

A CASE STUDY OF DHT FUNCTIONALITY:
RANGE QUERY ON DHT

A thesis submitted to the faculty of
San Francisco State University
In partial fulfillment of
The Requirements for
The Degree

Master of Science
in
Computer Science

by
Yoshihiro Tanaka
San Francisco, California
December, 2010

Copyright by
Yoshihiro Tanaka
2010

CERTIFICATION OF APPROVAL

I certify that I have read A CASE STUDY OF DHT FUNCTIONALITY: RANGE QUERY ON DHT by Yoshihiro Tanaka, and that in my opinion this work meets the criteria for approving a thesis submitted in partial fulfillment of the requirements for the degree: Master of Science in Computer Science at San Francisco State University.

James Wong

Professor of Computer Science

Marguerite C. Murphy

Professor of Computer Science

Arno Puder

Associate Professor of Computer Science

A CASE STUDY OF DHT FUNCTIONALITY:
RANGE QUERY ON DHT

Yoshihiro Tanaka

San Francisco State University

2010

Distributed Hash Table [1] (DHT) is a method to store and lookup data across distributed nodes. It provides a lookup service to locate a corresponding node for a given key using a hash function. DHT is scalable, fault-tolerant and widely used in P2P systems [1]. However, it does not inherently support range queries, since DHT is based on a hash function [2]. In this project, a method to perform range queries on DHT is introduced. To verify and evaluate this method, a simple Chord [3] based DHT system was implemented using Erlang/OTP [4, p. XIII] [4, p. 263] [5]. Experiments were performed using several machines. These experiments include measuring the performance under different conditions, comparison with a simple get method, memory usage of the range query method, and load balancing considerations. The experimental results show that the range query method outperforms exhaustive search using a simple get method; and the data load distribution was more balanced as the number of virtual nodes increases.

I certify that the Abstract is a correct representation of the content of this thesis.

Chair, Thesis Committee

Date

ACKNOWLEDGEMENT

First, I thank my family for allowing me to study for this long time regardless of our hardest time.

Next, I thank Prof. Wong for his help evaluating and writing the thesis, he took a lot of time reading, correcting, and suggesting for this project. In addition, I appreciate his generosity for accepting me as one of his students.

I also thank Prof. Murphy for allowing me to use machines for the development and evaluation of Harmonia, in addition to the valuable class material.

TABLE OF CONTENTS

List of Tables	ix
List of Figures	x
List of Listings	xii
List of Appendices	xv
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Related Work	1
1.3 Thesis Overview	2
2 THEORY AND TECHNICAL BACKGROUND	4
2.1 Distributed Hash Table	4
2.1.1 Mapping and Routing	5
2.1.2 Scalability and Fault Tolerance	9
2.2 Chord	10
2.2.1 Routing with Finger Table	10
2.2.2 Node Join and Stabilization	15
2.2.3 Successor List and Fault Tolerance	15
2.3 CAP Theorem	17
2.4 Erlang/OTP	18
2.4.1 Features of Erlang	18
2.4.2 Sample Erlang Application	19
2.4.3 Erlang/OTP	21
2.5 Background Summary	22
3 SYSTEM ARCHITECTURE	23
3.1 Design Overview	23

3.2	Functional Structure of Harmonia	24
3.2.1	Client I/F	24
3.2.2	Data Store Layer	25
3.2.3	Routing Layer	26
3.2.4	Auxiliary Functions	26
3.3	Routing Layer	28
3.3.1	API	28
3.3.2	Data Structure	28
3.3.3	Hash Function	29
3.3.4	Routing Method	30
3.3.5	Average Cost of Routing	31
3.4	Data Store Layer - Simple store/get	32
3.4.1	API	32
3.4.2	Data Structure	32
3.4.3	Algorithm	33
3.4.4	Parallel Store	34
3.5	Data Store Layer - Range Query	35
3.5.1	API	35
3.5.2	Data Structure	36
3.5.3	Algorithm	38
3.5.4	Example of Range Methods	38
3.6	Client Interface	41
3.6.1	API	41

3.7	System Architecture Summary	43
4	EVALUATION	44
4.1	Common Configuration	44
4.1.1	Hardware Configuration	44
4.1.2	Software Configuration	45
4.2	Verification Test	46
4.3	Comparison with Different Number of Nodes	46
4.3.1	Purpose	46
4.3.2	Specification	47
4.3.3	Metric	47
4.3.4	Configuration	47
4.3.5	Evaluation	49
4.4	Comparison with Different Number of Threads	51
4.4.1	Purpose	51
4.4.2	Specification	51
4.4.3	Metric	52
4.4.4	Configuration	52
4.4.5	Evaluation	52
4.5	Comparison with Different Range Query Conditions	54
4.5.1	Purpose	54
4.5.2	Specification	54
4.5.3	Metric	54
4.5.4	Configuration	54

4.5.5	Evaluation	55
4.6	Memory Usage	57
4.6.1	Purpose	57
4.6.2	Specification	57
4.6.3	Metric	58
4.6.4	Configuration	58
4.6.5	Evaluation	59
4.7	Load Balancing	61
4.7.1	Purpose	61
4.7.2	Specification	61
4.7.3	Metric	61
4.7.4	Configuration	61
4.7.5	Evaluation	62
4.8	Evaluation Summary	63
5	CONCLUSION	64
	BIBLIOGRAPHY	65
	APPENDICES	67

LIST OF TABLES

1	Finger Table Example	11
2	Definitions of Variables for Node with m-bit Identifiers	12
3	Finger Tables of All Nodes for Figure5	13
4	Successor Lists Sample	16
5	Data Maintained in Data Storage Layer (Simple store/get)	33
6	Data Maintained in Data Storage Layer (Range Query)	37
7	Table Configuration Example	39
8	Index Table Example	39
9	Data Table Example	40
10	Machine Specifications	45
11	OS and Erlang VM version	45
12	Harmonia Configuration	45
13	Number of Virtual Nodes for Each Machine (Different Number of Nodes) . . .	48
14	Number of Virtual Nodes for Each Machine (Different Number of Threads) . .	52
15	Number of Threads Allocated for Each Machine	52
16	Number of Virtual Nodes for Each Machine (Different Query Conditions) . . .	55
17	Data for store function (Short data configuration)	57
18	Data for rstore function (Short data configuration)	57
19	Data for store function (Long data configuration)	57
20	Data for rstore function (Long data configuration)	57
21	Main Data Type Size of Erlang	60
22	Number of Virtual Nodes for Each Machine	62
23	Process list of Harmonia	82
24	Environment variables of Harmonia	85

LIST OF FIGURES

1	Address Space of Hash Function in DHT	5
2	Mapping of Address Space in DHT	6
3	Address Space and Partition of DHT	7
4	Lookup Service of DHT	8
5	8-bit Identifier Circle of Chord with 7 Virtual Nodes	10
6	Routing Process of Chord	14
7	Node Join of Stabilization Process	15
8	Data Replication Using a Successor List	16
9	Generic Process Pattern	21
10	Harmonia System Architecture	24
11	Functional Structure of Harmonia	25
12	Routing Methods	30
13	Parallel Store Sequence	34
14	Store and Range Query	36
15	Average Latency of Harmonia APIs for a different Number of Virtual Nodes . .	49
16	Comparison with Different Number of Threads	53
17	Comparison with Different Range Query Conditions	55
18	Memory Usage with Theory and Real Case	59
19	Load Balancing for different Number of Virtual Nodes per Machine	62
20	Key Lookup using finger table	67
21	Pseudo code for Stabilization - create and join	68
22	Pseudo code for Stabilization - stabilize and notify	68
23	Pseudo code for Stabilization - fix_fingers and check_predecessor	69
24	Pseudo code for Stabilization - stabilize and notify	69

25	Supervisor tree of Harmonia	81
26	Object relationship of Harmonia	83
27	Directory structure of Harmonia	84

LIST OF LISTINGS

1	Routing Information Data Structure	28
2	Routing Information: example from log file	29
3	Hash function of Harmonia	29
4	Example User Interface	42
5	Sample Erlang program: start_ring function	70
6	Sample Erlang program: loop function	71
7	Sample Erlang program: user API	73
8	Sample Erlang program: output example	74
9	Sample Erlang program	75
10	Query condition of range query	78
11	Example command line execution of range query	78
12	Start up command: create	86
13	Start up command: join	86
14	Makefile	87
15	Include file	89
16	harmonia.app	91
17	harmonia.erl	92
18	hm_cache.erl	93
19	hm_cache_mgr.erl	95
20	hm_cli.erl	99
21	hm_config.erl	103
22	hm_config_if.erl	106
23	hm_ds.erl	109
24	hm_edge.erl	123

25	hm_event_mgr.erl	125
26	hm_log_h_file.erl	128
27	hm_log_h_term.erl	131
28	hm_misc.erl	134
29	hm_name_server.erl	138
30	hm_qp.erl	141
31	hm_router.erl	146
32	hm_stabilizer.erl	156
33	hm_sup.erl	162
34	hm_table.erl	165
35	Makefile	170
36	hm_cache_test.erl	171
37	hm_cli_test.erl	172
38	hm_qp_test.erl	185
39	hm_test.erl	186
40	Test Results:Comparison with different number of nodes:10 nodes	187
41	Test Results:Comparison with different number of nodes:17 nodes	189
42	Test Results:Comparison with different number of nodes:33 nodes	191
43	Test Results:Comparison with different number of nodes:65 nodes	194
44	Test Results:Comparison with different number of nodes:105 nodes	197
45	Test Results:Comparison with different number of threads	202
46	Test Results:Comparison with different query conditions	209
47	Test Results:Memory Usage:store function with short data	213

48	Test Results:Memory Usage:store function with long data	238
49	Test Results:Memory Usage:rstore function with short data	261
50	Test Results:Memory Usage:rstore function with long data	287
51	Test Results:Load balancing with 5 nodes per machine	314
52	Test Results:Load balancing with 10 nodes per machine	315
53	Test Results:Load balancing with 20 nodes per machine	317
54	Test Results:Load balancing with 40 nodes per machine	322
55	Test Results:Comparison with different number of nodes:plot data	331
56	Test Results:Comparison with different number of nodes:gnuplot script	331
57	Test Results:Comparison with different number of threads:plot data	331
58	Test Results:Comparison with different number of threads:gnuplot script	332
59	Test Results:Comparison with different query conditions:plot data	332
60	Test Results:Comparison with different query conditions:gnuplot script	332
61	Test Results:Memory Usage of Theory:plot data	333
62	Test Results:Memory Usage of Real:plot data	333
63	Test Results:Memory Usage with theory and real case:gnuplot script	333
64	Test Results:Load Balancing:plot data	334
65	Test Results:Load Balancing:gnuplot script	334

LIST OF APPENDICES

A	Chord Algorithms	67
B	Sample Erlang Application	70
C	Range Query Specification and Example	78
D	Implementation of Harmonia	80
D.1	Supervisor Tree of Harmonia	80
D.2	Object Relationship of Harmonia	81
D.3	Directory Structure and Startup Configuration	84
D.3.1	Directory structure	84
D.3.2	Application Resource File	84
D.3.3	Startup of Harmonia	86
E	Harmonia Source Files	87
F	Harmonia Test Codes	170
G	Test Results	187
G.1	Comparison with Different Number of Nodes	187
G.2	Comparison with Different Number of Threads	202
G.3	Comparison with Different Range Query Conditions	209
G.4	Memory Usage with Real Case	213
G.5	Load Balancing	314
H	Plot Data	331
H.1	Comparison with Different Number of Nodes	331
H.1.1	Plot Data	331
H.1.2	Gnuplot Script	331
H.2	Comparison with Different Number of Threads	331
H.2.1	Plot Data	331

H.2.2	Gnuplot Script	332
H.3	Comparison with Different Range Query Conditions	332
H.3.1	Plot Data	332
H.3.2	Gnuplot Script	332
H.4	Memory Usage with Theory and Real Case	333
H.4.1	Plot Data	333
H.4.2	Gnuplot Script	333
H.5	Load Balancing	334
H.5.1	Plot Data	334
H.5.2	Gnuplot Script	334

1 INTRODUCTION

1.1 Introduction

Distributed computing is currently one of the main forms of computing. Most systems work on the premise of the use of networks. The importance of distributed computing is growing partly because the amount of data to be processed is growing faster than the ability of hardware to process it [6]. Distributed Hash Table (DHT) is one possible solution to this problem. DHT is a technology to manage a hash table on distributed nodes in a network [7]. DHT originates from peer-to-peer systems such as Napster, Gnutella and Freenet [8, pp. 9-23]. It has been widely used and researched [9] because it is scalable and fault-tolerant [8, pp. 84-85].

While DHT has a number of good properties, support of range queries on DHT is not necessarily straightforward because “*DHTs are constrained to single-key, exact-match queries. The randomized property of DHT indexing works against the range query*” [10, p. 174]. However, the demand for range query on DHT exists and it has been researched as described in the next section.

The goal of this project is to provide a case study of another range query method on DHT. In order to evaluate the method, a simple DHT system was implemented and range query functionality was added to it. The system was named as *Harmonia*, and was tested and evaluated using several machines. The experimental results indicate that range query method works correctly and effectively on top of DHT.

1.2 Related Work

A range query is “*a query that describes a region(range) in space and asks for a subset of object points that belong to the region*” [10, p. 173]. While it is not straightforward to perform

range query on DHT, much research has explored the topic and followings are a part of it.

Gupta et al. [11] used Locality Sensitive Hashing to map similar data to the same nodes and enabled range query with high probability. Andrzejak and Zhichen [12] used Space Filling Curve as a hash function that fills d -dimensional cartesian space to support multi attribute range query. Ratnasamy et al. [13] used Prefix Hash Trees to build an index. Felber et al. [14] used XML to build a structured index. Ratnasamy et al. and Felber et al. took the same approach in terms of building index on DHT that is stored across nodes so that a query can traverse nodes to search index.

The range query method of Harmonia also uses index to perform a range query. However, it builds an index on a specific node, thus no node traversal is necessary to search an index. In addition, it provides a flexible multi attribute range query.

The research survey of range query on DHT is well summarized in RFC4981 [9] and Buford et al. [10].

1.3 Thesis Overview

The following are brief descriptions of each chapter. Chapter 2 describes technical background about DHT, Chord, CAP theorem and Erlang/OTP [5]. The routing algorithm and replication mechanism of Chord is presented with basic DHT concepts. Next, the limitation of distributed storage is described with CAP theorem [15]. Finally, overview of Erlang/OTP is presented. Chapter 3 describes the system architecture and implementation of Harmonia. The range query algorithm in chapter 3 and Harmonia described also in appendices are designed and implemented in this project from scratch. Chapter 4 describes evaluations of range query using Harmonia. Chapter 5 describes the conclusion of the project and future work. Appendices include Chord algorithm, sample Erlang application, implementation of Harmonia,

program source files, and test program source files with results. The contents of appendices are the results of this project. Harmonia was implemented in this project from scratch using DHT concepts, algorithms introduced in Chapter 2, and range query design and algorithm introduced in Chapter 3.

2 THEORY AND TECHNICAL BACKGROUND

The following chapter describes theory and technical background of this project. First, features of DHT will be described with Chord. Chord was chosen as a DHT algorithm in this project to implement Harmonia. Second, requirements and constraints of distributed storage systems are described along with CAP theorem. Third, the main features and advantages of Erlang/OTP are described.

2.1 Distributed Hash Table

This section describes features of DHT with Chord. The fundamental function of DHT is a lookup service to locate a correspondent node in a network for a given key. As an interface, it provides the following API:

```
lookup(key)
```

The *lookup(key)* function returns a node ID (e.g., IP address) which is in charge of the *key* passed as an argument. Using the *lookup(key)* function, APIs to store or get key-value pair data can be provided as following:

```
store(key, value)
get(key)
```

These functions rely on the *lookup(key)* function. *store(key, value)* function first retrieves a node ID for the *key* using the *lookup(key)* function. Then it can store a *key-value* pair to the node. The *get(key)* function first retrieves node IDs for the *key* using the *lookup(key)* function. Then it can retrieve *value* which corresponds to the *key* from the node.

The lookup service mechanism is described in the following section.

2.1.1 Mapping and Routing

This section describes the mechanism of lookup DHT service using Chord example. Hash functions usually return a certain range of integer value for a given key. This range of integer value is called *an address space* of a hash function. For example, SHA-1 [16] hash function has an 160 bits address space which starts from 0 to $2^{160} - 1$.

Then node IDs that participate in DHT are mapped to an address space of hash function. For example when the node ID is an IP address 130.212.4.32, hash function is $H()$, and hashed value is 37, it can be expressed as following:

$$H(130.212.4.32) = 37$$

The hashed value, 37 in this example, is called *virtual node ID*. Virtual node ID is actually mapped into the address space of hash function $H()$. When key-value pair is stored using $store(key, value)$ function, the *key* is also mapped to the same address space of the hash function. For example when the key is a string Key1, hash function is $H()$, and hashed value is 2038, it can be expressed as following:

$$H(Key1) = 2038$$

Figure 1 shows an example of an address space from 0 to $2^{12} - 1$. In this example, IP address is used for node ID. Keys are also mapped into the same address space. Both node IDs and Keys

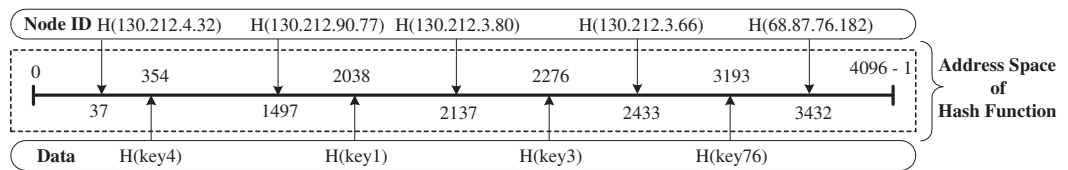


Figure 1: Address Space of hash function in DHT: Both node IDs and data are hashed and mapped to the same address space

are hashed with the same hash function $H()$ and mapped to the address space. Figure 2 shows the mapping between virtual nodes and real nodes for the example of Figure 1. An address

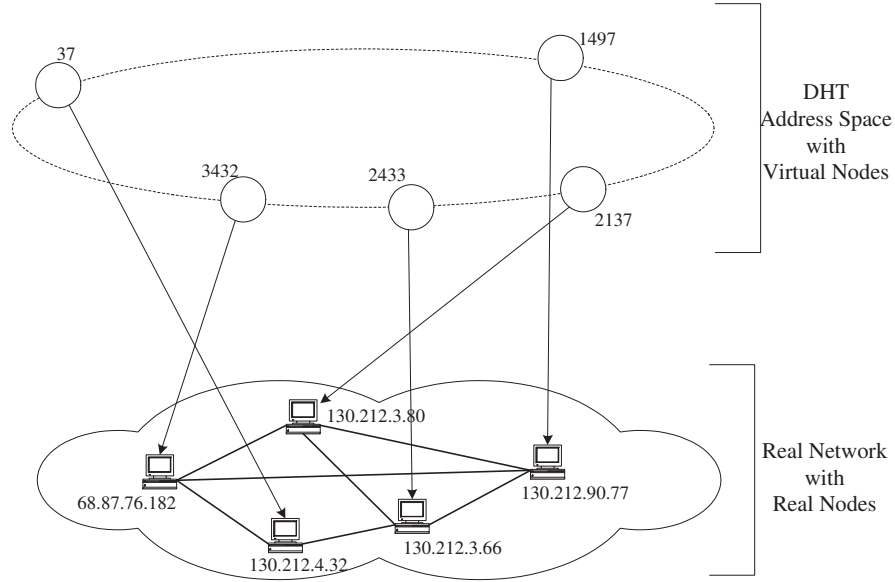


Figure 2: Mapping of Address Space in DHT

space can be considered a circle which is ordered clock-wise and modulo 2^m , where m denotes the number of bits for the range of address space. In order to assign a key to a node in network, DHT assigns contiguous portions of address space to each node, called *partition*. The way to assign a key to a partition of node depends on each DHT algorithm. In case of Chord, a hashed value of a key is assigned to the first node whose virtual node ID is equal to or follows the value in the address space. For example, hashed value of key1, $H(key1) = 2038$ is assigned to virtual node ID 2137 in the address space in Figure 1.

Figure 3 shows an address space from 0 to $2^{12} - 1$ and partitions for 5 virtual nodes. The figure shows 5 partitions of virtual nodes. In this example, the partition of virtual node 2137 is from 1498 to 2137. key1 , $H(key1) = 2038$, is assigned to the partition of virtual node 2137. In the same way $H(key4) = 354$ belongs to virtual node 1497, and $H(key76) = 3193$ belongs to virtual node 3432.

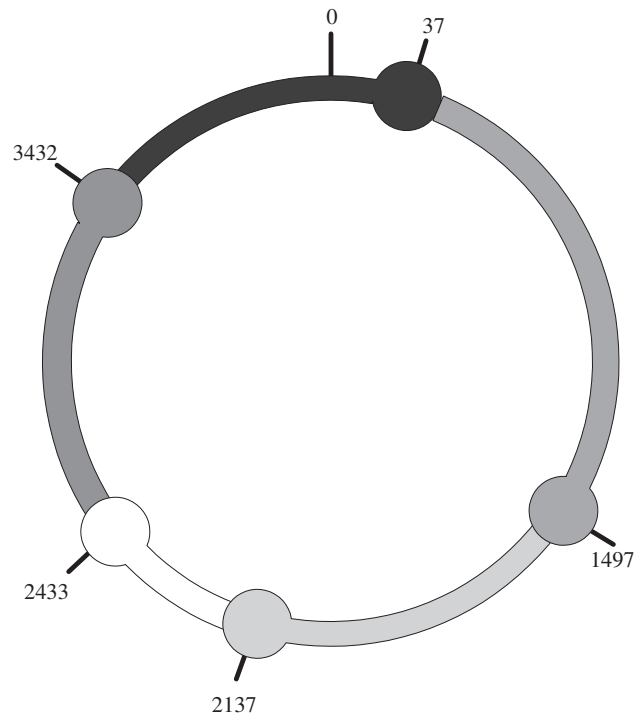


Figure 3: Address space and partition of DHT

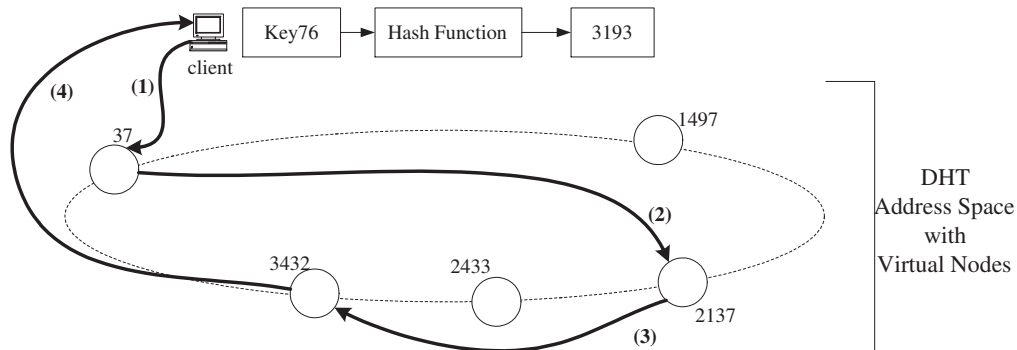


Figure 4: Lookup service of DHT: (1) When a user wants to know which node has data for the Key76, it asks arbitrary DHT node with the *lookup* function. (2) Virtual node 37 checks the hashed value of 3193, and forwards it to vnode 2137 since it does not have data for 3193. (3) Virtual node 2137 checks if it has data for 3193. It does not have data for 3193 and forwards the query to virtual node 3432. (4) Virtual node 3432 knows it has data for 3193, and returns the node ID to the originator.

Figure 4 describes a process enabling a client to find a correspondent node for a given key. It shows mapping of DHT address space with virtual nodes in a network. In Figure 4, the query is routed until the target node is found. Routing is necessary because each node has only a small portion of entire DHT network information. Routing is a core functionality of DHT [8, p. 88]. There are a variety of routing algorithms such as Chord [3], CAN [17], Pastry [18], Tapestry [19], and Kademlia [20]. However, common procedure is that each node maintains limited information of the whole system and refers to it when it receives a request for a correspondent node ID of a certain key. If it does not hold target information, it forwards the request to other node according to its DHT routing algorithm. This process continues recursively until target node ID is found [8, p. 88].

2.1.2 Scalability and Fault Tolerance

Scalability is an important characteristic of DHT. The path length, the number of nodes that need to be traversed during lookup service, can be considered the cost of a lookup service. The path length grows as the log of the total virtual nodes in DHT. Therefore, even if a large number of nodes participate in a DHT network, the lookup service is still feasible [8, pp. 80-85].

DHT is fault tolerant and has no single point of failure because it is designed to handle node failures and it has no central server. To handle node failure, routing and lookup procedures are typically designed to use alternative routes to find a target node ID when a failed node is encountered [8, p. 90]. Routing information is designed to be maintained periodically to reflect the latest status of the DHT network.

2.2 Chord

Chord [3] is one of the DHT algorithms. It was implemented in this project to evaluate the range query method. In this section the basic concept and algorithm of Chord is introduced.

2.2.1 Routing with Finger Table

This section describes the routing method of Chord. Chord uses an address space of a hash function. For example, if a hash function returns integers from 0 to $2^{12} - 1$, this range is the address space of the hash function. This address space of hash function is also called *identifier circle* in Stoica et al. [3]. Then a node ID (e.g., IP address) is hashed to an integer value from 0 to $2^m - 1$ where m depends on hash functions. This integer value, called *virtual node ID*, is mapped to an identifier circle.

Figure 5 shows 8-bit identifier circle of Chord. Here, 7 nodes are mapped to the 0 to $2^8 - 1$ space. In this example, the virtual node ID is expressed as N and a number, such as N22 for convenience. The number portion indicates the virtual node ID.

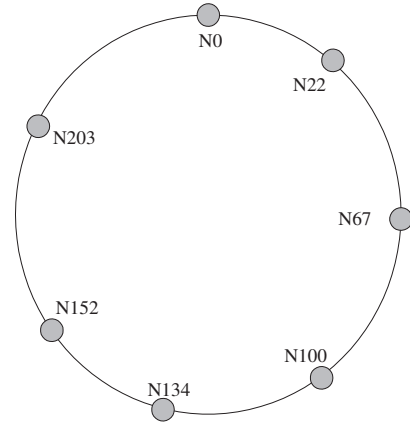


Figure 5: 8-bit Identifier Circle of Chord with 7 Virtual Nodes

In order to map data to an identifier circle, the key of data, k is hashed to an integer value. This integer value is then assigned to the first virtual node whose value is equal to or follows the value of k in the identifier

circle. This node is called *successor node* of key k . For example, in Figure 5, when k is $\{k \mid 22 < k \leq 67\}$, the successor node is N67. This contiguous portion of identifier circle for

which a certain node is responsible is called *partition*.

Routing is a process to find a successor node for a given key. For routing, each node maintains a table with m entries for m -bit identifier circle. This is called a *finger table*. The i th entry in a finger table of node n is node ID s that succeeds n by at least 2^{i-1} on the identifier circle [3]:

$$s = \text{successor}(n + 2^{i-1}) \bmod 2^m, \text{ where } 1 \leq i \leq m$$

For example, table 1 depicts the finger table of node N22 as seen in Figure 5. The actual finger table also includes the real node ID, such as the IP address and a port number as long as a virtual node ID. Note that the left 2 columns are shown only as an indicator of n -th entry and not included in the actual finger table:

$22 + 1$	$(22 + 2^{1-1}) \bmod 2^8 = 23$	N67
$22 + 2$	$(22 + 2^{2-1}) \bmod 2^8 = 24$	N67
$22 + 4$	$(22 + 2^{3-1}) \bmod 2^8 = 26$	N67
$22 + 8$	$(22 + 2^{4-1}) \bmod 2^8 = 30$	N67
$22 + 16$	$(22 + 2^{5-1}) \bmod 2^8 = 38$	N67
$22 + 32$	$(22 + 2^{6-1}) \bmod 2^8 = 54$	N67
$22 + 64$	$(22 + 2^{7-1}) \bmod 2^8 = 86$	N100
$22 + 128$	$(22 + 2^{8-1}) \bmod 2^8 = 150$	N152

Table 1: Finger Table of N22

There are two important characteristics for the finger table [3].

- Each node maintains a small portion of information for the entire system and knows more about nodes closely following it on the identifier circle than about nodes farther away.
- A node's finger table generally does not contain enough information to directly determine the successor of an arbitrary key k .

In addition to a finger table, each node maintains *predecessor*. Predecessor is an immediate previous node in the identifier circle. *Successor* is an immediate next node in identifier circle. Note that successor is the first entry of finger table and is denoted as *finger[1]*. Table 2 shows a definition of variables for node x , using m -bit identifiers from Stoica et al. [3].

Notation	Definition
$finger[k]$ in finger table	first node on circle that succeeds $(x + 2^{k-1}) \bmod 2^m, 1 \leq k \leq m$
<i>successor</i>	$finger[1]$; the next node on the identifier circle
<i>predecessor</i>	the previous node on the identifier circle

Table 2: Definition of variables for node x , using m -bit identifiers

To find a successor node of key k using a finger table, node x first checks if $k = \{k \mid x < k \leq successor\}$. If it is true, x 's successor is returned. If it is false, the finger table is checked in reverse order if $k = \{k \mid x < finger[i] < k\}$. For example, if $m=8$, a finger table is checked from $finger[8]$ down to $finger[1]$. This means the request is forwarded to the *closest predecessor* of k in the finger table. If no entry matches, it returns itself as a successor. The pseudo code of this algorithm is described in Stoica et al. [3] and Figure 20 in Appendix A. This scheme leads to the following important property of Chord [3].

Since each node has finger entries at a power of two intervals around the identifier circle, each node can forward a query at least halfway along the remaining distance between the node and the target identifier. From this intuition follows a theorem:

Theorem IV.2: With high probability, the number of nodes that must be contacted to find a successor in a N -node network is $O(\log N)$.

The following are scenarios for routing in Figure 5 with finger table Table 3.

	N0	N22	N67	N100	N134	N152	N203
$(n + 2^{1-1}) \bmod 2^8$	N22	N67	N100	N134	N152	N203	N0
$(n + 2^{2-1}) \bmod 2^8$	N22	N67	N100	N134	N152	N203	N0
$(n + 2^{3-1}) \bmod 2^8$	N22	N67	N100	N134	N152	N203	N0
$(n + 2^{4-1}) \bmod 2^8$	N22	N67	N100	N134	N152	N203	N0
$(n + 2^{5-1}) \bmod 2^8$	N22	N67	N100	N134	N152	N203	N0
$(n + 2^{6-1}) \bmod 2^8$	N67	N67	N100	N134	N203	N203	N0
$(n + 2^{7-1}) \bmod 2^8$	N67	N100	N134	N203	N203	N0	N22
$(n + 2^{8-1}) \bmod 2^8$	N134	N152	N203	N0	N22	N67	N100

Table 3: Finger Tables of All Nodes for Figure5

Scenario 1: Requested id falls between n and n 's successor

N67 receives a request for id 85. N67.find_successor is called and N67 checks if 85 falls between N67 and N67's successor. This is true, so N67 returns its successor N100.

Scenario 2: Requested id falls between n 's seventh and last entry of the finger table

N67 receives a request for id 152. N67.find_successor is called and N67 checks if 152 falls between N67 and N67's successor. This is false, so the function closest_preceding_node is called. It checks if $finger[8]=N203$ falls between N67 and 152, which is false. Therefore, It continues to check if $finger[7]=N134$ falls between N67 and 152, which is true. Therefore, it returns N134. N134.find_successor is called and it checks if 152 falls between N134 and N134's successor, which is true, so it returns N152.

Scenario 3: Requested id falls beyond n 's last entry of the finger table

N67 receives a request for id 230. N67.find_successor is called and N67 checks if 230 falls between N67 and N67's successor, which is false, so the function closest_preceding_node is called. It checks if $finger[8]=N203$ falls between N67 and 230, which is true. It returns N203.

N203.find_successor is called and it checks if 230 falls between N203 and N203's successor, which is true, so it returns its successor N0.

Scenario 4: Requested *id* falls between *n*'s predecessor and itself

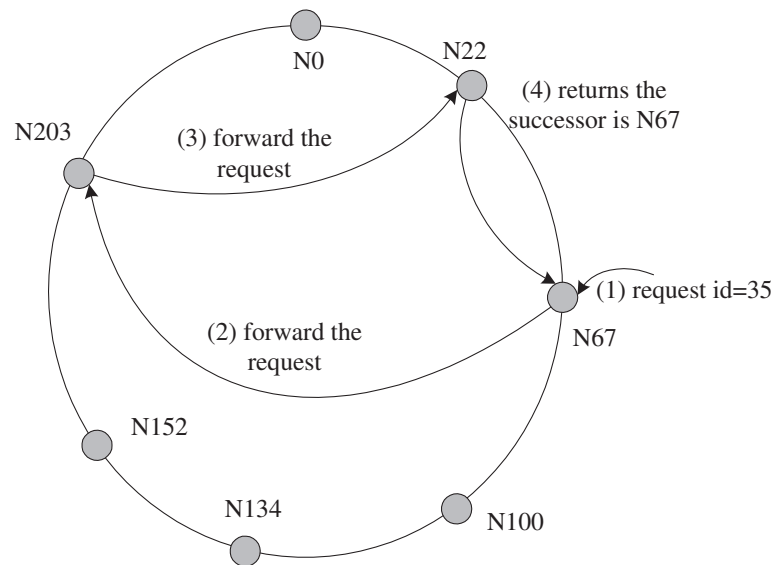


Figure 6: Routing Process of Chord: Scenario 4

N67 receives a request for *id* 35. N67.find_successor is called and N67 checks if 35 falls between N67 and N67's successor, which is false, so the function *closest_preceding_node* is called. It checks if *finger*[8]=N203 falls between N67 and 35, which is true. Therefore, it returns N203. N203.find_successor is called and it checks if 35 falls between N203 and N203's successor, which is false, so the function *closest_preceding_node* is called. It checks if *finger*[8]=N100 falls between N203 and 35, which is false, so it continues to check if *finger*[7]=N22 falls between N203 and 35, which is true. It returns N22. N22 receives request for *id* 35. N22.find_successor is called and it checks if 35 falls between N22 and N22's successor, which is true, so it returns N67. This process is shown in Figure 6.

2.2.2 Node Join and Stabilization

Stabilization is how a node creates or joins an identifier circle, and builds predecessor and a finger table. Figure 7 shows the process of node join. It is based on the figure in Stoica et al. [3]. At first, node N22 has successor N67, and node N67 has predecessor N22. Then node N36 joins. N36 gains a successor using *join* algorithm in Figure 21 of Appendix A. N36 notifies itself to N67 with *stabilize* and *notify* algorithms in Figure 22 of appendix A. In *notify* algorithm N67 checks if $N22 < N36 < N67$, which is true, so N67 sets N36 as its predecessor. N22 calls *stabilize* algorithm and finds N36 is now its successor and notifies N36 about N22. N36's predecessor is still *nil*, so N36 sets N22 as its predecessor. The pseudo codes are shown in Figure 21, 22, 23 of Appendix A from Stoica et al. [3].

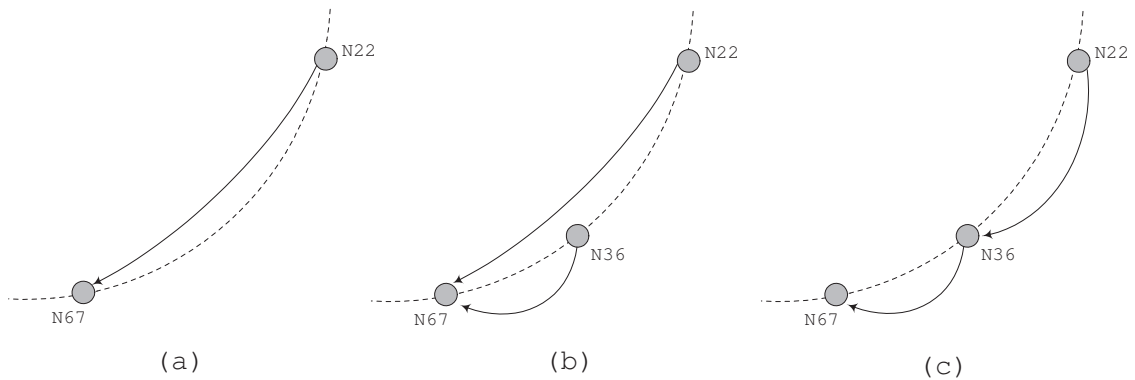


Figure 7: Node join of Stabilization process. (a) N22 has a successor as N67, and N67 has N22 as a predecessor. (b) N36 joins and notifies about it to N67. N36 has successor N67, and N67's predecessor is updated to N36. (c) N22 notifies N36. N22's successor is updated to N36 and N36's predecessor is set to N22.

2.2.3 Successor List and Fault Tolerance

The correctness of the Chord protocol relies on the fact that each node knows its successor. However this invariants can be compromised if nodes fail [3]. To increase robustness, each

node needs to maintain a *successor list* of size r where r is a number large enough to be robust, containing the node's first r successors [3].

A successor list is a list of consecutive nodes that follows a node's successor. Examples of successor lists are shown in Table 4. Using a successor list, a node can find its successor with high probability even if its successors fail simultaneously.

N0	N22	N67	N100	N134	N152	N203
N22	N67	N100	N134	N152	N203	N0
N67	N100	N134	N152	N203	N0	N22
N100	N134	N152	N203	N0	N22	N67
N134	N152	N203	N0	N22	N67	N100

Table 4: Successor List of All Nodes for Figure 5 with size $r=4$

Instructions on how to build a successor list is described in Stoica et al. [3]. The algorithm copies the successor list of its successor, then removes the last element of the successor list, and prepends it to its own successor. The pseudo code is shown in Figure 24 of Appendix A. The successor list mechanism also helps higher-layer software to replicate data [3]. Using a

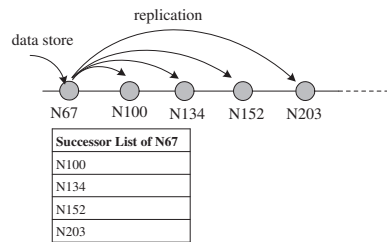


Figure 8: Data Replication using a Successor List

successor list, a typical application such as key value store can replicate data to successor nodes so that if the successor of a key fails, data is still available in other nodes of the successor.

2.3 CAP Theorem

This section describes the system limits of this project. For data storage purposes, traditional systems use relational databases in many cases. But in some cases, the use of relational databases is not necessarily suitable. Those cases include systems that require high availability while they do not need complex queries such as join operations. For example, Bala et al. [21] describes this situation as following:

For example, customers should be able to view and add items to their shopping cart even if disks are failing, network routes are flapping, or data centers are being destroyed by tornados. Therefore, the service responsible for managing shopping carts requires that it can always write to and read from its data store, and that its data be available across multiple data centers.

There are many services on Amazon's platform that only need primary-key access to a data store.

Since relational databases do not meet these requirements, systems that enable availability without strict consistency are designed and developed.

According to CAP theorem [15], three properties of consistency (all nodes see the same data), availability (node failure does not interrupt system service) and partition-tolerance (arbitrary message loss does not affect system) are impossible to achieve at the same time in a distributed environment. Therefore systems such as Amazon's Dynamo provide weaker consistency for availability and partition-tolerance [21].

The storage system in this project, Harmonia, also targets the same requirements and does not provide the mechanism for consistency such as concurrency control.

2.4 Erlang/OTP

This section describes basic concepts of Erlang/OTP. The system in this project was developed using Erlang/OTP because it has many advantages for developing distributed system.

2.4.1 Features of Erlang

Scalable and Efficient Concurrency Processes of Erlang is independent of underlying operating system. They are created, scheduled, and handled in the VM. Erlang processes communicate with each other through message passing. Regardless of the number of concurrent processes in a system, exchanging messages within the system takes microseconds [4, p. 6].

Message Passing [4, pp. 92-99] Processes of Erlang communicate by message passing. This is true for OTP behavior too. Sending a message is asynchronous and the sender does not wait for messages to be sent. To receive messages, each Erlang process has its own mail box that messages are stored to. A process takes messages out of a mail box one by one. Even if messages are sent from many processes asynchronously, the receiver does not need to take care of concurrency, since messages are already serialized in its mail box. For this reason, Erlang is suitable for writing concurrent programs.

Robustness Erlang has a number of mechanisms to make a system robust. They include process link and monitoring and use of OTP middleware. Processes linked together can be informed if one of them crashes and can choose what to do. OTP provides a set of generic process behaviors such as servers, finite state machines, and event handlers. These worker process behaviors have built-in robustness. Supervisor is another process behavior whose only task is to monitor and handle process start and termination. Using these built-in mechanisms, building fault tolerant systems with Erlang is efficient [4, p. 7].

Distributed Computation Erlang has a built-in distributed environment, language syntax and semantics that work transparently across nodes. Message passing and remote procedure call is incorporated in language in the distributed environment. Therefore, building distributed systems with Erlang is efficient [4, p. 7].

2.4.2 Sample Erlang Application

In order to illustrate the advantages of Erlang to write distributed software, a small application program was written. The requirements of the program are following:

“Create N processes in a ring. Send a message round the ring M times so that a total of $N * M$ messages get sent” [5, p. 150]. Furthermore each process is created on a node chosen randomly from a given node list, thus the processes run across network, messages are sent across network too.

To solve this problem, only about 70 lines of code were necessary in Erlang (without comments). The entire list is shown in appendix B. The APIs that make the program concise are described below, but these are only a part of the infrastructure provided by Erlang to support concurrent and distributed programming. In Erlang, one can create a process on an arbitrary node using *spawn* function:

```
spawn(Node, Module, Function, Args)
```

It returns the process ID of a new process started by the application of *Module:Function* to arguments of *Args* on *Node*.

To send a message to a process, *send* function is used:

```
send(Name, Msg)
```

It sends a message *Msg* to the process of name *Name*. There is no need to open or close socket.

To receive a message, *receive* statement is used:

```

loop() ->
  receive
    {m1, Message1} ->
      processes for Message1
      loop();
    {m2, Message2} ->
      processes for Message2
      loop();
    ...
  end.

```

In this example, a process waits for a message in the *loop* user function. When a message arrives, it is *pattern matched* with the messages specified in *receive* statement. For example, if a message has a form of {m1, Variables}, it matches to {m1, Message1}. After processing the message, loop function is called again and the process waits for messages.

2.4.3 Erlang/OTP

The system in this project was developed using Erlang/OTP. OTP behaviors in Erlang are a formalization of process design patterns [4, p. 263]. It does all of the generic process work and error handling, and programmers only need to write specific code as callback functions. The generic process structures include starting a process, receiving messages, sending replies, maintaining process state, and termination of a process. All these behaviors are common even if each process has different tasks to perform. These generic processes are provided by OTP.

These structures are common to all processes, so OTP handles this generic part. On the other hand, specific parts include how to initialize a process, which messages to receive and reply, which data to maintain as process state, and how to terminate process. These specific parts are left to programmers. The idea is to split the code in two parts. The generic part is called generic behavior and provided by OTP libraries, while specific part is called callback modules and is implemented by programmers. There are five generic behaviors provided by OTP:

- Generic Servers is client/server model behavior
- Generic Finite State Machines is finite state machine model behavior
- Generic Event Handler / Manager is used for event handling

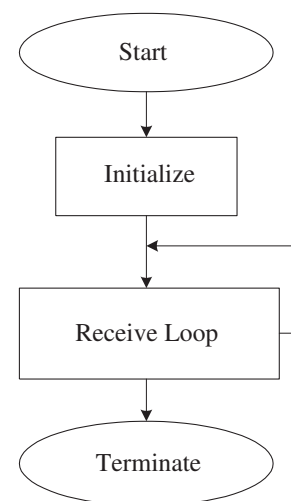


Figure 9: Generic Process Pattern

- Supervisor is used for fault-tolerant process management
- Application is used to encapsulate resources and functionality of packages

There are several advantages in using OTP. Error and special cases are handled by well-tested libraries and it reduces costs for development. It forces common programming style and enhances code readability. However, the learning curve is steep and performance may be affected by using OTP because many are working behind the scene.

2.5 Background Summary

This chapter described theory and technical background of this project. First, features and mechanisms of DHT was described with Chord. Second, requirements and constraints of distributed storage systems were described along with CAP theorem. Third, the main features and advantages of Erlang/OTP were described. In the next chapter, the system architecture of the Chord based DHT system with range query functionality is introduced.

3 SYSTEM ARCHITECTURE

This chapter describes system architecture of Harmonia, a DHT storage system with range query that was developed in this project from scratch. This includes the implementation and evaluations in appendices. The detail of each component, as well as the relationship between them is described. Along with Harmonia, other possible designs are also discussed.

3.1 Design Overview

The design goal of Harmonia is to build a flexible range query on DHT. Therefore DHT and range query are main functions.

The entire Harmonia network consists of DHT nodes and works by those DHT nodes communicating with each other on a network. A DHT node consists of a routing layer, a data storage layer, client interface, and other auxiliary components. Harmonia runs on Erlang distributed environment and many DHT nodes can run on one physical machine. For example, running five DHT nodes on each machine, a total of 20 nodes can run using 4 machines.

Routing layer provides a lookup service. It tells a responsible node for a given key. As a routing protocol, Chord was chosen in Harmonia. Since routing layer works with Chord protocol, joins and leaves of nodes are stabilized by the protocol itself dynamically. Routing layer also provides a successor list maintenance. This helps the routing to be robust against node failures, and upper layer can use it for replication.

Data store layer provides a simple key-value storage service as well as a range query service. This layer relies on lookup service of the routing layer. Harmonia enables range query function by building index. The detail mechanism of range query is described later in this chapter.

Client interface is provided on DHT nodes. Therefore, a client needs to connect at least

one DHT node to use the data store service of Harmonia. Figure 10 shows system architecture of Harmonia.

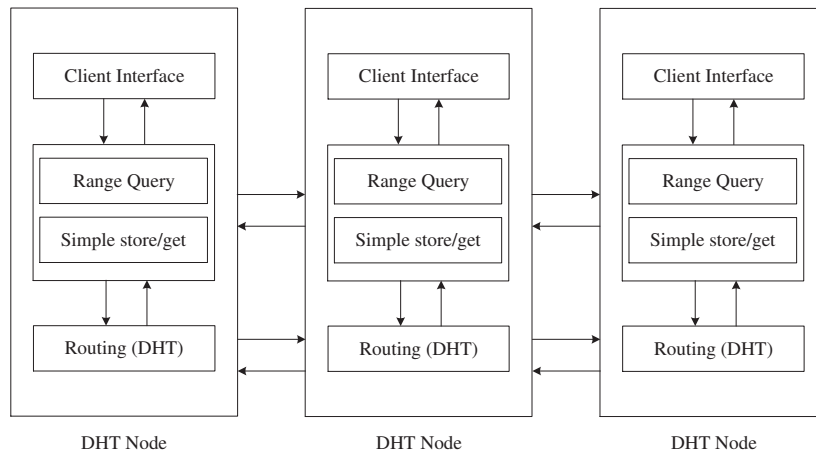


Figure 10: Harmonia System Architecture

3.2 Functional Structure of Harmonia

In this section Harmonia's functional structure is explained. Harmonia works as DHT nodes distributed across a network. Each DHT node has a similar structure. Main functions include a routing layer, a data store layer, and a client interface. In addition, there are other auxiliary components. Harmonia's DHT node consists of about 10 processes, each providing their own function. Some functions are provided as library modules. The functional structure of Harmonia node is shown in Figure 11. Each function is described below.

3.2.1 Client I/F

Client I/F provides a user interface for data storage and administrative commands. It is provided as a library module on DHT node, so clients need to connect at least one DHT node.

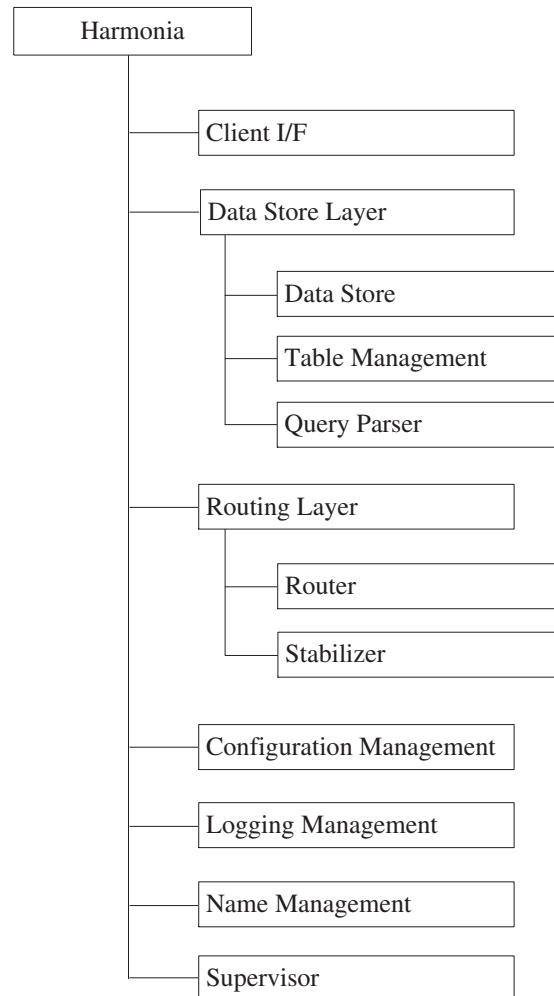


Figure 11: Functional Structure of Harmonia

3.2.2 Data Store Layer

Storage layer consists of a data store function, a table management function, and a query parser function. Data store function and table management function are provided as processes, and query parser as a library module. Data store function provides APIs for range query and simple store/get. It also supports caching for simple store/get functions, however this is an

experimental function and not a main topic in this project. Table management function creates and maintains table configuration needed for range query. Table is created in advance to define key fields to specify the range of values for specific fields. Data store function retrieves a table configuration using this function during range query. Query parser function is part of the range query function. It parses range query condition and converts it to the specific data storage native form. The data storage currently supported is ETS (Erlang Term Storage) memory storage.

3.2.3 Routing Layer

Routing layer supports Chord protocol and provides an upper-layer with lookup service. It consists of two processes, a router process and a stabilizer process. The router process maintains routing information such as a finger list and a successor list, and provides a lookup service. The stabilizer process calls stabilization functions periodically and keeps routing information up-to-date. The node join and leave is thus detected and stabilized dynamically. A successor list is also maintained by these processes so that the routing is robust against node failure, and upper layers can use it for replication.

3.2.4 Auxiliary Functions

There are auxiliary functions of DHT node other than main functions. These functions help to organize a DHT node.

Configuration Management Function The configuration management function makes configuration parameters available to other processes. It maintains configuration parameters and provides accessor/mutator methods.

Logging Management Function The logging management function outputs logging information to a file or a console. Using event handler and event manager OTP behavior, it provides an interface to start or stop logging.

Name Management Function The name management function manages Harmonia DHT node names. In order to join a Harmonia network, a node has to know at least one other node's name to ask beforehand, unless the node is the first node to create a Harmonia network. The node that is supposed to be known as default is called *bootstrap node*. In Stoica et al. [3], it is described as “*any known Chord node, or n.create() to create a new Chord network*”. According to Buford et al. [10, p. 155], “*Bootstrapping is the mechanism by which a newly joining peer identifies a peer already in the overlay to which it can issue the join request*”. How to manage a bootstrap node is out of the scope of this project. In this project, the first node simply keeps all node names in a Harmonia network.

Supervisor Function The supervisor function configures and monitors the start, restart, and termination of worker processes. It uses OTP supervisor behavior. In a Harmonia DHT node, the worker processes under the supervisor are restarted up to five times within 2000 seconds when they fail. If a process fails more than six times within 2000 seconds, the DHT node is stopped.

3.3 Routing Layer

The function of the routing layer is to find an address of a resource. Although Harmonia currently uses Chord as a protocol, it can use any protocol as long as the same interface is provided.

3.3.1 API

This layer provides the following interfaces to an upper layer:

```
lookup(key)
lookup_with_successor_list(key)
```

lookup(key) returns a node ID for a given key. This is used to determine the node ID that holds data associated with the key. *lookup_with_successor_list(key)* returns a successor list along with a node ID for a given key. A successor list is used for replication of data.

3.3.2 Data Structure

Using Chord protocol, this layer maintains a finger table, a successor list, and predecessor. The routing information is kept in the router process as a *record* of Erlang [4, p. 157]. The stabilizer process periodically calls *stabilize*, *fix_finger*, and *check_predecessor* functions described in Figure 22 and Figure 23 to keep these data up-to-date. The data structure of routing information is shown in List 1.

List 1: Routing Information Data Structure

```

1 -record(state, {node_name,           % node name
2                   node_vector,       % hash value of node name
3                   predecessor=nil,    % predecessor
4                   finger=[],         % finger table
5                   succlist=[],       % successor list of the node
6                   current_fix=0}).   % current target entry of fix_finger \
                                     -function

```

node_name is a name of a DHT node. *node_vector* is a hash value of the node name. *predecessor* is a predecessor (initial value is nil). *finger* is a finger table (initial value is empty list). *succlist* is a successor list (initial value is an empty list). *current_fix* indicates the status of which entry of finger table is now maintained by *fix_finger* function that is called periodically. The elements of predecessor, finger, and succlist are expressed as a tuple of a router process name and its node vector, such as {hm_router_foo,90}. An example of this data structure from log file of a Harmonia DHT node is shown in List .2.

List 2: Routing Information: example from log file

```

1 fixfinger_loop: Next:[1], State:[{state,hm_router_cat,245,
2                               {hm_router_dog,220},
3                               [{hm_router_hoge,17},
4                               {hm_router_bar,38},
5                               {hm_router_hoge,17},
6                               {hm_router_hoge,17},
7                               {hm_router_hoge,17},
8                               {hm_router_bar,38},
9                               {hm_router_foo,90},
10                              {hm_router_xxx,171}]},
11                              [{hm_router_hoge,17},
12                              {hm_router_bar,38},
13                              {hm_router_foo,90},
14                              {hm_router_xxx,171}]},
15                              8}]].

```

This example is data of node name *hm_router_cat*. The predecessor is *hm_router_dog*. It shows a finger table of 8 entries and a successor list of 4 entries. Lastly, it shows *current_fix*.

3.3.3 Hash Function

Harmonia uses SHA [16] as hash function. SHA is designed to be collision resistant and the probability of two different keys hashed to the same value is negligible. Another candidate for the hash function of Chord is MD5 [22] because it is also considered to be collision resistant. Harmonia uses the *crypto* application to generate SHA hash value as following:

List 3: Hash function of Harmonia

```

1 get_digest(Key) when is_integer(Key) ->
2     <<Vector:160>> = crypto:sha(integer_to_list(Key)),
3     Vector rem ?max_key_value;

```

The code generates a 160 bits message digest from integer *Key*. The latest crypto application (R14B) supports FIPS180-2 [23] as stated in Erlang Official Website [24].

3.3.4 Routing Method

The routing process of Harmonia is efficient. Generally, when a message is routed from node A through B to C, then returned to A, there are 3 types of control flow. Figure 12 shows these types of control flow. First, when A requests B, B requests C, then C returns to B, and B returns to A. Second, A requests B, then B returns C as forwarding destination. A requests

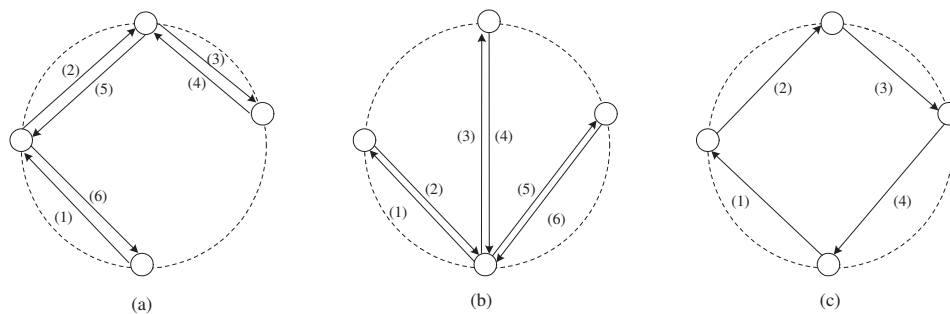


Figure 12: Routing methods:Harmonia takes (c) for routing

C again. C returns A with answer. Third, A requests B, and B requests C, then C returns to A with an answer. In this case, B does not reply to A, but forwards the request to C. There are two advantages of this method. First, there are fewer messages. This would alleviate network traffic and improve the latency. Second, this method helps avoid deadlock because routing processes are not blocked when they receive inquiries.

3.3.5 Average Cost of Routing

The cost of routing can be considered as the number of nodes to be traversed to reach the target node ID. This is called the *path length*. The average path length of Chord is about $\frac{1}{2}\log N$ where N denotes the total number of nodes in an identifier circle [3].

3.4 Data Store Layer - Simple store/get

Simple store/get methods provide primitive methods to store and retrieve data in the system. Although range query functions belong to the same layer, they are independent of these functions and use a different data structure and algorithms. Therefore simple store/get methods are described separately here.

3.4.1 API

This layer provides the following interfaces to the upper layer:

```
get(Key)
store(Key, Value)
```

store(key, value) function provides a way to store data on DHT. *get*(key) function provides a way to retrieve data stored on DHT. Both functions use the *lookup*(key) function of the routing layer to get target node to store/retrieve data. Then a request to store/get data is sent to the node.

Data is replicated on successor nodes of the target node in *store* function. Successor nodes can be retrieved with the *lookup_with_successor_list* function of the routing layer.

3.4.2 Data Structure

To store and get key-value pair data, each node creates a table for simple store/get API. In case of Harmonia, ETS (Erlang Term Storage) [4, p. 213] is used. The method to store data can be switched to other data storage methods as long as the interface is kept the same. A table to store data is created when a node starts.

Data	Description
data table	this is a table created as default on each node

Table 5: Data Maintained in Data Storage Layer (Simple store/get)

3.4.3 Algorithm

With *lookup* and *lookup_with_succlist* functions, performing simple store/get on DHT is quite simple. Steps to perform simple store and get are following:

store(*key*, *value*)

1. Get a node ID and a successor list of the *key* using *lookup_with_succlist* function.
2. store key-value pair to all those nodes

get(*key*)

1. Get a node ID and a successor list of the *key* using *lookup_with_succlist* function.
2. Retrieve data from the first successful node ID in the nodes above

3.4.4 Parallel Store

Data is stored on a successor node of the key as well as on all nodes in the successor list. These store processes are performed in parallel. Figure 13 shows the sequence.

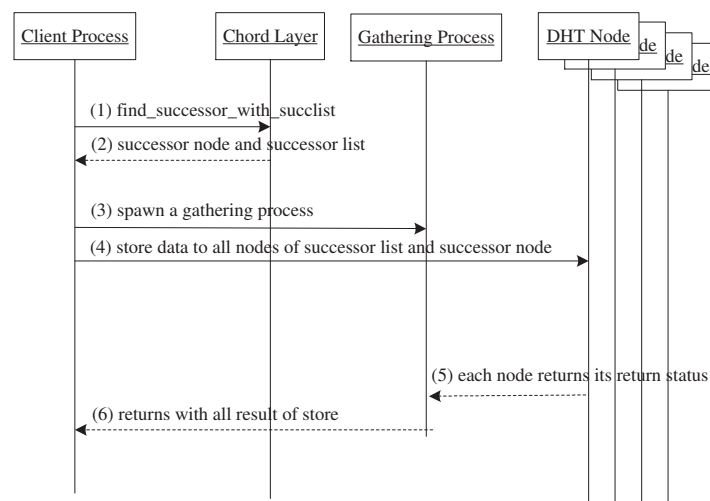


Figure 13: Parallel Store Sequence

(1) When a client process issues *store(key, value)* API, it asks the Chord layer for the responsible node for *key* and its successor list. (2) Then the client process gets a list of node to store data. (3) the client spawns a process to gather the result status of store for all nodes. This process waits for all nodes to return its return code. (4) The client process sends messages to store data to all nodes. When a process sends messages to store data to a node, the process is blocked until the result is returned. Therefore this is done by spawning a request process for each data store, and each process waits for its result and sends that result to the gathering process. This step is performed in parallel. (5) Each node returns its result status code to the gathering process. (6) When all process respond, the gathering process returns the status code to the client process.

3.5 Data Store Layer - Range Query

Range query methods depend on a lookup service of the routing layer, but are independent of simple store/get methods of the same layer. It provides range query functionality with replication.

3.5.1 API

This layer provides the following interfaces to the upper layer:

```
create_table(domain, table, table_configuration)
drop_table(domain, table)
rstore(domain, table,
        [data items for each field of table])
rget(domain, table, query_condition)
```

- *create_table* function creates index table needed to store index for a range query. It also saves the table configuration. It takes (1) domain name (2) table name and (3) table configuration as arguments. Data is partitioned by domain and table name. Table configuration includes the field information of the table. Each field has its name, key field flag (whether it is key field or not), and default value. Once a table is created, data can be stored to the table using the *rstore* function.
- *drop_table* function deletes table specified by the domain and table name.
- *rstore* function stores data specified in its arguments. First it stores index then data. It takes (1) domain name (2) table name and (3) list of data for each field as arguments. The list of data for each field consists of tuple of field name and its value.

- *rget* function performs range query for the specified arguments. It first retrieves DHT nodes that hold correspondent data for query from index table, then issues query to all those DHT nodes. This function takes (1) domain name (2) table name and (3) query condition as arguments. Range query of Harmonia allows relational operators ($=$, $!$, $=$, $<=$, $>=$, $<$, $>$) and logical operators (and, or).

3.5.2 Data Structure

The approach for range query in this project is to store table configuration and index along with data. Index exists for each domain/table name combination. An index table is created by *create_table* function. Index is built when data is stored and looked up when a range query is issued. This concept is expressed in Figure 14:

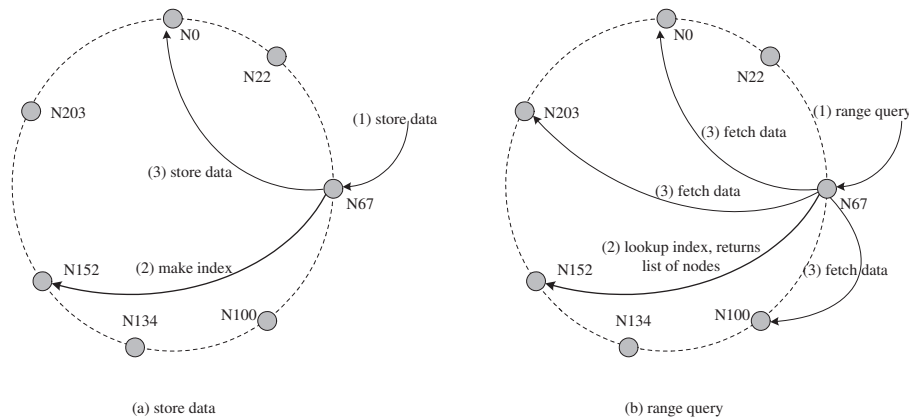


Figure 14: Store and range query: (a) When storing data, first, data store function is called with domain/table name along with data. Next, data items of key fields are stored into index table. Finally, all fields of data is stored into data table (b) When a range query is issued, first, a range query is issued with domain/table name and query condition. Next, the node that holds an index table is determined and a list of nodes that hold data is retrieved according to the query condition. Finally, data that matches query condition is retrieved from all nodes that hold corresponding data

Data	Description
table configuration	this data is used to recognize key fields of a table when index table is stored and searched
index table	this is used to determine nodes that hold data for the correspondent data of query
data table	this is data itself, stored in default table

Table 6: Data Maintained in Data Storage Layer (Range Query)

To perform a range query using an index, this layer maintains the table configuration information, an index table, and a data table.

Table configuration is used to parse query conditions to the native query form of underlying data storage method of Harmonia. It is also used to distinguish key fields from data fields because only key fields are stored to an index table. Table configuration consists of (1) domain name (2) table name (3) field names (4) key field flags and (5) default values of each field. Table configuration is kept by a DHT node that holds an index table for the same domain/table combination.

An index table is used to find nodes that hold data for the query condition. When range query is issued, the index table is searched with range query. Therefore the data storage method used in this project has to support range queries. Index table consists of (1) key fields of the table and (2) a field for DHT node ID that holds data for these key fields. Domain name and table name are used to partition data. One domain can have multiple tables. An index is created for each table when the *create_table* function is issued. A DHT node that holds an index table is determined by hashing domain/table name combination.

A data table is used to store data. All DHT nodes have a data table because it is created when DHT node is initiated. A data table consists of fields of (1) domain name (2) table name and (3) all fields of the table.

3.5.3 Algorithm

Using index and table configuration, the algorithms for *rget* and *rstore* functions are described below:

rstore(*domain*, *table*, *list of value*)

1. Get a successor *node ID1* and a successor list of the combination of *domain/table* using *lookup_with_succlist* function.
2. Get table configuration from the *node ID1* above
3. Determine the key fields of the *table*
4. Get a successor *node ID2* of the combination of *domain/table/value* of key fields in *list of value* using *lookup_with_succlist* function.
5. Store (1) key fields and (2) the *node ID2* into index table on the *node ID1* and all nodes in its successor list
6. Store (1) domain name and (2) table name (3) all key and data fields into data table on the *node ID2* and all nodes in the successor list of the node

rget(*domain*, *table*, *query condition*)

1. Get a *node ID1* and successor list of the combination of *domain/table* using *lookup_with_succlist* function.
2. Perform query with *query condition* to the index table on the first alive node in *node ID1* and nodes in its successor list and get a list of nodes that hold data
3. Perform query with *query condition* to the data table on all nodes returned in the previous step

3.5.4 Example of Range Methods

In this section an example of table configuration, index table, and data table are shown. When an index table is created using *create_table* function, table configuration is also saved on the same DHT node. An example is shown in Table 7, 8 and 9. In this example, domain name is *user1* and table name is *demo graph*.

Domain name : user1 Table name : demo graph

Fieldname	Key	Default vlaue
Age	true	0
City	true	nil
Gender	false	nil
Occupation	false	nil

Table 7: Table Configuration Example

Domain name : user1 Table name : demo graph

Age	City	Data node
85	San Francisco	N67
21	San Francisco	N203
34	Daly city	N203
28	San Jose	N0
30	San Mateo	N152
10	Daly city	N152

Table 8: Index Table Example

Using examples shown in Table 7 and 8 and 9, data is retrieved using the following steps:

1. *rget*("user1", "demo graph", "*age* >= 30") is issued.
2. Using *lookup*("user1, demo graph") by passing "user1" and "demo graph" combination(*lookup* function generates digest key from domain and table name combination), *rget* function retrieves the node that holds index table and table configuration.
3. Query "*age* >= 30" is performed to the user1/demo graph index table, and it gets node IDs N67, N152, N203.
4. Query "*age* >= 30" is performed to the user1/demo graph data table on N67, N152, and N203, and result is returned.

Using examples shown in Table 7, 8 and 9, data is stored using the following steps:

1. *rstore*(user1, demo graph, [{age,20},{city,San Francisco},{gender,male},{occupation,student}]) function is issued.

Domain name	Table name	Age	City	Gender	Occupation
user1	demo graph	85	San Francisco	male	school faculty
user1	demo graph	21	San Francisco	male	public officer
user1	demo graph	34	Daly city	female	doctor
user1	demo graph	28	San Jose	male	student
user1	demo graph	30	San Mateo	female	student
user1	demo graph	10	Daly city	male	student

Table 9: Data Table Example

2. Using *lookup*("user1, demo graph") function, *rstore* function determines the node that holds the index table and configuration table
3. Retrieves table configuration of user1/demo graph table, then it identifies key fields (age and city in this example)
4. Using *lookup*("user1, demo graph, 20, San Francisco"), it determines the node that will hold data.
5. Stores key fields (20 and San Francisco) and node ID for data into index table
6. Stores domain/table names (user1/demo graph) and all fields into data table on the node ID

3.6 Client Interface

Client interface provides data store APIs and administrative APIs. These APIs are provided as a library module on DHT node. Therefore, clients need to connect to DHT node or issue RPC (remote procedure call).

3.6.1 API

This layer provides the following interfaces:

```
create_table(domain, table, [attributes])
drop_table(domain, table)
get_table_info(domain, table)
store(key, value)
get(key)
rstore(domain, table,
        [data items for each field of table])
rget(domain, table, query condition)
log_start()
log_stop()
```

Domain and table as arguments are given as strings. A Domain includes multiple tables. A Table includes multiple fields.

- *create_table* function creates an index table needed to store index for range query. It returns two lists. One is a list of nodes on which the table creation succeeded. Another is a list of nodes on which the table creation failed.
- *drop_table* function deletes a table. It returns the same contents to *create_table* function.

- *get_table_info* function returns a table configuration.
- *store* function stores key-value pair data. It returns the result of the API, which are OK, Partial, and NG. OK means data was stored on all nodes in the successor list and successor node. Partial means some nodes failed. NG means data was not stored at all.
- *get* function returns a correspondent value to the key
- *rstore* function stores data to the system. It returns the same content to *store* function.
- *rget* function returns correspondent data to the query condition.
- *log_start* function starts logging of the system.
- *log_stop* function stops logging of the system.

Examples of command line execution are shown in List 4. Line 1 issues *create_table* function, creating table as “Tb11” in “Domain1” with fields specified in the list. It returns the list of nodes on which the table was created and replicated. Line 8 issues the *store* function. It stores data without specifying domain and table. Therefore an index is not created for this data, and range query cannot issue for this data. Line 10 issues *get* function, retrieving correspondent data for the *key1*. Line 12 issues *get_table_info* function. It retrieves correspondent table name and configuration of the table. Line 15 issues *rstore* function, storing data for range query. It specifies domain name as “Domain1”, table name as “Tb11”, and list of tuples that have a field name-value pair. Line 17 issues *rget* function, retrieving data with a query condition. Line 19 issues *drop_table* function, deleting the table.

List 4: Example User Interface

```

1 (xxx_node@ubu)1> hm_cli:create_table("Domain1", "Tb11", [{ "Fld1", true, nil}, {"Fld2", false, \
  → true}, {"Fld3", false, 0}]).
2 {ok, [{hm_router_hoge, instance},
3       {hm_router_bar, 38},

```

```

4      {hm_router_foo,90},
5      {hm_router_xxx,171},
6      {hm_router_dog,220}],
7      []}}
8 (xxx_node@ubu)2> hm_cli:store(key1, "value1").
9 {ok,5}
10 (xxx_node@ubu)3> hm_cli:get(key1).
11 {ok,[{key1,"value1"}]}
12 (xxx_node@ubu)4> hm_cli:get_table_info("Domain1", "Tbl1").
13 {ok,'Domain1Tbl1',
14     [{ "Fld1",true,nil},{ "Fld2",false,true},{ "Fld3",false,0}]}
15 (xxx_node@ubu)5> hm_cli:rstore("Domain1", "Tbl1", [{ "Fld1", key1},{ "Fld2", "Hello World"},{ "\
    →Fld3",100}]).
16 {ok,5}
17 (xxx_node@ubu)6> hm_cli:rget("Domain1", "Tbl1", "Fld1 == key1").
18 {ok,[{key1,"Hello World",100}]}
19 (xxx_node@ubu)7> hm_cli:drop_table("Domain1", "Tbl1").
20 {ok,[{hm_router_hoge,instance},
21      {hm_router_bar,38},
22      {hm_router_foo,90},
23      {hm_router_xxx,171},
24      {hm_router_dog,220}],
25      []}}

```

3.7 System Architecture Summary

This chapter described system architecture of Harmonia, a DHT storage system with range query that was developed in this project. The detail of each component, as well as the relationship between them was described. In the next chapter, Harmonia is evaluated from several aspects.

4 EVALUATION

In this chapter, Harmonia is evaluated from several aspects. These evaluations include (i) Verification of API correctness, (ii) Performance with different conditions of virtual nodes, threads, and range query (iii) Comparisons with memory usage, and (iv) Measurement of load balancing.

These evaluations are performed using the following APIs:

- *store* function: This API stores a key-value pair.
- *get* function: This API retrieves the correspondent value of key
- *rstore* function: This API provides range query function for the data stored using it. The index is built along with data.
- *rget* function: This API provides range query function.

The main differences between *store/get* and *rstore/rget* are:

- *rstore* and *rget* functions provide range query functionality
- *rstore* and *rget* functions use an index to perform range query, therefore *rstore* function writes an index before it writes data. *rget* function reads the index in order to retrieve data.
- Table needs to be created before *rstore* function is called.

4.1 Common Configuration

4.1.1 Hardware Configuration

/proc/cpuinfo file and *free* command were used to collect information. The specifications of 5 machines are shown below: netlab3 and netlab4 are dual core machines. The capacities of each machine are quite different. These environments can affect the results of the experiments. Machines are connected in a LAN (Local Area Network) and bandwidth is around 90 Mbits/sec (bandwidth was evaluated using *iperf* command).

Hostname	CPU	Memory
netlab3	Dual Core, Intel(R) Core(TM)2 CPU 4300 @ 1.80GHz	1006Mbytes
netlab4	Dual Core, Intel(R) Core(TM)2 CPU 4300 @ 1.80GHz	1006Mbytes
netlaba	Intel(R) Pentium(R) 4 CPU 1.70GHz	497Mbytes
netlabbb	Intel(R) Pentium(R) 4 CPU 1.70GHz	749Mbytes
dell	Pentium III (Coppermine) 1GHz	496Mbytes

Table 10: Machine Specifications

4.1.2 Software Configuration

OS and Erlang VM version are shown in Table 11.

Harmonia configuration is shown in Table 12.

Hostname	OS	Erlang (BEAM) emulator version
netlab3	2.6.25-gentoo-r7 #8 SMP	5.6.3
netlab4	2.6.25-gentoo-r7 #1 SMP	5.6.3
netlaba	2.6.31-14-generic #48-Ubuntu SMP	5.7.2
netlabbb	2.6.31-14-generic #48-Ubuntu SMP	5.8.1
dell	2.6.31-14-generic #48-Ubuntu SMP	5.8.1

Table 11: OS and Erlang VM version

Hash function is SHA 160 bits, therefore the length of a finger list is 160. *fix_finger* function to update finger list runs every seconds (one second intervals). Length of a successor list is configured to four. Therefore when data is stored, four copies are replicated.

Hash Key Bit	Finger List Length	Successor List Length
160	160	4

Table 12: Harmonia Configuration

4.2 Verification Test

Verification test performs a test function that runs a series of APIs and checks the correctness of APIs.

The tests cover following APIs.

- *store* function
- *get* function
- *create_table* function
- *drop_table* function
- *rstore* function
- *rget* function

The test function stores data and checks if APIs can retrieve the same data correctly. Since the data store layer is independent of the internal state of a routing layer, configuration of virtual nodes and physical machines does not affect the test results.

The number of data to be tested is 10. *store* test function stores 10 data as default. Then *get* test function retrieves all data stored, and checks the correctness of each value.

For a range query test, the test function creates a table with three fields and stores data. Data is stored to the table using *rstore* function. Then range query is performed with about 30 conditions, and results are checked if the functions return an expected set of data that matches the specified query conditions.

These test functions are included in *test_all* function of *hm_cli_test* module in Appendix F.

4.3 Comparison with Different Number of Nodes

4.3.1 Purpose

The purpose of this performance experiment examines the performance of APIs under the different number of virtual nodes. In addition, this test shows the relative performance between

APIs. This test evaluates the following APIs:

- *store* function
- *get* function
- *rstore* function
- *rget* function

4.3.2 Specification

The average latency in 10,000 operations are examined for each API. 10,000 data are stored using the *store* function, then all data are retrieved one by one using the *get* function. Therefore, there are 10,000 data stores and data retrievals. The same operations are performed for *rstore* and *rget* functions.

The above operations are performed with a different number of nodes (10, 17, 33, 65, 105).

4.3.3 Metric

The latency of one operation for each API is measured in milliseconds. The latency is calculated as the average of 10,000 operations.

Namely, *The average latency* = $\frac{\text{Total time for 10,000 operations}}{10,000}$.

4.3.4 Configuration

The configuration of the virtual node number for each machine in the test case is shown in Table 13. For example, to perform the test case of 10 virtual nodes, 4 virtual nodes run on netlab3, and 3 on netlab4, 1 on netlaba, netlabb, and dell.

The actual test configurations and results are shown in Appendix G.1.

Test Case	netlab3	netlab4	netlaba	netlabb	dell
10 nodes	4	3	1	1	1
17 nodes	6	5	2	2	2
33 nodes	11	10	4	4	4
65 nodes	21	20	8	8	8
105 nodes	41	40	8	8	8

Table 13: Number of Virtual Nodes for Each Machine (Different Number of Nodes)

4.3.5 Evaluation

Figure 15 shows the average latency of Harmonia APIs for a different number of virtual nodes. Plot data and script are shown in Appendix H.1.

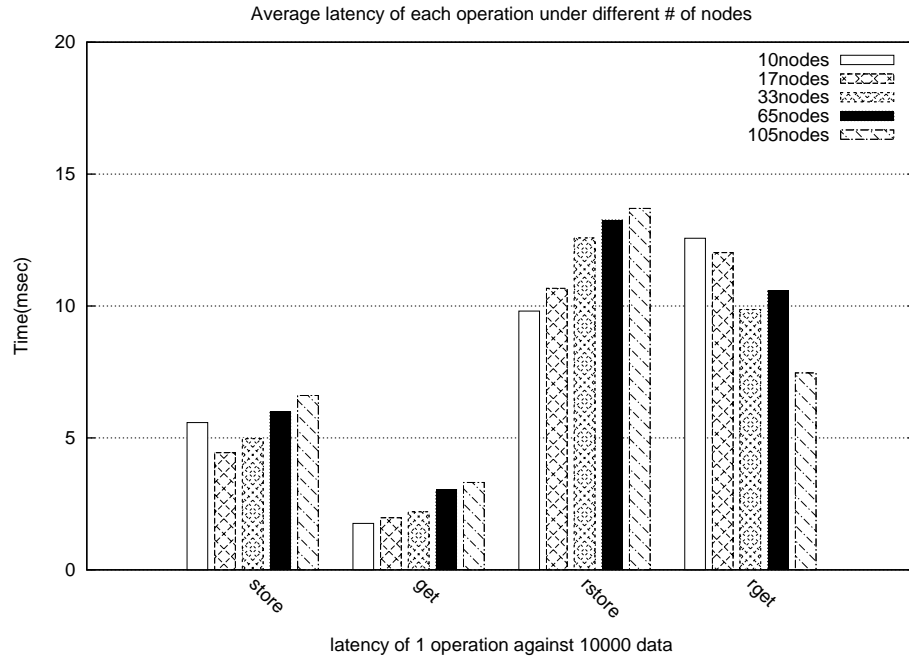


Figure 15: Average Latency of Harmonia APIs for a different Number of Virtual Nodes

First, *store* and *rstore* functions take more time than *get* and *rget* functions in general. This can be caused by replication during *store* and *rstore* functions.

Next, the latency increases as the number of virtual nodes increase, but not proportionally. Comparing the 10 node case with 105 node case (more than 1,000% increase) in *get*, *store*, *rstore*, the latencies increase less than about 100%. This can be caused because of the log scale path length of Chord routing. In this case, the number of virtual nodes increase from 10 to 105, then the path length increases from $\frac{1}{2}\log 10 \approx 1.66$ to $\frac{1}{2}\log 105 \approx 3.35$ in average [3], that is about 100% increase. In addition, the cost of data storage and retrieval for each operation

should be identical between the 10 node case and 105 node case.

Finally, the latency of *rget* function decreases as the number of virtual nodes increases. While the path length of routing should be longer when the number of nodes increases, this result can be caused by the biased data load. Since Harmonia currently uses ETS for data storage, range query requires an exhaustive search against a table. If the number of data held by one virtual node is large, this can cause the result in figure 15.

4.4 Comparison with Different Number of Threads

4.4.1 Purpose

The purpose of this experiment is to test the performance of APIs when operations are performed concurrently. Harmonia is based on DHT and the system consists of many virtual nodes and real machines. Therefore, it is possible to achieve better performance when operations are performed concurrently. Since Harmonia does not provide concurrency control and is not suitable for concurrent writes, only read APIs were tested.

This test evaluates the following APIs:

- *get* function
- *rget* function

4.4.2 Specification

The test performs a total 10,000 read operations. Data is stored in advance. The 10,000 read operations were divided into each thread. For example, a test with 2 threads divides a task into 2 parts. The first thread reads data from 1 to 5,000. The second thread reads data from 5,001 to 10,000. These 2 threads run concurrently.

The tests were performed for 1, 2, 4, 8, and 33 threads. The number of virtual nodes are set to 47. The steps of test is shown below:

1. Start measuring time
2. Mapper process maps each task to virtual nodes (one task for one virtual node).
3. Mapper process creates a gathering process
4. Each virtual node performs its task and returns the result to the gathering process
5. When all virtual nodes return, the gathering process returns to the mapper process
6. Stop measuring time

4.4.3 Metric

The time to perform 10,000 operations is measured in seconds.

4.4.4 Configuration

The virtual nodes and tasks allocated for each machine are shown below. The mapper process and gathering process run on netlab3, so the number of threads on netlab3 were reduced. The actual test configurations and results are shown in Appendix G.2.

Test Case	netlab3	netlab4	netlaba	netlabbb	dell
All cases	21	20	2	2	2

Table 14: Number of Virtual Nodes for Each Machine (Different Number of Threads)

Test Case	netlab3	netlab4	netlaba	netlabbb	dell
1 thread	1	0	0	0	0
2 threads	1	1	0	0	0
4 threads	1	1	1	1	0
8 threads	1	4	1	1	1
33 threads	13	14	2	2	2

Table 15: Number of Threads Allocated for Each Machine

4.4.5 Evaluation

Figure 16 shows total time for 10,000 operations of get and rget functions for different numbers of threads. Plot data and script are shown in Appendix H.2. First, the performance of *rget* function improves when the number of threads increases from 1 to 4, from about 52 seconds to about 34 seconds. However the performance does not improve when the number of threads increases from 4 to 33. One reason to be considered is that *rget* function needs to

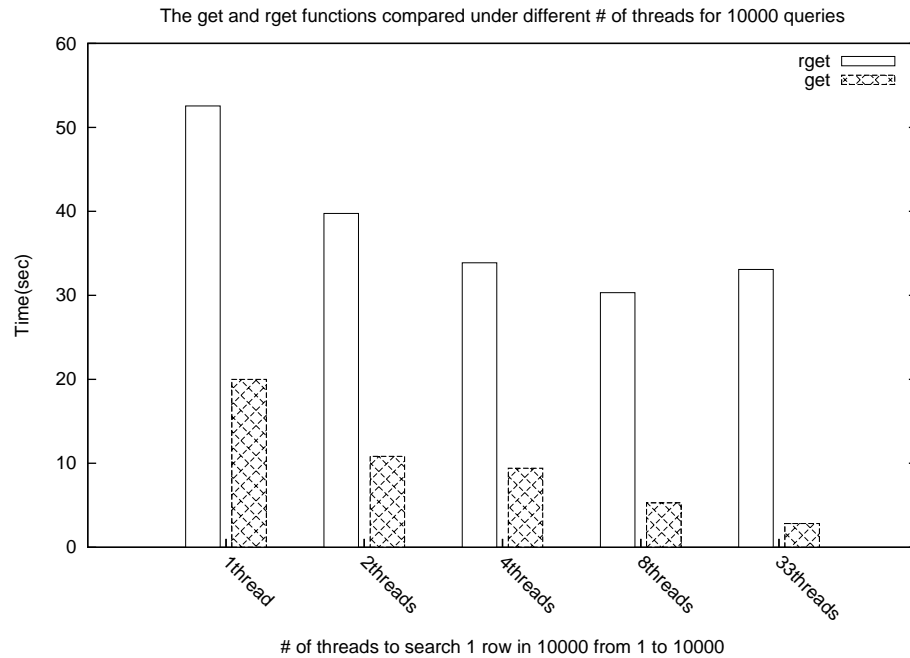


Figure 16: Comparison with Different Number of Threads

read the index. Index of Harmonia is held by a specific virtual node. Therefore, threads try to access the index concurrently, which can be a bottleneck for performance.

Next, the performance of *get* function improves when the number of threads increases from 1 to up to 33, from about 20 seconds to about 2.8 seconds. This is because *get* function does not access the index and data is fully distributed. It also indicates that routing protocol of Chord can endure concurrent processing.

The result shows that *rget* function does not fit for concurrent processing, while *get* function does. Therefore, when there are many jobs to run with different query conditions, the *get* function model is more efficient.

4.5 Comparison with Different Range Query Conditions

4.5.1 Purpose

The purpose of this experiment is to compare the simple *get* function with the *rget* function. Range query of Harmonia builds and reads the index, therefore the cost is greater than simple query method on DHT. This experiment determines whether the range query method using the *rget* function is cost effective. This test evaluates the following APIs:

- *get* function
- *rget* function

4.5.2 Specification

The test compares performance of the *get* function and the *rget* function by retrieving same range of data. 1,000 data are stored in advance. For example, the test compares the performance of retrieving the data range from 1 to 10 using the *get* function and the *rget* function. In the case of using the *get* function, it retrieves all data and checks if the value is between 1 and 10, since it has no functionality of range query. The ranges tested are (1) from 1 to 1, (2) from 1 to 10, (3) from 1 to 100, and (4) from 1 to 1,000.

4.5.3 Metric

The time to perform each range query is measured in seconds.

4.5.4 Configuration

The number of virtual nodes for each machine is shown in Table 16.

Test Case	netlab3	netlab4	netlaba	netlabb	dell
All cases	21	20	4	4	4

Table 16: Number of Virtual Nodes for Each Machine (Different Query Conditions)

4.5.5 Evaluation

Figure 17 shows total time for each range query using either *get* or *rget* function. The time is shown in log scale. Plot data and script is shown in Appendix H.3.

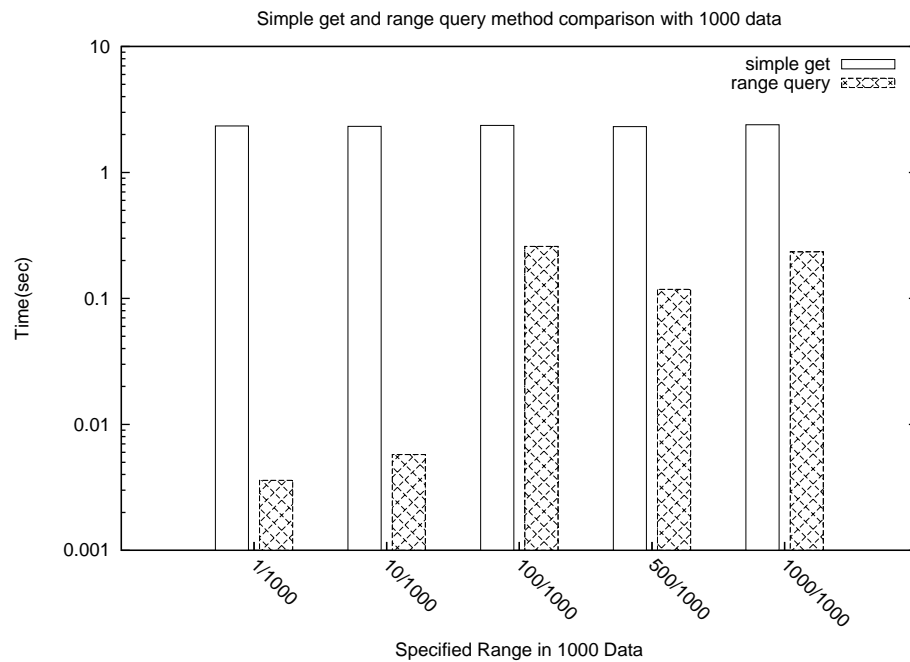


Figure 17: Comparison with Different Range Query Conditions

First, in all cases the *rget* function outperforms the *get* function. When searching one record out of 1,000 data, the *rget* function is about 650 times faster.

Second, even for the case of retrieving 1,000 records out of 1,000 data, the *rget* function is about 10 times faster. This is because the *get* function has to be called 1,000 times, while the

rget function has to be called only one time.

The experimental result shows that range query method of Harmonia is more efficient.

4.6 Memory Usage

4.6.1 Purpose

The purpose of this experiment is to test how much space is necessary for range query of Harmonia compared with a simple store function.

This test evaluates the following APIs:

- *store* function
- *rstore* function

4.6.2 Specification

The test compares the total size of 10,000 data stored in the system with different data configurations using *store* and *rstore* functions. There are four different configurations shown below: Table 17 and 18 are short data and only have one key and one data fields. Table 19 and 20 are long data. *store* function can only have one key field, so Table 19 has only one key. On the other hand, Table 20 has three key fields.

Field type	Size	Key
int	n/a	true
char	30	false

Table 17: Data for store(short)

Field type	Size	Key
int	n/a	true
char	30	false

Table 18: Data for rstore(short)

Field type	Size	Key
int	n/a	true
int	n/a	false
char	10	false
char	30	false

Table 19: Data for store(long)

Field type	Size	Key
int	n/a	true
int	n/a	true
char	10	true
char	30	false

Table 20: Data for rstore(long)

Both theoretical and actual size are measured. Theoretical size can be calculated knowing how Harmonia stores data with APIs. When calculating a theoretical value, Erlang specific data size was not considered because the main interest of this experiment is relative size of data stored by the *rstore* function to that of the *store* function. Therefore the data size of integer is considered 4 bytes and one character is one byte.

Real size was measured using *ets:i()* function of Erlang. *ets:i()* function displays information about all ETS tables on tty. The result is shown in Appendix G.4.

4.6.3 Metric

The space used by each operation is measured in mega bytes.

4.6.4 Configuration

Data was stored to 64 virtual nodes.

4.6.5 Evaluation

Figure 18 shows total data size for each operation using the *store* and the *rstore* functions. Plot data and script is shown in Appendix H.4. First, range query of Harmonia requires space

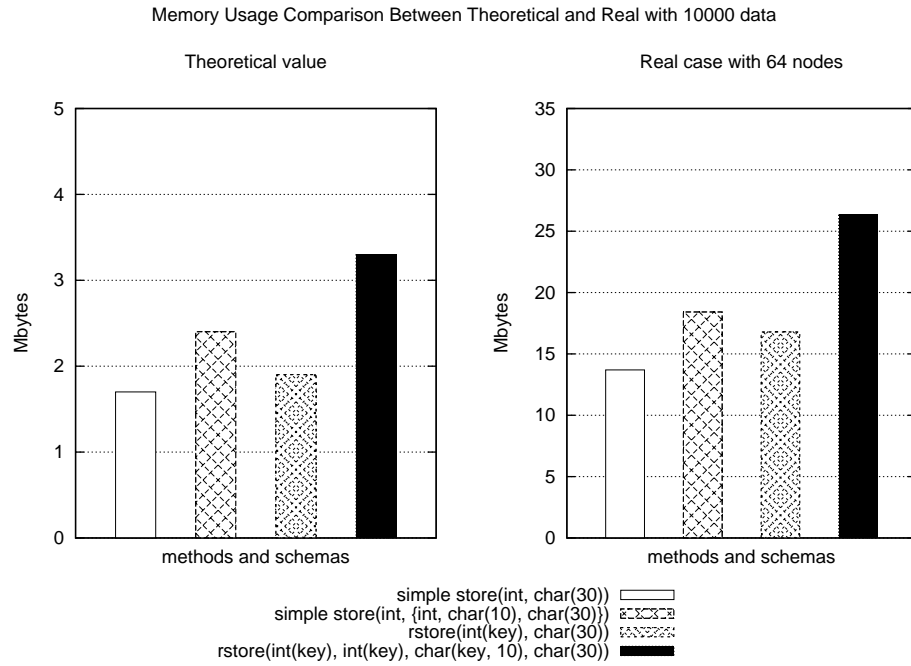


Figure 18: Memory Usage with Theory and Real Case

for an index table, a data table and table configuration. The size of table configuration is small and negligible.

The size of an index table is $s \times (\sum_{i=1}^k size(i))$, where s denotes a number or data, k denotes the number of key fields, $size(i)$ denotes the size of the field i th.

The size of a data table is $s \times (\sum_{i=1}^j size(i))$, where s denotes a number or data, j denotes the number of fields, and $size(i)$ denotes the size of the field i th.

Therefore, the total amount of theoretical space is $s \times (\sum_{i=1}^k 2size(i) + \sum_{i=k+1}^j size(i))$.

This simply means that the total space can also be denoted as $A + (A \times R)$, where A denotes the

size of data table, and R denotes the ratio $\frac{\text{the number of key fields}}{\text{the total number of fields}}$. Therefore, as this ratio increases, the space required increases up to the twice of data table. When replication is considered, it requires simply r times more space. Therefore, total amount of theoretical space is $s(r + 1) \times (\sum_{i=1}^k 2\text{size}(i) + \sum_{i=k+1}^j \text{size}(i))$.

The theoretical space was calculated according to above formulas.

Next, the theoretical sizes are quite smaller than actual values. This is because the object sizes of Erlang are different from those of theoretical size. According to *Erlang Efficiency Guide* in Erlang Official Site [24], the sizes of main objects are shown in Table 21. It shows the size of integer is nearly equal to the theoretical value, but a string of Erlang requires much more space than the theoretical value.

Data type	Memory size
Small integer	1 word(28 bits in 32-bit implementation)
String	1 word + 2 words per character
Tuple	2 words + the size of each element
ETS Table	Initially 768 words + the size of each element (6 words + size of Erlang data). The table will grow when necessary.

Table 21: Main Data Type Size of Erlang

Finally, the relative size of theory between *store* and *rstore* are rather similar to actual size, as seen in Figure 18.

4.7 Load Balancing

4.7.1 Purpose

The purpose of this experiment is to test how data is loaded to each virtual node when the number of virtual nodes per machine increases. Since load balancing affects performance, it is important that data is distributed evenly to each virtual node.

4.7.2 Specification

The test counts the number of data that is held by each virtual nodes. Next, it checks the maximum, average, and minimum number of data held by each virtual node.

The tests were performed for 5, 10, 20, and 40 virtual nodes per machine. The number of data stored to the system is 10,000, but data is replicated to 4 more nodes. Therefore 50,000 data exist in the system.

4.7.3 Metric

The maximum and minimum number of data were chosen from all numbers retrieved from each virtual node. For example, in the case of 10 virtual nodes per machine, there are 50 virtual nodes since there are 5 machines. Maximum and minimum number of data counts are chosen. An average is calculated by $\frac{\text{sum of data counts}}{\text{total number of virtual nodes in the system}}$.

4.7.4 Configuration

The configuration of the virtual node number of machines in each test case is shown in Table 22. The number in netlab3 is increased by one because the commands were issued from netlab3 virtual node. The actual test configurations and results are shown in Appendix G.5.

Test Case	netlab3	netlab4	netlaba	netlabb	dell
5 nodes	6	5	5	5	5
10 nodes	11	10	10	10	10
20 nodes	21	20	20	20	20
40 nodes	41	40	40	40	40

Table 22: Number of Virtual Nodes for Each Machine

4.7.5 Evaluation

Figure 19 shows average latency of Harmonia APIs for a different number of virtual nodes. Plot data and script are shown in Appendix H.5.

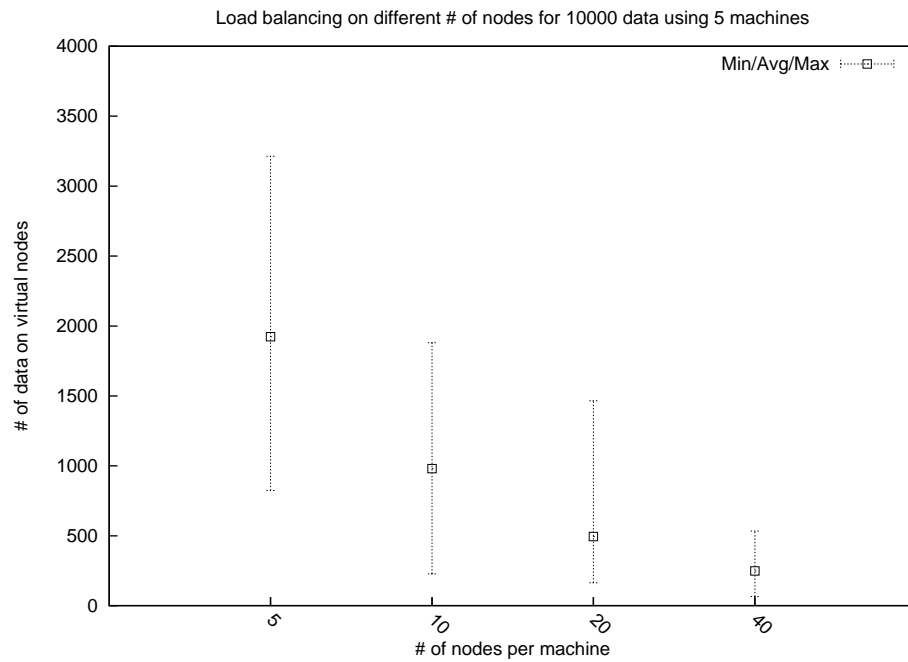


Figure 19: Load balancing for different number of virtual nodes per machine

As the number of virtual nodes per machine increases, the range of data between maximum and minimum becomes smaller. Also the maximum number decreases as the number of virtual

nodes per machine increases. This indicates that the load is more balanced when the number of virtual nodes per machine increases.

