

Distributed Transactional Database

HotDB Server -

[Standard]

Function Manual

Version No.: V2.5.6

Shanghai Hotpu Networks Technology Co., Ltd.

Nov. 2020

All rights reserved! Without the written consent of the company, no unit or individual is allowed to extract and copy the contents hereof in part or in whole, and disseminate them in any form.

Trademark statement

HotDB trademark of Hotpu is the registered trademark or trademark of our company or its affiliated company which is legally protected. Any infringement shall be investigated. Without the written permission of the company or the trademark owner, no unit and individual is allowed to use, copy, modify, disseminate, transcribe or bundling sell any part of this trademark in any form or for any reason. Any infringement of the trademark right of the company shall be investigated for legal liability.

Cautions

The products, services or features you purchase shall be subject to the commercial contracts and terms of Shanghai Hotpu Networks Technology Co., Ltd. The final interpretation right of the products, services or features in this document shall belong to Shanghai Hotpu Networks Technology Co., Ltd.

Shanghai Hotpu Networks Technology Co., Ltd.

Address: 603-605, Block A, SIM Technology building, No. 633, Jinzhong Road, Changning District, Shanghai

Postcode: 200050

Website: www.hotdb.com

Service email: service_hotdb@hotdb.cm

Service Tel: 021-5218 0789

Contents

1. Basic information	1
1.1. Profile of HotDB Server	1
1.1.1. Component architecture of HotDB Server.....	3
1.1.2. Explanation of technical terms.....	5
1.1.3. Compute node	6
1.1.4. Management platform.....	8
1.2. New functions and new features of Version 2.5.6	9
1.3. New compute node parameters of Version 2.5.6	11
2. Installation deployment and upgrading of HotDB Server	12
2.1. Service licensing	12
2.2. Installation deployment and upgrading.....	12
2.3. Config File	12
3. Rapid configuration of HotDB Server	15
3.1. Log in to management platform.....	15
3.2. Add compute node cluster.....	16
3.3. Add management platform user	17
3.4. Create MySQL database and data source user.....	17
3.5. Add LogicDB	18
3.6. Grant user LogicDB privilege.....	19
3.7. Add data source group	19
3.8. Add data node and data source.....	22
3.9. Add sharding function.....	25
3.10. Add table configuration.....	27
3.11. Check and reload configuration information	29
3.12. Log in to compute node and start to use	30
4. HotDB Server running related.....	34
4.1. Instruction on enabling compute node	34
4.1.1. Judgment of availability of LogicDB when enabling compute node	35
4.2. Parameter checking of MySQL server.....	37
4.2.1. Parameters required to be set as fixed value	37
4.2.2. Parameters requiring consistent configuration of all nodes	39
4.2.3. Parameters requiring not exceeding compute node configuration.....	40
4.3. Management port information monitoring.....	40
4.3.1. Management port command	41

4.3.2. Management port operation	42
4.4. Limit on number of front-end connections	44
4.5. Management of back-end connection pool	45
4.6. Limit on use of disk space	46
4.7. Reload	47
4.8. Config Checking	48
4.9. Deadlock Check	50
4.10. SQL error reporting log.....	54
5. Safety	59
5.1. User and privilege	59
5.2. SSL authentication	62
5.2.1. Generation of TLS secret key	62
5.2.2. Configuration of TLS secret key.....	65
5.2.3. TLS connection login.....	67
6. Data migration, backup and recovery	71
6.1. Use mysqldump backup	71
6.1.1. mysqldump - database backup program	71
6.2. Use binary log (increment) recovery	71
6.2.1. mysqlbinlog - utility program processing binary log files.....	71
6.2.2. Practical application of mysqldump and mysqlbinlog.....	73
7. Data consistency guarantee	78
7.1. Master/slave data consistency detection	78
7.2. Global AUTO_INCREMENT	80
7.2.1. When the parameter is set as 0.....	80
7.2.2. When the parameter is set as 1.....	81
7.2.3. When the parameter is set as 2.....	83
7.3. Data strong consistency (XA transaction).....	87
7.3.1. Use XA transaction	88
7.3.2. Relation between XA transaction and read/write splitting	91
7.4. Nondeterministic Function Proxy	92
7.5. Global Time Zone Support	93
7.6. Global Unique Constraint	94
7.6.1. Table level control when creating a table.....	95
7.6.2. Table level control when modifying a table.....	98
7.6.3. Online Change of Sharding Plan.....	99
7.6.4. Locate by secondary index when querying.....	100

7.7. Data accuracy guarantee after failover	101
8. High availability service	104
8.1. High availability service	104
8.1.1. High availability of data node.....	104
8.1.2. ConfigDB High Availability	111
8.1.3. Compute Node High Availability.....	113
8.1.4. Load-balancing of compute node.....	118
8.1.5. Compute node auto scaling.....	130
8.2. Read/write splitting.....	146
8.2.1. Description of Read/write splitting function.....	146
8.2.2. Weight configuration of Read/write splitting.....	148
9. Characteristic functions of HotDB Server.....	150
9.1. DNID.....	150
9.2. HINT	153
9.3. Connection binding.....	158
9.4. EXPLAIN	163
9.5. OnlineDDL	164
9.6. NDB Cluster SQL Node Service	165
9.6.1. NDB use restrictions.....	166
9.7. Online modification of sharding keys.....	168
9.7.1. Usage.....	168
9.7.2. Restrictions on use	168
10. Data type and character set support	170
10.1. Support of HotDB Server toward data type.....	170
10.1.1. Value type.....	170
10.1.2. Date and time type	171
10.1.3. Character string type	171
10.1.4. Space type	173
10.1.5. Other types	173
10.2. Support of HotDB Server toward character set	173
11. Function and operator support.....	177
11.1. Support of HotDB Server toward function	177
11.2. MERGE_RESULT	197
12. SQL syntax support	200
12.1. DML statement	200
12.1.1. DELETE statement	201

12.1.2. INSERT statement	203
12.1.3. LOAD DATA statement	209
12.1.4. REPLACE statement	211
12.1.5. SELECT statement	214
12.1.6. UPDATE statement	224
12.1.7. Cross-node JOIN	228
12.2. DDL statement	241
12.2.1. ALTER statement	241
12.2.2. CREATE statement	245
12.2.3. DROP statement	250
12.2.4. TRUNCATE AND RENAME statement	252
12.3. Transaction management and locking statement	253
12.4. Other MySQL statements	258
12.4.1. Storage procedure and function statement	258
12.4.2. Prepare SQL Statement	260
12.4.3. User management statement	261
12.4.4. Table maintenance statement	261
12.4.5. SET statement	267
12.4.6. SHOW statement	268
12.4.7. HotDB PROFILE	271
12.4.8. Other MySQL management statement	275
12.4.9. SHOW VARIABLES AND SHOW STATUS	276
12.5. Special functions of compute node syntax	283
12.5.1. Create Table is Sharding	283
12.5.2. Create table directly via existing Sharding Function	286
12.5.3. Guarantee that the data of dropped table is not lost for a period of time	308
13. INFORMATION_SCHEMA	312
14. Instruction on use of compute node parameters	318
14.1.1. adaptiveProcessor	318
14.1.2. allowRCWithoutReadConsistentInXA	321
14.1.3. autoIncrement	323
14.1.4. badConnAfterContinueGet	324
14.1.5. badConnAfterFastCheckAllIdle	325
14.1.6. bakUrl & bakUsername & bakPassword	325
14.1.7. checkConnLastUsedTime	328

14.1.8. CheckConnValid	329
14.1.9. checkConnValidTimeout.....	330
14.1.10. checkMySQLParamInterval	331
14.1.11. checkUpdate	332
14.1.12. clientFoundRows	334
14.1.13. clusterElectionTimeoutMs	336
14.1.14. clusterHeartbeatTimeoutMs	337
14.1.15. clusterHost	338
14.1.16. clusterName	339
14.1.17. clusterNetwork	340
14.1.18. clusterPacketTimeoutMs	341
14.1.19. clusterPort	342
14.1.20. clusterSize	343
14.1.21. clusterStartedPacketTimeoutMs	344
14.1.22. configMGR & bak1Url & bak1Username & bak1Password.....	345
14.1.23. cryptMandatory	347
14.1.24. dataNodeIdleCheckPeriod	351
14.1.25. deadlockCheckPeriod	353
14.1.26. defaultMaxLimit	354
14.1.27. dropTableRetentionTime	357
14.1.28. drBakUrl & drBakUsername & drBakPassword	358
14.1.29. drUrl & drUsername & drPassword	360
14.1.30. enableCursor	362
14.1.31. enableFlowControl.....	363
14.1.32. enableHeartbeat&heartbeatPeriod& heartbeatTimeoutMs	365
14.1.33. enableLatencyCheck & latencyCheckPeriod.....	367
14.1.34. enableListener	369
14.1.35. enableOracleFunction	372
14.1.36. enableSleep	374
14.1.37. enableSSL	376
14.1.38. enableSubquery	376
14.1.39. enableWatchdog	378
14.1.40. enableXA	381
14.1.41. errorsPermittedInTransaction	383
14.1.42. failoverAutoresetslave	385
14.1.43. frontConnectionTrxIsoLevel.....	386

14.1.44. frontWriteBlockTimeout	387
14.1.45. generatePrefetchCostRatio	388
14.1.46. globalUniqueConstraint	389
14.1.47. haMode	390
14.1.48. haState & haNodeHost	392
14.1.49. highCostSqlConcurrency	394
14.1.50. idcId & idcNodeHost	396
14.1.51. idleTimeout	398
14.1.52. joinable	401
14.1.53. joinBatchSize	403
14.1.54. joinCacheSize	404
14.1.55. joinLoopSize	406
14.1.56. keyStore	407
14.1.57. keyStorePass	408
14.1.58. lockWaitTimeout	409
14.1.59. masterSourceInitWaitTimeout	410
14.1.60. maxAllowedPacket	411
14.1.61. maxConnections & maxUserConnections	412
14.1.62. maxIdleTransactionTimeout	415
14.1.63. maxJoinSize	416
14.1.64. maxLatencyForRWSplit	419
14.1.65. maxNotInSubquery	421
14.1.66. maxReconnectConfigDBTimes	423
14.1.67. maxSqlRecordLength	424
14.1.68. ndbSqlAddr & ndbSqlUser & ndb SqlPass	425
14.1.69. ndbSqlDataAddr	427
14.1.70. ndbSqlMode	428
14.1.71. ndbSqlVersion & ndbVersion	428
14.1.72. operateMode	430
14.1.73. parkPeriod	433
14.1.74. pingLogCleanPeriod	434
14.1.75. pingLogCleanPeriodUnit	435
14.1.76. pingPeriod	436
14.1.77. prefetchBatchInit	438
14.1.78. prefetchBatchMax	439
14.1.79. prefetchBatchMin	440

14.1.80. prefetchValidTimeout.....	441
14.1.81. processorExecutor.....	442
14.1.82. Processors	443
14.1.83. readOnly.....	444
14.1.84. recordAuditlog	446
14.1.85. recordCrossDNJoin	447
14.1.86. recordDDL	448
14.1.87. recordDeadLockSQL	449
14.1.88. recordHotDBErrors.....	450
14.1.89. recordHotDBWarnings	451
14.1.90. recordLimitOffsetWithoutOrderby	454
14.1.91. recordMySQLErrors	455
14.1.92. recordMySQLWarnings	456
14.1.93. recordSql.....	458
14.1.94. recordSqlAuditlog.....	461
14.1.95. recordSQLIntercepted.....	463
14.1.96. recordSQLKeyConflict	464
14.1.97. recordSQLSyntaxError	466
14.1.98. recordSQLUnsupported	467
14.1.99. recordSubQuery	468
14.1.100. recordUNION	469
14.1.101. routeByRelativeCol.....	471
14.1.102. serverId	472
14.1.103. service port & management port.....	473
14.1.104. showAllAffectedRowsInGlobalTable.....	474
14.1.105. skipDatatypeCheck	475
14.1.106. socketBacklog	477
14.1.107. sqlTimeout	478
14.1.108. sslUseSM4	479
14.1.109. statisticsUpdatePeriod.....	481
14.1.110. strategyForRWSplit.....	482
14.1.111. switchByLogInFailover.....	487
14.1.112. switchoverTimeoutForTrans	488
14.1.113. timerExecutor.....	491
14.1.114. timestampProxy	492
14.1.115. url & username & password.....	493

14.1.116. usingAIO	496
14.1.117. version	497
14.1.118. versionComment	499
14.1.119. VIP & checkVIPPeriod	502
14.1.120. waitConfigSyncFinish	504
14.1.121. waitForSlaveInFailover	506
14.1.122. waitSyncFinishAtStartup	509
14.1.123. weightForSlaveRWSplit	511
15. Appendix	514
15.1. Notices for HotDB Server	514
15.1.1. Recommendation on JDBC version	514
15.1.2. Recommendation on JAVA database connection pool	514
15.1.3. Reserved field of database design	514

1. Basic information

This Manual is compiled based on *Distributed Transactional Database Product HotDB Server - V2.5.6*, and it mainly introduces basic use method and operating procedures of compute node, for reference and learning by the users.

Some functions in this manual could be used in combination with distributed transactional database management platform (hereinafter referred to as management platform), and if to know use method of management platform, please refer to [*Distributed Transactional Database HotDB Server \[Management Platform\] Function Manual*](#).

HotDB Server V.2.5.3.1 and above provide a solution based on MySQL native replication function to solve the problem of HotDB Server cross-IDC disaster recovery, which can realize the cross-IDC data synchronization function and solve the problem of cross-IDC distributed transactional database service disaster recovery. This document only describes the functions and features of HotDB Server in general mode in detail. To understand the functions and features in disaster recovery mode, please refer to the [*Distributed Transactional Database HotDB Server - Cross-IDC Disaster Recovery Function Specification*](#).

Special attention may not be paid to difference in version details of some screenshots, and the version number described in the document shall prevail. Since there are many contents in the document, it's recommended opening document map for the convenience of reading.

1.1. Profile of HotDB Server

HotDB Server is a distributed transactional database product which realizes horizontal expansion of data capacity and performance, and it could solve “two-large and three-high” (namely large-scale user, large-scale data, high availability, high concurrency and high throughput) problems in real-time transaction service system.

HotDB Server could provide centralized database operating experience for application

under the environment of data storage distribution, and meanwhile, it could provide a complete set of solutions on data safety, data disaster tolerance, data recovery, online capacity expansion without service stopped, cluster monitoring, intelligent topology, intelligent large screen, etc.

HotDB Server supports to split a table horizontally into multiple portions, and store them in different databases to realize sharding of the data, besides, it also supports vertical sharding and Global table. HotDB Server provides several table data sharding functions, and the user could use different data sharding functions according to transaction needs.

Basing R&D on Java NIO and MySQL protocol, HotDB Server supports multiplex of database connection, and is equipped with better concurrency performance.

HotDB Server provides automatic switching function of database service, and could effectively solve single-point failure of database.

The management platform in compatible use of compute node (also known as HotDB Management) is an important constituent part of the product.

1.1.1. Component architecture of HotDB Server

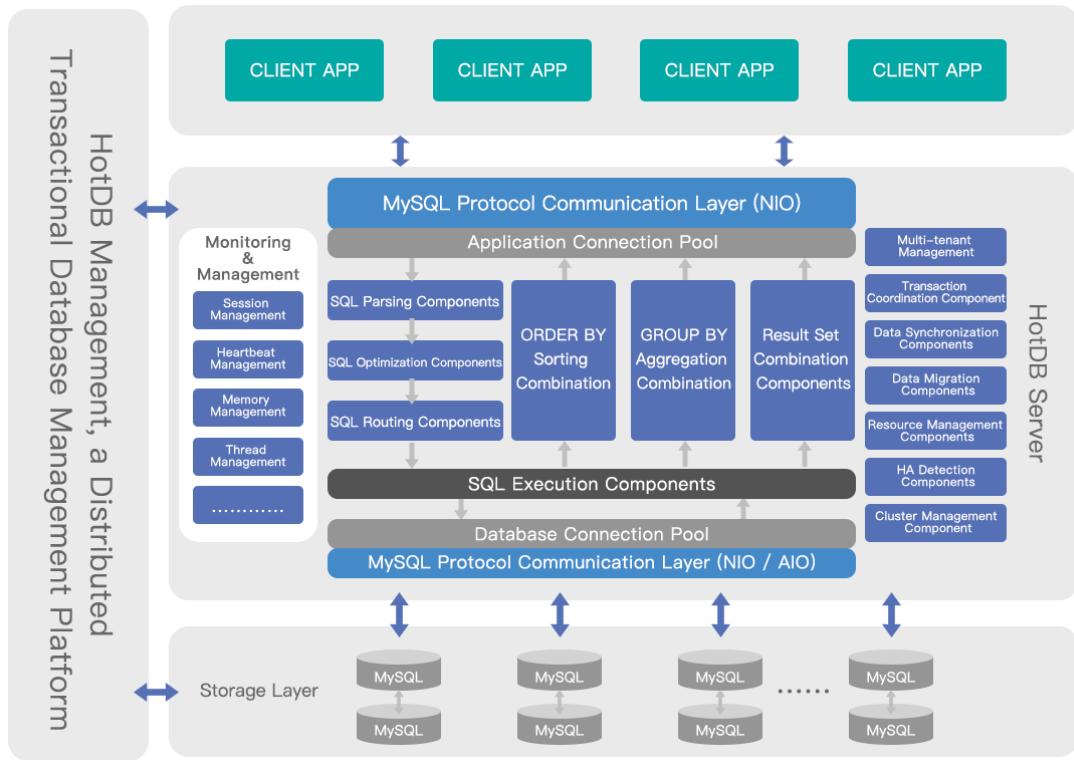


Figure 1.1.1-1 Functional component architecture diagram of HotDB Server

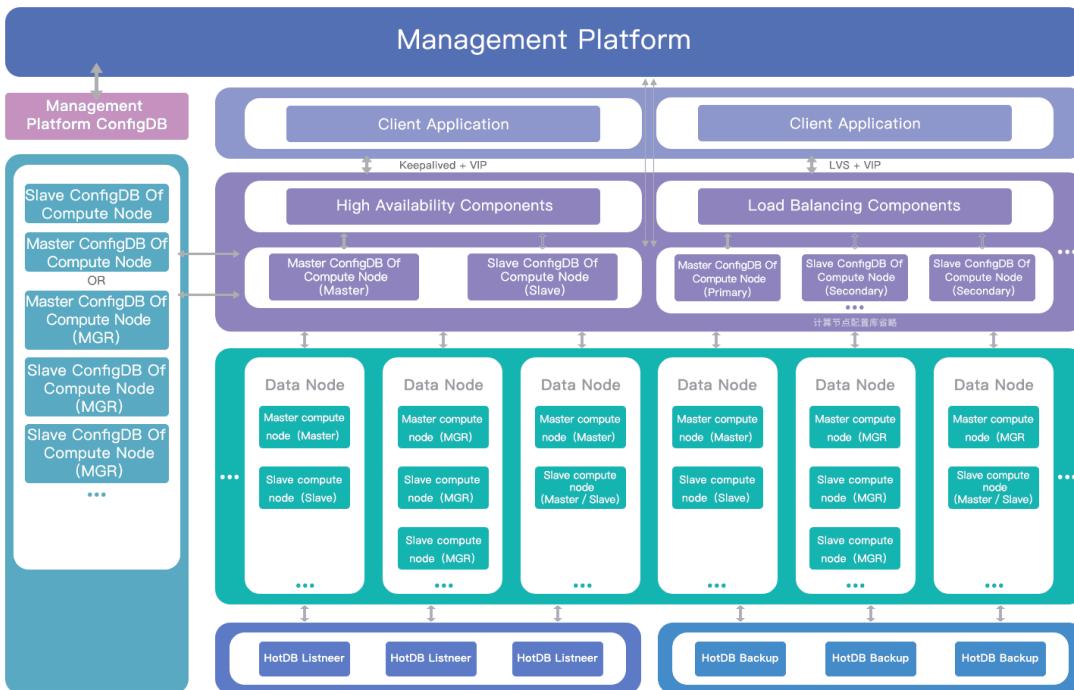


Figure 1.1.1-2 Component architecture diagram of HotDB Server

The distributed transactional database cluster (HotDB Server) is a database management system composed of a group of compute nodes, data sources, management platform and configDBs.

The distributed transactional database cluster (HotDB Server) is a database management system composed of a group of compute nodes, data sources, management platform and configDBs.

Compute node: Compute node is the core of distributed transactional database HotDB Server cluster system. It provides the core control functions of distributed transactional database, such as SQL parsing, routing distribution and result set merging, and is the lifeblood of the whole distributed service.

Data source: This is the MySQL database which makes actual storage of data, and IP+port + physical database could determine a data source. To realize the function of high availability and data multi-copy, a group of data sources with the same data copy are called one **data node** in HotDB.

Data nodes: A data node is a group of data sources with the same data copy. A data node can be a MySQL MGR cluster or a MySQL master-slave replication cluster. The data node manages the replication relations of a set of data sources with the same data copies. As sharding data in HotDB, all data nodes together constitute the total data of HotDB.

Management platform: Distributed transactional database management platform (hereinafter referred to as management platform) is also known as HotDB Management. It could realize usability configuration of compute node database user, data node, table type, sharding function and other information, and meanwhile, it could also provide monitoring of compute node service status, reminder of abnormal events, view of statements, management of tasks and other intelligent operation and maintenance related services.

ConfigDB: It is responsible for storing relevant configuration information of compute node and management platform. The configDB can perform high availability configuration through master-slave or MGR.

High availability: Compute node of HotDB Server could realize master/slave compute node availability checking and high availability switch by means of Keepalived high availability solution.

Load balancing: The compute node cluster mode of HotDB Server can achieve high availability and load balancing through LVS/F5. The application accesses the distributed transactional database service of HotDB Server through VIP of LVS. The distributed transactional database service is transparent to the application program, and the failure of single or multiple nodes in the compute node cluster has no effect on the application program.

HotDB Backup: Distributed transactional database backup program independently developed by Hotpu, is responsible for backup of transaction data.

HotDB Listener: A pluggable component developed by HotDB technology, which needs to be deployed separately and run in an independent process, so as to solve the problem of linear performance expansion under the strong consistency (XA) mode of cluster.

1.1.2. Explanation of technical terms

If to know relevant terms and relation of HotDB Server cluster system, please refer to [*Distributed Transactional Database HotDB Server \[Explanation of Terms\] Function Manual*](#).

1.1.3. Compute node

Compute node is data service provider, and its default service port is 3323, and could be logged in using the following MySQL command:

```
#mysql -uroot -proot -h127.0.0.1 -P3323
```

After login, compute node could be used the same as MySQL database, for example:

```
root> mysql -uroot -proot -h127.0.0.1 -P3323

mysql: [Warning] Using a password on the command line interface can be insecure.

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 515

Server version: 5.1.27-HotDB-2.5.0 HotDB Server by Hotpu Tech

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;

+-----+
| DATABASE           |
+-----+
| CLASSIC_LOGICDB |
| HotDB              |
+-----+

2 rows in set (0.01 sec)

mysql> use CLASSIC_LOGICDB
```

Database changed

```
mysql> show tables;

+-----+
| Tables_in_CLASSIC_LOGICDB |
+-----+
| customer |
+-----+
1 row in set (0.03 sec)
```

The application program connection of compute node is consistent with connection of MySQL, and only host, port, database, user and password information of the database Config File in the application program need to be modified. MySQL database driver and connection pool under different development platforms are supported, such as JDBC, c3p0, DHCP, DRUID connection pool of JAVA development platform. The following is configuration instance of c3p0 connection pool:

```
<!--database mapping-->

<!-- com.mchange.v2.c3p0.ComboPooledDataSource, org.apache.commons.dbcp.BasicDataSource -->

<bean id="dataSource1" class="com.mchange.v2.c3p0.ComboPooledDataSource"

    destroy-method="close">

        <property name="driverClass" value="com.mysql.jdbc.Driver" />

        <property name="driverClass" value="com.mysql.jdbc.Driver" />

        <property name="jdbcUrl"
value="jdbc:mysql://192.168.137.101:3323.cloth?characterEncoding=UTF-8" /> Where, port 3323 should be
changed to the service port of the compute node.

    <!--database mapping-->

        <property name="user" value="root" />

        <property name="password" value="$root" />
```

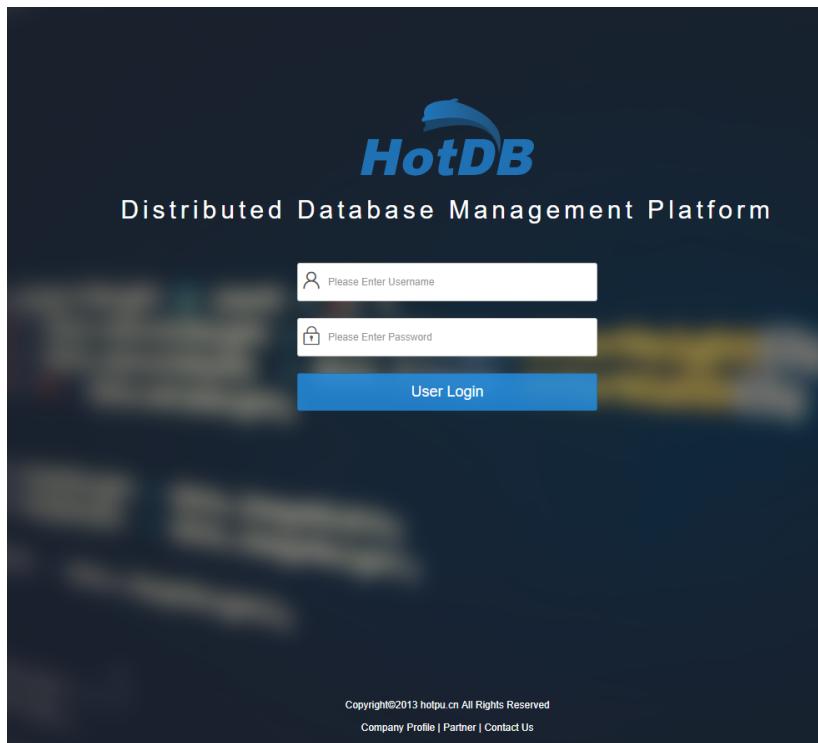
```
<property name="initialPoolSize" value="10" />  
  
<property name="maxPoolSize" value="$256" />  
  
<property name="minPoolSize" value="10" />  
  
<property name="maxIdleTime" value="1800" />  
  
<property name="maxStatements" value="1000" />
```

Meanwhile, compute node provides management port as 3325 by default, and the current service could be monitored and managed using command in management port. To know more information, please refer to [management port Information Monitoring](#).

1.1.4. Management platform

Management platform provides the compute node with configuration of user information, node information, table configuration and sharding, etc. Its default port is 3324, and by entering HTTP link address in the browser, access could be made to the management platform. Please use Chrome or FireFox browser.

For example: <http://192.168.200.191:3324/login>, the accesss page is shown as follow:



Both manager username and password are: admin by default, while other user accounts are created by the manager user, with the initial password being: service_hotdb@hotdb.com.

If to know detailed use method of the management platform, please refer to *Distributed Transactional Database HotDB Server [Management Platform] Function Manual*.

1.2. New functions and new features of Version 2.5.6

This chapter will briefly introduce the summary of functions which are added, prohibited or deleted in HotDB Server – V2.5.6. For detailed function usage, click the hyperlink to view the details:

- Support [the online expansion/reduction of compute node services](#), that is, the number of online compute node instances;
- Multi-node mode is supported for the cross-IDC disaster recovery function. For more details, please refer to the "[Distributed Transactional Database HotDB Server](#)

- V2.5.5 Cross-IDC Disaster Recovery Function Specification”.

- Support direct parsing and identifying some [Oracle functions and Sequence syntax](#) to reduce the amount of code modification when Oracle migrates to HotDB Server;
- Support direct parsing and identifying of some Oracle functions and Sequence syntax to reduce the amount of code modification when Oracle migrates to HotDB Server;
- Support SSL + [SM4](#) for client connection;
- Optimized function of creating [global tables](#) according to the default sharding node;
- Add parameter [operateMode](#) to meet one-click configuration of parameter combination under different scenarios, such as performance maximization, debugging mode, etc;
- Support [modification of sharding key](#) directly through SQL statements (alter table... change shard column...);
- Optimized [deadlock check](#) logic: control whether to roll back the transaction and start new transactions when deadlock occurs according to MySQL version number;
- Optimized log record of disconnection in [XA mode](#). You can analyze whether the transaction needs to be redone through the log;
- Optimized logic of role exchange after the failure of [data sources/ConfigDBs](#) or manual switching, and logic of reloading without changing the service of the original master;
- Optimized logic of the failure of data sources/ConfigDBs or manual switching. Compatible with the setting of [master_delay](#). Prevent the failure of switching due to the setting of master_delay;
- Optimization of [creating table directly via existing Sharding Function](#). The sharding attributes can be written after the table definition.
- Support the use of SQL statements to [CREATE / DROP user and to GRANT / REVOKE user](#);
- Support the use of SQL statements to [CREATE LogicDB](#);

- Support the use of SQL statements to CREATE / DROP / ALTER VIEW.

1.3. New compute node parameters of Version 2.5.6

This section will introduce the new compute node parameters added and optimized in Compute Node – V2.5.6, as listed below:

Parameter name of compute node	Description of compute node parameters	Default value	Re load is valid or not	Version supported
<u>enableOracleFunction</u>	Whether to parse Oracle functions first	false	N	2.5.6
<u>lockWaitTimeout</u>	Timeout for obtaining metadata lock (s)	31536000	Y	Synchronized downward to 2.5.3
<u>operateMode</u>	Compute node working mode	0	Y	Newly added in 2.5.6
<u>maxReconnectConfigDBTimes</u>	Maximum number of retries to connect to the ConfigDB	3	Y	2.5.6
<u>sslUseSM4</u>	Whether to support SM4	No	Y	Synchronized downward to 2.5.5
<u>haMode</u>	Added status: 4: master center in cluster mode 5: DR center in cluster mode	0	N	2.5.6
<u>crossDbXa</u>	Whether XA transactions are adopted in cross-LogicDB	false	N	2.5.5

2. Installation deployment and upgrading of HotDB Server

2.1. Service licensing

To make compute node of HotDB Server able to provide service normally, regular authorization license is required.

If to know how to obtain service licensing, please refer to *Distributed Transactional Database HotDB Server [Service Licensing] Function Manual*.

2.2. Installation deployment and upgrading

To deploy HotDB Server, JDK (JAVA operating environment), MySQL Database, USB KEY Driver Packet, Compute Node and Management Platform need to be installed. And a MySQL instance shall be configured as configDB of compute node. To deploy server operating system of compute node, 64-digit CentOS 6.x is recommended. If to know how to install, deploy and upgrade HotDB Server, or to know hardware configuration recommendation of HotDB Server, please refer to *Distributed Transactional Database HotDB Server [Install Deploy] Function Manual*.

2.3. Config File

Config File of compute node is located under conf directory after installation of compute node, and the file name is server.xml. If to view conditions of all parameters supported by compute node, please refer to [Instruction on use of compute node parameters](#).

After modification of some server.xml parameters, they will take effect only after reenabling compute node, and some parameters could take effect via “[Reload](#)”.

If the parameters listed in [Instruction on use of compute node parameters](#) do not exist in server.xml, that means the default value is used by the parameter; if you want to adjust a parameter value or add a parameter, please add the following code in server.xml, or add via “Configuration”->“Compute Node Parameters” page of the management platform.

```
<property name="dropTableRetentionTime">0</property><!--retention time of dropped table, 0 by default, not retained-->
```


3. Rapid configuration of HotDB Server

This section will introduce rapid configuration of HotDB Server. This section only introduces some necessary functions of Configuration to guarantee QuickStart, and to know more configuration functions, please refer to *Distributed Transactional Database HotDB Server [management platform] Function Manual*.

In the following example, a distributed transactional database cluster with three groups of data shardings (two data sources in each group) is configured. In this distributed transactional database cluster, a logicDB named "test" and a database table named "customer" is defined. This table is a sharding table; the sharding type is automatic sharding, and the sharding field is "provinceid".

Before configuring HotDB Server, please ensure that the management platform and compute node have been normally enabled, and 6 MySQL database instances have been well prepared (In this case, configuration of master-master data node will be taken for instance, and if only single-node data node is needed, 3 instances shall be prepared).

To know more about installation and deployment of HotDB Server and management platform, please refer to *Distributed Transactional Database HotDB Server [Install Deploy] Function Manual*.

3.1. Log in to management platform

Enter HTTP link address of management platform in browser, and log in to the management platform; HTTP link address is generally the server IP of the deployed management platform, the port is 3324 by default, for example, <http://192.168.200.89:3324/login.html>.

The management platform provides two kinds of user roles: super manager user and general user, super manager user has both the initial username and password as: admin by default; while the general user is created by the super manager user, with the default password being: hotdb@hotpu.cn.

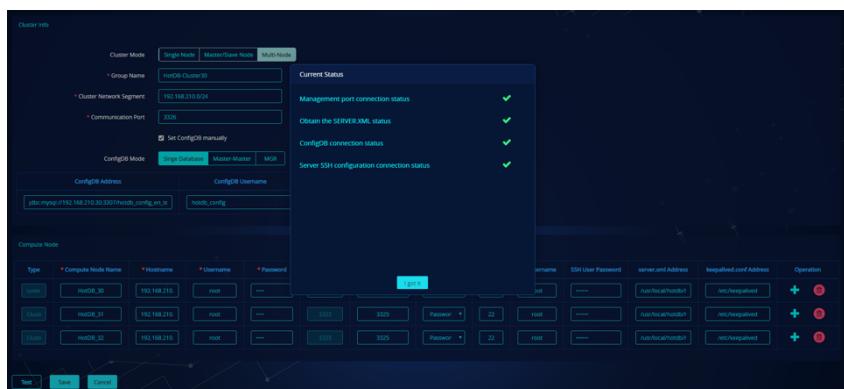
After the super manager user logs in, there are mainly “Compute Node Cluster Management” and “User Management” functions, the manager user could create and edit compute node cluster, and configure compute node connection information, add management platform user and add privilege for user, etc.

3.2. Add compute node cluster

Compute node cluster is a group of compute node services with high availability relation, add compute node cluster is to add the well deployed compute node to the management platform for management, and if to deploy a set of compute node cluster, please refer to *Distributed Transactional Database HotDB Server [Install Deploy Function Manual]*.

Click “Cluster Deployment and Configuration”->“Add Compute Node Cluster” on compute node cluster management page, enter compute node’s server IP service port, management port, username and password of management port connection, thus single compute node could be created, and after selecting master/slave node or multiple node, a group of high availability compute nodes could be added.

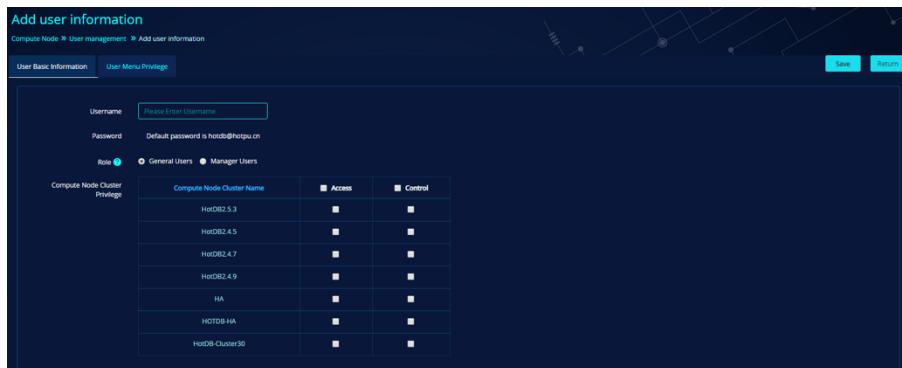
After entry, click Test, and if connection succeeded, this compute node cluster could be assigned to the management platform user for configuration management.



3.3. Add management platform user

Management platform user is the user who Manage, Configure, Monitor and Detect compute node cluster, and it has two privileges, one is access privilege (only some of the pages could be viewed), and the other is control privilege (including edit operation). When creating management platform user, the manager user assigns compute node cluster to management platform user, and assigns the control privilege at the same time, and after successful creation, the management platform user could log in to the page of Manage, Configure and Monitor Compute Node Cluster.

Log in to management platform, and on the Management Platform User page, click “Add New User”, enter username, and assign control privilege of compute node cluster. After completing the Add operation, the user could manage the compute node after login.



3.4. Create MySQL database and data source user

Create physical database and data source user respectively on 6 MySQL database instances (Data Source), log in to MySQL, and execute the following statements:

```

CREATE DATABASE db01 CHARACTER SET 'utf8' COLLATE 'utf8_general_ci';

GRANT
SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,INDEX,ALTER,PROCESS,REFERENCES,SUPER,LO
CK TABLES,REPLICATION SLAVE,REPLICATION CLIENT,TRIGGER,SHOW VIEW,CREATE

```

```
VIEW,CREATE ROUTINE,ALTER ROUTINE,EXECUTE,EVENT,RELOAD ON *.* TO
'hotdb_datasource'@'%' IDENTIFIED BY 'hotdb_datasource';
```

Note:

If MySQL version of the Data Source is 8.0 and above, authorization statements need to be added with XA_RECOVER_ADMIN privilege.

hotdb_Datasource account is the only account for HotDB to connect to each MySQL instance, through which all added platform users can connect to MySQL instances. Operations of users of each platform are only on front-end business connection and user access control.

Generally, the database created by create database in MySQL is called "LogicDB", while it is called "database" or "certain sharding database" in distributed transactional database system for it is where the data is actually saved.

For the convenience of configuration, the databases and users created on six MySQL database instances need to be consistent.

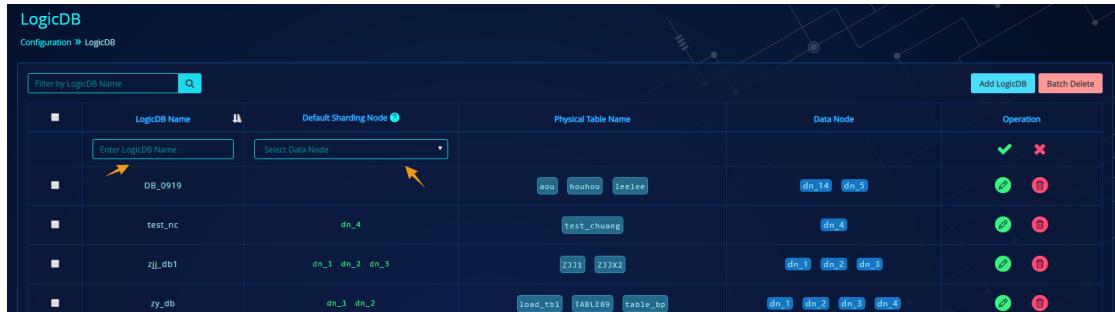
3.5. Add LogicDB

Usually, the database created by create database in MySQL is called "LogicDB". A "LogicDB" can provide database services for an application or a microservice. In order to maintain consistency, the distributed transactional database system also provides database services for an application or a microservice through the "LogicDB", but the distributed transactional database system is composed of multiple MySQL instances (in this case, 6 sets), so the "LogicDB" in HotDB Server is the "global LogicDB", that is, the set of "LogicDB" in 6 MySQL instances, which is still called LogicDB in HotDB Server.

LogicDB is a virtual database in compute node, after logging in to compute node by MySQL command, display the LogicDB list via the following statements:

show databases;

Log in to management platform page, select “Configuration”->“LogicDB”->“Add LogicDB”. Click “√”, save the configuration, and the LogicDB is successfully added.



The screenshot shows the LogicDB configuration interface. At the top, there's a search bar labeled "Filter by LogicDB Name". Below it is a table with columns: "LogicDB Name", "Default Sharding Node", "Physical Table Name", "Data Node", and "Operation". A new row is being added, indicated by a red-bordered input field for "LogicDB Name" containing "DB_0919". A dropdown menu for "Select Data Node" is open, showing options like "dn_1", "dn_2", "dn_3", etc. The "Operation" column for this row has a green checkmark icon. Other rows in the table represent existing LogicDBs: "test_nc" (sharded across "dn_1" to "dn_4"), "zjj_db1" (sharded across "dn_1" to "dn_3"), and "zy_db" (sharded across "dn_1" to "dn_2").

3.6. Grant user LogicDB privilege

Only the user granted with LogicDB privilege could use the LogicDB.

Log in to the management platform page, select “Configuration”->“Database User Management”, select Root User, and click “Edit” button. Jump to “Edit user privilege” page, and tick the created LogicDB“test” from the drop-down box, click “Save”, and the privilege is successfully granted.

Note:

After the management platform is installed, the system creates a platform user named root (password is root) by default.

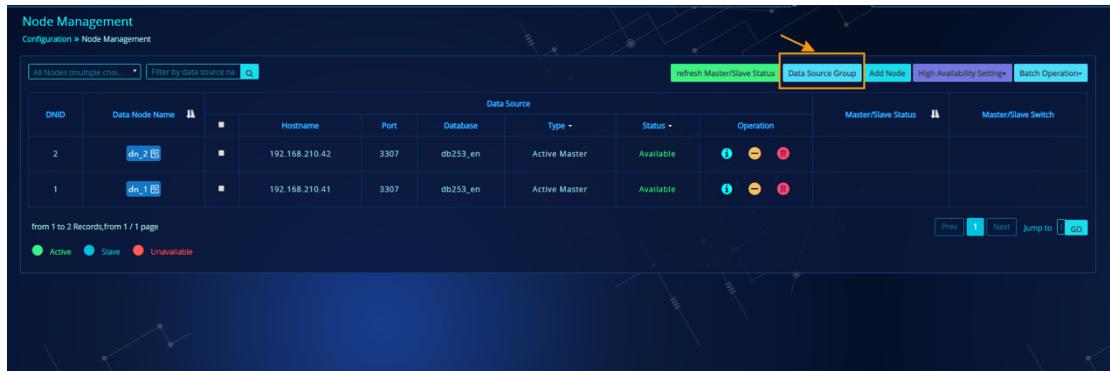


The screenshot shows the "Edit user rights" page under "LogicDB User Management". It has tabs for "Basic Information", "LogicDB privilege" (which is selected), and "Table privilege". A dropdown menu for "LogicDB" is open, showing "test" as the selected option. Below the dropdown, a note says "Select the corresponding LogicDB". A table below lists various database operations (SELECT, UPDATE, DELETE, INSERT, CREATE, DROP, ALTER, ALL) with checkboxes. The checkbox for "ALL" is checked for the "test" LogicDB row. The "Operation" column for this row has a green checkmark icon.

3.7. Add data source group

Add Data Source Group could make it more convenient to Add or Modify a group of data sources with the same parameter value.

Log in to Distributed Transactional Database Management Platform page, select “Configuration”->“Node Management”->“Data Source Group”->“Add Group”:



The parameters include:

- Group Name: enter data source group name;
- Connection User: The Username of user having access privilege to the physical database (the Username added in previous section);
- Connection User Password: the user password having access privilege to the physical database;
- Physical Database Name: citable database name in the data source, for example “db01” (the LogicDB added in previous section);
- Backup User: (optional) the username used for backup of the physical database;
- Backup User Password: (optional) the user password used for backup of the physical database;
- Character Set: Character Set of the connected physical database, utf8mb4 by default;
- Max Connections: Max Connections of MySQL physical database, 4200 by default;
- Initial Connections: Initial Connections of MySQL physical database, 32 by default;
- Max Idle Connections: Max Idle Connections of MySQL physical database, 512 by default;

- Min Idle Connections: [Min Idle Connections](#) of MySQL physical database, 32 by default;
- Idle Examination Period (second): Idle Examination Period of MySQL physical database, 600 by default. When the connection fails to send request to the server for a long time, the connection will be disconnected as timed, in order to avoid wasting the database connection.

According to business scenarios, select those with the same parameter value as a data source group, for example in this case, the parameter values ticked and entered in the figure below will constitute a data source group.

When Add Node, the data source group is applied on several data sources, which will Autofill the preset parameter value of the group; when Edit a parameter of the group, the parameter of all data sources in the group will be Edited in batches.

Add Data Source Group

Configuration > Node Management > Data Source group management > Add Data Source Group

Data Source Group Setting

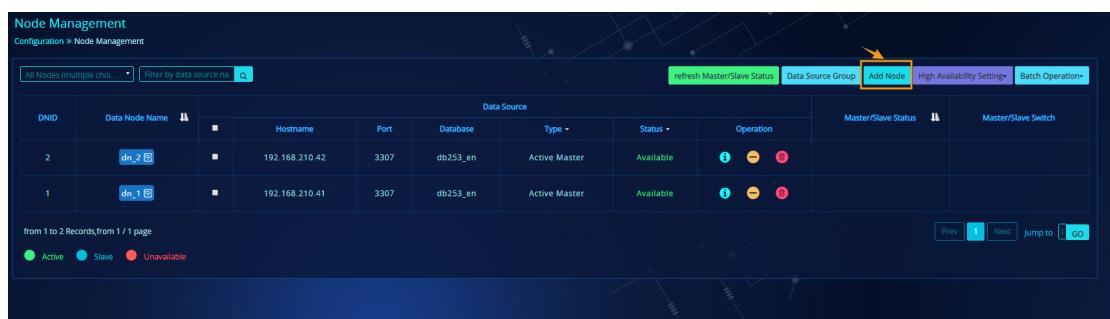
Group Name	group_name
<input checked="" type="checkbox"/> Connection User	root
<input checked="" type="checkbox"/> Connection User Password
<input checked="" type="checkbox"/> Database Name	db01
<input type="checkbox"/> Backup User	Enter backup user
<input type="checkbox"/> Backup User Password	Enter backup user password
<input checked="" type="checkbox"/> Character Set	utf8mb4
<input checked="" type="checkbox"/> Max Connections	4200
<input checked="" type="checkbox"/> Initial Connections	16
<input checked="" type="checkbox"/> Max Idle Connections	64
<input checked="" type="checkbox"/> Min Idle Connections	8
<input checked="" type="checkbox"/> Idle Examination Period (second)	600

Save and Return Return

3.8. Add data node and data source

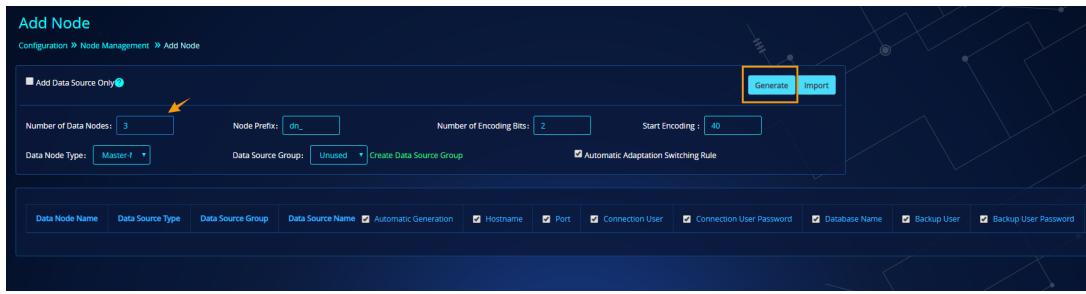
In this case, six MySQL instances are divided into three groups (three shardings) with two MySQL instances in each group (one active and one standby). The above description corresponds to the distributed transactional database system: the total data consists of three data nodes, each of which has two data sources. We need to do the following operations on the platform: add three data nodes and add two data sources for the three data nodes.

Log in to management platform page, select “Configuration”->“Node Management”->“Add Node”:

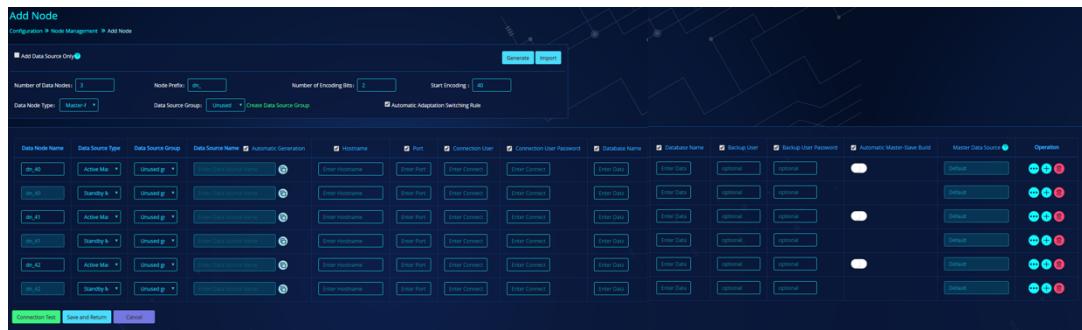


Either Add Data Node and its corresponding Data Source in batches or Add Data Source to existing Data Node is available, and only Add of Data Node and Data Source in batches is introduced here, and the operation is displayed as follow:

1. Fill in parameters of Data Node added: In this case, the Number of Data Nodes is 3, and the Data Node Type is Master-Master (other types could also be selected). In this case, Data Source Group selects not to use group, and you could also select to use the [Data Source Group](#) added in previous section from the drop-down menu, and then Add in batches or Edit similar parameters. Without special requirements, the Node Prefix, Number of Encoding Bits and Start Encoding could use the default value. After filling in the parameters, click [Generate].



2. Fill in Data Source Configuration Parameters according to the prompt message.

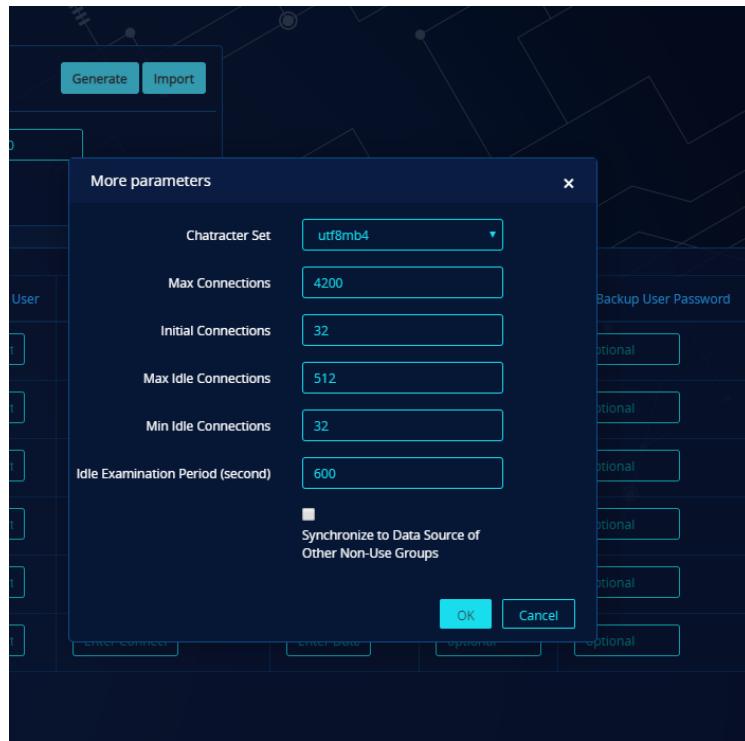


Parameters include:

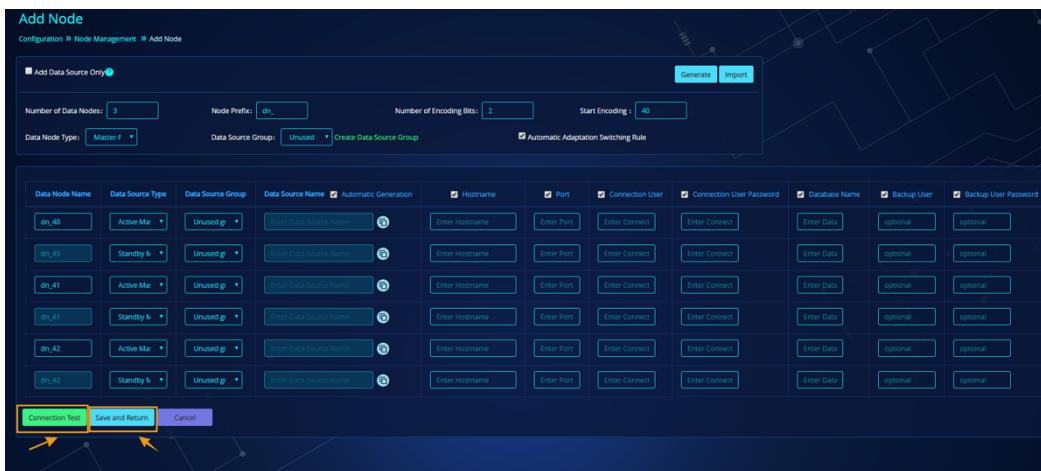
- Data Node: Generate according to the previous parameter filled in by default, Edit is available
- Data Source Type: Generate according to the previous parameter filled in by default, Edit is available
- Data Source Group: Generate according to the previous parameter filled in by default, Edit is available
- Data Source Name: Auto Check and Generate by default, but you could also enter Data Source Name in the textbox after un-checking, for example “ds_01”;
- Hostname: enter host Ip of MySQL database.
- Port Number: enter MySQL database port.
- Connection User: The Username with access privilege to the physical database (Username added in previous section);
- Connection User Password: The User Password with access privilege to the physical database;
- Physical Database Name: Database Name citable in Data Source, for example

- “db01” ([LogicDB](#) added in previous section);
- Backup User: (optional) Username used for backup of the physical database;
 - Backup User Password: (optional) User Password used for backup of the physical database;
 - Listener hostname(optional): The Listener is installed to solve the performance linear expansion problem of the compute node cluster mode. The Listener hostname is the host name of the server where the data source is located by default;
 - Monitoring port(optional): The port which is used to set the start of monitoring;
 - Listener service port (optional): Listener service port is the port where a compute node connects to a data source through a Listener. If a Listener needs to listen to multiple data sources, it needs to fill in different service ports for them;
 - Auto Master/Slave Build: after selection, Compute Node will Auto Build Replication Relation for the data source added according to the configuration information.
 - Master Data Source: This parameter shall be filled in only when it needs to set up replication relation like master-master with slave(s) or multiple levels of slaves. The master data source name needing to be built replication relation in the current data source could be copied and pasted here. By default, the system will make auto judgement according to the configuration.

Click [...] to unfold more parameters, including:



3. After completing the parameters, click [Connection Test] to verify that the entry is accurate and after all data sources are successfully connected, click [Save and Return], thus 3 data nodes and their respective corresponding 6 data sources have been successfully added.

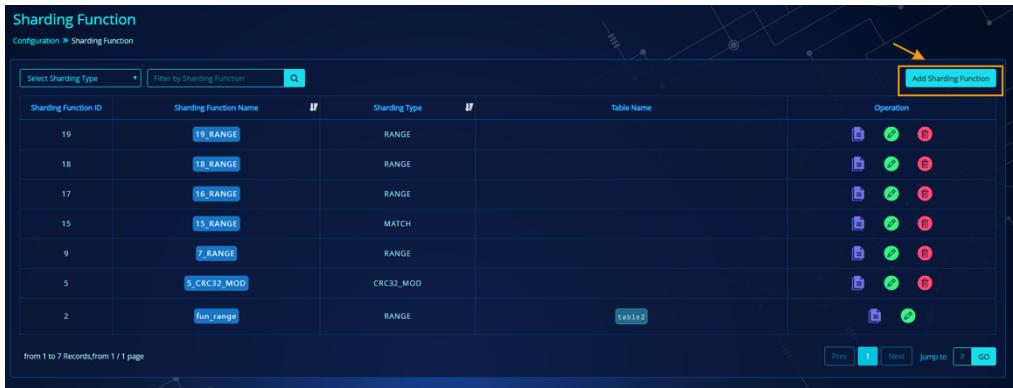


3.9. Add sharding function

The purpose of Add Sharding Function is to provide route method and algorithm of manual setting for table sharding, and if hoping to create table configuration by Auto

Sharding, this step could be skipped.

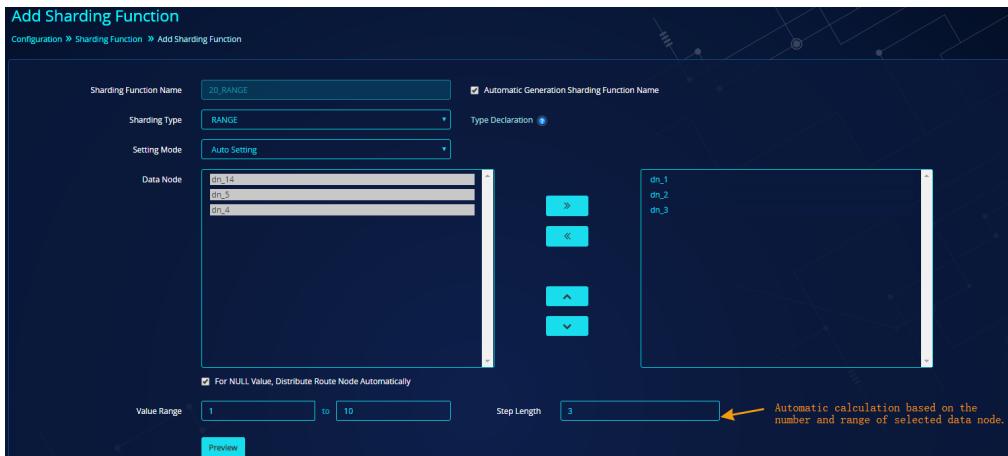
Log in to Management Platform page, select “Configuration”->“Sharding Function”->“Add Sharding Function”.



Sharding Function ID	Sharding Function Name	Sharding Type	Table Name	Operation
19	19_RANGE	RANGE		
18	18_RANGE	RANGE		
17	16_RANGE	RANGE		
15	15_RANGE	MATCH		
9	7_RANGE	RANGE		
5	5_CRC32_MOD	CRC32_MOD		
2	fun_range	RANGE	table2	

According to business scenarios, enter configuration parameters, including:

- Sharding Function Name: Generate by default, Edit is available after un-checking
- Sharding Type: includes ROUTE, RANGE, MATCH, SIMPLE_MOD, CRC32_MOD. Take RANGE for instance, to know more sharding functions, you could view more detailed function description document, and please refer to [*Distributed Transactional Database HotDB Server \[Management Platform\] Function Manual*](#).
- Setting Mode: includes Auto Setting and Manual Setting, and Auto Setting is taken for instance here. If selecting Auto Setting, the management platform will compute the Value Range automatically according to the configuration parameters, and partition the data nodes automatically; if selecting Manual Setting, the data nodes could be entered into corresponding Value Range manually.
- Data Node: select Sharding Data Node
- Value Range: enter full Value Range of sharding key, and the management platform will compute the Step Length automatically in combination of the Number of Nodes selected



Click [Preview] to view the generated results, and click [Modify] to modify the Value Range or Data Node, in order to solve data skew problem.

Value Range		Step Length					
1	to	10	3				
<small>from 1 to 5 Records, from 1 / 1 page</small>							
Data Node		Value Range					
dn_1		1:3					
dn_2		4:6					
dn_3		7:9					
dn_1		10:10					
dn_2		null					
<small>Filling Instruction: Input Numerical Range such as 1:10, and use English Commas and English Space between different Input Values for separation. Input Value could be NULL Value, and NULL Value is represented by null</small>							
<input type="button" value="Preview"/>		<input type="button" value="Prev"/>	<input type="button" value="1"/>	<input type="button" value="Next"/>	<input type="button" value="Jump to"/>	<input type="text" value="0"/>	<input type="button" value="GO"/>
<input type="button" value="Save and Return"/>		<input type="button" value="Return"/>		<input type="button" value="Modify"/>			

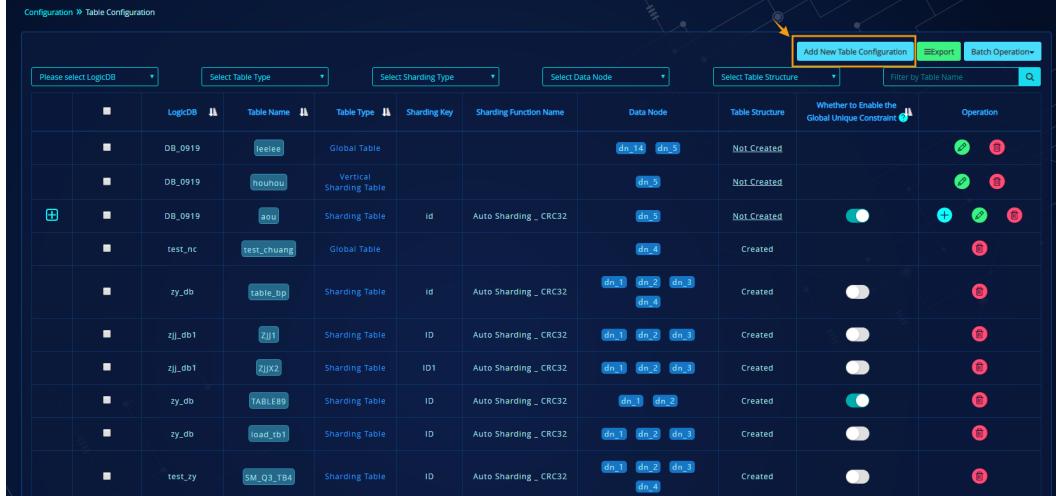
Click [Save and Return] to add sharding function.

3.10. Add table configuration

Log in to Management Platform page, select “Configuration”->“Table Configuration”->“Add Table Configuration”

Table Configuration

Configuration > Table Configuration

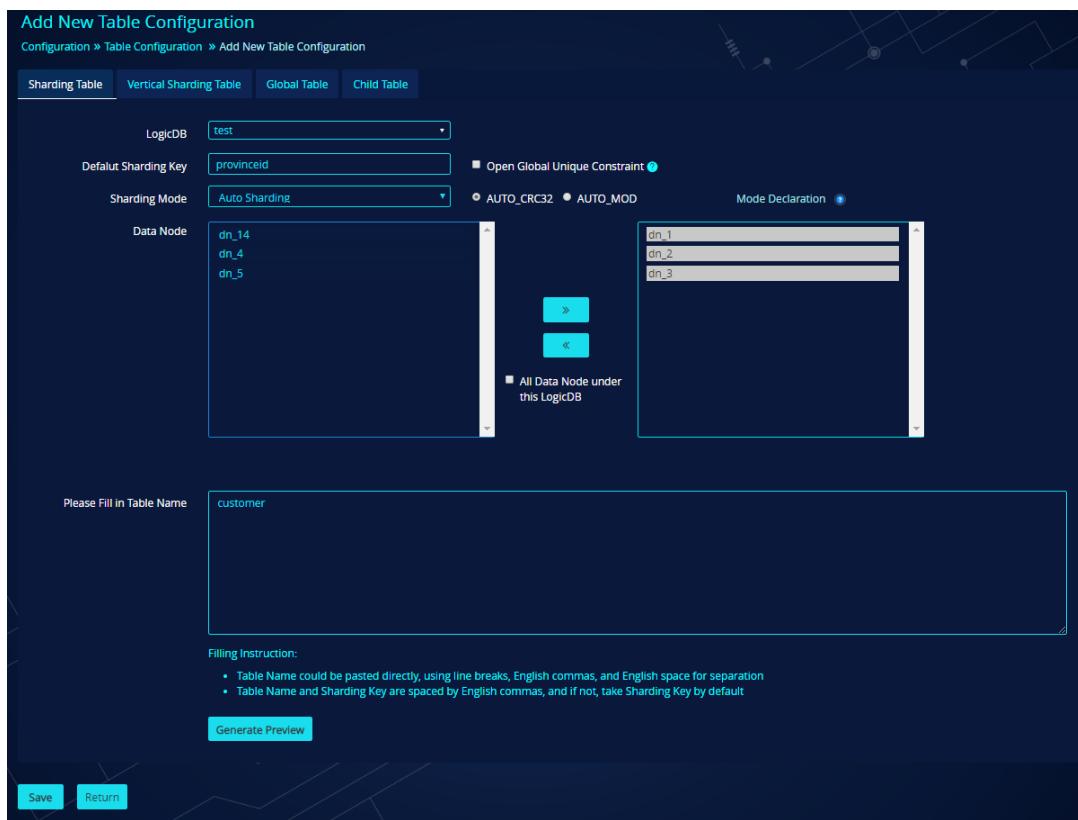


The screenshot shows a table configuration page with the following columns:

Action	LogicDB	Table Name	Table Type	Select Sharding Type	Sharding Key	Sharding Function Name	Data Node	Table Structure	Whether to Enable the Global Unique Constraint	Operation
	DB_0919	leeee	Global Table				dn_14 dn_5	Not Created	<input checked="" type="checkbox"/>	
	DB_0919	houhou	Vertical Sharding Table				dn_5	Not Created	<input checked="" type="checkbox"/>	
Add	DB_0919	asu	Sharding Table	id	Auto Sharding _ CRC32		dn_5	Not Created	<input checked="" type="checkbox"/>	
	test_nc	test_chuang	Global Table				dn_4	Created		
	zy_db	table_bp	Sharding Table	id	Auto Sharding _ CRC32	dn_1 dn_2 dn_3 dn_4		Created	<input checked="" type="checkbox"/>	
	zjj_db1	ZJ1	Sharding Table	ID	Auto Sharding _ CRC32	dn_1 dn_2 dn_3		Created	<input checked="" type="checkbox"/>	
	zjj_db1	ZJX2	Sharding Table	ID1	Auto Sharding _ CRC32	dn_1 dn_2 dn_3		Created	<input checked="" type="checkbox"/>	
	zy_db	TABLE89	Sharding Table	ID	Auto Sharding _ CRC32	dn_1 dn_2		Created	<input checked="" type="checkbox"/>	
	zy_db	load_tb1	Sharding Table	ID	Auto Sharding _ CRC32	dn_1 dn_2 dn_3		Created	<input checked="" type="checkbox"/>	
	test_zy	SM_Q3_TB4	Sharding Table	ID	Auto Sharding _ CRC32	dn_1 dn_2 dn_3 dn_4		Created	<input checked="" type="checkbox"/>	

According to business scenarios, after selecting the Table Type, enter the configuration parameters. In this case, under the Sharding Table page, add the parameter configuration as follow:

- LogicDB: From the drop-down menu, select the [LogicDB](#) test added in previous section.
- Default Sharding Key: When filling in the Table Name, Table Name and its corresponding Sharding Key shall be separated by English comma, and if no Sharding Key is filled in, the default Sharding Key shall be taken. Therefore, “provinceid” is filled in here
- Sharding Mode: In this case, either Auto Sharding, or the [Sharding Function](#) added in previous section could be selected here
- Data Node: select Sharding Data Node. in this case, the [Data Node](#) added in previous section shall be selected
- Please Fill in Table Name: enter “customer”, when Add multiple tables but different sharding keys, “customer:provinceid” could be entered.



Click [Save], and Customer Auto Sharding Table is successfully added. Note: The Sharding Function cited in this table is AUTO_CRC32 type (for difference of the sharding types AUTO_MOD and AUTO_CRC32, the “Mode Declaration” in the page could be viewed).

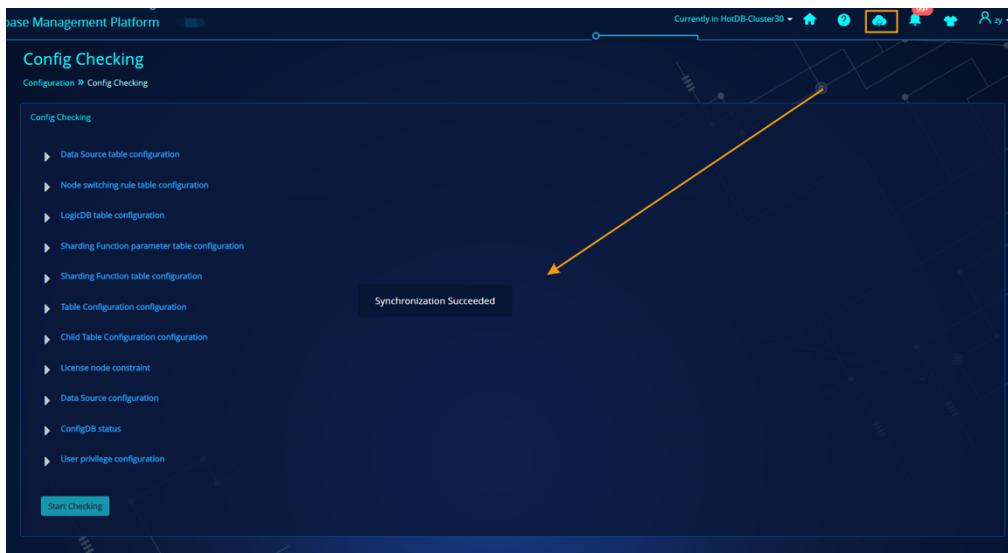
3.11. Check and reload configuration information

Log in to Management Platform page, and for any modification made to the HotDB Server User, LogicDB, Data Node, Data Source, Failover, Sharding Function, Table Configuration and Child Table Configuration, under the condition of not re-enabling the compute node service, the new configuration information shall take effect only after “Reload”.

If Compute Node is not enabled, reload can't be executed, therefore, Compute Node shall be enabled first.

