CellCycle Appendix

Alessandro Fazio, Andrea Gennusa

A. Memory Benchmark

a. Tests:

Test id	Operations	get/set ratio	values(Byte)	Allocated ram(MegaByte)	Slab Size (KiloByte)	Slabs purged
1	100.000	5	300	10	100	0
2	500.000	5	300	10	100	144
3	1.000.000	5	300	10	100	367
4	100.000	5	300	0.1	10	432
5	500.000	5	300	0.1	10	2173
6	1.000.000	5	300	0.1	10	4321
7	5.000.000	5	300	0.1	10	21674
8	100.000	5	900	1	10	1258
9	500.000	5	900	1	10	6492
10	1.000.000	5	900	1	10	13001
11	5.000.000	5	900	1	10	64971
12	100.000	5	900	10	10	507
13	500.000	5	900	10	10	5964
14	1.000.000	5	900	10	10	12597
15	5.000.000	5	900	10	10	64808
16	100.000	1	300	10	100	49
17	500.000	1	300	100	100	521
18	1.000.000	1	300	100	100	1055
19	5.000.000	1	300	100	100	5140
20	100.000	5	300	100	1.000	0
21	500.000	5	300	1.000	1.000	0
22	1.000.000	5	300	1.000	1.000	0
23	5.000.000	5	300	1.000	1.000	0

b. Get Operations:

Test id	Cumulative get for lld (s)	Cumulative get for lla_nu (s)	Cumulative get for lla_u (s)	Cumulative get for st_a (s)	Cumulative get for st_lla (s)
1	1.6	1.5	1.6	1.6	1.7
2	5.8	5.6	6.3	6.4	6.4
3	9.6	9.2	9.7	10.7	11.1
4	0.5	0.4	0.5	0.4	0.4
5	2.2	2.2	2.2	2.0	2.1
6	4.8	4.4	4.5	4.2	4.7
7	23.2	21.8	21.7	20.0	22.3
8	1.1	1.0	1.1	1.2	1.3
9	3.2	3.2	3.2	3.6	3.7
10	5.6	5.3	5.5	6.2	6.4
11	24.1	22.1	23.8	26.6	29.1
12	3.2	3.1	3.4	3.6	3.9
13	7.6	7.5	8.0	9.2	9.9
14	11.4	10.7	11.6	14.0	14.6
15	32.2	30.3	32.6	41.8	45.6
16	1.7	1.6	1.7	1.8	1.9
17	4.8	4.5	4.8	4.8	5.2
18	7.3	6.8	7.7	7.5	8.2
19	27.9	23.8	26.7	27.6	29.4
20	NA	NA	1.4	NA	1.4
21	NA	NA	7.7	NA	8.0
22	NA	NA	14.9	NA	16.1
23	NA	NA	76.1	NA	86.8

c. Set Operations:

Test id	Cumulative set for lld (s)	Cumulative set for lla_nu (s)	Cumulative set for lla_u (s)	Cumulative set for st_a (s)	Cumulative set for st_lla (s)
1	1.0	0.9	0.9	0.9	1.0
2	6.9	6.3	4.0	4.5	4.3
3	14.5	12.8	7.5	9.1	9.2
4	0.7	0.6	0.7	0.7	0.8
5	3.7	3.7	3.5	3.8	4.1
6	7.7	7.3	7.0	7.7	9.3
7	38.2	38.1	35.3	37.9	41.3
8	2.6	2.5	1.9	2.3	2.5
9	11.9	12.0	9.3	10.4	11.3
10	24.5	22.7	18.4	21.5	22.4
11	116.9	107.6	91.5	101.9	115.0
12	7.7	6.3	2.5	3.4	3.6
13	44.6	37.2	12.2	22.8	23.4
14	88.3	75.6	24.0	48.2	48.3
15	445.4	384.0	117.9	237.7	247.3
16	4.1	3.7	2.4	2.9	2.9
17	19.0	16.9	10.0	12.0	12.0
18	35.1	31.2	18.7	21.5	21.7
19	169.6	146.7	84.5	104.3	105.4
20	NA	NA	0.8	NA	0.9
21	NA	NA	4.1	NA	4.6
22	NA	NA	8.1	NA	9.1
23	NA	NA	40.2	NA	46.5

d. Memory Load Tests:

Test id	Object Pool (MB)	Actual preallocated RAM (MB)	Allocated RAM after workload (MB)
1	10	10.0	1.0
2	10	10.0	4.9
3	10	10.0	8.9
4	0.1	0.1	1.2
5	0.1	0.1	4.0
6	0.1	0.1	6.8
7	0.1	0.1	32.2
8	1	1.4	1.1
9	1	1.4	4.4
10	1	1.4	6.7
11	1	1.4	31.9
12	10	10.1	1.1
13	10	10.1	5.3
14	10	10.1	8.7
15	10	10.1	33.9
16	100	10.1	1.6
17	100	100.2	7.6
18	100	100.2	13.7
19	100	100.2	41.9
20	1.000	954.9	1.2
21	1.000	954.9	5.4
22	1.000	954.9	9.6
23	1.000	954.9	46.4