4.8 Evaluation Summary

In this chapter, Harmonia was evaluated from several aspects. These evaluations include (i) Verification of API correctness, (ii) Performance with different conditions of virtual nodes, threads, and range query (iii) Comparisons with memory usage, and (iv) Measurement of load balancing. The experimental results show that the range query method outperforms exhaustive search using a simple get method; and the data load distribution was more balanced as the number of virtual nodes increases. However, the performance of the range query does not improve linearly as the number of virtual nodes increases since the access to index table becomes bottleneck. In addition, it was confirmed that the memory usage of Harmonia is larger than theoretical value due to the data type definition of Erlang.

5 CONCLUSION

In this project, range query method on DHT was designed, implemented, and evaluated. Harmonia, the software implemented in this project, was developed from scratch using Erlang/OTP. Harmonia has a Chord based DHT component and data store functions with range query. It also provides a data replication mechanism.

In order to develop Harmonia, Erlang and OTP were used. Using Erlang and OTP, the development was efficient because of a number of advantages, including message passing and distributed programming facilities.

The range query design of Harmonia was evaluated and confirmed to work correctly. The method of range query proposed in this project outperforms the range query method using a simple get function, and data load was more balanced when the number of virtual nodes increased.

The future work of this project will be an attempt to distribute index for range query. Since the index is stored on a specific virtual node in Harmonia, this range query design is not scalable when the size of the table is large. Storing index to separate DHT network can be a candidate for the improvement of the current design.

Implementation of concurrency control and Restful user interface would also be possible.

References

- [1] Hari Balakrishnan et al. “Looking up data in P2P systems”. In: *Commun. ACM* 46 (2003), pp. 43–48. ISSN: 0001-0782. DOI: <http://doi.acm.org/10.1145/606272.606299>. URL: <http://doi.acm.org/10.1145/606272.606299>.
- [2] Ashwin R. Bharambe, Mukesh Agrawal, and Srinivasan Seshan. “Mercury: supporting scalable multi-attribute range queries”. In: *SIGCOMM Comput. Commun. Rev.* 34.4 (2004), pp. 353–366. ISSN: 0146-4833. DOI: <http://doi.acm.org/10.1145/1030194.1015507>.
- [3] Ion Stoica et al. “Chord: a scalable peer-to-peer lookup protocol for internet applications”. In: *IEEE/ACM Trans. Netw.* 11.1 (2003), pp. 17–32. ISSN: 1063-6692. DOI: <http://dx.doi.org/10.1109/TNET.2002.808407>.
- [4] Francesco Cesarini and Simon Thompson. *ERLANG Programming*. O’Reilly Media, Inc., 2009. ISBN: 0596518188, 9780596518189.
- [5] Joe Armstrong. *Programming Erlang: Software for a Concurrent World*. Pragmatic Bookshelf, 2007. ISBN: 193435600X, 9781934356005.
- [6] Adam Jacobs. “The Pathologies of Big Data”. In: *Queue* 7.6 (2009), pp. 10–19. ISSN: 1542-7730. DOI: <http://doi.acm.org/10.1145/1563821.1563874>.
- [7] Hiroshi ESAKI. *Peer to Peer Textbook*. Japanese. Tokyo, JAPAN: An Impress Group Company, 2008. ISBN: 978-4-8443-2504-8.
- [8] Ralf Steinmetz and Klaus Wehrle. *Peer-to-Peer Systems and Applications (Lecture Notes in Computer Science)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005. ISBN: 354029192X.
- [9] John Risson and Tim Moors. “Survey of research towards robust peer-to-peer networks: search methods”. In: *Comput. Netw.* 50.17 (2006), pp. 3485–3521. ISSN: 1389-1286. DOI: <http://dx.doi.org/10.1016/j.comnet.2006.02.001>.
- [10] John Buford, Heather Yu, and Eng Keong Lua. *P2P Networking and Applications*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008. ISBN: 0123742145, 9780123742148.
- [11] Abhishek Gupta, Divyakant Agrawal, and Amr El Abbadi. “Approximate Range Selection Queries in Peer-to-Peer Systems”. In: *CIDR*. 2003.
- [12] Artur Andrzejak and Zhichen Xu. *Scalable, Efficient Range Queries for Grid Information Services*. Washington, DC, USA. 2002. URL: <http://portal.acm.org/citation.cfm?id=824472.825527>.
- [13] Sylvia Ratnasamy et al. *Range Queries over DHTs*. Tech. rep. IRB-TR-03-009. Intel Research Berkeley, 2003.

- [14] P. A. Felber et al. “Data Indexing and Querying in DHT Peer-to-Peer Networks”. In: *In Proceedings of ICDCS 2004*. 2004.
- [15] Seth Gilbert and Nancy Lynch. “Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant web services”. In: *SIGACT News* 33.2 (2002), pp. 51–59. ISSN: 0163-5700. DOI: <http://doi.acm.org/10.1145/564585.564601>.
- [16] D. Eastlake 3rd and P. Jones. *US Secure Hash Algorithm 1 (SHA1)*. United States. 2001.
- [17] Sylvia Ratnasamy et al. “A scalable content-addressable network”. In: *SIGCOMM ’01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. San Diego, California, United States: ACM, 2001, pp. 161–172. ISBN: 1-58113-411-8. DOI: <http://doi.acm.org/10.1145/383059.383072>.
- [18] Antony I. T. Rowstron and Peter Druschel. “Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems”. In: *Middleware ’01: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*. London, UK: Springer-Verlag, 2001, pp. 329–350. ISBN: 3-540-42800-3.
- [19] Ben Y. Zhao, John D. Kubiatowicz, and Anthony D. Joseph. *Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and*. Tech. rep. UCB/CSD-01-1141. Berkeley, CA, USA 2001.
- [20] Petar Maymounkov and David Mazières. “Kademlia: A Peer-to-Peer Information System Based on the XOR Metric”. In: *IPTPS ’01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*. London, UK: Springer-Verlag, 2002, pp. 53–65. ISBN: 3-540-44179-4.
- [21] Vasanth Bala, Evelyn Duesterwald, and Sanjeev Banerjia. “Dynamo: a transparent dynamic optimization system”. In: *PLDI ’00: Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation*. Vancouver, British Columbia, Canada: ACM, 2000, pp. 1–12. ISBN: 1-58113-199-2. DOI: <http://doi.acm.org/10.1145/349299.349303>.
- [22] Ronald L. Rivest. *The MD5 Message-Digest Algorithm (RFC 1321)*. <http://www.ietf.org/rfc/rfc1321.txt?number=1321>.
- [23] National institute of standards and technology. *FIPS 180-2, Secure Hash Standard, Federal Information Processing Standard (FIPS), Publication 180-2*. Tech. rep. DEPARTMENT OF COMMERCE, 2002. URL: <http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf>.
- [24] *Erlang Programming Language, Official Web Site*. This is an electronic document. Date of publication: [Date unavailable]. Date retrieved: October 16, 2010. Date last modified: October 7, 2010. 2010. URL: <http://www.erlang.org/index.html>.

Appendices

A Chord Algorithms

This section explains the algorithms of Chord [3] excerpted from the original paper. Figure 21 shows *create* and *join* algorithms. *create* and *join* algorithms set its predecessor and successor and do not notify about itself to other nodes. successor and predecessor are maintained by *stabilize* and *notify* functions. Note that a node is supposed to know at least one bootstrap node to join an identification circle. A node first asks this bootstrap node its successor in the *join* algorithm.

```
//ask node n to find the successor of id
n.find_successor(id)
  if(id ∈ (n, successor])
    return successor;
  else
    n' = closest_preceding_node(id);
    return n'.find_successor(id)

//search the local table for the highest predecessor of id
n.closest_preceding_node(id)
  for i = m downto 1
    if(finger[i] ∈ (n, id))
      return finger[i];
  return n;
```

Figure 20: Key Lookup using finger table

Figure 22 shows *stabilize* and *notify* algorithms. They are called periodically and check newly joined nodes. *stabilize* asks its successor about successor's predecessor and decides if the predecessor should be its new successor. *stabilize* also notifies its successor about the node. This process shows why predecessor is necessary to maintain a successor.

```

//create a new Chord ring
n.create()
    predecessor = nil;
    successor = n;

//join a Chord ring containing node n'
n.join(n')
    predecessor = nil;
    successor = n'.find_successor(n);

```

Figure 21: Pseudo code for Stabilization - create and join

```

//called periodically. verifies n's immediate
//successor, and tells the successor about n.
n.stabilize()
    x = successor.predecessor;
    if( $x \in (n, \text{successor})$ )
        successor = x;
    successor.notify(n);

//n' thinks it might be our predecessor
n.notify(n')
    if(predecessor is nil or  $n' \in (\text{predecessor}, n)$ )
        predecessor = n';

```

Figure 22: Pseudo code for Stabilization - stabilize and notify

Figure 23 shows *fix_finger* and *check_predecessor* algorithms. They are called periodically and maintain a finger table. The *fix_finger* algorithm incorporates a newly joined node to its finger table. The *check_predecessor* algorithm clears predecessor if it fails in order to accept a new predecessor in the *notify* algorithm.

```

//called periodically. refreshes finger table entries
//next stores the index of the next finger to fix
n.fix_fingers()
  next = next+1;
  if(next > m)
    next = 1;
  finger[next] = find_successor(n + 2next-1);

//called periodically. checks whether predecessor has failed.
n.check_predecessor()
  if(predecessor has failed)
    predecessor = nil;

```

Figure 23: Pseudo code for Stabilization - fix_fingers and check_predecessor

```

//called periodically. verifies n's immediate
//successor, and tells the successor about n.
n.stabilize()
  //successor is the first alive ndoe in successor list
  n.successor = first_alive_node_in_successor_list();

  //maintain successor list
  //copies successor list of successor, append it to successor without last element
  ss = successor.successor_list;
  n.successor_list = head(n.successor_list) ++ reverse(tail(reverse(ss)));

  x = successor.predecessor;
  if(x ∈ (n, successor))
    successor = x;
  successor.notify(n);

```

Figure 24: Pseudo code for Stabilization - stabilize and notify

B Sample Erlang Application

The following is the problem statement of the sample Erlang application. “Create N processes in a ring. Send a message round the ring M times so that a total of $N * M$ messages get sent” [5, p. 150]. Furthermore, each process is created on a node chosen randomly from a given node list. Thus, the processes run across a network and messages are also sent across the network.

First, *start_ring* user function is written to make N processes on nodes as shown in List 5. This function (1) creates N processes on nodes and (2) initiate message circulation. To create a process on an arbitrary node, *spawn* function is used. By passing it a node name, a module name, and a function name, it creates a process by calling that function in the module on the node, and returns process ID. In this example, a new process is created with *loop* user function as shown in List 6. Creating a new process on another node using other programming language is not as easy. The *register_name* function registers the process name. The registered name can be referenced from other processes across a network, as following:

```
global:send('1', {print, Msg, N, M}),
```

It sends a message $\{print, Msg, N, M\}$ to the process of registered name ‘1’. The process ‘1’ can exist on another node across a network. To send a message to another process on a different node, only one line is necessary with Erlang. There is no need to open or close socket.

List 5: Sample Erlang program: start_ring function

```

1 %% when N == Cur, it means N - 1 processes have created, so
2 %% this is the last process to make
3 start_ring(N, M, Msg, Cur, NodeList) when N == Cur ->
4   Node = random_elem(NodeList), % chose a node from NodeList
5   Name = list_to_atom(integer_to_list(Cur)),
6
```

```

7  % generate a process with Name as it's registered name
8  global:register_name(Name, spawn(Node,
9                                ?MODULE,
10                               loop, [Name, '1', N]
11                               )
12                               ),
13  % now since all N processes are created,
14  % let's start circulating messages M times
15  global:send('1', {print, Msg, N, M}),
16  ok;
17 start_ring(N, M, Msg, Cur, NodeList) ->
18   Node = random_elem(NodeList),
19   Name = list_to_atom(integer_to_list(Cur)),
20   Next = list_to_atom(integer_to_list(Cur + 1)),
21   global:register_name(Name, spawn(Node,
22                                   ?MODULE,
23                                   loop, [Name, Next, N]
24                                   )
25                                   ),
26  % repeat N times to create N processes..
27  start_ring(N, M, Msg, Cur + 1, NodeList).

```

When a new process is created, it executes *loop* user function. This function waits to receive and forward messages and maintains process state This (process name), Next (process name to forward message), and ProcNum (total number of processes in system). If messages are sent in parallel to a process, the process receives messages one by one. When the process receives {stop, N} message, it stops by not calling itself.

List 6: Sample Erlang program: loop function

```

1  %% process to receive messages and forwards
2  %% process state
3  %%   This: Name of process
4  %%   Next: Name of next process
5  %%   ProcNum: Total # of processes in entire system
6  loop(This, Next, ProcNum) ->
7  receive
8      % receives the last message.
9      {print, _Msg, _N, 0} ->
10         io:format("Msg Complete.\n", []),
11         loop(This, Next, ProcNum);
12
13     % message round M ends, initiate M -1 round
14     {print, Msg, 1, M} ->

```



```

15     global:send(Next, {print, Msg, ProcNum, M - 1}),
16     io:format("Node:[~p] Proc:[~p] Msg:[~p] Round#:[~p]~n",
17               [node(), This, Msg, M]),
18     loop(This, Next, ProcNum);
19
20     % receives a message, forwarded to next process
21     {print, Msg, N, M} ->
22         io:format("Node:[~p] Proc:[~p] Msg:[~p] Round#:[~p]~n",
23                   [node(), This, Msg, M]),
24         global:send(Next, {print, Msg, N - 1, M}),
25         loop(This, Next, ProcNum);
26
27     % receives a stop messages.
28     % if this is not the last process, forward message
29     {stop, N} ->
30         io:format("Node:[~p] Proc:[~p] Good bye.~n",
31                   [node(), This]),
32         case N == ProcNum of
33             true -> ok;
34             false -> global:send(Next, {stop, N + 1})
35         end;
36     Other ->
37         io:format("Node:[~p] Proc:[~p] error, received:[~p]~n",
38                   [node(), This, Other]),
39         loop(This, Next, ProcNum)
40     end.

```

List 7 shows user API. The *start* function calls the *start_ring* function. The *stop* function stops all processes by sending {stop, 1} message to the '1' process. The *start_para* function does same thing as the *start_ring* function, but further it initiates the *start_ring* function on every node in NodeList list passed to it as an argument. The *rpc:call* function is used to call function on remote nodes.

List 7: Sample Erlang program: user API

```

1  %% start a ring, and send it messages to
2  %% circulate M times on nodes in NodeList
3  start(N,M,Msg,NodeList) ->
4      start_ring(N, M, Msg, 1, NodeList).
5
6  %% stop all processes
7  stop() ->
8      global:send('1', {stop, 1}),
9      ok.
10
11 %% start a ring, and circulate plural messages
12 %% in parallel
13 start_para(N,M,Msg,NodeList) ->
14     start_para_in(N,M,Msg,NodeList,NodeList).
15
16 start_para_in(_,_,_,_,[]) -> ok;
17 start_para_in(N,M,Msg,NodeList,[Node|Rest]) ->
18     % call start function of ring module on node Node
19     rpc:call(Node, ring, start,
20              [N, M, {Msg, originates, Node}, NodeList]),
21     start_para_in(N,M,Msg,NodeList,Rest).

```

List 8 shows how this program should work. It uses 3 nodes on 3 machines to send messages to 5 processes. First, it creates 5 processes on 3 nodes, then circulates a message 2 times around the ring, then stops all processes. Second, it creates 5 processes on 3 nodes, initiates message circulation on all 3 nodes in parallel, then stops all processes again. The whole program list is shown in List 9.

List 8: Sample Erlang program: output example

```

1 % starting Erlang node with name test1. Other nodes are started in a same manner
2 hirotnk@netlab3 ~ $ erl -sname test1 -setcookie abc
3 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
4
5 Eshell V5.6.3 (abort with ^G)
6 (test1@netlab3)1> c(ring). % compile the module
7 {ok,ring}
8 % connecting this node to other nodes
9 (test1@netlab3)2> net_adm:ping(test2@netlab4).
10 pong
11 (test1@netlab3)3> net_adm:ping(test3@netlab1).
12 pong
13 (test1@netlab3)4>
14 (test1@netlab3)4>
15 % start 5 processes on nodes randomly chosen from node list, then circulate
16 % a message 2 times
17 (test1@netlab3)4> ring:start(5,2,hi,[test1@netlab3,test2@netlab4,test3@netlab1]).
18 % proc '1' is on test1@netlab3
19 Node:[test1@netlab3] Proc:['1'] Msg:[hi] Round#:[2]
20 ok
21 % proc '2' is on test2@netlab4
22 Node:[test2@netlab4] Proc:['2'] Msg:[hi] Round#:[2]
23 % proc '3' is on test3@netlab1
24 Node:[test3@netlab1] Proc:['3'] Msg:[hi] Round#:[2]
25 % proc '4' is on test3@netlab1
26 Node:[test3@netlab1] Proc:['4'] Msg:[hi] Round#:[2]
27 Node:[test1@netlab3] Proc:['1'] Msg:[hi] Round#:[1]
28 % proc '5' is on test2@netlab4
29 Node:[test2@netlab4] Proc:['5'] Msg:[hi] Round#:[2]
30 Node:[test2@netlab4] Proc:['2'] Msg:[hi] Round#:[1]
31 Node:[test3@netlab1] Proc:['3'] Msg:[hi] Round#:[1]
32 Node:[test3@netlab1] Proc:['4'] Msg:[hi] Round#:[1]
33 Msg Complete.
34 Node:[test2@netlab4] Proc:['5'] Msg:[hi] Round#:[1]
35 % starting all 5 processes
36 (test1@netlab3)5> ring:stop().
37 Node:[test1@netlab3] Proc:['1'] Good bye.
38 ok
39 Node:[test2@netlab4] Proc:['2'] Good bye.
40 Node:[test3@netlab1] Proc:['3'] Good bye.
41 Node:[test3@netlab1] Proc:['4'] Good bye.
42 Node:[test2@netlab4] Proc:['5'] Good bye.
43 (test1@netlab3)6>
44 (test1@netlab3)6>
45 (test1@netlab3)6>
46 % next, circulate a message 2 times through 5 processes, but messages are

```

```

47 % initiated on each of 3 nodes.
48 % therefore, there are total 5 * 2 * 3 = 30 messages get sent.
49 (test1@netlab3)6> ring:start_para(5,2,hi,[test1@netlab3,test2@netlab4,test3@netlab1]).
50 Node:[test1@netlab3] Proc:['1'] Msg:[{hi,originates,test1@netlab3}] Round#:[2]
51 Node:[test2@netlab4] Proc:['2'] Msg:[{hi,originates,test1@netlab3}] Round#:[2]
52 Node:[test3@netlab1] Proc:['3'] Msg:[{hi,originates,test1@netlab3}] Round#:[2]
53 Node:[test3@netlab1] Proc:['4'] Msg:[{hi,originates,test1@netlab3}] Round#:[2]
54 Node:[test1@netlab3] Proc:['1'] Msg:[{hi,originates,test1@netlab3}] Round#:[1]
55 Node:[test2@netlab4] Proc:['5'] Msg:[{hi,originates,test1@netlab3}] Round#:[2]
56 Node:[test2@netlab4] Proc:['2'] Msg:[{hi,originates,test1@netlab3}] Round#:[1]
57 Node:[test3@netlab1] Proc:['3'] Msg:[{hi,originates,test1@netlab3}] Round#:[1]
58 Node:[test3@netlab1] Proc:['4'] Msg:[{hi,originates,test1@netlab3}] Round#:[1]
59 Msg Complete.
60 Node:[test2@netlab4] Proc:['5'] Msg:[{hi,originates,test1@netlab3}] Round#:[1]
61 Node:[test1@netlab3] Proc:['1'] Msg:[{hi,originates,test2@netlab4}] Round#:[2]
62 Node:[test2@netlab4] Proc:['2'] Msg:[{hi,originates,test2@netlab4}] Round#:[2]
63 Node:[test3@netlab1] Proc:['3'] Msg:[{hi,originates,test2@netlab4}] Round#:[2]
64 Node:[test3@netlab1] Proc:['4'] Msg:[{hi,originates,test2@netlab4}] Round#:[2]
65 Node:[test1@netlab3] Proc:['1'] Msg:[{hi,originates,test2@netlab4}] Round#:[1]
66 Node:[test2@netlab4] Proc:['5'] Msg:[{hi,originates,test2@netlab4}] Round#:[2]
67 Node:[test2@netlab4] Proc:['2'] Msg:[{hi,originates,test2@netlab4}] Round#:[1]
68 Node:[test3@netlab1] Proc:['3'] Msg:[{hi,originates,test2@netlab4}] Round#:[1]
69 Node:[test3@netlab1] Proc:['4'] Msg:[{hi,originates,test2@netlab4}] Round#:[1]
70 Node:[test2@netlab4] Proc:['5'] Msg:[{hi,originates,test2@netlab4}] Round#:[1]
71 Msg Complete.
72 Node:[test1@netlab3] Proc:['1'] Msg:[{hi,originates,test3@netlab1}] Round#:[2]
73 ok
74 Node:[test2@netlab4] Proc:['2'] Msg:[{hi,originates,test3@netlab1}] Round#:[2]
75 Node:[test3@netlab1] Proc:['3'] Msg:[{hi,originates,test3@netlab1}] Round#:[2]
76 Node:[test3@netlab1] Proc:['4'] Msg:[{hi,originates,test3@netlab1}] Round#:[2]
77 Node:[test2@netlab4] Proc:['5'] Msg:[{hi,originates,test3@netlab1}] Round#:[2]
78 Node:[test1@netlab3] Proc:['1'] Msg:[{hi,originates,test3@netlab1}] Round#:[1]
79 Node:[test2@netlab4] Proc:['2'] Msg:[{hi,originates,test3@netlab1}] Round#:[1]
80 Node:[test3@netlab1] Proc:['3'] Msg:[{hi,originates,test3@netlab1}] Round#:[1]
81 Node:[test3@netlab1] Proc:['4'] Msg:[{hi,originates,test3@netlab1}] Round#:[1]
82 Node:[test2@netlab4] Proc:['5'] Msg:[{hi,originates,test3@netlab1}] Round#:[1]
83 Msg Complete.
84 % stopping all 5 processes
85 (test1@netlab3)7> ring:stop().
86 Node:[test1@netlab3] Proc:['1'] Good bye.
87 ok
88 Node:[test2@netlab4] Proc:['2'] Good bye.
89 Node:[test3@netlab1] Proc:['3'] Good bye.
90 Node:[test3@netlab1] Proc:['4'] Good bye.
91 Node:[test2@netlab4] Proc:['5'] Good bye.
92 (test1@netlab3)8>

```

Following is the entire list of code.

List 9: Sample Erlang program

```

1 -module(ring).
2 -export([start/4, stop/0, start_para/4]).
3 -export([loop/3, start_ring/5]).
4
5
6 %% start a ring, and send it messages to
7 %% circulate M times on nodes in NodeList
8 start(N,M,Msg,NodeList) ->

```

```

9   start_ring(N, M, Msg, 1, NodeList).
10
11  %% stop all processes
12  stop() ->
13    global:send('1', {stop, 1}),
14    ok.
15
16  %% start a ring, and circulate plural messages
17  %% in parallel
18  start_para(N,M,Msg,NodeList) ->
19    start_para_in(N,M,Msg,NodeList,NodeList).
20
21  start_para_in(_,_,_,_,[]) -> ok;
22  start_para_in(N,M,Msg,NodeList,[Node|Rest]) ->
23    % call start function of ring module on node Node
24    rpc:call(Node, ring, start,
25      [N, M, {Msg, originates, Node}, NodeList]),
26    start_para_in(N,M,Msg,NodeList,Rest).
27
28  %%-----
29  %% @spec(start_ring(N::integer(),
30  %%           M::integer(),
31  %%           Msg::any(),
32  %%           Cur::integer(),
33  %%           NodeList::list()) -> ok).
34  %% @doc
35  %%   start_ring function creates a ring of N processes, then send
36  %%   messages to the ring so that the message circulates the ring
37  %%   M times.
38  %%   Processes are created on nodes chosen from NodeList at random.
39  %%
40  %% @end
41  %%-----
42  %% when N == Cur, it means N - 1 processes have created, so
43  %% this is the last process to make
44  start_ring(N, M, Msg, Cur, NodeList) when N == Cur ->
45    Node = random_elem(NodeList), % chose a node from NodeList
46    Name = list_to_atom(integer_to_list(Cur)),
47
48    % generate a process with Name as it's registered name
49    global:register_name(Name, spawn(Node,
50      ?MODULE,
51      loop, [Name, '1', N]
52    )
53  ),
54  % now since all N processes are created,
55  % let's start circulating messages M times
56  global:send('1', {print, Msg, N, M}),
57  ok;
58  start_ring(N, M, Msg, Cur, NodeList) ->
59    Node = random_elem(NodeList),
60    Name = list_to_atom(integer_to_list(Cur)),
61    Next = list_to_atom(integer_to_list(Cur + 1)),
62    global:register_name(Name, spawn(Node,
63      ?MODULE,
64      loop, [Name, Next, N]
65    )
66  ),
67  % repeat N times to create N processes..
68  start_ring(N, M, Msg, Cur + 1, NodeList).

```

```

69
70 %%
71 %% process to receive messages and forwards
72 %% process state
73 %% This: Name of process
74 %% Next: Name of next process
75 %% ProcNum: Total # of processes in entire system
76 loop(This, Next, ProcNum) ->
77     receive
78         {print, _Msg, _N, 0} ->
79             io:format("Msg Complete.\n", []),
80             loop(This, Next, ProcNum);
81         {print, Msg, 1, M} ->
82             global:send(Next, {print, Msg, ProcNum, M - 1}),
83             io:format("Node:[~p] Proc:[~p] Msg:[~p] Round#:[~p]\n",
84                 [node(), This, Msg, M]),
85             loop(This, Next, ProcNum);
86         {print, Msg, N, M} ->
87             io:format("Node:[~p] Proc:[~p] Msg:[~p] Round#:[~p]\n",
88                 [node(), This, Msg, M]),
89             global:send(Next, {print, Msg, N - 1, M}),
90             loop(This, Next, ProcNum);
91         {stop, N} ->
92             io:format("Node:[~p] Proc:[~p] Good bye.\n",
93                 [node(), This]),
94             case N == ProcNum of
95                 true -> ok;
96                 false -> global:send(Next, {stop, N + 1})
97             end;
98         Other ->
99             io:format("Node:[~p] Proc:[~p] error, received:[~p]\n",
100                 [node(), This, Other]),
101             loop(This, Next, ProcNum)
102     end.
103
104 random_elem(List) ->
105     Max = length(List),
106     lists:nth(random:uniform(Max), List).

```

C Range Query Specification and Example

Range query of Harmonia allows relational operators ($=$, $!$, $<$, $>$, \leq , \geq) and logical operators (and, or). Query condition grammar is expressed in EBNF (Extended Backus-Naur Form) as following:

List 10: Query condition of range query

```

1   Sts   = Exp , { 'or' Exp };
2   Exp   = Factor , { 'and' Factor };
3   Factor = Fname RelOp Term | '(' , Sts , ')';
4   Fname  = Char , { Char | Number };
5   Term   = Atom | Number | String;
6   RelOp  = '=' | '!=' | '<=' | '>=' | '>' | '<';
7   Number = Digit , { Digit };
8   String = '[' , { Char | Digit } , ']';
9   Atom   = ''' , { Char | Digit } , ''';
10  Digit  = '0-9';
11  Char   = 'a-Z';

```

```

29                                     {"Fld3", filename2}])).
30 {ok,5}
31 (xxx_node@ubu)41> hm_cli:rstore("Domain1",
32                                "Tbl5",
33                                [{"Fld1", 300},
34                                {"Fld2", "tex of field2"},
35                                {"Fld3", filename3}])).
36 {ok,5}
37
38 %% get where Fld1 >= 200
39 (xxx_node@ubu)42> hm_cli:rget("Domain1", "Tbl5", "Fld1 >= 200").
40 {ok,[[200,"text of field2",filename2],
41      [300,"tex of field2",filename3]]}
42
43 %% get where Fld1 >= 200 and Fld2 == "tex of field2"
44 (xxx_node@ubu)43> hm_cli:rget("Domain1", "Tbl5", "Fld1 >= 200 and Fld2 == [tex of field2]").
45 {ok,[[300,"tex of field2",filename3]]}
46 (xxx_node@ubu)44>

```

D Implementation of Harmonia

This section describes the implementation details of Harmonia. Harmonia is developed with Erlang/OTP. Harmonia system consists of *Harmonia nodes*. Harmonia node consists of Chord based DHT storage with range query. It communicates with other Harmonia nodes using Chord protocol and provides data storage over DHT. Harmonia is based on the distributed environment of Erlang. Therefore multiple nodes can run on a physical machine and each node can communicate with other nodes across the network transparently. Since Harmonia is Chord based system, a node can join or leave without any manual configuration. When nodes join or leave the system, the routing information is updated dynamically, and data is not lost because it is replicated. However, Harmonia does not support data migration when a node leaves. Therefore if r successive nodes on identifier circle fail or leave, the data will be lost where r denotes the degree of replication.

D.1 Supervisor Tree of Harmonia

Harmonia node consists of a supervisor tree in which worker processes are supervised by a supervisor process using OTP supervisor behavior. The advantage is that Harmonia node can maintain integrity and assure it does not run without any of its worker processes. Figure 25 shows supervisor tree of Harmonia. The worker processes do not necessarily have a one-to-one relationship with functions. Sometimes two worker processes correspond to a function, and some functions are provided as library modules, not processes. Harmonia works by exchanging messages within or across nodes. The description and type of OTP behavior of the processes are shown in table 23.

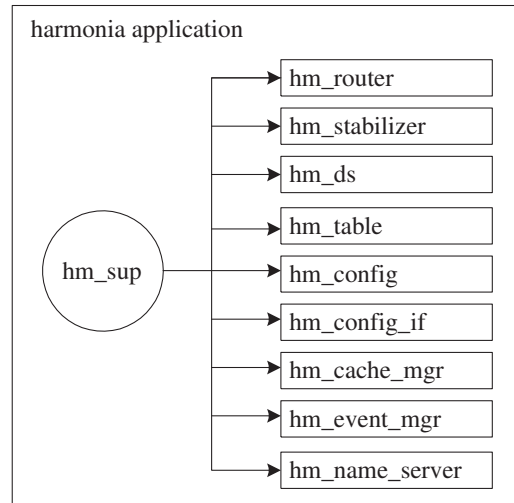


Figure 25: Supervisor tree of Harmonia

D.2 Object Relationship of Harmonia

Harmonia works by exchanging messages between nodes. However messages are hidden in APIs exported by modules. The relationship between processes and nodes are shown in Figure 26. In order to focus on the main functional aspect, the supervisor process and library modules as well as minor APIs are omitted in Figure 26. Client I/F is the entry point of Harmonia and it provides a set of user commands. Therefore, users need to connect at least one Harmonia node to use it.

In this Figure, three main layers are described. First, the client layer consists of Client I/F process. Second, the data storage layer consists of a data store process, a table management process, and a Cache management process. Third, the DHT layer consists of a router process and a stabilizer process. The client layer communicates with the lower data storage layer. Data storage layer communicates with the lower DHT layer. Each data storage layer and DHT layer communicates with the same layer of other nodes. The implementation of each layer is encaps-

Module Name	OTP behavior	Description
harmonia	application	This process wraps up modules as package
hm_sup	supervisor	This process manages worker processes to start/restart/terminate
hm_router	gen_server	This process maintains routing information and provides <i>lookup</i> API
hm_stabilizer	gen_fsm	This process periodically called and keep routing information up to date
hm_ds	gen_server	This process provides <i>store</i> , <i>get</i> , <i>rstore</i> , <i>rget</i> API
hm_table	gen_server	This process maintains table configuration data
hm_config	gen_server	This process maintains configuration data locally
hm_config_if	gen_server	This process provides interface of configuration data globally
hm_event_mgr	gen_event	This process manages logging events
hm_cache_mgr	gen_fsm	This process manages cache table
hm_name_server	gen_server	This process manages node names, and only bootstrap node has this process

Table 23: Process list of Harmonia

sulated by exported APIs of modules and messages. Therefore, switching the implementation logic is possible. For example, the data storage layer can be switched from current ETS implementation to PostgreSQL implementation. Similarly the DHT layer can be switched from Chord implementation to Kademlia [20] implementation.

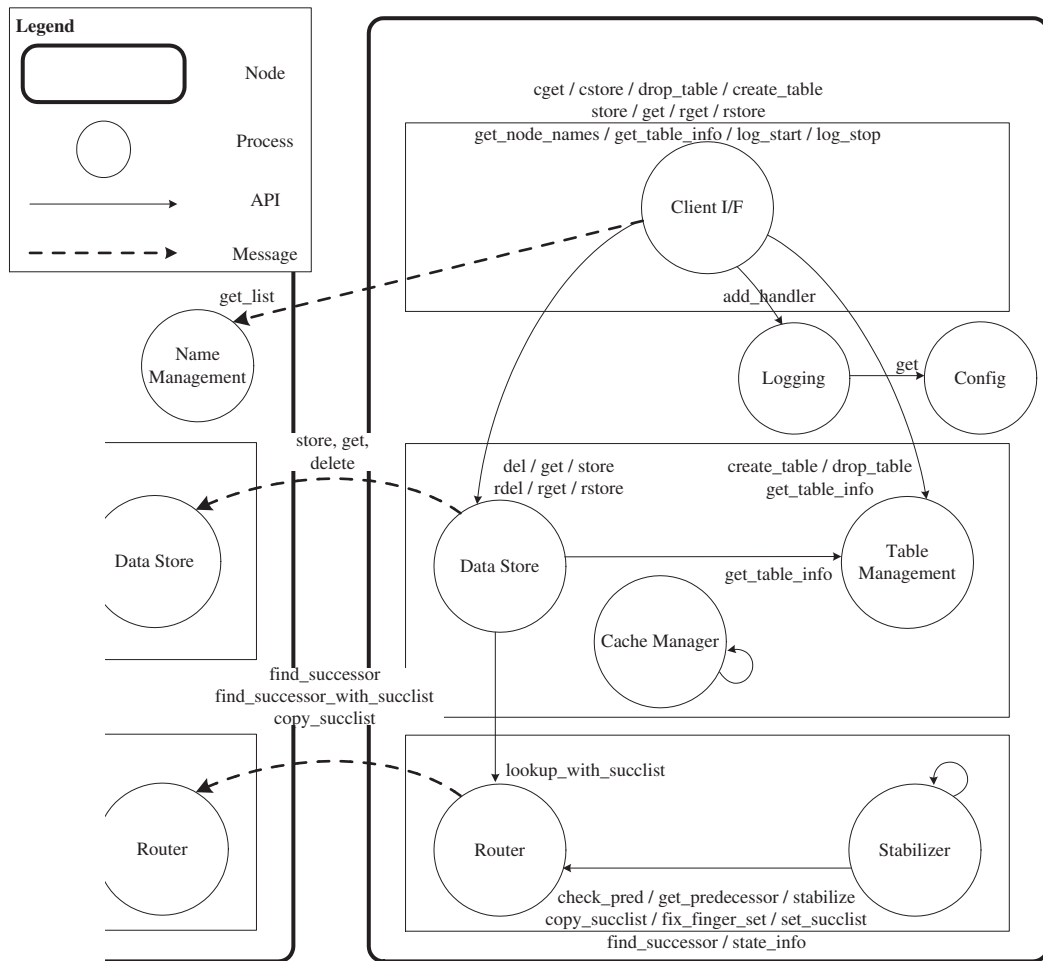


Figure 26: Object relationship of Harmonia

D.3 Directory Structure and Startup Configuration

D.3.1 Directory structure

Harmonia uses application behavior of OTP to make it packaged. Application behavior specifies a specific layout of directory [4, p. 281]. Detailed manual of Application behavior is available on Erlang Official website [24]. The directory layout is shown in Figure 27.

```

Harmonia
|-- harmonia.config
|-- ebin/
|-- include/
|-- priv/
|-- scripts/
|-- src/
|-- doc/
`-- test/

```

Figure 27: Directory structure of Harmonia

ebin contains compiled byte code files (*beam file*) and application resource file.

include contains include files.

priv is an optional directory and not being used in Harmonia.

src contains the source code of all the Erlang modules in the application.

doc contains documents (currently not used in Harmonia).

test contains test modules (Harmonia specific).

D.3.2 Application Resource File

Application resource file defines application resources and dependencies. In a resource file, *environment variables* can be specified and an application can access them. Furthermore,

these parameters can be overridden when an Erlang node is started. Environment variables of Harmonia are described in Table 24.

Name	Description
node_type	join or create. If a node is the first in Chord network, create is specified. A node initializes itself according these types.
name	Name of a Harmonia node. Harmonia uses this name as an node identifier instead of IP address.
sname	Name of a Erlang node
root	Name of a bootstrap Harmonia node. Specified when a node type is join
root_node	Name of a bootstrap Erlang node. Specified when a node type is join
logfile	Logfile name
logfile_ext	Logfile extension
logdir	Logfile directory

Table 24: Environment variables of Harmonia

D.3.3 Startup of Harmonia

A Harmonia node is started using the following command line: *-pa* adds a path to the directory. *-harmonia* specifies environment variables of an application and enables overrides. *-setcookie* specifies cookie value. Erlang nodes need to have the same cookie value to connect each other. *-sname* specifies the name of the Erlang node. Below is an example of a bootstrap node start. *-detached* executes the Erlang runtime system detached from the system console. *-run* executes a specified function of the Module. In this example, start function of `hm_edge` module is executed when system starts.

List 12: Start up command: create

```

1 erl -pa ebin -config harmonia \
2   -detached \
3   -run hm_edge start \
4   -harmonia node_type 'create' \
5       name foo \
6       sname 'node_foo@ubu' \
7   -setcookie harmonia_cookie -sname 'node_foo@ubu'

```

Next is an example of node start to join an existing Chord network.

List 13: Start up command: join

```

1 erl -pa ebin -config harmonia \
2   -detached \
3   -run hm_edge start \
4   -harmonia node_type 'join' \
5       name bar \
6       sname 'bar_node@ubu' \
7       root_node 'node_foo@ubu' \
8   -setcookie harmonia_cookie -sname 'bar_node@ubu'

```

The `hm_edge` module calls `harmonia applicatoin`, then `harmonia application` calls `hm_sup module`.

The supervisor module starts Harmonia node by starting worker processes.

E Harmonia Source Files

List 14: Makefile

```

1 #####
2 # Generic make script for compiling erlang code #
3 # The environment variable $ERLHOME has to be set to where erlang/OTP #
4 # is installed #
5 # Compiles the code into a ebin dir. relative to the source dir. #
6 # (../ebin) #
7 #####
8 #Compiles the code into a ebin dir. relative to the source dir.
9 EBIN = ../ebin
10 EDBG = -Ddebug
11 CFLG = -W +warn_unused_vars +warn_unused_import +debug_info +native +"{hipec, [o3]}"
12
13 ifeq ($(CFG),nolog)
14 NOLOG = -Dnolog
15 else
16 NOLOG =
17 endif
18
19 ERL = erl
20 PA = /usr/local/lib/erlang/lib/eunit-2.1.5/ebin
21 GEN = beam
22 ERIC_EMULATOR = erl -boot start_clean
23 PATH= .:$(ERLHOME)/bin:/bin:/usr/bin:
24 SOURCE = harmonia.erl \
25         hm_cache.erl \
26         hm_cache_mgr.erl \
27         hm_cli.erl \
28         hm_config.erl \
29         hm_config_if.erl \
30         hm_ds.erl \
31         hm_edge.erl \
32         hm_event_mgr.erl \
33         hm_log_h_file.erl \
34         hm_log_h_term.erl \
35         hm_misc.erl \
36         hm_name_server.erl \
37         hm_qp.erl \
38         hm_router.erl \
39         hm_stabilizer.erl \
40         hm_sup.erl \
41         hm_table.erl
42
43 TARGETS = $(SOURCE:%.erl=$(EBIN)/%.beam)
44 CODE = $(SOURCE:%.erl=$(EBIN)/%.beam)
45
46 $(EBIN)/%.beam: %.erl
47     /usr/bin/erlc -pa $(PA) $(NOLOG) $(EDBG) $(CFLG) -v -b beam -o $(EBIN) $(EFLAGS) $<
48
49 all: $(TARGETS)
50     cp harmonia.app ../ebin/harmonia.app
51
52 clean:
53     \rm -f $(CODE)
54     \rm -f ../ebin/harmonia.app
55

```



```

56
57 #####
58 # Template of for compiling erlang files #
59 # The environment variable $TOOLSHOME home has to be set to where #
60 # the generic make script is installed (erlang). #
61 #####
62 # code to compile
63
64 #Where include files are stored ".hrl"
65 EFLAGS = -I../include -I/usr/local/lib/erlang/lib/stdlib-1.16.5/
66
67
68 #####
69 # Do not edit below this line #
70 #####
71 #Include following generic make script
72 #include $(TOOLSHOME)/erlang

```

List 15: Include file

```

1  % length of the digest of crypto:sha/1
2  -ifdef(debug).
3  -define(key_bit_length, 8).
4  -else.
5  -define(key_bit_length, 160).
6  -endif.
7
8  % registered name server id
9  -define(name_server, hm_name_server).
10
11 % Interval of stabilizer
12 -define(stabilize_interval, 3000).
13
14 % Interval of fixfingers
15 -define(fixfinger_interval, 3000).
16
17 % get timeout
18 -define(TIMEOUT_GET, 30000).
19
20 % Length of finger table
21 -define(max_finger, ?key_bit_length).
22
23 % Length of successor list
24 %-define(succ_list_len, (?key_bit_length bsr 1)).
25 -define(succ_list_len, 4).
26
27 % max key value
28 -define(max_key_value, ((1 bsl ?key_bit_length) - 1)).
29
30 % logging info
31 -define(LOG_INFO, info).
32 -define(LOG_WARNING, warning).
33 -define(LOG_ERROR, error).
34
35 -ifdef(nolog).
36 -define(info_p (Fmt, RegName, Data), ok).
37 -define(warning_p (Fmt, RegName, Data), ok).
38 -define(error_p (Fmt, RegName, Data), ok).
39 -else.
40 -define(log_parts (RegName, Data), ([node(), self(), ?MODULE, ?LINE, RegName] ++ Data)).
41 -define(info_p (Fmt, RegName, Data), hm_event_mgr:log(?LOG_INFO, Fmt, ?log_parts(\
    →RegName, Data))).
42 -define(warning_p (Fmt, RegName, Data), hm_event_mgr:log(?LOG_WARNING, Fmt, ?log_parts(\
    →RegName, Data))).
43 -define(error_p (Fmt, RegName, Data), hm_event_mgr:log(?LOG_ERROR, Fmt, ?log_parts(\
    →RegName, Data))).
44 -endif.
45
46 % process prefix
47 -define(PROCESS_PREFIX, "hm_router_").
48
49 % cache
50 -define(hm_ets_cache_table, hm_ets_cache_table).
51 -define(cache_timeout, (60*60)). % default 1 hour later
52 -define(cache_limit_size, 10000). % default 10000 recs
53 -define(cache_cleanup_interval, (60 * 5 * 1000)). % 5minutes
54 -define(ets_cache_threshold_num, 1000000).
55
56

```

```
57 -record(state, {node_name, node_vector, predecessor=nil, finger=[], succlist=[], current_fix\
    →=0}).
```

List 16: harmonia.app

```

1 {application, harmonia,
2   [{description, "Harmonia - Key/Value store engine with flexible range query support"},
3    {vsns, "0.1"},
4    {modules, [
5      harmonia, hm_cache, hm_cache_mgr, hm_cli, hm_config, hm_config_if,
6      hm_ds, hm_edge, hm_event_mgr, hm_log_h_file, hm_log_h_term, hm_misc,
7      hm_name_server, hm_qp, hm_router, hm_stabilizer, hm_sup, hm_table
8    ]},
9    {registered, [
10     hm_ds, hm_event_mgr, hm_name_server, hm_router, hm_stabilizer,
11     hm_sup, hm_table, harmonia, hm_cache_mgr, hm_config, hm_config_if
12   ]},
13   {applications, [kernel, stdlib]},
14   {mod, {harmonia, []}},
15   {start_phases, []},
16   {env, [
17     {node_type,    create_or_join},
18     {name,         noname},
19     {sname,        node_name},
20     {root,         foo},
21     {root_node,    foo@node},
22     {logfile,       "harmonia_log"},
23     {logfile_ext,   ".txt"},
24     {logdir,        "log/"}
25   ]}
26 ]}.

```

List 17: `harmonia.erl`

```

1 % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2 % use this file except in compliance with the License. You may obtain a copy of
3 % the License at
4 %
5 % http://www.apache.org/licenses/LICENSE-2.0
6 %
7 % Unless required by applicable law or agreed to in writing, software
8 % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9 % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12
13 %%%-----
14 %%% File      : harmonia.erl
15 %%% Description : application to start supervisor
16 %%%
17 %%%-----
18
19 -module(harmonia).
20 -author('Yoshihiro TANAKA <hirotknkg@gmail.com>').
21 -behaviour(application).
22
23 -export([start/2, stop/1]).
24
25 start(_StartType, _StartArgs) ->
26     Env = application:get_all_env(harmonia),
27     case hm_sup:start_link(Env) of
28         {ok, Pid} -> {ok, Pid};
29         Other      -> {error, Other}
30     end.
31
32 stop(_State) ->
33     ok.

```

List 18: hm_cache.erl

```

1 % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2 % use this file except in compliance with the License. You may obtain a copy of
3 % the License at
4 %
5 % http://www.apache.org/licenses/LICENSE-2.0
6 %
7 % Unless required by applicable law or agreed to in writing, software
8 % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9 % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12
13 %%%-----
14 %%% File      : hm_cache.erl
15 %%% Description : cget/cstore interface
16 %%% The data structure of cache is following:
17 %%% Record format:
18 %%% {
19 %%%     Key           : key of this record
20 %%%     Value         : value of this record, type ::any()
21 %%%     Reference Count : this field is incremented every time
22 %%%                     the record is referenced
23 %%%     Expiration Time : indicates when this cache is expired
24 %%% }
25 %%%
26 %%% (1) data store
27 %%%     when data is stored, reference count is set to 0(zero), and
28 %%%     expiration time is set to current time + configured amount
29 %%%     of time
30 %%%
31 %%% (2) data get
32 %%%     when data is referenced, reference count is incremented by 1,
33 %%%     and expiration time is set to current time + configured amount
34 %%%     of time, then the record is update.
35 %%%
36 %%%-----
37
38 -module(hm_cache).
39 -author('Yoshihiro TANAKA <hirotknkg@gmail.com>').
40 -vsn('0.1').
41
42 -export([
43     get_cache/1,
44     del_cache/1,
45     start/0,
46     store_cache/2
47 ]).
48
49 -include("harmonia.hrl").
50
51 %%%=====
52 %%% API
53 %%%=====
54
55 start() ->
56     ets:new(?hm_ets_cache_table, [named_table, public]).
57
58 store_cache(Key, Value) ->
59     {MegaSecs, Secs, _Microsecs} = now(),

```

```

60     ets:insert(?hm_ets_cache_table,
61                 {Key, Value, 0, MegaSecs*1000000 + Secs + ?cache_timeout}),
62     ok.
63
64 %% @spec(get_cache(Key::list()/atom()/integer()) -> none|{ok, {Value::term(), Cnt::integer()}\
65         ->}}).
66 get_cache(Key) ->
67     case ets:lookup(?hm_ets_cache_table, Key) of
68     [] -> none;
69     [{Key, Value, Cnt, _}] ->
70         {MegaSecs, Secs, _Microsecs} = now(),
71         ets:update_element(
72             ?hm_ets_cache_table,
73             Key,
74             [{3,Cnt+1},{4, MegaSecs*1000000 + Secs + ?cache_timeout}]
75         ),
76         {ok, {Value, Cnt}}
77     end.
78
79 %% @spec(del_cache(Key::list()/atom()/integer()) -> {ok, Key}).
80 del_cache(Key) ->
81     ets:delete(?hm_ets_cache_table, Key),
82     {ok, Key}.

```

List 19: hm_cache_mgr.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% File      : hm_cache_mgr.erl
14 %%% Description : periodically maintain local cache with LRU manner
15 %%%
16 %%% (1) cache_cleanup_lru runs periodically, check if the number of
17 %%%     records exceeds the threshold of cache, if true, go to (2)
18 %%%     if false, sleep for configured period of time
19 %%%
20 %%% (2) perform the cleaning with following logic:
21 %%%     referenced count and expiration time is to be considered:
22 %%%
23 %%%     CheckCnt = 0
24 %%%     cleaned_record_num = 0
25 %%%     to_be_cleaned_num = total_cache_recnum - threshold_cache
26 %%%
27 %%%     while( cleaned_record_num < to_be_cleaned_num ){
28 %%%         Recset = select records where reference count == CheckCnt
29 %%%         while( cleaned_record_num < to_be_cleaned_num ){
30 %%%             if Recset.rec.expiration_time > now()
31 %%%                 delete Recset.rec
32 %%%                 cleaned_record_num++
33 %%%             end
34 %%%             if no more record in Recset then
35 %%%                 break
36 %%%             end
37 %%%         }
38 %%%         CheckCnt++
39 %%%     }
40 %%%-----
41 -module(hm_cache_mgr).
42 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
43 -behaviour(gen_fsm).
44 -vsn('0.1').
45 %% API
46 -export([
47     start_link/1,
48     stop/1
49     ]).
50 %% gen_fsm callbacks
51 -export([init/1, cache_cleanup_lru/2, handle_event/3,
52     handle_sync_event/4, handle_info/3, terminate/3, code_change/4]).
53
54 -include("harmonia.hrl").
55 -include_lib("include/ms_transform.hrl").
56
57 %%%=====
58 %% API
59 %%%=====

```



```

60 start_link(RegName) ->
61     gen_fsm:start_link({local, ?MODULE}, ?MODULE, RegName, []).
62
63 stop(RegName) ->
64     ?info_p("stop:stopping:[~p].~n", RegName, [RegName]),
65     gen_fsm:send_event({global, name(RegName)}, stop).
66
67 %%=====
68 %% gen_fsm callbacks
69 %%=====
70 %%-----
71 %% Function: init(Args) -> {ok, StateName, State} |
72 %%                          {ok, StateName, State, Timeout} |
73 %%                          ignore                               |
74 %%                          {stop, StopReason}
75 %% Description: Whenever a gen_fsm is started using gen_fsm:start/[3,4] or
76 %% gen_fsm:start_link/3,4, this function is called by the new process to
77 %% initialize.
78 %%-----
79 init(RegName) ->
80     hm_cache:start(),
81     {ok, cache_cleanup_lru, RegName, ?cache_cleanup_interval}.
82
83 %%-----
84 %% @private
85 %% @doc
86 %% Whenever a gen_fsm receives an event sent using
87 %% gen_fsm:send_all_state_event/2, this function is called to handle
88 %% the event.
89 %%
90 %% @spec handle_event(Event, StateName, State) ->
91 %%       {next_state, NextStateName, NextState} |
92 %%       {next_state, NextStateName, NextState, Timeout} |
93 %%       {stop, Reason, NewState}
94 %% @end
95 %%-----
96 handle_event(_Event, StateName, State) ->
97     {next_state, StateName, State}.
98
99 %%-----
100 %% @private
101 %% @doc
102 %% Whenever a gen_fsm receives an event sent using
103 %% gen_fsm:sync_send_all_state_event/[2,3], this function is called
104 %% to handle the event.
105 %%
106 %% @spec handle_sync_event(Event, From, StateName, State) ->
107 %%       {next_state, NextStateName, NextState} |
108 %%       {next_state, NextStateName, NextState, Timeout} |
109 %%       {reply, Reply, NextStateName, NextState} |
110 %%       {reply, Reply, NextStateName, NextState, Timeout} |
111 %%       {stop, Reason, NewState} |
112 %%       {stop, Reason, Reply, NewState}
113 %% @end
114 %%-----
115 handle_sync_event(_Event, _From, StateName, State) ->
116     Reply = ok,
117     {reply, Reply, StateName, State}.
118
119 %%-----

```

```

120 %% @private
121 %% @doc
122 %% This function is called by a gen_fsm when it receives any
123 %% message other than a synchronous or asynchronous event
124 %% (or a system message).
125 %%
126 %% @spec handle_info(Info,StateName,State)->
127 %%           {next_state, NextStateName, NextState} |
128 %%           {next_state, NextStateName, NextState, Timeout} |
129 %%           {stop, Reason, NewState}
130 %% @end
131 %%-----
132 handle_info(_Info, StateName, State) ->
133     {next_state, StateName, State}.
134
135 %%-----
136 %% @private
137 %% @doc
138 %% This function is called by a gen_fsm when it is about to
139 %% terminate. It should be the opposite of Module:init/1 and do any
140 %% necessary cleaning up. When it returns, the gen_fsm terminates with
141 %% Reason. The return value is ignored.
142 %%
143 %% @spec terminate(Reason, StateName, State) -> void()
144 %% @end
145 %%-----
146 terminate(Reason, StateName, State) ->
147     ?info_p("terminate:Reason:[~p] StateName:[~p], State:[~p]~n", none, [Reason, StateName, \
        ->State]),
148     ok.
149
150 %%-----
151 %% Function:
152 %% state_name(Event, State) -> {next_state, NextStateName, NextState}|
153 %%                               {next_state, NextStateName,
154 %%                               NextState, Timeout} |
155 %%                               {stop, Reason, NewState}
156 %% Description:There should be one instance of this function for each possible
157 %% state name. Whenever a gen_fsm receives an event sent using
158 %% gen_fsm:send_event/2, the instance of this function with the same name as
159 %% the current state name StateName is called to handle the event. It is also
160 %% called if a timeout occurs.
161 %%-----
162 cache_cleanup_lru(timeout, RegName) ->
163     RowCnt = ets:select_count(?hm_ets_cache_table, ets:fun2ms(fun(_)->true end)),
164     case RowCnt > ?ets_cache_threshold_num of
165         true ->
166             ?info_p("Current Num:[~p]~n", RegName, [RowCnt]),
167             clean_up_worker(RowCnt - ?ets_cache_threshold_num);
168         false -> ok
169     end,
170     {next_state, cache_cleanup_lru, RegName, ?cache_cleanup_interval}.
171
172 %%-----
173 %% @private
174 %% @doc
175 %% Convert process state when code is changed
176 %%
177 %% @spec code_change(OldVsn, StateName, State, Extra) ->
178 %%           {ok, StateName, NewState}

```

```

179 %% @end
180 %%-----
181 code_change(_OldVsn, StateName, State, _Extra) ->
182     {ok, StateName, State}.
183
184 %%%=====
185 %%% Internal functions
186 %%%=====
187 clean_up_worker(ToDelCnt) ->
188     {Megasec, Sec, _Microsec} = now(),
189     clean_up_worker_in(0, Megasec*1000000 + Sec, ToDelCnt).
190
191 clean_up_worker_in(CurCnt, CurTimesec, ToDelCnt) ->
192     %% first condition: reference count == count & expired
193     MS = ets:fun2ms(
194         fun({K,_,Cnt,Timesec}) when Cnt == CurCnt, Timesec < CurTimesec ->
195             K
196         end
197     ),
198     NextDelCnt =
199         case ets:select(?hm_ets_cache_table, MS) of
200             [] ->
201                 %% second condition: reference count == count or expired
202                 MS2 = ets:fun2ms(
203                     fun({K,_,Cnt,Timesec}) when Cnt == CurCnt; Timesec < CurTimesec ->
204                         K
205                     end
206                 ),
207                 case ets:select(?hm_ets_cache_table, MS2) of
208                     %% check with next reference count(count+1)
209                     [] -> ToDelCnt;
210                     RecMatch2 ->
211                         %% delete ToDelCnt records from these records
212                         del_until_threshold(RecMatch2, ToDelCnt)
213                     end;
214                     %% delete ToDelCnt records from these records
215                     RecMatch -> del_until_threshold(RecMatch, ToDelCnt)
216                 end,
217             RowCnt = ets:select_count(?hm_ets_cache_table, ets:fun2ms(fun(_) -> true end)),
218             ?info_p("New Row Num:[~p]\n", none, [RowCnt]),
219             case NextDelCnt > 0 of
220                 true ->
221                     %% there is still need to delete some records
222                     clean_up_worker_in(CurCnt+1, CurTimesec, NextDelCnt);
223                 false ->
224                     ok
225             end.
226
227 del_until_threshold([], ToDelCnt) -> ToDelCnt;
228 del_until_threshold(_Any, 0) -> 0;
229 del_until_threshold([Key|RecList], ToDelCnt) ->
230     ets:delete(?hm_ets_cache_table, Key),
231     del_until_threshold(RecList, ToDelCnt - 1).
232
233 name(Name) -> list_to_atom(atom_to_list(?MODULE) ++ "_" ++ atom_to_list(Name)).

```

List 20: hm_cli.erl

```

1 % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2 % use this file except in compliance with the License. You may obtain a copy of
3 % the License at
4 %
5 %   http://www.apache.org/licenses/LICENSE-2.0
6 %
7 % Unless required by applicable law or agreed to in writing, software
8 % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9 % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, Yoshihiro TANAKA
15 %%% @doc
16 %%%   user interface of Harmonia
17 %%% @end
18 %%% Created : 2 Oct 2010 by Yoshihiro <hirotnkg@gmail.com>
19 %%%-----
20 -module(hm_cli).
21 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
22 %% API
23 -export([
24     cget/1,
25     create_table/3,
26     cstore/2,
27     drop_table/2,
28     get/1,
29     get_node_names/0,
30     get_table_info/2,
31     log_start/0,
32     log_stop/0,
33     rget/3,
34     rstore/3,
35     store/2,
36     get_data_count/1 % only for test purpose
37     ]).
38 -include_lib("eunit/include/eunit.hrl").
39
40 %%%=====
41 %%% API
42 %%%=====
43 %%%-----
44 %% @doc simple K/V store api
45 %% @spec(store(Key::atom()|string()|integer(), Value::any()) ->
46 %%     {ok, Cnt::integer()} |
47 %%     {partial, Cnt::integer()} |
48 %%     {ng, Msg::string}).
49 %% @end
50 %%%-----
51 store(Key, Value) ->
52     hm_ds:store(Key, Value).
53
54 %%%-----
55 %% @doc simple K/V store api with cache enabled
56 %% @spec(store(Key::atom()|string()|integer(), Value::any()) ->
57 %%     {ok, Cnt::integer()} |
58 %%     {partial, Cnt::integer()} |
59 %%     {ng, Msg::string}).

```

```

60 %%
61 %% @end
62 %%-----
63 cstore(Key, Value) ->
64     hm_ds:cstore(Key, Value).
65
66 %%-----
67 %% @doc store API for range query
68 %%     this API stores index and data.
69 %%
70 %%     format of KVList::[{FldName::string(), Vaue::any()}]
71 %%     Note: table for this domain & table name, and field definition
72 %%           need to be made in advance.
73 %% @spec(rstore(DomainName::string(), TableName::string(), KVList::list())
74 %%     ->
75 %%         {ok, Cnt::integer()} |
76 %%         {partial, Cnt::integer()} |
77 %%         {ng, Msg::string} |
78 %%         {error, Msg::string}).
79 %% @end
80 %%-----
81 rstore(DomainName, TableName, KVList) ->
82     hm_ds:rstore(DomainName, TableName, KVList).
83
84 %%-----
85 %% @doc simple K/V get api
86 %%     Recset::Records, Records::ListofRec
87 %% @spec(get(Key::atom()|string()|integer()) -> {ok, Result::Recset}|
88 %%     {error, nodata}).
89 %% @end
90 %%-----
91 get(Key) ->
92     hm_ds:get(Key).
93
94 %%-----
95 %% @doc simple K/V get api with cache enabled
96 %% @spec(get(Key::atom()|string()|integer()) -> {ok, Result::any}|
97 %%     {error, nodata}).
98 %%
99 %% @end
100 %%-----
101 cget(Key) ->
102     hm_ds:cget(Key).
103
104 %%-----
105 %% @doc rget API for table format.
106 %%     - format of Cond::string
107 %%     - available relational operator::=|!=|<|>|<=|>=|>=|>=]
108 %%     - available logical operator::and/or
109 %%     - parenthesis also available::( )
110 %%     - name of fields and tables need to be stringnum(start with character)
111 %% @spec(rget(DomainName::string(), TableName::string(), Cond::string()) ->
112 %%     {ok, Result}|{error, Msg}).
113 %% @end
114 %%-----
115 rget(DomainName, TableName, Cond) ->
116     hm_ds:rget(DomainName, TableName, Cond).
117
118 %%-----
119 %% @doc get list of nodes participating Harmonia ring

```

```

120 %% @spec(get_node_names() -> {ok, NameList::list()}).
121 %% @end
122 %%-----
123 get_node_names() ->
124     hm_name_server:get_list().
125
126 %%-----
127 %% @doc make table for storing index info of this table
128 %%     returns list of nodes in which tables were created,
129 %%     and list of nodes which table creation failed
130 %%     TODO: this return value make sense??
131 %% @spec(create_table(DomainName::string(), TableName::string(), AttList::list()) ->
132 %%     {ok, {NodeList::list(), FailedList::list()}}).
133 %% @end
134 %%-----
135 create_table(DomainName, TableName, AttList) ->
136     hm_table:create_table(DomainName, TableName, AttList).
137
138 %%-----
139 %% @doc drop index table
140 %%     returns list of nodes in which tables were created,
141 %%     and list of nodes which table creation failed
142 %%     TODO: this return value make sense??
143 %% @spec(drop_table(DomainName::string(), TableName::string()) ->
144 %%     {ok, {NodeList::list(), FailedList::list()}}).
145 %% @end
146 %%-----
147 drop_table(DomainName, TableName) ->
148     hm_table:drop_table(DomainName, TableName).
149
150 %%-----
151 %% @doc returns Table Id to access it, and field attributes
152 %%     TODO: Tid should be known to users??
153 %% @spec(get_table_info(DomainName::string(), TableName::string()) ->
154 %%     {ok, Tid, AttList::list()} |
155 %%     {error, no_node_available} | {error, no_table}
156 %% @end
157 %%-----
158 get_table_info(DomainName, TableName) ->
159     hm_table:get_table_info(DomainName, TableName).
160
161 %%-----
162 %% @doc start logging for all nodes
163 %% @spec(log_start() -> ok).
164 %% @end
165 %%-----
166 log_start() ->
167     {ok, NameList} = hm_name_server:get_list(),
168     log_start_in(NameList).
169
170 %%-----
171 %% @doc stop logging for all nodes
172 %% @spec(log_stop() -> ok).
173 %% @end
174 %%-----
175 log_stop() ->
176     {ok, NameList} = hm_name_server:get_list(),
177     log_stop_in(NameList).
178
179 %%-----

```

```

180 %% @doc get current data count
181 %% @spec(get_data_count() -> integer).
182 %% @end
183 %-----
184 get_data_count(Name) ->
185     hm_ds:get_data_count(Name).
186
187 %=====
188 %%% Internal functions
189 %=====
190 log_start_in([]) -> ok;
191 log_start_in([{_Name, NodeName}|NodeList]) ->
192     Res = rpc:call(NodeName, hm_log_h_file, add, []),
193     io:format("log start:[~p] Result:[~p]\n", [NodeName, Res]),
194     log_start_in(NodeList).
195
196 log_stop_in([]) -> ok;
197 log_stop_in([{_Name, NodeName}|NameList]) ->
198     Res = rpc:call(NodeName, hm_event_mgr, delete_file_handler, []),
199     case Res of
200     {error,module_not_found} ->
201         io:format("log stop:[~p] : The file handler is not installed.\n", [NodeName]);
202     _ ->
203         io:format("log stop:[~p] Result:[~p]\n", [NodeName, Res])
204     end,
205     log_stop_in(NameList).
206
207 %=====
208 %%% EUnit test functions
209 %=====
210
211 rangeq_test_() ->
212     [
213         ?_assertEqual(hm_cli:store("Domain1", "Tbl2", [{ "Fld1", xxx }, { "Fld2", 32 }, { "Fld3", \
214             -> textfile1 }]), {ok, 5}),
215         ?_assertEqual(hm_cli:store("Domain1", "Tbl2", [{ "Fld1", yyy }, { "Fld2", 150 }, { "Fld3", \
216             -> textfile2 }]), {ok, 5}),
217         ?_assertEqual(hm_cli:store("Domain1", "Tbl2", [{ "Fld1", zzz }, { "Fld2", 3000 }, { "Fld3", \
218             -> textfile3 }]), {ok, 5}),
219         ?_assertEqual(hm_cli:store("Domain1", "Tbl2", [{ "Fld1", aaa }, { "Fld2", 9000 }, { "Fld3", \
220             -> textfile4 }]), {ok, 5}),
221
222         ?_assertEqual({ok, [[xxx,32,textfile1]]}, hm_cli:get("Domain1", "Tbl2", "Fld2 == \
223             -> 32")),
224         ?_assertEqual({ok, [[yyy,150,textfile2]]}, hm_cli:get("Domain1", "Tbl2", "Fld2 == \
225             -> 150")),
226         ?_assertEqual({ok, [[zzz,3000,textfile3]]}, hm_cli:get("Domain1", "Tbl2", "Fld2 == \
227             -> 3000")),
228         ?_assertEqual({ok, [[aaa,9000,textfile4]]}, hm_cli:get("Domain1", "Tbl2", "Fld2 == \
229             -> 9000"))
230     ].

```

List 21: hm_config.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, hiro
15 %%% @doc I/F to node modules for configuration information
16 %%% @end
17 %%% Created : 2 Oct 2010 by Yoshihiro TANAKA <hirotnkg@gmail.com>
18 %%%-----
19 -module(hm_config).
20 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
21 -behaviour(gen_server).
22 -vsn('0.1').
23
24 %% API
25 -export([
26     get/1,
27     name/1,
28     start_link/1,
29     stop/0
30     ]).
31 %% gen_server callbacks
32 -export([init/1, handle_call/3, handle_cast/2, handle_info/2,
33     terminate/2, code_change/3]).
34
35 -include("harmonia.hrl").
36
37 %%%=====
38 %%% API
39 %%%=====
40
41 %%%-----
42 %%% @doc
43 %%% Starts the server
44 %%%
45 %%% @spec start_link(Env) -> {ok, Pid} | ignore | {error, Error}
46 %%% @end
47 %%%-----
48 start_link(Env) ->
49     gen_server:start_link({local, ?MODULE}, ?MODULE, Env, []).
50
51 stop() ->
52     gen_server:cast(?MODULE, stop).
53
54 get(Key) ->
55     {ok, _Value} = gen_server:call(?MODULE, {get, Key}).
56
57 name(RegName) ->
58     list_to_atom(atom_to_list(?MODULE) ++ "_" ++ atom_to_list(RegName)).
59

```



```

60 %%%=====
61 %%% gen_server callbacks
62 %%%=====
63
64 %%%-----
65 %%% @private
66 %%% @doc
67 %%% Initializes the server
68 %%%
69 %%% @spec init(Args) -> {ok, State} |
70 %%%                      {ok, State, Timeout} |
71 %%%                      ignore |
72 %%%                      {stop, Reason}
73 %%% @end
74 %%%-----
75 init(Env) -> {ok, Env}.
76
77 %%%-----
78 %%% @private
79 %%% @doc
80 %%% This function is called by a gen_server when it is about to
81 %%% terminate. It should be the opposite of Module:init/1 and do any
82 %%% necessary cleaning up. When it returns, the gen_server terminates
83 %%% with Reason. The return value is ignored.
84 %%%
85 %%% @spec terminate(Reason, State) -> void()
86 %%% @end
87 %%%-----
88 terminate(Reason, State) ->
89     ?info_p("terminate:Reason:[~p] State:[~p]~n", none, [Reason, State]),
90     ok.
91
92 %%%-----
93 %%% @private
94 %%% @doc
95 %%% Handling cast messages
96 %%%
97 %%% @spec handle_cast(Msg, State) -> {noreply, State} |
98 %%%                      {noreply, State, Timeout} |
99 %%%                      {stop, Reason, State}
100 %%% @end
101 %%%-----
102 handle_cast(stop, State) ->
103     {stop, normal, State}.
104
105 %%%-----
106 %%% @private
107 %%% @doc
108 %%% Handling call messages
109 %%%
110 %%% @spec handle_call(Request, From, State) ->
111 %%%                      {reply, Reply, State} |
112 %%%                      {reply, Reply, State, Timeout} |
113 %%%                      {noreply, State} |
114 %%%                      {noreply, State, Timeout} |
115 %%%                      {stop, Reason, Reply, State} |
116 %%%                      {stop, Reason, State}
117 %%% @end
118 %%%-----
119 handle_call({get, Key}, _From, Env) ->

```

```

120     Reply =
121         case proplists:get_value(Key, Env) of
122             undefined -> {error, key_undefined};
123             Value -> {ok, {Key, Value}}
124         end,
125     {reply, Reply, Env}.
126
127 %%-----
128 %% @private
129 %% @doc
130 %% Handling all non call/cast messages
131 %%
132 %% @spec handle_info(Info, State) -> {noreply, State} |
133 %%                                     {noreply, State, Timeout} |
134 %%                                     {stop, Reason, State}
135 %% @end
136 %%-----
137 handle_info(_Info, State) ->
138     {noreply, State}.
139
140 %%-----
141 %% @private
142 %% @doc
143 %% Convert process state when code is changed
144 %%
145 %% @spec code_change(OldVsn, State, Extra) -> {ok, NewState}
146 %% @end
147 %%-----
148 code_change(_OldVsn, State, _Extra) ->
149     {ok, State}.

```

List 22: hm_config_if.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, hiro
15 %%% @doc global I/F to for configuration information
16 %%% @end
17 %%% Created : 2 Oct 2010 by Yoshihiro TANAKA <hirotnkg@gmail.com>
18 %%%-----
19 -module(hm_config_if).
20 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
21 -behaviour(gen_server).
22 -vsn('0.1').
23
24 %% API
25 -export([
26     get/1,
27     name/1,
28     start_link/1,
29     stop/0
30     ]).
31 %% gen_server callbacks
32 -export([init/1, handle_call/3, handle_cast/2, handle_info/2,
33     terminate/2, code_change/3]).
34
35 -include("harmonia.hrl").
36
37 %%%=====
38 %%% API
39 %%%=====
40
41 %%%-----
42 %%% @doc
43 %%% Starts the server
44 %%%
45 %%% @spec start_link(Env) -> {ok, Pid} | ignore | {error, Error}
46 %%% @end
47 %%%-----
48 start_link({Name, Env}) ->
49     gen_server:start_link({global, name(Name)}, ?MODULE, Env, []).
50
51 stop() ->
52     gen_server:cast(?MODULE, stop).
53
54 get(Key) ->
55     {ok, _Value} = gen_server:call(?MODULE, {get, Key}).
56
57 name(RegName) -> list_to_atom(atom_to_list(?MODULE) ++ "_" ++ atom_to_list(RegName)).
58
59 %%%=====

```

```

60 %%% gen_server callbacks
61 %%%=====
62
63 %%%-----
64 %%% @private
65 %%% @doc
66 %%% Initializes the server
67 %%%
68 %%% @spec init(Args) -> {ok, State} |
69 %%%                      {ok, State, Timeout} |
70 %%%                      ignore |
71 %%%                      {stop, Reason}
72 %%% @end
73 %%%-----
74 init(Env) -> {ok, Env}.
75
76 %%%-----
77 %%% @private
78 %%% @doc
79 %%% This function is called by a gen_server when it is about to
80 %%% terminate. It should be the opposite of Module:init/1 and do any
81 %%% necessary cleaning up. When it returns, the gen_server terminates
82 %%% with Reason. The return value is ignored.
83 %%%
84 %%% @spec terminate(Reason, State) -> void()
85 %%% @end
86 %%%-----
87 terminate(Reason, State) ->
88     ?info_p("terminate:Reason:[~p] State:[~p]~n", none, [Reason, State]),
89     ok.
90
91 %%%-----
92 %%% @private
93 %%% @doc
94 %%% Handling cast messages
95 %%%
96 %%% @spec handle_cast(Msg, State) -> {noreply, State} |
97 %%%                      {noreply, State, Timeout} |
98 %%%                      {stop, Reason, State}
99 %%% @end
100 %%%-----
101 handle_cast(stop, State) ->
102     {stop, normal, State}.
103
104 %%%-----
105 %%% @private
106 %%% @doc
107 %%% Handling call messages
108 %%%
109 %%% @spec handle_call(Request, From, State) ->
110 %%%                      {reply, Reply, State} |
111 %%%                      {reply, Reply, State, Timeout} |
112 %%%                      {noreply, State} |
113 %%%                      {noreply, State, Timeout} |
114 %%%                      {stop, Reason, Reply, State} |
115 %%%                      {stop, Reason, State}
116 %%% @end
117 %%%-----
118 handle_call({get, Key}, _From, Env) ->
119     Reply =

```

```

120         case proplists:get_value(Key, Env) of
121             undefined -> {error, key_undefined};
122             Value -> {ok, {Key, Value}}
123         end,
124         {reply, Reply, Env}.
125
126 %%-----
127 %% @private
128 %% @doc
129 %% Handling all non call/cast messages
130 %%
131 %% @spec handle_info(Info, State) -> {noreply, State} |
132 %%                                     {noreply, State, Timeout} |
133 %%                                     {stop, Reason, State}
134 %% @end
135 %%-----
136 handle_info(_Info, State) ->
137     {noreply, State}.
138
139 %%-----
140 %% @private
141 %% @doc
142 %% Convert process state when code is changed
143 %%
144 %% @spec code_change(OldVsn, State, Extra) -> {ok, NewState}
145 %% @end
146 %%-----
147 code_change(_OldVsn, State, _Extra) ->
148     {ok, State}.

```

List 23: hm_ds.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, hiro
15 %%% @doc data storage I/F
16 %%% @end
17 %%% Created : 2 Oct 2010 by Yoshihiro TANAKA <hirotnkg@gmail.com>
18 %%%-----
19 -module(hm_ds).
20 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
21 -behaviour(gen_server).
22 -vsn('0.1').
23
24 %% API
25 -export([
26     cdel/1,
27     cget/1,
28     cstore/2,
29     del/1,
30     fun_for_data/2,
31     fun_for_index/2,
32     get/1,
33     name/1,
34     rdel/3,
35     rget/3,
36     rstore/3,
37     start_link/1,
38     stop/0,
39     stop/1,
40     store/2,
41     get_data_count/1 % only for test purpose
42     ]).
43 -export([
44     gather_get/4,
45     gather_store/4,
46     gather_delete/4,
47     delete_data_table_solo/6,
48     lookup_data_table_solo/6,
49     store_solo/6
50     ]).
51 %% gen_server callbacks
52 -export([init/1, handle_call/3, handle_cast/2, handle_info/2,
53     terminate/2, code_change/3]).
54
55 -include("harmonia.hrl").
56
57 -define(MODULE_NAME, atom_to_list(?MODULE)).
58 -define(hm_global_table, hm_table_global).
59

```

```

60
61 %%%=====
62 %%% API
63 %%%=====
64 %%%-----
65 %%% @doc
66 %%% Starts the server
67 %%%
68 %%% @spec start_link(Env) -> {ok, Pid} | ignore | {error, Error}
69 %%% @end
70 %%%-----
71 start_link(RegName) ->
72     gen_server:start_link({global, name(RegName)}, ?MODULE, RegName, []).
73
74 stop() ->
75     gen_server:cast({global, ?MODULE}, stop).
76
77 stop(RegName) ->
78     gen_server:cast({global, name(RegName)}, stop).
79
80 %%%-----
81 %%% @doc Key::atom()/string()/integer()
82 %%% @spec get(Key) -> {ok, {Key, Value::any()}} | {error, Msg}
83 %%% @end
84 %%%-----
85 get(Key) ->
86     get_in(Key).
87
88 %%%-----
89 %%% @doc Key::atom()/string()/integer()
90 %%%     simple K/V store with cache enabled
91 %%% @spec cget(Key) -> {ok, {Key, Value::any()}} | {error, Msg}
92 %%% @end
93 %%%-----
94 cget(Key) ->
95     cget_in(Key).
96
97 %%%-----
98 %%% @doc Range Query API
99 %%%     Returned list does not include DTName
100 %%% @spec rget(DomainName::string(), TableName::string(), Cond::string()) ->
101 %%%     {ok, Rec::list()} | {error, Msg::any()}
102 %%% @end
103 %%%-----
104 rget(DomainName, TableName, Cond) ->
105     get_in(DomainName, TableName, Cond).
106
107 %%%-----
108 %%% @doc simple K/V store
109 %%% @spec store(Key::atom(), Value::any() ) ->
110 %%%     {ok, Cnt::integer()} | {partial, Cnt::integer()} | {ng, Msg::string}
111 %%% @end
112 %%%-----
113 store(Key, Value) ->
114     store_in(Key, Value).
115
116 %%%-----
117 %%% @doc simple K/V store with cache enabled
118 %%% @spec cstore(Key::atom(), Value::any() ) ->
119 %%%     {ok, Cnt::integer()} | {partial, Cnt::integer()} | {ng, Msg::string}

```

```

120 %% @end
121 %%-----
122 cstore(Key, Value) ->
123     cstore_in(Key, Value).
124
125 %%-----
126 %% @doc Range Query API
127 %%     KVList: [{FieldName::string(), Value::any()}]
128 %% @spec(rstore(DomainName::string(), TableName::string(), KVList)) ->
129 %%     {ok, Rec::list()} | {error, Msg::any()}
130 %% @end
131 %%-----
132 rstore(DomainName, TableName, KVList) ->
133     store_in(DomainName, TableName, KVList).
134
135 %%-----
136 %% @doc Key::atom()/string()/integer()
137 %% @spec(del(Key) -> {ok, {Key, Value::any()}} | {error, Msg})
138 %% @end
139 %%-----
140 del(Key) ->
141     del_in(Key).
142
143 %%-----
144 %% @doc Key::atom()/string()/integer()
145 %%     simple K/V delete with cache enabled
146 %% @spec(cdel(Key) -> {ok, {Key, Value::any()}} | {error, Msg})
147 %% @end
148 %%-----
149 cdel(Key) ->
150     cdel_in(Key).
151
152 %%-----
153 %% @doc Range Delete API
154 %%     Returned list does not include DTName
155 %% @spec(rdel(DomainName::string(), TableName::string(), Cond::string())) ->
156 %%     {ok, DeletedNum} | {error, Msg::any()}
157 %%
158 %% @end
159 %%-----
160 rdel(DomainName, TableName, Cond) ->
161     del_in(DomainName, TableName, Cond).
162
163 %%-----
164 %% @doc Current Data counter
165 %% @spec(get_data_count(Name) -> integer())
166 %%
167 %% @end
168 %%-----
169 get_data_count(Name) ->
170     gen_server:call({global, name(Name)}, get_current_data_count).
171
172 %%-----
173 %% @doc return node specific module name
174 %% @spec
175 %% @end
176 %%-----
177 name(Name) when is_list(Name) ->
178     list_to_atom(?MODULE_NAME ++ "_" ++ Name);
179 name(Name) ->

```



```

180     list_to_atom(?MODULE_NAME ++ "_" ++ atom_to_list(Name)).
181
182     %%=====
183     %% gen_server callbacks
184     %%=====
185
186     %%-----
187     %% @private
188     %% @doc
189     %% Initializes the server
190     %%
191     %% @spec init(Args) -> {ok, State} |
192     %%                      {ok, State, Timeout} |
193     %%                      ignore |
194     %%                      {stop, Reason}
195     %% @end
196     %%-----
197     init(RegName) ->
198         ets:new(?hm_global_table, [bag, public, named_table]),
199         {ok, {RegName, [{hm_global_table, 0}]}.
200
201     %%-----
202     %% @private
203     %% @doc
204     %% This function is called by a gen_server when it is about to
205     %% terminate. It should be the opposite of Module:init/1 and do any
206     %% necessary cleaning up. When it returns, the gen_server terminates
207     %% with Reason. The return value is ignored.
208     %%
209     %% @spec terminate(Reason, State) -> void()
210     %% @end
211     %%-----
212     terminate(Reason, State) ->
213         ?info_p("terminate:Reason:[~p] State:[~p]~n", none, [Reason, State]),
214         ok.
215
216     %%-----
217     %% @private
218     %% @doc
219     %% Handling cast messages
220     %%
221     %% @spec handle_cast(Msg, State) -> {noreply, State} |
222     %%                      {noreply, State, Timeout} |
223     %%                      {stop, Reason, State}
224     %% @end
225     %%-----
226     handle_cast(stop, State) ->
227         {stop, normal, State}.
228
229     %%-----
230     %% @private
231     %% @doc
232     %% Handling call messages
233     %%
234     %% @spec handle_call(Request, From, State) ->
235     %%                      {reply, Reply, State} |
236     %%                      {reply, Reply, State, Timeout} |
237     %%                      {noreply, State} |
238     %%                      {noreply, State, Timeout} |
239     %%                      {stop, Reason, Reply, State} |

```

```

240 %%                                     {stop, Reason, State}
241 %% @end
242 %%-----
243 handle_call(get_current_data_count, _From, {RegName, TableList, Cnt}) ->
244     {reply, {ok, Cnt}, {RegName, TableList, Cnt}};
245
246 handle_call({register_table, TableName}, _From, {RegName, TableList, Cnt}) ->
247     ?info_p("register_table info:[~p] new table:[~p].~n", RegName, [TableList, TableName]),
248     {reply, {ok, register_table, TableName}, {RegName, [TableName|TableList], Cnt}};
249
250 handle_call({unregister_table, TableName}, _From, {RegName, TableList, Cnt}) ->
251     ?info_p("unregister_table info:[~p] new table:[~p].~n", RegName, [TableList, TableName])\
252     →,
253     NewTableList = TableList -- [TableName],
254     {reply, {ok, unregister_table}, {RegName, NewTableList, Cnt}};
255
256 handle_call({get_table_info, DTName}, _From, {_RegName, TableList, _Cnt}=State) ->
257     case lists:member(DTName, TableList) of
258     true -> {reply, {ok, DTName}, State};
259     false -> {reply, {error, no_table}, State}
260     end;
261
262 handle_call({select_table, ?hm_global_table, DTName, FlistModified, MS}, _From, {_RegName, \
263     →_TableList, _Cnt}=State) ->
264     % ets:select(Tid, [{{'$1', ['$2', '$3', '$4']}, {'and', {'==', '$2', yyy}, {'==', '$1', '\
265     →Domain1Tbl2'}}}], ['$1'])).
266     Reply = ets:select(?hm_global_table, [{{'$1', FlistModified}, {'and', {'==', '$1', DTName}, \
267     →MS}], ['$1'])),
268     {reply, {ok, lists:usort(Reply)}, State};
269
270 handle_call({select_table, DTName, FlistModified, MS}, _From, State) ->
271     Reply = ets:select(DTName, [{{'$1', FlistModified}, MS, ['$1']}]},
272     {reply, {ok, lists:usort(Reply)}, State};
273
274 handle_call({select_delete_table, ?hm_global_table, DTName, FlistModified, MS}, _From, {_RegName, \
275     →_TableList, _Cnt}=State) ->
276     NumDeleted = ets:select_delete(?hm_global_table, [{{'$1', FlistModified}, {'and', {'==', '\
277     →$1', DTName}, MS}], ['$1'])),
278     {reply, {ok, NumDeleted}, State};
279
280 handle_call({select_delete_table, DTName, FlistModified, MS}, _From, {RegName, TableList, \
281     →Cnt}) ->
282     NumDeleted = ets:select_delete(DTName, [{{'$1', FlistModified}, MS, ['$1']}]},
283     {reply, {ok, NumDeleted}, {RegName, TableList, Cnt - 1}};
284
285 handle_call({store, TableName, Key, Value}, _From, {RegName, TableList, Cnt}) ->
286     ?info_p("store:TableName:[~p] Key:[~p] Value:[~p] TableList:[~p].~n", store, [TableName, \
287     → Key, Value, TableList]),
288     case lists:member(TableName, TableList) of
289     false ->
290         {reply, {error, {store, no_table_found}}, {RegName, TableList, Cnt}};
291     true ->
292         Ret = (catch ets:insert(TableName, {Key, Value})),
293         case Ret of
294         true ->
295             {reply, {ok, insert}, {RegName, TableList, Cnt + 1}};
296         Any ->
297             {reply,
298             {error, {store, exception, TableName, {Key, Value}, Any}},
299             {RegName, TableList, Cnt}}
300         end
301     end

```

```

292         end
293     end;
294
295     handle_call({get, Key}, _From, {RegName, TableList, Cnt}) ->
296         Reply = ets:lookup(?hm_global_table, Key),
297         {reply, Reply, {RegName, TableList, Cnt}};
298
299     handle_call({delete, Key}, _From, {RegName, TableList, Cnt}) ->
300         ets:delete(?hm_global_table, Key),
301         {reply, {ok, delete, Key}, {RegName, TableList, Cnt - 1}}.
302
303     %%-----
304     %% @private
305     %% @doc
306     %% Handling all non call/cast messages
307     %%
308     %% @spec handle_info(Info, State) -> {noreply, State} |
309     %%                                     {noreply, State, Timeout} |
310     %%                                     {stop, Reason, State}
311     %% @end
312     %%-----
313     handle_info(_Info, State) ->
314         {noreply, State}.
315
316     %%-----
317     %% @private
318     %% @doc
319     %% Convert process state when code is changed
320     %%
321     %% @spec code_change(OldVsn, State, Extra) -> {ok, NewState}
322     %% @end
323     %%-----
324     code_change(_OldVsn, State, _Extra) ->
325         {ok, State}.
326
327     %% -----
328     %% Internal Functions
329     %% -----
330
331     get_in(Key) ->
332         {ok, TargetName, SuccListTarget} = hm_router:lookup_with_succlist(Key),
333         SuccList = hm_misc:make_request_list(TargetName, SuccListTarget),
334         get_from_succlist(SuccList, Key).
335
336     del_in(Key) ->
337         {ok, TargetName, SuccListTarget} = hm_router:lookup_with_succlist(Key),
338         SuccList = hm_misc:make_request_list(TargetName, SuccListTarget),
339         del_from_succlist(SuccList, Key).
340
341     store_in(Key, Value) ->
342         {ok, RouterName, SuccList} = hm_router:lookup_with_succlist(Key),
343         store_in_to(RouterName, SuccList, ?hm_global_table, {Key, Value}).
344
345     cget_in(Key) ->
346         case hm_cache:get_cache(Key) of
347             none ->
348                 case hm_ds:get(Key) of
349                     {ok, {Key, Value}} ->
350                         hm_cache:store_cache(Key, Value),
351                         {ok, {Key, Value}};

```

```

352         {error, Msg} ->
353             {error, Msg}
354     end;
355
356     {ok, {Value, _Cnt}} ->
357         {ok, {Key, Value}}
358 end.
359
360 cdel_in(Key) ->
361     hm_cache:del_cache(Key),
362     del_in(Key).
363
364 cstore_in(Key, Value) ->
365     hm_cache:store_cache(Key, Value),
366     store_in(Key, Value).
367
368 get_in(DomainName, TableName, Cond) ->
369     DTName = list_to_atom(DomainName ++ TableName),
370     %dynamite_prof:start_prof(make_request_list_from_dt),
371     NodeList = hm_misc:make_request_list_from_dt(DomainName, TableName),
372     %dynamite_prof:stop_prof(make_request_list_from_dt),
373     %dynamite_prof:start_prof(lookup_index_table_node),
374     {ok, IndexTableNode, IndexDsNode} = lookup_index_table_node(NodeList),
375     %dynamite_prof:stop_prof(lookup_index_table_node),
376     %dynamite_prof:start_prof(lookup_index_table_attribute),
377     {ok, _Tid, AttList} = lookup_index_table_attribute(IndexTableNode, DTName),
378     %dynamite_prof:stop_prof(lookup_index_table_attribute),
379     %dynamite_prof:start_prof(get_query_spec),
380     {ok, MS, MSData, FlistIndex, FlistData} = get_query_spec(Cond, AttList),
381     %dynamite_prof:stop_prof(get_query_spec),
382     %dynamite_prof:start_prof(get_data_node_list),
383     {ok, DataNodeList} = get_data_node_list(IndexDsNode, DTName, FlistIndex, MS, get),
384     %dynamite_prof:stop_prof(get_data_node_list),
385     %dynamite_prof:start_prof(scatter_gather),
386     Res = scatter_gather(DomainName, DataNodeList, DTName, FlistData, MSData, TableName, \
        ->Cond),
387     %dynamite_prof:stop_prof(scatter_gather),
388     Res.
389
390 del_in(DomainName, TableName, Cond) ->
391     DTName = list_to_atom(DomainName ++ TableName),
392     NodeList = hm_misc:make_request_list_from_dt(DomainName, TableName),
393     {ok, IndexTableNode, IndexDsNode} = lookup_index_table_node(NodeList),
394     {ok, _Tid, AttList} = lookup_index_table_attribute(IndexTableNode, DTName),
395     {ok, MS, MSData, FlistIndex, FlistData} = get_query_spec(Cond, AttList),
396
397     %% get data node list from index table AND delete index information
398     {ok, DataNodeList} = get_data_node_list(IndexDsNode, DTName, FlistIndex, MS, delete),
399     scatter_delete(DomainName, DataNodeList, DTName, FlistData, MSData, TableName, Cond).
400
401 store_in(DomainName, TableName, KVList) ->
402
403     %dynamite_prof:start_prof(make_request_list_from_dt),
404     NodeList = hm_misc:make_request_list_from_dt(DomainName, TableName),
405     %dynamite_prof:stop_prof(make_request_list_from_dt),
406
407     {IndexTableNode, _} = hd(NodeList), % index record is stored on this node in DTName \
        ->table
408     case hm_table:get_table_info(DomainName, TableName, NodeList) of
409         {ok, DTName, AttList} ->

```

```

410
411     %% Key is calculated with DTName+values of key fields
412     {ok, Key} = calc_key_from_key_data(DTName, KVList, AttList),
413
414     %% data record is stored on this node in ?hm_global_table table
415     %% at first data is stored, then index is stored
416     {ok, DataTableNode, SuccList} = hm_router:lookup_with_succlist(Key),
417     case store_data(DTName, DataTableNode, SuccList, KVList) of
418         {ok, _} ->
419             store_index(DTName, NodeList, DataTableNode, IndexTableName, KVList, \
420                 ->AttList);
421             {partial, Cnt} ->
422                 ?warning_p("store_data failed(partial) :DTName:[~p] DataTableNode:[~p] \
423                     ->KVList:[~p] Cnt:[~p].~n",
424                     none, [DTName, DataTableNode, KVList, Cnt]),
425                 store_index(DTName, NodeList, DataTableNode, IndexTableName, KVList, \
426                     ->AttList);
427             {ng, Cnt} ->
428                 ?error_p("store_data failed :DTName:[~p] DataTableNode:[~p] KVList:[~p] \
429                     ->Cnt:[~p].~n",
430                     none, [DTName, DataTableNode, KVList, Cnt]),
431                 {error, store_into_datatable_failed}
432     end;
433     {error, ErrInfo} -> {error, ErrInfo}
434 end.
435
436 scatter_gather(DomainName, DataNodeList, DTName, FlistData, MSData, TableName, Cond) ->
437     Ref = make_ref(),
438
439     % spawn a gathering process
440     LoopPid = spawn_link( ?MODULE, gather_get, [length(DataNodeList), self(), Ref, []]),
441
442     % spawn a scattering processes for each node
443     lists:foreach(
444         fun(E1) ->
445             spawn_link(
446                 ?MODULE,
447                 lookup_data_table_solo,
448                 [E1, DTName, FlistData, MSData, LoopPid, Ref]
449             )
450         end,
451         DataNodeList
452     ),
453     % receive total result
454     receive
455         {ok, Ref, Result} -> {ok, Result};
456         {error, Ref, Msg} -> {error, Msg}
457     after ?TIMEOUT_GET ->
458         ?error_p("get timeout :DomainName:[~p] TabName:[~p] Cond:[~p]~n",
459             get, [DomainName, TableName, Cond]),
460         {error, timeout}
461     end.
462
463 get_query_spec(Cond, AttList) ->
464     MS = hm_qp:parse(hm_qp:scan(Cond, {index, AttList})),
465     [MSData] = hm_qp:parse(hm_qp:scan(Cond, {data, AttList})),
466     {_, FlistIndex} = lists:foldl(fun ?MODULE:fun_for_index/2, {1,[]}, AttList),
467     {_, FlistData} = lists:foldl(fun ?MODULE:fun_for_data/2, {1,[]}, AttList),
468     {ok, MS, MSData, FlistIndex, FlistData}.
469

```

```

466 get_data_node_list(IndexDsNode, DTName, FlistIndex, MS, Type) when Type == get ->
467     {ok, RowList} = gen_server:call({global, IndexDsNode}, {select_table, DTName, FlistIndex\
    →, MS}),
468     DataNodeList =
469         lists:usort(
470             lists:foldl(fun (Node, AccIn) ->
471                 [hd(Node) | AccIn]
472             end, [], RowList)
473         ),
474     {ok, DataNodeList};
475 get_data_node_list(IndexDsNode, DTName, FlistIndex, MS, Type) when Type == delete ->
476     {ok, RowList} = gen_server:call({global, IndexDsNode}, {select_table, DTName, FlistIndex\
    →, MS}),
477     DataNodeList =
478         lists:usort(
479             lists:foldl(fun (Node, AccIn) ->
480                 [hd(Node) | AccIn]
481             end, [], RowList)
482         ),
483     {ok, _DeletedNum} = gen_server:call({global, IndexDsNode}, {select_delete_table, DTName,\
    → FlistIndex, MS}),
484     {ok, DataNodeList}.
485
486 lookup_index_table_attribute(IndexTableName, DTNameTable) ->
487     %% TODO: don't call gen_server:call of other module directly
488     case gen_server:call({global, IndexTableName}, {get_table_info, DTNameTable}) of
489         {ok, TableName, AttList} -> {ok, TableName, AttList};
490         {error, Msg} ->
491             {error, Msg}
492     end.
493
494 scatter_delete(DomainName, DataNodeList, DTName, FlistData, MSData, TableName, Cond) ->
495     Ref = make_ref(),
496
497     % spawn a gathering process
498     LoopPid = spawn_link(?MODULE, gather_delete, [length(DataNodeList), self(), Ref, 0]),
499
500     % spawn a scattering processes for each node
501     lists:foreach(
502         fun(E1) ->
503             spawn_link(
504                 ?MODULE,
505                 delete_data_table_solo,
506                 [E1, DTName, FlistData, MSData, LoopPid, Ref]
507             )
508         end,
509         DataNodeList
510     ),
511     % receive total result
512     receive
513         {ok, Ref, DeletedNum} -> {ok, DeletedNum};
514         {error, Ref, Msg} -> {error, Msg}
515     after ?TIMEOUT_GET ->
516         ?error_p("get timeout :DomainName:[~p] TabName:[~p] Cond:[~p]~n",
517             get, [DomainName, TableName, Cond]),
518         {error, timeout}
519     end.
520
521 gather_get(0, Pid, Ref, ResList) ->
522     ?info_p("gather_get OK:ResList:[~p]~n", gather_get, [ResList]),

```

```

523     DataList = lists:usort(ResList),
524     log_get_data(DataList),
525     Pid ! {ok, Ref, lists:usort(ResList)};
526 gather_get(Cnt, Pid, Ref, ResList) ->
527     receive
528         {ok, Ref, RowList} ->
529             gather_get( Cnt - 1, Pid, Ref, lists:merge(RowList,ResList) );
530         {error, Ref, Msg} ->
531             ?error_p("gather_get Error:Msg:[~p]~n", gather_get, [Msg]),
532             gather_get( Cnt - 1, Pid, Ref, ResList )
533     after ?TIMEOUT_GET ->
534         Pid ! {error, Ref, timeout}
535     end.
536
537 gather_delete(0, Pid, Ref, Acc) ->
538     ?info_p("gather_delete OK:[~p]~n", gather_delete, [Acc]),
539     Pid ! {ok, Ref, Acc};
540 gather_delete(Cnt, Pid, Ref, Acc) ->
541     receive
542         {ok, Ref, DeletedNum} ->
543             gather_delete( Cnt - 1, Pid, Ref, Acc + DeletedNum );
544         {error, Ref, Msg} ->
545             ?error_p("gather_delete Error:Msg:[~p]~n", gather_delete, [Msg]),
546             gather_delete( Cnt - 1, Pid, Ref, Acc )
547     after ?TIMEOUT_GET ->
548         Pid ! {error, Ref, timeout}
549     end.
550
551 lookup_data_table_solo(NodeName, DTNameTable, FlistModified, MS, LoopPid, Ref) ->
552     BareName = list_to_atom(atom_to_list(NodeName) -- ?PROCESS_PREFIX),
553     DataNodeName = name(BareName),
554
555     case hm_misc:is_alive(BareName) of
556         false ->
557             ?warning_p("DataNode Not Alive : Node:[~p].~n", none, [DataNodeName]),
558             LoopPid ! {error, Ref, node_not_alive};
559         true ->
560             {ok, RowList} =
561                 gen_server:call(
562                     {global, DataNodeName},
563                     {select_table, ?hm_global_table, DTNameTable, FlistModified, MS}
564                 ),
565             LoopPid ! {ok, Ref, lists:map(fun({_,Rec}) -> Rec end, RowList)}
566     end.
567
568 delete_data_table_solo(NodeName, DTNameTable, FlistModified, MS, LoopPid, Ref) ->
569     BareName = list_to_atom(atom_to_list(NodeName) -- ?PROCESS_PREFIX),
570     DataNodeName = name(BareName),
571
572     case hm_misc:is_alive(BareName) of
573         false ->
574             ?warning_p("DataNode Not Alive : Node:[~p].~n", none, [DataNodeName]),
575             LoopPid ! {error, Ref, node_not_alive};
576         true ->
577             {ok, NumDeleted} =
578                 gen_server:call(
579                     {global, DataNodeName},
580                     {select_delete_table, ?hm_global_table, DTNameTable, FlistModified, MS}
581                 ),
582             LoopPid ! {ok, Ref, NumDeleted}

```

```

583     end.
584
585 %% @doc returns hm_table/hm_ds name of the server first looked alive
586 lookup_index_table_node(NodeList) ->
587     case hm_misc:get_first_alive_entry(NodeList) of
588     {error, none} -> {error, no_node_available};
589     {IndexNode, _Vector} ->
590         IndexTableNode = hm_table:name(atom_to_list(IndexNode) -- ?PROCESS_PREFIX),
591         IndexNodeDs = name(atom_to_list(IndexNode) -- ?PROCESS_PREFIX),
592         {ok, IndexTableNode, IndexNodeDs}
593     end.
594
595 fun_for_index({_Fname, Bool, _},{N,FList}) ->
596     case Bool == true of
597     true -> {N+1, FList ++ [list_to_atom("$" ++ integer_to_list(N+1))]};
598     false -> {N, FList}
599     end.
600
601 fun_for_data(_T,{N,FList}) ->
602     {N+1, FList ++ [list_to_atom("$" ++ integer_to_list(N+1))]}..
603
604 %%
605 %% @spec store_sta(DTName::atom(), DataTableNode::atom(), KVList::list()) ->
606 %%                                     {ok, Cnt::integer()} |
607 %%                                     {partial, Cnt::integer()}
608 %%                                     {ng, Msg::string} |
609 %%
610 %% @doc
611 %% store data of all fields in the following record form:
612 %% Row: {DmainName++TableName::atom(), [{Fld1, Value}, {Fld2, Value}, ...]}
613 %% Key : calculate from key fields' data
614 %% Target Nodes: calculate from Key
615 %%
616 store_data(DTName, DataTableNode, SuccList, KVList) ->
617     VList = lists:map(fun({_Fld,Val}) -> Val end, KVList),
618     store_in_to(DataTableNode, SuccList, ?hm_global_table, {DTName, VList}).
619 %%
620 %% @spec store_index(DTName::atom(), DataTableNode::atom(),
621 %%                   IndexTableNode::atom(), KVList::list(), AttList::list()) ->
622 %%                                     {ok, Cnt::integer()} |
623 %%                                     {partial, Cnt::integer()}
624 %%                                     {ng, Msg::string} |
625 %%
626 %% @doc
627 %% store index data to global_table in the following record form:
628 %% RouterName indicates the successor from the key DTName.
629 %% Data: {RouterName, [{Fld1, Value}, {Fld2, Value}, ...]}
630 %% Table: DomainName ++ TableName
631 store_index(DTNameTable, SuccList, DataTableNode, IndexTableNode, KVList, AttList) ->
632     {ok, Row} = extract_kv_tuples(KVList, AttList, true),
633     Vlist =
634         lists:foldl(
635             fun({_Fname,Value}, AccIn) ->
636                 AccIn ++ [Value]
637             end, [], Row),
638     store_in_to(IndexTableNode, SuccList, DTNameTable, {DataTableNode, Vlist}).
639
640 %% retrieve tuples of fields with data
641 %% KVFlag: true -> returns key-tuples list
642 %% KVFlag: false -> returns data-tuples list

```



```

643 extract_kv_tuples(KVList, AttList, KVFlag) ->
644     DataFields =
645         lists:filter(
646             fun({_Fname, BoolVal, _Init}) ->
647                 case BoolVal == KVFlag of
648                     true -> true;
649                     false -> false
650                 end
651             end,
652             AttList),
653     StoreDataList =
654         lists:filter(
655             fun({_Fname, _}) ->
656                 case lists:keysearch(Fname, 1, DataFields) of
657                     false -> false;
658                     _ -> true
659                 end
660             end,
661             KVList),
662     {ok, StoreDataList}.
663
664 %%
665 %% @spec(calc_key_from_key_data(DomainName, TableName, KVList, AttList) -> {ok, Key::atom()})
666 %% @doc generates key from DTName and key values
667 %% @TODO: the type of data in key fields are now confined to following:
668 %%         may need to figure out more flexibility
669 %%         - string()
670 %%         - atom()
671 %%         - integer()
672 calc_key_from_key_data(DTName, KVList, AttList) ->
673     % retrieve tuples of key fields
674     {ok, KeyList} =
675         extract_kv_tuples(KVList, AttList, true),
676
677     % the value of the key fields must be list/atom/integer
678     Key = lists:foldl(fun
679         ({_, Data}, AccIn) when is_list(Data) ->
680             Data ++ AccIn;
681
682         ({_, Data}, AccIn) when is_atom(Data) ->
683             atom_to_list(Data) ++ AccIn;
684
685         ({_, Data}, AccIn) when is_integer(Data) ->
686             integer_to_list(Data) ++ AccIn
687     end,
688     [], KeyList),
689     {ok, list_to_atom(atom_to_list(DTName) ++ Key)}.
690
691 store_in_to(RouterName, SuccListTemp, TableName, {Key, Value}) ->
692     % store to all successor list nodes
693     SuccList = hm_misc:make_request_list(RouterName, SuccListTemp),
694     ?info_p("store_to_succlist:SuccListTemp:[~p].~n", store, [SuccListTemp]),
695     ?info_p("store_to_succlist:SuccList:[~p].~n", store, [SuccList]),
696     ?info_p("DATA-STORE>>>> Key:[~p] Value:[~p].~n", store, [Key, Value]),
697     store_to_succlist(SuccList, TableName, Key, Value, {length(SuccList), 0}).
698
699 gather_store(0, Acc, Pid, Ref) -> Pid ! {ok, Ref, Acc};
700 gather_store(Len, Acc, Pid, Ref) ->
701     receive
702         {ok, Ref, {ok, _Res}} -> gather_store(Len - 1, Acc + 1, Pid, Ref);

```

```

703         {ok, Ref, {error, _Res}} -> gather_store(Len - 1, Acc, Pid, Ref);
704         {error, Ref, _Msg}         -> gather_store(Len - 1, Acc, Pid, Ref)
705     after ?TIMEOUT_GET ->
706         ?error_p("store timeout ~n", gather_store, []),
707         Pid ! {error, Ref, timeout}
708     end.
709
710 store_solo(RouterName, TableName, Key, Value, Pid, Ref) ->
711     TargetName = name(atom_to_list(RouterName) -- ?PROCESS_PREFIX),
712     case global:whereis_name(TargetName) of
713         undefined ->
714             ?error_p("store_to_succlist:Target undefined ~nKey:[~p] TargetName:[~p].~n",
715                 store, [Key, TargetName]),
716             Pid ! {error, Ref, undefined};
717         _ ->
718             ?info_p("store_to_succlist:Key:[~p] TargetName:[~p].~n", store, [Key, \
719                 →TargetName]),
720             Reply = gen_server:call({global, TargetName}, {store, TableName, Key, Value}),
721             Pid ! {ok, Ref, Reply}
722     end.
723
724 store_to_succlist(SuccList, TableName, Key, Value, {Len, Cnt}) ->
725     Ref = make_ref(),
726     Pid = spawn_link(?MODULE, gather_store, [Len, Cnt, self(), Ref]),
727
728     % spawn a scattering processes for each node
729     lists:foreach(
730         fun({El, _}) ->
731             spawn_link(
732                 ?MODULE,
733                 store_solo,
734                 [El, TableName, Key, Value, Pid, Ref]
735             ),
736         SuccList
737     ),
738     % receive total result
739     receive
740         {ok, Ref, Acc} ->
741             case Acc == length(SuccList) of
742                 true -> {ok, Acc};
743                 false -> {partial, Acc}
744             end;
745         {error, Ref, timeout} -> {ng, timeout}
746     after ?TIMEOUT_GET ->
747         ?error_p("store timeout :TabName:[~p] Key:[~p] Value:[~p]~n", store, [TableName, Key\
748             →, Value]),
749         {error, timeout}
750     end.
751
752 get_from_succlist([], _Key) -> {error, nodata};
753 get_from_succlist(Succlist, Key) ->
754     {RouterName, _} = hd(Succlist),
755     TargetName = name(atom_to_list(RouterName) -- ?PROCESS_PREFIX),
756     case global:whereis_name(TargetName) of
757         undefined ->
758             get_from_succlist(tl(Succlist), Key);
759         _ ->
760             Result = gen_server:call({global, TargetName}, {get, Key}),

```

```

760         ?info_p("get:Key:[~p] TargetName:[~p] Result:[~p] .~n", get, [Key, TargetName, \
        →Result]),
761         case length(Result) of
762         0 -> get_from_succlist(tl(Succlist), Key);
763         _ -> {ok, Result}
764         end
765     end.
766
767 del_from_succlist([], _Key) -> ok;
768 del_from_succlist(Succlist, Key) ->
769     {RouterName, _} = hd(Succlist),
770     TargetName = name(atom_to_list(RouterName) -- ?PROCESS_PREFIX),
771     case global:whereis_name(TargetName) of
772     undefined ->
773         ?warning_p("del:Key:[~p] TargetName:[~p] global name undefined .~n", delete, [\
        →Key, TargetName]);
774     _ ->
775         {ok, delete, Key} = gen_server:call({global, TargetName}, {delete, Key}),
776         ?info_p("del:Key:[~p] TargetName:[~p].~n", delete, [Key, TargetName])
777     end,
778     del_from_succlist(tl(Succlist), Key).
779
780 log_get_data([]) -> ok;
781 log_get_data(DataList) ->
782     ?info_p("DATA-GET >>>> Value:[~p].~n", store, [hd(DataList)]),
783     log_get_data(tl(DataList)).