Log in to Management Platform page, select “Configuration”->“Config Checking”, and click “Start Checking” in the page, if there is no prompt of configuration error, it means that the configuration information is correct:

Click “Reload” in the page, if it’s promoted “Reload Succeeded” in the page, then the configuration information has taken effect successfully in the compute node:



3.12. Log in to compute node and start to use

Use MySQL command line, to specify actual IP address, and log in to the compute node:

```
mysql -uroot -proot -h127.0.0.1 -P3323 -Dtest
```

For example:

```
root> mysql -h127.0.0.1 -uroot -proot -P3323 -Dtest

mysql: [Warning] Using a password on the command line interface can be insecure.

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 100728

Server version: 5.7.19-HotDB-2.5.2 HotDB Server by Hotpu Tech
```

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or 'h' for help. Type '\c' to clear the current input statement.

mysql>

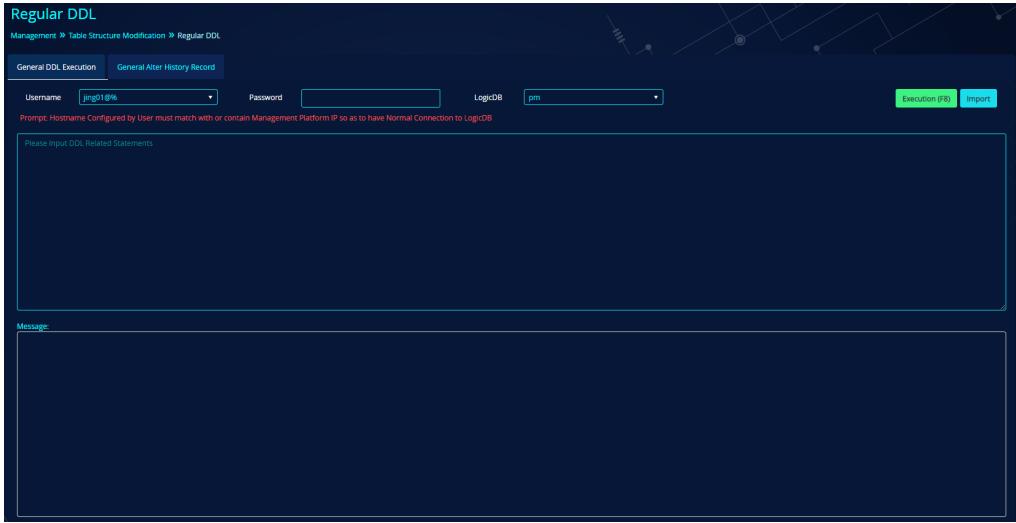
Execute customer Create Table statement:

```
CREATE TABLE `customer`(  
  
    `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,  
  
    `name` VARCHAR(32) NOT NULL,  
  
    `telephone` VARCHAR(16) NOT NULL,  
  
    `provinceid` TINYINT UNSIGNED NOT NULL DEFAULT 0,  
  
    `province` ENUM  
        ('Anhui','Aomen','Beijing','Chongqing','Fujian','Gansu','Guangdong','Guangxi','Guizhou','Hainan','Hebei','Heilongji  
        ang','Henan','Hubei','Hunan','Jiangsu','Jiangxi','Jilin','Liaoning','Neimenggu','Ningxia','Qinghai','Shaanxi','Shandong  
        ','Shanghai','Shanxi','Sichuan','Taiwan','Tianjin','Xianggang','Xinjiang','Xizang','Yunnan','Zhejiang') NULL,  
  
    `city` VARCHAR(16) NULL default ",  
  
    `address` VARCHAR(64) NULL,  
  
    PRIMARY KEY(`id`),  
  
    UNIQUE KEY(`telephone`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

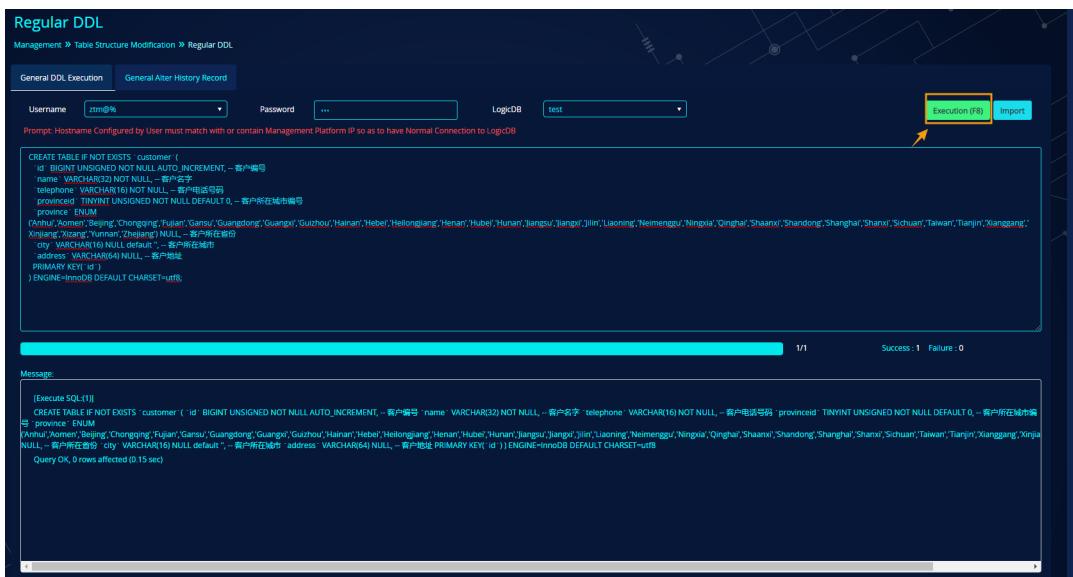
Compute node will Create Customer Table in various data nodes. You can log in to various MySQL data sources, to verify whether customer has been created or not.

Or find the [Table Configuration](#) added in previous section on the “Configuration”->“Table Configuration” page, and click [Not Created] in the Table

Structure column to jump to Ordinary DDL page.



Enter [LogicDB Username Password](#), and after selecting Test LogicDB, enter Create Table statement, click [Execute] to Add Table Structure.



After the Sharding Table Customer is successfully created, you could execute the following SQL statements in Compute Node, and write in data:

```
INSERT INTO customer VALUES (21,'何重庆','13912340021',4,'Chongqing','重庆','某某街某某号');

INSERT INTO customer VALUES (22,'吕重庆','13912340022',4,'Chongqing','重庆','某某街某某号');

INSERT INTO customer VALUES (25,'孔福州','13912340025',5,'Fujian','福州','某某街某某号');

INSERT INTO customer VALUES (26,'曹兰州','13912340026',6,'Gansu','兰州','某某街某某号');
```

```
INSERT INTO customer VALUES (67,'岑南昌','13912340067',17,'Jiangxi','南昌','某某街某某号');

INSERT INTO customer VALUES (68,'薛长春','13912340068',18,'Jilin','长春','某某街某某号');

INSERT INTO customer VALUES (69,'雷沈阳','13912340069',19,'Liaoning','沈阳','某某街某某号');

INSERT INTO customer VALUES (70,'贺呼和浩特','13912340070',20,'Neimenggu','呼和浩特','某某街某某号');

INSERT INTO customer VALUES (71,'倪银川','13912340071',21,'Ningxia','银川','某某街某某号');

INSERT INTO customer VALUES (72,'汤西宁','13912340072',22,'Qinghai','西宁','某某街某某号');

INSERT INTO customer VALUES (73,'滕西安','13912340073',23,'Shaanxi','西安','某某街某某号');

INSERT INTO customer VALUES (74,'殷济南','13912340074',24,'Shandong','济南','某某街某某号');

INSERT INTO customer VALUES (93,'顾台北','13912340093',28,'Taiwan','台北','某某街某某号');

INSERT INTO customer VALUES (94,'孟天津','13912340094',29,'Tianjin','天津','某某街某某号');

INSERT INTO customer VALUES (95,'平香港','13912340095',30,'Xianggang','香港','某某街某某号');

INSERT INTO customer VALUES (96,'黄乌鲁木齐','13912340096',31,'Xinjiang','乌鲁木齐','某某街某某号');

INSERT INTO customer VALUES (99,'萧杭州','13912340099',34,'Zhejiang','杭州','某某街某某号');

INSERT INTO customer VALUES (100,'尹杭州','13912340100',34,'Zhejiang','杭州','某某街某某号');
```

Next, you could log in to Compute Node Service, to execute DELETE, UPDATE and SELECT operations toward the Customer Sharding Table.

4. HotDB Server running related

4.1. Instruction on enabling compute node

- When the configuration error of Data Node, Data Source and other information can't pass the Config Checking, Compute Node cannot be enabled;
- To enable compute node, you could switch to /usr/local/hotdb/hotdb-server/bin directory first, and then run the Startup Script, or add the following path directly:
`sh /usr/local/hotdb/hotdb-server/bin/hotdb_server start;`
- Synchronization State of configDB could influence Compute Node Enable, when compute node is enabled or in case of online High Availability Switch, configDB must ensure synchronization to catch up;
- Synchronization Status of Data Source could influence Compute Node Enable. Use true/false attribute of the configuration parameter [waitSyncFinishAtStartup](#) in server.xml to Control whether to wait for synchronization of data source to catch up when Compute Node is enabled. Wait by default;
- When Compute Node is enabled, if the Data Source is of abnormal connection, you could control whether the Master Data Source in Data Node shall be re-initialized or not and the Initialization Timeout time through editing the configuration parameter [masterSourceInitWaitTimeout](#) in server.xml, and please refer to [Judgment of availability of LogicDB when enabling compute node](#) for specific control logic.

4.1.1. Judgment of availability of LogicDB when enabling compute node

In order to ensure that in case of Unavailable Status of Data Node in Vertical sharding scenario, the unassociated business scenarios of different LogicDB are not influenced, therefore, at the time of Compute Node Enable, special judgment and treatment have been made toward available status of all LogicDB, and the description is as follow:

- If the configured Master Data Source is in Available status, but this data source can't be connected in reality, then at the time of Compute Node Enable, it will wait for configuration time of [masterSourceInitWaitTimeout](#) (default:300s), to judge whether the data source is really un-connectable or not, and if during this period, the data source comes into reconnection without abnormality, then this node is successfully initialized;
- If there are Unavailable nodes under all LogicDB, then the compute node can't be enabled, log prompt: 04/13 10:50:54.644 ERROR [main] (HotdbServer.java:436) - datanodes:[3] init failed. System exit.
- As long as corresponding data node of a LogicDB is available, then the compute node could start, and the table under corresponding logic could come into normal operation. If there is unavailable node under other LogicDB, then the table under the LogicDB can't make normal Read/Write, Client prompt: ERROR 1003 (HY000): DATABASE is unavailable when datanodes: [datanode_id] unavailable.

For example: ALogicDB includes Node 1, 2, and BLogicDB includes Node 3, 4. If Node 1, 2 are Unavailable, but Node 3, 4 are available, then compute node could start, the table under BLogicDB could come into normal operation, the table under ALogicDB can't make Read/Write; if Node 1, 3 are Unavailable, then the compute node can't be enabled.

➤ Judgment of whether a node is available or not, is related with the status of the data source in configDB and the actual available status of the data source, and it's required that the configuration status shall be consistent with the data source status. Otherwise, Compute Node Enable will be influenced, and when compute node is enabled, connect configured available data source of configDB. If connection succeeded, it shall be deemed as available; if a data source configured available cannot be connected, even if the node has other available data sources which could be connected, the node shall be deemed as Unavailable, and each node shall be configured with at least one available data source, otherwise compute node can't be enabled. The specific conditions are as follow:

1. Master/slave data source is configured available

If master/slave data source could be connected, then the node is available. If Active Master can't be connected, Standby Slave could be connected, then there will be Switch, and the Active Master will be set Unavailable, and the Standby Slave will be used, but the compute node shall still judge that the node is Unavailable. If Active Master could be connected, but Standby Slave can't be connected, then Active Master will be used, and Standby Slave will be set Unavailable, and the compute node will judge that the node is Unavailable. If master/slave database can't be connected, then the node is Unavailable.

2. Active Master Configuration is Unavailable, Standby Slave Configuration is available

If Standby Slave could be connected, then Standby Slave shall be used, and the node is available. If Standby Slave can't be connected, then the node is Unavailable

3. Active Master Configuration is available, but Standby Slave Configuration is Unavailable

If Active Master could be connected, then Active Master shall be used, and the node is available. If Active Master can't be connected, then the node is Unavailable

4.2. Parameter checking of MySQL server

In order to guarantee data consistency, when compute node is enabled, it will make checking of the parameters of MySQL data source service port. For different parameters, if parameter configuration doesn't conform to checking rule, compute node will report warning message, or can't be enabled. There are two kinds of requirements of compute node for parameters of MySQL data source service port: one is that all service port parameters between the data sources shall be consistent; the other is that all service port parameters of data source must be fixed value.

4.2.1. Parameters required to be set as fixed value

As for following parameters of MySQL data source service port, they are required to be set as uniform fixed value by compute node:

1. **completion_type must be NO_CHAN**, if the parameter is not standard, then reload fails;
2. **innodb_rollback_on_timeout shall be ON**, and the innodb_rollback_on_timeout parameter shown by show [global|session] variables at any time shall be on, the description is as follow:
 - If innodb_rollback_on_timeout parameters are all off, then compute node allows successful load, but the behavior of the compute node equals to the transaction rollback method when innodb_rollback_on_timeout parameter is on, and the following prompts will be given at the time of Config Checking:

Config Checking

Compute Node Configuration ➤ Config Checking

Config Checking

- ▶ Data source table Config Checking passed ✓
- ▶ Node switching table Config Checking passed ✓
- ▶ LogicDB table Config Checking passed ✓
- ▶ Sharding function parameter table Config Checking passed ✓
- ▶ Sharding Function table Config Checking passed ✓
- ▶ Table Configuration Config Checking passed ✓
- ▶ Child Table Configuration Config Checking passed ✓
- ▶ License node constraint checking passed ✓
- ▼ Data source Config Checking error ✖
- ▼ a. Whether Data source configuration is correct +
- Data Node: dn_1 Data Source under:192.168.210.41_3307_db253(192.168.210.41:3307/db253) of The values of innodb_rollback_on_timeout are inconsistent, and the current value is OFF, it is recommended to change it to ON
- Data Node: dn_2 Data Source under:192.168.210.41_3307_phy253(192.168.210.41:3307/phy253) of The values of innodb_rollback_on_timeout are inconsistent, and the current value is OFF, it is recommended to change it to ON
- Data Node: dn_5 Data Source under:192.168.210.41_3307_phy1416(192.168.210.41:3307/phy1416) of The values of innodb_rollback_on_timeout are inconsistent, and the current value is OFF, it is recommended to change it to ON

And at the time of Reload, the log output will be: innodb_rollback_on_timeout=off is not supported, HotDB behavior will be equivalent to innodb_rollback_on_timeout=on.

- If innodb_rollback_on_timeout parameter data sources are inconsistent, Reload will fail, and there will be prompt as follow at the time of Config Checking:

- ▶ Child Table Configuration Config Checking passed ✓
- ▶ License node constraint checking passed ✓
- ▼ Data source Config Checking error ✖
- ▼ a. Whether Data source configuration is correct ✖
 - Data Node: dn_1 Data Source under:192.168.210.41_3307_db253(192.168.210.41:3307/db253) of The values of innodb_rollback_on_timeout are inconsistent, and the current value is ON
 - Data Node: dn_2 Data Source under:192.168.210.41_3307_phy253(192.168.210.41:3307/phy253) of The values of innodb_rollback_on_timeout are inconsistent, and the current value is ON
 - Data Node: dn_5 Data Source under:192.168.210.41_3307_phy1416(192.168.210.41:3307/phy1416) of The values of innodb_rollback_on_timeout are inconsistent, and the current value is ON
 - Data Node: dn_3 Data Source under:192.168.210.44_3309_phy253(192.168.210.44:3309/phy253) of The values of innodb_rollback_on_timeout are inconsistent, and the current value is OFF, it is recommended to change it to ON
 - Data Node: dn_4 Data Source under:192.168.210.42_3308_db253(192.168.210.42:3308/db253) of The values of innodb_rollback_on_timeout are inconsistent, and the current value is ON
 - Data Node: dn_14 Data Source under:192.168.210.45_3307_qianku1(192.168.210.45:3307/qianku1) of The values of innodb_rollback_on_timeout are inconsistent, and the current value is OFF, it is recommended to change it to ON
 - Data Node: dn_4 Data Source under:192.168.210.42_3308_db253(192.168.210.42:3308/db253) The time difference with Compute Node is more than 3 seconds, with a difference of 2h
 - b. Reload requirement: the replication delay between available master Data source and original master Data source cannot exceed 10s ✓
 - c. Reload requires that the replication status between available master data source and original master data source shall be normal. ✓
- ▶ ConfigDB connection is normal ✓

And at the time of Reload, the data source being off will have log output: MySQL variables 'innodb_rollback_on_timeout' is not consistent, the current value is OFF ,neet to bu changed to ON, and the data source being on will have log output: MySQL variables 'innodb_rollback_on_timeout' is not consistent, the current value is ON

3. **read_only**, the parameter description is as follow:

- If for master data source, the parameter `read_only=1`, then compute node will refuse to start, and reload fails.
- If for the Slave, the parameter `read_only=1` and is configured with the configuration rule of Switch to the Slave, then compute node could start, RELOAD fails.
- If for the Slave, the parameter `read_only=1` and is not configured with the configuration rule of Switch to the Slave, then compute node could start, and reload will be successful if without other error.

4.2.2. Parameters requiring consistent configuration of all nodes

For the following parameters of MySQL data source service port, parameter values between the data sources are required to be set consistent by the compute node:

- `autocommit`
- `transaction_isolation`
- `div_precision_increment`

If the above parameters are configured inconsistent between the data sources, compute node will give warning message. For `transaction_isolation` parameter, if the min configuration value is lower than REPEATABLE-READ, compute node will use REPEATABLE-READ mode; if the max configuration value is higher than REPEATABLE-READ, SERIALIZABLE will be used.

4.2.3. Parameters requiring not exceeding compute node configuration

Considering that sending super-large SQL by Client may threaten HotDB Server (no practical case has yet been discovered at present), HotDB Server could be configured with MAX_ALLOWED_PACKET the same as MySQL, to control the max packet size of SQL sent to compute node by Client, and the parameter could be preset in server.xml via parameter name maxAllowedPacket, if maxAllowedPacket of compute node has its default value bigger than MySQL, the log will give warning prompt, and Config Checking on the management platform will also give prompt:



4.3. Management port information monitoring

HotDB Server provides the customer a set of information monitoring, statistics and service management functions which are perfect and easy to operate. The user could log in to monitoring management port of compute node via MySQL Client to view the detailed information, please refer to [Distributed Transactional Database HotDB Server \[management port Command\] Function Manual](#) for detailed description.

4.3.1. Management port command

The user could log in to management port (default port: 3325) to use show @@help command to view the supported management port command and corresponding role.

```
root> mysql -uroot -proot -P3325 -h192.168.200.201
mysql: [Warning] Using a password on the command line interface can be insecure.

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 992081

Server version: 5.1.27-HotDB-2.5.0 HotDB Manager by Hotpu Tech

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or 'h' for help. Type '\c' to clear the current input statement.

mysql> show @@help;
+-----+-----+
| statement | description |
+-----+-----+
| check @@datasource_config | Inspect MySQL parameter configuration information | |
| check @@route [db_name.tb_name | tb_name] | Detect the data routing correctness of Sharding Table |
| kill @@connection [connection_id] | Close one appointed connection |
| onlineddl "[DDLSTATEMENT]" | Execution onlineddl |
| rebuild @@pool | Rebuild currently available datasources of all nodes |
| reload @@config | Re-load the configuration information |
| restart @@heartbeat [datanode_id] | Recover the heartbeat detection on the appointed data node |
|
```

show @@auxtable	Displays the created auxiliary table information
-----------------	--

--

...more contents are omitted, and you could log in to view...

4.3.2. Management port operation

The user can enter corresponding command to monitor service condition of compute node, such as showing the data source information:

```
mysql> show @@datasource;
```

dn	ds	name	type	status	host	port	schema
186	227	192.168.210.68_3307_db252	1	1	192.168.210.68	3307	db252
	0	512	512	NULL	0/64		
186	228	192.168.210.68_3308_db252	3	1	192.168.210.68	3308	db252
	0	512	512	NULL	0/64		
1	7	192.168.210.41_3307_db252	1	1	192.168.210.41	3307	db252
	0	512	512	NULL	0/64		
2	8	192.168.210.42_3307_db252	1	1	192.168.210.42	3307	db252
	0	512	512	NULL	0/64		
101	74	192.168.210.41_3307_clu_db30	1	1	192.168.210.41	3307	clu_db30
	0	512	512	NULL	0/64		

...more contents are omitted, and you could log in to view...

The content behind show @@ command is a table name, for example in the previous instance, “show @@datasource;”, datasource is a table name.

The user could also make DESC operation of the table name behind show @@ command, to view meanings of various fields in this table, such as viewing the meaning of various fields in data source information:

```
mysql> desc datasource;
```

fieldname	description
dn	Datanode
ds	Datasource
name	Datasource name
type	Datasource type
status	Datasource status
host	Host
port	Port
schema	Physical database name
active	Active connections
idle	Idle connections
size	Total connections
unavailable_reason	Reason for datasource unavailable
flow_control	Remaining available quantity in flow control
idc_id	ID of IDC
listener_id	LISTENER ID(ID of LISTENER)
listener_status	LISTENER STATUS(STATUS of LISTENER)

16 rows in set (0.00 sec)

The user could also make SELECT operation of the table name behind show @@ command, to make SQL query under arbitrary condition, such as viewing the data source on No. 11 data node:

```
mysql> select * from datasource where dn=11;

+-----+-----+-----+-----+-----+-----+-----+
| dn   | ds   | name          | type | status | host      | port | schema | active | idle |
+-----+-----+-----+-----+-----+-----+-----+
| 11   | 9    | 192.168.210.70_3310_yds_db1 | 1     | 1       | 192.168.210.70 | 3310 | yds_db1 | 2     |
| 32   | 34   | NULL          | 0/16 |         |           |       |         |        |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

4.4. Limit on number of front-end connections

Compute node supports the function of Limit on Number of Front-end Connections, which could provide guarantee in case of access overload, and the function is the same as MySQL. The use method is: make configuration in server.xml:

```
<property name="maxConnections">5000</property><!-- Front maximum connections -->
<property name="maxUserConnections">0</property><!-- User's front maximum connections, Unlimited: 0 -->
```

maxConnections is front-end Max Connections, 5000 by default;

maxUserConnections is front-end max user connections, the default 0 means unlimited;

Meanwhile, it supports the user with super privilege to Modify on set global

```
max_connections=1; set global max_user_connections=0 service port;.
```

Show variables could be used to view limit conditions of current connections.

4.5. Management of back-end connection pool

Compute node will make connection with data source during the start and running, and when [Add Data Source](#), four configurations could be used to control the number of connections:

- Max Connections: Max Connections available between compute node and data source (that is MySQL physical database), SQL can't come into normal execution if exceeded;
- Initial Connections: Initial Connections established between compute node and data source;
- Min Idle Connections: Min Idle Connections established between compute node and data source;
- Max Idle Connections: Max Idle Connections established between compute node and data source.

When the Timing Detection Thread finds that the Idle Connections in the connection pool are smaller than the Min Idle Connections, then Create Connection; if bigger than Max Idle Connections, then Disable Connection. That is: $\text{Min Idle Connections} \leq \text{Idle Connections in connection pool} \leq \text{Max Idle Connections}$, and the Max and Min Idle Connections are mainly used for controlling Idle Connections in connection pool within a certain range.

For example:

Take single compute node as the unit for instance (multiple compute node service will multiply according to the number of compute nodes), HotDB Server data source configuration: Max Connections is 4200, Initial Connections is 32, Min Idle Connections is 64, Max Idle Connections is 512

Then, when HotDB Server compute node is enabled, it will establish 32 back-end connections with each data source, and when Timing Detection Thread detects that the current connections are less than the Min Idle Connections, then the connections shall be increased to the 64 Min Idle Connections;

At this time, if there is a 2048 Concurrent Pressurizing Scenario to impose pressure to compute node, it will be found that there are insufficient available connections in the connection pool, and the compute node will automatically increase the connections with data source, and the Max Connections could be up to 4200.

After Pressurizing, these connections will not be destroyed immediately. Instead, they will wait until examination by Idle Examination Period: if Idle (that is the management port show @@backend is marked as Idle) connections are bigger than 512, then the connections shall be destroyed to 512; if smaller than 512, they will be kept intact;

If to make Idle Connections back to Initial Connections, then during running process of compute node, you could rebuild connection pool by referring to rebuild connection pool rebuild @@pool related chapters in *Distributed Transactional Database HotDB Server [management port Command] Function Manual*, then Initial Connections status shall be recovered.

4.6. Limit on use of disk space

Since insufficient disk space would result in many problems such as the compute node can't come into normal operation, etc., therefore, at the time of Auto Deploy, the

compute node will detect whether free disk space of installation directory of HotDB Server is bigger than 10G or not, meanwhile, it will also detect the free disk space when writing temp-files. The details are as follow:

When the user conducts operations such as SQL Query Insert, etc. after Create Session, the compute node will write temp-files under installation directory of HotDB Server. While writing temp-files, if the compute node detects that the free disk space of the installation directory is insufficient, then it will end the current session and report error and record log.

Error-level logs recorded by compute node logs are as follow, and when End Session, the prompt message is the same with it:

```
2019-06-10 18:03:24.423 [ERROR] [DISKSPACE] [Employee-2]
cn.hotpu.hotdb.mysql.nio.handler.MultiNodeHandler(88) - session[1606] was killed,due to less than 1G space left
on device,and the size of temp-file is larger than the usable space.
```

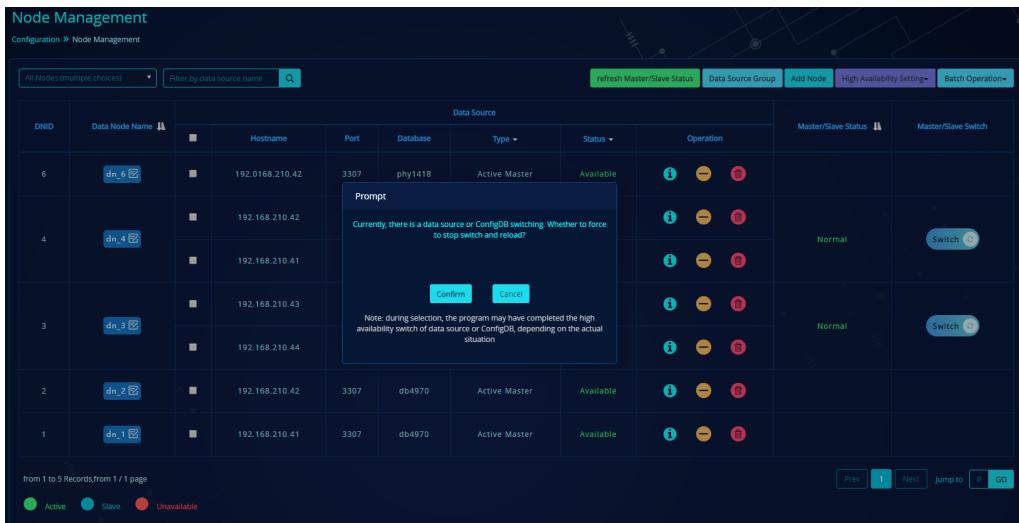
4.7. Reload

Compute node could make Online Reload of configuration information without re-enabling the service. For the parameters which could take effect immediately via “Reload” function, please refer to [Instruction on use of compute node parameters](#).

There are two Reload methods, one is to log in to [management port \(3325\)](#) to execute: reload @@config command; the other is to log in to management platform, click “Reload” button on top right corner of the menu bar, and reload the new configuration items to the compute node for use. As shown in the following figure:

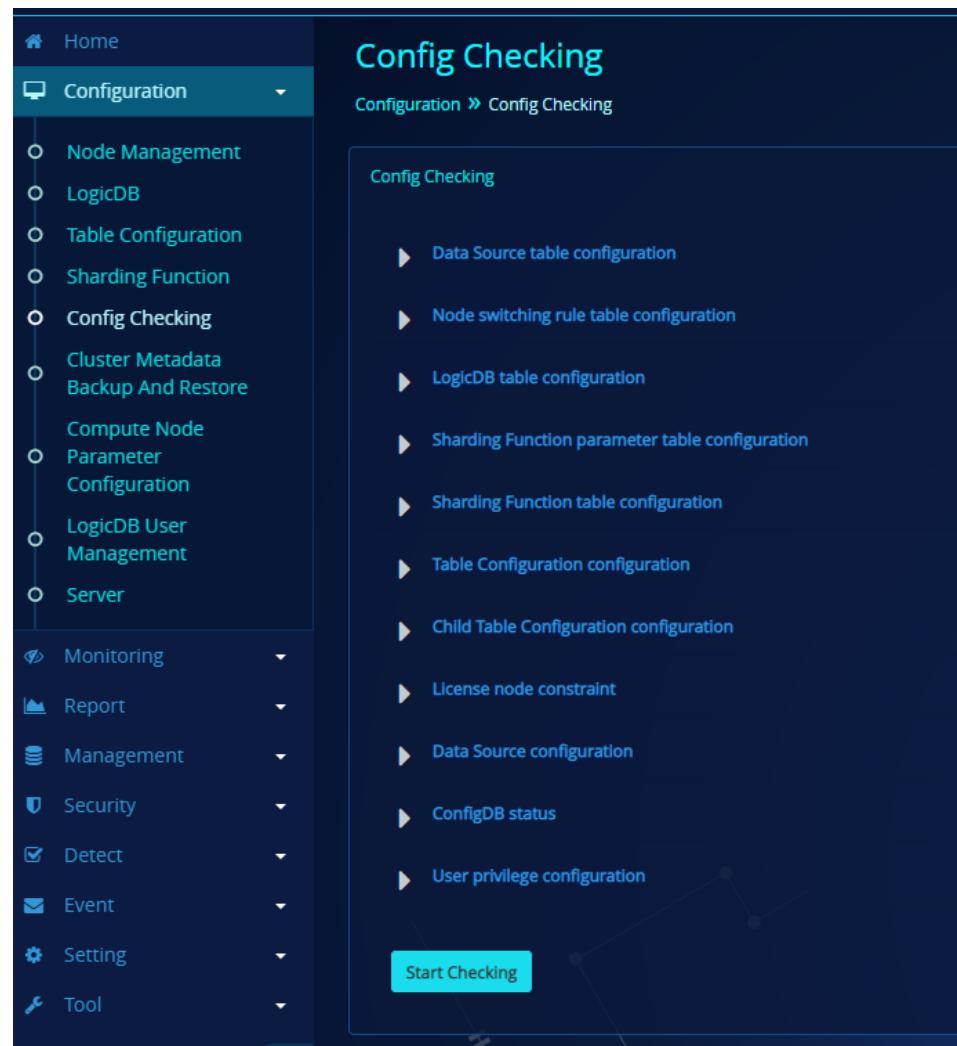


In order to guarantee that compute node makes accurate loading of configuration information, before executing Reload, configuration information could be checked first. During Reload process, in case of master/slave configDB, master/slave data source switching, it will give prompt to the user and provide two optional schemes: force to stop switch and cancel reload.



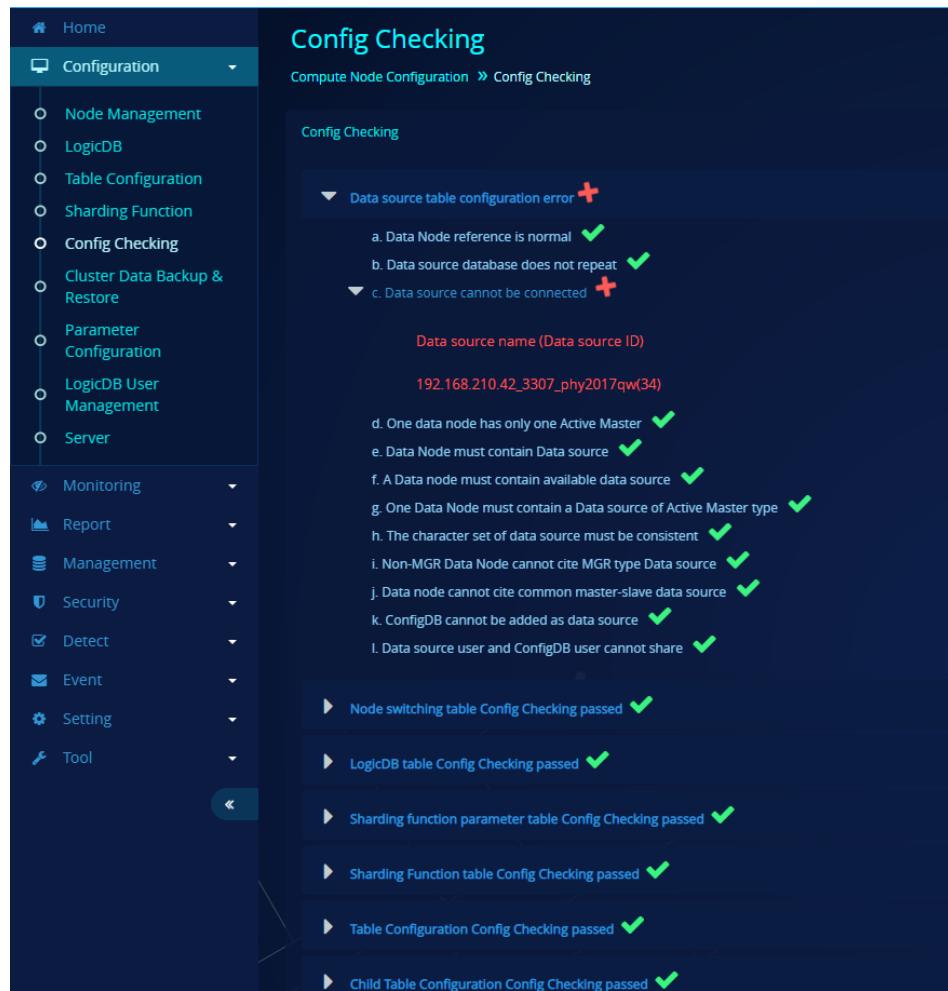
4.8. Config Checking

Log in to management platform, select “Configuration”->“Config Checking” to enter the Config Checking panel, click “Start Checking” button, and then it will check the configuration items in “Config Checking” menu of Distributed Transactional Database Management Platform, as shown in the following figure:



As shown in the above figure, all configuration items have passed the checking normally.

In case of inaccurate configuration items, modify correspondingly according to the error prompt:



When executing reload @@config command to reload via compute node management port, Config Checking will be conducted first as well by default, and successful reload is allowed only after the checking passed.

4.9. Deadlock Check

In Distributed Transactional Database system, in case of deadlock between two data sources under data node, then Deadlock Check mechanism of MySQL can't detect the deadlock.

The operations in the following table, describe the deadlock process of two data nodes in distributed system. Session 1 and Session 2 execute DELETE operation respectively on the two data nodes:

	Session 1	Session 2
Session 1 Start transaction	start transaction;	
Session 2 Start transaction		start transaction;
Session 1 Execute DELETE statement on data node where DNID = 15	delete from customer where dnid=15 and id=1;	
Session 2 Execute DELETE statement on data node where DNID = 13		delete from customer where dnid=13 and id=4;
Session 1 Execute DELETE statement on data node where DNID = 13; DELETE operation will be blocked by Session 2	delete from customer where dnid=13 and id=4;	
Session 2 Execute DELETE statement on data node where DNID = 15; this operation will be blocked by Session 1; since Session 1 is blocked by Session 2,		delete from customer where dnid=15 and id=1;

and Session 2 is also blocked by Session 1, therefore, there will be deadlock at this time		
--	--	--

Under the above condition, Session 1 and Session 2 are mutually blocked, thus there will be deadlock. Since the deadlock occurs between two data sources of data node, MySQL can't detect existence of the deadlock.

In Distributed Transactional Database system of HotDB Server, compute node could detect deadlock among multiple data sources of data node, and rollback the transaction of least overheads.

In Config File server.xml of compute node, set Deadlock Check Period at the value bigger than 0, which will enable Deadlock Auto Check function. By default, Deadlock Check is enabled, and the Check Period is 3000ms.

```
<property name="deadlockCheckPeriod ">3000</property>
```

When the value of deadlockCheckPeriod is set as 0, Deadlock Check function will not start.

When Start Deadlock Check of compute node, re-execute the above-mentioned DELETE operation:

Session 1, start transaction:

```
mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)
```

Session 2, start transaction:

```
mysql> start transaction;
```

```
Query OK, 0 rows affected (0.00 sec)
```

Session 1, Execute DELETE statement on data node where DNID = 15:

```
mysql> delete from customer where dnid=15 and id=1;
```

```
Query OK, 1 row affected (0.00 sec)
```

Session 2, Execute DELETE statement on data node where DNID = 13

```
mysql> delete from customer where dnid=13 and id=4;
```

```
Query OK, 1 row affected (0.00 sec)
```

Session 1, Execute DELETE statement on data node where DNID = 13; DELETE operation will be blocked by Session 2:

```
mysql> delete from customer where dnid=13 and id=4;
```

Session 2, Execute DELETE statement on data node where DNID = 15; this operation will be blocked by Session 1; since Session 1 is blocked by Session 2, and Session 2 is also blocked by Session 1, therefore, there will be deadlock at this time

```
mysql> delete from customer where dnid=15 and id=1;
```

```
Query OK, 1 row affected (1.59 sec)
```

Compute node checks deadlock, and rolls back to transaction in Session 1:

```
mysql> delete from customer where dnid=13 and id=4;
```

```
ERROR 1213 (HY000): Deadlock found when trying to get lock; try restarting transaction
```

Note: In MySQL 5.7 and above, a new transaction will not be started immediately after a deadlock rollback occurs in the transaction. You can refer to the official BUG link: <https://bugs.mysql.com/bug.php?id=98133>. HotDB Server does compatibility processing for the above BUGs: for the lock timeout, deadlock detection, and back-end

disconnection, MySQL 5.7 and above will determine whether to start a new transaction according to the front-end connection of autocommit.

4.10. SQL error reporting log

If the following error information is returned during SQL execution, the compute node will record it in the compute node log (hotdb-unusualsql.log):

- ERROR information caused by primary key\unique key conflict or foreign key constraint not satisfied (i.e. MySQL error code 1062, 1216, 1217, 1451, 1452, 1557, 1761, 1762, 3008)
- Data overflow (i.e. MySQL error code 1264, 1690, 3155, 3669) and data truncation (i.e. MySQL error code 1265, 1292, 1366) caused by data type conversion or implicit conversion
- Involving binlog unsafe statements (i.e. MySQL error codes 1418, 1592, 1663, 1668, 1669, 1671, 1673, 1674, 1675, 1693, 1714, 1715, 1716, 1719, 1722, 1724, 1727, 1785, 3006, 3199, 3570, 3571, MY-010908, MY-013098)
- When the INSERT operation is performed on the sharding table whose sharding key is not an auto-incremental field, the INSERT of auto increment is specified externally.
- Mismatch between MATCH and AFFECT in UPDATE statement
- Zero rows are deleted in DELETE
- UPDATE or DELETE AFFECT are more than 10000 rows
- INSERT, UPDATE, DELETE, and DDL statements are used in HINT statements
- ERROR information is reported during the execution of DDL statements. In addition, there will be additional logging in the following two cases: 1. When the character set of the specified table or field in CREATE TABLE or ALTER TABLE

is inconsistent with the character set of the data source or the character set used in the connection; 2. In the sharding table of CREATE TABLE or ALTER TABLE, the primary key or unique key does not contain the sharding key and does not use the global unique constraint to ensure field uniqueness.

- A partial commit occurs, that is, when some nodes have issued a commit, the rest nodes have not issued a commit but the connection is stopped, or when the rest of the back-end connections have issued a commit, there is no response and the connection is stopped. In the above cases, the whole transaction will be recorded to the compute node log.
- A syntax error occurs.
- Execution of SQL that cannot be routed due to lack of routing rules, such as ROUTE rules that do not exist in INSERT
- Execution of SQL blocked by SQL firewall
- Execution of SQL which timed out
- Transaction killed due to the dead lock
- Transaction killed due to data source switching and other reasons
- Execution of SQL with lock which timed out and rolled back
- SQL killed after Kill Command is executed
- SQL being rolled back
- SQL rolled back caused by abnormal front-end connection and connection stopping
- Rollback caused by abnormal back-end connection and connection stopping or other exceptions
- Unexpected throwing of exception by compute node

- If the above SQLs are too long, the SQL statements will be intercepted, and the WARNING information will be additionally recorded

For example, execute a SQL with primary key conflict as follows:

```
mysql> insert into table01 (id,title,author,submission_date) values (3,"apple",
"apple pie", '2019-10-11-20-05');

ERROR 1062 (23000): Duplicate entry '3' for key 'PRIMARY'
```

View the compute node log (hotdb-unusualsql.log) :

```
2019-10-12 15:27:45.051 [INFO] [UNUSUALSQL] [$NIOREACTOR-7-RW]
cn.hotpu.hotdb.mysql.nio.MySQLConnection(415) - ERROR 1062:Duplicate entry '3' for
key 'PRIMARY' [frontend:[thread=$NIOREACTOR-7-
RW,id=453,user=root,host=192.168.210.225,port=3323,localport=65442,schema=DBY];
backend:null; frontend_sql:insert into table01 (id,title,author,submission_date)
values (3,"apple", "apple pie", '2019-10-11-20-05');backend_sql:null]
```

For another example, execute a SQL intercepted by SQL firewall as follows:

```
mysql> select * from test;

ERROR 1064 (HY000): Intercepted by sql firewall, because: not allowed to execute
select without where expression
```

View the compute node log (hotdb-unusualsql.log) :

```
2019-10-14 15:41:42.246 [INFO] [UNUSUALSQL] [$NIOExecutor-1-2]
cn.hotpu.hotdb.route.RouteService(415) - ERROR 10029:not pass sql firewall
[frontend:[thread=$NIOExecutor-1-
2,id=1433,user=root,host=192.168.210.225,port=3323,localport=64658,schema=DBY];
backend:null; frontend_sql:null; backend_sql:null] [DBY.count]=33
```

Note:

For MySQL error code explanations, please refer to the official document;

<https://dev.mysql.com/doc/refman/8.0/en/server-error-reference.html>

By default, this type of log information is saved in the file hotdb_unusualsql.log under the HotDB Server installation directory /logs/extra/unusualsql/. If the log is not recorded to a file, check whether the following configuration exists under log4j2.xml in the HotDB Server installation directory /conf:

```
</RollingFile>

<RollingFile

    name="Unusualsql"

    filename="${sys:HOSTDB_HOME}/logs/extra/unusualsql/hotdb-
unusualsql.log"

    filepattern="${sys:HOSTDB_HOME}/logs/extra/unusualsql/hotdb-
unusualsql-%d{yyyy-MM-dd-HH-mm-ss}.log">

    <PatternLayout

        pattern="%d{yyyy-MM-dd HH:mm:ss.SSS} [%-4p] [%marker] [%t] %c(%L)
- %msg%n"/>

    <Policies>

        <SizeBasedTriggeringPolicy size="100 MB"/>

    </Policies>

    <!-- only record unusual sql log -->

    <filters>

        <MarkerFilter marker="UNUSUALSQL" onMatch="ACCEPT"
onMismatch="DENY"></MarkerFilter>

    </filters>

    <DefaultRolloverStrategy max="1000"/>

</RollingFile>
```

```
</Appenders>

<Loggers>

<Root level="info">

    <AppenderRef ref="Unusualsql" />
```

5. Safety

5.1. User and privilege

HotDB Server has two kinds of users, one is Compute Node User who operates data, and executes SELECT, UPDATE, DELETE, INSERT and other SQL statements. The other is Distributed Transactional Database Management Platform User, who manages the configuration information. This chapter will emphasize on introducing compute node user related contents.

LogicDB must be accessed with granted compute node user privilege. Compute node provides LogicDB with operating privilege similar to that of MySQL, as follow:

Privilege type	Executable SQL statement
CREATE	CREATE TABLE,CREATE INDEX
DROP	DROP TABLE,DROP INDEX,TRUNCATE TABLE,RENAME TABLE
ALTER	ALTER TABLE,RENAME TABLE
SELECT	SELECT,INSERT…SELECT
UPDATE	UPDATE
DELETE	DELETE,REPLACE
INSERT	INSERT,REPLACE,INSERT…SELECT
SUPER	management port statement, /*!HotDB:dnid=?*/

Description of SUPER privilege:

The user with SUPER privilege, could log in to 3325port of compute node, and could execute all SQL statements of management port; otherwise, it can't log in to

management port or execute SQL statements of management port.

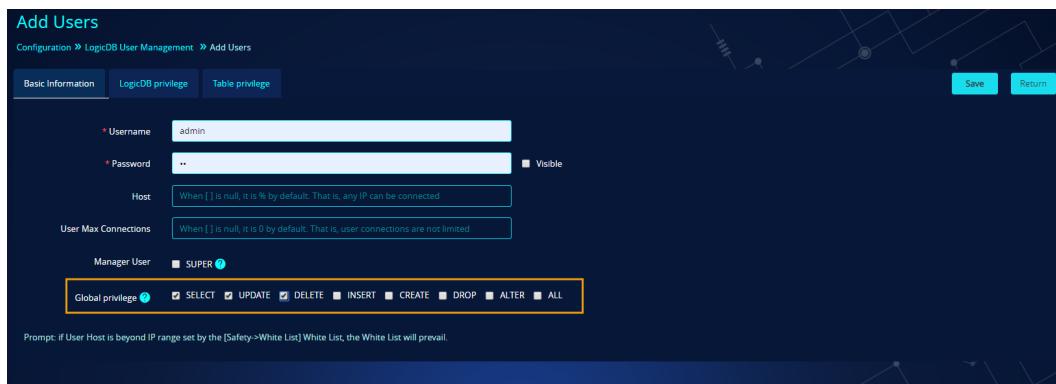
The user with SUPER privilege could execute hint statement at 3323port. For example:

```
/*!hotdb:dnid=1*/select * from table
```

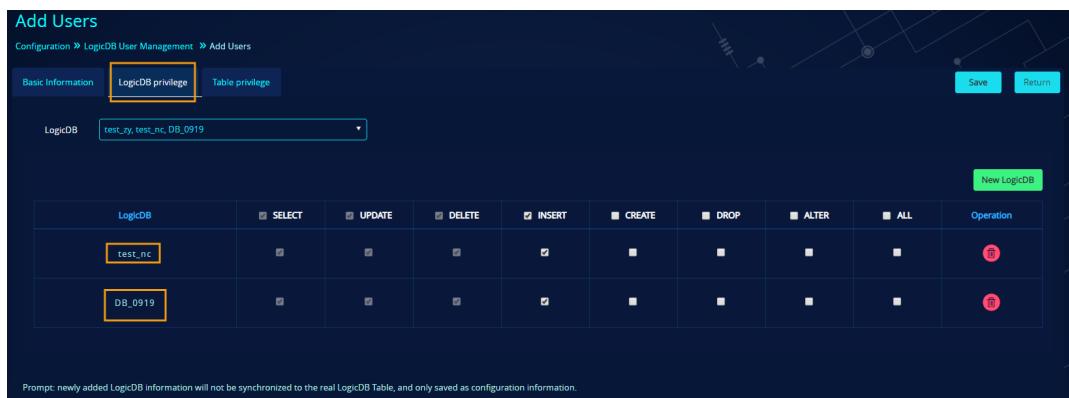
Privilege range:

When granting privilege for compute node user, in addition to SUPER privilege, operating privilege of the user toward LogicDB or table will be specified. Privilege range is divided into Global Privilege, LogicDB Privilege and Table Privilege:

- Global privilege: the user with Global Privilege has the privilege to specify all objects under LogicDB. For example: if check Global privilege: SELECT, UPDATE, INSERT, CREATE, and click Save, then the current user could make S/U/I/C operation toward all LogicDB and tables.

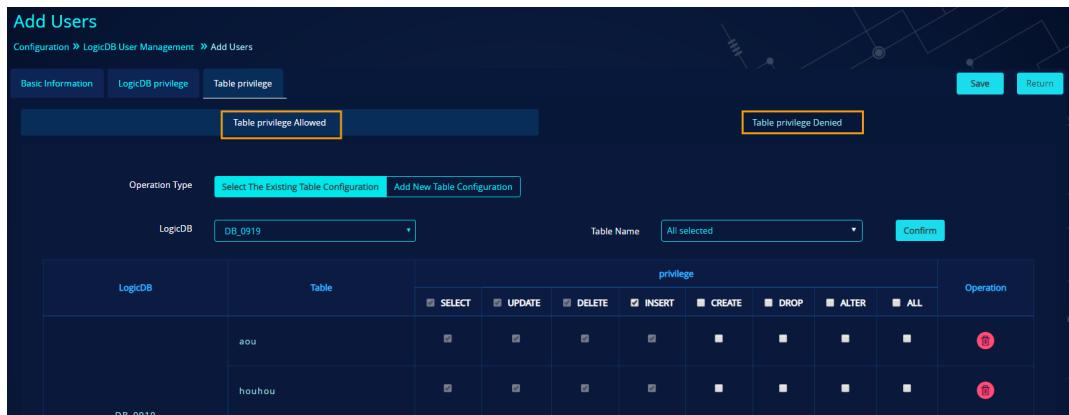


- LogicDB privilege: the user with LogicDB Privilege has the privilege to specify all objects under LogicDB.



Prompt: newly added LogicDB information will not be synchronized to the real LogicDB Table, and only saved as configuration information.

- Table privilege: Table privilege is also divided into Table privilege Allowed and Table privilege Denied. The user with Table privilege Allowed owns the privilege to check the table; the user with Table privilege Denied owns all privileges other than check privilege toward the table; for example: check Table privilege Denied: SELECT, UPDATE, INSERT, CREATE, and click Save, then the current user can't make S/U/I/C operation of the table, but has the privilege to DELETE, DROP, ALTER.



SUPER privilege doesn't specify specific LogicDB. Only the user with SUPER privilege could execute management port statements, please refer to the chapter [management port Information Monitoring](#) for detailed functions of management port.

The privileges are mutually independent, and owning the Table UPDATE privilege doesn't mean owning the Table SELECT privilege; owning the SUPER privilege doesn't mean owning the Table Operation privilege. Besides, TRIGGER related privilege is not separately maintained at present, and the privilege rule observed is:

CREATE TRIGGER requires CREATE privilege; DROP TRIGGER requires DROP privilege; TRIGGER internal statement doesn't verify privilege; DEFINER related are all removed; In case of SHOW TRIGGERS, the related field shall be the current user.

5.2. SSL authentication

Introduction: SSL (Secure Socket Layer) is a protocol encryption layer under HTTPS, with 1, 2 and 3 versions. Currently, we use SSL 3.0. After IETF standardizes SSL, TLS1.0 (Transport Layer Security) is released based on SSL 3.0. TLS protocol now has 1.0, 1.1, 1.2, 1.3 four versions.

Since HotDB-Server v.2.5.5, SSL encrypted connection mode to log in to the compute node has been supported.

5.2.1. Generation of TLS secret key

5.2.1.1. Generation of the certificate and key files

Please refer to the official MySQL documents to generate a self-signed secret key. For example, you can generate the certificate and key files using MySQL's own command `mysql ssl rsa setup`.

Among them, the secret keys required by the client are: ca.pem, client-cert.pem, client-key.pem;

The secret keys required by HotDB are: ca.pem、server-cert.pem、server-key.pem;

Note: the certificate generated by the MySQL command cannot be CA certified, please refer to the link: <https://dev.mysql.com/doc/refman/5.7/en/using-encrypted-connections.html>

Note

Host name identity verification with `VERIFY_IDENTITY` does not work with self-signed certificates that are created automatically by the server or manually using `mysql_ssl_rsa_setup` (see Section 6.3.3.1, "Creating SSL and RSA Certificates and Keys using MySQL"). Such self-signed certificates do not contain the server name as the Common Name value.

Host name identity verification also does not work with certificates that specify the Common Name using wildcards because that name is compared verbatim to the server name.

If you need to generate a self-signed certificate capable of CA authentication, you need to use openssl. Please refer to the following steps:

1. Generate CA root certificate private key: `openssl genrsa 2048 > ca-key.pem`
2. Generate CA root certificate: `openssl req -new -x509 -nodes -days 3600 -key ca-key.pem -out ca.pem`. Please note that in the information filling step, Common Name is better to be filled in the valid domain name, and the name cannot be the same as Common Name in the issued certificate. Here we fill in 127.0.0.1
3. Generate server certificate request file: `openssl req -newkey rsa:2048 -days 3600 -nodes -keyout server-key.pem -out server-req.pem`. Please note that in the information filling step, Common Name needs to be filled in the IP address/domain name monitored by HotDB-Server, and the client will use this IP for service connection. It cannot be the same as the information in CA certificate.
4. Use command `openssl rsa` to process the secret key to delete the password:
`openssl rsa -in server-key.pem -out server-key.pem`
5. Generate the self signed certificate for the server: `openssl x509 -req -in server-req.pem -days 3600 -CA ca.pem -CAkey ca-key.pem -set_serial 01 -out server-cert.pem`

```

6. Generate the client certificate request file: openssl req -newkey rsa:2048 -days
3600 -nodes -keyout client-key.pem -out client-req.pem. Note that the Common Name
in the information filling step cannot be the same as the information in the CA
certificate.

7. Use command openssl rsa to process the secret key to delete password: openssl
rsa -in client-key.pem -out client-key.pem

8. Generate self signed certificate for client: openssl x509 -req -in client-
req.pem -days 3600 -CA ca.pem -CAkey ca-key.pem -set_serial 01 -out client-cert.pem

```

5.2.1.2. Generation of server.jks file

For a compute node, the secret key needs to be converted to a Java-standard KeyStore file. That is .jks as mentioned below. The generation steps are:

1) First, synthesize cert and key files into pfx files using openssl:

In this example, SDcrtest should be entered as password (the password of the key file that comes with the program is hotdb.com, which can be used directly. This example is for when generation of a new secret key is required.)

```
openssl pkcs12 -export -out server.pfx -inkey server-key.pem -in server-cert.pem -
CAfile ca.pem
```

Enter password SDcrtest

```
[root@drds-240-117 crt]# openssl pkcs12 -export -out server.pfx -inkey server-key.pem -in server-cert.pem -CAfile ca.pem
Enter Export Password:
Verifying - Enter Export Password:
[root@drds-240-117 crt]# ll
total 36
-rw----- 1 root root 1679 May  8 15:06 ca-key.pem
-rw-r--r-- 1 root root 1107 May  8 15:06 ca.pem
-rw-r--r-- 1 root root 1107 May  8 15:06 client-cert.pem
-rw----- 1 root root 1679 May  8 15:06 client-key.pem
-rw----- 1 root root 1675 May  8 15:06 private_key.pem
-rw-r--r-- 1 root root  451 May  8 15:06 public_key.pem
-rw-r--r-- 1 root root 1107 May  8 15:06 server-cert.pem
-rw----- 1 root root 1679 May  8 15:06 server-key.pem
-rw-r--r-- 1 root root 2381 May  8 15:13 server.pfx
```

2) Convert pfx to jks file using keytool provided by Java:

```
keytool -importkeystore -srckeystore server.pfx -destkeystore server.jks -
srcstoretype PKCS12
```

Enter password SDcrtest

```
[root@drds-240-117 crt]# keytool -importkeystore -srckeystore server.pfx -destkeystore server.jks -srcstoretype PKCS12
Enter destination keystore password:
Re-enter new password:
Enter source keystore password:
Entry for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled
[root@drds-240-117 crt]# ll
total 40
-rw----- 1 root root 1679 May  8 15:06 ca-key.pem
-rw-r--r-- 1 root root 1107 May  8 15:06 ca.pem
-rw-r--r-- 1 root root 1107 May  8 15:06 client-cert.pem
-rw----- 1 root root 1679 May  8 15:06 client-key.pem
-rw----- 1 root root 1675 May  8 15:06 private_key.pem
-rw-r--r-- 1 root root  451 May  8 15:06 public_key.pem
-rw-r--r-- 1 root root 1107 May  8 15:06 server-cert.pem
-rw-r--r-- 1 root root 2124 May  8 15:19 server.jks
-rw----- 1 root root 1679 May  8 15:06 server-key.pem
-rw-r--r-- 1 root root 2381 May  8 15:13 server.pfx
```

5.2.2. Configuration of TLS secret key

After the TLS secret key is generated, the corresponding secret key file should be transferred to the server where the server and client of the compute node are located and configured with the following three parameters in the compute node as required before using:

```
<property name="enableSSL">false</property><!-- Enable SSL connection or not -->
```

Parameter description: true means to enable SSL function; false means to disable SSL function; default value is false.

```
<property name="keyStore">/server.jks</property><!-- Path to the data
```

certificate .jks file for TLS connection -->

Parameter description: a set of pem file related to server.jks and client is provided by default by compute nodes under /conf directory, with password of hotdb.com, which can be used for simple connection testing. When you choose to use your own generated TLS certificate or pay-for-use TLS certificate to connect, you need to fill in according to the actual path and name. For example, /usr/local/crt/server.jks.

```
<property name="keyStorePass">BB5A70F75DD5FEB214A5623DD171CEEB</property><!--
```

Password of the data certificate .jks file for TLS connection -->

Parameter description: the password in the key file that comes with the program is hotdb.com, which can be encrypted by users through select hex(aes_encrypt('hotdb.com',unhex(md5('Hotpu@2013#shanghai#2017')))); to get the default value BB5A70F75DD5FEB214A5623DD171CEEB and fill the value in keyStorePass. If users use their own generated key file, the value be filled is based on the password which is actually entered. If SDcrtest is entered as password, users can get the value of keyStorePass through select hex(aes_encrypt('SDcrtest',unhex(md5('Hotpu@2013#shanghai#2017')))) and fill the value C43BD9DDE9C908FEE7683AED7A301E33 in keyStorePass.

The configured parameters are as follows:

```
<property name="enableSSL">true</property><!-- 是否开启SSL连接功能(Enable SSL connection or not) -->
<property name="keystore"/>/usr/local/crt/server.jks</property><!-- 用于TLS连接的数据证书.jks文件的路径(Path to the data certificate .jks file for TLS connection) -->
<property name="keyStorePass">c43bd9dde9c908fee7683aed7a301e33</property><!-- 用于TLS连接的数据证书.jks文件的密码>Password of the data certificate .jks file for TLS connection -->
```

Users have no need to restart the compute node service for the parameter modification, for server.jks documents will be read again during dynamic loading. If SSL-related logic initialization fails, the dynamic loading will not fail, though the subsequent SSL connections cannot be established normally. Non-SSL connections will not be affected.

Notes:

- If the compute node cannot find any available server.jks file, the following error messages will be output when starting or synchronously loading.

- If the keyStorePass configuration is wrong, the following error messages will be output during startup or synchronously loading.

```
2020-05-08 16:39:23.286 [INFO] [INIT] [$NIOExecutor-2-1] cn.hutox.hotdb.a(2356) - using config datasource in start up:[id:-1,nodeId:-1 192.168.240.117:3306/hotdb_config]
2020-05-08 16:39:23.295 [INFO] [INIT] [$NIOExecutor-2-1] cn.hutox.hotdb.a(803) - master config datasource [id:-1,nodeId:-1 192.168.240.117:3306/hotdb_config status:s1.charset:utf8] connect success.
2020-05-08 16:39:23.343 [WARN] [INNER] [$!-NIOREACTOR-6-RW] cn.hutox.hotdb.mysql.handler.Z256 - Datasource: 2 isn't a slave of Datasource: 1 in HotDB_Config hotdb_follower.
2020-05-08 16:39:23.659 [ERROR] [SSL] [Labor-10] cn.hutox.hotdb.a(680) - wrong password for /server_jks
2020-05-08 16:39:23.662 [INFO] [INIT] [Labor-10] cn.hutox.hotdb.a(707) - datanode 1 persist current datasource 1 to config database.
2020-05-08 16:39:23.672 [INFO] [TThread-141] cn.hutox.hotdb.a(3189) - clear datanode 1 reason : below
```

- If the certificate configuration is wrong, the following error messages will be output during login

```
[root@ndc-240-117 ~]# mysql -uroot -p123456 -h192.168.240.117 -p32333 --ssl=ca=/usr/local/hotdb/hotdb-server/conf/ca.pem --ssl-cert=/usr/local/hotdb/hotdb-server/conf/client-key.pem --ssl-mode=verify_ca  
mysql: [Warning] Using a password on the command line interface can be insecure.  
ERROR 2026 (HY000): SSL connection error: protocol version mismatch
```

5.2.3.1. MySQL clients

5.2.3.1. MySQL clients

For MySQL clients, users can specify a secret key file to connect using the following method:

```
mysql -ujing01 -p123456 -h192.168.240.117 -P3323 --ssl-ca=/usr/local/crt/ca.pem --  
ssl-cert=/usr/local/crt/client-cert.pem --ssl-key=/usr/local/crt/client-key.pem --  
ssl-mode=verify_ca
```

```
[root@drdb-240-117 conf]# mysql -ujing01 -p123456 -h192.168.240.117 -P3323 --ssl-ca=/usr/local/crt/ca.pem --ssl-cert=/usr/local/crt/client-cert.pem --ssl-key=/usr/local/crt/client-key.pem >ssl-modifycertify.ca
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 44
server version: 5.7.25-HotDB-2.5.5 HotDB Server by Hotpu Tech
copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
jing01@192.168.240.117:(none) 5.7.25-HotDB-2.5.5 03:45:57> ■
```

Check whether SSL is enabled:

```
jing01@192.168.240.117:(none) 5.7.25-HotDB-2.5.5 03:45:57> \s
-----
mysql Ver 14.14 Distrib 5.7.25, for linux-glibc2.12 (x86_64) using EditLine wrapper

Connection id: 44
Current database:
Current user: jing01@192.168.240.117
SSL: Cipher in use is DHE-RSA-AES128-SHA
Current pager: stdout
Using outfile:
Using delimiter:
Server version: 5.7.25-HotDB-2.5.5 HotDB Server by Hotpu Tech
Protocol version: 10
Connection: 192.168.240.117 via TCP/IP
Server characterset: utf8mb4
Db characterset: utf8mb4
Client characterset: utf8mb4
Conn. characterset: utf8mb4
TCP port: 3323
Uptime: 1 hour 56 min 32 sec

Threads: 6 Questions: 0 Slow queries: 0 Opens: 0 Flush tables: 0 Open tables: 0 Queries per second avg: 0
-----
```

5.2.3.2. JDBC

For JDBC, the corresponding key file is also required. Please refer to the [official MySQL manual](#). Here are two ways for reference:

1) By importing CA into Java trust store:

```
keytool -importcert -alias MySQLCACert -file ca.pem -keystore truststore

[root@hotdb_210_71 tmp]# keytool -importcert -alias MySQLCACert -file ca.pem -keystore truststore
Enter keystore password:
Re-enter new password:
Owner: CN=MySQL_Server_5.7.25_Auto_Generated_CA_Certificate
Issuer: CN=MySQL_Server_5.7.25_Auto_Generated_CA_Certificate
Serial number: 1
Valid from: Fri Mar 27 18:28:25 CST 2020 until: Mon Mar 25 18:28:25 CST 2030
Certificate fingerprints:
MD5: F0:91:C8:5E:E7:41:7A:CC:30:84:A0:41:A7:B0:24:00
SHA1: E8:59:F1:35:99:72:B9:46:74:4D:19:06:5D:E7:4D:90:19:5E:71:D1
SHA256: 20:DF:87:B7:D8:69:F2:A6:78:68:7F:56:B9:AE:9B:3B:07:C0:73:F6:39:40:A9:93:BE:AC:32:81:41:4E:21:89
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:
#1: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]
Trust this certificate? [no]: yes
Certificate was added to keystore
[root@hotdb_210_71 tmp]# ls
ca-key.pem  ca.pem  client-cert.pem  private_key.pem  public_key.pem  server-cert.pem  server.jks  server-key.pem  server.pfx  truststore
```

The truststore file is used for JDBC connection:

```
jdbc:mysql://192.168.240.117:3323/smoketest?clientCertificateKeyStoreUrl=file:/usr/
local/crt/truststore&clientCertificateKeyStorePassword=hotdb.com&verifyServerCertif
icate=true
```

2) By using a certificate:

```
openssl pkcs12 -export -in client-cert.pem -inkey client-key.pem -name
"mysqlclient" -out client-keystore.p12

keytool -importkeystore -srckeystore client-keystore.p12 -srcstoretype pkcs12 -
destkeystore keystore -deststoretype JKS
```

```
[root@hotdb_210_71 tmp]# openssl pkcs12 -export -in client-cert.pem -inkey client-key.pem -name "mysqlclient" -out client-keystore.p12
Enter Export Password:
Verifying - Enter Export Password:
[root@hotdb_210_71 tmp]# keytool -importkeystore -srckeystore client-keystore.p12 -srcstoretype pkcs12 -destkeystore keystore -deststoretype JKS
Enter destination keystore password:
Re-enter new password:
Enter source keystore password:
Entry for alias mysqlclient successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled
[root@hotdb_210_71 tmp]# ll
total 52
-rw----- 1 root root 1675 Mar 27 18:28 ca-key.pem
-rw-r--r-- 1 root root 1107 Mar 27 18:28 ca.pem
-rw-r--r-- 1 root root 1107 Mar 27 18:28 client-cert.pem
-rw----- 1 root root 1679 Mar 27 18:28 client-key.pem
-rw-r--r-- 1 root root 2452 Mar 27 18:41 client-keystore.p12
-rw-r--r-- 1 root root 2135 Mar 27 18:41 keystore
-rw----- 1 root root 1679 Mar 27 18:28 private_key.pem
-rw-r--r-- 1 root root 451 Mar 27 18:28 public_key.pem
-rw-r--r-- 1 root root 1107 Mar 27 18:28 server-cert.pem
-rw-r--r-- 1 root root 2910 Mar 27 18:36 server.jks
-rw----- 1 root root 1679 Mar 27 18:28 server-key.pem
-rw-r--r-- 1 root root 3197 Mar 27 18:34 server.pfx
-rw-r--r-- 1 root root 843 Mar 27 18:40 truststore
```

The truststore file is used for JDBC connection:

```
jdbc:mysql://192.168.240.117:3323/smoketest?clientCertificateKeyStoreUrl=file:/usr/
local/crt/keystore&clientCertificateKeyStorePassword=hotdb.com
```

5.2.3.3. Navicat and other similar clients

For Navicat and other similar clients, users can configure the relevant file location in the client settings to connect:



Note: for some versions of Navicat, it may be unable to connect after checking the CA certificate name verification, which may be because the DLL of this version is not compatible. For example, the following warning: "2026 SSL connection error: ASN: bad other signature confirmation". In this case, you need to replace the "libmysql.dll" with files with the same name in MySQL Workbench, or update it to a higher version.

Please refer to the [link](#).

6. Data migration, backup and recovery

6.1. Use mysqldump backup

6.1.1. mysqldump - database backup program

HotDB Server supports mysqldump function, and the usage is the same with MySQL.

When using mysqldump to export data from compute node, it's required to Add the following specified parameters:

```
--set-gtid-purged=OFF --no-tablespaces --skip-triggers --single-transaction --default-character-set=utf8mb4 --  
complete-insert --compact --skip-tz-utc [--replace|--insert-ignore] [--hex-blob] [--where=xxx]
```

When using mysqldump to export data from MySQL and then import to compute node, it's required to Add the following parameters:

```
--no-defaults --no-tablespaces --complete-insert --default-character-set=utf8mb4 --hex-blob --master-data=2 --no-  
create-db --set-gtid-purged=OFF --single-transaction --skip-add-locks --skip-disable-keys --skip-triggers --skip-tz-  
utc [--replace|--insert-ignore] [--no-create-info|--no-data] [--where=xxx] --databases xxx
```

Note: Please fill in value of default-character-set parameter according to actual conditions, for example utf8 or utf8mb4, etc.

If specified parameters are not used, there may be problem of time difference, and error reporting of some function commands not supported.

6.2. Use binary log (increment) recovery

6.2.1. mysqlbinlog - utility program processing binary log files

Compute node supports parsing mysqlbinlog syntax to synchronize incremental data, in order to reduce downtime for migrating standalone MySQL data to compute node.

Use mysqlbinlog to execute SQL statement of a certain binlog file in MySQL connection, so as to import data from a certain database to a certain LogicDB of compute node. Firstly, log in to [management port](#) (default port: 3325), execute dbremapping command to Add database mapping relation, and for use method of dbremapping command, please refer to [*Distributed Transactional Database HotDB Server \[management port Command\] Function Manual*](#).

```
dbremapping @@add@ database name: LogicDB name expected to be imported
```

Then use mysqlbinlog statement to execute SQL statement in binlog of the selected part, and it's required to use the following syntax and parameters:

```
mysqlbinlog --base64-output=decode-rows --skip-gtids --to-last-log --stop-never --database= database name --start-position=binlog initial position binlog file name | mysql -u username -p password -h server -Pservice port -c -show-warnings=false
```

Note: --to-last-log could be replaced with --stop-position, specify end position of binlog instead of the latest execution position of binlog.

For example, if hoping to import physical database db01 in 192.168.200.77:3306 to LogicDB logicdb01 configured on the management platform, then the belonging compute node of the LogicDB is 192.168.210.30.

1. Firstly, go to 192.168.210.30 to log in to [management port 3325](#), and execute:

```
dbremapping @@add@db01:logicdb01
```

2. Then, go to 192.168.210.30 server to make remote login to 192.168.200.77 to execute the following command:

```
mysqlbinlog -R -h 192.168.200.77 -P3306 -v --base64-output=decode-rows --skip-gtids --to-last-log --stop-never --database=db01 --start-position=0 mysql-bin.000009 | mysql -uroot -proot -h192.168.210.30 -P3323 -c -A
```

6.2.2. Practical application of mysqldump and mysqlbinlog

This section will show how to in practical application scenario, combine complete backup of mysqldump with incremental backup of mysqlbinlog to migrate data from source-end standalone MySQL to compute node.

Notice: During the whole operation process, it's not recommended executing any DDL, parameter change and other irregular operations in source end or compute node of data migration. Due to single thread operation and restriction by network latency, the data-catching execution speed of this mode is slower than execution speed of MySQL replication. Therefore, it's not guaranteed that execution speed of compute node could meet real-time catching of data, and there may be increasing data latency. Therefore, at this time, find the transaction trough to retry or plan a scheme separately.

Scenario description: Hoping to import physical database db01 at source-end 192.168.210.45:3309 (this instance is ordinary MySQL instance with production data) to LogicDB logicdb01 configured on the management platform, and the belonging master compute node of the LogicDB is 192.168.210.32. Reference steps are as follow:

1. Use mysqldump to export table structure from source end of data migration (that is 192.168.210.45:3309), and execute the following commands on 192.168.210.45 server (the following parameters must be added):

```
root> mysqldump --no-defaults -h127.0.0.1 -P3309 -uhotdb_datasource -photdb_datasource --no-data --skip-triggers --set-gtid-purged=OFF --no-tablespaces --single-transaction --default-character-set=utf8mb4 --hex-blob --master-data=2 --no-create-db --skip-add-locks --skip-disable-keys --skip-tz-utc --databases db01 >db01.sql
```

2. After uploading SQL file of table structure to the server where compute node is, that is 192.168.210.32, log in to compute node to execute the following commands, and then Import Table Structure succeeded:

```
mysql> source /root/db01.sql
```

3. Use mysqldump to export table data from source end of data migration (that is 192.168.210.45:3309), execute the following commands on 192.168.210.45 server (the following parameters must be added):

```
root> mysqldump --no-defaults -h127.0.0.1 -P3309 -uhotdb_datasource -photdb_datasource --no-create-info --skip-triggers --set-gtid-purged=OFF --no-tablespaces --single-transaction --default-character-set=utf8mb4 --hex-blob --master-data=2 --no-create-db --skip-add-locks --skip-disable-keys --skip-tz-utc --databases db01 >db01-1.sql
```

4. Open export file of table data, view the current binlog position, and the display below means that the binlog position is 2410, and binlog file is mysql-bin.000076:

```
CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.000076', MASTER_LOG_POS=2410;
```

5. After uploading SQL file of table structure to the server where compute node is, that is 192.168.210.32, log in to compute node to execute the following commands, and then Import Table Data succeeded:

```
mysql> source /root/db01.sql
```

Pay attention: if foreign key is used, the following commands shall be executed additionally:

```
mysql> set foreign_key_checks=0
```

```
mysql> source /root/db01.sql
```

During the execution process, pay close attention to whether there is Warning or Error, otherwise, there will be problem of data inconsistency.