B. Settings Parameters

Field	Example	Description
LogFile	log.txt	The relative path of the file where the system writes log
Verbose	True	Specify if the system writes logs only on LogFile (False) or in console and LogFile (True)
PreallocatedPool	1000000000	Specify the ram available to store values in Bytes. It must be less than available ram on system where application is executed. You must consider a light overhead due to data to maintain values in memory, it's near to be 10 MB for each million of values. There's no a too large value, it was tested at max at 31 GB on a 32 GB ram machine.
SlabSize	1000000	Specify the dimension of each slab in Bytes, that represents the maximum size of a single value.
GetterThreadNumber	2	Specify the number of threads reserved for get operations from memory. 1 is min
MasterSetPort	5550	Specify TCP port for set operations of master memory
MasterGetPort	5551	Specify TCP port for get operations of master memory
SlaveSetPort	5552	Specify TCP port for set operations of slave memory
SlaveGetPort	5553	Specify TCP port for get operations of slave memory
InternalChannelPort	5557	Specify TCP port for internal channel operations
ExternalChannelPort	5558	Specify TCP port for external channel operations
MemoryObjectPort	5559	Specify TCP port for memory transfer operations
ServiceThreadNumber	4	Specify the number of threads reserved to entrypoint memcached like interfaces

ClientEntrypointPort	5555	Specify the TCP port for memcached like interface
ScalePeriod	60	Number of seconds of usage metricator period. Interval of time between checks
GetScaleUpLevel	0.5	Value between 0,1 to trigger scale up from get threads usage percent. When Get threads usage level reaches that value, system will requests a scale up. 0 means ignore value
GetScaleDownLevel	0.001	Value between 0,1 to trigger scale down from get threads usage percent. When Get threads usage level reaches that value, system will requests a scale down. 0 means ignore value
SetScaleUpLevel	0.5	Value between 0,1 to trigger scale up from Set thread usage percent. When Set thread usage level reaches that value, system will requests a scale up. 0 means ignore value
SetScaleDownLevel	0.001	Value between 0,1 to trigger scale down from set thread usage percent. When set thread usage level reaches that value, system will requests a scale down. 0 means ignore value
AwsImageId	ami-9707fcf8	This string identifies the AWS ami id to invoke launching a new instance of the application.
AwsSecurityGroup	SSHToAll	This value identifies the security group applied to new AWS EC2 instances. Security group must permit telnet and ssh incoming communication.
AwsKeyName	CellCycleBot	Name of key used for ssh communication for new AWS EC2 instances.
GitBranch	master	New Instances, booting, pull from git the current source of the application. Updating AMI image is expensive, in this way we can mantain an old version of AMI image and update automatically the code. This value identifies the branch where you are working on.
StartFileRelativePath	startOnBoot.py	This is the relative path (from CellsCycle root folder) to file that is invoked on boot on new EC2 instances.
AwsProfileName	default	aws profile name in ~/.aws folder

MaxInstance	12	max aws instances
MinInstance	5	min aws instances

C. ExtraCycle Commands

Memcached operation (Memcached, Redis compatible):
-SET (SET <key> <flag> <exp> <byte> <data>) - store a value
-ADD (ADD <key> <flag> <exp> <byte> <data>) - store a new value
-GET (GET <key>) - get a value from a key
-DELETE (DELETE <key>) - delete a value

Manage operations:
-CELLCYCLE
KILLYOURSELF <TERMINATE or STOP> - stop or terminate the current virtual machine
KILLALL<TERMINATE or STOP> - stop or terminate all virtual machines of AWSprofilename
NEWCELL <params> - (debug) manually starts a virtual machines with custom params
SCALEUP - manually asks for a scaleup
SCALEDOWN - manually asks for a scaledown
KEYS - returns the keys list of nodes
WHOHAS <key> - returns the node that manage that key
LOG - returns the logFile of the current instance

D. Write, Crash and Read Use Case Test

```
ubuntu@ip-172-31-21-1:~$ telnet 172.31.20.1 5555
Trying 172.31.20.1...
Connected to 172.31.20.1.
Escape character is '^]'.
help
ERROR
SUPPORTED OPERATIONS:
-SET (SET <key> <flag> <exp> <byte> <data>)
-ADD (ADD <key> <flag> <exp> <byte> <data>)
-GET (GET <key>)
-DELETE (DELETE <key>)
-CELLCYCLE
    KILLYOURSELF <TERMINATE or STOP>
    KILLALL <TERMINATE or STOP>
    NEWCELL <params>
    SCALEUP
    SCALEDOWN
    KEYS
    WHOHAS <key>
```

```
     LOG

BYE
cellcycle keys
Node 1, Node : myself 1, master 5, slave 2
Keys : master 3435973836:4294967294, myself 0:858993458, slave
858993459:1717986917
Node 3, Node : myself 3, master 2, slave 4
Keys : master 858993459:1717986917, myself 1717986918:2576980376, slave
2576980377:3435973835
Node 2, Node : myself 2, master 1, slave 3
Keys : master 0:858993458, myself 858993459:1717986917, slave
1717986918:2576980376
Node 5, Node : myself 5, master 4, slave 1
Keys : master 2576980377:3435973835, myself 3435973836:4294967294, slave
0:858993458
Node 4, Node : myself 4, master 3, slave 5
Keys : master 1717986918:2576980376, myself 2576980377:3435973835, slave
3435973836:4294967294

cellcycle whohas ciao
Key 3996799345 is assigned to: 172.31.20.5
set ciao 1 1 4 ciaone
STORED
get ciao
VALUE 3996799345 1 6
ciaone
END
quit
Connection closed by foreign host.
ubuntu@ip-172-31-21-1:~$ telnet 172.31.20.5 5555
Trying 172.31.20.5...
Connected to 172.31.20.5.
Escape character is '^]'.
cellcycle killyourself terminate
HELLO DARKNESS MY OLD FRIEND...
Connection closed by foreign host.
ubuntu@ip-172-31-21-1:~$ telnet 172.31.20.1 5555
Trying 172.31.20.1...
Connected to 172.31.20.1.
Escape character is '^]'.
get ciao
VALUE 3996799345 1 6
ciaone
END
```