```

List 24: hm_edge.erl

```

1 % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2 % use this file except in compliance with the License. You may obtain a copy of
3 % the License at
4 %
5 % http://www.apache.org/licenses/LICENSE-2.0
6 %
7 % Unless required by applicable law or agreed to in writing, software
8 % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9 % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, hiro
15 %%% @doc I/F to start/stop Harmonia from outside of application
16 %%% (from shell, etc)
17 %%% @end
18 %%% Created : 2 Oct 2010 by Yoshihiro TANAKA <hirotnkg@gmail.com>
19 %%%-----
20 -module(hm_edge).
21 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
22 -vsn('0.1').
23 %% API
24 -export([
25     start/0,
26     stop/0,
27     stop/1
28     ]).
29
30 %%%=====
31 %%% API
32 %%%=====
33
34 %%-----
35 %% @doc
36 %% Starts the server
37 %%
38 %% @spec start() -> ok
39 %% @end
40 %%-----
41 start() ->
42     {ok, [[Nodename]]} = init:get_argument(sname),
43     error_logger:logfile({open, "SYS_INFO_" ++ Nodename ++ ".log"}),
44     case application:start(harmonia) of
45         ok -> ok;
46         {error, _Msg} -> init:stop(1)
47     end.
48
49 stop([RootNode]) -> rpc:call(RootNode, hm_edge, stop, []).
50
51 stop() ->
52     {ok, NameList} = hm_name_server:get_list(),
53     stop_in(NameList).
54 stop_in([]) -> ok;
55 stop_in([{_Name, NodeName}|NameList]) ->
56     case rpc:call(NodeName, application, stop, [harmonia]) of
57         ok ->
58             case rpc:call(NodeName, init, stop, []) of
59                 ok ->

```

```
60         io:fwrite("Stopped the node [%p].~n", [NodeName]);
61         {badrpc, Reason} ->
62             io:fwrite("Failed to stop the node [%p].~n", [Reason])
63         end;
64     {badrpc, Reason} ->
65         io:fwrite("Failed to stop harmonia on the node [%p].~n", [Reason])
66     end,
67     stop_in(NameList).
```

List 25: hm_event_mgr.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, hiro
15 %%% @doc event manager to add/delete event handlers
16 %%% @end
17 %%% Created : 2 Oct 2010 by Yoshihiro TANAKA <hirotnkg@gmail.com>
18 %%%-----
19 -module(hm_event_mgr).
20 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
21 -vsn('0.1').
22 %% API
23 -export([start_link/0,
24         add_file_handler/0,
25         add_handler/2,
26         delete_file_handler/0,
27         log/1,
28         log/3
29         ]).
30 %% gen_event callbacks
31 -export([init/1, handle_event/2, handle_call/2,
32         handle_info/2, terminate/2, code_change/3]).
33
34 -include("harmonia.hrl").
35
36 %%%=====
37 %%% API
38 %%%=====
39 log(Msg) ->
40     gen_event:notify(?MODULE, {log, Msg}).
41
42 log(Type, Format, DataList) ->
43     {{Year,Month,Day}, {Hour,Minute,Second}} = erlang:localtime(),
44     {_MegaSec, _Sec, Usec} = now(),
45     Buf = io_lib:format(
46         "~4..0w-~2..0w-~2..0w ~2..0w:~2..0w:~2..0w.~6..0w :: " ++ "[~p:~p:~p:~p]:~n" ++ \
47             ~Format,
48         [Year, Month, Day, Hour, Minute, Second, Usec] ++ DataList
49     ),
50     case Type of
51         ?LOG_INFO    -> log("[INFO]    === " ++ Buf);
52         ?LOG_WARNING -> log("[WARNING] === " ++ Buf);
53         ?LOG_ERROR   -> log("[ERROR]  === " ++ Buf)
54     end.
55 %%%=====
56 %%% gen_event callbacks
57 %%%=====
58

```

```

59 %%-----
60 %% @doc
61 %% Creates an event manager
62 %%
63 %% @spec start_link() -> {ok, Pid} | {error, Error}
64 %% @end
65 %%-----
66 start_link() ->
67     gen_event:start_link({local, ?MODULE}).
68
69 %%-----
70 %% @doc
71 %% Adds an event handler
72 %%
73 %% @spec add_handler() -> ok | {'EXIT', Reason} | term()
74 %% @end
75 %%-----
76 add_handler(Handler, Args) ->
77     gen_event:add_handler(?MODULE, Handler, Args).
78
79 add_file_handler() ->
80     gen_event:add_handler(?MODULE, hm_log_h_file, []).
81
82 delete_file_handler() ->
83     gen_event:delete_handler(?MODULE, hm_log_h_file, []).
84
85 %%-----
86 %% @private
87 %% @doc
88 %% Whenever a new event handler is added to an event manager,
89 %% this function is called to initialize the event handler.
90 %%
91 %% @spec init(Args) -> {ok, State}
92 %% @end
93 %%-----
94 init([]) ->
95     {ok, #state{}}.
96
97 %%-----
98 %% @private
99 %% @doc
100 %% Whenever an event manager receives an event sent using
101 %% gen_event:notify/2 or gen_event:sync_notify/2, this function is
102 %% called for each installed event handler to handle the event.
103 %%
104 %% @spec handle_event(Event, State) ->
105 %%                                     {ok, State} |
106 %%                                     {swap_handler, Args1, State1, Mod2, Args2} |
107 %%                                     remove_handler
108 %% @end
109 %%-----
110 handle_event(_Event, State) ->
111     {ok, State}.
112
113 %%-----
114 %% @private
115 %% @doc
116 %% Whenever an event manager receives a request sent using
117 %% gen_event:call/3,4, this function is called for the specified
118 %% event handler to handle the request.

```

```

119 %%
120 %% @spec handle_call(Request, State) ->
121 %%           {ok, Reply, State} |
122 %%           {swap_handler, Reply, Args1, State1, Mod2, Args2} |
123 %%           {remove_handler, Reply}
124 %% @end
125 %-----
126 handle_call(_Request, State) ->
127     Reply = ok,
128     {ok, Reply, State}.
129
130 %-----
131 %% @private
132 %% @doc
133 %% This function is called for each installed event handler when
134 %% an event manager receives any other message than an event or a
135 %% synchronous request (or a system message).
136 %%
137 %% @spec handle_info(Info, State) ->
138 %%           {ok, State} |
139 %%           {swap_handler, Args1, State1, Mod2, Args2} |
140 %%           remove_handler
141 %% @end
142 %-----
143 handle_info(_Info, State) ->
144     {ok, State}.
145
146 %-----
147 %% @private
148 %% @doc
149 %% Whenever an event handler is deleted from an event manager, this
150 %% function is called. It should be the opposite of Module:init/1 and
151 %% do any necessary cleaning up.
152 %%
153 %% @spec terminate(Reason, State) -> void()
154 %% @end
155 %-----
156 terminate(_Reason, _State) ->
157     ok.
158
159 %-----
160 %% @private
161 %% @doc
162 %% Convert process state when code is changed
163 %%
164 %% @spec code_change(OldVsn, State, Extra) -> {ok, NewState}
165 %% @end
166 %-----
167 code_change(_OldVsn, State, _Extra) ->
168     {ok, State}.

```

List 26: hm_log_h_file.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, hiro
15 %%% @doc logging file event handler
16 %%% @end
17 %%% Created : 2 Oct 2010 by Yoshihiro TANAKA <hirotnkg@gmail.com>
18 %%%-----
19 -module(hm_log_h_file).
20 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
21 -vsn('0.1').
22 -behaviour(gen_event).
23 %% API
24 -export([
25     add/0,
26     delete/0
27 ]).
28 %% gen_event callbacks
29 -export([init/1, handle_event/2, handle_call/2,
30     handle_info/2, code_change/3, terminate/2]).
31
32 -include("harmonia.hrl").
33
34 %%%=====
35 %%% API
36 %%%=====
37 add() ->
38     FileName = hm_misc:make_log_file_name(),
39     %io:format("~p~n", [FileName]),
40     hm_event_mgr:add_handler(?MODULE, list_to_atom(FileName)).
41
42 delete() ->
43     hm_event_mgr:delete_handler(?MODULE, []).
44
45 %%%=====
46 %%% gen_event callbacks
47 %%%=====
48
49 %%%-----
50 %% @private
51 %% @doc
52 %% Whenever a new event handler is added to an event manager,
53 %% this function is called to initialize the event handler.
54 %%
55 %% @spec init(Args) -> {ok, State}
56 %% @end
57 %%%-----
58 init(Filename) ->
59     %io:format("~p~n", [Filename]),

```

```

60     {ok, Fd} = file:open(Filename, [write]), % truncate if exists
61     {ok, {Fd, Filename}}.
62
63 %%-----
64 %% @private
65 %% @doc
66 %% Whenever an event manager receives an event sent using
67 %% gen_event:notify/2 or gen_event:sync_notify/2, this function is
68 %% called for each installed event handler to handle the event.
69 %%
70 %% @spec handle_event(Event, State) ->
71 %%     {ok, State} |
72 %%     {swap_handler, Args1, State1, Mod2, Args2} |
73 %%     remove_handler
74 %% @end
75 %%-----
76 handle_event({log, Msg}, {Fd, _Filename}=State) ->
77     io:format(Fd, "~s", [Msg]),
78     {ok, State}.
79
80 %%-----
81 %% @private
82 %% @doc
83 %% Whenever an event manager receives a request sent using
84 %% gen_event:call/3,4, this function is called for the specified
85 %% event handler to handle the request.
86 %%
87 %% @spec handle_call(Request, State) ->
88 %%     {ok, Reply, State} |
89 %%     {swap_handler, Reply, Args1, State1, Mod2, Args2} |
90 %%     {remove_handler, Reply}
91 %% @end
92 %%-----
93 handle_call(_Request, State) ->
94     Reply = ok,
95     {ok, Reply, State}.
96
97 %%-----
98 %% @private
99 %% @doc
100 %% This function is called for each installed event handler when
101 %% an event manager receives any other message than an event or a
102 %% synchronous request (or a system message).
103 %%
104 %% @spec handle_info(Info, State) ->
105 %%     {ok, State} |
106 %%     {swap_handler, Args1, State1, Mod2, Args2} |
107 %%     remove_handler
108 %% @end
109 %%-----
110 handle_info(_Info, State) ->
111     {ok, State}.
112
113 %%-----
114 %% @private
115 %% @doc
116 %% Whenever an event handler is deleted from an event manager, this
117 %% function is called. It should be the opposite of Module:init/1 and
118 %% do any necessary cleaning up.
119 %%

```

```

120 %% @spec terminate(Reason, State) -> void()
121 %% @end
122 %%-----
123 terminate(_Reason, {Fd, File}) ->
124     io:format("~p:closing ~p~n", [?MODULE, File]),
125     file:close(Fd),
126     ok.
127
128 %%-----
129 %% @private
130 %% @doc
131 %% Convert process state when code is changed
132 %%
133 %% @spec code_change(OldVsn, State, Extra) -> {ok, NewState}
134 %% @end
135 %%-----
136 code_change(_OldVsn, State, _Extra) ->
137     {ok, State}.

```

List 27: hm_log_h_term.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, hiro
15 %%% @doc logging terminal event handler
16 %%% @end
17 %%% Created : 2 Oct 2010 by Yoshihiro TANAKA <hirotnkg@gmail.com>
18 %%%-----
19 -module(hm_log_h_term).
20 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
21 -vsn('0.1').
22 -behaviour(gen_event).
23 %% API
24 -export([add_handler/0, delete_handler/0]).
25 %% gen_event callbacks
26 -export([init/1, handle_event/2, handle_call/2,
27         handle_info/2, code_change/3, terminate/2]).
28 -include("harmonia.hrl").
29
30 %%%=====
31 %%% API
32 %%%=====
33 add_handler() ->
34     hm_event_mgr:add_handler(?MODULE, []).
35
36 delete_handler() ->
37     hm_event_mgr:delete_handler(?MODULE, []).
38
39 %%%=====
40 %%% gen_event callbacks
41 %%%=====
42
43 %%-----
44 %% @private
45 %% @doc
46 %% Whenever a new event handler is added to an event manager,
47 %% this function is called to initialize the event handler.
48 %%
49 %% @spec init(Args) -> {ok, State}
50 %% @end
51 %%-----
52 init(_) ->
53     {ok, []}.
54
55 %%%-----
56 %% @private
57 %% @doc
58 %% Whenever an event manager receives an event sent using
59 %% gen_event:notify/2 or gen_event:sync_notify/2, this function is

```

```

60 %% called for each installed event handler to handle the event.
61 %%
62 %% @spec handle_event(Event, State) ->
63 %%     {ok, State} |
64 %%     {swap_handler, Args1, State1, Mod2, Args2} |
65 %%     remove_handler
66 %% @end
67 %%-----
68 handle_event({log, Msg}, State) ->
69     io:format("~s", [Msg]),
70     {ok, State}.
71
72 %%-----
73 %% @private
74 %% @doc
75 %% Whenever an event manager receives a request sent using
76 %% gen_event:call/3,4, this function is called for the specified
77 %% event handler to handle the request.
78 %%
79 %% @spec handle_call(Request, State) ->
80 %%     {ok, Reply, State} |
81 %%     {swap_handler, Reply, Args1, State1, Mod2, Args2} |
82 %%     {remove_handler, Reply}
83 %% @end
84 %%-----
85 handle_call(_Request, State) ->
86     Reply = ok,
87     {ok, Reply, State}.
88
89 %%-----
90 %% @private
91 %% @doc
92 %% This function is called for each installed event handler when
93 %% an event manager receives any other message than an event or a
94 %% synchronous request (or a system message).
95 %%
96 %% @spec handle_info(Info, State) ->
97 %%     {ok, State} |
98 %%     {swap_handler, Args1, State1, Mod2, Args2} |
99 %%     remove_handler
100 %% @end
101 %%-----
102 handle_info(_Info, State) ->
103     {ok, State}.
104
105 %%-----
106 %% @private
107 %% @doc
108 %% Whenever an event handler is deleted from an event manager, this
109 %% function is called. It should be the opposite of Module:init/1 and
110 %% do any necessary cleaning up.
111 %%
112 %% @spec terminate(Reason, State) -> void()
113 %% @end
114 %%-----
115 terminate(_Reason, {Fd, File}) ->
116     io:format("~p:closing ~p~n", [?MODULE, File]),
117     file:close(Fd),
118     ok.
119

```

```
120 %%-----
121 %% @private
122 %% @doc
123 %% Convert process state when code is changed
124 %%
125 %% @spec code_change(OldVsn, State, Extra) -> {ok, NewState}
126 %% @end
127 %%-----
128 code_change(_OldVsn, State, _Extra) ->
129     {ok, State}.
```

List 28: hm_misc.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, hiro
15 %%% @doc library module
16 %%% @end
17 %%% Created : 2 Oct 2010 by Yoshihiro TANAKA <hirotnkg@gmail.com>
18 %%%-----
19 -module(hm_misc).
20 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
21 -vsn('0.1').
22 %% API
23 -export([
24     check_exist/2,
25     check_pred_and_successor/1,
26     crypto_start/0,
27     crypto_stop/0,
28     del_dup/1,
29     get_digest/1,
30     get_first_alive_entry/1,
31     get_node/0,
32     get_rand_procname/0,
33     get_successor_alive/1,
34     get_vector_from_name/2,
35     is_alive/1,
36     is_pred_nil/1,
37     make_log_file_name/0,
38     make_request_list/2,
39     make_request_list_from_dt/2,
40     replace_nth/3,
41     search_table_attlist/2
42     ]).
43
44 -include("harmonia.hrl").
45
46 %%%=====
47 %%% API
48 %%%=====
49 get_successor_alive(State) when length(State#state.finger) > 0 ->
50     {SuccName, SuccVector} = hd(State#state.finger),
51     % check if the Succ is alive
52     BareName = list_to_atom(atom_to_list(SuccName) -- ?PROCESS_PREFIX),
53     case is_alive(BareName) of
54         true -> {ok, {SuccName, SuccVector}};
55         false -> {error, successor_dead}
56     end;
57 get_successor_alive(_State) -> {error, no_successor}.
58
59 get_vector_from_name(_RegName, []) -> {error, instance};

```

```

60 get_vector_from_name(RegName, Finger) ->
61     {Name, Vector} = hd(Finger),
62     case (Name == RegName) of
63         true -> {ok, Vector};
64         false -> get_vector_from_name(RegName, tl(Finger))
65     end.
66
67 get_node() ->
68     atom_to_list(node()).
69
70 get_rand_procname() ->
71     % bypassing get_name_list API for performance
72     Candidates =
73         case get(router_names) of
74             undefined ->
75                 NameList = global:registered_names(),
76                 {ok, RouterList} = get_first_fit_router(NameList, []), % clash, if error.
77                 put(router_names, RouterList),
78                 RouterList;
79             RouterList ->
80                 RouterList
81         end,
82     Name = lists:nth(random:uniform(length(Candidates)), Candidates),
83     {ok, Name}.
84
85 get_first_fit_router([], Candidates) -> {ok, Candidates};
86 get_first_fit_router([Candidate|NameList], Candidates) ->
87     case lists:prefix("hm_router_", atom_to_list(Candidate)) of
88         true ->
89             get_first_fit_router(NameList, [Candidate | Candidates]);
90         false ->
91             get_first_fit_router(NameList, Candidates)
92     end.
93
94 get_first_alive_entry([]) -> {error, none};
95 get_first_alive_entry([{Name, _Vector} = FirstNode|NodeList]) ->
96
97     BareName = list_to_atom(atom_to_list(Name) -- ?PROCESS_PREFIX),
98     case is_alive(BareName) of
99         true -> FirstNode;
100        false ->
101            get_first_alive_entry(NodeList)
102    end.
103
104 is_alive(BareName) ->
105     % 'foo', 'hm_router_foo', both type of name is available,
106     % but need to specify 'foo' type name in case it's dead
107     % to remove from name server
108     case global:whereis_name(BareName) of
109         undefined ->
110             hm_name_server:unregister(BareName),
111             false;
112         _ -> true
113     end.
114
115 check_exist(_AppName, []) -> false;
116 check_exist(AppName, [{AppName,_,_}|_Applist]) -> true;
117 check_exist(AppName, Applist) ->
118     check_exist(AppName, tl(Applist)).
119

```



```

120 crypto_start() ->
121   Apps = application:which_applications(),
122   case check_exist(crypto, Apps) of
123     true -> {ok, exists};
124     false -> crypto:start(), {ok, started}
125   end.
126
127 crypto_stop() ->
128   Apps = application:which_applications(),
129   case check_exist(crypto, Apps) of
130     true -> crypto:stop(), {ok, stopped};
131     false -> {ok, not_exists}
132   end.
133
134 replace_nth(N, Val, List) ->
135   replace_nth_in(1, N, length(List), Val, List, []).
136
137 check_pred_and_successor(State) ->
138   % includes info about predecessor also
139   case State#state.predecessor == nil of
140     true ->
141       Pred = pred_is_nil;
142     false ->
143       Pred = pred_is_not_nil
144   end,
145   case length(State#state.finger) > 0 of
146     true -> {succ_exists, Pred};
147     false -> {no_succ_exists, Pred}
148   end.
149
150 is_pred_nil(State) ->
151   case State#state.predecessor == nil of
152     true -> true;
153     false -> false
154   end.
155
156 make_request_list(TargetName, SuccListTarget) ->
157   % this order is important, don't sort !!
158   % also, if elements of list duplicate, system ends up with trying same
159   % operation multiple times, so this list must have uniq elements
160   del_dup([TargetName, instance] | SuccListTarget []).
161
162 make_request_list_from_dt(DomainName, TableName) ->
163   {ok, NodeName, SuccList} = hm_router:lookup_with_succlist(list_to_atom(DomainName ++ \
164     →TableName)),
165   make_request_list(NodeName, SuccList).
166
167 get_digest(Key) when is_integer(Key) ->
168   <<Vector:160>> = crypto:sha(integer_to_list(Key)),
169   Vector rem ?max_key_value;
170 get_digest(Key) when is_atom(Key) ->
171   <<Vector:160>> = crypto:sha(atom_to_list(Key)),
172   Vector rem ?max_key_value;
173 get_digest(Key) when is_list(Key) ->
174   <<Vector:160>> = crypto:sha(Key),
175   Vector rem ?max_key_value.
176
177 search_table_attlist(DTName, TblList) ->
178   case lists:keysearch(DTName, 1, TblList) of

```

```

179         false -> {error, no_table};
180         {value, {DTName, AttList}} -> {ok, DTName, AttList}
181     end.
182
183 del_dup(List) ->
184     lists:reverse(del_dup_in(List)).
185
186 make_log_file_name() ->
187     {ok, {_, Name}} = hm_config:get(name),
188     {ok, {_, Logfile}} = hm_config:get(logfile),
189     {ok, {_, Ext}} = hm_config:get(logfile_ext),
190     {ok, {_, Logdir}} = hm_config:get(logdir),
191     Logdir ++ Logfile ++ "_" ++ atom_to_list(Name) ++ Ext.
192
193 %%%=====
194 %%% Internal functions
195 %%%=====
196 replace_nth_in(C, N, Len, _Val, _Old, New) when (C > N) and (C > Len) ->
197     lists:reverse(New);
198 replace_nth_in(C, C, Len, Val, Old, New) ->
199     case (length(Old) > 0) of
200         true -> replace_nth_in(C+1, C, Len, Val, tl(Old), [Val|New]);
201         false -> replace_nth_in(C+1, C, Len, Val, [], [Val|New])
202     end;
203 replace_nth_in(C, N, Len, Val, [], New) ->
204     replace_nth_in(C+1, N, Len, Val, [], [nil|New]);
205 replace_nth_in(C, N, Len, Val, Old, New) ->
206     case (length(Old) > 0) of
207         true -> replace_nth_in(C+1, N, Len, Val, tl(Old), [hd(Old)|New]);
208         false -> replace_nth_in(C+1, N, Len, Val, [], [hd(Old)|New])
209     end.
210
211 del_dup_in(Old) ->
212     lists:foldl(
213         fun({Node,Vec}, Acc) ->
214             case lists:keysearch(Node, 1, Acc) of
215                 false -> [{Node,Vec}|Acc];
216                 {value, _} -> Acc
217             end
218         end, [], Old
219     ).

```

List 29: hm_name_server.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, Yoshihiro TANAKA
15 %%% @doc
16 %%%   manage names of the chord ring
17 %%% @end
18 %%% Created : 2 Oct 2010 by Yoshihiro <hirotnkg@gmail.com>
19 %%%-----
20 -module(hm_name_server).
21 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
22 -behaviour(gen_server).
23 -vsn('0.1').
24 %% API
25 -export(
26     [
27         get_list/0,
28         name/1,
29         register/1,
30         unregister/1,
31         start_link/0,
32         stop/0
33     ]).
34 %% gen_server callbacks
35 -export([init/1, handle_call/3, handle_cast/2, handle_info/2,
36         terminate/2, code_change/3]).
37
38 -include("harmonia.hrl").
39
40 %%%=====
41 %%% API
42 %%%=====
43
44 %%-----
45 %% @doc
46 %%   Starts the server
47 %%
48 %% @spec start_link(Env) -> {ok, Pid} | ignore | {error, Error}
49 %% @end
50 %%-----
51 start_link() ->
52     gen_server:start_link({global, ?MODULE}, ?MODULE, [], []).
53
54 stop() ->
55     gen_server:cast(?MODULE, stop).
56
57 register({Name, NodeName}) ->
58     gen_server:call({global, ?name_server}, {register_name, {Name, NodeName}}).
59

```

```

60 unregister(Name) ->
61     gen_server:call({global, ?name_server}, {unregister_name, {name, Name}}).
62
63 get_list() ->
64     gen_server:call({global, ?name_server}, get_name_list).
65
66 name(RegName) -> list_to_atom(atom_to_list(?MODULE) ++ "_" ++ atom_to_list(RegName)).
67
68 %%%=====
69 %%% gen_server callbacks
70 %%%=====
71
72 %%%-----
73 %%% @private
74 %%% @doc
75 %%% Initializes the server
76 %%%
77 %%% @spec init(Args) -> {ok, State} |
78 %%%                      {ok, State, Timeout} |
79 %%%                      ignore |
80 %%%                      {stop, Reason}
81 %%% @end
82 %%%-----
83 init(_) -> {ok, []}.
84
85
86 %%%-----
87 %%% @private
88 %%% @doc
89 %%% This function is called by a gen_server when it is about to
90 %%% terminate. It should be the opposite of Module:init/1 and do any
91 %%% necessary cleaning up. When it returns, the gen_server terminates
92 %%% with Reason. The return value is ignored.
93 %%%
94 %%% @spec terminate(Reason, State) -> void()
95 %%% @end
96 %%%-----
97 terminate(Reason, State) ->
98     ?info_p("terminate:Reason:[~p] State:[~p]~n", none, [Reason, State]),
99     ok.
100
101 %%%-----
102 %%% @private
103 %%% @doc
104 %%% Handling cast messages
105 %%%
106 %%% @spec handle_cast(Msg, State) -> {noreply, State} |
107 %%%                      {noreply, State, Timeout} |
108 %%%                      {stop, Reason, State}
109 %%% @end
110 %%%-----
111 handle_cast(stop, State) ->
112     {stop, normal, State}.
113
114 %%%-----
115 %%% @private
116 %%% @doc
117 %%% Handling call messages
118 %%%
119 %%% @spec handle_call(Request, From, State) ->

```

```

120 %%                                {reply, Reply, State} |
121 %%                                {reply, Reply, State, Timeout} |
122 %%                                {noreply, State} |
123 %%                                {noreply, State, Timeout} |
124 %%                                {stop, Reason, Reply, State} |
125 %%                                {stop, Reason, State}
126 %% @end
127 %%-----
128 handle_call(get_name_list, _From, State) ->
129     {reply, {ok, State}, State};
130
131 handle_call({register_name, {RegName, NodeName}}, _From, State) ->
132     {reply, {ok, {RegName, NodeName}}, [{RegName, NodeName}|State]};
133
134 handle_call({unregister_name, {name, RegName}}, _From, State) ->
135     {reply, ok, lists:keydelete(RegName, 1, State)}.
136
137 %%-----
138 %% @private
139 %% @doc
140 %% Handling all non call/cast messages
141 %%
142 %% @spec handle_info(Info, State) -> {noreply, State} |
143 %%                                   {noreply, State, Timeout} |
144 %%                                   {stop, Reason, State}
145 %% @end
146 %%-----
147 handle_info(_Info, State) ->
148     {noreply, State}.
149
150 %%-----
151 %% @private
152 %% @doc
153 %% Convert process state when code is changed
154 %%
155 %% @spec code_change(OldVsn, State, Extra) -> {ok, NewState}
156 %% @end
157 %%-----
158 code_change(_OldVsn, State, _Extra) ->
159     {ok, State}.

```

List 30: hm_qp.erl

```

1 % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2 % use this file except in compliance with the License. You may obtain a copy of
3 % the License at
4 %
5 % http://www.apache.org/licenses/LICENSE-2.0
6 %
7 % Unless required by applicable law or agreed to in writing, software
8 % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9 % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, hiro
15 %%% @doc query parser for ets
16 %%% convert query condition into match specification of ets
17 %%%
18 %%% Sts = Exp , {'or' Exp};
19 %%% Exp = Factor , {'and' Factor};
20 %%% Factor = Fname RelOp Term | '(' , Sts , ')';
21 %%% Fname = Char , {Char|Number};
22 %%% Term = Atom|Number|String;
23 %%% RelOp = '=='/'!='/'<='/'>='/'>'/'<';
24 %%% Number = Digit , {Digit};
25 %%% String = '[' , {Char|Digit} , ']';
26 %%% Atom = ''' , {Char|Digit} , ''';
27 %%% Digit = '0-9';
28 %%% Char = 'a-Z';
29 %%%
30 %%% @end
31 %%% Created : 2 Oct 2010 by Yoshihiro TANAKA <hirotnkg@gmail.com>
32 %%%-----
33 -module(hm_qp).
34 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
35 -vsn('0.1').
36 %% API
37 -export([
38     eval/1,
39     parse/1,
40     scan/2,
41     scan/3
42     ]).
43 -include_lib("eunit/include/eunit.hrl").
44
45 -define(CHARACTERS, "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_").
46 -define(NUMBERS, "0123456789").
47
48 %%%=====
49 %%% API
50 %%%=====
51 parse(Tokens) ->
52     {Result, []} = parse_sts(Tokens, []),
53     [Result].
54
55 eval({Op}) ->
56     {result, Op};
57 eval({Rel, Op1, Op2}) ->
58     Ret1 = eval(Op1),
59     Ret2 = eval(Op2),

```

```

60     {result, Rel, Ret1, Ret2};
61 eval(Op) ->
62     {result, Op}.
63
64 scan(Query, {IsKey, AttList})->
65     Tokens = scan(Query, [], []),
66     scan_ret(Tokens, {IsKey, AttList}).
67
68 scan([], Cur, Tokens) -> Tokens ++ Cur;
69
70 % scan atom
71 scan([$'|Query], [], Tokens) ->
72     scan(Query, [{atom, []}], Tokens);
73 scan([$'|Query], [{atom, Cur}], Tokens) ->
74     scan(Query, [], Tokens++[{atom, Cur}]);
75 scan([$'|Query], Cur, Tokens) ->
76     scan(Query, [{atom, []}], Tokens++Cur);
77 scan([Char|Query], [{atom, Cur}], Tokens) ->
78     scan(Query, [{atom, [Char|Cur]}], Tokens);
79
80 % scan string
81 scan([$|Query], [], Tokens) ->
82     scan(Query, [{string, []}], Tokens);
83 scan([$|Query], [{string, Cur}], Tokens) ->
84     scan(Query, [], Tokens++[{string, lists:reverse(Cur)}]);
85 scan([$|Query], Cur, Tokens) ->
86     scan(Query, [{string, []}], Tokens++Cur);
87 scan([Char|Query], [{string, Cur}], Tokens) ->
88     scan(Query, [{string, [Char|Cur]}], Tokens);
89
90
91 % general cases
92 scan(" " ++ Query, [], Tokens) ->
93     scan(Query, [], Tokens);
94
95 scan(" " ++ Query, Cur, Tokens) ->
96     scan(Query, [], Tokens++Cur);
97
98 scan("and" ++ Query, [], Tokens) ->
99     scan(Query, [], Tokens++[{and_op, "and"}]);
100 scan("or" ++ Query, [], Tokens) ->
101     scan(Query, [], Tokens++[{or_op, "or"}]);
102
103 scan(">=" ++ Query, Cur, Tokens) ->
104     scan(Query, [], Tokens++Cur++[{relational_operator, ">="}]);
105 scan("<=" ++ Query, Cur, Tokens) ->
106     scan(Query, [], Tokens++Cur++[{relational_operator, "<="}]);
107 scan(">" ++ Query, Cur, Tokens) ->
108     scan(Query, [], Tokens++Cur++[{relational_operator, ">"}]);
109 scan("<" ++ Query, Cur, Tokens) ->
110     scan(Query, [], Tokens++Cur++[{relational_operator, "<"}]);
111 scan("==" ++ Query, Cur, Tokens) ->
112     scan(Query, [], Tokens++Cur++[{relational_operator, "=="}]);
113 scan("!=" ++ Query, Cur, Tokens) ->
114     scan(Query, [], Tokens++Cur++[{relational_operator, "!="}]);
115 scan("<" ++ Query, Cur, Tokens) ->
116     scan(Query, [], Tokens++Cur++[{relational_operator, "<"}]);
117 scan(">" ++ Query, Cur, Tokens) ->
118     scan(Query, [], Tokens++Cur++[{relational_operator, ">"}]);
119

```

```

120 scan("." ++ Query, Cur, Tokens) ->
121     scan(Query, [], Tokens++Cur++[{dot_op, "."}]);
122
123 scan("(" ++ Query, Cur, Tokens) ->
124     scan(Query, [], Tokens++Cur++[{left_paren, "("}]);
125 scan(")" ++ Query, Cur, Tokens) ->
126     scan(Query, [], Tokens++Cur++[{right_paren, "("}]);
127
128 scan([Char|Query], [], Tokens) ->
129     case char_type(Char) of
130         character ->
131             scan(Query, [{identifier, [Char]}], Tokens);
132         number ->
133             scan(Query, [{const, [Char]}], Tokens);
134         _ ->
135             error
136     end;
137
138 scan([Char|Query], [{identifier, String}], Tokens) ->
139     scan(Query, [{identifier, [Char|String]}], Tokens);
140 scan([Char|Query], [{const, String}], Tokens) ->
141     case char_type(Char) of
142         number ->
143             scan(Query, [{const, [Char|String]}], Tokens);
144         _ -> error
145     end.
146
147 %%%=====
148 %%% Internal functions
149 %%%=====
150 scan_ret(Tokens, {IsKey, AttList}) ->
151     scan_ret_in(Tokens, {IsKey, AttList}, []).
152
153 scan_ret_in([], {_IsKey, _AttList}, Result) -> lists:reverse(Result);
154 scan_ret_in([{identifier, String}|Tokens], {IsKey, AttList}, Result) when IsKey == index ->
155     Token = lists:reverse(String),
156     case find_nth(Token, AttList) of
157         {no_field, _} ->
158             scan_ret_in(
159                 Tokens,
160                 {IsKey, AttList},
161                 [{identifier, list_to_atom(Token)}|Result]
162             );
163
164         {false, _N} ->
165             %% if the field name is NOT key field, then ignore this field condition
166             %% for key table search
167             {_Discard, NewTokens} = lists:split(2, Tokens),
168             scan_ret_in(
169                 NewTokens,
170                 {IsKey, AttList},
171                 [{atom, true}, {relational_operator, "=="}, {atom, true}|Result]
172             );
173
174         {true, N} ->
175             %% add 1, because $1 is the table name
176             scan_ret_in(
177                 Tokens,
178                 {IsKey, AttList},
179                 [{identifier, list_to_atom("$" ++ integer_to_list(N+1))}|Result]

```



```

180         )
181     end;
182 scan_ret_in([{{identifier, String}}|Tokens], {IsKey, AttList}, Result) ->
183     Token = lists:reverse(String),
184     NewToken =
185         case find_nth(Token, AttList) of
186             {no_field, _} ->
187                 {identifier, list_to_atom(Token)};
188             {_, N} ->
189                 %% add 1, because $1 is the table name
190                 {identifier, list_to_atom("$" ++ integer_to_list(N+1))}
191         end,
192     scan_ret_in(Tokens, {IsKey, AttList}, [NewToken|Result]);
193 scan_ret_in([{{const, String}}|Tokens], {IsKey, AttList}, Result) ->
194     NewToken = {const, list_to_integer(lists:reverse(String))},
195     scan_ret_in(Tokens, {IsKey, AttList}, [NewToken|Result]);
196 scan_ret_in([{{atom, String}}|Tokens], {IsKey, AttList}, Result) ->
197     NewToken = {atom, list_to_atom(lists:reverse(String))},
198     scan_ret_in(Tokens, {IsKey, AttList}, [NewToken|Result]);
199 scan_ret_in([{{Type, String}}|Tokens], {IsKey, AttList}, Result) ->
200     NewToken = {Type, String},
201     scan_ret_in(Tokens, {IsKey, AttList}, [NewToken|Result]).
202
203 char_type(Char) ->
204     case lists:member(Char, ?CHARACTERS) of
205         true -> character;
206         false ->
207             case lists:member(Char, ?NUMBERS) of
208                 true -> number;
209                 false ->
210                     case Char of
211                         $' -> single_quote;
212                         $" -> double_quote;
213                         _ -> other_char
214                     end
215             end
216     end.
217
218 find_nth(Token, AttList) ->
219     find_nth_in(Token, AttList, 1).
220
221 find_nth_in(Token, [], _N) -> {no_field, Token};
222 find_nth_in(Token, [{FieldName, IsKey, _} | AttList], N) ->
223     case Token ==: FieldName of
224         true -> {IsKey, N};
225         false ->
226             find_nth_in(Token, AttList, N+1)
227     end.
228
229 parse_sts(Tokens, []) ->
230     case parse_exp(Tokens, []) of
231         {Exp1, [{or_op, _}|Tokens2]} ->
232             {Exp2, Tokens3} = parse_sts(Tokens2, []),
233             {'or', Exp1, Exp2}, Tokens3;
234         {Exp1, [{right_paren, _}|Tokens2]} ->
235             {Exp1, Tokens2};
236         {Exp1, []} -> {Exp1, []};
237         _ -> {error, parse_sts, Tokens}
238     end.
239

```

```

240 parse_exp(Tokens, []) ->
241   case parse_factor(Tokens, []) of
242     {Factor1, [{and_op,_}|Tokens2]} ->
243       {Factor2, Tokens3} = parse_exp(Tokens2, []),
244       {{'and', Factor1, Factor2}, Tokens3};
245     {Factor1, Tokens1} -> {Factor1, Tokens1};
246   _ -> {error, parse_exp, Tokens}
247   end.
248
249 parse_factor([{left_paren, _} | Tokens], []) ->
250   {Exp, Tokens2} = parse_sts(Tokens, []),
251   {Exp, Tokens2};
252 parse_factor(Tokens, []) ->
253   {Term, [RelOp|Tokens2]} = parse_term(Tokens, []),
254   case RelOp of
255     {relational_operator, Op} ->
256       {Term2, Tokens3} = parse_term(Tokens2, []),
257       AtomOp = list_to_atom(Op),
258       {{AtomOp, Term, Term2}, Tokens3};
259   _ -> {error, parse_factor, Tokens2}
260   end.
261
262
263 parse_term([{identifier,_}=Tble,{dot_op, _},{identifier, _}=Fld|Tokens], []) ->
264   {{table_field, Tble, Fld}, Tokens};
265 parse_term([{identifier,Val}|Tokens], []) ->
266   {Val, Tokens};
267 parse_term([{atom,Val}|Tokens], []) ->
268   {Val, Tokens};
269 parse_term([{string,Val}|Tokens], []) ->
270   {Val, Tokens};
271 parse_term([{const,Val}|Tokens], []) ->
272   {{const, Val}, Tokens}.
273
274 %%%=====
275 %%% EUnit test functions
276 %%%=====
277
278 rangeq_test_() ->
279   [
280     ?_assertEqual({ok, [[xxx,32,textfile1]]}, hm_cli:get("Domain1", "Tbl2", "Fld2 == \
281       →32")),
282     ?_assertEqual({ok, [[yyy,150,textfile2]]}, hm_cli:get("Domain1", "Tbl2", "Fld2 == \
283       →150")),
284     ?_assertEqual({ok, [[zzz,3000,textfile3]]}, hm_cli:get("Domain1", "Tbl2", "Fld2 == \
285       →3000")),
286     ?_assertEqual({ok, [[aaa,9000,textfile4]]}, hm_cli:get("Domain1", "Tbl2", "Fld2 == \
287       →9000"))
288   ].

```

List 31: hm_router.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, hiro
15 %%% @doc Chord base router
16 %%%
17 %%% Chord algorithm is based on the following paper:
18 %%% [1] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R.,
19 %%%     Kaashoek, M. F., Dabek, F., and Balakrishnan, H.
20 %%%     2003.
21 %%%     Chord: a scalable peer-to-peer lookup protocol for internet applications.
22 %%%     IEEE/ACM Trans. Netw. 11, 1 (Feb. 2003), 17-32.
23 %%%     DOI= http://dx.doi.org/10.1109/TNET.2002.808407
24 %%%
25 %%% in this source, when reffered to as [1], it means above paper.
26 %%%
27 %%% @end
28 %%% Created : 2 Oct 2010 by Yoshihiro TANAKA <hirotnkg@gmail.com>
29 %%%-----
30 -module(hm_router).
31 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
32 -behaviour(gen_server).
33 -vsn('0.1').
34
35 %% API
36 -export([
37     check_pred/1,
38     get_predecessor/1,
39     stabilize/2,
40     copy_succlist/1,
41     fix_finger_set/4,
42     set_succlist/2,
43     find_successor/3,
44     lookup/1,
45     lookup_with_succlist/1,
46     name/1,
47     start/1,
48     start_link/1,
49     state_info/1,
50     state_info/2,
51     stop/0,
52     stop/1
53 ]).
54 %% gen_server callbacks
55 -export([init/1, handle_call/3, handle_cast/2, handle_info/2,
56     terminate/2, code_change/3]).
57
58 -include("harmonia.hrl").
59

```

```

60 %%%=====
61 %%% API
62 %%%=====
63 start([]) -> ok;
64 start(NodeNameList) ->
65     start_link(hd(NodeNameList)),
66     start(tl(NodeNameList)).
67
68 start_link({create, RegName}) ->
69     gen_server:start_link( {global, name(RegName)}, ?MODULE, {create, RegName}, []);
70 start_link({join, RegName, RootName}) ->
71     gen_server:start_link( {global, name(RegName)}, ?MODULE, {{join, RootName}, RegName}, []\
    →).
72
73 stop() ->
74     gen_server:cast(?MODULE, stop).
75
76 stop(RegName) ->
77     gen_server:cast({global, name(RegName)}, stop).
78
79 %%%-----
80 %%% @spec(lookup(Key::atom()) -> atom()).
81 %%% @doc returns successor name in the form of harmonia_foo(bare name)
82 %%% @end
83 %%%-----
84 lookup(Key) ->
85     KeyVector = hm_misc:get_digest(Key),
86     {ok, RegName} = hm_misc:get_rand_procname(),
87     {SuccName, _} = gen_server:call({global, RegName}, {find_successor, KeyVector, nil}),
88     SuccName.
89
90 %%%-----
91 %%% @spec(lookup_with_succlist(Key::atom()) -> {ok, atom(), SuccList::list()}).
92 %%% @doc returns successor name with its successor list
93 %%% @end
94 %%%-----
95 lookup_with_succlist(Key) ->
96     KeyVector = hm_misc:get_digest(Key),
97     {ok, RegName} = hm_misc:get_rand_procname(),
98     {{SuccName, _}, SuccList} =
99         gen_server:call({global, RegName}, {find_successor_with_succlist, KeyVector, nil}),
100     {ok, SuccName, SuccList}.
101
102 %%%-----
103 %%% @spec(find_successor(RegName::atom(), NodeVector::integer(),
104 %%%                               Current::integer()) -> Successor).
105 %%% @doc this routine is mainly called by hm_stabilizer module
106 %%% @end
107 %%%-----
108 find_successor(RegName, NodeVector, Current) ->
109     gen_server:call({global, hm_router:name(RegName)}, {find_successor, NodeVector, Current}\
    →).
110
111 state_info(RegName) ->
112     gen_server:call({global, name(RegName)}, state_info).
113 state_info(RegName, NodeName) ->
114     gen_server:call({name(RegName), NodeName}, state_info).
115
116 check_pred(RegName) ->
117     gen_server:cast({global, hm_router:name(RegName)}, check_pred).

```

```

118
119 get_predecessor(SuccName) ->
120     gen_server:call({global, SuccName}, get_predecessor).
121
122 copy_succlist(SuccName) ->
123     gen_server:call({global, SuccName}, copy_succlist).
124
125 set_succlist(MyNodeName, NewSuccList) ->
126     gen_server:cast({global, MyNodeName}, {set_succlist, NewSuccList}).
127
128 stabilize(RegName, PredOfSucc) ->
129     gen_server:cast({global, hm_router:name(RegName)}, {stabilize, PredOfSucc}).
130
131 %%-----
132 %% @spec(fix_finger_set(RegName::atom(), Current::integer(), NewSucc, Finger) -> ok).
133 %% @doc replace nth entry in finger list
134 %% @end
135 %%-----
136 fix_finger_set(RegName, Current, NewSucc, Finger) ->
137     gen_server:cast({global, hm_router:name(RegName)}, {fix_finger, Current, NewSucc, Finger\
->}),
138     ok.
139
140 name(Name) -> list_to_atom("hm_router_" ++ atom_to_list(Name)).
141
142 %%%=====
143 %%% gen_server callbacks
144 %%%=====
145
146 %%-----
147 %% @private
148 %% @doc
149 %% Initializes the server
150 %%
151 %% @spec init(Args) -> {ok, State} |
152 %%                      {ok, State, Timeout} |
153 %%                      ignore |
154 %%                      {stop, Reason}
155 %% @end
156 %%-----
157 init({Op, RegName}) ->
158     hm_misc:crypto_start(),
159     NodeName = name(RegName),
160     NodeVector = hm_misc:get_digest(NodeName),
161     State = #state{node_name = NodeName, node_vector = NodeVector},
162     NewState =
163         case Op of
164             create ->
165                 State#state{finger = [{NodeName, NodeVector}]};
166             {join, RootNodeName} ->
167                 NewSucc = gen_server:call(
168                     {global, name(RootNodeName)},
169                     {find_successor, NodeVector, nil}
170                 ),
171                 State#state{finger = [NewSucc]}
172         end,
173     {ok, NewState}.
174
175 %%-----
176 %% @private

```



```

236 %%                                {noreply, State} |
237 %%                                {noreply, State, Timeout} |
238 %%                                {stop, Reason, Reply, State} |
239 %%                                {stop, Reason, State}
240 %% @end
241 %%-----
242 handle_call({find_successor, NodeVector, CurrentFix}, From, State) ->
243     find_successor_handle_call_in(find_successor, NodeVector, CurrentFix, From, State);
244
245 handle_call({find_successor_with_succlist, NodeVector, CurrentFix}, From, State) ->
246     find_successor_handle_call_in(find_successor_with_succlist, NodeVector, CurrentFix, From\
        →, State);
247
248 handle_call(copy_succlist, _From, State) ->
249     {reply, State#state.succlist, State};
250
251 handle_call(get_predecessor, _From, State) ->
252     {reply, State#state.predecessor, State};
253
254 handle_call(get_successor, _From, State) ->
255     {reply, hm_misc:get_successor_alive(State), State};
256
257 handle_call(state_info, _From, State) ->
258     {reply, {ok, State}, State}.
259
260 %%-----
261 %% @private
262 %% @doc
263 %% Handling all non call/cast messages
264 %%
265 %% @spec handle_info(Info, State) -> {noreply, State} |
266 %%                                {noreply, State, Timeout} |
267 %%                                {stop, Reason, State}
268 %% @end
269 %%-----
270 handle_info(_Info, State) ->
271     {noreply, State}.
272
273 %%-----
274 %% @private
275 %% @doc
276 %% Convert process state when code is changed
277 %%
278 %% @spec code_change(OldVsn, State, Extra) -> {ok, NewState}
279 %% @end
280 %%-----
281 code_change(_OldVsn, State, _Extra) ->
282     {ok, State}.
283
284 %%%=====
285 %%% Internal functions
286 %%%=====
287
288 %%-----
289 %% @doc this API is called by handle_call immediately,
290 %%      and forwarding to other nodes starts from this point
291 %%
292 %% algorithm: refer to [1]:
293 %% // ask node n to find the successor of id
294 %% n.find_successor(id)

```

```

295 %%   if(id in (n, successor])
296 %%       return successor
297 %%   else
298 %%       n' = closest_preceding_node(id)
299 %%       return n'.find_successor(id)
300 %%
301 %% @end
302 %%-----
303 find_successor_handle_call_in(Handlecall_Key, NodeVector, CurrentFix, From, State) ->
304     NewState =
305         case CurrentFix of
306             nil -> State; % inquired by other, no change to current_fix
307             _ -> State#state{current_fix = CurrentFix} % inquired by self-stabilizer
308         end,
309         case NodeVector == State#state.node_vector of
310
311             % when node vector == current node, return current node
312             true ->
313                 case Handlecall_Key == find_successor_with_succlist of
314                     true -> {reply, {{State#state.node_name, State#state.node_vector}, State#\
315                                     ->state.succlist}, NewState};
316                     false -> {reply, {State#state.node_name, State#state.node_vector}, NewState}
317                 end;
318             false ->
319                 return_successor_info(
320                     Handlecall_Key,
321                     find_successor_in(NodeVector, State),
322                     NodeVector,
323                     From,
324                     NewState
325                 )
326             end.
327
328 %%-----
329 %% @spec(find_successor_in(NodeVector, State) -> {found_me, {Succ, SuccVector}} |
330 %%                                               {found_other, TargetNode} |
331 %%                                               {not_found, Node}).
332 %% @doc this API seaches the routing information for the 'NodeVector' in this
333 %%       node's 'State' and returns 3 types of judgements described above
334 %% @end
335 %%-----
336 find_successor_in(NodeVector, State) ->
337     {Succ, SuccVector} =
338         case hm_misc:get_first_alive_entry(State#state.finger) of
339             {error, none} ->
340                 {State#state.node_name, State#state.node_vector};
341             {TmpSucc, TmpSuccVector} ->
342                 {TmpSucc, TmpSuccVector}
343         end,
344         case is_between2(State#state.node_vector, NodeVector, SuccVector) of
345             % my successor is in charge of the node vector
346             true ->
347                 {found_other, {Succ, SuccVector}};
348
349             % my successor is not the successor of NodeVector,
350             % search in the finger list later the successor
351             false ->
352                 case closest_predecessor(State, NodeVector) of
353                     % no info available

```



```

354         nil ->
355             {found_me, {State#state.node_name, State#state.node_vector}};
356
357         % ask this closest pred to *other* node(need to pass nil)
358         {RetNodeName, RetNodeVector} ->
359             {not_found, {RetNodeName, RetNodeVector}}
360     end
361 end.
362
363 return_successor_info(find_successor_with_succlist, RetVal, NodeVector, From, State) ->
364     case RetVal of
365         {found_me, MyNode} ->
366             {reply, {MyNode, State#state.succlist}, State};
367         {_, {InqNode, _InqVector}} ->
368             gen_server:cast({global, InqNode},
369                             {find_successor_ask_other,
370                              find_successor_with_succlist, NodeVector, From}),
371             {noreply, State}
372     end;
373 return_successor_info(find_successor, RetVal, NodeVector, From, State) ->
374     case RetVal of
375         {not_found, {InqNode, _InqVector}} ->
376             gen_server:cast({global, InqNode},
377                             {find_successor_ask_other,
378                              find_successor, NodeVector, From}),
379             {noreply, State};
380         {_, NewSucc} -> % both found_other & found_me, in case only name is needed
381             {reply, NewSucc, State}
382     end.
383
384 %%-----
385 %% @doc it is trying to find the closest preceding node in local state.
386 %%
387 %% (1) the successor of the target id is in this node's finger table range:
388 %%     in this case I want to forward the query to the closest preceding node.
389 %%
390 %% (2) the successor of the target id is out of my finger table range:
391 %%     in this case, from the algorithm, the query is forwarded to
392 %%     the node which is the farthest node from this node in the local
393 %%     finger table
394 %%
395 %% algorithm: refer to [1]:
396 %%
397 %% // search the local table for the highest predecessor of id
398 %% n.closest_preceding_node(id)
399 %%   for i = m downto 1
400 %%       if(finger[i] in (n, id))
401 %%           return finger[i]
402 %%   return n
403 %% @end
404 %%-----
405 closest_predecessor(State, NodeVector) ->
406     closest_predecessor_in(State#state.node_vector,
407                             NodeVector,
408                             lists:reverse(State#state.finger)).
409
410 closest_predecessor_in(_LocalVector, _NodeVector, []) -> nil;
411 closest_predecessor_in(LocalVector, NodeVector, FingerList) ->
412     {FingerName, FingerVector} = hd(FingerList),
413     case is_between(LocalVector, FingerVector, NodeVector) of

```

```

414         true ->
415             BareName = list_to_atom( atom_to_list(FingerName) -- ?PROCESS_PREFIX ),
416             case hm_misc:is_alive(BareName) of
417                 true -> {FingerName, FingerVector};
418                 false ->
419                     closest_predecessor_in(LocalVector, NodeVector, tl(FingerList))
420             end;
421         false ->
422             closest_predecessor_in(LocalVector, NodeVector, tl(FingerList))
423     end.
424
425     %%-----
426     %% @doc condition: (From, To) = {Target | From < Target < To}
427     %% used by closest_preceding_node, notify algorithm
428     %% @end
429     %%-----
430     is_between(From, _Target, To) when From == To -> true;
431     is_between(From, Target, To) when From < To ->
432         case (From < Target) and (Target < To) of
433             true -> true;
434             false -> false
435         end;
436     is_between(From, Target, To) when From > To ->
437         case ((From < Target) or (Target < To)) of
438             true -> true;
439             false -> false
440         end.
441
442     %%-----
443     %% @doc condition: (From, To) = {Target | From < Target <= To}
444     %% used by find_successor algorithm
445     %% @end
446     %%-----
447     is_between2(From, _Target, To) when From == To ->
448         true;
449     is_between2(From, Target, To) when From < To ->
450         case (From < Target) and (Target <= To) of
451             true -> true;
452             false -> false
453         end;
454     is_between2(From, Target, To) when From > To ->
455         case ((From < Target) or (Target <= To)) of
456             true -> true;
457             false -> false
458         end.
459
460     find_successor_ask_other(Key, NodeVector, From, State) ->
461         case find_successor_in(NodeVector, State) of
462             %% what you are looking for is me, I'm returning my information to you
463             {found_me, NewSucc} ->
464                 case Key == find_successor_with_succlist of
465                     true -> gen_server:reply(From, {NewSucc, State#state.succlist});
466                     false -> gen_server:reply(From, NewSucc)
467                 end;
468
469             %% what you are looking for is my successor,
470             %% if you only need a name, I'm giving it to you now,
471             %% if you also need successor list of it, I'll forward you to it.
472             {found_other, {InqNode, InqNodeVector}} ->
473                 case Key == find_successor_with_succlist of

```

```

474         true -> gen_server:cast({global, InqNode}, {return_with_succlist, From});
475         false -> gen_server:reply(From, {InqNode, InqNodeVector})
476     end;
477
478     %% not found here, forward it to other again...
479     {not_found, {InqNode, _InqVector}} ->
480         gen_server:cast({global, InqNode}, {find_successor_ask_other, Key, NodeVector, \
->From})
481 end.
482
483 %%-----
484 %% @doc this routine is called periodically from hm_stabilizer
485 %%      to maintail predecessor
486 %%
487 %% algorithm: refer to [1]:
488 %%
489 %% // n' thinks it might be my predecessor
490 %% n.notify(n')
491 %% if(predecessor is nil or n' in (predecessor, n))
492 %%     predecessor = n'
493 %%
494 %% @end
495 %%-----
496 notify(NodeInfo, State) ->
497     case hm_misc:check_pred_and_successor(State) of
498         % predecessor == nil
499         {succ_exists, pred_is_nil} -> State#state{predecessor = NodeInfo};
500
501         % predecessor == other node
502         {succ_exists, pred_is_not_nil} ->
503             {_, PredVector} = State#state.predecessor,
504             {_, NodeVector} = NodeInfo,
505             case is_between(PredVector, NodeVector, State#state.node_vector) of
506                 true -> State#state{predecessor = NodeInfo};
507                 false -> State
508             end;
509
510         % finger table lentgh == 0, possible bug
511         {no_succ_exists, _} -> State
512     end.
513
514 %%-----
515 %% @doc this routine is called periodically from hm_stabilizer
516 %%      to maintain its successor by checking its current successor
517 %%      that if it is the immediate predecessor of it
518 %%
519 %% algorithm: refer to [1]:
520 %%
521 %% // called periodically, veries n's immediate successor,
522 %% // and tells the successor about n
523 %% n.stabilize()
524 %% x = successor.predecessor,
525 %% if( x in (n, successor))
526 %%     successor = x
527 %%     successor.notify(n)
528 %%
529 %% @end
530 %%-----
531 stabilize_in(nil, State) -> State;

```

```

533 stabilize_in({PredName, PredVector}, State) ->
534     % check if new successor exists
535     {_SuccName, SuccVector} = get_successor(State),
536     case is_between(State#state.node_vector, PredVector, SuccVector) of
537         % update successor
538         true  -> State#state{finger=[{PredName,PredVector}|
539                                     tl(State#state.finger)]};
540         % no need to update
541         false -> State
542     end.
543
544 get_successor(State) ->
545     case length(State#state.finger) > 0 of
546         true  -> hd(State#state.finger);
547         false -> nil % basically I don't see the case of successor is nil,
548                     % if it is, I let it clash.
549     end.
550
551 check_pred_in(State) when State#state.predecessor == nil ->
552     State;
553 check_pred_in(State) ->
554     % Check predecessor and set nil if failed
555     {PredName, _PredVector} = State#state.predecessor,
556     BareName = list_to_atom( atom_to_list(PredName) -- ?PROCESS_PREFIX ),
557     case hm_misc:is_alive(BareName) of
558         true  -> State;
559         false -> State#state{predecessor = nil}
560     end.

```

List 32: hm_stabilizer.erl

```

1 % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2 % use this file except in compliance with the License. You may obtain a copy of
3 % the License at
4 %
5 %   http://www.apache.org/licenses/LICENSE-2.0
6 %
7 % Unless required by applicable law or agreed to in writing, software
8 % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9 % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% File      : hm_stabilizer.erl
14 %%% Description : this module is in charge of stabilization of the
15 %%%               routing information in cooperation with hm_router
16 %%%               module
17 %%%
18 %%% control flow:
19 %%% stabilize_loop(with check_pred) ----->      sleep(stabilize_interval)
20 %%%           ^                                   |
21 %%%           |                                   |
22 %%%           |                                   v
23 %%% sleep(fix_finger_interval) <----- fix_finger_loop
24 %%%
25 %%% Chord algorithm is based on the following paper:
26 %%% [1] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R.,
27 %%%     Kaashoek, M. F., Dabek, F., and Balakrishnan, H.
28 %%%     2003.
29 %%%     Chord: a scalable peer-to-peer lookup protocol for internet applications.
30 %%%     IEEE/ACM Trans. Netw. 11, 1 (Feb. 2003), 17-32.
31 %%%     DOI= http://dx.doi.org/10.1109/TNET.2002.808407
32 %%%
33 %%% in this source, when referred to as [1], it means above paper.
34 %%%
35 %%%
36 %%% @end
37 %%% Created : 2 Oct 2010 by Yoshihiro TANAKA <hirotnkg@gmail.com>
38 %%%-----
39 -module(hm_stabilizer).
40 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
41 -behaviour(gen_fsm).
42 -vsn('0.1').
43 %% API
44 -export([
45     name/1,
46     start_link/1,
47     stop/1
48     ]).
49 %% gen_fsm callbacks
50 -export([init/1, stabilize_loop/2, fixfinger_loop/2, handle_event/3,
51     handle_sync_event/4, handle_info/3, terminate/3, code_change/4]).
52
53 -include("harmonia.hrl").
54
55 %%%=====
56 %% API
57 %%%=====
58 start_link(RegName) ->
59     gen_fsm:start_link({global, name(RegName)}, ?MODULE, RegName, []).

```

```

60
61 stop(RegName) ->
62     ?info_p("stop:stopping:[~p].~n", RegName, [RegName]),
63     gen_fsm:send_event({global, name(RegName)}, stop).
64
65 name(Name) -> list_to_atom(atom_to_list(?MODULE) ++ "_" ++ atom_to_list(Name)).
66
67 %%=====
68 %% gen_fsm callbacks
69 %%=====
70 %%-----
71 %% Function: init(Args) -> {ok, StateName, State} |
72 %%                          {ok, StateName, State, Timeout} |
73 %%                          ignore                               |
74 %%                          {stop, StopReason}
75 %% Description:Whenever a gen_fsm is started using gen_fsm:start/[3,4] or
76 %% gen_fsm:start_link/3,4, this function is called by the new process to
77 %% initialize.
78 %%-----
79 init(RegName) ->
80     {ok, stabilize_loop, RegName, ?stabilize_interval}.
81
82 %%-----
83 %% @private
84 %% @doc
85 %% Whenever a gen_fsm receives an event sent using
86 %% gen_fsm:send_all_state_event/2, this function is called to handle
87 %% the event.
88 %%
89 %% @spec handle_event(Event, StateName, State) ->
90 %%       {next_state, NextStateName, NextState} |
91 %%       {next_state, NextStateName, NextState, Timeout} |
92 %%       {stop, Reason, NewState}
93 %% @end
94 %%-----
95 handle_event(_Event, StateName, State) ->
96     {next_state, StateName, State}.
97
98 %%-----
99 %% @private
100 %% @doc
101 %% Whenever a gen_fsm receives an event sent using
102 %% gen_fsm:sync_send_all_state_event/[2,3], this function is called
103 %% to handle the event.
104 %%
105 %% @spec handle_sync_event(Event, From, StateName, State) ->
106 %%       {next_state, NextStateName, NextState} |
107 %%       {next_state, NextStateName, NextState, Timeout} |
108 %%       {reply, Reply, NextStateName, NextState} |
109 %%       {reply, Reply, NextStateName, NextState, Timeout} |
110 %%       {stop, Reason, NewState} |
111 %%       {stop, Reason, Reply, NewState}
112 %% @end
113 %%-----
114 handle_sync_event(_Event, _From, StateName, State) ->
115     Reply = ok,
116     {reply, Reply, StateName, State}.
117
118 %%-----
119 %% @private

```

```

120 %% @doc
121 %% This function is called by a gen_fsm when it receives any
122 %% message other than a synchronous or asynchronous event
123 %% (or a system message).
124 %%
125 %% @spec handle_info(Info,StateName,State)->
126 %%           {next_state, NextStateName, NextState} |
127 %%           {next_state, NextStateName, NextState, Timeout} |
128 %%           {stop, Reason, NewState}
129 %% @end
130 %%-----
131 handle_info(_Info, StateName, State) ->
132     {next_state, StateName, State}.
133
134 %%-----
135 %% @private
136 %% @doc
137 %% This function is called by a gen_fsm when it is about to
138 %% terminate. It should be the opposite of Module:init/1 and do any
139 %% necessary cleaning up. When it returns, the gen_fsm terminates with
140 %% Reason. The return value is ignored.
141 %%
142 %% @spec terminate(Reason, StateName, State) -> void()
143 %% @end
144 %%-----
145 terminate(Reason, StateName, State) ->
146     ?info_p("terminate:Reason:[~p] StateName:[~p], State:[~p]~n", none, [Reason, StateName, \
        ->State]),
147     ok.
148
149 %%-----
150 %% @private
151 %% @doc
152 %% Function:
153 %% state_name(Event, State) -> {next_state, NextStateName, NextState}|
154 %%                               {next_state, NextStateName,
155 %%                               NextState, Timeout} |
156 %%                               {stop, Reason, NewState}
157 %% Description:There should be one instance of this function for each possible
158 %% state name. Whenever a gen_fsm receives an event sent using
159 %% gen_fsm:send_event/2, the instance of this function with the same name as
160 %% the current state name StateName is called to handle the event. It is also
161 %% called if a timeout occurs.
162 %%
163 %%
164 %% this routine is called periodically from hm_stabilizer
165 %% to maintain its successor by checking its current successor
166 %% that if it is the immediate predecessor of it
167 %% then notify it's successor of it
168 %%
169 %% algorithm: refer to [1]:
170 %%
171 %% 1. x = get pred of successor
172 %% 2. check if x is my immediage successor
173 %% 3. notify about me to successor
174 %% @end
175 %%-----
176 stabilize_loop(stop, RegName) ->
177     {stop, normal, RegName};
178

```

```

179 stabilize_loop(timeout, RegName) ->
180
181 % check predecessor - if it's dead, set nil to my pred
182 hm_router:check_pred(RegName),
183
184 % node is checked to be alive inside get_successor
185 {ok, State} = hm_router:state_info(RegName),
186 Succ = hm_misc:get_successor_alive(State),
187 ?info_p("stabilize_loop:SuccName:[~p].~n", RegName, [Succ]),
188
189 {NodeName, _NodeVector} =
190     case Succ of
191         % successor is dead
192         {error, successor_dead} ->
193             case hm_misc:get_first_alive_entry(tl(State#state.succlist)) of
194                 {error, none} ->
195                     ?error_p("stabilize_loop:{error, none}.~n", RegName, []),
196                     {State#state.node_name, State#state.node_vector};
197
198                     % if successor is dead and there is some alive successor,
199                     % build a new successor list
200                     {NewSucc, NewSuccVector} ->
201                         ?info_p("stabilize_loop:NewSucc:[~p] NewSuccVector:[~p].~n", RegName\
202                             →,
203                             [NewSucc, NewSuccVector]),
204                             make_succ_list({NewSucc, NewSuccVector}, hm_router:name(RegName)),
205                             {NewSucc, NewSuccVector}
206
207                         end;
208
209         % successor is alive
210         {ok, {SuccName, SuccVector}} ->
211             % need to always maintain(don't skip)
212             make_succ_list({SuccName, SuccVector}, hm_router:name(RegName)),
213             {SuccName, SuccVector};
214
215         % currently no successor
216         {error, no_successor} ->
217             {State#state.node_name, State#state.node_vector}
218
219     end,
220
221 % get predecessor of successor, and check if new successor exists
222 % them, notify to "successor" about me
223 %
224 % algorithm: refer to [1]:
225 %
226 % // called periodically. verifies n's immediate successor
227 % // and tells the successor about n
228 % n.stabilize()
229 %     x = successor.predecessor
230 %     if( x in (n, successor) )
231 %         successor = x
232 %     successor.notify(n)
233 %
234 check_new_successor(NodeName, RegName),
235 gen_server:cast({global, NodeName}, {notify, {State#state.node_name, State#state.\
236     →node_vector}}}),
237
238 {next_state, fixfinger_loop, RegName, ?stabilize_interval}.

```



```

237
238 %%-----
239 %% @doc this API refresh finger table periodically
240 %%
241 %% algorithm: refer to [1]:
242 %%
243 %% // called periodically. refreshes finger table entries
244 %% // next stores the index of the next finger to fix
245 %% n.fix_fingers()
246 %%   next = next + 1
247 %%   if(next > m)
248 %%       next = 1
249 %%   finger[next] = find_successor(n + 2^(next-1))
250 %%
251 %% @end
252 %%-----
253 fixfinger_loop(stop, RegName) ->
254   {stop, normal, RegName};
255 fixfinger_loop(timeout, RegName) ->
256   {ok, State} = hm_router:state_info(RegName),
257   LVector = State#state.node_vector,
258   Next =
259     case (State#state.current_fix + 1) > ?max_finger of
260       true -> 1;
261       false -> State#state.current_fix + 1
262     end,
263
264   % get a fresh Next-th successor
265   NextVector = (LVector + round(math:pow(2, Next - 1))) rem ?max_key_value,
266   NewSucc = hm_router:find_successor(RegName, NextVector, Next),
267
268   % replace my finger's Next-th entry
269   hm_router:fix_finger_set(RegName, Next, NewSucc, State#state.finger),
270
271   ?info_p("fixfinger_loop: Next:[~p], State:[~p].~n", RegName, [Next, State]),
272   {next_state, stabilize_loop, RegName, ?fixfinger_interval}.
273
274 %%-----
275 %% @private
276 %% @doc
277 %% Convert process state when code is changed
278 %%
279 %% @spec code_change(OldVsn, StateName, State, Extra) ->
280 %%   {ok, StateName, NewState}
281 %% @end
282 %%-----
283 code_change(_OldVsn, StateName, State, _Extra) ->
284   {ok, StateName, State}.
285
286 %%%=====
287 %%% Internal functions
288 %%%=====
289
290 check_new_successor(SuccName, RegName) ->
291   hm_router:stabilize(RegName, hm_router:get_predecessor(SuccName)).
292
293 make_succ_list({SuccName, SuccVector}, MyNodeName) ->
294   SuccList = [{SuccName, SuccVector} | hm_router:copy_succlist(SuccName)],
295   NewSuccList =
296     case length(SuccList) > ?succ_list_len of

```

```
297         true -> lists:sublist(SuccList, 1, ?succ_list_len);
298         false -> SuccList
299     end,
300     ?info_p("stabilize_loop:Updated SuccList:[~p].~n", MyNodeName, [NewSuccList]),
301     hm_router:set_succlist(MyNodeName, NewSuccList).
```

List 33: hm_sup.erl

```

1 % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2 % use this file except in compliance with the License. You may obtain a copy of
3 % the License at
4 %
5 % http://www.apache.org/licenses/LICENSE-2.0
6 %
7 % Unless required by applicable law or agreed to in writing, software
8 % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9 % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, Yoshihiro TANAKA
15 %%% @doc
16 %%%   Harmonia supervisor
17 %%% @end
18 %%% Created : 2 Oct 2010 by Yoshihiro <hirotnkg@gmail.com>
19 %%%-----
20 -module(hm_sup).
21 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
22 -behaviour(supervisor).
23 %% API
24 -export([
25     create/2,
26     join/3,
27     start_link/1,
28     stop/1
29     ]).
30 %% Supervisor callbacks
31 -export([init/1]).
32
33 -include("harmonia.hrl").
34 -define(connect_try, 5).
35
36 %%%=====
37 %%% API functions
38 %%%=====
39
40 %%-----
41 %% @doc
42 %% Starts the supervisor
43 %%
44 %% @spec start_link() -> {ok, Pid} | ignore | {error, Error}
45 %% @end
46 %%-----
47 start_link(Env) ->
48     Name = proplists:get_value(name, Env),
49     Root = proplists:get_value(root, Env),
50     NodeName = proplists:get_value(sname, Env),
51     RootNode = proplists:get_value(root_node, Env),
52     io:format("~p~n", [Env]),
53     {ok, Pid} =
54         case proplists:get_value(node_type, Env) of
55             create ->
56                 create(Name, Env);
57             join ->
58                 ok = connect_node(RootNode),
59                 join(Name, Root, Env);

```

```

60         Any ->
61             io:fwrite("ERR:~p\n", [Any])
62     end,
63     {ok, {Name, NodeName}} = hm_name_server:register({Name, NodeName}),
64     case hm_log_h_file:add() of
65         ok -> ok;
66         Err -> io:fwrite("ERR: log file handler add:[~p]~n", [Err])
67     end,
68     io:fwrite("start Pid:[~p]~n", [Pid]),
69     {ok, Pid}.
70
71 %%-----
72 %% @private
73 %% @doc
74 %% Whenever a supervisor is started using supervisor:start_link/[2,3],
75 %% this function is called by the new process to find out about
76 %% restart strategy, maximum restart frequency and child
77 %% specifications.
78 %%
79 %% @spec init(Args) -> {ok, {SupFlags, [ChildSpec]}} |
80 %%                     ignore |
81 %%                     {error, Reason}
82 %% @end
83 %%-----
84 init({{create, Name}          = Arg, Env}) -> create_children(create, Arg, Name, Env);
85 init({{join, Name, _RootName} = Arg, Env}) -> create_children(join, Arg, Name, Env).
86
87 %%%=====
88 %%% Internal functions
89 %%%=====
90 create(Name, Env)          -> supervisor:start_link({global, Name}, ?MODULE, {{create, Name}, \
    -> Env}).
91
92 join(Name, RootName, Env) -> supervisor:start_link({global, Name}, ?MODULE, {{join, Name, \
    -> RootName}, Env}).
93
94 create_children(Type, Arg, Name, Env) ->
95     ServerListTmp =
96     [
97         child(hm_config,      hm_config,      [Env], worker),
98         child(hm_config_if,   hm_config_if,   [{Name, Env}], worker),
99         child(hm_event_mgr,    gen_event,      [{local, hm_event_mgr}], worker),
100        child(hm_router,       hm_router,      [Arg], worker),
101        child(hm_stabilizer,    hm_stabilizer,  [Name], worker),
102        child(hm_ds,           hm_ds,          [Name], worker),
103        child(hm_table,        hm_table,       [Name], worker),
104        child(hm_cache_mgr,    hm_cache_mgr,    [Name], worker)
105    ],
106     ServerList =
107         case Type == create of
108             true ->
109                 [child(?name_server, ?name_server, [], worker)] ++
110                 ServerListTmp;
111             false ->
112                 ServerListTmp
113         end,
114     {ok, { {one_for_one, 5, 2000}, ServerList } }.
115
116 child(Name, Module, Arg, Type) ->
117     {Name,
        % Name

```

```

118     {           % Start Function
119         Module,      % Module
120         start_link,   % Function
121         Arg           % Arg
122     },
123     permanent,      % restart type
124     1000,           % Shutdown time
125     Type,           % Process type
126     [Module]        % Modules
127 }.
128
129 stop(Name) -> exit(global:whereis_name(Name), kill).
130
131 connect_node(Node) ->
132     connect_node_in(Node, ?connect_try).
133
134 connect_node_in(_Node, 0) -> {error, fail_to_connect};
135 connect_node_in(Node, Cnt) ->
136     case net_kernel:connect_node(Node) of
137         true ->
138             timer:sleep(1000),
139             global:sync(),
140             ok;
141         false ->
142             timer:sleep(1000),
143             connect_node_in(Node, Cnt - 1);
144         ignored -> {error, localnode_not_alive}
145     end.
146
147 %% start_hm_name() ->
148 %%     supervisor:start_child(?NAME_SUP, child(?NAME_SUP,
149 %%                                             ?NAME_SUP,
150 %%                                             [], supervisor)).
151 %% stop_hm_name(RegName) ->
152 %%     supervisor:terminate_child(name(RegName), ?name_server).

```

List 34: hm_table.erl

```

1 % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2 % use this file except in compliance with the License. You may obtain a copy of
3 % the License at
4 %
5 % http://www.apache.org/licenses/LICENSE-2.0
6 %
7 % Unless required by applicable law or agreed to in writing, software
8 % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9 % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12 %%%-----
13 %%% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%% @copyright (C) 2010, hiro
15 %%% @doc maintain table info for range query
16 %%% @end
17 %%% Created : 2 Oct 2010 by Yoshihiro TANAKA <hirotnkg@gmail.com>
18 %%%-----
19 -module(hm_table).
20 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
21 -behaviour(gen_server).
22 -vsn('0.1').
23
24 %% API
25 -export([
26     create_table/3,
27     drop_table/2,
28     get_table_info/2,
29     get_table_info/3,
30     name/1,
31     start_link/1,
32     stop/1
33     ]).
34 %% gen_server callbacks
35 -export([init/1, handle_call/3, handle_cast/2, handle_info/2,
36     terminate/2, code_change/3]).
37
38 -include("harmonia.hrl").
39
40 %%%=====
41 %%% API
42 %%%=====
43
44 %%-----
45 %% @doc
46 %% Starts the server
47 %%
48 %% @spec start_link(Env) -> {ok, Pid} | ignore | {error, Error}
49 %% @end
50 %%%-----
51 start_link(RegName) ->
52     gen_server:start_link({global, name(RegName)}, ?MODULE, RegName, []).
53
54 stop(RegName) ->
55     gen_server:cast({global, name(RegName)}, stop).
56
57 %% spec(get_table_info(string(), string()) -> {ok, AttList::list()} |
58 %% {error, no_node_available} |
59 %% {error, no_table}

```

```

60 get_table_info(DomainName, TableName) ->
61     NodeList = hm_misc:make_request_list_from_dt(DomainName, TableName),
62     get_table_info(DomainName, TableName, NodeList).
63
64 %% if you have nodelist to ask, use this API
65 get_table_info(DomainName, TableName, NodeList) ->
66     case hm_misc:get_first_alive_entry(NodeList) of
67     {error, none} -> {error, no_node_available};
68     {NodeName, _Vector} ->
69         TargetName = name(list_to_atom( atom_to_list(NodeName) -- ?PROCESS_PREFIX )),
70         gen_server:call({global, TargetName}, {get_table_info, DomainName, TableName})
71     end.
72
73 %% spec(create_table(string(), string(), list()) -> {ok, {list(),list()}}).
74 create_table(DomainName, TableName, AttList) ->
75     NodeList = hm_misc:make_request_list_from_dt(DomainName, TableName),
76
77     % table is made on every node in successor list of correspondent node
78     {ok, FailedList} = create_table_in(NodeList, [], DomainName, TableName, AttList),
79     {ok, {NodeList, FailedList}}.
80
81 %% spec(drop_table(string(), string(), list()) -> {ok, {list(),list()}}).
82 drop_table(DomainName, TableName) ->
83     NodeList = hm_misc:make_request_list_from_dt(DomainName, TableName),
84
85     % table is made on every node in successor list of correspondent node
86     {ok, FailedList} = drop_table_in(NodeList, [], DomainName, TableName),
87     {ok, {NodeList, FailedList}}.
88
89 name(Name) when is_list(Name) -> list_to_atom(atom_to_list(?MODULE) ++ "_" ++ Name);
90 name(Name) when is_atom(Name) -> list_to_atom(atom_to_list(?MODULE) ++ "_" ++ atom_to_list(\
    →Name)).
91
92 %%%=====
93 %%% gen_server callbacks
94 %%%=====
95
96 %%-----
97 %% @private
98 %% @doc
99 %% Initializes the server
100 %%
101 %% @spec init(Args) -> {ok, State} |
102 %% {ok, State, Timeout} |
103 %% ignore |
104 %% {stop, Reason}
105 %% @end
106 %%-----
107 init(RegName) ->
108     {ok, {RegName, []}}.
109
110 %%-----
111 %% @private
112 %% @doc
113 %% This function is called by a gen_server when it is about to
114 %% terminate. It should be the opposite of Module:init/1 and do any
115 %% necessary cleaning up. When it returns, the gen_server terminates
116 %% with Reason. The return value is ignored.
117 %%
118 %% @spec terminate(Reason, State) -> void()

```

```

119 %% @end
120 %%-----
121 terminate(Reason, State) ->
122     ?info_p("terminate:Reason:[~p] State:[~p]~n", none, [Reason, State]),
123     ok.
124
125 %%-----
126 %% @private
127 %% @doc
128 %% Handling cast messages
129 %%
130 %% @spec handle_cast(Msg, State) -> {noreply, State} |
131 %%                                   {noreply, State, Timeout} |
132 %%                                   {stop, Reason, State}
133 %% @end
134 %%-----
135 handle_cast(stop, State) ->
136     {stop, normal, State}.
137
138 %%-----
139 %% @private
140 %% @doc
141 %% Handling call messages
142 %%
143 %% @spec handle_call(Request, From, State) ->
144 %%                                   {reply, Reply, State} |
145 %%                                   {reply, Reply, State, Timeout} |
146 %%                                   {noreply, State} |
147 %%                                   {noreply, State, Timeout} |
148 %%                                   {stop, Reason, Reply, State} |
149 %%                                   {stop, Reason, State}
150 %% @end
151 %%-----
152 handle_call({get_table_info, DomainName, TableName}, _From, {_RegName, TblList}=State) ->
153     DTName=list_to_atom(DomainName ++ TableName),
154     ReplyData = hm_misc:search_table_attlist(DTName, TblList),
155     {reply, ReplyData, State};
156
157 handle_call({get_table_info, DTName}, _From, {_RegName, TblList}=State) ->
158     ReplyData = hm_misc:search_table_attlist(DTName, TblList),
159     {reply, ReplyData, State};
160
161 handle_call({create_table, DomainName, TableName, AttList}, _From, {RegName, TblList}) ->
162     DTName=list_to_atom(DomainName ++ TableName),
163     case ets:info(DTName) of
164     undefined ->
165         DTName = ets:new(DTName, [bag, named_table, public]),
166         {reply, {ok, DTName}, {RegName, [{DTName, AttList}|TblList]}};
167     _ ->
168         {reply, {error, already_exists}, {RegName, TblList}}
169     end;
170
171
172 handle_call({drop_table, DomainName, TableName}, _From, {RegName, TblList}) ->
173     DTName=list_to_atom(DomainName ++ TableName),
174     case ets:info(DTName) of
175     undefined ->
176         {reply, {error, not_exists}, {RegName, TblList}};
177     _ ->
178         ets:delete(DTName),

```



```

179         NewState = lists:keydelete(DTName, 1, TblList),
180         {reply, {ok, delete_table, DTName}, {RegName, NewState}}
181     end.
182
183     %%-----
184     %% @private
185     %% @doc
186     %% Handling all non call/cast messages
187     %%
188     %% @spec handle_info(Info, State) -> {noreply, State} |
189     %%                                     {noreply, State, Timeout} |
190     %%                                     {stop, Reason, State}
191     %% @end
192     %%-----
193     handle_info(_Info, State) ->
194         {noreply, State}.
195
196     %%-----
197     %% @private
198     %% @doc
199     %% Convert process state when code is changed
200     %%
201     %% @spec code_change(OldVsn, State, Extra) -> {ok, NewState}
202     %% @end
203     %%-----
204     code_change(_OldVsn, State, _Extra) ->
205         {ok, State}.
206
207     %%%=====
208     %%% Internal functions
209     %%%=====
210     %% @doc this module makes tables, and register table info to hm_ds process
211     create_table_in([], FailedList, _DomainName, _TableName, _AttList) ->
212         {ok, FailedList};
213     create_table_in([{NodeName,_NodeVector}=CurNode|Tail],
214                     FailedList, DomainName, TableName, AttList) ->
215         BareName = list_to_atom( atom_to_list(NodeName) -- ?PROCESS_PREFIX ),
216         TargetName = name(BareName),
217         case hm_misc:is_alive(BareName) of
218             true ->
219                 {ok, DTName} = gen_server:call({global, TargetName}, {create_table, DomainName,\
220                                     → TableName, AttList}),
221                 TargetName_ds = hm_ds:name(list_to_atom( atom_to_list(NodeName) -- ?\
222                                     → PROCESS_PREFIX )),
223                 case gen_server:call({global, TargetName_ds}, {register_table, DTName}) of
224                     {ok, register_table, DTName} ->
225                         create_table_in(Tail, FailedList, DomainName, TableName, AttList);
226                     _Any ->
227                         ?warning_p("register_table fail : TargetName_ds:[~p] DTName:[~p].~n", \
228                                     → none,
229                                     [TargetName_ds, DTName]),
230                         create_table_in(Tail, [CurNode|FailedList], DomainName, TableName, \
231                                     → AttList)
232                 end;
233             false ->
234                 ?warning_p("Target Not Alive : Node:[~p].~n", none, [TargetName]),
235                 create_table_in(Tail, [CurNode|FailedList], DomainName, TableName, AttList)
236         end.
237
238     drop_table_in([], FailedList, _DomainName, _TableName) ->

```

```

235     {ok, FailedList};
236 drop_table_in([ {NodeName, _NodeVector}=CurNode|Tail],
237               FailedList, DomainName, TableName) ->
238     BareName = list_to_atom( atom_to_list(NodeName) -- ?PROCESS_PREFIX ),
239     TargetName = name(BareName),
240     case hm_misc:is_alive(BareName) of
241     true ->
242         case gen_server:call({global, TargetName}, {drop_table, DomainName, TableName}) \
243         ->of
244             {ok, delete_table, DTName} ->
245                 TargetName_ds = hm_ds:name(list_to_atom( atom_to_list(NodeName) -- ?\
246                 ->PROCESS_PREFIX )),
247                 case gen_server:call({global, TargetName_ds}, {unregister_table, DTName}\
248                 ->) of
249                     {ok, unregister_table} ->
250                         drop_table_in(Tail, FailedList, DomainName, TableName);
251                     _Any ->
252                         drop_table_in(Tail, [CurNode|FailedList], DomainName, TableName)
253                     end;
254                 _ -> drop_table_in(Tail, [CurNode|FailedList], DomainName, TableName)
255             end;
256     false ->
257         drop_table_in(Tail, [CurNode|FailedList], DomainName, TableName)
258     end.

```

F Harmonia Test Codes

List 35: Makefile

```

1 #####
2 # Generic make script for compiling erlang code #
3 # The environment variable $ERLHOME has to be set to where erlang/OTP #
4 # is installed #
5 # Compiles the code into a ebin dir. relative to the source dir. #
6 # (../ebin) #
7 #####
8 #Compiles the code into a ebin dir. relative to the source dir.
9 EBIN = ../ebin
10 EDBG = -Ddebug
11 CFLG = -W +warn_unused_vars +warn_unused_import +debug_info
12 PA = /usr/local/lib/erlang/lib/eunit-2.1.5/ebin
13 ERL = erl
14 GEN = beam
15 ERLC_EMULATOR = erl -boot start_clean
16 PATH= .:${ERLHOME}/bin:/bin:/usr/bin:/usr/local/bin:/usr/local/bin
17 SOURCE = hm_cli_test.erl \
18         hm_cache_test.erl \
19         hm_test.erl \
20         hm_qp_test.erl
21
22 TARGETS = $(SOURCE:%.erl=$(EBIN)/%.beam)
23 CODE = $(SOURCE:%.erl=$(EBIN)/%.beam)
24
25 $(EBIN)/%.beam: %.erl
26     /usr/bin/erlc -pa $(PA) $(EDBG) $(CFLG) -v -o $(EBIN) -b beam $(EFLAGS) $<
27
28 all: $(TARGETS)
29
30 debug: $(TARGETS)
31     /usr/bin/erlc -pa $(PA) $(EDBG) $(CFLG) -v -o $(EBIN) -b beam $(EFLAGS) $<
32
33 clean:
34     \rm -f $(CODE)
35
36
37 #####
38 # Template of for compiling erlang files #
39 # The environment variable $TOOLSHOME home has to be set to where #
40 # the generic make script is installed (erlang). #
41 #####
42 # code to compile
43
44 #Where include files are stored ".hrl"
45 EFLAGS = -I../include -I/usr/local/lib/erlang/lib
46
47 #####
48 # Do not edit below this line #
49 #####
50 #Include following generic make script
51 #include $(TOOLSHOME)/erlang

```

List 36: hm_cache_test.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12
13 -module(hm_cache_test).
14 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
15 -compile(export_all).
16
17 start() ->
18     hm_cache:start().
19
20 stop() ->
21     hm_cache:stop([]).
22
23 store(N) ->
24     store_in(N).
25
26 store_in(0) -> ok;
27 store_in(N) ->
28     Key = "key" ++ integer_to_list(N),
29     ok = hm_cache:store_cache(Key, {N, Key}),
30     store_in(N-1).
31
32 get(N) ->
33     get_in(N).
34
35 get_in(0) -> ok;
36 get_in(N) ->
37     Key = "key" ++ integer_to_list(N),
38     case hm_cache:get_cache(Key) of
39         {ok, _Rec} -> ok;
40         none -> io:format("not found Key:[~p]~n", [Key])
41     end,
42     get_in(N-1).

```

List 37: hm_cli_test.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12
13 -module(hm_cli_test).
14 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
15 -export([
16     cget/1,
17     cget_cond/1,
18     create_table/0,
19     create_table_test_all/0,
20     cstore/1,
21     drop_table/0,
22     get/1,
23     get/2,
24     get_cond/1,
25     get_node_name/0,
26     rangeq_test0/0,
27     rangeq_test1/0,
28     rangeq_test2/0,
29     rangeq_test3/0,
30     rangeq_test4/0,
31     rangeq_test5/0,
32     rangeq_test_all/0,
33     rget/1,
34     rget/2,
35     rget_cond/1,
36     rget_test_all/1,
37     rstore/1,
38     store/1,
39     store_short/1,
40     store_long/1,
41     test_all/0,
42     test_all/1,
43     test_comp_get/1,
44     test_perf/1,
45     test_all_short/1,
46     test_all_long/1,
47     create_table_short/0,
48     create_table_long/0,
49     rstore_short/1,
50     rstore_long/1,
51     thread_test/1,
52     thread_get_test/1,
53     thread_rget_test/1,
54     thread_gather/4,
55     thread_get_solo/5,
56     thread_rget_solo/5,
57     check_size/0,
58     check_data_num/0
59 ]).