Tips: If transaction data is free of messy code problem, it could be considered splitting file and importing it to compute node in parallel, in order to accelerate the processing speed.

6. Use mysqlbinlog to make incremental data synchronization. If database name of the

source end is different from LogicDB name of the compute node, then it needs to add database mapping relation in management port first, for example:

```
dbremapping @@add@db01:logicdb01
```

Add is unnecessary in this case, and go to the server where compute node is (192.168.210.32) directly to make remote login to the source end (192.168.200.77) to execute the following command, binlog initial position is the position recorded in Step 4 (in this case it is 2410, binlog file is mysql-bin.000076):

```
mysqlbinlog -R -h192.168.210.45 -P3309 -uhotdb_datasource -photdb_datasource -v --base64-output=decode-rows --skip-gtids --to-last-log --stop-never --database=db01 --start-position=2410 mysql-bin.000076 | mysql -uroot -proot -h192.168.210.32 -P3323 -c -A
```

In order to accelerate data-catching speed, it's recommended that the server executing the mysqlbinlog command is the server where compute node is, which will save the time overhead of SQL and ok packet via network when the MySQL command line client executes SQL, and could greatly improve SQL execution speed of compute node single thread.

7. Check accuracy of data synchronization: at this time, it is necessary to stop service for a short time, interrupt the write-in operation of the transaction system toward database. After manual execution of a special data at the source end, view whether the data has been synchronized or not. After confirming that the compute node has been synchronized with the latest data, stop mysqlbinlog command, and if needed, cancel database name mapping.

Tips: After both the source end and the compute node have executed the following commands, you could view whether the select results are consistent or not to approximately judge whether the data is consistent or not

```
use xxx  
  
set session group_concat_max_len=1048576;
```

```
set @mytablename='xxx';

set @mydbname=database();

select concat('select sum(crc32(concat(ifnull(',group_concat(column_name separator
',\'NULL\'),ifnull(','NULL'))))) as sum from ',table_name,') as sqltext from information_schema.columns where
table_schema=@mydbname and table_name=@mytablename \G
```

If consistent, then data increment synchronization completed, for example, execute the following in the source end (192.168.200.77) MySQL instance:

```
mysql> use db01
Database changed

mysql> set session group_concat_max_len=1048576;
Query OK, 0 rows affected (0.00 sec)

mysql> set @mytablename='table02';
Query OK, 0 rows affected (0.00 sec)

mysql> set @mydbname=database();
Query OK, 0 rows affected (0.00 sec)

mysql> select concat('select sum(crc32(concat(ifnull(',group_concat(column_name separator
',\'NULL\'),ifnull(','NULL'))))) as sum from ',table_name,') as sqltext from information_schema.columns where
table_schema=@mydbname and table_name=@mytablename \G
***** 1. row *****

sqltext: select sum(crc32(concat(ifnull(id,'NULL'),ifnull(name,'NULL')))) as sum from table02;
1 row in set (0.00 sec)

mysql> select sum(crc32(concat(ifnull(id,'NULL'),ifnull(name,'NULL')))) as sum from table02;
```

```
+-----+  
| sum |  
+-----+  
| 1812521567 |  
+-----+  
1 row in set (0.00 sec)
```

If the result (1812521567) is consistent with execution result of compute node, then data synchronization completed.

7. Data consistency guarantee

7.1. Master/slave data consistency detection

HotDB Server provides Master/Slave Data Source Consistency Detection function in Data Node. The master/slave database, Active Master and Standby Slave needing checking must be respectively configured as master data source and slave data source, and shall belong to the same data node.

Master/slave data consistency detection, could check whether table structure of various tables in Active Master and Standby Slave is the same or not, whether the table data is consistent or not, and whether the master/slave has delay or not. When there is a few data inconsistency in table data between Active Master and Standby Slave, Master/slave data consistency detection could locate the Primary Key value of the inconsistent data line.

Log in to [management port \(3325 Port\)](#) of the compute node to execute show @@masterslaveconsistency command, thus it could be viewed whether the table is consistent on Active Master and Standby Slave:

```
mysql> show @@masterslaveconsistency;
+-----+-----+-----+-----+
| db      | table          | dn    | result | info
|-----+-----+-----+-----+
|       |
+-----+-----+-----+-----+
| DB_T      | FB_STUDENT      | dn_04 | NO     | There is
data inconsistency, because data source: 5, table: FB_STUDENT, MySQL error: Table 'db252.fb_student' doesn't
exist
|-----+-----+-----+-----+
| DB_A      | SP              | dn_04 | NO     | table: SP
has data inconsistency in node: 4, column: ID, distribution interval is: 0-17;, and unique key of the inconsistent
line is: (ID):(2),(1)
|-----+-----+-----+-----+
```

DB_T	JOIN_Z		YES		
<hr/>					
3 row in set (0.07 sec)					

It's shown in the result that for the JOIN_Z table in LogicDBDB_T, among master/slave data sources of all nodes, the data is consistent. Table structure is as follow:

- db: LogicDB name.
- table: Table Name.
- dn: data node name; when the table is inconsistent in master/slave data source, this column will display data node name;
- result: if checking result is YES, it means that the table is consistent between master/slave data sources; if NO, it means that the table is inconsistent between master/slave data sources, and meanwhile, inconsistency information will be output in info; UNKNOWN means unknown error, which may have the condition of table structure inconsistency, and master/slave replication interruption may also occur in UNKNOWN.
- info: In case of master/slave data consistency, there is no information output; in case of master/slave data inconsistency, there will be several kinds of information as follow:

A large amount of data inconsistency in table	Table: ... in datanode: ... exist a large amount of data inconsistency
Some data inconsistency in table	Table : ... in datanode: ... exist data inconsistency where ID in range:...;and inconsistent rows' Primary Key (...):
Standby Slave Table doesn't exist	exist data inconsistency, because DS: ... Table '...' doesn't exist
Index of table doesn't exist	DN: ... not exsit index of table:...

Master/slave failure detection (for example the Slave Slave_SQL_Running: NO state)	DN: ... ERROR! Check your replication.
Master/slave delay exceeds 10S	DN : ... delay too much,can't check master-slave data consistency
Delay exceeds 2S	Table: ... in datanode: ... exist a large amount of data inconsistency

7.2. Global AUTO_INCREMENT

Global AUTO_INCREMENT, refers to that the AUTO_INCREMENT column of the table makes orderly auto-increment in various nodes of the whole distribution system.

HotDB Server provides Global AUTO_INCREMENT support. When the table contains AUTO_INCREMENT column, and in server.xml file, the value of the parameter autoIncrement is set as non-zero (one or two), Global AUTO_INCREMENT of compute node could be used the same as using AUTO_INCREMENT of MySQL:

```
<property name="autoIncrement">1</property>
```

7.2.1. When the parameter is set as 0

If the parameter autoIncrement is set as 0, the auto-incremental field will be maintained in the data source MySQL, which is obvious when the table type is sharding table. Repeated auto-incremental sequences may occur between different data sources in the same sharding table.

For example: customer is an auto sharding table, and the sharding field is id, and name is defined as auto-incremental sequence. Then the auto increment feature of name is controlled by each data source:

```
mysql> create table customer(id int ,name int auto_increment primary key);

mysql> insert into customer values (1,null),(2,null),(3,null),(4,null);

Query OK, 4 rows affected (0.01 sec)

Records: 4 Duplicates:0 Warnings: 0

mysql> select * from customer;

+----+----+----+
| id | name | DNID |
+----+----+----+
| 4 | 1 | 1001 |
| 2 | 1 | 1006 |
| 3 | 1 | 1004 |
| 1 | 1 | 1008 |
+----+----+----+

4 rows in set (0.00 sec)
```

7.2.2. When the parameter is set as 1

If the parameter autoIncrement is set as 1, the compute node takes over the auto increment of all tables, which can ensure the global auto increment.

```
<property name="autoIncrement">1</property>
```

For example: customer is an auto sharding table, and the sharding field is id, and name is defined as auto-incremental sequence. Then the auto increment feature of name is controlled by the compute node, which can realize global auto increment:

```
mysql> create table customer(id int ,name int auto_increment primary key);

mysql> insert into customer values (1,null),(2,null),(3,null),(4,null);

Query OK, 4 rows affected (0.01 sec)

Records: 4  Duplicates: 0  Warnings: 0

mysql> select * From customer order by id;

+-----+-----+-----+
| id   | name | DNID |
+-----+-----+-----+
|    1 |    1 | 1008 |
|    2 |    2 | 1006 |
|    3 |    3 | 1004 |
|    4 |    4 | 1001 |
+-----+-----+-----+

4 rows in set (0.00 sec)
```

若将参数 [autoIncrement](#) 设置为 1，自增字段类型必须为 INT 或 BIGINT，否则建表提示 warning：

If the parameter autoIncrement is set as 1, the auto-incremental field type must be INT or BIGINT, otherwise, the table creation prompt warning:

```
mysql> create table table_test(id tinyint auto_increment primary key);

Query OK, 0 rows affected, 1 warning (0.05 sec)

Warning (Code 10212): auto_increment column must be bigint or int
```

The auto-incremental sequence 1 mode can guarantee the global uniqueness and strict positive increment, but does not guarantee the auto-incremental continuity.

7.2.3. When the parameter is set as 2

If the parameter autoIncrement is set as 2, the compute node takes over the auto increment of all tables. In this mode, when the compute node mode is cluster mode and the table contains the auto-incremental sequence, only the global uniqueness and relative increment in the long run of auto-incremental sequence is guaranteed, but the continuity of auto increment is not guaranteed (in a short time, the auto increment between different nodes will be staggered). The intelligent control of the auto increment feature by compute node can improve the performance of the compute node in the cluster mode. If the compute node mode is high availability or single node mode, the outcome of setting as 2 is the same as that of setting as 1.

For example, if the prefetchbatchinit of the existing Primary compute node A, Secondary compute node B and Secondary compute node C is set as 100, the prefetch interval of the auto-incremental sequence of compute node A is [1,100], the prefetch interval of compute node B is [101,200] and the prefetch interval of compute node C is [201,300], that is:

```
mysql> create table test(id int auto_increment primary key,num int);
```

Execute the code on compute node A:

```
mysql> insert into test values(null,1),(null,2),(null,3),(null,4);
```

```
mysql> select * from test order by id;
```

```
+-----+
```

```
| id | num |
```

```
+-----+
```

```
| 1 | 1 |
```

```
| 2 | 2 |
```

```
| 3 | 3 |
```

```
| 4 | 4 |
+----+
// the prefetch interval of the auto-incremental sequence is [1,100]
```

Execute the code on compute node B:

```
mysql> insert into test values(null,1),(null,2),(null,3),(null,4);
```

```
mysql> select * from test order by id;
```

```
+----+
```

```
| id | num |
+----+
```

```
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 101 | 1 |
| 102 | 2 |
| 103 | 3 |
| 104 | 4 |
+----+
```

```
// the prefetch interval of the auto-incremental sequence is [101,200]
```

Execute the code on compute node C

```
mysql> insert into test values(null,1),(null,2),(null,3),(null,4);
```

```
mysql> select * from test order by id;
```

```
+-----+  
| id  | num  |  
+-----+  
| 1   | 1    |  
| 2   | 2    |  
| 3   | 3    |  
| 4   | 4    |  
| 101 | 1    |  
| 102 | 2    |  
| 103 | 3    |  
| 104 | 4    |  
| 201 | 1    |  
| 202 | 2    |  
| 203 | 3    |  
| 204 | 4    |  
+-----+  
  
// the prefetch interval of the auto-incremental sequence is [201,300]
```

In the following two cases, it will judge whether to re prefetch the batch and recalculate the next batch size, so as to adjust the batch size suitable for the current business environment:

- (1) If the current batch utilization reaches the consumed proportion configured by the hidden parameter generatePrefetchCostRatio, the next batch will be prefetched. For example, if the consumed proportion is 90%, the current batch size is 100, and the auto increment has reached 90, the next batch will be prefetched at this time.
- (2) It is calculated from the time of fetching the batch. If it has reached the prefetchvalidtimeout, it will determine whether to prefetch the next batch according to

the utilization rate of the current batch. For example, if you set the consumed proportion as 90%, the current batch size as 100, and the current auto increment has reached 80, when the timeout time is reached and the current batch utilization rate reaches 50% of the configured consumed proportion, the next batch will be prefetched.

Currently, users are allowed to insert a specified auto increment. Whether the size of the auto increment is larger than the batch size or not, it can ensure the global orderly increment of the auto increment. For example, if the current batch size is 100, insert an auto increment value smaller than the batch size:

Execute the code on compute node A

```
mysql> insert into test values(null,1);

mysql> insert into test values (11,5);

mysql> insert into test values(null,1);

mysql> select * from test order by id;

+----+----+
| id | num |
+----+----+
| 1  |    1 |
| 11 |    5 |
| 12 |    1 |
+----+----+
```

Execute the code on compute node B

```
mysql> insert into test values(null,1);

+----+----+
| id | num |
+----+----+
| 1  |    1 |
```

11	5
12	1
101	1

Notes:

The auto-incremental sequence 2 mode can guarantee the global uniqueness and approximately positive growth in the long run, and does not guarantee the continuity of the auto increment;

The compute node can perceive the field type range of the auto-incremental sequence; the behavior of the node beyond the range is the same as that of MySQL;

If the parameter autoIncrement is set as 2, the auto-incremental field type must be bigint, otherwise the table creation fails:

```
mysql> create table table_test(id tinyint auto_increment primary key);

ERROR 10212 (HY000): auto_increment column must be bigint
```

7.3. Data strong consistency (XA transaction)

In Distributed Transactional Database system, after data sharding, the same transaction will operate multiple data nodes, and generate cross-node transaction. In cross-node transaction, after transaction COMMIT, if transaction is successfully committed in one of the data nodes, while of COMMIT failure in the other data node; for the data node with COMMIT completed, the data has been persistent and can no longer be modified; while for the data node with COMMIT failed, the data has lost, which result in data inconsistency between data nodes.

Using foreign XA TRANSACTION provided by MySQL, HotDB Server could solve

data strong consistency in cross-node transaction scenario: either COMMIT transactions of all nodes, or ROLLBACK all nodes.

7.3.1. Use XA transaction

In compute node, by default, XA TRANSACTION is disabled. To use XA transaction, the attribute enableXA shall be set TRUE in server.xml file:

```
<property name="enableXA">true</property>
```

It could take effect after restart the compute node, i.e. XA TRANSACTION function is enabled. If after XA TRANSACTION switch is modified, reload is directly conducted without reenabling compute node, then the Modify results shall not take effect and there will be INFO message in compute node log:

```
Can't reset XA in reloading, please restart the hotdb to enable XA
```

After enabling XA TRANSACTION function in compute node, the operations toward the application or client MySQL command are transparent, while the SQL command and transaction procedure are free of change, and could be used the same as ordinary transaction. Use START TRANSACTION or BEGIN, SET AUTOCOMMIT=0 to enable transaction, COMMIT or ROLLBACK to commit or rollback transaction.

To use XA TRANSACTION of compute node in the system, in order to guarantee strong consistency of transaction, several points below shall be paid attention to:

- MySQL Version must be 5.7.17 and above. As for the versions before 5.7.17, MySQL has defect in processing XA TRANSACTION. Therefore, under XA mode, if it's detected when enabling compute node that MySQL version of any data source is lower than 5.7.17, then Compute Node Enable failed; if data source lower than the version 5.7.17. is added after starting the compute node, then reload will fail;

if before starting the compute node, the data source lower than the version 5.7.17 can't be connected, then even if reconnection succeeded after restart, the data source is still Unavailable and reload will fail. There will be output of ERROR level log prompt for the above circumstances: Currently in XA mode, MySQL version is not allowed to be lower than 5.7.17.

- Data Source and configDB shall enable semi-synchronization replication (**extra attention: when enabling XA mode, MySQL Group Replication mode is not allowed**), when enabling semi-synchronization replication, it's not recommended enabling Master-1 (`innodb_flush_log_at_trx_commit = 1, sync_binlog = 1`) mode, and simultaneous enabling of both will also influence the performance;
- To deploy and use XA strong consistency mode, high availability of data node shall be configured, and attention shall be paid to that after host failover to Slave, the original host can't be reused, and can't be marked Available directly in compute node, and the original host must be redeployed in later period in order to be marked Available.
- Isolation level of XA TRANSACTION supports serializable and repeatable read, if splitting level of data source is set as read uncommitted and read committed, the compute node will process directly according to repeatable read. Meanwhile, under XA mode, isolation level of sessions can't be set as read uncommitted and read committed.
- XA transaction fully supports serializable, repeatable read and read committed, while read uncommitted is not supported for the moment. When the front-end isolation level is set to be read committed, you need to refer to the parameter description of `allowRCWithoutReadConsistentInXA` to avoid strong read-write consistency.
- After enabling XA mode and using HINT, since compute node can't control

modified contents of HINT statement, therefore, for any operations related with the connection in later period, the compute node will no longer control accuracy of the isolation level.

Special description:

Under XA mode: refer to SQL99 Standard, begin\start transaction will immediately start a transaction. That is, under the condition of enabling XA mode, begin\start transaction will equal to start transaction with consistent snapshot.

When the compute node version is 2.5.6 and above, if the front end is disconnected under XA mode, the transaction status will be recorded to the log and ConfigDB. You can check whether the transaction needs to be redone by executing SHOW ABNORMAL_XA_TRX directly at the service port.

```
2020-10-30 15:42:29.857 [WARN] [MANAGER] [$NIOExecutor-2-10]
cn.hotpu.hotdb.manager.response.v(39) - [thread=$NIOExecutor-2-
10,id=17,user=root,host=127.0.0.1,port=3323,localport=58902,schema=TEST_CT]killed
by manager

2020-10-30 15:42:29.857 [INFO] [INNER] [$NIOExecutor-2-10]
cn.hotpu.hotdb.server.d.c(1066) - XATransactionSession in [thread=$NIOExecutor-2-
10,id=17,user=root,host=127.0.0.1,port=3323,localport=58902,schema=TEST_CT]'s query
will be killed due to a kill command, current sql:null

2020-10-30 15:42:29.859 [INFO] [CONNECTION] [$NIOExecutor-2-10]
cn.hotpu.hotdb.server.b(3599) - [thread=$NIOExecutor-2-
10,id=17,user=root,host=127.0.0.1,port=3323,localport=58902,schema=TEST_CT] will be
closed because a kill command.
```

```
root@localhost:hotdb_config 5.7.25-log 03:44:23> select * from abnormal_xa_trx\G;
***** 1. row *****
    xid: HotXA7600198_
    connection_id: 17
    connect_time: 2020-10-30 15:39:50.996
last_sql_sent_time: 2020-10-30 15:41:47.646
    disconnect_time: 2020-10-30 15:42:29.868
disconnect_reason: kill
    trx_start_time: 2020-10-30 15:41:42.929
    trx_state: ROLLBACKED_BY_HOTDB
    logic_db: TEST_CT
***** 2. row *****
    xid: HotXA7600368_
    connection_id: 22
    connect_time: 2020-10-30 15:47:15.777
last_sql_sent_time: 2020-10-30 15:47:42.236
    disconnect_time: 2020-10-30 15:47:59.679
disconnect_reason: kill
    trx_start_time: 2020-10-30 15:47:21.776
    trx_state: ROLLBACKED_BY_HOTDB
    logic_db: TEST_CT
2 rows in set (0.00 sec)
```

```
root@127.0.0.1:TEST_CT 5.7.22 03:49:21> root@127.0.0.1:TEST_CT 5.7.22 03:49:21> SHOW ABNORMAL_XA_TRX;
+-----+-----+-----+-----+-----+-----+-----+-----+
| xid | connection_id | connect_time | last_sql_sent_time | disconnect_time | disconnect_reason | trx_start_time | trx_state | logic_db |
+-----+-----+-----+-----+-----+-----+-----+-----+
| HotXA7600198 | 17 | 2020-10-30 15:39:50.996 | 2020-10-30 15:41:47.646 | 2020-10-30 15:42:29.868 | kill | 2020-10-30 15:41:42.929 | ROLLBACKED_BY_HOTDB | TEST_CT |
| HotXA7600368 | 22 | 2020-10-30 15:47:15.777 | 2020-10-30 15:47:42.236 | 2020-10-30 15:47:59.679 | kill | 2020-10-30 15:47:21.776 | ROLLBACKED_BY_HOTDB | TEST_CT |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

disconnect_reason: reasons for disconnection, such as kill, TCP disconnection (program err:java.io.IOException: Connection reset by peer), SQL execution timeout (stream closed, read return -1), idle timeout, etc.

trx_state: the transaction status of disconnection, including:

1. ROLLBACKED_BY_HOTDB: in a transaction and the transaction is rolled back by the compute node (when the transaction is not committed automatically, the application program does not issue the commit command or lose the commit command halfway);
2. COMMITTED_BY_HOTDB: in a transaction and the transaction is committed by the compute node (when the transaction is not committed automatically, the compute node receives a commit and commits it successfully, but the front-end connection is disconnected in the middle of commit, so the compute node fails to send ok package).

7.3.2. Relation between XA transaction and read/write splitting

splitting

Under XA mode, when start Read/write splitting: only read request of single node in non-transaction is splittable, and Mode 3 degrades to Mode 2;

Under XA mode, when start Read/write splitting: it can't guarantee accuracy of isolation level but could guarantee no loss of data, and no occurrence of partial

Commit/partial Rollback;

Meanwhile, at the time of use, compute node log will output relevant Warning prompt.

7.4. Nondeterministic Function Proxy

The Non-Deterministic Function will bring some column problems in use, especially data consistency problem of Global Table, therefore, HotDB Server provides the function of Nondeterministic Function Proxy. Non-Deterministic Function is generally divided into types, one is Time Function with known value, such as CURDATE(), CURRENT_TIMESTAMP(), etc., the other is Random Value Function and Unique Value Function with unknown value, such as RAND(), UUID(), etc.

1. For Time Function, compute node makes Uniform Proxy.

- When the Table Field type is datetime (or timestamp) and free of default value, the parameter [timestampProxy](#) will control Proxy Range of the compute node (Auto Mode by default, and Global Processing/Auto Detection are optional), and insert such Function Proxy as specific value into Table;
- In case the Functions such as curdate(), curtime(), etc. appear in select/insert/update/delete statement, compute node will insert such Function Proxy as specific value into Table;

2. For Random Value Function, compute node conducts different Proxy methods toward different SQL statements.

3. For Unique Value Function, compute node makes Uniform Proxy.

- In case uuid() or uuid_short() appear in select/insert/update/delete statement, compute node will make Proxy Unique Value according to standard UUIDv1 algorithm;

- In case of server_id conflict between data source and configDB, compute node will Auto Disable uuid_short() and inform the user of adjusting the server_id manually. You can refer to description in official website of MySQL:
<https://dev.mysql.com/doc/refman/5.7/en/replication-options.html>.

7.5. Global Time Zone Support

In order to guarantee data accuracy, for different time zones are set due to existence of different data source servers, and thus resulting in data error of the time type in database, HotDB Server provided support for Global Time Zone, including:

- When the time_zone parameters are specific same value or for SYSTEM the system_time_zone are all the same specific value, HotDB Server makes no special treatment, otherwise, HotDB Server will set the time_zone as uniform fixed value: “+8:00” and will record warning level log (The datasources' time_zones are not consistent);

For example, after login service port, enter the command:

```
set time_zone = '+0:00';  
  
show variables like '%time_zone';
```

It will still display that the time_zone is '+8:00':

```
mysql> set time_zone='+0:00';  
  
mysql> show variables like '%time_zone';  
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| system_time_zone | CST |
```

```
| time_zone      | +08:00 |
+-----+-----+
2 rows in set (0.09 sec)
```

- At the time of Backup and Recovery, HotDB Server could guarantee that the time type data after recovery is consistent with that before backup.

7.6. Global Unique Constraint

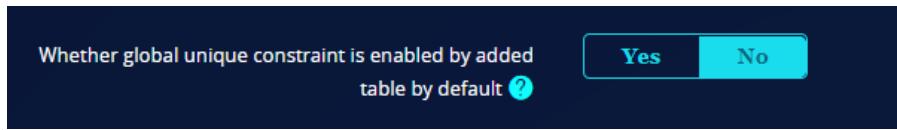
If start Global Unique Constraint function, HotDB Server could guarantee that the column with Unique Constraint (UNIQUE, PRIMARY KEY) is unique in all nodes, including but not limited to the following scenarios:

- Unique Constraint Key is not sharding key or doesn't include sharding key
- Under Parent/Child Table, associate field of Child Table and Parent Table is not the same column with Unique Constraint Key of Child Table

HotDB Server 2.5.3 optimizes and accurate Global Unique Constraint to Table Level, and Global Unique Constraint is disabled for all tables added in the future by default, or Global Unique Constraint could be separately disabled / enabled for some tables manually when Add Table.

You could either modify the following parameter in server.xml or modify the parameter in compute node parameter configuration on the management platform. Modify parameter only sets Global unique default value for the table added in the future, but doesn't influence Global Uniqueness of history data table.

```
<property name="globalUniqueConstraint">false</property><!-- Global unique constraints enable on newly added table by default or not -->
```

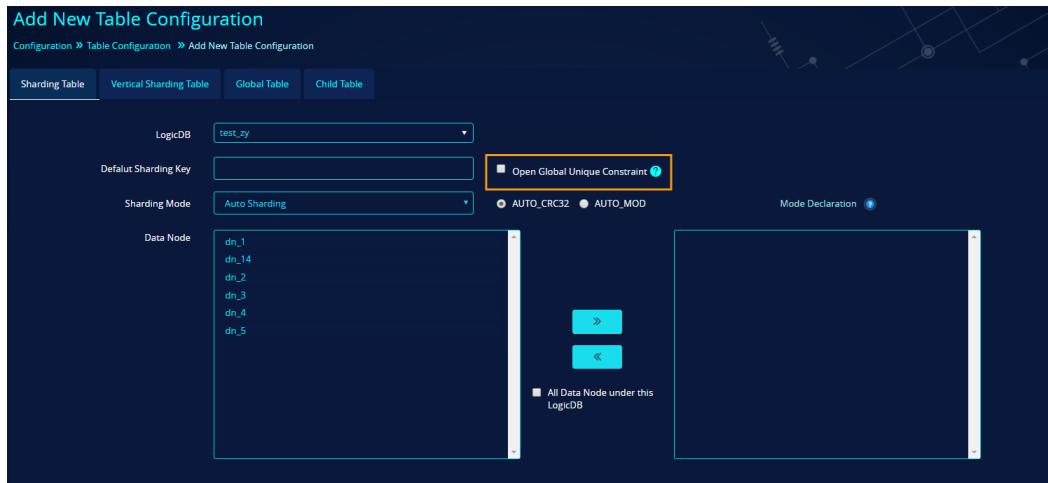


Notice: After enable the function, there may be great influence on execution efficiency of INSERT, UPDATE and DELETE of SQL statements, and may result in increasing delay of SQL operation; it may also increase the circumstance of Lock Wait and Deadlock. Please make the choice upon careful consideration.

7.6.1. Table level control when creating a table

At the time of Add Table Configuration, it could Enable/Disable Global Unique Constraint separately for a table

- When Add Table Configuration on the management platform, Enable/Disable status of Global Unique Constraint will be displayed by default according to compute node parameter, which could also be manually modified:



Both Vertical Sharding Table and Global Table have no such exit, because no special treatment is required for Unique Constraint. After Add Table and Configuration, it could be used after using Create Table statement to Add Table Structure.

- Using [Auto Create Table](#) function, the switch of Global Unique Constraint could be set via table option GLOBAL_UNIQUE [=] {0 | 1}. For example:

```
mysql> create table test02(id not null auto_increment Primary Key,a char(8),b decimal(4,2),c int)
```

```
GLOBAL_UNIQUE=0;
```

```
mysql> create table test03(id int Primary Key,id1 int) GLOBAL_UNIQUE =1;
```

If not using GLOBAL_UNIQUE [=] {0 | 1}, then by default, it will be set Enable or Disable according to default value of the compute node parameter configuration or the Table Configuration added on the management platform; if GLOBAL_UNIQUE=1, it shall be judged as Enable; if GLOBAL_UNIQUE=0, it shall be judged as Disable.

- If GLOBAL_UNIQUE setting is different from the default value, then GLOBAL_UNIQUE shall prevail;
- If GLOBAL_UNIQUE setting is different from Global Unique Constraint configuration of this table on the Management Platform, then Create Table will fail, and error prompt will be given, for example when Add test01 on the management platform, Global Unique Constraint is disabled:

```
mysql> create table test01(id int)global_unique=1;
```

```
ERROR 10172 (HY000): CREATE TABLE FAILED due to generated table config already in HotDB config datasource. You may need to check config datasource or reload HotDB config.
```

- If using GLOBAL_UNIQUE in Create Table statement in Vertical Sharding Table or Global Table, then Create Table will succeed, but warning message will be given, because no extra treatment is required for its Unique Constraint, for example, test03 is a Vertical Sharding Table:

```
mysql> create table test03(id int)global_unique=1;
```

```
Query OK, 0 rows affected, 2 warnings (0.09 sec)
```

```
Warning (Code 10032): Create table without Primary Key and unique key
```

```
Note (Code 10210): Global_unique is not applicable to vertical-sharding tables or global tables.
```

In order to meet compatibility of MySQL, for example, when using mysqldump, there is no worry about that the backup results will be interfered by GLOBAL_UNIQUE, therefore, GLOBAL_UNIQUE syntax is parsed as Annotation Format, for example:

```
mysql> create table test02(id int) GLOBAL_UNIQUE=1;

mysql> show create table test02

+-----+-----+
| Table | Create Table           |
+-----+-----+
| test3 | CREATE TABLE `test3` (
  `id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4/*hotdb:020503 global_unique=1*/ |

+-----+-----+
1 row in set (0.00 sec)
```

Importing the Create Table statement containing GLOBAL_UNIQUE Syntax into MySQL will have no influence on the result. In compute node, Annotation Syntax could also be directly used to operate GLOBAL_UNIQUE, and use -c as login parameter of MySQL, and annotation is allowed to be executed.

```
root> mysql -c -uroot -proot -h127.0.0.1 -P3323
```

For example, executing the following statement, means that when the version of HotDB Server is higher than 2.5.3, GLOBAL_UNIQUE=0 will be executed:

```
mysql> create table test02(id not null auto_increment Primary Key,a char(8),b decimal(4,2),c int) /*hotdb:020503
GLOBAL_UNIQUE=0*/;
```

7.6.2. Table level control when modifying a table

1. Table Configuration could be modified on Table Configuration Management page on the Management Platform:

Table Configuration									
Configuration > Table Configuration									
	Please select LogicDB	Select Table Type	Select Sharding Type	Select Data Node	Select Table Structure	Filter by Table Name	Add New Table Configuration	Export	Batch Operation
	■	LogicDB	Table Name	Table Type	Sharding Key	Sharding Function Name	Data Node	Table Structure	Whether to Enable the Global Unique Constraint
	■	test_zy	table3	Sharding Table	id	Auto Sharding - CRC32	dn_1 dn_2 dn_3 dn_4	Created	<input type="checkbox"/>
	■	test_zy	table1	Sharding Table	id	Auto Sharding - CRC32	dn_1 dn_2 dn_3 dn_4	Created	<input type="checkbox"/>
	■	test_zy	sm_d_tb3	Sharding Table	id	Auto Sharding - CRC32	dn_1 dn_2 dn_3 dn_4	Created	<input type="checkbox"/>
	■	test_zy	sm_d_tb2	Sharding Table	id	Auto Sharding - CRC32	dn_1 dn_2 dn_3 dn_4	Created	<input type="checkbox"/>
⊕	■	test_zy	sm_c_tb3	Sharding Table	id	Auto Sharding - CRC32	dn_1 dn_2 dn_3 dn_4	Created	<input type="checkbox"/>
⊕	■	test_zy	sm_c_tb2	Sharding Table	id	Auto Sharding - CRC32	dn_1 dn_2 dn_3 dn_4	Created	<input type="checkbox"/>

If the Table Structure is Created Table, then after Modify the Global Unique Constraint to Enable status, click Reload and refresh the page, if the prompt as in the figure below appears, it means that unique @@create shall be executed on management port, and check history data of the Unique Constraint Key of the table, and after the return result is unique, the compute node will Auto Create secondary index, only in this way could Global Unique Constraint take effect, for details of this command, please refer to [Distributed Transactional Database HotDB Server \[management port Command Function Manual\]](#)

[Function Manual](#):

Table Configuration										
Configuration > Table Configuration										
	Please select LogicDB	Select Table Type	Select Sharding Type	Select Data Node	Select Table Structure	Filter by Table Name	Add New Table Configuration	Export	Batch Operation	
	■	LogicDB	Table Name	Table Type	Sharding Key	Sharding Function Name	Data Node	Table Structure	Global Unique Constraint	
	⊕	■	test_zy	sm_c_tb1	Sharding Table	id	Auto Sharding - CRC32	dn_1 dn_2 dn_3 dn_4	Created	<input type="checkbox"/>
	⊕	■	test_zy	sm_b_tb4	Sharding Table	id	Auto Sharding - CRC32	dn_1 dn_2 dn_3 dn_4	Created	<input type="checkbox"/>
	⊕	■	test_zy	sm_b_tb3	Sharding Table	id	Auto Sharding - CRC32	dn_1 dn_2 dn_3 dn_4	Created	<input type="checkbox"/>

2. Use GLOBAL_UNIQUE Syntax in the compute node via ALTER TABLE, enable Global Unique, and similarly, if appearing warning message, it means that it could take

effect only after executing unique @@create:

```
mysql> alter table keevey01 global_unique=1;
```

```
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

```
mysql> show warnings;
```

```
+-----+-----+-----+
```

Level	Code	Message
-------	------	---------

--

```
+-----+-----+-----+
```

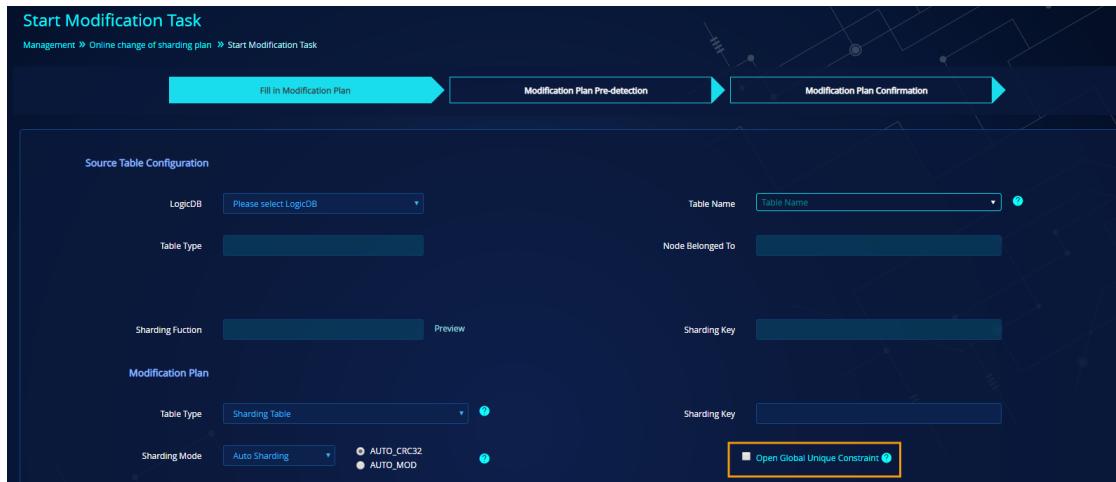
Note 10210 please go to HotDB Server manager port and execute this command: unique @@create, otherwise this global_unique setting doesn't work.

```
+-----+-----+-----+
```

```
1 row in set (0.00 sec)
```

7.6.3. Online Change of Sharding Plan

At the time of Online Change of Sharding Plan, you could also Enable or Disable Global Unique Constraint manually for the table after change.



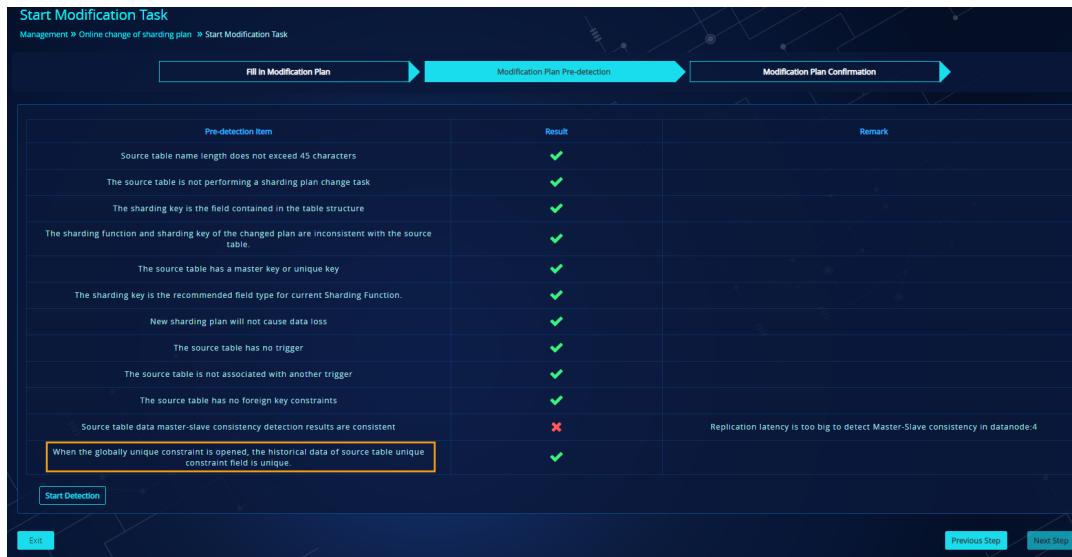
The screenshot shows the 'Start Modification Task' interface with the following steps:

- Fill in Modification Plan
- Modification Plan Pre-detection
- Modification Plan Confirmation

In the 'Modification Plan Confirmation' section, there are several configuration fields:

- Source Table Configuration:
 - LogicDB: Please select LogicDB
 - Table Name: Table Name
 - Table Type: Table Type
 - Node Belonged To: Node Belonged To
 - Sharding Function: Sharding Function
 - Sharding Key: Sharding Key
- Modification Plan:
 - Table Type: Sharding Table
 - Sharding Mode: Auto Sharding (radio button selected)
 - AUTO_CRC32 (radio button)
 - AUTO_MOD (radio button)
 - Open Global Unique Constraint (checkbox highlighted with a yellow border)

After Enable, the Modification Plan Pre-detection will detect whether history data of Unique Constraint Key of the table is unique or not, if yes, then Test Passed.



7.6.4. Locate by secondary index when querying

This function also supports that when the SELECT Query statement doesn't contain sharding key but contains Unique Constraint field, locate to fixed node via Query Secondary Index, and distribute SELECT Query statement to specified node only instead of all nodes.

```
<property name="routeByRelativeCol">false</property><!--When not containing sharding key, via Secondary Index Field Route-->
```

This function is Disable by default, which could be enabled by modifying the routeByRelativeCol parameter in server.xml or adding the parameter “When not containing sharding key, enable via Secondary Index Field Route or not” in compute node parameter configuration under Management Platform Configuration Menu.

After the parameter is enabled, example of its role is as follow: for an existing Sharding Table table01, sharding key is id, sharding function is auto_mod, when executing the following Query statement:

```
SELECT * FROM table01 WHERE unique_col = 100; //unique_col is Unique Constraint column
```

This Query statement will be distributed to the data node with unique_col = 100 only, instead of all data nodes

7.7. Data accuracy guarantee after failover

Whether Asynchronous Replication or Semi-synchronization Replication, there may be such a circumstance as follow: after failover completed, the original Active Master will recover, the new Active Master (the original Standby Slave) Replication IO Thread may Auto Reconnect and acquire the transaction not acquired before Switch, which such transaction has been deemed as transaction failing to be committed during the processing process of Switch, and can't continue to be replicated, otherwise, there will be risk of messy data.

Therefore, compute node parameter [failoverAutoresetslave](#) is added, which is Disable by default.

```
<property name="failoverAutoresetslave">false</property><!-- At the time of failover, Auto Reset master/slave replication relation or not -->
```

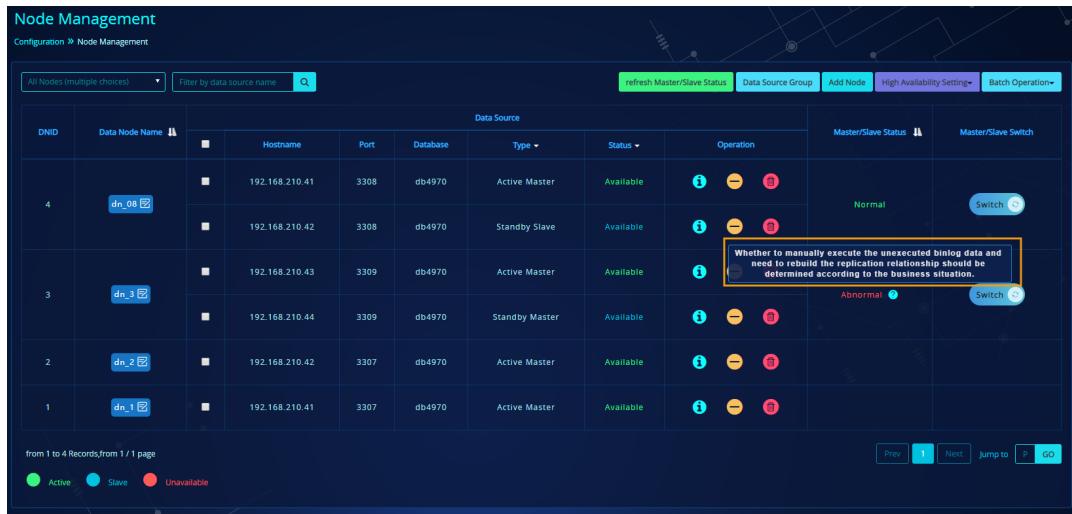
After failover, IO thread between the original Master/Slave will be suspended, and Heartbeat Detection of the original Active Master will be conducted once every minute, until the original Active Master recovers normal. After the original Active Master recovers normal, comparing with binlog position of the original Active Master, detect whether the original Standby Slave (the existing Active Master) has the transaction not acquired before Switch, if yes, enable this parameter will Auto Reset the master/slave replication relation. If not, there will be no treatment after Re-enable the IO Thread.

Notice:

- The precondition for detecting whether there is transaction not received or not is that both the Active Master and Standby Slave need to enable GTID, otherwise, when this parameter is enabled, there will be Auto Reset of master/slave replication relation after failover completed.
- If at the time of Heartbeat Detection, the original Active Master is still in Unavailable state after more than 10080 times of Retry, at this time, parameter is in Enable status, and there will also be Auto Reset of master/slave replication relation.
- For example, after Auto Reset of replication relation, warning log recorded by compute node at warning level will be as follow:

you should decide whether to manually execute the unexecuted part of binlog or rebuild the replication according to the actual situation.

And the Master/Slave Status in the management platform will display Abnormal, and the Pre-click display is as the prompt message in the figure:



Data Source							Master/Slave Status	Master/Slave Switch	
DNID	Data Node Name	Hostname	Port	Database	Type	Status			Operation
4	dn_08	192.168.210.41	3308	db4970	Active Master	Available	Info Sync Unsync	Normal	Switch
		192.168.210.42	3308	db4970	Standby Slave	Available	Info Sync Unsync	Normal	Switch
3	dn_3	192.168.210.43	3309	db4970	Active Master	Available	Info Sync Unsync	Abnormal	Switch
		192.168.210.44	3309	db4970	Standby Master	Available	Info Sync Unsync	Abnormal	Switch
2	dn_2	192.168.210.42	3307	db4970	Active Master	Available	Info Sync Unsync	Abnormal	Switch
1	dn_1	192.168.210.41	3307	db4970	Active Master	Available	Info Sync Unsync	Abnormal	Switch

from 1 to 4 Records, from 1 / 1 page

● Active ● Slave ● Unavailable

- If after failover completed, the Master Active and Standby Slave neither Switch GTID or have transaction not received, but this parameter is disabled, then the compute node will also record warning log at warning level as follow:

DBA is required to deal with the new master, which is the original slave before switching and decide whether to stop replication or continue replication regardless. In addition, there is risk of data error caused by automatic reconnection of replication after manual or unexpected restart of the new master.

8. High availability service

This chapter mainly describes the high availability service of the computing node cluster in the single-IDC mode. To understand the high availability service in the DR mode, please refer to the [Distributed Transactional Database HotDB Server - Cross-IDC Disaster Recovery Function Specification](#).

8.1. High availability service

HotDB Server provides MySQL high availability in data node, when the Master data source is Unavailable, compute node will switch to slave data source automatically.

If to use high availability of data node, the following preconditions shall be met:

- Rules for configuration of master/slave data source and failover priority in data node;
- Master/slave data source must have been configured master/slave or standby master replication relation;
- Enable Heartbeat function in compute node Config File.

8.1.1. High availability of data node

For master/slave configuration mode of MySQL database, please refer to official website of MySQL (pay attention to official document of corresponding version, for example: <http://dev.mysql.com/doc/refman/5.6/en/replication.html>)

By default, compute node Heartbeat function is enabled:

```
<property name="enableHeartbeat">true</property>
```

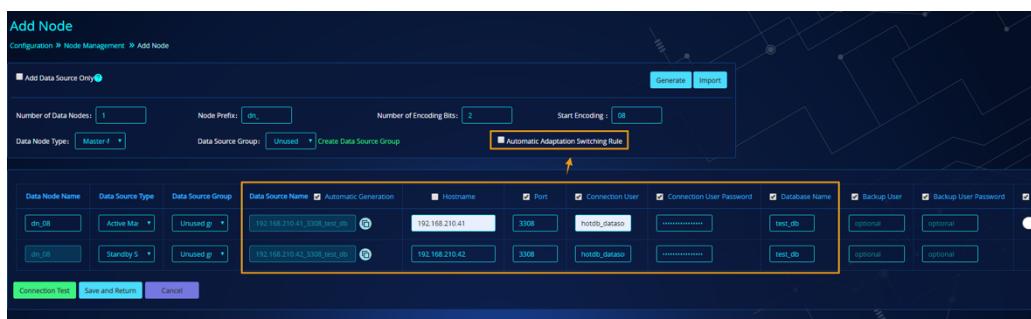
Assuming that instance 3308 of 192.168.210.41 and instance 3308 of 192.168.210.42 are a pair of MySQL databases of master/slave replication.

The master/slave data source configured in the same node, could be set in “Add Node”

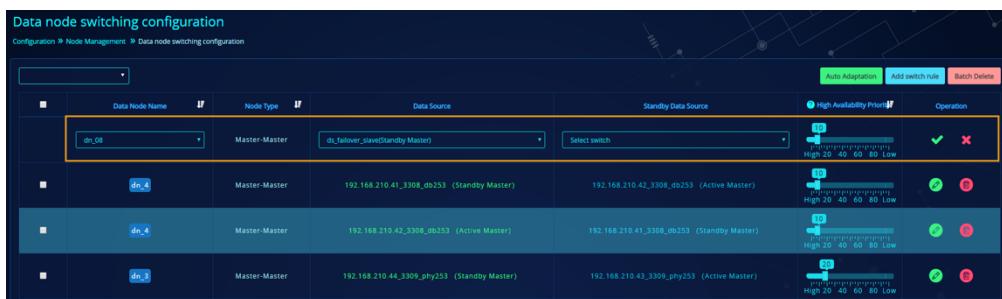
page or “Data Source Edit” page on the management platform.

Assuming that the data node “dn_08” has already existed, Add Master data source and slave data source for this node. On management platform page, select “Configuration”->“Node Management”->“Add Node”, and jump to the “Add Node” page:

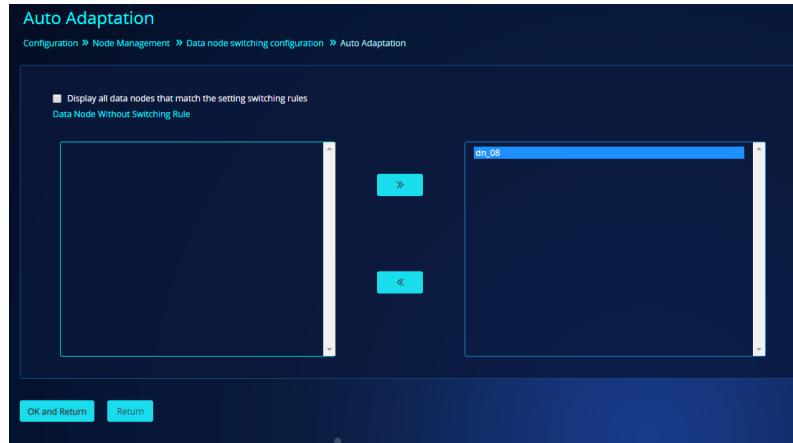
In the following operation, generate a data node “dn_08”, and add a Master data source“ds_failover_master” and a slave data source“ds_failover_slave” for this data node:



“Automatic Adaptation Switching Rule” could be directly checked, when Add Node, make Automatic Adaptation of failover priority at the same time. Or, on the management platform page, select “Configuration”->“Node Management”->“High Availability Setting”->“Switching Rule”->“Add Switch Rule”, select “dn_08” from drop-down box of data node, select Master data source “ds_failover_master” from drop-down box of data source, select “ds_failover_slave” from drop-down box of standby data source, and select High in Failover Priority:

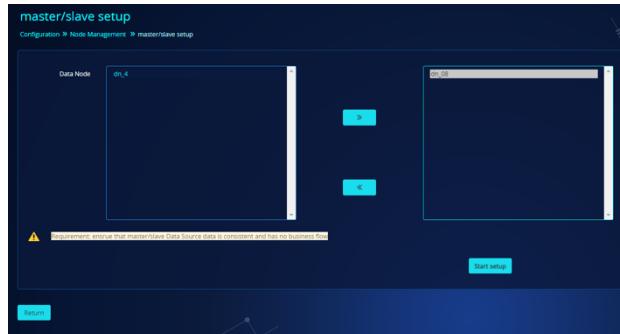


Or click “Auto Adaptation”, select dn_08 node, and then Save.



Build master/slave replication relation:

Although a pair of master/slave data sources have been added under node dn_08, if the 2 data sources haven't been built master/slave replication relation in practice, then at this time, you could select "dn_08" node in "Configuration"->"Node Management"->"High Availability Setting"->"Master/Slave Build".



After clicking "Start Build", the system will build master/slave replication relation for the data source automatically. After Build succeeded, the Master/Slave status is the list will display normally:

Node Management										
Configuration > Node Management										
All Nodes (multiple choices)		Filter by data source name		Data Source			refresh Master/Slave Status	Data Source Group	Add Node	High Availability Setting
4	dn_08	Hostname	Port	Database	Type	Status	Operation	Normal	Switch	
		192.168.210.41	3308	test_db	Active Master	Available				
3	dn_3	192.168.210.42	3308	test_db	Standby Slave	Available		Normal	Switch	
		192.168.210.43	3309	db4970	Active Master	Available				
2	dn_2	192.168.210.44	3309	db4970	Standby Master	Available		Normal	Switch	
		192.168.210.42	3307	db4970	Active Master	Available				
1	dn_1	192.168.210.41	3307	db4970	Active Master	Available				

8.1.1.1. Manual Switch

In “Configuration”->“Node Management”, click [Switch] of a data node, thus Manual Switch could be completed:

Node Management									
Configuration > Node Management									
		Data Source							
DNID	Data Node Name	Hostname	Port	Database	Type	Status	Operation		
4	dn_08	192.168.210.41	3308	test_db	Active Master	Available			
		192.168.210.42	3308	test_db	Standby Slave	Available			
3	dn_3	192.168.210.43	3309	db4970	Active Master	Available			
		192.168.210.44	3309	db4970	Standby Master	Available			
2	dn_2	192.168.210.42	3307	db4970	Active Master	Available			
1	dn_1	192.168.210.41	3307	db4970	Active Master	Available			

Description:

- If being Master - Slave, select the data source with highest priority for Switch, and after Switch, the compute node will set the original master and the other Slaves of the original master as Unavailable, which can no longer be switched.
- If being standby master, after Switch, the Active Master will not be set as Unavailable, and you can switch back and forth.
- If the Standby Slave with the highest-priority is Unavailable or delays for more than 10s, select the one with higher priority from the remaining Standby Slaves for Switch, if they are all Unavailable or delay for more than 10s, there shall be no Switch, and will be error prompt (Switch Failure Log will prompt: switch datasource datasourceid failed due to: no available backup found)
- In V2.5.6 and above, in the case of manual switching, it will check first whether the current table hotdb_datanode/hotdb_datasource/hotdb_failover is consistent with the running table. If they are not consistent, it will prompt: “Unable to switch because the configuration of the current data source is inconsistent with the

configuration in memory. Please try again after reloading.” If they are consistent, the taken data source will be updated to the master before new data source taking over, and the original master data source will be updated to the standby master or slave. (Note: if it is master/slave relation, the original master data source and its related slave nodes will be cascaded to be unavailable; the original replication relations will be cleaned up synchronously during the switching; the failover rules of the original master data source and the original slave data source will be exchanged, waiting for the manual offline reconstruction of replication relations.

- After Switch succeeded, the compute node will record switch process log:

```
INFO [pool-1-thread-1064] (SwitchDataSource.java:78) -received switch datasourceid command from Manager : [Connection Information]

WARN [pool-1-thread-1339] (BackendDataNode.java:263) -datanode id switch datasource:id to datasource:id in failover. due to: Manual Switch by User: username

INFO [pool-1-thread-1339] (SwitchDataSource.java:68) -switch datasource:id for datanode:id successfully by Manager.
```

- In case of no Switching Rule configured, there will be no Switch, and will be the error prompt: switch datasource id failed due to:found no backup information)

8.1.1.2. Failover

Failover is generally the Auto Switch after data source failure, and the description is as follow:

- Whether Standby Master or master/slave, after Failure and Switch of the Active Master, they will set the Active Master as Unavailable, which can no longer be switched. The Active Master needs to be recovered manually, and after the Active Master becomes normal, set the Active Master as Available manually, and the Active Master will be restored as available after Reload.

- If the Standby Slave Status is Unavailable, there will be no Switch, and the compute node will record log

```
WARN [pool-1-thread-2614] datanode id failover failed due to found no available backup
```

- In server.xml, you could configure the parameter waitForSlaveInFailover to control whether the Switch shall wait for the Standby Slave to catch up with replication. If the parameter is true by default, which means Waiting, then during Switch process, it will wait for the Standby Slave to catch up with replication. If set as false, which means Not Waiting, then it will Switch immediately. (There is a risk of data loss in the immediate switch, which is not recommended).
- In V2.5.6 and above, in the case of failover, the taken over data source will be updated to the master before new data source taking over, and the original master data source will be updated to the standby master or slave. The original master data source and its related slave nodes will be cascaded to be unavailable. (Note: if it is master/slave relation, the original replication relations will be cleaned up synchronously during the switching; the failover rules of the original master data source and the original slave data source will be exchanged, waiting for the manual offline reconstruction of replication relations).
- During Failover, Active Master Heartbeat is continuous. If Heartbeat succeed for two successive time, then give up Switch, and the compute node will record log

```
INFO [$NIOREACTOR-6-RW] (Heartbeat.java:502) -heartbeat continue success twice for datasource 5  
192.168.200.52:3310/phy243_05, give up failover.
```

- After Switch succeeded, the compute node will record log, and record the cause for Switch:

If being network failure, server down, power down, etc., then it will record: Network is unreachable

If the Network is reachable, MySQL service stops and is without response, then it will record: MySQL Service Stopped

If MySQL service is enabled, but the response is abnormal, then it will record: MySQL Service Exception

For example: when stopping the data source service, the whole Switch process is promoted as follow:

```
02/21 15:57:29.342 INFO [HeartbeatTimer] (BackendDataNode.java:396) -start failover for datanode:5
02/21 15:57:29.344 INFO [HeartbeatTimer] (BackendDataNode.java:405) -found candidate backup for
datanode 5 :[id:9,nodeId:5 192.168.200.51:331001_3310_ms status:1] in failover, start checking slave status.
02/21 15:57:29.344 WARN [$NIOREACTOR-0-RW] (HeartbeatInitHandler.java:44) -datasource 5
192.168.200.52:331001_3310_ms init heartbeat failed due to:MySQL Error Packet{length=36,id=1}
02/21 15:57:29.344 INFO [pool-1-thread-1020] (CheckSlaveHandler.java:241) -slave_sql_running is Yes
in :[id:9,nodeId:5 192.168.200.51:331001_3310_ms status:1] during failover of datanode 5
02/21 15:57:29.424 WARN [pool-1-thread-1066] (BackendDataNode.java:847) -datanode 5 switch datasource
5 to 9 in failover. due to: MySQL Service Stopped
02/21 15:57:29.429 WARN [pool-1-thread-1066] (Heartbeat.java:416) -datasource 5
192.168.200.52:331001_3310_ms heartbeat failed and will be no longer used.
```

- In case of Switching Rule configured, there will be no Switch, and the compute node will record log:

```
WARN [pool-1-thread-177] (?:?) -datanode id failover failed due to found no backup information
```

- After Failover of data source, whether Master/Slave or Standby Master, we have the uniform requirement that, the precondition for setting the data source as Available manually is that the operating personnel must be clear about that the existing master/slave service is free of abnormalities, the data synchronization is free of abnormalities, and especially for the master/slave mode, it must be guaranteed that the data at the time of Slave service during the period has been synchronized to the Master data source. When data source is enabled, we shall form

the habit of not skipping the Master / Slave Consistency Detection.

8.1.2. ConfigDB High Availability

When the configDB and some data sources are deployed on the same MySQL instance or the same server, in case of failure with this instance, the failure information can't be recorded into the configDB, therefore, compute node supports the ConfigDB High Availability function, in order to guarantee normal use of configDB.

1. Configure connection information of master/slave configDB in server.xml, in order to guarantee normal master/slave relation

```
<property name="url">jdbc:mysql://192.168.200.191:3310/hotdb_config</property><!-- Master configDB address-->

<property name="username">hotdb_config</property><!-- Master configDB username -->

<property name="password">hotdb_config</property><!-- Master configDB password -->

<property name="bakUrl">jdbc:mysql://192.168.200.190:3310/hotdb_config</property><!-- Slave configDB address -->

<property name="bakUsername">hotdb_config</property><!-- Slave configDB username -->

<property name="bakPassword">hotdb_config</property><!-- Slave configDB password -->
```

- In case of failure with the Master configDB, it will switch to the Slave configDB automatically. In case of delay in the Switch process, it will wait for the Slave configDB to catch up with replication.
- When the compute node version is 2.5.6 or above, in case of failure with the Master configDB, the slave ConfigDB will be updated to the master before taking over, and the original master ConfigDB will be updated to the slave. Server.xml will be adjusted synchronously, as shown below:

```
<property name="url">jdbc:mysql://192.168.200.190:3310/hotdb_config</property><!-- Master configDB address -->

<property name="username">hotdb_config</property><!-- Master configDB username -->

<property name="password">hotdb_config</property><!-- Master configDB password -->

<property name="bakUrl">jdbc:mysql://192.168.200.191:3310/hotdb_config</property><!-- Slave configDB address -->

<property name="bakUsername">hotdb_config</property><!-- Slave configDB username -->

<property name="bakPassword">hotdb_config</property><!-- Slave configDB password -->
```

- If Compute Node High Availability service involves master/slave relation of configDB, then it shall be guaranteed that configuration of a group of Compute Node High Availability master/slave configDB in server.xml is identical, and there shall be no staggered configuration.

2. configDB also supports MGR configDB at the same time (MySQL must be above the Version 5.7, configDB MGR only supports three nodes for the time being), in server.xml, configure configDB information with MGR relation, and guarantee that the MGR relation is normal. Under the condition where MGR relation is inaccurate, configDB may can't provide normal service, and may result in Startup Failure.

```
<property name="url">jdbc:mysql://192.168.210.22:3308/hotdb_config_test250</property><!-- configDB address -->

<property name="username">hotdb_config</property><!-- configDB username -->

<property name="password">hotdb_config</property><!-- configDB password -->

<property name="bakUrl">jdbc:mysql://192.168.210.23:3308/hotdb_config_test250</property><!-- Slave configDB address (if configDB uses MGR, this item must be configured) -->

<property name="bakUsername">hotdb_config</property><!-- Slave configDB username (if configDB uses MGR, this item must be configured) -->

<property name="bakPassword">hotdb_config</property><!-- Slave configDB password (if configDB uses MGR, this item must be configured) -->
```

```
<property name="configMGR">true</property><!-- Whether configDB uses MGR or not-->

<property name="bak1Url">jdbc:mysql://192.168.210.24:3308/hotdb_config_test250</property><!-- MGR
configDB address (if configDB uses MGR, this item must be configured) -->

<property name="bak1Username">hotdb_config</property><!-- MGR configDB username (if configDB uses
MGR, this item must be configured) -->

<property name="bak1Password">hotdb_config</property><!-- MGR configDB password (if configDB uses
MGR, this item must be configured) -->
```

In case of failure with the Active Master, MGR configDB will re-select the Master Logic according to actual MySQL, and when new Active Master is selected, it will switch to new Master configDB timely.

8.1.3. Compute Node High Availability

HotDB Server supports High Availability Architecture Deployment, which utilizes the principle of keepalived high availability service to build master/slave service relation, and it could guarantee that after failure with the master compute node (that is Activecompute node) service, it will switch to the slave compute node (that is Standbycompute node), and in terms of the application layer, it could have access to the database service via VIP of Keepalived, in order to guarantee the service free of interruption.

8.1.3.1. Startup description

When start the master/slave compute node service under High Availability Architecture, attention shall be paid to the problem of startup sequence, and the following is the standard startup sequence:

1. Firstly, start the Keepalived on the server where the master compute node resides, and then start the master compute node:

View the compute node log:

```
2018-06-13 09:40:04.408 [INFO] [INIT] [Labor-3] j(-1) – HotDB-Manager listening on 3325
2018-06-13 09:40:04.412 [INFO] [INIT] [Labor-3] j(-1) – HotDB-Server listening on 3323
```

View the port listening status:

```
root> ss -npl | grep 3323
LISTEN      0          1000                               *:3323                         *:*
users:(("java",12639,87))

root> ps -aux |grep hotdb
Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.8/FAQ
root      12639  60.7  34.0 4194112  2032134 ? SJun04  7043:58 /usr/java/jdk1.7.0_80/bin/java -
DHOTDB_HOME=/usr/local/hotdb-2.4/hotdb-server -classpath /usr/local/hotdb-2.4/hotdb-server/conf: ...省略更多... -Xdebug -Xrunjdwp:transport=dt_socket,address=8065,server=y,suspend=n -Djava.net.preferIPv4Stack=true cn.hotpu.hotdb.HotdbStartup
```

Using command“ip a”, you could view whether Keepalived VIP of the existing master compute node has been successfully bound or not, for example in the following instance, 192.168.200.190 is the address of the server where master compute node resides; 192.168.200.140 is the configured VIP address

```
root> ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
```

```
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 00:1d:0f:14:8b:fa brd ff:ff:ff:ff:ff:ff
      inet 192.168.200.190/24 brd 192.168.200.255 scope global eth1
        inet 192.168.200.140/24 scope global secondary eth1:1
          inet6 fe80::21d:ff:fe14:8bfa/64 scope link
            valid_lft forever preferred_lft forever
```

- Restart the Keepalived on the server where the slave compute node resides, and restart the slave compute node:

View the compute node log:

```
2018-06-04 18:14:32:321 [INFO] [INIT] [main] j(-1) – Using nio network handler
2018-06-04 18:14:32:356 [INFO] [INIT] [main] j(-1) – HotDB-Manager listening on 3325
2018-06-04 18:14:32:356 [INFO] [AUTHORITY] [checker] Z(-1) – Thanks for choosing HotDB
```

View the port listening status:

```
root> ss -npl | grep 3325
LISTEN      0      1000                  *:3325                  *.**
users:(("java",11603,83))

root> ps -aux |grep hotdb
Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.8/FAQ
root      11603  12.0  13.6 3788976  1086196 ?  Sl   Jun04  1389:44 /usr/java/jdk1.7.0_80/bin/java -
DHOTDB_HOME=/usr/local/hotdb-2.4/hotdb-server -classpath /usr/local/hotdb-2.4/hotdb-server/conf: ...省略更多... -Xdebug -Xrunjdwp:transport=dt_socket,address=8065,server=y,suspend=n -Djava.net.preferIPv4Stack=true cn.hotpu.hotdb.HotdbStartup
```

8.1.3.2. Description of High Availability Switch

In case of failure with the server where master compute node resides (take 192.168.200.190 for instance), and the Detection Script (vrrp_scripts) detects that the service port of the master compute node is Unreachable or more than 3 consecutive hacheck failures, then the priority will be adjusted to 90 (weight-10).

After the keepalived on the server where the slave compute node resides (take 192.168.200.191 for instance) receives vrrp packet inferior to its own priority (the priority of 192.168.200.191 is 95), it will switch to the master status, and preempt the vip (take 192.168.200.140 for instance). Meanwhile, after entering the master status, it will execute the notify_master script, have access to management port of the compute node on 192.168.200.191 to execute online command, start and initialize the service port of the compute node on 192.168.200.191. If this compute node is enabled successfully, then Master/Slave Switch succeeded, and it will continue to provide service. Log of the compute node on 192.168.200.191 is as follow:

```
2018-06-12 21:54:45.128 [INFO] [INIT] [Labor-3] j(-1) – HotDB-Server listening on 3323
2018-06-12 21:54:45.128 [INFO] [INIT] [Labor-3] j(-1) –
=====
2018-06-12 21:54:45.141 [INFO] [MANAGER] [Labor-4] q(-1) – Failed to offline master Because mysql:
[Warning] Using a password on the command line interface can be insecure.
ERROR 2003 (HY000): Can't connect to MySQL server on '192.168.200.190' (111)
2018-06-12 21:54:45.141 [INFO] [RESPONSE] [$NIOREACTOR-8-RW] af(-1) – connection killed for HotDB
backup startup
...More are omitted...
```

VIP of Keepalived has been on the 192.168.200.191 server already:

```
root> ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
```

```
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
link/ether 18:a9:05:1b:0f:a8 brd ff:ff:ff:ff:ff:ff
inet 192.168.200.191/24 brd 192.168.200.255 scope global eth0
inet 192.168.200.140/24 brd 192.168.200.255 scope global secondary eth0:1
inet6 fe80::1aa9:5ff:fe1b:fa8/64 scope link
    valid_lft forever preferred_lft forever
```

Note: For Manual Switch on the management platform, if Switch succeeded, it will modify the configuration in server.xml (haState、haNodeHost), interchange the Master/Slave information, while the configuration will not be modified in case of failover.

8.1.3.3. High Availability Rebuild

The compute node under Master/Slave mode mainly marks the Master/Slave role via the server.xml and keepalived.conf Config File. High Availability Switch shall only from Master role to Slave role, and after Failover or Manual Switch of the compute node, in order to guarantee smooth Failback in case of failure with the compute node next time, it needs to make configuration of the master/slave compute node accurate via the High Availability Rebuild.

For example, through this function it could be realized that: in case of failover with the master compute node (take 192.168.200.190 for instance), after switch to the slave

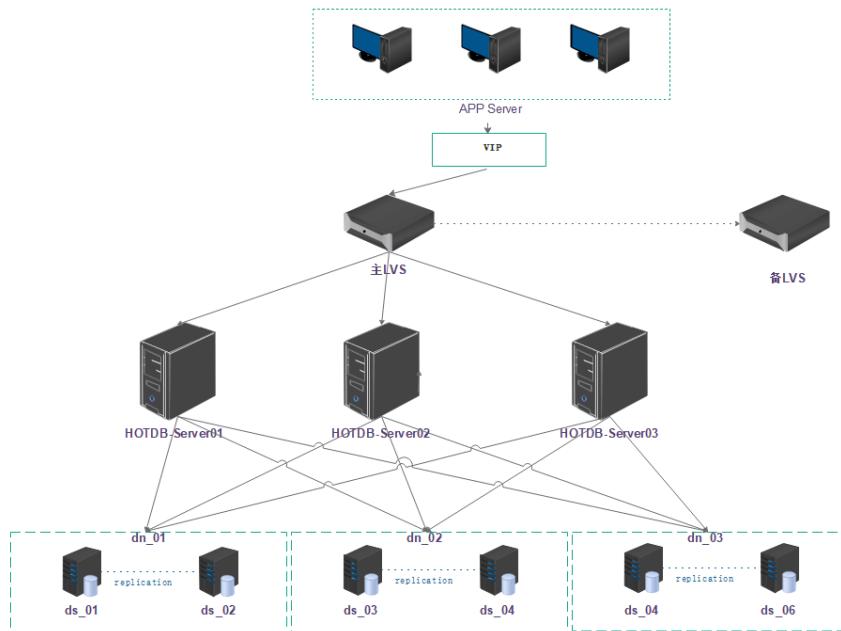
compute node (take 192.168.200.191 for instance), by manual trigger of Compute Node High Availability Rebuild function, the compute node service and Master/Slave relation on the 192.168.200.190 server could be recovered, and in case of failure again with the compute node on the 192.168.200.191 server, it could fallback to the 192.168.200.190 automatically. For details of High Availability Rebuild, please refer to *Distributed Transactional Database HotDB Server [Management Platform] Function Manual*.

8.1.4. Load-balancing of compute node

HotDB Server supports node autonomy of multiple compute node cluster. Hereinafter, the compute node in Primary status in the compute node cluster is referred to as Primary Compute Node; the compute node in Secondary status in the compute node cluster is referred to as Secondary Compute Node. Data service of both Primary and Secondary Compute Node is completely equivalent, and they both support all types of data operation and guarantee data consistency. After failure with one or more nodes in the cluster (not support failure with multiple compute nodes simultaneously), as long as there is one or more nodes available, the whole data service shall still be available.

HotDB Server supports load-balancing: you could select LVS and other means to distribute SQL request. The application client could have access to database service of HotDB Server via VIP of LVS, and at the same time, use transparency and service uninterrupted are guaranteed. It could also use the remaining load-balancing plans for processing, for example F5 plus Custom Test; apply the mode of direct connection compute node, but replace node in case of abnormality, etc.