E. Memaslap Benchmark

a. Ratio 9-1 Get-Set

```
22:37:01 start 2 x memaslap
22:37:59 new node 4.5
22:38:49 new node 3.5
22:39:56 new node 7.0
22:40:55 new node 2.5
22:41:34 new node 2.25
22:42:36 new node 1.5
22:47:03 memaslap finish
22:47:30 recovered death of 7.0
22:47:58 recovered death of 4.5
22:48:32 recovered death of 1
22:49:16 recovered death of 5
22:50:02 recovered death of 4
22:50:54 recovered death of 2


cmd_get: 1926005
cmd_set: 214014
get_misses: 8525
written_bytes: 371.299.440
read_bytes: 2.029.972.872

Run time: 600.1s Ops: 2131802 TPS: 1782 Net_rate: 4.0M/s
```

b. Ratio 5-5 Get-Set

```
18:23:00 start 2 x memaslap
18:25:19 new node 2.5
18:25:56 new node 4.5
18:27:03 new node 7.0
18:27:57 new node 1.5
18:28:00 memaslap finish
18:28:37 recovered death of 5
18:29:36 recovered death of 4
18:30:13 recovered death of 2
18:31:08 recovered death of 7.0
18:32:36 new node 1.25
18:33:25 recovered death of 2.5



cmd_get: 278070
cmd_set: 269991
```

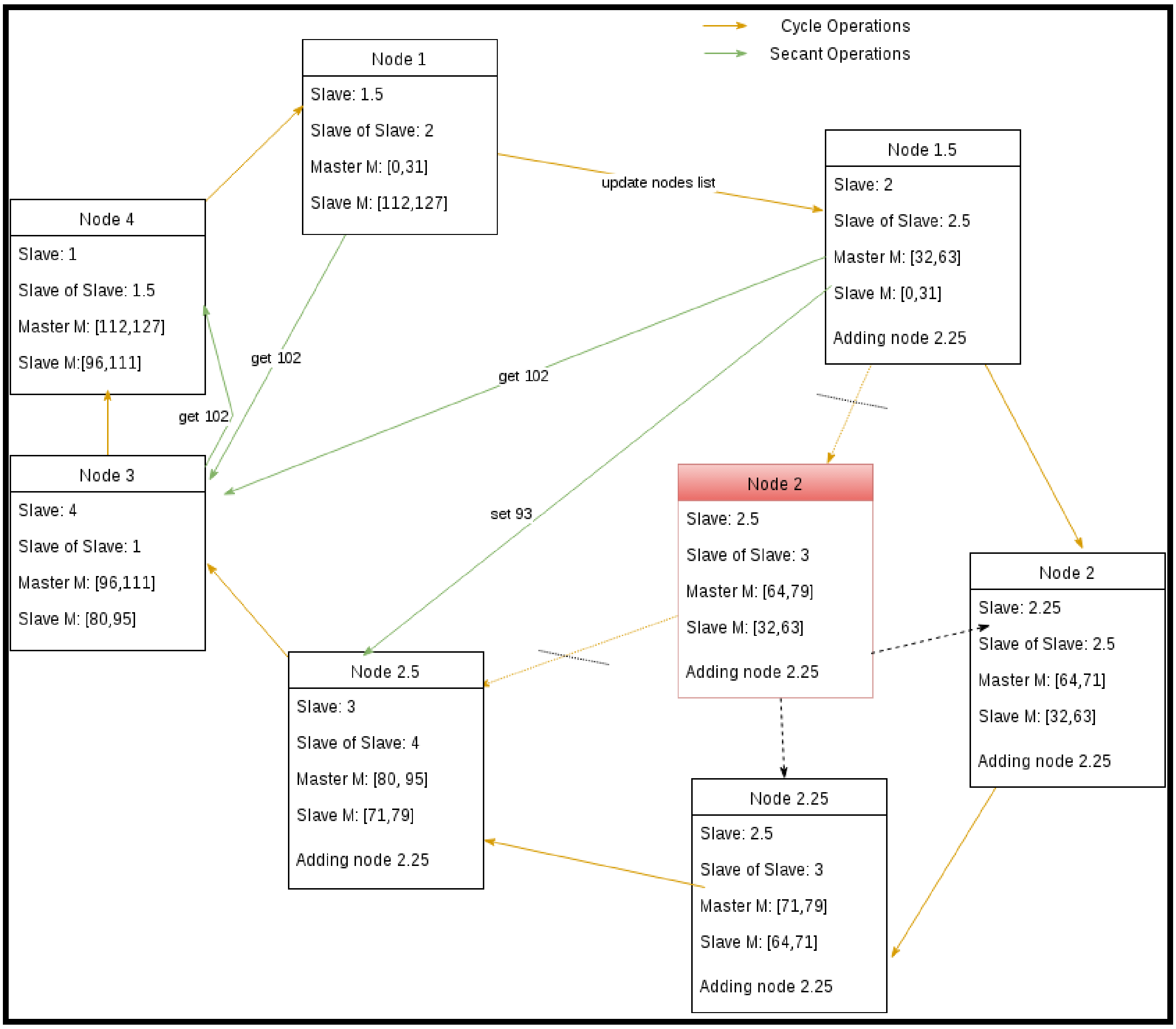
```
get_misses: 9291
written_bytes: 282.951.569
read_bytes: 282.372.663

Run time: 300.0s Ops: 931202 TPS: 1235 Net_rate: 3.7M/s
```