```

```

60 -define(microsec, (1000*1000)).
61
62 -include("harmonia.hrl").
63
64 -include_lib("eunit/include/eunit.hrl").
65
66
67
68 test_all() ->
69     test_all(10),
70     rangeq_test_all().
71
72 test_all(N) ->
73     F =
74         fun({Format, Mod, Func, K, Ret}) ->
75             Ret = Mod:Func(K),
76             io:format(Format, [K]);
77         ({Format, Func, K, _Ret}) ->
78             Func(K),
79             io:format(Format, [K]);
80         ({Format, Func, Ret}) when Ret == {ok, any} ->
81             case Func() of
82                 {ok, _} -> io:format(Format, ["OK"]);
83                 Msg -> io:format(Format, [Msg])
84             end;
85         ({Format}) ->
86             io:format(Format)
87     end,
88     lists:foreach(F,
89     [
90         {"starting....\n"},
91         {"store(~p)      OK....\n", fun store/1,          N, ok},
92         {"get(~p)       OK....\n", fun hm_cli_test:get/1, N, ok},
93         {"cstore(~p)    OK....\n", fun cstore/1,         N, ok},
94         {"cget(~p)      OK....\n", fun cget/1,           N, ok},
95         {"drop_table()  ~p....\n", fun drop_table/0,     {ok, any}},
96         {"create_table() ~p....\n", fun create_table_test_all/0, {ok, any}},
97         {"rstore(~p)    OK....\n", fun rstore_test_all/1, N, ok},
98         {"rget(~p)      OK....\n", fun rget_test_all/1,  N, ok},
99         {"..end\n"}
100    ]
101    ),
102    rangeq_test_all().
103
104 test_perf(N) ->
105     F =
106         fun({Format, Mod, Func, K, Ret}) ->
107             {Time, Ret} = timer:tc(Mod, Func, [K]),
108             io:format(Format, [Time/?microsec]);
109
110         ({Format, Func, K, Ret}) ->
111             {Time, Ret} = timer:tc(?MODULE, Func, [K]),
112             io:format(Format, [Time/?microsec]);
113
114         ({Format, Func, Ret}) when Ret == {ok, any} ->
115             {Time, {ok, _}} = timer:tc(?MODULE, Func, []),
116             io:format(Format, [Time/?microsec]);
117
118         ({Format}) ->
119             io:format(Format)

```

```

120         end,
121     lists:foreach(F,
122     [
123         {"starting....\n"},
124         {"store          OK....[~20.10f] sec\n", store, N, ok},
125         {"get           OK....[~20.10f] sec\n", get, N, ok},
126         {"cstore        OK....[~20.10f] sec\n", cstore, N, ok},
127         {"cget          OK....[~20.10f] sec\n", cget, N, ok},
128         {"drop_table()  OK....[~20.10f] sec\n", drop_table, {ok, any}},
129         {"create_table  OK....[~20.10f] sec\n", create_table, {ok, any}},
130         {"rstore        OK....[~20.10f] sec\n", rstore, N, ok},
131         {"rget          OK....[~20.10f] sec\n", rget, N, ok},
132         {"..end\n"}
133     ]
134     ).
135
136 %
137 % Functions for load balancing tests
138 %
139 check_data_num() ->
140     {ok, NodeList} = hm_cli:get_node_names(),
141     disp_each_node(NodeList).
142
143 disp_each_node([]) -> ok;
144 disp_each_node([Node | NodeList]) ->
145     {Name, SName} = Node,
146     {ok, N} = hm_cli:get_data_count(Name),
147     io:format("Name:~p Node:~p count:~p\n", [Name, SName, N]),
148     disp_each_node(NodeList).
149
150
151 %
152 % Functions for space complexity tests
153 %
154 test_all_short(N) ->
155     F =
156         fun({Format, Mod, Func, K, Ret}) ->
157             Ret = Mod:Func(K),
158             io:format(Format, [K]);
159         ({Format, Func, K, _Ret}) ->
160             Func(K),
161             io:format(Format, [K]);
162         ({Format, Func, Ret}) when Ret == {ok, any} ->
163             case Func() of
164                 {ok, _} -> io:format(Format, ["OK"]);
165                 Msg -> io:format(Format, [Msg])
166             end;
167         ({Format}) ->
168             io:format(Format)
169     end,
170     lists:foreach(F,
171     [
172         {"starting....\n"},
173         {"create_table_short ~p....\n", fun create_table_short/0, {ok, any}},
174         {"rstore_short(~p)      OK....\n", fun rstore_short/1, N, ok},
175         {"..end\n"}
176     ]
177     ).
178
179 test_all_long(N) ->

```

```

180     F =
181         fun({Format, Mod, Func, K, Ret}) ->
182             Ret = Mod:Func(K),
183             io:format(Format, [K]);
184         ({Format, Func, K, _Ret}) ->
185             Func(K),
186             io:format(Format, [K]);
187         ({Format, Func, Ret}) when Ret == {ok, any} ->
188             case Func() of
189                 {ok, _} -> io:format(Format, ["OK"]);
190                 Msg -> io:format(Format, [Msg])
191             end;
192         ({Format}) ->
193             io:format(Format)
194     end,
195     lists:foreach(F,
196     [
197         {"starting...\n"},
198         {"create_table_long ~p...\n", fun create_table_long/0, {ok, any}},
199         {"rstore_long(~p) OK...\n", fun rstore_long/1, N, ok},
200         {"..end\n"}
201     ]
202     ).
203
204 create_table_short() ->
205     Domain = "Domain1",
206     Tbl = "Tbl2",
207     % int(key), char(30)
208     FldList = [{ "Fld1", true, 0 }, { "Fld2", false, "" }],
209     hm_cli:create_table(Domain, Tbl, FldList).
210
211 create_table_long() ->
212     Domain = "Domain1",
213     Tbl = "Tbl2",
214     % int(key), int(key), char(10,key), char(30)
215     FldList = [{ "Fld1", true, 0 }, { "Fld2", true, 0 }, { "Fld3", true, "" }, { "Fld4", false, "" }],
216     hm_cli:create_table(Domain, Tbl, FldList).
217
218 rstore_short(Len) ->
219     FldList = [ "Fld1", "Fld2" ],
220     rstore_in_short(Len, "Domain1", "Tbl2", FldList).
221
222 rstore_in_short(0, _Domain, _Tbl, [_,_]) -> ok;
223 rstore_in_short(Len, Domain, Tbl, [Fld1,Fld2]) ->
224     hm_cli:rstore(Domain, Tbl, [{ Fld1, Len }, { Fld2, "012345678901234567890123456789" }]),
225     rstore_in_short(Len - 1, Domain, Tbl, [Fld1,Fld2]).
226
227 store_short(0) -> ok;
228 store_short(Len) ->
229     hm_cli:store(Len, "012345678901234567890123456789"),
230     store_short(Len-1).
231
232
233
234 rstore_long(Len) ->
235     FldList = [ "Fld1", "Fld2", "Fld3", "Fld4" ],
236     rstore_in_long(Len, "Domain1", "Tbl2", FldList).
237
238 rstore_in_long(0, _Domain, _Tbl, [_,_,_,_]) -> ok;
239 rstore_in_long(Len, Domain, Tbl, [Fld1,Fld2,Fld3,Fld4]) ->

```



```

240     hm_cli:rstore(Domain, Tbl, [{Fld1, Len},
241                               {Fld2, Len+10},
242                               {Fld3, "0123456789"},
243                               {Fld4, "012345678901234567890123456789"}]),
244     rstore_in_long(Len - 1, Domain, Tbl, [Fld1,Fld2,Fld3,Fld4]).
245
246 store_long(0) -> ok;
247 store_long(Len) ->
248     hm_cli:store(Len, {Len+10, "0123456789", "012345678901234567890123456789"}),
249     store_long(Len-1).
250
251
252 check_size() ->
253     % 64 nodes
254     NodeList = [
255     netlab31@netlab3, netlab32@netlab3, netlab33@netlab3,
256     netlab34@netlab3, netlab35@netlab3, netlab36@netlab3,
257     netlab37@netlab3, netlab38@netlab3, netlab39@netlab3,
258     netlab310@netlab3, netlab311@netlab3, netlab312@netlab3,
259     netlab313@netlab3, netlab314@netlab3, netlab315@netlab3,
260     netlab316@netlab3, netlab317@netlab3, netlab318@netlab3,
261     netlab319@netlab3, netlab320@netlab3, netlab41@netlab4,
262     netlab42@netlab4, netlab43@netlab4, netlab44@netlab4,
263     netlab45@netlab4, netlab46@netlab4, netlab47@netlab4,
264     netlab48@netlab4, netlab49@netlab4, netlab410@netlab4,
265     netlab411@netlab4, netlab412@netlab4, netlab413@netlab4,
266     netlab414@netlab4, netlab415@netlab4, netlab416@netlab4,
267     netlab417@netlab4, netlab418@netlab4, netlab419@netlab4,
268     netlab420@netlab4, dell1@dell, dell2@dell,
269     dell3@dell, dell4@dell, dell5@dell, dell6@dell,
270     dell7@dell, dell8@dell, netlaba1@netlaba,
271     netlaba2@netlaba, netlaba3@netlaba, netlaba4@netlaba,
272     netlaba5@netlaba, netlaba6@netlaba, netlaba7@netlaba,
273     netlaba8@netlaba, netlabb1@netlabb, netlabb2@netlabb,
274     netlabb3@netlabb, netlabb4@netlabb, netlabb5@netlabb,
275     netlabb6@netlabb, netlabb7@netlabb, netlabb8@netlabb],
276
277     {ok, S} = file:open("data_size.dat", write),
278     check_size_in(NodeList, S).
279
280 check_size_in([], S) -> file:close(S);
281 check_size_in([Node | NodeList],S) ->
282     io:format("==== ~p =====\n", [Node]),
283     Dat = rpc:call(Node, ets, i, []),
284     io:format(S, "~p\n", [Dat]),
285     check_size_in(NodeList, S).
286
287
288 %
289 % Functions for different query conditions
290 %
291 test_comp_get(R) ->
292     F =
293     fun
294         ({Format, Func, {_,_,_,Last}=K}) ->
295             {Time, _} = timer:tc(?MODULE, Func, [K]),
296             io:format(Format, [Last, Time/?microsec]);
297         ({Format, Func, K, Last}) ->
298             {Time, _} = timer:tc(?MODULE, Func, [K]),
299             %io:format("~p~n", [Ret]),

```

```

300         io:format(Format, [Last, Time/?microsec]);
301         ({Format}) ->
302         io:format(Format)
303     end,
304     lists:foreach(F,
305     [
306         {"starting....\n"},
307         {"get   Between 1 and 1,    in ~p    OK....[~20.10f] sec\n", get_cond, {1,1,1,R\
308         →}},
309         %{"cget   Between 1 and 1,    in ~p    OK....[~20.10f] sec\n", cget_cond, {\
310         →1,1,1,R}},
311         {"rget   Between 1 and 1,    in ~p    OK....[~20.10f] sec\n", rget_cond, {1,1}, \
312         →R},
313         {"get   Between 1 and 10,   in ~p    OK....[~20.10f] sec\n", get_cond, {1,10,1,\
314         →R}},
315         %{"cget   Between 1 and 10,   in ~p    OK....[~20.10f] sec\n", cget_cond, {\
316         →1,10,1,R}},
317         {"rget   Between 1 and 10,   in ~p    OK....[~20.10f] sec\n", rget_cond, {1,10},\
318         → R},
319         {"get   Between 1 and 100,  in ~p    OK....[~20.10f] sec\n", get_cond, {\
320         →1,100,1,R}},
321         %{"cget   Between 1 and 100,  in ~p    OK....[~20.10f] sec\n", cget_cond, {\
322         →1,100,1,R}},
323         {"rget   Between 1 and 100,  in ~p    OK....[~20.10f] sec\n", rget_cond, {1,100},\
324         →, R},
325         {"get   Between 1 and 500,   in ~p    OK....[~20.10f] sec\n", get_cond, {\
326         →1,500,1,R}},
327         %{"cget   Between 1 and 500,   in ~p    OK....[~20.10f] sec\n", cget_cond, {\
328         →1,500,1,R}},
329         {"rget   Between 1 and 500,   in ~p    OK....[~20.10f] sec\n", rget_cond, {1,500},\
330         →, R},
331         {"get   Between 1 and 1000,  in ~p    OK....[~20.10f] sec\n", get_cond, {\
332         →1,1000,1,R}},
333         %{"cget   Between 1 and 500,   in ~p    OK....[~20.10f] sec\n", cget_cond, {\
334         →1,500,1,R}},
335         {"rget   Between 1 and 1000,  in ~p    OK....[~20.10f] sec\n", rget_cond, {\
336         →1,1000}, R},
337         {"..end\n"}
338     ]
339     ).
340
341 get_cond({Min, Max, From, To}) ->
342     get_cond_in(Min,Max,From,To,[ ]).
343
344 get_cond_in(_Min,_Max,From,To, L) when From > To -> L;
345 get_cond_in(Min,Max,From,To, L) ->
346     {ok,[{Key,Dat}]} = hm_cli:get(From),
347     NewList =
348     case (Key >= Min) and (Key <= Max) of
349         true -> [{Key,Dat}|L];
350         false -> L
351     end,
352     get_cond_in(Min, Max, From + 1, To, NewList).
353
354 cget_cond({Min, Max, From, To}) ->
355     cget_cond_in(Min,Max,From,To,[ ]).
356
357 cget_cond_in(_Min,_Max,From,To, L) when From > To -> L;
358 cget_cond_in(Min,Max,From,To, L) ->

```

```

345     {ok,[{Key,Dat}]} = hm_cli:cget(From),
346     NewList =
347         case (Key >= Min) and (Key <= Max) of
348             true -> [{Key,Dat}|L];
349             false -> L
350         end,
351     cget_cond_in(Min, Max, From + 1, To, NewList).
352
353 rget_cond({Min,Max}) ->
354     Domain = "Domain1",
355     Tbl = "Tbl2",
356     hm_cli:rget(Domain, Tbl, "Fld2 >= " ++ integer_to_list(Min) ++ " and Fld2 <= " ++ \
    →integer_to_list(Max)).
357
358 store(Len) -> store_in(Len).
359
360 cstore(Len) -> cstore_cache_in(Len).
361
362 get(Len) -> get_in(Len).
363
364 get(Start, End) when Start == End -> ok;
365 get(Start, End) ->
366     Ret = hm_cli:get(Start),
367     io:format("~p~n", [Ret]),
368     get(Start+1, End).
369
370
371 cget(Len) -> cget_in(Len).
372
373 create_table_test_all() ->
374     Domain = "Domain1",
375     Tbl = "Tbl2",
376     FldList = [{{"Fld1",true,true},{{"Fld2",true,true},{{"Fld3",false,nil}}],
377     hm_cli:create_table(Domain, Tbl, FldList).
378
379 create_table() ->
380     Domain = "Domain1",
381     Tbl = "Tbl2",
382     FldList = [{{"Fld1",true,true},{{"Fld2",true,true}}],
383     hm_cli:create_table(Domain, Tbl, FldList).
384
385 drop_table() ->
386     Domain = "Domain1",
387     Tbl = "Tbl2",
388     hm_cli:drop_table(Domain, Tbl).
389
390 rstore_test_all(Len) ->
391     Domain = "Domain1",
392     Tbl = "Tbl2",
393     FldList = [{{"Fld1",true,true},{{"Fld2",true,true},{{"Fld3",false,nil}}],
394     {Fld1, _, _} = lists:nth(1, FldList),
395     {Fld2, _, _} = lists:nth(2, FldList),
396     {Fld3, _, _} = lists:nth(3, FldList),
397     rstore_in_test_all(Len, Domain, Tbl, [Fld1,Fld2,Fld3]).
398
399 rstore(Len) ->
400     Domain = "Domain1",
401     Tbl = "Tbl2",
402     FldList = [{{"Fld1",true,true},{{"Fld2",true,true}}],
403     {Fld1, _, _} = lists:nth(1, FldList),

```

```

404     {Fld2, _, _} = lists:nth(2, FldList),
405     rstore_in(Len, Domain, Tbl, [Fld1,Fld2]).
406
407 rget_test_all(Len) ->
408     Domain = "Domain1",
409     Tbl     = "Tbl2",
410     rget_in_test_all(Len, Domain, Tbl).
411
412 rget(Len) ->
413     Domain = "Domain1",
414     Tbl     = "Tbl2",
415     rget_in(Len, Domain, Tbl).
416
417 rget(Start, End) ->
418     Domain = "Domain1",
419     Tbl     = "Tbl2",
420     rget_in(Start, End, Domain, Tbl).
421
422 rangeq_test_all() ->
423     %% TODO: here, you need to delete all records.
424     rangeq_test0(),
425     rangeq_test1(),
426     rangeq_test2(),
427     rangeq_test3(),
428     rangeq_test4(),
429     rangeq_test5().
430
431 rangeq_test0() ->
432     Result1 = {ok, ?succ_list_len + 1},
433     L1 = [{"Fld1", xxx}, {"Fld2", 32}, {"Fld3", textfile1}],
434     L2 = [{"Fld1", yyy}, {"Fld2", 150}, {"Fld3", textfile2}],
435     L3 = [{"Fld1", zzz}, {"Fld2", 3000}, {"Fld3", textfile3}],
436     L4 = [{"Fld1", aaa}, {"Fld2", 9000}, {"Fld3", textfile4}],
437
438     ?assertEqual(Result1, hm_cli:rstore("Domain1", "Tbl2", L1)),
439     ?assertEqual(Result1, hm_cli:rstore("Domain1", "Tbl2", L2)),
440     ?assertEqual(Result1, hm_cli:rstore("Domain1", "Tbl2", L3)),
441     ?assertEqual(Result1, hm_cli:rstore("Domain1", "Tbl2", L4)).
442
443 rangeq_test1() ->
444     io:format("rangeq_test1 start~n"),
445     D = "Domain1", T = "Tbl2",
446     Q1 = "Fld2 == 32",
447     Q2 = "Fld2 == 150",
448     Q3 = "Fld2 == 3000",
449     Q4 = "Fld2 == 9000",
450
451     ?assertEqual({ok, [[xxx,32,textfile1]]}, hm_cli:rget(D, T, Q1)),
452     io:format("[~p ~p ~p ~p]:ok~n",["case1", D,T,Q1]),
453     ?assertEqual({ok, [[yyy,150,textfile2]]}, hm_cli:rget(D, T, Q2)),
454     io:format("[~p ~p ~p ~p]:ok~n",["case2",D,T,Q2]),
455     ?assertEqual({ok, [[zzz,3000,textfile3]]}, hm_cli:rget(D, T, Q3)),
456     io:format("[~p ~p ~p ~p]:ok~n",["case3",D,T,Q3]),
457     ?assertEqual({ok, [[aaa,9000,textfile4]]}, hm_cli:rget(D, T, Q4)),
458     io:format("[~p ~p ~p ~p]:ok~n",["case4",D,T,Q4]).
459
460 rangeq_test2() ->
461     io:format("rangeq_test2 start~n"),
462     {ok, RowList1} = hm_cli:rget(D = "Domain1", T = "Tbl2", Q1 = "Fld2 != 32"),
463     lists:foreach(fun([_,N,_]) -> ?assert(N /= 32) end, RowList1),

```

```

464     io:format("[~p ~p ~p ~p]:ok~n",["case1",D,T,Q1]),
465
466     {ok, RowList2} = hm_cli:rget(D = "Domain1", T = "Tbl2", Q2 = "Fld2 != 150"),
467     lists:foreach(fun([_,N,_]) -> ?assert(N /= 150) end, RowList2),
468     io:format("[~p ~p ~p ~p]:ok~n",["case2",D,T,Q2]),
469
470     {ok, RowList3} = hm_cli:rget(D = "Domain1", T = "Tbl2", Q3 = "Fld2 != 3000"),
471     lists:foreach(fun([_,N,_]) -> ?assert(N /= 3000) end, RowList3),
472     io:format("[~p ~p ~p ~p]:ok~n",["case3",D,T,Q3]),
473
474     {ok, RowList4} = hm_cli:rget(D = "Domain1", T = "Tbl2", Q4 = "Fld2 != 9000"),
475     lists:foreach(fun([_,N,_]) -> ?assert(N /= 9000) end, RowList4),
476     io:format("[~p ~p ~p ~p]:ok~n",["case4",D,T,Q4]).
477
478 rangeq_test3() ->
479     io:format("rangeq_test3 start~n"),
480     D = "Domain1", T = "Tbl2",
481     Q1 = "Fld2 > 32",
482     Q2 = "Fld2 > 150",
483     Q3 = "Fld2 > 3000",
484     Q4 = "Fld2 > 9000",
485     Q5 = "Fld2 < 32",
486     Q6 = "Fld2 < 150",
487     Q7 = "Fld2 < 3000",
488     Q8 = "Fld2 < 9000",
489
490     {ok, RowList1} = hm_cli:rget(D, T, Q1),
491     lists:foreach(fun([_,N,_]) -> ?assert(N > 32) end, RowList1),
492     io:format("[~p ~p ~p ~p]:ok~n",["case1",D,T,Q1]),
493     {ok, RowList2} = hm_cli:rget(D, T, Q2),
494     lists:foreach(fun([_,N,_]) -> ?assert(N > 150) end, RowList2),
495     io:format("[~p ~p ~p ~p]:ok~n",["case2",D,T,Q2]),
496     {ok, RowList3} = hm_cli:rget(D, T, Q3),
497     lists:foreach(fun([_,N,_]) -> ?assert(N > 3000) end, RowList3),
498     io:format("[~p ~p ~p ~p]:ok~n",["case3",D,T,Q3]),
499     {ok, RowList4} = hm_cli:rget(D, T, Q4),
500     lists:foreach(fun([_,N,_]) -> ?assert(N > 9000) end, RowList4),
501     io:format("[~p ~p ~p ~p]:ok~n",["case4",D,T,Q4]),
502
503     {ok, RowList5} = hm_cli:rget(D, T, Q5),
504     lists:foreach(fun([_,N,_]) -> ?assert(N < 32) end, RowList5),
505     io:format("[~p ~p ~p ~p]:ok~n",["case5",D,T,Q5]),
506     {ok, RowList6} = hm_cli:rget(D, T, Q6),
507     lists:foreach(fun([_,N,_]) -> ?assert(N < 150) end, RowList6),
508     io:format("[~p ~p ~p ~p]:ok~n",["case6",D,T,Q6]),
509     {ok, RowList7} = hm_cli:rget(D, T, Q7),
510     lists:foreach(fun([_,N,_]) -> ?assert(N < 3000) end, RowList7),
511     io:format("[~p ~p ~p ~p]:ok~n",["case7",D,T,Q7]),
512     {ok, RowList8} = hm_cli:rget(D, T, Q8),
513     lists:foreach(fun([_,N,_]) -> ?assert(N < 9000) end, RowList8),
514     io:format("[~p ~p ~p ~p]:ok~n",["case8",D,T,Q8]).
515
516 rangeq_test4() ->
517     io:format("rangeq_test4 start~n"),
518     D = "Domain1", T = "Tbl2",
519     Q1 = "Fld2 >= 32",
520     Q2 = "Fld2 >= 150",
521     Q3 = "Fld2 >= 3000",
522     Q4 = "Fld2 >= 9000",
523     Q5 = "Fld2 <= 32",

```

```

524 Q6 = "Fld2 <= 150",
525 Q7 = "Fld2 <= 3000",
526 Q8 = "Fld2 <= 9000",
527
528 {ok, RowList1} = hm_cli:rget(D, T, Q1),
529 lists:foreach(fun([_,N,_]) -> ?assert(N >= 32) end, RowList1),
530 io:format("[~p ~p ~p ~p]:ok~n",["case1",D,T,Q1]),
531 {ok, RowList2} = hm_cli:rget(D, T, Q2),
532 lists:foreach(fun([_,N,_]) -> ?assert(N >= 150) end, RowList2),
533 io:format("[~p ~p ~p ~p]:ok~n",["case2",D,T,Q2]),
534 {ok, RowList3} = hm_cli:rget(D, T, Q3),
535 lists:foreach(fun([_,N,_]) -> ?assert(N >= 3000) end, RowList3),
536 io:format("[~p ~p ~p ~p]:ok~n",["case2",D,T,Q3]),
537 {ok, RowList4} = hm_cli:rget(D, T, Q4),
538 lists:foreach(fun([_,N,_]) -> ?assert(N >= 9000) end, RowList4),
539 io:format("[~p ~p ~p ~p]:ok~n",["case3",D,T,Q4]),
540
541 {ok, RowList5} = hm_cli:rget(D, T, Q5),
542 lists:foreach(fun([_,N,_]) -> ?assert(N <= 32) end, RowList5),
543 io:format("[~p ~p ~p ~p]:ok~n",["case4",D,T,Q5]),
544 {ok, RowList6} = hm_cli:rget(D, T, Q6),
545 lists:foreach(fun([_,N,_]) -> ?assert(N <= 150) end, RowList6),
546 io:format("[~p ~p ~p ~p]:ok~n",["case5",D,T,Q6]),
547 {ok, RowList7} = hm_cli:rget(D, T, Q7),
548 lists:foreach(fun([_,N,_]) -> ?assert(N <= 3000) end, RowList7),
549 io:format("[~p ~p ~p ~p]:ok~n",["case6",D,T,Q7]),
550 {ok, RowList8} = hm_cli:rget(D, T, Q8),
551 lists:foreach(fun([_,N,_]) -> ?assert(N <= 9000) end, RowList8),
552 io:format("[~p ~p ~p ~p]:ok~n",["case7",D,T,Q8]),
553
554
555
556 rangeq_test5() ->
557 io:format("rangeq_test5 start~n"),
558 D = "Domain1", T = "Tbl2",
559 Q1 = "Fld2 >= 32 and Fld2 <= 150",
560 Q2 = "Fld2 >= 3000 or Fld2 <= 150",
561 Q3 = "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1",
562 Q4 = "Fld2 >= 3000 and Fld2 <= 150",
563 Q5 = "Fld1 == yyy and Fld2 == 150",
564 Q6 = "Fld1 == yyy or Fld2 == 32",
565 Q7 = "Fld1 == yyy and Fld2 == 150",
566
567 {ok, RowList1} = hm_cli:rget(D, T, Q1),
568 lists:foreach(fun([_,N,_]) -> ?assert((N >= 32) and (N <= 150)) end, RowList1),
569 io:format("[~p ~p ~p ~p]:ok~n",["case1",D,T,Q1]),
570
571 {ok, RowList2} = hm_cli:rget(D, T, Q2),
572 lists:foreach(fun([_,N,_]) -> ?assert((N >= 3000) or (N <= 150)) end, RowList2),
573 io:format("[~p ~p ~p ~p]:ok~n",["case2",D,T,Q2]),
574
575 {ok, RowList3} = hm_cli:rget(D, T, Q3),
576 lists:foreach(fun([_,N,O]) -> ?assert(((N >= 3000) or (N <= 150)) and (O == textfile1))\
    -> end, RowList3),
577 io:format("[~p ~p ~p ~p]:ok~n",["case3",D,T,Q3]),
578
579 {ok, RowList4} = hm_cli:rget(D, T, Q4),
580 lists:foreach(fun([_,N,_]) -> ?assert(((N >= 3000) and (N <= 150))) end, RowList4),
581 io:format("[~p ~p ~p ~p]:ok~n",["case4",D,T,Q4]),
582

```

```

583     {ok, RowList5} = hm_cli:rget(D, T, Q5),
584     lists:foreach(fun([M,N,_]) -> ?assert(((M == yyy) and (N == 150))) end, RowList5),
585     io:format("~p ~p ~p ~p:ok~n",["case5",D,T,Q5]),
586
587     {ok, RowList6} = hm_cli:rget(D, T, Q6),
588     lists:foreach(fun([M,N,_]) -> ?assert(((M == yyy) or (N == 32))) end, RowList6),
589     io:format("~p ~p ~p ~p:ok~n",["case6",D,T,Q6]),
590
591     {ok, RowList7} = hm_cli:rget(D, T, Q7),
592     lists:foreach(fun([M,N,O]) -> ?assert((M == yyy) and (N == 150)) end, RowList7),
593     io:format("~p ~p ~p ~p:ok~n",["case7",D,T,Q7]).
594
595
596 %
597 % Functions for thread tests
598 %
599 thread_test(List) ->
600     {Time1, _} = timer:tc(?MODULE, thread_rget_test, [List]),
601     io:format("~20.10f] sec~n", [Time1/?microsec]),
602     {Time2, _} = timer:tc(?MODULE, thread_get_test, [List]),
603     io:format("~20.10f] sec~n", [Time2/?microsec]).
604
605 thread_rget_test(List) ->
606     Ref = make_ref(),
607     global:register_name(thread_gather, spawn(?MODULE, thread_gather, [Ref, length(List), \
        ->self(), 0])),
608     spawn_all_rget_threads(List, thread_gather, Ref),
609     receive
610         {ok, thread_done, _Time} -> ok
611     end.
612
613 thread_get_test(List) ->
614     Ref = make_ref(),
615     global:register_name(thread_gather, spawn(?MODULE, thread_gather, [Ref, length(List), \
        ->self(), 0])),
616     spawn_all_get_threads(List, thread_gather, Ref),
617     receive
618         {ok, thread_done, _Time} -> ok
619     end.
620
621 thread_gather(_Ref, 0, Pid, Timeacc) -> Pid ! {ok, thread_done, Timeacc};
622 thread_gather(Ref, N, Pid, Timeacc) ->
623     receive
624         {ok, Ref, _Node, Time} ->
625             thread_gather(Ref, N-1, Pid, Timeacc + Time)
626     end.
627
628 spawn_all_rget_threads([], _Name, _Ref) -> ok;
629 spawn_all_rget_threads([_Node, Start, End] | List, Name, Ref) ->
630     spawn(Node, ?MODULE, thread_rget_solo, [Start, End, Name, Ref, Node]),
631     spawn_all_rget_threads(List, Name, Ref).
632
633
634 thread_rget_solo(Start, End, Name, Ref, Node) ->
635     {Time, _} = timer:tc(?MODULE, rget, [Start, End]),
636     io:format("Node:[~p] rget Time:[~20.10f] sec~n", [Node, Time/?microsec]),
637     global:send(Name, {ok, Ref, Node, Time}),
638     ok.
639
640

```

```

641 spawn_all_get_threads([], _Name, _Ref) -> ok;
642 spawn_all_get_threads([{{Node, Start, End}} | List], Name, Ref) ->
643     spawn(Node, ?MODULE, thread_get_solo, [Start, End, Name, Ref, Node]),
644     spawn_all_get_threads(List, Name, Ref).
645
646
647 thread_get_solo(Start, End, Name, Ref, Node) ->
648     {Time, _} = timer:tc(?MODULE, get, [Start, End]),
649     io:format("Node:[~p] get Time:[~20.10f] sec\n", [Node, Time/?microsec]),
650     global:send(Name, {ok, Ref, Node, Time}),
651     ok.
652
653
654
655
656
657
658
659 %% -----
660 %% Internal Functions
661 %% -----
662 store_in(0) -> ok;
663 store_in(Len) when is_integer(Len) ->
664     Val = Len + 100,
665     hm_cli:store(Len, Val),
666     store_in(Len-1).
667
668 cstore_cache_in(0) -> ok;
669 cstore_cache_in(Len) when is_integer(Len) ->
670     Val = Len + 100,
671     hm_cli:cstore(Len, Val),
672     cstore_cache_in(Len-1).
673
674 get_in(0) -> ok;
675 get_in(Len) ->
676     Val = Len + 100,
677     {ok, [{Len, Val}]} = hm_cli:get(Len),
678     get_in(Len-1).
679
680 cget_in(0) -> ok;
681 cget_in(Len) ->
682     hm_cli:cget(Len),
683     cget_in(Len-1).
684
685 rstore_in_test_all(0, _Domain, _Tbl, [_,_]) -> ok;
686 rstore_in_test_all(Len, Domain, Tbl, [Fld1,Fld2,Fld3]) ->
687     hm_cli:rstore(Domain, Tbl, [{Fld1, xxx},{Fld2, Len},{Fld3, textfile1}]),
688     rstore_in_test_all(Len - 1, Domain, Tbl, [Fld1,Fld2,Fld3]).
689
690 rstore_in(0, _Domain, _Tbl, [_,_]) -> ok;
691 rstore_in(Len, Domain, Tbl, [Fld1,Fld2]) ->
692     hm_cli:rstore(Domain, Tbl, [{Fld1, xxx},{Fld2, Len}]),
693     rstore_in(Len - 1, Domain, Tbl, [Fld1,Fld2]).
694
695 rget_in_test_all(0, _Domain, _Tbl) -> ok;
696 rget_in_test_all(Len, Domain, Tbl) ->
697     {ok, [[xxx,Len,_]]} = hm_cli:rget(Domain, Tbl, "Fld2 == " ++ integer_to_list(Len)),
698     rget_in_test_all(Len - 1, Domain, Tbl).
699
700 rget_in(0, _Domain, _Tbl) -> ok;

```



```

701 rget_in(Len, Domain, Tbl) ->
702     {ok, [[xxx,Len]]} = hm_cli:rget(Domain, Tbl, "Fld2 == " ++ integer_to_list(Len)),
703     rget_in(Len - 1, Domain, Tbl).
704
705 rget_in(Start, End, _Domain, _Tbl) when Start > End -> ok;
706 rget_in(Start, End, Domain, Tbl) ->
707     {ok, [[xxx,Start]]} = hm_cli:rget(Domain, Tbl, "Fld2 == " ++ integer_to_list(Start)),
708     rget_in(Start + 1, End, Domain, Tbl).
709
710 get_node_name() ->
711     {ok, NameList} = gen_server:call({global, hm_name_server}, get_name_list),
712     {_Name, NodeName} = lists:nth(random:uniform(length(NameList)), NameList),
713     {ok, NodeName}.

```

List 38: hm_qp_test.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12
13 %% @author Yoshihiro TANAKA <hirotnkg@gmail.com>
14 %%
15 %% @doc query scanner/parser test suite
16 %%
17 %% to run:
18 %% erlc hm_qp_test.erl
19 %% erl -pa ../ebin -s hm_qp_test test -s init stop
20 %%
21 %% @end
22
23 -module(hm_qp_test).
24 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
25 -compile([export_all]).
26
27 -include_lib("eunit/include/eunit.hrl").
28
29 scan_test_() -> [
30     % simple scan test
31     ?_assertEqual([{{identifier,'$2'},
32                     {relational_operator,"=="},
33                     {identifier,xxx}} , hm_qp:scan("Fld1 == xxx", [{ "Fld1",true,true} ])),
34
35     % atom test
36     ?_assertEqual([{{identifier,'$2'},
37                     {relational_operator,"=="},
38                     {atom,'test test'}}], hm_qp:scan("Fld1 == 'test test'", [{ "Fld1",true,true} ])\
39                     ->)),
40
41     % no space between == and atom 'test test'
42     ?_assertEqual([{{identifier,'$2'},
43                     {relational_operator,"=="},
44                     {atom,'test test'}}], hm_qp:scan("Fld1 =='test test'", [{ "Fld1",true,true} ])\
45                     ->)]

```

List 39: hm_test.erl

```

1  % Licensed under the Apache License, Version 2.0 (the "License"); you may not
2  % use this file except in compliance with the License. You may obtain a copy of
3  % the License at
4  %
5  %   http://www.apache.org/licenses/LICENSE-2.0
6  %
7  % Unless required by applicable law or agreed to in writing, software
8  % distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
9  % WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
10 % License for the specific language governing permissions and limitations under
11 % the License.
12
13 -module(hm_test).
14 -author('Yoshihiro TANAKA <hirotnkg@gmail.com>').
15 -export([
16     dbgadd/1,
17     dbgall/0,
18     dbgdel/1
19 ]).
20
21 dbgall() ->
22     dbg:tracer(),
23     dbg:p(all, call),
24     dbg:tpl(test,[]),
25     dbg:tpl(test,x),
26     dbg:tpl(tp,[]),
27     dbg:tpl(tp,x),
28     dbg:tpl(hm_cache,[]),
29     dbg:tpl(hm_cache,x),
30     dbg:tpl(hm_router,[]),
31     dbg:tpl(hm_router,x),
32     dbg:tpl(harmonia,[]),
33     dbg:tpl(harmonia,x),
34     dbg:tpl(hm_misc,[]),
35     dbg:tpl(hm_misc,x),
36     dbg:tpl(hm_ds,[]),
37     dbg:tpl(hm_ds,x),
38     dbg:tpl(hm_sup,[]),
39     dbg:tpl(hm_sup,x),
40     dbg:tpl(hm_name_server,[]),
41     dbg:tpl(hm_name_server,x),
42     dbg:tpl(hm_log_h_file,[]),
43     dbg:tpl(hm_log_h_file,x),
44     dbg:tpl(hm_event_mgr,[]),
45     dbg:tpl(hm_event_mgr,x),
46     dbg:tpl(hm_event,[]),
47     dbg:tpl(hm_event,x),
48     application:start(sasl).
49
50 dbgadd(Mod) ->
51     dbg:tracer(),
52     dbg:p(Mod, call),
53     dbg:tpl(Mod,[]),
54     dbg:tpl(Mod,x).
55
56 dbgdel(Mod) ->
57     dbg:ctpl(Mod).

```

G Test Results

G.1 Comparison with Different Number of Nodes

List 40: Test Results: Comparison with different number of nodes: 10 nodes

```

1 hirotnk@netlab3 ~/harmonia/Harmonia $ erl -pa ebin -config harmonia -run hm_edge start -\
  →harmonia node_type 'join' name xxx sname 'xxx_node@netlab3' root_node 'netlab31@netlab3' \
  →setcookie harmonia_cookie -sname 'xxx_node@netlab3'
2 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
3
4 Eshell V5.6.3 (abort with ^G)
5 (xxx_node@netlab3)1> [{name,xxx},
6 {root_node,netlab31@netlab3},
7 {root,netlab31},
8 {included_applications,[],},
9 {sname,xxx_node@netlab3},
10 {logfile_ext, ".txt"},
11 {node_type, join},
12 {logdir, "log/"},
13 {logfile, "harmonia_log"}]
14 "log/harmonia_log_xxx.txt"
15 start Pid:[<0.72.0>]
16
17 (xxx_node@netlab3)1> hm_cli_test:test_all().
18 starting....
19 store(10)      OK....
20 get(10)        OK....
21 cstore(10)     OK....
22 cget(10)       OK....
23 drop_table()  "OK"....
24 create_table() "OK"....
25 rstore(10)     OK....
26 rget(10)       OK....
27 ..end
28 rangeq_test1 start
29 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
30 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
31 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
32 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
33 rangeq_test2 start
34 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok
35 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
36 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
37 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
38 rangeq_test3 start
39 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok
40 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok
41 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
42 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok
43 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
44 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
45 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok
46 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
47 rangeq_test4 start
48 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
49 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
50 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok

```

```

51 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
52 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok
53 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
54 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
55 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
56 rangeq_test5 start
57 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
58 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
59 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
60 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
61 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
62 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
63 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
64 rangeq_test1 start
65 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
66 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
67 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
68 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
69 rangeq_test2 start
70 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok
71 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
72 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
73 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
74 rangeq_test3 start
75 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok
76 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok
77 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
78 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok
79 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
80 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
81 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok
82 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
83 rangeq_test4 start
84 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
85 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
86 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok
87 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
88 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok
89 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
90 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
91 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
92 rangeq_test5 start
93 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
94 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
95 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
96 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
97 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
98 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
99 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
100 ok
101 (xxx_node@netlab3)2> hm_cli:log_stop().
102 log stop:[dell1@dell] Result:[ok]
103 log stop:[netlabbl@netlabbb] Result:[ok]
104 log stop:[netlabal@netlaba] Result:[ok]
105 log stop:[netlab43@netlab4] Result:[ok]
106 log stop:[netlab42@netlab4] Result:[ok]
107 log stop:[netlab41@netlab4] Result:[ok]
108 hm_log_h_file:closing 'log/harmonia_log_xxx.txt'
109 log stop:[xxx_node@netlab3] Result:[ok]
110 log stop:[netlab33@netlab3] Result:[ok]

```

```

111 log stop:[netlab32@netlab3] Result:[ok]
112 log stop:[netlab31@netlab3] Result:[ok]
113 ok
114 (xxx_node@netlab3)3> hm_cli_test:test_perf(10000).
115 starting....
116 store      OK....[      55.8763870000] sec
117 get        OK....[      17.6551860000] sec
118 cstore     OK....[      37.5985010000] sec
119 cget       OK....[       0.0320320000] sec
120 drop_table() OK....[       0.0072790000] sec
121 create_table OK....[       0.0068260000] sec
122 rstore     OK....[      98.1345750000] sec
123 rget       OK....[     125.7586580000] sec
124 ..end
125 ok
126 (xxx_node@netlab3)4> hm_cli:get_data_count(dell1).
127 {ok,23500}

```

List 41: Test Results:Comparison with different number of nodes:17 nodes

```

1 ame 'xxx_node@netlab3' root_node 'netlab31@netlab3' -setcookie harmonia_cookie -sname '\
  →xxx_node@netlab3'
2 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
3
4 Eshell V5.6.3 (abort with ^G)
5 (xxx_node@netlab3)1> [{name,xxx},
6 {root_node,netlab31@netlab3},
7 {root,netlab31},
8 {included_applications,[],},
9 {sname,xxx_node@netlab3},
10 {logfile_ext,".txt"},
11 {node_type,join},
12 {logdir,"log/"},
13 {logfile,"harmonia_log"}]
14 "log/harmonia_log_xxx.txt"
15 start Pid:[<0.94.0>]
16
17 (xxx_node@netlab3)1> hm_cli_test:test_all().
18 starting....
19 store(10)      OK....
20 get(10)        OK....
21 cstore(10)     OK....
22 cget(10)       OK....
23 drop_table()   "OK"....
24 create_table() "OK"....
25 rstore(10)     OK....
26 rget(10)       OK....
27 ..end
28 rangeq_test1 start
29 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
30 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
31 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
32 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
33 rangeq_test2 start
34 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok
35 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
36 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
37 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
38 rangeq_test3 start
39 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok

```

```

40 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok
41 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
42 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok
43 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
44 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
45 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok
46 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
47 rangeq_test4 start
48 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
49 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
50 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok
51 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
52 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok
53 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
54 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
55 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
56 rangeq_test5 start
57 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
58 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
59 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
60 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
61 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
62 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
63 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
64 rangeq_test1 start
65 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
66 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
67 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
68 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
69 rangeq_test2 start
70 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok
71 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
72 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
73 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
74 rangeq_test3 start
75 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok
76 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok
77 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
78 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok
79 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
80 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
81 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok
82 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
83 rangeq_test4 start
84 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
85 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
86 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok
87 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
88 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok
89 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
90 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
91 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
92 rangeq_test5 start
93 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
94 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
95 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
96 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
97 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
98 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
99 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok

```

```

100 ok
101 (xxx_node@netlab3)2> hm_cli:log_stop().
102 log stop:[dell2@dell] Result:[ok]
103 log stop:[dell1@dell] Result:[ok]
104 log stop:[netlab2@netlab] Result:[ok]
105 log stop:[netlab1@netlab] Result:[ok]
106 log stop:[netlab2@netlab] Result:[ok]
107 log stop:[netlab1@netlab] Result:[ok]
108 log stop:[netlab45@netlab4] Result:[ok]
109 log stop:[netlab41@netlab4] Result:[ok]
110 log stop:[netlab42@netlab4] Result:[ok]
111 log stop:[netlab44@netlab4] Result:[ok]
112 log stop:[netlab43@netlab4] Result:[ok]
113 hm_log_h_file:closing 'log/harmonia_log_xxx.txt'
114 log stop:[xxx_node@netlab3] Result:[ok]
115 log stop:[netlab35@netlab3] Result:[ok]
116 log stop:[netlab34@netlab3] Result:[ok]
117 log stop:[netlab33@netlab3] Result:[ok]
118 log stop:[netlab32@netlab3] Result:[ok]
119 log stop:[netlab31@netlab3] Result:[ok]
120 ok
121 (xxx_node@netlab3)3> hm_cli_test:test_perf(10000).
122 starting....
123 store          OK....[      44.4140180000] sec
124 get            OK....[      19.7205070000] sec
125 cstore        OK....[      44.4498950000] sec
126 cget          OK....[       0.0317460000] sec
127 drop_table()  OK....[       0.0073830000] sec
128 create_table  OK....[       0.0068740000] sec
129 rstore        OK....[     106.7740510000] sec
130 rget          OK....[     120.1923330000] sec
131 ..end
132 ok
133 (xxx_node@netlab3)4>

```

List 42: Test Results:Comparison with different number of nodes:33 nodes

```

1 hirotnk@netlab3 ~/harmonia/Harmonia $ erl -pa ebin -config harmonia -run hm_edge start -\
  →harmonia node_type 'join' name xxx sname 'xxx_node@netlab3' root_node 'netlab31@netlab3' \
  →setcookie harmonia_cookie -sname 'xxx_node@netlab3'
2 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
3
4 Eshell V5.6.3 (abort with ^G)
5 (xxx_node@netlab3)1> [{name,xxx},
6 {root_node,netlab31@netlab3},
7 {root,netlab31},
8 {included_applications,[]},
9 {sname,xxx_node@netlab3},
10 {logfile_ext,".txt"},
11 {node_type,join},
12 {logdir,"log/"},
13 {logfile,"harmonia_log"}]
14 "log/harmonia_log_xxx.txt"
15 start Pid:[<0.149.0>]
16
17 (xxx_node@netlab3)1> hm_cli_test:test_all().
18 starting....
19 store(10)      OK....
20 get(10)        OK....
21 cstore(10)     OK....

```



```

22 cget(10)          OK...
23 drop_table()      "OK"....
24 create_table()    "OK"....
25 rstore(10)        OK...
26 rget(10)          OK...
27 ..end
28 rangeq_test1 start
29 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
30 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
31 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
32 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
33 rangeq_test2 start
34 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok
35 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
36 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
37 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
38 rangeq_test3 start
39 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok
40 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok
41 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
42 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok
43 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
44 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
45 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok
46 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
47 rangeq_test4 start
48 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
49 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
50 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok
51 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
52 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok
53 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
54 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
55 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
56 rangeq_test5 start
57 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
58 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
59 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
60 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
61 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
62 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
63 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
64 rangeq_test1 start
65 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
66 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
67 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
68 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
69 rangeq_test2 start
70 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok
71 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
72 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
73 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
74 rangeq_test3 start
75 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok
76 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok
77 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
78 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok
79 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
80 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
81 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok

```

```

82 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
83 rangeq_test4 start
84 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
85 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
86 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok
87 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
88 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok
89 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
90 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
91 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
92 rangeq_test5 start
93 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
94 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
95 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
96 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
97 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
98 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
99 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
100 ok
101 (xxx_node@netlab3)2> hm_cli:log_stop().
102 log stop:[dell3@dell] Result:[ok]
103 log stop:[dell4@dell] Result:[ok]
104 log stop:[dell1@dell] Result:[ok]
105 log stop:[dell2@dell] Result:[ok]
106 log stop:[netlabb1@netlabb] Result:[ok]
107 log stop:[netlabb4@netlabb] Result:[ok]
108 log stop:[netlabb3@netlabb] Result:[ok]
109 log stop:[netlabb2@netlabb] Result:[ok]
110 log stop:[netlaba4@netlaba] Result:[ok]
111 log stop:[netlaba1@netlaba] Result:[ok]
112 log stop:[netlaba3@netlaba] Result:[ok]
113 log stop:[netlaba2@netlaba] Result:[ok]
114 log stop:[netlab410@netlab4] Result:[ok]
115 log stop:[netlab49@netlab4] Result:[ok]
116 log stop:[netlab45@netlab4] Result:[ok]
117 log stop:[netlab48@netlab4] Result:[ok]
118 log stop:[netlab47@netlab4] Result:[ok]
119 log stop:[netlab46@netlab4] Result:[ok]
120 log stop:[netlab44@netlab4] Result:[ok]
121 log stop:[netlab43@netlab4] Result:[ok]
122 log stop:[netlab42@netlab4] Result:[ok]
123 log stop:[netlab41@netlab4] Result:[ok]
124 hm_log_h_file:closing 'log/harmonia_log_xxx.txt'
125 log stop:[xxx_node@netlab3] Result:[ok]
126 log stop:[netlab35@netlab3] Result:[ok]
127 log stop:[netlab36@netlab3] Result:[ok]
128 log stop:[netlab310@netlab3] Result:[ok]
129 log stop:[netlab37@netlab3] Result:[ok]
130 log stop:[netlab39@netlab3] Result:[ok]
131 log stop:[netlab38@netlab3] Result:[ok]
132 log stop:[netlab34@netlab3] Result:[ok]
133 log stop:[netlab33@netlab3] Result:[ok]
134 log stop:[netlab32@netlab3] Result:[ok]
135 log stop:[netlab31@netlab3] Result:[ok]
136 ok
137 (xxx_node@netlab3)3> hm_cli_test:test_perf(10000).
138 starting....
139 store      OK....[      49.6963220000] sec
140 get        OK....[      21.9204730000] sec
141 cstore     OK....[      49.8137730000] sec

```

```

142 cget          OK....[      0.0323190000] sec
143 drop_table()  OK....[      0.0090010000] sec
144 create_table  OK....[      0.0082430000] sec
145 rstore        OK....[    125.8592480000] sec
146 rget          OK....[    98.7583810000] sec
147 ..end
148 ok
149 (xxx_node@netlab3)4>

```

List 43: Test Results: Comparison with different number of nodes: 65 nodes

```

1 Eshell V5.6.3 (abort with ^G)
2 (xxx_node@netlab3)1> [{name,xxx},
3 {root_node,netlab31@netlab3},
4 {root,netlab31},
5 {included_applications,[],},
6 {sname,xxx_node@netlab3},
7 {logfile_ext, ".txt"},
8 {node_type, join},
9 {logdir, "log/"},
10 {logfile, "harmonia_log"}]
11 "log/harmonia_log_xxx.txt"
12 start Pid:[<0.259.0>]
13
14 (xxx_node@netlab3)1> hm_cli_test:test_all().
15 starting....
16 store(10)      OK....
17 get(10)        OK....
18 cstore(10)     OK....
19 cget(10)       OK....
20 drop_table()   "OK"....
21 create_table() "OK"....
22 rstore(10)     OK....
23 rget(10)       OK....
24 ..end
25 rangeq_test1 start
26 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
27 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
28 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
29 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
30 rangeq_test2 start
31 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok
32 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
33 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
34 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
35 rangeq_test3 start
36 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok
37 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok
38 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
39 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok
40 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
41 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
42 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok
43 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
44 rangeq_test4 start
45 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
46 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
47 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok
48 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
49 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok

```

```

50 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
51 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
52 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
53 rangeq_test5 start
54 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
55 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
56 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
57 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
58 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
59 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
60 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
61 rangeq_test1 start
62 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
63 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
64 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
65 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
66 rangeq_test2 start
67 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok
68 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
69 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
70 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
71 rangeq_test3 start
72 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok
73 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok
74 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
75 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok
76 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
77 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
78 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok
79 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
80 rangeq_test4 start
81 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
82 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
83 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok
84 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
85 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok
86 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
87 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
88 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
89 rangeq_test5 start
90 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
91 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
92 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
93 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
94 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
95 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
96 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
97 ok
98 (xxx_node@netlab3)2>
99 (xxx_node@netlab3)2>
100 (xxx_node@netlab3)2>
101 (xxx_node@netlab3)2>
102 (xxx_node@netlab3)2>
103 (xxx_node@netlab3)2>
104 (xxx_node@netlab3)2> hm_cli:log_stop().
105 log stop:[dell1@dell] Result:[ok]
106 log stop:[dell5@dell] Result:[ok]
107 log stop:[dell6@dell] Result:[ok]
108 log stop:[dell2@dell] Result:[ok]
109 log stop:[dell7@dell] Result:[ok]

```

```

110 log stop:[dell8@dell] Result:[ok]
111 log stop:[dell3@dell] Result:[ok]
112 log stop:[dell4@dell] Result:[ok]
113 log stop:[netlabb2@netlabb] Result:[ok]
114 log stop:[netlabb1@netlabb] Result:[ok]
115 log stop:[netlabb4@netlabb] Result:[ok]
116 log stop:[netlabb7@netlabb] Result:[ok]
117 log stop:[netlabb8@netlabb] Result:[ok]
118 log stop:[netlabb6@netlabb] Result:[ok]
119 log stop:[netlabb5@netlabb] Result:[ok]
120 log stop:[netlabb3@netlabb] Result:[ok]
121 log stop:[netlaba4@netlaba] Result:[ok]
122 log stop:[netlaba7@netlaba] Result:[ok]
123 log stop:[netlaba5@netlaba] Result:[ok]
124 log stop:[netlaba1@netlaba] Result:[ok]
125 log stop:[netlaba2@netlaba] Result:[ok]
126 log stop:[netlaba8@netlaba] Result:[ok]
127 log stop:[netlaba3@netlaba] Result:[ok]
128 log stop:[netlaba6@netlaba] Result:[ok]
129 log stop:[netlab420@netlab4] Result:[ok]
130 log stop:[netlab415@netlab4] Result:[ok]
131 log stop:[netlab47@netlab4] Result:[ok]
132 log stop:[netlab45@netlab4] Result:[ok]
133 log stop:[netlab418@netlab4] Result:[ok]
134 log stop:[netlab419@netlab4] Result:[ok]
135 log stop:[netlab44@netlab4] Result:[ok]
136 log stop:[netlab46@netlab4] Result:[ok]
137 log stop:[netlab48@netlab4] Result:[ok]
138 log stop:[netlab412@netlab4] Result:[ok]
139 log stop:[netlab411@netlab4] Result:[ok]
140 log stop:[netlab413@netlab4] Result:[ok]
141 log stop:[netlab416@netlab4] Result:[ok]
142 log stop:[netlab417@netlab4] Result:[ok]
143 log stop:[netlab410@netlab4] Result:[ok]
144 log stop:[netlab414@netlab4] Result:[ok]
145 log stop:[netlab49@netlab4] Result:[ok]
146 log stop:[netlab43@netlab4] Result:[ok]
147 log stop:[netlab42@netlab4] Result:[ok]
148 log stop:[netlab41@netlab4] Result:[ok]
149 hm_log_h_file:closing 'log/harmonia_log_xxx.txt'
150 log stop:[xxx_node@netlab3] Result:[ok]
151 log stop:[netlab319@netlab3] Result:[ok]
152 log stop:[netlab34@netlab3] Result:[ok]
153 log stop:[netlab38@netlab3] Result:[ok]
154 log stop:[netlab318@netlab3] Result:[ok]
155 log stop:[netlab317@netlab3] Result:[ok]
156 log stop:[netlab32@netlab3] Result:[ok]
157 log stop:[netlab35@netlab3] Result:[ok]
158 log stop:[netlab315@netlab3] Result:[ok]
159 log stop:[netlab320@netlab3] Result:[ok]
160 log stop:[netlab316@netlab3] Result:[ok]
161 log stop:[netlab313@netlab3] Result:[ok]
162 log stop:[netlab314@netlab3] Result:[ok]
163 log stop:[netlab312@netlab3] Result:[ok]
164 log stop:[netlab311@netlab3] Result:[ok]
165 log stop:[netlab39@netlab3] Result:[ok]
166 log stop:[netlab310@netlab3] Result:[ok]
167 log stop:[netlab37@netlab3] Result:[ok]
168 log stop:[netlab36@netlab3] Result:[ok]
169 log stop:[netlab33@netlab3] Result:[ok]

```

```

170 log stop:[netlab31@netlab3] Result:[ok]
171 ok
172 (xxx_node@netlab3)3> hm_cli_test:test_perf(10000).
173 starting....
174 store          OK....[      59.9408410000] sec
175 get            OK....[      30.4779760000] sec
176 cstore         OK....[      59.6026520000] sec
177 cget           OK....[       0.0324080000] sec
178 drop_table()   OK....[       0.0086890000] sec
179 create_table   OK....[       0.0084170000] sec
180 rstore         OK....[     132.6259150000] sec
181 rget           OK....[     105.6835490000] sec
182 ..end
183 ok
184 (xxx_node@netlab3)4>

```

List 44: Test Results:Comparison with different number of nodes:105 nodes

```

1 hirotnk@netlab3 ~/harmonia/Harmonia $ erl -pa ebin -config harmonia -run hm_edge start -\
  →harmonia node_type 'join' name xxx sname 'xxx_node@netlab3' root_node 'netlab31@netlab3' \
  →setcookie harmonia_cookie -sname 'xxx_node@netlab3'
2 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
3
4 Eshell V5.6.3 (abort with ^G)
5 (xxx_node@netlab3)1> [{name,xxx},
6 {root_node,netlab31@netlab3},
7 {root,netlab31},
8 {included_applications,[],},
9 {sname,xxx_node@netlab3},
10 {logfile_ext, ".txt"},
11 {node_type,join},
12 {logdir,"log/"},
13 {logfile,"harmonia_log"}]
14 "log/harmonia_log_xxx.txt"
15 start Pid:[<0.384.0>]
16
17 (xxx_node@netlab3)1> hm_cli_test:test_all().
18 starting....
19 store(10)      OK....
20 get(10)        OK....
21 cstore(10)     OK....
22 cget(10)       OK....
23 drop_table()   "OK"....
24 create_table() "OK"....
25 rstore(10)     OK....
26 rget(10)       OK....
27 ..end
28 rangeq_test1 start
29 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
30 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
31 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
32 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
33 rangeq_test2 start
34 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok
35 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
36 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
37 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
38 rangeq_test3 start
39 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok
40 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok

```

```

41 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
42 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok
43 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
44 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
45 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok
46 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
47 rangeq_test4 start
48 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
49 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
50 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok
51 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
52 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok
53 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
54 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
55 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
56 rangeq_test5 start
57 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
58 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
59 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
60 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
61 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
62 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
63 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
64 rangeq_test1 start
65 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
66 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
67 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
68 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
69 rangeq_test2 start
70 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok
71 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
72 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
73 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
74 rangeq_test3 start
75 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok
76 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok
77 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
78 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok
79 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
80 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
81 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok
82 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
83 rangeq_test4 start
84 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
85 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
86 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok
87 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
88 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok
89 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
90 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
91 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
92 rangeq_test5 start
93 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
94 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
95 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
96 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
97 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
98 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
99 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
100 ok

```

```

101 (xxx_node@netlab3)2> hm_cli:log_stop().
102 hm_log_h_file:closing 'log/harmonia_log_xxx.txt'
103 log stop:[xxx_node@netlab3] Result:[ok]
104 log stop:[dell8@dell] Result:[ok]
105 log stop:[dell7@dell] Result:[ok]
106 log stop:[netlab2@netlab] Result:[ok]
107 log stop:[dell5@dell] Result:[ok]
108 log stop:[dell3@dell] Result:[ok]
109 log stop:[dell2@dell] Result:[ok]
110 log stop:[netlab8@netlab] Result:[ok]
111 log stop:[dell6@dell] Result:[ok]
112 log stop:[dell1@dell] Result:[ok]
113 log stop:[dell4@dell] Result:[ok]
114 log stop:[netlab6@netlab] Result:[ok]
115 log stop:[netlab5@netlab] Result:[ok]
116 log stop:[netlab7@netlab] Result:[ok]
117 log stop:[netlab3@netlab] Result:[ok]
118 log stop:[netlab1@netlab] Result:[ok]
119 log stop:[netlab4@netlab] Result:[ok]
120 log stop:[netlab4@netlab] Result:[ok]
121 log stop:[netlab3@netlab] Result:[ok]
122 log stop:[netlab2@netlab] Result:[ok]
123 log stop:[netlab1@netlab] Result:[ok]
124 log stop:[netlab6@netlab] Result:[ok]
125 log stop:[netlab5@netlab] Result:[ok]
126 log stop:[netlab7@netlab] Result:[ok]
127 log stop:[netlab8@netlab] Result:[ok]
128 log stop:[netlab427@netlab4] Result:[ok]
129 log stop:[netlab426@netlab4] Result:[ok]
130 log stop:[netlab417@netlab4] Result:[ok]
131 log stop:[netlab425@netlab4] Result:[ok]
132 log stop:[netlab434@netlab4] Result:[ok]
133 log stop:[netlab413@netlab4] Result:[ok]
134 log stop:[netlab439@netlab4] Result:[ok]
135 log stop:[netlab440@netlab4] Result:[ok]
136 log stop:[netlab49@netlab4] Result:[ok]
137 log stop:[netlab416@netlab4] Result:[ok]
138 log stop:[netlab415@netlab4] Result:[ok]
139 log stop:[netlab431@netlab4] Result:[ok]
140 log stop:[netlab412@netlab4] Result:[ok]
141 log stop:[netlab437@netlab4] Result:[ok]
142 log stop:[netlab421@netlab4] Result:[ok]
143 log stop:[netlab414@netlab4] Result:[ok]
144 log stop:[netlab436@netlab4] Result:[ok]
145 log stop:[netlab435@netlab4] Result:[ok]
146 log stop:[netlab429@netlab4] Result:[ok]
147 log stop:[netlab46@netlab4] Result:[ok]
148 log stop:[netlab432@netlab4] Result:[ok]
149 log stop:[netlab438@netlab4] Result:[ok]
150 log stop:[netlab48@netlab4] Result:[ok]
151 log stop:[netlab418@netlab4] Result:[ok]
152 log stop:[netlab430@netlab4] Result:[ok]
153 log stop:[netlab419@netlab4] Result:[ok]
154 log stop:[netlab428@netlab4] Result:[ok]
155 log stop:[netlab43@netlab4] Result:[ok]
156 log stop:[netlab433@netlab4] Result:[ok]
157 log stop:[netlab42@netlab4] Result:[ok]
158 log stop:[netlab41@netlab4] Result:[ok]
159 log stop:[netlab422@netlab4] Result:[ok]
160 log stop:[netlab411@netlab4] Result:[ok]

```



```

161 log stop:[netlab420@netlab4] Result:[ok]
162 log stop:[netlab424@netlab4] Result:[ok]
163 log stop:[netlab445@netlab4] Result:[ok]
164 log stop:[netlab47@netlab4] Result:[ok]
165 log stop:[netlab423@netlab4] Result:[ok]
166 log stop:[netlab44@netlab4] Result:[ok]
167 log stop:[netlab410@netlab4] Result:[ok]
168 log stop:[netlab37@netlab3] Result:[ok]
169 log stop:[netlab336@netlab3] Result:[ok]
170 log stop:[netlab318@netlab3] Result:[ok]
171 log stop:[netlab327@netlab3] Result:[ok]
172 log stop:[netlab331@netlab3] Result:[ok]
173 log stop:[netlab36@netlab3] Result:[ok]
174 log stop:[netlab333@netlab3] Result:[ok]
175 log stop:[netlab328@netlab3] Result:[ok]
176 log stop:[netlab311@netlab3] Result:[ok]
177 log stop:[netlab312@netlab3] Result:[ok]
178 log stop:[netlab324@netlab3] Result:[ok]
179 log stop:[netlab323@netlab3] Result:[ok]
180 log stop:[netlab321@netlab3] Result:[ok]
181 log stop:[netlab335@netlab3] Result:[ok]
182 log stop:[netlab337@netlab3] Result:[ok]
183 log stop:[netlab33@netlab3] Result:[ok]
184 log stop:[netlab332@netlab3] Result:[ok]
185 log stop:[netlab326@netlab3] Result:[ok]
186 log stop:[netlab329@netlab3] Result:[ok]
187 log stop:[netlab317@netlab3] Result:[ok]
188 log stop:[netlab330@netlab3] Result:[ok]
189 log stop:[netlab315@netlab3] Result:[ok]
190 log stop:[netlab339@netlab3] Result:[ok]
191 log stop:[netlab316@netlab3] Result:[ok]
192 log stop:[netlab320@netlab3] Result:[ok]
193 log stop:[netlab340@netlab3] Result:[ok]
194 log stop:[netlab338@netlab3] Result:[ok]
195 log stop:[netlab319@netlab3] Result:[ok]
196 log stop:[netlab334@netlab3] Result:[ok]
197 log stop:[netlab325@netlab3] Result:[ok]
198 log stop:[netlab322@netlab3] Result:[ok]
199 log stop:[netlab34@netlab3] Result:[ok]
200 log stop:[netlab314@netlab3] Result:[ok]
201 log stop:[netlab38@netlab3] Result:[ok]
202 log stop:[netlab313@netlab3] Result:[ok]
203 log stop:[netlab310@netlab3] Result:[ok]
204 log stop:[netlab39@netlab3] Result:[ok]
205 log stop:[netlab35@netlab3] Result:[ok]
206 log stop:[netlab32@netlab3] Result:[ok]
207 log stop:[netlab31@netlab3] Result:[ok]
208 ok
209 (xxx_node@netlab3)3> hm_cli_test:test_perf(10000).
210 starting....
211 store      OK....[      66.1194410000] sec
212 get        OK....[      33.1734910000] sec
213 cstore     OK....[      65.9298560000] sec
214 cget       OK....[       0.0333200000] sec
215 drop_table() OK....[       0.0110690000] sec
216 create_table OK....[       0.0154710000] sec
217 rstore     OK....[     136.9953950000] sec
218 rget       OK....[     74.6138880000] sec
219 ..end
220 ok

```

221 (xxx_node@netlab3)4>

G.2 Comparison with Different Number of Threads

List 45: Test Results: Comparison with different number of threads

```

1 hirotnk@netlab3 ~/harmonia/Harmonia $ test/test20.sh -s
2 Launched the node:[netlab31@netlab3]
3 Launched the node:[netlab32@netlab3]
4 Launched the node:[netlab33@netlab3]
5 Launched the node:[netlab34@netlab3]
6 Launched the node:[netlab35@netlab3]
7 Launched the node:[netlab36@netlab3]
8 Launched the node:[netlab37@netlab3]
9 Launched the node:[netlab38@netlab3]
10 Launched the node:[netlab39@netlab3]
11 Launched the node:[netlab310@netlab3]
12 Launched the node:[netlab311@netlab3]
13 Launched the node:[netlab312@netlab3]
14 Launched the node:[netlab313@netlab3]
15 Launched the node:[netlab314@netlab3]
16 Launched the node:[netlab315@netlab3]
17 Launched the node:[netlab316@netlab3]
18 Launched the node:[netlab317@netlab3]
19 Launched the node:[netlab318@netlab3]
20 Launched the node:[netlab319@netlab3]
21 Launched the node:[netlab320@netlab3]
22 epmd: up and running on port 4369 with data:
23 name netlab315 at port 46474
24 name netlab317 at port 34650
25 name netlab314 at port 42667
26 name netlab313 at port 33677
27 name netlab312 at port 34679
28 name netlab311 at port 56225
29 name netlab310 at port 39671
30 name netlab39 at port 50195
31 name netlab37 at port 60907
32 name netlab38 at port 47783
33 name netlab36 at port 34496
34 name netlab35 at port 46560
35 name netlab34 at port 44108
36 name netlab33 at port 59481
37 name netlab32 at port 42054
38 name netlab31 at port 42949
39 hirotnk@netlab3 ~/harmonia/Harmonia $ erl -pa ebin -config harmonia -run hm_edge start -\
    →harmonia node_type 'join' name xxx sname 'xxx_node@netlab3' root_node 'netlab31@netlab3' \
    →setcookie harmonia_cookie -sname 'xxx_node@netlab3'
40 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
41
42 Eshell V5.6.3 (abort with ^G)
43 (xxx_node@netlab3)1> [{name,xxx},
44 {root_node,netlab31@netlab3},
45 {root,netlab31},
46 {included_applications,[],},
47 {sname,xxx_node@netlab3},
48 {logfile_ext,".txt"},
49 {node_type,join},
50 {logdir,"log/"},
51 {logfile,"harmonia_log"}]
52 "log/harmonia_log_xxx.txt"
53 start Pid:[<0.259.0>]
54

```

```

55 (xxx_node@netlab3)1>
56 (xxx_node@netlab3)1>
57 (xxx_node@netlab3)1>
58 (xxx_node@netlab3)1>
59 (xxx_node@netlab3)1>
60 (xxx_node@netlab3)1>
61 (xxx_node@netlab3)1> hm_cli_test:test_all().
62 starting....
63 store(10)      OK....
64 get(10)        OK....
65 cstore(10)     OK....
66 cget(10)       OK....
67 drop_table()  "OK"....
68 create_table() "OK"....
69 rstore(10)     OK....
70 rget(10)       OK....
71 ..end
72 rangeq_test1 start
73 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
74 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
75 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
76 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
77 rangeq_test2 start
78 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok
79 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
80 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
81 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
82 rangeq_test3 start
83 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok
84 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok
85 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
86 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok
87 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
88 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
89 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok
90 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
91 rangeq_test4 start
92 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
93 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
94 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok
95 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
96 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok
97 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
98 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
99 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
100 rangeq_test5 start
101 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
102 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
103 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
104 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
105 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
106 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
107 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
108 rangeq_test1 start
109 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
110 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
111 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
112 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
113 rangeq_test2 start
114 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok

```

```

115 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
116 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
117 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
118 rangeq_test3 start
119 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok
120 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok
121 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
122 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok
123 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
124 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
125 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok
126 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
127 rangeq_test4 start
128 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
129 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
130 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok
131 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
132 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok
133 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
134 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
135 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
136 rangeq_test5 start
137 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
138 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
139 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
140 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
141 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
142 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
143 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
144 ok
145 (xxx_node@netlab3)2> hm_cli:log_stop().
146 log stop:[dell2@dell] Result:[ok]
147 log stop:[dell1@dell] Result:[ok]
148 log stop:[netlab2@netlab] Result:[ok]
149 log stop:[netlab1@netlab] Result:[ok]
150 log stop:[netlab1@netlab] Result:[ok]
151 log stop:[netlab2@netlab] Result:[ok]
152 log stop:[netlab419@netlab4] Result:[ok]
153 log stop:[netlab411@netlab4] Result:[ok]
154 log stop:[netlab416@netlab4] Result:[ok]
155 log stop:[netlab418@netlab4] Result:[ok]
156 log stop:[netlab42@netlab4] Result:[ok]
157 log stop:[netlab41@netlab4] Result:[ok]
158 log stop:[netlab415@netlab4] Result:[ok]
159 log stop:[netlab48@netlab4] Result:[ok]
160 log stop:[netlab49@netlab4] Result:[ok]
161 log stop:[netlab43@netlab4] Result:[ok]
162 log stop:[netlab414@netlab4] Result:[ok]
163 log stop:[netlab412@netlab4] Result:[ok]
164 log stop:[netlab413@netlab4] Result:[ok]
165 log stop:[netlab420@netlab4] Result:[ok]
166 log stop:[netlab46@netlab4] Result:[ok]
167 log stop:[netlab417@netlab4] Result:[ok]
168 log stop:[netlab410@netlab4] Result:[ok]
169 log stop:[netlab47@netlab4] Result:[ok]
170 log stop:[netlab45@netlab4] Result:[ok]
171 log stop:[netlab44@netlab4] Result:[ok]
172 hm_log_h_file:closing 'log/harmonia_log_xxx.txt'
173 log stop:[xxx_node@netlab3] Result:[ok]
174 log stop:[netlab34@netlab3] Result:[ok]

```

```

175 log stop:[netlab315@netlab3] Result:[ok]
176 log stop:[netlab320@netlab3] Result:[ok]
177 log stop:[netlab37@netlab3] Result:[ok]
178 log stop:[netlab38@netlab3] Result:[ok]
179 log stop:[netlab33@netlab3] Result:[ok]
180 log stop:[netlab316@netlab3] Result:[ok]
181 log stop:[netlab39@netlab3] Result:[ok]
182 log stop:[netlab319@netlab3] Result:[ok]
183 log stop:[netlab318@netlab3] Result:[ok]
184 log stop:[netlab317@netlab3] Result:[ok]
185 log stop:[netlab311@netlab3] Result:[ok]
186 log stop:[netlab314@netlab3] Result:[ok]
187 log stop:[netlab313@netlab3] Result:[ok]
188 log stop:[netlab312@netlab3] Result:[ok]
189 log stop:[netlab310@netlab3] Result:[ok]
190 log stop:[netlab36@netlab3] Result:[ok]
191 log stop:[netlab35@netlab3] Result:[ok]
192 log stop:[netlab32@netlab3] Result:[ok]
193 log stop:[netlab31@netlab3] Result:[ok]
194 ok
195 (xxx_node@netlab3)3> hm_cli_test:test_perf(10000).
196 starting....
197 store          OK....[          46.9935960000] sec
198 get            OK....[          20.0107650000] sec
199 cstore         OK....[          47.0702910000] sec
200 cget           OK....[           0.0321730000] sec
201 drop_table()   OK....[           0.0053280000] sec
202 create_table    OK....[           0.0060850000] sec
203 rstore         OK....[          104.4188560000] sec
204 rget           OK....[           52.5535400000] sec
205 ..end
206 ok
207 (xxx_node@netlab3)4> hm_cli_test:thread_test([netlab31@netlab3, 1 , 5000}, {\
->netlab41@netlab4, 5001, 10000})).
208 Node:[netlab31@netlab3] rget Time:[          34.8632630000] sec
209 Node:[netlab41@netlab4] rget Time:[          39.7251560000] sec
210 [          39.7513980000] sec
211 Node:[netlab41@netlab4] get Time:[          10.6438910000] sec
212 Node:[netlab31@netlab3] get Time:[          10.7430180000] sec
213 [          10.8382960000] sec
214 ok
215 (xxx_node@netlab3)5>
216 (xxx_node@netlab3)5>
217 (xxx_node@netlab3)5> hm_cli_test:thread_test([
218 (xxx_node@netlab3)5> {netlab31@netlab3, 1 , 2500},
219 (xxx_node@netlab3)5> {netlab41@netlab4, 2501 , 5000},
220 (xxx_node@netlab3)5> {netlabal@netlaba, 5001 , 7500},
221 (xxx_node@netlab3)5> {netlabbl@netlabbb, 7501 , 10000}
222 (xxx_node@netlab3)5> ]).
223 Node:[netlab31@netlab3] rget Time:[          25.1710290000] sec
224 Node:[netlab41@netlab4] rget Time:[          28.3082240000] sec
225 Node:[netlabbl@netlabbb] rget Time:[          33.0683130000] sec
226 Node:[netlabal@netlaba] rget Time:[          33.8500710000] sec
227 [          33.8717430000] sec
228 Node:[netlab41@netlab4] get Time:[           6.1612500000] sec
229 Node:[netlab31@netlab3] get Time:[           6.2409330000] sec
230 Node:[netlabbl@netlabbb] get Time:[           8.8720420000] sec
231 Node:[netlabal@netlaba] get Time:[           9.2702770000] sec
232 [           9.4164350000] sec
233 ok

```

```

234 (xxx_node@netlab3)6>
235 (xxx_node@netlab3)6>
236 (xxx_node@netlab3)6>
237 (xxx_node@netlab3)6> hm_cli_test:thread_test([
238 (xxx_node@netlab3)6> {netlab31@netlab3, 1 , 1250},
239 (xxx_node@netlab3)6> {netlab41@netlab4, 2501 , 3750},
240 (xxx_node@netlab3)6> {netlab42@netlab4, 3751 , 5000},
241 (xxx_node@netlab3)6> {netlab43@netlab4, 5001 , 6250},
242 (xxx_node@netlab3)6> {netlab44@netlab4, 1251 , 2500},
243 (xxx_node@netlab3)6> {netlab41@netlab4, 6251 , 7500},
244 (xxx_node@netlab3)6> {netlab41@netlab4, 7501 , 8750},
245 (xxx_node@netlab3)6> {dell1@dell, 8751 , 10000}
246 (xxx_node@netlab3)6> ]).
247 Node:[netlab31@netlab3] rget Time:[ 27.0090040000] sec
248 Node:[netlab42@netlab4] rget Time:[ 28.6678240000] sec
249 Node:[netlab44@netlab4] rget Time:[ 28.8462720000] sec
250 Node:[netlab43@netlab4] rget Time:[ 28.9583950000] sec
251 Node:[netlab41@netlab4] rget Time:[ 28.9750950000] sec
252 Node:[dell1@dell] rget Time:[ 29.9464060000] sec
253 Node:[netlab41@netlab4] rget Time:[ 30.1593430000] sec
254 Node:[netlab41@netlab4] rget Time:[ 30.2790660000] sec
255 [ 30.3188790000] sec
256 Node:[netlab31@netlab3] get Time:[ 3.8653860000] sec
257 Node:[netlab42@netlab4] get Time:[ 3.9150540000] sec
258 Node:[netlab43@netlab4] get Time:[ 3.9824040000] sec
259 Node:[netlab41@netlab4] get Time:[ 4.0593460000] sec
260 Node:[netlab44@netlab4] get Time:[ 4.1094260000] sec
261 Node:[dell1@dell] get Time:[ 4.6611080000] sec
262 Node:[netlab41@netlab4] get Time:[ 4.9480370000] sec
263 Node:[netlab41@netlab4] get Time:[ 5.1802750000] sec
264 [ 5.2974320000] sec
265 ok
266 (xxx_node@netlab3)7>
267 (xxx_node@netlab3)7>
268 (xxx_node@netlab3)7>
269 (xxx_node@netlab3)7>
270 (xxx_node@netlab3)7> hm_cli_test:thread_test([
271 (xxx_node@netlab3)7> {netlab31@netlab3, 1 , 310},
272 (xxx_node@netlab3)7> {netlab32@netlab3, 311 , 620},
273 (xxx_node@netlab3)7> {netlab33@netlab3, 621 , 930},
274 (xxx_node@netlab3)7> {netlab34@netlab3, 931 , 1240},
275 (xxx_node@netlab3)7> {netlab35@netlab3, 1241 , 1550},
276 (xxx_node@netlab3)7> {netlab36@netlab3, 1551 , 1860},
277 (xxx_node@netlab3)7> {netlab37@netlab3, 1861 , 2170},
278 (xxx_node@netlab3)7> {netlab38@netlab3, 2171 , 2480},
279 (xxx_node@netlab3)7> {netlab39@netlab3, 2481 , 2790},
280 (xxx_node@netlab3)7> {netlab310@netlab3, 2791 , 3100},
281 (xxx_node@netlab3)7> {netlab311@netlab3, 3101 , 3410},
282 (xxx_node@netlab3)7> {netlab312@netlab3, 3411 , 3720},
283 (xxx_node@netlab3)7> {netlab313@netlab3, 3721 , 4030},
284 (xxx_node@netlab3)7> {netlab41@netlab4, 4031 , 4340},
285 (xxx_node@netlab3)7> {netlab42@netlab4, 4341 , 4650},
286 (xxx_node@netlab3)7> {netlab43@netlab4, 4651 , 4960},
287 (xxx_node@netlab3)7> {netlab44@netlab4, 4961 , 5270},
288 (xxx_node@netlab3)7> {netlab45@netlab4, 5271 , 5580},
289 (xxx_node@netlab3)7> {netlab46@netlab4, 5581 , 5890},
290 (xxx_node@netlab3)7> {netlab47@netlab4, 5891 , 6200},
291 (xxx_node@netlab3)7> {netlab48@netlab4, 6201 , 6510},
292 (xxx_node@netlab3)7> {netlab49@netlab4, 6511 , 6820},
293 (xxx_node@netlab3)7> {netlab410@netlab4, 6821 , 7130},

```

```

294 (xxx_node@netlab3)7> {netlab411@netlab4, 7131 , 7440},
295 (xxx_node@netlab3)7> {netlab412@netlab4, 7441 , 7750},
296 (xxx_node@netlab3)7> {netlab413@netlab4, 7751 , 8060},
297 (xxx_node@netlab3)7> {netlaba1@netlaba, 8061 , 8370},
298 (xxx_node@netlab3)7> {netlaba2@netlaba, 8371 , 8680},
299 (xxx_node@netlab3)7> {netlabb1@netlabb, 8681 , 8990},
300 (xxx_node@netlab3)7> {netlabb2@netlabb, 8991 , 9200},
301 (xxx_node@netlab3)7> {dell1@dell, 9201 , 9510},
302 (xxx_node@netlab3)7> {dell2@dell, 9511 , 9820},
303 (xxx_node@netlab3)7> {netlab414@netlab4, 9821 , 10000}
304 (xxx_node@netlab3)7> ]).
305 Node:[netlab414@netlab4] rget Time:[ 19.3991870000] sec
306 Node:[netlabb2@netlabb] rget Time:[ 23.4295940000] sec
307 Node:[netlab31@netlab3] rget Time:[ 32.1423410000] sec
308 Node:[netlab37@netlab3] rget Time:[ 32.3318740000] sec
309 Node:[netlab312@netlab3] rget Time:[ 32.3359090000] sec
310 Node:[netlab32@netlab3] rget Time:[ 32.4679180000] sec
311 Node:[netlab34@netlab3] rget Time:[ 32.5500760000] sec
312 Node:[netlab38@netlab3] rget Time:[ 32.5357480000] sec
313 Node:[netlab39@netlab3] rget Time:[ 32.5320050000] sec
314 Node:[netlab35@netlab3] rget Time:[ 32.6639460000] sec
315 Node:[netlab310@netlab3] rget Time:[ 32.6522910000] sec
316 Node:[netlab311@netlab3] rget Time:[ 32.6494250000] sec
317 Node:[netlab33@netlab3] rget Time:[ 32.7071810000] sec
318 Node:[netlab36@netlab3] rget Time:[ 32.7112620000] sec
319 Node:[netlab45@netlab4] rget Time:[ 32.6632660000] sec
320 Node:[netlab313@netlab3] rget Time:[ 32.6904370000] sec
321 Node:[netlab46@netlab4] rget Time:[ 32.6879860000] sec
322 Node:[netlab42@netlab4] rget Time:[ 32.7077970000] sec
323 Node:[netlab47@netlab4] rget Time:[ 32.7153310000] sec
324 Node:[netlab48@netlab4] rget Time:[ 32.7485480000] sec
325 Node:[netlab410@netlab4] rget Time:[ 32.7560200000] sec
326 Node:[netlab43@netlab4] rget Time:[ 32.7988560000] sec
327 Node:[netlab41@netlab4] rget Time:[ 32.8327040000] sec
328 Node:[netlab411@netlab4] rget Time:[ 32.8165490000] sec
329 Node:[netlab44@netlab4] rget Time:[ 32.8913060000] sec
330 Node:[netlab49@netlab4] rget Time:[ 32.8852860000] sec
331 Node:[netlab413@netlab4] rget Time:[ 32.8909320000] sec
332 Node:[netlab412@netlab4] rget Time:[ 32.8978420000] sec
333 Node:[dell1@dell] rget Time:[ 32.9266430000] sec
334 Node:[netlaba2@netlaba] rget Time:[ 32.9328340000] sec
335 Node:[netlabb1@netlabb] rget Time:[ 32.9460550000] sec
336 Node:[dell2@dell] rget Time:[ 32.9372570000] sec
337 Node:[netlaba1@netlaba] rget Time:[ 32.9534890000] sec
338 [ 33.0868620000] sec
339 Node:[netlab414@netlab4] get Time:[ 1.1750410000] sec
340 Node:[netlabb2@netlabb] get Time:[ 1.8860840000] sec
341 Node:[netlab31@netlab3] get Time:[ 1.9701320000] sec
342 Node:[netlab310@netlab3] get Time:[ 1.9579320000] sec
343 Node:[netlab42@netlab4] get Time:[ 1.9646300000] sec
344 Node:[netlab45@netlab4] get Time:[ 1.9623810000] sec
345 Node:[netlab33@netlab3] get Time:[ 2.0262730000] sec
346 Node:[netlab39@netlab3] get Time:[ 2.0327000000] sec
347 Node:[netlab48@netlab4] get Time:[ 1.9990740000] sec
348 Node:[netlab43@netlab4] get Time:[ 2.0180690000] sec
349 Node:[netlab411@netlab4] get Time:[ 2.0295240000] sec
350 Node:[netlab410@netlab4] get Time:[ 2.0460110000] sec
351 Node:[netlab413@netlab4] get Time:[ 2.0446530000] sec
352 Node:[netlab35@netlab3] get Time:[ 2.1402860000] sec
353 Node:[netlab311@netlab3] get Time:[ 2.1980090000] sec

```



```

354 Node:[netlab38@netlab3] get Time:[      2.2101450000] sec
355 Node:[netlab47@netlab4] get Time:[      2.1771450000] sec
356 Node:[netlab36@netlab3] get Time:[      2.2284290000] sec
357 Node:[netlab312@netlab3] get Time:[      2.2423930000] sec
358 Node:[netlab313@netlab3] get Time:[      2.2465110000] sec
359 Node:[netlab37@netlab3] get Time:[      2.2698890000] sec
360 Node:[netlab34@netlab3] get Time:[      2.2891460000] sec
361 Node:[netlab32@netlab3] get Time:[      2.3888460000] sec
362 Node:[netlab46@netlab4] get Time:[      2.3477140000] sec
363 Node:[netlab412@netlab4] get Time:[      2.3651260000] sec
364 Node:[dell2@dell] get Time:[      2.3208310000] sec
365 Node:[netlab44@netlab4] get Time:[      2.3956170000] sec
366 Node:[netlab41@netlab4] get Time:[      2.4062270000] sec
367 Node:[netlab41@netlab4] get Time:[      2.4062270000] sec
368 Node:[netlab49@netlab4] get Time:[      2.4096100000] sec
369 Node:[dell1@dell] get Time:[      2.3908860000] sec
370 Node:[netlab2@netlab2] get Time:[      2.5266660000] sec
371 Node:[netlab1@netlab1] get Time:[      2.6388460000] sec
372 [      2.8253160000] sec
373 ok
374 (xxx_node@netlab3)8>

```

G.3 Comparison with Different Range Query Conditions

List 46: Test Results: Comparison with different query conditions

```

1 hirotnk@netlab3 ~/harmonia/Harmonia $ test/test20.sh -s
2 Launched the node:[netlab31@netlab3]
3 Launched the node:[netlab32@netlab3]
4 Launched the node:[netlab33@netlab3]
5 Launched the node:[netlab34@netlab3]
6 Launched the node:[netlab35@netlab3]
7 Launched the node:[netlab36@netlab3]
8 Launched the node:[netlab37@netlab3]
9 Launched the node:[netlab38@netlab3]
10 Launched the node:[netlab39@netlab3]
11 Launched the node:[netlab310@netlab3]
12 Launched the node:[netlab311@netlab3]
13 Launched the node:[netlab312@netlab3]
14 Launched the node:[netlab313@netlab3]
15 Launched the node:[netlab314@netlab3]
16 Launched the node:[netlab315@netlab3]
17 Launched the node:[netlab316@netlab3]
18 Launched the node:[netlab317@netlab3]
19 Launched the node:[netlab318@netlab3]
20 Launched the node:[netlab319@netlab3]
21 Launched the node:[netlab320@netlab3]
22 epmd: up and running on port 4369 with data:
23 name netlab316 at port 47401
24 name netlab315 at port 60855
25 name netlab312 at port 34283
26 name netlab311 at port 36128
27 name netlab39 at port 53578
28 name netlab310 at port 36914
29 name netlab38 at port 34030
30 name netlab37 at port 41609
31 name netlab36 at port 51462
32 name netlab35 at port 33478
33 name netlab34 at port 33252
34 name netlab33 at port 52701
35 name netlab32 at port 40227
36 name netlab31 at port 59164
37 hirotnk@netlab3 ~/harmonia/Harmonia $ erl -pa ebin -config harmonia -run hm_edge start -\
    →harmonia node_type 'join' name xxx sname 'xxx_node@netlab3' root_node 'netlab31@netlab3' \
    →setcookie harmonia_cookie -sname 'xxx_node@netlab3'
38 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
39
40 Eshell V5.6.3 (abort with ^G)
41 (xxx_node@netlab3)1> [{name,xxx},
42 {root_node,netlab31@netlab3},
43 {root,netlab31},
44 {included_applications,[],},
45 {sname,xxx_node@netlab3},
46 {logfile_ext,".txt"},
47 {node_type,join},
48 {logdir,"log/"},
49 {logfile,"harmonia_log"}]
50 "log/harmonia_log_xxx.txt"
51 start Pid:[<0.245.0>]
52
53 (xxx_node@netlab3)1> hm_cli_test:test_all().
54 starting....

```

```

55 store(10)      OK....
56 get(10)        OK....
57 cstore(10)     OK....
58 cget(10)       OK....
59 drop_table()   "OK"....
60 create_table() "OK"....
61 rstore(10)     OK....
62 rget(10)       OK....
63 ..end
64 rangeq_test1 start
65 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
66 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
67 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
68 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
69 rangeq_test2 start
70 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok
71 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
72 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
73 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
74 rangeq_test3 start
75 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok
76 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok
77 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
78 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok
79 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
80 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
81 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok
82 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
83 rangeq_test4 start
84 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
85 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
86 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok
87 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
88 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok
89 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
90 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
91 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
92 rangeq_test5 start
93 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
94 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
95 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
96 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
97 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
98 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
99 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
100 rangeq_test1 start
101 ["case1" "Domain1" "Tbl2" "Fld2 == 32"]:ok
102 ["case2" "Domain1" "Tbl2" "Fld2 == 150"]:ok
103 ["case3" "Domain1" "Tbl2" "Fld2 == 3000"]:ok
104 ["case4" "Domain1" "Tbl2" "Fld2 == 9000"]:ok
105 rangeq_test2 start
106 ["case1" "Domain1" "Tbl2" "Fld2 != 32"]:ok
107 ["case2" "Domain1" "Tbl2" "Fld2 != 150"]:ok
108 ["case3" "Domain1" "Tbl2" "Fld2 != 3000"]:ok
109 ["case4" "Domain1" "Tbl2" "Fld2 != 9000"]:ok
110 rangeq_test3 start
111 ["case1" "Domain1" "Tbl2" "Fld2 > 32"]:ok
112 ["case2" "Domain1" "Tbl2" "Fld2 > 150"]:ok
113 ["case3" "Domain1" "Tbl2" "Fld2 > 3000"]:ok
114 ["case4" "Domain1" "Tbl2" "Fld2 > 9000"]:ok

```

```

115 ["case5" "Domain1" "Tbl2" "Fld2 < 32"]:ok
116 ["case6" "Domain1" "Tbl2" "Fld2 < 150"]:ok
117 ["case7" "Domain1" "Tbl2" "Fld2 < 3000"]:ok
118 ["case8" "Domain1" "Tbl2" "Fld2 < 9000"]:ok
119 rangeq_test4 start
120 ["case1" "Domain1" "Tbl2" "Fld2 >= 32"]:ok
121 ["case2" "Domain1" "Tbl2" "Fld2 >= 150"]:ok
122 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000"]:ok
123 ["case3" "Domain1" "Tbl2" "Fld2 >= 9000"]:ok
124 ["case4" "Domain1" "Tbl2" "Fld2 <= 32"]:ok
125 ["case5" "Domain1" "Tbl2" "Fld2 <= 150"]:ok
126 ["case6" "Domain1" "Tbl2" "Fld2 <= 3000"]:ok
127 ["case7" "Domain1" "Tbl2" "Fld2 <= 9000"]:ok
128 rangeq_test5 start
129 ["case1" "Domain1" "Tbl2" "Fld2 >= 32 and Fld2 <= 150"]:ok
130 ["case2" "Domain1" "Tbl2" "Fld2 >= 3000 or Fld2 <= 150"]:ok
131 ["case3" "Domain1" "Tbl2" "(Fld2 >= 32 and Fld2 <= 150) and Fld3 == textfile1"]:ok
132 ["case4" "Domain1" "Tbl2" "Fld2 >= 3000 and Fld2 <= 150"]:ok
133 ["case5" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
134 ["case6" "Domain1" "Tbl2" "Fld1 == yyy or Fld2 == 32"]:ok
135 ["case7" "Domain1" "Tbl2" "Fld1 == yyy and Fld2 == 150"]:ok
136 ok
137 (xxx_node@netlab3)2> hm_cli:log_stop().
138 log stop:[dell3@dell] Result:[ok]
139 log stop:[dell4@dell] Result:[ok]
140 log stop:[dell2@dell] Result:[ok]
141 log stop:[dell1@dell] Result:[ok]
142 log stop:[netlab2@netlab] Result:[ok]
143 log stop:[netlab3@netlab] Result:[ok]
144 log stop:[netlab4@netlab] Result:[ok]
145 log stop:[netlab1@netlab] Result:[ok]
146 log stop:[netlab4@netlab] Result:[ok]
147 log stop:[netlab3@netlab] Result:[ok]
148 log stop:[netlab1@netlab] Result:[ok]
149 log stop:[netlab2@netlab] Result:[ok]
150 log stop:[netlab4@netlab] Result:[ok]
151 log stop:[netlab413@netlab4] Result:[ok]
152 log stop:[netlab43@netlab4] Result:[ok]
153 log stop:[netlab418@netlab4] Result:[ok]
154 log stop:[netlab415@netlab4] Result:[ok]
155 log stop:[netlab410@netlab4] Result:[ok]
156 log stop:[netlab47@netlab4] Result:[ok]
157 log stop:[netlab411@netlab4] Result:[ok]
158 log stop:[netlab419@netlab4] Result:[ok]
159 log stop:[netlab420@netlab4] Result:[ok]
160 log stop:[netlab48@netlab4] Result:[ok]
161 log stop:[netlab417@netlab4] Result:[ok]
162 log stop:[netlab416@netlab4] Result:[ok]
163 log stop:[netlab44@netlab4] Result:[ok]
164 log stop:[netlab414@netlab4] Result:[ok]
165 log stop:[netlab49@netlab4] Result:[ok]
166 log stop:[netlab412@netlab4] Result:[ok]
167 log stop:[netlab42@netlab4] Result:[ok]
168 log stop:[netlab45@netlab4] Result:[ok]
169 log stop:[netlab41@netlab4] Result:[ok]
170 hm_log_h_file:closing 'log/harmonia_log_xxx.txt'
171 log stop:[xxx_node@netlab3] Result:[ok]
172 log stop:[netlab316@netlab3] Result:[ok]
173 log stop:[netlab39@netlab3] Result:[ok]
174 log stop:[netlab37@netlab3] Result:[ok]

```

```

175 log stop:[netlab33@netlab3] Result:[ok]
176 log stop:[netlab310@netlab3] Result:[ok]
177 log stop:[netlab319@netlab3] Result:[ok]
178 log stop:[netlab315@netlab3] Result:[ok]
179 log stop:[netlab313@netlab3] Result:[ok]
180 log stop:[netlab320@netlab3] Result:[ok]
181 log stop:[netlab312@netlab3] Result:[ok]
182 log stop:[netlab317@netlab3] Result:[ok]
183 log stop:[netlab314@netlab3] Result:[ok]
184 log stop:[netlab318@netlab3] Result:[ok]
185 log stop:[netlab311@netlab3] Result:[ok]
186 log stop:[netlab36@netlab3] Result:[ok]
187 log stop:[netlab38@netlab3] Result:[ok]
188 log stop:[netlab35@netlab3] Result:[ok]
189 log stop:[netlab34@netlab3] Result:[ok]
190 log stop:[netlab32@netlab3] Result:[ok]
191 log stop:[netlab31@netlab3] Result:[ok]
192 ok
193 (xxx_node@netlab3)3> hm_cli_test:test_perf(1000).
194 starting....
195 store      OK....[      5.0779480000] sec
196 get        OK....[      2.2984420000] sec
197 cstore     OK....[      5.0843840000] sec
198 cget       OK....[      0.0031910000] sec
199 drop_table() OK....[      0.0070440000] sec
200 create_table OK....[      0.0062670000] sec
201 rstore     OK....[     10.7119100000] sec
202 rget       OK....[      2.8442240000] sec
203 ..end
204 ok
205 (xxx_node@netlab3)4> hm_cli_test:test_comp_get(1000).
206 starting....
207 get  Between 1 and 1,    in 1000    OK....[      2.3449630000] sec
208 rget Between 1 and 1,    in 1000    OK....[      0.0036200000] sec
209 get  Between 1 and 10,   in 1000    OK....[      2.3220820000] sec
210 rget Between 1 and 10,   in 1000    OK....[      0.0057500000] sec
211 get  Between 1 and 100,  in 1000    OK....[      2.3593630000] sec
212 rget Between 1 and 100,  in 1000    OK....[      0.0259570000] sec
213 get  Between 1 and 500,  in 1000    OK....[      2.3153030000] sec
214 rget Between 1 and 500,  in 1000    OK....[      0.1184570000] sec
215 get  Between 1 and 1000, in 1000    OK....[      2.3882260000] sec
216 rget Between 1 and 1000, in 1000    OK....[      0.2341260000] sec
217 ..end
218 ok
219 (xxx_node@netlab3)5>

```

G.4 Memory Usage with Real Case

List 47: Test Results:Memory Usage:store function with short data