分布式事务数据库HotDB Server 负载均衡部署架构示意图



8.1.4.1. Startup description

After environment deployment of compute node cluster succeeded, start the compute node:

```
#cd /usr/local/hotdb-2.5.0/hotdb-server/bin  
  
#sh hotdb server start
```

View start status of HotDB Server:

After compute node is enabled, it doesn't enable service port, and will set its role status as Started for the time being. After all compute nodes in the cluster start, one compute node of them will become Primary, while the remaining compute nodes will become Secondary, and all compute nodes have their data service port open, and then the whole cluster comes into normal running status, for example:

Primary node:

```

2018-10-11 14:43:26.521 [INFO] [INIT] [main] HotdbServer(475) - Using nio network handler
2018-10-11 14:43:26.548 [INFO] [INIT] [main] HotdbServer(496) - HotDB-Manager listening on 3325
2018-10-11 14:43:26.584 [ERROR] [AUTHORITY] [checker] Validation(128) - USB-Key hard lock check failed. Error code:50, Error info:HASP_SCOPE_RESULTS_EMPTY:Unable to locate any Feature matching scope.
2018-10-11 14:43:29.670 [WARN] [AUTHORITY] [checker] Validation(271) - HotDB Trial expires in 482 hours and 21 minutes and 54 seconds.
2018-10-11 14:43:32.829 [INFO] [CLUSTER] [Packet-Handler-3] ClusterNode(467) - Cluster state changes: [started]->[candidate]
2018-10-11 14:43:32.843 [INFO] [CLUSTER] [Packet-Handler-3] ClusterNode(1168) - received vote HotDBVotePacket, From:192.168.200.4:3326:3:3323:3325, ToIP:192.168.200.3, ToPort:3326.
2018-10-11 14:43:32.844 [INFO] [CLUSTER] [Packet-Handler-3] ClusterNode(1191) - ack vote HotDBAckPacket, From:192.168.200.3:3326:4:3323:3325, ToIP:192.168.200.4, ToPort:3326.
2018-10-11 14:43:32.845 [INFO] [CLUSTER] [Packet-Handler-2] ClusterNode(1168) - received vote HotDBVotePacket, From:192.168.200.2:3326:3:3323:3325, ToIP:192.168.200.3, ToPort:3326.
2018-10-11 14:43:32.847 [INFO] [CLUSTER] [Packet-Handler-2] ClusterNode(1191) - ack vote HotDBAckPacket, From:192.168.200.3:3326:4:3323:3325, ToIP:192.168.200.2, ToPort:3326.
2018-10-11 14:43:32.847 [INFO] [CLUSTER] [Packet-Handler-2] ClusterNode(1191) - enough vote 3, target 3
2018-10-11 14:43:32.848 [INFO] [CLUSTER] [Packet-Handler-2] ClusterNode(467) - Cluster state changes: [candidate]->[primary]
2018-10-11 14:43:32.892 [INFO] [INIT] [$NIOExecutor-3-0] BackendDataSource(282) - datasource [id:-1,nodeId:-1 192.168.210.43:3308/hotdb_config_test250 status:0;charset:utf8] is unavailable and will not be initied.
2018-10-11 14:43:33.033 [INFO] [INIT] [$NIOExecutor-3-0] BackendDataSource(626) - init datasource success,initSize= 8,success=8,ds=[id:-2,nodeId:-1 192.168.210.44:3308/hotdb_config_test250 status:1;charset:utf8]
2018-10-11 14:43:33.048 [INFO] [INIT] [$NIOExecutor-3-0] HotdbServer(648) - init hotdb_config datanode success.
2018-10-11 14:43:33.049 [INFO] [INIT] [$NIOExecutor-3-0] HotdbServer(652) - Initialize data nodes ...
2018-10-11 14:43:33.178 [INFO] [INIT] [$NIOExecutor-3-0] BackendDataSource(626) - init datasource success,initSize= 32,success=32,ds=[id:1,nodeId:1 192.168.210.43:3309/multi_zj_jjjstatus:1;charset:utf8mb4]
2018-10-11 14:43:33.181 [WARN] [INNER] [$NIOREACTOR-5-RW] InitSlaveStatusHandler(85) - Datasource: 1's master datasource: 192.168.210.44:3309 not in datanode: 1
2018-10-11 14:43:33.190 [INFO] [INIT] [$NIOExecutor-3-0] HotdbServer(664) - init datanode 1 success. current datasource:[id:1,nodeId:1 192.168.210.43:3309/multi_zj_jjj status:1,ccharset:utf8mb4]
2018-10-11 14:43:33.442 [INFO] [INIT] [$NIOExecutor-4-RW] HotdbServer(2099) - persist sequence at abnormal starting server.
2018-10-11 14:43:33.449 [INFO] [INIT] [$NIOREACTOR-4-RW] XAInitRecoverHandler(128) - wait for config datasource synchronizing...
2018-10-11 14:43:33.512 [INFO] [INIT] [$NIOREACTOR-4-RW] XAInitRecoverHandler(203) - wait datanodes synchronizing to recover XA transactions.
2018-10-11 14:43:33.557 [INFO] [INIT] [$NIOREACTOR-12-RW] XARecoverHandler(125) - no xa recover result
2018-10-11 14:43:33.587 [INFO] [INIT] [$NIOREACTOR-7-RW] XARecoverHandler(913) - persist XID at abnormal starting server.
2018-10-11 14:43:33.606 [INFO] [INIT] [Labor-41] HotdbServer(1181) - HotDB-Server listening on 3233
2018-10-11 14:43:33.615 [INFO] [INIT] [Labor-41] HotdbServer(1183) - =====
2018-10-11 14:43:33.723 [INFO] [INIT] [Labor-17] BackendDataSource(626) - init datasource success,initSize= 4,success=4,ds=[id:-101,nodeId:-101 192.168.200.2:3325/null status:1,charset:utf8mb4]
2018-10-11 14:43:33.928 [INFO] [INIT] [Labor-17] BackendDataSource(626) - init datasource success,initSize= 4,success=4,ds=[id:-102,nodeId:-102 192.168.200.2:3325/null status:1,charset:utf8mb4]

```

Secondary node:

```

2018-10-11 14:43:31.731 [INFO] [INIT] [main] HotdbServer(475) - Using nio network handler
2018-10-11 14:43:31.754 [INFO] [INIT] [main] HotdbServer(496) - HotDB-Manager listening on 3325
2018-10-11 14:43:32.831 [INFO] [CLUSTER] [Packet-Handler-4] ClusterNode(467) - Cluster state changes: [started]->[secondary]
2018-10-11 14:43:33.929 [INFO] [INIT] [$NIOExecutor-1-1] Online(68) - received online from: [thread=$NIOExecutor-1-1,id=5,user=_HotDB_Cluster_USER_,host=192.168.200.3,port=3325,localport=50894,schema=null]
2018-10-11 14:43:33.931 [INFO] [MANAGER] [$NIOExecutor-1-1] Online(164) - received online command from:[thread=$NIOExecutor-1-1,id=5,user=_HotDB_Cluster_USER_,host=192.168.200.3,port=3325,localport=50894,schema=null]
2018-10-11 14:43:34.027 [INFO] [INIT] [$NIOExecutor-1-1] HotdbConfig(405) - master config datasource [id:-1,nodeId:-1 192.168.210.43:3308/hotdb_config_test250 status:1;charset:utf8] connect success.
2018-10-11 14:43:34.027 [INFO] [INIT] [$NIOExecutor-1-1] HotdbConfig(409) - backup config datasource [id:-2,nodeId:-1 192.168.210.44:3308/hotdb_config_test250 status:1;charset:utf8] connect success.
2018-10-11 14:43:34.031 [WARN] [INIT] [$NIOExecutor-1-1] HotdbConfig(418) - master config datasource [id:-1,nodeId:-1 192.168.210.43:3308/hotdb_config_test250 status:1;charset:utf8] is unavailable.
2018-10-11 14:43:34.032 [WARN] [INIT] [$NIOExecutor-1-1] HotdbConfig(622) - using backup config in start up:[id:-2,nodeId:-1 192.168.210.44:3308/hotdb_config_test250 status:1,charset:utf8]
2018-10-11 14:43:34.325 [INFO] [INIT] [$NIOExecutor-1-1] BackendDataSource(282) - datasource [id:-1,nodeId:-1 192.168.210.43:3308/hotdb_config_test250 status:0;charset:utf8] is unavailable and will not be initied.
2018-10-11 14:43:34.466 [INFO] [INIT] [$NIOExecutor-1-1] BackendDataSource(626) - init datasource success,initSize= 8,success=8,ds=[id:-2,nodeId:-1 192.168.210.44:3308/hotdb_config_test250 status:1;charset:utf8]
2018-10-11 14:43:34.482 [INFO] [INIT] [$NIOExecutor-1-1] HotdbServer(648) - init hotdb_config datanode success.
2018-10-11 14:43:34.482 [INFO] [INIT] [$NIOExecutor-1-1] HotdbServer(652) - Initialize data nodes ...
2018-10-11 14:43:34.619 [INFO] [INIT] [$NIOExecutor-1-1] BackendDataSource(626) - init datasource success,initSize= 32,success=32,ds=[id:1,nodeId:1 192.168.210.43:3309/multi_zj_jjjstatus:1,charset:utf8mb4]
2018-10-11 14:43:34.623 [WARN] [INNER] [$NIOREACTOR-9-RW] InitSlaveStatusHandler(85) - Datasource: 1's master datasource: 192.168.210.44:3309 not in datanode: 1
2018-10-11 14:43:34.630 [INFO] [INIT] [$NIOExecutor-1-1] HotdbServer(664) - init datanode 1 success. current datasource:[id:1,nodeId:1 192.168.210.43:3309/multi_zj_jjj status:1,ccharset:utf8mb4]
2018-10-11 14:43:34.676 [ERROR] [AUTHORITY] [checker] Validation(128) - USB-Key hard lock check failed. Error code:50, Error info:HASP_SCOPE_RESULTS_EMPTY:Unable to locate any Feature matching scope
2018-10-11 14:43:34.779 [INFO] [INIT] [Labor-1] BackendDataSource(626) - init datasource success,initSize= 4,success=4,ds=[id:-100,nodeId:-100 192.168.200.3:3323/null status:1,charset:utf8mb4]
2018-10-11 14:43:35.957 [WARN] [AUTHORITY] [checker] Validation(271) - HotDB Trial expires in 482 hours and 21 minutes and 39 seconds.
2018-10-11 14:43:35.311 [INFO] [INIT] [$NIOREACTOR-10-RW] HotdbServer(2099) - persist sequence at abnormal starting server.
2018-10-11 14:43:35.337 [INFO] [INIT] [Labor-33] HotdbServer(1181) - HotDB-Server listening on 3233
2018-10-11 14:43:35.339 [INFO] [INIT] [Labor-33] HotdbServer(1183) - =====
2018-10-11 14:43:35.453 [INFO] [INIT] [Labor-32] BackendDataSource(626) - init datasource success,initSize= 4,success=4,ds=[id:-100,nodeId:-100 192.168.200.3:3323/null status:1,charset:utf8mb4]
2018-10-11 14:43:35.463 [INFO] [CLUSTER] [$NIOExecutor-1-0] EnableIUD(41) - received enable_iud from [thread=$NIOExecutor-1-0,id=5,user=_HotDB_Cluster_USER_,host=192.168.200.3,port=3325,localport=50894,schema=null]
2018-10-11 14:43:40.352 [INFO] [CLUSTER] [$NIOExecutor-1-2] ConfigSync(26) - received config sync from [thread=$NIOExecutor-1-2,id=5,user=_HotDB_Cluster_USER_,host=192.168.200.3,port=3325,localport=50894,schema=null]

```

View management port 3325 status:

```
[root@datasource_ms ~]# mysql -ujzl -pjzl -h192.168.200.2 -P3325
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 5.7.19-HotDB-2.5.0 HotDB Manager by Hotpu Tech

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

jzl@192.168.200.2:(none) 5.7.19-HotDB-2.5.0 02:28:22> show @@cluster;
+-----+-----+-----+-----+
| status | host   | port | server_port | manager_port |
+-----+-----+-----+-----+
| PRIMARY | 192.168.200.3 | 3326 | 3323      | 3325      |
| SECONDARY | 192.168.200.2 | 3326 | 3323      | 3325      |
| SECONDARY | 192.168.200.4 | 3326 | 3323      | 3325      |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

When Primary service is abnormal, one from the remaining Secondary will become the new Primary, and the original Primary will be kicked out of the cluster.

Original Primary:

```
#cd /usr/local/hotdb-2.5.0/hotdb-server/bin
#sh hotdb_server stop
```

```
2018-10-11 14:54:13.933 [INFO] [EXIT[ FLOW ]] [ShutdownHook] HotdbServer(517) - begin to exit...
2018-10-11 14:54:14.000 [WARN] [CONNECTION] [ShutdownHook] NIODceptor(175) - HotDB SocketChannel close due to:System exit
2018-10-11 14:54:14.001 [WARN] [CONNECTION] [ShutdownHook] NIODceptor(175) - HotDB SocketChannel close due to:System exit
2018-10-11 14:54:14.006 [WARN] [CONNECTION] [ShutdownHook] IOPProcessor(325) - processor close due to:System exit
2018-10-11 14:54:14.006 [WARN] [CONNECTION] [ShutdownHook] IOPProcessor(325) - processor close due to:System exit
2018-10-11 14:54:14.007 [WARN] [CONNECTION] [ShutdownHook] IOPProcessor(325) - processor close due to:System exit
2018-10-11 14:54:14.008 [WARN] [CONNECTION] [ShutdownHook] IOPProcessor(325) - processor close due to:System exit
2018-10-11 14:54:14.009 [WARN] [CONNECTION] [ShutdownHook] IOPProcessor(325) - processor close due to:System exit
2018-10-11 14:54:14.010 [WARN] [CONNECTION] [ShutdownHook] IOPProcessor(325) - processor close due to:System exit
2018-10-11 14:54:14.011 [WARN] [CONNECTION] [ShutdownHook] IOPProcessor(325) - processor close due to:System exit
2018-10-11 14:54:14.012 [WARN] [CONNECTION] [ShutdownHook] IOPProcessor(325) - processor close due to:System exit
2018-10-11 14:54:14.013 [WARN] [CONNECTION] [ShutdownHook] IOPProcessor(325) - processor close due to:System exit
2018-10-11 14:54:14.014 [WARN] [CONNECTION] [ShutdownHook] IOPProcessor(325) - processor close due to:System exit
2018-10-11 14:54:14.015 [WARN] [CONNECTION] [ShutdownHook] IOPProcessor(325) - processor close due to:System exit
2018-10-11 14:54:14.016 [WARN] [CONNECTION] [ShutdownHook] IOPProcessor(325) - processor close due to:System exit
2018-10-11 14:54:14.017 [WARN] [CONNECTION] [ShutdownHook] IOPProcessor(325) - processor close due to:System exit
2018-10-11 14:54:14.020 [WARN] [EXIT[ FLOW ]] [ShutdownHook] HotdbServer(609) - begin to update hotdb_server_status...
2018-10-11 14:54:14.022 [WARN] [EXIT[ FLOW ]] [Labor-17] HotdbServer(550) - begin to update hotdb_table_sequence...
2018-10-11 14:54:14.042 [INFO] [INIT] [$NIOREACTOR-5-RW] HotdbServer(2102) - persist sequence at stopping server.
2018-10-11 14:54:14.043 [WARN] [EXIT[ FLOW ]] [Labor-17] HotdbServer(556) - begin to persist XID...
2018-10-11 14:54:14.044 [INFO] [INIT] [$NIOREACTOR-5-RW] HotdbServer(916) - persist XID at stopping server.
2018-10-11 14:54:14.045 [WARN] [EXIT[ FLOW ]] [Labor-17] HotdbServer(571) - begin to clear onlineddl temporary tables...
2018-10-11 14:54:14.069 [WARN] [EXIT[ FLOW ]] [Labor-41] HotdbServer(587) - begin to clear global consistency check result...
2018-10-11 14:54:14.080 [INFO] [INNER] [Labor-41] BackendDataNode(1446) - clear datanode 1, reason : system exit
2018-10-11 14:54:14.081 [INFO] [INNER] [Labor-41] BackendDataNode(1452) - clear datasource [id:1,nodeId:1 192.168.210.43:3309/multi_zjj status:1;charset:utf8mb4],size:49, reason:system exit
```

New Primary:

```
2018-10-11 14:54:18.864 [INFO] [INNER] [Timer-2] BackendDataNode(1446) - clear datanode -100, reason : primary heartbeat timeout
2018-10-11 14:54:18.865 [INFO] [INNER] [Timer-2] BackendDataNode(1452) - clear datasource [id:-100,nodeId:-100 192.168.200.3:3325/null status:1;charset:utf8mb4],size:0, reason :primary heartbeat timeout
2018-10-11 14:54:18.865 [INFO] [INNER] [Timer-2] BackendDataNode(1446) - clear datanode -100, reason : primary heartbeat timeout
2018-10-11 14:54:18.866 [INFO] [INNER] [Timer-2] BackendDataNode(1452) - clear datasource [id:-100,nodeId:-100 192.168.200.3:3323/null status:1;charset:utf8mb4],size:0, reason :primary heartbeat timeout
2018-10-11 14:54:18.866 [INFO] [CLUSTER] [Timer-2] ClusterNode(467) - Cluster state changes: [secondary]-->[candidate]
2018-10-11 14:54:18.873 [INFO] [CLUSTER] [Packet-Handler-3] ClusterNode(1168) - received vote HotDBVotePacket, From:192.168.200.4:3326:3:3323:3325, ToIP:192.168.200.2, ToPort:3326.
2018-10-11 14:54:18.874 [INFO] [CLUSTER] [Packet-Handler-3] ClusterNode(1191) - ack vote HotDBAckPacket, From:192.168.200.2:3326:4:3323:3325, ToIP:192.168.200.4, ToPort:3326.
2018-10-11 14:54:18.874 [INFO] [CLUSTER] [Packet-Handler-3] ClusterNode(1193) - enough vote 2, target 2
2018-10-11 14:54:18.874 [INFO] [CLUSTER] [Packet-Handler-3] ClusterNode(467) - Cluster state changes: [candidate]-->[primary]
2018-10-11 14:54:18.878 [INFO] [CLUSTER] [Packet-Handler-3] ClusterNode(581) - send replace:REPLACE INTO hotdb_heartbeat VALUES(10,now(),'19216820023326') to config datasource
```

View management port status:

```
[root@datasource_ms ~]# mysql -ujz1 -pjz1 -h192.168.200.2 -P3325
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 243
Server version: 5.7.19-HotDB-2.5.0 HotDB Manager by Hotpu Tech

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

jzl@192.168.200.2:(none) 5.7.19-HotDB-2.5.0 02:57:25> show @@cluster;
+-----+-----+-----+-----+
| status | host   | port | server_port | manager_port |
+-----+-----+-----+-----+
| PRIMARY | 192.168.200.2 | 3326 | 3323      | 3325      |
| SECONDARY | 192.168.200.4 | 3326 | 3323      | 3325      |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

If the original Primary service restarts (equivalent to Add of new node), for the Primary node found to be enabled at present, new node will join the cluster to be Secondary.

Original Primary:

```
#cd /usr/local/hotdb-2.5.0/hotdb-server/bin
```

```
#sh hotdb_server start
```

```

2018-10-11 14:58:25.704 [INFO] [CLUSTER] [Packet-Handler-3] ClusterNode(1222) - received member update HotDBMembersPacket, members:[192.168.200.3:3326:3:3323:3325, 192.168.200
.2:3326:2;3323:3325, 192.168.200.4:3326:3:3323:3325]
2018-10-11 14:58:25.740 [INFO] [CLUSTER] [Packet-Handler-3] ClusterNode(1275) - ack member update HotDBVotePacket, From:192.168.200.3:3326:1:3323:3325, ToIP:192.168.200.2, ToP
ort:3326.
2018-10-11 14:58:25.742 [INFO] [CLUSTER] [Packet-Handler-2] ClusterNode(467) - Cluster state changes: [started]->[secondary]
2018-10-11 14:58:25.832 [INFO] [] [$NIOExecutor-3-1] Online(68) - received online from: [thread=$NIOExecutor-3-1,id=10,user=_HotDB_Cluster_USER_,host=192.168.200.2,port=3325,1
ocalport=3325,schema=null]
2018-10-11 14:58:25.834 [INFO] [MANAGER] [$NIOExecutor-3-1] Online(164) - received online command from:[thread=$NIOExecutor-3-1,id=10,user=_HotDB_Cluster_USER_,host=192.168.20
.2,port=3325,localport=34090,schema=null]
2018-10-11 14:58:25.915 [INFO] [INIT] [$NIOExecutor-3-1] HotdbConfig(405) - master config datasource [id:-1,nodeId:-1 192.168.210.43:3308/hotdb_config_test250 status:1;charset
:utf8] connect success.
2018-10-11 14:58:25.916 [INFO] [INIT] [$NIOExecutor-3-1] HotdbConfig(409) - backup config datasource [id:-2,nodeId:-1 192.168.210.44:3308/hotdb_config_test250 status:1;charset
:utf8] connect success.
2018-10-11 14:58:25.919 [WARN] [INIT] [$NIOExecutor-3-1] HotdbConfig(418) - master config datasource [id:-1,nodeId:-1 192.168.210.43:3308/hotdb_config_test250 status:1;charset
:utf8] is unavailable.
2018-10-11 14:58:25.920 [WARN] [INIT] [$NIOExecutor-3-1] HotdbConfig(622) - using backup config in start up:[id:-2,nodeId:-1 192.168.210.44:3308/hotdb_config_test250 status:1,
localport=405]
2018-10-11 14:58:26.148 [INFO] [INIT] [$NIOExecutor-3-1] BackendDataNode(282) - datasource [id:-1,nodeId:-1 192.168.210.43:3308/hotdb_config_test250 status:0;charset:utf8] is
unavailable and will not be init.
2018-10-11 14:58:26.288 [INFO] [INIT] [$NIOExecutor-3-1] BackendDatasource(626) - init datasource success,initSize= 8,success=8,ds=[id:-2,nodeId:-1 192.168.210.44:3308/hotdb_c
onfig_test250 status:1;charset:utf8]
2018-10-11 14:58:26.303 [INFO] [INIT] [$NIOExecutor-3-1] HotdbServer(648) - init hotdb_config datanode success.
2018-10-11 14:58:26.303 [INFO] [INIT] [$NIOExecutor-3-1] HotdbServer(652) - Initialize data nodes ...
2018-10-11 14:58:26.303 [INFO] [INIT] [$NIOExecutor-3-1] BackendDatasource(626) - init datasource success,initSize= 32,success=32,ds=[id:1,nodeId:1 192.168.210.43:3309/multi_z
jj,status:1;charset:utf8mb4]
2018-10-11 14:58:26.437 [WARN] [SNIOREACTOR-3-RW] InitSlaveStatusHandler(85) - Datasource: 1's master datasource: 192.168.210.44:3309 not in datanode: 1
2018-10-11 14:58:26.446 [INFO] [INIT] [$NIOExecutor-3-1] HotdbServer(664) - init datanode 1 success. current datasource:[id:1,nodeId:1 192.168.210.43:3309/multi_zjj status:1,c
harset:utf8mb4]
2018-10-11 14:58:26.570 [INFO] [INIT] [Labor-1] BackendDatasource(626) - init datasource success,initSize= 4,success=4,ds=[id:-100,nodeId:-100 192.168.200.2:3323/null status:1
;charset:utf8mb4]
2018-10-11 14:58:26.987 [INFO] [INIT] [$NIOREACTOR-8-RW] HotdbServer(2099) - persist sequence at abnormal starting server.
2018-10-11 14:58:27.017 [INFO] [INIT] [Labor-52] HotdbServer(1181) - HotDB-Server listening on 3323
2018-10-11 14:58:27.020 [INFO] [INIT] [Labor-52] HotdbServer(1183) - =====
2018-10-11 14:58:27.131 [INFO] [INIT] [Labor-41] BackendDatasource(626) - init datasource success,initSize= 4,success=4,ds=[id:-100,nodeId:-100 192.168.200.2:3325/null status:
1,charset:utf8mb4]
2018-10-11 14:58:27.140 [INFO] [CLUSTER] [$NIOExecutor-3-2] EnableIUD(41) - received enable_iud from [thread=$NIOExecutor-3-2,id=10,user=_HotDB_Cluster_USER_,host=192.168.200.
2,port=3325,localport=34090,schema=null]
    
```

View management port status:

```

[root@datasource_ms ~]# mysql -ujz1 -pjz1 -h192.168.200.2 -P3325
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 405
Server version: 5.7.19-HotDB-2.5.0 HotDB Manager by Hotpu Tech

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

jzl@192.168.200.2:(none) 5.7.19-HotDB-2.5.0 03:00:38> show @@cluster;
+-----+-----+-----+-----+
| status | host   | port | server_port | manager_port |
+-----+-----+-----+-----+
| SECONDARY | 192.168.200.3 | 3326 | 3323      | 3325      |
| PRIMARY   | 192.168.200.2 | 3326 | 3323      | 3325      |
| SECONDARY | 192.168.200.4 | 3326 | 3323      | 3325      |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
    
```

After multiple compute node cluster starts, by accessing to the database service via VIP, transparent load-balancing could be realized, and uninterrupted service could be guaranteed.

```
[root@datasource_ms ~]# mysql -uroot -proot -h192.168.200.33 -P3323
mysql: [warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36003
Server version: 5.6.29-HotDB-2.5.0 HotDB Server by Hotpu Tech

copyright (c) 2000, 2018, oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

root@192.168.200.33:(none) 5.6.29-HotDB-2.5.0 10:30:11> use test;
Database changed
root@192.168.200.33:test 5.6.29-HotDB-2.5.0 10:30:14> select * from father;
+----+
| id |
+----+
| 2 |
| 1 |
+----+
2 rows in set (0.04 sec)
```

Notices for multiple compute node cluster:

1. When compute node cluster starts, the Primary is random, the compute node on the server where the Master configDB resides can't become Primary;
2. After a certain period, the compute node with failure will disable the service port independently to become Started status;
3. In case the Secondary finds that the Primary loses response and itself is not on the server where Master configDB resides, it will launch new Election, and will become the new Primary if receiving the majority votes;
4. Add new node, and the Primary will find join of the new Started node, and will Add the new node found; if the Primary finds that the Secondary loses response, it will kick out this node;
5. For cluster environment upgrade version, if the business is not influenced, it's recommended enabling after disabling the Cluster Upgrade;
6. For server.xml configuration of various nodes in the cluster, all parameters must be consistent except the cluster related parameters;
7. Time difference of various compute node servers shall be less than 1s;
8. It's required that network latency between compute node servers shall be less than 1s at any time;

9. It's recommended deploying only a set of multiple compute node cluster within a local-area network segment (It's just a suggestion, not a mandatory requirement. The reason for the suggestion is to reserve more space for future expansion);
10. configDB IP requires configuration of actual IP.

8.1.4.2. Linear expansion

In the multi-node cluster mode, if you want to achieve linear growth of performance throughput in the strong consistency (XA) mode with the expansion performance of the number of compute nodes, you can use HotDB Listener components.

8.1.4.2.1. Deployment of Listener

Please refer to the Distributed Transactional Database HotDB Server - [Installation & Deployment] Function Manual for the deployment of Listener components.

8.1.4.2.2. Use of Listener

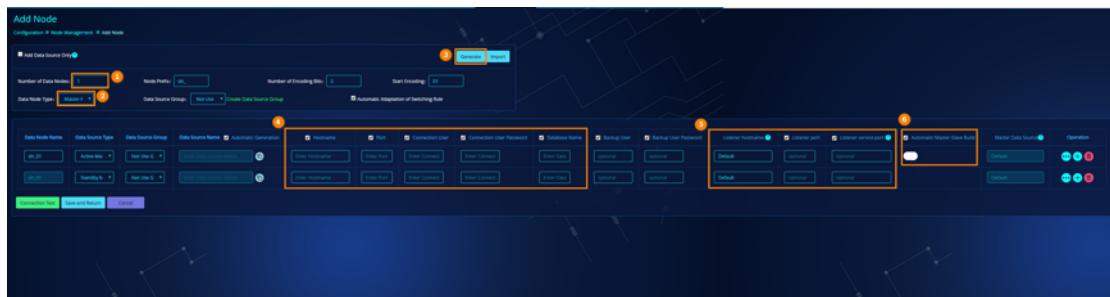
These requirements must be met before using the Listener:

- Compute node version is 2.5.5 and above
- Multi-node cluster mode
- XA mode is enabled
- The parameter [enableListener](#) is set to true in server.xml.

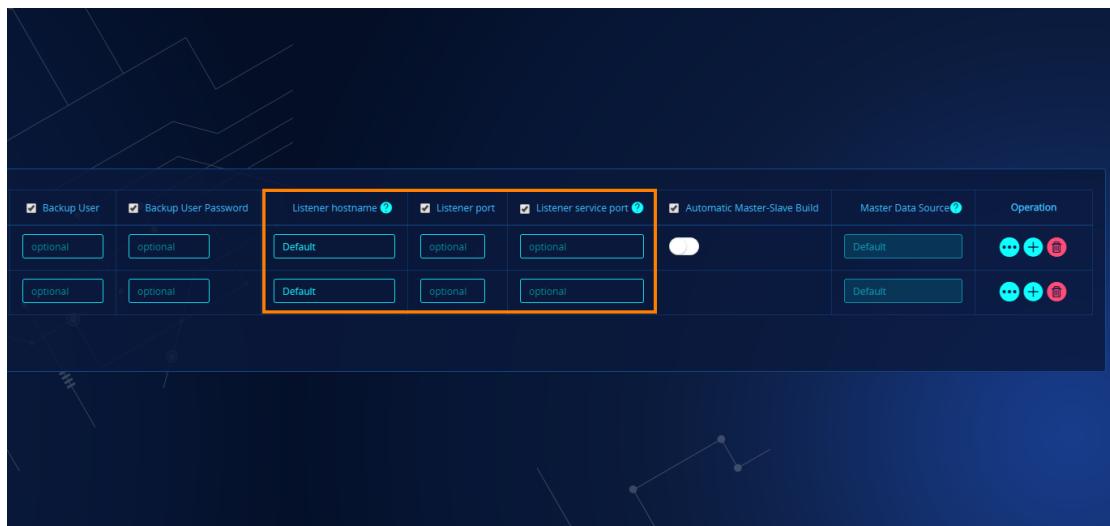
8.1.4.2.2.1. Add node configuration of Listener

After the Listener deployment is completed, the configuration of Listener can be introduced onto the data node page.

Take adding a group of data nodes with master-master type as an example:



In steps 1-4, fill in the host name and port number of the data source, connection user and password, database, etc. according to the previous rules. If the group of data sources needs to bind Listener, fill in the Listener related information in step 5.



The rules are as follows:

- Listener host name: "default" by default, no need to modify
- Listening port: the start port of Listener. 3330 by default
- Listener service port: the port where Listener provides services to HotDB Server. 4001 by default. If there are multiple MySQL instances on the same data source server that need to bind Listener, the service port needs to be unique.

After filling is achieved, click Connection Test, and click Save and Return after passing the test.

In dynamic loading, if the status on the node management list is **可用**, it means that the Listener can be connected; if the status is **不可用**, it means that the Listener cannot be connected. You should check whether enableXA is true and enableListener is true.

Verify whether the Listener service is enabled or not: execute show @@datasource on port 3325 to view.

8.1.4.2.2.2. Edit node configuration of Listener

This method is suitable for adding Listener configuration based on existing data nodes.

On the node management page, take dn_26 as an example:



DNID	Data Node Name	IP	Data source					Master/Slave Status	Master/Slave Switch
			Hostname	Port	Database	Type	Status		
26	dn_26	192.168.210.44	3307	zy157	Active Master	Available			
25	dn_25	192.168.210.43	3308	zy157	Active Master	Available			
24	dn_24	192.168.210.42	3308	zy157	Active Master	Available			
23	dn_23	192.168.210.41	3308	zy157	Active Master	Available			

Click the icon *i* in the operation bar, i.e. information, to enter the data source management page.

Data Source Management

Configuration » Node Management » Data Source Management

Node Selection: dn_26

Select Data Source: 192.168.210.44_3307_zy157

Data Source Information

Data Source Group	No Group
Hostname	192.168.210.44
Port	3307
Connection User	hotdb_datasource
Backup User	
Database Name	zy157
Data Source Type	Active Master
Data Source Status	Available
Character Set	utf8mb4
Max Connections	4200
Initial Connections	32
Max Idle Connections	512
Min Idle Connections	32
Idle Examination Period (second)	600
Master Data Source Name	<input type="text" value="None"/>
Listener hostname	
Listener port	
Listener service port	

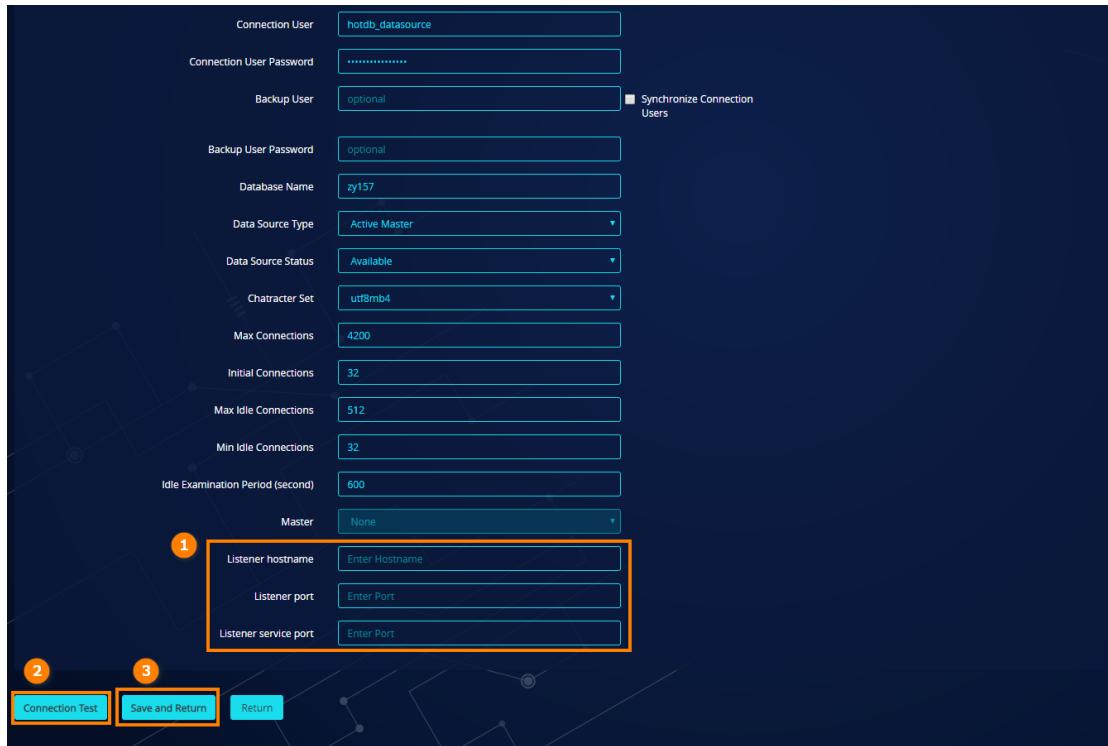
Edit

Return

The screenshot shows the 'Data Source Management' page of the HotDB Server function manual. It displays configuration details for a data source named 'zy157'. The 'Data Source Information' section lists various parameters like Hostname (192.168.210.44), Port (3307), and Connection User (hotdb_datasource). A specific row for 'Master Data Source Name' has a dropdown menu set to 'None'. Below this, there is a group of three input fields labeled 'Listener hostname', 'Listener port', and 'Listener service port', all of which are currently empty. An orange rectangle highlights this group of fields. At the bottom right of the form is a blue 'Edit' button.

For a data source that is not bound to a Listener, the last three items are empty by default.

Click Edit to add information about the Listener.



The rules are as follows:

- Listener host name: "default" by default, no need to modify
- Listener port: the start port of Listener. 3330 by default
- Listener service port: the port where Listener provides services to HotDB Server. 4001 by default. If there are multiple MySQL instances on the same data source server that need to bind Listener, the service port needs to be unique.

After filling is achieved, click Connection Test, and click Save and Return after passing the test.

In dynamic loading, if the status on the node management list is **可用**, it means that the Listener can be connected; if the status is **不可用**, it means that the Listener cannot be connected. You should check whether enableXA is true and enableListener is true.

Verify whether the Listener service is enabled or not: execute show @@datasource on port 3325 to view.

8.1.4.2.3. Notes

1. Once deployed and correctly identified, the Listener is of no concern to users in the process of daily compute node operation;
2. Listener components should be installed on the same server as the data source;
3. If a Listener needs to listen to multiple data sources, different service ports need to be filled in;
4. When a certain data source is cancelled to be listened by Listener, the allocated Listener service port will always exist. The original data source can use the Listener service port to bind the Listener again.
5. When a certain data source is cancelled to be listened by Listener, the allocated Listener service port will always exist. At this time, if other data sources use the Listener service port, the Listener log will report an error: Port conflicts. Port already exists. Therefore, you need to restart Listener before you can use the Listener service port.
6. When the cluster needs to be restarted, it is recommended that the Listener components be restarted together. The restart sequence is: restart Listener first, and then restart the cluster, so that the cluster can recognize the Listener faster;
7. As a pluggable component, when Listener is not available, the cluster and data sources can still provide services.

8.1.5. Compute node auto scaling

In order to satisfy the demand for business development and application data growth, HotDB Server V2.5.6 supports the online expansion/reduction function of compute nodes, by manually adjusting the relevant parameters of the compute node server.xml and dynamically loading. For example, the single-node mode can be

expanded to HA mode or cluster mode and the cluster mode can be reduced to HA mode or single-node mode.

8.1.5.1. Expansion of compute nodes

8.1.5.1.1. Parameter introduction

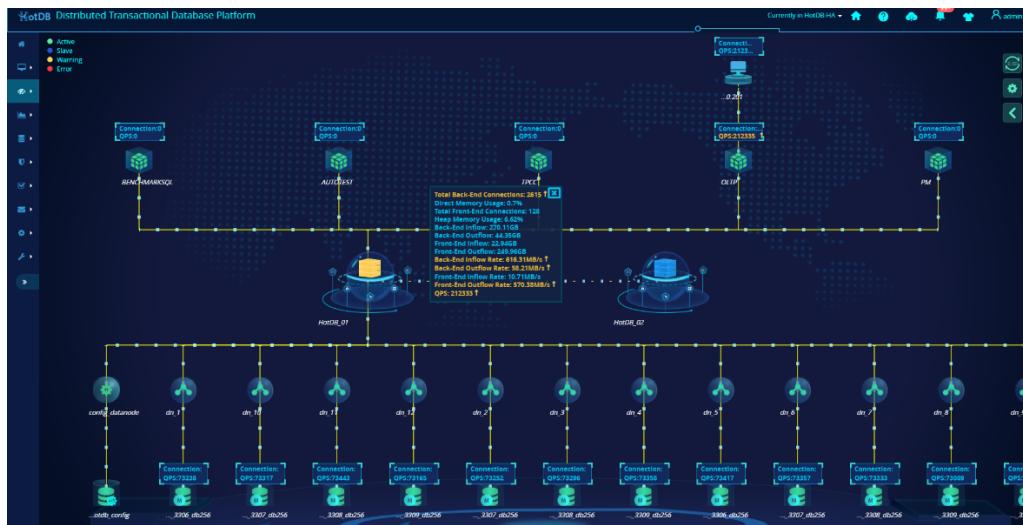
Parameters involved are as follows:

Parameters	Instructions	Reference value	Whether the reloading is valid
haMode	High availability mode: 0: master and slave; 1: cluster	In cluster environment, the parameter value is 1	yes
serverId	Cluster node number 1-N (number of nodes), unique in the cluster and $N \leq$ total number of nodes in the cluster	The serverID should start from 1 and should not be repeated continuously in the cluster	yes
clusterName	Cluster group name	HotDB-Cluster	yes
clusterSize	Total number of cluster nodes	The default value is 3, which is configured according to the actual number of nodes	yes
clusterNetwork	Network segment of cluster	192.168.200.0/24, same network segment as cluster IP	yes
clusterHost	IP address of this node	192.168.200.1, matched according to the specific IP	yes
clusterPort	Cluster communication port	3326 by default	yes

8.1.5.1.2. Expand from HA mode to multi-node mode

Expand from HA mode to multi-node mode is mainly about how to switch keepalived to LVS. This section mainly describes the expansion operation of HA to cluster. The information of components involved is as follows:

Role	Connection information	Name
Master compute node	192.168.210.67_3323_3325	HotDB_01
Slave compute node	192.168.210.68_3325	HotDB_02
LVS service	192.168.210.136	VIP:192.168.210.218
New compute node	192.168.210.134	HotDB_03



Step 1: Disable standby compute node / standby keepalived service

Disable keepalived and compute node service of HotDB_02

```
[root@hotdb_210_68 conf]# service keepalived stop
Stopping keepalived: [OK]
[root@hotdb_210_68 conf]# sh ./bin/hotdb_server stop
INFO: Stopping HotDB-server ...
INFO: HotDB-server stop successed.
```

Step 2: Deploy and enable LVS

Taking single LVS service as an example, select 192.168.210.136 as LVS server, and 192.168.210.218 is used for VIP.

(1) Deploy LVS service on LVS server

```
cd /usr/local/hotdb/Install_Package

sh hotdbinstall_v*.sh --dry-run=no --install-lvs=master --lvs-vip-with-prefix=192.168.210.218/24 --lvs-port=3323
--lvs-virtual-router-id=44 --lvs-net-interface-name=eth0:1 --lvs-real-server-
list=192.168.210.134:3323:3325,192.168.210.67:3323:3325,192.168.210.68:3323:3325 --ntpdate-server-
ip=182.92.12.11
```

(2) Configure LVS service on compute node server

(HotDB 01/HotDB 02/HotDB 03)

HotDB_01/HotDB_02/HotDB_03 服务器分别执行脚本：

```
cd /usr/local/hotdb/Install_Package
```

```
sh hotdbinstall_v*.sh --dry-run=no --lvs-real-server-startup-type=service --lvs-vip-with-prefix=192.168.210.218/24 --install-ntpd=yes --ntpdate-server-host=182.92.12.11
```

(3) Enable LVS service on LVS server

```
service keepalived start
```

Step 3: Adjust parameters and enable cluster service

(1) server.xml of compute node (HotDB_01/HotDB_02/HotDB_03) is adjusted according to relevant [parameters](#), as in the following figure:

For HotDB 01 parameters, refer to the configuration of selected area:

For HotDB_02 parameters, refer to the configuration of selected area:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hotdb:server SYSTEM "server.dtd">
<hotdb:server xmlns:hotdb="http://cn.hotpu.com">
    <System>
        <property name="url">jdbc:mysql://192.168.210.67:3306/hotdb_config</property>!-- 主配置库地址, 需指定配置库服务所在的真实IP地址(ConfigDB address) -->
        <property name="username">hotdb_config</property>!-- 主配置库用户名(ConfigDB user name) -->
        <property name="password">hotdb_config</property>!-- 主配置库密码(ConfigDB password) -->
        <property name="balukername">hotdb_config</property>!-- 备机热备库用户名(Slave configDB user name) -->
        <property name="drUrl">jdbc:mysql://192.168.200.1:3306/hotdb_config</property>!-- 备机热备库地址(ConfigDB address in DR center) -->
        <property name="drPassword">hotdb_config</property>!-- 备机热备库密码(ConfigDB password in DR center) -->
        <property name="drBalUsername">hotdb_config</property>!-- 备机热备库用户名(Slave configDB user name in DR center) -->
        <property name="drBalPassword">hotdb_config</property>!-- 备机热备库密码(ConfigDB password in DR center) -->
        <property name="balu1Name"></property>!-- MGR配置库连接配置项, 必须填写此项, 指定从配置库服务所在的真实IP地址(MGR ConfigDB address(for example, ConfigDB uses MGR, this item must be configured)) -->
        <property name="balu1Port"></property>!-- MGR配置库连接配置项, 必须填写此项, 指定从配置库服务所在的真实IP地址(MGR ConfigDB password(for example, ConfigDB uses MGR, this item must be configured)) -->
        <property name="idcId"></property>!-- 机房ID, 1中机房ID, 2大机房ID(C, 1:master center, 2:DR center) -->
        <property name="nodeCorr"></property>!-- 对应机房ID的节点数, 且必须等于节点数(Cluster computer node info in the other TC) -->
        <property name="ClusterName">hotDB_02</property>!-- 集群名称(Cluster group name) -->
        <property name="ClusterSize"></property>!-- 集群中节点数(Cluster number of nodes) -->
        <property name="ClusterPort">3325</property>!-- 集群通信端口(Cluster communication port) -->
    </System>
    <Compute>
        <property name="activeComputeMode">1</property>!-- 活动计算机模式, 0:standby(热备模式), 1:master(主模式), 备机热备模式为 backup, 集群模式下, 此项无效。(the active/standby configuration of compute nodes in the high availability mode. The active compute mode is configured as master, while the standby compute mode is configured as backup (this item is invalid in the cluster mode)) -->
        <property name="correspondingComputeMode">0</property>!-- 对应计算机模式, 在高可用性模式下, 配置信息的格式为 IP:PORT, 例如, 192.168.200.1:3325 (this item is invalid in the cluster mode) -->
        <property name="serverPort"></property>!-- 管理端口(Cluster management port) -->
        <property name="processorCount"></property>!-- CPU核心数(Number of processors) -->
        <property name="timerExecutor">4</property>!-- 定时线程数(Number of timer threads) -->
        <property name="autoIncr">1</property>!-- 自增是否启用(Global autoincrement or not, 0:No, 1:global autoincrement, 2: only unique is guaranteed in global autoincrement) -->
        <property name="enableHeartbeat">true</property>!-- 心跳是否开启, 是: true, 否: false(heartbeat enabled or not: Yes: true, No: false) -->
        <property name="heartbeatPeriod"></property>!-- 心跳周期(Heartbeat period) -->
        <property name="heartbeatTimeouts">300</property>!-- 心跳超时(Heartbeat timeout) -->
    </Compute>
</hotdb:server>
```

For HotDB_03 parameters, refer to the configuration of selected area:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hotdb:server SYSTEM "server.dtd">
<hotdb:server xmlns:hotdb="http://cn.hotpu.com">
    <System>
        <property name="url">jdbc:mysql://192.168.210.67:3306/hotdb_config</property>!-- 主配置库地址, 需指定配置库服务所在的真实IP地址(ConfigDB address) -->
        <property name="username">hotdb_config</property>!-- 主配置库用户名(ConfigDB user name) -->
        <property name="password">hotdb_config</property>!-- 主配置库密码(ConfigDB password) -->
        <property name="balukername">hotdb_config</property>!-- 备机热备库用户名(Slave configDB user name) -->
        <property name="drUrl">jdbc:mysql://192.168.200.1:3306/hotdb_config</property>!-- 备机热备库地址(ConfigDB address in DR center) -->
        <property name="drPassword">hotdb_config</property>!-- 备机热备库密码(ConfigDB password in DR center) -->
        <property name="drBalUsername">hotdb_config</property>!-- 备机热备库用户名(Slave configDB user name in DR center) -->
        <property name="drBalPassword">hotdb_config</property>!-- 备机热备库密码(ConfigDB password in DR center) -->
        <property name="ConfigDB">false</property>!-- MGR是否使用(MGR, 0:否, 1:是)(MGR ConfigDB is used or not) -->
        <property name="balu1Name"></property>!-- MGR配置库连接配置项, 必须填写此项, 指定从配置库服务所在的真实IP地址(MGR ConfigDB address(for example, ConfigDB uses MGR, this item must be configured)) -->
        <property name="balu1Port"></property>!-- MGR配置库连接配置项, 必须填写此项, 指定从配置库服务所在的真实IP地址(MGR ConfigDB password(for example, ConfigDB uses MGR, this item must be configured)) -->
        <property name="idcId"></property>!-- 机房ID, 1中机房ID, 2大机房ID(C, 1:master center, 2:DR center) -->
        <property name="nodeCorr"></property>!-- 对应机房ID的节点数, 且必须等于节点数(Cluster node number 1~N(number of nodes), unique in the cluster) -->
        <property name="serverPort"></property>!-- 管理端口(Cluster management port) -->
        <property name="clusterSize"></property>!-- 集群中节点数(Cluster number of nodes) -->
        <property name="clusterPort">3325</property>!-- 集群通信端口(Cluster communication port) -->
    </System>
    <Compute>
        <property name="activeComputeMode">1</property>!-- 活动计算机模式, 0:standby(热备模式), 1:master(主模式), 备机热备模式为 backup, 集群模式下, 此项无效。(the active/standby configuration of compute nodes in the high availability mode. The active compute mode is configured as master, while the standby compute mode is configured as backup (this item is invalid in the cluster mode)) -->
        <property name="correspondingComputeMode">0</property>!-- 对应计算机模式, 在高可用性模式下, 配置信息的格式为 IP:PORT, 例如, 192.168.200.1:3325 (this item is invalid in the cluster mode) -->
        <property name="managerPort"></property>!-- 管理端口(Cluster management port) -->
        <property name="processorExecutor"></property>!-- CPU处理线程数(Thread count of each processor) -->
        <property name="dataNodeIdleCheckTime">120</property>!-- 默认空闲检测时间(秒)(Default idle inspection time of data node(s)) -->
        <property name="autoIncr">1</property>!-- 是否启用全局自增(0:否, 1:全局自增, 2:只启用全局唯一)(Global autoincrement or not, 0:No, 1:global autoincrement, 2: only unique is guaranteed in global autoincrement) -->
        <property name="enableHeartbeat">true</property>!-- 心跳是否开启, 是: true, 否: false(heartbeat enabled or not: Yes: true, No: false) -->
        <property name="heartbeatTimeouts">500</property>!-- 心跳超时(Heartbeat timeout) -->
        <property name="latencyCheckPeriod">500</property>!-- 延迟检测周期(Latency detection cycle(ms)) -->
        <property name="deadlockCheckPeriod">30000</property>!-- 死锁检测周期(Deadlock detection cycle(ms), Disabled: 0) -->
        <property name="cryptoDictionary"></property>!-- 密码字典(Crypto dictionary) -->
        <property name="DINQueryAllow"></property>!-- DIN查询是否允许(DIN query allowed or not, Yes: true, No: false) -->
    </Compute>
</hotdb:server>
```

- (2) Executes reload @@ config HotDB_01 at the management end of HotDB_01, and execute show @@cluster you can see HotDB_01 join the cluster as PRIMARY.

```
[root@hotdb_210.67 conf]# mysql -uroot -proot -h192.168.210.67 -p3325
mysql: [Warning] using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 200
Server version: 8.0.15 HotDB-2.5.6 HotDB Server by Hotpu Tech

copyright (c) 2000, 2019, oracle and/or its affiliates. All rights reserved.

oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
root@192.168.210.67:(none) 8.0.15 12:06:25> reload @@config;
Query OK, 1 row affected (0.46 sec)
Reload config success
root@192.168.210.67:(none) 8.0.15 12:07:03> show @@cluster;
+-----+-----+-----+-----+
| status | host   | port  | server_port | manager_port |
+-----+-----+-----+-----+
| PRIMARY | 192.168.210.67 | 3326 | 3323 | 3325 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- (3) Disable keepalived service of HotDB_01

service keepalived stop

```
[root@hotdb_210_67 conf]# ifconfig
eth0      Link encap:Ethernet HWaddr 52:54:00:33:8A:D9
          inet addr:192.168.210.67  Bcast:192.168.210.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fe33:8ad9/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
             RX packets:32127383 errors:0 dropped:0 overruns:0 frame:0
             TX packets:7380186 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:20480
             RX bytes:1073969479 (9.9 GiB)  TX bytes:2282289950 (2.1 GiB)

eth0:1    Link encap:Ethernet HWaddr 52:54:00:33:8A:D9
          inet addr:192.168.210.218  Bcast:0.0.0.0  Mask:255.255.255.0
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo       Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
             UP LOOPBACK RUNNING  MTU:65536  Metric:1
             RX packets:5345753 errors:0 dropped:0 overruns:0 frame:0
             TX packets:5345753 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:0
             RX bytes:461345900 (439.9 MiB)  TX bytes:461345900 (439.9 MiB)

[root@hotdb_210_67 conf]# service keepalived stop ← [ OK ]
Stopping keepalived:                                           [ OK ]
[root@hotdb_210_67 conf]# ifconfig
eth0      Link encap:Ethernet HWaddr 52:54:00:33:8A:D9
          inet addr:192.168.210.67  Bcast:192.168.210.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fe33:8ad9/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
             RX packets:32151709 errors:0 dropped:0 overruns:0 frame:0
             TX packets:7397831 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:20480
             RX bytes:10769491704 (10.0 GiB)  TX bytes:2337380941 (2.1 GiB)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
             UP LOOPBACK RUNNING  MTU:65536  Metric:1
             RX packets:5346442 errors:0 dropped:0 overruns:0 frame:0
             TX packets:5346442 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:0
             RX bytes:461456130 (440.0 MiB)  TX bytes:461456130 (440.0 MiB)
```

- (4) Start HotDB_02, HotDB_03, and then execute show @@cluster at the management end of HotDB_01; you can see that all cluster members have joined.

```
[root@localhost conf]# mysql -uzy -pzy123456 -h192.168.210.67 -P3325
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 875
Server version: 5.7.25 HotDB-2.5.6 HotDB Server by Hotpu Tech

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
zy@192.168.210.67: (none) 5.7.25 06:43:20> show @@cluster;
+-----+-----+-----+-----+
| status | host   | port | server_port | manager_port |
+-----+-----+-----+-----+
| PRIMARY | 192.168.210.67 | 3326 | 3323 | 3325 |
| SECONDARY | 192.168.210.68 | 3326 | 3323 | 3325 |
| SECONDARY | 192.168.210.134 | 3326 | 3323 | 3325 |
+-----+-----+-----+-----+
3 rows in set (0.02 sec)

zy@192.168.210.67: (none) 5.7.25 06:43:23> █
```

Step 4: Adaptation adjustment of management platform

The adaptation mode is the same as "[Expand compute nodes in cluster mode](#)". Edit the cluster of compute nodes to add new compute nodes and convert the HA mode to cluster mode, as shown in the following figure:

Edit Compute Node Cluster

Cluster Management > Compute Node Cluster > Edit Compute Node Cluster

Cluster Info

Compute Node Mode: **Single Node** (Master/Slave Node) **Multi-Node**

* Group Name: HA-DB

Set ConfigDB manually

enable DR mode

ConfigDB Address: sdb:mying://192.168.240.67:3306/hosts_config

ConfigDB Username: hosts.config

ConfigDB Password:
.....

Compute Node

Type	* Compute Node Name	* Hostname	* Username	* Password	* Service Port	* Management Port	SSH Login Mode	SSH Port	SSH Username	SSH User Password	server.xml Address	keepalived.conf Address	Operation
Master	HostDB_01	192.168.240.67	root	3323	3325	Passwd	22	root	/usr/local/hosts/host	/etc/keepalived	
Slave	Enter Node Name	192.168.240.68	root	3323	3325	Passwd	22	root	/usr/local/hosts/host	/etc/keepalived	

Test **Save** **Cancel**

HotDB: Distributed Transactional Database Platform

Currently in HotDB-ZB6

admin

The dashboard displays the following key metrics and components:

- Connections:** QPS: 2719, OPS: 5.0, Latency: 0.2ms.
- Nodes:** AUTOTEST, BENCHMARKSQL, OLTP, PM, TPCC.
- Metrics:** Total Back-End Connections: 4441, Direct Memory Usage: 4.3%, Total Back-End Connections: 43, Heap Memory Usage: 48.27%, Back-End Inflow: 152.63GB/s, Back-End Outflow: 536.09GB/s, Back-End Inflow Rate: 27.65MB/s ↑, Back-End Outflow Rate: 273.45MB/s ↑, Back-End Inflow Rate: 27.65MB/s ↑, Back-End Outflow Rate: 273.45MB/s ↑, Back-End Inflow Rate: 24.07MB/s ↑, Back-End Outflow Rate: 235.22MB/s ↑, Back-End Inflow Rate: 24.07MB/s ↑, Back-End Outflow Rate: 235.22MB/s ↑, Back-End Inflow Rate: 27.04MB/s ↑, Back-End Outflow Rate: 250.95MB/s ↑, Back-End Inflow Rate: 27.04MB/s ↑, Back-End Outflow Rate: 250.95MB/s ↑.
- Network:** A detailed network diagram at the bottom shows connections between nodes (dn.1 to dn.9) and their corresponding connection details (QPS, OPS, Latency).

8.1.5.1.3. Expand compute nodes in cluster mode

This section mainly describes the operation of expanding compute nodes in cluster mode. The information of involved components is as follows:

Role	Connection information	Name
Master compute node	192.168.210.157_3323_3325	HotDB_01
Slave compute node	192.168.210.156_3323_3325	HotDB_02
Slave compute node	192.168.210.155_3323_3325	HotDB_03

Master/slave LVS service	192.168.210.135/137	VIP:192.168.210.216
New compute node	192.168.210.134	HotDB_04

Step 1: add a new compute node to LVS server

- (1) Add virtual service of HotDB_04 to the master/slave LVS server.

```
ipvsadm -a -t 192.168.210.216:3323 -r 192.168.210.134
```

- (2) Add the service info of HotDB_04 to keepalived.conf of the master/slave LVS, as shown in the following figure:

```
! Configuration File for keepalived
global_defs {
    router_id hotdb_lvs_backup
    vrrp_garp_master_refresh 20
    vrrp_garp_master_refresh_repeat 2
}

vrrp_instance hotdb_lvs {
    state BACKUP #指定keepalived的角色, MASTER为主, BACKUP为备
    interface eth0
    virtual_router_id 216 #虚拟路由编号, 主备要一致, 同一个网段\广播域内要唯一
    priority 100 #优先级
    advert_int 1 #检查间隔, 默认为1s
    authentication {
        auth_type PASS
        auth_pass 1221
    }
    virtual_ipaddress {
        192.168.210.216/24 dev eth0 label eth0:2
    }
}

virtual_server 192.168.210.216 3323 {
    delay_loop 6 # 设置健康检查的算时, 单位是秒, 会影响节点权重增加间隔时间
    lb_algo wrr # 设置负载调度的算法
    lb_kind DR # 设置LVS实现负载的机制
    protocol TCP #指定转发协议类型
    # persistence_timeout 50 # 一个源地址可连接到同一个目标地址的持续时间, 单位是秒

    real_server 192.168.210.134 3323 {
        weight 100
        inhibit_on_failure
        MISC_CHECK {
            misc_path "/usr/local/hotdb/hotdb-server/bin/hotdb_lvs_check.sh root root 192.168.210.134 3325 3323"
            misc_timeout 5
            misc_dynamic
        }
    }

    real_server 192.168.210.155 3323 {
        weight 100
        inhibit_on_failure
        MISC_CHECK {
            misc_path "/usr/local/hotdb/hotdb-server/bin/hotdb_lvs_check.sh root root 192.168.210.155 3325 3323"
            misc_timeout 5
            misc_dynamic
        }
    }

    real_server 192.168.210.156 3323 {
        weight 100
        inhibit_on_failure
        MISC_CHECK {
            misc_path "/usr/local/hotdb/hotdb-server/bin/hotdb_lvs_check.sh root root 192.168.210.156 3325 3323"
            misc_timeout 5
            misc_dynamic
        }
    }

    real_server 192.168.210.157 3323 {
        weight 100
        inhibit_on_failure
        MISC_CHECK {
            misc_path "/usr/local/hotdb/hotdb-server/bin/hotdb_lvs_check.sh root root 192.168.210.157 3325 3323"
        }
    }
}
```

Step 2: configure LVS for the new compute node server

Execute deployment script and configure LVS at the HotDB_04 server:

```
cd /usr/local/hotdb/Install_Package/
```

```
sh hotdbinstall_v*.sh --dry-run=no --lvs-real-server-startup-type=service --lvs-vip-with-prefix=192.168.210.216/24 --install-ntpd=no --ntpdate-server-host=182.92.12.11
```

说明：--lvs-vip-with-prefix：当前集群的 VIP

Step 3: adjust parameters and start a new cluster member

(1) Modify the value of ClusterSize in server.xml of all compute node servers (HotDB_01/HotDB_02/HotDB_03/HotDB_04) to ensure that the value of ClusterSize is equal to the actual number of cluster members (here is 4). Other parameters require no adjustment, but it should be noted that the values of clusterName, clusterSize, clusterNetwork, clusterPort are consistent in the same cluster.

```

<!-- XML version 1.0 encoding: UTF-8 -->
<!DOCTYPE htdb SYSTEM "server.dtd">
<htdb>
  <!-- 系统参数 -->
  <system>
    <!-- 配置连接地址，需指定配置库服务所在的真实IP地址(ConfigDB address) -->
    <property name="userDbConfig"></property>--> 主配置库用户名(ConfigDB user name)-->
    <property name="password"></property>--> 主配置库密码(ConfigDB password)-->
    <property name="userSlaveConfig"></property>--> 从配置库用户名(Slave configDB user name)-->
    <property name="passwordSlaveConfig"></property>--> 从配置库密码( Slave configDB password)-->
    <property name="url">jdbc:mysql://192.168.200.1:3306/htdb</property>--> 配置机配置库地址(ConfigDB address in DR center)-->
    <property name="drakarUser"></property>--> 客户端连接配置库用户名(ConfigDB user in DR center)-->
    <property name="drakarPassWord"></property>--> 客户端连接配置库密码(ConfigDB password in DR center)-->
    <property name="drakarUrl">jdbc:mysql://192.168.200.1:3306/htdb</property>--> 客户端连接配置库地址(ConfigDB address in DR center)-->
    <property name="drakarPort"></property>--> 客户端连接配置库端口(ConfigDB port in DR center)-->
    <property name="drakarProtocol"></property>--> 客户端连接配置库协议(ConfigDB protocol in DR center)-->
    <property name="balDriver"></property>--> MGR驱动器(MGR driver)必须配置，指配置库连接的真是IP地址(MGR ConfigDB address(for example, ConfigDB uses MGR, this item must be configured))-->
    <property name="balPassWord"></property>--> MGR连接密码(MGR password)必须配置，指配置库连接的真是IP地址(MGR ConfigDB password(for example, ConfigDB uses MGR, this item must be configured))-->
    <property name="balDriver"></property>--> MGR连接驱动器(MGR driver)必须配置，指配置库连接的真是IP地址(MGR ConfigDB driver(for example, ConfigDB uses MGR, this item must be configured))-->
    <property name="balPort"></property>--> MGR连接端口(MGR port)必须配置，指配置库连接的真是IP地址(MGR ConfigDB port)-->
    <property name="balProtocol"></property>--> MGR连接协议(MGR protocol)必须配置，指配置库连接的真是IP地址(MGR ConfigDB protocol)-->
    <property name="idcNodes" value="192.168.200.1:3325,192.168.200.1:3325" />--> 一个机房部署的节点数(Computer node info in the other IDC)-->
    <property name="clusterNodes" value="192.168.200.1:3325" />--> 一个集群部署的节点数(Cluster node number 1~(number of nodes), unique in the cluster)-->
    <property name="clusterName" value="cluster" />--> 集群名(Cluster group name)-->
    <property name="clusterNetwork" value="192.168.0.0/24" />--> 集群所在网段(Cluster network segment)-->
    <property name="clusterPort" value="3326" />--> 集群通信端口(Cluster communication port)-->
    <property name="master_standby" value="true" />--> 计算节点采用主备配置，手工点对点配置为 master，会计算节点配置为 standby(集群模式下, 此项无效)(The active/standby configuration of compute nodes in the high availability mode is invalid if this item is set to true)-->
    <property name="handover" value="true" />--> 手动切换是否可用(Handover is available or not) (The connection information of the current active compute node corresponding to the handover item is invalid in the cluster mode)-->
    <property name="serverPort" value="2323" />--> 异步端口(Asservice port)-->
    <property name="processors" value="16" />--> 处理器数(Number of processors)-->
    <property name="processorExecutor" value="4" />--> 处理器线程数(Thread count of each processor)-->
    <property name="deadlockCheckPeriod" value="1000" />--> 死锁检测周期(Detect period of deadlock)-->
    <property name="deadnodeIdCheckPeriod" value="120" />--> 死节点检测默认时间(秒)(Default idle inspection time of data node(s))-->
    <property name="autoIncrement" value="0" />--> 自增长是否启用(Global autoincrement or not, 0:No, 1:Global autoincrement, 2: Only unique is guaranteed in global auto increment)-->
    <property name="heartbeatEnabled" value="true" />--> 心跳是否启用(Heartbeat enable or not)-->
    <property name="heartbeatPeriod" value="1000" />--> 心跳周期(Heartbeat period)-->
    <property name="deadlockDetected" value="true" />--> 死锁检测是否已检测到(Deadlock detection detected or not)-->
    <property name="latencyCheckPeriod" value="500" />--> 延时检测周期(Latency detection period)-->
    <property name="deadlockCheckCycles" value="1000" />--> 死锁检测周期(Deadlock detection cycle(m), disabled: 0)-->
    <property name="deadlockCheckTime" value="1000" />--> 死锁检测时间(Deadlock detection time(m), disabled: 0)-->
    <property name="joinable" value="true" />--> 允许JOIN查询(Allow JOIN query, true, false) (join query allowed or not, Yes: true, No: false)-->
    <property name="joinBatchSize" value="1000" />--> JOIN查询批处理大小(Numbers of queries transferred into IN queries from each batch in equivalent JOIN query)-->
    <property name="maxJoinSize" value="1000000" />--> JOIN查询最大行数(100万行, K: thousand) (Limitation of intermediate result set row(s) in JOIN)-->
    <property name="maxConnections" value="5000" />--> 最大连接数(Fixed maximum connections)-->
    <property name="maxConnectionsUnlimited" value="false" />--> 最大连接数是否无限(Fixed maximum connections unlimited or not)-->
    <property name="enableX" value="false" />--> 是否开启X事务(true, false)(transaction enabled or not, Yes: true, No: false)-->
    <property name="strategy" value="readWrite" />--> 读写分离策略(Read/Write splitting strategy)-->
    <!-- 分离读写请求 -->
    <!-- 可以通过设置为 0 分离所有表的读写操作，但部分表的读写操作可能无法使用读分离。2) 事务中发生故障的读请求会用到备数据源；3) Read/write splitting is not enabled: 0, Separate Read Request is sent to all available datasources; 1, Separate Read request is available to standby datasources; 2, Separable Read In Transaction request is sent to available standby datasources; 3) -->
    <property name="weightForSlaveRead" value="50" />--> 为代理表分配权重(Slave Read weight, 0~100, 百分比) (Weight for slave read allocation, 0~100 percentage, Disable: 0)-->
    <property name="idleTimeout" value="100000" />--> 读超时时间(Read timeout time, ms) (Read timeout time in ms)-->
    <property name="waitForSlaveRecovery" value="true" />--> 等待从机恢复(等待从机恢复, true, false)(Waiting for slave recovery)-->
    <property name="waitForSlaveRecover" value="true" />--> 等待从机恢复中(等待从机恢复中, true, false)(Waiting for slave catching up with copy or not in the recover)-->
  </system>

```

(2) Adjust other cluster parameters in server.xml and start the service in the new compute node server (HotDB 04), as shown in the following figure:

Step 4: Reload the configuration

Execute reload @@config at the management end of the primary compute node(HotDB_01), you can see HotDB_04 join the cluster:

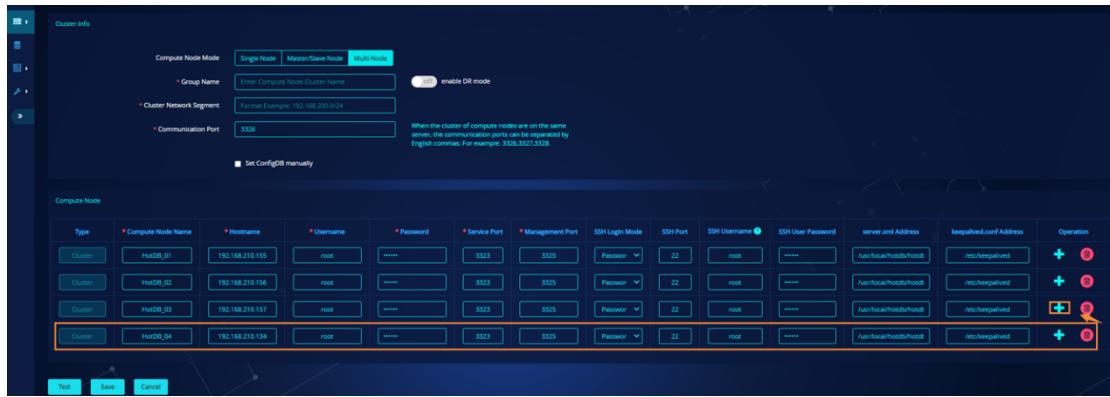
```
zy@192.168.210.157:(none) 5.7.25 02:27:00> reload @@config;
Query OK, 1 row affected (0.96 sec)
Reload config success

zy@192.168.210.157:(none) 5.7.25 02:27:06> show @@cluster;
+-----+-----+-----+-----+-----+
| status | host   | port | server_port | manager_port |
+-----+-----+-----+-----+-----+
| PRIMARY | 192.168.210.157 | 3326 | 3323 | 3325 |
| SECONDARY | 192.168.210.155 | 3326 | 3323 | 3325 |
| SECONDARY | 192.168.210.156 | 3326 | 3323 | 3325 |
| SECONDARY | 192.168.210.134 | 3326 | 3323 | 3325 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Step5: Adaption adjustment of management platform

Enter the “Cluster management” -> “Compute node cluster” page to bring the newly added compute nodes into the management.

Edit the compute node cluster, and add the newly introduced compute nodes via the “+” button in the operation bar on the right side of the compute node. After saving, the management platform will automatically identify the compute node mode according to the number of compute nodes, as shown in the following figure:



Notes:

- If new compute nodes are added in the cluster, repeat the operation from the first step.
- If the value of clusterSize and haMode of the compute node does not match the actual configured cluster, reload @@config in step 4 will fail, therefore it shall be ensured that the configuration be consistent with the actual situation.
- The parameter serverId of new compute nodes should be unique and continuous with the original cluster, otherwise it will cause abnormal startup.

8.1.5.2. Reduction of compute nodes

The number of compute nodes can be reduced by compute node reduction function.

After disabling the compute node, you can directly modify other multi-node-related configuration in the cluster. After reloading, the number of compute nodes can be reduced according to the new configuration.