F. Test and Benchmark Configuration

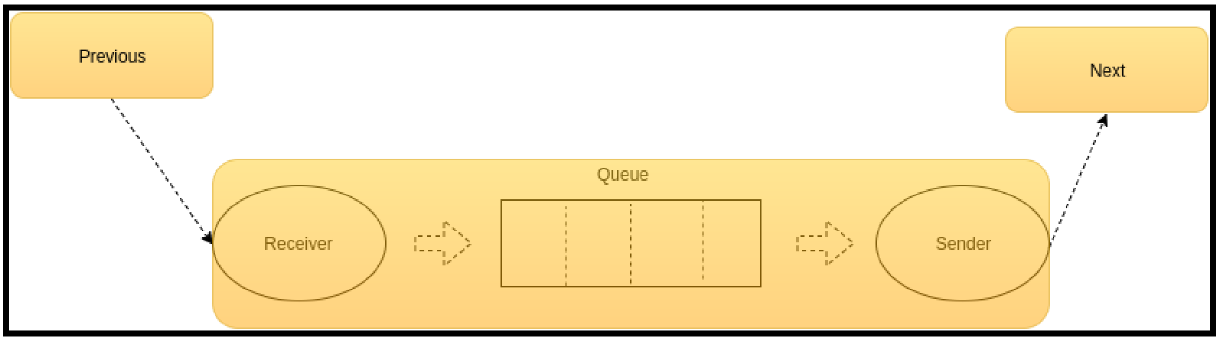
```
LogFile logFile.txt
Verbose False
PreallocatedPool  200000000
SlabSize 100000
ValueMaxSize 4096
GetterThreadNumber 1
MasterSetPort 5550
MasterGetPort 5551
SlaveSetPort 5552
SlaveGetPort 5553
ServiceThreadNumber 16
ClientEntrypointPort 5555
ScalePeriod 30
GetScaleUpLevel 0.05
GetScaleDownLevel 0.000001
SetScaleUpLevel 0.09
SetScaleDownLevel 0.0000001
AwsImageId ami-9fc501f0
AwsSecurityGroup SSHToAll
AwsKeyName AWSCellCycle
EC2Type t2.micro
GitBranch master
StartFileRelativePath startOnBoot.py
IntPort 5557
ExtPort 5558
MinInstance 5
MaxInstance 13
MemoryObjectPort 5559
AwsProfileName default
```