```

1 hirotnk@netlab3 ~/harmonia/Harmonia $ erl -pa ebin -config harmonia -run hm_edge start -\
  →harmonia node_type 'join' name xxx sname 'xxx_node@netlab3' root_node 'netlab31@netlab3' \
  →-setcookie harmonia_cookie -sname 'xxx_node@netlab3'
2 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
3
4 Eshell V5.6.3 (abort with ^G)
5 (xxx_node@netlab3)1> [{name,xxx},
6   {root_node,netlab31@netlab3},
7   {root,netlab31},
8   {included_applications,[],},
9   {sname,xxx_node@netlab3},
10  {logfile_ext,".txt"},
11  {node_type,join},
12  {logdir,"log/"},
13  {logfile,"harmonia_log"}]
14 "log/harmonia_log_xxx.txt"
15 start Pid:[<0.253.0>]
16
17 (xxx_node@netlab3)1>
18 application            application_controller    application_master
19 application_starter     auth                    beam_lib
20 c                       code                    code_aux
21 code_server            crypto                  crypto_app
22 crypto_server          crypto_sup              dict
23 dist_util              edlin                  edlin_expand
24 erl_ddll               erl_distribution        erl_epmd
25 erl_eval               erlLint                erl_parse
26 erl_prim_loader        erl_scan               erlang
27 error_handler          error_logger            error_logger_file_h
28 error_logger_tty_h     ets                    file
29 file_io_server         file_server            filename
30 gb_sets                gb_trees               gen
31 gen_event              gen_fsm                gen_server
32 gen_tcp                global                 global_group
33 group                 harmonia               heart
34 hipe_unified_loader    hm_cache               hm_cache_mgr
35 hm_config              hm_config_if           hm_ds
36 hm_edge                hm_event_mgr           hm_log_h_file
37 hm_misc                hm_name_server         hm_router
38 hm_stabilizer          hm_sup                 hm_table
39 inet                   inet_config            inet_db
40 inet_gethost_native    inet_parse             inet_tcp
41 inet_tcp_dist          inet_udp               init
42 io                     io_lib                 io_lib_format
43 io_lib_pretty          kernel                 kernel_config
44 lists                  net_kernel             orddict
45 ordsets                os                     otp_ring0
46 packages               prim_file              prim_inet
47 proc_lib               proplists              ram_file
48 random                 rpc                    sets
49 shell                  supervisor             supervisor_bridge
50 sys                    timer                  user_drv
51 user_sup
52 (xxx_node@netlab3)1> hm_cli:log_stop().
53 log stop:[dell5@dell] Result:[ok]
54 log stop:[dell8@dell] Result:[ok]

```

```

55 log stop:[dell7@dell] Result:[ok]
56 log stop:[dell6@dell] Result:[ok]
57 log stop:[dell1@dell] Result:[ok]
58 log stop:[dell3@dell] Result:[ok]
59 log stop:[dell4@dell] Result:[ok]
60 log stop:[dell2@dell] Result:[ok]
61 log stop:[netlab5@netlab] Result:[ok]
62 log stop:[netlab3@netlab] Result:[ok]
63 log stop:[netlab6@netlab] Result:[ok]
64 log stop:[netlab7@netlab] Result:[ok]
65 log stop:[netlab8@netlab] Result:[ok]
66 log stop:[netlab7@netlab] Result:[ok]
67 log stop:[netlab4@netlab] Result:[ok]
68 log stop:[netlab2@netlab] Result:[ok]
69 log stop:[netlab1@netlab] Result:[ok]
70 log stop:[netlab2@netlab] Result:[ok]
71 log stop:[netlab4@netlab] Result:[ok]
72 log stop:[netlab5@netlab] Result:[ok]
73 log stop:[netlab1@netlab] Result:[ok]
74 log stop:[netlab8@netlab] Result:[ok]
75 log stop:[netlab6@netlab] Result:[ok]
76 log stop:[netlab3@netlab] Result:[ok]
77 log stop:[netlab412@netlab4] Result:[ok]
78 log stop:[netlab46@netlab4] Result:[ok]
79 log stop:[netlab48@netlab4] Result:[ok]
80 log stop:[netlab42@netlab4] Result:[ok]
81 log stop:[netlab49@netlab4] Result:[ok]
82 log stop:[netlab418@netlab4] Result:[ok]
83 log stop:[netlab414@netlab4] Result:[ok]
84 log stop:[netlab420@netlab4] Result:[ok]
85 log stop:[netlab43@netlab4] Result:[ok]
86 log stop:[netlab410@netlab4] Result:[ok]
87 log stop:[netlab419@netlab4] Result:[ok]
88 log stop:[netlab411@netlab4] Result:[ok]
89 log stop:[netlab44@netlab4] Result:[ok]
90 log stop:[netlab415@netlab4] Result:[ok]
91 log stop:[netlab417@netlab4] Result:[ok]
92 log stop:[netlab416@netlab4] Result:[ok]
93 log stop:[netlab413@netlab4] Result:[ok]
94 log stop:[netlab47@netlab4] Result:[ok]
95 log stop:[netlab45@netlab4] Result:[ok]
96 log stop:[netlab41@netlab4] Result:[ok]
97 hm_log_h_file:closing 'log/harmonia_log_xxx.txt'
98 log stop:[xxx_node@netlab3] Result:[ok]
99 log stop:[netlab315@netlab3] Result:[ok]
100 log stop:[netlab318@netlab3] Result:[ok]
101 log stop:[netlab319@netlab3] Result:[ok]
102 log stop:[netlab317@netlab3] Result:[ok]
103 log stop:[netlab314@netlab3] Result:[ok]
104 log stop:[netlab36@netlab3] Result:[ok]
105 log stop:[netlab39@netlab3] Result:[ok]
106 log stop:[netlab38@netlab3] Result:[ok]
107 log stop:[netlab313@netlab3] Result:[ok]
108 log stop:[netlab35@netlab3] Result:[ok]
109 log stop:[netlab320@netlab3] Result:[ok]
110 log stop:[netlab312@netlab3] Result:[ok]
111 log stop:[netlab310@netlab3] Result:[ok]
112 log stop:[netlab316@netlab3] Result:[ok]
113 log stop:[netlab311@netlab3] Result:[ok]
114 log stop:[netlab37@netlab3] Result:[ok]

```

```

115 log stop:[netlab34@netlab3] Result:[ok]
116 log stop:[netlab33@netlab3] Result:[ok]
117 log stop:[netlab32@netlab3] Result:[ok]
118 log stop:[netlab31@netlab3] Result:[ok]
119 ok
120 (xxx_node@netlab3)2>
121 (xxx_node@netlab3)2>
122 (xxx_node@netlab3)2> hm_cli_test:store_short(10000).
123 ok
124 (xxx_node@netlab3)3> hm_cli_test:check_size().
125 ===== netlab31@netlab3 =====
126 id          name          type  size  mem  owner
127 -----
128 10          cookies          set    0   279   auth
129 13          code            set   278 12598  code_server
130 14          code_names       set    48  4649  code_server
131 ac_tab      ac_tab           set    20  1389  application_controller
132 file_io_servers file_io_servers set     1   347   file_server_2
133 global_locks global_locks     set     0   279   global_name_server
134 global_names global_names     set   391 11843  global_name_server
135 global_names_ext global_names_ext set     0   279   global_name_server
136 global_pid_ids global_pid_ids   bag     0   279   global_name_server
137 global_pid_names global_pid_names bag   782 11199  global_name_server
138 hm_ets_cache_table hm_ets_cache_table set    0   279   hm_cache_mgr
139 hm_table_global hm_table_global bag   520 36679  <0.52.0>
140 inet_cache    inet_cache       bag     0   279   inet_db
141 inet_db       inet_db          set    21   541   inet_db
142 inet_hosts    inet_hosts       set     0   279   inet_db
143 sys_dist      sys_dist         set    64  3047  net_kernel
144 ===== netlab32@netlab3 =====
145 id          name          type  size  mem  owner
146 -----
147 10          cookies          set    0   279   auth
148 13          code            set   279 12706  code_server
149 14          code_names       set    48  4649  code_server
150 15          ign_requests     set     0   279   inet_gethost_native
151 16          ign_req_index    set     0   279   inet_gethost_native
152 ac_tab      ac_tab           set    20  1389  application_controller
153 file_io_servers file_io_servers set     1   347   file_server_2
154 global_locks global_locks     set     0   279   global_name_server
155 global_names global_names     set   391 11851  global_name_server
156 global_names_ext global_names_ext set     0   279   global_name_server
157 global_pid_ids global_pid_ids   bag     0   279   global_name_server
158 global_pid_names global_pid_names bag   782 11203  global_name_server
159 hm_ets_cache_table hm_ets_cache_table set    0   279   hm_cache_mgr
160 hm_table_global hm_table_global bag   426 30099  <0.90.0>
161 inet_cache    inet_cache       bag     0   279   inet_db
162 inet_db       inet_db          set    21   541   inet_db
163 inet_hosts    inet_hosts       set     0   279   inet_db
164 sys_dist      sys_dist         set    64  3047  net_kernel
165 ===== netlab33@netlab3 =====
166 id          name          type  size  mem  owner
167 -----
168 10          cookies          set    0   279   auth
169 13          code            set   279 12706  code_server
170 14          code_names       set    48  4649  code_server
171 15          ign_requests     set     0   279   inet_gethost_native
172 16          ign_req_index    set     0   279   inet_gethost_native
173 ac_tab      ac_tab           set    20  1389  application_controller
174 file_io_servers file_io_servers set     1   347   file_server_2

```



```

175 global_locks      global_locks      set    0      279      global_name_server
176 global_names      global_names      set   391    11851    global_name_server
177 global_names_ext   global_names_ext   set    0      279      global_name_server
178 global_pid_ids     global_pid_ids     bag    0      279      global_name_server
179 global_pid_names   global_pid_names   bag   782    11203    global_name_server
180 hm_ets_cache_table hm_ets_cache_table set    0      279      hm_cache_mgr
181 hm_table_global    hm_table_global    bag   790    55579    <0.88.0>
182 inet_cache         inet_cache         bag    0      279      inet_db
183 inet_db            inet_db            set   21     541      inet_db
184 inet_hosts         inet_hosts         set    0      279      inet_db
185 sys_dist           sys_dist           set   64    3047     net_kernel
186 ===== netlab34@netlab3 =====
187 id                name                type  size  mem  owner
188 -----
189 10                 cookies             set    0    279   auth
190 13                 code                set   279  12706  code_server
191 14                 code_names          set   48   4649  code_server
192 15                 ign_requests        set    0    279   inet_gethost_native
193 16                 ign_req_index        set    0    279   inet_gethost_native
194 ac_tab             ac_tab              set   20   1389  application_controller
195 file_io_servers    file_io_servers     set    1    347   file_server_2
196 global_locks       global_locks         set    0    279   global_name_server
197 global_names       global_names         set   391  11851  global_name_server
198 global_names_ext   global_names_ext     set    0    279   global_name_server
199 global_pid_ids     global_pid_ids       bag    0    279   global_name_server
200 global_pid_names   global_pid_names     bag   782  11203  global_name_server
201 hm_ets_cache_table hm_ets_cache_table  set    0    279   hm_cache_mgr
202 hm_table_global    hm_table_global      bag  1026  72099  <0.89.0>
203 inet_cache         inet_cache           bag    0    279   inet_db
204 inet_db            inet_db              set   21   541   inet_db
205 inet_hosts         inet_hosts           set    0    279   inet_db
206 sys_dist           sys_dist             set   64   3047   net_kernel
207 ===== netlab35@netlab3 =====
208 id                name                type  size  mem  owner
209 -----
210 10                 cookies             set    0    279   auth
211 13                 code                set   279  12706  code_server
212 14                 code_names          set   48   4649  code_server
213 15                 ign_requests        set    0    279   inet_gethost_native
214 16                 ign_req_index        set    0    279   inet_gethost_native
215 ac_tab             ac_tab              set   20   1389  application_controller
216 file_io_servers    file_io_servers     set    1    347   file_server_2
217 global_locks       global_locks         set    0    279   global_name_server
218 global_names       global_names         set   391  11851  global_name_server
219 global_names_ext   global_names_ext     set    0    279   global_name_server
220 global_pid_ids     global_pid_ids       bag    0    279   global_name_server
221 global_pid_names   global_pid_names     bag   782  11203  global_name_server
222 hm_ets_cache_table hm_ets_cache_table  set    0    279   hm_cache_mgr
223 hm_table_global    hm_table_global      bag   768  54039  <0.129.0>
224 inet_cache         inet_cache           bag    0    279   inet_db
225 inet_db            inet_db              set   21   541   inet_db
226 inet_hosts         inet_hosts           set    0    279   inet_db
227 sys_dist           sys_dist             set   64   3047   net_kernel
228 ===== netlab36@netlab3 =====
229 id                name                type  size  mem  owner
230 -----
231 10                 cookies             set    0    279   auth
232 13                 code                set   279  12706  code_server
233 14                 code_names          set   48   4649  code_server
234 15                 ign_requests        set    0    279   inet_gethost_native

```

```

235 16          ign_req_index      set 0      279      inet_gethost_native
236 ac_tab      ac_tab             set 20     1389     application_controller
237 file_io_servers file_io_servers set 1      347     file_server_2
238 global_locks global_locks      set 0      279     global_name_server
239 global_names global_names      set 391    11851   global_name_server
240 global_names_ext global_names_ext set 0      279     global_name_server
241 global_pid_ids global_pid_ids    bag 0      279     global_name_server
242 global_pid_names global_pid_names bag 782     11203   global_name_server
243 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
244 hm_table_global hm_table_global bag 764     53759   <0.137.0>
245 inet_cache     inet_cache      bag 0      279     inet_db
246 inet_db        inet_db         set 21     541     inet_db
247 inet_hosts     inet_hosts      set 0      279     inet_db
248 sys_dist       sys_dist       set 64     3047    net_kernel
249 ===== netlab37@netlab3 =====
250 id             name                type  size    mem    owner
251 -----
252 10             cookies             set 0      279     auth
253 13             code               set 279    12706   code_server
254 14             code_names         set 48     4649    code_server
255 15             ign_requests       set 0      279     inet_gethost_native
256 16             ign_req_index      set 0      279     inet_gethost_native
257 ac_tab         ac_tab             set 20     1389    application_controller
258 file_io_servers file_io_servers    set 1      347     file_server_2
259 global_locks   global_locks       set 0      279     global_name_server
260 global_names   global_names       set 391    11851   global_name_server
261 global_names_ext global_names_ext   set 0      279     global_name_server
262 global_pid_ids global_pid_ids     bag 0      279     global_name_server
263 global_pid_names global_pid_names   bag 782     11203   global_name_server
264 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
265 hm_table_global hm_table_global   bag 1543   108289  <0.90.0>
266 inet_cache     inet_cache      bag 0      279     inet_db
267 inet_db        inet_db         set 21     541     inet_db
268 inet_hosts     inet_hosts      set 0      279     inet_db
269 sys_dist       sys_dist       set 64     3047    net_kernel
270 ===== netlab38@netlab3 =====
271 id             name                type  size    mem    owner
272 -----
273 10             cookies             set 0      279     auth
274 13             code               set 279    12706   code_server
275 14             code_names         set 48     4649    code_server
276 15             ign_requests       set 0      279     inet_gethost_native
277 16             ign_req_index      set 0      279     inet_gethost_native
278 ac_tab         ac_tab             set 20     1389    application_controller
279 file_io_servers file_io_servers    set 1      347     file_server_2
280 global_locks   global_locks       set 0      279     global_name_server
281 global_names   global_names       set 391    11851   global_name_server
282 global_names_ext global_names_ext   set 0      279     global_name_server
283 global_pid_ids global_pid_ids     bag 0      279     global_name_server
284 global_pid_names global_pid_names   bag 782     11203   global_name_server
285 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
286 hm_table_global hm_table_global   bag 1226   86099   <0.129.0>
287 inet_cache     inet_cache      bag 0      279     inet_db
288 inet_db        inet_db         set 21     541     inet_db
289 inet_hosts     inet_hosts      set 0      279     inet_db
290 sys_dist       sys_dist       set 64     3047    net_kernel
291 ===== netlab39@netlab3 =====
292 id             name                type  size    mem    owner
293 -----
294 10             cookies             set 0      279     auth

```

```

295 13          code          set 279 12706 code_server
296 14          code_names    set 48 4649 code_server
297 15          ign_requests  set 0 279 inet_gethost_native
298 16          ign_req_index set 0 279 inet_gethost_native
299 ac_tab      ac_tab        set 20 1389 application_controller
300 file_io_servers file_io_servers set 1 347 file_server_2
301 global_locks global_locks set 0 279 global_name_server
302 global_names global_names set 391 11851 global_name_server
303 global_names_ext global_names_ext set 0 279 global_name_server
304 global_pid_ids global_pid_ids bag 0 279 global_name_server
305 global_pid_names global_pid_names bag 782 11203 global_name_server
306 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
307 hm_table_global hm_table_global bag 682 48019 <0.139.0>
308 inet_cache    inet_cache    bag 0 279 inet_db
309 inet_db       inet_db       set 21 541 inet_db
310 inet_hosts    inet_hosts    set 0 279 inet_db
311 sys_dist      sys_dist      set 64 3047 net_kernel
312 ===== netlab310@netlab3 =====
313 id            name          type  size  mem  owner
314 -----
315 10            cookies          set 0 279 auth
316 13            code          set 279 12706 code_server
317 14            code_names    set 48 4649 code_server
318 15            ign_requests  set 0 279 inet_gethost_native
319 16            ign_req_index set 0 279 inet_gethost_native
320 ac_tab        ac_tab          set 20 1389 application_controller
321 file_io_servers file_io_servers set 1 349 file_server_2
322 global_locks  global_locks    set 0 279 global_name_server
323 global_names  global_names    set 391 11739 global_name_server
324 global_names_ext global_names_ext set 0 279 global_name_server
325 global_pid_ids global_pid_ids  bag 0 279 global_name_server
326 global_pid_names global_pid_names bag 782 11203 global_name_server
327 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
328 hm_table_global hm_table_global bag 449 31709 <0.159.0>
329 inet_cache    inet_cache    bag 0 279 inet_db
330 inet_db       inet_db       set 21 541 inet_db
331 inet_hosts    inet_hosts    set 0 279 inet_db
332 sys_dist      sys_dist      set 64 3047 net_kernel
333 ===== netlab311@netlab3 =====
334 id            name          type  size  mem  owner
335 -----
336 10            cookies          set 0 279 auth
337 13            code          set 279 12706 code_server
338 14            code_names    set 48 4649 code_server
339 15            ign_requests  set 0 279 inet_gethost_native
340 16            ign_req_index set 0 279 inet_gethost_native
341 ac_tab        ac_tab          set 20 1389 application_controller
342 file_io_servers file_io_servers set 1 349 file_server_2
343 global_locks  global_locks    set 0 279 global_name_server
344 global_names  global_names    set 391 11767 global_name_server
345 global_names_ext global_names_ext set 0 279 global_name_server
346 global_pid_ids global_pid_ids  bag 0 279 global_name_server
347 global_pid_names global_pid_names bag 782 11203 global_name_server
348 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
349 hm_table_global hm_table_global bag 1552 108919 <0.153.0>
350 inet_cache    inet_cache    bag 0 279 inet_db
351 inet_db       inet_db       set 21 541 inet_db
352 inet_hosts    inet_hosts    set 0 279 inet_db
353 sys_dist      sys_dist      set 64 3047 net_kernel
354 ===== netlab312@netlab3 =====

```

```

355 id          name          type  size  mem  owner
356 -----
357 10           cookies        set    0    279  auth
358 13           code          set   279  12706 code_server
359 14           code_names       set    48  4649  code_server
360 15           ign_requests   set     0  279  inet_gethost_native
361 16           ign_req_index   set     0  279  inet_gethost_native
362 ac_tab       ac_tab            set    20  1389  application_controller
363 file_io_servers file_io_servers set     1   349  file_server_2
364 global_locks global_locks      set     0  279  global_name_server
365 global_names global_names      set   391  11739 global_name_server
366 global_names_ext global_names_ext set     0   279  global_name_server
367 global_pid_ids global_pid_ids    bag     0  279  global_name_server
368 global_pid_names global_pid_names bag    782  11203 global_name_server
369 hm_ets_cache_table hm_ets_cache_table set     0   279  hm_cache_mgr
370 hm_table_global hm_table_global  bag   853  59989 <0.166.0>
371 inet_cache     inet_cache       bag     0  279  inet_db
372 inet_db        inet_db          set    21  541  inet_db
373 inet_hosts     inet_hosts       set     0  279  inet_db
374 sys_dist      sys_dist        set    64  3047 net_kernel
375 ===== netlab313@netlab3 =====
376 id          name          type  size  mem  owner
377 -----
378 10           cookies        set    0    279  auth
379 13           code          set   279  12706 code_server
380 14           code_names       set    48  4649  code_server
381 15           ign_requests   set     0  279  inet_gethost_native
382 16           ign_req_index   set     0  279  inet_gethost_native
383 ac_tab       ac_tab            set    20  1389  application_controller
384 file_io_servers file_io_servers set     1   349  file_server_2
385 global_locks global_locks      set     0  279  global_name_server
386 global_names global_names      set   391  11739 global_name_server
387 global_names_ext global_names_ext set     0   279  global_name_server
388 global_pid_ids global_pid_ids    bag     0  279  global_name_server
389 global_pid_names global_pid_names bag    782  11203 global_name_server
390 hm_ets_cache_table hm_ets_cache_table set     0   279  hm_cache_mgr
391 hm_table_global hm_table_global  bag   499  35209 <0.163.0>
392 inet_cache     inet_cache       bag     0  279  inet_db
393 inet_db        inet_db          set    21  541  inet_db
394 inet_hosts     inet_hosts       set     0  279  inet_db
395 sys_dist      sys_dist        set    64  3047 net_kernel
396 ===== netlab314@netlab3 =====
397 id          name          type  size  mem  owner
398 -----
399 10           cookies        set    0    279  auth
400 13           code          set   279  12706 code_server
401 14           code_names       set    48  4649  code_server
402 15           ign_requests   set     0  279  inet_gethost_native
403 16           ign_req_index   set     0  279  inet_gethost_native
404 ac_tab       ac_tab            set    20  1389  application_controller
405 file_io_servers file_io_servers set     1   349  file_server_2
406 global_locks global_locks      set     0  279  global_name_server
407 global_names global_names      set   391  11731 global_name_server
408 global_names_ext global_names_ext set     0   279  global_name_server
409 global_pid_ids global_pid_ids    bag     0  279  global_name_server
410 global_pid_names global_pid_names bag    782  11203 global_name_server
411 hm_ets_cache_table hm_ets_cache_table set     0   279  hm_cache_mgr
412 hm_table_global hm_table_global  bag   724  50959 <0.165.0>
413 inet_cache     inet_cache       bag     0  279  inet_db
414 inet_db        inet_db          set    21  541  inet_db

```

```

415 inet_hosts      inet_hosts      set  0      279      inet_db
416 sys_dist       sys_dist       set  64     3047     net_kernel
417 ===== netlab315@netlab3 =====
418 id              name              type  size  mem      owner
419 -----
420 10              cookies          set   0      279      auth
421 13              code             set  279    12706    code_server
422 14              code_names      set   48     4649     code_server
423 15              ign_requests    set   0      279      inet_gethost_native
424 16              ign_req_index   set   0      279      inet_gethost_native
425 ac_tab          ac_tab          set  20     1389     application_controller
426 file_io_servers file_io_servers set   1      349      file_server_2
427 global_locks    global_locks    set   0      279      global_name_server
428 global_names    global_names    set  391    11739    global_name_server
429 global_names_ext global_names_ext set   0      279      global_name_server
430 global_pid_ids  global_pid_ids  bag   0      279      global_name_server
431 global_pid_names global_pid_names bag  782    11203    global_name_server
432 hm_ets_cache_table hm_ets_cache_table set   0      279      hm_cache_mgr
433 hm_table_global hm_table_global bag  569    40109    <0.172.0>
434 inet_cache      inet_cache      bag   0      279      inet_db
435 inet_db         inet_db         set  21     541      inet_db
436 inet_hosts      inet_hosts      set   0      279      inet_db
437 sys_dist       sys_dist       set  64     3047     net_kernel
438 ===== netlab316@netlab3 =====
439 id              name              type  size  mem      owner
440 -----
441 10              cookies          set   0      279      auth
442 13              code             set  279    12706    code_server
443 14              code_names      set   48     4649     code_server
444 15              ign_requests    set   0      279      inet_gethost_native
445 16              ign_req_index   set   0      279      inet_gethost_native
446 ac_tab          ac_tab          set  20     1389     application_controller
447 file_io_servers file_io_servers set   1      349      file_server_2
448 global_locks    global_locks    set   0      279      global_name_server
449 global_names    global_names    set  391    11767    global_name_server
450 global_names_ext global_names_ext set   0      279      global_name_server
451 global_pid_ids  global_pid_ids  bag   0      279      global_name_server
452 global_pid_names global_pid_names bag  782    11203    global_name_server
453 hm_ets_cache_table hm_ets_cache_table set   0      279      hm_cache_mgr
454 hm_table_global hm_table_global bag  710    49979    <0.158.0>
455 inet_cache      inet_cache      bag   0      279      inet_db
456 inet_db         inet_db         set  21     541      inet_db
457 inet_hosts      inet_hosts      set   0      279      inet_db
458 sys_dist       sys_dist       set  64     3047     net_kernel
459 ===== netlab317@netlab3 =====
460 id              name              type  size  mem      owner
461 -----
462 10              cookies          set   0      279      auth
463 13              code             set  279    12706    code_server
464 14              code_names      set   48     4649     code_server
465 15              ign_requests    set   0      279      inet_gethost_native
466 16              ign_req_index   set   0      279      inet_gethost_native
467 ac_tab          ac_tab          set  20     1389     application_controller
468 file_io_servers file_io_servers set   1      349      file_server_2
469 global_locks    global_locks    set   0      279      global_name_server
470 global_names    global_names    set  391    11739    global_name_server
471 global_names_ext global_names_ext set   0      279      global_name_server
472 global_pid_ids  global_pid_ids  bag   0      279      global_name_server
473 global_pid_names global_pid_names bag  782    11203    global_name_server
474 hm_ets_cache_table hm_ets_cache_table set   0      279      hm_cache_mgr

```

```

475 hm_table_global hm_table_global bag 880 61879 <0.158.0>
476 inet_cache inet_cache bag 0 279 inet_db
477 inet_db inet_db set 21 541 inet_db
478 inet_hosts inet_hosts set 0 279 inet_db
479 sys_dist sys_dist set 64 3047 net_kernel
480 ===== netlab318@netlab3 =====
481 id name type size mem owner
482 -----
483 10 cookies set 0 279 auth
484 13 code set 279 12706 code_server
485 14 code_names set 48 4649 code_server
486 15 ign_requests set 0 279 inet_gethost_native
487 16 ign_req_index set 0 279 inet_gethost_native
488 ac_tab ac_tab set 20 1389 application_controller
489 file_io_servers file_io_servers set 1 349 file_server_2
490 global_locks global_locks set 0 279 global_name_server
491 global_names global_names set 391 11731 global_name_server
492 global_names_ext global_names_ext set 0 279 global_name_server
493 global_pid_ids global_pid_ids bag 0 279 global_name_server
494 global_pid_names global_pid_names bag 782 11203 global_name_server
495 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
496 hm_table_global hm_table_global bag 752 52919 <0.170.0>
497 inet_cache inet_cache bag 0 279 inet_db
498 inet_db inet_db set 21 541 inet_db
499 inet_hosts inet_hosts set 0 279 inet_db
500 sys_dist sys_dist set 64 3047 net_kernel
501 ===== netlab319@netlab3 =====
502 id name type size mem owner
503 -----
504 10 cookies set 0 279 auth
505 13 code set 279 12706 code_server
506 14 code_names set 48 4649 code_server
507 15 ign_requests set 0 279 inet_gethost_native
508 16 ign_req_index set 0 279 inet_gethost_native
509 ac_tab ac_tab set 20 1389 application_controller
510 file_io_servers file_io_servers set 1 349 file_server_2
511 global_locks global_locks set 0 279 global_name_server
512 global_names global_names set 391 11739 global_name_server
513 global_names_ext global_names_ext set 0 279 global_name_server
514 global_pid_ids global_pid_ids bag 0 279 global_name_server
515 global_pid_names global_pid_names bag 782 11203 global_name_server
516 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
517 hm_table_global hm_table_global bag 472 33319 <0.160.0>
518 inet_cache inet_cache bag 0 279 inet_db
519 inet_db inet_db set 21 541 inet_db
520 inet_hosts inet_hosts set 0 279 inet_db
521 sys_dist sys_dist set 64 3047 net_kernel
522 ===== netlab320@netlab3 =====
523 id name type size mem owner
524 -----
525 10 cookies set 0 279 auth
526 13 code set 279 12706 code_server
527 14 code_names set 48 4649 code_server
528 15 ign_requests set 0 279 inet_gethost_native
529 16 ign_req_index set 0 279 inet_gethost_native
530 ac_tab ac_tab set 20 1389 application_controller
531 file_io_servers file_io_servers set 1 349 file_server_2
532 global_locks global_locks set 0 279 global_name_server
533 global_names global_names set 391 11739 global_name_server
534 global_names_ext global_names_ext set 0 279 global_name_server

```

```

535 global_pid_ids global_pid_ids bag 0 279 global_name_server
536 global_pid_names global_pid_names bag 782 11203 global_name_server
537 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
538 hm_table_global hm_table_global bag 384 27159 <0.161.0>
539 inet_cache inet_cache bag 0 279 inet_db
540 inet_db inet_db set 21 541 inet_db
541 inet_hosts inet_hosts set 0 279 inet_db
542 sys_dist sys_dist set 64 3047 net_kernel
543 ===== netlab41@netlab4 =====
544 id name type size mem owner
545 -----
546 10 cookies set 0 279 auth
547 13 code set 279 12706 code_server
548 14 code_names set 48 4649 code_server
549 15 ign_requests set 0 279 inet_gethost_native
550 16 ign_req_index set 0 279 inet_gethost_native
551 ac_tab ac_tab set 20 1389 application_controller
552 file_io_servers file_io_servers set 1 347 file_server_2
553 global_locks global_locks set 0 279 global_name_server
554 global_names global_names set 391 11371 global_name_server
555 global_names_ext global_names_ext set 0 279 global_name_server
556 global_pid_ids global_pid_ids bag 0 279 global_name_server
557 global_pid_names global_pid_names bag 782 11203 global_name_server
558 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
559 hm_table_global hm_table_global bag 892 62719 <0.305.0>
560 inet_cache inet_cache bag 0 279 inet_db
561 inet_db inet_db set 21 541 inet_db
562 inet_hosts inet_hosts set 0 279 inet_db
563 sys_dist sys_dist set 64 3047 net_kernel
564 ===== netlab42@netlab4 =====
565 id name type size mem owner
566 -----
567 10 cookies set 0 279 auth
568 13 code set 279 12706 code_server
569 14 code_names set 48 4649 code_server
570 15 ign_requests set 0 279 inet_gethost_native
571 16 ign_req_index set 0 279 inet_gethost_native
572 ac_tab ac_tab set 20 1389 application_controller
573 file_io_servers file_io_servers set 1 347 file_server_2
574 global_locks global_locks set 0 279 global_name_server
575 global_names global_names set 391 11371 global_name_server
576 global_names_ext global_names_ext set 0 279 global_name_server
577 global_pid_ids global_pid_ids bag 0 279 global_name_server
578 global_pid_names global_pid_names bag 782 11203 global_name_server
579 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
580 hm_table_global hm_table_global bag 462 32619 <0.354.0>
581 inet_cache inet_cache bag 0 279 inet_db
582 inet_db inet_db set 21 541 inet_db
583 inet_hosts inet_hosts set 0 279 inet_db
584 sys_dist sys_dist set 64 3047 net_kernel
585 ===== netlab43@netlab4 =====
586 id name type size mem owner
587 -----
588 10 cookies set 0 279 auth
589 13 code set 279 12706 code_server
590 14 code_names set 48 4649 code_server
591 15 ign_requests set 0 279 inet_gethost_native
592 16 ign_req_index set 0 279 inet_gethost_native
593 ac_tab ac_tab set 20 1389 application_controller
594 file_io_servers file_io_servers set 1 347 file_server_2

```

```

595 global_locks      global_locks      set    0      279      global_name_server
596 global_names      global_names      set   391    11371    global_name_server
597 global_names_ext   global_names_ext   set    0      279      global_name_server
598 global_pid_ids     global_pid_ids     bag    0      279      global_name_server
599 global_pid_names   global_pid_names   bag   782    11203    global_name_server
600 hm_ets_cache_table hm_ets_cache_table set    0      279      hm_cache_mgr
601 hm_table_global    hm_table_global    bag   395    27929    <0.351.0>
602 inet_cache         inet_cache         bag    0      279      inet_db
603 inet_db            inet_db            set   21     541      inet_db
604 inet_hosts         inet_hosts         set    0      279      inet_db
605 sys_dist           sys_dist           set   64    3047     net_kernel
606 ===== netlab44@netlab4 =====
607 id                 name                 type  size  mem  owner
608 -----
609 10                  cookies              set    0    279   auth
610 13                  code                 set   279   12706  code_server
611 14                  code_names           set   48    4649   code_server
612 15                  ign_requests         set    0    279   inet_gethost_native
613 16                  ign_req_index        set    0    279   inet_gethost_native
614 ac_tab             ac_tab               set   20    1389   application_controller
615 file_io_servers     file_io_servers      set    1    347   file_server_2
616 global_locks        global_locks         set    0    279   global_name_server
617 global_names        global_names         set   391   11371  global_name_server
618 global_names_ext    global_names_ext     set    0    279   global_name_server
619 global_pid_ids      global_pid_ids       bag    0    279   global_name_server
620 global_pid_names    global_pid_names     bag   782    11203  global_name_server
621 hm_ets_cache_table  hm_ets_cache_table  set    0    279   hm_cache_mgr
622 hm_table_global     hm_table_global      bag  1054   74059  <0.348.0>
623 inet_cache          inet_cache           bag    0    279   inet_db
624 inet_db             inet_db              set   21     541   inet_db
625 inet_hosts          inet_hosts           set    0    279   inet_db
626 sys_dist            sys_dist             set   64    3047   net_kernel
627 ===== netlab45@netlab4 =====
628 id                 name                 type  size  mem  owner
629 -----
630 10                  cookies              set    0    279   auth
631 13                  code                 set   279   12706  code_server
632 14                  code_names           set   48    4649   code_server
633 15                  ign_requests         set    0    279   inet_gethost_native
634 16                  ign_req_index        set    0    279   inet_gethost_native
635 ac_tab             ac_tab               set   20    1389   application_controller
636 file_io_servers     file_io_servers      set    1    347   file_server_2
637 global_locks        global_locks         set    0    279   global_name_server
638 global_names        global_names         set   391   11371  global_name_server
639 global_names_ext    global_names_ext     set    0    279   global_name_server
640 global_pid_ids      global_pid_ids       bag    0    279   global_name_server
641 global_pid_names    global_pid_names     bag   782    11203  global_name_server
642 hm_ets_cache_table  hm_ets_cache_table  set    0    279   hm_cache_mgr
643 hm_table_global     hm_table_global      bag   640   45079  <0.318.0>
644 inet_cache          inet_cache           bag    0    279   inet_db
645 inet_db             inet_db              set   21     541   inet_db
646 inet_hosts          inet_hosts           set    0    279   inet_db
647 sys_dist            sys_dist             set   64    3047   net_kernel
648 ===== netlab46@netlab4 =====
649 id                 name                 type  size  mem  owner
650 -----
651 10                  cookies              set    0    279   auth
652 13                  code                 set   279   12706  code_server
653 14                  code_names           set   48    4649   code_server
654 15                  ign_requests         set    0    279   inet_gethost_native

```



```

655 16          ign_req_index      set 0      279      inet_gethost_native
656 ac_tab      ac_tab             set 20     1389     application_controller
657 file_io_servers file_io_servers set 1      347     file_server_2
658 global_locks global_locks      set 0      279     global_name_server
659 global_names global_names      set 391    11371   global_name_server
660 global_names_ext global_names_ext set 0      279     global_name_server
661 global_pid_ids global_pid_ids    bag 0      279     global_name_server
662 global_pid_names global_pid_names bag 782    11203   global_name_server
663 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
664 hm_table_global hm_table_global bag 710    49979   <0.338.0>
665 inet_cache     inet_cache         bag 0      279     inet_db
666 inet_db        inet_db           set 21     541     inet_db
667 inet_hosts     inet_hosts         set 0      279     inet_db
668 sys_dist       sys_dist         set 64     3047    net_kernel
669 ===== netlab47@netlab4 =====
670 id             name                type  size    mem    owner
671 -----
672 10             cookies             set 0      279     auth
673 13             code               set 279    12706   code_server
674 14             code_names         set 48     4649    code_server
675 15             ign_requests       set 0      279     inet_gethost_native
676 16             ign_req_index      set 0      279     inet_gethost_native
677 ac_tab         ac_tab             set 20     1389     application_controller
678 file_io_servers file_io_servers    set 1      347     file_server_2
679 global_locks   global_locks       set 0      279     global_name_server
680 global_names   global_names       set 391    11371   global_name_server
681 global_names_ext global_names_ext   set 0      279     global_name_server
682 global_pid_ids global_pid_ids     bag 0      279     global_name_server
683 global_pid_names global_pid_names   bag 782    11203   global_name_server
684 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
685 hm_table_global hm_table_global   bag 441    31149   <0.326.0>
686 inet_cache     inet_cache         bag 0      279     inet_db
687 inet_db        inet_db           set 21     541     inet_db
688 inet_hosts     inet_hosts         set 0      279     inet_db
689 sys_dist       sys_dist         set 64     3047    net_kernel
690 ===== netlab48@netlab4 =====
691 id             name                type  size    mem    owner
692 -----
693 10             cookies             set 0      279     auth
694 13             code               set 279    12706   code_server
695 14             code_names         set 48     4649    code_server
696 15             ign_requests       set 0      279     inet_gethost_native
697 16             ign_req_index      set 0      279     inet_gethost_native
698 ac_tab         ac_tab             set 20     1389     application_controller
699 file_io_servers file_io_servers    set 1      347     file_server_2
700 global_locks   global_locks       set 0      279     global_name_server
701 global_names   global_names       set 391    11371   global_name_server
702 global_names_ext global_names_ext   set 0      279     global_name_server
703 global_pid_ids global_pid_ids     bag 0      279     global_name_server
704 global_pid_names global_pid_names   bag 782    11203   global_name_server
705 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
706 hm_table_global hm_table_global   bag 752    52919   <0.338.0>
707 inet_cache     inet_cache         bag 0      279     inet_db
708 inet_db        inet_db           set 21     541     inet_db
709 inet_hosts     inet_hosts         set 0      279     inet_db
710 sys_dist       sys_dist         set 64     3047    net_kernel
711 ===== netlab49@netlab4 =====
712 id             name                type  size    mem    owner
713 -----
714 10             cookies             set 0      279     auth

```

```

715 13          code          set 279 12706 code_server
716 14          code_names    set 48 4649 code_server
717 15          ign_requests   set 0 279 inet_gethost_native
718 16          ign_req_index  set 0 279 inet_gethost_native
719 ac_tab      ac_tab        set 20 1389 application_controller
720 file_io_servers file_io_servers set 1 347 file_server_2
721 global_locks global_locks set 0 279 global_name_server
722 global_names global_names set 391 11271 global_name_server
723 global_names_ext global_names_ext set 0 279 global_name_server
724 global_pid_ids global_pid_ids bag 0 279 global_name_server
725 global_pid_names global_pid_names bag 782 11203 global_name_server
726 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
727 hm_table_global hm_table_global bag 976 68599 <0.395.0>
728 inet_cache    inet_cache    bag 0 279 inet_db
729 inet_db       inet_db       set 21 541 inet_db
730 inet_hosts    inet_hosts    set 0 279 inet_db
731 sys_dist     sys_dist     set 64 3047 net_kernel
732 ===== netlab410@netlab4 =====
733 id           name          type  size  mem  owner
734 -----
735 10           cookies          set 0 279 auth
736 13           code          set 279 12706 code_server
737 14           code_names    set 48 4649 code_server
738 15           ign_requests   set 0 279 inet_gethost_native
739 16           ign_req_index  set 0 279 inet_gethost_native
740 ac_tab      ac_tab        set 20 1389 application_controller
741 file_io_servers file_io_servers set 1 349 file_server_2
742 global_locks global_locks set 0 279 global_name_server
743 global_names global_names set 391 11371 global_name_server
744 global_names_ext global_names_ext set 0 279 global_name_server
745 global_pid_ids global_pid_ids bag 0 279 global_name_server
746 global_pid_names global_pid_names bag 782 11203 global_name_server
747 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
748 hm_table_global hm_table_global bag 847 59569 <0.366.0>
749 inet_cache    inet_cache    bag 0 279 inet_db
750 inet_db       inet_db       set 21 541 inet_db
751 inet_hosts    inet_hosts    set 0 279 inet_db
752 sys_dist     sys_dist     set 64 3047 net_kernel
753 ===== netlab411@netlab4 =====
754 id           name          type  size  mem  owner
755 -----
756 10           cookies          set 0 279 auth
757 13           code          set 279 12706 code_server
758 14           code_names    set 48 4649 code_server
759 15           ign_requests   set 0 279 inet_gethost_native
760 16           ign_req_index  set 0 279 inet_gethost_native
761 ac_tab      ac_tab        set 20 1389 application_controller
762 file_io_servers file_io_servers set 1 349 file_server_2
763 global_locks global_locks set 0 279 global_name_server
764 global_names global_names set 391 11347 global_name_server
765 global_names_ext global_names_ext set 0 279 global_name_server
766 global_pid_ids global_pid_ids bag 0 279 global_name_server
767 global_pid_names global_pid_names bag 782 11203 global_name_server
768 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
769 hm_table_global hm_table_global bag 795 55929 <0.358.0>
770 inet_cache    inet_cache    bag 0 279 inet_db
771 inet_db       inet_db       set 21 541 inet_db
772 inet_hosts    inet_hosts    set 0 279 inet_db
773 sys_dist     sys_dist     set 64 3047 net_kernel
774 ===== netlab412@netlab4 =====

```

```

775 id          name          type  size  mem  owner
776 -----
777 10           cookies       set   0    279  auth
778 13           code         set  279  12706 code_server
779 14           code_names    set   48  4649  code_server
780 15           ign_requests  set   0    279  inet_gethost_native
781 16           ign_req_index set   0    279  inet_gethost_native
782 ac_tab       ac_tab         set  20   1389  application_controller
783 file_io_servers file_io_servers set   1    349  file_server_2
784 global_locks global_locks    set   0    279  global_name_server
785 global_names global_names    set  391  11295  global_name_server
786 global_names_ext global_names_ext set   0    279  global_name_server
787 global_pid_ids global_pid_ids  bag   0    279  global_name_server
788 global_pid_names global_pid_names bag  782  11203  global_name_server
789 hm_ets_cache_table hm_ets_cache_table set  0    279  hm_cache_mgr
790 hm_table_global hm_table_global bag  881  61949  <0.369.0>
791 inet_cache    inet_cache     bag   0    279  inet_db
792 inet_db       inet_db       set  21   541  inet_db
793 inet_hosts    inet_hosts    set   0    279  inet_db
794 sys_dist     sys_dist     set   64  3047  net_kernel
795 ===== netlab413@netlab4 =====
796 id          name          type  size  mem  owner
797 -----
798 10           cookies       set   0    279  auth
799 13           code         set  279  12706 code_server
800 14           code_names    set   48  4649  code_server
801 15           ign_requests  set   0    279  inet_gethost_native
802 16           ign_req_index set   0    279  inet_gethost_native
803 ac_tab       ac_tab         set  20   1389  application_controller
804 file_io_servers file_io_servers set   1    349  file_server_2
805 global_locks global_locks    set   0    279  global_name_server
806 global_names global_names    set  391  11271  global_name_server
807 global_names_ext global_names_ext set   0    279  global_name_server
808 global_pid_ids global_pid_ids  bag   0    279  global_name_server
809 global_pid_names global_pid_names bag  782  11203  global_name_server
810 hm_ets_cache_table hm_ets_cache_table set  0    279  hm_cache_mgr
811 hm_table_global hm_table_global bag  420  29679  <0.386.0>
812 inet_cache    inet_cache     bag   0    279  inet_db
813 inet_db       inet_db       set  21   541  inet_db
814 inet_hosts    inet_hosts    set   0    279  inet_db
815 sys_dist     sys_dist     set   64  3047  net_kernel
816 ===== netlab414@netlab4 =====
817 id          name          type  size  mem  owner
818 -----
819 10           cookies       set   0    279  auth
820 13           code         set  279  12706 code_server
821 14           code_names    set   48  4649  code_server
822 15           ign_requests  set   0    279  inet_gethost_native
823 16           ign_req_index set   0    279  inet_gethost_native
824 ac_tab       ac_tab         set  20   1389  application_controller
825 file_io_servers file_io_servers set   1    349  file_server_2
826 global_locks global_locks    set   0    279  global_name_server
827 global_names global_names    set  391  11319  global_name_server
828 global_names_ext global_names_ext set   0    279  global_name_server
829 global_pid_ids global_pid_ids  bag   0    279  global_name_server
830 global_pid_names global_pid_names bag  782  11203  global_name_server
831 hm_ets_cache_table hm_ets_cache_table set  0    279  hm_cache_mgr
832 hm_table_global hm_table_global bag 1187  83369  <0.375.0>
833 inet_cache    inet_cache     bag   0    279  inet_db
834 inet_db       inet_db       set  21   541  inet_db

```

```

835 inet_hosts      inet_hosts      set  0      279      inet_db
836 sys_dist        sys_dist        set  64     3047     net_kernel
837 ===== netlab415@netlab4 =====
838 id              name              type  size  mem      owner
839 -----
840 10              cookies           set   0      279      auth
841 13              code              set  279    12706    code_server
842 14              code_names        set   48     4649     code_server
843 15              ign_requests      set   0      279      inet_gethost_native
844 16              ign_req_index     set   0      279      inet_gethost_native
845 ac_tab          ac_tab            set  20     1389     application_controller
846 file_io_servers file_io_servers    set   1      349      file_server_2
847 global_locks    global_locks       set   0      279      global_name_server
848 global_names     global_names       set  391    11307    global_name_server
849 global_names_ext global_names_ext    set   0      279      global_name_server
850 global_pid_ids   global_pid_ids     bag   0      279      global_name_server
851 global_pid_names global_pid_names    bag  782    11203    global_name_server
852 hm_ets_cache_table hm_ets_cache_table set   0      279      hm_cache_mgr
853 hm_table_global hm_table_global    bag  644    45359    <0.378.0>
854 inet_cache       inet_cache         bag   0      279      inet_db
855 inet_db          inet_db            set  21     541      inet_db
856 inet_hosts       inet_hosts         set   0      279      inet_db
857 sys_dist        sys_dist          set  64     3047     net_kernel
858 ===== netlab416@netlab4 =====
859 id              name              type  size  mem      owner
860 -----
861 10              cookies           set   0      279      auth
862 13              code              set  279    12706    code_server
863 14              code_names        set   48     4649     code_server
864 15              ign_requests      set   0      279      inet_gethost_native
865 16              ign_req_index     set   0      279      inet_gethost_native
866 ac_tab          ac_tab            set  20     1389     application_controller
867 file_io_servers file_io_servers    set   1      349      file_server_2
868 global_locks    global_locks       set   0      279      global_name_server
869 global_names     global_names       set  391    11271    global_name_server
870 global_names_ext global_names_ext    set   0      279      global_name_server
871 global_pid_ids   global_pid_ids     bag   0      279      global_name_server
872 global_pid_names global_pid_names    bag  782    11203    global_name_server
873 hm_ets_cache_table hm_ets_cache_table set   0      279      hm_cache_mgr
874 hm_table_global hm_table_global    bag  901    63349    <0.382.0>
875 inet_cache       inet_cache         bag   0      279      inet_db
876 inet_db          inet_db            set  21     541      inet_db
877 inet_hosts       inet_hosts         set   0      279      inet_db
878 sys_dist        sys_dist          set  64     3047     net_kernel
879 ===== netlab417@netlab4 =====
880 id              name              type  size  mem      owner
881 -----
882 10              cookies           set   0      279      auth
883 13              code              set  279    12706    code_server
884 14              code_names        set   48     4649     code_server
885 15              ign_requests      set   0      279      inet_gethost_native
886 16              ign_req_index     set   0      279      inet_gethost_native
887 ac_tab          ac_tab            set  20     1389     application_controller
888 file_io_servers file_io_servers    set   1      349      file_server_2
889 global_locks    global_locks       set   0      279      global_name_server
890 global_names     global_names       set  391    11271    global_name_server
891 global_names_ext global_names_ext    set   0      279      global_name_server
892 global_pid_ids   global_pid_ids     bag   0      279      global_name_server
893 global_pid_names global_pid_names    bag  782    11203    global_name_server
894 hm_ets_cache_table hm_ets_cache_table set   0      279      hm_cache_mgr

```

```

895 hm_table_global hm_table_global bag 1474 103459 <0.390.0>
896 inet_cache inet_cache bag 0 279 inet_db
897 inet_db inet_db set 21 541 inet_db
898 inet_hosts inet_hosts set 0 279 inet_db
899 sys_dist sys_dist set 64 3047 net_kernel
900 ===== netlab418@netlab4 =====
901 id name type size mem owner
902 -----
903 10 cookies set 0 279 auth
904 13 code set 279 12706 code_server
905 14 code_names set 48 4649 code_server
906 15 ign_requests set 0 279 inet_gethost_native
907 16 ign_req_index set 0 279 inet_gethost_native
908 ac_tab ac_tab set 20 1389 application_controller
909 file_io_servers file_io_servers set 1 349 file_server_2
910 global_locks global_locks set 0 279 global_name_server
911 global_names global_names set 391 11271 global_name_server
912 global_names_ext global_names_ext set 0 279 global_name_server
913 global_pid_ids global_pid_ids bag 0 279 global_name_server
914 global_pid_names global_pid_names bag 782 11203 global_name_server
915 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
916 hm_table_global hm_table_global bag 401 28349 <0.386.0>
917 inet_cache inet_cache bag 0 279 inet_db
918 inet_db inet_db set 21 541 inet_db
919 inet_hosts inet_hosts set 0 279 inet_db
920 sys_dist sys_dist set 64 3047 net_kernel
921 ===== netlab419@netlab4 =====
922 id name type size mem owner
923 -----
924 10 cookies set 0 279 auth
925 13 code set 279 12706 code_server
926 14 code_names set 48 4649 code_server
927 15 ign_requests set 0 279 inet_gethost_native
928 16 ign_req_index set 0 279 inet_gethost_native
929 ac_tab ac_tab set 20 1389 application_controller
930 file_io_servers file_io_servers set 1 349 file_server_2
931 global_locks global_locks set 0 279 global_name_server
932 global_names global_names set 391 11271 global_name_server
933 global_names_ext global_names_ext set 0 279 global_name_server
934 global_pid_ids global_pid_ids bag 0 279 global_name_server
935 global_pid_names global_pid_names bag 782 11203 global_name_server
936 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
937 hm_table_global hm_table_global bag 1234 86659 <0.390.0>
938 inet_cache inet_cache bag 0 279 inet_db
939 inet_db inet_db set 21 541 inet_db
940 inet_hosts inet_hosts set 0 279 inet_db
941 sys_dist sys_dist set 64 3047 net_kernel
942 ===== netlab420@netlab4 =====
943 id name type size mem owner
944 -----
945 10 cookies set 0 279 auth
946 13 code set 279 12706 code_server
947 14 code_names set 48 4649 code_server
948 15 ign_requests set 0 279 inet_gethost_native
949 16 ign_req_index set 0 279 inet_gethost_native
950 ac_tab ac_tab set 20 1389 application_controller
951 file_io_servers file_io_servers set 1 349 file_server_2
952 global_locks global_locks set 0 279 global_name_server
953 global_names global_names set 391 11271 global_name_server
954 global_names_ext global_names_ext set 0 279 global_name_server

```

```

955 global_pid_ids global_pid_ids bag 0 279 global_name_server
956 global_pid_names global_pid_names bag 782 11203 global_name_server
957 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
958 hm_table_global hm_table_global bag 833 58589 <0.383.0>
959 inet_cache inet_cache bag 0 279 inet_db
960 inet_db inet_db set 21 541 inet_db
961 inet_hosts inet_hosts set 0 279 inet_db
962 sys_dist sys_dist set 64 3047 net_kernel
963 ===== dell1@ dell =====
964 id name type size mem owner
965 -----
966 12 cookies set 0 286 auth
967 4111 code set 281 12978 code_server
968 8208 code_names set 56 7510 code_server
969 12305 ign_requests set 0 286 inet_gethost_native
970 16402 ign_req_index set 0 286 inet_gethost_native
971 ac_tab ac_tab set 20 1328 application_controller
972 file_io_servers file_io_servers set 1 339 file_server_2
973 global_locks global_locks set 0 286 global_name_server
974 global_names global_names set 391 10161 global_name_server
975 global_names_ext global_names_ext set 0 286 global_name_server
976 global_pid_ids global_pid_ids bag 0 286 global_name_server
977 global_pid_names global_pid_names bag 782 8864 global_name_server
978 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
979 hm_table_global hm_table_global bag 333 22597 <0.456.0>
980 inet_cache inet_cache bag 0 286 inet_db
981 inet_db inet_db set 29 553 inet_db
982 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
983 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
984 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
985 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
986 sys_dist sys_dist set 64 2868 net_kernel
987 ===== dell2@ dell =====
988 id name type size mem owner
989 -----
990 12 cookies set 0 286 auth
991 4111 code set 281 12978 code_server
992 8208 code_names set 56 7510 code_server
993 12305 ign_requests set 0 286 inet_gethost_native
994 16402 ign_req_index set 0 286 inet_gethost_native
995 ac_tab ac_tab set 20 1328 application_controller
996 file_io_servers file_io_servers set 1 339 file_server_2
997 global_locks global_locks set 0 286 global_name_server
998 global_names global_names set 391 10509 global_name_server
999 global_names_ext global_names_ext set 0 286 global_name_server
1000 global_pid_ids global_pid_ids bag 0 286 global_name_server
1001 global_pid_names global_pid_names bag 782 8864 global_name_server
1002 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1003 hm_table_global hm_table_global bag 466 31508 <0.356.0>
1004 inet_cache inet_cache bag 0 286 inet_db
1005 inet_db inet_db set 29 553 inet_db
1006 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1007 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1008 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1009 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1010 sys_dist sys_dist set 64 2868 net_kernel
1011 ===== dell3@ dell =====
1012 id name type size mem owner
1013 -----
1014 12 cookies set 0 286 auth

```

```

1015 4111          code          set 281 12978 code_server
1016 8208          code_names     set 56 7510 code_server
1017 12305         ign_requests   set 0 286 inet_gethost_native
1018 16402         ign_req_index  set 0 286 inet_gethost_native
1019 ac_tab         ac_tab        set 20 1328 application_controller
1020 file_io_servers file_io_servers set 1 339 file_server_2
1021 global_locks   global_locks   set 0 286 global_name_server
1022 global_names   global_names   set 391 10109 global_name_server
1023 global_names_ext global_names_ext set 0 286 global_name_server
1024 global_pid_ids global_pid_ids bag 0 286 global_name_server
1025 global_pid_names global_pid_names bag 782 8864 global_name_server
1026 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1027 hm_table_global hm_table_global bag 507 34255 <0.469.0>
1028 inet_cache     inet_cache     bag 0 286 inet_db
1029 inet_db        inet_db        set 29 553 inet_db
1030 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1031 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1032 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1033 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1034 sys_dist       sys_dist       set 64 2868 net_kernel
1035 ===== dell4@dell =====
1036 id             name             type  size  mem  owner
1037 -----
1038 12             cookies          set 0 286 auth
1039 4111           code            set 281 12978 code_server
1040 8208           code_names     set 56 7510 code_server
1041 12305         ign_requests   set 0 286 inet_gethost_native
1042 16402         ign_req_index  set 0 286 inet_gethost_native
1043 ac_tab        ac_tab         set 20 1328 application_controller
1044 file_io_servers file_io_servers set 1 339 file_server_2
1045 global_locks   global_locks   set 0 286 global_name_server
1046 global_names   global_names   set 391 10461 global_name_server
1047 global_names_ext global_names_ext set 0 286 global_name_server
1048 global_pid_ids global_pid_ids bag 0 286 global_name_server
1049 global_pid_names global_pid_names bag 782 8864 global_name_server
1050 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1051 hm_table_global hm_table_global bag 1545 103801 <0.327.0>
1052 inet_cache     inet_cache     bag 0 286 inet_db
1053 inet_db        inet_db        set 29 553 inet_db
1054 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1055 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1056 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1057 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1058 sys_dist       sys_dist       set 64 2868 net_kernel
1059 ===== dell5@dell =====
1060 id             name             type  size  mem  owner
1061 -----
1062 12             cookies          set 0 286 auth
1063 4111           code            set 281 12978 code_server
1064 8208           code_names     set 56 7510 code_server
1065 12305         ign_requests   set 0 286 inet_gethost_native
1066 16402         ign_req_index  set 0 286 inet_gethost_native
1067 ac_tab        ac_tab         set 20 1328 application_controller
1068 file_io_servers file_io_servers set 1 339 file_server_2
1069 global_locks   global_locks   set 0 286 global_name_server
1070 global_names   global_names   set 391 10349 global_name_server
1071 global_names_ext global_names_ext set 0 286 global_name_server
1072 global_pid_ids global_pid_ids bag 0 286 global_name_server
1073 global_pid_names global_pid_names bag 782 8864 global_name_server
1074 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr

```

```

1075 hm_table_global hm_table_global bag 912 61390 <0.375.0>
1076 inet_cache inet_cache bag 0 286 inet_db
1077 inet_db inet_db set 29 553 inet_db
1078 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1079 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1080 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1081 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1082 sys_dist sys_dist set 64 2868 net_kernel
1083 ===== dell6@dell =====
1084 id name type size mem owner
1085 -----
1086 12 cookies set 0 286 auth
1087 4111 code set 281 12978 code_server
1088 8208 code_names set 56 7510 code_server
1089 12305 ign_requests set 0 286 inet_gethost_native
1090 16402 ign_req_index set 0 286 inet_gethost_native
1091 ac_tab ac_tab set 20 1328 application_controller
1092 file_io_servers file_io_servers set 1 339 file_server_2
1093 global_locks global_locks set 0 286 global_name_server
1094 global_names global_names set 391 10349 global_name_server
1095 global_names_ext global_names_ext set 0 286 global_name_server
1096 global_pid_ids global_pid_ids bag 0 286 global_name_server
1097 global_pid_names global_pid_names bag 782 8864 global_name_server
1098 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1099 hm_table_global hm_table_global bag 497 33585 <0.353.0>
1100 inet_cache inet_cache bag 0 286 inet_db
1101 inet_db inet_db set 29 553 inet_db
1102 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1103 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1104 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1105 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1106 sys_dist sys_dist set 64 2868 net_kernel
1107 ===== dell7@dell =====
1108 id name type size mem owner
1109 -----
1110 12 cookies set 0 286 auth
1111 4111 code set 281 12978 code_server
1112 8208 code_names set 56 7510 code_server
1113 12305 ign_requests set 0 286 inet_gethost_native
1114 16402 ign_req_index set 0 286 inet_gethost_native
1115 ac_tab ac_tab set 20 1328 application_controller
1116 file_io_servers file_io_servers set 1 339 file_server_2
1117 global_locks global_locks set 0 286 global_name_server
1118 global_names global_names set 391 10453 global_name_server
1119 global_names_ext global_names_ext set 0 286 global_name_server
1120 global_pid_ids global_pid_ids bag 0 286 global_name_server
1121 global_pid_names global_pid_names bag 782 8864 global_name_server
1122 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1123 hm_table_global hm_table_global bag 390 26416 <0.358.0>
1124 inet_cache inet_cache bag 0 286 inet_db
1125 inet_db inet_db set 29 553 inet_db
1126 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1127 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1128 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1129 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1130 sys_dist sys_dist set 64 2868 net_kernel
1131 ===== dell8@dell =====
1132 id name type size mem owner
1133 -----
1134 12 cookies set 0 286 auth

```



```

1135 4111          code          set 281 12978 code_server
1136 8208          code_names     set 56 7510 code_server
1137 12305         ign_requests    set 0 286 inet_gethost_native
1138 16402         ign_req_index   set 0 286 inet_gethost_native
1139 ac_tab        ac_tab         set 20 1328 application_controller
1140 file_io_servers file_io_servers set 1 339 file_server_2
1141 global_locks   global_locks   set 0 286 global_name_server
1142 global_names   global_names   set 391 10101 global_name_server
1143 global_names_ext global_names_ext set 0 286 global_name_server
1144 global_pid_ids global_pid_ids bag 0 286 global_name_server
1145 global_pid_names global_pid_names bag 782 8864 global_name_server
1146 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1147 hm_table_global hm_table_global bag 228 15562 <0.478.0>
1148 inet_cache     inet_cache     bag 0 286 inet_db
1149 inet_db        inet_db        set 29 553 inet_db
1150 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1151 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1152 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1153 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1154 sys_dist      sys_dist      set 64 2868 net_kernel
1155 ===== netlabal@netlaba =====
1156 id            name            type  size  mem  owner
1157 -----
1158 9             cookies          set 0 284 auth
1159 4108          code            set 278 12787 code_server
1160 8205          code_names     set 54 5196 code_server
1161 12302         ign_requests    set 0 284 inet_gethost_native
1162 16399         ign_req_index   set 0 284 inet_gethost_native
1163 ac_tab        ac_tab         set 20 1380 application_controller
1164 file_io_servers file_io_servers set 1 352 file_server_2
1165 global_locks   global_locks   set 0 284 global_name_server
1166 global_names   global_names   set 391 10896 global_name_server
1167 global_names_ext global_names_ext set 0 284 global_name_server
1168 global_pid_ids global_pid_ids bag 0 284 global_name_server
1169 global_pid_names global_pid_names bag 782 11208 global_name_server
1170 hm_ets_cache_table hm_ets_cache_table set 0 284 hm_cache_mgr
1171 hm_table_global hm_table_global bag 503 35494 <0.530.0>
1172 inet_cache     inet_cache     bag 0 284 inet_db
1173 inet_db        inet_db        set 21 512 inet_db
1174 inet_hosts     inet_hosts     set 0 284 inet_db
1175 sys_dist      sys_dist      set 64 3052 net_kernel
1176 ===== netlaba2@netlaba =====
1177 id            name            type  size  mem  owner
1178 -----
1179 9             cookies          set 0 284 auth
1180 4108          code            set 278 12787 code_server
1181 8205          code_names     set 54 5196 code_server
1182 12302         ign_requests    set 0 284 inet_gethost_native
1183 16399         ign_req_index   set 0 284 inet_gethost_native
1184 ac_tab        ac_tab         set 20 1380 application_controller
1185 file_io_servers file_io_servers set 1 352 file_server_2
1186 global_locks   global_locks   set 0 284 global_name_server
1187 global_names   global_names   set 391 10880 global_name_server
1188 global_names_ext global_names_ext set 0 284 global_name_server
1189 global_pid_ids global_pid_ids bag 0 284 global_name_server
1190 global_pid_names global_pid_names bag 782 11208 global_name_server
1191 hm_ets_cache_table hm_ets_cache_table set 0 284 hm_cache_mgr
1192 hm_table_global hm_table_global bag 1476 103604 <0.532.0>
1193 inet_cache     inet_cache     bag 0 284 inet_db
1194 inet_db        inet_db        set 21 512 inet_db

```

```

1195 inet_hosts      inet_hosts      set  0      284      inet_db
1196 sys_dist        sys_dist        set  64     3052     net_kernel
1197 ===== netlaba3@netlaba =====
1198 id              name              type  size    mem      owner
1199 -----
1200 9               cookies           set   0       284      auth
1201 4108            code             set   278     12787    code_server
1202 8205            code_names       set   54      5196     code_server
1203 12302           ign_requests     set   0       284      inet_gethost_native
1204 16399           ign_req_index    set   0       284      inet_gethost_native
1205 ac_tab          ac_tab           set   20      1380     application_controller
1206 file_io_servers file_io_servers  set   1       352      file_server_2
1207 global_locks    global_locks     set   0       284      global_name_server
1208 global_names    global_names     set   391     10904    global_name_server
1209 global_names_ext global_names_ext set   0       284      global_name_server
1210 global_pid_ids  global_pid_ids   bag   0       284      global_name_server
1211 global_pid_names global_pid_names bag   782     11208    global_name_server
1212 hm_ets_cache_table hm_ets_cache_table set  0       284      hm_cache_mgr
1213 hm_table_global hm_table_global  bag   623     43894    <0.522.0>
1214 inet_cache      inet_cache       bag   0       284      inet_db
1215 inet_db         inet_db          set   21      512      inet_db
1216 inet_hosts      inet_hosts      set   0       284      inet_db
1217 sys_dist        sys_dist        set   64     3052     net_kernel
1218 ===== netlaba4@netlaba =====
1219 id              name              type  size    mem      owner
1220 -----
1221 9               cookies           set   0       284      auth
1222 4108            code             set   278     12787    code_server
1223 8205            code_names       set   54      5196     code_server
1224 12302           ign_requests     set   0       284      inet_gethost_native
1225 16399           ign_req_index    set   0       284      inet_gethost_native
1226 ac_tab          ac_tab           set   20      1380     application_controller
1227 file_io_servers file_io_servers  set   1       352      file_server_2
1228 global_locks    global_locks     set   0       284      global_name_server
1229 global_names    global_names     set   391     10896    global_name_server
1230 global_names_ext global_names_ext set   0       284      global_name_server
1231 global_pid_ids  global_pid_ids   bag   0       284      global_name_server
1232 global_pid_names global_pid_names bag   782     11208    global_name_server
1233 hm_ets_cache_table hm_ets_cache_table set  0       284      hm_cache_mgr
1234 hm_table_global hm_table_global  bag   715     50334    <0.528.0>
1235 inet_cache      inet_cache       bag   0       284      inet_db
1236 inet_db         inet_db          set   21      512      inet_db
1237 inet_hosts      inet_hosts      set   0       284      inet_db
1238 sys_dist        sys_dist        set   64     3052     net_kernel
1239 ===== netlaba5@netlaba =====
1240 id              name              type  size    mem      owner
1241 -----
1242 9               cookies           set   0       284      auth
1243 4108            code             set   278     12787    code_server
1244 8205            code_names       set   54      5196     code_server
1245 12302           ign_requests     set   0       284      inet_gethost_native
1246 16399           ign_req_index    set   0       284      inet_gethost_native
1247 ac_tab          ac_tab           set   20      1380     application_controller
1248 file_io_servers file_io_servers  set   1       352      file_server_2
1249 global_locks    global_locks     set   0       284      global_name_server
1250 global_names    global_names     set   391     10848    global_name_server
1251 global_names_ext global_names_ext set   0       284      global_name_server
1252 global_pid_ids  global_pid_ids   bag   0       284      global_name_server
1253 global_pid_names global_pid_names bag   782     11208    global_name_server
1254 hm_ets_cache_table hm_ets_cache_table set  0       284      hm_cache_mgr

```

```

1255 hm_table_global hm_table_global bag 824 57964 <0.541.0>
1256 inet_cache inet_cache bag 0 284 inet_db
1257 inet_db inet_db set 21 512 inet_db
1258 inet_hosts inet_hosts set 0 284 inet_db
1259 sys_dist sys_dist set 64 3052 net_kernel
1260 ===== netlaba6@netlaba =====
1261 id name type size mem owner
1262 -----
1263 9 cookies set 0 284 auth
1264 4108 code set 278 12787 code_server
1265 8205 code_names set 54 5196 code_server
1266 12302 ign_requests set 0 284 inet_gethost_native
1267 16399 ign_req_index set 0 284 inet_gethost_native
1268 ac_tab ac_tab set 20 1380 application_controller
1269 file_io_servers file_io_servers set 1 352 file_server_2
1270 global_locks global_locks set 0 284 global_name_server
1271 global_names global_names set 391 11216 global_name_server
1272 global_names_ext global_names_ext set 0 284 global_name_server
1273 global_pid_ids global_pid_ids bag 0 284 global_name_server
1274 global_pid_names global_pid_names bag 782 11208 global_name_server
1275 hm_ets_cache_table hm_ets_cache_table set 0 284 hm_cache_mgr
1276 hm_table_global hm_table_global bag 824 57964 <0.400.0>
1277 inet_cache inet_cache bag 0 284 inet_db
1278 inet_db inet_db set 21 512 inet_db
1279 inet_hosts inet_hosts set 0 284 inet_db
1280 sys_dist sys_dist set 64 3052 net_kernel
1281 ===== netlaba7@netlaba =====
1282 id name type size mem owner
1283 -----
1284 9 cookies set 0 284 auth
1285 4108 code set 278 12787 code_server
1286 8205 code_names set 54 5196 code_server
1287 12302 ign_requests set 0 284 inet_gethost_native
1288 16399 ign_req_index set 0 284 inet_gethost_native
1289 ac_tab ac_tab set 20 1380 application_controller
1290 file_io_servers file_io_servers set 1 352 file_server_2
1291 global_locks global_locks set 0 284 global_name_server
1292 global_names global_names set 391 10892 global_name_server
1293 global_names_ext global_names_ext set 0 284 global_name_server
1294 global_pid_ids global_pid_ids bag 0 284 global_name_server
1295 global_pid_names global_pid_names bag 782 11208 global_name_server
1296 hm_ets_cache_table hm_ets_cache_table set 0 284 hm_cache_mgr
1297 hm_table_global hm_table_global bag 597 42074 <0.561.0>
1298 inet_cache inet_cache bag 0 284 inet_db
1299 inet_db inet_db set 21 512 inet_db
1300 inet_hosts inet_hosts set 0 284 inet_db
1301 sys_dist sys_dist set 64 3052 net_kernel
1302 ===== netlaba8@netlaba =====
1303 id name type size mem owner
1304 -----
1305 9 cookies set 0 284 auth
1306 4108 code set 278 12787 code_server
1307 8205 code_names set 54 5196 code_server
1308 12302 ign_requests set 0 284 inet_gethost_native
1309 16399 ign_req_index set 0 284 inet_gethost_native
1310 ac_tab ac_tab set 20 1380 application_controller
1311 file_io_servers file_io_servers set 1 352 file_server_2
1312 global_locks global_locks set 0 284 global_name_server
1313 global_names global_names set 391 10896 global_name_server
1314 global_names_ext global_names_ext set 0 284 global_name_server

```

```

1315 global_pid_ids global_pid_ids bag 0 284 global_name_server
1316 global_pid_names global_pid_names bag 782 11208 global_name_server
1317 hm_ets_cache_table hm_ets_cache_table set 0 284 hm_cache_mgr
1318 hm_table_global hm_table_global bag 551 38854 <0.526.0>
1319 inet_cache inet_cache bag 0 284 inet_db
1320 inet_db inet_db set 21 512 inet_db
1321 inet_hosts inet_hosts set 0 284 inet_db
1322 sys_dist sys_dist set 64 3052 net_kernel
1323 ===== netlabbl@netlabbb =====
1324 id name type size mem owner
1325 -----
1326 12 cookies set 0 286 auth
1327 4111 code set 281 12978 code_server
1328 8208 code_names set 56 7510 code_server
1329 12305 ign_requests set 0 286 inet_gethost_native
1330 16402 ign_req_index set 0 286 inet_gethost_native
1331 ac_tab ac_tab set 21 1339 application_controller
1332 file_io_servers file_io_servers set 1 351 file_server_2
1333 global_locks global_locks set 0 286 global_name_server
1334 global_names global_names set 391 9993 global_name_server
1335 global_names_ext global_names_ext set 0 286 global_name_server
1336 global_pid_ids global_pid_ids bag 0 286 global_name_server
1337 global_pid_names global_pid_names bag 782 8864 global_name_server
1338 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1339 hm_table_global hm_table_global bag 1255 84371 <0.449.0>
1340 inet_cache inet_cache bag 0 286 inet_db
1341 inet_db inet_db set 29 559 inet_db
1342 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1343 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1344 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1345 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1346 sys_dist sys_dist set 64 2862 net_kernel
1347 ===== netlabbb2@netlabbb =====
1348 id name type size mem owner
1349 -----
1350 12 cookies set 0 286 auth
1351 4111 code set 281 12978 code_server
1352 8208 code_names set 56 7510 code_server
1353 12305 ign_requests set 0 286 inet_gethost_native
1354 16402 ign_req_index set 0 286 inet_gethost_native
1355 ac_tab ac_tab set 21 1339 application_controller
1356 file_io_servers file_io_servers set 1 351 file_server_2
1357 global_locks global_locks set 0 286 global_name_server
1358 global_names global_names set 391 9997 global_name_server
1359 global_names_ext global_names_ext set 0 286 global_name_server
1360 global_pid_ids global_pid_ids bag 0 286 global_name_server
1361 global_pid_names global_pid_names bag 782 8864 global_name_server
1362 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1363 hm_table_global hm_table_global bag 510 34456 <0.459.0>
1364 inet_cache inet_cache bag 0 286 inet_db
1365 inet_db inet_db set 29 559 inet_db
1366 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1367 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1368 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1369 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1370 sys_dist sys_dist set 64 2862 net_kernel
1371 ===== netlabbb3@netlabbb =====
1372 id name type size mem owner
1373 -----
1374 12 cookies set 0 286 auth

```

```

1375 4111          code          set 281 12978 code_server
1376 8208          code_names     set 56 7510 code_server
1377 12305         ign_requests    set 0 286 inet_gethost_native
1378 16402         ign_req_index   set 0 286 inet_gethost_native
1379 ac_tab         ac_tab         set 21 1339 application_controller
1380 file_io_servers file_io_servers set 1 351 file_server_2
1381 global_locks   global_locks   set 0 286 global_name_server
1382 global_names   global_names   set 391 9597 global_name_server
1383 global_names_ext global_names_ext set 0 286 global_name_server
1384 global_pid_ids global_pid_ids bag 0 286 global_name_server
1385 global_pid_names global_pid_names bag 782 8864 global_name_server
1386 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1387 hm_table_global hm_table_global bag 797 53685 <0.615.0>
1388 inet_cache     inet_cache     bag 0 286 inet_db
1389 inet_db        inet_db        set 29 559 inet_db
1390 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1391 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1392 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1393 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1394 sys_dist       sys_dist       set 64 2862 net_kernel
1395 ===== netlabbb4@netlabbb =====
1396 id            name            type size mem owner
1397 -----
1398 12             cookies          set 0 286 auth
1399 4111           code          set 281 12978 code_server
1400 8208           code_names     set 56 7510 code_server
1401 12305          ign_requests    set 0 286 inet_gethost_native
1402 16402          ign_req_index   set 0 286 inet_gethost_native
1403 ac_tab         ac_tab         set 21 1339 application_controller
1404 file_io_servers file_io_servers set 1 351 file_server_2
1405 global_locks   global_locks   set 0 286 global_name_server
1406 global_names   global_names   set 391 9681 global_name_server
1407 global_names_ext global_names_ext set 0 286 global_name_server
1408 global_pid_ids global_pid_ids bag 0 286 global_name_server
1409 global_pid_names global_pid_names bag 782 8864 global_name_server
1410 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1411 hm_table_global hm_table_global bag 1017 68425 <0.588.0>
1412 inet_cache     inet_cache     bag 0 286 inet_db
1413 inet_db        inet_db        set 29 559 inet_db
1414 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1415 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1416 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1417 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1418 sys_dist       sys_dist       set 64 2862 net_kernel
1419 ===== netlabbb5@netlabbb =====
1420 id            name            type size mem owner
1421 -----
1422 12             cookies          set 0 286 auth
1423 4111           code          set 281 12978 code_server
1424 8208           code_names     set 56 7510 code_server
1425 12305          ign_requests    set 0 286 inet_gethost_native
1426 16402          ign_req_index   set 0 286 inet_gethost_native
1427 ac_tab         ac_tab         set 21 1339 application_controller
1428 file_io_servers file_io_servers set 1 351 file_server_2
1429 global_locks   global_locks   set 0 286 global_name_server
1430 global_names   global_names   set 391 9569 global_name_server
1431 global_names_ext global_names_ext set 0 286 global_name_server
1432 global_pid_ids global_pid_ids bag 0 286 global_name_server
1433 global_pid_names global_pid_names bag 782 8864 global_name_server
1434 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr

```

```

1435 hm_table_global hm_table_global bag 243 16567 <0.626.0>
1436 inet_cache inet_cache bag 0 286 inet_db
1437 inet_db inet_db set 29 559 inet_db
1438 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1439 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1440 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1441 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1442 sys_dist sys_dist set 64 2862 net_kernel
1443 ===== netlab6@netlab6 =====
1444 id name type size mem owner
1445 -----
1446 12 cookies set 0 286 auth
1447 4111 code set 281 12978 code_server
1448 8208 code_names set 56 7510 code_server
1449 12305 ign_requests set 0 286 inet_gethost_native
1450 16402 ign_req_index set 0 286 inet_gethost_native
1451 ac_tab ac_tab set 21 1339 application_controller
1452 file_io_servers file_io_servers set 1 351 file_server_2
1453 global_locks global_locks set 0 286 global_name_server
1454 global_names global_names set 391 9629 global_name_server
1455 global_names_ext global_names_ext set 0 286 global_name_server
1456 global_pid_ids global_pid_ids bag 0 286 global_name_server
1457 global_pid_names global_pid_names bag 782 8864 global_name_server
1458 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1459 hm_table_global hm_table_global bag 987 66415 <0.607.0>
1460 inet_cache inet_cache bag 0 286 inet_db
1461 inet_db inet_db set 29 559 inet_db
1462 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1463 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1464 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1465 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1466 sys_dist sys_dist set 64 2862 net_kernel
1467 ===== netlab7@netlab7 =====
1468 id name type size mem owner
1469 -----
1470 12 cookies set 0 286 auth
1471 4111 code set 281 12978 code_server
1472 8208 code_names set 56 7510 code_server
1473 12305 ign_requests set 0 286 inet_gethost_native
1474 16402 ign_req_index set 0 286 inet_gethost_native
1475 ac_tab ac_tab set 21 1339 application_controller
1476 file_io_servers file_io_servers set 1 351 file_server_2
1477 global_locks global_locks set 0 286 global_name_server
1478 global_names global_names set 391 9593 global_name_server
1479 global_names_ext global_names_ext set 0 286 global_name_server
1480 global_pid_ids global_pid_ids bag 0 286 global_name_server
1481 global_pid_names global_pid_names bag 782 8864 global_name_server
1482 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1483 hm_table_global hm_table_global bag 885 59581 <0.615.0>
1484 inet_cache inet_cache bag 0 286 inet_db
1485 inet_db inet_db set 29 559 inet_db
1486 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1487 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1488 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1489 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1490 sys_dist sys_dist set 64 2862 net_kernel
1491 ===== netlab8@netlab8 =====
1492 id name type size mem owner
1493 -----
1494 12 cookies set 0 286 auth

```

```

1495 4111          code          set 281 12978 code_server
1496 8208          code_names    set 56 7510 code_server
1497 12305         ign_requests   set 0 286 inet_gethost_native
1498 16402         ign_req_index  set 0 286 inet_gethost_native
1499 ac_tab         ac_tab        set 21 1339 application_controller
1500 file_io_servers file_io_servers set 1 351 file_server_2
1501 global_locks   global_locks   set 0 286 global_name_server
1502 global_names   global_names   set 391 9793 global_name_server
1503 global_names_ext global_names_ext set 0 286 global_name_server
1504 global_pid_ids global_pid_ids bag 0 286 global_name_server
1505 global_pid_names global_pid_names bag 782 8864 global_name_server
1506 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1507 hm_table_global hm_table_global bag 935 62931 <0.541.0>
1508 inet_cache     inet_cache     bag 0 286 inet_db
1509 inet_db        inet_db        set 29 559 inet_db
1510 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1511 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1512 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1513 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1514 sys_dist       sys_dist       set 64 2862 net_kernel
1515 ok
1516 (xxx_node@netlab3)4>

```

List 48: Test Results:Memory Usage:store function with long data

```

1 (xxx_node@netlab3)5> hm_cli_test:store_long(10000).
2 ok
3 (xxx_node@netlab3)6> hm_cli_test:check_size().
4 ===== netlab31@netlab3 =====
5 id          name          type  size  mem  owner
6 -----
7 10          cookies          set   0    279  auth
8 13          code            set 278 12598 code_server
9 14          code_names        set 48 4649 code_server
10 ac_tab      ac_tab            set 20 1389 application_controller
11 file_io_servers file_io_servers  set 1 347 file_server_2
12 global_locks global_locks      set 0 279 global_name_server
13 global_names global_names      set 391 11843 global_name_server
14 global_names_ext global_names_ext set 0 279 global_name_server
15 global_pid_ids global_pid_ids   bag 0 279 global_name_server
16 global_pid_names global_pid_names bag 782 11199 global_name_server
17 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
18 hm_table_global hm_table_global bag 520 49159 <0.55.0>
19 inet_cache   inet_cache       bag 0 279 inet_db
20 inet_db      inet_db          set 21 541 inet_db
21 inet_hosts   inet_hosts       set 0 279 inet_db
22 sys_dist     sys_dist         set 64 3047 net_kernel
23 ===== netlab32@netlab3 =====
24 id          name          type  size  mem  owner
25 -----
26 10          cookies          set   0    279  auth
27 13          code            set 279 12706 code_server
28 14          code_names        set 48 4649 code_server
29 15          ign_requests      set 0 279 inet_gethost_native
30 16          ign_req_index     set 0 279 inet_gethost_native
31 ac_tab      ac_tab            set 20 1389 application_controller
32 file_io_servers file_io_servers  set 1 347 file_server_2
33 global_locks global_locks      set 0 279 global_name_server
34 global_names global_names      set 391 11851 global_name_server
35 global_names_ext global_names_ext set 0 279 global_name_server

```

```

36 global_pid_ids global_pid_ids bag 0 279 global_name_server
37 global_pid_names global_pid_names bag 782 11203 global_name_server
38 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
39 hm_table_global hm_table_global bag 426 40323 <0.138.0>
40 inet_cache inet_cache bag 0 279 inet_db
41 inet_db inet_db set 21 541 inet_db
42 inet_hosts inet_hosts set 0 279 inet_db
43 sys_dist sys_dist set 64 3047 net_kernel
44 ===== netlab33@netlab3 =====
45 id name type size mem owner
46 -----
47 10 cookies set 0 279 auth
48 13 code set 279 12706 code_server
49 14 code_names set 48 4649 code_server
50 15 ign_requests set 0 279 inet_gethost_native
51 16 ign_req_index set 0 279 inet_gethost_native
52 ac_tab ac_tab set 20 1389 application_controller
53 file_io_servers file_io_servers set 1 347 file_server_2
54 global_locks global_locks set 0 279 global_name_server
55 global_names global_names set 391 11851 global_name_server
56 global_names_ext global_names_ext set 0 279 global_name_server
57 global_pid_ids global_pid_ids bag 0 279 global_name_server
58 global_pid_names global_pid_names bag 782 11203 global_name_server
59 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
60 hm_table_global hm_table_global bag 790 74539 <0.96.0>
61 inet_cache inet_cache bag 0 279 inet_db
62 inet_db inet_db set 21 541 inet_db
63 inet_hosts inet_hosts set 0 279 inet_db
64 sys_dist sys_dist set 64 3047 net_kernel
65 ===== netlab34@netlab3 =====
66 id name type size mem owner
67 -----
68 10 cookies set 0 279 auth
69 13 code set 279 12706 code_server
70 14 code_names set 48 4649 code_server
71 15 ign_requests set 0 279 inet_gethost_native
72 16 ign_req_index set 0 279 inet_gethost_native
73 ac_tab ac_tab set 20 1389 application_controller
74 file_io_servers file_io_servers set 1 347 file_server_2
75 global_locks global_locks set 0 279 global_name_server
76 global_names global_names set 391 11851 global_name_server
77 global_names_ext global_names_ext set 0 279 global_name_server
78 global_pid_ids global_pid_ids bag 0 279 global_name_server
79 global_pid_names global_pid_names bag 782 11203 global_name_server
80 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
81 hm_table_global hm_table_global bag 1026 96723 <0.138.0>
82 inet_cache inet_cache bag 0 279 inet_db
83 inet_db inet_db set 21 541 inet_db
84 inet_hosts inet_hosts set 0 279 inet_db
85 sys_dist sys_dist set 64 3047 net_kernel
86 ===== netlab35@netlab3 =====
87 id name type size mem owner
88 -----
89 10 cookies set 0 279 auth
90 13 code set 279 12706 code_server
91 14 code_names set 48 4649 code_server
92 15 ign_requests set 0 279 inet_gethost_native
93 16 ign_req_index set 0 279 inet_gethost_native
94 ac_tab ac_tab set 20 1389 application_controller
95 file_io_servers file_io_servers set 1 347 file_server_2

```



```

96 global_locks      global_locks      set    0      279      global_name_server
97 global_names      global_names      set   391    11851    global_name_server
98 global_names_ext  global_names_ext  set    0      279      global_name_server
99 global_pid_ids    global_pid_ids    bag    0      279      global_name_server
100 global_pid_names  global_pid_names  bag   782    11203    global_name_server
101 hm_ets_cache_table hm_ets_cache_table set    0      279      hm_cache_mgr
102 hm_table_global  hm_table_global  bag   768    72471    <0.103.0>
103 inet_cache       inet_cache       bag    0      279      inet_db
104 inet_db          inet_db          set   21     541      inet_db
105 inet_hosts       inet_hosts       set    0      279      inet_db
106 sys_dist         sys_dist        set   64     3047     net_kernel
107 ===== netlab36@netlab3 =====
108 id              name              type    size    mem      owner
109 -----
110 10               cookies           set     0      279      auth
111 13               code             set    279    12706    code_server
112 14               code_names       set    48     4649     code_server
113 15               ign_requests     set     0      279      inet_gethost_native
114 16               ign_req_index    set     0      279      inet_gethost_native
115 ac_tab          ac_tab           set    20     1389     application_controller
116 file_io_servers file_io_servers  set     1      347      file_server_2
117 global_locks     global_locks     set     0      279      global_name_server
118 global_names     global_names     set   391    11851    global_name_server
119 global_names_ext global_names_ext set     0      279      global_name_server
120 global_pid_ids   global_pid_ids   bag     0      279      global_name_server
121 global_pid_names global_pid_names bag   782    11203    global_name_server
122 hm_ets_cache_table hm_ets_cache_table set    0      279      hm_cache_mgr
123 hm_table_global  hm_table_global  bag   764    72095    <0.133.0>
124 inet_cache       inet_cache       bag     0      279      inet_db
125 inet_db          inet_db          set   21     541      inet_db
126 inet_hosts       inet_hosts       set     0      279      inet_db
127 sys_dist         sys_dist        set   64     3047     net_kernel
128 ===== netlab37@netlab3 =====
129 id              name              type    size    mem      owner
130 -----
131 10               cookies           set     0      279      auth
132 13               code             set    279    12706    code_server
133 14               code_names       set    48     4649     code_server
134 15               ign_requests     set     0      279      inet_gethost_native
135 16               ign_req_index    set     0      279      inet_gethost_native
136 ac_tab          ac_tab           set    20     1389     application_controller
137 file_io_servers file_io_servers  set     1      347      file_server_2
138 global_locks     global_locks     set     0      279      global_name_server
139 global_names     global_names     set   391    11851    global_name_server
140 global_names_ext global_names_ext set     0      279      global_name_server
141 global_pid_ids   global_pid_ids   bag     0      279      global_name_server
142 global_pid_names global_pid_names bag   782    11203    global_name_server
143 hm_ets_cache_table hm_ets_cache_table set    0      279      hm_cache_mgr
144 hm_table_global  hm_table_global  bag  1543   145321    <0.98.0>
145 inet_cache       inet_cache       bag     0      279      inet_db
146 inet_db          inet_db          set   21     541      inet_db
147 inet_hosts       inet_hosts       set     0      279      inet_db
148 sys_dist         sys_dist        set   64     3047     net_kernel
149 ===== netlab38@netlab3 =====
150 id              name              type    size    mem      owner
151 -----
152 10               cookies           set     0      279      auth
153 13               code             set    279    12706    code_server
154 14               code_names       set    48     4649     code_server
155 15               ign_requests     set     0      279      inet_gethost_native

```

```

156 16          ign_req_index      set 0      279      inet_gethost_native
157 ac_tab      ac_tab             set 20     1389     application_controller
158 file_io_servers file_io_servers set 1      347     file_server_2
159 global_locks global_locks       set 0      279     global_name_server
160 global_names global_names       set 391    11851   global_name_server
161 global_names_ext global_names_ext set 0      279     global_name_server
162 global_pid_ids global_pid_ids     bag 0      279     global_name_server
163 global_pid_names global_pid_names bag 782     11203   global_name_server
164 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
165 hm_table_global hm_table_global bag 1226    115523  <0.123.0>
166 inet_cache     inet_cache         bag 0      279     inet_db
167 inet_db        inet_db           set 21     541     inet_db
168 inet_hosts     inet_hosts         set 0      279     inet_db
169 sys_dist       sys_dist         set 64     3047    net_kernel
170 ===== netlab39@netlab3 =====
171 id            name                type  size    mem    owner
172 -----
173 10            cookies              set 0      279     auth
174 13            code                 set 279    12706   code_server
175 14            code_names           set 48     4649    code_server
176 15            ign_requests         set 0      279     inet_gethost_native
177 16            ign_req_index        set 0      279     inet_gethost_native
178 ac_tab        ac_tab              set 20     1389    application_controller
179 file_io_servers file_io_servers     set 1      347     file_server_2
180 global_locks global_locks        set 0      279     global_name_server
181 global_names global_names        set 391    11839   global_name_server
182 global_names_ext global_names_ext    set 0      279     global_name_server
183 global_pid_ids global_pid_ids      bag 0      279     global_name_server
184 global_pid_names global_pid_names    bag 782     11203   global_name_server
185 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
186 hm_table_global hm_table_global    bag 682    64387   <0.136.0>
187 inet_cache     inet_cache         bag 0      279     inet_db
188 inet_db        inet_db           set 21     541     inet_db
189 inet_hosts     inet_hosts         set 0      279     inet_db
190 sys_dist       sys_dist         set 64     3047    net_kernel
191 ===== netlab310@netlab3 =====
192 id            name                type  size    mem    owner
193 -----
194 10            cookies              set 0      279     auth
195 13            code                 set 279    12706   code_server
196 14            code_names           set 48     4649    code_server
197 15            ign_requests         set 0      279     inet_gethost_native
198 16            ign_req_index        set 0      279     inet_gethost_native
199 ac_tab        ac_tab              set 20     1389    application_controller
200 file_io_servers file_io_servers     set 1      349     file_server_2
201 global_locks global_locks        set 0      279     global_name_server
202 global_names global_names        set 391    11839   global_name_server
203 global_names_ext global_names_ext    set 0      279     global_name_server
204 global_pid_ids global_pid_ids      bag 0      279     global_name_server
205 global_pid_names global_pid_names    bag 782     11203   global_name_server
206 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
207 hm_table_global hm_table_global    bag 449    42485   <0.128.0>
208 inet_cache     inet_cache         bag 0      279     inet_db
209 inet_db        inet_db           set 21     541     inet_db
210 inet_hosts     inet_hosts         set 0      279     inet_db
211 sys_dist       sys_dist         set 64     3047    net_kernel
212 ===== netlab311@netlab3 =====
213 id            name                type  size    mem    owner
214 -----
215 10            cookies              set 0      279     auth

```

```

216 13          code          set 279 12706 code_server
217 14          code_names    set 48 4649 code_server
218 15          ign_requests  set 0 279 inet_gethost_native
219 16          ign_req_index set 0 279 inet_gethost_native
220 ac_tab      ac_tab        set 20 1389 application_controller
221 file_io_servers file_io_servers set 1 349 file_server_2
222 global_locks global_locks set 0 279 global_name_server
223 global_names global_names set 391 11839 global_name_server
224 global_names_ext global_names_ext set 0 279 global_name_server
225 global_pid_ids global_pid_ids bag 0 279 global_name_server
226 global_pid_names global_pid_names bag 782 11203 global_name_server
227 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
228 hm_table_global hm_table_global bag 1552 146167 <0.139.0>
229 inet_cache     inet_cache     bag 0 279 inet_db
230 inet_db        inet_db        set 21 541 inet_db
231 inet_hosts     inet_hosts     set 0 279 inet_db
232 sys_dist      sys_dist      set 64 3047 net_kernel
233 ===== netlab312@netlab3 =====
234 id            name          type  size  mem  owner
235 -----
236 10            cookies         set 0 279 auth
237 13            code          set 279 12706 code_server
238 14            code_names    set 48 4649 code_server
239 15            ign_requests  set 0 279 inet_gethost_native
240 16            ign_req_index set 0 279 inet_gethost_native
241 ac_tab      ac_tab        set 20 1389 application_controller
242 file_io_servers file_io_servers set 1 349 file_server_2
243 global_locks global_locks set 0 279 global_name_server
244 global_names global_names set 391 11803 global_name_server
245 global_names_ext global_names_ext set 0 279 global_name_server
246 global_pid_ids global_pid_ids bag 0 279 global_name_server
247 global_pid_names global_pid_names bag 782 11203 global_name_server
248 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
249 hm_table_global hm_table_global bag 853 80461 <0.156.0>
250 inet_cache     inet_cache     bag 0 279 inet_db
251 inet_db        inet_db        set 21 541 inet_db
252 inet_hosts     inet_hosts     set 0 279 inet_db
253 sys_dist      sys_dist      set 64 3047 net_kernel
254 ===== netlab313@netlab3 =====
255 id            name          type  size  mem  owner
256 -----
257 10            cookies         set 0 279 auth
258 13            code          set 279 12706 code_server
259 14            code_names    set 48 4649 code_server
260 15            ign_requests  set 0 279 inet_gethost_native
261 16            ign_req_index set 0 279 inet_gethost_native
262 ac_tab      ac_tab        set 20 1389 application_controller
263 file_io_servers file_io_servers set 1 349 file_server_2
264 global_locks global_locks set 0 279 global_name_server
265 global_names global_names set 391 11803 global_name_server
266 global_names_ext global_names_ext set 0 279 global_name_server
267 global_pid_ids global_pid_ids bag 0 279 global_name_server
268 global_pid_names global_pid_names bag 782 11203 global_name_server
269 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
270 hm_table_global hm_table_global bag 499 47185 <0.139.0>
271 inet_cache     inet_cache     bag 0 279 inet_db
272 inet_db        inet_db        set 21 541 inet_db
273 inet_hosts     inet_hosts     set 0 279 inet_db
274 sys_dist      sys_dist      set 64 3047 net_kernel
275 ===== netlab314@netlab3 =====

```