8.1.5.2.1. Parameter introduction

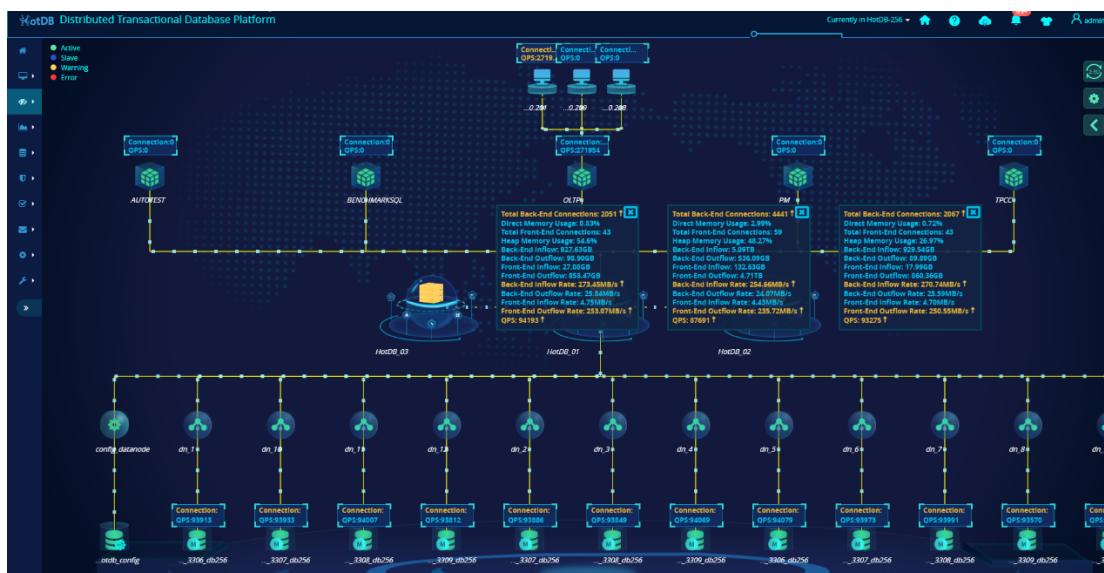
Parameters involved are as follows:

Parameters	Instructions	Reference value	Whether the reloading is valid
------------	--------------	-----------------	--------------------------------

haMode	High availability: 0: master/slave; 1: cluster	In cluster environment, the parameter value is 1.	Yes
HaState	Master / slave role configuration in HA mode.	Configuration of primary compute node: master, configuration of standby compute node: backup.	Yes
haNodeHost	The connection information of the current active compute node in the high availability mode.	The configuration format is IP: PORT 192.168.200.1:3325	Yes

8.1.5.2.2. Reduce from cluster mode to HA mode

This section mainly describes the operation of reducing a cluster that normally provides services to HA mode. The components involved are consistent with the [Expand from HA mode to multi-node mode](#).



Step 1: disable the standby compute node service

Disable the compute node services of HotDB_02 and HotDB_03. This process will

trigger cluster election. If there is a pressure measurement task at this time, it will flash off and return to normal in a few seconds.

```
[ 90s ] thds: 4 tps: 341.99 qps: 5469.27 (r/w/o: 4785.30/0.00/683.97) lat (ms,99%): 21.50 err/s: 0.00 reconn/s: 0.00
[ 92s ] thds: 4 tps: 325.52 qps: 5206.78 (r/w/o: 4555.74/0.00/651.03) lat (ms,99%): 23.10 err/s: 0.00 reconn/s: 0.00
[ 94s ] thds: 4 tps: 322.51 qps: 5170.71 (r/w/o: 4525.69/0.00/645.03) lat (ms,99%): 20.74 err/s: 0.00 reconn/s: 0.00
[ 96s ] thds: 4 tps: 49.94 qps: 3106.70 (r/w/o: 702.14/0.00/2404.56) lat (ms,99%): 19.29 err/s: 2304.18 reconn/s: 1.00
[ 98s ] thds: 4 tps: 0.00 qps: 3454.70 (r/w/o: 0.00/0.00/3454.70) lat (ms,99%): 0.00 err/s: 3454.70 reconn/s: 0.00
[ 100s ] thds: 4 tps: 0.00 qps: 3309.18 (r/w/o: 0.00/0.00/3309.18) lat (ms,99%): 0.00 err/s: 3308.68 reconn/s: 0.00
[ 102s ] thds: 4 tps: 37.04 qps: 628.17 (r/w/o: 553.09/0.00/75.08) lat (ms,99%): 7346.49 err/s: 0.50 reconn/s: 1.00
[ 104s ] thds: 4 tps: 262.48 qps: 4200.25 (r/w/o: 3675.28/0.00/524.97) lat (ms,99%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 106s ] thds: 4 tps: 301.02 qps: 4816.76 (r/w/o: 4214.72/0.00/602.03) lat (ms,99%): 18.61 err/s: 0.00 reconn/s: 0.00
[ 108s ] thds: 4 tps: 314.63 qps: 5040.03 (r/w/o: 4410.77/0.00/629.25) lat (ms,99%): 19.65 err/s: 0.00 reconn/s: 0.00
```

Step 2: deploy keepalived and adjust the compute node configuration

(1) HotDB_01、HotDB_02 servers are respectively deployed with master and slave keepalived (the corresponding VIP can be the same with and VIP of LVS, but virtual_router_id cannot be the same)

HotDB_01 server execution script:

```
/usr/local/hotdb/Install_Package

sh hotdbinstall_v*.sh --dry-run=no --install-keepalived=master --keepalived-vip-with-prefix=192.168.210.218/24 --keepalived-virtual-router-id=218 --keepalived-net-interface-name=eth0:1 --ntpdate-server-host=182.92.12.11 --install-ntpd=yes
```

HotDB_02 server execution script:

```
cd /usr/local/hotdb/Install_Package

sh hotdbinstall_v*.sh --dry-run=no --install-keepalived=backup --keepalived-vip-with-prefix=192.168.210.218/24 --keepalived-virtual-router-id=218 --keepalived-net-interface-name=eth0:1 --ntpdate-server-host=182.92.12.11 --install-ntpd=yes
```

(2) Modify the configuration of keepalived.conf of HotDB_01、HotDB_02 servers, refer to the chapter 2.1.2.2 in the document Installation and Deployment

(3) Modify server.xml of HotDB_01、HotDB_02 compute nodes. Configure relevant parameters as HA mode, as in the following figure:

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hotdb:server SYSTEM "Server.dtd">
<hotdb:server>
<!--System-->
<property name="user">idc:mysql://192.168.210.67:3306/hotdb_config</property><!-- 主配置库地址，需指定配置库服务所在的真IP地址(Config0B address) -->
<property name="username">hotdb_config</property><!-- 主配置库用户名(Config0B user name) -->
<property name="password">hotdb_config</property><!-- 主配置库密码(Config0B password) -->
<property name="bakUsername">hotdb_config</property><!-- 备份配置库用户名(Config0B user name) -->
<property name="drurl">jdbc:mysql://192.168.200.1:3306/hotdb_config</property><!-- 备份配置库地址(Config0B address in DR center) -->
<property name="drusername">hotdb_config</property><!-- 备份配置库用户名(Config0B user name in DR center) -->
<property name="drpassword">hotdb_config</property><!-- 备份配置库密码(Config0B password in DR center) -->
<property name="drBakUser">hotdb_config</property><!-- 备份配置库用户名(Config0B user name in DR center) -->
<property name="drBakPasswd">hotdb_config</property><!-- 备份配置库密码(Config0B password in DR center) -->
<!--Compute-->
<property name="haMode">false</property><!-- 高可用模式。0:HA, 1:集群, 2:HA模式中心机房, 3:HA模式次备机房[High-availability mode, 0:HA, 1:Cluster, 2:HA in master center, 3:HA in DR center] -->
<property name="bakUser"></property><!-- MGR配置库用户名(如配置库使用MGR, 必须配置此项)(MGR Config0B user name(for example, Config0B uses MGR, this item must be configured)) -->
<property name="bakPasswd"></property><!-- MGR配置库密码(如配置库使用MGR, 必须配置此项)(MGR Config0B password(for example, Config0B uses MGR, this item must be configured)) -->
<property name="hotdbPort">3325</property><!-- 热备端口(Service port) -->
<property name="hotdbPort">3325</property><!-- 热备端口(Service port) -->
<property name="processors">1</property><!-- 处理器数(Number of processors) -->
<property name="tmerExecutor">4</property><!-- 定时器线程数(Number of timer thread) -->
<property name="dataModelIdleCheckPeriod">120</property><!-- 数据模型空闲时检查时间(秒) (Default idle check time of data model(s)) -->
<property name="autoIncrement">1</property><!-- 是否采用全局自增为1。0:不开启全局自增, 1:开启全局自增, 2:开启全局自增并保证(Globally autoincrement or not, 0:No, 1:Global autoincrement, 2: Only unique is guaranteed in global auto increment) -->
```

```
<!--Compute-->
<property name="bakUser"></property><!-- MGR配置库用户名(如配置库使用MGR, 必须配置此项)(MGR Config0B user name(for example, Config0B uses MGR, this item must be configured)) -->
<property name="bakPasswd"></property><!-- MGR配置库密码(如配置库使用MGR, 必须配置此项)(MGR Config0B password(for example, Config0B uses MGR, this item must be configured)) -->
<property name="hotdbPort">3325</property><!-- 热备端口(Service port) -->
<property name="hotdbPort">3325</property><!-- 热备端口(Service port) -->
<property name="processors">1</property><!-- 处理器数(Number of processors) -->
<property name="tmerExecutor">4</property><!-- 定时器线程数(Number of timer thread) -->
<property name="dataModelIdleCheckPeriod">120</property><!-- 数据模型空闲时检查时间(秒) (Default idle check time of data model(s)) -->
<property name="autoIncrement">1</property><!-- 是否采用全局自增为1。0:不开启全局自增, 1:开启全局自增, 2:开启全局自增并保证(Globally autoincrement or not, 0:No, 1:Global autoincrement, 2: Only unique is guaranteed in global auto increment) -->
```

(4) Enable keepalived of HotDB_01, until the VIP of keepalived is mounted.

```
[root@hotdb_210_67 lib]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet 192.168.210.218/32 brd 192.168.210.218 scope global lo:3
            inet6 ::1/128 scope host
                valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 20480
    link/ether 52:54:00:33:8a:d9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.210.67/24 brd 192.168.210.255 scope global eth0
        inet6 fe80::5054:ff:fe33:8ad9/64 scope link
            valid_lft forever preferred_lft forever
[root@hotdb_210_67 lib]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet 192.168.210.218/32 brd 192.168.210.218 scope global lo:3
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 20480
    link/ether 52:54:00:33:8a:d9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.210.67/24 brd 192.168.210.255 scope global eth0:1
        inet6 fe80::5054:ff:fe33:8ad9/64 scope link
            valid_lft forever preferred_lft forever
```

(5) Execute reload @@ config at managemtn end of HotDB_01 to make the remaining compute nodes become the master compute nodes in HA mode.

```
[root@localhost Logs]# mysql -uzy -pzy123456 -h192.168.210.67 -P3325
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38009
Server version: 5.7.25 HotDB-2.5.6 HotDB Server by Hotpu Tech

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

zy@192.168.210.67:(none) 5.7.25 11:00:17> reload @@config;
Query OK, 1 row affected (0.77 sec)
Reload config success
```

At this time, if there is a pressure measurement task, it will flash off and return to normal after a few seconds.

```
[ 808s ] thds: 4 tps: 385.52 qps: 6169.88 (r/w/o: 5398.84/0.00/771.05) lat (ms,99%): 20.00 err/s: 0.00 reconn/s: 0.00
[ 810s ] thds: 4 tps: 230.69 qps: 3688.59 (r/w/o: 3227.20/0.00/461.39) lat (ms,99%): 23.52 err/s: 0.00 reconn/s: 0.00
[ 812s ] thds: 4 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,99%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 814s ] thds: 4 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,99%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 816s ] thds: 4 tps: 84.57 qps: 1342.60 (r/w/o: 1173.46/0.00/169.14) lat (ms,99%): 5607.61 err/s: 0.00 reconn/s: 0.00
[ 818s ] thds: 4 tps: 151.50 qps: 2420.54 (r/w/o: 2118.03/0.00/302.50) lat (ms,99%): 57.87 err/s: 0.00 reconn/s: 0.00
[ 820s ] thds: 4 tps: 275.45 qps: 4411.64 (r/w/o: 3860.25/0.00/551.39) lat (ms,99%): 43.39 err/s: 0.00 reconn/s: 0.00
```

Step 3: Disable LVS service on LVS server

Disables LVS service on LVS server

```
systemctl stop keepalived.service
```

Step 4: Clear LVS configuration of compute node server

(1) HotDB_01、HotDB_02、HotDB_03 stop lvsrs

```
/etc/init.d/lvsrs stop
```

(2) HotDB_01、HotDB_02、HotDB_03 delete lvsrs

```
cd /etc/init.d
```

```
rm -rf lvsrs
```

Step 5: Enable slave compute node and slave keepalived service.

Enable HotDB_02 and keepalived service.

```
[root@hotdb_210_70 hotdb-server]# sh bin/hotdb_server start
INFO: Start HotDB-server ...
INFO: HotDB-Server start successed.
[root@hotdb_210_70 hotdb-server]# service keepalived start
Starting keepalived: [ OK ]
```

Step 6: Adaptation adjustment of management platform

The adaptation mode is the same as "[Expand compute nodes in cluster mode](#)". Edit the cluster of compute nodes to delete the reduced compute nodes and convert the cluster mode to HA mode, as shown in the following figure:

Cluster Info

Compute Node Mode: Single Node, Master/Slave Node, Multi-Node
 Group Name: HA_DB
 enable DR mode

* Cluster Network Segment: Format Example: 192.168.200.0/24
 * Communication Port: 3326
When the cluster of compute nodes are on the same segment, the communication ports can be separated by commas. For example: 3326,3327,3328

Set ConfigDB manually
 Single Database, Master-Master, MGR
 ConfigDB Address: jdbc:mysql://192.168.210.67:3306/hotdb_config
 ConfigDB Username: hotdb_config
 Com:

Compute Node

Type	Compute Node Name	Hostname	Username	Password	Service Port	Management Port	SSH Login Mode	SSH Port	SSH Username	SSH User Password	server.xml Address	keepalive.conf Address	Operation
Master	HotDB_01	192.168.210.67	root	3323	3325	Password	22	root	Autofocus/hosts/ha	/etc/keepalive	<input type="button" value="+"/> <input type="radio"/>
Slave	HotDB_02	192.168.210.68	root	3323	3325	Password	22	root	Autofocus/hosts/ha	/etc/keepalive	<input type="button" value="+"/> <input type="radio"/>
Slave	HotDB_03	192.168.210.13	root	3323	3325	Password	22	root	Autofocus/hosts/ha	/etc/keepalive	<input type="button" value="+"/> <input checked="" type="radio"/>

Edit Compute Node Cluster

Cluster Management > Compute Node Cluster > Edit Compute Node Cluster

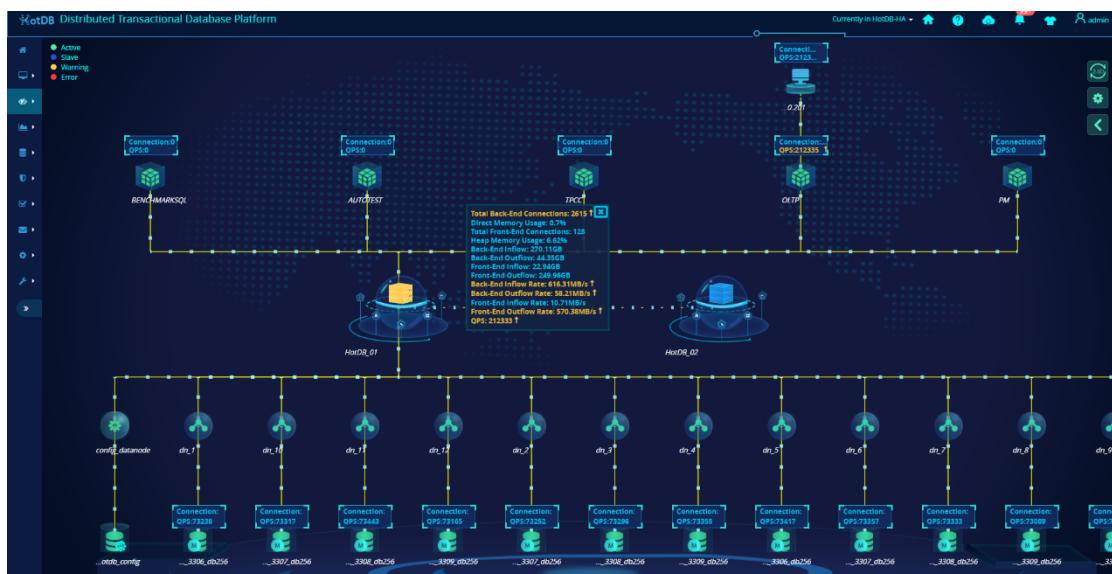
Cluster Info

Compute Node Mode: Single Node, Master/Slave Node, Multi-Node
 Group Name: HA_DB
 enable DR mode

Set ConfigDB manually
 Single Database, Master-Master, MGR
 ConfigDB Address: jdbc:mysql://192.168.210.67:3306/hotdb_config
 ConfigDB Username: hotdb_config
 Com:

Compute Node

Type	Compute Node Name	Hostname	Username	Password	Service Port	Management Port	SSH Login Mode	SSH Port	SSH Username	SSH User Password	server.xml Address	keepalive.conf Address	Operation
Master	HotDB_01	192.168.210.67	root	3323	3325	Password	22	root	Autofocus/hosts/ha	/etc/keepalive	<input type="button" value="+"/> <input type="radio"/>
Slave	HotDB_02	192.168.210.68	root	3323	3325	Password	22	root	Autofocus/hosts/ha	/etc/keepalive	<input type="button" value="+"/> <input type="radio"/>



Notes:

If you want to perform online reduction of cluster compute nodes, you shall disable the

compute nodes, and then reload before modifying the total number of cluster members and member serverId.

8.2. Read/write splitting

HotDB Server supports Master/Slave Read/write splitting function of MySQL data source in data node

8.2.1. Description of Read/write splitting function

To use the Read/write splitting function, it needs to configure Master/Slave data source in the data node.

By default, Read/write splitting function is Disable in the compute node. To enable the Read/write splitting function, it needs to set property value of strategyForRWSplit bigger than 0 in Config Fileserver.xml of the compute node.

```
<property name="strategyForRWSplit">1</property>
```

strategyForRWSplit is allowed to be set as 0, 1, 2, 3. If set as 0, the Read/Write operations are conducted in the Master data source.

- When set as 1, the Master data source participates in the Read/Write, and the Slave participates in Read.
- When set as 2, the Active Master in non-transaction only participates in Write, and the Slave only participates in Read.
- When set as 3, then before write request, the Active Master in transaction only participates in Write, and the Slave only participates in Read, after Write occurs in transaction, all Read and Write in the transaction are sent to the Active Master.

Note: when HINT is not used for read-write splitting, the "separable read request" mainly refers to the auto committed read request and the read request in the explicit

read-only transaction. The rest of the read requests are "inseparable read requests". For example, a read request in a non-auto-commit transaction.

In server.xml, it could configure max latency time, parameter name: maxLatencyForRWSplit, unit: ms of readable Standby Slave in Read/write splitting, when the actual master/slave latency time is longer than the latency time set, Read/write splitting will force to read data from the Active Master, and if shorter than the latency time set, it will read data according to the configured Read/write splitting strategy. The configured latency time is 1s by default, that is: when the data source data synchronization in the node has latency for more than 1s, then under the condition of enabling the Read/write splitting function, the compute node will not allow the Slave data source with data synchronization latency to participate in Read; if in data node, all Slave data sources have data latency, then at this time, the Read operation will be taken over by the Master data source, and only when there is no latency in data source data synchronization could the Slave data source participate in Read.

When Read/write splitting starts, even if the Heartbeat is not started, it will also force to make latency check. The Latency Check Period could be configured via the parameter latencyCheckPeriod in server.xml.

In the compute node, through [HINT](#) function, it could specify the data source type executed by SQL statement.

Specify that SQL statement shall execute the following in the Master data source:

```
/*!hotdb:rw=master*/select * from customer;
```

Specify that SQL statement shall execute the following in the Standby Slave data source:

```
/*!hotdb:rw=slave*/select * from customer;
```

8.2.2. Weight configuration of Read/write splitting

While supporting Read/write splitting, compute node could control the master/slave Read proportion via the configuration parameter in server.xml. Enter conf directory under installation directory of compute node, and edit server.xml, to modify the following related settings:

```
<property name="strategyForRWSplit">1</property><!-- Read/write splitting strategy, only Active Master participates in Read/Write: 0; Active Master participates in Read/Write, the Slave participates in Read; 1:Active Master only participates in Write, the Slave participates in Read: 2: Splittable Read request is distributed to available Slave data sources; 3: Read request in transaction before Write is distributed to available Slave data sources; -->

<property name="weightForSlaveRWSplit">100</property><!-- the Slave Read proportion, 50 (percent) by default, and 0 means that the modified parameter is invalid -->
```

Description:

- When the Read/write splitting strategy strategyForRWSplit is configured as 0, it only makes Read/Write on the Active Master, and the setting of weightForSlaveRWSplit Slave Read Proportion is meaningless;
- When Read/write splitting strategy strategyForRWSplit is configured as 1, Active Master participates in Read/Write, the Slave participates in Read, and only in this way could weightForSlaveRWSplit have practical significance. Read proportion of Active Master is 100% —percentage value of weightForSlaveRWSplit, and the remaining Slaves shares equal distribution of all proportions of weightForSlaveRWSplit setting, for example: if the weightForSlaveRWSplit value is set as 60%, and the existing node is of one-master and two-slave architecture, then the Master Read 40%, while the two Slave Read 30% respectively;
- When Read/write splitting strategy strategyForRWSplit is configured as 2, Active Master only participates in Write, and the Slave participates in Read; all Slave share

equal distribution of Read proportion, and for one-master and two-slave architecture, the Slave Read 50% each;

- When Read/write splitting strategy strategyForRWSplit is configured as 3, for Read in transaction before Write, the Slave participates in Read, and the Active Master participates in Write; the Read operations in transaction after Write will be taken over by the Master data source; while the operations beyond the transaction are consistent with that in mode 2.

9. Characteristic functions of HotDB Server

HotDB Server provides some expanded functions based on Distributed Transactional Database design, which are easy to use and manage.

9.1. DNID

DNID is abbreviation of DATANODE_ID

On compute node, DNID could be used as filter condition in WHERE Clause, and as a Query item in SELECT statement; DNID (data node) of the results of each row could also be displayed in Result Set.

➤ Use DNID field in SELECT, UPDATE, DELETE Clauses

```
SELECT * FROM customer WHERE dnid=1;
```

If executing this SELECT statement, compute node will return the data of the Sharding Table customer being 1 on DNID.

```
DELETE FROM customer WHERE dnid=1 AND id=3;
```

If executing this DELETE statement, compute node will delete data of Sharding Table customer with DNID being 1 and Field ID being 3.

```
UPDATE customer SET id=4 WHERE dnid=1 AND name='a';
```

If executing this UPDATE statement, compute node will modify the data of Sharding Table customer with DNID being 1 and Field Name = 'a'.

➤ Use DNID as a Query item to execute SELECT statement

```
SELECT *,dnid FROM tab_name;
```

If executing this SELECT statement, compute node will display dnid value of all results in the Result Set, and dnid must be placed behind *, otherwise, there will be Syntax Error.

➤ Show DNID in Result Set

After log in the compute node, if executing the SET SHOW_DNID=1 statement, compute node will return DNID (data node ID) of results of each row in the SELECT statement.

```
mysql> set show_dnid=1;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from customer where id in (77,67,52,20);

+----+-----+-----+-----+-----+-----+
| id | name      | telephone    | provinceid | province    | city | address          | DNID |
+----+-----+-----+-----+-----+-----+
| 52 | 马深圳    | 13912340052 |           7 | Guangdong | 深圳   | 某某街某某号 |     13 |
| 77 | 郝上海    | 13912340077 |           25 | Shanghai  | 上海   | 某某街某某号 |     14 |
| 20 | 许重庆    | 13912340020 |           4  | Chongqing | 重庆   | 某某街某某号 |     12 |
| 67 | 岑南昌    | 13912340067 |           17 | Jiangxi   | 南昌   | 某某街某某号 |     15 |
+----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

The results in above figure show data rows with data node ID as 12, 13, 14, 15 respectively.

When executing SET SHOW_DNID=1 statement, and Query Global Table, compute node will return DNID (GLOBAL) of results of each row in the SELECT statement.

```
mysql> set show_dnid=1;  
  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from tb_quan;  
  
+----+---+---+-----+  
| id | a | b | DNID |  
+----+---+---+-----+  
| 1 | 1 | 1.10 | GLOBAL |  
| 2 | 2 | 1.20 | GLOBAL |  
| 3 | 3 | 1.30 | GLOBAL |  
+----+---+---+-----+
```

SET SHOW_DNID=0, will cancel showing DNID column in Result Reset.

```
mysql> set show_dnid=0;  
  
Query OK, 0 rows affected (0.00 sec)  
  
  
mysql> select * from customer where id in (77,67,52,20);  
  
+----+-----+-----+-----+-----+  
| id | name | telephone | provinceid | province | city | address |  
+----+-----+-----+-----+-----+  
| 52 | 马深圳 | 13912340052 | 7 | Guangdong | 深圳 | 某某街某某号 |  
| 77 | 郝上海 | 13912340077 | 25 | Shanghai | 上海 | 某某街某某号 |  
| 20 | 许重庆 | 13912340020 | 4 | Chongqing | 重庆 | 某某街某某号 |  
| 67 | 岑南昌 | 13912340067 | 17 | Jiangxi | 南昌 | 某某街某某号 |  
+----+-----+-----+-----+-----+  
  
4 rows in set (0.00 sec)
```

➤ Limit for DNID

DNID Field is reserved field of the compute node, and it's forbidden from using DNID Field in Transaction Table, and using DNID as alias in SQL statement.

DNID is only applicable to simple single-table statements of SELECT, UPDATE, DELETE; besides, DNID could only serve as filer condition of WHERE Clause, and DNID Field can't be used in ORDER BY, GROUP BY, HAVING; similarly, it doesn't support using DNID in JOIN statement, UNION/UNION ALL, Subquery; DNID Field are forbidden from being used in Function and Expression.

9.2. HINT

If using HINT Syntax on compute node, you can bypass the HotDB Server Parser, and execute arbitrary SQL statement of MySQL on specified data node directly. Compute node supports two modes of HINT Syntax:

➤ Use DNID (data node ID) in HINT:

Syntax:

```
/*!hotdb:dnid = dnid_value*/ SQL to be executed
```

Note: the value of dnid_value is the ID of a data node. The user could specify the specific sharding node by replacing the value of dnid_value.

For example:

```
/*!hotdb:dnid = 1*/select * from customer where age > 20;
```

This statement will be executed on database node 1. The user could, via “Data Node” page on Distributed Transactional Database Management Platform, find the name of the data source with data node ID being 1, and search specified data source name of the “Data Node” page, thus the practical MySQL database could be located.

➤ Use DSID (data source ID) in HINT:

HINT statement supports the specified datasource_id to skip the compute node and send the statement directly to the data source. You can view the data source datasource_id through the service port command.

```
Syntax: SHOW [full] HOTDB { datasources } [LIKE 'pattern' | WHERE expr]
```

For example:

```
hotdb> show hotdb datasources where datasource_id like '22';
+-----+-----+-----+-----+
| DATANODE_ID | DATASOURCE_ID | DATASOURCE_NAME           | DATASOURCE_TYPE |
| DATASOURCE_STATUS | HOST          | PORT | SCHEMA | IDC_ID |
+-----+-----+-----+-----+
| 23          | 22           | 192.168.210.41_3308_hotdb157 | 1             |
| 192.168.210.41 | 3308 | hotdb157 | 1 |
```

- Specify a specific datasource_id without writing binlog:

```
/*!hotdb:dsid=nobinlog:datasource_id*/SQL to be executed
```

Note: the value of datasource_id is the ID of a certain data source, and you can specify multiple nodes and separate them by English commas. This statement will not record the executed statements into binlogof data source. If it is

not used properly, it may lead to the data inconsistency between the two masters and the disorder of GTID location. You should be careful when using it.

For Example:

Create the user hpt on the data source datasource_id=22 without writing binlog.

```
hotdb> /*!hotdb:dsid=nobinlog:22*/create user 'hpt'@'%' identified by '123456';
Query OK, 0 rows affected (0.00 sec)
```

Set parameters on the data source datasource_id=22.

```
hotdb> /*!hotdb:dsid=nobinlog:22*/set wait_timeout=1200;
Query OK, 0 rows affected (0.01 sec)

hotdb> /*!hotdb:dsid=nobinlog:22*/show variables like 'wait_timeout';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wait_timeout | 1200 |
+-----+-----+
1 row in set (0.01 sec)
```

- Specify all datasource_ids without writing binlog:

```
/*!hotdb:dsid=nobinlog:all*/SQL to be executed
```

Note: all refers to all data sources (including data sources in the DR center when the DR mode is enabled). This statement will not record the executed statements into binlog.

For example:

Update table 1 on all data sources without writing binlog:

```
/*!hotdb:dsid=nobinlog:all*/update table1 set name='hotdb' where id=100;
```

Set parameters on all data sources.

```
hotdb> /*!hotdb:dsid=nobinlog:all*/set wait_timeout=1200;
Query OK, 0 rows affected (0.00 sec)

hotdb> /*!hotdb:dsid=nobinlog:all*/show variables like 'wait_timeout';

+-----+-----+
| Variable_name | Value |
+-----+-----+
| wait_timeout | 1200 |
8 rows in set (0.00 sec)
```

- Specify a specific datasource_id with writing binlog:

```
/*!hotdb:dsid=datasource_id*/SQL to be executed
```

Note: this statement will record the executed statements into binlog of the corresponding data source. When operating data sources with replication relations, care should be taken to avoid the master-slave replication synchronization exception.

- Specify all datasource_ids with writing binlog:

```
/*!hotdb:dsid=all*/SQL to be executed
```

Note: all refers to all data sources (including data sources in the DR center when the DR mode is enabled). This statement will record the executed statements into binlog. For writing binlog may lead to replication exception of data sources with replication relations and disorder of GTID location. You should be careful when using it.

Note: if multiple data sources are distributed on the same instance, use of datasource_id in HINT needs to be executed by the specified data source.

➤ Use sharding key in HINT:

Syntax:

```
/*!hotdb:table = table_name:column_value*/ SQL to be executed
```

Note: table_name is the Table Name of a Sharding Table; column_value is a value of the Sharding Key in this table. The user could replace the table_name to specify corresponding Splitting Rule, and then via replacing the column_value, specify corresponding Sharding Node of using this Sharding Key.

For example:

```
/*!hotdb: table = customer:10001*/select * from customer where age > 20;
```

Use method:

Connect compute node (refer to Log in compute node and start to use), select the set LogicDB ('test'LogicDB is used here), and then execute specified statement by the above means (illustration is made here, and you could write SQL as needed in practical use).

Search customer table on sharding node with dn_id=2

```
mysql> /*!hotdb: dnid=2*/ select count(*) from customer;
```

```
+-----+
| count(*) |
+-----+
|      50 |
+-----+
1 row in set (0.00 sec)
```

Search the customer table with provinceid = 1 on sharding node

```
mysql> /*!hotdb: table=customer:1*/ select count(*) from customer;
+-----+
| count(*) |
+-----+
|      11 |
+-----+
1 row in set (0.00 sec)
```

Note: In terms of transaction layer, it's not recommended operating data source directly by HINT, because after using HINT, both the data and status will beyond control of the compute node. Besides, after using HINT operation, compute node will automatically perform the connection binding. Please follow the instructions after the connection binding.

9.3. Connection binding

In order to prevent the connection pool from being contaminated, after using HINT operation, compute node will bind associated back-end connection of the LogicDB used by the existing HINT Quer (that is connection between compute node and data source), and all operations involving back-end are allowed within the bound connection range. Therefore, after using HINT, it's recommended rebuilding new front-end connection to guarantee that the new Session connection is in clean and stable status. If not rebuild connection, then after using HINT, when other operations involve new data nodes beyond back-end connection bound by the original LogicDB, then the back-end connection bound previous will fail, and the front-end connection will be rebuilt automatically.

In addition to HINT, the statements involving connection binding also include the following statements:

```
set [session] foreign_key_checks=0;  
  
START TRANSACTION /*!40100 WITH CONSISTENT SNAPSHOT */  
  
set [session] UNIQUE_CHECKS=0;
```

After the SQL with connection binding is executed, the compute node will output Warning and log prompt:

For example, when executing the following statement, there will be warning prompt:

```
mysql> use db_a  
Database changed  
  
mysql> /*!hotdb:dnid=all*/select * From tba;  
+----+----+  
| id | a |  
+----+----+  
| 1 | 1 |
```

```
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
+---+---+
```

5 rows in set, 1 warning (0.01 sec)

Warning (Code 10041): The current session has been bound to the backend connection associated with the current LogicDB. It is recommended to rebuild the session after use.

Meanwhile, the following info information will be logged:

```
2019-04-01 19:11:29.662 [INFO] [CONNECTION] [$NIOEecutor-3-1] ServerConnection(1565) – 31 has been
bound to the backend connection:[2,1]
```

When operating new data node beyond bound back-end connection of the original LogicDB, SHOW WARNING will prompt as follow and the connection will be disconnected:

```
mysql> use db_b
Database changed

mysql> show warnings;
+-----+-----+
| Level | Code  | Message
|
+-----+-----+
| Note  | 10042 | The connection in current LogicDB was a binded connection, operations under current LogicDB
may cause connect abort. |
+
1 row in set (0.00 sec)
```

```
mysql> select * from tbb;  
  
ERROR 2013 (HY000): Lost connection to MySQL server during query  
  
ERROR 2016 (HY000): MySQL server has gone away  
  
No connection. Trying to reconnect...  
  
Connection id: 63  
  
Current database: db_b
```

When the HotDB Server version is 2.5.5 below, if LogicDB is not USEd, the compute node will bind all the back-end data sources connection by default, and other SQL statements after the connection binding statement will be distributed for execution, which may lead to modification or overwriting of the historical data of the original data source.

For example: log in to the service port, execute the connection binding statement without using the LogicDB:

```
set [session] foreign_key_checks=0;
```

Then directly execute

```
DROP TABLE IF EXISTS table_test;
```

At this time, the DROP statement will be directly distributed to all data sources for execution, and then the unrelated data will be deleted.

Therefore, when using the statement of connection binding, you need to pay extra attention: you need to use the LogicDB in advance for INSERT/REPLACE/UPDATE/DELETE/LOAD DATA/CREATE TABLE /ALTER TABLE/DROP TABLE/TRUNCATE TABLE/RENAME TABLE and other operations that will modify the data.

When HotDB Server version is 2.5.5 and above, this kind of scenario is further optimized to support the content parsing of connection binding.

In connection binding, whether the LogicDB is USED or not, it is not allowed to execute SQL that may destroy the data source (please note that HINT itself does not restrict the following types of SQL):

CREATE|ALTER|DROP DATABASE, SET SESSION SQL_LOG_BIN|GTID_NEXT,
SET GLOBAL, RESET MASTER|SLAVE, CHANGE MASTER, START|STOP
SLAVE|GROUP_REPLICATION, CREATE|ALTER|DROP|RENAME USER|ROLE,
GRANT, REVOKE, SET PASSWORD, SET DEFAULT ROLE, CLONE and other
SQL types.

For example: execute binding connection statement before executing DROP TABLE:

```
set foreign_key_checks=0;  
  
drop table if exists test1;  
  
ERROR 1289 (HY000): Command 'drop table test1' is forbidden when sql don't use database with table or use  
multi database, because he current session has been bound to the backend connection.
```

When the LogicDB is not USED, after executing the binding connection statement, you can execute SQL with LogicDB which is limited to be single: SELECT/INSERT/
REPLACE/UPDATE/DELETE/LOAD/CREATE TABLE/ALTER TABLE/DROP
TABLE/TRUNCATE TABLE/RENAME
TABLE/PREPARE/EXECUTE/DEALLOCATE/SET SESSION/SHOW etc.

For example: execute the binding connection statement before executing CREATE TABLE:

```
set foreign_key_checks=0;
```

Then directly execute

```
create table db_b.table_test (id int);
```

Query OK, 0 rows affected, 1 warning (0.18 sec)

9.4. EXPLAIN

In compute node, EXPLAIN statement is used for showing route plan of SQL statement.

```
mysql> explain select id,name,telephone from customer;
+-----+-----+
| DATANODE | SQL
+-----+-----+
| 1      | SELECT id, name, telephone FROM customer |
| 2      | SELECT id, name, telephone FROM customer |
| 3      | SELECT id, name, telephone FROM customer |
| 4      | SELECT id, name, telephone FROM customer |
| 5      | SELECT id, name, telephone FROM customer |
+-----+-----+
5 rows in set (0.01 sec)
```

DATANODE column is data node ID, and as shown in the above results, this SQL statement will be executed on data node with ID being 1, 2, 3, 4, 5.

If to show execution plan of MySQL in EXPLAIN, you could combine the [HINT](#) function in compute node:

```
mysql> /*!hotdb:dnid=13*/explain select * from customer;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | type | possible_keys | key | key_len | ref | rows | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

```
+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | customer | NULL   | NULL           | NULL | NULL      | NULL | 53 | NULL  |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Limit for EXPLAIN

EXPLAIN statement is only applicable to simple single-table statements of INSERT, SELECT, UPDATE, DELETE. It doesn't support EXPLAIN to show route plan of JOIN statement, UNION/UNION ALL, Subquery statement.

9.5. OnlineDDL

SupportOnlineDDL function on [management port \(3325\)](#) of compute node, guarantees that the online Read/Write transaction will not be blocked at the time of Table Change, the database can still provide foreign access, and the specific use methods are as follow:

- Log in 3325 management port, and onlineddl "[DDLSTATEMENT]" Syntax could be used to execute onlineddl statement, for example: onlineddl "alter table customer add column testddl varchar (20) default '测试 onlineddl'";
- Executing show @@onlineddl statement, it could show the current running OnlineDDL statement and execution speed of the statement, progress shows the current DDL execution progress (unit: %), speed shows the current DDL running speed (unit: row/ms), for example:

```
mysql> show @@onlineddl;
```

schema	onlineddl
progress	speed
<hr/>	
TEST_DML_JWY	ALTER TABLE CUSTOMER ADD COLUMN TESTDDL VARCHAR(20) DEFAULT '测 试 ONLINEDDL'
0.2300	23.3561
<hr/>	

Special description: execution of onlineddl statement doesn't mean completion of DDL, the return of "Query OK, 0 rows affected" only means that the DDL statement is executable, and if to see whether execution is completed, please view the progress showed in progress in show @@onlineddl. When the result of show @@onlineddl is null, it means that all DDL execution completed and there is no other DDL task at present, in case of midway DDL disconnection due to network or other abnormalities, it will roll back to the whole DDL;

9.6. NDB Cluster SQL Node Service

The compute node below the version of V2.5.2 has incomplete support toward SQL statement in SELECT Query type, especially SQL of Subquery type, therefore, NDB Cluster SQL Node Service (hereinafter referred to as NDB SQL) is introduced. This service uses SQL node of NDB Cluster for compatibility of SQL not supported by the compute node, and to be used for completing relatively complex Query statement computation under distributed environment. If to know how to install and deploy NDB Cluster SQL Node Service, please refer to the chapter of NDB Cluster SQL Node Service Deployment in [*Distributed Transactional Database HotDB Server \[Install Deploy\] Function Manual*](#).

9.6.1. NDB use restrictions

When NDB SQL mode starts, NDB SQL logic could be used only after meeting the following conditions simultaneously:

- If Global Unique is not enabled, there must be Primary Key or Unique Key, and both the Primary Key and Unique Key must sharding key or Auto Increment;
- When Global Unique is enabled, there must be Primary Key or Unique Key;
- The Primary Key and Unique Key in the above two items must be Single Field Value type;
- Execute the Query statement originally not supported by the compute node.

For the following SQL type statements, the compute node doesn't support itself, but supports Query after NDB SQL is enabled

MySQL statement type	Clause type	Function	Description
SELECT	INNER/LEFT JOIN/RIGHT JOIN	Operation Expression	column1+column2、column1-column2、column1*column2、column1/column2
		<=> or <>	
	T JOIN	% or MOD	Only support column% constant; not support column1% column2
		RAND()	2.3 Not support all rand() related Syntax, including group by rand(), order by rand()
	WHERE	/ or DIV	Only support column div constant; not support column1 div column2

	IN/IS NOT NULL/IS NULL/BETWEEN. ..AND/LIKE	
	<=> or <>	
	XOR	
	CAST()	
	INNER/LEFT JOIN/RIGHT JOIN ON	<p>Not support concat() to serve as JOIN condition in Operation Expression (on Clause condition), or as association condition in where Clause</p>
	CASE...WHEN...END	<p>Only support Field of the table judged single by CASE WHEN; not support conditional judgement of multi-table Field, such as: CASE WHEN column_name1=xx THEN column_name2 END; CASE WHEN must use table alias</p>
Function	MIN(MIN(column_name)) ABS(MAX())	Nested Function not supported
Multi-table (above three tables) Query	Mixed LEFT/INNER/NATURAL JOIN Single NATURAL JOIN	
Subquery	Query Operational Condition (any, all)	

	Nested Multi-layer Association Subquery	
--	---	--

In case of table structure non-geometry type Space Type Field and json type Field in the table structure, then the Query SQL originally not supported is still not supported;

Field type in WHERE condition of Query SQL originally not supported only supports: all shaping integration, decimal, char, enum, varchar type.

9.7. Online modification of sharding key

The online modification of sharding key by directly using SQL statements on the service port is supported in V2.5.6 and above. The business table will not lock the table during the modification, and the business can perform SIUD on the original table.

9.7.1. Usage

The statement of alter is as follows:

```
alter table table_name change shard column new_column;
```

For example, modify the sharding key id of source table sbtest1 to k, and execute:

```
root@127.0.0.1:hotdb 5.7.25 06:44:26> alter table sbtest1 change shard column k;
```

```
Query OK, 0 rows affected (2 min 2.27 sec)
```

9.7.2. Restrictions on use

- The source table must have a primary key or a unique key, and the length of the table name cannot exceed 45 characters;
- There can be no triggers on the source table, or the source table cannot be associated

with other triggers;

- The source table cannot have foreign key constraints;
- The new sharding key must be contained in the table structure, and cannot be the one currently used by the table;
- Data types of the new sharding key cannot be BIT, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT, TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB, GEOMETRY, POINT, LINESTRING, POLYGON, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, GEOMETRYCOLLECTION, JSON;
- The global table, vertical sharding table, parent table and child table are not supported. It only supports using alter to modify the sharding key for sharding tables;
- In the sharding table, using alter to modify the sharding key is not supported for the source table with sharding functions of RANGE, MATCH or ROUTE;
- When alter is used to modify the sharding key, there should not be a sharding plan modification task in progress in the source table;
- If there is master/slave data inconsistency in the source table, the detection will be directly skipped when alter is used to modify the sharding key (it is recommended to manually perform the master/slave data consistency detection through the management platform before execution);
- When the global unique constraint is enabled for the source table, the historical data of the unique constraint field of the source table must be unique when using alter to modify the sharding key;

10. Data type and character set support

10.1. Support of HotDB Server toward data type

10.1.1. Value type

MySQL data type	Support status	Description
BIT	Support	
TINYINT	Support	
SMALLINT	Support	
MEDIUMINT	Support	
INT	Support	
INTEGER	Support	
BIGINT	Support	
SERIAL	Support	Synchronous with BIGINT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE
SERIAL DEFAULT VALUE	Support	Synonymous with NOT NULL AUTO_INCREMENT UNIQUE
REAL	Not support	It's forbidden from being set as associate field type of sharding key and Parent/Child Table
DOUBLE	Not support	It's forbidden from being set as associate field type of sharding key and Parent/Child Table

FLOAT	Not support	It's forbidden from being set as associate field type of sharding key and Parent/Child Table
DECIMAL	Support	
NUMERIC	Support	

10.1.2. Date and time type

MySQL data type	Support status	Description
DATE	Support	
TIME	Support	
TIMESTAMP	Support	
	Support	
DATETIME	Support	
	Support	
YEAR	Support	

10.1.3. Character string type

MySQL data type	Support status	Description
CHAR	Support	

VARCHAR	Support	
BINARY	Support	
VARBINARY	Support	
TINYBLOB	Support	It's forbidden from being set as associate field type of sharding key and Parent/Child Table
BLOB	Support	It's forbidden from being set as associate field type of sharding key and Parent/Child Table
MEDIUMBLOB	Support	It's forbidden from being set as associate field type of sharding key and Parent/Child Table
LONGBLOB	Support	It's forbidden from being set as associate field type of sharding key and Parent/Child Table
TINYTEXT	Support	It's forbidden from being set as associate field type of sharding key and Parent/Child Table
TEXT	Support	It's forbidden from being set as associate field type of sharding key and Parent/Child Table
MEDIUMTEXT	Support	It's forbidden from being set as associate field type of sharding key and Parent/Child Table
LONGTEXT	Support	It's forbidden from being set as associate field type of sharding key and Parent/Child Table
ENUM	Support	
SET	Support	

10.1.4. Space type

Compute node supports use Space typespatial_type when Create Table; it supports use spatial_type in single-node SQL statement; it doesn't support secondary computation of spatial_type in cross-node SQL statement.

10.1.5. Other types

MySQL data type	Support status	Description
JSON	Support	It's forbidden from being used either as sharding key, Parent/Child Table associated Field, or as join Field

10.2. Support of HotDB Server toward character set

HotDB Server supports relevant setting of Character Set, and the Character Set and Collation Set supported at present are as follow:

Collation	Charset
latin1_swedish_ci	latin1
latin1_bin	latin1
gbk_chinese_ci	gbk
gbk_bin	gbk
utf8_general_ci	utf8
utf8_bin	utf8

```
| utf8mb4_general_ci | utf8mb4 |
| utf8mb4_bin        | utf8mb4 |
+-----+-----+
```

The Syntax associated with the Character Set is as follow, HotDB Server could also make synchronous Support, and the functions are consistent with that of MySQL:

Function classification	Syntax related
CREATE TABLE	col_name {CHAR VARCHAR TEXT} (col_length) [CHARACTER SET charset_name] [COLLATE collation_name] col_name {ENUM SET} (val_list) [CHARACTER SET charset_name] [COLLATE collation_name]
ALTER TABLE	ALTER TABLE tbl_name CONVERT TO CHARACTER SET charset_name [COLLATE collation_name]; ALTER TABLE tbl_name DEFAULT CHARACTER SET charset_name [COLLATE collation_name];

	<pre>ALTER TABLE t MODIFY tbl_name column_definition CHARACTER SET charset_name [COLLATE collation_name];</pre>
SET	<pre>SET NAMES 'charset_name' [COLLATE 'collation_name']</pre> <pre>SET CHARACTER SET charset_name</pre> <pre>set [session] {character_set_client character_set_results character_set_connection collation_connection} = xxx;</pre>
WITH	<p>With ORDER BY:</p> <pre>SELECT k FROM t1 ORDER BY k COLLATE latin1_german2_ci;</pre> <p>With AS:</p> <pre>SELECT k COLLATE latin1_german2_ci AS k1 FROM t1 ORDER BY k1;</pre> <p>With GROUP BY:</p> <pre>SELECT k FROM t1 GROUP BY k COLLATE latin1_german2_ci;</pre> <p>With aggregate functions:</p> <pre>SELECT MAX(k COLLATE latin1_german2_ci) FROM t1;</pre> <p>With DISTINCT:</p> <pre>SELECT DISTINCT k COLLATE latin1_german2_ci FROM t1;</pre>

	With WHERE: SELECT * FROM k WHERE a='a' COLLATE utf8_bin = 'a';
	With HAVING: SELECT * FROM k WHERE a='a' having a='a' COLLATE utf8_bin = a order by id;

11. Function and operator support

11.1. Support of HotDB Server toward function

This document only lists some functions upon special treatment, and if to know all functions supported by the compute node, please obtain the official *HotDB Server-v2.5.3 Latest List of Functions*.

Function name	Support status	Intercept or not	Description
ABS()	Support	No	
ACOS()	Support	No	
ADDDATE()	Support	No	
ADDTIME()	Support	No	
AES_DECRYPT()	Support	No	
AES_ENCRYPT()	Support	No	
AND, &&	Support	No	
Area()	Support	No	
AsBinary(), AsWKB()	Support	No	
ASCII()	Support	No	
ASIN()	Support	No	
=	Support	No	
:=	Not support	Yes	

AsText(), AsWKT()	Support	No	
ATAN2(), ATAN()	Support	No	
ATAN()	Support	No	
AVG()	Support	No	
BENCHMARK()	Not support	Yes	
BETWEEN ... AND ...	Support	No	
BIN()	Support	No	
BINARY	Support	No	
BIT_AND()	Not support	Yes	
BIT_COUNT()	Support	No	
BIT_LENGTH()	Support	No	
BIT_OR()	Not support	Yes	
BIT_XOR()	Not support	Yes	
&	Support	No	
~	Support	No	
	Support	No	
^	Support	No	
Buffer()	Support	No	

CASE	Support	No	
CAST()	Support	No	
CEIL()	Support	No	
CEILING()	Support	No	
Centroid()	Support	No	
CHAR_LENGTH()	Support	No	
CHAR()	Support	No	
CHARACTER_LENGTH()	Support	No	
CHARSET()	Support	No	
COALESCE()	Support	No	
COERCIBILITY()	Support	No	
COLLATION()	Support	No	
COMPRESS()	Support	No	
CONCAT_WS()	Support	No	
CONCAT()	Support	No	
CONNECTION_ID()	Support	No	connection_id of front-end session with the compute node
Contains()	Support	No	
CONV()	Support	No	
CONVERT_TZ()	Support	No	

CONVERT()	Support	No	Whether Sharding Table or Global Table, compute node doesn't support CONVERT (value, type) writing mode, and it only supports CONVERT (value using Character Set);
COS()	Support	No	
COT()	Support	No	
COUNT()	Support	No	
COUNT(DISTINCT)	Support	No	
CRC32()	Support	No	
Crosses()	Support	No	
CURDATE()	Support	No	
CURDATE(), CURRENT_DATE	Support	No	
CURRENT_ROLE()	Not support	Yes	Compute node doesn't support new role function of MySQL8.0
CURRENT_TIME(), CURRENT_TIME	Support	No	
CURRENT_TIMESTAMP(), CURRENT_TIMESTAMP	Support	No	
CURRENT_USER(), CURRENT_USER	Support	No	Return the current LogicDB username

CURTIME()	Support	No	
DATABASE()	Support	No	Return database name of the current compute node
DATE_ADD()	Support	No	
DATE_FORMAT()	Support	No	
DATE_SUB()	Support	No	
DATE()	Support	No	
DATEDIFF()	Support	No	
DAY()	Support	No	
DAYNAME()	Support	No	
DAYOFMONTH()	Support	No	
DAYOFWEEK()	Support	No	
DAYOFYEAR()	Support	No	
DECODE()	Support	No	
DEFAULT()	Support	No	
DEGREES()	Support	No	
DES_DECRYPT() (deprecated 5.7.6)	Support	No	
DES_ENCRYPT() (deprecated 5.7.6)	Support	No	
Dimension()	Support	No	

Disjoint()	Support	No	
DIV	Support	No	
/	Support	No	
ELT()	Support	No	
ENCODE()	Support	No	
ENCRYPT() (deprecated 5.7.6)	Support	No	
EndPoint()	Support	No	
Envelope()	Support	No	
<=>	Support	No	
=	Support	No	
Equals()	Support	No	
EXP()	Support	No	
EXPORT_SET()	Support	No	
ExteriorRing()	Support	No	
EXTRACT()	Support	No	
ExtractValue()	Support	No	
FIELD()	Support	No	
FIND_IN_SET()	Support	No	
FLOOR()	Support	No	
FORMAT()	Support	No	

FOUND_ROWS()	Not support	Yes	
FROM_BASE64()	Support	No	
FROM_DAYS()	Support	No	
FROM_UNIXTIME()	Support	No	
GeomCollFromText(), GeometryCollectionFromText()	Support	No	
GeomCollFromWKB(), GeometryCollectionFromWKB()	Support	No	
GeometryCollection()	Support	No	
GeometryN()	Support	No	
GeometryType()	Support	No	
GeomFromText(), GeometryFromText()	Support	No	
GeomFromWKB()	Support	No	
GET_FORMAT()	Support	No	
GET_LOCK()	Not support	Yes	
GLength()	Support	No	
>=	Support	No	
>	Support	No	
GREATEST()	Support	No	

GROUP_CONCAT()	Support	No	
GROUPING()	Not support	Yes	New function of MySQL8.0
GTID_SUBSET()	Support	No	
GTID_SUBTRACT()	Support	No	
HEX()	Support	No	
HOUR()	Support	No	
IF()	Support	No	
IFNULL()	Support	No	
IN()	Support	No	
INET_ATON()	Support	No	
INET_NTOA()	Support	No	
INET6_ATON()	Support	No	
INET6_NTOA()	Support	No	
INSERT()	Support	No	
INSTR()	Support	No	
InteriorRingN()	Support	No	
Intersects()	Support	No	
INTERVAL()	Support	No	
IS_FREE_LOCK()	Not support	Yes	

IS_IPV4_COMPAT()	Support	No	
IS_IPV4_MAPPED()	Support	No	
IS_IPV4()	Support	No	
IS_IPV6()	Support	No	
IS NOT NULL	Support	No	
IS NOT	Support	No	
IS NULL	Support	No	
IS_USED_LOCK()	Not support	Yes	
IS	Support	No	
IsClosed()	Support	No	
IsEmpty()	Support	No	
ISNULL()	Support	No	
IsSimple()	Support	No	
JSON_ARRAYAGG(col_or_expr) [over_clause]	Not support	Yes	New function of MySQL8.0 and 5.7
JSON_OBJECTAGG(key, value) [over_clause]	Not support	Yes	New function of MySQL8.0 and 5.7
JSON_PRETTY(json_val)	Not support	Yes	New function of MySQL8.0 and 5.7
JSON_STORAGE_FREE(json_val)	Not support	Yes	New function of MySQL8.0

JSON_STORAGE_SIZE(json_val)	Not support	Yes	New function of MySQL8.0 and 5.7
JSON_MERGE_PATCH(json_doc, json_doc[, json_doc] ...)	Not support	Yes	New function of MySQL8.0 and 5.7
JSON_TABLE(expr, path COLUMNS (column_list) [AS] alias)	Not support	Yes	New function of MySQL8.0
LAST_DAY	Support	No	
LAST_INSERT_ID()	Support	No	
LCASE()	Support	No	
LEAST()	Support	No	
<<	Support	No	
LEFT()	Support	No	
LENGTH()	Support	No	
<=	Support	No	
<	Support	No	
LIKE	Support	No	
LineFromText()	Support	No	
LineFromWKB(), LineStringFromWKB()	Support	No	
LineString()	Support	No	

LN()	Support	No	
LOAD_FILE()	Not support	Yes	
LOCALTIME(), LOCALTIME	Support	No	
LOCALTIMESTAMP, LOCALTIMESTAMP()	Support	No	
LOCATE()	Support	No	
LOG10()	Support	No	
LOG2()	Support	No	
LOG()	Support	No	
LOWER()	Support	No	
LPAD()	Support	No	
LTRIM()	Support	No	
MAKE_SET()	Support	No	
MAKEDATE()	Support	No	
MAKETIME()	Support	No	
MASTER_POS_WAIT()	Not support	Yes	
MATCH	Support	No	
MAX()	Support	No	
MBRContains()	Support	No	

MBRDisjoint()	Support	No	
MBREqual() (deprecated 5.7.6)	Support	No	
MBRIntersects()	Support	No	
MBROverlaps()	Support	No	
MBRTouches()	Support	No	
MBRWithin()	Support	No	
MD5()	Support	No	
MICROSECOND()	Support	No	
MID()	Support	No	
-	Support	No	
MIN()	Support	No	
MINUTE()	Support	No	
MLineFromText(), MultiLineStringFromText()	Support	No	
MLineFromWKB(), MultiLineStringFromWKB()	Support	No	
MOD()	Support	No	
% or MOD	Support	No	
MONTH()	Support	No	
MONTHNAME()	Support	No	

MPointFromText(),MultiPointFromText()	Support	No	
MPointFromWKB(), MultiPointFromWKB()	Support	No	
MPolyFromText(),MultiPolygonFromText()	Support	No	
MPolyFromWKB(),MultiPolygonFromWKB()	Support	No	
MultiLineString()	Support	No	
MultiPoint()	Support	No	
MultiPolygon()	Support	No	
NAME_CONST()	Support	No	
NOT BETWEEN ... AND ...	Support	No	
!=, <=>	Support	No	
NOT IN()	Support	No	
NOT LIKE	Support	No	
NOT REGEXP	Support	No	
NOT, !	Support	No	
NOW()	Support	No	
NULLIF()	Support	No	
NumGeometries()	Support	No	
NumInteriorRings()	Support	No	

NumPoints()	Support	No	
OCT()	Support	No	
OCTET_LENGTH()	Support	No	
OLD_PASSWORD() (deprecated 5.6.5)	Support	No	
, OR	Support	No	
ORD()	Support	No	
Overlaps()	Support	No	
PASSWORD()	Support	No	
PERIOD_ADD()	Support	No	
PERIOD_DIFF()	Support	No	
PI()	Support	No	
+	Support	No	
Point()	Support	No	
PointFromText()	Support	No	
PointFromWKB()	Support	No	
PointN()	Support	No	
PolyFromText(), PolygonFromText()	Support	No	
PolyFromWKB(), PolygonFromWKB()	Support	No	

Polygon()	Support	No	
POSITION()	Support	No	
POW()	Support	No	
POWER()	Support	No	
PROCEDURE ANALYSE()	Not support	Yes	
PS_CURRENT_THREAD_ID()	Not support	Yes	New function of MySQL8.0
PS_THREAD_ID(connection_id)	Not support	Yes	New function of MySQL8.0
QUARTER()	Support	No	
QUOTE()	Support	No	
RADIANS()	Support	No	
RAND()	Support	No	In join, the Sharding Table doesn't support any rand Syntax
RANDOM_BYTIES()	Support	No	
REGEXP	Support	No	
RELEASE_LOCK()	Not support	Yes	
REPEAT()	Support	No	
REPLACE()	Support	No	
REVERSE()	Support	No	

>>	Support	No	
RIGHT()	Support	No	
RLIKE	Support	No	
ROLES_GRAPHML()	Not support	Yes	New function of MySQL8.0
ROUND()	Support	No	
ROW_COUNT()	Not support	Yes	
RPAD()	Support	No	
RTRIM()	Support	No	
SCHEMA()	Support	No	1. select schema() return to LogicDB name; 2. show tables from information_schema; compute node is not supported, Query result is null;
SEC_TO_TIME()	Support	No	
SECOND()	Support	No	
SESSION_USER()	Support	No	select session_user(); the Query result is not information of the current user login the LogicDB but the user information of LogicDB associated node

SHA1(), SHA()	Support	No	
SHA2()	Support	No	
SIGN()	Support	No	
SIN()	Support	No	
SLEEP()	Not support	Yes	
SOUNDEX()	Support	No	
SOUNDS LIKE	Support	No	
SPACE()	Support	No	
SQL_THREAD_WAIT_AFT ER_GTIDS()(deprecated 5.6.9)	Not support	Yes	
SQRT()	Support	No	
SRID()	Support	No	
StartPoint()	Support	No	
STD()	Not support	Yes	
STDDEV_POP()	Not support	Yes	
STDDEV_SAMP()	Not support	Yes	
STDDEV()	Not	Yes	

	support		
STR_TO_DATE()	Support	No	
STRCMP()	Support	No	
SUBDATE()	Support	No	
SUBSTR()	Support	No	
SUBSTRING_INDEX()	Support	No	
SUBSTRING()	Support	No	
SUBTIME()	Support	No	
SUM()	Support	No	
SYSDATE()	Support	No	(Notice: SYSDATE of the test server adds parameter, making it equal to now(), therefore, there will be no difference in latency, in order to avoid Master/Slave data inconsistency)
SYSTEM_USER()	Support	No	The Query result is not user information of LogicDB, but random user information of LogicDB associated node
TAN()	Support	No	
TIME_FORMAT()	Support	No	
TIME_TO_SEC()	Support	No	

TIME()	Support	No	
TIMEDIFF()	Support	No	
*	Support	No	
TIMESTAMP()	Support	No	
TIMESTAMPADD()	Support	No	
TIMESTAMPDIFF()	Support	No	
TO_BASE64()	Support	No	
TO_DAYS()	Support	No	
TO_SECONDS()	Support	No	
Touches()	Support	No	
TRIM()	Support	No	
TRUNCATE()	Support	No	
UCASE()	Support	No	
-	Support	No	
UNCOMPRESS()	Support	No	
UNCOMPRESSED_LENGTH()	Support	No	
UNHEX()	Support	No	
UNIX_TIMESTAMP()	Support	No	
UpdateXML()	Support	No	
UPPER()	Support	No	

USER()	Support	No	select user(); the Query result is information of the current user of LogicDB
UTC_DATE()	Support	No	
UTC_TIME()	Support	No	
UTC_TIMESTAMP()	Support	No	
UUID_SHORT()	Support	No	
UUID()	Support	No	
VALIDATE_PASSWORD_STRENGTH()	Support	No	
VALUES()	Support	No	
VAR_POP()	Not support	Yes	
VAR_SAMP()	Not support	Yes	
VARIANCE()	Not support	Yes	
VERSION()	Support	No	The Query result shows the version of compute node
WAIT_UNTIL_SQL_THREADS_After_GTIDS()	Not support	Yes	
WEEK()	Support	No	
WEEKDAY()	Support	No	

WEEKOFYEAR()	Support	No	
WEIGHT_STRING()	Support	No	
窗口函数 Window Function	Not support	Yes	New function of MySQL8.0
Within()	Support	No	
X()	Support	No	
XOR	Support	No	
Y()	Support	No	
YEAR()	Support	No	
YEARWEEK()	Support	No	

11.2. MERGE_RESULT

MERGE_RESULT controls whether the compute node merges the Aggregate Function results or not. When this is set as 1, the compute node will merge the Aggregate Function results; when this value is set as 0, the compute node will not merge the Aggregate Function results.

By default, the MERGE_RESULT value is 1.

When MERGE_RESULT=0, for the SQL statement containing Aggregate Function, the compute node will not merge the Result Set, and the query result of each data node will be returned separately:

```
mysql> select count(*) from customer;
```

```
+-----+
```

```
| COUNT(*) |
```

```
+-----+
```

```
|      23 |
```

```
|      11 |
```

```
|      13 |
```

```
|      53 |
```

```
+-----+
```

```
4 rows in set (0.00 sec)
```

SET MERGE_RESULT=0 and SET SHOW_DNID=1, is available for statistics of distribution condition of transaction tables on various data nodes:

```
mysql> set MERGE_RESULT=0;
```

```
mysql> set show_dnid=1;
```

```
mysql> select count(*) from customer;
```

```
+-----+-----+
```

```
| COUNT(*) | DNID |
```

```
+-----+-----+
```

```
|      13 |    12 |
```

```
|      11 |    15 |
```

```
|      53 |    13 |
```

```
|      23 |    14 |
```

```
+-----+-----+
```

```
4 rows in set (0.00 sec)
```

DNID column in the Result Set shows unique ID of each data node. In the result, the actual data amount of the customer table on various data nodes has been showed intuitively.

When MERGE_RESULT=1, for the SQL statement containing Aggregate Function, the compute node shall return query results of all data nodes by SQL Semantics:

```
mysql> set show_dnid=0;

mysql> set MERGE_RESULT=1;

mysql> select count(*) from customer;

+-----+
| COUNT(*)      |
+-----+
|          100 |
+-----+

1 row in set (0.00 sec)
```

12. SQL syntax support

12.1. DML statement

In distributed database, logic of DML statement will become more complex. Compute node divides DML statement into two types: single-node DML statement and cross-node DML statement.

Single-node DML statement, refers to that SQL statement only needs to run on one node, and the accurate results could be computed. For example, assuming that in Sharding Table the customer has sharding key provinceid, then the following statement is single-node SELECT, because this statement will only run on the node where provinceid=1 resides:

```
SELECT * FROM customer WHERE provinceid=1;
```

Cross-node DML statement, refers to that SQL statement requires data of multiple data nodes, and then upon secondary processing of compute nodes, the final results could be integrated and computed. For example, assuming that in the Sharding Table the customer has sharding key provinceid, then the following SELECT statement is cross-node statement, because the id is not sharding key, the data with id>10 may distribute on multiple nodes, and in order to obtain the final result through integration and ranking, data of multiple nodes is required:

```
SELECT * FROM customer WHERE id>10 ORDER BY id;
```

Obviously, single-node SQL statement has better performance than cross-node SQL statement. When using compute node, try to use single-node DML statement as much as possible.

In the instance above, it only describes single-node and cross-node Query of simple single-table SELECT. Then in JOIN, if requiring data of multiple data nodes, it is called

cross-node JOIN; if requiring data of single data node, it is called single-node JOIN.

For Subquery statement, if requiring Query of data from multiple data nodes, it is called cross-node Subquery; if requiring data from single data node only, it is called single-node Subquery.

Query Support function of the compute node toward single-node JOIN, is the same as Support function of single-node SELECT statement. For support of compute node toward cross-node JOIN statement, please refer to [Cross-node JOIN](#)

12.1.1. DELETE statement

Since MySQL5.6.2, DELETE statement supports delete data from specified partition. If there are Table Name t and Partition name p0, the statement will delete all data from the Partition p0:

```
DELETE FROM t PARTITION(p0);
```

12.1.1.1. Single-node DELETE statement

MySQL statement type	Clause type	Function	Support status	Description
DELETE	PARTITION		Support	
	ORDER BY		Support	
	LIMIT		Support	

	WHERE	dnid	Support	1. dnid in DML where Clause shall be a necessary requirement, and it is not supported if not being a necessary requirement 2. Global Table does not support use dnid.
		Function	Support	
	Multi-table DELETE		Support	

12.1.1.2. Cross-node DELETE statement

MySQL statement type	Clause type	Function	Support status	Description
DELETE	PARTITION		Support	
	ORDER BY DESC ASC		Support	
	LIMIT		Support	
	ORDER BY ... LIMIT ...		Support	Parent/Child Table is not supported
	ORDER BY case sensitive of Field value		Support	
	WHERE	Function in WHERE	Support	

	JOIN		Support	Scenarios with temporary tables are not supported
--	------	--	---------	---

In cross-node DELETE statement, the following Multi-table statement is not supported:

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
```

```
tbl_name[.*] [, tbl_name[.*]] ...
```

```
FROM table_references
```

```
[WHERE where_condition]
```

Or:

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
```

```
FROM tbl_name[.*] [, tbl_name[.*]] ...
```

```
USING table_references
```

```
[WHERE where_condition]
```

12.1.2. INSERT statement

12.1.2.1. Single-node INSERT statement

MySQL statement type	Clause type	Function	Support status	Description
INSERT	INSERT ... SELECT ...	Single-node simple single-table Query	Support	
		Single-node JOIN	Support	

	Single-node Subquery	Support	
	Single-node UNION/UNION ALL	Support	
IGNORE		Support	
PARTITION		Support	
ON DUPLICATE KEY UPDATE		Support	
INSERT INTO table_name(columns ...) VALUES(values...)		Support	
INSERT INTO ... VALUES()		Support	
INSERT INTO ... SET		Support	
Sharding Table Splitting-free Field		Not support	
Sharding Table Splitting Field value is NULL		Support	NULL value parameter needs to be configured in Sharding Function

	Child Table Non-Associated Field value		Not support	INSERT operation of child table data must meet foreign key condition
	Child Table Associated Field value is NULL		Not support	INSERT operation of child table data must meet foreign key condition
		Sharding Table	Support	
		Global Table	Support	
	INSERT BATCH	Child Table	Conditional limit	Associated field of Parent Table is not supported if it is not sharding key.

- INSERT INTO...SELECT...
- For INSERT INTO... SELECT ... statement, if the SELECT clause is not supported, INSERT INTO... SELECT...is not supported either. It can be executed in other circumstances.

- **INSERT IGNORE**

- On compute node, INSERT IGNORE reserves original features of MySQL. In case of Primary Key/Unique Key conflict, data and conflict information will be ignored.
- Parent/Child Table does not support inser/replace into... select....

```
mysql> create table test(id int not null Primary Key,provinceid int)engine=innodb;
```

```
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> insert into test set id = 1,provinceid=2;
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from test;
```

```
+----+-----+
```

```
| id | provinceid |
```

```
+----+-----+
```

```
| 1 | 2 |
```

```
+----+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> insert ignore into test set id = 1,provinceid=2; -- there has already existed the record of Primary Keyid = 1, data is ignored.
```

```
Query OK, 0 rows affected (0.00 sec)
```

For operation of INSERT IGNORE statement in Sharding Table, if in INSERT statement, sharding key and sharding key value are not given, then the compute node will judge whether to ignore the SQL statement according to whether the global unique constraint

is enabled.

For example, test is a sharding table, and id is a sharding key.

```
mysql> CREATE TABLE `test2` (
    `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
    `name` varchar(20) DEFAULT NULL,
    UNIQUE KEY `id` (`id`),
    UNIQUE KEY `name` (`name`)
) ENGINE=InnoDB;
```

```
mysql> insert ignore into test2(name) values ('e');
```

```
mysql> insert ignore into test2(name) values ('e')
```

With the global unique constraint disabled, when inserting the second row, if the values of id column 1 and 2 are routed to the same node, the second SQL will be ignored. If it is not the same node, the second row will be inserted successfully.

With the global unique constraint enabled, when inserting the second row, if the values of id column 1 and 2 are routed to the same node, the second SQL will be ignored. If it is not the same node, the second SQL will also be ignored.

● INSERT no sharding key

If the value of the sharding key is not specified in the INSERT clause:

If the sharding key has a default value, the default value is used for routing by default;

If the sharding key has no default value, null will be filled in. If the null value is configured with routing rules, it can be inserted. If the null value is not configured with routing rules, it is not allowed to insert (for example, range\match is the type that requires configuration of null sharding functions; auto_crc32 is the type that automatically route according to null)

12.1.2.2. Cross-node INSERT statement

In Distributed Transactional Database, INSERT statement could generate cross-node INSERT statement only under two conditions: INSERT ... SELECT and INSERT BATCH.