G. Pictures

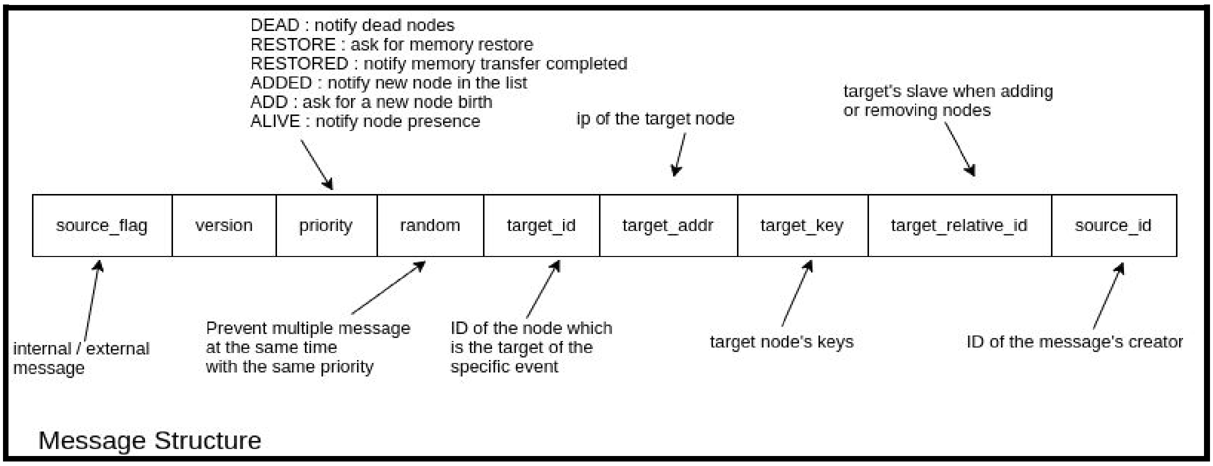
a. CellCycle example



b. Queue

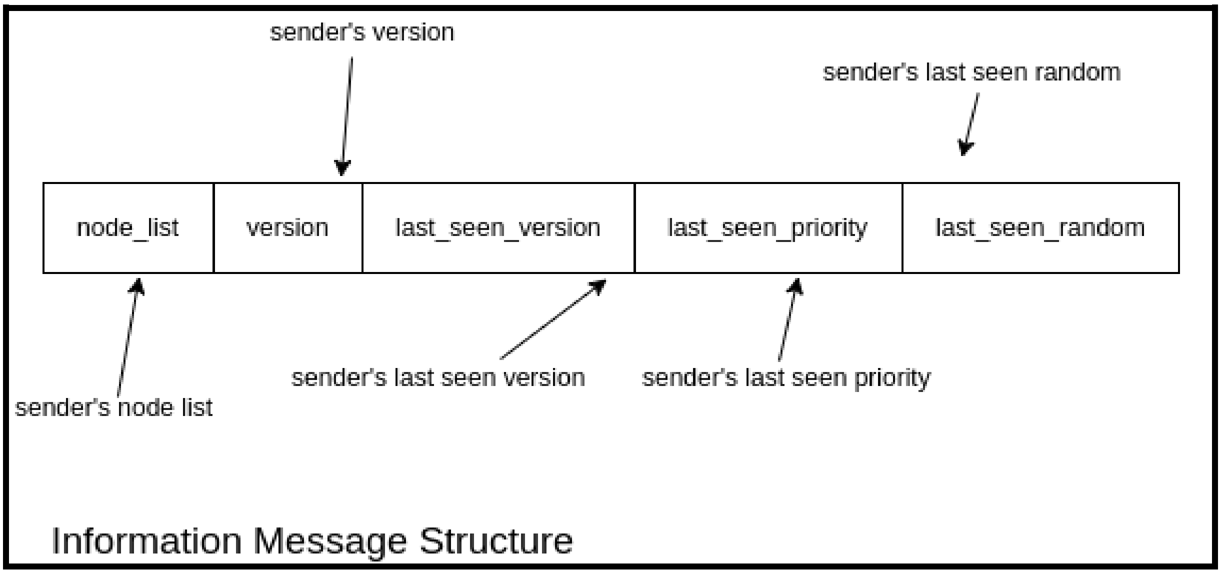


c. Message Structure


DEAD : notify dead nodes
RESTORE : ask for memory restore
RESTORED : notify memory transfer completed
ADDED : notify new node in the list
ADD : ask for a new node birth
ALIVE : notify node presence
ip of the target node
target's slave when adding or removing nodes
source_flag
version
priority
random
target_id
target_addr
target_key
target_relative_id
source_id
internal / external message
Prevent multiple message at the same time with the same priority
ID of the node which is the target of the specific event
target node's keys
ID of the message's creator