```

276 id          name          type  size  mem  owner
277 -----
278 10           cookies        set    0    279  auth
279 13           code          set   279  12706 code_server
280 14           code_names      set    48  4649  code_server
281 15           ign_requests   set     0  279  inet_gethost_native
282 16           ign_req_index  set     0  279  inet_gethost_native
283 ac_tab       ac_tab          set    20  1389  application_controller
284 file_io_servers file_io_servers set     1   349  file_server_2
285 global_locks global_locks    set     0  279  global_name_server
286 global_names global_names    set   391  11755 global_name_server
287 global_names_ext global_names_ext set     0   279  global_name_server
288 global_pid_ids global_pid_ids  bag     0  279  global_name_server
289 global_pid_names global_pid_names bag    782  11203 global_name_server
290 hm_ets_cache_table hm_ets_cache_table set     0   279  hm_cache_mgr
291 hm_table_global hm_table_global bag    724  68335 <0.171.0>
292 inet_cache     inet_cache     bag     0   279  inet_db
293 inet_db        inet_db        set    21   541  inet_db
294 inet_hosts     inet_hosts     set     0   279  inet_db
295 sys_dist       sys_dist       set    64  3047  net_kernel
296 ===== netlab315@netlab3 =====
297 id          name          type  size  mem  owner
298 -----
299 10           cookies        set     0   279  auth
300 13           code          set   279  12706 code_server
301 14           code_names      set    48  4649  code_server
302 15           ign_requests   set     0  279  inet_gethost_native
303 16           ign_req_index  set     0  279  inet_gethost_native
304 ac_tab       ac_tab          set    20  1389  application_controller
305 file_io_servers file_io_servers set     1   349  file_server_2
306 global_locks global_locks    set     0  279  global_name_server
307 global_names global_names    set   391  11755 global_name_server
308 global_names_ext global_names_ext set     0   279  global_name_server
309 global_pid_ids global_pid_ids  bag     0  279  global_name_server
310 global_pid_names global_pid_names bag    782  11203 global_name_server
311 hm_ets_cache_table hm_ets_cache_table set     0   279  hm_cache_mgr
312 hm_table_global hm_table_global bag    569  53765 <0.158.0>
313 inet_cache     inet_cache     bag     0   279  inet_db
314 inet_db        inet_db        set    21   541  inet_db
315 inet_hosts     inet_hosts     set     0   279  inet_db
316 sys_dist       sys_dist       set    64  3047  net_kernel
317 ===== netlab316@netlab3 =====
318 id          name          type  size  mem  owner
319 -----
320 10           cookies        set     0   279  auth
321 13           code          set   279  12706 code_server
322 14           code_names      set    48  4649  code_server
323 15           ign_requests   set     0  279  inet_gethost_native
324 16           ign_req_index  set     0  279  inet_gethost_native
325 ac_tab       ac_tab          set    20  1389  application_controller
326 file_io_servers file_io_servers set     1   349  file_server_2
327 global_locks global_locks    set     0  279  global_name_server
328 global_names global_names    set   391  11755 global_name_server
329 global_names_ext global_names_ext set     0   279  global_name_server
330 global_pid_ids global_pid_ids  bag     0  279  global_name_server
331 global_pid_names global_pid_names bag    782  11203 global_name_server
332 hm_ets_cache_table hm_ets_cache_table set     0   279  hm_cache_mgr
333 hm_table_global hm_table_global bag    710  67019 <0.159.0>
334 inet_cache     inet_cache     bag     0   279  inet_db
335 inet_db        inet_db        set    21   541  inet_db

```

```

336 inet_hosts      inet_hosts      set  0      279      inet_db
337 sys_dist        sys_dist        set  64     3047     net_kernel
338 ===== netlab317@netlab3 =====
339 id              name              type  size  mem      owner
340 -----
341 10              cookies           set   0      279      auth
342 13              code              set  279    12706    code_server
343 14              code_names        set   48    4649    code_server
344 15              ign_requests      set   0      279      inet_gethost_native
345 16              ign_req_index     set   0      279      inet_gethost_native
346 ac_tab          ac_tab            set  20     1389     application_controller
347 file_io_servers file_io_servers    set   1      349      file_server_2
348 global_locks    global_locks      set   0      279      global_name_server
349 global_names    global_names      set  391    11695    global_name_server
350 global_names_ext global_names_ext  set   0      279      global_name_server
351 global_pid_ids  global_pid_ids    bag   0      279      global_name_server
352 global_pid_names global_pid_names  bag  782    11203    global_name_server
353 hm_ets_cache_table hm_ets_cache_table set   0      279      hm_cache_mgr
354 hm_table_global hm_table_global   bag  880    82999    <0.169.0>
355 inet_cache      inet_cache        bag   0      279      inet_db
356 inet_db         inet_db           set  21     541      inet_db
357 inet_hosts      inet_hosts        set   0      279      inet_db
358 sys_dist        sys_dist          set  64     3047     net_kernel
359 ===== netlab318@netlab3 =====
360 id              name              type  size  mem      owner
361 -----
362 10              cookies           set   0      279      auth
363 13              code              set  279    12706    code_server
364 14              code_names        set   48    4649    code_server
365 15              ign_requests      set   0      279      inet_gethost_native
366 16              ign_req_index     set   0      279      inet_gethost_native
367 ac_tab          ac_tab            set  20     1389     application_controller
368 file_io_servers file_io_servers    set   1      349      file_server_2
369 global_locks    global_locks      set   0      279      global_name_server
370 global_names    global_names      set  391    11695    global_name_server
371 global_names_ext global_names_ext  set   0      279      global_name_server
372 global_pid_ids  global_pid_ids    bag   0      279      global_name_server
373 global_pid_names global_pid_names  bag  782    11203    global_name_server
374 hm_ets_cache_table hm_ets_cache_table set   0      279      hm_cache_mgr
375 hm_table_global hm_table_global   bag  752    70967    <0.174.0>
376 inet_cache      inet_cache        bag   0      279      inet_db
377 inet_db         inet_db           set  21     541      inet_db
378 inet_hosts      inet_hosts        set   0      279      inet_db
379 sys_dist        sys_dist          set  64     3047     net_kernel
380 ===== netlab319@netlab3 =====
381 id              name              type  size  mem      owner
382 -----
383 10              cookies           set   0      279      auth
384 13              code              set  279    12706    code_server
385 14              code_names        set   48    4649    code_server
386 15              ign_requests      set   0      279      inet_gethost_native
387 16              ign_req_index     set   0      279      inet_gethost_native
388 ac_tab          ac_tab            set  20     1389     application_controller
389 file_io_servers file_io_servers    set   1      349      file_server_2
390 global_locks    global_locks      set   0      279      global_name_server
391 global_names    global_names      set  391    11755    global_name_server
392 global_names_ext global_names_ext  set   0      279      global_name_server
393 global_pid_ids  global_pid_ids    bag   0      279      global_name_server
394 global_pid_names global_pid_names  bag  782    11203    global_name_server
395 hm_ets_cache_table hm_ets_cache_table set   0      279      hm_cache_mgr

```

```

396 hm_table_global hm_table_global bag 472 44647 <0.169.0>
397 inet_cache inet_cache bag 0 279 inet_db
398 inet_db inet_db set 21 541 inet_db
399 inet_hosts inet_hosts set 0 279 inet_db
400 sys_dist sys_dist set 64 3047 net_kernel
401 ===== netlab320@netlab3 =====
402 id name type size mem owner
403 -----
404 10 cookies set 0 279 auth
405 13 code set 279 12706 code_server
406 14 code_names set 48 4649 code_server
407 15 ign_requests set 0 279 inet_gethost_native
408 16 ign_req_index set 0 279 inet_gethost_native
409 ac_tab ac_tab set 20 1389 application_controller
410 file_io_servers file_io_servers set 1 349 file_server_2
411 global_locks global_locks set 0 279 global_name_server
412 global_names global_names set 391 11695 global_name_server
413 global_names_ext global_names_ext set 0 279 global_name_server
414 global_pid_ids global_pid_ids bag 0 279 global_name_server
415 global_pid_names global_pid_names bag 782 11203 global_name_server
416 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
417 hm_table_global hm_table_global bag 384 36375 <0.179.0>
418 inet_cache inet_cache bag 0 279 inet_db
419 inet_db inet_db set 21 541 inet_db
420 inet_hosts inet_hosts set 0 279 inet_db
421 sys_dist sys_dist set 64 3047 net_kernel
422 ===== netlab41@netlab4 =====
423 id name type size mem owner
424 -----
425 10 cookies set 0 279 auth
426 13 code set 279 12706 code_server
427 14 code_names set 48 4649 code_server
428 15 ign_requests set 0 279 inet_gethost_native
429 16 ign_req_index set 0 279 inet_gethost_native
430 ac_tab ac_tab set 20 1389 application_controller
431 file_io_servers file_io_servers set 1 347 file_server_2
432 global_locks global_locks set 0 279 global_name_server
433 global_names global_names set 391 11395 global_name_server
434 global_names_ext global_names_ext set 0 279 global_name_server
435 global_pid_ids global_pid_ids bag 0 279 global_name_server
436 global_pid_names global_pid_names bag 782 11203 global_name_server
437 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
438 hm_table_global hm_table_global bag 892 84127 <0.304.0>
439 inet_cache inet_cache bag 0 279 inet_db
440 inet_db inet_db set 21 541 inet_db
441 inet_hosts inet_hosts set 0 279 inet_db
442 sys_dist sys_dist set 64 3047 net_kernel
443 ===== netlab42@netlab4 =====
444 id name type size mem owner
445 -----
446 10 cookies set 0 279 auth
447 13 code set 279 12706 code_server
448 14 code_names set 48 4649 code_server
449 15 ign_requests set 0 279 inet_gethost_native
450 16 ign_req_index set 0 279 inet_gethost_native
451 ac_tab ac_tab set 20 1389 application_controller
452 file_io_servers file_io_servers set 1 347 file_server_2
453 global_locks global_locks set 0 279 global_name_server
454 global_names global_names set 391 11395 global_name_server
455 global_names_ext global_names_ext set 0 279 global_name_server

```

```

456 global_pid_ids global_pid_ids bag 0 279 global_name_server
457 global_pid_names global_pid_names bag 782 11203 global_name_server
458 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
459 hm_table_global hm_table_global bag 462 43707 <0.300.0>
460 inet_cache inet_cache bag 0 279 inet_db
461 inet_db inet_db set 21 541 inet_db
462 inet_hosts inet_hosts set 0 279 inet_db
463 sys_dist sys_dist set 64 3047 net_kernel
464 ===== netlab43@netlab4 =====
465 id name type size mem owner
466 -----
467 10 cookies set 0 279 auth
468 13 code set 279 12706 code_server
469 14 code_names set 48 4649 code_server
470 15 ign_requests set 0 279 inet_gethost_native
471 16 ign_req_index set 0 279 inet_gethost_native
472 ac_tab ac_tab set 20 1389 application_controller
473 file_io_servers file_io_servers set 1 347 file_server_2
474 global_locks global_locks set 0 279 global_name_server
475 global_names global_names set 391 11395 global_name_server
476 global_names_ext global_names_ext set 0 279 global_name_server
477 global_pid_ids global_pid_ids bag 0 279 global_name_server
478 global_pid_names global_pid_names bag 782 11203 global_name_server
479 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
480 hm_table_global hm_table_global bag 395 37409 <0.351.0>
481 inet_cache inet_cache bag 0 279 inet_db
482 inet_db inet_db set 21 541 inet_db
483 inet_hosts inet_hosts set 0 279 inet_db
484 sys_dist sys_dist set 64 3047 net_kernel
485 ===== netlab44@netlab4 =====
486 id name type size mem owner
487 -----
488 10 cookies set 0 279 auth
489 13 code set 279 12706 code_server
490 14 code_names set 48 4649 code_server
491 15 ign_requests set 0 279 inet_gethost_native
492 16 ign_req_index set 0 279 inet_gethost_native
493 ac_tab ac_tab set 20 1389 application_controller
494 file_io_servers file_io_servers set 1 347 file_server_2
495 global_locks global_locks set 0 279 global_name_server
496 global_names global_names set 391 11395 global_name_server
497 global_names_ext global_names_ext set 0 279 global_name_server
498 global_pid_ids global_pid_ids bag 0 279 global_name_server
499 global_pid_names global_pid_names bag 782 11203 global_name_server
500 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
501 hm_table_global hm_table_global bag 1054 99355 <0.343.0>
502 inet_cache inet_cache bag 0 279 inet_db
503 inet_db inet_db set 21 541 inet_db
504 inet_hosts inet_hosts set 0 279 inet_db
505 sys_dist sys_dist set 64 3047 net_kernel
506 ===== netlab45@netlab4 =====
507 id name type size mem owner
508 -----
509 10 cookies set 0 279 auth
510 13 code set 279 12706 code_server
511 14 code_names set 48 4649 code_server
512 15 ign_requests set 0 279 inet_gethost_native
513 16 ign_req_index set 0 279 inet_gethost_native
514 ac_tab ac_tab set 20 1389 application_controller
515 file_io_servers file_io_servers set 1 347 file_server_2

```

```

516 global_locks      global_locks      set    0      279      global_name_server
517 global_names      global_names      set   391    11395    global_name_server
518 global_names_ext  global_names_ext  set    0      279      global_name_server
519 global_pid_ids    global_pid_ids    bag    0      279      global_name_server
520 global_pid_names  global_pid_names  bag   782   11203    global_name_server
521 hm_ets_cache_table hm_ets_cache_table set    0      279      hm_cache_mgr
522 hm_table_global   hm_table_global   bag   640   60439    <0.319.0>
523 inet_cache        inet_cache        bag    0      279      inet_db
524 inet_db           inet_db           set   21     541      inet_db
525 inet_hosts        inet_hosts        set    0      279      inet_db
526 sys_dist          sys_dist          set   64    3047     net_kernel
527 ===== netlab46@netlab4 =====
528 id                name                type    size    mem      owner
529 -----
530 10                 cookies             set     0      279      auth
531 13                 code                set    279    12706    code_server
532 14                 code_names          set     48    4649     code_server
533 15                 ign_requests        set     0      279      inet_gethost_native
534 16                 ign_req_index       set     0      279      inet_gethost_native
535 ac_tab            ac_tab              set    20    1389     application_controller
536 file_io_servers   file_io_servers     set     1      347      file_server_2
537 global_locks      global_locks        set     0      279      global_name_server
538 global_names      global_names        set   391    11395    global_name_server
539 global_names_ext  global_names_ext    set     0      279      global_name_server
540 global_pid_ids    global_pid_ids      bag     0      279      global_name_server
541 global_pid_names  global_pid_names    bag   782   11203    global_name_server
542 hm_ets_cache_table hm_ets_cache_table set     0      279      hm_cache_mgr
543 hm_table_global   hm_table_global     bag   710   67019    <0.351.0>
544 inet_cache        inet_cache          bag     0      279      inet_db
545 inet_db           inet_db             set   21     541      inet_db
546 inet_hosts        inet_hosts          set     0      279      inet_db
547 sys_dist          sys_dist            set    64    3047     net_kernel
548 ===== netlab47@netlab4 =====
549 id                name                type    size    mem      owner
550 -----
551 10                 cookies             set     0      279      auth
552 13                 code                set    279    12706    code_server
553 14                 code_names          set     48    4649     code_server
554 15                 ign_requests        set     0      279      inet_gethost_native
555 16                 ign_req_index       set     0      279      inet_gethost_native
556 ac_tab            ac_tab              set    20    1389     application_controller
557 file_io_servers   file_io_servers     set     1      347      file_server_2
558 global_locks      global_locks        set     0      279      global_name_server
559 global_names      global_names        set   391    11395    global_name_server
560 global_names_ext  global_names_ext    set     0      279      global_name_server
561 global_pid_ids    global_pid_ids      bag     0      279      global_name_server
562 global_pid_names  global_pid_names    bag   782   11203    global_name_server
563 hm_ets_cache_table hm_ets_cache_table set     0      279      hm_cache_mgr
564 hm_table_global   hm_table_global     bag   441   41733    <0.345.0>
565 inet_cache        inet_cache          bag     0      279      inet_db
566 inet_db           inet_db             set   21     541      inet_db
567 inet_hosts        inet_hosts          set     0      279      inet_db
568 sys_dist          sys_dist            set    64    3047     net_kernel
569 ===== netlab48@netlab4 =====
570 id                name                type    size    mem      owner
571 -----
572 10                 cookies             set     0      279      auth
573 13                 code                set    279    12706    code_server
574 14                 code_names          set     48    4649     code_server
575 15                 ign_requests        set     0      279      inet_gethost_native

```



```

576 16          ign_req_index      set 0      279      inet_gethost_native
577 ac_tab      ac_tab             set 20     1389     application_controller
578 file_io_servers file_io_servers set 1      347     file_server_2
579 global_locks global_locks      set 0      279     global_name_server
580 global_names global_names      set 391    11323   global_name_server
581 global_names_ext global_names_ext set 0      279     global_name_server
582 global_pid_ids global_pid_ids    bag 0      279     global_name_server
583 global_pid_names global_pid_names bag 782    11203   global_name_server
584 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
585 hm_table_global hm_table_global bag 752    70967   <0.373.0>
586 inet_cache    inet_cache      bag 0      279     inet_db
587 inet_db       inet_db        set 21     541     inet_db
588 inet_hosts    inet_hosts      set 0      279     inet_db
589 sys_dist      sys_dist        set 64     3047    net_kernel
590 ===== netlab49@netlab4 =====
591 id            name            type  size    mem    owner
592 -----
593 10            cookies          set 0      279     auth
594 13            code            set 279    12706   code_server
595 14            code_names      set 48     4649    code_server
596 15            ign_requests    set 0      279     inet_gethost_native
597 16            ign_req_index   set 0      279     inet_gethost_native
598 ac_tab      ac_tab             set 20     1389     application_controller
599 file_io_servers file_io_servers set 1      347     file_server_2
600 global_locks global_locks      set 0      279     global_name_server
601 global_names global_names      set 391    11395   global_name_server
602 global_names_ext global_names_ext set 0      279     global_name_server
603 global_pid_ids global_pid_ids    bag 0      279     global_name_server
604 global_pid_names global_pid_names bag 782    11203   global_name_server
605 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
606 hm_table_global hm_table_global bag 976    92023   <0.301.0>
607 inet_cache    inet_cache      bag 0      279     inet_db
608 inet_db       inet_db        set 21     541     inet_db
609 inet_hosts    inet_hosts      set 0      279     inet_db
610 sys_dist      sys_dist        set 64     3047    net_kernel
611 ===== netlab410@netlab4 =====
612 id            name            type  size    mem    owner
613 -----
614 10            cookies          set 0      279     auth
615 13            code            set 279    12706   code_server
616 14            code_names      set 48     4649    code_server
617 15            ign_requests    set 0      279     inet_gethost_native
618 16            ign_req_index   set 0      279     inet_gethost_native
619 ac_tab      ac_tab             set 20     1389     application_controller
620 file_io_servers file_io_servers set 1      349     file_server_2
621 global_locks global_locks      set 0      279     global_name_server
622 global_names global_names      set 391    11375   global_name_server
623 global_names_ext global_names_ext set 0      279     global_name_server
624 global_pid_ids global_pid_ids    bag 0      279     global_name_server
625 global_pid_names global_pid_names bag 782    11203   global_name_server
626 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
627 hm_table_global hm_table_global bag 847    79897   <0.353.0>
628 inet_cache    inet_cache      bag 0      279     inet_db
629 inet_db       inet_db        set 21     541     inet_db
630 inet_hosts    inet_hosts      set 0      279     inet_db
631 sys_dist      sys_dist        set 64     3047    net_kernel
632 ===== netlab411@netlab4 =====
633 id            name            type  size    mem    owner
634 -----
635 10            cookies          set 0      279     auth

```

```

636 13          code          set 279 12706 code_server
637 14          code_names    set 48 4649 code_server
638 15          ign_requests   set 0 279 inet_gethost_native
639 16          ign_req_index  set 0 279 inet_gethost_native
640 ac_tab      ac_tab        set 20 1389 application_controller
641 file_io_servers file_io_servers set 1 349 file_server_2
642 global_locks global_locks set 0 279 global_name_server
643 global_names global_names set 391 11299 global_name_server
644 global_names_ext global_names_ext set 0 279 global_name_server
645 global_pid_ids global_pid_ids bag 0 279 global_name_server
646 global_pid_names global_pid_names bag 782 11203 global_name_server
647 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
648 hm_table_global hm_table_global bag 795 75009 <0.387.0>
649 inet_cache    inet_cache    bag 0 279 inet_db
650 inet_db        inet_db        set 21 541 inet_db
651 inet_hosts     inet_hosts     set 0 279 inet_db
652 sys_dist       sys_dist       set 64 3047 net_kernel
653 ===== netlab412@netlab4 =====
654 id            name            type  size  mem  owner
655 -----
656 10            cookies          set 0 279 auth
657 13            code          set 279 12706 code_server
658 14            code_names    set 48 4649 code_server
659 15            ign_requests   set 0 279 inet_gethost_native
660 16            ign_req_index  set 0 279 inet_gethost_native
661 ac_tab        ac_tab          set 20 1389 application_controller
662 file_io_servers file_io_servers set 1 349 file_server_2
663 global_locks  global_locks set 0 279 global_name_server
664 global_names  global_names set 391 11299 global_name_server
665 global_names_ext global_names_ext set 0 279 global_name_server
666 global_pid_ids global_pid_ids bag 0 279 global_name_server
667 global_pid_names global_pid_names bag 782 11203 global_name_server
668 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
669 hm_table_global hm_table_global bag 881 83093 <0.379.0>
670 inet_cache    inet_cache    bag 0 279 inet_db
671 inet_db        inet_db        set 21 541 inet_db
672 inet_hosts     inet_hosts     set 0 279 inet_db
673 sys_dist       sys_dist       set 64 3047 net_kernel
674 ===== netlab413@netlab4 =====
675 id            name            type  size  mem  owner
676 -----
677 10            cookies          set 0 279 auth
678 13            code          set 279 12706 code_server
679 14            code_names    set 48 4649 code_server
680 15            ign_requests   set 0 279 inet_gethost_native
681 16            ign_req_index  set 0 279 inet_gethost_native
682 ac_tab        ac_tab          set 20 1389 application_controller
683 file_io_servers file_io_servers set 1 349 file_server_2
684 global_locks  global_locks set 0 279 global_name_server
685 global_names  global_names set 391 11299 global_name_server
686 global_names_ext global_names_ext set 0 279 global_name_server
687 global_pid_ids global_pid_ids bag 0 279 global_name_server
688 global_pid_names global_pid_names bag 782 11203 global_name_server
689 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
690 hm_table_global hm_table_global bag 420 39759 <0.387.0>
691 inet_cache    inet_cache    bag 0 279 inet_db
692 inet_db        inet_db        set 21 541 inet_db
693 inet_hosts     inet_hosts     set 0 279 inet_db
694 sys_dist       sys_dist       set 64 3047 net_kernel
695 ===== netlab414@netlab4 =====

```

```

696 id          name          type  size  mem  owner
697 -----
698 10           cookies        set    0    279  auth
699 13           code          set   279  12706 code_server
700 14           code_names       set    48  4649  code_server
701 15           ign_requests    set     0  279  inet_gethost_native
702 16           ign_req_index    set     0  279  inet_gethost_native
703 ac_tab       ac_tab           set    20  1389  application_controller
704 file_io_servers file_io_servers set     1   349  file_server_2
705 global_locks global_locks     set     0  279  global_name_server
706 global_names global_names     set   391  11275 global_name_server
707 global_names_ext global_names_ext set     0   279  global_name_server
708 global_pid_ids global_pid_ids   bag     0  279  global_name_server
709 global_pid_names global_pid_names bag    782  11203 global_name_server
710 hm_ets_cache_table hm_ets_cache_table set     0   279  hm_cache_mgr
711 hm_table_global hm_table_global bag   1187  111857 <0.393.0>
712 inet_cache     inet_cache       bag     0   279  inet_db
713 inet_db        inet_db          set    21   541  inet_db
714 inet_hosts     inet_hosts       set     0   279  inet_db
715 sys_dist       sys_dist         set    64  3047  net_kernel
716 ===== netlab415@netlab4 =====
717 id          name          type  size  mem  owner
718 -----
719 10           cookies        set    0    279  auth
720 13           code          set   279  12706 code_server
721 14           code_names       set    48  4649  code_server
722 15           ign_requests    set     0  279  inet_gethost_native
723 16           ign_req_index    set     0  279  inet_gethost_native
724 ac_tab       ac_tab           set    20  1389  application_controller
725 file_io_servers file_io_servers set     1   349  file_server_2
726 global_locks global_locks     set     0  279  global_name_server
727 global_names global_names     set   391  11275 global_name_server
728 global_names_ext global_names_ext set     0   279  global_name_server
729 global_pid_ids global_pid_ids   bag     0  279  global_name_server
730 global_pid_names global_pid_names bag    782  11203 global_name_server
731 hm_ets_cache_table hm_ets_cache_table set     0   279  hm_cache_mgr
732 hm_table_global hm_table_global bag    644  60815 <0.390.0>
733 inet_cache     inet_cache       bag     0   279  inet_db
734 inet_db        inet_db          set    21   541  inet_db
735 inet_hosts     inet_hosts       set     0   279  inet_db
736 sys_dist       sys_dist         set    64  3047  net_kernel
737 ===== netlab416@netlab4 =====
738 id          name          type  size  mem  owner
739 -----
740 10           cookies        set    0    279  auth
741 13           code          set   279  12706 code_server
742 14           code_names       set    48  4649  code_server
743 15           ign_requests    set     0  279  inet_gethost_native
744 16           ign_req_index    set     0  279  inet_gethost_native
745 ac_tab       ac_tab           set    20  1389  application_controller
746 file_io_servers file_io_servers set     1   349  file_server_2
747 global_locks global_locks     set     0  279  global_name_server
748 global_names global_names     set   391  11275 global_name_server
749 global_names_ext global_names_ext set     0   279  global_name_server
750 global_pid_ids global_pid_ids   bag     0  279  global_name_server
751 global_pid_names global_pid_names bag    782  11203 global_name_server
752 hm_ets_cache_table hm_ets_cache_table set     0   279  hm_cache_mgr
753 hm_table_global hm_table_global bag    901  84973 <0.375.0>
754 inet_cache     inet_cache       bag     0   279  inet_db
755 inet_db        inet_db          set    21   541  inet_db

```

```

756 inet_hosts      inet_hosts      set  0      279      inet_db
757 sys_dist        sys_dist        set  64     3047     net_kernel
758 ===== netlab417@netlab4 =====
759 id              name              type  size  mem      owner
760 -----
761 10              cookies           set   0      279      auth
762 13              code              set  279    12706    code_server
763 14              code_names        set   48    4649    code_server
764 15              ign_requests      set   0      279      inet_gethost_native
765 16              ign_req_index     set   0      279      inet_gethost_native
766 ac_tab          ac_tab            set  20     1389     application_controller
767 file_io_servers file_io_servers    set   1      349      file_server_2
768 global_locks    global_locks       set   0      279      global_name_server
769 global_names    global_names       set  391    11275    global_name_server
770 global_names_ext global_names_ext   set   0      279      global_name_server
771 global_pid_ids  global_pid_ids     bag   0      279      global_name_server
772 global_pid_names global_pid_names   bag  782    11203    global_name_server
773 hm_ets_cache_table hm_ets_cache_table set   0      279      hm_cache_mgr
774 hm_table_global hm_table_global    bag  1474   138835   <0.382.0>
775 inet_cache      inet_cache         bag   0      279      inet_db
776 inet_db         inet_db            set  21     541      inet_db
777 inet_hosts      inet_hosts         set   0      279      inet_db
778 sys_dist        sys_dist          set  64     3047     net_kernel
779 ===== netlab418@netlab4 =====
780 id              name              type  size  mem      owner
781 -----
782 10              cookies           set   0      279      auth
783 13              code              set  279    12706    code_server
784 14              code_names        set   48    4649    code_server
785 15              ign_requests      set   0      279      inet_gethost_native
786 16              ign_req_index     set   0      279      inet_gethost_native
787 ac_tab          ac_tab            set  20     1389     application_controller
788 file_io_servers file_io_servers    set   1      349      file_server_2
789 global_locks    global_locks       set   0      279      global_name_server
790 global_names    global_names       set  391    11239    global_name_server
791 global_names_ext global_names_ext   set   0      279      global_name_server
792 global_pid_ids  global_pid_ids     bag   0      279      global_name_server
793 global_pid_names global_pid_names   bag  782    11203    global_name_server
794 hm_ets_cache_table hm_ets_cache_table set   0      279      hm_cache_mgr
795 hm_table_global hm_table_global    bag  401    37973    <0.404.0>
796 inet_cache      inet_cache         bag   0      279      inet_db
797 inet_db         inet_db            set  21     541      inet_db
798 inet_hosts      inet_hosts         set   0      279      inet_db
799 sys_dist        sys_dist          set  64     3047     net_kernel
800 ===== netlab419@netlab4 =====
801 id              name              type  size  mem      owner
802 -----
803 10              cookies           set   0      279      auth
804 13              code              set  279    12706    code_server
805 14              code_names        set   48    4649    code_server
806 15              ign_requests      set   0      279      inet_gethost_native
807 16              ign_req_index     set   0      279      inet_gethost_native
808 ac_tab          ac_tab            set  20     1389     application_controller
809 file_io_servers file_io_servers    set   1      349      file_server_2
810 global_locks    global_locks       set   0      279      global_name_server
811 global_names    global_names       set  391    11239    global_name_server
812 global_names_ext global_names_ext   set   0      279      global_name_server
813 global_pid_ids  global_pid_ids     bag   0      279      global_name_server
814 global_pid_names global_pid_names   bag  782    11203    global_name_server
815 hm_ets_cache_table hm_ets_cache_table set   0      279      hm_cache_mgr

```

```

816 hm_table_global hm_table_global bag 1234 116275 <0.403.0>
817 inet_cache inet_cache bag 0 279 inet_db
818 inet_db inet_db set 21 541 inet_db
819 inet_hosts inet_hosts set 0 279 inet_db
820 sys_dist sys_dist set 64 3047 net_kernel
821 ===== netlab420@netlab4 =====
822 id name type size mem owner
823 -----
824 10 cookies set 0 279 auth
825 13 code set 279 12706 code_server
826 14 code_names set 48 4649 code_server
827 15 ign_requests set 0 279 inet_gethost_native
828 16 ign_req_index set 0 279 inet_gethost_native
829 ac_tab ac_tab set 20 1389 application_controller
830 file_io_servers file_io_servers set 1 349 file_server_2
831 global_locks global_locks set 0 279 global_name_server
832 global_names global_names set 391 11239 global_name_server
833 global_names_ext global_names_ext set 0 279 global_name_server
834 global_pid_ids global_pid_ids bag 0 279 global_name_server
835 global_pid_names global_pid_names bag 782 11203 global_name_server
836 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
837 hm_table_global hm_table_global bag 833 78581 <0.430.0>
838 inet_cache inet_cache bag 0 279 inet_db
839 inet_db inet_db set 21 541 inet_db
840 inet_hosts inet_hosts set 0 279 inet_db
841 sys_dist sys_dist set 64 3047 net_kernel
842 ===== dell1@dell =====
843 id name type size mem owner
844 -----
845 12 cookies set 0 286 auth
846 4111 code set 281 12978 code_server
847 8208 code_names set 56 7510 code_server
848 12305 ign_requests set 0 286 inet_gethost_native
849 16402 ign_req_index set 0 286 inet_gethost_native
850 ac_tab ac_tab set 20 1328 application_controller
851 file_io_servers file_io_servers set 1 339 file_server_2
852 global_locks global_locks set 0 286 global_name_server
853 global_names global_names set 391 10437 global_name_server
854 global_names_ext global_names_ext set 0 286 global_name_server
855 global_pid_ids global_pid_ids bag 0 286 global_name_server
856 global_pid_names global_pid_names bag 782 8864 global_name_server
857 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
858 hm_table_global hm_table_global bag 333 30589 <0.379.0>
859 inet_cache inet_cache bag 0 286 inet_db
860 inet_db inet_db set 29 553 inet_db
861 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
862 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
863 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
864 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
865 sys_dist sys_dist set 64 2868 net_kernel
866 ===== dell2@dell =====
867 id name type size mem owner
868 -----
869 12 cookies set 0 286 auth
870 4111 code set 281 12978 code_server
871 8208 code_names set 56 7510 code_server
872 12305 ign_requests set 0 286 inet_gethost_native
873 16402 ign_req_index set 0 286 inet_gethost_native
874 ac_tab ac_tab set 20 1328 application_controller
875 file_io_servers file_io_servers set 1 339 file_server_2

```

```

876 global_locks      global_locks      set    0      286      global_name_server
877 global_names      global_names      set   391    10457    global_name_server
878 global_names_ext   global_names_ext   set    0      286      global_name_server
879 global_pid_ids     global_pid_ids     bag    0      286      global_name_server
880 global_pid_names   global_pid_names   bag   782    8864     global_name_server
881 hm_ets_cache_table hm_ets_cache_table set    0      286      hm_cache_mgr
882 hm_table_global    hm_table_global    bag   466   42692    <0.360.0>
883 inet_cache         inet_cache         bag    0      286      inet_db
884 inet_db            inet_db            set   29     553      inet_db
885 inet_hosts_byaddr  inet_hosts_byaddr  bag    0      286      inet_db
886 inet_hosts_byname  inet_hosts_byname  bag    0      286      inet_db
887 inet_hosts_file_byaddr inet_hosts_file_byaddr bag    0      286      inet_db
888 inet_hosts_file_byname inet_hosts_file_byname bag    0      286      inet_db
889 sys_dist           sys_dist           set   64    2868     net_kernel
890 ===== dell13@dell =====
891 id                 name                 type  size  mem  owner
892 -----
893 12                 cookies              set   0     286  auth
894 4111               code                 set  281   12978 code_server
895 8208               code_names           set   56   7510  code_server
896 12305              ign_requests         set   0     286  inet_gethost_native
897 16402              ign_req_index        set   0     286  inet_gethost_native
898 ac_tab             ac_tab               set   20   1328  application_controller
899 file_io_servers    file_io_servers      set    1    339  file_server_2
900 global_locks       global_locks         set   0     286  global_name_server
901 global_names       global_names         set  391   10105 global_name_server
902 global_names_ext   global_names_ext     set   0     286  global_name_server
903 global_pid_ids     global_pid_ids       bag    0     286  global_name_server
904 global_pid_names   global_pid_names     bag   782   8864  global_name_server
905 hm_ets_cache_table hm_ets_cache_table  set    0     286  hm_cache_mgr
906 hm_table_global    hm_table_global      bag  507   46423 <0.476.0>
907 inet_cache         inet_cache           bag    0     286  inet_db
908 inet_db            inet_db              set   29     553  inet_db
909 inet_hosts_byaddr  inet_hosts_byaddr    bag    0     286  inet_db
910 inet_hosts_byname  inet_hosts_byname    bag    0     286  inet_db
911 inet_hosts_file_byaddr inet_hosts_file_byaddr bag    0     286  inet_db
912 inet_hosts_file_byname inet_hosts_file_byname bag    0     286  inet_db
913 sys_dist           sys_dist             set   64    2868  net_kernel
914 ===== dell14@dell =====
915 id                 name                 type  size  mem  owner
916 -----
917 12                 cookies              set   0     286  auth
918 4111               code                 set  281   12978 code_server
919 8208               code_names           set   56   7510  code_server
920 12305              ign_requests         set   0     286  inet_gethost_native
921 16402              ign_req_index        set   0     286  inet_gethost_native
922 ac_tab             ac_tab               set   20   1328  application_controller
923 file_io_servers    file_io_servers      set    1    339  file_server_2
924 global_locks       global_locks         set   0     286  global_name_server
925 global_names       global_names         set  391   10441 global_name_server
926 global_names_ext   global_names_ext     set   0     286  global_name_server
927 global_pid_ids     global_pid_ids       bag    0     286  global_name_server
928 global_pid_names   global_pid_names     bag   782   8864  global_name_server
929 hm_ets_cache_table hm_ets_cache_table  set    0     286  hm_cache_mgr
930 hm_table_global    hm_table_global      bag 1545  140881 <0.370.0>
931 inet_cache         inet_cache           bag    0     286  inet_db
932 inet_db            inet_db              set   29     553  inet_db
933 inet_hosts_byaddr  inet_hosts_byaddr    bag    0     286  inet_db
934 inet_hosts_byname  inet_hosts_byname    bag    0     286  inet_db
935 inet_hosts_file_byaddr inet_hosts_file_byaddr bag    0     286  inet_db

```

```

936 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
937 sys_dist sys_dist set 64 2868 net_kernel
938 ===== dell5@dell =====
939 id name type size mem owner
940 -----
941 12 cookies set 0 286 auth
942 4111 code set 281 12978 code_server
943 8208 code_names set 56 7510 code_server
944 12305 ign_requests set 0 286 inet_gethost_native
945 16402 ign_req_index set 0 286 inet_gethost_native
946 ac_tab ac_tab set 20 1328 application_controller
947 file_io_servers file_io_servers set 1 339 file_server_2
948 global_locks global_locks set 0 286 global_name_server
949 global_names global_names set 391 10121 global_name_server
950 global_names_ext global_names_ext set 0 286 global_name_server
951 global_pid_ids global_pid_ids bag 0 286 global_name_server
952 global_pid_names global_pid_names bag 782 8864 global_name_server
953 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
954 hm_table_global hm_table_global bag 912 83278 <0.476.0>
955 inet_cache inet_cache bag 0 286 inet_db
956 inet_db inet_db set 29 553 inet_db
957 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
958 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
959 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
960 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
961 sys_dist sys_dist set 64 2868 net_kernel
962 ===== dell6@dell =====
963 id name type size mem owner
964 -----
965 12 cookies set 0 286 auth
966 4111 code set 281 12978 code_server
967 8208 code_names set 56 7510 code_server
968 12305 ign_requests set 0 286 inet_gethost_native
969 16402 ign_req_index set 0 286 inet_gethost_native
970 ac_tab ac_tab set 20 1328 application_controller
971 file_io_servers file_io_servers set 1 339 file_server_2
972 global_locks global_locks set 0 286 global_name_server
973 global_names global_names set 391 10413 global_name_server
974 global_names_ext global_names_ext set 0 286 global_name_server
975 global_pid_ids global_pid_ids bag 0 286 global_name_server
976 global_pid_names global_pid_names bag 782 8864 global_name_server
977 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
978 hm_table_global hm_table_global bag 497 45513 <0.399.0>
979 inet_cache inet_cache bag 0 286 inet_db
980 inet_db inet_db set 29 553 inet_db
981 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
982 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
983 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
984 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
985 sys_dist sys_dist set 64 2868 net_kernel
986 ===== dell7@dell =====
987 id name type size mem owner
988 -----
989 12 cookies set 0 286 auth
990 4111 code set 281 12978 code_server
991 8208 code_names set 56 7510 code_server
992 12305 ign_requests set 0 286 inet_gethost_native
993 16402 ign_req_index set 0 286 inet_gethost_native
994 ac_tab ac_tab set 20 1328 application_controller
995 file_io_servers file_io_servers set 1 339 file_server_2

```

```

996 global_locks      global_locks      set    0      286      global_name_server
997 global_names      global_names      set   391    10441    global_name_server
998 global_names_ext  global_names_ext  set    0      286      global_name_server
999 global_pid_ids    global_pid_ids    bag    0      286      global_name_server
1000 global_pid_names  global_pid_names  bag   782    8864     global_name_server
1001 hm_ets_cache_table hm_ets_cache_table set    0      286      hm_cache_mgr
1002 hm_table_global   hm_table_global   bag   390    35776    <0.372.0>
1003 inet_cache        inet_cache        bag    0      286      inet_db
1004 inet_db           inet_db           set   29     553      inet_db
1005 inet_hosts_byaddr inet_hosts_byaddr bag    0      286      inet_db
1006 inet_hosts_byname inet_hosts_byname bag    0      286      inet_db
1007 inet_hosts_file_byaddr inet_hosts_file_byaddr bag    0      286      inet_db
1008 inet_hosts_file_byname inet_hosts_file_byname bag    0      286      inet_db
1009 sys_dist         sys_dist         set   64     2868     net_kernel
1010 ===== dell18@dell =====
1011 id                name                type  size  mem  owner
1012 -----
1013 12                cookies             set    0     286  auth
1014 4111              code                set   281   12978  code_server
1015 8208              code_names          set    56   7510  code_server
1016 12305             ign_requests        set    0     286  inet_gethost_native
1017 16402             ign_req_index       set    0     286  inet_gethost_native
1018 ac_tab            ac_tab              set   20   1328  application_controller
1019 file_io_servers   file_io_servers     set    1     339  file_server_2
1020 global_locks      global_locks        set    0     286  global_name_server
1021 global_names      global_names        set   391   10441  global_name_server
1022 global_names_ext  global_names_ext    set    0     286  global_name_server
1023 global_pid_ids    global_pid_ids      bag    0     286  global_name_server
1024 global_pid_names  global_pid_names    bag   782   8864   global_name_server
1025 hm_ets_cache_table hm_ets_cache_table set    0     286  hm_cache_mgr
1026 hm_table_global   hm_table_global     bag   228   21034  <0.378.0>
1027 inet_cache        inet_cache          bag    0     286  inet_db
1028 inet_db           inet_db             set   29     553  inet_db
1029 inet_hosts_byaddr inet_hosts_byaddr   bag    0     286  inet_db
1030 inet_hosts_byname inet_hosts_byname   bag    0     286  inet_db
1031 inet_hosts_file_byaddr inet_hosts_file_byaddr bag    0     286  inet_db
1032 inet_hosts_file_byname inet_hosts_file_byname bag    0     286  inet_db
1033 sys_dist         sys_dist           set   64     2868  net_kernel
1034 ===== netlab1@netlaba =====
1035 id                name                type  size  mem  owner
1036 -----
1037 9                 cookies             set    0     284  auth
1038 4108              code                set   278   12787  code_server
1039 8205              code_names          set    54   5196  code_server
1040 12302             ign_requests        set    0     284  inet_gethost_native
1041 16399             ign_req_index       set    0     284  inet_gethost_native
1042 ac_tab            ac_tab              set   20   1380  application_controller
1043 file_io_servers   file_io_servers     set    1     352  file_server_2
1044 global_locks      global_locks        set    0     284  global_name_server
1045 global_names      global_names        set   391   10932  global_name_server
1046 global_names_ext  global_names_ext    set    0     284  global_name_server
1047 global_pid_ids    global_pid_ids      bag    0     284  global_name_server
1048 global_pid_names  global_pid_names    bag   782   11208  global_name_server
1049 hm_ets_cache_table hm_ets_cache_table set    0     284  hm_cache_mgr
1050 hm_table_global   hm_table_global     bag   503   47566  <0.562.0>
1051 inet_cache        inet_cache          bag    0     284  inet_db
1052 inet_db           inet_db             set   21     512  inet_db
1053 inet_hosts        inet_hosts          set    0     284  inet_db
1054 sys_dist         sys_dist           set   64    3052  net_kernel
1055 ===== netlaba2@netlaba =====

```



```

1056 id          name          type  size  mem  owner
1057 -----
1058 9             cookies        set    0    284  auth
1059 4108          code          set   278  12787 code_server
1060 8205          code_names       set    54  5196  code_server
1061 12302         ign_requests     set     0  284  inet_gethost_native
1062 16399         ign_req_index     set     0  284  inet_gethost_native
1063 ac_tab        ac_tab           set    20  1380  application_controller
1064 file_io_servers file_io_servers set     1   352  file_server_2
1065 global_locks  global_locks     set     0  284  global_name_server
1066 global_names  global_names     set   391  10932 global_name_server
1067 global_names_ext global_names_ext set     0   284  global_name_server
1068 global_pid_ids global_pid_ids   bag     0  284  global_name_server
1069 global_pid_names global_pid_names bag    782  11208 global_name_server
1070 hm_ets_cache_table hm_ets_cache_table set     0   284  hm_cache_mgr
1071 hm_table_global hm_table_global bag   1476  139028 <0.524.0>
1072 inet_cache    inet_cache       bag     0  284  inet_db
1073 inet_db       inet_db          set    21  512  inet_db
1074 inet_hosts    inet_hosts       set     0  284  inet_db
1075 sys_dist      sys_dist         set    64  3052  net_kernel
1076 ===== netlaba3@netlaba =====
1077 id          name          type  size  mem  owner
1078 -----
1079 9             cookies        set    0    284  auth
1080 4108          code          set   278  12787 code_server
1081 8205          code_names       set    54  5196  code_server
1082 12302         ign_requests     set     0  284  inet_gethost_native
1083 16399         ign_req_index     set     0  284  inet_gethost_native
1084 ac_tab        ac_tab           set    20  1380  application_controller
1085 file_io_servers file_io_servers set     1   352  file_server_2
1086 global_locks  global_locks     set     0  284  global_name_server
1087 global_names  global_names     set   391  10872 global_name_server
1088 global_names_ext global_names_ext set     0   284  global_name_server
1089 global_pid_ids global_pid_ids   bag     0  284  global_name_server
1090 global_pid_names global_pid_names bag    782  11208 global_name_server
1091 hm_ets_cache_table hm_ets_cache_table set     0   284  hm_cache_mgr
1092 hm_table_global hm_table_global bag    623  58846 <0.564.0>
1093 inet_cache    inet_cache       bag     0  284  inet_db
1094 inet_db       inet_db          set    21  512  inet_db
1095 inet_hosts    inet_hosts       set     0  284  inet_db
1096 sys_dist      sys_dist         set    64  3052  net_kernel
1097 ===== netlaba4@netlaba =====
1098 id          name          type  size  mem  owner
1099 -----
1100 9             cookies        set    0    284  auth
1101 4108          code          set   278  12787 code_server
1102 8205          code_names       set    54  5196  code_server
1103 12302         ign_requests     set     0  284  inet_gethost_native
1104 16399         ign_req_index     set     0  284  inet_gethost_native
1105 ac_tab        ac_tab           set    20  1380  application_controller
1106 file_io_servers file_io_servers set     1   352  file_server_2
1107 global_locks  global_locks     set     0  284  global_name_server
1108 global_names  global_names     set   391  10932 global_name_server
1109 global_names_ext global_names_ext set     0   284  global_name_server
1110 global_pid_ids global_pid_ids   bag     0  284  global_name_server
1111 global_pid_names global_pid_names bag    782  11208 global_name_server
1112 hm_ets_cache_table hm_ets_cache_table set     0   284  hm_cache_mgr
1113 hm_table_global hm_table_global bag    715  67494 <0.521.0>
1114 inet_cache    inet_cache       bag     0  284  inet_db
1115 inet_db       inet_db          set    21  512  inet_db

```

```

1116 inet_hosts      inet_hosts      set  0      284      inet_db
1117 sys_dist        sys_dist        set  64     3052     net_kernel
1118 ===== netlaba5@netlaba =====
1119 id              name              type  size  mem      owner
1120 -----
1121 9               cookies           set   0      284      auth
1122 4108            code             set  278    12787    code_server
1123 8205            code_names       set   54     5196     code_server
1124 12302           ign_requests     set   0      284      inet_gethost_native
1125 16399           ign_req_index    set   0      284      inet_gethost_native
1126 ac_tab          ac_tab           set  20     1380     application_controller
1127 file_io_servers file_io_servers  set   1      352      file_server_2
1128 global_locks    global_locks     set   0      284      global_name_server
1129 global_names    global_names     set  391    10932    global_name_server
1130 global_names_ext global_names_ext set   0      284      global_name_server
1131 global_pid_ids  global_pid_ids   bag   0      284      global_name_server
1132 global_pid_names global_pid_names bag  782    11208    global_name_server
1133 hm_ets_cache_table hm_ets_cache_table set  0      284      hm_cache_mgr
1134 hm_table_global hm_table_global  bag  824    77740    <0.515.0>
1135 inet_cache      inet_cache       bag   0      284      inet_db
1136 inet_db         inet_db          set  21     512      inet_db
1137 inet_hosts      inet_hosts      set   0      284      inet_db
1138 sys_dist        sys_dist        set  64     3052     net_kernel
1139 ===== netlaba6@netlaba =====
1140 id              name              type  size  mem      owner
1141 -----
1142 9               cookies           set   0      284      auth
1143 4108            code             set  278    12787    code_server
1144 8205            code_names       set   54     5196     code_server
1145 12302           ign_requests     set   0      284      inet_gethost_native
1146 16399           ign_req_index    set   0      284      inet_gethost_native
1147 ac_tab          ac_tab           set  20     1380     application_controller
1148 file_io_servers file_io_servers  set   1      352      file_server_2
1149 global_locks    global_locks     set   0      284      global_name_server
1150 global_names    global_names     set  391    10932    global_name_server
1151 global_names_ext global_names_ext set   0      284      global_name_server
1152 global_pid_ids  global_pid_ids   bag   0      284      global_name_server
1153 global_pid_names global_pid_names bag  782    11208    global_name_server
1154 hm_ets_cache_table hm_ets_cache_table set  0      284      hm_cache_mgr
1155 hm_table_global hm_table_global  bag  824    77740    <0.520.0>
1156 inet_cache      inet_cache       bag   0      284      inet_db
1157 inet_db         inet_db          set  21     512      inet_db
1158 inet_hosts      inet_hosts      set   0      284      inet_db
1159 sys_dist        sys_dist        set  64     3052     net_kernel
1160 ===== netlaba7@netlaba =====
1161 id              name              type  size  mem      owner
1162 -----
1163 9               cookies           set   0      284      auth
1164 4108            code             set  278    12787    code_server
1165 8205            code_names       set   54     5196     code_server
1166 12302           ign_requests     set   0      284      inet_gethost_native
1167 16399           ign_req_index    set   0      284      inet_gethost_native
1168 ac_tab          ac_tab           set  20     1380     application_controller
1169 file_io_servers file_io_servers  set   1      352      file_server_2
1170 global_locks    global_locks     set   0      284      global_name_server
1171 global_names    global_names     set  391    10920    global_name_server
1172 global_names_ext global_names_ext set   0      284      global_name_server
1173 global_pid_ids  global_pid_ids   bag   0      284      global_name_server
1174 global_pid_names global_pid_names bag  782    11208    global_name_server
1175 hm_ets_cache_table hm_ets_cache_table set  0      284      hm_cache_mgr

```

```

1176 hm_table_global hm_table_global bag 597 56402 <0.528.0>
1177 inet_cache inet_cache bag 0 284 inet_db
1178 inet_db inet_db set 21 512 inet_db
1179 inet_hosts inet_hosts set 0 284 inet_db
1180 sys_dist sys_dist set 64 3052 net_kernel
1181 ===== netlaba8@netlaba =====
1182 id name type size mem owner
1183 -----
1184 9 cookies set 0 284 auth
1185 4108 code set 278 12787 code_server
1186 8205 code_names set 54 5196 code_server
1187 12302 ign_requests set 0 284 inet_gethost_native
1188 16399 ign_req_index set 0 284 inet_gethost_native
1189 ac_tab ac_tab set 20 1380 application_controller
1190 file_io_servers file_io_servers set 1 352 file_server_2
1191 global_locks global_locks set 0 284 global_name_server
1192 global_names global_names set 391 11020 global_name_server
1193 global_names_ext global_names_ext set 0 284 global_name_server
1194 global_pid_ids global_pid_ids bag 0 284 global_name_server
1195 global_pid_names global_pid_names bag 782 11208 global_name_server
1196 hm_ets_cache_table hm_ets_cache_table set 0 284 hm_cache_mgr
1197 hm_table_global hm_table_global bag 551 52078 <0.446.0>
1198 inet_cache inet_cache bag 0 284 inet_db
1199 inet_db inet_db set 21 512 inet_db
1200 inet_hosts inet_hosts set 0 284 inet_db
1201 sys_dist sys_dist set 64 3052 net_kernel
1202 ===== netlabbl@netlabbb =====
1203 id name type size mem owner
1204 -----
1205 12 cookies set 0 286 auth
1206 4111 code set 281 12978 code_server
1207 8208 code_names set 56 7510 code_server
1208 12305 ign_requests set 0 286 inet_gethost_native
1209 16402 ign_req_index set 0 286 inet_gethost_native
1210 ac_tab ac_tab set 21 1339 application_controller
1211 file_io_servers file_io_servers set 1 351 file_server_2
1212 global_locks global_locks set 0 286 global_name_server
1213 global_names global_names set 391 9609 global_name_server
1214 global_names_ext global_names_ext set 0 286 global_name_server
1215 global_pid_ids global_pid_ids bag 0 286 global_name_server
1216 global_pid_names global_pid_names bag 782 8864 global_name_server
1217 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1218 hm_table_global hm_table_global bag 1255 114491 <0.639.0>
1219 inet_cache inet_cache bag 0 286 inet_db
1220 inet_db inet_db set 29 559 inet_db
1221 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1222 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1223 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1224 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1225 sys_dist sys_dist set 64 2862 net_kernel
1226 ===== netlabbb2@netlabbb =====
1227 id name type size mem owner
1228 -----
1229 12 cookies set 0 286 auth
1230 4111 code set 281 12978 code_server
1231 8208 code_names set 56 7510 code_server
1232 12305 ign_requests set 0 286 inet_gethost_native
1233 16402 ign_req_index set 0 286 inet_gethost_native
1234 ac_tab ac_tab set 21 1339 application_controller
1235 file_io_servers file_io_servers set 1 351 file_server_2

```

```

1236 global_locks      global_locks      set    0      286      global_name_server
1237 global_names      global_names      set   391    9657    global_name_server
1238 global_names_ext   global_names_ext   set    0      286      global_name_server
1239 global_pid_ids     global_pid_ids     bag    0      286      global_name_server
1240 global_pid_names   global_pid_names   bag   782    8864    global_name_server
1241 hm_ets_cache_table hm_ets_cache_table set    0      286      hm_cache_mgr
1242 hm_table_global    hm_table_global    bag   510   46696    <0.591.0>
1243 inet_cache         inet_cache         bag    0      286      inet_db
1244 inet_db            inet_db            set   29     559      inet_db
1245 inet_hosts_byaddr  inet_hosts_byaddr  bag    0      286      inet_db
1246 inet_hosts_byname  inet_hosts_byname  bag    0      286      inet_db
1247 inet_hosts_file_byaddr inet_hosts_file_byaddr bag    0      286      inet_db
1248 inet_hosts_file_byname inet_hosts_file_byname bag    0      286      inet_db
1249 sys_dist           sys_dist           set   64    2862    net_kernel
1250 ===== netlab3@netlab3 =====
1251 id                 name                 type  size  mem  owner
1252 -----
1253 12                 cookies              set    0    286  auth
1254 4111               code                 set   281  12978 code_server
1255 8208               code_names           set    56  7510  code_server
1256 12305              ign_requests         set    0    286  inet_gethost_native
1257 16402              ign_req_index        set    0    286  inet_gethost_native
1258 ac_tab             ac_tab               set   21   1339  application_controller
1259 file_io_servers    file_io_servers      set    1    351  file_server_2
1260 global_locks       global_locks         set    0    286  global_name_server
1261 global_names       global_names         set   391  9697  global_name_server
1262 global_names_ext   global_names_ext     set    0    286  global_name_server
1263 global_pid_ids     global_pid_ids       bag    0    286  global_name_server
1264 global_pid_names   global_pid_names     bag   782  8864  global_name_server
1265 hm_ets_cache_table hm_ets_cache_table  set    0    286  hm_cache_mgr
1266 hm_table_global    hm_table_global      bag   797  72813 <0.570.0>
1267 inet_cache         inet_cache           bag    0    286  inet_db
1268 inet_db            inet_db              set   29     559  inet_db
1269 inet_hosts_byaddr  inet_hosts_byaddr    bag    0    286  inet_db
1270 inet_hosts_byname  inet_hosts_byname    bag    0    286  inet_db
1271 inet_hosts_file_byaddr inet_hosts_file_byaddr bag    0    286  inet_db
1272 inet_hosts_file_byname inet_hosts_file_byname bag    0    286  inet_db
1273 sys_dist           sys_dist             set   64    2862  net_kernel
1274 ===== netlab4@netlab4 =====
1275 id                 name                 type  size  mem  owner
1276 -----
1277 12                 cookies              set    0    286  auth
1278 4111               code                 set   281  12978 code_server
1279 8208               code_names           set    56  7510  code_server
1280 12305              ign_requests         set    0    286  inet_gethost_native
1281 16402              ign_req_index        set    0    286  inet_gethost_native
1282 ac_tab             ac_tab               set   21   1339  application_controller
1283 file_io_servers    file_io_servers      set    1    351  file_server_2
1284 global_locks       global_locks         set    0    286  global_name_server
1285 global_names       global_names         set   391  9697  global_name_server
1286 global_names_ext   global_names_ext     set    0    286  global_name_server
1287 global_pid_ids     global_pid_ids       bag    0    286  global_name_server
1288 global_pid_names   global_pid_names     bag   782  8864  global_name_server
1289 hm_ets_cache_table hm_ets_cache_table  set    0    286  hm_cache_mgr
1290 hm_table_global    hm_table_global      bag  1017  92833 <0.604.0>
1291 inet_cache         inet_cache           bag    0    286  inet_db
1292 inet_db            inet_db              set   29     559  inet_db
1293 inet_hosts_byaddr  inet_hosts_byaddr    bag    0    286  inet_db
1294 inet_hosts_byname  inet_hosts_byname    bag    0    286  inet_db
1295 inet_hosts_file_byaddr inet_hosts_file_byaddr bag    0    286  inet_db

```

```

1296 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1297 sys_dist sys_dist set 64 2862 net_kernel
1298 ===== netlab5@netlab5 =====
1299 id name type size mem owner
1300 -----
1301 12 cookies set 0 286 auth
1302 4111 code set 281 12978 code_server
1303 8208 code_names set 56 7510 code_server
1304 12305 ign_requests set 0 286 inet_gethost_native
1305 16402 ign_req_index set 0 286 inet_gethost_native
1306 ac_tab ac_tab set 21 1339 application_controller
1307 file_io_servers file_io_servers set 1 351 file_server_2
1308 global_locks global_locks set 0 286 global_name_server
1309 global_names global_names set 391 9797 global_name_server
1310 global_names_ext global_names_ext set 0 286 global_name_server
1311 global_pid_ids global_pid_ids bag 0 286 global_name_server
1312 global_pid_names global_pid_names bag 782 8864 global_name_server
1313 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1314 hm_table_global hm_table_global bag 243 22399 <0.541.0>
1315 inet_cache inet_cache bag 0 286 inet_db
1316 inet_db inet_db set 29 559 inet_db
1317 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1318 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1319 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1320 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1321 sys_dist sys_dist set 64 2862 net_kernel
1322 ===== netlab6@netlab6 =====
1323 id name type size mem owner
1324 -----
1325 12 cookies set 0 286 auth
1326 4111 code set 281 12978 code_server
1327 8208 code_names set 56 7510 code_server
1328 12305 ign_requests set 0 286 inet_gethost_native
1329 16402 ign_req_index set 0 286 inet_gethost_native
1330 ac_tab ac_tab set 21 1339 application_controller
1331 file_io_servers file_io_servers set 1 351 file_server_2
1332 global_locks global_locks set 0 286 global_name_server
1333 global_names global_names set 391 9677 global_name_server
1334 global_names_ext global_names_ext set 0 286 global_name_server
1335 global_pid_ids global_pid_ids bag 0 286 global_name_server
1336 global_pid_names global_pid_names bag 782 8864 global_name_server
1337 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1338 hm_table_global hm_table_global bag 987 90103 <0.595.0>
1339 inet_cache inet_cache bag 0 286 inet_db
1340 inet_db inet_db set 29 559 inet_db
1341 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1342 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1343 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1344 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1345 sys_dist sys_dist set 64 2862 net_kernel
1346 ===== netlab7@netlab7 =====
1347 id name type size mem owner
1348 -----
1349 12 cookies set 0 286 auth
1350 4111 code set 281 12978 code_server
1351 8208 code_names set 56 7510 code_server
1352 12305 ign_requests set 0 286 inet_gethost_native
1353 16402 ign_req_index set 0 286 inet_gethost_native
1354 ac_tab ac_tab set 21 1339 application_controller
1355 file_io_servers file_io_servers set 1 351 file_server_2

```

```

1356 global_locks      global_locks      set    0      286      global_name_server
1357 global_names      global_names      set   391    9677    global_name_server
1358 global_names_ext  global_names_ext  set    0      286      global_name_server
1359 global_pid_ids    global_pid_ids    bag    0      286      global_name_server
1360 global_pid_names  global_pid_names  bag   782   8864    global_name_server
1361 hm_ets_cache_table hm_ets_cache_table set    0      286      hm_cache_mgr
1362 hm_table_global   hm_table_global   bag   885   80821   <0.596.0>
1363 inet_cache        inet_cache        bag    0      286      inet_db
1364 inet_db           inet_db           set   29     559     inet_db
1365 inet_hosts_byaddr inet_hosts_byaddr bag    0      286      inet_db
1366 inet_hosts_byname inet_hosts_byname bag    0      286      inet_db
1367 inet_hosts_file_byaddr inet_hosts_file_byaddr bag    0      286      inet_db
1368 inet_hosts_file_byname inet_hosts_file_byname bag    0      286      inet_db
1369 sys_dist          sys_dist          set   64    2862    net_kernel
1370 ===== netlab38@netlab3 =====
1371 id                name                type  size  mem  owner
1372 -----
1373 12                cookies             set    0    286  auth
1374 4111              code                set   281  12978 code_server
1375 8208              code_names          set    56  7510  code_server
1376 12305             ign_requests        set    0    286  inet_gethost_native
1377 16402             ign_req_index        set    0    286  inet_gethost_native
1378 ac_tab            ac_tab              set   21   1339  application_controller
1379 file_io_servers   file_io_servers     set    1    351  file_server_2
1380 global_locks      global_locks        set    0    286  global_name_server
1381 global_names      global_names        set   391  9669  global_name_server
1382 global_names_ext  global_names_ext    set    0    286  global_name_server
1383 global_pid_ids    global_pid_ids      bag    0    286  global_name_server
1384 global_pid_names  global_pid_names    bag   782  8864  global_name_server
1385 hm_ets_cache_table hm_ets_cache_table set    0    286  hm_cache_mgr
1386 hm_table_global   hm_table_global     bag   935  85371 <0.586.0>
1387 inet_cache        inet_cache          bag    0    286  inet_db
1388 inet_db           inet_db             set   29     559  inet_db
1389 inet_hosts_byaddr inet_hosts_byaddr   bag    0    286  inet_db
1390 inet_hosts_byname inet_hosts_byname   bag    0    286  inet_db
1391 inet_hosts_file_byaddr inet_hosts_file_byaddr bag    0    286  inet_db
1392 inet_hosts_file_byname inet_hosts_file_byname bag    0    286  inet_db
1393 sys_dist          sys_dist            set   64    2862  net_kernel
1394 ok
1395 (xxx_node@netlab3)7>

```

List 49: Test Results:Memory Usage:rstore function with short data

```

1 hirotnk@netlab3 ~/harmonia/Harmonia $ test/test20.sh -s
2 Launched the node:[netlab31@netlab3]
3 Launched the node:[netlab32@netlab3]
4 Launched the node:[netlab33@netlab3]
5 Launched the node:[netlab34@netlab3]
6 Launched the node:[netlab35@netlab3]
7 Launched the node:[netlab36@netlab3]
8 Launched the node:[netlab37@netlab3]
9 Launched the node:[netlab38@netlab3]
10 Launched the node:[netlab39@netlab3]
11 Launched the node:[netlab310@netlab3]
12 Launched the node:[netlab311@netlab3]
13 Launched the node:[netlab312@netlab3]
14 Launched the node:[netlab313@netlab3]
15 Launched the node:[netlab314@netlab3]
16 Launched the node:[netlab315@netlab3]
17 Launched the node:[netlab316@netlab3]

```

```

18 Launched the node:[netlab317@netlab3]
19 Launched the node:[netlab318@netlab3]
20 Launched the node:[netlab319@netlab3]
21 Launched the node:[netlab320@netlab3]
22 epmd: up and running on port 4369 with data:
23 name netlab318 at port 41230
24 name netlab317 at port 60602
25 name netlab315 at port 45201
26 name netlab316 at port 46023
27 name netlab314 at port 41235
28 name netlab313 at port 49597
29 name netlab312 at port 36392
30 name netlab311 at port 44408
31 name netlab310 at port 58577
32 name netlab38 at port 37367
33 name netlab36 at port 56002
34 name netlab39 at port 59533
35 name netlab37 at port 33769
36 name netlab35 at port 58130
37 name netlab34 at port 38391
38 name netlab33 at port 52165
39 name netlab32 at port 59215
40 name netlab31 at port 59603
41 hirotnk@netlab3 ~/harmonia/Harmonia $ erl -pa ebin -config harmonia -run hm_edge start -\
    →harmonia node_type 'join' name xxx sname 'xxx_node@netlab3' root_node 'netlab31@netlab3' \
    →setcookie harmonia_cookie -sname 'xxx_node@netlab3'
42 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
43
44 Eshell V5.6.3 (abort with ^G)
45 (xxx_node@netlab3)1> [{name,xxx},
46 {root_node,netlab31@netlab3},
47 {root,netlab31},
48 {included_applications,[],},
49 {sname,xxx_node@netlab3},
50 {logfile_ext, ".txt"},
51 {node_type,join},
52 {logdir,"log/"},
53 {logfile,"harmonia_log"}]
54 "log/harmonia_log_xxx.txt"
55 start Pid:[<0.259.0>]
56
57 (xxx_node@netlab3)1> hm_cli:log_stop().
58 log stop:[dell2@dell] Result:[ok]
59 log stop:[dell8@dell] Result:[ok]
60 log stop:[netlab32@netlab3] Result:[ok]
61 log stop:[dell6@dell] Result:[ok]
62 log stop:[dell5@dell] Result:[ok]
63 log stop:[dell1@dell] Result:[ok]
64 log stop:[dell4@dell] Result:[ok]
65 log stop:[netlab38@netlab3] Result:[ok]
66 log stop:[dell3@dell] Result:[ok]
67 log stop:[dell7@dell] Result:[ok]
68 log stop:[netlab33@netlab3] Result:[ok]
69 log stop:[netlab35@netlab3] Result:[ok]
70 log stop:[netlab36@netlab3] Result:[ok]
71 log stop:[netlab31@netlab3] Result:[ok]
72 log stop:[netlab34@netlab3] Result:[ok]
73 log stop:[netlab37@netlab3] Result:[ok]
74 log stop:[netlab39@netlab3] Result:[ok]
75 log stop:[netlab32@netlab3] Result:[ok]

```

```

76 log stop:[netlaba2@netlaba] Result:[ok]
77 log stop:[netlaba4@netlaba] Result:[ok]
78 log stop:[netlaba6@netlaba] Result:[ok]
79 log stop:[netlaba7@netlaba] Result:[ok]
80 log stop:[netlaba8@netlaba] Result:[ok]
81 log stop:[netlabal@netlaba] Result:[ok]
82 log stop:[netlab49@netlab4] Result:[ok]
83 log stop:[netlab417@netlab4] Result:[ok]
84 log stop:[netlab412@netlab4] Result:[ok]
85 log stop:[netlab413@netlab4] Result:[ok]
86 log stop:[netlab46@netlab4] Result:[ok]
87 log stop:[netlab416@netlab4] Result:[ok]
88 log stop:[netlab420@netlab4] Result:[ok]
89 log stop:[netlab410@netlab4] Result:[ok]
90 log stop:[netlab47@netlab4] Result:[ok]
91 log stop:[netlab419@netlab4] Result:[ok]
92 log stop:[netlab418@netlab4] Result:[ok]
93 log stop:[netlab411@netlab4] Result:[ok]
94 log stop:[netlab415@netlab4] Result:[ok]
95 log stop:[netlab414@netlab4] Result:[ok]
96 log stop:[netlab41@netlab4] Result:[ok]
97 log stop:[netlab48@netlab4] Result:[ok]
98 log stop:[netlab44@netlab4] Result:[ok]
99 log stop:[netlab45@netlab4] Result:[ok]
100 log stop:[netlab42@netlab4] Result:[ok]
101 log stop:[netlab43@netlab4] Result:[ok]
102 hm_log_h_file:closing 'log/harmonia_log_xxx.txt'
103 log stop:[xxx_node@netlab3] Result:[ok]
104 log stop:[netlab319@netlab3] Result:[ok]
105 log stop:[netlab37@netlab3] Result:[ok]
106 log stop:[netlab317@netlab3] Result:[ok]
107 log stop:[netlab38@netlab3] Result:[ok]
108 log stop:[netlab315@netlab3] Result:[ok]
109 log stop:[netlab36@netlab3] Result:[ok]
110 log stop:[netlab312@netlab3] Result:[ok]
111 log stop:[netlab318@netlab3] Result:[ok]
112 log stop:[netlab310@netlab3] Result:[ok]
113 log stop:[netlab320@netlab3] Result:[ok]
114 log stop:[netlab316@netlab3] Result:[ok]
115 log stop:[netlab314@netlab3] Result:[ok]
116 log stop:[netlab313@netlab3] Result:[ok]
117 log stop:[netlab34@netlab3] Result:[ok]
118 log stop:[netlab311@netlab3] Result:[ok]
119 log stop:[netlab35@netlab3] Result:[ok]
120 log stop:[netlab39@netlab3] Result:[ok]
121 log stop:[netlab33@netlab3] Result:[ok]
122 log stop:[netlab32@netlab3] Result:[ok]
123 log stop:[netlab31@netlab3] Result:[ok]
124 ok
125 (xxx_node@netlab3)2> hm_cli_test:test_all_short(10000).
126 starting....
127 create_table_short "OK"....
128 rstore_short(10000)      OK....
129 ..end
130 ok
131 (xxx_node@netlab3)3> hm_cli_test:check_size().
132 ===== netlab31@netlab3 =====
133 id            name                type  size  mem    owner
134 -----
135 10            cookies                set   0     279    auth

```



```

136 13          code          set 278 12598 code_server
137 14          code_names    set 48 4649 code_server
138 ac_tab      ac_tab        set 20 1389 application_controller
139 file_io_servers file_io_servers set 1 347 file_server_2
140 global_locks global_locks set 0 279 global_name_server
141 global_names global_names set 391 11843 global_name_server
142 global_names_ext global_names_ext set 0 279 global_name_server
143 global_pid_ids global_pid_ids bag 0 279 global_name_server
144 global_pid_names global_pid_names bag 782 11199 global_name_server
145 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
146 hm_table_global hm_table_global bag 535 39869 <0.55.0>
147 inet_cache     inet_cache     bag 0 279 inet_db
148 inet_db        inet_db        set 21 541 inet_db
149 inet_hosts     inet_hosts     set 0 279 inet_db
150 sys_dist       sys_dist       set 64 3047 net_kernel
151 ===== netlab32@netlab3 =====
152 id            name          type  size  mem  owner
153 -----
154 10            cookies          set 0 279 auth
155 13            code          set 279 12706 code_server
156 14            code_names    set 48 4649 code_server
157 15            ign_requests set 0 279 inet_gethost_native
158 16            ign_req_index set 0 279 inet_gethost_native
159 ac_tab        ac_tab        set 20 1389 application_controller
160 file_io_servers file_io_servers set 1 347 file_server_2
161 global_locks global_locks set 0 279 global_name_server
162 global_names global_names set 391 11851 global_name_server
163 global_names_ext global_names_ext set 0 279 global_name_server
164 global_pid_ids global_pid_ids bag 0 279 global_name_server
165 global_pid_names global_pid_names bag 782 11203 global_name_server
166 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
167 hm_table_global hm_table_global bag 448 33431 <0.90.0>
168 inet_cache     inet_cache     bag 0 279 inet_db
169 inet_db        inet_db        set 21 541 inet_db
170 inet_hosts     inet_hosts     set 0 279 inet_db
171 sys_dist       sys_dist       set 64 3047 net_kernel
172 ===== netlab33@netlab3 =====
173 id            name          type  size  mem  owner
174 -----
175 10            cookies          set 0 279 auth
176 13            code          set 279 12706 code_server
177 14            code_names    set 48 4649 code_server
178 15            ign_requests set 0 279 inet_gethost_native
179 16            ign_req_index set 0 279 inet_gethost_native
180 ac_tab        ac_tab        set 20 1389 application_controller
181 file_io_servers file_io_servers set 1 347 file_server_2
182 global_locks global_locks set 0 279 global_name_server
183 global_names global_names set 391 11851 global_name_server
184 global_names_ext global_names_ext set 0 279 global_name_server
185 global_pid_ids global_pid_ids bag 0 279 global_name_server
186 global_pid_names global_pid_names bag 782 11203 global_name_server
187 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
188 hm_table_global hm_table_global bag 828 61551 <0.102.0>
189 inet_cache     inet_cache     bag 0 279 inet_db
190 inet_db        inet_db        set 21 541 inet_db
191 inet_hosts     inet_hosts     set 0 279 inet_db
192 sys_dist       sys_dist       set 64 3047 net_kernel
193 ===== netlab34@netlab3 =====
194 id            name          type  size  mem  owner
195 -----

```

```

196 10          cookies          set 0      279      auth
197 13          code             set 279    12706   code_server
198 14          code_names       set 48     4649   code_server
199 15          ign_requests      set 0      279    inet_gethost_native
200 16          ign_req_index     set 0      279    inet_gethost_native
201 ac_tab      ac_tab           set 20     1389   application_controller
202 file_io_servers file_io_servers set 1      347    file_server_2
203 global_locks global_locks    set 0      279    global_name_server
204 global_names global_names    set 391    11851  global_name_server
205 global_names_ext global_names_ext set 0      279    global_name_server
206 global_pid_ids global_pid_ids bag 0      279    global_name_server
207 global_pid_names global_pid_names bag 782    11203  global_name_server
208 hm_ets_cache_table hm_ets_cache_table set 0      279    hm_cache_mgr
209 hm_table_global hm_table_global bag 967    71837  <0.123.0>
210 inet_cache     inet_cache     bag 0      279    inet_db
211 inet_db        inet_db        set 21     541    inet_db
212 inet_hosts     inet_hosts     set 0      279    inet_db
213 sys_dist      sys_dist      set 64     3047   net_kernel
214 ===== netlab35@netlab3 =====
215 id            name            type size    mem    owner
216 -----
217 10            cookies          set 0      279    auth
218 13            code             set 279    12706  code_server
219 14            code_names       set 48     4649   code_server
220 15            ign_requests      set 0      279    inet_gethost_native
221 16            ign_req_index     set 0      279    inet_gethost_native
222 ac_tab        ac_tab           set 20     1389   application_controller
223 file_io_servers file_io_servers set 1      347    file_server_2
224 global_locks global_locks    set 0      279    global_name_server
225 global_names global_names    set 391    11851  global_name_server
226 global_names_ext global_names_ext set 0      279    global_name_server
227 global_pid_ids global_pid_ids bag 0      279    global_name_server
228 global_pid_names global_pid_names bag 782    11203  global_name_server
229 hm_ets_cache_table hm_ets_cache_table set 0      279    hm_cache_mgr
230 hm_table_global hm_table_global bag 771    57333  <0.111.0>
231 inet_cache     inet_cache     bag 0      279    inet_db
232 inet_db        inet_db        set 21     541    inet_db
233 inet_hosts     inet_hosts     set 0      279    inet_db
234 sys_dist      sys_dist      set 64     3047   net_kernel
235 ===== netlab36@netlab3 =====
236 id            name            type size    mem    owner
237 -----
238 10            cookies          set 0      279    auth
239 13            code             set 279    12706  code_server
240 14            code_names       set 48     4649   code_server
241 15            ign_requests      set 0      279    inet_gethost_native
242 16            ign_req_index     set 0      279    inet_gethost_native
243 ac_tab        ac_tab           set 20     1389   application_controller
244 file_io_servers file_io_servers set 1      347    file_server_2
245 global_locks global_locks    set 0      279    global_name_server
246 global_names global_names    set 391    11851  global_name_server
247 global_names_ext global_names_ext set 0      279    global_name_server
248 global_pid_ids global_pid_ids bag 0      279    global_name_server
249 global_pid_names global_pid_names bag 782    11203  global_name_server
250 hm_ets_cache_table hm_ets_cache_table set 0      279    hm_cache_mgr
251 hm_table_global hm_table_global bag 754    56075  <0.130.0>
252 inet_cache     inet_cache     bag 0      279    inet_db
253 inet_db        inet_db        set 21     541    inet_db
254 inet_hosts     inet_hosts     set 0      279    inet_db
255 sys_dist      sys_dist      set 64     3047   net_kernel

```

256 ===== netlab37@netlab3 =====

id	name	type	size	mem	owner
10	cookies	set	0	279	auth
13	code	set	279	12706	code_server
14	code_names	set	48	4649	code_server
15	ign_requests	set	0	279	inet_gethost_native
16	ign_req_index	set	0	279	inet_gethost_native
ac_tab	ac_tab	set	20	1389	application_controller
file_io_servers	file_io_servers	set	1	347	file_server_2
global_locks	global_locks	set	0	279	global_name_server
global_names	global_names	set	391	11851	global_name_server
global_names_ext	global_names_ext	set	0	279	global_name_server
global_pid_ids	global_pid_ids	bag	0	279	global_name_server
global_pid_names	global_pid_names	bag	782	11203	global_name_server
hm_ets_cache_table	hm_ets_cache_table	set	0	279	hm_cache_mgr
hm_table_global	hm_table_global	bag	1659	123045	<0.124.0>
inet_cache	inet_cache	bag	0	279	inet_db
inet_db	inet_db	set	21	541	inet_db
inet_hosts	inet_hosts	set	0	279	inet_db
sys_dist	sys_dist	set	64	3047	net_kernel

277 ===== netlab38@netlab3 =====

id	name	type	size	mem	owner
10	cookies	set	0	279	auth
13	code	set	279	12706	code_server
14	code_names	set	48	4649	code_server
15	ign_requests	set	0	279	inet_gethost_native
16	ign_req_index	set	0	279	inet_gethost_native
ac_tab	ac_tab	set	20	1389	application_controller
file_io_servers	file_io_servers	set	1	347	file_server_2
global_locks	global_locks	set	0	279	global_name_server
global_names	global_names	set	391	11851	global_name_server
global_names_ext	global_names_ext	set	0	279	global_name_server
global_pid_ids	global_pid_ids	bag	0	279	global_name_server
global_pid_names	global_pid_names	bag	782	11203	global_name_server
hm_ets_cache_table	hm_ets_cache_table	set	0	279	hm_cache_mgr
hm_table_global	hm_table_global	bag	1185	87969	<0.121.0>
inet_cache	inet_cache	bag	0	279	inet_db
inet_db	inet_db	set	21	541	inet_db
inet_hosts	inet_hosts	set	0	279	inet_db
sys_dist	sys_dist	set	64	3047	net_kernel

298 ===== netlab39@netlab3 =====

id	name	type	size	mem	owner
10	cookies	set	0	279	auth
13	code	set	279	12706	code_server
14	code_names	set	48	4649	code_server
15	ign_requests	set	0	279	inet_gethost_native
16	ign_req_index	set	0	279	inet_gethost_native
ac_tab	ac_tab	set	20	1389	application_controller
file_io_servers	file_io_servers	set	1	347	file_server_2
global_locks	global_locks	set	0	279	global_name_server
global_names	global_names	set	391	11851	global_name_server
global_names_ext	global_names_ext	set	0	279	global_name_server
global_pid_ids	global_pid_ids	bag	0	279	global_name_server
global_pid_names	global_pid_names	bag	782	11203	global_name_server
hm_ets_cache_table	hm_ets_cache_table	set	0	279	hm_cache_mgr
hm_table_global	hm_table_global	bag	713	53041	<0.104.0>
inet_cache	inet_cache	bag	0	279	inet_db

```

316 inet_db          inet_db          set  21    541    inet_db
317 inet_hosts       inet_hosts       set   0    279    inet_db
318 sys_dist         sys_dist         set  64   3047    net_kernel
319 ===== netlab310@netlab3 =====
320 id              name              type  size  mem    owner
321 -----
322 10              cookies           set   0    279    auth
323 13              code              set  279   12706   code_server
324 14              code_names        set   48   4649   code_server
325 15              ign_requests      set   0    279    inet_gethost_native
326 16              ign_req_index     set   0    279    inet_gethost_native
327 ac_tab          ac_tab            set  20   1389   application_controller
328 file_io_servers file_io_servers   set   1    349    file_server_2
329 global_locks     global_locks      set   0    279    global_name_server
330 global_names     global_names      set  391   11815   global_name_server
331 global_names_ext global_names_ext  set   0    279    global_name_server
332 global_pid_ids   global_pid_ids    bag   0    279    global_name_server
333 global_pid_names global_pid_names  bag  782   11203   global_name_server
334 hm_ets_cache_table hm_ets_cache_table set   0    279    hm_cache_mgr
335 hm_table_global  hm_table_global   bag  466   34763   <0.142.0>
336 inet_cache       inet_cache        bag   0    279    inet_db
337 inet_db          inet_db           set  21    541    inet_db
338 inet_hosts       inet_hosts        set   0    279    inet_db
339 sys_dist         sys_dist          set  64   3047    net_kernel
340 ===== netlab311@netlab3 =====
341 id              name              type  size  mem    owner
342 -----
343 10              cookies           set   0    279    auth
344 13              code              set  279   12706   code_server
345 14              code_names        set   48   4649   code_server
346 15              ign_requests      set   0    279    inet_gethost_native
347 16              ign_req_index     set   0    279    inet_gethost_native
348 ac_tab          ac_tab            set  20   1389   application_controller
349 file_io_servers file_io_servers   set   1    349    file_server_2
350 global_locks     global_locks      set   0    279    global_name_server
351 global_names     global_names      set  391   11827   global_name_server
352 global_names_ext global_names_ext  set   0    279    global_name_server
353 global_pid_ids   global_pid_ids    bag   0    279    global_name_server
354 global_pid_names global_pid_names  bag  782   11203   global_name_server
355 hm_ets_cache_table hm_ets_cache_table set   0    279    hm_cache_mgr
356 hm_table_global  hm_table_global   bag 1621  120233   <0.110.0>
357 inet_cache       inet_cache        bag   0    279    inet_db
358 inet_db          inet_db           set  21    541    inet_db
359 inet_hosts       inet_hosts        set   0    279    inet_db
360 sys_dist         sys_dist          set  64   3047    net_kernel
361 ===== netlab312@netlab3 =====
362 id              name              type  size  mem    owner
363 -----
364 10              cookies           set   0    279    auth
365 13              code              set  279   12706   code_server
366 14              code_names        set   48   4649   code_server
367 15              ign_requests      set   0    279    inet_gethost_native
368 16              ign_req_index     set   0    279    inet_gethost_native
369 ac_tab          ac_tab            set  20   1389   application_controller
370 file_io_servers file_io_servers   set   1    349    file_server_2
371 global_locks     global_locks      set   0    279    global_name_server
372 global_names     global_names      set  391   11815   global_name_server
373 global_names_ext global_names_ext  set   0    279    global_name_server
374 global_pid_ids   global_pid_ids    bag   0    279    global_name_server
375 global_pid_names global_pid_names  bag  782   11203   global_name_server

```

```

376 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
377 hm_table_global hm_table_global bag 924 68655 <0.147.0>
378 inet_cache inet_cache bag 0 279 inet_db
379 inet_db inet_db set 21 541 inet_db
380 inet_hosts inet_hosts set 0 279 inet_db
381 sys_dist sys_dist set 64 3047 net_kernel
382 ===== netlab313@netlab3 =====
383 id name type size mem owner
384 -----
385 10 cookies set 0 279 auth
386 13 code set 279 12706 code_server
387 14 code_names set 48 4649 code_server
388 15 ign_requests set 0 279 inet_gethost_native
389 16 ign_req_index set 0 279 inet_gethost_native
390 ac_tab ac_tab set 20 1389 application_controller
391 file_io_servers file_io_servers set 1 349 file_server_2
392 global_locks global_locks set 0 279 global_name_server
393 global_names global_names set 391 11815 global_name_server
394 global_names_ext global_names_ext set 0 279 global_name_server
395 global_pid_ids global_pid_ids bag 0 279 global_name_server
396 global_pid_names global_pid_names bag 782 11203 global_name_server
397 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
398 hm_table_global hm_table_global bag 548 40831 <0.139.0>
399 inet_cache inet_cache bag 0 279 inet_db
400 inet_db inet_db set 21 541 inet_db
401 inet_hosts inet_hosts set 0 279 inet_db
402 sys_dist sys_dist set 64 3047 net_kernel
403 ===== netlab314@netlab3 =====
404 id name type size mem owner
405 -----
406 10 cookies set 0 279 auth
407 13 code set 279 12706 code_server
408 14 code_names set 48 4649 code_server
409 15 ign_requests set 0 279 inet_gethost_native
410 16 ign_req_index set 0 279 inet_gethost_native
411 ac_tab ac_tab set 20 1389 application_controller
412 file_io_servers file_io_servers set 1 349 file_server_2
413 global_locks global_locks set 0 279 global_name_server
414 global_names global_names set 391 11703 global_name_server
415 global_names_ext global_names_ext set 0 279 global_name_server
416 global_pid_ids global_pid_ids bag 0 279 global_name_server
417 global_pid_names global_pid_names bag 782 11203 global_name_server
418 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
419 hm_table_global hm_table_global bag 736 54743 <0.171.0>
420 inet_cache inet_cache bag 0 279 inet_db
421 inet_db inet_db set 21 541 inet_db
422 inet_hosts inet_hosts set 0 279 inet_db
423 sys_dist sys_dist set 64 3047 net_kernel
424 ===== netlab315@netlab3 =====
425 id name type size mem owner
426 -----
427 10 cookies set 0 279 auth
428 13 code set 279 12706 code_server
429 14 code_names set 48 4649 code_server
430 15 ign_requests set 0 279 inet_gethost_native
431 16 ign_req_index set 0 279 inet_gethost_native
432 ac_tab ac_tab set 20 1389 application_controller
433 file_io_servers file_io_servers set 1 349 file_server_2
434 global_locks global_locks set 0 279 global_name_server
435 global_names global_names set 391 11767 global_name_server

```

```

436 global_names_ext global_names_ext set 0 279 global_name_server
437 global_pid_ids global_pid_ids bag 0 279 global_name_server
438 global_pid_names global_pid_names bag 782 11203 global_name_server
439 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
440 hm_table_global hm_table_global bag 523 38981 <0.148.0>
441 inet_cache inet_cache bag 0 279 inet_db
442 inet_db inet_db set 21 541 inet_db
443 inet_hosts inet_hosts set 0 279 inet_db
444 sys_dist sys_dist set 64 3047 net_kernel
445 ===== netlab316@netlab3 =====
446 id name type size mem owner
447 -----
448 10 cookies set 0 279 auth
449 13 code set 279 12706 code_server
450 14 code_names set 48 4649 code_server
451 15 ign_requests set 0 279 inet_gethost_native
452 16 ign_req_index set 0 279 inet_gethost_native
453 ac_tab ac_tab set 20 1389 application_controller
454 file_io_servers file_io_servers set 1 349 file_server_2
455 global_locks global_locks set 0 279 global_name_server
456 global_names global_names set 391 11691 global_name_server
457 global_names_ext global_names_ext set 0 279 global_name_server
458 global_pid_ids global_pid_ids bag 0 279 global_name_server
459 global_pid_names global_pid_names bag 782 11203 global_name_server
460 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
461 hm_table_global hm_table_global bag 672 50007 <0.171.0>
462 inet_cache inet_cache bag 0 279 inet_db
463 inet_db inet_db set 21 541 inet_db
464 inet_hosts inet_hosts set 0 279 inet_db
465 sys_dist sys_dist set 64 3047 net_kernel
466 ===== netlab317@netlab3 =====
467 id name type size mem owner
468 -----
469 10 cookies set 0 279 auth
470 13 code set 279 12706 code_server
471 14 code_names set 48 4649 code_server
472 15 ign_requests set 0 279 inet_gethost_native
473 16 ign_req_index set 0 279 inet_gethost_native
474 ac_tab ac_tab set 20 1389 application_controller
475 file_io_servers file_io_servers set 1 349 file_server_2
476 global_locks global_locks set 0 279 global_name_server
477 global_names global_names set 391 11703 global_name_server
478 global_names_ext global_names_ext set 0 279 global_name_server
479 global_pid_ids global_pid_ids bag 0 279 global_name_server
480 global_pid_names global_pid_names bag 782 11203 global_name_server
481 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
482 hm_table_global hm_table_global bag 882 65547 <0.172.0>
483 inet_cache inet_cache bag 0 279 inet_db
484 inet_db inet_db set 21 541 inet_db
485 inet_hosts inet_hosts set 0 279 inet_db
486 sys_dist sys_dist set 64 3047 net_kernel
487 ===== netlab318@netlab3 =====
488 id name type size mem owner
489 -----
490 10 cookies set 0 279 auth
491 13 code set 279 12706 code_server
492 14 code_names set 48 4649 code_server
493 15 ign_requests set 0 279 inet_gethost_native
494 16 ign_req_index set 0 279 inet_gethost_native
495 'Domain1Tbl2' 'Domain1Tbl2' bag 10000 121814 <0.168.0>

```

```

496 ac_tab          ac_tab          set 20    1389    application_controller
497 file_io_servers file_io_servers set 1      349     file_server_2
498 global_locks    global_locks    set 0      279     global_name_server
499 global_names    global_names    set 391   11715   global_name_server
500 global_names_ext global_names_ext set 0      279     global_name_server
501 global_pid_ids  global_pid_ids  bag 0      279     global_name_server
502 global_pid_names global_pid_names bag 782   11203   global_name_server
503 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
504 hm_table_global hm_table_global bag 708   52671   <0.167.0>
505 inet_cache      inet_cache      bag 0      279     inet_db
506 inet_db         inet_db         set 21     541     inet_db
507 inet_hosts      inet_hosts      set 0      279     inet_db
508 sys_dist        sys_dist        set 64    3047    net_kernel
509 ===== netlab319@netlab3 =====
510 id              name              type  size    mem      owner
511 -----
512 10              cookies           set 0      279     auth
513 13              code             set 279    12706   code_server
514 14              code_names       set 48     4649    code_server
515 15              ign_requests     set 0      279     inet_gethost_native
516 16              ign_req_index    set 0      279     inet_gethost_native
517 'Domain1Tbl2'  'Domain1Tbl2'    bag 10000  121814  <0.171.0>
518 ac_tab          ac_tab          set 20    1389    application_controller
519 file_io_servers file_io_servers set 1      349     file_server_2
520 global_locks    global_locks    set 0      279     global_name_server
521 global_names    global_names    set 391   11703   global_name_server
522 global_names_ext global_names_ext set 0      279     global_name_server
523 global_pid_ids  global_pid_ids  bag 0      279     global_name_server
524 global_pid_names global_pid_names bag 782   11203   global_name_server
525 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
526 hm_table_global hm_table_global bag 416   31063   <0.170.0>
527 inet_cache      inet_cache      bag 0      279     inet_db
528 inet_db         inet_db         set 21     541     inet_db
529 inet_hosts      inet_hosts      set 0      279     inet_db
530 sys_dist        sys_dist        set 64    3047    net_kernel
531 ===== netlab320@netlab3 =====
532 id              name              type  size    mem      owner
533 -----
534 10              cookies           set 0      279     auth
535 13              code             set 279    12706   code_server
536 14              code_names       set 48     4649    code_server
537 15              ign_requests     set 0      279     inet_gethost_native
538 16              ign_req_index    set 0      279     inet_gethost_native
539 'Domain1Tbl2'  'Domain1Tbl2'    bag 10000  121814  <0.169.0>
540 ac_tab          ac_tab          set 20    1389    application_controller
541 file_io_servers file_io_servers set 1      349     file_server_2
542 global_locks    global_locks    set 0      279     global_name_server
543 global_names    global_names    set 391   11703   global_name_server
544 global_names_ext global_names_ext set 0      279     global_name_server
545 global_pid_ids  global_pid_ids  bag 0      279     global_name_server
546 global_pid_names global_pid_names bag 782   11203   global_name_server
547 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
548 hm_table_global hm_table_global bag 333   24921   <0.168.0>
549 inet_cache      inet_cache      bag 0      279     inet_db
550 inet_db         inet_db         set 21     541     inet_db
551 inet_hosts      inet_hosts      set 0      279     inet_db
552 sys_dist        sys_dist        set 64    3047    net_kernel
553 ===== netlab41@netlab4 =====
554 id              name              type  size    mem      owner
555 -----

```

```

556 10          cookies          set 0      279      auth
557 13          code             set 279    12706   code_server
558 14          code_names       set 48    4649   code_server
559 15          ign_requests      set 0     279    inet_gethost_native
560 16          ign_req_index     set 0     279    inet_gethost_native
561 ac_tab      ac_tab           set 20    1389   application_controller
562 file_io_servers file_io_servers set 1     347    file_server_2
563 global_locks global_locks    set 0     279    global_name_server
564 global_names global_names     set 391   11371  global_name_server
565 global_names_ext global_names_ext set 0     279    global_name_server
566 global_pid_ids global_pid_ids bag 0     279    global_name_server
567 global_pid_names global_pid_names bag 782   11203  global_name_server
568 hm_ets_cache_table hm_ets_cache_table set 0     279    hm_cache_mgr
569 hm_table_global hm_table_global bag 891   66213  <0.344.0>
570 inet_cache    inet_cache      bag 0     279    inet_db
571 inet_db       inet_db        set 21    541    inet_db
572 inet_hosts    inet_hosts      set 0     279    inet_db
573 sys_dist      sys_dist       set 64    3047   net_kernel
574 ===== netlab42@netlab4 =====
575 id            name            type  size    mem    owner
576 -----
577 10            cookies          set 0     279    auth
578 13            code             set 279    12706  code_server
579 14            code_names       set 48    4649   code_server
580 15            ign_requests      set 0     279    inet_gethost_native
581 16            ign_req_index     set 0     279    inet_gethost_native
582 ac_tab        ac_tab           set 20    1389   application_controller
583 file_io_servers file_io_servers set 1     347    file_server_2
584 global_locks  global_locks    set 0     279    global_name_server
585 global_names  global_names     set 391   11371  global_name_server
586 global_names_ext global_names_ext set 0     279    global_name_server
587 global_pid_ids global_pid_ids bag 0     279    global_name_server
588 global_pid_names global_pid_names bag 782   11203  global_name_server
589 hm_ets_cache_table hm_ets_cache_table set 0     279    hm_cache_mgr
590 hm_table_global hm_table_global bag 478   35651  <0.315.0>
591 inet_cache    inet_cache      bag 0     279    inet_db
592 inet_db       inet_db        set 21    541    inet_db
593 inet_hosts    inet_hosts      set 0     279    inet_db
594 sys_dist      sys_dist       set 64    3047   net_kernel
595 ===== netlab43@netlab4 =====
596 id            name            type  size    mem    owner
597 -----
598 10            cookies          set 0     279    auth
599 13            code             set 279    12706  code_server
600 14            code_names       set 48    4649   code_server
601 15            ign_requests      set 0     279    inet_gethost_native
602 16            ign_req_index     set 0     279    inet_gethost_native
603 ac_tab        ac_tab           set 20    1389   application_controller
604 file_io_servers file_io_servers set 1     347    file_server_2
605 global_locks  global_locks    set 0     279    global_name_server
606 global_names  global_names     set 391   11371  global_name_server
607 global_names_ext global_names_ext set 0     279    global_name_server
608 global_pid_ids global_pid_ids bag 0     279    global_name_server
609 global_pid_names global_pid_names bag 782   11203  global_name_server
610 hm_ets_cache_table hm_ets_cache_table set 0     279    hm_cache_mgr
611 hm_table_global hm_table_global bag 382   28547  <0.313.0>
612 inet_cache    inet_cache      bag 0     279    inet_db
613 inet_db       inet_db        set 21    541    inet_db
614 inet_hosts    inet_hosts      set 0     279    inet_db
615 sys_dist      sys_dist       set 64    3047   net_kernel

```


616 ===== netlab44@netlab4 =====

id	name	type	size	mem	owner
10	cookies	set	0	279	auth
13	code	set	279	12706	code_server
14	code_names	set	48	4649	code_server
15	ign_requests	set	0	279	inet_gethost_native
16	ign_req_index	set	0	279	inet_gethost_native
ac_tab	ac_tab	set	20	1389	application_controller
file_io_servers	file_io_servers	set	1	347	file_server_2
global_locks	global_locks	set	0	279	global_name_server
global_names	global_names	set	391	11371	global_name_server
global_names_ext	global_names_ext	set	0	279	global_name_server
global_pid_ids	global_pid_ids	bag	0	279	global_name_server
global_pid_names	global_pid_names	bag	782	11203	global_name_server
hm_ets_cache_table	hm_ets_cache_table	set	0	279	hm_cache_mgr
hm_table_global	hm_table_global	bag	959	71245	<0.309.0>
inet_cache	inet_cache	bag	0	279	inet_db
inet_db	inet_db	set	21	541	inet_db
inet_hosts	inet_hosts	set	0	279	inet_db
sys_dist	sys_dist	set	64	3047	net_kernel

637 ===== netlab45@netlab4 =====

id	name	type	size	mem	owner
10	cookies	set	0	279	auth
13	code	set	279	12706	code_server
14	code_names	set	48	4649	code_server
15	ign_requests	set	0	279	inet_gethost_native
16	ign_req_index	set	0	279	inet_gethost_native
ac_tab	ac_tab	set	20	1389	application_controller
file_io_servers	file_io_servers	set	1	347	file_server_2
global_locks	global_locks	set	0	279	global_name_server
global_names	global_names	set	391	11371	global_name_server
global_names_ext	global_names_ext	set	0	279	global_name_server
global_pid_ids	global_pid_ids	bag	0	279	global_name_server
global_pid_names	global_pid_names	bag	782	11203	global_name_server
hm_ets_cache_table	hm_ets_cache_table	set	0	279	hm_cache_mgr
hm_table_global	hm_table_global	bag	601	44753	<0.316.0>
inet_cache	inet_cache	bag	0	279	inet_db
inet_db	inet_db	set	21	541	inet_db
inet_hosts	inet_hosts	set	0	279	inet_db
sys_dist	sys_dist	set	64	3047	net_kernel

658 ===== netlab46@netlab4 =====

id	name	type	size	mem	owner
10	cookies	set	0	279	auth
13	code	set	279	12706	code_server
14	code_names	set	48	4649	code_server
15	ign_requests	set	0	279	inet_gethost_native
16	ign_req_index	set	0	279	inet_gethost_native
ac_tab	ac_tab	set	20	1389	application_controller
file_io_servers	file_io_servers	set	1	347	file_server_2
global_locks	global_locks	set	0	279	global_name_server
global_names	global_names	set	391	11371	global_name_server
global_names_ext	global_names_ext	set	0	279	global_name_server
global_pid_ids	global_pid_ids	bag	0	279	global_name_server
global_pid_names	global_pid_names	bag	782	11203	global_name_server
hm_ets_cache_table	hm_ets_cache_table	set	0	279	hm_cache_mgr
hm_table_global	hm_table_global	bag	737	54817	<0.351.0>
inet_cache	inet_cache	bag	0	279	inet_db