INSERT BATCH refers to the mode of writing single INSERT statement into multi-row records:

```
INSERT INTO ... table_name VALUES(),VALUES(),VALUES();
```

MySQL statement type	Clause type	Function	Support status	Description
INSERT	INSERT ... SELECT ...	Cross-node simple single-table Query	Support	
		Cross-node JOIN	Not support	
		Cross-node UNION	Not support	
	INSERT BATCH	Child Table	Support	JOIN Field of Parent Table is not supported if not being sharding key
		Global Table	Support	
		Sharding Table	Support	

● Special instructions for INSERT BATCH

For INSERT BATCH in a transaction, if parts succeed and parts failed, it will automatically roll back to the previous SAVEPOINT.

12.1.3. LOAD DATA statement

MySQL statement type	Clause type	Function	Support status	Description
LOAD DATA	LOAD DATA ... INFILE ... INTO TABLE		Support	<p>1. It is required that the database user of the compute node who executes the statement has the FILE privilege.</p> <p>2. When the compute node is in cluster mode, no matter on which server in the cluster this statement is executed, the imported file must be uploaded to the fixed path on the current active compute node server: <code>/usr/local/hotdb/hotdb-server/HotDB-TEMP.</code></p>
	LOW_PRIORITY		Not support	
	CONCURRENT		Not support	

LOCAL		Not support	
REPLACE		Support	
IGNORE		Support	
PARTITION		Not support	
CHARACTER SET		Not support	
{FIELDS COLUMNS} [TERMINATED BY 'string'] [[OPTIONALLY] ENCLOSED BY 'char'] [ESCAPED BY 'char']		Support	
LINES STARTING BY 'string'		Not support	
LINES TERMINATED BY 'string'		Support	
Import the specified field		Support	
SET		Support	

	IGNORE number {LINES ROWS}		Support	
--	---------------------------------	--	---------	--

12.1.4. REPLACE statement

12.1.4.1. Single-node REPLACE statement

MySQL statement type	Clause type	Function	Support status	Description
REPLACE	REPLACE SELECT	Single-node simple single-table Query	Support	
		Single-node JOIN	Support	
		Single-node Subquery	Support	
		Single-node UNION/UNION ALL	Support	
	IGNORE		Support	
	PARTITION		Support	
	ON DUPLICATE KEY UPDATE		Support	

	REPLACE INTO table_name(columns ...) VALUES(values...)	Support	
	REPALCE INTO ... VALUES()	Support	
	REPLACE INTO ... SET	Support	
	Sharding table has no sharding key	Not support	
	Sharding table sharding key value is NULL	Support	NULL value parameter needs to be configured in Sharding Function parameter
	Child table has no related field value	Not support	INSERT operation of child table data must meet foreign key condition

	Child table related field value is NULL		Not support	INSERT operation of child table data must meet foreign key condition
REPLACE BATCH	Sharding Table	Sharding Table	Support	
		Global Table	Support	
	Child Table		Conditional support	Associated field of Parent Table is not supported if it is not sharding key.

12.1.4.2. Cross-node REPLACE statement

In Distributed Transactional Database, REPLACE statement could generate cross-node INSERT statement only under two conditions: REPLACE ... SELECT and REPLACE BATCH.

REPLACE BATCH refers to the mode of writing single REPLACE statement into multiple rows of records:

```
REPLACE INTO ... table_name VALUES(),VALUES(),VALUES();
```

MySQL statement type	Clause type	Function	Support status	Description
REPLACE	REPLACE ... SELECT ...	Cross-node simple single-table Query	Support	
		Cross-node JOIN	Not support	
		Cross-node UNION	Not support	
	REPLACE BATCH	Child Table	Support	JOIN Field of Parent Table is not supported if it is not sharding key.
		Global Table	Support	
		Sharding Table	Support	

12.1.5. SELECT statement

12.1.5.1. Single-node SELECT statement

MySQL statement type	Clause type	Function	Support status	Description
SELECT	JOIN	LEFT JOIN	Support	
		INNER JOIN	Support	

	RIGHT JOIN	Support	
	CROSS JOIN	Support	
	Ordinary JOIN (Multi-table Query without JOIN key word)	Support	
	PARTITION Table	Support	
	Mixed JOIN of single table type	Support	
	Mixed JOIN of multi-table type	Support	
Subquery	JOIN	Support	
	IFNULL/NULLIF	Support	
	UNION/UNION ALL	Support	
	IS NULL/IS NOT NULL	Support	
	PARTITION Table	Support	
	Select from where Expression	Support	
	Select select Expression	Support	

	SELECT FROM SELECT Expression	Support	NDB service is used and NDB limit requirements are met in compute nodes.
UNION/UNION ALL	Simple single-table Query	Support	
	JOIN	Support	
	Subquery	Support	The same Support Syntax as Subquery
	Having Aggregate Function	Support	
	PARTITION Table	Support	
DISTINCTROW		Support	
DISTINCT		Support	
SELECT INTO		Not support	
STRAIGHT_JOIN		Support	
SQL_NO_CACHE		Support	
PARTITION		Support	

	WHERE	dnid	Support	<p>1. After set show_dnid=1, do not support where condition with dnid;</p> <p>2. dnid and Other conditions use or association, and only take dnid Condition;</p> <p>3. Not support SELECT Clause with dnid Expression, for example: select dnid=4 from dml_a_jwy;</p>
		Function	Support	Please refer to Function Description
	GROUP BY ASC DESC WITH ROLLUP		Support	
	HAVING		Support	
	ORDER BY ASC DESC		Support	
	LIMIT n,m		Support	
	PROCEDURE		Not support	

	INTO OUTFILE	Support	<p>1. It is required that the database user of the compute node who executes the statement has the FILE privilege.</p> <p>2. When the compute node is in cluster mode, no matter on which server in the cluster this statement is executed, the exported file must be uploaded to the fixed path on the current active compute node server:</p> <p>/usr/local/hotdb/hotdb-server/HotDB-TEMP.</p> <p>3. If the cluster is switched during the export, the data output can still be normal.</p>
	INTO DUMPFILE	Not support	

	INTO Variable		Not support	
	FOR UPDATE		Support	Not support collocation with NOWAIT or SKIP LOCKED
	LOCK IN SHARE MODE		Support	The same as FOR SHARE function of MySQL8.0, in order to guarantee downward compatibility, it's still reserved and supported
	FOR SHARE		Support	Support use on data source of MySQL8.0 and above; Not support collocation with NOWAIT or SKIP LOCKED
	Function	Including Aggregate Function	Support	Support complex operation beyond bracket of single-table Aggregate Function
	DUAL		Support	

	FORCE INDEX		Support	
	USING INDEX		Support	
	IGNORE INDEX		Support	

12.1.5.2. Cross-node SELECT statement

MySQL statement type	Clause type	Function	Support status	Description
SELECT	LIMIT n,m		Support	
	ORDER BY		Support	
	ORDER BY LIMIT n,m		Support	
	GROUP BY ASC DESC WITH ROLLUP		Support	
	GROUP BY ORDER BY LIMIT m,n		Support	
	GROUP BY/ORDER BY case sensitive of Field value		Support	
	Aggregate Function	Aggregate Function in SELECT Clause	Support	

	Aggregate Function in HAVING Clause	Support	
	COUNT(DISTINCT)	Support	
DISTINCT		Support	
INTO		Not support	
WHERE	Function	Support	
PARTITION		Support	
HAVING		Support	
PROCEDURE		Not support	

			<p>1. It is required that the database user of the compute node who executes the statement has the FILE privilege.</p> <p>2. When the compute node is in cluster mode, no matter on which server in the cluster this statement is executed, the exported file must be uploaded to the fixed path on the current active compute node server: /usr/local/hotdb/hotdb-server/HotDB-TEMP.</p> <p>3. If the cluster is switched during the export, the data output can still be normal.</p>
INTO OUTFILE	Support		

	INTO DUMPFILE		Not support	
	INTO Variable		Not support	
	FOR UPDATE		Support	
	LOCK IN SHARE MODE		Support	
	FORCE INDEX		Support	
	USING INDEX		Support	
	IGNORE INDEX		Support	
	STRAIGHT_JOIN		Support	
	JOIN		Limited support	<p>Please refer to Cross-node JOIN; For some JOIN SQL not supported by compute nodes, they can be supported when NDB service is used and NDB limit requirements are met in compute nodes.</p>
Subquery	JOIN	Support		
	IFNULL/NUL LIF	Support		

	UNION/UNION ALL	Support	
	IS NULL /IS NOT NULL	Support	
	PARTITION Table	Support	
	AVG/SUM/MIN/MAX Function	Support	
	Horizontal derived table	Not support	New function of MySQL8.0
	UNION/UNION ALL	join	Support

12.1.6. UPDATE statement

12.1.6.1. Single-node UPDATE statement

MySQL statement type	Clause type	Function	Support status	Description
UPDATE	LOW_PRIORITY		Support	
	IGNORE		Support	
	ORDER BY		Support	
	LIMIT n		Support	

	SET	Support	1. It is allowed to update the sharding key, but it is required that the change of the value of the sharding key will not affect the data routing, that is, the modified value of the sharding key and the value before the modification are routed to the same node, otherwise the execution is not successful. 2. The parent-child table is not allowed to use expression statement to update the associated fields of the parent-child table, even if the change of the value of the sharding key will not affect the data routing, such as SET id=id or SET id=id+3. 3. It is not supported to update a sharding key multiple times by one statement, for example: UPDATE table1 SET id =31, id=41 WHERE id =1;	

	WHERE	dnid	Support	When dnid serves as or Condition in DML where Condition, only dnid Condition is judged, while other limit conditions will be ignored
		Function	Support	
	Function		Support	
	Multi-table associated		Support	

12.1.6.2. Cross-node UPDATE statement

MySQL statement type	Clause type	Function	Support status	Description
UPDATE	ORDER BY DESC ASC		Support	
	LIMIT n		Support	
	ORDER BY DESC ASC LIMIT n,m		Support	Parent/Child Table is not supported
	ORDER BY case sensitive of Field value		Support	
	WHERE		Support	

				1. It is allowed to update the sharding key, but it is required that the change of the value of the sharding key will not affect the data routing, that is, the modified value of the sharding key and the value before the modification are routed to the same node, otherwise the execution is not successful. 2. The parent-child table is not allowed to use expression statement to update the associated fields of the parent-child table, even if the change of the value of the sharding key will not affect the data routing, such as SET id=id or SET id=id+3. 3. It is not supported to update a sharding key multiple times by one statement, for example: UPDATE table1 SET id =31, id=41 WHERE id =1;
	Function in SET Clause	Support		
	Function in WHERE	Support		

	PARTITION		Support	
	JOIN		Support	

12.1.7. Cross-node JOIN

Primary function	Secondary function	Tertiary function	Support status	Description
INNER/ LEFT JOIN	UNION ALL		Support	
	UNION		Support	
	HAVING	Unconditional Field	Not support	SELECT Clause must contain HAVING Filter Field, so does MySQL
	COUNT(*)		Support	
	AVG()		Support	
	MAX()		Support	
	MIN()		Support	
	SUM()		Support	
	别名 Alias		Support	
	ORDER BY	Single Field	Support	
		Multiple Field in the same order	Support	order by <i>column_name1</i> desc, <i>column_name2</i> desc

	Multiple Fields in different orders	Support	order by <i>column_name1</i> desc, <i>column_name2</i> asc
	Field Alias	Support	The Alias can't be the same with the Field Name in the Table
	Field value Case	Support	
	ENUM type	Support	
	Function	Support	
OR		Limited support	Condition in which Cross-node JOIN supports can transfer to in Condition; For some JOIN SQL not supported by compute nodes, they can be supported when NDB service is used and NDB limit requirements are met in compute nodes.

	WHERE	OR condition of different fields	Limited support	<p>a=x and b=x or c=x not supported, only support the condition that OR Expression is sub-node of AND Expression, and the condition that there is only one or Expression, for example: select xxx from a,b where (c1 OR c2) and c3 and (c4 OR c5 OR c6) and c7..AND cN.. statement:</p> <p>Among which, every condition (C1, C2, etc.) in OR Clause only supports table.column [= < <= > = !=] value or IS [NOT] NULL or specific value (0/1/TRUE/FALSE/character string, etc.);</p> <p>For some JOIN SQL not supported by compute nodes, they can be supported when NDB service is used and NDB limit requirements are met</p>
--	-------	----------------------------------	-----------------	--

				in compute nodes.
	or Condition of Single Field	Limited support		or Expression in left join which is not sub-node of and Expression is not supported; For some JOIN SQL not supported by compute nodes, they can be supported when NDB service is used and NDB limit requirements are met in compute nodes.
	IN	Support		
	AND	Support		
	IS NOT NULL	Support		
	IS NULL	Support		

	BETWEEN ... AND ...	Support	
	>、>=、<、<=	Support	
	NOW() and other constant Expression	Support	column1 > NOW() or column1 > DATE_ADD(NOW(), INTERVAL +3 day)
	Operation Expression	Special support	column1=column2+1(Sup port of using NDB and meeting NDB limit)
	LIKE	Support	
GROUP BY	Single Field	Support	
	Multiple Field	Support	
	ORDER BY NULL	Support	
	WITH ROLLUP	Support	
	Field Alias	Support	The Alias can't be the same with the Field name in the Table Name
	Field value Case	Support	
FORCE INDEX		Support	
USING INDEX		Support	

	IGNORE INDEX		Support	
	AVG	AVG()	Support	Not support nested Function, AVG(SUM(<i>column_name</i>))
		AVG()	Support	
		AVG(IFNULL())	Support	
		AVG(<i>column1-column2</i>)	Support	Only support operation of single-table columns, Multi-table Field is not supported; operation of Multi-table Field has been intercepted
	COUNT	COUNT DISTINCT	Support	
		COUNT()	Support	
		COUNT(*)	Support	
		COUNT(1)	Support	
	MIN	MIN()	Support	Nested Function is not supported
	MAX	MAX()	Support	Nested Function is not supported
	SUM	SUM()	Support	

	SUM(CASE ... WHEN...)	Support	Only support Field judged as single table by CASE WHEN, and the CASE WHEN Field must have Table Alias
	SUM(IFNULL())	Support	Procedure Control Function
	SUM(<i>column1</i> - <i>column2</i>)	Support	Only support operation of single-table columns, Multi-table Field is not supported; operation of Multi-table Field has been intercepted

	INTO OUTFILE	Support	<p>1. It is required that the database user of the compute node who executes the statement has the FILE privilege.</p> <p>2. When the compute node is in cluster mode, no matter on which server in the cluster this statement is executed, the exported file must be uploaded to the fixed path on the current active compute node server: <code>/usr/local/hotdb/hotdb-server/HotDB-TEMP.</code></p> <p>3. If the cluster is switched during the export, the data output can still be normal.</p>
	FOR UPDATE	Not support	
	LOCK IN SHARE MODE	Not support	
	Subquery	Support	See SELECT statement

	Table Alias		Support	Support using the Table Alias where a.column or select a.column
ON Clause	Single =	Support		
	<=>	Special support	Support of using NDB and meeting NDB limit	
	!= <>	Support		
	>= > <= <	Support		
	Multiple >= > <= <Condition	Support		
	Multiple and = Condition	Support		
	IN	Support	When LEFT JOIN, the Left Table Field using IN condition to filter is not supported in ON condition	
	IS NOT NULL	Support	When LEFT JOIN, the Left Table Field using IS NOT NULL condition to filter is not supported in ON condition	

	IS NULL	Support	When LEFT JOIN, the Left Table or Right Table Field using IS NULL condition to filter is not supported in ON condition
	BETWEEN ... AND ...	Support	When LEFT JOIN, the Left Table Field using BETWEEN ... AND... condition to filter is not supported in ON condition
	LIKE	Support	When LEFT JOIN, the Left Table Field using LIKE condition to filter is not supported in ON condition
	Or Condition	Special support	Support of using NDB and meeting NDB limit
	Mathematical Expression	Special support	Support of using NDB and meeting NDB limit, such as: column1=column2+1
SELECT Clause	Show Null Column	Support	SELECT " AS A FROM ... Query result could show accurate null column
	STRAIGHT_JOIN	Support	

Function	UNIX_TIMESTAMP()	Support	
	NOW()	Support	
	DATE_FORMAT()	Support	
	DATE_ADD()	Support	
	DATEDIFF()	Support	
	FROM_UNIXTIME()	Support	
	CONVERT	Support	
	SUBSTRING_INDE X()	Support	
	SUBSTRING()	Support	
	TRIM()	Support	
	RTRIM()	Support	
	LTRIM()	Support	
	UCASE()	Support	
	UPPER()	Support	
	FLOOR()	Support	
	% or MOD	Support	Only support column% constant; Not support column1%column2

	RAND()	Special support	Support of using NDB and meeting NDB limit
	TRUNCATE()	Support	
	/ or DIV	Support	Only support column div constant; Not support column1 div column2
	ABS()	Support	
	LENGTH()	Support	
	CONCAT()	Support	Not support concat() to serve as JOIN condition (on Clause condition) in Operation Expression, or as association condition in where Clause
	CAST()	Support	
	IF()	Support	
	IFNULL	Support	

	CASE...WHEN...END	Support	Only support the Field judged as single table by CASE WHEN; Not support condition judgment of Multi-table Field, such as: CASE WHEN column_name1=xx THEN column_name2 END; CASE WHEN must use table alias
	DISTINCT	Support	
USING(column)		Support	
PARTITION		Support	
LIMIT	LIMIT n,m	Support	
	LIMIT n	Support	
Multi-table (above three tables) Query	Single LEFT JOIN	Support	
	Single INNER JOIN	Support	
	Single NATURAL JOIN	Special support	Support of using NDB and meeting NDB limit
	Mixed LEFT/INNER JOIN/RIGHT JOIN	Support	

	Mixed LEFT/INNER/NATU RAL JOIN	Special support	Support of using NDB and meeting NDB limit
	table a ... join (table b,table c) ...	Support	left join, right join does not support in of on condition
	NATURAL JOIN	Special support	Support of using NDB and meeting NDB limit
	Table of different nodes JOIN	Support	
JOIN	UPDATE ... JOIN	Support	
	DELETE ... JOIN	Support	

12.2. DDL statement

12.2.1. ALTER statement

MySQL statement type	Clause type	Support status	Description
ALTER	ADD COLUMN	Support	

TABLE	ADD PRIMARY KEY/UNIQUE/FOREIGN KEY/FULLTEXT/INDEX/KEY	Support	Support ADD UNIQUE [index_name][index_type]index_col_name
	ADD FOREIGN KEY for child table(s)	Partial Support	When the non-sharding key is used as the foreign key associated field, foreign key reference between parent and child tables cannot be guaranteed when crossing nodes. That is to say, in MySQL, if the foreign key values of the parent table and the child table are equal, the data can be inserted after they are matched. In the distributed environment, however, when the non-sharding key is used as the foreign key associated field, the foreign key values corresponding to the parent table cannot be found in the data source of the final route of the child table, for the nodes routed by the foreign key associated field of the child table are inconsistent with the routed nodes of the sharding key of the parent table, hence the insertion failed: ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails

	ADD SPATIAL [INDEX KEY]	Support	
	ADD CONSTRAINT [CONSTRAINT [symbol]] PRIMARY KEY/UNIQUE KEY/FOREIGN KEY	Support	

	ADD CONSTRAINT [CONSTRAINT [symbol]] FOREIGN KEY for child table(s)	Partial Support	<p>When the non-sharding key is used as the foreign key associated field, foreign key reference between parent and child tables cannot be guaranteed when crossing nodes. That is to say, in MySQL, if the foreign key values of the parent table and the child table are equal, the data can be inserted after they are matched. In the distributed environment, however, when the non-sharding key is used as the foreign key associated field, the foreign key values corresponding to the parent table cannot be found in the data source of the final route of the child table, for the nodes routed by the foreign key associated field of the child table are inconsistent with the routed nodes of the sharding key of the parent table, hence the insertion failed: ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails</p>
	ALGORITHM	Support	New INSTANT of MySQL8.0, and INSTANT is used by default
	ALTER COLUMN	Support	
	LOCK	Support	

	MODIFY/CHANGE [COLUMN]	Support	
	DROP COLUMN	Support	
	DROP PRIMARY KEY/KEY/INDEX/FOREIG N KEY	Support	
	DISABLE KEYS	Support	
	ENABLE KEYS	Support	
	DISCARD TABLESPACE	Not support	
	IMPORT TABLESPACE	Not support	
	ADD/DROP/TRUNCATE PARTITION	Support	
	GENERATED COLUMNS	Support	New function of MySQL8.0 and 5.7
	SECONDARY INDEXES	Support	New function of MySQL8.0 and 5.7
	CHECK	Support	New function of MySQL8.0
ALTER	VIEW	Support	Supported in V2.5.6 and above

12.2.2. CREATE statement

MySQL statement type	Clause type	Support status	Description

CREATE DATABASE		Support	Create database is supported in V2.5.6 and above. The function instructions can be seen at the bottom of the table.
CREATE EVENT		Forbidden	
CREATE FUNCTION		Limited	Supported in single node
CREATE INDEX	FOREIGN KEY	Support	
	UNIQUE	Support	

	FOREIGN KEY for child table(s)	Partial Support	When the non-sharding key is used as the foreign key associated field, foreign key reference between parent and child tables cannot be guaranteed when crossing nodes. That is to say, in MySQL, if the foreign key values of the parent table and the child table are equal, the data can be inserted after they are matched. In the distributed environment, however, when the non-sharding key is used as the foreign key associated field, the foreign key values corresponding to the parent table cannot be found in the data source of the final route of the child table, for the nodes routed by the foreign key associated field of the child table are inconsistent with the routed nodes of the sharding key of the parent table, hence the insertion failed: ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
	FULLTEXT	Support	

	SPATIAL	Support	
	ALGORITHM	Support	
	LOCK	Support	
	FUNCTIONAL KEYS	Support	New function of MySQL8.0
CREATE TABLE	CREATE TEMPORARY TABLE	Forbidden	SQL Parser supports this Syntax, not supports Temporary Table function
	CREATE TABLE [IF NOT EXISTS]	Support	
	CREATE TABLE LIKE	Support	

	CREATE TABLE AS SELECT ...	Support	<p>1. The data source user is required to have CREATE TEMPORARY TABLE privilege.</p> <p>2. It is required that the CREATE table and the SELECT table are associated with at least one same data node, otherwise the execution is unsuccessful: ERROR 10215 (HY000): [LOADTEST1] no overlapping datanode .</p> <p>3. CREATE TABLE ... IGNORE SELECT and CREATE TABLE ... REPLACE SELECT are not supported.</p>
	GENERATED COLUMNS	Support	New function of MySQL8.0 and 5.7
	SECONDARY INDEXES	Support	New function of MySQL8.0 and 5.7
	CHECK	Support	New function of MySQL8.0
CREATE TRIGGER		Support	At present, it only supports single-node, and need CREATE privilege granted, internal statement does not verify the privilege, DEFINER related is not

			supported at present, related Field show the current user when show triggers
CREATE VIEW		Support	

CREATE DATABASE is used to create LogicDB, the usage is as follows:

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name [create_option] ... [DEFAULT
DATANODE 'datanodeid']
```

Description:

create_option: [DEFAULT] { CHARACTER SET [=] charset_name | COLLATE [=] collation_name }

[DEFAULT DATANODE 'datanodeid'] You can specify the default sharding node. When it is not specified, all data nodes will be associated by default; when it is specified, the specified data node will be associated as the default sharding node of the LogicDB; when the specified datanodeid does not exist, it prompts: datanodeid not exists.

The statement of CREATING DATABASE on the server end is:

```
create database if not exists zjj_d3 default datanode '1,4';
```

- Associate non-existent data nodes

```
jing01@127.0.0.1:hotdb 8.0.13 03:00:45> create schema if not exists zjj_d3 default datanode '1,2';
ERROR 10040 (HY000): Datanode 2 not exists
```

- When the character set is specified, a warning will be given as follows

```
jing01@127.0.0.1:(none) 8.0.13 03:15:29> create schema if not exists zjj_d6 default datanode '1' default character set utf8 def
ault collate utf8_general_ci;
Query OK, 0 rows affected, 1 warning (0.05 sec)

Warning (Code 1300): The setting of character sets and collations will be ignored
```

Note: the warning prompt will be given when the character set and collations is specified, because setting the node-level character set and collations is actually invalid for the compute node and it is recognized according to the default configuration of the data source itself.

12.2.3. DROP statement

MySQL statement type	Clause type	Support status	Description
DROP DATABASE		Forbidden	
DROP EVENT		Forbidden	
DROP FUNCTION		Forbidden	
DROP INDEX	UNIQUE	Support	
	Regular index KEY	Support	
	FOREIGN KEY	Support	
		Support	
	FULLTEXT	Support	
	SPATIAL	Support	
	ALGORITHM	Support	
DROP TABLE	DROP [TEMPORARY] TABLE [IF EXISTS]	Forbidden	
	DROP TABLE	Support	

	DROP TABLE multi-table	Support	Multi-table must be guaranteed in the same node
	DROP TABLE table_name [RESTRICT CASCADE]	Support	
DROP TRIGGER		Support	DROP privilege shall be granted
DROP VIEW		Support	

12.2.4. TRUNCATE AND RENAME statement

MySQL statement type	Clause type	Support status	Description
RENAME TABLE		Support	<p>1. RENAME multiple tables is supported, but these tables are required to be on the same node. Otherwise, the execution will fail and an error will be reported: ERROR 10042 (HY000): unsupported to rename multi table with different datanodes.</p> <p>2. The target table of RENAME does not need to be added with the table configuration in advance. If you add the table configuration to</p>

			<p>the new table, you need to ensure that the configuration of the new table is consistent with the old table, otherwise RENAME will not succeed</p> <p>Note: database users of the compute node need to have ALTER and DROP privileges of the old table, and CREATE and INSERT privileges on the new table.</p>
TRUNCATE TABLE		Support	

12.3. Transaction management and locking statement

Statement type	Transaction statement	Statement parameter	Status	Description
Transaction management	START TRANSACTION	No parameter	Support	
		WITH CONSISTENT SNAPSHOT	Support	
		READ WRITE	Support	
		READ ONLY	Support	
		BEGIN	Support	
		COMMIT	Support	

	COMMIT	[AND CHAIN] RELEASE]	[NO] [[NO]]	Support	
	ROLLBACK			Support	
	ROLLBACK	[AND CHAIN] RELEASE]	[NO] [[NO]]	Support	
	SET autocommit	0 1		Support	
SAVEPOINT NT	SAVEPOINT			Support	
	ROLLBACK TO		Support	
	RELEASE SAVEPOINT			Support	
LOCK	LOCK TABLES	READ [LOCAL]		Forbidden	
		[LOW_PRIORITY]		Forbidden	
		WRITE		Forbidden	
	UNLOCK TABLES			Forbidden	
	LOCK INSTANCE FOR BACKUP			Forbidden	
	UNLOCK INSTANCE;			Forbidden	

Transactio n isolation level statement	SET TRANSACTION	SESSION	REPEATABLE READ	Support	Fully supported in XA mode. In general mode, partial commit may be read.
---	--------------------	---------	--------------------	---------	--

	READ COMMITTED	Support	<p>Read&write inconsistency may exist in general mode;</p> <p>In XA mode, it is not supported in v.2.5.5 below, and it is supported in v.2.5.5 and above.</p> <p>In v.2.5.5 and above, however, strong read-write consistency will not be guaranteed under multiple cross-node queries. That is, for SQL such as select and insert select, if one SQL is converted to multiple SQL statements, the SQL execution result may be incorrect at this isolation level.</p> <p>Refer to the description of data strong consistency (XA transaction)</p>

		READ UNCOMMITTED	Not support	
		SERIALIZABLE	Support	Fully supported in XA mode. In general mode, partial commit may be read.
		REPEATABLE READ	Not support	Not support SET GLOBAL mode, only support SET SESSION
SET TRANSACTION	GLOBAL	READ COMMITTED	Not support	Not support SET GLOBAL mode, only support SET SESSION
		READ UNCOMMITTED	Not support	
		SERIALIZABLE	Not support	
SET TRANSACTION	SESSION	READ ONLY	Support	
		READ WRITE	Support	

	SET GLOBAL TRANSACTION	READ ONLY	Not support	
		READ WRITE	Not support	
Distributed transaction	XA START BEGIN ...	[JOIN RESUME]	Forbidden	
	XA END	[SUSPEND [FOR MIGRATE]]	Forbidden	
	XA PREPARE		Forbidden	
	XA COMMIT	[ONE PHASE]	Forbidden	
	XA ROLLBACK		Forbidden	
	XA RECOVER		Forbidden	
	XA RECOVER	[CONVERT XID]	Forbidden	New parameter of 5.7

12.4. Other MySQL statements

12.4.1. Storage procedure and custom function statement

HotDB Server only supports Storage Procedure, Custom Function statement in vertical table. (i.e. the LogicDB only associate one data node).

Statement type	SQL statement	Support status	Description
Storage Procedure	BEGIN ... END ...	Limited	Can be used in vertical table
	DECLARE	Limited	

	CASE	Limited	
	IF	Limited	
	ITRATE	Limited	
	LEAVE	Limited	
	LOOP	Limited	
	REPEAT	Limited	
	RETURN	Limited	
	WHILE	Limited	
	CURSOR	Limited	
	DECLARE ... CONDITION...	Limited	
	DECLARE ... HANDLER ...	Limited	
	GET DIAGNOSTICS	Limited	
	RESIGNAL	Limited	
	SIGNAL	Limited	

Plugin and User-Defined Function statement	CREATE [AGGREGATE] FUNCTION function_name RETURN S {STRING INT REAL DECIMAL}	Limited	
	DROP FUNCTION	Limited	
	INSTALL PLUGIN	Forbidden	
	UNINSTALL PLUGIN	Forbidden	

12.4.2. Prepare SQL Statement

Statement type	SQL statement	Support status	Description
Prepare SQL Statement	PREPARE ... FROM ...	Support	
	EXECUTE ...	Support	

{DEALLOCATE DROP} PREPARE	Support	
---------------------------------------	---------	--

12.4.3. User management statement

HotDB Server realizes a set of its own username and privilege management system, which could be merely operated on the Distributed Transactional Database Management Platform page. SQL statements of MySQL database user management are all Forbidden.

Statement type	SQL statement	Support status	Description
User management statement	ALTER USER	Forbidden	
	CREATE USER	Support	
	DROP USER	Support	
	GRANT	Support	
	RENAME USER	Forbidden	
	REVOKE	Support	
	SET PASSWORD	Forbidden	

Support the use of SQL statements to CREATE / DROP user and to GRANT / REVOKE user when the compute node version is higher than 2.5.6.

12.4.3.1. CREATE USER

The statement is:

```
CREATE USER [IF NOT EXISTS] 'user_name'@'host_name' IDENTIFIED BY 'password_auth_string'  
['user_name'@'host_name' IDENTIFIED BY 'password_auth_string']...
```

The statement of CREATE USER on the server end is:

```
create user 'jingjingjing'@'%' identified by 'jing' with max_user_connections 3;
```

When CREATE USER, the execution user must have super privilege. NULL password creation is not supported. The max length of user name is limited to 64 characters, and the password is not limited temporarily.

- When the execution user does not have super privilege, it will prompt:

```
jing01@127.0.0.1:(none) 8.0.13 11:45:12> create user test01 identified by 'test';  
ERROR 1211 (HY000): 'jing01'@'%' is not allowed to create new users  
jing01@127.0.0.1:(none) 8.0.13 11:45:39> 
```

- When creating with NULL password, it will prompt:

```
jing01@127.0.0.1:(none) 8.0.13 11:46:34> create user jing1;
ERROR 1396 (HY000): Operation CREATE USER failed for 'jing1'@'%', reason: empty password is not allowed
```

- If the user name exceeds the limit, it will prompt:

```
jing01@127.0.0.1:(none) 8.0.13 11:46:32> create user 'jingjingjingjingjingjingjingjingjingjingjing1'@'192.168.210.%' identified by 'jing';
ERROR 1396 (HY000): Operation CREATE USER failed, reason:recvie error response:Data too long for column 'username' at row 1
jing01@127.0.0.1:(none) 8.0.13 11:46:34>
```

- When creating user repeatedly, it will prompt:

```
jing01@127.0.0.1:(none) 8.0.13 11:48:10> create user jing1 identified by 'jing';
Query OK, 0 rows affected (0.12 sec)

jing01@127.0.0.1:(none) 8.0.13 11:48:10> create user jing1 identified by 'jing';
ERROR 1396 (HY000): Operation CREATE USER failed for 'jing1'@'%'
jing01@127.0.0.1:(none) 8.0.13 11:48:11>
```

12.4.3.2. DROP USER

The statement is:

```
DROP USER [IF EXISTS] 'user_name'@'host_name' [, 'user_name'@'host_name']...
```

The statement of DROP USER on the server end is:

```
drop user 'jingjingjing'@'%';
```

When DROP USER, the execution user must have super privilege.

- When the execution user does not have super privilege, it will prompt:

```
jing01@127.0.0.1:(none) 8.0.13 12:02:41> drop user jing1;
ERROR 1227 (HY000): Access denied; you need (at least one of) the SUPER privilege(s) for this operation
```

- When deleting a non-existent user, it will prompt:

```
jing01@127.0.0.1:(none) 8.0.13 01:55:21> drop user testtest;
ERROR 1396 (HY000): Operation DROP USER failed for 'testtest'@'%'
```

12.4.3.3. GRANT

The statement is:

```
GRANT
priv_type[, priv_type ] ...
ON priv_level TO 'user_name'@'host_name'[,'user_name'@'host_name'] ...
[WITH MAX_USER_CONNECTIONS con_num ]
```

Description:

priv_type includes: SELECT、 UPDATE、 DELETE、 INSERT 、CREATE 、DROP 、
ALTER 、FILE 、 SUPER

You can use ALL [PRIVILEGES] to give users all privileges (including super privilege), which is the same as MySQL.

priv_Level includes: * | *.* | db_name.* |db_name.tbl_name |tbl_name |

*: represents all tables in the current database (which can only be executed after “use” the LogicDB);
* *: represents all tables in all databases; db_name.*: represents all tables in a certain database, db_name specifies the database name; db_name.tbl_name: a certain table in a certain database, db_name, database name, tbl_name: represents a table, tbl_name specifies the name of the table (which can only be executed after “use” the LogicDB).

The statement of GRANT on the server end is:

The global grant:

```
grant all on *.* to 'test_ct'@'localhost' identified by 'test_ct' with max_user_connections 3;
```

The node-level grant:

```
grant all on test_ct.* to 'test_ct'@'localhost' identified by 'test_ct';
```

The table-level grant

```
grant update on test_ct.test_aa to 'test_ct'@'localhost' identified by 'test_ct';
```

Notes on GRANT:

1. The user who GRANTS must have super privilege.
2. User can be created synchronously when GRANT, but with password.
3. “super” and “file” must be granted global management privilege, and node-level and table-level grant is not supported.
4. “all” can not be used with other privileges at the same time. It can only be granted separately, which is the same as MySQL.
5. Privilege modification is only valid for the new connection and will not change the privilege of the created connection.

➤ When the user who GRANTS does not have super privilege, it will prompt:

```
jing01@127.0.0.1:(none) 8.0.13 02:04:30> grant update on *.* to 'jing1'@'localhost' identified by 'jing1' with max_user_connections 3;
ERROR 1227 (HY000): Access denied; you need (at least one of) the SUPER privilege(s) for this operation
```

➤ User is created synchronously when GRANT.

```
jing01@127.0.0.1:(none) 8.0.13 02:06:00> grant update on *.* to 'jing2'@'localhost' identified by 'jing1' with max_user_connections 3;
Query OK, 0 rows affected (0.13 sec)
```

➤ User is created synchronously when GRANT without password, it will prompt:

```
jing01@127.0.0.1:(none) 8.0.13 02:06:44> grant drop on *.* to jing3;
ERROR 1133 (HY000): Can't find any matching row in the user table
jing01@127.0.0.1:(none) 8.0.13 02:07:38>
```

- super must be granted global management privilege, and node-level and table-level grant is not supported.

```
jing01@127.0.0.1:(none) 8.0.13 02:07:38> grant super on *.* to jing2 identified by 'jing';
Query OK, 0 rows affected (0.19 sec)

jing01@127.0.0.1:(none) 8.0.13 02:08:30> grant super on hotdb.* to jing3 identified by 'jing';
ERROR 1049 (HY000): Illegal GRANT command; please refer to the manual to find out which privileges can be used
jing01@127.0.0.1:(none) 8.0.13 02:08:41>
```

- file must be granted global privilege, and node-level and table-level grant is not supported.

```
jing01@127.0.0.1:hotdb 8.0.13 02:15:46> grant file on *.* to jing5 identified by 'jing';
Query OK, 0 rows affected (0.05 sec)

jing01@127.0.0.1:hotdb 8.0.13 02:16:15> use hotdb
Database changed
jing01@127.0.0.1:hotdb 8.0.13 02:16:20> grant file on zjj11 to jing6 identified by 'jing';
ERROR 1049 (HY000): Illegal GRANT command; please refer to the manual to find out which privileges can be used
jing01@127.0.0.1:hotdb 8.0.13 02:16:25>
```

- all can not be used with other privileges at the same time. It can only be granted separately.

```
jing01@127.0.0.1:(none) 8.0.13 02:09:25> grant all privileges on *.* to 'jing12'@'%' identified by 'jing' with grant option;
Query OK, 0 rows affected (0.05 sec)

jing01@127.0.0.1:(none) 8.0.13 02:09:27> grant all,select on *.* to jing11 identified by 'jing';
ERROR 10010 (HY000): unsupported statement:grant all,select on *.* to jing11 identified by 'jing'
jing01@127.0.0.1:(none) 8.0.13 02:09:52>
```

12.4.3.4. REVOKE

The statement is:

```
REVOKE priv_type [, priv_type ] ...ON priv_level FROM 'user_name'@'host_name' [,  
'user_name'@'host_name'] ...
```

The statement of REVOKE on the server end is:

```
revoke select,update,delete,insert,create,drop,alter,file,super on *.* from jingjing05;
```

Notes on REVOKE:

1. The user who REVOKE must have super privilege.
2. You can REVOKE some privileges and all privileges, and you can also REVOKE

the corresponding privileges of node-level and table-level.

3. Privilege items can be REVOKEd repeatedly, but error will be reported when REVOKEing non-existent types.
4. Privilege modification is only valid for the new connection and will not change the privilege of the created connection.

- When the user who REVOKEs does not have super privilege, it will prompt:

```
jing01@127.0.0.1:(none) 8.0.13 02:26:27> revoke insert on *.* from 'jingjing1'@'localhost';
ERROR 1227 (HY000): Access denied; you need (at least one of) the SUPER privilege(s) for this operation
```

- REVOKE of some privileges is supported:

```
jing01@127.0.0.1:(none) 8.0.13 02:27:10> grant select,update,delete,insert,create,drop,alter,file,super on *.* to jingjing5 identified by 'jing' with max_user_connections 11;
Query OK, 0 rows affected (0.03 sec)

jing01@127.0.0.1:(none) 8.0.13 02:27:12> revoke select,update on *.* from jingjing5;
Query OK, 0 rows affected (0.05 sec)
```

- REVOKE of all privileges is supported:

```
jing01@127.0.0.1:(none) 8.0.13 02:27:18> grant all on *.* to 'jingjing2'@'localhost' identified by 'jing';
Query OK, 0 rows affected (0.04 sec)

jing01@127.0.0.1:(none) 8.0.13 02:27:59> revoke all on *.* from 'jingjing2'@'localhost';
Query OK, 0 rows affected (0.04 sec)
```

- REVOKE of node-level privileges is supported:

```
jing01@127.0.0.1:(none) 8.0.13 02:28:04> grant select,update,delete,insert,create,drop,alter,file,super on *.* to jingjing10 identified by 'jing' ;
Query OK, 0 rows affected (0.05 sec)

jing01@127.0.0.1:(none) 8.0.13 02:29:25> revoke select on hotdb.* from jingjing10;
Query OK, 0 rows affected (0.05 sec)
```

- REVOKE of table-level privileges is supported:

```
jing01@127.0.0.1:(none) 8.0.13 02:29:31> grant select,update,delete,insert,create,drop,alter,file,super on *.* to jingjing11 identified by 'jing' ;
Query OK, 0 rows affected (0.05 sec)

jing01@127.0.0.1:(none) 8.0.13 02:30:00> revoke drop on hotdb.zjj1 from jingjing11;
Query OK, 0 rows affected (0.04 sec)
```

- If you REVOKE a privilege and use it again, error will be reported as follows:

```
jing01@127.0.0.1:hotdb 8.0.13 02:38:37> revoke select on hotdb.zjj from jing1;
Query OK, 0 rows affected (0.07 sec)
```

```
jing1@127.0.0.1:hotdb 8.0.13 02:39:12> select * from zjj;
ERROR 1142 (HY000): [SELECT] command denied to user 'jing1' to table 'ZJJ' of logic database 'HOTDB'
jing1@127.0.0.1:hotdb 8.0.13 02:39:12>
```

12.4.4. Table maintenance statement

Statement type	SQL statement	Support status	Description
Table maintenance statement	ANALYZE TABLE	Forbidden	
	CHECK TABLE	Forbidden	
	CHECKSUM TABLE	Forbidden	
	OPTIMIZE TABLE	Forbidden	
	REPAIR TABLE	Forbidden	

12.4.5. SET statement

Statement type	SQL statement	Support status	Description
SET statement	SET GLOBAL	Not support	
	SET SESSION	Partial support	Such as: SET SESSION TRANSACTION/SET TX_READONLY/SET NAMES, etc.
	SET @@global.	Not support	

	SET @@session.	Not support	
	SET @@	Not support	
	SET ROLE	Forbidden	Compute node does not support new role function of MySQL8.0
User Variable	Custom	Support	Only support recall under single-node
	SET CHARACTER SET	Support	Only support: CHARACTER_SET_CLIENT CHARACTER_SET_CONNECTION CHARACTER_SET_RESULTS
	SET NAMES	Support	
	SET TRANSACTION ISOLATION LEVEL	Support	Under ordinary mode, the level supported is REPEATABLE READ, READ COMMITTED, SERIALIZABLE XA mode only supports REPEATABLE READ, SERIALIZABLE

12.4.6. SHOW statement

Statement type	SQL statement	Support status	Description

SHOW statement	SHOW AUTHORS	Support	
	SHOW BINARY LOGS	Support	
	SHOW BINLOG EVENTS	Support	
	SHOW CHARACTER SET	Support	
	SHOW COLLATION	Support	
	SHOW FIELDS FROM	Support	
	SHOW COLUMNS FROM IN tbl_name	Support	
	SHOW FULL COLUMNS FROM IN tbl_name	Support	
	SHOW CONTRIBUTORS	Support	
	SHOW CREATE DATABASE	Support	
	SHOW CREATE EVENT	Support	
	SHOW CREATE FUNCTION	Support	
	SHOW CREATE PROCEDURE	Support	
	SHOW CREATE TABLE	Support	
	SHOW CREATE TRIGGER	Support	
	SHOW CREATE VIEW	Support	
	SHOW DATABASES	Support	
	SHOW ENGINES	Support	
	SHOW ERRORS	Support	

	SHOW EVENTS	Support	
	SHOW FUNCTION STATUS	Support	
	SHOW GRANTS	Support	Show privilege control condition of the compute node
	SHOW INDEX FROM <i>db_name.table_name</i>	Support	
	SHOW INDEX FROM <i>table_name</i> WHERE…	Support	
	SHOW MASTER STATUS	Support	
	SHOW OPEN TABLES	Support	Show uniform null set
	SHOW PLUGINS	Support	
	SHOW PRIVILEGES	Support	
	SHOW PROCEDURE STATUS	Support	
	SHOW PROCESSLIST	Support	Show connection condition of the compute node
	SHOW PROFILES	Support	
	SHOW RELAYLOG EVENTS [IN ' <i>log_name</i> '] [FROM <i>pos</i>] [LIMIT [<i>offset</i> ,] <i>row_count</i>]	Support	
	SHOW SLAVE HOSTS	Support	
	SHOW SLAVE STATUS	Support	

	SHOW GLOBAL STATUS	Support	
	SHOW SESSION STATUS	Support	
	SHOW STATUS	Support	
	SHOW TABLE STATUS	Support	
	SHOW FULL TABLES	Support	
	SHOW TABLES	Support	
	SHOW TRIGGERS	Support	
	SHOW GLOBAL SESSION VARIABLES	Support	
	SHOW WARNINGS	Support	
	Show HOTDB tables	Support	<p>Support [{FROM IN} db_name] [LIKE 'pattern' WHERE expr]</p> <p>Show sharding information of the compute node</p>

12.4.7. HotDB PROFILE

Statement type	SQL statement	Support status	Description
SET statement	set hotdb_profiling={0 1 on off}	Support	Support set [session] hotdb_profiling

	show hotdb_profiles	Support	
SHOW statement	show hotdb_profile for query N [relative time real time]	Support	N represents the SQL id executed

Function Description: this function is limited to Session level only

show hotdb_profiles output is identical with MySQL:

Query_ID of the 1st column is id of SQL

Duration of the 2nd column is the time length of (the time point when the compute node completes writing the last Result Set to the front-end – the time point when the compute node starts to receive the first data packet of SQL), and the unit is: ms.

Query of the 3rd column is SQL text

Instance:

```
mysql> show hotdb_profiles;
+-----+-----+-----+
| Query_ID | Duration | Query           |
+-----+-----+-----+
|       1 |      422 | SELECT DATABASE() |
|       2 |    1962 | select * from aa |
+-----+-----+-----+
2 rows in set (0.01 sec)
```

Show hotdb_profile for query N [relative time|real time] is similar with MySQL:

Status of the 1st column, which is execution status of SQL statement

The 2nd column is relative time or real time, if the type is not specified, it is relative time by default, if the request has real time, then it shall be real time

Duration of the 3rd column, which is compute time of every step during SQL execution

process

Instance:

```
mysql> show hotdb_profile for query 2 relative time;

+-----+-----+-----+
| Status | Relative_time | Duration |
+-----+-----+-----+
| SQL receive start time | 0 | NULL |
| SQL receive end time | 67 | 67 |
| multiquery parse end time | 125 | 58 |
| multiquery other SQL execute wait end time | 155 | 30 |
| SQL parse end time | 709 | 553 |
| backend SQL 1 request generate start time | 709 | NULL |
| backend SQL 1 request generate end time | 892 | 182 |
| backend SQL 1 request send start time | 1329 | NULL |
| backend SQL 1 request send end time | 1390 | 61 |
| backend SQL 1 result receive start time | 1733 | NULL |
| backend SQL 1 result receive end time | 1842 | 109 |
| backend SQL 1 result rewrite start time | 1733 | NULL |
| backend SQL 1 result rewrite end time | 1849 | 116 |
| result send start time | 1849 | NULL |
| result send end time | 1962 | 112 |
+-----+-----+-----+
15 rows in set (0.00 sec)
```

Note: Description of the status column:

SQL receive start time: the time point when the compute node starts to receive the first

data packet of SQL

SQL receive end time: the time point when the compute node completes receive of SQL
(the SQL failed to be received will not continue to be output in later period)

multiquery parse end time: the time point when the compute node completes Multi-query parse (after parse failure, the subsequent rows will no longer be output)

multiquery other SQL execute wait end time: the time point when the compute node starts SQL parse (if there is no SQL to run ahead, then this row is unnecessary)

SQL parse end time: the time point when the compute node completes SQL parse (after parse failure, the subsequent rows will no longer be output)

backend SQL N request generate start time: the time point when the compute node starts generation of back-end SQL N (N is serial No. of back-end SQL, such as 1, 2, 3, 4...)

backend SQL N request generate end time: the time point when the compute node completes back-end SQL N

backend SQL N request send start time: the time point when the compute node starts to send the first back-end SQL N to the first MySQL

backend SQL N request send end time: the time point when the compute node completes sending the last back-end SQL N to the last MySQL

backend SQL N result receive start time: the time point when the compute node starts to receive the data packet of first SQL N Result Set from the first MySQL

backend SQL N result receive end time: the time point when the compute node completes receiving data packet of the last SQL N Result Set from the last MySQL

backend SQL N result rewrite start time: the time point when the compute node starts to rewrite the Result Set received (in case of SQL execution error or warning, this time period will contain processing time of the error or warning)

backend SQL N result rewrite end time: the time point when the compute node completes rewriting the Result Set received

result send start time: the time point when the compute node starts to send the first Result Set packet to the front-end

result send end time: the time point when the compute node completes sending the last Result Set packet to the front-end

12.4.8. Other MySQL management statement

Statement type	SQL statement	Support status	Description
Other management statements	BINLOG 'str'	Forbidden	
	CACHE INDEX	Forbidden	
	KILL [CONNECTION QUERY]	Support	
	LOAD INDEX INTO CACHE	Forbidden	
	RESET MASTER	Forbidden	
	RESET QUERY CACHE	Forbidden	
	RESET SLAVE	Forbidden	
MySQL Utility Statements	DESCRIBE DESC	Support	
	EXPLAIN	Support	Please refer to EXPLAIN
	EXPLAIN EXTENDED	Not support	

	HELP	Not support	
	USE	Support	

The use method of KILL statement is the same with that of MySQL KILL statement. KILL will simultaneously disable the front-end connection of the compute node, and data source connection of MySQL database.

12.4.9. SHOW VARIABLES AND SHOW STATUS

HotDB Server supports show results of some of variables and status of MySQL, and you could view information of the data source variable connected by the compute node via relevant Syntax.

Show description of the parameters without special treatment: when show_dnld=1, merge_result=0, the output data has a dnld column added, and all data show original value in the in-read data source; when show_dnld=1, merge_result=1, the data sources with the same variable value will converge to be shown in the same row; when show_dnld=0, the first data source variable value will be returned if not specially specified,

The following parameter are of special processing, the for its specific show results, please see the Show description:

MySQL VARIABLES	Show description
BIND_ADDRESS	*Always show*
TX_ISOLATION	REPEATABLE-READ is default. If it is session, show session's value. This parameter was removed in MySQL8.0, and was

	replaced with transaction_isolation
TRANSACTION_ISOLATION	New parameter of MySQL8.0, used for replacing tx_isolation
AUTO_INCREMENT_OFFSET	Show 1 at present
CHARACTER_SET_CONNECTION	Only support utf8/gbk/latin1/utf8mb4 Character Set
CHARACTER_SET_RESULTS	Only support utf8/gbk/latin1/utf8mb4 Character Set
MAX_CONNECTIONS	Show according to actual configuration of the compute node
MAX_USER_CONNECTIONS	Show according to actual configuration of the compute node
MAX_JOIN_SIZE	Only support set session max_join_size=xxx, show according to set value of the compute node
CHARACTER_SET_SERVER	Only support utf8/gbk/latin1/utf8mb4 Character Set
VERSION_COMMENT	HotDB Server by Hotpu Tech
INTERACTIVE_TIMEOUT	172800
SERVER_UUID	Always show 00000000-0000-0000-0000-000000000000
TX_READ_ONLY	OFF is default. If it is session, show session status. This parameter was removed in MySQL8.0, and was replaced with transaction_ready_only
TRANSACTION_READ_ONLY	New parameter of MySQL8.0, used for replacing tx_read_only

PORT	Show according to the configured service port value
AUTOCOMMIT	ON is default. If it is session, show session status.
HOSTNAME	MySQL5.7, show as Hostname of the compute node server
AUTO_INCREMENT	Distribute to any data source of the LogicDB to take AUTO_INCREMENT value
COLLATION_DATABASE	Only support utf8/gbk/latin1/utf8mb4 Character Set
CHARACTER_SET_DATABASE	Only support utf8/gbk/latin1/utf8mb4 Character Set
PROTOCOL_VERSION	Show according to practical Communication Protocol Version used by the compute node
READ_ONLY	Set according to the practical mode used by the compute node
VERSION	For MySQL Version No. - HotDB Server Version No., show according to the practical one used by the compute node
COLLATION_SERVER	<p>At present, only support:</p> <p>latin1_swedish_ci</p> <p>latin1_bin</p> <p>gbk_chinese_ci</p> <p>gbk_bin</p> <p>utf8_general_ci</p> <p>utf8_bin</p>

	utf8mb4_general_ci utf8mb4_bin
SOCKET	Show Null Character String
SERVER_ID	Show 0
WAIT_TIMEOUT	172800
SSL_CIPHER	Return Null Character String
COLLATION_CONNECTION	At present, only support: latin1_swedish_ci latin1_bin gbk_chinese_ci gbk_bin utf8_general_ci utf8_bin utf8mb4_general_ci utf8mb4_bin
FOREIGN_KEY_CHECKS	Show ON
CHARACTER_SET_CLIENT	Only support utf8/gbk/latin1/utf8mb4 Character Set
TIME_ZONE	Show SYSTEM
MAX_ALLOWED_PACKET	Compute node control, Default: 64M

ADMIN_ADDRESS	Always show Null Character String, new of MySQL8.0
INNODB_BUFFER_POOL_SIZE	Sum of all nodes under the LogicDB, Master/Slave node is counted as master node

Status Name	Show description
Compression	All OFF (compute node does not support Compression Protocol for the time being)
Innodb_buffer_pool_dump_status	The first status which is not ended with Not Started, otherwise, take the value of the first node of LogicDB
Innodb_buffer_pool_load_status	The first status which is not ended with Not Started, otherwise, take the value of the first node of LogicDB
Innodb_have_atomic_builtins	If there is OFF among all nodes under LogicDB, then it is OFF; if there are all ON, then it is ON
Innodb_page_size	Take the value of the first node of LogicDB
Innodb_row_lock_time_avg	Take simple average of all nodes under the LogicDB
Innodb_row_lock_time_max	Take Max Value of all nodes under the LogicDB
Last_query_cost	Always 0.000000
Last_query_partial_plans	Always 0

Max_used_connections	Take Max Value of all nodes under the LogicDB
Slave_heartbeat_period	Take Max Value of all nodes under the LogicDB
Slave_last_heartbeat	For date type value, take Min Value of all nodes under the LogicDB; if they are all Null Character String, then it is Null Character String
Slave_running	If there is OFF among all nodes under LogicDB, then it is OFF, if they are all ON, then it is ON
Ssl_cipher	Always return Null Character String
Ssl_cipher_list	Always return Null Character String
Ssl_ctx_verify_depth	Take the value of the first node of LogicDB
Ssl_ctx_verify_mode	Take the value of the first node of LogicDB
Ssl_default_timeout	Take the value of the first node of LogicDB
Ssl_server_not_after	Always return Null Character String
Ssl_server_not_before	Always return Null Character String
Ssl_session_cache_mode	Take the value of the first node of LogicDB
Ssl_verify_depth	Take the value of the first node of LogicDB
Ssl_verify_mode	Take the value of the first node of LogicDB
Ssl_version	Take the value of the first node of LogicDB
Tc_log_page_size	Take the value of the first node of LogicDB
Uptime	Take Max Value of all nodes under the LogicDB

Uptime_since_flush_status	Take Max Value of all nodes under the LogicDB
Caching_sha2_password_rsa_public_key	Always show Null Character String, new of MySQL8.0
Current_tls_ca	Always show Null Character String, new of MySQL8.0
Current_tls_capath	Always show Null Character String, new of MySQL8.0
Current_tls_cert	Always show Null Character String, new of MySQL8.0
Current_tls_cipher	Always show Null Character String, new of MySQL8.0
Current_tls_ciphersuites	Always show Null Character String, new of MySQL8.0
Current_tls_crl	Always show Null Character String, new of MySQL8.0
Current_tls_crlpath	Always show Null Character String, new of MySQL8.0
Current_tls_key	Always show Null Character String, new of MySQL8.0
Current_tls_version	Always show Null Character String, new of MySQL8.0
group_replication_primary_member	Always show Null Character String, new of MySQL8.0

mecab_charset	The first of LogicDB, New of MySQL8.0
Performance_schema_session_connect_attrs_longest_seen	Max of LogicDB, New of MySQL8.0
Rpl_semi_sync_master_clients	Always show 0, New of MySQL8.0
Rpl_semi_sync_master_net_avg_wait_time	Average of LogicDB, New of MySQL8.0
Rpl_semi_sync_master_status	Always show ON, New of MySQL8.0
Rpl_semi_sync_master_tx_avg_wait_time	Average of LogicDB, New of MySQL8.0
Rpl_semi_sync_slave_status	Always show ON, New of MySQL8.0
Rsa_public_key	Always show Null Character String, New of MySQL8.0

12.5. Special functions of compute node syntax

12.5.1. Create Table is Sharding

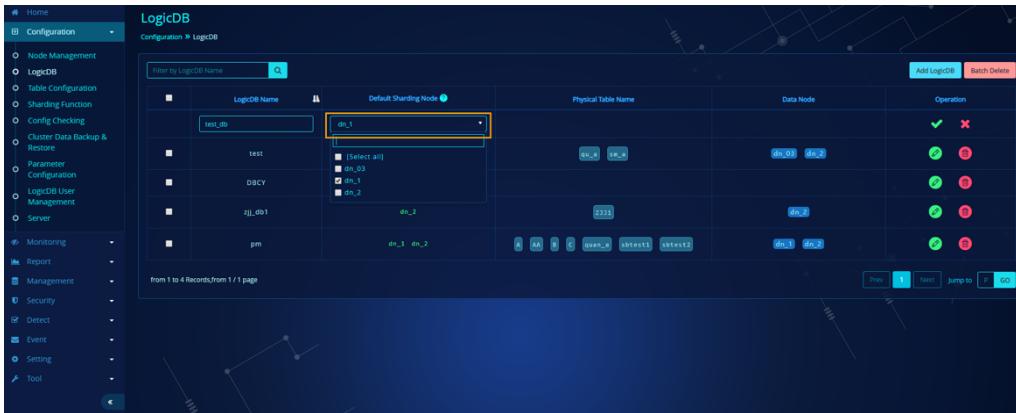
By default, the compute node could Create Table only after making appropriate Table Configuration. But in practical use process, user does not understand the compute node when contacting HotDB Server at the very beginning, and has no concept of Distributed Database; under this condition, there may be a simple method for transiting from MySQL to HotDB Server slowly and smoothly, that is: when Create Table, the function of associating node sharding by default according to LogicDB.

Function description: Add default node when Configure LogicDB. After Reload, Log in to compute node to Create Table directly, without needing to configure the Table Configuration Sharding Function. If LogicDB associates a node, then create Vertical Sharding Table; if LogicDB associates multiple nodes, then create Sharding Table of

AUTO_CRC32 type.

Premise for use: LogicDB sets default node

1. Log in to Distributed Transactional Database Management Platform, select “Configuration”->“LogicDB”, set LogicDB default node



- By default on the management platform, after Reload of LogicDB Configuration, table could be directly created by Create statement (without being defined on the management platform any longer), configuration will be generated automatically after Create Table (including table configuration and Sharding Function, etc.)
- If LogicDB selects single node, then Vertical Sharding Table will be created by default; if it selects multiple nodes, Sharding Table with sharding type of AUTO_CRC32 will be created by default, and will give priority to select Primary Key and Unique Key as the sharding key
- If to change the default node of LogicDB in later period, it will have no influence on tables created before change, and only applies the new LogicDB node to the new tables added after change

LogicDB		Physical Table Name				Data Node	Operation
LogicDB Name	Default Sharding Node						
test		customer	dn_1 dn_2 dn_3				
DB_0919		au houhou leele	dn_4 dn_5				
test_nc	dn_4	test_chuang	dn_6				
zjj_db1	dn_1 dn_2 dn_3	zzx1 zzx2	dn_1 dn_2 dn_3				
zy_db	dn_1 dn_2	load_tb1 TABLE09 table_bp	dn_1 dn_2 dn_3 dn_4				
test_zy	dn_1 dn_2 dn_3 dn_4	se_a_tb1 se_a_tb2 se_a_tb3 se_a_tb4 se_b_tb1 se_b_tb2 se_b_tb3 se_b_tb4 se_c_tb1 se_c_tb2 etc. 34	dn_1 dn_2 dn_3 dn_4				

from 1 to 8 Records, from 1 / 1 page

2. Log in to compute node to select “test001” LogicDB, and Create Table, create succeeded

For example, if the “test001” has only one default node, then “test01” is Vertical Sharding Table; if “test002” has multiple default nodes, then “test02” is Sharding Table

```
mysql> use TEST001;
```

Database changed

```
mysql> create table test01(id not null auto_increment Primary Key,a char(8),b decimal(4,2),c int);
```

```
mysql> use TEST002;
```

Database changed

```
mysql> create table test02(id not null auto_increment Primary Key,a char(8),b decimal(4,2),c int);
```

Table Configuration							
Please select LogicDB		Select Table Type	Select Sharding Type	Select Data Node	Select Table Structure	Batch Operation	
	LogicDB	Table Name	Table Type	Sharding Key	Sharding Function Name	Data Node	Table Structure
	test002	TEST02	Sharding Table	ID	Auto Sharding _ CRC32	dn_1 dn_2	Created
	test001	TEST01	Vertical Sharding Table			dn_1	Created
	test	customer	Sharding Table	provinceid	Auto Sharding _ CRC32	dn_1 dn_2 dn_3	Not Created
	DB_0919	leele	Global Table			dn_4 dn_5	Not Created
	DB_0919	houhou	Vertical Sharding Table			dn_5	Not Created

- For Sharding Table automatically created, selection sequence of sharding key is:

Primary Key Field -> Unique Key Field ->the 1st Integer Field (BIGINT, INT, MEDIUMINT, SMALLINT, TINYINT) ->after taking the Integer Field, take the Character String Type Field (CHAR, VARCHAR), and in case of no appropriate one after taking all types above, a Field will be randomly selected as the Sharding Key by default.

Notice: This function is only recommended to be used when contacting customer for the first time, and is not recommended for formal delivery and online, and Sharding shall be made according to the practical transaction scenarios;

12.5.2. Create table directly via existing Sharding Function

Function description: According to the added Sharding Functions on the management platform, Create Table directly via special statement on service port of the compute node, without configuring Table Sharding Information any longer. For tables created using the existing Sharding Functions, after deleting the tables according to rules, relevant tables on the management platform will be deleted synchronously.

Premise for use: If Sharding Functions already exist, to add Sharding Function, please refer to *Distributed Transactional Database HotDB Server [Management Platform] Function Manual*.

Add Sharding Function on the management platform, and reload.

Sharding Function					
Select Sharding Type		Filter by Sharding Function		Add Sharding Function	
Sharding Function ID	Sharding Function Name	Sharding Type	Table Name	Operation	
38	test_match1	MATCH			
37	37_CRC32_MOD	CRC32_MOD			
36	36_SIMPLE_MOD	SIMPLE_MOD			
35	35_SIMPLE_MOD	SIMPLE_MOD			
34	34_RANGE	RANGE			
33	33_RANGE	RANGE			
32	32_SIMPLE_MOD	SIMPLE_MOD			
31	31_CRC32_MOD	CRC32_MOD			
30	26_SIMPLE_MOD	SIMPLE_MOD			
25	25_MATCH	MATCH			

from 1 to 10 Records, from 1 / 2 page

Prev 1 2 Next Jump to GO

Use [service port command](#) to view functionid | functionname| functiontype| ruleid | rulename and other information of Sharding Function, and Create Table according to relevant Field information.

```
mysql> show hotdb functions;
```

Function_id	Function_name	Function_type	Auto_generated	
1	AUTO_CRC32_4	AUTO_CRC32	0	
2	fun_range	RANGE	0	
3	AUTO_CRC32_2	AUTO_CRC32	0	
4	AUTO_GENERATE_CRC32	AUTO_CRC32	1	
5	5_CRC32_MOD	CRC32_MOD	0	
6	AUTO_CRC32_3	AUTO_CRC32	0	
9	7_RANGE	RANGE	0	
11	AUTO_MOD_4	AUTO_MOD	0	
12	AUTO_MOD_3	AUTO_MOD	0	
13	AUTO_CRC32_1	AUTO_CRC32	0	
14	AUTO_MOD_1	AUTO_MOD	0	

15	15_RANGE	MATCH	0	
17	16_RANGE	RANGE	0	
18	18_RANGE	RANGE	0	
24	24_SIMPLE_MOD	SIMPLE_MOD	0	
25	25_MATCH	MATCH	0	
30	26_SIMPLE_MOD	SIMPLE_MOD	0	
31	31_CRC32_MOD	CRC32_MOD	0	
32	32_SIMPLE_MOD	SIMPLE_MOD	0	
33	33_RANGE	RANGE	0	
34	34_RANGE	RANGE	0	
35	35_SIMPLE_MOD	SIMPLE_MOD	0	
36	36_SIMPLE_MOD	SIMPLE_MOD	0	
37	37_CRC32_MOD	CRC32_MOD	0	
+-----+-----+-----+-----+				

24 rows in set (0.01 sec)

mysql> show hotdb rules;

Rule_id	Rule_name	Rule_column
Function_id	Auto_generated	
1	hotdb-cloud_eb04c4ba-2116-4ea5-ba32-40f97ba8616d	id 1
0		
2	hotdb-cloud_a3527f8c-fc7f-44d9-ad1f-243944e5c573	id 2
0		
3	hotdb-cloud_3703e424-69d7-474c-b925-03f65ec5f6c0	a 2
0		

4	hotdb-cloud_2fa6dfaa-1f82-449b-bff3-036bc0d762ce	a	1
0			
5	hotdb-cloud_8bd4884f-7bca-405f-b6bf-4b11da71dd2d	id	3
0			
6	AUTO_GENERATE_1_TABLEEMP_ID		ID
4	1		
7	hotdb-cloud_3ceac42d-015a-4a3b-8cc7-d4edcc34a5b1	id	1
0			
8	hotdb-server_cf24a81b-6ffa-494c-b69c-28c7d8318cb8_20190905185548	ID	6
1			
12	hotdb-server_bc53bcae-139a-4238-a598-47b52f6f9787_20190909104803	ID1	6
1			
13	hotdb-server_e662af7d-b968-4131-a7db-eac8d66c53c1_20190909134505	NAME	6
1			
14	hotdb-server_171fe946-63c9-47b9-bf52-10b9743fd064_20190909135113	NAME	6
1			
15	hotdb-server_f3f0dcc2-3d31-4536-8a5c-ec229c2983d7_20190909135444	NAME	6
1			
16	hotdb-server_261a36af-40d6-4854-9205-081e358e275d_20190909160204	ID	12
1			
17	hotdb-server_742cae4b-416f-4df4-92cf-cf45f070a2ec_20190909160303	NAME	12
1			
18	hotdb-server_e3e0db99-25e4-499c-a7b1-550919cb87aa_20190909165736	ID1	12
1			
19	hotdb-server_604d4b31-b369-4372-86e8-ce5f1c0a61b5_20190909190106	ID1	1
1			
20	hotdb-server_b3bb5cfcd-42b6-4cde-86c3-318ddee4bc1c_20190909190906	ID1	1
1			
21	hotdb-server_b385ebde-a28b-478a-8662-d8fbea964bfe_20190909191129	ID1	1
1			

```

| 26      | hotdb-cloud_3abf9ad2-d436-4e2d-ad32-2a60246a6102           | id          | 13
| 0      |
| 27      | hotdb-server_227c12e2-0c6c-4380-8f66-bab30f521a74_20190919163004 | ID          | 11
| 1      |
| 28      | hotdb-cloud_2ab203ed-885b-4cd2-9879-5e0f7191bafa           | provinceid | 6
| 0      |
+-----+-----+-----+-----+
21 rows in set (0.00 sec)

```

12.5.2.1. Sharding Table

Sharding table is the kind of table that splits data of the table by rows according to the sharding rules of sharding columns, and stores the data on different data nodes after splitting. Tables with large amount of data are suitable to be defined as sharding tables.

The syntax for creating a sharding table is as follows:

```

CREATE TABLE [IF NOT EXISTS] tbl_name SHARD BY {functionid | functionname} 'functionid |
functionname' USING COLUMN 'shardcolumnname' (....)