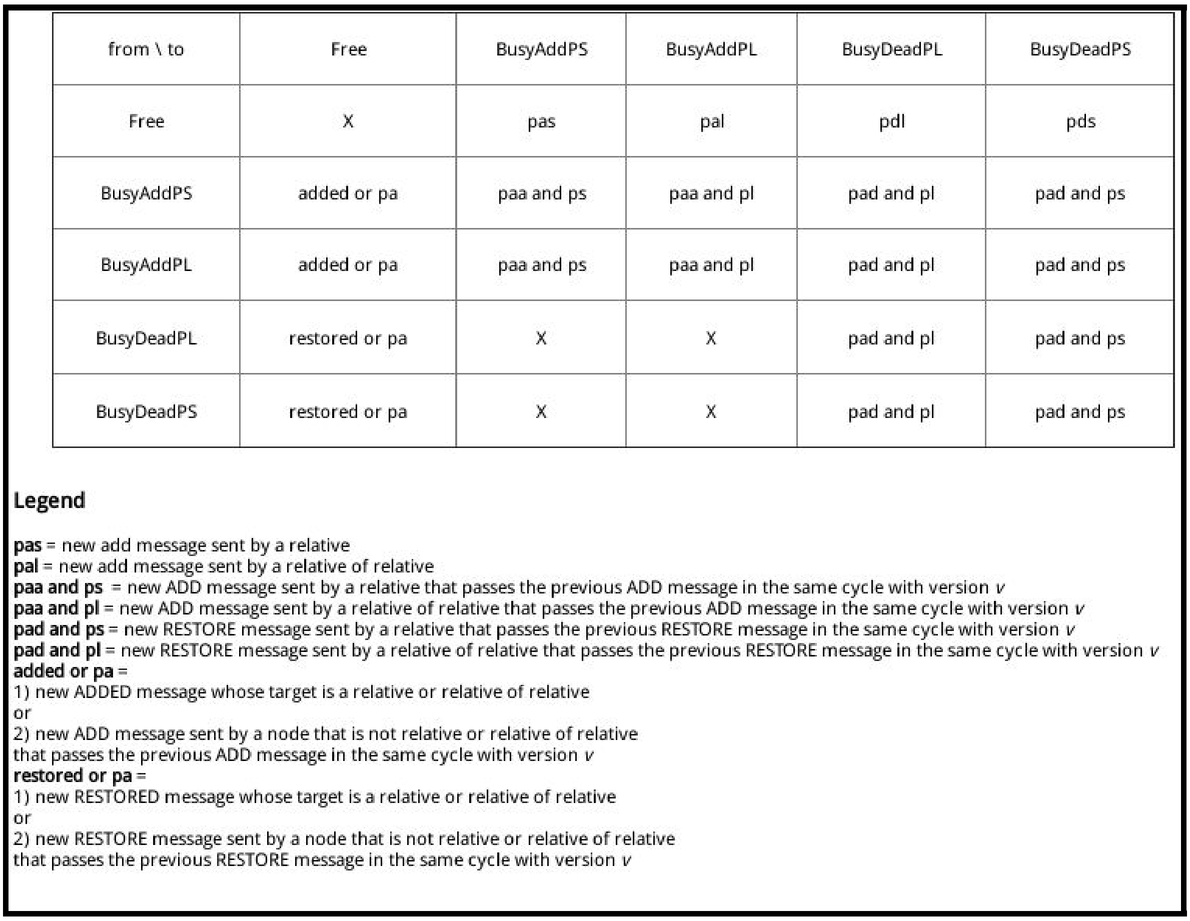
Message Structure

d. Information Message Structure



Information Message Structure

e. Transition Table

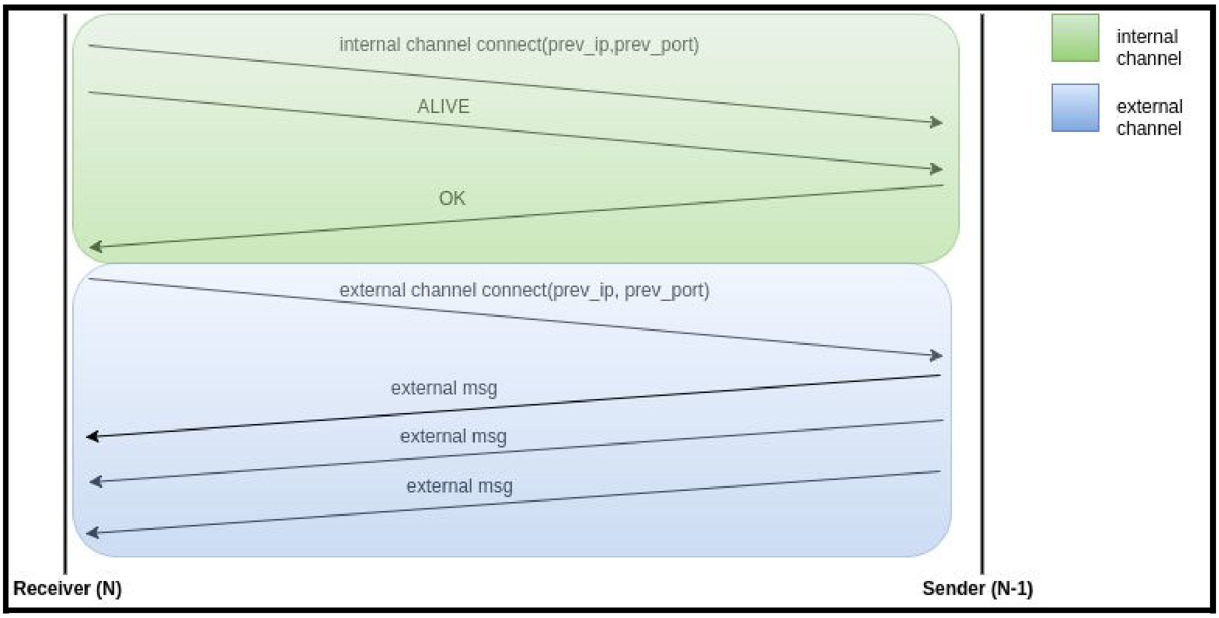


from \ to	Free	BusyAddPS	BusyAddPL	BusyDeadPL	BusyDeadPS
Free	X	pas	pal	pdl	pds
BusyAddPS	added or pa	paa and ps	paa and pl	pad and pl	pad and ps
BusyAddPL	added or pa	paa and ps	paa and pl	pad and pl	pad and ps
BusyDeadPL	restored or pa	X	X	pad and pl	pad and ps
BusyDeadPS	restored or pa	X	X	pad and pl	pad and ps