```

676 inet_db          inet_db          set 21    541    inet_db
677 inet_hosts       inet_hosts       set 0     279    inet_db
678 sys_dist         sys_dist         set 64   3047   net_kernel
679 ===== netlab47@netlab4 =====
680 id              name              type  size    mem    owner
681 -----
682 10              cookies           set 0      279    auth
683 13              code              set 279    12706  code_server
684 14              code_names        set 48     4649   code_server
685 15              ign_requests      set 0      279    inet_gethost_native
686 16              ign_req_index     set 0      279    inet_gethost_native
687 ac_tab          ac_tab           set 20    1389   application_controller
688 file_io_servers file_io_servers  set 1     347    file_server_2
689 global_locks    global_locks     set 0      279    global_name_server
690 global_names     global_names     set 391   11371  global_name_server
691 global_names_ext global_names_ext set 0      279    global_name_server
692 global_pid_ids  global_pid_ids   bag 0      279    global_name_server
693 global_pid_names global_pid_names bag 782   11203  global_name_server
694 hm_ets_cache_table hm_ets_cache_table set 0      279    hm_cache_mgr
695 hm_table_global hm_table_global  bag 426   31803  <0.339.0>
696 inet_cache      inet_cache       bag 0      279    inet_db
697 inet_db         inet_db          set 21    541    inet_db
698 inet_hosts      inet_hosts       set 0      279    inet_db
699 sys_dist        sys_dist         set 64   3047   net_kernel
700 ===== netlab48@netlab4 =====
701 id              name              type  size    mem    owner
702 -----
703 10              cookies           set 0      279    auth
704 13              code              set 279    12706  code_server
705 14              code_names        set 48     4649   code_server
706 15              ign_requests      set 0      279    inet_gethost_native
707 16              ign_req_index     set 0      279    inet_gethost_native
708 ac_tab          ac_tab           set 20    1389   application_controller
709 file_io_servers file_io_servers  set 1     347    file_server_2
710 global_locks    global_locks     set 0      279    global_name_server
711 global_names     global_names     set 391   11363  global_name_server
712 global_names_ext global_names_ext set 0      279    global_name_server
713 global_pid_ids  global_pid_ids   bag 0      279    global_name_server
714 global_pid_names global_pid_names bag 782   11203  global_name_server
715 hm_ets_cache_table hm_ets_cache_table set 0      279    hm_cache_mgr
716 hm_table_global hm_table_global  bag 790   58739  <0.309.0>
717 inet_cache      inet_cache       bag 0      279    inet_db
718 inet_db         inet_db          set 21    541    inet_db
719 inet_hosts      inet_hosts       set 0      279    inet_db
720 sys_dist        sys_dist         set 64   3047   net_kernel
721 ===== netlab49@netlab4 =====
722 id              name              type  size    mem    owner
723 -----
724 10              cookies           set 0      279    auth
725 13              code              set 279    12706  code_server
726 14              code_names        set 48     4649   code_server
727 15              ign_requests      set 0      279    inet_gethost_native
728 16              ign_req_index     set 0      279    inet_gethost_native
729 ac_tab          ac_tab           set 20    1389   application_controller
730 file_io_servers file_io_servers  set 1     347    file_server_2
731 global_locks    global_locks     set 0      279    global_name_server
732 global_names     global_names     set 391   11371  global_name_server
733 global_names_ext global_names_ext set 0      279    global_name_server
734 global_pid_ids  global_pid_ids   bag 0      279    global_name_server
735 global_pid_names global_pid_names bag 782   11203  global_name_server

```

```

736 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
737 hm_table_global hm_table_global bag 943 70061 <0.346.0>
738 inet_cache inet_cache bag 0 279 inet_db
739 inet_db inet_db set 21 541 inet_db
740 inet_hosts inet_hosts set 0 279 inet_db
741 sys_dist sys_dist set 64 3047 net_kernel
742 ===== netlab410@netlab4 =====
743 id name type size mem owner
744 -----
745 10 cookies set 0 279 auth
746 13 code set 279 12706 code_server
747 14 code_names set 48 4649 code_server
748 15 ign_requests set 0 279 inet_gethost_native
749 16 ign_req_index set 0 279 inet_gethost_native
750 ac_tab ac_tab set 20 1389 application_controller
751 file_io_servers file_io_servers set 1 349 file_server_2
752 global_locks global_locks set 0 279 global_name_server
753 global_names global_names set 391 11239 global_name_server
754 global_names_ext global_names_ext set 0 279 global_name_server
755 global_pid_ids global_pid_ids bag 0 279 global_name_server
756 global_pid_names global_pid_names bag 782 11203 global_name_server
757 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
758 hm_table_global hm_table_global bag 889 66065 <0.382.0>
759 inet_cache inet_cache bag 0 279 inet_db
760 inet_db inet_db set 21 541 inet_db
761 inet_hosts inet_hosts set 0 279 inet_db
762 sys_dist sys_dist set 64 3047 net_kernel
763 ===== netlab411@netlab4 =====
764 id name type size mem owner
765 -----
766 10 cookies set 0 279 auth
767 13 code set 279 12706 code_server
768 14 code_names set 48 4649 code_server
769 15 ign_requests set 0 279 inet_gethost_native
770 16 ign_req_index set 0 279 inet_gethost_native
771 ac_tab ac_tab set 20 1389 application_controller
772 file_io_servers file_io_servers set 1 349 file_server_2
773 global_locks global_locks set 0 279 global_name_server
774 global_names global_names set 391 11307 global_name_server
775 global_names_ext global_names_ext set 0 279 global_name_server
776 global_pid_ids global_pid_ids bag 0 279 global_name_server
777 global_pid_names global_pid_names bag 782 11203 global_name_server
778 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
779 hm_table_global hm_table_global bag 841 62513 <0.376.0>
780 inet_cache inet_cache bag 0 279 inet_db
781 inet_db inet_db set 21 541 inet_db
782 inet_hosts inet_hosts set 0 279 inet_db
783 sys_dist sys_dist set 64 3047 net_kernel
784 ===== netlab412@netlab4 =====
785 id name type size mem owner
786 -----
787 10 cookies set 0 279 auth
788 13 code set 279 12706 code_server
789 14 code_names set 48 4649 code_server
790 15 ign_requests set 0 279 inet_gethost_native
791 16 ign_req_index set 0 279 inet_gethost_native
792 ac_tab ac_tab set 20 1389 application_controller
793 file_io_servers file_io_servers set 1 349 file_server_2
794 global_locks global_locks set 0 279 global_name_server
795 global_names global_names set 391 11263 global_name_server

```

```

796 global_names_ext global_names_ext set 0 279 global_name_server
797 global_pid_ids global_pid_ids bag 0 279 global_name_server
798 global_pid_names global_pid_names bag 782 11203 global_name_server
799 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
800 hm_table_global hm_table_global bag 829 61625 <0.378.0>
801 inet_cache inet_cache bag 0 279 inet_db
802 inet_db inet_db set 21 541 inet_db
803 inet_hosts inet_hosts set 0 279 inet_db
804 sys_dist sys_dist set 64 3047 net_kernel
805 ===== netlab413@netlab4 =====
806 id name type size mem owner
807 -----
808 10 cookies set 0 279 auth
809 13 code set 279 12706 code_server
810 14 code_names set 48 4649 code_server
811 15 ign_requests set 0 279 inet_gethost_native
812 16 ign_req_index set 0 279 inet_gethost_native
813 ac_tab ac_tab set 20 1389 application_controller
814 file_io_servers file_io_servers set 1 349 file_server_2
815 global_locks global_locks set 0 279 global_name_server
816 global_names global_names set 391 11239 global_name_server
817 global_names_ext global_names_ext set 0 279 global_name_server
818 global_pid_ids global_pid_ids bag 0 279 global_name_server
819 global_pid_names global_pid_names bag 782 11203 global_name_server
820 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
821 hm_table_global hm_table_global bag 454 33875 <0.386.0>
822 inet_cache inet_cache bag 0 279 inet_db
823 inet_db inet_db set 21 541 inet_db
824 inet_hosts inet_hosts set 0 279 inet_db
825 sys_dist sys_dist set 64 3047 net_kernel
826 ===== netlab414@netlab4 =====
827 id name type size mem owner
828 -----
829 10 cookies set 0 279 auth
830 13 code set 279 12706 code_server
831 14 code_names set 48 4649 code_server
832 15 ign_requests set 0 279 inet_gethost_native
833 16 ign_req_index set 0 279 inet_gethost_native
834 ac_tab ac_tab set 20 1389 application_controller
835 file_io_servers file_io_servers set 1 349 file_server_2
836 global_locks global_locks set 0 279 global_name_server
837 global_names global_names set 391 11239 global_name_server
838 global_names_ext global_names_ext set 0 279 global_name_server
839 global_pid_ids global_pid_ids bag 0 279 global_name_server
840 global_pid_names global_pid_names bag 782 11203 global_name_server
841 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
842 hm_table_global hm_table_global bag 1163 86341 <0.383.0>
843 inet_cache inet_cache bag 0 279 inet_db
844 inet_db inet_db set 21 541 inet_db
845 inet_hosts inet_hosts set 0 279 inet_db
846 sys_dist sys_dist set 64 3047 net_kernel
847 ===== netlab415@netlab4 =====
848 id name type size mem owner
849 -----
850 10 cookies set 0 279 auth
851 13 code set 279 12706 code_server
852 14 code_names set 48 4649 code_server
853 15 ign_requests set 0 279 inet_gethost_native
854 16 ign_req_index set 0 279 inet_gethost_native
855 ac_tab ac_tab set 20 1389 application_controller

```

```

856 file_io_servers file_io_servers set 1 349 file_server_2
857 global_locks global_locks set 0 279 global_name_server
858 global_names global_names set 391 11239 global_name_server
859 global_names_ext global_names_ext set 0 279 global_name_server
860 global_pid_ids global_pid_ids bag 0 279 global_name_server
861 global_pid_names global_pid_names bag 782 11203 global_name_server
862 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
863 hm_table_global hm_table_global bag 570 42459 <0.395.0>
864 inet_cache inet_cache bag 0 279 inet_db
865 inet_db inet_db set 21 541 inet_db
866 inet_hosts inet_hosts set 0 279 inet_db
867 sys_dist sys_dist set 64 3047 net_kernel
868 ===== netlab416@netlab4 =====
869 id name type size mem owner
870 -----
871 10 cookies set 0 279 auth
872 13 code set 279 12706 code_server
873 14 code_names set 48 4649 code_server
874 15 ign_requests set 0 279 inet_gethost_native
875 16 ign_req_index set 0 279 inet_gethost_native
876 ac_tab ac_tab set 20 1389 application_controller
877 file_io_servers file_io_servers set 1 349 file_server_2
878 global_locks global_locks set 0 279 global_name_server
879 global_names global_names set 391 11239 global_name_server
880 global_names_ext global_names_ext set 0 279 global_name_server
881 global_pid_ids global_pid_ids bag 0 279 global_name_server
882 global_pid_names global_pid_names bag 782 11203 global_name_server
883 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
884 hm_table_global hm_table_global bag 909 67545 <0.395.0>
885 inet_cache inet_cache bag 0 279 inet_db
886 inet_db inet_db set 21 541 inet_db
887 inet_hosts inet_hosts set 0 279 inet_db
888 sys_dist sys_dist set 64 3047 net_kernel
889 ===== netlab417@netlab4 =====
890 id name type size mem owner
891 -----
892 10 cookies set 0 279 auth
893 13 code set 279 12706 code_server
894 14 code_names set 48 4649 code_server
895 15 ign_requests set 0 279 inet_gethost_native
896 16 ign_req_index set 0 279 inet_gethost_native
897 ac_tab ac_tab set 20 1389 application_controller
898 file_io_servers file_io_servers set 1 349 file_server_2
899 global_locks global_locks set 0 279 global_name_server
900 global_names global_names set 391 11239 global_name_server
901 global_names_ext global_names_ext set 0 279 global_name_server
902 global_pid_ids global_pid_ids bag 0 279 global_name_server
903 global_pid_names global_pid_names bag 782 11203 global_name_server
904 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
905 hm_table_global hm_table_global bag 1546 114683 <0.395.0>
906 inet_cache inet_cache bag 0 279 inet_db
907 inet_db inet_db set 21 541 inet_db
908 inet_hosts inet_hosts set 0 279 inet_db
909 sys_dist sys_dist set 64 3047 net_kernel
910 ===== netlab418@netlab4 =====
911 id name type size mem owner
912 -----
913 10 cookies set 0 279 auth
914 13 code set 279 12706 code_server
915 14 code_names set 48 4649 code_server

```

```

916 15          ign_requests      set 0      279      inet_gethost_native
917 16          ign_req_index      set 0      279      inet_gethost_native
918 ac_tab       ac_tab            set 20     1389     application_controller
919 file_io_servers file_io_servers set 1      349     file_server_2
920 global_locks global_locks      set 0      279     global_name_server
921 global_names global_names      set 391    11215   global_name_server
922 global_names_ext global_names_ext set 0      279     global_name_server
923 global_pid_ids global_pid_ids    bag 0      279     global_name_server
924 global_pid_names global_pid_names bag 782    11203   global_name_server
925 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
926 hm_table_global hm_table_global bag 361    26993   <0.398.0>
927 inet_cache    inet_cache      bag 0      279     inet_db
928 inet_db       inet_db        set 21     541     inet_db
929 inet_hosts    inet_hosts      set 0      279     inet_db
930 sys_dist      sys_dist        set 64     3047    net_kernel
931 ===== netlab419@netlab4 =====
932 id            name                type  size    mem      owner
933 -----
934 10            cookies              set 0      279     auth
935 13            code              set 279    12706   code_server
936 14            code_names       set 48     4649   code_server
937 15            ign_requests      set 0      279     inet_gethost_native
938 16            ign_req_index      set 0      279     inet_gethost_native
939 ac_tab       ac_tab            set 20     1389     application_controller
940 file_io_servers file_io_servers set 1      349     file_server_2
941 global_locks global_locks      set 0      279     global_name_server
942 global_names global_names      set 391    11215   global_name_server
943 global_names_ext global_names_ext set 0      279     global_name_server
944 global_pid_ids global_pid_ids    bag 0      279     global_name_server
945 global_pid_names global_pid_names bag 782    11203   global_name_server
946 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
947 hm_table_global hm_table_global bag 1205   89449   <0.389.0>
948 inet_cache    inet_cache      bag 0      279     inet_db
949 inet_db       inet_db        set 21     541     inet_db
950 inet_hosts    inet_hosts      set 0      279     inet_db
951 sys_dist      sys_dist        set 64     3047    net_kernel
952 ===== netlab420@netlab4 =====
953 id            name                type  size    mem      owner
954 -----
955 10            cookies              set 0      279     auth
956 13            code              set 279    12706   code_server
957 14            code_names       set 48     4649   code_server
958 15            ign_requests      set 0      279     inet_gethost_native
959 16            ign_req_index      set 0      279     inet_gethost_native
960 ac_tab       ac_tab            set 20     1389     application_controller
961 file_io_servers file_io_servers set 1      349     file_server_2
962 global_locks global_locks      set 0      279     global_name_server
963 global_names global_names      set 391    11215   global_name_server
964 global_names_ext global_names_ext set 0      279     global_name_server
965 global_pid_ids global_pid_ids    bag 0      279     global_name_server
966 global_pid_names global_pid_names bag 782    11203   global_name_server
967 hm_ets_cache_table hm_ets_cache_table set 0      279     hm_cache_mgr
968 hm_table_global hm_table_global bag 862    64067   <0.364.0>
969 inet_cache    inet_cache      bag 0      279     inet_db
970 inet_db       inet_db        set 21     541     inet_db
971 inet_hosts    inet_hosts      set 0      279     inet_db
972 sys_dist      sys_dist        set 64     3047    net_kernel
973 ===== dell1@dell =====
974 id            name                type  size    mem      owner
975 -----

```

```

976 12          cookies          set 0      286      auth
977 4111         code            set 281    12978   code_server
978 8208         code_names       set 56    7510   code_server
979 12305        ign_requests     set 0     286    inet_gethost_native
980 16402        ign_req_index    set 0     286    inet_gethost_native
981 ac_tab       ac_tab           set 20    1328   application_controller
982 file_io_servers file_io_servers set 1     339    file_server_2
983 global_locks  global_locks   set 0     286    global_name_server
984 global_names  global_names   set 391   10433  global_name_server
985 global_names_ext global_names_ext set 0     286    global_name_server
986 global_pid_ids global_pid_ids bag 0     286    global_name_server
987 global_pid_names global_pid_names bag 782   8864   global_name_server
988 hm_ets_cache_table hm_ets_cache_table set 0     286    hm_cache_mgr
989 hm_table_global hm_table_global bag 300   21586  <0.339.0>
990 inet_cache    inet_cache     bag 0     286    inet_db
991 inet_db       inet_db       set 29    553    inet_db
992 inet_hosts_byaddr inet_hosts_byaddr bag 0     286    inet_db
993 inet_hosts_byname inet_hosts_byname bag 0     286    inet_db
994 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0     286    inet_db
995 inet_hosts_file_byname inet_hosts_file_byname bag 0     286    inet_db
996 sys_dist     sys_dist     set 64    2868   net_kernel
997 ===== dell2@dell =====
998 id           name           type  size  mem  owner
999 -----
1000 12           cookies          set 0    286   auth
1001 4111         code            set 281  12978 code_server
1002 8208         code_names       set 56   7510  code_server
1003 12305        ign_requests     set 0    286   inet_gethost_native
1004 16402        ign_req_index    set 0    286   inet_gethost_native
1005 ac_tab       ac_tab           set 20   1328  application_controller
1006 file_io_servers file_io_servers set 1    339   file_server_2
1007 global_locks  global_locks   set 0    286   global_name_server
1008 global_names  global_names   set 391  10089 global_name_server
1009 global_names_ext global_names_ext set 0    286   global_name_server
1010 global_pid_ids global_pid_ids bag 0    286   global_name_server
1011 global_pid_names global_pid_names bag 782  8864   global_name_server
1012 hm_ets_cache_table hm_ets_cache_table set 0    286   hm_cache_mgr
1013 hm_table_global hm_table_global bag 513  36709 <0.476.0>
1014 inet_cache    inet_cache     bag 0    286   inet_db
1015 inet_db       inet_db       set 29   553    inet_db
1016 inet_hosts_byaddr inet_hosts_byaddr bag 0    286   inet_db
1017 inet_hosts_byname inet_hosts_byname bag 0    286   inet_db
1018 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0    286   inet_db
1019 inet_hosts_file_byname inet_hosts_file_byname bag 0    286   inet_db
1020 sys_dist     sys_dist     set 64   2868  net_kernel
1021 ===== dell3@dell =====
1022 id           name           type  size  mem  owner
1023 -----
1024 12           cookies          set 0    286   auth
1025 4111         code            set 281  12978 code_server
1026 8208         code_names       set 56   7510  code_server
1027 12305        ign_requests     set 0    286   inet_gethost_native
1028 16402        ign_req_index    set 0    286   inet_gethost_native
1029 ac_tab       ac_tab           set 20   1328  application_controller
1030 file_io_servers file_io_servers set 1    339   file_server_2
1031 global_locks  global_locks   set 0    286   global_name_server
1032 global_names  global_names   set 391  10469 global_name_server
1033 global_names_ext global_names_ext set 0    286   global_name_server
1034 global_pid_ids global_pid_ids bag 0    286   global_name_server
1035 global_pid_names global_pid_names bag 782  8864   global_name_server

```

```

1036 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1037 hm_table_global hm_table_global bag 444 31810 <0.323.0>
1038 inet_cache inet_cache bag 0 286 inet_db
1039 inet_db inet_db set 29 553 inet_db
1040 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1041 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1042 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1043 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1044 sys_dist sys_dist set 64 2868 net_kernel
1045 ===== dell14@dell =====
1046 id name type size mem owner
1047 -----
1048 12 cookies set 0 286 auth
1049 4111 code set 281 12978 code_server
1050 8208 code_names set 56 7510 code_server
1051 12305 ign_requests set 0 286 inet_gethost_native
1052 16402 ign_req_index set 0 286 inet_gethost_native
1053 ac_tab ac_tab set 20 1328 application_controller
1054 file_io_servers file_io_servers set 1 339 file_server_2
1055 global_locks global_locks set 0 286 global_name_server
1056 global_names global_names set 391 10437 global_name_server
1057 global_names_ext global_names_ext set 0 286 global_name_server
1058 global_pid_ids global_pid_ids bag 0 286 global_name_server
1059 global_pid_names global_pid_names bag 782 8864 global_name_server
1060 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1061 hm_table_global hm_table_global bag 1658 118004 <0.348.0>
1062 inet_cache inet_cache bag 0 286 inet_db
1063 inet_db inet_db set 29 553 inet_db
1064 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1065 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1066 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1067 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1068 sys_dist sys_dist set 64 2868 net_kernel
1069 ===== dell15@dell =====
1070 id name type size mem owner
1071 -----
1072 12 cookies set 0 286 auth
1073 4111 code set 281 12978 code_server
1074 8208 code_names set 56 7510 code_server
1075 12305 ign_requests set 0 286 inet_gethost_native
1076 16402 ign_req_index set 0 286 inet_gethost_native
1077 ac_tab ac_tab set 20 1328 application_controller
1078 file_io_servers file_io_servers set 1 339 file_server_2
1079 global_locks global_locks set 0 286 global_name_server
1080 global_names global_names set 391 10109 global_name_server
1081 global_names_ext global_names_ext set 0 286 global_name_server
1082 global_pid_ids global_pid_ids bag 0 286 global_name_server
1083 global_pid_names global_pid_names bag 782 8864 global_name_server
1084 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1085 hm_table_global hm_table_global bag 950 67736 <0.475.0>
1086 inet_cache inet_cache bag 0 286 inet_db
1087 inet_db inet_db set 29 553 inet_db
1088 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1089 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1090 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1091 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1092 sys_dist sys_dist set 64 2868 net_kernel
1093 ===== dell16@dell =====
1094 id name type size mem owner
1095 -----

```



```

1096 12 cookies set 0 286 auth
1097 4111 code set 281 12978 code_server
1098 8208 code_names set 56 7510 code_server
1099 12305 ign_requests set 0 286 inet_gethost_native
1100 16402 ign_req_index set 0 286 inet_gethost_native
1101 ac_tab ac_tab set 20 1328 application_controller
1102 file_io_servers file_io_servers set 1 339 file_server_2
1103 global_locks global_locks set 0 286 global_name_server
1104 global_names global_names set 391 10437 global_name_server
1105 global_names_ext global_names_ext set 0 286 global_name_server
1106 global_pid_ids global_pid_ids bag 0 286 global_name_server
1107 global_pid_names global_pid_names bag 782 8864 global_name_server
1108 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1109 hm_table_global hm_table_global bag 537 38413 <0.353.0>
1110 inet_cache inet_cache bag 0 286 inet_db
1111 inet_db inet_db set 29 553 inet_db
1112 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1113 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1114 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1115 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1116 sys_dist sys_dist set 64 2868 net_kernel
1117 ===== dell7@dell =====
1118 id name type size mem owner
1119 -----
1120 12 cookies set 0 286 auth
1121 4111 code set 281 12978 code_server
1122 8208 code_names set 56 7510 code_server
1123 12305 ign_requests set 0 286 inet_gethost_native
1124 16402 ign_req_index set 0 286 inet_gethost_native
1125 ac_tab ac_tab set 20 1328 application_controller
1126 file_io_servers file_io_servers set 1 339 file_server_2
1127 global_locks global_locks set 0 286 global_name_server
1128 global_names global_names set 391 10433 global_name_server
1129 global_names_ext global_names_ext set 0 286 global_name_server
1130 global_pid_ids global_pid_ids bag 0 286 global_name_server
1131 global_pid_names global_pid_names bag 782 8864 global_name_server
1132 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1133 hm_table_global hm_table_global bag 375 26911 <0.373.0>
1134 inet_cache inet_cache bag 0 286 inet_db
1135 inet_db inet_db set 29 553 inet_db
1136 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1137 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1138 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1139 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1140 sys_dist sys_dist set 64 2868 net_kernel
1141 ===== dell8@dell =====
1142 id name type size mem owner
1143 -----
1144 12 cookies set 0 286 auth
1145 4111 code set 281 12978 code_server
1146 8208 code_names set 56 7510 code_server
1147 12305 ign_requests set 0 286 inet_gethost_native
1148 16402 ign_req_index set 0 286 inet_gethost_native
1149 ac_tab ac_tab set 20 1328 application_controller
1150 file_io_servers file_io_servers set 1 339 file_server_2
1151 global_locks global_locks set 0 286 global_name_server
1152 global_names global_names set 391 9973 global_name_server
1153 global_names_ext global_names_ext set 0 286 global_name_server
1154 global_pid_ids global_pid_ids bag 0 286 global_name_server
1155 global_pid_names global_pid_names bag 782 8864 global_name_server

```

```

1156 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1157 hm_table_global hm_table_global bag 223 16119 <0.511.0>
1158 inet_cache inet_cache bag 0 286 inet_db
1159 inet_db inet_db set 29 553 inet_db
1160 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1161 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1162 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1163 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1164 sys_dist sys_dist set 64 2868 net_kernel
1165 ===== netlabal@netlaba =====
1166 id name type size mem owner
1167 -----
1168 9 cookies set 0 284 auth
1169 4108 code set 278 12787 code_server
1170 8205 code_names set 54 5196 code_server
1171 12302 ign_requests set 0 284 inet_gethost_native
1172 16399 ign_req_index set 0 284 inet_gethost_native
1173 ac_tab ac_tab set 20 1380 application_controller
1174 file_io_servers file_io_servers set 1 352 file_server_2
1175 global_locks global_locks set 0 284 global_name_server
1176 global_names global_names set 391 10896 global_name_server
1177 global_names_ext global_names_ext set 0 284 global_name_server
1178 global_pid_ids global_pid_ids bag 0 284 global_name_server
1179 global_pid_names global_pid_names bag 782 11208 global_name_server
1180 hm_ets_cache_table hm_ets_cache_table set 0 284 hm_cache_mgr
1181 hm_table_global hm_table_global bag 517 38542 <0.531.0>
1182 inet_cache inet_cache bag 0 284 inet_db
1183 inet_db inet_db set 21 512 inet_db
1184 inet_hosts inet_hosts set 0 284 inet_db
1185 sys_dist sys_dist set 64 3052 net_kernel
1186 ===== netlaba2@netlaba =====
1187 id name type size mem owner
1188 -----
1189 9 cookies set 0 284 auth
1190 4108 code set 278 12787 code_server
1191 8205 code_names set 54 5196 code_server
1192 12302 ign_requests set 0 284 inet_gethost_native
1193 16399 ign_req_index set 0 284 inet_gethost_native
1194 ac_tab ac_tab set 20 1380 application_controller
1195 file_io_servers file_io_servers set 1 352 file_server_2
1196 global_locks global_locks set 0 284 global_name_server
1197 global_names global_names set 391 10896 global_name_server
1198 global_names_ext global_names_ext set 0 284 global_name_server
1199 global_pid_ids global_pid_ids bag 0 284 global_name_server
1200 global_pid_names global_pid_names bag 782 11208 global_name_server
1201 hm_ets_cache_table hm_ets_cache_table set 0 284 hm_cache_mgr
1202 hm_table_global hm_table_global bag 1528 113356 <0.532.0>
1203 inet_cache inet_cache bag 0 284 inet_db
1204 inet_db inet_db set 21 512 inet_db
1205 inet_hosts inet_hosts set 0 284 inet_db
1206 sys_dist sys_dist set 64 3052 net_kernel
1207 ===== netlaba3@netlaba =====
1208 id name type size mem owner
1209 -----
1210 9 cookies set 0 284 auth
1211 4108 code set 278 12787 code_server
1212 8205 code_names set 54 5196 code_server
1213 12302 ign_requests set 0 284 inet_gethost_native
1214 16399 ign_req_index set 0 284 inet_gethost_native
1215 ac_tab ac_tab set 20 1380 application_controller

```

```

1216 file_io_servers file_io_servers set 1 352 file_server_2
1217 global_locks global_locks set 0 284 global_name_server
1218 global_names global_names set 391 10896 global_name_server
1219 global_names_ext global_names_ext set 0 284 global_name_server
1220 global_pid_ids global_pid_ids bag 0 284 global_name_server
1221 global_pid_names global_pid_names bag 782 11208 global_name_server
1222 hm_ets_cache_table hm_ets_cache_table set 0 284 hm_cache_mgr
1223 hm_table_global hm_table_global bag 600 44684 <0.531.0>
1224 inet_cache inet_cache bag 0 284 inet_db
1225 inet_db inet_db set 21 512 inet_db
1226 inet_hosts inet_hosts set 0 284 inet_db
1227 sys_dist sys_dist set 64 3052 net_kernel
1228 ===== netlaba4@netlaba =====
1229 id name type size mem owner
1230 -----
1231 9 cookies set 0 284 auth
1232 4108 code set 278 12787 code_server
1233 8205 code_names set 54 5196 code_server
1234 12302 ign_requests set 0 284 inet_gethost_native
1235 16399 ign_req_index set 0 284 inet_gethost_native
1236 'Domain1Tbl2' 'Domain1Tbl2' bag 10000 121821 <0.534.0>
1237 ac_tab ac_tab set 20 1380 application_controller
1238 file_io_servers file_io_servers set 1 352 file_server_2
1239 global_locks global_locks set 0 284 global_name_server
1240 global_names global_names set 391 10880 global_name_server
1241 global_names_ext global_names_ext set 0 284 global_name_server
1242 global_pid_ids global_pid_ids bag 0 284 global_name_server
1243 global_pid_names global_pid_names bag 782 11208 global_name_server
1244 hm_ets_cache_table hm_ets_cache_table set 0 284 hm_cache_mgr
1245 hm_table_global hm_table_global bag 664 49420 <0.533.0>
1246 inet_cache inet_cache bag 0 284 inet_db
1247 inet_db inet_db set 21 512 inet_db
1248 inet_hosts inet_hosts set 0 284 inet_db
1249 sys_dist sys_dist set 64 3052 net_kernel
1250 ===== netlaba5@netlaba =====
1251 id name type size mem owner
1252 -----
1253 9 cookies set 0 284 auth
1254 4108 code set 278 12787 code_server
1255 8205 code_names set 54 5196 code_server
1256 12302 ign_requests set 0 284 inet_gethost_native
1257 16399 ign_req_index set 0 284 inet_gethost_native
1258 ac_tab ac_tab set 20 1380 application_controller
1259 file_io_servers file_io_servers set 1 352 file_server_2
1260 global_locks global_locks set 0 284 global_name_server
1261 global_names global_names set 391 10888 global_name_server
1262 global_names_ext global_names_ext set 0 284 global_name_server
1263 global_pid_ids global_pid_ids bag 0 284 global_name_server
1264 global_pid_names global_pid_names bag 782 11208 global_name_server
1265 hm_ets_cache_table hm_ets_cache_table set 0 284 hm_cache_mgr
1266 hm_table_global hm_table_global bag 786 58448 <0.567.0>
1267 inet_cache inet_cache bag 0 284 inet_db
1268 inet_db inet_db set 21 512 inet_db
1269 inet_hosts inet_hosts set 0 284 inet_db
1270 sys_dist sys_dist set 64 3052 net_kernel
1271 ===== netlaba6@netlaba =====
1272 id name type size mem owner
1273 -----
1274 9 cookies set 0 284 auth
1275 4108 code set 278 12787 code_server

```

```

1276 8205          code_names      set    54    5196    code_server
1277 12302          ign_requests    set    0     284    inet_gethost_native
1278 16399          ign_req_index    set    0     284    inet_gethost_native
1279 ac_tab          ac_tab          set   20   1380    application_controller
1280 file_io_servers file_io_servers  set    1     352    file_server_2
1281 global_locks    global_locks    set    0     284    global_name_server
1282 global_names    global_names    set   391  10880    global_name_server
1283 global_names_ext global_names_ext set    0     284    global_name_server
1284 global_pid_ids  global_pid_ids  bag    0     284    global_name_server
1285 global_pid_names global_pid_names bag   782  11208    global_name_server
1286 hm_ets_cache_table hm_ets_cache_table set    0     284    hm_cache_mgr
1287 hm_table_global hm_table_global  bag   786  58448    <0.534.0>
1288 inet_cache      inet_cache      bag    0     284    inet_db
1289 inet_db         inet_db         set   21     512    inet_db
1290 inet_hosts      inet_hosts      set    0     284    inet_db
1291 sys_dist        sys_dist        set   64   3052    net_kernel
1292 ===== netlaba7@netlaba =====
1293 id              name              type    size    mem    owner
1294 -----
1295 9                cookies            set     0     284    auth
1296 4108             code              set   278  12787    code_server
1297 8205             code_names        set     54   5196    code_server
1298 12302            ign_requests      set     0     284    inet_gethost_native
1299 16399            ign_req_index     set     0     284    inet_gethost_native
1300 ac_tab           ac_tab            set    20   1380    application_controller
1301 file_io_servers  file_io_servers    set     1     352    file_server_2
1302 global_locks     global_locks       set     0     284    global_name_server
1303 global_names     global_names       set   391  10880    global_name_server
1304 global_names_ext global_names_ext   set     0     284    global_name_server
1305 global_pid_ids   global_pid_ids     bag     0     284    global_name_server
1306 global_pid_names global_pid_names   bag   782  11208    global_name_server
1307 hm_ets_cache_table hm_ets_cache_table set     0     284    hm_cache_mgr
1308 hm_table_global  hm_table_global    bag    570  42464    <0.532.0>
1309 inet_cache       inet_cache         bag     0     284    inet_db
1310 inet_db          inet_db           set    21     512    inet_db
1311 inet_hosts       inet_hosts         set     0     284    inet_db
1312 sys_dist         sys_dist          set    64   3052    net_kernel
1313 ===== netlaba8@netlaba =====
1314 id              name              type    size    mem    owner
1315 -----
1316 9                cookies            set     0     284    auth
1317 4108             code              set   278  12787    code_server
1318 8205             code_names        set     54   5196    code_server
1319 12302            ign_requests      set     0     284    inet_gethost_native
1320 16399            ign_req_index     set     0     284    inet_gethost_native
1321 ac_tab           ac_tab            set    20   1380    application_controller
1322 file_io_servers  file_io_servers    set     1     352    file_server_2
1323 global_locks     global_locks       set     0     284    global_name_server
1324 global_names     global_names       set   391  10880    global_name_server
1325 global_names_ext global_names_ext   set     0     284    global_name_server
1326 global_pid_ids   global_pid_ids     bag     0     284    global_name_server
1327 global_pid_names global_pid_names   bag   782  11208    global_name_server
1328 hm_ets_cache_table hm_ets_cache_table set     0     284    hm_cache_mgr
1329 hm_table_global  hm_table_global    bag    567  42242    <0.527.0>
1330 inet_cache       inet_cache         bag     0     284    inet_db
1331 inet_db          inet_db           set    21     512    inet_db
1332 inet_hosts       inet_hosts         set     0     284    inet_db
1333 sys_dist         sys_dist          set    64   3052    net_kernel
1334 ===== netlabbl@netlabbb =====
1335 id              name              type    size    mem    owner

```

```

1336 -----
1337 12          cookies          set 0      286      auth
1338 4111        code            set 281    12978   code_server
1339 8208        code_names      set 56    7510   code_server
1340 12305       ign_requests     set 0      286     inet_gethost_native
1341 16402       ign_req_index    set 0      286     inet_gethost_native
1342 ac_tab      ac_tab          set 21    1339   application_controller
1343 file_io_servers file_io_servers set 1      351     file_server_2
1344 global_locks global_locks    set 0      286     global_name_server
1345 global_names global_names    set 391   10193  global_name_server
1346 global_names_ext global_names_ext set 0      286     global_name_server
1347 global_pid_ids global_pid_ids    bag 0      286     global_name_server
1348 global_pid_names global_pid_names bag 782    8864   global_name_server
1349 hm_ets_cache_table hm_ets_cache_table set 0      286     hm_cache_mgr
1350 hm_table_global hm_table_global bag 1206   85912  <0.390.0>
1351 inet_cache   inet_cache     bag 0      286     inet_db
1352 inet_db      inet_db        set 29    559     inet_db
1353 inet_hosts_byaddr inet_hosts_byaddr bag 0      286     inet_db
1354 inet_hosts_byname inet_hosts_byname bag 0      286     inet_db
1355 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0      286     inet_db
1356 inet_hosts_file_byname inet_hosts_file_byname bag 0      286     inet_db
1357 sys_dist    sys_dist      set 64    2862   net_kernel
1358 ===== netlabbb2@netlabbb =====
1359 id          name          type  size  mem  owner
1360 -----
1361 12          cookies          set 0      286      auth
1362 4111        code            set 281    12978   code_server
1363 8208        code_names      set 56    7510   code_server
1364 12305       ign_requests     set 0      286     inet_gethost_native
1365 16402       ign_req_index    set 0      286     inet_gethost_native
1366 ac_tab      ac_tab          set 21    1339   application_controller
1367 file_io_servers file_io_servers set 1      351     file_server_2
1368 global_locks global_locks    set 0      286     global_name_server
1369 global_names global_names    set 391   10109  global_name_server
1370 global_names_ext global_names_ext set 0      286     global_name_server
1371 global_pid_ids global_pid_ids    bag 0      286     global_name_server
1372 global_pid_names global_pid_names bag 782    8864   global_name_server
1373 hm_ets_cache_table hm_ets_cache_table set 0      286     hm_cache_mgr
1374 hm_table_global hm_table_global bag 514    36780  <0.454.0>
1375 inet_cache   inet_cache     bag 0      286     inet_db
1376 inet_db      inet_db        set 29    559     inet_db
1377 inet_hosts_byaddr inet_hosts_byaddr bag 0      286     inet_db
1378 inet_hosts_byname inet_hosts_byname bag 0      286     inet_db
1379 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0      286     inet_db
1380 inet_hosts_file_byname inet_hosts_file_byname bag 0      286     inet_db
1381 sys_dist    sys_dist      set 64    2862   net_kernel
1382 ===== netlabbb3@netlabbb =====
1383 id          name          type  size  mem  owner
1384 -----
1385 12          cookies          set 0      286      auth
1386 4111        code            set 281    12978   code_server
1387 8208        code_names      set 56    7510   code_server
1388 12305       ign_requests     set 0      286     inet_gethost_native
1389 16402       ign_req_index    set 0      286     inet_gethost_native
1390 ac_tab      ac_tab          set 21    1339   application_controller
1391 file_io_servers file_io_servers set 1      351     file_server_2
1392 global_locks global_locks    set 0      286     global_name_server
1393 global_names global_names    set 391   9637   global_name_server
1394 global_names_ext global_names_ext set 0      286     global_name_server
1395 global_pid_ids global_pid_ids    bag 0      286     global_name_server

```

```

1396 global_pid_names global_pid_names bag 782 8864 global_name_server
1397 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1398 hm_table_global hm_table_global bag 809 57725 <0.620.0>
1399 inet_cache inet_cache bag 0 286 inet_db
1400 inet_db inet_db set 29 559 inet_db
1401 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1402 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1403 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1404 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1405 sys_dist sys_dist set 64 2862 net_kernel
1406 ===== netlab4@netlab4 =====
1407 id name type size mem owner
1408 -----
1409 12 cookies set 0 286 auth
1410 4111 code set 281 12978 code_server
1411 8208 code_names set 56 7510 code_server
1412 12305 ign_requests set 0 286 inet_gethost_native
1413 16402 ign_req_index set 0 286 inet_gethost_native
1414 ac_tab ac_tab set 21 1339 application_controller
1415 file_io_servers file_io_servers set 1 351 file_server_2
1416 global_locks global_locks set 0 286 global_name_server
1417 global_names global_names set 391 9641 global_name_server
1418 global_names_ext global_names_ext set 0 286 global_name_server
1419 global_pid_ids global_pid_ids bag 0 286 global_name_server
1420 global_pid_names global_pid_names bag 782 8864 global_name_server
1421 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1422 hm_table_global hm_table_global bag 924 65890 <0.604.0>
1423 inet_cache inet_cache bag 0 286 inet_db
1424 inet_db inet_db set 29 559 inet_db
1425 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1426 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1427 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1428 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1429 sys_dist sys_dist set 64 2862 net_kernel
1430 ===== netlab5@netlab5 =====
1431 id name type size mem owner
1432 -----
1433 12 cookies set 0 286 auth
1434 4111 code set 281 12978 code_server
1435 8208 code_names set 56 7510 code_server
1436 12305 ign_requests set 0 286 inet_gethost_native
1437 16402 ign_req_index set 0 286 inet_gethost_native
1438 ac_tab ac_tab set 21 1339 application_controller
1439 file_io_servers file_io_servers set 1 351 file_server_2
1440 global_locks global_locks set 0 286 global_name_server
1441 global_names global_names set 391 10109 global_name_server
1442 global_names_ext global_names_ext set 0 286 global_name_server
1443 global_pid_ids global_pid_ids bag 0 286 global_name_server
1444 global_pid_names global_pid_names bag 782 8864 global_name_server
1445 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1446 hm_table_global hm_table_global bag 245 17681 <0.421.0>
1447 inet_cache inet_cache bag 0 286 inet_db
1448 inet_db inet_db set 29 559 inet_db
1449 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1450 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1451 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1452 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1453 sys_dist sys_dist set 64 2862 net_kernel
1454 ===== netlab6@netlab6 =====
1455 id name type size mem owner

```

```

1456 -----
1457 12          cookies          set 0      286      auth
1458 4111        code             set 281    12978   code_server
1459 8208        code_names       set 56    7510    code_server
1460 12305       ign_requests     set 0      286     inet_gethost_native
1461 16402       ign_req_index    set 0      286     inet_gethost_native
1462 ac_tab      ac_tab           set 21    1339    application_controller
1463 file_io_servers file_io_servers set 1      351     file_server_2
1464 global_locks global_locks    set 0      286     global_name_server
1465 global_names global_names    set 391    9641    global_name_server
1466 global_names_ext global_names_ext set 0      286     global_name_server
1467 global_pid_ids global_pid_ids  bag 0      286     global_name_server
1468 global_pid_names global_pid_names bag 782    8864    global_name_server
1469 hm_ets_cache_table hm_ets_cache_table set 0      286     hm_cache_mgr
1470 hm_table_global hm_table_global bag 985    70221   <0.604.0>
1471 inet_cache   inet_cache     bag 0      286     inet_db
1472 inet_db      inet_db        set 29    559     inet_db
1473 inet_hosts_byaddr inet_hosts_byaddr bag 0      286     inet_db
1474 inet_hosts_byname inet_hosts_byname bag 0      286     inet_db
1475 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0      286     inet_db
1476 inet_hosts_file_byname inet_hosts_file_byname bag 0      286     inet_db
1477 sys_dist     sys_dist      set 64    2862    net_kernel
1478 ===== netlab7@netlab7 =====
1479 id          name          type  size  mem  owner
1480 -----
1481 12          cookies          set 0      286      auth
1482 4111        code             set 281    12978   code_server
1483 8208        code_names       set 56    7510    code_server
1484 12305       ign_requests     set 0      286     inet_gethost_native
1485 16402       ign_req_index    set 0      286     inet_gethost_native
1486 ac_tab      ac_tab           set 21    1339    application_controller
1487 file_io_servers file_io_servers set 1      351     file_server_2
1488 global_locks global_locks    set 0      286     global_name_server
1489 global_names global_names    set 391    9645    global_name_server
1490 global_names_ext global_names_ext set 0      286     global_name_server
1491 global_pid_ids global_pid_ids  bag 0      286     global_name_server
1492 global_pid_names global_pid_names bag 782    8864    global_name_server
1493 hm_ets_cache_table hm_ets_cache_table set 0      286     hm_cache_mgr
1494 hm_table_global hm_table_global bag 973    69369   <0.604.0>
1495 inet_cache   inet_cache     bag 0      286     inet_db
1496 inet_db      inet_db        set 29    559     inet_db
1497 inet_hosts_byaddr inet_hosts_byaddr bag 0      286     inet_db
1498 inet_hosts_byname inet_hosts_byname bag 0      286     inet_db
1499 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0      286     inet_db
1500 inet_hosts_file_byname inet_hosts_file_byname bag 0      286     inet_db
1501 sys_dist     sys_dist      set 64    2862    net_kernel
1502 ===== netlab8@netlab8 =====
1503 id          name          type  size  mem  owner
1504 -----
1505 12          cookies          set 0      286      auth
1506 4111        code             set 281    12978   code_server
1507 8208        code_names       set 56    7510    code_server
1508 12305       ign_requests     set 0      286     inet_gethost_native
1509 16402       ign_req_index    set 0      286     inet_gethost_native
1510 'Domain1Tbl2' 'Domain1Tbl2'  bag 10000  91824   <0.639.0>
1511 ac_tab      ac_tab           set 21    1339    application_controller
1512 file_io_servers file_io_servers set 1      351     file_server_2
1513 global_locks global_locks    set 0      286     global_name_server
1514 global_names global_names    set 391    9621    global_name_server
1515 global_names_ext global_names_ext set 0      286     global_name_server

```

```

1516 global_pid_ids global_pid_ids bag 0 286 global_name_server
1517 global_pid_names global_pid_names bag 782 8864 global_name_server
1518 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1519 hm_table_global hm_table_global bag 911 64967 <0.638.0>
1520 inet_cache inet_cache bag 0 286 inet_db
1521 inet_db inet_db set 29 559 inet_db
1522 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1523 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1524 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1525 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1526 sys_dist sys_dist set 64 2862 net_kernel
1527 ok
1528 (xxx_node@netlab3)4>

```

List 50: Test Results:Memory Usage:rstore function with long data

```

1 hirotnk@netlab3 ~/harmonia/Harmonia $ test/test20.sh -s
2 Launched the node:[netlab31@netlab3]
3 Launched the node:[netlab32@netlab3]
4 Launched the node:[netlab33@netlab3]
5 Launched the node:[netlab34@netlab3]
6 Launched the node:[netlab35@netlab3]
7 Launched the node:[netlab36@netlab3]
8 Launched the node:[netlab37@netlab3]
9 Launched the node:[netlab38@netlab3]
10 Launched the node:[netlab39@netlab3]
11 Launched the node:[netlab310@netlab3]
12 Launched the node:[netlab311@netlab3]
13 Launched the node:[netlab312@netlab3]
14 Launched the node:[netlab313@netlab3]
15 Launched the node:[netlab314@netlab3]
16 Launched the node:[netlab315@netlab3]
17 Launched the node:[netlab316@netlab3]
18 Launched the node:[netlab317@netlab3]
19 Launched the node:[netlab318@netlab3]
20 Launched the node:[netlab319@netlab3]
21 Launched the node:[netlab320@netlab3]
22 epmd: up and running on port 4369 with data:
23 name netlab315 at port 47235
24 name netlab316 at port 56275
25 name netlab314 at port 40308
26 name netlab313 at port 38916
27 name netlab312 at port 47697
28 name netlab311 at port 43661
29 name netlab310 at port 38361
30 name netlab38 at port 41764
31 name netlab39 at port 36329
32 name netlab37 at port 54912
33 name netlab36 at port 34150
34 name netlab35 at port 47077
35 name netlab34 at port 41331
36 name netlab33 at port 57458
37 name netlab32 at port 40668
38 name netlab31 at port 39735
39 hirotnk@netlab3 ~/harmonia/Harmonia $ erl -pa ebin -config harmonia -run hm_edge start -\
    -harmonia node_type 'join' name xxx sname 'xxx_node@netlab3' root_node 'netlab31@netlab3' \
    -setcookie harmonia_cookie -sname 'xxx_node@netlab3'
40 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
41
42 Eshell V5.6.3 (abort with ^G)

```



```

43 (xxx_node@netlab3)1> [{name,xxx},
44 {root_node,netlab31@netlab3},
45 {root,netlab31},
46 {included_applications,[]},
47 {sname,xxx_node@netlab3},
48 {logfile_ext,".txt"},
49 {node_type,join},
50 {logdir,"log/"},
51 {logfile,"harmonia_log"}]
52 "log/harmonia_log_xxx.txt"
53 start Pid:[<0.256.0>]
54
55 (xxx_node@netlab3)1> hm_cli:log_stop().
56 log stop:[dell5@dell] Result:[ok]
57 log stop:[dell6@dell] Result:[ok]
58 log stop:[dell2@dell] Result:[ok]
59 log stop:[dell1@dell] Result:[ok]
60 log stop:[dell4@dell] Result:[ok]
61 log stop:[dell8@dell] Result:[ok]
62 log stop:[dell3@dell] Result:[ok]
63 log stop:[dell7@dell] Result:[ok]
64 log stop:[netlab4@netlab] Result:[ok]
65 log stop:[netlaba8@netlaba] Result:[ok]
66 log stop:[netlab7@netlab] Result:[ok]
67 log stop:[netlab3@netlab] Result:[ok]
68 log stop:[netlab6@netlab] Result:[ok]
69 log stop:[netlab1@netlab] Result:[ok]
70 log stop:[netlaba6@netlaba] Result:[ok]
71 log stop:[netlab415@netlab4] Result:[ok]
72 log stop:[netlab2@netlab] Result:[ok]
73 log stop:[netlab8@netlab] Result:[ok]
74 log stop:[netlab5@netlab] Result:[ok]
75 log stop:[netlaba3@netlaba] Result:[ok]
76 log stop:[netlaba4@netlaba] Result:[ok]
77 log stop:[netlaba2@netlaba] Result:[ok]
78 log stop:[netlaba5@netlaba] Result:[ok]
79 log stop:[netlaba1@netlaba] Result:[ok]
80 log stop:[netlaba7@netlaba] Result:[ok]
81 log stop:[netlab45@netlab4] Result:[ok]
82 log stop:[netlab413@netlab4] Result:[ok]
83 log stop:[netlab48@netlab4] Result:[ok]
84 log stop:[netlab412@netlab4] Result:[ok]
85 log stop:[netlab420@netlab4] Result:[ok]
86 log stop:[netlab419@netlab4] Result:[ok]
87 log stop:[netlab42@netlab4] Result:[ok]
88 log stop:[netlab44@netlab4] Result:[ok]
89 log stop:[netlab411@netlab4] Result:[ok]
90 log stop:[netlab414@netlab4] Result:[ok]
91 log stop:[netlab43@netlab4] Result:[ok]
92 log stop:[netlab418@netlab4] Result:[ok]
93 log stop:[netlab417@netlab4] Result:[ok]
94 log stop:[netlab416@netlab4] Result:[ok]
95 log stop:[netlab49@netlab4] Result:[ok]
96 log stop:[netlab47@netlab4] Result:[ok]
97 log stop:[netlab410@netlab4] Result:[ok]
98 log stop:[netlab46@netlab4] Result:[ok]
99 log stop:[netlab41@netlab4] Result:[ok]
100 hm_log_h_file:closing 'log/harmonia_log_xxx.txt'
101 log stop:[xxx_node@netlab3] Result:[ok]
102 log stop:[netlab315@netlab3] Result:[ok]

```

```

103 log stop:[netlab320@netlab3] Result:[ok]
104 log stop:[netlab317@netlab3] Result:[ok]
105 log stop:[netlab319@netlab3] Result:[ok]
106 log stop:[netlab316@netlab3] Result:[ok]
107 log stop:[netlab38@netlab3] Result:[ok]
108 log stop:[netlab39@netlab3] Result:[ok]
109 log stop:[netlab313@netlab3] Result:[ok]
110 log stop:[netlab318@netlab3] Result:[ok]
111 log stop:[netlab36@netlab3] Result:[ok]
112 log stop:[netlab314@netlab3] Result:[ok]
113 log stop:[netlab312@netlab3] Result:[ok]
114 log stop:[netlab311@netlab3] Result:[ok]
115 log stop:[netlab35@netlab3] Result:[ok]
116 log stop:[netlab310@netlab3] Result:[ok]
117 log stop:[netlab37@netlab3] Result:[ok]
118 log stop:[netlab33@netlab3] Result:[ok]
119 log stop:[netlab34@netlab3] Result:[ok]
120 log stop:[netlab32@netlab3] Result:[ok]
121 log stop:[netlab31@netlab3] Result:[ok]
122 ok
123 (xxx_node@netlab3)2> hm_cli_test:test_all_long(10000).
124 starting....
125 create_table_long "OK"....
126 rstore_long(10000)      OK....
127 ..end
128 ok
129 (xxx_node@netlab3)3>
130 (xxx_node@netlab3)3>
131 (xxx_node@netlab3)3>
132 (xxx_node@netlab3)3>
133 (xxx_node@netlab3)3>
134 (xxx_node@netlab3)3>
135 (xxx_node@netlab3)3>
136 (xxx_node@netlab3)3>
137 (xxx_node@netlab3)3>
138 (xxx_node@netlab3)3>
139 (xxx_node@netlab3)3>
140 (xxx_node@netlab3)3>
141 (xxx_node@netlab3)3>
142 (xxx_node@netlab3)3>
143 (xxx_node@netlab3)3>
144 (xxx_node@netlab3)3>
145 (xxx_node@netlab3)3>
146 (xxx_node@netlab3)3> hm_cli_test:check_size().
147 ===== netlab31@netlab3 =====
148 id            name                type  size  mem  owner
149 -----
150 10             cookies                set   0     279  auth
151 13             code                   set   278   12598 code_server
152 14             code_names             set   48    4649 code_server
153 ac_tab         ac_tab                 set   20    1389 application_controller
154 file_io_servers file_io_servers        set   1     347 file_server_2
155 global_locks   global_locks           set   0     279 global_name_server
156 global_names   global_names           set   391   11843 global_name_server
157 global_names_ext global_names_ext       set   0     279 global_name_server
158 global_pid_ids global_pid_ids         bag   0     279 global_name_server
159 global_pid_names global_pid_names       bag   782   11199 global_name_server
160 hm_ets_cache_table hm_ets_cache_table set   0     279 hm_cache_mgr
161 hm_table_global hm_table_global       bag   517   50945 <0.52.0>
162 inet_cache     inet_cache            bag   0     279 inet_db

```

```

163 inet_db          inet_db          set 21    541    inet_db
164 inet_hosts       inet_hosts       set 0     279    inet_db
165 sys_dist         sys_dist         set 64   3047   net_kernel
166 ===== netlab32@netlab3 =====
167 id              name              type  size    mem    owner
168 -----
169 10              cookies           set 0      279    auth
170 13              code              set 279    12706  code_server
171 14              code_names        set 48     4649   code_server
172 15              ign_requests      set 0      279    inet_gethost_native
173 16              ign_req_index     set 0      279    inet_gethost_native
174 ac_tab          ac_tab           set 20     1389   application_controller
175 file_io_servers file_io_servers   set 1      347    file_server_2
176 global_locks     global_locks      set 0      279    global_name_server
177 global_names     global_names      set 391    11851  global_name_server
178 global_names_ext global_names_ext  set 0      279    global_name_server
179 global_pid_ids   global_pid_ids    bag 0      279    global_name_server
180 global_pid_names global_pid_names  bag 782    11203  global_name_server
181 hm_ets_cache_table hm_ets_cache_table set 0      279    hm_cache_mgr
182 hm_table_global hm_table_global  bag 438    43203  <0.95.0>
183 inet_cache       inet_cache        bag 0      279    inet_db
184 inet_db          inet_db          set 21    541    inet_db
185 inet_hosts       inet_hosts       set 0     279    inet_db
186 sys_dist         sys_dist         set 64   3047   net_kernel
187 ===== netlab33@netlab3 =====
188 id              name              type  size    mem    owner
189 -----
190 10              cookies           set 0      279    auth
191 13              code              set 279    12706  code_server
192 14              code_names        set 48     4649   code_server
193 15              ign_requests      set 0      279    inet_gethost_native
194 16              ign_req_index     set 0      279    inet_gethost_native
195 ac_tab          ac_tab           set 20     1389   application_controller
196 file_io_servers file_io_servers   set 1      347    file_server_2
197 global_locks     global_locks      set 0      279    global_name_server
198 global_names     global_names      set 391    11851  global_name_server
199 global_names_ext global_names_ext  set 0      279    global_name_server
200 global_pid_ids   global_pid_ids    bag 0      279    global_name_server
201 global_pid_names global_pid_names  bag 782    11203  global_name_server
202 hm_ets_cache_table hm_ets_cache_table set 0      279    hm_cache_mgr
203 hm_table_global hm_table_global  bag 844    82991  <0.94.0>
204 inet_cache       inet_cache        bag 0      279    inet_db
205 inet_db          inet_db          set 21    541    inet_db
206 inet_hosts       inet_hosts       set 0     279    inet_db
207 sys_dist         sys_dist         set 64   3047   net_kernel
208 ===== netlab34@netlab3 =====
209 id              name              type  size    mem    owner
210 -----
211 10              cookies           set 0      279    auth
212 13              code              set 279    12706  code_server
213 14              code_names        set 48     4649   code_server
214 15              ign_requests      set 0      279    inet_gethost_native
215 16              ign_req_index     set 0      279    inet_gethost_native
216 ac_tab          ac_tab           set 20     1389   application_controller
217 file_io_servers file_io_servers   set 1      347    file_server_2
218 global_locks     global_locks      set 0      279    global_name_server
219 global_names     global_names      set 391    11851  global_name_server
220 global_names_ext global_names_ext  set 0      279    global_name_server
221 global_pid_ids   global_pid_ids    bag 0      279    global_name_server
222 global_pid_names global_pid_names  bag 782    11203  global_name_server

```

```

223 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
224 hm_table_global hm_table_global bag 1033 101513 <0.98.0>
225 inet_cache inet_cache bag 0 279 inet_db
226 inet_db inet_db set 21 541 inet_db
227 inet_hosts inet_hosts set 0 279 inet_db
228 sys_dist sys_dist set 64 3047 net_kernel
229 ===== netlab35@netlab3 =====
230 id name type size mem owner
231 -----
232 10 cookies set 0 279 auth
233 13 code set 279 12706 code_server
234 14 code_names set 48 4649 code_server
235 15 ign_requests set 0 279 inet_gethost_native
236 16 ign_req_index set 0 279 inet_gethost_native
237 ac_tab ac_tab set 20 1389 application_controller
238 file_io_servers file_io_servers set 1 347 file_server_2
239 global_locks global_locks set 0 279 global_name_server
240 global_names global_names set 391 11851 global_name_server
241 global_names_ext global_names_ext set 0 279 global_name_server
242 global_pid_ids global_pid_ids bag 0 279 global_name_server
243 global_pid_names global_pid_names bag 782 11203 global_name_server
244 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
245 hm_table_global hm_table_global bag 746 73387 <0.109.0>
246 inet_cache inet_cache bag 0 279 inet_db
247 inet_db inet_db set 21 541 inet_db
248 inet_hosts inet_hosts set 0 279 inet_db
249 sys_dist sys_dist set 64 3047 net_kernel
250 ===== netlab36@netlab3 =====
251 id name type size mem owner
252 -----
253 10 cookies set 0 279 auth
254 13 code set 279 12706 code_server
255 14 code_names set 48 4649 code_server
256 15 ign_requests set 0 279 inet_gethost_native
257 16 ign_req_index set 0 279 inet_gethost_native
258 ac_tab ac_tab set 20 1389 application_controller
259 file_io_servers file_io_servers set 1 347 file_server_2
260 global_locks global_locks set 0 279 global_name_server
261 global_names global_names set 391 11851 global_name_server
262 global_names_ext global_names_ext set 0 279 global_name_server
263 global_pid_ids global_pid_ids bag 0 279 global_name_server
264 global_pid_names global_pid_names bag 782 11203 global_name_server
265 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
266 hm_table_global hm_table_global bag 770 75739 <0.121.0>
267 inet_cache inet_cache bag 0 279 inet_db
268 inet_db inet_db set 21 541 inet_db
269 inet_hosts inet_hosts set 0 279 inet_db
270 sys_dist sys_dist set 64 3047 net_kernel
271 ===== netlab37@netlab3 =====
272 id name type size mem owner
273 -----
274 10 cookies set 0 279 auth
275 13 code set 279 12706 code_server
276 14 code_names set 48 4649 code_server
277 15 ign_requests set 0 279 inet_gethost_native
278 16 ign_req_index set 0 279 inet_gethost_native
279 ac_tab ac_tab set 20 1389 application_controller
280 file_io_servers file_io_servers set 1 347 file_server_2
281 global_locks global_locks set 0 279 global_name_server
282 global_names global_names set 391 11851 global_name_server

```

```

283 global_names_ext global_names_ext set 0 279 global_name_server
284 global_pid_ids global_pid_ids bag 0 279 global_name_server
285 global_pid_names global_pid_names bag 782 11203 global_name_server
286 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
287 hm_table_global hm_table_global bag 1604 157471 <0.103.0>
288 inet_cache inet_cache bag 0 279 inet_db
289 inet_db inet_db set 21 541 inet_db
290 inet_hosts inet_hosts set 0 279 inet_db
291 sys_dist sys_dist set 64 3047 net_kernel
292 ===== netlab38@netlab3 =====
293 id name type size mem owner
294 -----
295 10 cookies set 0 279 auth
296 13 code set 279 12706 code_server
297 14 code_names set 48 4649 code_server
298 15 ign_requests set 0 279 inet_gethost_native
299 16 ign_req_index set 0 279 inet_gethost_native
300 ac_tab ac_tab set 20 1389 application_controller
301 file_io_servers file_io_servers set 1 347 file_server_2
302 global_locks global_locks set 0 279 global_name_server
303 global_names global_names set 391 11851 global_name_server
304 global_names_ext global_names_ext set 0 279 global_name_server
305 global_pid_ids global_pid_ids bag 0 279 global_name_server
306 global_pid_names global_pid_names bag 782 11203 global_name_server
307 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
308 hm_table_global hm_table_global bag 1199 117781 <0.124.0>
309 inet_cache inet_cache bag 0 279 inet_db
310 inet_db inet_db set 21 541 inet_db
311 inet_hosts inet_hosts set 0 279 inet_db
312 sys_dist sys_dist set 64 3047 net_kernel
313 ===== netlab39@netlab3 =====
314 id name type size mem owner
315 -----
316 10 cookies set 0 279 auth
317 13 code set 279 12706 code_server
318 14 code_names set 48 4649 code_server
319 15 ign_requests set 0 279 inet_gethost_native
320 16 ign_req_index set 0 279 inet_gethost_native
321 ac_tab ac_tab set 20 1389 application_controller
322 file_io_servers file_io_servers set 1 347 file_server_2
323 global_locks global_locks set 0 279 global_name_server
324 global_names global_names set 391 11851 global_name_server
325 global_names_ext global_names_ext set 0 279 global_name_server
326 global_pid_ids global_pid_ids bag 0 279 global_name_server
327 global_pid_names global_pid_names bag 782 11203 global_name_server
328 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
329 hm_table_global hm_table_global bag 738 72603 <0.126.0>
330 inet_cache inet_cache bag 0 279 inet_db
331 inet_db inet_db set 21 541 inet_db
332 inet_hosts inet_hosts set 0 279 inet_db
333 sys_dist sys_dist set 64 3047 net_kernel
334 ===== netlab310@netlab3 =====
335 id name type size mem owner
336 -----
337 10 cookies set 0 279 auth
338 13 code set 279 12706 code_server
339 14 code_names set 48 4649 code_server
340 15 ign_requests set 0 279 inet_gethost_native
341 16 ign_req_index set 0 279 inet_gethost_native
342 ac_tab ac_tab set 20 1389 application_controller

```

```

343 file_io_servers file_io_servers set 1 349 file_server_2
344 global_locks global_locks set 0 279 global_name_server
345 global_names global_names set 391 11851 global_name_server
346 global_names_ext global_names_ext set 0 279 global_name_server
347 global_pid_ids global_pid_ids bag 0 279 global_name_server
348 global_pid_names global_pid_names bag 782 11203 global_name_server
349 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
350 hm_table_global hm_table_global bag 464 45751 <0.119.0>
351 inet_cache inet_cache bag 0 279 inet_db
352 inet_db inet_db set 21 541 inet_db
353 inet_hosts inet_hosts set 0 279 inet_db
354 sys_dist sys_dist set 64 3047 net_kernel
355 ===== netlab311@netlab3 =====
356 id name type size mem owner
357 -----
358 10 cookies set 0 279 auth
359 13 code set 279 12706 code_server
360 14 code_names set 48 4649 code_server
361 15 ign_requests set 0 279 inet_gethost_native
362 16 ign_req_index set 0 279 inet_gethost_native
363 ac_tab ac_tab set 20 1389 application_controller
364 file_io_servers file_io_servers set 1 349 file_server_2
365 global_locks global_locks set 0 279 global_name_server
366 global_names global_names set 391 11807 global_name_server
367 global_names_ext global_names_ext set 0 279 global_name_server
368 global_pid_ids global_pid_ids bag 0 279 global_name_server
369 global_pid_names global_pid_names bag 782 11203 global_name_server
370 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
371 hm_table_global hm_table_global bag 1625 159529 <0.125.0>
372 inet_cache inet_cache bag 0 279 inet_db
373 inet_db inet_db set 21 541 inet_db
374 inet_hosts inet_hosts set 0 279 inet_db
375 sys_dist sys_dist set 64 3047 net_kernel
376 ===== netlab312@netlab3 =====
377 id name type size mem owner
378 -----
379 10 cookies set 0 279 auth
380 13 code set 279 12706 code_server
381 14 code_names set 48 4649 code_server
382 15 ign_requests set 0 279 inet_gethost_native
383 16 ign_req_index set 0 279 inet_gethost_native
384 ac_tab ac_tab set 20 1389 application_controller
385 file_io_servers file_io_servers set 1 349 file_server_2
386 global_locks global_locks set 0 279 global_name_server
387 global_names global_names set 391 11779 global_name_server
388 global_names_ext global_names_ext set 0 279 global_name_server
389 global_pid_ids global_pid_ids bag 0 279 global_name_server
390 global_pid_names global_pid_names bag 782 11203 global_name_server
391 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
392 hm_table_global hm_table_global bag 966 94947 <0.154.0>
393 inet_cache inet_cache bag 0 279 inet_db
394 inet_db inet_db set 21 541 inet_db
395 inet_hosts inet_hosts set 0 279 inet_db
396 sys_dist sys_dist set 64 3047 net_kernel
397 ===== netlab313@netlab3 =====
398 id name type size mem owner
399 -----
400 10 cookies set 0 279 auth
401 13 code set 279 12706 code_server
402 14 code_names set 48 4649 code_server

```

```

403 15          ign_requests      set 0      279      inet_gethost_native
404 16          ign_req_index      set 0      279      inet_gethost_native
405 ac_tab      ac_tab            set 20     1389     application_controller
406 file_io_servers file_io_servers set 1      349      file_server_2
407 global_locks global_locks      set 0      279      global_name_server
408 global_names global_names      set 391    11807    global_name_server
409 global_names_ext global_names_ext set 0      279      global_name_server
410 global_pid_ids global_pid_ids      bag 0      279      global_name_server
411 global_pid_names global_pid_names bag 782     11203    global_name_server
412 hm_ets_cache_table hm_ets_cache_table set 0      279      hm_cache_mgr
413 hm_table_global hm_table_global bag 532     52415    <0.148.0>
414 inet_cache   inet_cache      bag 0      279      inet_db
415 inet_db      inet_db            set 21     541      inet_db
416 inet_hosts   inet_hosts      set 0      279      inet_db
417 sys_dist     sys_dist          set 64     3047     net_kernel
418 ===== netlab314@netlab3 =====
419 id           name                type  size  mem  owner
420 -----
421 10           cookies             set 0      279      auth
422 13           code               set 279    12706    code_server
423 14           code_names         set 48     4649     code_server
424 15           ign_requests       set 0      279      inet_gethost_native
425 16           ign_req_index      set 0      279      inet_gethost_native
426 ac_tab      ac_tab            set 20     1389     application_controller
427 file_io_servers file_io_servers set 1      349      file_server_2
428 global_locks global_locks      set 0      279      global_name_server
429 global_names global_names      set 391    11671    global_name_server
430 global_names_ext global_names_ext set 0      279      global_name_server
431 global_pid_ids global_pid_ids      bag 0      279      global_name_server
432 global_pid_names global_pid_names bag 782     11203    global_name_server
433 hm_ets_cache_table hm_ets_cache_table set 0      279      hm_cache_mgr
434 hm_table_global hm_table_global bag 714     70251    <0.177.0>
435 inet_cache   inet_cache      bag 0      279      inet_db
436 inet_db      inet_db            set 21     541      inet_db
437 inet_hosts   inet_hosts      set 0      279      inet_db
438 sys_dist     sys_dist          set 64     3047     net_kernel
439 ===== netlab315@netlab3 =====
440 id           name                type  size  mem  owner
441 -----
442 10           cookies             set 0      279      auth
443 13           code               set 279    12706    code_server
444 14           code_names         set 48     4649     code_server
445 15           ign_requests       set 0      279      inet_gethost_native
446 16           ign_req_index      set 0      279      inet_gethost_native
447 ac_tab      ac_tab            set 20     1389     application_controller
448 file_io_servers file_io_servers set 1      349      file_server_2
449 global_locks global_locks      set 0      279      global_name_server
450 global_names global_names      set 391    11731    global_name_server
451 global_names_ext global_names_ext set 0      279      global_name_server
452 global_pid_ids global_pid_ids      bag 0      279      global_name_server
453 global_pid_names global_pid_names bag 782     11203    global_name_server
454 hm_ets_cache_table hm_ets_cache_table set 0      279      hm_cache_mgr
455 hm_table_global hm_table_global bag 539     53101    <0.168.0>
456 inet_cache   inet_cache      bag 0      279      inet_db
457 inet_db      inet_db            set 21     541      inet_db
458 inet_hosts   inet_hosts      set 0      279      inet_db
459 sys_dist     sys_dist          set 64     3047     net_kernel
460 ===== netlab316@netlab3 =====
461 id           name                type  size  mem  owner
462 -----

```

```

463 10          cookies          set 0      279      auth
464 13          code             set 279    12706   code_server
465 14          code_names       set 48     4649   code_server
466 15          ign_requests     set 0      279    inet_gethost_native
467 16          ign_req_index     set 0      279    inet_gethost_native
468 ac_tab      ac_tab           set 20     1389   application_controller
469 file_io_servers file_io_servers set 1      349    file_server_2
470 global_locks global_locks    set 0      279    global_name_server
471 global_names global_names     set 391    11731  global_name_server
472 global_names_ext global_names_ext set 0      279    global_name_server
473 global_pid_ids global_pid_ids bag 0      279    global_name_server
474 global_pid_names global_pid_names bag 782    11203  global_name_server
475 hm_ets_cache_table hm_ets_cache_table set 0      279    hm_cache_mgr
476 hm_table_global hm_table_global bag 714    70251  <0.158.0>
477 inet_cache     inet_cache     bag 0      279    inet_db
478 inet_db        inet_db        set 21     541    inet_db
479 inet_hosts     inet_hosts     set 0      279    inet_db
480 sys_dist       sys_dist       set 64     3047   net_kernel
481 ===== netlab317@netlab3 =====
482 id            name            type  size    mem    owner
483 -----
484 10            cookies          set 0      279    auth
485 13            code             set 279    12706   code_server
486 14            code_names       set 48     4649   code_server
487 15            ign_requests     set 0      279    inet_gethost_native
488 16            ign_req_index     set 0      279    inet_gethost_native
489 ac_tab        ac_tab           set 20     1389   application_controller
490 file_io_servers file_io_servers set 1      349    file_server_2
491 global_locks global_locks    set 0      279    global_name_server
492 global_names global_names     set 391    11671  global_name_server
493 global_names_ext global_names_ext set 0      279    global_name_server
494 global_pid_ids global_pid_ids bag 0      279    global_name_server
495 global_pid_names global_pid_names bag 782    11203  global_name_server
496 hm_ets_cache_table hm_ets_cache_table set 0      279    hm_cache_mgr
497 hm_table_global hm_table_global bag 808    79463  <0.183.0>
498 inet_cache     inet_cache     bag 0      279    inet_db
499 inet_db        inet_db        set 21     541    inet_db
500 inet_hosts     inet_hosts     set 0      279    inet_db
501 sys_dist       sys_dist       set 64     3047   net_kernel
502 ===== netlab318@netlab3 =====
503 id            name            type  size    mem    owner
504 -----
505 10            cookies          set 0      279    auth
506 13            code             set 279    12706   code_server
507 14            code_names       set 48     4649   code_server
508 15            ign_requests     set 0      279    inet_gethost_native
509 16            ign_req_index     set 0      279    inet_gethost_native
510 'Domain1Tbl2' 'Domain1Tbl2' bag 10000  361814 <0.180.0>
511 ac_tab        ac_tab           set 20     1389   application_controller
512 file_io_servers file_io_servers set 1      349    file_server_2
513 global_locks global_locks    set 0      279    global_name_server
514 global_names global_names     set 391    11671  global_name_server
515 global_names_ext global_names_ext set 0      279    global_name_server
516 global_pid_ids global_pid_ids bag 0      279    global_name_server
517 global_pid_names global_pid_names bag 782    11203  global_name_server
518 hm_ets_cache_table hm_ets_cache_table set 0      279    hm_cache_mgr
519 hm_table_global hm_table_global bag 723    71133  <0.179.0>
520 inet_cache     inet_cache     bag 0      279    inet_db
521 inet_db        inet_db        set 21     541    inet_db
522 inet_hosts     inet_hosts     set 0      279    inet_db

```



```

523 sys_dist      sys_dist      set    64    3047    net_kernel
524 ===== netlab319@netlab3 =====
525 id            name            type    size    mem    owner
526 -----
527 10            cookies        set     0      279    auth
528 13            code           set    279    12706  code_server
529 14            code_names     set     48    4649   code_server
530 15            ign_requests   set     0      279    inet_gethost_native
531 16            ign_req_index   set     0      279    inet_gethost_native
532 'Domain1Tbl2' 'Domain1Tbl2'   bag   10000  361814 <0.180.0>
533 ac_tab        ac_tab         set     20    1389   application_controller
534 file_io_servers file_io_servers set     1      349    file_server_2
535 global_locks  global_locks   set     0      279    global_name_server
536 global_names  global_names   set    391   11671  global_name_server
537 global_names_ext global_names_ext set     0      279    global_name_server
538 global_pid_ids global_pid_ids bag     0      279    global_name_server
539 global_pid_names global_pid_names bag    782   11203  global_name_server
540 hm_ets_cache_table hm_ets_cache_table set     0      279    hm_cache_mgr
541 hm_table_global hm_table_global bag    443   43693  <0.179.0>
542 inet_cache    inet_cache     bag     0      279    inet_db
543 inet_db       inet_db        set     21    541    inet_db
544 inet_hosts    inet_hosts     set     0      279    inet_db
545 sys_dist      sys_dist      set     64    3047    net_kernel
546 ===== netlab320@netlab3 =====
547 id            name            type    size    mem    owner
548 -----
549 10            cookies        set     0      279    auth
550 13            code           set    279    12706  code_server
551 14            code_names     set     48    4649   code_server
552 15            ign_requests   set     0      279    inet_gethost_native
553 16            ign_req_index   set     0      279    inet_gethost_native
554 'Domain1Tbl2' 'Domain1Tbl2'   bag   10000  361814 <0.173.0>
555 ac_tab        ac_tab         set     20    1389   application_controller
556 file_io_servers file_io_servers set     1      349    file_server_2
557 global_locks  global_locks   set     0      279    global_name_server
558 global_names  global_names   set    391   11671  global_name_server
559 global_names_ext global_names_ext set     0      279    global_name_server
560 global_pid_ids global_pid_ids bag     0      279    global_name_server
561 global_pid_names global_pid_names bag    782   11203  global_name_server
562 hm_ets_cache_table hm_ets_cache_table set     0      279    hm_cache_mgr
563 hm_table_global hm_table_global bag    362   35755  <0.172.0>
564 inet_cache    inet_cache     bag     0      279    inet_db
565 inet_db       inet_db        set     21    541    inet_db
566 inet_hosts    inet_hosts     set     0      279    inet_db
567 sys_dist      sys_dist      set     64    3047    net_kernel
568 ===== netlab41@netlab4 =====
569 id            name            type    size    mem    owner
570 -----
571 10            cookies        set     0      279    auth
572 13            code           set    279    12706  code_server
573 14            code_names     set     48    4649   code_server
574 15            ign_requests   set     0      279    inet_gethost_native
575 16            ign_req_index   set     0      279    inet_gethost_native
576 ac_tab        ac_tab         set     20    1389   application_controller
577 file_io_servers file_io_servers set     1      347    file_server_2
578 global_locks  global_locks   set     0      279    global_name_server
579 global_names  global_names   set    391   11371  global_name_server
580 global_names_ext global_names_ext set     0      279    global_name_server
581 global_pid_ids global_pid_ids bag     0      279    global_name_server
582 global_pid_names global_pid_names bag    782   11203  global_name_server

```

```

583 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
584 hm_table_global hm_table_global bag 894 87891 <0.306.0>
585 inet_cache inet_cache bag 0 279 inet_db
586 inet_db inet_db set 21 541 inet_db
587 inet_hosts inet_hosts set 0 279 inet_db
588 sys_dist sys_dist set 64 3047 net_kernel
589 ===== netlab42@netlab4 =====
590 id name type size mem owner
591 -----
592 10 cookies set 0 279 auth
593 13 code set 279 12706 code_server
594 14 code_names set 48 4649 code_server
595 15 ign_requests set 0 279 inet_gethost_native
596 16 ign_req_index set 0 279 inet_gethost_native
597 ac_tab ac_tab set 20 1389 application_controller
598 file_io_servers file_io_servers set 1 347 file_server_2
599 global_locks global_locks set 0 279 global_name_server
600 global_names global_names set 391 11371 global_name_server
601 global_names_ext global_names_ext set 0 279 global_name_server
602 global_pid_ids global_pid_ids bag 0 279 global_name_server
603 global_pid_names global_pid_names bag 782 11203 global_name_server
604 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
605 hm_table_global hm_table_global bag 465 45849 <0.352.0>
606 inet_cache inet_cache bag 0 279 inet_db
607 inet_db inet_db set 21 541 inet_db
608 inet_hosts inet_hosts set 0 279 inet_db
609 sys_dist sys_dist set 64 3047 net_kernel
610 ===== netlab43@netlab4 =====
611 id name type size mem owner
612 -----
613 10 cookies set 0 279 auth
614 13 code set 279 12706 code_server
615 14 code_names set 48 4649 code_server
616 15 ign_requests set 0 279 inet_gethost_native
617 16 ign_req_index set 0 279 inet_gethost_native
618 ac_tab ac_tab set 20 1389 application_controller
619 file_io_servers file_io_servers set 1 347 file_server_2
620 global_locks global_locks set 0 279 global_name_server
621 global_names global_names set 391 11371 global_name_server
622 global_names_ext global_names_ext set 0 279 global_name_server
623 global_pid_ids global_pid_ids bag 0 279 global_name_server
624 global_pid_names global_pid_names bag 782 11203 global_name_server
625 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
626 hm_table_global hm_table_global bag 370 36539 <0.351.0>
627 inet_cache inet_cache bag 0 279 inet_db
628 inet_db inet_db set 21 541 inet_db
629 inet_hosts inet_hosts set 0 279 inet_db
630 sys_dist sys_dist set 64 3047 net_kernel
631 ===== netlab44@netlab4 =====
632 id name type size mem owner
633 -----
634 10 cookies set 0 279 auth
635 13 code set 279 12706 code_server
636 14 code_names set 48 4649 code_server
637 15 ign_requests set 0 279 inet_gethost_native
638 16 ign_req_index set 0 279 inet_gethost_native
639 ac_tab ac_tab set 20 1389 application_controller
640 file_io_servers file_io_servers set 1 347 file_server_2
641 global_locks global_locks set 0 279 global_name_server
642 global_names global_names set 391 11371 global_name_server

```

```

643 global_names_ext global_names_ext set 0 279 global_name_server
644 global_pid_ids global_pid_ids bag 0 279 global_name_server
645 global_pid_names global_pid_names bag 782 11203 global_name_server
646 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
647 hm_table_global hm_table_global bag 1078 105923 <0.328.0>
648 inet_cache inet_cache bag 0 279 inet_db
649 inet_db inet_db set 21 541 inet_db
650 inet_hosts inet_hosts set 0 279 inet_db
651 sys_dist sys_dist set 64 3047 net_kernel
652 ===== netlab45@netlab4 =====
653 id name type size mem owner
654 -----
655 10 cookies set 0 279 auth
656 13 code set 279 12706 code_server
657 14 code_names set 48 4649 code_server
658 15 ign_requests set 0 279 inet_gethost_native
659 16 ign_req_index set 0 279 inet_gethost_native
660 ac_tab ac_tab set 20 1389 application_controller
661 file_io_servers file_io_servers set 1 347 file_server_2
662 global_locks global_locks set 0 279 global_name_server
663 global_names global_names set 391 11371 global_name_server
664 global_names_ext global_names_ext set 0 279 global_name_server
665 global_pid_ids global_pid_ids bag 0 279 global_name_server
666 global_pid_names global_pid_names bag 782 11203 global_name_server
667 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
668 hm_table_global hm_table_global bag 640 62999 <0.356.0>
669 inet_cache inet_cache bag 0 279 inet_db
670 inet_db inet_db set 21 541 inet_db
671 inet_hosts inet_hosts set 0 279 inet_db
672 sys_dist sys_dist set 64 3047 net_kernel
673 ===== netlab46@netlab4 =====
674 id name type size mem owner
675 -----
676 10 cookies set 0 279 auth
677 13 code set 279 12706 code_server
678 14 code_names set 48 4649 code_server
679 15 ign_requests set 0 279 inet_gethost_native
680 16 ign_req_index set 0 279 inet_gethost_native
681 ac_tab ac_tab set 20 1389 application_controller
682 file_io_servers file_io_servers set 1 347 file_server_2
683 global_locks global_locks set 0 279 global_name_server
684 global_names global_names set 391 11371 global_name_server
685 global_names_ext global_names_ext set 0 279 global_name_server
686 global_pid_ids global_pid_ids bag 0 279 global_name_server
687 global_pid_names global_pid_names bag 782 11203 global_name_server
688 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
689 hm_table_global hm_table_global bag 704 69271 <0.305.0>
690 inet_cache inet_cache bag 0 279 inet_db
691 inet_db inet_db set 21 541 inet_db
692 inet_hosts inet_hosts set 0 279 inet_db
693 sys_dist sys_dist set 64 3047 net_kernel
694 ===== netlab47@netlab4 =====
695 id name type size mem owner
696 -----
697 10 cookies set 0 279 auth
698 13 code set 279 12706 code_server
699 14 code_names set 48 4649 code_server
700 15 ign_requests set 0 279 inet_gethost_native
701 16 ign_req_index set 0 279 inet_gethost_native
702 ac_tab ac_tab set 20 1389 application_controller

```

```

703 file_io_servers file_io_servers set 1 347 file_server_2
704 global_locks global_locks set 0 279 global_name_server
705 global_names global_names set 391 11371 global_name_server
706 global_names_ext global_names_ext set 0 279 global_name_server
707 global_pid_ids global_pid_ids bag 0 279 global_name_server
708 global_pid_names global_pid_names bag 782 11203 global_name_server
709 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
710 hm_table_global hm_table_global bag 413 40753 <0.345.0>
711 inet_cache inet_cache bag 0 279 inet_db
712 inet_db inet_db set 21 541 inet_db
713 inet_hosts inet_hosts set 0 279 inet_db
714 sys_dist sys_dist set 64 3047 net_kernel
715 ===== netlab48@netlab4 =====
716 id name type size mem owner
717 -----
718 10 cookies set 0 279 auth
719 13 code set 279 12706 code_server
720 14 code_names set 48 4649 code_server
721 15 ign_requests set 0 279 inet_gethost_native
722 16 ign_req_index set 0 279 inet_gethost_native
723 ac_tab ac_tab set 20 1389 application_controller
724 file_io_servers file_io_servers set 1 347 file_server_2
725 global_locks global_locks set 0 279 global_name_server
726 global_names global_names set 391 11287 global_name_server
727 global_names_ext global_names_ext set 0 279 global_name_server
728 global_pid_ids global_pid_ids bag 0 279 global_name_server
729 global_pid_names global_pid_names bag 782 11203 global_name_server
730 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
731 hm_table_global hm_table_global bag 676 66527 <0.366.0>
732 inet_cache inet_cache bag 0 279 inet_db
733 inet_db inet_db set 21 541 inet_db
734 inet_hosts inet_hosts set 0 279 inet_db
735 sys_dist sys_dist set 64 3047 net_kernel
736 ===== netlab49@netlab4 =====
737 id name type size mem owner
738 -----
739 10 cookies set 0 279 auth
740 13 code set 279 12706 code_server
741 14 code_names set 48 4649 code_server
742 15 ign_requests set 0 279 inet_gethost_native
743 16 ign_req_index set 0 279 inet_gethost_native
744 ac_tab ac_tab set 20 1389 application_controller
745 file_io_servers file_io_servers set 1 347 file_server_2
746 global_locks global_locks set 0 279 global_name_server
747 global_names global_names set 391 11343 global_name_server
748 global_names_ext global_names_ext set 0 279 global_name_server
749 global_pid_ids global_pid_ids bag 0 279 global_name_server
750 global_pid_names global_pid_names bag 782 11203 global_name_server
751 hm_ets_cache_table hm_ets_cache_table set 0 279 hm_cache_mgr
752 hm_table_global hm_table_global bag 945 92889 <0.328.0>
753 inet_cache inet_cache bag 0 279 inet_db
754 inet_db inet_db set 21 541 inet_db
755 inet_hosts inet_hosts set 0 279 inet_db
756 sys_dist sys_dist set 64 3047 net_kernel
757 ===== netlab410@netlab4 =====
758 id name type size mem owner
759 -----
760 10 cookies set 0 279 auth
761 13 code set 279 12706 code_server
762 14 code_names set 48 4649 code_server

```

```

763 15          ign_requests      set 0      279      inet_gethost_native
764 16          ign_req_index      set 0      279      inet_gethost_native
765 ac_tab      ac_tab            set 20     1389     application_controller
766 file_io_servers file_io_servers set 1      349      file_server_2
767 global_locks global_locks      set 0      279      global_name_server
768 global_names global_names      set 391    11287    global_name_server
769 global_names_ext global_names_ext set 0      279      global_name_server
770 global_pid_ids global_pid_ids    bag 0      279      global_name_server
771 global_pid_names global_pid_names bag 782    11203    global_name_server
772 hm_ets_cache_table hm_ets_cache_table set 0      279      hm_cache_mgr
773 hm_table_global hm_table_global bag 907    89165    <0.369.0>
774 inet_cache   inet_cache      bag 0      279      inet_db
775 inet_db      inet_db        set 21     541      inet_db
776 inet_hosts   inet_hosts      set 0      279      inet_db
777 sys_dist     sys_dist        set 64     3047     net_kernel
778 ===== netlab411@netlab4 =====
779 id           name                type  size  mem  owner
780 -----
781 10           cookies             set 0      279      auth
782 13           code               set 279    12706    code_server
783 14           code_names         set 48     4649     code_server
784 15           ign_requests       set 0      279      inet_gethost_native
785 16           ign_req_index      set 0      279      inet_gethost_native
786 ac_tab      ac_tab            set 20     1389     application_controller
787 file_io_servers file_io_servers set 1      349      file_server_2
788 global_locks global_locks      set 0      279      global_name_server
789 global_names global_names      set 391    11271    global_name_server
790 global_names_ext global_names_ext set 0      279      global_name_server
791 global_pid_ids global_pid_ids    bag 0      279      global_name_server
792 global_pid_names global_pid_names bag 782    11203    global_name_server
793 hm_ets_cache_table hm_ets_cache_table set 0      279      hm_cache_mgr
794 hm_table_global hm_table_global bag 861    84657    <0.362.0>
795 inet_cache   inet_cache      bag 0      279      inet_db
796 inet_db      inet_db        set 21     541      inet_db
797 inet_hosts   inet_hosts      set 0      279      inet_db
798 sys_dist     sys_dist        set 64     3047     net_kernel
799 ===== netlab412@netlab4 =====
800 id           name                type  size  mem  owner
801 -----
802 10           cookies             set 0      279      auth
803 13           code               set 279    12706    code_server
804 14           code_names         set 48     4649     code_server
805 15           ign_requests       set 0      279      inet_gethost_native
806 16           ign_req_index      set 0      279      inet_gethost_native
807 ac_tab      ac_tab            set 20     1389     application_controller
808 file_io_servers file_io_servers set 1      349      file_server_2
809 global_locks global_locks      set 0      279      global_name_server
810 global_names global_names      set 391    11287    global_name_server
811 global_names_ext global_names_ext set 0      279      global_name_server
812 global_pid_ids global_pid_ids    bag 0      279      global_name_server
813 global_pid_names global_pid_names bag 782    11203    global_name_server
814 hm_ets_cache_table hm_ets_cache_table set 0      279      hm_cache_mgr
815 hm_table_global hm_table_global bag 932    91615    <0.371.0>
816 inet_cache   inet_cache      bag 0      279      inet_db
817 inet_db      inet_db        set 21     541      inet_db
818 inet_hosts   inet_hosts      set 0      279      inet_db
819 sys_dist     sys_dist        set 64     3047     net_kernel
820 ===== netlab413@netlab4 =====
821 id           name                type  size  mem  owner
822 -----

```

```

823 10          cookies          set 0      279      auth
824 13          code             set 279    12706   code_server
825 14          code_names       set 48    4649   code_server
826 15          ign_requests      set 0     279    inet_gethost_native
827 16          ign_req_index     set 0     279    inet_gethost_native
828 ac_tab      ac_tab           set 20    1389   application_controller
829 file_io_servers file_io_servers set 1     349    file_server_2
830 global_locks global_locks    set 0     279    global_name_server
831 global_names global_names    set 391   11287   global_name_server
832 global_names_ext global_names_ext set 0     279    global_name_server
833 global_pid_ids global_pid_ids bag 0     279    global_name_server
834 global_pid_names global_pid_names bag 782   11203   global_name_server
835 hm_ets_cache_table hm_ets_cache_table set 0     279    hm_cache_mgr
836 hm_table_global hm_table_global bag 439   43301   <0.377.0>
837 inet_cache     inet_cache     bag 0     279    inet_db
838 inet_db        inet_db        set 21    541    inet_db
839 inet_hosts     inet_hosts     set 0     279    inet_db
840 sys_dist       sys_dist       set 64    3047   net_kernel
841 ===== netlab414@netlab4 =====
842 id            name            type  size    mem      owner
843 -----
844 10            cookies          set 0     279      auth
845 13            code             set 279    12706   code_server
846 14            code_names       set 48    4649   code_server
847 15            ign_requests      set 0     279    inet_gethost_native
848 16            ign_req_index     set 0     279    inet_gethost_native
849 ac_tab        ac_tab           set 20    1389   application_controller
850 file_io_servers file_io_servers set 1     349    file_server_2
851 global_locks global_locks    set 0     279    global_name_server
852 global_names global_names    set 391   11271   global_name_server
853 global_names_ext global_names_ext set 0     279    global_name_server
854 global_pid_ids global_pid_ids bag 0     279    global_name_server
855 global_pid_names global_pid_names bag 782   11203   global_name_server
856 hm_ets_cache_table hm_ets_cache_table set 0     279    hm_cache_mgr
857 hm_table_global hm_table_global bag 1145   112489   <0.375.0>
858 inet_cache     inet_cache     bag 0     279    inet_db
859 inet_db        inet_db        set 21    541    inet_db
860 inet_hosts     inet_hosts     set 0     279    inet_db
861 sys_dist       sys_dist       set 64    3047   net_kernel
862 ===== netlab415@netlab4 =====
863 id            name            type  size    mem      owner
864 -----
865 10            cookies          set 0     279      auth
866 13            code             set 279    12706   code_server
867 14            code_names       set 48    4649   code_server
868 15            ign_requests      set 0     279    inet_gethost_native
869 16            ign_req_index     set 0     279    inet_gethost_native
870 ac_tab        ac_tab           set 20    1389   application_controller
871 file_io_servers file_io_servers set 1     349    file_server_2
872 global_locks global_locks    set 0     279    global_name_server
873 global_names global_names    set 391   11287   global_name_server
874 global_names_ext global_names_ext set 0     279    global_name_server
875 global_pid_ids global_pid_ids bag 0     279    global_name_server
876 global_pid_names global_pid_names bag 782   11203   global_name_server
877 hm_ets_cache_table hm_ets_cache_table set 0     279    hm_cache_mgr
878 hm_table_global hm_table_global bag 651   64077   <0.455.0>
879 inet_cache     inet_cache     bag 0     279    inet_db
880 inet_db        inet_db        set 21    541    inet_db
881 inet_hosts     inet_hosts     set 0     279    inet_db
882 sys_dist       sys_dist       set 64    3047   net_kernel

```

```

883 ===== netlab416@netlab4 =====
884 id          name          type  size  mem  owner
885 -----
886 10          cookies        set    0    279  auth
887 13          code           set   279  12706 code_server
888 14          code_names       set    48  4649  code_server
889 15          ign_requests   set    0    279  inet_gethost_native
890 16          ign_req_index  set    0    279  inet_gethost_native
891 ac_tab      ac_tab             set   20  1389  application_controller
892 file_io_servers file_io_servers set    1    349  file_server_2
893 global_locks global_locks      set    0    279  global_name_server
894 global_names global_names      set   391  11287 global_name_server
895 global_names_ext global_names_ext set    0    279  global_name_server
896 global_pid_ids global_pid_ids    bag    0    279  global_name_server
897 global_pid_names global_pid_names bag   782  11203 global_name_server
898 hm_ets_cache_table hm_ets_cache_table set    0    279  hm_cache_mgr
899 hm_table_global hm_table_global  bag   817  80345 <0.378.0>
900 inet_cache    inet_cache        bag    0    279  inet_db
901 inet_db       inet_db           set   21   541  inet_db
902 inet_hosts    inet_hosts        set    0    279  inet_db
903 sys_dist     sys_dist         set    64  3047  net_kernel
904 ===== netlab417@netlab4 =====
905 id          name          type  size  mem  owner
906 -----
907 10          cookies        set    0    279  auth
908 13          code           set   279  12706 code_server
909 14          code_names       set    48  4649  code_server
910 15          ign_requests   set    0    279  inet_gethost_native
911 16          ign_req_index  set    0    279  inet_gethost_native
912 ac_tab      ac_tab             set   20  1389  application_controller
913 file_io_servers file_io_servers set    1    349  file_server_2
914 global_locks global_locks      set    0    279  global_name_server
915 global_names global_names      set   391  11287 global_name_server
916 global_names_ext global_names_ext set    0    279  global_name_server
917 global_pid_ids global_pid_ids    bag    0    279  global_name_server
918 global_pid_names global_pid_names bag   782  11203 global_name_server
919 hm_ets_cache_table hm_ets_cache_table set    0    279  hm_cache_mgr
920 hm_table_global hm_table_global  bag  1526  149827 <0.366.0>
921 inet_cache    inet_cache        bag    0    279  inet_db
922 inet_db       inet_db           set   21   541  inet_db
923 inet_hosts    inet_hosts        set    0    279  inet_db
924 sys_dist     sys_dist         set    64  3047  net_kernel
925 ===== netlab418@netlab4 =====
926 id          name          type  size  mem  owner
927 -----
928 10          cookies        set    0    279  auth
929 13          code           set   279  12706 code_server
930 14          code_names       set    48  4649  code_server
931 15          ign_requests   set    0    279  inet_gethost_native
932 16          ign_req_index  set    0    279  inet_gethost_native
933 ac_tab      ac_tab             set   20  1389  application_controller
934 file_io_servers file_io_servers set    1    349  file_server_2
935 global_locks global_locks      set    0    279  global_name_server
936 global_names global_names      set   391  11271 global_name_server
937 global_names_ext global_names_ext set    0    279  global_name_server
938 global_pid_ids global_pid_ids    bag    0    279  global_name_server
939 global_pid_names global_pid_names bag   782  11203 global_name_server
940 hm_ets_cache_table hm_ets_cache_table set    0    279  hm_cache_mgr
941 hm_table_global hm_table_global  bag   378  37323 <0.377.0>
942 inet_cache    inet_cache        bag    0    279  inet_db

```