CREATE TABLE [IF NOT EXISTS] tbl_name SHARD BY {functiontype} 'functiontype' USING COLUMN
'shardcolumnname' on datanode 'datanodeid'(....)

```

Besides, the keywords after SHARD BY can also be placed after the table definition (the same is true for the vertical sharding table and the global table), for example:

```

CREATE TABLE [IF NOT EXISTS] tbl_name (table define...) SHARD BY {functiontype} 'functiontype'
USING COLUMN 'shardcolumnname' on datanode 'datanodeid'(....);

```

Sharding table creating syntax description:

SHARD BY {FUNCTIONID | FUNCTIONNAME | FUNCTIONTYPE} refers to the keyword of sharding function ID, sharding function name and partition function type.
' functionid_value | functionname_value | functiontype_value ' refers to the specific

value of sharding function ID, sharding function name and sharding function type.

USING COLUMN refers to the keyword of the sharding column.

'shardcolumnname' refers to the value of the sharding column.

ON DATANODE refers to the keyword of the data node.

'datanodeid' refers to the value of a specific data node. Multiple discontinuous values can be separated by commas. Multiple consecutive values can be specified in the form of intervals, such as 1, 3, 4, 5-10, 12-40.

Log in to the service port, use the LogicDB, enter Create Table statement, and execution succeeded. The management platform will show this table as defined status

```
mysql> use fun_zy
Database changed

mysql> CREATE TABLE match1_tb shard by functionname 'test_match1' using column 'aname' (id INT
UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, adnid INT DEFAULT NULL, aname
VARCHAR(32) DEFAULT "", adept VARCHAR(40), adate datetime DEFAULT NULL)ENGINE =INNODB;
Query OK, 0 rows affected (0.09 sec)
```

Table Configuration									
Configuration > Table Configuration									
	Table Name	Table Type	Select Sharding Type	Select Data Node	Select Table Structure	Table Structure	Whether to Enable the Global Unique Constraint	Operation	
	test	Sharding Table	ANAME	test_match1	dn_1 dn_2 dn_3 dn_4	Created	<input checked="" type="checkbox"/>		
	DB1018	Sharding Table	id	Auto Sharding_CRC32	dn_010	Not Created	<input checked="" type="checkbox"/>		
	test_zy	Sharding Table	id	Auto Sharding_CRC32	dn_1 dn_2	Not Created	<input checked="" type="checkbox"/>		
	zy_db	Sharding Table	FAS_SETTLE_DT	Auto Sharding_CRC32	dn_1 dn_2	Created	<input checked="" type="checkbox"/>		
	test	Sharding Table	id	test_route1	dn_1 dn_2 dn_3 dn_4	Not Created	<input checked="" type="checkbox"/>		
	test002	Sharding Table	ID	Auto Sharding_CRC32	dn_1 dn_2	Created	<input checked="" type="checkbox"/>		
	test001	Vertical Sharding Table			dn_1	Created	<input checked="" type="checkbox"/>		
	test	Sharing Table	provinceid	Auto Sharding_CRC32	dn_1 dn_2 dn_3	Not Created	<input checked="" type="checkbox"/>		
	DB_0919	Global Table			dn_14 dn_15	Not Created	<input checked="" type="checkbox"/>		
	DB_0919	Vertical Sharing Table			dn_5	Not Created	<input checked="" type="checkbox"/>		

For Create Table according to this Syntax Rule, pay attention to several points below:

- 'functionid | functionname | functiontype' is specific specified Sharding Function ID, Sharding Function name and Sharding Function type

- 'shardcolumnname' is specified sharding key
- 'datanodeid' is node ID, which could be separated by comma, and support specification in interval form, such as: '1,3,4,5-10,12-40', the node ID could log in to Distributed Transactional Database Management Platform page, and select “Configuration”->“ Node Management” to view, and could also log in to compute node [service port Use Command](#) to show hotdb datanodes; view:

```
mysql> show hotdb datanodes;

+-----+-----+-----+
| datanode_id | datanode_name | datanode_type |
+-----+-----+-----+
| 9          | dn_01       | 0           |
| 11         | dn_02       | 0           |
| 13         | dn_03       | 0           |
| 15         | dn_04       | 0           |
| 19         | dn_failover | 0           |
| 20         | dn_rmb_01   | 0           |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

- Function Type only support auto_crc32/auto_mod; if other types are used, it will prompt: ERROR:The fucntiontype can only be auto_crc32/auto_mod.

```
mysql> create table ft_match shard by functiontype 'match' using column 'id' on datanode '11,13'(id int(10)
Primary Key, a char(20) not null);

ERROR 10070 (HY000): The functiontype can only by auto_crc32/auto_mode.
```

- When using functionid | functionname to Create Table, if the specified function information associated function_type is auto_crc32/auto_mod, it needs to specify

on datanode 'datanodes', otherwise, it will prompt: The function must be specified datanodes. If being other types, specification is unnecessary.

```
mysql> create table mod_ft shard by functionid '15' using column 'id'(id int(10) Primary Key, a char(20) not null);
```

ERROR 10090 (HY000): The function must be specified datanodes.

```
mysql> create table testsa shard by functionid '3' using column 'id'(id int,a int);
```

Query OK, 0 rows affected, 1 warning (0.10 sec)

```
mysql> CREATE TABLE match_tb shard by functionname 'test_match1' using column 'aname' on datanode
'1,2'(id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, adnid INT DEFAULT NULL,
aname VARCHAR(32) DEFAULT "", adept VARCHAR(40), adate datetime DEFAULT NULL)ENGINE
=INNODB;
```

ERROR 10090 (HY000): This rule doesn't need to specify a datanodes;

Tables with similar table structure could use the same Sharding Function, and the following Syntax could be used to make direct citation of Sharding Function to create Sharding Table

```
CREATE TABLE [IF NOT EXISTS] tbl_name SHARD BY {ruleid | rulename} 'ruleid\rulename' [on datanode
'datanodes'] (.....
```

Log in to compute node [service port Use Command](#), show hotdb rules; and show hotdb functions; you could see the Sharding Function associated with its sharding Function

```
mysql> show hotdb rules;
```

rule_id	rule_name	rule_column	function_id
	auto_generated		

17	AUTO_GENERATE_3_ROUTE1_TB	A	1	1
----	---------------------------	---	---	---

|

+-----+-----+-----+

21 rows in set (0.01 sec)

mysql> show hotdb functions;

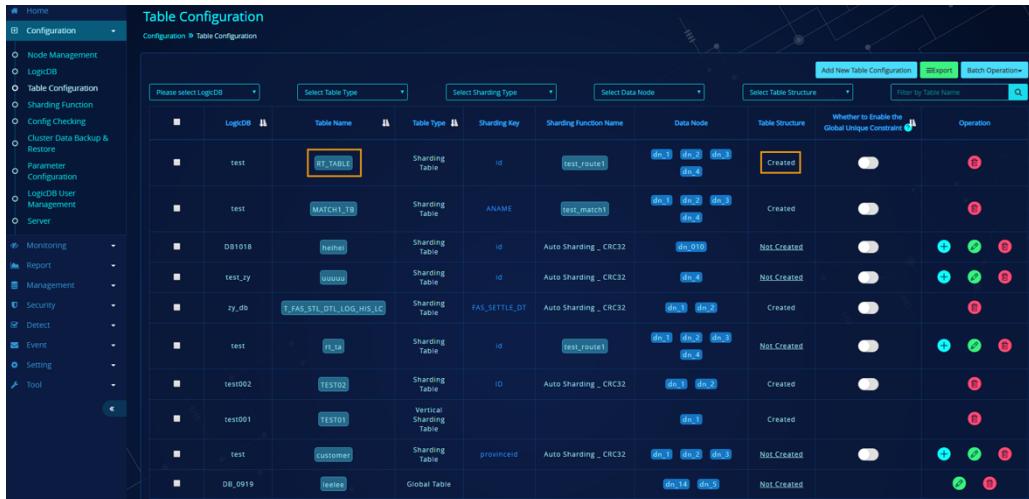
+-----+-----+-----+
function_id function_name function_type auto_generated
+-----+-----+-----+
1 test_route1 ROUTE 1
+-----+-----+-----+

13 rows in set (0.01 sec)

The user could use ruleid/rulename to Create Table directly

mysql> CREATE TABLE rt_table shard by ruleid '30'(id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, adnid INT DEFAULT NULL, aname VARCHAR(32) DEFAULT "", adept VARCHAR(40), adate datetime DEFAULT NULL)ENGINE =INNODB;

Query OK, 0 rows affected (0.07 sec)



The screenshot shows the 'Table Configuration' page of the HotDB management interface. On the left, there's a sidebar with navigation links like Home, Configuration, Monitoring, Report, Management, Security, Detect, Event, Setting, and Tool. The main area is titled 'Table Configuration' and contains a table with the following data:

	Please select Logic DB	Select Table Type	Select Sharding Type	Select Data Node	Select Table Structure	Filter by Table Name	Operation
1	LogiDB	RT_TABLE	Sharding Table	id	test_route1	dn_1 dn_2 dn_3 dn_4	Created
2	test	MATCH1_TB	Sharding Table	ANAME	test_match1	dn_1 dn_2 dn_3 dn_4	Created
3	DB1018	hehe1	Sharding Table	id	Auto Sharding_CRC32	dn_010	Not Created
4	test_zy	uuuuu	Sharding Table	id	Auto Sharding_CRC32	dn_4	Not Created
5	zy_db	FAS_SETTLE_DF	Sharding Table	FAS_SETTLE_DF	Auto Sharding_CRC32	dn_1 dn_2	Created
6	test	FE_TS	Sharding Table	id	test_route1	dn_1 dn_2 dn_3 dn_4	Not Created
7	test002	TEST02	Sharding Table	ID	Auto Sharding_CRC32	dn_1 dn_2	Created
8	test001	TEST01	Vertical Sharding Table			dn_3	Created
9	test	customer	Sharding Table	provinces	Auto Sharding_CRC32	dn_1 dn_2 dn_3	Not Created
10	DB_0919	lelele	Global Table			dn_1 dn_2 dn_3	Not Created

For Create Table according to this Syntax Rule, pay attention to several points below:

- If associated function_type with specified rule is auto_crc32/auto_mod, then it needs to specify on datanode 'datanodes', if being other types, there is no need to specify datanode.

```
mysql> show hotdb rules;
```

rule_id	rule_name	rule_column	function_id
	auto_generated		
17	AUTO_GENERATE_3_ROUTE1_TB	A	1

21 rows in set (0.01 sec)

```
mysql> show hotdb functions;
```

function_id	function_name	function_type	auto_generated
1	test_route1	ROUTE	1

13 rows in set (0.01 sec)

```
mysql> CREATE TABLE route2_rptb1 shard by rulename 'AUTO_GENERATE_3_ROUTE1_TB' on datanode
'9,11,13'(id int not null auto_increment Primary Key,a int(10),b decimal(5,2),c decimal(5,2),d date,e time(6),f
timestamp(6) DEFAULT CURRENT_TIMESTAMP(6),g datetime(6) DEFAULT CURRENT_TIMESTAMP(6),h
year,i char(20) character set utf8,j varchar(30) character set utf8mb4,k char(20) character set gbk,l text character set
latin1,m enum(",'null','1','2','3'),n set(",'null','1','2','3'));
```

ERROR 10090 (HY000): This rule doesn't need to specify a datanodes;

- When associated function_type with specified rule is auto_crc32/auto_mod, in case of inconsistence between specified data nodes and the parameter, then it will prompt:
ERROR: The total number of datanodes must be XXX (XXX is the column_value of the practical rule id associated function info)

```
mysql> CREATE TABLE auto_c shard by ruleid '63' on datanode '9'(id INT UNSIGNED NOT NULL
AUTO_INCREMENT PRIMARY KEY, adnid INT DEFAULT NULL, aname VARCHAR(32) DEFAULT
",adept VARCHAR(40), adate datetime DEFAULT NULL)ENGINE=INNODB;
```

ERROR 10090 (HY000): The total number of datanodes must be 2

```
mysql> CREATE TABLE auto_c shard by ruleid '63' on datanode '9,15'(id INT UNSIGNED NOT NULL
AUTO_INCREMENT PRIMARY KEY, adnid INT DEFAULT NULL, aname VARCHAR(32) DEFAULT
",adept VARCHAR(40), adate datetime DEFAULT NULL)ENGINE=INNODB;
```

Query OK, 0 rows affected (0.13 sec)

12.5.2.2. Vertical Sharding Table

The vertical sharding table is a global unique total data table with no sharding, and the total data of the vertical sharding table is only stored in one data node.

The syntax for creating a vertical sharding table is as follows:

```
CREATE TABLE [IF NOT EXISTS] tbl_name SHARD BY vertical on datanode 'datanodeid'(....)
```

Syntax description:

SHARD BY VERTICAL is a vertical sharding keyword. ON DATANODE 'datanodeid' can only specify one data node. If you do not specify any data node or specify multiple data nodes, an error will be reported.

```
mysql> CREATE TABLE tb_vertical shard by vertical on datanode'2'(id INT UNSIGNED NOT NULL
AUTO_INCREMENT PRIMARY KEY, admid INT DEFAULT NULL, aname VARCHAR(32) DEFAULT '',
adept VARCHAR(40), adate datetime DEFAULT NULL)ENGINE =INNODB;
```

Query OK, 0 rows affected(0.07 sec)

Table Configuration								
Please select LogicDB		Select Table Type	Select Sharding Type	Select Data Node	Select Table Structure	Filter by Table Name	Add New Table Configuration	Export
	LogicDB	Table Name	Table Type	Sharding Key	Sharding Function Name	Data Node	Table Structure	Whether to Enable the Global Unique Constraint
■	test	tb_vertical	Vertical Sharding Table			dn_1 dn_2 dn_3 dn_4	Created	
■	test	rt_table	Sharding Table	id	(test_route1)	dn_1 dn_2 dn_3 dn_4	Created	
■	test	rt_ts	Sharding Table	id	(test_route1)	dn_1 dn_2 dn_3 dn_4	Created	
■	test002	TEST02	Sharding Table	ID	Auto Sharding _ CRC32	dn_1 dn_2	Created	
■	test001	TEST01	Vertical Sharding Table			dn_1	Created	
■	test	customer	Sharding Table	provinceid	Auto Sharding _ CRC32	dn_1 dn_2 dn_3	Not Created	
■	DB_0919	leeee	Global Table			dn_14 dn_5	Not Created	
■	DB_0919	houhou	Vertical Sharding Table			dn_5	Not Created	

When datanode is not specified

```
mysql> CREATE TABLE tb1_vertical shard by vertical( id INT UNSIGNED NOT NULL AUTO_INCREMENT
PRIMARY KEY, admid INT DEFAULT NULL, aname VARCAHR(32) DEFAULT '', adept VARCHAR(40),
adate DATETIME DEFAULT NULL)ENGINE=INNODB;
```

ERROR 10090 (HY000): This table has to specify a datanodes.

When multiple nodes are specified

```
mysql> CREATE TABLE tb1_vertical shard by vertical on datanode'9,11,13'( id INT UNSIGNED NOT NULL
AUTO_INCREMENT PRIMARY KEY, admid INT DEFAULT NULL, aname VARCAHR(32) DEFAULT '',
adept VARCHAR(40), adate DATETIME DEFAULT NULL)ENGINE=INNODB;
```

ERROR 10090 (HY000): Can only specify one datanodes.

12.5.2.3. Global Table

Global table refers to the table stored in all data nodes under the LogicDB. The table

structure and data in all data nodes are completely consistent. Tables with small amount of data, infrequent DML, and frequent JOIN operations with other tables are suitable as global tables.

The syntax for creating a global table is as follows:

```
CREATE TABLE [IF NOT EXISTS] tbl_name SHARD BY global on datanode 'datanodeid'(....)
```

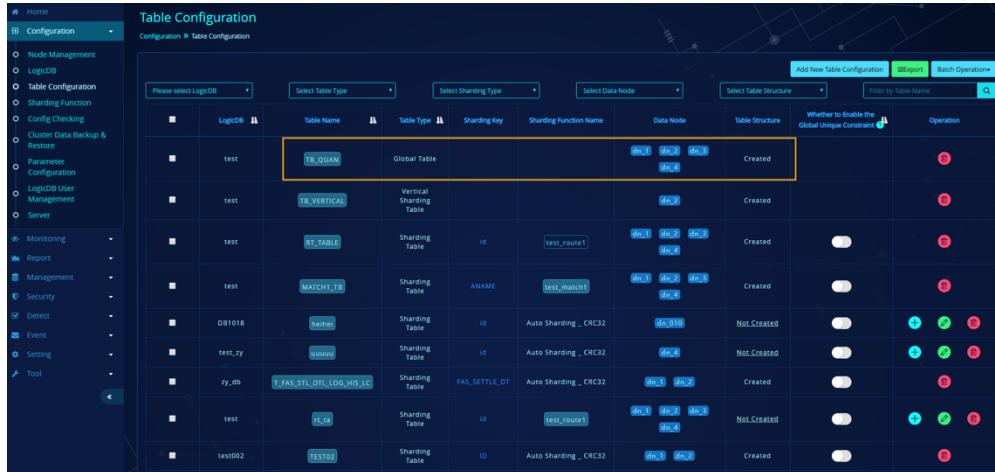
Syntax description:

SHARD BY GLOBAL is the keyword of global table.

[ON DATANODE 'datanodeid'] is the syntax for specifying a data node. When the compute node version is 2.5.6 or above, if it is not specified, the table will be created by default based on the default sharding node of the LogicDB + the union set of all the associated nodes of the tables in the logicDB; if it is specified, all data nodes must be included, errors will be reported if only some data nodes are specified.

```
mysql> CREATE TABLE tb_quan shard by global(id INT UNSIGNED NOT NULL AUTO_INCREMENT
PRIMARY KEY, adnid INT DEFAULT NULL, aname VARCHAR(32) DEFAULT "", adept VARCHAR(40),
adate datetime DEFAULT NULL)ENGINE =INNODB;
```

Query OK, 0 rows affected (0.07 sec)



The screenshot shows the 'Table Configuration' page of the HotDB management console. On the left, there's a navigation sidebar with various options like Home, Configuration, Node Management, LogDB, Table Configuration, Sharding Function, Config Checking, Cluster Data Backup & Restore, Monitoring, Report, Management, Security, Detect, Event, Setting, and Tool. The 'Table Configuration' option is selected.

The main area is titled 'Table Configuration' and shows a table with the following columns: Please select LogDB, Table Name, Table Type, Select Sharding Type, Select Data Node, Select Table Structure, Whether to Enable the Global Unique Constraint, and Operation.

The table contains several entries:

- tb_quan**: Global Table, Vertical Sharding Table, Data Node: dn_1, dn_2, dn_3, dn_4, Status: Created.
- tb_vertical**: Vertical Sharding Table, Data Node: dn_2, Status: Created.
- rt_table**: Sharding Table, id, ANAME: test_route1, Data Node: dn_1, dn_2, dn_3, dn_4, Status: Created.
- match_tb**: Sharding Table, id, ANAME: test_match, Data Node: dn_1, dn_2, dn_3, dn_4, Status: Created.
- DB1018**: Sharding Table, id, Auto Sharding_, CRC32, Data Node: dn_310, Status: Not Created.
- test_zx**: Sharding Table, id, Auto Sharding_, CRC32, Data Node: dn_4, Status: Not Created.
- zy_db**: Sharding Table, FAS_SETTLE_DT, Auto Sharding_, CRC32, Data Node: dn_1, dn_2, Status: Created.
- ft_tb**: Sharding Table, id, test_route1, Data Node: dn_1, dn_2, dn_3, dn_4, Status: Not Created.
- test02**: Sharding Table, ID, Auto Sharding_, CRC32, Data Node: dn_1, dn_2, Status: Created.

global in syntax rules means creating a global table. The 'datanodeid' is the node ID, which can be separated by English commas and can be specified in the form of interval,

such as 1, 3, 4, 5-10, 12-40. The nodes of the global table with sharding functions created by global should include all nodes under the LogicDB.

- If there is no sharding node or defined table under the LogicDB, you need to specify the nodes of the global table distribution when creating the global table using special syntax:

```
mysql> CREATE TABLE tb2_quan shard by global(id int not null auto_increment primary key,a int(10),b decimal(5,2),c decimal(5,2),d date,e time(6),f timestamp(6) DEFAULT CURRENT_TIMESTAMP(6),g datetime(6) DEFAULT CURRENT_TIMESTAMP(6),h year,i char(20) null,j varchar(30),k blob,l text, m enum("','null','1','2','3'),n set("','null','1','2','3"));

ERROR 10090 (HY000): This table has to specify a datanodes.
```

If a defined table already exists in the LogicDB, whenever you do not need to specify a node or need to specify a node, you need to specify the maximum number of non-duplicate nodes (that is, the union of all the selected nodes of all the table) that are included in the LogicDB. Otherwise, it will warn you of a table-creating error if some data nodes are specified:

```
mysql> CREATE TABLE tb1_quan shard by global on datanodes'9,11'(id int not null auto_increment primary key,a int(10),b decimal(5,2),c decimal(5,2),d date,e time(6),f timestamp(6) DEFAULT CURRENT_TIMESTAMP(6),g datetime(6) DEFAULT CURRENT_TIMESTAMP(6),h year,i char(20) null,j varchar(30),k blob,l text, m enum("','null','1','2','3'),n set("','null','1','2','3"));

ERROR 10090 (HY000): The specified datanodes must cover all datanodes of the LogicDB.
```

12.5.2.4. Related command of create table using existing sharding function

The command introduced in this section could be executed by service port and management port of the compute node.

1. show hotdb datanodes - show the existing available nodes

This command is used for View of hotdb_datanodes table in configDB, Syntax:

```
mysql> show hotdb datanodes [LIKE 'pattern' | WHERE expr];
```

The command contains parameters and its description:

Parameter	Description	Type
pattern	Optional, Fuzzy Query Expression, Match rule_name Field	STRING
expr	Optional, Fuzzy Query Expression, Match Specified Field	STRING

The result contains Field and its Description:

Column name	Description	Value type/Range
datanode_id	Node ID	INTEGER
datanode_name	Node name	STRING
datanode_type	0: Master/Slave; 1: MGR	INTEGER

For example:

```
mysql> show hotdb datanodes;
+-----+-----+-----+
| datanode_id | datanode_name | datanode_type |
+-----+-----+-----+
```

1	dn_01	0	
2	dn_02	0	
4	dn_04	0	
101	dn_101	0	
127	dn_03	0	
186	dn_199	0	
203	dn_19	0	
+-----+-----+-----+			

For another example:

mysql> show hotdb datanodes like 'dn_0%';
+-----+-----+-----+
datanode_id datanode_name datanode_type
+-----+-----+-----+
1 dn_01 0
2 dn_02 0
4 dn_04 0
127 dn_03 0
+-----+-----+-----+

2. show hotdb functions – show the existing available Sharding Function

This command is used for View of hotdb_function table in configDB, Syntax:

mysql> show hotdb functions;

The command contains parameters and its description:

Parameter	Description	Type
pattern	Optional, Fuzzy Query Expression, Match function_name Field	STRING
expr	Optional, Fuzzy Query Expression, Match function_name Field	STRING

The result contains Field and its Description:

Column name	Description	Value type/Range
function_id	Sharding Function ID	INTEGER
function_name	Sharding Function name	STRING
function_type	Sharding Type	STRING
auto_generated	Auto-generated configuration of HotDB or not (1: Auto Generated, Other: Non-auto Generated)	INTEGER

For example:

```
mysql> show hotdb functions;

+-----+-----+-----+-----+
| function_id | function_name | function_type | auto_generated |
+-----+-----+-----+-----+
| 40          | AUTO_CRC32_8   | AUTO_CRC32    | 0              |
| 41          | th_fun_range    | RANGE         | 0              |
| 42          | AUTO_MOD_5      | AUTO_MOD     | 0              |
...More are omitted...
```

43	43_RANGE	RANGE	0	
44	AUTO_CRC32_15	AUTO_CRC32	0	
45	AUTO_CRC32_5	AUTO_CRC32	0	
46	yds_RANGE	RANGE	0	
47	AUTO_CRC32_11	AUTO_CRC32	0	
+-----+-----+-----+-----+				

For another example:

```
mysql> show hotdb functions like '%range%';
+-----+-----+-----+-----+
| function_id | function_name | function_type | auto_generated |
+-----+-----+-----+-----+
| 41          | th_fun_range | RANGE        | 0             |
| 43          | 43_RANGE     | RANGE        | 0             |
| 46          | yds_RANGE    | RANGE        | 0             |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> show hotdb functions where function_name like '%range%';
+-----+-----+-----+-----+
| function_id | function_name | function_type | auto_generated |
+-----+-----+-----+-----+
| 41          | th_fun_range | RANGE        | 0             |
| 43          | 43_RANGE     | RANGE        | 0             |
| 46          | yds_RANGE    | RANGE        | 0             |
+-----+-----+-----+-----+
```

```
3 rows in set (0.00 sec)
```

3. show hotdb function infos - show the existing available Sharding Function information

This command is used for View of hotdb_function_info table in configDB, Syntax:

```
mysql> show hotdb function infos [WHERE expr];
```

The command contains parameters and its description:

Parameter	Description	Type
expr	Optional, Fuzzy Query Expression, Match Specified Field	STRING

The result contains Field and its Description:

Column name	Description	Value type/Range
function_id	Sharding Function ID	INTEGER
column_value	Sharding Key value	STRING
datanode_id	Data node id	INTEGER

For example:

```
mysql> show hotdb function infos;
```

```
+-----+-----+-----+
```

function_id column_value datanode_id
2 4 0
3 1 0
4 2 0
31 " 4
31 1 1
31 2 2
31 null 127
33 0:1 1
33 10:10 191
33 11:11 186
33 12 0
33 2:3 2

...More are omitted...

For another example:

mysql> show hotdb function infos where function_id=38;
+-----+-----+-----+
function_id column_value datanode_id
+-----+-----+-----+
38 10:12 1
38 1:2 1
38 20 0
38 4:8 1
+-----+-----+-----+

```
4 rows in set (0.00 sec)
```

4. show hotdb rules – show the existing available Sharding Function

This command is used for View of hotdb_rule table in configDB, Syntax:

```
mysql> show hotdb rules [LIKE 'pattern' | WHERE expr];
```

The command contains parameters and its description:

Parameter	Description	Type
pattern	Optional, Fuzzy Query Expression, Match rule_name Field	STRING
expr	Optional, Fuzzy Query Expression, Match rule_name Field	STRING

The result contains Field and its Description:

Column name	Description	Value type/Range
rule_id	Sharding Function ID	INTEGER
rule_name	Sharding Function name	STRING
rule_column	Sharding Key Name	STRING
function_id	Sharding Type ID	INTEGER
auto_generated	Auto-generated configuration of HotDB or not (1: Auto Generated, Other: Non-auto Generated)	INTEGER

For example:

```
mysql> show hotdb rules;
```

rule_id	rule_name	rule_column	function_id	auto_generated
21	AUTO_GENERATE_3_JOIN_A_JWY	ID	1	
22	hotdb-cloud_0374c02e-58a7-4263-9b80-9c5b46fb42af	id	5	0
25	hotdb-cloud_f3979d19-93cb-4925-8dee-e4fbf8803c7c	id	5	0
32	hotdb-cloud_6ccd2f69-cf53-4e81-ab3d-61345134fb7a	id	5	0
33	hotdb-cloud_b5bc16e6-3481-40ed-83ff-e81d488e47a5	ID	4	0
...More are omitted...				

For another example:

```
mysql> show hotdb rules like '%auto%';
```

rule_id	rule_name	rule_column	function_id	auto_generated
21	AUTO_GENERATE_3_JOIN_A_JWY	ID	1	1
50	AUTO_GENERATE_9_FT_ADDR	ID	26	1
64	AUTO_GENERATE_23_S03	A	1	1
65	AUTO_GENERATE_23_S04	B	1	1
...More are omitted...				

```
mysql> show hotdb rules where rule_name like '%auto%';
```

rule_id	rule_name	rule_column	function_id	auto_generated	
21	AUTO_GENERATE_3_JOIN_A_JWY	ID	1	1	
50	AUTO_GENERATE_9_FT_ADDR	ID	26	1	
64	AUTO_GENERATE_23_S03	A	1	1	
65	AUTO_GENERATE_23_S04	B	1	1	
...More are omitted...					

12.5.3. Guarantee that the data of dropped table is not lost for a period of time

Considering the high risk of DROP TABLE, TRUNCATE TABLE, DELETE TABLE without WHERE condition statements in practical production or online environment, HotDB Server supports Guarantee that the data of dropped table is not lost for a period of time. For DELETE TABLE without WHERE condition statement, data can be retained only in auto commit, and operations within a transaction cannot be retained temporarily.

You can either modify the dropTableRetentionTime parameter in server.xml or Add the parameter “dropTableRetentionTime(h)” in Compute Node Parameter Configuration under the management platform configuration menu.

```
<property name="dropTableRetentionTime">0</property><!—the dropTableRetentionTime is 0 by default,  
meaning not retained-->
```

dropTableRetentionTime parameter is 0 by default; when dropTableRetentionTime is 0, it means that the DROP Table is not retained, DROP TABLE is to delete table; when

dropTableRetentionTime is greater than 0, computed by hour, and retain the DROP Table for set time, and the table will be automatically deleted if exceeding the set time.

At present, compute node does not provide direct command of restoring the deleted table, but the user could restore the deleted table in data source by means of RENAME. Since operation is directly made on data source, therefore, this operation has risk, please use carefully. The risk includes: the data route after Restore may be different from the practical route, the original foreign key and trigger are deleted, Parent/Child Table relation does not exist, etc., and for detailed risks, please refer to [Notices for Restoration](#).

When this function is enabled, this document takes dropTableRetentionTime=24 for instance, that is to retain the DROP Table, and delete the table 24h later. If to restore the table deleted, firstly, Query hotdb_dropped_table_log in the compute node configDB, to view the mapping relation of the table deleted. For example:

```
mysql> select * from hotdb_dropped_table_log;
+-----+-----+-----+-----+
| table_name | datanodes | drop_time | renamed_table_name |
+-----+-----+-----+-----+
| TABLE25 | 11 | 2019-03-26 17:18:07 | HOTDB_TEMP_33_20190326171807 |
| TABLE30 | 11 | 2019-03-26 17:18:13 | HOTDB_TEMP_34_20190326171812 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Including, renamed_table_name is temporary table of data of the deleted table, and the data could be restored through RENAME this temporary table.

```
RENAME TABLE tbl_name TO new_tbl_name
```

The restoration operation could either RENAME temporary table respectively under corresponding data source, or RENAME temporary table directly under the compute

node via HINT statement. For operation description and notices for using HINT statement, please refer to [HINT](#) Chapter in this document

12.5.3.1. Notices for Restoration

Pay attention to the key points as following when making restoration:

- The table created after management platform configuration, could be RENAME as the original table name directly. For the table created via Auto Create Table function, Table Configuration is not retained at the time of DROP TABLE, therefore, it can't be RENAME as the original table name directly.
- The table created after restoring the Auto Create Table, could be RENAME as this table name after Add Configuration on the management platform. That is, the restoration operation allows to restore the deleted name to any table configured but not defined on the management platform via RENAME the table name.
- In case of change in the Sharding Function cited by the deleted table, or configuration of different Sharding Function, after restoration, the data route will be consistent with that before Delete, that is, the practical data route will not match with the configured route. It's recommended configuring restoration according to the original Sharding Function, in order to prevent the problem of data loss or inconsistency.
- If there is foreign key or trigger on the deleted table, then at the time of DROP TABLE, foreign key and trigger will be deleted in the temporary table. Since foreign key is deleted, after restoration, the Parent/Child Table no longer has Parent/Child Table relation.
- The restoration operation of RENAME temporary table could take effect only after Reload. The Reload supported by the compute node at present could take effect only under the condition of change in configDB, therefore, if there is no other change in configDB except RENAME operation, the Reload will take effect only

upon manual change in configDB, that is, the restored table could be viewed.

13. INFORMATION_SCHEMA

INFORMATION_SCHEMA database provides information and data of the existing compute node, for example database name, Table Name, data type of a certain Column, etc.

This chapter lists the tables and its special processing contents in INFORMATION_SCHEMA supported by the compute node, as follows:

Table name	Special processing
character_sets	Only return the Character Set and Collation Set data supported by the compute node
collations	Only return the Character Set and Collation Set data supported by the compute node
collation_character_set_applicability	Only return the Character Set and Collation Set data supported by the compute node
columns	If a table is distributed on multiple nodes, the compute node will return columns information on a selected node. The return result does not include information of system database ('mysql','information_schema','performance_schema','sys')
column_privileges	Return Null Set
engines	Only return innodb
events	Return Null Set
files	Return Null Set
global_status	The same as result of show global status
global_variables	The same as result of show global variables
innodb_buffer_page	Return Null Set

innodb_buffer_page_lru	Return Null Set
innodb_buffer_pool_stats	Return Null Set
innodb_cmp	Return Null Set
innodb_cmpmem	Return Null Set
innodb_cmpmem_reset	Return Null Set
innodb_cmp_per_index	Return Null Set
innodb_cmp_per_index_reset	Return Null Set
innodb_cmp_reset	Return Null Set
innodb_ft_being_deleted	Return Null Set
innodb_ft_config	Return Null Set
innodb_ft_default_stopword	Return Null Set
innodb_ft_deleted	Return Null Set
innodb_ft_index_cache	Return Null Set
innodb_ft_index_table	Return Null Set
innodb_locks	Return Null Set
innodb_lock_waits	Return Null Set
innodb_metrics	Return Null Set
innodb_sys_columns	Return Null Set
innodb_sys_datafiles	Return Null Set
innodb_sys_fields	Return Null Set
innodb_sys_foreign	Return Null Set

innodb_sys_foreign_cols	Return Null Set
innodb_sys_indexes	Return Null Set
innodb_sys_tables	Return Null Set
innodb_sys_tablespaces	Return Null Set
innodb_sys_tablestats	Return Null Set
innodb_trx	Return Null Set
key_column_usage	If a table is distributed on multiple nodes, the compute node will return columns information on a selected node. The return result does not include information of system database ('mysql','information_schema','performance_schema','sys')
optimizer_trace	Return Null Set
parameters	Return Null Set
partitions	If a table is distributed on multiple nodes, the compute node will return columns information on a selected node. The return result does not include information of system database ('mysql','information_schema','performance_schema','sys'). Sequencing and grouped queries of tables is supported.
plugins	Return Null Set
processlist	The returned result and server command show that processlist is consistent
profiling	Return Null Set

referential_constraints	If a table is distributed on multiple nodes, the compute node will return columns information on a selected node. The return result does not include information of system database ('mysql','information_schema','performance_schema','sys')
routines	Return Null Set
schemata	Return LogicDB related information
schema_privileges	Return Null Set
session_status	The same as the result of show session status
session_variables	The same as the result of show session variables
statistics	The compute node will show after making de-repetition or sum processing of statistics information of all nodes under the LogicDB. The return result does not include information of the system database ('mysql','information_schema','performance_schema','sys')
tables	If a table is distributed on multiple nodes, the compute node will return columns information on a selected node. The return result does not include information of system database ('mysql','information_schema','performance_schema','sys')
tablespaces	Return Null Set
table_constraints	If a table is distributed on multiple nodes, the compute node will return columns information on a selected node. The return result does not include information of system database

	('mysql','information_schema','performance_schema','sys')
table_privileges	Return Null Set
triggers	Return Null Set
user_privileges	Return Null Set
views	Return Null Set

In order to be compatible with the data source above MySQL 8.0, conduct the following special processing for the New contents of MySQL8.0:

Table Name	Special processing
check_constraints	If a table is distributed on multiple nodes, the compute node will return columns information on a selected node. The return result does not include information of system database ('mysql','information_schema','performance_schema','sys')
column_statistics	If a table is distributed on multiple nodes, the compute node will return columns information on a selected node. The return result does not include information of system database ('mysql','information_schema','performance_schema','sys')
keywords	Return Null Set
resource_groups	Return Null Set
st_geometry_columns	If a table is distributed on multiple nodes, the compute node will return columns information on a selected node. The return result does not include information of

	system database ('mysql','information_schema','performance_schema','sys')
st_spatial_reference_systems	No special processing
st_units_of_measure	No special processing
view_table_usage	Return Null Set
view_routine_usage	Return Null Set

14. Instruction on use of compute node parameters

During the use process, the compute node has maintained many system configuration parameters, and this document describes how to use these parameters and the influence on function. Every parameter has a default value, which can be either modified in server.xml Config File when service starts, or modified on the Parameter Configuration page after Log in the management platform. Most of these parameters could use Reload (reload @@config) to operate dynamic modification during running, without the need to stop and restart the service. Some parameters could also be modified in Expression by means of set.

14.1.1. adaptiveProcessor

Description of parameter:

Property	Value
Parameter value	adaptiveProcessor
Visible or not	Hidden
Description of parameters	Control whether the startup service is Automatic Adaptation or not.
Default value	true
Whether Reload is valid or not	No
Min Compatible Version	2.4.5

Parameter Setting:

adaptiveProcessor parameter in Server.xml is configured as follow:

```
<property name="adaptiveProcessor">true</property><!-- Control whether the startup service is Automatic Adaptation or not, true means Automatic Adaptation, false means non-Automatic Adaptation, the parameter is true by default, and is hidden -->
```

adaptiveProcessor parameter is true by default, that is the Automatic Adaptation startup service is enabled, and all values including [processor](#), [processorExecutor](#) and [timerExecutor](#) will come into Automatic Adaptation. If being false, the Automatic Adaptation startup service is disabled

Role of parameter:

After the Automatic Adaptation startup service is enabled, the compute node will set parameters according to the current server configuration and Automatic Adaptation Rule, that is to modify the following parameter value in server.xml, and after the startup service, the modification can't take effect, and the parameter value will still be set according to Adaptation Rule.

```
<property name="processors">16</property><!--number of processors-->

<property name="processorExecutor">4</property><!--number of threads of various processorExecutor -->

<property name="timerExecutor">4</property><!--number of threads of timerExecutor -->
```

Log in to 3325 port, execute show @@@threadpool; command, view the current processor, processorExecutor and timerExecutor value. for example:

```
mysql> show @@@threadpool;

+-----+-----+-----+-----+-----+
| name          | pool_size | active_count | task_queue_size | completed_task | total_task |
+-----+-----+-----+-----+-----+
| TimerExecutor |        4 |          0 |         15 |      50376807 |    50376822 |
| $NIOExecutor-0- |     4 |          0 |          0 |       99254 |     99254 |
| $NIOExecutor-1- |     4 |          1 |          0 |       81195 |     81196 |
| $NIOExecutor-2- |     4 |          2 |          0 |      140921 |    140923 |
```

\$NIOExecutor-3-	4	1	0	48218	48219
\$NIOExecutor-4-	4	0	0	39073	39073
\$NIOExecutor-5-	4	0	0	31656	31656
\$NIOExecutor-6-	4	0	0	167007	167007
\$NIOExecutor-7-	4	1	0	27221	27222
+-----+-----+-----+-----+-----+					
9 rows in set (0.00 sec)					

\$NIOExecutor has 0-7, meaning that the current processor=8, corresponding pool_size = 4, meaning that the processorExecutor=4, corresponding pool_size of TimerExecutor = 4, meaning that the timerExecutor=4.

The following command could be used to view the number of the current logic CPU

```
cat /proc/cpuinfo| grep "processor"| wc -l
```

Notice: When starting Initialize at the very beginning, the compute node does not generate all threads, instead, it generates the threads actually used, therefore, executing show @@threadpool; command may show the result as follow:

mysql> show @@threadpool;					
name	pool_size	active_count	task_queue_size	completed_task	total_task
TimerExecutor	4	0	14	73720	73734
\$NIOExecutor-0-	1	0	0	1	1
\$NIOExecutor-1-	1	0	0	1	1
\$NIOExecutor-2-	1	0	0	1	1
\$NIOExecutor-3-	1	0	0	1	1
\$NIOExecutor-4-	2	0	0	2	2

```

| $NIOExecutor-5- |      5 |      0 |      0 |      5 |      5 |
| $NIOExecutor-6- |      1 |      0 |      0 |      1 |      1 |
| $NIOExecutor-7- |      1 |      1 |      0 |      1 |      1 |
+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

```

Only when the compute node is under pressure could it reach the Automatic Adaptation value, that's to say, the Automatic Adaptation value is Max Value.

14.1.2. allowRCWithoutReadConsistentInXA

Description of parameter:

Property	Value
Parameter value	allowRCWithoutReadConsistentInXA
Visible or not	No
Description of parameters	RC isolation level that does not guarantee strong read-write consistency is allowed in XA mode
Default value	0
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.9

Parameter Setting:

allowRCWithoutReadConsistentInXA in server.xml is configured as follows:

```
<property name="allowRCWithoutReadConsistentInXA">1</property><!-- whether allow RC isolation level  
that does not guarantee strong read-write consistency in XA mode -->
```

Role of parameter:

By default, only two isolation levels, REPEATABLE READ and SERIALIZABLE, can be selected in XA mode.

When the compute node version is below 2.5.3.1 and the parameter allowRCWithoutReadConsistentInXA is set to 0, the isolation level READ COMMITTED is not allowed in XA mode; when it is set to 1, READ COMMITTED is allowed in XA mode, and the session isolation level can be modified to be READ COMMITTED. It should be noted that the current isolation level READ COMMITTED is essentially between READ COMMITTED and READ UNCOMMITTED, which does not guarantee the strong read-write consistency.

When the compute node version is 2.5.3.1 and above, if the parameter allowRCWithoutReadConsistentInXA is set to 0, the isolation level READ COMMITTED which guarantees the read-write consistency of transactions is allowed in XA mode, which is the same as MySQL (it should be noted that when the original SQL involves cross-node queries and is split into multiple statements for multiple queries, the latest transactions will be read continuously, so single node queries should be used as much as possible in this mode);

When the parameter allowRCWithoutReadConsistentInXA is set to 1, the isolation level READ COMMITTED is also allowed in XA mode, which however does not guarantee strong read-write consistency. The isolation level READ COMMITTED is essentially between READ COMMITTED and READ UNCOMMITTED, whose performance is better than when it is set to 0, and there will be a log prompt when starting and synchronously loading, as follows:

2020-03-12 15:36:03.719 [WARN] [main] cn.hotpu.hotdb.a(519) -- Note that the READ COMMITTED isolation level in XA mode is essentially between READ COMMITTED and READ UNCOMMITTED at this time, which does not guarantee strong consistency of reading and writing.

14.1.3. autoIncrement

Description of parameter:

Property	Value
Parameter value	autoIncrement
Visible or not	Yes
Description of parameters	Management platform v.2.5.4 below: adopt Global Auto-Incremental serial number or not Management platform v.2.5.4 and above: Global Auto-Incremental serial number mode
Default value	1
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Role of parameter:

To control whether Auto-Incremental sequence of the table could be Global Auto-Incremental under distributed environment. For details, please refer to the chapter Global AUTO_INCREMENT. In version 2.5.4 and above, the parameter setting range is: 0, 1, 2;

In version 2.5.3 and below, only true or false can be set. Setting as true is equivalent to setting as 1; setting as false is equivalent to setting as 0.

14.1.4. badConnAfterContinueGet

Description of parameter:

Property	Value
Parameter value	badConnAfterContinueGet
Visible or not	No
Description of parameters	Continue to obtain connection or not
Default value	true
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="badConnAfterContinueGet">true</property><!-- Continue to obtain connection or not: true means continue to obtain connection, false means return null and not continue to obtain connection, the outer layer shall create new connection or other operation -->
```

Role of parameter:

After compute node obtaining connection from the connection pool and having conducted connection validity check and obtained an invalid connection, then when this parameter is true, the connection pool will continue to obtain available connection; when it is false, the connection pool will return null, and the outer code logic will continue the processing.

14.1.5. badConnAfterFastCheckAllIdle

Description of parameter:

Property	Value
Parameter value	badConnAfterFastCheckAllIdle
Visible or not	No
Description of parameters	When broken back-end connection is obtained, whether to check all idle connections rapidly or not
Default value	true
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="badConnAfterFastCheckAllIdle">true</property><!-- When broken back-end connection is obtained, whether to check all idle connections rapidly or not, true means check, false means not check, true by default-->
```

Role of parameter:

When broken back-end connection is obtained, the connection pool of the compute node will check all idle connections rapidly

14.1.6. bakUrl & bakUsername & bakPassword

Description of parameter:

Property	Value
Parameter value	bakUrl
Visible or not	Yes
Description of parameters	Slave configDB address
Default value	jdbc:mysql://127.0.0.1:3306/hotdb_config
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.4

Property	Value
Parameter value	bakUsername
Visible or not	Yes
Description of parameters	Slave configDB username
Default value	hotdb_config
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.4

Property	Value
Parameter value	bakPassword
Visible or not	Yes
Description of parameters	Slave configDB password
Default value	hotdb_config

Property	Value
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.4

Role of parameter:

bakUrl and bakUsername and bakPassword are supporting parameters, and are used for the ConfigDB High Availability function. If to use ConfigDB High Availability, it shall set information of corresponding Slave configDB and guarantee normal replication relation of the master/slave configDB instance, and shall be Master/Slave mutually, it will switch to Slave configDB automatically in case of failure with the Master configDB.

In case of latency during the switch process, it will wait for the Slave configDB to catch up with replication, and then switch succeeded and it will provide service

```
<property name="url">jdbc:mysql://192.168.210.30:3306/hotdb_config</property><!-- Master configDB address, real IP address of the configDB service shall be specified -->

<property name="username">hotdb_config</property><!-- Master configDB username -->

<property name="password">hotdb_config</property><!-- Master configDB password -->
```

If not requiring Master/Slave configDB, then the configuration here shall be consistent with the master configuration information or Slave configDB shall not be configured.

```
property name="bakUrl">jdbc:mysql://192.168.210.31:3306/hotdb_config</property><!-- Slave configDB address, real IP address of the configDB service shall be specified -->

<property name="bakUsername">hotdb_config</property><!-- Slave configDB username -->

<property name="bakPassword">hotdb_config</property><!-- Slave configDB password -->
```

In case of configDB switch due to failure with the Active Master, to restore the Active Master to normal, it needs to update v value in the row hotdb_master_config_status named k in houdb_config_info table of the configDB from 0 to 1, and execute reload

`@@config` on management port, and only in this way could the Master configDB be reused (for operating method of enabling Master configDB on the management platform, please refer to *Distributed Transactional Database HotDB Server [Management Platform] Function Manual*).

```
mysql> select * from hotdb_config_info\G
*****
1.row*****
k: hotdb_master_config_status
v: 1
description: NULL
```

14.1.7. checkConnLastUsedTime

Description of parameter:

Property	Value
Parameter value	checkConnLastUsedTime
Visible or not	No
Description of parameters	Max allowed interval time of last use by the back-end connection. If exceeded, it will check whether this connection is valid or not, unit: ms
Default value	3000
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="checkConnLastUsedTime">false</property><!-- Max allowed interval time of last use by the
back-end connection. If exceeded, it will check whether this connection is valid or not, unit: ms -->
```

Role of parameter:

Under the condition of still no Read/Write of bk_last_write_time and the timeout, the compute node will check the connection connectivity first when obtaining connection from the connection pool, in order to guarantee that the connection obtained is available

```
mysql> show @@session;

+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| id      | running | trx_started | trx_time | trx_query | bk_count | bk_dnid | bk_dsid | bk_id | bk_mysqlid | bk_state |
| bk_closed | bk_autocommit | bk_host | bk_port | bk_db | bk_query | bk_last_read_time | bk_last_write_time |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| 60615 | FALSE    | NULL        | NULL      | NULL      | 0          | NULL      | NULL      | NULL      | NULL
| NULL      | NULL      | NULL        | NULL      | NULL      | NULL      | NULL      | NULL      | NULL      | NULL
| NULL      | NULL      |           |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

14.1.8. CheckConnValid

Description of parameter:

Property	Value
Parameter value	CheckConnValid

Property	Value
Visible or not	No
Description of parameters	Whether to check whether the back-end connection is valid or not
Default value	true
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Parameter Setting:

Add configuration of CheckConnValid manually in server.xml

```
<property name="CheckConnValid">true</property>
```

Role of parameter:

When obtaining connection from the connection pool, check availability of the connection, and in case of Unavailable connection, the connection will be disabled and kicked out from the connection pool.

14.1.9. checkConnValidTimeout

Description of parameter:

Property	Value
Parameter value	checkConnValidTimeout
Visible or not	No

Property	Value
Description of parameters	At the time of checking validity of back-end connection, max timeout, unit: ms
Default value	500
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="checkConnValidTimeout">500</property><!-- At the time of checking validity of back-end connection, max timeout, unit: ms -->
```

Role of parameter:

At the time of checking validity of back-end connection, when the checking time exceeds “back-end connection timeout”, the connection shall be judged invalid, and when the back-end connection is checked as timeout connection, then the connection will be kicked out from the connection pool.

14.1.10. checkMySQLParamInterval

Description of parameter:

Property	Value
Parameter value	checkMySQLParamInterval
Visible or not	No

Property	Value
Description of parameters	Interval time of checking MySQL Parameter Setting (Unit: ms)
Default value	600000
Min value	1000
Max value	86400000
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="checkMySQLParamInterval">60000</property><!--interval time of checking whether MySQL
Parameter Setting is reasonable or not (Unit: ms) -->
```

Role of parameter:

Check whether interval time of MySQL Parameter Setting is reasonable or not. The checking parameters include: completion_type, nnodb_rollback_on_timeout, div_precision_increment, autocommit, read_only, tx_isolation, MAX_ALLOWED_PACKET.

14.1.11. checkUpdate

Description of parameter:

Property	Value
Parameter value	checkUpdate
Visible or not	No

Property	Value
Description of parameters	Whether to intercept update operation of the sharding key or not
Default value	true
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Role of parameter:

Control whether allow modifying the sharding key or not. Unless under special circumstances, it's not recommended modifying this parameter, otherwise, it may result in inconsistency between the data route and the configured Sharding Function, thus influencing the Query result.

If set true, after updating the sharding key, there will be prompt as follow:

```
mysql> update ss set id=13 where a='aa';
ERROR 1064 (HY000): sharding column's value cannot be changed.
```

If set false, updating the sharding key could make execution succeeded.

```
mysql> update ss set id=13 where a='aa';
Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> select * from ss where a='aa';

+----+----+
| id | a   |
+----+----+
```

```
| 13 | aa |
+----+----+
1 row in set (0.00 sec)
```

14.1.12. clientFoundRows

Description of parameter:

Property	Value
Parameter value	clientFoundRows
Visible or not	No
Description of parameters	Use found rows to replace the affected rows in OK package
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.9 (abandoned in v.2.5.5)

Parameter Setting:

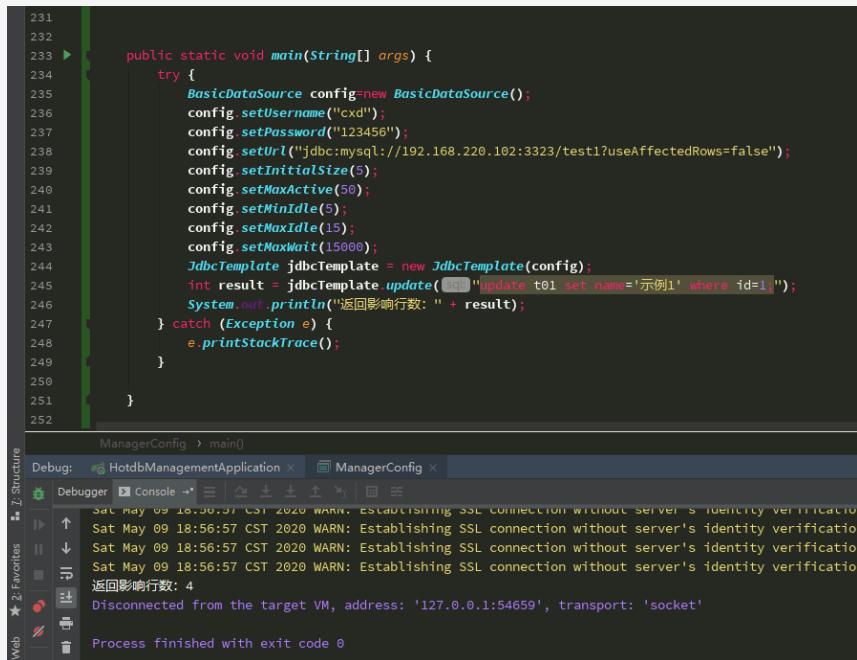
clientFoundRows parameter in Server.xml is configured as follow:

```
<property name=" clientFoundRows ">false</property><!--用 found rows 代替 OK 包中的 affected rows -->
```

Role of parameter:

When the version is 2.5.5 below, this parameter is used to judge the execution status of SQL statements when performing SQL statements. If the client connection uses the parameter useAffectedRow, set clientFoundRows=false, and update returns the actual number of rows affected; set clientFoundRows=true, and update returns the number of rows matched; when the version is 2.5.5 and above, the compute node service supports self-adaption, that is, if the parameter useAffectedRows is used for client connection, the setting of the client will prevail, and the parameter will be abandoned.

For example: jdbc is committed useAffectedRows=false, the number of rows matched will be returned.



The screenshot shows an IDE interface with a code editor and a debugger tool window.

Code Editor:

```
231
232
233 public static void main(String[] args) {
234     try {
235         BasicDataSource config=new BasicDataSource();
236         config.setUsername("cxd");
237         config.setPassword("123456");
238         config.setUrl("jdbc:mysql://192.168.220.102:3323/test1?useAffectedRows=false");
239         config.setInitialSize(5);
240         config.setMaxActive(50);
241         config.setMinIdle(5);
242         config.setMaxIdle(15);
243         config.setMaxWait(15000);
244         JdbcTemplate jdbcTemplate = new JdbcTemplate(config);
245         int result = jdbcTemplate.update("update t01 set name='示例1' where id=1");
246         System.out.println("返回影响行数: " + result);
247     } catch (Exception e) {
248         e.printStackTrace();
249     }
250 }
251
252 }
```

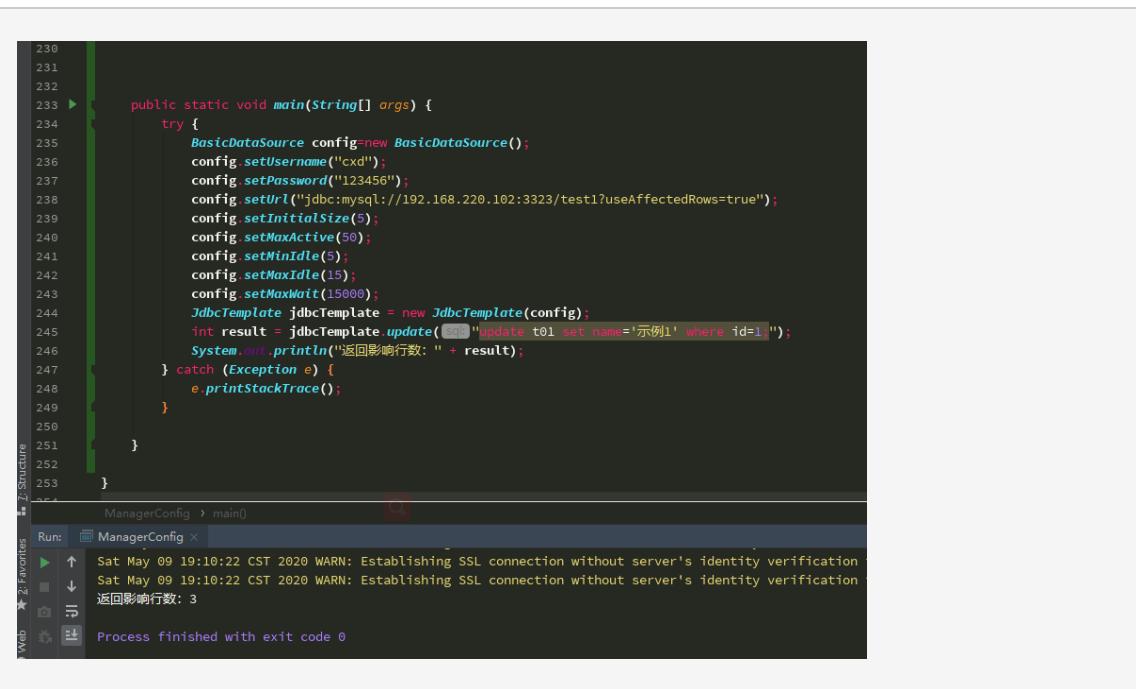
Debugger Tool Window:

- Shows the current stack trace:

```
Sat May 09 18:56:57 CST 2020 WARN: Establishing SSL connection without server's identity verification
Sat May 09 18:56:57 CST 2020 WARN: Establishing SSL connection without server's identity verification
Sat May 09 18:56:57 CST 2020 WARN: Establishing SSL connection without server's identity verification
Sat May 09 18:56:57 CST 2020 WARN: Establishing SSL connection without server's identity verification
返回影响行数: 4
Disconnected from the target VM, address: '127.0.0.1:54659', transport: 'socket'
```
- Shows the process status:

```
Process finished with exit code 0
```

jdbc is committed useAffectedRows=true, the actual number of rows affected will be returned.



```

230
231
232
233 public static void main(String[] args) {
234     try {
235         BasicDataSource config=new BasicDataSource();
236         config.setUsername("xd");
237         config.setPassword("123456");
238         config.setUrl("jdbc:mysql://192.168.220.102:3323/test1?useAffectedRows=true");
239         config.setInitialSize(5);
240         config.setMaxActive(50);
241         config.setMinIdle(5);
242         config.setMaxIdle(15);
243         config.setMaxWait(15000);
244         JdbcTemplate jdbcTemplate = new JdbcTemplate(config);
245         int result = jdbcTemplate.update("update t01 set name='示例1' where id=1");
246         System.out.println("返回影响行数: " + result);
247     } catch (Exception e) {
248         e.printStackTrace();
249     }
250 }
251
252 }
253

```

ManagerConfig > main()

Run: ManagerConfig X

Sat May 09 19:10:22 CST 2020 WARN: Establishing SSL connection without server's identity verification
Sat May 09 19:10:22 CST 2020 WARN: Establishing SSL connection without server's identity verification

返回影响行数: 3

Process finished with exit code 0

14.1.13. clusterElectionTimeoutMs

Description of parameter:

Property	Value
Parameter value	clusterElectionTimeoutMs
Visible or not	No
Description of parameters	Cluster Election Timeout (ms)
Default value	2000
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.3

Parameter Setting:

clusterElectionTimeoutMs parameter configuration in Server.xml is configured as follow:

```
<property name="clusterElectionTimeoutMs">2000</property><!-- Cluster Election Timeout (ms) -->
```

Role of parameter:

This parameter is used for setting Cluster Election Timeout of the compute node, and generally modification is not recommended, and appropriate adjustment could be made according to the service network quality. For example, setting the parameter clusterElectionTimeoutMs as 2000ms, then if election fails, in case of failure with the master compute node in the cluster, it will wait for election within the timeout 2000ms until election succeeded, and the selection will fail if exceeding 2000ms.

14.1.14. clusterHeartbeatTimeoutMs

Description of parameter:

Property	Value
Parameter value	clusterHeartbeatTimeoutMs
Visible or not	No
Description of parameters	Cluster Heartbeat Timeout (ms)
Default value	5000
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.3

Parameter Setting:

clusterHeartbeatTimeoutMs parameter configuration in Server.xml is configured as follow:

```
<property name="clusterHeartbeatTimeoutMs">5000</property><!-- Cluster Heartbeat Timeout (ms) -->
```

Role of parameter:

This parameter is used for setting Cluster Heartbeat Timeout of the compute node, and generally modification is not recommended, and appropriate adjustment could be made according to the service network quality.

14.1.15. clusterHost

Description of parameter:

Property	Value
Parameter value	clusterHost
Visible or not	Yes
Description of parameters	IP of the current node
Default value	192.168.200.1
Whether Reload is valid or not	No
Min Compatible Version	2.5.0

Parameter Setting:

clusterHost parameter configuration in Server.xml is configured as follow:

```
<property name="clusterHost">192.168.200.1</property><!-- IP of the current node -->
```

Role of parameter:

This parameter shall be set consistent with the actual IP of the compute node (can't be replaced with 127.0.0.1), and at the time of cluster election, it shall provide communication address between this compute node and other compute nodes.

14.1.16. clusterName

Description of parameter:

Property	Value
Parameter value	clusterName
Visible or not	Yes
Description of parameters	Cluster Group Name
Default value	HotDB-Cluster
Whether Reload is valid or not	No
Min Compatible Version	2.5.0

Parameter Setting:

clusterName parameter configuration in Server.xml is configured as follow:

```
<property name=" HotDB-Cluster "> HotDB-Cluster </property><!-- cluster group name -->
```

Role of parameter:

Specify the name of the group added after cluster startup, and this parameter of the compute nodes within the same cluster group shall be set identical, while this parameter of the compute nodes of different cluster groups shall be set different.

14.1.17. clusterNetwork

Description of parameter:

Property	Value
Parameter value	clusterNetwork
Visible or not	Yes
Description of parameters	Cluster Network Segment
Default value	192.168.200.0/24
Whether Reload is valid or not	No
Min Compatible Version	2.5.0

Parameter Setting:

clusterNetwork parameter configuration in Server.xml is configured as follow:

```
<property name="clusterNetwork">192.168.200.0/24</property><!-- Cluster Network Segment -->
```

Role of parameter:

This parameter is the network segment of the whole cluster, and it's limited that all compute nodes IP of the cluster must be within this network segment. Otherwise, even if the cluster group makes the same start, it can't join the cluster

14.1.18. clusterPacketTimeoutMs

Description of parameter:

Property	Value
Parameter value	clusterPacketTimeoutMs
Visible or not	No
Description of parameters	Failure time of inter-cluster communication packet (ms)
Default value	5000
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.3

Parameter Setting:

clusterPacketTimeoutMs parameter configuration in Server.xml is configured as follow:

```
<property name="clusterPacketTimeoutMs">5000</property><!-- Failure time of inter-cluster communication  
packet (ms) -->
```

Role of parameter:

This parameter is used for setting Cluster Packet Timeout, and generally modification is not recommended, and appropriate adjustment could be made according to the service network quality. Cluster packet refers to all point-to-point packets ought to be sent during normal running of the cluster, including but not limited to Heartbeat, Election, member change and other data packets.

14.1.19. clusterPort

Description of parameter:

Property	Value
Parameter value	clusterPort
Visible or not	Yes
Description of parameters	Cluster Communication Port
Default value	3326
Whether Reload is valid or not	No
Min Compatible Version	2.5.0

Parameter Setting:

clusterPort parameter configuration in Server.xml is configured as follow:

```
<property name="clusterPort">3326</property><!-- Cluster Communication Port -->
```

Role of parameter:

The default value 3326 specifies the port of listening cluster information. This parameter is used for inter-communication within the cluster, and the communication port within the same cluster must be the same.

14.1.20. clusterSize

Description of parameter:

Property	Value
Parameter value	clusterSize
Visible or not	Yes
Description of parameters	Total number of nodes in cluster
Default value	3
Whether Reload is valid or not	No
Min Compatible Version	2.5.0

Parameter Setting:

clusterSize parameter configuration in Server.xml is configured as follow:

```
<property name="clusterSize">3</property><!-- Total number of nodes in cluster -->
```

Role of parameter:

This parameter is total number of compute node in cluster. If haMode is set as 1 (that is the cluster mode), then it shall be configured as actual number of compute nodes of this cluster.

14.1.21. clusterStartedPacketTimeoutMs

Description of parameter:

Property	Value
Parameter value	clusterStartedPacketTimeoutMs
Visible or not	No
Description of parameters	Failure Time of Cluster Started Broadcast Packet (ms)
Default value	5000
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.3

Parameter Setting:

clusterStartedPacketTimeoutMs parameter configuration in Server.xml is configured as follow:

```
<property name="clusterStartedPacketTimeoutMs">2000</property><!-- Failure Time of Cluster Started  
Broadcast Packet (ms) -->
```

Role of parameter:

This parameter is used for setting Cluster Started Packet Timeout, and generally modification is not recommended, and appropriate adjustment could be made according to the service network quality. Cluster Started Packet refers to a broadcast packet targeting the network segment when cluster is enabled. In case of failure with the Started Packet due to timeout, it may result that service port of this compute node can't be enabled.

14.1.22. configMGR & bak1Url & bak1Username & bak1Password

Description of parameter:

Property	Value
Parameter value	configMGR
Visible or not	Yes
Description of parameters	Whether configDB uses MGR or not
Default value	false
Whether Reload is valid or not	YesY
Min Compatible Version	2.5.0

Property	Value
Parameter value	bak1Url
Visible or not	Yes
Description of parameters	MGR configDB address
Default value	空 Null
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.0

Property	Value
Parameter value	bak1Username
Visible or not	Yes
Description of parameters	MGR configDB username
Default value	Null
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.0

Property	Value
Parameter value	bak1Password
Visible or not	Yes
Description of parameters	MGR configDB password

Default value	Null
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.0

Role of parameter:

configMGR and bak1Url and bak1Username and bak1Password are supporting parameters, and are used for MGR configDB function. If to use MGR configDB, it needs to be set as information of corresponding MGR configDB and guarantee normal replication relation of the MGR configDB instance, and they are MGR mutually, and in case of failure with the Master configDB, it will switch to new Master configDB automatically. At most 3 MGR configDB are supported.

```
<property name="configMGR">true</property> <!-- Whether configDB uses MGR or not -->

<property name="bak1Url">jdbc:mysql://192.168.210.32:3306/hotdb_config</property> <!-- MGR configDB address(if configDB uses MGR, this item must be configured), real IP address of the configDB service shall be specified -->

<property name="bak1Username">hotdb_config</property> <!-- MGR configDB username (if configDB uses MGR, this item must be configured) -->

<property name="bak1Password">hotdb_config</property> <!-- MGR configDB password (if configDB uses MGR, this item must be configured) -->
```

14.1.23. crossDbXa

Description of parameter:

Property	Value
Parameter value	crossDbXa
Visible or not	No

Property	Value
Description of parameters	Whether XA transactions are adopted in cross-LogicDB
Default value	false
Whether Reload is valid or not	No
Min Compatible Version	2.5.5

Parameter Setting:

crossDbXa is configured in server.xml as follows:

```
<property name="crossDbXa">false</property>
```

Role of parameter:

When enableXA is enabled, if there are XA transactions queries across LogicDBs, you need to enable crossDbXa to ensure strong consistency. It can also be supported when crossDbXa is not enabled, but the strong consistency of data is not guaranteed, and if a new node is added to the transaction, the query will report an error. For example:

Data preparation:

1. Enable XA
2. LogicDB A, the default node is 2, 4; and LogicDB B, the default node is 2, 3, 4
3. LogicDB A creates table a; LogicDB B creates table b; the table structure of the two tables is consistent
4. 1000 pieces of data are inserted into table a, but no data in table b

Scenario 1: when crossDbXa is disabled, strong data consistency is not guaranteed:

1. Open a session and execute the following SQL:

```
use A;
begin;insert into B.b select * from A.a;commit;
```

```
use B;
begin;delete from b;commit;
```

The two transactions are executed alternately without interval;

2. Open another session and execute it repeatedly:

```
use A;
select count(*) from B.b;
```

Result: the results of Count (*) doesn't have to be all 0 or 1000

```
root@127.0.0.1:test_db_a 5.7.25 06:18:06> select count(*) from test_db_b.test_b;
+-----+
| count(*) |
+-----+
|      674 |
+-----+
1 row in set (0.01 sec)
```

Scenario 2: when crossDbXa is enabled, strong data consistency is guaranteed:

1. Open a session and execute the following SQL:

```
use A;
begin;insert into B.b select * from A.a;commit;
use B;
begin;delete from b;commit;
```

The two transactions are executed alternately without interval;

2. Open another session and execute it repeatedly:

```
use A;
select count(*) from B.b;
```

Result: the results of Count (*) be 0 or 1000

```
root@127.0.0.1:test_db_a 5.7.25 06:20:44> select count(*) from test_db_b.test_b;
+-----+
| count(*) |
+-----+
|      0 |
+-----+
1 row in set (0.01 sec)

root@127.0.0.1:test_db_a 5.7.25 06:20:46> select count(*) from test_db_b.test_b;
+-----+
| count(*) |
+-----+
|    1000 |
+-----+
1 row in set (0.00 sec)
```

Scenario 3: when crossDbXa is disabled, error will be reported when a node is added to the transaction:

1. Open a session and execute the following SQL:

```
use A;
begin;
select * from A.a;
select * from B.b;
```

Result: select * from B.b; error will be reported when executes

```
root@127.0.0.1:test_db_a 5.7.25 06:53:47>
root@127.0.0.1:test_db_a 5.7.25 06:53:48> select * from test_db_b.test_b;;
ERROR 1064 (HY000): request a new backend connection in an active xa transaction is not supported
```

Scenario 4: when crossDbXa is enabled, execute normally when a node is added to the transaction:

1. Open a session and execute the following SQL:

```
use A;
begin;select * from A.a;select * from B.b;
```

Result: select * from B.b; execute normally

```
root@127.0.0.1:test_db_a 5.7.25 06:20:46> select count(*) from test_db_b.test_b;
+-----+
| count(*) |
+-----+
|    1000 |
+-----+
1 row in set (0.00 sec)
```

14.1.24. cryptMandatory

Description of parameter:

Property	Value
Parameter value	cryptMandatory
Visible or not	Yes
Description of parameters	Mandatory password encryption or not
Default value	False

Property	Value
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="cryptMandatory">false</property><!-- Mandatory password encryption or not, Yes: true, No:  
false -->
```

Role of parameter:

It's used for setting whether to make mandatory password identification or not when the compute node reads the data source password

- True status:
 - When the data source password is plaintext, the compute node will fail to connect to this data source
 - When the data source password is ciphertext, the compute node can connect to this data source
- False status:
 - When the data source password is plaintext, the compute node can connect to this data source
 - When the data source password is ciphertext, the compute node will fail to connect to this data source

14.1.25. dataNodeIdleCheckPeriod

Description of parameter:

Property	Value
Parameter value	dataNodeIdleCheckPeriod
Visible or not	Yes
Description of parameters	Default Data Node Idle Check Period (S)
Default value	120
Min value	1
Max value	3600
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="dataNodeIdleCheckPeriod">120</property><!-- Default Data Node Idle Check Period (S) -->
```

Role of parameter:

It is used for setting the periodical task period of data node idle check. The compute node will check connection condition of the back-end data source periodically, disable excessive idle connection or supplement available connections of the connection pool, guarantee that the connections can't be disabled by MySQL, and maintain normal running of the connection pool.

For example: Conduct on large concurrent insert operation on the 3323 service port, execute Show @@backend on the 3325 management port to view back-end connection. When execution of 3323 service port completed, if the compute node checks that there are idle back-end connections in the data node, the compute node will clear up these connections.

14.1.26. deadlockCheckPeriod

Description of parameter:

Property	Value
Parameter value	deadlockCheckPeriod
Visible or not	Yes
Description of parameters	Deadlock Check Period (Ms), 0 means Disable
Default value	3000
Min value	0
Max value	100000
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.3

Role of parameter:

When Deadlock Check is enabled, it will check cross-node deadlock periodically according to the set period.

When deadlock is enabled, it will check cross-node deadlock periodically according to the Deadlock Check Period, and once cross-node deadlock occurs, it will kick out the connection with the Min trx_weight, and rollback the whole transaction.

```
mysql> select * from autoi where id=4 for update;
```

```
ERROR 1213 (HY000): Deadlock found when trying to get lock; try restarting transaction
```

When deadlock is disabled, it will wait for Lock Timeout, and the Lock Timeout is set according to the innodb_lock_wait_timeout parameter value in MySQL data source.

```
mysql> select * from autoi where id=10 for update;
ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting transaction
```

14.1.27. defaultMaxLimit

Description of parameter:

Property	Value
Parameter value	defaultMaxLimit
Visible or not	No
Description of parameters	default max limit
Default value	0
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

defaultMaxLimit parameter configuration in Server.xml is configured as follow:

```
<property name="defaultMaxLimit">10000</property><!-- default max limit -->
```

Role of parameter:

This parameter is related to the HotDB overload protection and is used in combination with the highCostSqlConcurrency parameter. When the front-end concurrency executes the cross-node update/delete limit n scenario, if n exceeds the defaultMaxLimit setting, the highCostSqlConcurrency parameter control will be triggered to limit the concurrency number of the high memory-consumption statements, and the related

connections will be held until the previous execution is completed, thus the next batch can be executed.

Reflected in show processlist, State is Flow control, waiting for the next batch of execution. In the following graph, to facilitate testing, set defaultMaxLimit=5, highCostSqlConcurrency=10, and use 20 concurrencies to execute cross-node update limit n. It can be seen that 10 connections are being executed, and the other 10 connections have been limited.