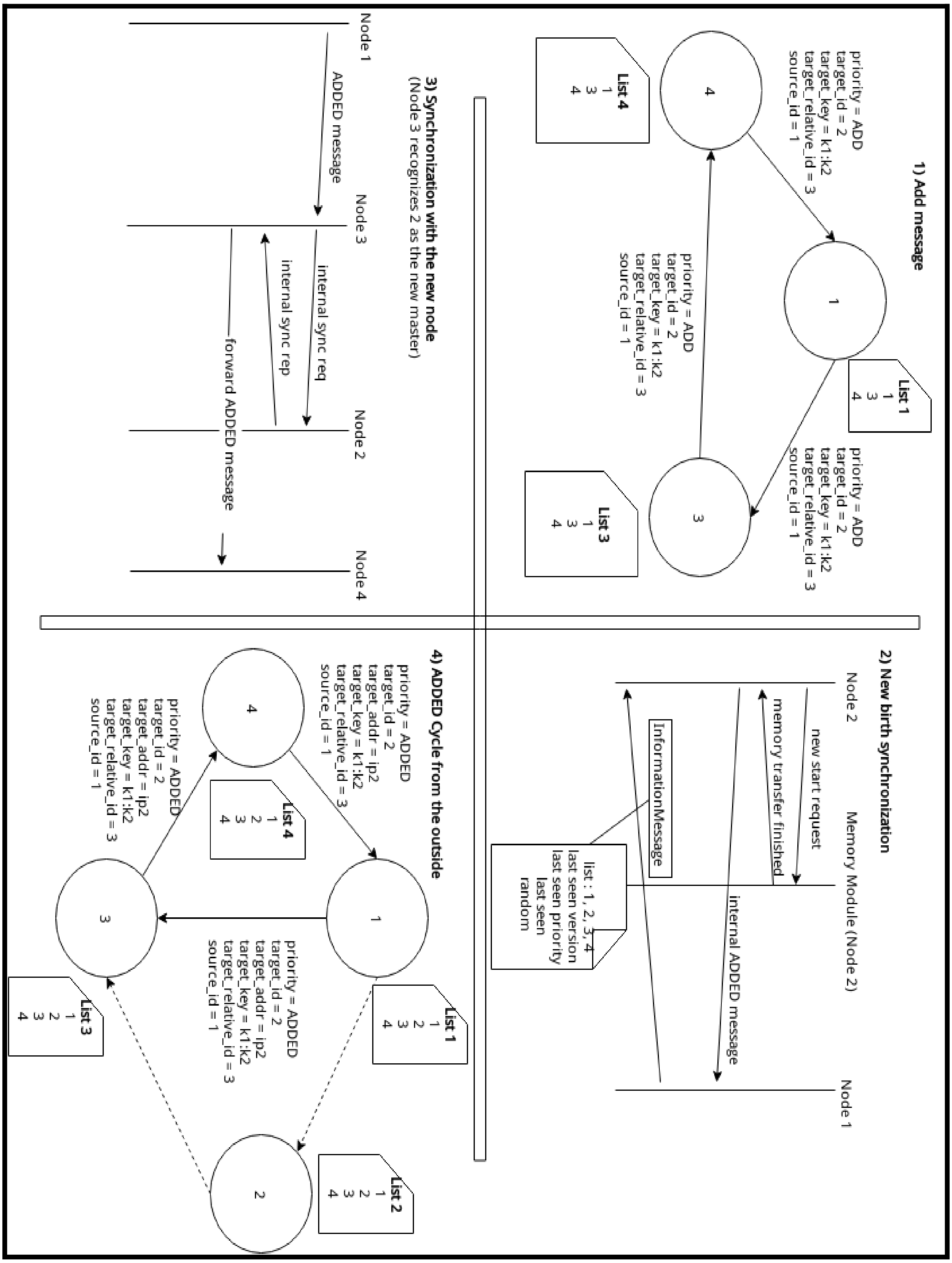
Legend

pas = new add message sent by a relative
pal = new add message sent by a relative of relative
paa and ps = new ADD message sent by a relative that passes the previous ADD message in the same cycle with version *v*
paa and pl = new ADD message sent by a relative of relative that passes the previous ADD message in the same cycle with version *v*
pad and ps = new RESTORE message sent by a relative that passes the previous RESTORE message in the same cycle with version *v*
pad and pl = new RESTORE message sent by a relative of relative that passes the previous RESTORE message in the same cycle with version *v*
added or pa =
1) new ADDED message whose target is a relative or relative of relative
or
2) new ADD message sent by a node that is not relative or relative of relative
that passes the previous ADD message in the same cycle with version *v*
restored or pa =
1) new RESTORED message whose target is a relative or relative of relative
or
2) new RESTORE message sent by a node that is not relative or relative of relative
that passes the previous RESTORE message in the same cycle with version *v*