```

943 inet_db          inet_db          set 21    541    inet_db
944 inet_hosts       inet_hosts       set 0     279    inet_db
945 sys_dist         sys_dist         set 64   3047   net_kernel
946 ===== netlab419@netlab4 =====
947 id              name              type  size    mem    owner
948 -----
949 10              cookies           set 0      279    auth
950 13              code              set 279    12706  code_server
951 14              code_names        set 48     4649   code_server
952 15              ign_requests      set 0      279    inet_gethost_native
953 16              ign_req_index     set 0      279    inet_gethost_native
954 ac_tab          ac_tab           set 20    1389   application_controller
955 file_io_servers file_io_servers  set 1     349    file_server_2
956 global_locks    global_locks     set 0      279    global_name_server
957 global_names    global_names     set 391   11271  global_name_server
958 global_names_ext global_names_ext set 0      279    global_name_server
959 global_pid_ids  global_pid_ids   bag 0      279    global_name_server
960 global_pid_names global_pid_names bag 782   11203  global_name_server
961 hm_ets_cache_table hm_ets_cache_table set 0      279    hm_cache_mgr
962 hm_table_global hm_table_global  bag 1198  117683 <0.388.0>
963 inet_cache      inet_cache       bag 0      279    inet_db
964 inet_db         inet_db          set 21    541    inet_db
965 inet_hosts      inet_hosts       set 0      279    inet_db
966 sys_dist        sys_dist         set 64   3047   net_kernel
967 ===== netlab420@netlab4 =====
968 id              name              type  size    mem    owner
969 -----
970 10              cookies           set 0      279    auth
971 13              code              set 279    12706  code_server
972 14              code_names        set 48     4649   code_server
973 15              ign_requests      set 0      279    inet_gethost_native
974 16              ign_req_index     set 0      279    inet_gethost_native
975 ac_tab          ac_tab           set 20    1389   application_controller
976 file_io_servers file_io_servers  set 1     349    file_server_2
977 global_locks    global_locks     set 0      279    global_name_server
978 global_names    global_names     set 391   11243  global_name_server
979 global_names_ext global_names_ext set 0      279    global_name_server
980 global_pid_ids  global_pid_ids   bag 0      279    global_name_server
981 global_pid_names global_pid_names bag 782   11203  global_name_server
982 hm_ets_cache_table hm_ets_cache_table set 0      279    hm_cache_mgr
983 hm_table_global hm_table_global  bag 751   73877  <0.406.0>
984 inet_cache      inet_cache       bag 0      279    inet_db
985 inet_db         inet_db          set 21    541    inet_db
986 inet_hosts      inet_hosts       set 0      279    inet_db
987 sys_dist        sys_dist         set 64   3047   net_kernel
988 ===== dell1@dell =====
989 id              name              type  size    mem    owner
990 -----
991 12              cookies           set 0      286    auth
992 4111            code              set 281   12978  code_server
993 8208            code_names        set 56    7510   code_server
994 12305           ign_requests      set 0      286    inet_gethost_native
995 16402           ign_req_index     set 0      286    inet_gethost_native
996 ac_tab          ac_tab           set 20    1328   application_controller
997 file_io_servers file_io_servers  set 1     339    file_server_2
998 global_locks    global_locks     set 0      286    global_name_server
999 global_names    global_names     set 391   10505  global_name_server
1000 global_names_ext global_names_ext set 0      286    global_name_server
1001 global_pid_ids  global_pid_ids   bag 0      286    global_name_server
1002 global_pid_names global_pid_names bag 782   8864   global_name_server

```



```

1003 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1004 hm_table_global hm_table_global bag 315 30211 <0.359.0>
1005 inet_cache inet_cache bag 0 286 inet_db
1006 inet_db inet_db set 29 553 inet_db
1007 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1008 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1009 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1010 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1011 sys_dist sys_dist set 64 2868 net_kernel
1012 ===== dell2@dell =====
1013 id name type size mem owner
1014 -----
1015 12 cookies set 0 286 auth
1016 4111 code set 281 12978 code_server
1017 8208 code_names set 56 7510 code_server
1018 12305 ign_requests set 0 286 inet_gethost_native
1019 16402 ign_req_index set 0 286 inet_gethost_native
1020 ac_tab ac_tab set 20 1328 application_controller
1021 file_io_servers file_io_servers set 1 339 file_server_2
1022 global_locks global_locks set 0 286 global_name_server
1023 global_names global_names set 391 10461 global_name_server
1024 global_names_ext global_names_ext set 0 286 global_name_server
1025 global_pid_ids global_pid_ids bag 0 286 global_name_server
1026 global_pid_names global_pid_names bag 782 8864 global_name_server
1027 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1028 hm_table_global hm_table_global bag 483 46171 <0.382.0>
1029 inet_cache inet_cache bag 0 286 inet_db
1030 inet_db inet_db set 29 553 inet_db
1031 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1032 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1033 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1034 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1035 sys_dist sys_dist set 64 2868 net_kernel
1036 ===== dell3@dell =====
1037 id name type size mem owner
1038 -----
1039 12 cookies set 0 286 auth
1040 4111 code set 281 12978 code_server
1041 8208 code_names set 56 7510 code_server
1042 12305 ign_requests set 0 286 inet_gethost_native
1043 16402 ign_req_index set 0 286 inet_gethost_native
1044 ac_tab ac_tab set 20 1328 application_controller
1045 file_io_servers file_io_servers set 1 339 file_server_2
1046 global_locks global_locks set 0 286 global_name_server
1047 global_names global_names set 391 10461 global_name_server
1048 global_names_ext global_names_ext set 0 286 global_name_server
1049 global_pid_ids global_pid_ids bag 0 286 global_name_server
1050 global_pid_names global_pid_names bag 782 8864 global_name_server
1051 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1052 hm_table_global hm_table_global bag 459 43891 <0.327.0>
1053 inet_cache inet_cache bag 0 286 inet_db
1054 inet_db inet_db set 29 553 inet_db
1055 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1056 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1057 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1058 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1059 sys_dist sys_dist set 64 2868 net_kernel
1060 ===== dell4@dell =====
1061 id name type size mem owner
1062 -----

```

```

1063 12 cookies set 0 286 auth
1064 4111 code set 281 12978 code_server
1065 8208 code_names set 56 7510 code_server
1066 12305 ign_requests set 0 286 inet_gethost_native
1067 16402 ign_req_index set 0 286 inet_gethost_native
1068 ac_tab ac_tab set 20 1328 application_controller
1069 file_io_servers file_io_servers set 1 339 file_server_2
1070 global_locks global_locks set 0 286 global_name_server
1071 global_names global_names set 391 10453 global_name_server
1072 global_names_ext global_names_ext set 0 286 global_name_server
1073 global_pid_ids global_pid_ids bag 0 286 global_name_server
1074 global_pid_names global_pid_names bag 782 8864 global_name_server
1075 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1076 hm_table_global hm_table_global bag 1595 151811 <0.343.0>
1077 inet_cache inet_cache bag 0 286 inet_db
1078 inet_db inet_db set 29 553 inet_db
1079 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1080 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1081 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1082 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1083 sys_dist sys_dist set 64 2868 net_kernel
1084 ===== dell5@dell =====
1085 id name type size mem owner
1086 -----
1087 12 cookies set 0 286 auth
1088 4111 code set 281 12978 code_server
1089 8208 code_names set 56 7510 code_server
1090 12305 ign_requests set 0 286 inet_gethost_native
1091 16402 ign_req_index set 0 286 inet_gethost_native
1092 ac_tab ac_tab set 20 1328 application_controller
1093 file_io_servers file_io_servers set 1 339 file_server_2
1094 global_locks global_locks set 0 286 global_name_server
1095 global_names global_names set 391 10345 global_name_server
1096 global_names_ext global_names_ext set 0 286 global_name_server
1097 global_pid_ids global_pid_ids bag 0 286 global_name_server
1098 global_pid_names global_pid_names bag 782 8864 global_name_server
1099 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1100 hm_table_global hm_table_global bag 922 87876 <0.376.0>
1101 inet_cache inet_cache bag 0 286 inet_db
1102 inet_db inet_db set 29 553 inet_db
1103 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1104 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1105 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1106 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1107 sys_dist sys_dist set 64 2868 net_kernel
1108 ===== dell6@dell =====
1109 id name type size mem owner
1110 -----
1111 12 cookies set 0 286 auth
1112 4111 code set 281 12978 code_server
1113 8208 code_names set 56 7510 code_server
1114 12305 ign_requests set 0 286 inet_gethost_native
1115 16402 ign_req_index set 0 286 inet_gethost_native
1116 ac_tab ac_tab set 20 1328 application_controller
1117 file_io_servers file_io_servers set 1 339 file_server_2
1118 global_locks global_locks set 0 286 global_name_server
1119 global_names global_names set 391 10345 global_name_server
1120 global_names_ext global_names_ext set 0 286 global_name_server
1121 global_pid_ids global_pid_ids bag 0 286 global_name_server
1122 global_pid_names global_pid_names bag 782 8864 global_name_server

```

```

1123 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1124 hm_table_global hm_table_global bag 517 49401 <0.397.0>
1125 inet_cache inet_cache bag 0 286 inet_db
1126 inet_db inet_db set 29 553 inet_db
1127 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1128 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1129 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1130 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1131 sys_dist sys_dist set 64 2868 net_kernel
1132 ===== dell7@dell =====
1133 id name type size mem owner
1134 -----
1135 12 cookies set 0 286 auth
1136 4111 code set 281 12978 code_server
1137 8208 code_names set 56 7510 code_server
1138 12305 ign_requests set 0 286 inet_gethost_native
1139 16402 ign_req_index set 0 286 inet_gethost_native
1140 ac_tab ac_tab set 20 1328 application_controller
1141 file_io_servers file_io_servers set 1 339 file_server_2
1142 global_locks global_locks set 0 286 global_name_server
1143 global_names global_names set 391 10341 global_name_server
1144 global_names_ext global_names_ext set 0 286 global_name_server
1145 global_pid_ids global_pid_ids bag 0 286 global_name_server
1146 global_pid_names global_pid_names bag 782 8864 global_name_server
1147 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1148 hm_table_global hm_table_global bag 384 36766 <0.407.0>
1149 inet_cache inet_cache bag 0 286 inet_db
1150 inet_db inet_db set 29 553 inet_db
1151 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1152 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1153 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1154 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1155 sys_dist sys_dist set 64 2868 net_kernel
1156 ===== dell8@dell =====
1157 id name type size mem owner
1158 -----
1159 12 cookies set 0 286 auth
1160 4111 code set 281 12978 code_server
1161 8208 code_names set 56 7510 code_server
1162 12305 ign_requests set 0 286 inet_gethost_native
1163 16402 ign_req_index set 0 286 inet_gethost_native
1164 ac_tab ac_tab set 20 1328 application_controller
1165 file_io_servers file_io_servers set 1 339 file_server_2
1166 global_locks global_locks set 0 286 global_name_server
1167 global_names global_names set 391 10521 global_name_server
1168 global_names_ext global_names_ext set 0 286 global_name_server
1169 global_pid_ids global_pid_ids bag 0 286 global_name_server
1170 global_pid_names global_pid_names bag 782 8864 global_name_server
1171 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1172 hm_table_global hm_table_global bag 215 20711 <0.309.0>
1173 inet_cache inet_cache bag 0 286 inet_db
1174 inet_db inet_db set 29 553 inet_db
1175 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1176 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1177 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1178 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1179 sys_dist sys_dist set 64 2868 net_kernel
1180 ===== netlabal@netlaba =====
1181 id name type size mem owner
1182 -----

```

```

1183 9          cookies          set 0      284      auth
1184 4108        code             set 278    12787   code_server
1185 8205        code_names       set 54     5196   code_server
1186 12302       ign_requests      set 0      284    inet_gethost_native
1187 16399       ign_req_index     set 0      284    inet_gethost_native
1188 ac_tab      ac_tab              set 20     1380   application_controller
1189 file_io_servers file_io_servers set 1      352    file_server_2
1190 global_locks global_locks    set 0      284    global_name_server
1191 global_names global_names    set 391    10916  global_name_server
1192 global_names_ext global_names_ext set 0      284    global_name_server
1193 global_pid_ids global_pid_ids bag 0      284    global_name_server
1194 global_pid_names global_pid_names bag 782    11208  global_name_server
1195 hm_ets_cache_table hm_ets_cache_table set 0      284    hm_cache_mgr
1196 hm_table_global hm_table_global bag 491    48402  <0.520.0>
1197 inet_cache     inet_cache     bag 0      284    inet_db
1198 inet_db        inet_db        set 21     512    inet_db
1199 inet_hosts     inet_hosts     set 0      284    inet_db
1200 sys_dist       sys_dist       set 64     3052   net_kernel
1201 ===== netlaba2@netlaba =====
1202 id            name              type  size    mem    owner
1203 -----
1204 9             cookies          set 0      284    auth
1205 4108          code             set 278    12787  code_server
1206 8205          code_names       set 54     5196   code_server
1207 12302         ign_requests      set 0      284    inet_gethost_native
1208 16399         ign_req_index     set 0      284    inet_gethost_native
1209 ac_tab        ac_tab              set 20     1380   application_controller
1210 file_io_servers file_io_servers set 1      352    file_server_2
1211 global_locks  global_locks    set 0      284    global_name_server
1212 global_names  global_names    set 391    10912  global_name_server
1213 global_names_ext global_names_ext set 0      284    global_name_server
1214 global_pid_ids global_pid_ids bag 0      284    global_name_server
1215 global_pid_names global_pid_names bag 782    11208  global_name_server
1216 hm_ets_cache_table hm_ets_cache_table set 0      284    hm_cache_mgr
1217 hm_table_global hm_table_global bag 1558    152968 <0.522.0>
1218 inet_cache     inet_cache     bag 0      284    inet_db
1219 inet_db        inet_db        set 21     512    inet_db
1220 inet_hosts     inet_hosts     set 0      284    inet_db
1221 sys_dist       sys_dist       set 64     3052   net_kernel
1222 ===== netlaba3@netlaba =====
1223 id            name              type  size    mem    owner
1224 -----
1225 9             cookies          set 0      284    auth
1226 4108          code             set 278    12787  code_server
1227 8205          code_names       set 54     5196   code_server
1228 12302         ign_requests      set 0      284    inet_gethost_native
1229 16399         ign_req_index     set 0      284    inet_gethost_native
1230 ac_tab        ac_tab              set 20     1380   application_controller
1231 file_io_servers file_io_servers set 1      352    file_server_2
1232 global_locks  global_locks    set 0      284    global_name_server
1233 global_names  global_names    set 391    10916  global_name_server
1234 global_names_ext global_names_ext set 0      284    global_name_server
1235 global_pid_ids global_pid_ids bag 0      284    global_name_server
1236 global_pid_names global_pid_names bag 782    11208  global_name_server
1237 hm_ets_cache_table hm_ets_cache_table set 0      284    hm_cache_mgr
1238 hm_table_global hm_table_global bag 606    59672  <0.521.0>
1239 inet_cache     inet_cache     bag 0      284    inet_db
1240 inet_db        inet_db        set 21     512    inet_db
1241 inet_hosts     inet_hosts     set 0      284    inet_db
1242 sys_dist       sys_dist       set 64     3052   net_kernel

```

1243 ===== netlaba4@netlaba =====

id	name	type	size	mem	owner
9	cookies	set	0	284	auth
4108	code	set	278	12787	code_server
8205	code_names	set	54	5196	code_server
12302	ign_requests	set	0	284	inet_gethost_native
16399	ign_req_index	set	0	284	inet_gethost_native
'Domain1Tbl2'	'Domain1Tbl2'	bag	10000	361821	<0.543.0>
ac_tab	ac_tab	set	20	1380	application_controller
file_io_servers	file_io_servers	set	1	352	file_server_2
global_locks	global_locks	set	0	284	global_name_server
global_names	global_names	set	391	10840	global_name_server
global_names_ext	global_names_ext	set	0	284	global_name_server
global_pid_ids	global_pid_ids	bag	0	284	global_name_server
global_pid_names	global_pid_names	bag	782	11208	global_name_server
hm_ets_cache_table	hm_ets_cache_table	set	0	284	hm_cache_mgr
hm_table_global	hm_table_global	bag	667	65650	<0.542.0>
inet_cache	inet_cache	bag	0	284	inet_db
inet_db	inet_db	set	21	512	inet_db
inet_hosts	inet_hosts	set	0	284	inet_db
sys_dist	sys_dist	set	64	3052	net_kernel

1265 ===== netlaba5@netlaba =====

id	name	type	size	mem	owner
9	cookies	set	0	284	auth
4108	code	set	278	12787	code_server
8205	code_names	set	54	5196	code_server
12302	ign_requests	set	0	284	inet_gethost_native
16399	ign_req_index	set	0	284	inet_gethost_native
ac_tab	ac_tab	set	20	1380	application_controller
file_io_servers	file_io_servers	set	1	352	file_server_2
global_locks	global_locks	set	0	284	global_name_server
global_names	global_names	set	391	10912	global_name_server
global_names_ext	global_names_ext	set	0	284	global_name_server
global_pid_ids	global_pid_ids	bag	0	284	global_name_server
global_pid_names	global_pid_names	bag	782	11208	global_name_server
hm_ets_cache_table	hm_ets_cache_table	set	0	284	hm_cache_mgr
hm_table_global	hm_table_global	bag	833	81918	<0.516.0>
inet_cache	inet_cache	bag	0	284	inet_db
inet_db	inet_db	set	21	512	inet_db
inet_hosts	inet_hosts	set	0	284	inet_db
sys_dist	sys_dist	set	64	3052	net_kernel

1286 ===== netlaba6@netlaba =====

id	name	type	size	mem	owner
9	cookies	set	0	284	auth
4108	code	set	278	12787	code_server
8205	code_names	set	54	5196	code_server
12302	ign_requests	set	0	284	inet_gethost_native
16399	ign_req_index	set	0	284	inet_gethost_native
ac_tab	ac_tab	set	20	1380	application_controller
file_io_servers	file_io_servers	set	1	352	file_server_2
global_locks	global_locks	set	0	284	global_name_server
global_names	global_names	set	391	10912	global_name_server
global_names_ext	global_names_ext	set	0	284	global_name_server
global_pid_ids	global_pid_ids	bag	0	284	global_name_server
global_pid_names	global_pid_names	bag	782	11208	global_name_server
hm_ets_cache_table	hm_ets_cache_table	set	0	284	hm_cache_mgr
hm_table_global	hm_table_global	bag	829	81526	<0.554.0>

```

1303 inet_cache      inet_cache      bag    0      284      inet_db
1304 inet_db          inet_db          set    21     512     inet_db
1305 inet_hosts       inet_hosts       set    0      284     inet_db
1306 sys_dist         sys_dist         set    64    3052    net_kernel
1307 ===== netlaba7@netlaba =====
1308 id              name              type    size    mem      owner
1309 -----
1310 9                cookies           set     0       284      auth
1311 4108             code              set    278    12787    code_server
1312 8205             code_names        set     54    5196     code_server
1313 12302            ign_requests      set     0       284      inet_gethost_native
1314 16399            ign_req_index     set     0       284      inet_gethost_native
1315 ac_tab           ac_tab            set    20    1380     application_controller
1316 file_io_servers  file_io_servers   set     1       352      file_server_2
1317 global_locks     global_locks      set     0       284      global_name_server
1318 global_names     global_names      set    391   10912    global_name_server
1319 global_names_ext global_names_ext  set     0       284      global_name_server
1320 global_pid_ids   global_pid_ids    bag     0       284      global_name_server
1321 global_pid_names global_pid_names  bag    782   11208    global_name_server
1322 hm_ets_cache_table hm_ets_cache_table set     0       284      hm_cache_mgr
1323 hm_table_global  hm_table_global   bag    638   62808    <0.511.0>
1324 inet_cache      inet_cache      bag     0       284      inet_db
1325 inet_db          inet_db          set    21     512     inet_db
1326 inet_hosts       inet_hosts       set     0       284      inet_db
1327 sys_dist         sys_dist         set    64    3052    net_kernel
1328 ===== netlaba8@netlaba =====
1329 id              name              type    size    mem      owner
1330 -----
1331 9                cookies           set     0       284      auth
1332 4108             code              set    278    12787    code_server
1333 8205             code_names        set     54    5196     code_server
1334 12302            ign_requests      set     0       284      inet_gethost_native
1335 16399            ign_req_index     set     0       284      inet_gethost_native
1336 ac_tab           ac_tab            set    20    1380     application_controller
1337 file_io_servers  file_io_servers   set     1       352      file_server_2
1338 global_locks     global_locks      set     0       284      global_name_server
1339 global_names     global_names      set    391   10916    global_name_server
1340 global_names_ext global_names_ext  set     0       284      global_name_server
1341 global_pid_ids   global_pid_ids    bag     0       284      global_name_server
1342 global_pid_names global_pid_names  bag    782   11208    global_name_server
1343 hm_ets_cache_table hm_ets_cache_table set     0       284      hm_cache_mgr
1344 hm_table_global  hm_table_global   bag    496   48892    <0.533.0>
1345 inet_cache      inet_cache      bag     0       284      inet_db
1346 inet_db          inet_db          set    21     512     inet_db
1347 inet_hosts       inet_hosts       set     0       284      inet_db
1348 sys_dist         sys_dist         set    64    3052    net_kernel
1349 ===== netlabbl@netlabl =====
1350 id              name              type    size    mem      owner
1351 -----
1352 12               cookies           set     0       286      auth
1353 4111             code              set    281   12978    code_server
1354 8208             code_names        set     56    7510     code_server
1355 12305            ign_requests      set     0       286      inet_gethost_native
1356 16402            ign_req_index     set     0       286      inet_gethost_native
1357 ac_tab           ac_tab            set    21   1339     application_controller
1358 file_io_servers  file_io_servers   set     1       351      file_server_2
1359 global_locks     global_locks      set     0       286      global_name_server
1360 global_names     global_names      set    391   9997     global_name_server
1361 global_names_ext global_names_ext  set     0       286      global_name_server
1362 global_pid_ids   global_pid_ids    bag     0       286      global_name_server

```

```

1363 global_pid_names global_pid_names bag 782 8864 global_name_server
1364 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1365 hm_table_global hm_table_global bag 1208 115046 <0.468.0>
1366 inet_cache inet_cache bag 0 286 inet_db
1367 inet_db inet_db set 29 559 inet_db
1368 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1369 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1370 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1371 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1372 sys_dist sys_dist set 64 2862 net_kernel
1373 ===== netlab2@netlab2 =====
1374 id name type size mem owner
1375 -----
1376 12 cookies set 0 286 auth
1377 4111 code set 281 12978 code_server
1378 8208 code_names set 56 7510 code_server
1379 12305 ign_requests set 0 286 inet_gethost_native
1380 16402 ign_req_index set 0 286 inet_gethost_native
1381 ac_tab ac_tab set 21 1339 application_controller
1382 file_io_servers file_io_servers set 1 351 file_server_2
1383 global_locks global_locks set 0 286 global_name_server
1384 global_names global_names set 391 9665 global_name_server
1385 global_names_ext global_names_ext set 0 286 global_name_server
1386 global_pid_ids global_pid_ids bag 0 286 global_name_server
1387 global_pid_names global_pid_names bag 782 8864 global_name_server
1388 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1389 hm_table_global hm_table_global bag 542 51776 <0.595.0>
1390 inet_cache inet_cache bag 0 286 inet_db
1391 inet_db inet_db set 29 559 inet_db
1392 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1393 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1394 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1395 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1396 sys_dist sys_dist set 64 2862 net_kernel
1397 ===== netlab3@netlab3 =====
1398 id name type size mem owner
1399 -----
1400 12 cookies set 0 286 auth
1401 4111 code set 281 12978 code_server
1402 8208 code_names set 56 7510 code_server
1403 12305 ign_requests set 0 286 inet_gethost_native
1404 16402 ign_req_index set 0 286 inet_gethost_native
1405 ac_tab ac_tab set 21 1339 application_controller
1406 file_io_servers file_io_servers set 1 351 file_server_2
1407 global_locks global_locks set 0 286 global_name_server
1408 global_names global_names set 391 9805 global_name_server
1409 global_names_ext global_names_ext set 0 286 global_name_server
1410 global_pid_ids global_pid_ids bag 0 286 global_name_server
1411 global_pid_names global_pid_names bag 782 8864 global_name_server
1412 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1413 hm_table_global hm_table_global bag 783 74671 <0.539.0>
1414 inet_cache inet_cache bag 0 286 inet_db
1415 inet_db inet_db set 29 559 inet_db
1416 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1417 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1418 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1419 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1420 sys_dist sys_dist set 64 2862 net_kernel
1421 ===== netlab4@netlab4 =====
1422 id name type size mem owner

```

```

1423 -----
1424 12          cookies          set 0      286      auth
1425 4111        code            set 281    12978   code_server
1426 8208        code_names      set 56    7510   code_server
1427 12305       ign_requests     set 0      286    inet_gethost_native
1428 16402       ign_req_index    set 0      286    inet_gethost_native
1429 ac_tab      ac_tab          set 21    1339   application_controller
1430 file_io_servers file_io_servers set 1      351    file_server_2
1431 global_locks global_locks    set 0      286    global_name_server
1432 global_names global_names    set 391   10313  global_name_server
1433 global_names_ext global_names_ext set 0      286    global_name_server
1434 global_pid_ids global_pid_ids  bag 0      286    global_name_server
1435 global_pid_names global_pid_names bag 782   8864   global_name_server
1436 hm_ets_cache_table hm_ets_cache_table set 0      286    hm_cache_mgr
1437 hm_table_global hm_table_global bag 1058  100796 <0.344.0>
1438 inet_cache   inet_cache     bag 0      286    inet_db
1439 inet_db      inet_db        set 29    559    inet_db
1440 inet_hosts_byaddr inet_hosts_byaddr bag 0      286    inet_db
1441 inet_hosts_byname inet_hosts_byname bag 0      286    inet_db
1442 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0      286    inet_db
1443 inet_hosts_file_byname inet_hosts_file_byname bag 0      286    inet_db
1444 sys_dist     sys_dist      set 64    2862   net_kernel
1445 ===== netlabbb5@netlabbb =====
1446 id          name          type  size  mem  owner
1447 -----
1448 12          cookies          set 0      286    auth
1449 4111        code            set 281    12978   code_server
1450 8208        code_names      set 56    7510   code_server
1451 12305       ign_requests     set 0      286    inet_gethost_native
1452 16402       ign_req_index    set 0      286    inet_gethost_native
1453 ac_tab      ac_tab          set 21    1339   application_controller
1454 file_io_servers file_io_servers set 1      351    file_server_2
1455 global_locks global_locks    set 0      286    global_name_server
1456 global_names global_names    set 391   9661   global_name_server
1457 global_names_ext global_names_ext set 0      286    global_name_server
1458 global_pid_ids global_pid_ids  bag 0      286    global_name_server
1459 global_pid_names global_pid_names bag 782   8864   global_name_server
1460 hm_ets_cache_table hm_ets_cache_table set 0      286    hm_cache_mgr
1461 hm_table_global hm_table_global bag 223   21471  <0.581.0>
1462 inet_cache   inet_cache     bag 0      286    inet_db
1463 inet_db      inet_db        set 29    559    inet_db
1464 inet_hosts_byaddr inet_hosts_byaddr bag 0      286    inet_db
1465 inet_hosts_byname inet_hosts_byname bag 0      286    inet_db
1466 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0      286    inet_db
1467 inet_hosts_file_byname inet_hosts_file_byname bag 0      286    inet_db
1468 sys_dist     sys_dist      set 64    2862   net_kernel
1469 ===== netlabbb6@netlabbb =====
1470 id          name          type  size  mem  owner
1471 -----
1472 12          cookies          set 0      286    auth
1473 4111        code            set 281    12978   code_server
1474 8208        code_names      set 56    7510   code_server
1475 12305       ign_requests     set 0      286    inet_gethost_native
1476 16402       ign_req_index    set 0      286    inet_gethost_native
1477 ac_tab      ac_tab          set 21    1339   application_controller
1478 file_io_servers file_io_servers set 1      351    file_server_2
1479 global_locks global_locks    set 0      286    global_name_server
1480 global_names global_names    set 391   9645   global_name_server
1481 global_names_ext global_names_ext set 0      286    global_name_server
1482 global_pid_ids global_pid_ids  bag 0      286    global_name_server

```



```

1483 global_pid_names global_pid_names bag 782 8864 global_name_server
1484 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1485 hm_table_global hm_table_global bag 981 93481 <0.603.0>
1486 inet_cache inet_cache bag 0 286 inet_db
1487 inet_db inet_db set 29 559 inet_db
1488 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1489 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1490 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1491 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1492 sys_dist sys_dist set 64 2862 net_kernel
1493 ===== netlab7@netlab7 =====
1494 id name type size mem owner
1495 -----
1496 12 cookies set 0 286 auth
1497 4111 code set 281 12978 code_server
1498 8208 code_names set 56 7510 code_server
1499 12305 ign_requests set 0 286 inet_gethost_native
1500 16402 ign_req_index set 0 286 inet_gethost_native
1501 ac_tab ac_tab set 21 1339 application_controller
1502 file_io_servers file_io_servers set 1 351 file_server_2
1503 global_locks global_locks set 0 286 global_name_server
1504 global_names global_names set 391 9893 global_name_server
1505 global_names_ext global_names_ext set 0 286 global_name_server
1506 global_pid_ids global_pid_ids bag 0 286 global_name_server
1507 global_pid_names global_pid_names bag 782 8864 global_name_server
1508 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1509 hm_table_global hm_table_global bag 867 82651 <0.501.0>
1510 inet_cache inet_cache bag 0 286 inet_db
1511 inet_db inet_db set 29 559 inet_db
1512 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1513 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1514 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1515 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1516 sys_dist sys_dist set 64 2862 net_kernel
1517 ===== netlab8@netlab8 =====
1518 id name type size mem owner
1519 -----
1520 12 cookies set 0 286 auth
1521 4111 code set 281 12978 code_server
1522 8208 code_names set 56 7510 code_server
1523 12305 ign_requests set 0 286 inet_gethost_native
1524 16402 ign_req_index set 0 286 inet_gethost_native
1525 'Domain1Tbl2' 'Domain1Tbl2' bag 10000 331824 <0.584.0>
1526 ac_tab ac_tab set 21 1339 application_controller
1527 file_io_servers file_io_servers set 1 351 file_server_2
1528 global_locks global_locks set 0 286 global_name_server
1529 global_names global_names set 391 9681 global_name_server
1530 global_names_ext global_names_ext set 0 286 global_name_server
1531 global_pid_ids global_pid_ids bag 0 286 global_name_server
1532 global_pid_names global_pid_names bag 782 8864 global_name_server
1533 hm_ets_cache_table hm_ets_cache_table set 0 286 hm_cache_mgr
1534 hm_table_global hm_table_global bag 908 86546 <0.583.0>
1535 inet_cache inet_cache bag 0 286 inet_db
1536 inet_db inet_db set 29 559 inet_db
1537 inet_hosts_byaddr inet_hosts_byaddr bag 0 286 inet_db
1538 inet_hosts_byname inet_hosts_byname bag 0 286 inet_db
1539 inet_hosts_file_byaddr inet_hosts_file_byaddr bag 0 286 inet_db
1540 inet_hosts_file_byname inet_hosts_file_byname bag 0 286 inet_db
1541 sys_dist sys_dist set 64 2862 net_kernel
1542 ok

```

1543 (xxx_node@netlab3)4>

G.5 Load Balancing

List 51: Test Results:Load balancing with 5 nodes per machine

```

1 hirotnk@netlab3 ~/harmonia/Harmonia/test $ cd ..
2 hirotnk@netlab3 ~/harmonia/Harmonia $ test/test5.sh -s
3 Launched the node:[netlab31@netlab3]
4 Launched the node:[netlab32@netlab3]
5 Launched the node:[netlab33@netlab3]
6 Launched the node:[netlab34@netlab3]
7 Launched the node:[netlab35@netlab3]
8 epmd: up and running on port 4369 with data:
9 name netlab32 at port 32921
10 name netlab31 at port 38087
11 hirotnk@netlab3 ~/harmonia/Harmonia $ erl -pa ebin -config harmonia -run hm_edge start -\
    →harmonia node_type 'join' name xxx sname 'xxx_node@netlab3' root_node 'netlab31@netlab3' \
    →setcookie harmonia_cookie -sname 'xxx_node@netlab3'
12 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
13
14 Eshell V5.6.3 (abort with ^G)
15 (xxx_node@netlab3)1> [{name,xxx},
16 {root_node,netlab31@netlab3},
17 {root,netlab31},
18 {included_applications,[],},
19 {sname,xxx_node@netlab3},
20 {logfile_ext,".txt"},
21 {node_type,join},
22 {logdir,"log/"},
23 {logfile,"harmonia_log"}]
24 "log/harmonia_log_xxx.txt"
25 start Pid:[<0.94.0>]
26
27 (xxx_node@netlab3)1> hm_cli_test:store_short(10000).
28 ok
29 (xxx_node@netlab3)2> hm_cli_test:check_data_num().
30 Name:dell4 Node:dell4@dell count:3106
31 Name:dell2 Node:dell2@dell count:2394
32 Name:dell5 Node:dell5@dell count:2081
33 Name:dell3 Node:dell3@dell count:1634
34 Name:dell1 Node:dell1@dell count:1547
35 Name:netlabbb3 Node:netlabbb3@netlabbb count:1297
36 Name:netlabbb1 Node:netlabbb1@netlabbb count:2706
37 Name:netlabbb2 Node:netlabbb2@netlabbb count:2833
38 Name:netlabbb5 Node:netlabbb5@netlabbb count:1559
39 Name:netlabbb4 Node:netlabbb4@netlabbb count:1154
40 Name:netlaba3 Node:netlaba3@netlaba count:2048
41 Name:netlaba4 Node:netlaba4@netlaba count:1843
42 Name:netlaba5 Node:netlaba5@netlaba count:824
43 Name:netlaba1 Node:netlaba1@netlaba count:2053
44 Name:netlaba2 Node:netlaba2@netlaba count:3054
45 Name:netlab43 Node:netlab43@netlab4 count:1596
46 Name:netlab42 Node:netlab42@netlab4 count:1370
47 Name:netlab44 Node:netlab44@netlab4 count:1293
48 Name:netlab41 Node:netlab41@netlab4 count:3213
49 Name:netlab45 Node:netlab45@netlab4 count:1657
50 Name:xxx Node:xxx_node@netlab3 count:1571
51 Name:netlab34 Node:netlab34@netlab3 count:1649
52 Name:netlab35 Node:netlab35@netlab3 count:2172
53 Name:netlab33 Node:netlab33@netlab3 count:2590
54 Name:netlab32 Node:netlab32@netlab3 count:913

```

```

55 Name:netlab31 Node:netlab31@netlab3 count:1843
56 ok
57 (xxx_node@netlab3)3>

```

List 52: Test Results:Load balancing with 10 nodes per machine

```

1 hirotnk@netlab3 ~/harmonia/Harmonia $ test/test10.sh -s
2 Launched the node:[netlab31@netlab3]
3 Launched the node:[netlab32@netlab3]
4 Launched the node:[netlab33@netlab3]
5 Launched the node:[netlab34@netlab3]
6 Launched the node:[netlab35@netlab3]
7 Launched the node:[netlab36@netlab3]
8 Launched the node:[netlab37@netlab3]
9 Launched the node:[netlab38@netlab3]
10 Launched the node:[netlab39@netlab3]
11 Launched the node:[netlab310@netlab3]
12 epmd: up and running on port 4369 with data:
13 name netlab38 at port 55574
14 name netlab37 at port 36332
15 name netlab36 at port 53886
16 name netlab35 at port 49095
17 name netlab34 at port 55526
18 name netlab32 at port 33998
19 name netlab33 at port 53064
20 name netlab31 at port 41007
21 hirotnk@netlab3 ~/harmonia/Harmonia $ erl -pa ebin -config harmonia -run hm_edge start -\
    -harmonia node_type 'join' name xxx sname 'xxx_node@netlab3' root_node 'netlab31@netlab3' \
    -setcookie harmonia_cookie -sname 'xxx_node@netlab3'
22 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
23
24 Eshell V5.6.3 (abort with ^G)
25 (xxx_node@netlab3)1> [{name,xxx},
26 {root_node,netlab31@netlab3},
27 {root,netlab31},
28 {included_applications,[],},
29 {sname,xxx_node@netlab3},
30 {logfile_ext,".txt"},
31 {node_type,join},
32 {logdir,"log/"},
33 {logfile,"harmonia_log"}]
34 "log/harmonia_log_xxx.txt"
35 start Pid:[<0.149.0>]
36
37 (xxx_node@netlab3)1> hm_cli:log_stop().
38 log stop:[dell1@dell] Result:[ok]
39 log stop:[dell6@dell] Result:[ok]
40 log stop:[dell8@dell] Result:[ok]
41 log stop:[dell2@dell] Result:[ok]
42 log stop:[dell10@dell] Result:[ok]
43 log stop:[dell5@dell] Result:[ok]
44 log stop:[dell4@dell] Result:[ok]
45 log stop:[dell3@dell] Result:[ok]
46 log stop:[dell7@dell] Result:[ok]
47 log stop:[dell9@dell] Result:[ok]
48 log stop:[netlab3@netlab] Result:[ok]
49 log stop:[netlab5@netlab] Result:[ok]
50 log stop:[netlab2@netlab] Result:[ok]
51 log stop:[netlab9@netlab] Result:[ok]
52 log stop:[netlab1@netlab] Result:[ok]

```

```

53 log stop:[netlabb10@netlabb] Result:[ok]
54 log stop:[netlabb4@netlabb] Result:[ok]
55 log stop:[netlabb6@netlabb] Result:[ok]
56 log stop:[netlabb7@netlabb] Result:[ok]
57 log stop:[netlabb8@netlabb] Result:[ok]
58 log stop:[netlaba3@netlaba] Result:[ok]
59 log stop:[netlaba8@netlaba] Result:[ok]
60 log stop:[netlaba9@netlaba] Result:[ok]
61 log stop:[netlaba5@netlaba] Result:[ok]
62 log stop:[netlaba4@netlaba] Result:[ok]
63 log stop:[netlaba2@netlaba] Result:[ok]
64 log stop:[netlaba6@netlaba] Result:[ok]
65 log stop:[netlaba7@netlaba] Result:[ok]
66 log stop:[netlaba10@netlaba] Result:[ok]
67 log stop:[netlaba1@netlaba] Result:[ok]
68 log stop:[netlab43@netlab4] Result:[ok]
69 log stop:[netlab46@netlab4] Result:[ok]
70 log stop:[netlab49@netlab4] Result:[ok]
71 log stop:[netlab410@netlab4] Result:[ok]
72 log stop:[netlab48@netlab4] Result:[ok]
73 log stop:[netlab47@netlab4] Result:[ok]
74 log stop:[netlab45@netlab4] Result:[ok]
75 log stop:[netlab42@netlab4] Result:[ok]
76 log stop:[netlab44@netlab4] Result:[ok]
77 log stop:[netlab41@netlab4] Result:[ok]
78 hm_log_h_file:closing 'log/harmonia_log_xxx.txt'
79 log stop:[xxx_node@netlab3] Result:[ok]
80 log stop:[netlab36@netlab3] Result:[ok]
81 log stop:[netlab37@netlab3] Result:[ok]
82 log stop:[netlab310@netlab3] Result:[ok]
83 log stop:[netlab39@netlab3] Result:[ok]
84 log stop:[netlab38@netlab3] Result:[ok]
85 log stop:[netlab35@netlab3] Result:[ok]
86 log stop:[netlab34@netlab3] Result:[ok]
87 log stop:[netlab32@netlab3] Result:[ok]
88 log stop:[netlab33@netlab3] Result:[ok]
89 log stop:[netlab31@netlab3] Result:[ok]
90 ok
91 (xxx_node@netlab3)2> hm_cli_test:store_short(10000).
92 ok
93 (xxx_node@netlab3)3> hm_cli_test:check_data_num().
94 Name:del11 Node:del11@del1 count:1268
95 Name:del16 Node:del16@del1 count:1106
96 Name:del18 Node:del18@del1 count:228
97 Name:del12 Node:del12@del1 count:1077
98 Name:del110 Node:del110@del1 count:984
99 Name:del15 Node:del15@del1 count:912
100 Name:del14 Node:del14@del1 count:1880
101 Name:del13 Node:del13@del1 count:782
102 Name:del17 Node:del17@del1 count:390
103 Name:del19 Node:del19@del1 count:1647
104 Name:netlabb3 Node:netlabb3@netlabb count:797
105 Name:netlabb5 Node:netlabb5@netlabb count:243
106 Name:netlabb2 Node:netlabb2@netlabb count:1063
107 Name:netlabb9 Node:netlabb9@netlabb count:857
108 Name:netlabb1 Node:netlabb1@netlabb count:1255
109 Name:netlabb10 Node:netlabb10@netlabb count:681
110 Name:netlabb4 Node:netlabb4@netlabb count:1133
111 Name:netlabb6 Node:netlabb6@netlabb count:1305
112 Name:netlabb7 Node:netlabb7@netlabb count:1096

```

```

113 Name:netlabb8 Node:netlabb8@netlabb count:1090
114 Name:netlaba3 Node:netlaba3@netlaba count:824
115 Name:netlaba8 Node:netlaba8@netlaba count:1619
116 Name:netlaba9 Node:netlaba9@netlaba count:1011
117 Name:netlaba5 Node:netlaba5@netlaba count:824
118 Name:netlaba4 Node:netlaba4@netlaba count:1057
119 Name:netlaba2 Node:netlaba2@netlaba count:1833
120 Name:netlaba6 Node:netlaba6@netlaba count:824
121 Name:netlaba7 Node:netlaba7@netlaba count:1039
122 Name:netlaba10 Node:netlaba10@netlaba count:985
123 Name:netlaba1 Node:netlaba1@netlaba count:774
124 Name:netlab43 Node:netlab43@netlab4 count:391
125 Name:netlab46 Node:netlab46@netlab4 count:796
126 Name:netlab49 Node:netlab49@netlab4 count:1197
127 Name:netlab410 Node:netlab410@netlab4 count:929
128 Name:netlab48 Node:netlab48@netlab4 count:972
129 Name:netlab47 Node:netlab47@netlab4 count:374
130 Name:netlab45 Node:netlab45@netlab4 count:1188
131 Name:netlab42 Node:netlab42@netlab4 count:795
132 Name:netlab44 Node:netlab44@netlab4 count:1075
133 Name:netlab41 Node:netlab41@netlab4 count:1031
134 Name:xxx Node:xxx_node@netlab3 count:976
135 Name:netlab36 Node:netlab36@netlab3 count:865
136 Name:netlab37 Node:netlab37@netlab3 count:1800
137 Name:netlab310 Node:netlab310@netlab3 count:990
138 Name:netlab39 Node:netlab39@netlab3 count:1050
139 Name:netlab38 Node:netlab38@netlab3 count:1226
140 Name:netlab35 Node:netlab35@netlab3 count:919
141 Name:netlab34 Node:netlab34@netlab3 count:1091
142 Name:netlab32 Node:netlab32@netlab3 count:426
143 Name:netlab33 Node:netlab33@netlab3 count:805
144 Name:netlab31 Node:netlab31@netlab3 count:520
145 ok
146 (xxx_node@netlab3)4>

```

List 53: Test Results:Load balancing with 20 nodes per machine

```

1 hirotnk@netlab3 ~/harmonia/Harmonia $ test/test20.sh -s
2 Launched the node: [netlab31@netlab3]
3 Launched the node: [netlab32@netlab3]
4 Launched the node: [netlab33@netlab3]
5 Launched the node: [netlab34@netlab3]
6 Launched the node: [netlab35@netlab3]
7 Launched the node: [netlab36@netlab3]
8 Launched the node: [netlab37@netlab3]
9 Launched the node: [netlab38@netlab3]
10 Launched the node: [netlab39@netlab3]
11 Launched the node: [netlab310@netlab3]
12 Launched the node: [netlab311@netlab3]
13 Launched the node: [netlab312@netlab3]
14 Launched the node: [netlab313@netlab3]
15 Launched the node: [netlab314@netlab3]
16 Launched the node: [netlab315@netlab3]
17 Launched the node: [netlab316@netlab3]
18 Launched the node: [netlab317@netlab3]
19 Launched the node: [netlab318@netlab3]
20 Launched the node: [netlab319@netlab3]
21 Launched the node: [netlab320@netlab3]
22 epmd: up and running on port 4369 with data:
23 name netlab318 at port 47425

```

```

24 name netlab317 at port 43219
25 name netlab316 at port 37070
26 name netlab315 at port 41974
27 name netlab314 at port 55565
28 name netlab313 at port 35794
29 name netlab312 at port 40085
30 name netlab311 at port 52843
31 name netlab310 at port 55175
32 name netlab39 at port 36536
33 name netlab38 at port 54983
34 name netlab36 at port 54023
35 name netlab37 at port 46562
36 name netlab35 at port 38514
37 name netlab34 at port 39453
38 name netlab32 at port 55082
39 name netlab33 at port 42509
40 name netlab31 at port 36113
41 hirotnk@netlab3 ~/harmonia/Harmonia $ erl -pa ebin -config harmonia -run hm_edge start -\
    -harmonia node_type 'join' name xxx sname 'xxx_node@netlab3' root_node 'netlab31@netlab3' \
    ->-setcookie harmonia_cookie -sname 'xxx_node@netlab3'
42 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
43
44 Eshell V5.6.3 (abort with ^G)
45 (xxx_node@netlab3)1> [{name,xxx},
46 {root_node,netlab31@netlab3},
47 {root,netlab31},
48 {included_applications,[]},
49 {sname,xxx_node@netlab3},
50 {logfile_ext, ".txt"},
51 {node_type,join},
52 {logdir,"log/"},
53 {logfile,"harmonia_log"}]
54 "log/harmonia_log_xxx.txt"
55 start Pid:[<0.253.0>]
56
57 (xxx_node@netlab3)1>
58 (xxx_node@netlab3)1> hm_cli:log_stop().
59 log stop:[netlab37@netlab3] Result:[ok]
60 log stop:[dell1@ dell] Result:[ok]
61 log stop:[dell3@ dell] Result:[ok]
62 log stop:[dell12@ dell] Result:[ok]
63 log stop:[netlab317@netlab3] Result:[ok]
64 log stop:[netlab312@netlab3] Result:[ok]
65 log stop:[dell15@ dell] Result:[ok]
66 log stop:[netlab311@netlab3] Result:[ok]
67 log stop:[netlab34@netlab3] Result:[ok]
68 log stop:[dell9@ dell] Result:[ok]
69 log stop:[netlab36@netlab3] Result:[ok]
70 log stop:[dell13@ dell] Result:[ok]
71 log stop:[dell16@ dell] Result:[ok]
72 log stop:[dell6@ dell] Result:[ok]
73 log stop:[netlab38@netlab3] Result:[ok]
74 log stop:[netlab35@netlab3] Result:[ok]
75 log stop:[dell5@ dell] Result:[ok]
76 log stop:[dell19@ dell] Result:[ok]
77 log stop:[netlab39@netlab3] Result:[ok]
78 log stop:[netlab320@netlab3] Result:[ok]
79 log stop:[dell11@ dell] Result:[ok]
80 log stop:[dell4@ dell] Result:[ok]
81 log stop:[dell2@ dell] Result:[ok]

```

```

82 log stop:[dell8@dell] Result:[ok]
83 log stop:[netlabal0@netlaba] Result:[ok]
84 log stop:[dell17@dell] Result:[ok]
85 log stop:[dell14@dell] Result:[ok]
86 log stop:[netlabbb4@netlabbb] Result:[ok]
87 log stop:[dell7@dell] Result:[ok]
88 log stop:[netlabbb16@netlabbb] Result:[ok]
89 log stop:[dell110@dell] Result:[ok]
90 log stop:[netlabbb19@netlabbb] Result:[ok]
91 log stop:[dell20@dell] Result:[ok]
92 log stop:[netlabbb14@netlabbb] Result:[ok]
93 log stop:[netlabbb15@netlabbb] Result:[ok]
94 log stop:[dell118@dell] Result:[ok]
95 log stop:[netlabbb13@netlabbb] Result:[ok]
96 log stop:[netlabbb10@netlabbb] Result:[ok]
97 log stop:[netlabbb2@netlabbb] Result:[ok]
98 log stop:[netlabbb3@netlabbb] Result:[ok]
99 log stop:[netlabbb18@netlabbb] Result:[ok]
100 log stop:[netlabbb1@netlabbb] Result:[ok]
101 log stop:[netlabbb5@netlabbb] Result:[ok]
102 log stop:[netlabal9@netlaba] Result:[ok]
103 log stop:[netlaba8@netlaba] Result:[ok]
104 log stop:[netlaba6@netlaba] Result:[ok]
105 log stop:[netlaba20@netlaba] Result:[ok]
106 log stop:[netlabal3@netlaba] Result:[ok]
107 log stop:[netlabal7@netlaba] Result:[ok]
108 log stop:[netlaba2@netlaba] Result:[ok]
109 log stop:[netlaba9@netlaba] Result:[ok]
110 log stop:[netlaba7@netlaba] Result:[ok]
111 log stop:[netlabal1@netlaba] Result:[ok]
112 log stop:[netlabal8@netlaba] Result:[ok]
113 log stop:[netlabal6@netlaba] Result:[ok]
114 log stop:[netlabal4@netlaba] Result:[ok]
115 log stop:[netlabal2@netlaba] Result:[ok]
116 log stop:[netlabal5@netlaba] Result:[ok]
117 log stop:[netlaba5@netlaba] Result:[ok]
118 log stop:[netlabal@netlaba] Result:[ok]
119 log stop:[netlab416@netlab4] Result:[ok]
120 log stop:[netlab413@netlab4] Result:[ok]
121 log stop:[netlab417@netlab4] Result:[ok]
122 log stop:[netlab412@netlab4] Result:[ok]
123 log stop:[netlab414@netlab4] Result:[ok]
124 log stop:[netlab411@netlab4] Result:[ok]
125 log stop:[netlab44@netlab4] Result:[ok]
126 log stop:[netlab42@netlab4] Result:[ok]
127 log stop:[netlab48@netlab4] Result:[ok]
128 log stop:[netlab47@netlab4] Result:[ok]
129 log stop:[netlab43@netlab4] Result:[ok]
130 log stop:[netlab419@netlab4] Result:[ok]
131 log stop:[netlab420@netlab4] Result:[ok]
132 log stop:[netlab49@netlab4] Result:[ok]
133 log stop:[netlab418@netlab4] Result:[ok]
134 log stop:[netlab415@netlab4] Result:[ok]
135 log stop:[netlab410@netlab4] Result:[ok]
136 log stop:[netlab46@netlab4] Result:[ok]
137 log stop:[netlab45@netlab4] Result:[ok]
138 log stop:[netlab41@netlab4] Result:[ok]
139 log stop:[netlab316@netlab3] Result:[ok]
140 hm_log_h_file:closing 'log/harmonia_log_xxx.txt'
141 log stop:[xxx_node@netlab3] Result:[ok]

```



```

142 log stop:[netlab319@netlab3] Result:[ok]
143 log stop:[netlab39@netlab3] Result:[ok]
144 log stop:[netlab312@netlab3] Result:[ok]
145 log stop:[netlab320@netlab3] Result:[ok]
146 log stop:[netlab33@netlab3] Result:[ok]
147 log stop:[netlab34@netlab3] Result:[ok]
148 log stop:[netlab314@netlab3] Result:[ok]
149 log stop:[netlab318@netlab3] Result:[ok]
150 log stop:[netlab317@netlab3] Result:[ok]
151 log stop:[netlab315@netlab3] Result:[ok]
152 log stop:[netlab313@netlab3] Result:[ok]
153 log stop:[netlab310@netlab3] Result:[ok]
154 log stop:[netlab37@netlab3] Result:[ok]
155 log stop:[netlab311@netlab3] Result:[ok]
156 log stop:[netlab36@netlab3] Result:[ok]
157 log stop:[netlab38@netlab3] Result:[ok]
158 log stop:[netlab35@netlab3] Result:[ok]
159 log stop:[netlab32@netlab3] Result:[ok]
160 log stop:[netlab31@netlab3] Result:[ok]
161 ok
162 (xxx_node@netlab3)2> hm_cli_test:store_short(10000).
163 ok
164 (xxx_node@netlab3)3> hm_cli_test:check_data_num().
165 Name:netlabbb7 Node:netlabbb7@netlabbb count:472
166 Name:dell1 Node:dell1@dell count:333
167 Name:dell3 Node:dell3@dell count:398
168 Name:dell12 Node:dell12@dell count:572
169 Name:netlabbb17 Node:netlabbb17@netlabbb count:336
170 Name:netlabbb12 Node:netlabbb12@netlabbb count:602
171 Name:dell15 Node:dell15@dell count:211
172 Name:netlabbb11 Node:netlabbb11@netlabbb count:176
173 Name:netlaba4 Node:netlaba4@netlaba count:715
174 Name:dell9 Node:dell9@dell count:342
175 Name:netlabbb6 Node:netlabbb6@netlabbb count:704
176 Name:dell13 Node:dell13@dell count:212
177 Name:dell16 Node:dell16@dell count:338
178 Name:dell6 Node:dell6@dell count:333
179 Name:netlabbb8 Node:netlabbb8@netlabbb count:755
180 Name:netlaba3 Node:netlaba3@netlaba count:352
181 Name:dell5 Node:dell5@dell count:289
182 Name:dell19 Node:dell19@dell count:312
183 Name:netlabbb9 Node:netlabbb9@netlabbb count:453
184 Name:netlabbb20 Node:netlabbb20@netlabbb count:1331
185 Name:dell11 Node:dell11@dell count:467
186 Name:dell4 Node:dell4@dell count:1377
187 Name:dell2 Node:dell2@dell count:466
188 Name:dell8 Node:dell8@dell count:220
189 Name:netlaba10 Node:netlaba10@netlaba count:608
190 Name:dell17 Node:dell17@dell count:417
191 Name:dell14 Node:dell14@dell count:177
192 Name:netlabbb4 Node:netlabbb4@netlabbb count:1017
193 Name:dell7 Node:dell7@dell count:323
194 Name:netlabbb16 Node:netlabbb16@netlabbb count:252
195 Name:dell10 Node:dell10@dell count:345
196 Name:netlabbb19 Node:netlabbb19@netlabbb count:1372
197 Name:dell20 Node:dell20@dell count:400
198 Name:netlabbb14 Node:netlabbb14@netlabbb count:634
199 Name:netlabbb15 Node:netlabbb15@netlabbb count:313
200 Name:dell18 Node:dell18@dell count:413
201 Name:netlabbb13 Node:netlabbb13@netlabbb count:318

```

202 Name:netlabbb10 Node:netlabbb10@netlabbb count:299
203 Name:netlabbb2 Node:netlabbb2@netlabbb count:500
204 Name:netlabbb3 Node:netlabbb3@netlabbb count:289
205 Name:netlabbb18 Node:netlabbb18@netlabbb count:530
206 Name:netlabbb1 Node:netlabbb1@netlabbb count:366
207 Name:netlabbb5 Node:netlabbb5@netlabbb count:213
208 Name:netlaba19 Node:netlaba19@netlaba count:718
209 Name:netlaba8 Node:netlaba8@netlaba count:439
210 Name:netlaba6 Node:netlaba6@netlaba count:773
211 Name:netlaba20 Node:netlaba20@netlaba count:436
212 Name:netlaba13 Node:netlaba13@netlaba count:341
213 Name:netlaba17 Node:netlaba17@netlaba count:289
214 Name:netlaba2 Node:netlaba2@netlaba count:1466
215 Name:netlaba9 Node:netlaba9@netlaba count:486
216 Name:netlaba7 Node:netlaba7@netlaba count:597
217 Name:netlabal1 Node:netlabal1@netlaba count:248
218 Name:netlabal8 Node:netlabal8@netlaba count:357
219 Name:netlabal6 Node:netlabal6@netlaba count:1022
220 Name:netlabal4 Node:netlabal4@netlaba count:283
221 Name:netlabal2 Node:netlabal2@netlaba count:288
222 Name:netlabal5 Node:netlabal5@netlaba count:398
223 Name:netlaba5 Node:netlaba5@netlaba count:797
224 Name:netlabal Node:netlabal@netlaba count:449
225 Name:netlab416 Node:netlab416@netlab4 count:373
226 Name:netlab413 Node:netlab413@netlab4 count:420
227 Name:netlab417 Node:netlab417@netlab4 count:406
228 Name:netlab412 Node:netlab412@netlab4 count:881
229 Name:netlab414 Node:netlab414@netlab4 count:367
230 Name:netlab411 Node:netlab411@netlab4 count:553
231 Name:netlab44 Node:netlab44@netlab4 count:1054
232 Name:netlab42 Node:netlab42@netlab4 count:164
233 Name:netlab48 Node:netlab48@netlab4 count:318
234 Name:netlab47 Node:netlab47@netlab4 count:370
235 Name:netlab43 Node:netlab43@netlab4 count:324
236 Name:netlab419 Node:netlab419@netlab4 count:557
237 Name:netlab420 Node:netlab420@netlab4 count:404
238 Name:netlab49 Node:netlab49@netlab4 count:628
239 Name:netlab418 Node:netlab418@netlab4 count:386
240 Name:netlab415 Node:netlab415@netlab4 count:644
241 Name:netlab410 Node:netlab410@netlab4 count:529
242 Name:netlab46 Node:netlab46@netlab4 count:368
243 Name:netlab45 Node:netlab45@netlab4 count:307
244 Name:netlab41 Node:netlab41@netlab4 count:667
245 Name:netlab316 Node:netlab316@netlab3 count:609
246 Name:xxx Node:xxx_node@netlab3 count:676
247 Name:netlab319 Node:netlab319@netlab3 count:472
248 Name:netlab39 Node:netlab39@netlab3 count:374
249 Name:netlab312 Node:netlab312@netlab3 count:608
250 Name:netlab320 Node:netlab320@netlab3 count:384
251 Name:netlab33 Node:netlab33@netlab3 count:517
252 Name:netlab34 Node:netlab34@netlab3 count:595
253 Name:netlab314 Node:netlab314@netlab3 count:569
254 Name:netlab318 Node:netlab318@netlab3 count:752
255 Name:netlab317 Node:netlab317@netlab3 count:324
256 Name:netlab315 Node:netlab315@netlab3 count:261
257 Name:netlab313 Node:netlab313@netlab3 count:229
258 Name:netlab310 Node:netlab310@netlab3 count:449
259 Name:netlab37 Node:netlab37@netlab3 count:428
260 Name:netlab311 Node:netlab311@netlab3 count:1322
261 Name:netlab36 Node:netlab36@netlab3 count:184

```

262 Name:netlab38 Node:netlab38@netlab3 count:472
263 Name:netlab35 Node:netlab35@netlab3 count:558
264 Name:netlab32 Node:netlab32@netlab3 count:382
265 Name:netlab31 Node:netlab31@netlab3 count:176
266 ok
267 (xxx_node@netlab3)4>

```

List 54: Test Results:Load balancing with 40 nodes per machine

```

1 hirotnk@netlab3 ~/harmonia/Harmonia $ test/test40.sh -s
2 Launched the node: [netlab31@netlab3]
3 Launched the node: [netlab32@netlab3]
4 Launched the node: [netlab33@netlab3]
5 Launched the node: [netlab34@netlab3]
6 Launched the node: [netlab35@netlab3]
7 Launched the node: [netlab36@netlab3]
8 Launched the node: [netlab37@netlab3]
9 Launched the node: [netlab38@netlab3]
10 Launched the node: [netlab39@netlab3]
11 Launched the node: [netlab310@netlab3]
12 Launched the node: [netlab311@netlab3]
13 Launched the node: [netlab312@netlab3]
14 Launched the node: [netlab313@netlab3]
15 Launched the node: [netlab314@netlab3]
16 Launched the node: [netlab315@netlab3]
17 Launched the node: [netlab316@netlab3]
18 Launched the node: [netlab317@netlab3]
19 Launched the node: [netlab318@netlab3]
20 Launched the node: [netlab319@netlab3]
21 Launched the node: [netlab320@netlab3]
22 Launched the node: [netlab321@netlab3]
23 Launched the node: [netlab322@netlab3]
24 Launched the node: [netlab323@netlab3]
25 Launched the node: [netlab324@netlab3]
26 Launched the node: [netlab325@netlab3]
27 Launched the node: [netlab326@netlab3]
28 Launched the node: [netlab327@netlab3]
29 Launched the node: [netlab328@netlab3]
30 Launched the node: [netlab329@netlab3]
31 Launched the node: [netlab330@netlab3]
32 Launched the node: [netlab331@netlab3]
33 Launched the node: [netlab332@netlab3]
34 Launched the node: [netlab333@netlab3]
35 Launched the node: [netlab334@netlab3]
36 Launched the node: [netlab335@netlab3]
37 Launched the node: [netlab336@netlab3]
38 Launched the node: [netlab337@netlab3]
39 Launched the node: [netlab338@netlab3]
40 Launched the node: [netlab339@netlab3]
41 Launched the node: [netlab340@netlab3]
42 epmd: up and running on port 4369 with data:
43 name netlab336 at port 44437
44 name netlab335 at port 48765
45 name netlab333 at port 44152
46 name netlab332 at port 37271
47 name netlab330 at port 58569
48 name netlab331 at port 44804
49 name netlab328 at port 42182
50 name netlab329 at port 54008
51 name netlab327 at port 41155

```

```

52 name netlab326 at port 39569
53 name netlab324 at port 36459
54 name netlab325 at port 59340
55 name netlab323 at port 39322
56 name netlab322 at port 47324
57 name netlab321 at port 56202
58 name netlab319 at port 55754
59 name netlab320 at port 41728
60 name netlab318 at port 34629
61 name netlab316 at port 47136
62 name netlab317 at port 48657
63 name netlab313 at port 45815
64 name netlab314 at port 33069
65 name netlab315 at port 41144
66 name netlab312 at port 50679
67 name netlab38 at port 45085
68 name netlab311 at port 56535
69 name netlab37 at port 49482
70 name netlab310 at port 57924
71 name netlab39 at port 40405
72 name netlab36 at port 33598
73 name netlab35 at port 48060
74 name netlab34 at port 51029
75 name netlab33 at port 43583
76 name netlab32 at port 42600
77 name netlab31 at port 37589
78 hirotnk@netlab3 ~/harmonia/Harmonia $ erl -pa ebin -config harmonia -run hm_edge start -\
    ->harmonia node_type 'join' name xxx sname 'xxx_node@netlab3' root_node 'netlab31@netlab3' \
    ->setcookie harmonia_cookie -sname 'xxx_node@netlab3'
79 Erlang (BEAM) emulator version 5.6.3 [source] [async-threads:0]
80
81 Eshell V5.6.3 (abort with ^G)
82 (xxx_node@netlab3)1> [{name,xxx},
83 {root_node,netlab31@netlab3},
84 {root,netlab31},
85 {included_applications,[],},
86 {sname,xxx_node@netlab3},
87 {logfile_ext,".txt"},
88 {node_type,join},
89 {logdir,"log/"},
90 {logfile,"harmonia_log"}]
91 "log/harmonia_log_xxx.txt"
92 start Pid:[<0.390.0>]
93
94 (xxx_node@netlab3)1>
95 (xxx_node@netlab3)1>
96 (xxx_node@netlab3)1>
97 (xxx_node@netlab3)1>
98 (xxx_node@netlab3)1> hm_cli:log_stop().
99 log stop:[dell39@dell] Result:[ok]
100 log stop:[netlabal@netlaba] Result:[ok]
101 log stop:[dell3@dell] Result:[ok]
102 log stop:[netlabal5@netlaba] Result:[ok]
103 log stop:[netlabbl4@netlabbb] Result:[ok]
104 log stop:[dell18@dell] Result:[ok]
105 log stop:[netlaba4@netlaba] Result:[ok]
106 log stop:[dell16@dell] Result:[ok]
107 log stop:[netlaba39@netlaba] Result:[ok]
108 log stop:[netlabal6@netlaba] Result:[ok]
109 log stop:[dell20@dell] Result:[ok]

```

```
110 log stop:[dell29@dell] Result:[ok]
111 log stop:[dell31@dell] Result:[ok]
112 log stop:[netlaba13@netlaba] Result:[ok]
113 log stop:[netlaba28@netlaba] Result:[ok]
114 log stop:[dell6@dell] Result:[ok]
115 log stop:[dell23@dell] Result:[ok]
116 log stop:[netlabbb31@netlabbb] Result:[ok]
117 log stop:[netlabbb7@netlabbb] Result:[ok]
118 log stop:[netlaba11@netlaba] Result:[ok]
119 log stop:[netlaba17@netlaba] Result:[ok]
120 log stop:[dell5@dell] Result:[ok]
121 log stop:[netlaba20@netlaba] Result:[ok]
122 log stop:[dell33@dell] Result:[ok]
123 log stop:[netlabbb26@netlabbb] Result:[ok]
124 log stop:[netlabbb19@netlabbb] Result:[ok]
125 log stop:[netlabbb2@netlabbb] Result:[ok]
126 log stop:[dell9@dell] Result:[ok]
127 log stop:[dell7@dell] Result:[ok]
128 log stop:[dell2@dell] Result:[ok]
129 log stop:[netlabbb29@netlabbb] Result:[ok]
130 log stop:[netlaba9@netlaba] Result:[ok]
131 log stop:[netlaba36@netlaba] Result:[ok]
132 log stop:[netlaba34@netlaba] Result:[ok]
133 log stop:[dell21@dell] Result:[ok]
134 log stop:[netlabbb37@netlabbb] Result:[ok]
135 log stop:[netlabbb20@netlabbb] Result:[ok]
136 log stop:[dell36@dell] Result:[ok]
137 log stop:[netlabbb9@netlabbb] Result:[ok]
138 log stop:[netlaba38@netlaba] Result:[ok]
139 log stop:[netlaba25@netlaba] Result:[ok]
140 log stop:[netlabbb11@netlabbb] Result:[ok]
141 log stop:[dell1@dell] Result:[ok]
142 log stop:[netlaba3@netlaba] Result:[ok]
143 log stop:[dell22@dell] Result:[ok]
144 log stop:[netlabbb8@netlabbb] Result:[ok]
145 log stop:[dell14@dell] Result:[ok]
146 log stop:[dell15@dell] Result:[ok]
147 log stop:[netlabbb4@netlabbb] Result:[ok]
148 log stop:[netlabbb38@netlabbb] Result:[ok]
149 log stop:[netlaba2@netlaba] Result:[ok]
150 log stop:[dell32@dell] Result:[ok]
151 log stop:[netlabbb28@netlabbb] Result:[ok]
152 log stop:[netlaba21@netlaba] Result:[ok]
153 log stop:[netlaba10@netlaba] Result:[ok]
154 log stop:[netlaba5@netlaba] Result:[ok]
155 log stop:[netlabbb33@netlabbb] Result:[ok]
156 log stop:[netlaba40@netlaba] Result:[ok]
157 log stop:[netlabbb22@netlabbb] Result:[ok]
158 log stop:[netlabbb17@netlabbb] Result:[ok]
159 log stop:[netlabbb30@netlabbb] Result:[ok]
160 log stop:[netlabbb13@netlabbb] Result:[ok]
161 log stop:[dell35@dell] Result:[ok]
162 log stop:[netlaba27@netlaba] Result:[ok]
163 log stop:[dell38@dell] Result:[ok]
164 log stop:[netlabbb15@netlabbb] Result:[ok]
165 log stop:[netlaba29@netlaba] Result:[ok]
166 log stop:[dell25@dell] Result:[ok]
167 log stop:[netlabbb10@netlabbb] Result:[ok]
168 log stop:[dell13@dell] Result:[ok]
169 log stop:[dell12@dell] Result:[ok]
```

```
170 log stop:[dell28@dell] Result:[ok]
171 log stop:[netlabbb12@netlabbb] Result:[ok]
172 log stop:[netlabbb3@netlabbb] Result:[ok]
173 log stop:[netlab413@netlab4] Result:[ok]
174 log stop:[netlaba35@netlaba] Result:[ok]
175 log stop:[dell34@dell] Result:[ok]
176 log stop:[dell40@dell] Result:[ok]
177 log stop:[netlabbb1@netlabbb] Result:[ok]
178 log stop:[dell4@dell] Result:[ok]
179 log stop:[dell26@dell] Result:[ok]
180 log stop:[dell10@dell] Result:[ok]
181 log stop:[netlabbb5@netlabbb] Result:[ok]
182 log stop:[netlabbb6@netlabbb] Result:[ok]
183 log stop:[dell11@dell] Result:[ok]
184 log stop:[netlaba22@netlaba] Result:[ok]
185 log stop:[dell37@dell] Result:[ok]
186 log stop:[netlabbb24@netlabbb] Result:[ok]
187 log stop:[netlaba30@netlaba] Result:[ok]
188 log stop:[netlaba8@netlaba] Result:[ok]
189 log stop:[netlaba18@netlaba] Result:[ok]
190 log stop:[dell27@dell] Result:[ok]
191 log stop:[netlabbb34@netlabbb] Result:[ok]
192 log stop:[netlaba32@netlaba] Result:[ok]
193 log stop:[netlaba24@netlaba] Result:[ok]
194 log stop:[netlaba19@netlaba] Result:[ok]
195 log stop:[netlaba12@netlaba] Result:[ok]
196 log stop:[netlabbb35@netlabbb] Result:[ok]
197 log stop:[netlabbb36@netlabbb] Result:[ok]
198 log stop:[dell30@dell] Result:[ok]
199 log stop:[netlabbb25@netlabbb] Result:[ok]
200 log stop:[netlaba33@netlaba] Result:[ok]
201 log stop:[netlabbb32@netlabbb] Result:[ok]
202 log stop:[dell8@dell] Result:[ok]
203 log stop:[netlabbb21@netlabbb] Result:[ok]
204 log stop:[netlaba14@netlaba] Result:[ok]
205 log stop:[netlaba37@netlaba] Result:[ok]
206 log stop:[dell19@dell] Result:[ok]
207 log stop:[netlabbb40@netlabbb] Result:[ok]
208 log stop:[netlabbb23@netlabbb] Result:[ok]
209 log stop:[netlaba7@netlaba] Result:[ok]
210 log stop:[netlabbb16@netlabbb] Result:[ok]
211 log stop:[netlaba31@netlaba] Result:[ok]
212 log stop:[dell24@dell] Result:[ok]
213 log stop:[netlaba6@netlaba] Result:[ok]
214 log stop:[netlaba26@netlaba] Result:[ok]
215 log stop:[netlabbb39@netlabbb] Result:[ok]
216 log stop:[netlaba23@netlaba] Result:[ok]
217 log stop:[netlabbb27@netlabbb] Result:[ok]
218 log stop:[netlabbb18@netlabbb] Result:[ok]
219 log stop:[dell17@dell] Result:[ok]
220 log stop:[netlab426@netlab4] Result:[ok]
221 log stop:[netlab433@netlab4] Result:[ok]
222 log stop:[netlab435@netlab4] Result:[ok]
223 log stop:[netlab414@netlab4] Result:[ok]
224 log stop:[netlab423@netlab4] Result:[ok]
225 log stop:[netlab410@netlab4] Result:[ok]
226 log stop:[netlab416@netlab4] Result:[ok]
227 log stop:[netlab417@netlab4] Result:[ok]
228 log stop:[netlab44@netlab4] Result:[ok]
229 log stop:[netlab431@netlab4] Result:[ok]
```

```

230 log stop:[netlab412@netlab4] Result:[ok]
231 log stop:[netlab440@netlab4] Result:[ok]
232 log stop:[netlab434@netlab4] Result:[ok]
233 log stop:[netlab411@netlab4] Result:[ok]
234 log stop:[netlab43@netlab4] Result:[ok]
235 log stop:[netlab432@netlab4] Result:[ok]
236 log stop:[netlab438@netlab4] Result:[ok]
237 log stop:[netlab429@netlab4] Result:[ok]
238 log stop:[netlab430@netlab4] Result:[ok]
239 log stop:[netlab436@netlab4] Result:[ok]
240 log stop:[netlab419@netlab4] Result:[ok]
241 log stop:[netlab47@netlab4] Result:[ok]
242 log stop:[netlab439@netlab4] Result:[ok]
243 log stop:[netlab427@netlab4] Result:[ok]
244 log stop:[netlab437@netlab4] Result:[ok]
245 log stop:[netlab415@netlab4] Result:[ok]
246 log stop:[netlab425@netlab4] Result:[ok]
247 log stop:[netlab45@netlab4] Result:[ok]
248 log stop:[netlab428@netlab4] Result:[ok]
249 log stop:[netlab424@netlab4] Result:[ok]
250 log stop:[netlab418@netlab4] Result:[ok]
251 log stop:[netlab422@netlab4] Result:[ok]
252 log stop:[netlab48@netlab4] Result:[ok]
253 log stop:[netlab420@netlab4] Result:[ok]
254 log stop:[netlab421@netlab4] Result:[ok]
255 log stop:[netlab49@netlab4] Result:[ok]
256 log stop:[netlab41@netlab4] Result:[ok]
257 log stop:[netlab46@netlab4] Result:[ok]
258 log stop:[netlab42@netlab4] Result:[ok]
259 log stop:[netlab323@netlab3] Result:[ok]
260 hm_log_h_file:closing 'log/harmonia_log_xxx.txt'
261 log stop:[xxx_node@netlab3] Result:[ok]
262 log stop:[netlab311@netlab3] Result:[ok]
263 log stop:[netlab316@netlab3] Result:[ok]
264 log stop:[netlab332@netlab3] Result:[ok]
265 log stop:[netlab330@netlab3] Result:[ok]
266 log stop:[netlab339@netlab3] Result:[ok]
267 log stop:[netlab334@netlab3] Result:[ok]
268 log stop:[netlab336@netlab3] Result:[ok]
269 log stop:[netlab337@netlab3] Result:[ok]
270 log stop:[netlab331@netlab3] Result:[ok]
271 log stop:[netlab328@netlab3] Result:[ok]
272 log stop:[netlab318@netlab3] Result:[ok]
273 log stop:[netlab335@netlab3] Result:[ok]
274 log stop:[netlab327@netlab3] Result:[ok]
275 log stop:[netlab340@netlab3] Result:[ok]
276 log stop:[netlab35@netlab3] Result:[ok]
277 log stop:[netlab325@netlab3] Result:[ok]
278 log stop:[netlab315@netlab3] Result:[ok]
279 log stop:[netlab322@netlab3] Result:[ok]
280 log stop:[netlab329@netlab3] Result:[ok]
281 log stop:[netlab321@netlab3] Result:[ok]
282 log stop:[netlab324@netlab3] Result:[ok]
283 log stop:[netlab338@netlab3] Result:[ok]
284 log stop:[netlab313@netlab3] Result:[ok]
285 log stop:[netlab333@netlab3] Result:[ok]
286 log stop:[netlab37@netlab3] Result:[ok]
287 log stop:[netlab319@netlab3] Result:[ok]
288 log stop:[netlab36@netlab3] Result:[ok]
289 log stop:[netlab326@netlab3] Result:[ok]

```



```

290 log stop:[netlab312@netlab3] Result:[ok]
291 log stop:[netlab320@netlab3] Result:[ok]
292 log stop:[netlab317@netlab3] Result:[ok]
293 log stop:[netlab314@netlab3] Result:[ok]
294 log stop:[netlab38@netlab3] Result:[ok]
295 log stop:[netlab34@netlab3] Result:[ok]
296 log stop:[netlab310@netlab3] Result:[ok]
297 log stop:[netlab39@netlab3] Result:[ok]
298 log stop:[netlab32@netlab3] Result:[ok]
299 log stop:[netlab33@netlab3] Result:[ok]
300 log stop:[netlab31@netlab3] Result:[ok]
301 ok
302 (xxx_node@netlab3)2> hm_cli_test:store_short(10000).
303 ok
304 (xxx_node@netlab3)3> hm_cli_test:check_data_num().
305 Name:dell39 Node:dell39@dell count:77
306 Name:netlaba1 Node:netlaba1@netlaba count:214
307 Name:dell3 Node:dell3@dell count:113
308 Name:netlaba15 Node:netlaba15@netlaba count:254
309 Name:netlabb14 Node:netlabb14@netlabb count:261
310 Name:dell18 Node:dell18@dell count:276
311 Name:netlaba4 Node:netlaba4@netlaba count:187
312 Name:dell16 Node:dell16@dell count:300
313 Name:netlaba39 Node:netlaba39@netlaba count:282
314 Name:netlaba16 Node:netlaba16@netlaba count:490
315 Name:dell20 Node:dell20@dell count:400
316 Name:dell29 Node:dell29@dell count:325
317 Name:dell31 Node:dell31@dell count:163
318 Name:netlaba13 Node:netlaba13@netlaba count:201
319 Name:netlaba28 Node:netlaba28@netlaba count:110
320 Name:dell6 Node:dell6@dell count:161
321 Name:dell23 Node:dell23@dell count:407
322 Name:netlabb31 Node:netlabb31@netlabb count:294
323 Name:netlabb7 Node:netlabb7@netlabb count:212
324 Name:netlaba11 Node:netlaba11@netlaba count:168
325 Name:netlaba17 Node:netlaba17@netlaba count:289
326 Name:dell5 Node:dell5@dell count:284
327 Name:netlaba20 Node:netlaba20@netlaba count:123
328 Name:dell33 Node:dell33@dell count:218
329 Name:netlabb26 Node:netlabb26@netlabb count:354
330 Name:netlabb19 Node:netlabb19@netlabb count:397
331 Name:netlabb2 Node:netlabb2@netlabb count:361
332 Name:dell9 Node:dell9@dell count:184
333 Name:dell7 Node:dell7@dell count:227
334 Name:dell2 Node:dell2@dell count:230
335 Name:netlabb29 Node:netlabb29@netlabb count:271
336 Name:netlaba9 Node:netlaba9@netlaba count:164
337 Name:netlaba36 Node:netlaba36@netlaba count:195
338 Name:netlaba34 Node:netlaba34@netlaba count:326
339 Name:dell21 Node:dell21@dell count:193
340 Name:netlabb37 Node:netlabb37@netlabb count:263
341 Name:netlabb20 Node:netlabb20@netlabb count:216
342 Name:dell36 Node:dell36@dell count:334
343 Name:netlabb9 Node:netlabb9@netlabb count:109
344 Name:netlaba38 Node:netlaba38@netlaba count:346
345 Name:netlaba25 Node:netlaba25@netlaba count:451
346 Name:netlabb11 Node:netlabb11@netlabb count:66
347 Name:dell1 Node:dell1@dell count:225
348 Name:netlaba3 Node:netlaba3@netlaba count:257
349 Name:dell22 Node:dell22@dell count:80

```



```

350 Name:netlab8 Node:netlab8@netlab count:220
351 Name:dell14 Node:dell14@dell count:173
352 Name:dell15 Node:dell15@dell count:125
353 Name:netlab4 Node:netlab4@netlab count:510
354 Name:netlab38 Node:netlab38@netlab count:219
355 Name:netlaba2 Node:netlaba2@netlaba count:416
356 Name:dell32 Node:dell32@dell count:336
357 Name:netlab28 Node:netlab28@netlab count:360
358 Name:netlaba21 Node:netlaba21@netlaba count:260
359 Name:netlaba10 Node:netlaba10@netlaba count:151
360 Name:netlaba5 Node:netlaba5@netlaba count:395
361 Name:netlab33 Node:netlab33@netlab count:281
362 Name:netlaba40 Node:netlaba40@netlaba count:428
363 Name:netlab22 Node:netlab22@netlab count:167
364 Name:netlab17 Node:netlab17@netlab count:196
365 Name:netlab30 Node:netlab30@netlab count:200
366 Name:netlab13 Node:netlab13@netlab count:285
367 Name:dell35 Node:dell35@dell count:429
368 Name:netlaba27 Node:netlaba27@netlaba count:229
369 Name:dell38 Node:dell38@dell count:453
370 Name:netlab15 Node:netlab15@netlab count:313
371 Name:netlaba29 Node:netlaba29@netlaba count:254
372 Name:dell25 Node:dell25@dell count:184
373 Name:netlab10 Node:netlab10@netlab count:154
374 Name:dell13 Node:dell13@dell count:126
375 Name:dell12 Node:dell12@dell count:122
376 Name:dell28 Node:dell28@dell count:264
377 Name:netlab12 Node:netlab12@netlab count:440
378 Name:netlab3 Node:netlab3@netlab count:289
379 Name:netlab413 Node:netlab413@netlab4 count:197
380 Name:netlaba35 Node:netlaba35@netlaba count:307
381 Name:dell34 Node:dell34@dell count:304
382 Name:dell40 Node:dell40@dell count:194
383 Name:netlab1 Node:netlab1@netlab count:147
384 Name:dell4 Node:dell4@dell count:309
385 Name:dell26 Node:dell26@dell count:253
386 Name:dell10 Node:dell10@dell count:269
387 Name:netlab5 Node:netlab5@netlab count:205
388 Name:netlab6 Node:netlab6@netlab count:447
389 Name:dell11 Node:dell11@dell count:243
390 Name:netlaba22 Node:netlaba22@netlaba count:172
391 Name:dell37 Node:dell37@dell count:163
392 Name:netlab24 Node:netlab24@netlab count:194
393 Name:netlaba30 Node:netlaba30@netlaba count:336
394 Name:netlaba8 Node:netlaba8@netlaba count:242
395 Name:netlaba18 Node:netlaba18@netlaba count:357
396 Name:dell27 Node:dell27@dell count:534
397 Name:netlab34 Node:netlab34@netlab count:304
398 Name:netlaba32 Node:netlaba32@netlaba count:341
399 Name:netlaba24 Node:netlaba24@netlaba count:250
400 Name:netlaba19 Node:netlaba19@netlaba count:324
401 Name:netlaba12 Node:netlaba12@netlaba count:273
402 Name:netlab35 Node:netlab35@netlab count:171
403 Name:netlab36 Node:netlab36@netlab count:145
404 Name:dell30 Node:dell30@dell count:347
405 Name:netlab25 Node:netlab25@netlab count:206
406 Name:netlaba33 Node:netlaba33@netlaba count:236
407 Name:netlab32 Node:netlab32@netlab count:290
408 Name:dell8 Node:dell8@dell count:153
409 Name:netlab21 Node:netlab21@netlab count:277

```

```

410 Name:netlaba14 Node:netlaba14@netlaba count:283
411 Name:netlaba37 Node:netlaba37@netlaba count:320
412 Name:dell119 Node:dell119@dell count:169
413 Name:netlabbb40 Node:netlabbb40@netlabbb count:143
414 Name:netlabbb23 Node:netlabbb23@netlabbb count:261
415 Name:netlaba7 Node:netlaba7@netlaba count:472
416 Name:netlabbb16 Node:netlabbb16@netlabbb count:155
417 Name:netlaba31 Node:netlaba31@netlaba count:220
418 Name:dell24 Node:dell24@dell count:306
419 Name:netlaba6 Node:netlaba6@netlaba count:193
420 Name:netlaba26 Node:netlaba26@netlaba count:121
421 Name:netlabbb39 Node:netlabbb39@netlabbb count:178
422 Name:netlaba23 Node:netlaba23@netlaba count:356
423 Name:netlabbb27 Node:netlabbb27@netlabbb count:342
424 Name:netlabbb18 Node:netlabbb18@netlabbb count:200
425 Name:dell117 Node:dell117@dell count:223
426 Name:netlab426 Node:netlab426@netlab4 count:249
427 Name:netlab433 Node:netlab433@netlab4 count:373
428 Name:netlab435 Node:netlab435@netlab4 count:188
429 Name:netlab414 Node:netlab414@netlab4 count:118
430 Name:netlab423 Node:netlab423@netlab4 count:139
431 Name:netlab410 Node:netlab410@netlab4 count:139
432 Name:netlab416 Node:netlab416@netlab4 count:280
433 Name:netlab417 Node:netlab417@netlab4 count:294
434 Name:netlab44 Node:netlab44@netlab4 count:454
435 Name:netlab431 Node:netlab431@netlab4 count:117
436 Name:netlab412 Node:netlab412@netlab4 count:412
437 Name:netlab440 Node:netlab440@netlab4 count:421
438 Name:netlab434 Node:netlab434@netlab4 count:194
439 Name:netlab411 Node:netlab411@netlab4 count:320
440 Name:netlab43 Node:netlab43@netlab4 count:195
441 Name:netlab432 Node:netlab432@netlab4 count:392
442 Name:netlab438 Node:netlab438@netlab4 count:142
443 Name:netlab429 Node:netlab429@netlab4 count:235
444 Name:netlab430 Node:netlab430@netlab4 count:153
445 Name:netlab436 Node:netlab436@netlab4 count:282
446 Name:netlab419 Node:netlab419@netlab4 count:245
447 Name:netlab47 Node:netlab47@netlab4 count:221
448 Name:netlab439 Node:netlab439@netlab4 count:223
449 Name:netlab427 Node:netlab427@netlab4 count:152
450 Name:netlab437 Node:netlab437@netlab4 count:156
451 Name:netlab415 Node:netlab415@netlab4 count:189
452 Name:netlab425 Node:netlab425@netlab4 count:386
453 Name:netlab45 Node:netlab45@netlab4 count:128
454 Name:netlab428 Node:netlab428@netlab4 count:196
455 Name:netlab424 Node:netlab424@netlab4 count:271
456 Name:netlab418 Node:netlab418@netlab4 count:272
457 Name:netlab422 Node:netlab422@netlab4 count:297
458 Name:netlab48 Node:netlab48@netlab4 count:217
459 Name:netlab420 Node:netlab420@netlab4 count:221
460 Name:netlab421 Node:netlab421@netlab4 count:213
461 Name:netlab49 Node:netlab49@netlab4 count:179
462 Name:netlab41 Node:netlab41@netlab4 count:254
463 Name:netlab46 Node:netlab46@netlab4 count:245
464 Name:netlab42 Node:netlab42@netlab4 count:106
465 Name:netlab323 Node:netlab323@netlab3 count:199
466 Name:xxx Node:xxx_node@netlab3 count:364
467 Name:netlab311 Node:netlab311@netlab3 count:482
468 Name:netlab316 Node:netlab316@netlab3 count:297
469 Name:netlab332 Node:netlab332@netlab3 count:207

```

```
470 Name:netlab330 Node:netlab330@netlab3 count:375
471 Name:netlab339 Node:netlab339@netlab3 count:140
472 Name:netlab334 Node:netlab334@netlab3 count:120
473 Name:netlab336 Node:netlab336@netlab3 count:200
474 Name:netlab337 Node:netlab337@netlab3 count:137
475 Name:netlab331 Node:netlab331@netlab3 count:138
476 Name:netlab328 Node:netlab328@netlab3 count:194
477 Name:netlab318 Node:netlab318@netlab3 count:228
478 Name:netlab335 Node:netlab335@netlab3 count:241
479 Name:netlab327 Node:netlab327@netlab3 count:238
480 Name:netlab340 Node:netlab340@netlab3 count:226
481 Name:netlab35 Node:netlab35@netlab3 count:217
482 Name:netlab325 Node:netlab325@netlab3 count:207
483 Name:netlab315 Node:netlab315@netlab3 count:261
484 Name:netlab322 Node:netlab322@netlab3 count:284
485 Name:netlab329 Node:netlab329@netlab3 count:261
486 Name:netlab321 Node:netlab321@netlab3 count:134
487 Name:netlab324 Node:netlab324@netlab3 count:305
488 Name:netlab338 Node:netlab338@netlab3 count:138
489 Name:netlab313 Node:netlab313@netlab3 count:199
490 Name:netlab333 Node:netlab333@netlab3 count:310
491 Name:netlab37 Node:netlab37@netlab3 count:214
492 Name:netlab319 Node:netlab319@netlab3 count:215
493 Name:netlab36 Node:netlab36@netlab3 count:146
494 Name:netlab326 Node:netlab326@netlab3 count:129
495 Name:netlab312 Node:netlab312@netlab3 count:246
496 Name:netlab320 Node:netlab320@netlab3 count:184
497 Name:netlab317 Node:netlab317@netlab3 count:278
498 Name:netlab314 Node:netlab314@netlab3 count:274
499 Name:netlab38 Node:netlab38@netlab3 count:222
500 Name:netlab34 Node:netlab34@netlab3 count:419
501 Name:netlab310 Node:netlab310@netlab3 count:113
502 Name:netlab39 Node:netlab39@netlab3 count:253
503 Name:netlab32 Node:netlab32@netlab3 count:304
504 Name:netlab33 Node:netlab33@netlab3 count:318
505 Name:netlab31 Node:netlab31@netlab3 count:146
506 ok
507 (xxx_node@netlab3)4>
```

H Plot Data

H.1 Comparison with Different Number of Nodes

H.1.1 Plot Data

List 55: Test Results:Comparison with different number of nodes:plot data

1	dummy	10nodes	17nodes	33nodes	65nodes	105nodes
2	store	5.58	4.44	4.97	6	6.61
3	get	1.76	1.97	2.2	3.04	3.31
4	rstore	9.81	10.67	12.58	13.26	13.7
5	rget	12.57	12.02	9.87	10.57	7.46

H.1.2 Gnuplot Script

List 56: Test Results:Comparison with different number of nodes:gnuplot script

```

1 #set terminal jpeg medium
2 #set output "graph1.jpeg"
3 set terminal post eps
4 set output "graph1.eps"
5 set auto x
6 set grid y
7 set boxwidth 0.75 absolute
8 set style fill solid 1.00 border -1
9 set style data histogram
10 set style histogram cluster gap 1
11 set xtic rotate by -45
12 set xtics border in scale 1,0.5 nomirror rotate by -45 offset character 0, 0, 0
13 set xtics norangelimit
14 set yrange [0:20]
15 set ylabel "Time(msec)"
16 set xlabel "latency of 1 operation against 10000 data"
17 set title "Average latency of each operation under different # of nodes"
18 set style fill pattern border
19
20 plot 'graph1.dat' using 2:xtic(1) title col, '' u 3 ti col, '' u 4 ti col, '' u 5 ti col, '' \
    → u 6 ti col

```

H.2 Comparison with Different Number of Threads

H.2.1 Plot Data

List 57: Test Results:Comparison with different number of threads:plot data

1	Dummy	rget	get
2	1thread	52.55	20
3	2threads	39.75	10.83
4	4threads	33.87	9.41
5	8threads	30.31	5.29
6	33threads	33.08	2.82

H.2.2 Gnuplot Script

List 58: Test Results:Comparison with different number of threads:gnuplot script

```

1 #set terminal jpeg medium
2 #set output "graph2.jpeg"
3 set terminal post eps
4 set output "graph2.eps"
5 set auto x
6 set boxwidth 0.75 absolute
7 set style fill solid 1.00 border -1
8 set style data histogram
9 set style histogram cluster gap 1
10 set xtic rotate by -45
11 set xtics border in scale 1,0.5 nomirror rotate by -45 offset character 0, 0, 0
12 set xtics norangelimit
13 set yrange [0:60]
14 set ylabel "Time(sec)"
15 set xlabel "# of threads to search 1 row in 10000 from 1 to 10000"
16 set title "The get and rget functions compared under different # of threads for 10000 \
    →queries"
17 set style fill pattern border
18
19 plot 'graph2.dat' using 2:xtic(1) title col, '' u 3 ti col

```

H.3 Comparison with Different Range Query Conditions

H.3.1 Plot Data

List 59: Test Results:Comparison with different query conditions:plot data

1	1/1000	2.34	0.0036
2	1-10/1000	2.32	0.00575
3	1-100/1000	2.36	0.259
4	1-500/1000	2.31	0.118
5	1-1000/1000	2.388	0.234

H.3.2 Gnuplot Script

List 60: Test Results:Comparison with different query conditions:gnuplot script

```

1 #set terminal jpeg medium
2 #set output "graph3.jpeg"
3 set terminal post eps
4 set output "graph3.eps"
5 set auto x
6 set boxwidth 0.75 absolute
7 set style fill solid 1.00 border -1
8 set style data histogram
9 set style histogram cluster gap 1
10 set xtic rotate by -45
11 set xtics border in scale 1,0.5 nomirror rotate by -45 offset character 0, 0, 0
12 set xtics norangelimit

```

```

13 set xtics    ("1/1000" 1.00000 -1, "10/1000" 2.00000 -1, "100/1000" 3.00000 -1, "500/1000" \
    →4.00000 -1, "1000/1000" 5.00000 -1)
14 set logscale y
15 set yrange [0.001:10]
16 set ylabel "Time(sec)"
17 set xlabel "Specified Range in 1000 Data"
18 set title "Simple get and range query method comparison with 1000 data"
19 set style fill pattern border
20
21 plot 'graph3.dat' using 2 t "simple get", '' using 3 t "range query"

```

H.4 Memory Usage with Theory and Real Case

H.4.1 Plot Data

List 61: Test Results:Memory Usage of Theory:plot data

1	1.7	2.4	1.9	3.3
---	-----	-----	-----	-----

List 62: Test Results:Memory Usage of Real:plot data

1	13.695	18.413	16.786	26.316
---	--------	--------	--------	--------

H.4.2 Gnuplot Script

List 63: Test Results:Memory Usage with theory and real case:gnuplot script

```

1  #set terminal jpeg medium
2  #set output "graph3.jpeg"
3  set terminal post eps
4  set output "graph4.eps"
5  set multiplot layout 1,2 title "Memory Usage Comparison Between Theoretical and Real with \
    →10000 data"
6  set xtics rotate
7  set bmargin 7
8
9
10
11
12 set auto x
13 set grid y
14 set boxwidth 0.5 absolute
15 set style fill solid 1.00 border -1
16 set style data histogram
17 set style histogram cluster gap 5
18 set xtic rotate by -45
19 unset xtics
20 set xrange[-0.25:0.25]
21 set yrange[0:5]
22 set ylabel "Mbytes"
23 set xlabel "methods and schemas"
24 set title "Theoretical value\n"
25 set style fill pattern border

```

```

26 set key bottom below
27 plot 'graph4-2.dat' using 1 title "simple store(int, char(30))", '' u 2 ti "simple store(int\
→, {int, char(10), char(30)})", '' u 3 ti "rstore(int(key), char(30))", '' u 4 ti "rstore(\
→int(key), int(key), char(key, 10), char(30))"
28
29 set auto x
30 set grid y
31 set boxwidth 0.5 absolute
32 set style fill solid 1.00 border -1
33 set style data histogram
34 set style histogram cluster gap 5
35 set xtic rotate by -45
36 unset xtics
37 set xrange[-0.25:0.25]
38 set yrange[0:35]
39 set ylabel "Mbytes"
40 set xlabel "methods and schemas"
41 set title "Real case with 64 nodes\n"
42 set style fill pattern border
43
44 set nokey
45 plot 'graph4-1.dat' using 1 title "simple store(int, char(30))", '' u 2 ti "simple store(int\
→, {int, char(10), char(30)})", '' u 3 ti "rstore(int(key), char(30))", '' u 4 ti "rstore(\
→int(key), int(key), char(key, 10), char(30))"
46
47 unset multiplot

```

H.5 Load Balancing

H.5.1 Plot Data

List 64: Test Results:Load Balancing:plot data

1	#	# of nodes/machine,	SD(Machine),	AVG(Node),	SD(Node),	SD/AVG(%)	Min	Max	AVG
2	5	727.84 670.24 400	7.27	824 3213 1923	1				
3	10	857.80 366.16 200	8.57	228 1880 980.39	2				
4	20	1266.3 274.77 100	12.66	164 1466 495.05	3				
5	40	610.16 95.226 50	6.1	66 534 248.75	4				

H.5.2 Gnuplot Script

List 65: Test Results:Load Balancing:gnuplot script

```

1 #set terminal jpeg medium
2 #set output "graph5.jpeg"
3 set term post eps
4 set output "graph5.eps"
5 set xtic rotate by -45
6 set xtics border in scale 1,0.5 nomirror rotate by -45 offset character 0, 0, 0
7 set xtics ("5" 1.00000, "10" 2.00000, "20" 3.00000, "40" 4.00000)
8 set xtics norangelimit
9 set xrange [0:5]
10 set yrange [1:4000]

```

```
11 set ytics nomirror
12 set ylabel "# of data on virtual nodes"
13 set xlabel "# of nodes per machine"
14 set title "Load balancing on different # of nodes for 10000 data using 5 machines"
15 set style fill pattern border
16
17 plot 'graph5.dat' using 9:8:6:7 title "Min/Avg/Max"with errorbars 4
```