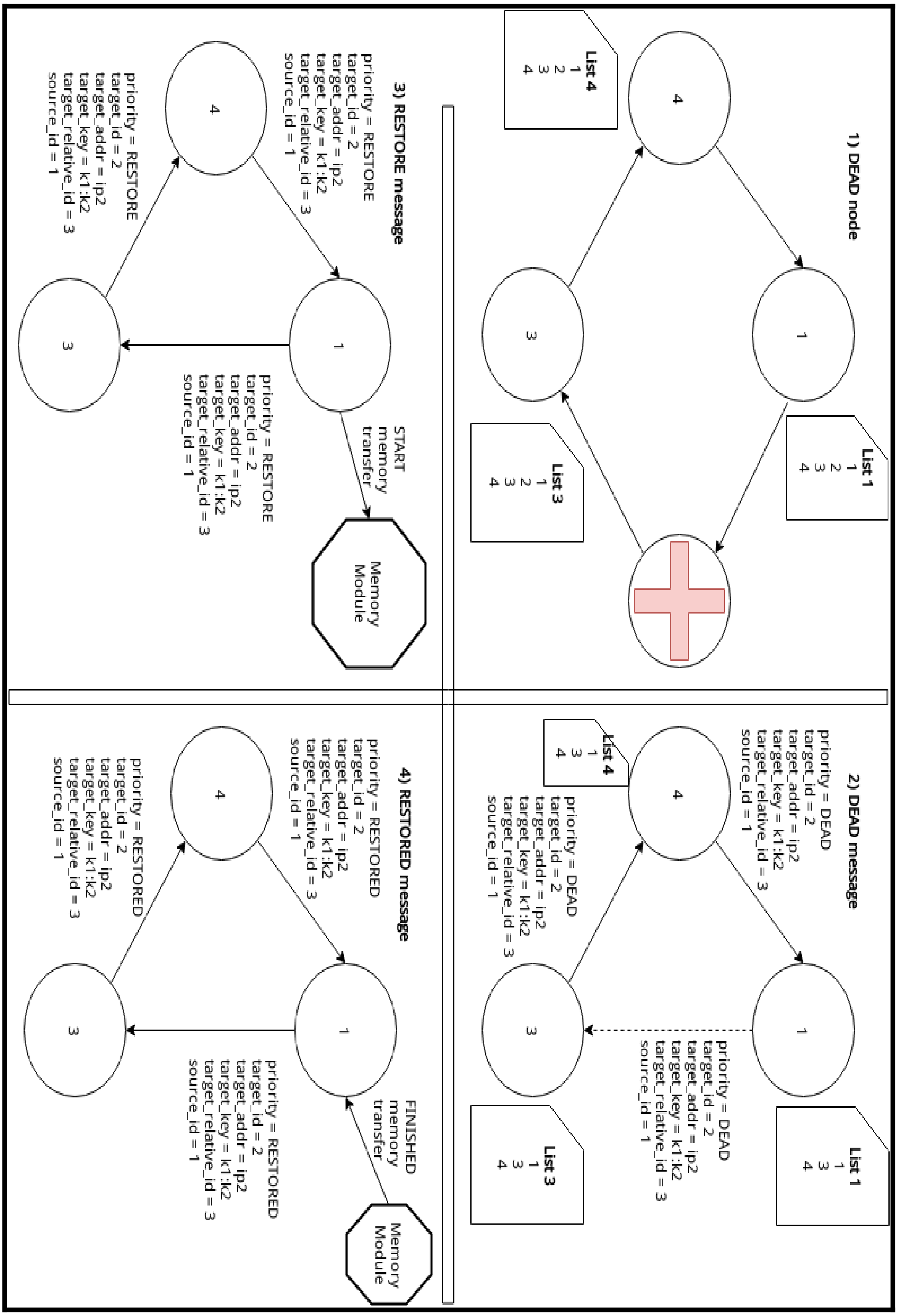
f. Receiver - Sender



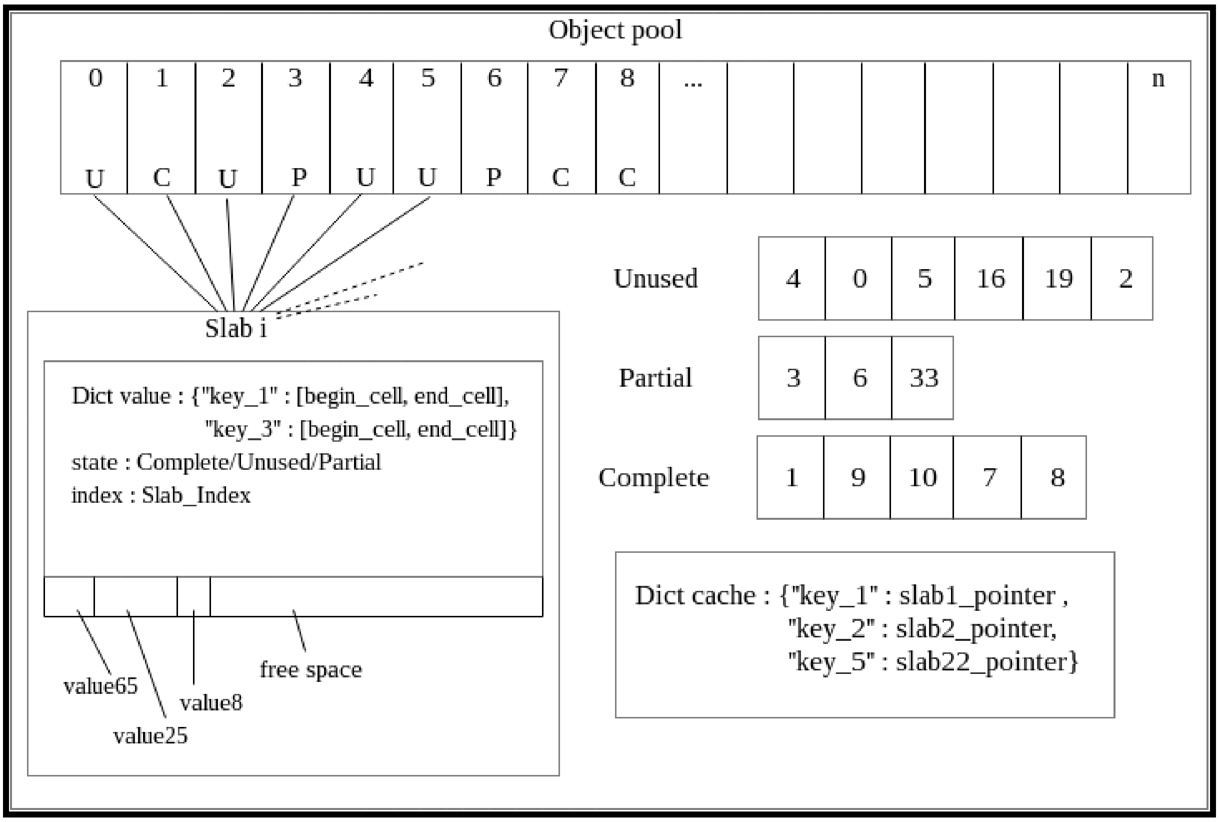
g. Scale Out


1) Add message
priority = ADD
target_id = 2
target_key = k1:k2
target_relative_id = 3
source_id = 1
4
List 4
1
3
4
1
List 1
1
3
4
priority = ADD
target_id = 2
target_key = k1:k2
target_relative_id = 3
source_id = 1
priority = ADD
target_id = 2
target_key = k1:k2
target_relative_id = 3
source_id = 1
3
List 3
1
3
4
2) New birth synchronization
Node 2
Memory Module (Node 2)
Node 1
new start request
memory transfer finished
InformationMessage
list : 1, 2, 3, 4
last seen version
last seen priority
last seen
random
internal ADDED message
3) Synchronization with the new node
(Node 3 recognizes 2 as the new master)
Node 1
Node 3
Node 2
Node 4
ADDED message
internal sync req
internal sync rep
forward ADDED message
4) ADDED Cycle from the outside
priority = ADDED
target_id = 2
target_addr = ip2
target_key = k1:k2
target_relative_id = 3
source_id = 1
4
List 4
1
2
3
4
1
List 1
1
2
3
4
priority = ADDED
target_id = 2
target_addr = ip2
target_key = k1:k2
target_relative_id = 3
source_id = 1
priority = ADDED
target_id = 2
target_addr = ip2
target_key = k1:k2
target_relative_id = 3
source_id = 1
3
List 3
1
2
3
4
2
List 2
1
2
3
4

h. Scale In


1) DEAD node
4
1
3
List 4
1
2
3
4
List 1
1
2
3
4
List 3
1
2
3
4
2) DEAD message
priority = DEAD
target_id = 2
target_addr = ip2
target_key = k1:k2
target_relative_id = 3
source_id = 1
List 4
1
3
4
List 1
1
3
4
List 3
1
3
4
3) RESTORE message
priority = RESTORE
target_id = 2
target_addr = ip2
target_key = k1:k2
target_relative_id = 3
source_id = 1
START
memory
transfer
Memory
Module
4) RESTORED message
priority = RESTORED
target_id = 2
target_addr = ip2
target_key = k1:k2
target_relative_id = 3
source_id = 1
FINISHED
memory
transfer
Memory
Module

i. Memory Module Our Slub Model


Object pool
0
1
2
3
4
5
6
7
8
...
n
U
C
U
P
U
U
P
C
C
Slab i
Dict value : {"key_1" : [begin_cell, end_cell],
"key_3" : [begin_cell, end_cell]}
state : Complete/Unused/Partial
index : Slab_Index
value65
value25
value8
free space
Unused
4
0
5
16
19
2
Partial
3
6
33
Complete
1
9
10
7
8
Dict cache : {"key_1" : slab1_pointer ,
"key_2" : slab2_pointer,
"key_5" : slab22_pointer}