```
ztm@10.10.0.207:pm 5.7.19-HotDB-2.5.1 06:10:45> show processlist;
+-----+-----+-----+-----+-----+
-----+
| Id | User | Host | db | Command | Time | State | Info
|-----+-----+-----+-----+-----+
-----+
| 4 | ztm | 10.10.0.201:57882 | PM | Query | 0 | executing | show processlist
|
+-----+-----+-----+-----+-----+
-----+
| 6 | ztm | 10.10.0.201:57905 | PM | Query | 1 | Sending data | UPDATE customer_route_1 SET
address = 'abcd' order by id LIMIT 20 |
|
| 7 | ztm | 10.10.0.201:57902 | PM | Query | 1 | Flow control | UPDATE customer_route_1 SET
address = 'abcd' order by id LIMIT 20 |
|
| 8 | ztm | 10.10.0.201:57912 | PM | Query | 1 | Sending data | UPDATE customer_route_1 SET
address = 'abcd' order by id LIMIT 20 |
|
| 9 | ztm | 10.10.0.201:57900 | PM | Query | 1 | Sending data | UPDATE customer_route_1 SET
address = 'abcd' order by id LIMIT 20 |
|
| 10 | ztm | 10.10.0.201:57919 | PM | Query | 1 | Sending data | UPDATE customer_route_1 SET
address = 'abcd' order by id LIMIT 20 |
```

```
| 11 | ztm | 10.10.0.201:57911 | PM | Query | 1 | Sending data | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 12 | ztm | 10.10.0.201:57904 | PM | Query | 1 | Flow control | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 13 | ztm | 10.10.0.201:57906 | PM | Query | 1 | Flow control | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 14 | ztm | 10.10.0.201:57903 | PM | Query | 1 | Sending data | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 15 | ztm | 10.10.0.201:57910 | PM | Query | 1 | Flow control | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 16 | ztm | 10.10.0.201:57908 | PM | Query | 1 | Flow control | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 17 | ztm | 10.10.0.201:57920 | PM | Query | 1 | Flow control | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 18 | ztm | 10.10.0.201:57907 | PM | Query | 1 | Sending data | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 19 | ztm | 10.10.0.201:57913 | PM | Query | 1 | Flow control | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 20 | ztm | 10.10.0.201:57909 | PM | Query | 1 | Flow control | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 21 | ztm | 10.10.0.201:57921 | PM | Query | 1 | Sending data | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 22 | ztm | 10.10.0.201:57918 | PM | Query | 1 | Sending data | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 23 | ztm | 10.10.0.201:57962 | PM | Query | 1 | Flow control | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 24 | ztm | 10.10.0.201:57915 | PM | Query | 1 | Flow control | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |  
  
| 25 | ztm | 10.10.0.201:57914 | PM | Query | 1 | Sending data | UPDATE customer_route_1 SET  
address = 'abcd' order by id LIMIT 20 |
```

```
+-----+-----+-----+-----+-----+
```

```
|-----+
```

```
21 rows in set (0.01 sec)
```

14.1.28. dropTableRetentionTime

Description of parameter:

Property	Value
Parameter value	dropTableRetentionTime
Visible or not	Yes
Description of parameters	dropTableRetentionTime, 0 by default, no retention
Default value	0
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.2

Parameter Setting:

dropTableRetentionTime parameter configuration in Server.xml is configured as follow:

```
<property name=" dropTableRetentionTime">0</property><!-- dropTableRetentionTime, 0 by default, no
retention -->
```

Role of parameter:

In v.2.5.5, dropTableRetentionTime parameter is 0 by default, meaning not to retain the DROP table, and execution of DROP TABLE statement will delete the table; when

dropTableRetentionTime is greater than 0, it's computed by hour, and the DROP table will be retained to the set time, and this table will be deleted automatically if exceeding the set time. For example, dropTableRetentionTime=24 means that the DROP table will be retained, and then be deleted 24h later.

14.1.29. drBakUrl & drBakUsername & drBakPassword

Description of parameter:

Property	Value
Parameter value	drBakUrl
Visible or not	Yes
Description of parameters	Slave ConfigDB address of DR center
Default value	jdbc:mysql://127.0.0.1:3306/hotdb_config
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.3.1

Property	Value
Parameter value	drBakUsername
Visible or not	Yes
Description of parameters	Slave ConfigDB username of DR center
Default value	hotdb_config

Property	Value
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.3.1

Property	Value
Parameter value	drBakPassword
Visible or not	Yes
Description of parameters	Slave ConfigDB password of DR center
Default value	hotdb_config
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.3.1

Role of parameter:

drBakUrl, drBakUsername and drBakPassword are supporting parameters, which are used for the high availability functions of ConfigDB in the DR center. When the DR center is switched to be the current active center, if the ConfigDB is highly available, the information of the corresponding slave ConfigDB needs to be set and the replication relation of the master-slave ConfigDB instances should be normal, and the master-slave ConfigDB instances should be mutually standby. When the DR center is switched to be the current active center, the master ConfigDB will automatically switch to the slave

ConfigDB in the case of failure. At this time, for the high availability switching of the ConfigDB, you can refer to the description of parameters [bakUrl](#), [bakUsername](#), [bakPassword](#) of the slave ConfigDB in the master center.

```
<property name="drBakUrl">jdbc:mysql://192.168.240.77:3316/hotdb_config</property><!-- Slave ConfigDB address of DR center -->

<property name="drBakUsername">hotdb_config</property><!-- Slave ConfigDB username of DR center -->

<property name="drBakPassword">hotdb_config</property><!-- Slave ConfigDB password of DR center -->
```

14.1.30. drUrl & drUsername & drPassword

Description of parameter:

Property	Value
Parameter value	drUrl
Visible or not	Yes
Description of parameters	ConfigDB address of DR center
Default value	jdbc:mysql://127.0.0.1:3306/hotdb_config
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.3.1

Property	Value
Parameter value	drUsername
Visible or not	Yes

Property	Value
Description of parameters	ConfigDB usernameof DR center
Default value	hotdb_config
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.3.1

Property	Value
Parameter value	drPassword
Visible or not	Yes
Description of parameters	ConfigDB passwordof DR center
Default value	hotdb_config
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.3.1

Role of parameter:

drUrl, drUsername and drPassword are supporting parameters, among which drUrl refers to the ConfigDB path of the compute node configuration in theDR center; drUsername and drPassword refer to the username and password connecting the

database, which is used to store the configuration of the DR center. Please refer to the related parameters [url, username and password](#) of the master center ConfigDBs.

```
<property name="drUrl">jdbc:mysql://192.168.240.76:3316/hotdb_config</property><!-- ConfigDB address of DR center -->

<property name="drUsername">hotdb_config</property><!-- ConfigDB username of DR center -->

<property name="drPassword">hotdb_config</property><!-- ConfigDB password of DR center -->
```

14.1.31. enableCursor

Description of parameter:

Property	Value
Parameter value	enableCursor
Visible or not	Yes
Description of parameters	Allow PREPARE statement to obtain data via CURSOR or not
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.6

Parameter Setting:

enableCursor parameter in Server.xml

```
<property name="enableCursor">false</property>
```

Role of parameter:

Allow PREPARE to obtain data content via Cursor or not
(jdbcURL:useCursorFetch=true)

14.1.32. enableFlowControl**Description of parameter:**

Property	Value
Parameter value	enableFlowControl
Visible or not	Yes
Description of parameters	Enable data source flow control or not
Default value	False
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.5

Role of parameter:

Enable by default in Version 2.4.8 and later versions. After Enable, it will limit back-end concurrency, protect the data source, and control the data source pressure. View the control condition flow_control via show @@datasource on management port.

```
mysql> show @@datasource;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
|-----+-----+
```

dn	ds	name	type	status	host	port	schema
active	idle	size	unavailable_reason	flow_control			
<hr/>							
<hr/>							
186	227	192.168.210.68_3307_db252	1	1	192.168.210.68	3307	db252
2	512	514	NULL	0/64			
186	228	192.168.210.68_3308_db252	3	1	192.168.210.68	3308	db252
1	512	513	NULL	0/64			
1	7	192.168.210.41_3307_db252	1	1	192.168.210.41	3307	db252
2	512	514	NULL	0/64			
2	8	192.168.210.42_3307_db252	1	1	192.168.210.42	3307	db252
2	512	514	NULL	0/16			
101	74	192.168.210.41_3307_clu_db30	1	1	192.168.210.41	3307	clu_db30
2	512	514	NULL	0/64			
190	237	192.168.200.191_3307_db252	1	1	192.168.200.191	3307	db252
2	512	514	NULL	0/64			
190	238	192.168.200.190_3307_db252	2	1	192.168.200.190	3307	db252
1	512	513	NULL	0/64			
4	5	192.168.210.43_3308_db252	1	1	192.168.210.43	3308	db252
2	32	34	NULL	0/64			
4	77	192.168.210.44_3308_db252	3	1	192.168.210.44	3308	db252
1	512	513	NULL	0/64			
191	239	192.168.200.191_3308_db252	1	1	192.168.200.191	3308	db252
2	512	514	NULL	0/64			
191	240	192.168.200.190_3308_db252	2	1	192.168.200.190	3308	db252
1	512	513	NULL	0/64			
127	88	192.168.210.42_3308_db252	3	1	192.168.210.42	3308	db252
1	512	513	NULL	0/64			
127	87	192.168.210.41_3308_db252	1	1	192.168.210.41	3308	db252
2	512	514	NULL	0/16			

```

| -1 | -1 | configDatasource | 1 | 1 | 192.168.210.30 | 3307 | hotdb_config_test252
|     1 |    7 |    8 | NULL | N/A |
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+
14 rows in set (0.00 sec)

```

Note: Data source flow control is internal flow control algorithm of the compute node.

14.1.33. enableHeartbeat&heartbeatPeriod& heartbeatTimeoutMs

Description of parameter:

Property	Value
Parameter value	enableHeartbeat
Visible or not	Yes
Description of parameters	Enable Heartbeat or not, Yes: true, No: false
Default value	true
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.3

Property	Value
Parameter value	heartbeatPeriod
Visible or not	Yes

Property	Value
Description of parameters	Heartbeat Period (S)
Default value	2
Max value	60
Min value	1
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.3

Property	Value
Parameter value	heartbeatTimeoutMs
Visible or not	Yes
Description of parameters	Heartbeat Timeout (Ms)
Default value	500
Max value	10000
Min value	100
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.3

The default value of Heartbeat Check Period is 2s, that is Periodical Heartbeat Check is executed every 2s. The default value of Heartbeat Timeout is 500ms.

Role of parameter:

Detect whether the data source is available and whether there is sharing of use.

After Heartbeat Check is enabled, it will check whether the data source is normally connected or not according to Heartbeat Check Period. When the network is unreachable or data source fails, under the condition of existing Slave data source and Configuration Switching Rule, it will switch to Slave data source, and the Master data source will be set Unavailable; for data source switch logic, please refer to [*Distributed Transactional Database HotDB Server \[Management Platform\] Function Manual*](#).

When a data source is cited by multiple compute node services simultaneously, then after reload, the compute node will prompt via log that: there's another HotDB using this datasource...restart heartbeat. Under the condition that Heartbeat Check is disabled, then Data Node/ConfigDB High Availability can't be realized, failover can't be made, and sharing condition of the data source can't be checked, etc.

For Heartbeat Timeout under the condition that Heartbeat is enabled, in case of data source failure or the Slave network latency, the Heartbeat failure exceeds the threshold value, then there will be log heartbeat timeout output:

```
2018-05-29 16:32:52.924 [WARN] [HEARTBEAT] [HeartbeatTimer] a(-1) – Datasource:-1  
128.0.0.1:3306/hotdb_config time out! Last packet sent at:2018-05-29 04:32:49.886...omitted...
```

Note: If the current data source is the last data source of the node, then the data source will not be set Unavailable and will try connection all the time; if being pure Slave data source, even if Heartbeat failure times have exceeded the threshold value, as long as Heartbeat could make successful connection with the data source, it will not be marked Unavailable.

14.1.34. enableLatencyCheck & latencyCheckPeriod

Description of parameter:

Property	Value
Parameter value	enableLatencyCheck
Visible or not	Yes
Description of parameters	Enable master/slave latency check or not
Default value	true
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.5

Property	Value
Parameter value	latencyCheckPeriod
Visible or not	Yes
Description of parameters	Master/Slave Latency Check Period (ms)
Default value	500
Max value	1000
Min value	100
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.5

The default value of Master/Slave Latency Check Period is 500ms, that is, the periodical check executes master/slave latency check every 500ms.

Role of parameter:

Latency check relies on Heartbeat table. If master/slave latency check is enabled, it will check whether the Standby Slave has replication latency, whether it catches up with replication synchronously. This parameter has a certain influence when the data source/configDB switch and the compute node is enabled. If to conduct master/slave latency check before data source failover, there shall be Configuration Failover Rule in advance; with no Configuration Switching Rule, there will be no Master/Slave data source switch and master/slave latency check, please refer to [*Distributed Transactional Database HotDB Server \[Management Platform\] Function Manual*](#).

Log in to management port, use show @@latency; and you could view the master/slave latency time.

```
mysql> show @@latency;

+-----+-----+-----+
| dn   | info           | backup_info          | latency |
+-----+-----+-----+
| 186  | 192.168.210.68:3307/db252 | 192.168.210.68:3308/db252 | 501 ms    |
| 4    | 192.168.210.43:3308/db252 | 192.168.210.44:3308/db252 | 0 ms     |
| 190  | 192.168.200.191:3307/db252 | 192.168.200.190:3307/db252 | 0 ms     |
| 191  | 192.168.200.191:3308/db252 | 192.168.200.190:3308/db252 | 0 ms     |
| 127  | 192.168.210.41:3308/db252 | 192.168.210.42:3308/db252 | STOPPED |
+-----+-----+-----+
5 rows in set (0.02 sec)
```

14.1.35. enableListener

Description of parameter:

Property	Value
Parameter value	enableListener
Visible or not	Yes
Description of parameters	Enable Listener mode or not.
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.5

Parameter Setting:

```
<property name="enableListener">false</property><!--Enable Listener mode or not-->
```

Role of parameter:

HotDB Listener is a pluggable component of compute node, which can solve the problem of linear performance expansion under the strong consistency (XA) mode of cluster. The following requirements shall be met before using the Listener: the compute node is in multi-node cluster mode with XA enabled; Listener is successfully deployed on the data source server with enableListener enabled. Executing the dynamic loading and then the following command, you can view whether the identification is successful or not and the real-time status of Listener via `listener_status`.

```
MySQL [(none)]> show @@datasource;
+-----+-----+-----+-----+-----+-----+-----+-----+
| dn   | ds   | name          | type | status | host          | port | schema      | active | idle | size |
+-----+-----+-----+-----+-----+-----+-----+-----+
| unavailable_reason | flow_control | idc_id | listener_id | listener_status |
```

17	17	10.10.0.140_3313_db01 1	1	10.10.0.140 3313 db01	0 45
45 NULL		0/64	1 8 1		
16	16	10.10.0.140_3312_db01 1	1	10.10.0.140 3312 db01	0 47
47 NULL		0/64	1 7 1		
19	19	10.10.0.155_3311_db01 1	1	10.10.0.155 3311 db01	0 49
49 NULL		0/64	1 10 1		
18	18	10.10.0.155_3310_db01 1	1	10.10.0.155 3310 db01	0 53
53 NULL		0/64	1 9 1		
21	21	10.10.0.155_3313_db01 1	1	10.10.0.155 3313 db01	0 55
55 NULL		0/64	1 12 1		
20	20	10.10.0.155_3312_db01 1	1	10.10.0.155 3312 db01	0 47
47 NULL		0/64	1 11 1		
10	10	10.10.0.125_3310_db01 1	1	10.10.0.125 3310 db01	0 44
44 NULL		0/64	1 1 1		
11	11	10.10.0.125_3311_db01 1	1	10.10.0.125 3311 db01	0 43
43 NULL		0/64	1 2 1		
12	12	10.10.0.125_3312_db01 1	1	10.10.0.125 3312 db01	0 44
44 NULL		0/64	1 3 1		
13	13	10.10.0.125_3313_db01 1	1	10.10.0.125 3313 db01	0 48
48 NULL		0/64	1 4 1		
14	14	10.10.0.140_3310_db01 1	1	10.10.0.140 3310 db01	0 44
44 NULL		0/64	1 5 1		
15	15	10.10.0.140_3311_db01 1	1	10.10.0.140 3311 db01	0 62
62 NULL		0/64	1 6 1		
-1	-1	configDatasource	1 1 10.10.0.121 3306 hotdb_config	1 11	
12 NULL		N/A	-1 0 1		

13 rows in set (0.00 sec)

注: listener_status 为 1, 代表 Listener 可用; listener_status 为 0, 代表 Listener 不可用

Please refer to the description of [linear expansion](#) for precautions and other supporting methods.

14.1.36. enableOracleFunction

Description of parameter:

Property	Value
Parameter value	enableOracleFunction
Visible or not	No
Description of parameters	Support oracle function or not
Default value	false
Max value	/
Min value	/
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.6

Parameter Setting:

enableOracleFunction is a hidden parameter. If you want to enable it, you need to add it into Server.xml. The default parameter value is false, configured as follows:

```
<property name="enableOracleFunction">false</property><!-- support oracle function or not -->
```

Role of parameter:

When Oracle data is migrated to MySQL, some functions are replaced to make it run successfully and reduce the cost of migration. Besides, sequence objects and its related functions of Oracle are supported. When this parameter is enabled, it will be parsed according to Oracle mode first, otherwise it will be parsed according to MySQL mode.

- For functions supported by Oracle but not supported by MySQL, some of them support rewriting. If you want to know functions that compute nodes support to rewrite, please refer to “HotDB Server Latest Function List”.

When it is set to true, the Oracle function parse identifies rewriting support and executes successfully. Example:

```
root@192.168.210.202:cc 5.7.23 05:09:07> select to_char(sysdate,'yyyy-MM-dd H
H24:mi:ss') from dual;

+-----+
| to_char(sysdate,'yyyy-MM-dd HH24:mi:ss') |
+-----+
| 2020-09-24 17:09:30 |
+-----+
1 row in set (0.01 sec)
```

When it is set to false, for functions not supported by MySQL, it will report “not support” or “function does not exist”.

```
root@192.168.210.225:cc 5.7.23 05:09:11> select to_char(sysdate,'yyyy-MM-dd H
H24:mi:ss') from dual;

ERROR 1305 (42000): FUNCTION db256_01.TO_CHAR does not exist
```

- For Sequence related functions, please refer to “HotDB Server Latest Function List”

When it is set to true, the sequence related functions execute successfully. Example:

```
root@192.168.210.202:cc 5.7.23 11:30:09> create sequence sequence_test
-> minvalue 1
-> maxvalue 1000
-> start with 1
-> increment by 10;
Query OK, 1 row affected (0.04 sec)
```

When it is set to false, it will prompt error:

```
root@192.168.210.225:cc 5.7.23 11:43:11> create sequence sequence_256
-> minvalue 1
-> maxvalue 1000
-> start with 1
-> increment by 10;
ERROR 10010 (HY000): expect VIEW. lexer state: token=IDENTIFIER, sqlLeft=sequence_256
```

14.1.37. enableSleep

Description of parameter:

Property	Value
Parameter value	enableSleep
Visible or not	Yes
Description of parameters	Whether SLEEP Function is allowed or not
Default value	false

Property	Value
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="enableSleep">false</property><!-- Whether SLEEP Function is allowed or not, Yes: true, No:  
false -->
```

Role of parameter:

It's used for setting whether execution of sleep parameter is allowed in compute node

Sleep Function is not allowed:

```
mysql> select sleep(2);  
  
ERROR 1064 (HY000): forbidden function:SLEEP, go check your config file to enable it.
```

Sleep Function is allowed:

```
mysql> select sleep(2);  
  
+-----+  
| sleep(2) |  
+-----+  
| 0 |  
+-----+  
  
1 row in set (2.00 sec)
```

14.1.38. enableSSL

Description of parameter:

Property	Value
Parameter value	enableSSL
Visible or not	Yes
Description of parameters	Whether to enable SSL connection function
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.5

Role of parameter:

```
<property name="enableSSL">false</property><!-- Whether to enable SSL connection function, yes: true,  
no: false -->
```

This parameter is used to set whether the compute node is allowed to connect using SSL authentication. For details, please refer to the [TLS connection login](#), and use it together with parameters [keyStore](#) and [keyStorePass](#).

14.1.39. enableSubquery

Description of parameter:

Property	Value
Parameter value	enableSubquery
Visible or not	No

Property	Value
Description of parameters	Whether Subquery under special scenarios is allowed or not
Default value	true
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Role of parameter:

Whether the table in Subquery is allowed to be Sharding Table. After Version 2.4.7, this parameter is Enabled by default, and can support Subquery under more scenarios. In the previous versions, Enabling this parameter does not guarantee accuracy of the Subquery results.

When set as false, it means that the table in Subquery is not allowed to be Sharding Table, and there will be prompt as follow:

```
mysql> select * from test3 where id in (select id from test31);
ERROR 1064 (HY000): Unsupported table type in subquery.
```

When set as true, it means that the table in Subquery is not allowed to be Sharding Table, there will be prompt as follow:

```
mysql> select * from test3 where id in (select id from test31);
+----+-----+
| id | name |
+----+-----+
| 5 | dfff56f |
| 7 | aa78bca |
```

15	dfff56f	
----	---------	--

...省略更多...

...More are omitted...

14.1.40. enableWatchdog

Description of parameter:

Property	Value
Parameter value	enableWatchdog
Visible or not	Yes
Description of parameters	Enable Watchdog or not
Default value	False
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.5

Parameter Setting:

```
<property name="enableWatchdog">true</property><!-- Enable Watchdog or not -->
```

Role of parameter:

It's used for checking abnormal connection and other abnormal status of front-end connection and back-end connection pool of the compute node. In case of abnormality, record log and terminate the connection.

You could view log tail -f hotdb.log|grep to see whether “watchdog” has been enabled or not

```
root> cat hotdb.log|grep 'watchdog'
```

```
2018-06-01 18:26:50.983 [WARN] [WATCHDOG] [$NIOREACTOR-7-RW] watchdogTableCheckHandler(78) -  
Table TABLEB not found in watchdog table structure check in HotOB memory, but was found in  
MySQLConnection [node=i, id=18, threadId=199616, state=running, closed=false, autocommit=true,  
host=192.168.200.5q, port=3308, database=db249, localPort=51691, isClose:false, toBeClose:false]. You may  
need to contact HotDB administrator to get help.
```

```
2018-06-01 18:26:50.986 [WARN] [WATCHDOG] [$NIOREACTOR-7-RW] watchdogTableCheckHandler(78) -  
Table TESTB not found in watchdog table structure check in HotOB memory, but was found in  
MySQLConnection [node=i, id=18, threadId=199616, state=running, closed=false, autocommit=true,  
host=192.168.200.5q, port=3308, database=db249, localPort=51691, isClose:false, toBeClose:false]. You may  
need to contact HotDB administrator to get help.
```

```
2018-06-01 18:26:50.988 [WARN] [WATCHDOG] [$NIOREACTOR-7-RW] watchdogTableCheckHandler(78) -  
Table JOIN_DN02 not found in watchdog table structure check in HotOB memory, but was found in  
MySQLConnection [node=i, id=18, threadId=199616, state=running, closed=false, autocommit=true,  
host=192.168.200.5q, port=3308, database=db249, localPort=51691, isClose:false, toBeClose:false]. You may  
need to contact HotDB administrator to get help.
```

...More are omitted...

You could view inconsistency check information of the table structure and memory via log:

```
2018-10-31 18:46:44.834 [WARN] [WATCHDOG] [$NIOREACTOR-0-RW] WatchdogTableCheckHandler(85) -  
Table CCC is inconsistent in watchdog table structure check between HotDB memory and MySQL:  
MySQLConnection [node=20, id=299, threadId=3748, state=running, closed=false, autocommit=true,  
host=192.168.210.41, port=3310, database=rmb01, localPort=58808, isClose:false, toBeClose:false]. You may  
need to contact HOtDB administrator to get help.
```

You could view inconsistency check information of configDB and memory via log:

```
2018-10-31 17:45:39.617 [INFO] [WATCHDOG] [Watchdog] WatchDog(500) – HotDB user config is  
inconsistent between config database and HotDB memory, Logic tables are not the same in FUN_RMB. you may  
need to reload HotDB config to bring into effect.
```

You could view check information of the transactions not committed within 24h via log:

```
2018-10-26 16:14:55.787 [INFO] [WATCHDOG] [$NIOREACTOR-0-RW]
WatchDogLongTransactionCheckHandler(123) - Session [thread=Thread-5,id=1720,user=rmb,host=192.168.200.3,port=3323,localport=54330,schema=FUNTEST_RMB] has not been
queried for 839s. executed IUD5:[INSERT INTO rmb_cbc VALUES (tuanjian, 4000)]. binded
connection:[MySQLConnection [node=11, id=1330, threadId=18085, state=borrowed, closed=false,
autocommit=false, host=192.168.210.42, port=3307, database=db251, localPort=15722, isClose:false,
toBeClose:false] lastSQL:INSERT INTO rmb_cbc VALUES (tuanjian, 4000)]. innodb_trx:[(ds:11
trx_id:25765462 trx_state:RUNNING trx_started:2018-10-26 16:00:56 trx_requested_lock_id:NULL
trx_wait_started:NULL trx_weight:2 trx_mysql_thread_id:18085 trx_query:NULL trx_operation_state:NULL
trx_tables_in_use:0 trx_tables_locked:1 trx_lock_structs:1 trx_lock_memory_bytes:1136 trx_rows_locked:0
trx_rows_modified:1 trx_concurrency_tickets:0 trx_isolation_level:REPEATABLE READ trx_unique_checks:1
trx_foreign_key_checks:1 trx_last_foreign_key_error:NULL trx_adaptive_hash_latched:0
trx_adaptive_hash_timeout:0 trx_is_read_only:0 trx_autocommit_non_locking:0)]. we will close this session now.
```

You could view check information of data source switch via log:

```
2018-10-26 19:29:01.146 [INFO] [MANAGER] [Labor-478] HotdbConfig(2164) - reload config successfully for
connection:[thread=Labor-478,id=1609,user=root,host=192.168.200.2,port=3325,localport=57440,schema=null]

2018-10-26 19:30:24.384 [INFO] [FAILOVER] [$NIOExecutor-7-2] SwitchDataSource(111) - received switch
datasource 24 command from Manager: [thread=$NIOExecutor-7-
2,id=1609,user=root,host=192.168.200.2,port=3325,localport=57440,schema=null]

2018-10-26 19:30:24.387 [WARN] [RESPONSE] [Labor-484] InitSequenceHandler(270) - FUN_RMB.BC's
sequence in Backup datasource: 25 is greater than current sequence

2018-10-26 19:30:24.387 [WARN] [RESPONSE] [Labor-474] InitSequenceHandler(270) - FUN_RMB.CBC's
sequence in Backup datasource: 25 is greater than current sequence

2018-10-26 19:30:24.387 [WARN] [RESPONSE] [Labor-484] InitSequenceHandler(270) • FUN_RMB.BC's
sequence in Backup datasource: 25 is greater than current sequence

2018-10-26 19:30:24.387 [WARN] [RESPONSE] [Labor-474] InitSequenceHandler(270) - FUN_RMB.CBC's
sequence in Backup datasource: 25 is greater than current sequence

2018-10-26 19:30:24.407 [INFO] [FAILOVER] [Labor-464] CheckSlaveHandler(852) - DN:20(dn_rmb_01)
switch datasource 24(192.168.210.41_3310_rmb0l)->25(192.168.210.42_3310_rmb0l). current slave
status:Slave_IO_State:Waiting for master to send event Master_Host:192.168.210.41
```

```

Master_User:hotdb_datasource Master_Port:3310 Connect_Retry:60 Master_Log_File:mysql-bin.000002
Read_Master_Log_Pos:3871570 Relay_Log_File:mysql-relay-bin.000006 Relay_Log_Pos:3871783
Relay_Master_Log_File:mysql-bin.000002 Slave_IO_Running:Yes Slave_SQL_Running:Yes Replicate_Do_DB:
Replicate_Ignore_DB: Replicate_Do_Table: Replicate_Ignore_Table: Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table: Last_Error:0 Last_Error: Skip_Counter:0 Exec_Master_Log_Pos:3871570
Relay_Log_Space:5058376 Until_Condition:None Until_Log_File: Until_Log_Pos:0 Master_SSL_Allowed:No
Master_SSL_CA_File: Master_SSL_CA_Path: Master_SSL_Cert: Master_SSL_Cipher: Master_SSL_Key:
Seconds_Behind_Master:0 Master_SSL_Verify_Server_Cert:No Last_IO_Errno:0 Last_IO_Error:
Last_SQL_Errno:0 Last_SQL_Error: Replicate_Ignore_Server_Ids: Master_Server_Id:210413310
Master_UUID:919cbf03-9f2d-11e8-b8af-525400636cd2 Master_Info_File:mysql.slave_master_info SQL_Delay:0
SQL_Remaining_Delay:NULL Slave_SQL_Running_State:Slave has read all relay log; waiting for more updates
Master_Retry_Count:86400 Master_Bind: Last_IO_Error_Timestamp: Last_SQL_Error_Timestamp:
Master_SSL_Crl: Master_SSL_Crlpath: Retrieved_Gtid_Set:919cbf03-9f2d-11e8-b8af-525400636cd2:22735-
1367727 Executed_Gtid_Set:1aef7172-9f2e-11e8-b62c-525400fcfb5b: 1-3281,919cbf03-9f2d-11e8-b8af-
525400636cd2:1-1367727 Auto_Position:1 Replicate_Rewrite_DB: Channel_Name: Master_TLS_Version:
2018-10-26 19:30:24.407 [WARN] [FAILOVER] [Labor-484] BackendDataNode(726) - datanode 20 switch
datasource 24 to 25 in failover. due to: Manual Switch by User: root

2018-10-26 19:30:24.408 [INFO] [FAILOVER] [Labor-484] BackendDataNode(762) -
datasource:[id:24,nodeld:20 192.168.210.41:3310/rmb01 status:1,charset:utf8mb4] will be set to unavailable due to
datanode switch datasource.

2018-10-26 19:30:24.410 [INFO] [FAILOVER] [Thread-55] BackendDataNode(834) - starting updating
datasource_status in failover of datanode:20

2018-10-26 19:30:24.415 [INFO] [FAILOVER] [Labor-484] SwitchDataSource(94) - switch datasource:24 for
datanode:20 successfully by Manager.

```

14.1.41. enableXA

Description of parameter:

Property	Value
Parameter value	enableXA

Property	Value
Visible or not	Yes
Description of parameters	Apply XA TRANSACTION or not
Default value	False
Whether Reload is valid or not	No
Min Compatible Version	2.4.3

Role of parameter:

XA mode refers to strong consistency mode. In distributed database system, after data sharding, the same transaction will operate multiple data nodes, and generate cross-node transactions. In cross-node transactions, after the transaction being committed, if the transaction is successfully COMMIT in one of the data node, while COMMIT fails in the other data node; for the data node with COMMIT operation completed, the data has been persistent and can no longer be modified; while for the data node whose COMMIT fails, the data has been lost, thus resulting in data inconsistency among data nodes.

Using the external XA TRANSACTION provided by MySQL, the compute node could solve data strong consistency in all cross-node transaction scenario: either COMMIT transaction of all nodes, or ROLLBACK all nodes. Please refer to the [Data Strong Consistency \(XA Transaction\) Chapter](#)

14.1.42. errorsPermittedInTransaction

Description of parameter:

Property	Value
Parameter value	errorsPermittedInTransaction
Visible or not	Yes
Description of parameters	Whether error is allowed in transaction or not
Default value	False
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Role of parameter:

When set as False, execute SQL in transaction, and after MySQL returns error, the transaction is no longer allowed to be operated, and could only be rolled back. All operations which may result in transaction COMMIT will result in rollback of this transaction. When set as True, ignore the error in transaction.

When set as False, and MySQL returns error, there will be prompt as follow:

```
mysql> begin;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> insert into autoi values(null,'aa');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from autoi;

ERROR 1146 (HY000): Table 'db249.autoi' doesn't exist

mysql> select * from autoi where id=1;

ERROR 1003 (HY000): errors occurred in transaction, you need to rollback now

mysql> commit;

Query OK, 0 rows affected (0.00 sec)

mysql> select * from autoi where id=1;

Empty set (0.00 sec)
```

When set as true, even if there is error in transaction, the transaction can still be successfully committed.

```
mysql> begin;

Query OK, 0 rows affected (0.00 sec)

mysql> insert into autoi values(null,'aa');

Query OK, 1 row affected (0.00 sec)

mysql> select * from ss;

ERROR 1146 (HY000): Table 'db249.ss' doesn't exist

mysql> select * from ss where id=1;

+----+----+
| id | a   |
+----+----+
|   1 | aa  |
+----+----+
```

```
+-----+
1 row in set (0.01 sec)
```

```
mysql> commit;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select * from ss where id=1;
+-----+
| id | a |
+-----+
| 1 | aa |
+-----+
1 row in set (0.01 sec)
```

14.1.43. failoverAutoresetslave

Description of parameter:

Property	Value
Parameter value	failoverAutoresetslave
Visible or not	Yes
Description of parameters	When failover, auto reset the master/slave replication relation or not
Default value	false
Whether Reload is valid or not	No

Property	Value
Min Compatible Version	2.5.3

Parameter Setting:

```
<property name="failoverAutoresetslave">false</property><!-- When failover, auto reset the master/slave
replication relation or not -->
```

Role of parameter:

This parameter is used for guaranteeing data accuracy after data source failover. When enabled, after failover, it will suspend the IO thread between the original Master/Slave, and after the original Active Master recovers normal, check whether there is still transaction not received after the original Standby Slave (the existing Active Master) catching up with replication; if exists, then auto reset the master/slave replication relation. Please refer to [Data accuracy guarantee after failover](#) for details.

14.1.44. frontConnectionTrxIsoLevel

Description of parameter:

Property	Value
Parameter value	frontConnectionTrxIsoLevel
Visible or not	No
Description of parameters	Front-end connection default isolation level
Default value	2

Property	Value
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.5

Parameter Setting:

```
<property name="frontConnectionTrxIsoLevel">2</property>
```

Role of parameter:

It's used for setting user isolation level of front-end connection of the compute node, there are four isolation levels to be selected

```
0=read-uncommitted;1= read-committed;2= repeatable-read;3=serializable
```

14.1.45. frontWriteBlockTimeout

Description of parameter:

Property	Value
Parameter value	frontWriteBlockTimeout
Visible or not	Yes
Description of parameters	Front-end Connection Write Block Timeout
Default value	10000ms
Min value	2000ms
Max value	60000ms
Whether Reload is valid or not	N for v.2.4.5

Property	Value
	Y for v.2.4.7 and above
Min Compatible Version	2.4.5

Role of parameter:

When Front-end Connection Write Block Timeout, it will disable front-end connection, and output corresponding log prompting “closed, due to write block timeout”, as follow:

Great network latency or unreachable network from the compute node to the client, slow data receiving of the client, etc. may lead to front-end write block.

```
2018-06-14 13:46:48.355 [INFO] [] [TimerExecutor1] FrontendConnection(695) -
[thread=TimerExecutor1,id=9,user=cara,host=192.168.200.82,port=8883,localport=61893,schema=TEST_LGG]
closed, due to write block timeout, executing SQL: select * from customer_auto_1
```

14.1.46. generatePrefetchCostRatio

Description of parameter:

Property	Value
Parameter value	generatePrefetchCostRatio
Visible or not	No
Description of parameters	触发提前预取的已消耗比例
Default value	90
Min value	50
Max value	100

Property	Value
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.4

参数设置:

```
<property name="generatePrefetchCostRatio">70</property>
```

参数作用:

隐藏参数，配置批次已消耗比例，已消耗比例是指当前自增值占当前批次大小的比例，例如当前自增值为 89，当前批次大小为 100，则已消耗比例为 89%。

若批次使用率达到已消耗比例，则会触发提前预取新的批次。例如参数设置为 70，若批次使用率达到 70%，则开始预取下一批次。

14.1.47. globalUniqueConstraint

Description of parameter:

Property	Value
Parameter value	globalUniqueConstraint
Visible or not	No
Description of parameters	Whether enable Global Unique Constraint for the new tables by default or not
Default value	false
Whether Reload is valid or not	N for v.2.4.5

	Y for v.2.4.7 and above
Min Compatible Version	2.5.2

Parameter Setting:

globalUniqueConstraint parameter configuration in Server.xml is configured as follow:

```
<property name="globalUniqueConstraint">false</property><!--Whether enable Global Unique Constraint for  
the new tables by default or not-->
```

Role of parameter:

This parameter is different from the 2.5.2 globalUniqueConstraint parameter function, please pay attention. The parameter in 2.5.3 means that whether enable Global Unique Constraint for the new tables by default or not. After set as true, Global Unique Constraint could be enabled by default for the added table. For details, please refer to [Global Unique Constraint](#).

To Enable Global Unique Constraint guarantees that the column with Unique Constraint (UNIQUE, PRIMARY KEY) is unique on all data nodes. Notice: after enabling this function, there may be great influence on execution efficiency of the SQL statements such as INSERT, UPDATE, DELETE, which may result in increasing latency of SQL operation; and it may also increase lock wait and deadlock.

14.1.48. haMode

Description of parameter:

Property	Value

Parameter value	haMode
Visible or not	Yes
Description of parameters	High-availability mode, 0:HA, 1:Cluster, 2:HA in master center, 3:HA in DR center 4:Cluster in master center, 5:Cluster in DR center
Default value	0
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.0

Parameter Setting:

haMode parameter configuration in Server.xml is configured as follow:

```
<property name="haMode">0</property><!-- High-availability mode, 0:HA, 1:Cluster, 2:HA in master center,  
3:HA in DR center 4:Cluster in master center, 5:Cluster in DR center -->
```

Role of parameter:

In HotDB Server 2.5.3.1 and below, the default value of haMode is 0. When this parameter is set as 0, it means that the current compute node cluster uses single-node or high availability mode, and the relevant parameters of the cluster can be ignored. When this parameter is set as 1, the relevant parameters of the cluster are required and the compute node will run in cluster mode.

In HotDB Server 2.5.3.1 and above, haMode can be set as 0,1,2,3. For the compute node cluster in the single-IDC mode, the same method is used as that in the lower version. Setting the haMode as 0 or 1 represents the single node, high availability and cluster mode in the single-IDC mode. For the compute node cluster in the DR mode, setting this parameter as 2 in the master center and 3 in the DR center represents the single node or high availability mode in the DR mode. Cluster mode is not supported for compute node cluster in DR mode.

In HotDB Server 2.5.6 and above, haMode can be set to 0,1,2,3,4,5. Among them, 4 means the master center in multi-node mode after the DR mode is enabled; 5 means the DR center in multi-node mode after the DR mode is enabled.

14.1.49. haState & haNodeHost

Description of parameter:

Property	Value
Parameter value	haState
Visible or not	Yes
Description of parameters	Master node: master; Slave node: backup
Default value	master
Whether Reload is valid or not	No
Min Compatible Version	2.4.3

Property	Value
Parameter value	haNodeHost

Property	Value
Visible or not	Yes
Description of parameters	HA role, Other node IP:PORT
Default value	(Null)
Whether Reload is valid or not	No
Min Compatible Version	2.4.3

hastate and haNodeHost are supporting parameters. When hastate is master node, haNodeHost is null; when hastate is slave node, haNodeHost could be configured as connection information of management port of end node, i.e. IP:PORT. This group of parameters are applicable to Compute Node High Availability environment, while this parameter could be ignored in compute node service. For details, please refer to [Distributed Transactional Database HotDB Server \[Install Deploy\] Function Manual](#). If the cluster mode haMode is enabled as 1, then this parameter shall set other node IP:PORT; PORT is the communication port, and separate multiple nodes by comma (refer to Reference Value Setting).

Role of parameter:

This parameter is only valid for compute node of backup role under high availability mode. When the backup compute node is triggered online by keepalived, it will actively send offline command to the original Master service on haNodeHost, in order to reduce multi-activity scenarios as much as possible.

For example, 192.168.200.51:3325 and 192.168.200.52:3325 belong to Compute Node High Availability environment, and this group of parameters are key configuration for the user in using Compute Node High Availability relation; master compute node plays the role as master, while standby compute node as backup, and it needs to specify and

configure IP and management port of master service related with it.

For example, 192.168.210.22:3326,192.168.210.23:3326 and 192.168.210.24:3326 belong to multiple compute nodes, and it needs to specify and configure IP and Communication Port of master service related with it.

```
<property name="haState">master</property><!--HA role, Master node: master, standby node: backup (invalid in cluster mode) -->

<property name="haNodeHost"/><!-- HA role, Other node IP:PORT (in Master/Standy mode, PORT means management port, for example: 192.168.200.2:3325) -->

<property name="haState">backup</property><!-- HA role, Master node: master, standby node: backup (invalid in cluster mode) -->

<property name="haNodeHost"/>192.168.200.51:3325<!-- HA role, Other node IP:PORT (in Master/Standy mode, PORT means management port, for example: 192.168.200.2:3325) -->

<property name="haState">backup</property><!-- HA role, Master node: master, standby node: backup (invalid in cluster mode) -->

<property name="haNodeHost"/>192.168.210.23:3326,192.168.310.24:3326<!-- HA role, Other node IP:PORT (in Master/Standy mode, PORT means management port, for example: 192.168.200.2:3325) -->
```

14.1.50. highCostSqlConcurrency

Description of parameter:

Property	Value
Parameter value	highCostSqlConcurrency
Visible or not	No
Description of parameters	Number of high cost statement concurrencies
Default value	32
Min value	0

Property	Value
Max value	1024
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.3

Role of parameter:

This parameter is the compute node overload protection related parameter, which is used for controlling number of high cost statement concurrencies (including cross-node join, union, update/delete...limit, etc.). When front-end execution concurrencies exceed the setting, relevant connection will Hold, and wait for completion of previous execution, then the next batch could be executed.

Flow control in Show processlist is in lock status, waiting for execution of the next batch.

You could view the remaining concurrencies available at present from the management port.

Id	User	Host	db	Command	Time	State	Info
150	_HotDB_Cluster_USER_	192.168.210.31:51428	TEST_LGG	Query	0	Sending data	select a.* , b.x from customer_auto_1 a join customer_auto_2 on ...Omitted
126	_HotDB_Cluster_USER_	192.168.210.31:51412	TEST_LGG	Query	0	Flow control	select a.* , b.x from customer_auto_1 a join customer_auto_2 on ...Omitted
222	_HotDB_Cluster_USER_	192.168.210.32:16636	TEST_LGG	Query	0	optimizing	select a.* , b.x from customer_auto_1 a join customer_auto_2 on ...Omitted

```

| 174 | _HotDB_Cluster_USER_ | 192.168.210.32:16604 | TEST_LGG | Query | 0 | Sending data | select
a.* , b.x from customer_auto_1 a join customer_auto_2 on ...Omitted |

| 129 | _HotDB_Cluster_USER_ | 192.168.210.31:51414 | TEST_LGG | Query | 0 | Flow control | select
a.* , b.x from customer_auto_1 a join customer_auto_2 on ...Omitted |

...More are omitted...

```

mysql> show @@debug;

```

+-----+-----+
| join_limit | committing |
+-----+-----+
| 32 | 0 |
+-----+-----+

```

1 row in set (0.00 sec)

14.1.51. idcId & idcNodeHost

Description of parameter:

Property	Value
Parameter value	idcId
Visible or not	Yes
Description of parameters	ID of IDC, 1:master center,2:DR center
Default value	0
Whether Reload is valid or not	Yes

Property	Value
Min Compatible Version	2.5.3.1

Property	Value
Parameter value	idcNodeHost
Visible or not	Yes
Description of parameters	connection information of another IDC
Default value	192.168.200.1:3325,1 92.168.200.1:3325
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.3.1

Role of parameter:

When the DR mode is enabled, the parameters idcId and idcNodeHost need to be configured. For idcId, the ID of IDC, the current default setting is 1 for the master center and 2 for the DR center; for idcNodeHost, the connection information of all compute nodes in the other IDC, the configuration format is IP:PORT. The compute nodes are separated by English commas, for example: 192.168.200.186:3325,192.168.200.187:3325.

For example, set idcId as 1 in server.xml of the master center, idcNodeHost for all the compute node information of the DR center; set idcId as 2 in server.xml of the DR center, idcNodeHost for all the compute node information of the master center.

```
<property name="idcId">2</property><!-- ID of IDC, 1:master center,2:DR center -->
<property name="idcNodeHost">192.168.220.188:3325,192.168.220.189:3325</property><!-- connection
information of another IDC-->
```

14.1.52. idleTimeout

Description of parameter:

Property	Value
Parameter value	idleTimeout
Visible or not	No
Description of parameters	Front-end idle connection timeout time
Default value	28800
Whether Reload is valid or not	No
Min Compatible Version	2.4.3

Parameter Setting:

In server.xml, the parameter idleTimeout is configured as follows:

```
<property name="idleTimeout">28800</property><!-- Front-end idle connection timeout time,
unit: seconds -->
```

Role of parameter:

This parameter is used to detect the timeout time of idle connections at the front end. If Time of connection in the "sleep" state at the front-end exceeds the set value, HotDB will close the idle connection. When the parameter is set to 0, it means that the current front-end idle connection never timeout.

To facilitate the demonstration, the value is set to 60 seconds in the test.

```
mysql > show processlist;

+-----+-----+-----+-----+-----+
| Id | User | Host | db | Command | Time | State |
+-----+-----+-----+-----+-----+
| Info |          |
+-----+-----+-----+-----+-----+
| 9 | root | 192.168.220.211:26568 | NULL | Query | 0 |
executing | show processlist |

| 7 | ztm | 192.168.220.211:26470 | INFORMATION_SCHEMA | Sleep | 59 |
| NULL |          |
+-----+-----+-----+-----+-----+
|          |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql > show processlist;

+-----+-----+-----+-----+-----+
| Id | User | Host | db | Command | Time | State |
+-----+-----+-----+-----+-----+
| Info |          |
+-----+-----+-----+-----+-----+
|          |
+-----+-----+-----+-----+-----+
```

At this time, the front-end connection session time outs. Entering SQL in, it will prompt that the connection has been disconnected, and try to reconnect. Finally, the reconnection is successful:

```
mysql> show databases;
```

DATABASE

```
+-----+
| INFORMATION_SCHEMA |
+-----+
1 row in set (0.00 sec)

mysql> show databases;

ERROR 2013 (HY000): Lost connection to MySQL server during query

ERROR 2006 (HY000): MySQL server has gone away

No connection. Trying to reconnect...

Connection id:      10

Current database: INFORMATION_SCHEMA
```

If it is set to 0, the front-end idle connection will never time out, and the connection time in sleep status will increase all the time.

14.1.53. joinable

Description of parameter:

Property	Value
Parameter value	joinable
Visible or not	Yes
Description of parameters	Allow JOIN Query or not, Yes: true, No: false
Default value	true
Whether Reload is valid or not	Yes

Property	Value
Min Compatible Version	2.4.3

Role of parameter:

This parameter could limit some SQL statement execution, including cross-node join queries and single-node join queries among sharding table that are judged to be distributed by sharding key. Under Global Table join and Vertical Sharding Table join, there will be no corresponding limit when this parameter is enabled. When this parameter is disabled , the execution of some join statements will be restricted, including cross-node join queries and some single-node join queries among sharding table that are judged to be distributed by sharding key.

Set JOIN Query as false, and execute statement under this environment. Report ERROR 1064 (HY000): joinable is not configured.

```
mysql> select * from join_cross_a_jwy a inner join join_cross_b_jwy b on a.adnid between 108 and 110;
```

```
ERROR 1064 (HY000): joinable is not configured.
```

```
mysql> select a.addept from join_a_jwy a join join_b_jwy b on a.addept=b.bdept limit 5;
```

```
ERROR 1064 (HY000): joinable is not configured.
```

Set JOIN Query as true, and execute the statement under this environment:

```
mysql> select a.addept from join_a_jwy a join join_b_jwy b on a.addept=b.bdept limit 5;
```

```
+-----+
```

```
| adept |
```

```
+-----+
```

```
| aa      |
| bb      |
| cc      |
+-----+
3 rows in set (0.03 sec)
```

14.1.54. joinBatchSize

Description of parameter:

Property	Value
Parameter value	joinBatchSize
Visible or not	Yes
Description of parameters	At equi-join query, record number of equi-join queries turning into IN queries per batch
Default value	1000
Min value	100
Max value	100000
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Role of parameter:

At equi-join query, record number of equi-join queries turning into IN queries per batch.

The value exceeding the setting will be transferred in several times. This parameter

belongs to optimization parameter of join Query, and it could improve the join Query speed. For example:

```
<property name="joinBatchSize">3</property><!—At equi-join query, record number of equi-join queries turning
into IN queries per batch -->
```

At this time, execute:

```
mysql> select b.* from customer_auto_1 a join customer_auto_3 b on a.id=b.id where a.postcode=123456;
```

View actual execution result of `general_log` as follow:

```
1993 Query SELECT B.`ID`, B.`name`, B.`telephone`, B.`provinceid`, B.`province`,
B.`city`, B.`address`, B.`postcode`, B.`birthday`, b.id AS `hotdb_tmp_col_alias_1` FROM customer_auto_3 AS b
WHERE B.ID IN (4064622, 4068449, 4071461)
```

```
1993 Query SELECT B.`ID`, B.`name`, B.`telephone`, B.`provinceid`, B.`province`,
B.`city`, B.`address`, B.`postcode`, B.`birthday`, b.id AS `hotdb_tmp_col_alias_1` FROM customer_auto_3 AS b
WHERE B.ID IN (4043006, 4053408, 4056542)
```

...More are omitted...

Notice: The parameter value is for illustration only, not for practical reference.

14.1.55. joinCacheSize

Description of parameter:

Property	Value
Parameter value	joinCacheSize
Visible or not	No

Property	Value
Description of parameters	Off-heap memory size of JOIN cache (M)
Default value	32
Min value	0
Max value	128
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Role of parameter:

It presents available direct memory size of join, and could influence speed of JOIN query cache.

When direct memory used by join exceeds the set value, it will be stored on local disk temporarily, and after execution of join statement completed, the temporary file will be deleted automatically.

```
root> pwd
/usr/local/hotdb-2.4.9/hotdb-server/HotDB-TEMP

You have mail in /var/spool/mail/root

root> ll
-rw-r--r-- 1 root root 8778410 May  9 17:28 positions_5302007528422328273.tmp
-rw-r--r-- 1 root root 141868981 May  9 17:28 row_411809270296834018.tmp
-rw-r--r-- 1 root root 26113612 May  9 18:01 row_4342139033645193593.tmp
```

14.1.56. joinLoopSize

Description of parameter:

Property	Value
Parameter value	joinLoopSize
Visible or not	Yes
Description of parameters	JOIN Query times per batch of each node when using BNL algorithm
Default value	1000
Min value	100
Max value	10000
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Role of parameter:

JOIN Query times per batch of each node when using BNL algorithm belongs to optimization parameter of join Query, and it could improve the join Query speed.

```
<property name="joinLoopSize">1000</property><!-- 使用 BNL 算法做 JOIN 时各节点每批次查询数量 -->
<property name="joinLoopSize">1000</property><!-- JOIN Query times per batch of each node when using BNL
algorithm -->
```

For example: joinLoopSize is set as 1000. bn_a_jwy is autoSharding Table, sharding key is id; bn_b_jwy is matchSharding Table, sharding key is a; bn_c_jwy is autoSharding Table, sharding key is a, and the data amount of all the three tables is 2w.

```
mysql> select * from bn_a_Jwy as a inner join bn_b_jwy as b on a.a=b.a limit 9000;
```

View actual execution result of general_log:

```
1187022 Query    SELECT A.id, A.a, A.bchar, A.cdeci, A.dtime FROM bn_a_jwy AS a ORDER BY A.ID
LIMIT 1001

1187022 Query    SELECT C.id, C.a, C.bchar, C.cdeci, C.dtime FROM bn_c_jwy AS c
WHERE C.id IN (0) ORDER BY C.ID LIMIT 0 , 1001

1187022 Query    SELECT B.id, B.a, B.bchar, B.cdeci, B.dtime FROM bn_b_jwy AS b
WHERE B.a COLLATE utf8_general_ci IN ('d') ORDER BY B.ID LIMIT 0 , 1001      ...More are omitted...
```

14.1.57. keyStore

Description of parameter:

Property	Value
Parameter value	keyStore
Visible or not	Yes
Description of parameters	Path to the data certificate .jks file for TLS connection
Default value	/server.jks
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.5

Parameter setting:

```
<property name="keyStore">/server.jks</property><!-- Path to the data certificate .jks file
for TLS connection -->
```

Role of parameter:

This parameter is used to set the path to the data certificate .jks file for connection using SSL authentication. For details, please refer to the [TLS connection login](#), and use it together with parameters [enableSSL](#) and [keyStorePass](#).

14.1.58. keyStorePass

Description of parameter:

Property	Value
Parameter value	keyStorePass
Visible or not	Yes
Description of parameters	Password of the data certificate .jks file for TLS connection
Default value	BB5A70F75DD5FEB214A5623DD171CEE B
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.5

Parameter setting:

```
<property name="keyStore">/server.jks</property><!-- Password of the data certificate .jks  
file for TLS connection -->
```

Role of parameter:

This parameter is used to set the password of the data certificate .jks file for connection using SSL authentication. For details, please refer to the [TLS connection login](#), and use it together with parameters [enableSSL](#) and [keyStore](#).

14.1.59. lockWaitTimeout

Description of parameter:

Property	Value
Parameter value	lockWaitTimeout
Visible or not	Yes
Description of parameters	Timeout for obtaining metadata lock (s)
Default value	31536000
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.3
Max value	31536000
Min value	1

Parameter Setting:

This parameter is used to set the timeout (s) for obtaining metadata lock. The value range is 1-31536000s, and the default value is 31536000s, i.e. 365 days, representing if the metadata lock timeout exceeds 365 days, the client will prompt the lock timeout.

```
<property name="lockWaitTimeout">31536000</property> <!-- Timeout for obtaining metadata lock -->
```

session A execute:

```
jing01@127.0.0.1:zjj_db1 5.7.23 01:41:44> begin;select * from aaa limit 10;
Query OK, 0 rows affected (0.00 sec)
```

session B execute: if the set value of lockWaitTimeout is exceeded, the following prompt will be given:

```
jing01@127.0.0.1:zjj_db1 5.7.23 11:57:45> alter table aaa drop name;
ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting transaction
jing01@127.0.0.1:zjj_db1 5.7.23 11:59:16>
```

14.1.60. masterSourceInitWaitTimeout

Description of parameter:

Property	Value
Parameter value	masterSourceInitWaitTimeout
Visible or not	No
Description of parameters	When start, Master Data Source Initialization Timeout in data node
Default value	300
Min value	0
Max value	600
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Role of parameter:

When start, the Master data source will reconnect continuously after initial initialization failure; in case of existing standby data source and initialization timeout time of the Master data source is exceeded, it will switch to available standby data source; in case of initialization failure with all data sources of this node, then the whole node is

Unavailable. If all nodes of at least one LogicDB in the compute node has Initialized successfully, then the compute node Start succeeded, otherwise, Start failed.

```
2018-05-28 18:07:29.719 [WARN] [INIT] [main] r(-1) – failed in connecting datasource:[id:182,nodeId:11
192.168.220.101:3306/db01 status:1,charset:utf8], exception:...omitted...
```

The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the sever.

```
2018-05-28 18:07:31.719 [INFO] [INIT] [main] b(-1) – try reinit datasource:[id:182,nodeId:11
192.168.220.101:3306/db01 status:1,charset:utf8]
```

Causes for data source timeout are: beyond limit of the system or database connection, authentication failure of data source user password, great network latency, etc.

14.1.61. maxAllowedPacket

Description of parameter:

Property	Value
Parameter value	maxAllowedPacket
Visible or not	No
Description of parameters	Max data packet allowed to be received
Default value	65536
Min value	1
Max value	1048576
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.5

Role of parameter:

Control packet size sent from the front-end connection. 64M by default, when the SQL statement size sent exceeds the default value 64M, the compute node will give prompt (Get a packet bigger than 'max_allowed_packet').

ERROR 1153 (HY000): Get a packet bigger than 'max allowed packet'

Meanwhile, show variables could show the configuration value.

```
mysql> show variables like '%allowed%';

+-----+-----+
| variable_name      | value           |
+-----+-----+
| max_allowed_packet | 16777216        |
+-----+-----+
```

14.1.62. maxConnections & maxUserConnections

Description of parameter:

Property	Value
Parameter value	maxConnections
Visible or not	Yes
Description of parameters	Front-end Max Connections
Default value	5000
Min value	300000

Property	Value
Max value	1
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.4

Property	Value
Parameter value	maxUserConnections
Visible or not	Yes
Description of parameters	User Front-end Max Connections, 0 means no limit
Default value	0
Min value	300000
Max value	0
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.4

Role of parameter:

The compute node supports the Front-End Connections Control function, which could provide guarantee in case of access overload.

maxConnections is the Max Connections allowed for user front-end connection, and is the upper limit of simultaneous sessions allowed by the compute node. The user could set maxConnections according to actual needs, adjust this value appropriately, but can't increase the set value blindly.

maxUserConnections is commonly known to be the Max Connections of the same account to the compute node simultaneously. User Front-end Max Connections could be null, and 0 is given by default in case of null, meaning no limit for user connections, and at this time, this connection shall be subject to the front-end Max Connections.

When number of connections exceeds the set value, if executing front-end connection, there will be prompt as follow:

```
root> mysql -uzy -pzy -h127.0.0.1 -P9993
Warning: Using a password on the command line interface can be insecure.
ERROR 1040 (HY000): too many connections
```

The value of maxConnections and maxUserConnections could be modified via Set, the parameters are at GLOBAL level:

```
mysql> set global max_connections = 5000;
Query OK, 0 rows affected (0.00 sec)

mysql> show variables like '%max_connections%';
+-----+-----+
| variable_name | value |
+-----+-----+
| max_connections | 5000 |
+-----+-----+

mysql> set global max_user_connections = 1000;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show variables like "%max_user_connections%;
```

```
+-----+-----+
| variable_name          | value      |
+-----+-----+
| max_user_connections    | 1000       |
+-----+-----+
```

14.1.63. maxIdleTransactionTimeout

Description of parameter:

Property	Value
Parameter value	maxIdleTransactionTimeout
Visible or not	Yes
Description of parameters	Non-committed Idle Transaction Timeout (ms)
Default value	86400000
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.1

Parameter Setting:

```
<property name="maxIdleTransactionTimeout">86400000</property>
```

maxIdleTransactionTimeout parameter has the default value of 86400000ms, that is 24h, which means that when the last SQL is completed in the transaction, the transaction fails to be submitted within 24h, it will be judged as timeout transaction, and HotDB records connection IP, port, username, LogicDB, lastsql, whether to autocommit or not,

innodb_trx of back-end connection and other information with the mark [INFO] [WATCHDOG] WatchDogLongTransactionCheckHandler in hotdb.log, and disable the connection, and auto rollback the transaction.

The parameter is only valid when enableWatchdog=true. In Watchdog, maxIdleTransactionTimeout is checked every 10 mins. Idle time of the transaction connected is judged in next check of Watchdog toward maxIdleTransactionTimeout; if exceeding the set threshold value, disable the connection, therefore, the practical transaction Idle time doesn't equal to the set threshold value.

For example, if the transaction Idle time exceeds the set threshold value, then disable the connection, and at this time, view log:

```
2019-07-01 18:09:24.528 [INFO] [WATCHDOG] [$NIOREACTOR-20-RW]
cn.hotpu.hotdb.mysql.nio.handler.WatchDogLongTransactionCheckHandler(123) - Session [thread=Thread-13,id=1,user=ztm,host=127.0.0.1,port=3323,localport=46138,schema=PM] has not been queried for 593s.
executed IUDs:[UPDATE customer_auto_1 SET city = 'xxxx' WHERE id = 1]. binded
connection:[MySQLConnection [node=2, id=59, threadId=14921, state=borrowed, closed=false,
autocommit=false, host=10.10.0.202, port=3307, database=db_test251, localPort=52736, isClose:false,
toBeClose:false] lastSQL:SET autocommit=0;UPDATE customer_auto_1 SET city = 'xxxx' WHERE id = 1].
innodb_trx:[(ds:2 trx_id:3435056156 trx_state:RUNNING trx_started:2019-07-01 17:59:33
trx_requested_lock_id:NULL trx_wait_started:NULL trx_weight:3 trx_mysql_thread_id:14921 trx_query:NULL
trx_operation_state:NULL trx_tables_in_use:0 trx_tables_locked:1 trx_lock_structs:2
trx_lock_memory_bytes:1136 trx_rows_locked:1 trx_rows_modified:1 trx_concurrency_tickets:0
trx_isolation_level:REPEATABLE READ trx_unique_checks:1 trx_foreign_key_checks:1
trx_last_foreign_key_error:NULL trx_adaptive_hash_latched:0 trx_adaptive_hash_timeout:0 trx_is_read_only:0
trx_autocommit_non_locking:0 )]. we will close this session now.
```

When the parameter is set as 0, it means never timeout, that is, no limit for COMMIT time of the transaction.

14.1.64. maxJoinSize

Description of parameter:

Property	Value
Parameter value	maxJoinSize
Visible or not	Yes
Description of parameters	Row limits in JOIN query cache (M: Million, K: Thousand)
Default value	10M
Min value	1K
Max value	1000M
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Role of parameter:

Max rows allowed for Join query cache. Computation method of JOIN query cache is: when there is no condition in SQL statement, compute the Cartesian product; when there is condition in SQL statement, compute the rows meeting join condition.

When the JOIN query cache has rows greater than the set value, the message will be prompted as follow:

```
mysql> select * from customer_auto_1 a join customer_auto_3 b on a.postcode=b.postcode;
ERROR 1104 (HY000): The SELECT would examine more than MAX_JOIN_SIZE rows; check your
maxJoinSize in server.xml
```

The current session parameter value could be modified via set session max_join_size, to make the JOIN query cache within 1~ 2124000000:

```
mysql> show variables like '%max_join_size%;
```

variable_name	value

```
+-----+-----+
| max_join_size | 5000 |
+-----+-----+
```

```
mysql> set global max_user_connections = 1000;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show variables like '%max_user_connections%';
```

```
+-----+-----+
| variable_name | value |
+-----+-----+
| max_user_connections | 2124000000 |
+-----+-----+
```

```
mysql> set session max_join_size=1;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show variables like '%max_user_connections%';
```

```
+-----+-----+
| variable_name | value |
+-----+-----+
| max_user_connections | 1 |
+-----+-----+
```

```
mysql> select * from bn_a_jwy a join bn_c_jwy b on a.a=b.a where a.a='d';
```

ERROR 1104 (HY000): The SELECT would examine more than MAX_JOIN_SIZE rows; check your maxJoinSize in server.xml

14.1.65. maxLatencyForRWSplit

Description of parameter:

Property	Value
Parameter value	maxLatencyForRWSplit
Visible or not	Yes
Description of parameters	Max Latency of Readable Standby Slave in Read/write splitting
Default value	1000ms
Min value	200
Max value	10000
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.5

Role of parameter:

After Read/write splitting is enabled, when the master/slave latency is smaller than the set latency time, read the Standby Slave:

```
mysql> select * from cd;
```

```
+-----+-----+
```

```
| id | name |
```

```
+-----+-----+
```

```
| 1 | slave |
```

```
| 2 | slave |
```

```
| 3 | slave |
```

```
| 4 | slave |
```

```
| 5 | slave |
```

```
+-----+-----+
```

```
5 rows in set (0.00 sec)
```

After Read/write splitting is enabled, when latency of readable Standby Slave exceeds the set time, it will read the Active Master:

```
mysql> select * from cd;
```

```
+-----+-----+
```

```
| id | name |
```

```
+-----+-----+
```

```
| 1 | master |
```

```
| 2 | master |
```

```
| 3 | master |
```

```
| 4 | master |
```

```
| 5 | master |
```

```
+-----+-----+
```

```
5 rows in set (0.00 sec)
```

14.1.66. maxNotInSubquery

Description of parameter:

Property	Value
Parameter value	maxNotInSubquery
Visible or not	Hidden
Description of parameters	Max number of not in in subquery
Default value	20000
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.9

Parameter Setting:

maxNotInSubquery parameter in server.xml is configured as follow:

```
<property name="maxNotInSubquery">20000</property><!-- Max number of not in in subquery -->
```

Role of parameter:

It's used to control max number of not in in subquery, which is 20000 by default, and when max number of not in in subquery of SQL statement executed exceeds the default value 20000, HotDB will limit execution of this SQL, and give ERROR prompt

```
(ERROR 1104 (HY000): The sub SELECT would examine more than maxNotInSubquery rows; check your maxNotInSubquery in server.xml) .
```

For example:

(For the convenience of test, maxNotInSubquery is set as 10)

```
mysql> use pm
Database changed

mysql> show tables;

+-----+
| Tables_in_PM      |
+-----+
| customer_quan_2  |
| customer_route_1 |
| customer_route_2 |
+-----+
3 rows in set (0.00 sec)
```

```
mysql> select * from customer_route_2 a where a.postcode not in (select postcode from customer_route_1 b where b.id > 205119 limit 20);
```

```
ERROR 1104 (HY000): The sub SELECT would examine more than maxNotInSubquery rows; check your
maxNotInSubquery in server.xml
```

The log will record corresponding information with [INFO] [SQL] mark

```
2019-10-08 14:33:41.725 [INFO] [SQL] [$NIOExecutor-3-2] cn.hotpu.hotdb.j.h(2626) - unsupported
subquery:[thread=$NIOExecutor-3-
2,id=152197,user=ztm,host=127.0.0.1,port=3323,localport=49458,schema=PM] AutoCommitTransactionSession
in [thread=$NIOExecutor-3-2,id=152197,user=ztm,host=127.0.0.1,port=3323,localport=49458,schema=PM],
sql:select * from customer_route_2 a where a.postcode not in (select postcode from customer_route_1 b where
b.id > 205119 limit 20), error code:1104, error msg:The sub SELECT would examine more than
maxNotInSubquery rows; check your maxNotInSubquery in server.xml
```

Meanwhile, you could view the configured value in log and 3325 port show @@systemconfig, and this parameter after modification could be valid upon reload.

```
mysql> show @@systemconfig;
+-----+
| config |
+-----+
| {"enableFlowControl":"true","recordSql":"false","defaultMaxLimit":"10000","bakPassword":"hotdb_config","bakUrl":"jdbc:mysql://192.168.220.138:3306/hotdb_config_249ha","managementPort":"3325","heartbeatPeriod":"2","cryptMandatory":"false","password":"hotdb_config","enableCursor":"false","username":"hotdb_config","enableXA":"false","errorsPermittedInTransaction":"true","strategyForRWSplit":"0","enableWatchdog":"false","haNodeHost":"192.168.220.139:3325","maxJoinSize":"9148M","maxNotInSubquery":"10","pingLogCleanPeriodUnit":"0","clientFoundRows":"false","joinCacheSize":"236","enableHeartbeat":"true","url":"jdbc:mysql://192.168.220.138:3306/hotdb_config_249ha","parkPeriod":"100000","maxSqlRecordLength":"4000","joinBatchSize":"46000","enableSubquery":"true","heartbeatTimeoutMs":"500","pingPeriod":"300","joinLoopSize":"18500","VIP":"192.168.220.171","joinable":"true","maxUserConnections":"4900","pingLogCleanPeriod":"1","dataNodeIdleCheckPeriod":"120","deadlockCheckPeriod":"3000","sqlTimeout":"3600","bakUsername":"hotdb_config","enableLatencyCheck":"true","waitSyncFinishAtStartup":"true","checkVIPPeriod":"500","statisticsUpdatePeriod":"0","usingAIO":"0","showAllAffectedRowsInGlobalTable":"false","maxLatencyForRWSplit":"1000","maxConnections":"5000","enableSleep":"false","waitForSlaveInFailover":"true","autoIncrement":"true","processOrExecutor":"4","highCostSqlConcurrency":"400","latencyCheckPeriod":"500","processors":"16","weightForSlaveRWSplit":"50","haState":"master","readOnly":"false","timerExecutor":"4","servicePort":"3323","frontWriteBlockTimeout":"10000","switchoverTimeoutForTrans":"3000"} |
+-----+
1 row in set (0.01 sec)
```

14.1.67. maxReconnectConfigDBTimes

Description of parameter:

Property	Value
Parameter value	maxReconnectConfigDBTimes
Visible or not	No

Description of parameters	Max times of reconnecting ConfigDB
Default value	3
Max value	1000
Min value	0
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.4

Parameter Setting:

In server.xml, maxReconnectConfigDBTimes is configured as follows:

```
<property name=" maxReconnectConfigDBTimes ">3</property><!-- Max times of reconnecting ConfigDB -->
```

Role of parameter:

The parameter can prevent long time consumption for configDB connection during the compute node start, the HA switch, or reloading, and increase the reconnection times of configDB. If the max times of reconnections is exceeded (the default reconnection time is 3*2s), it will automatically switch to connecting from the ConfigDB.

14.1.68. maxSqlRecordLength

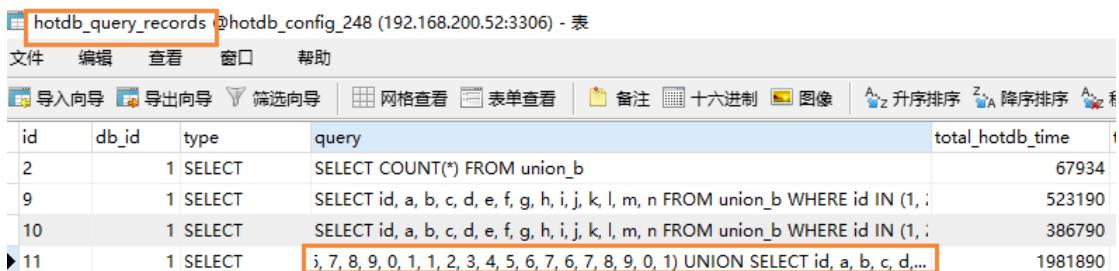
Description of parameter:

Property	Value
Parameter value	maxSqlRecordLength
Visible or not	Yes
Description of parameters	Max length of SQL statement record in SQL execution statistics
Default value	1000
Min value	1000
Max value	16000
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.5

Role of parameter:

This parameter refers to the max length of SQL statistics in Slow Query Log Analysis.

When the length of the executed SQL statement exceeds the set length, it will be cut out automatically, and replaced with ellipsis..., as shown in the following figure:



hotdb_query_records (hotdb_config_248 (192.168.200.52:3306)) - 表				
id	db_id	type	query	total_hotdb_time
2	1	SELECT	SELECT COUNT(*) FROM union_b	67934
9	1	SELECT	SELECT id, a, b, c, d, e, f, g, h, i, j, k, l, m, n FROM union_b WHERE id IN (1, :)	523190
10	1	SELECT	SELECT id, a, b, c, d, e, f, g, h, i, j, k, l, m, n FROM union_b WHERE id IN (1, :)	386790
11	1	SELECT	i, 7, 8, 9, 0, 1, 1, 2, 3, 4, 5, 6, 7, 6, 7, 8, 9, 0, 1) UNION SELECT id, a, b, c, d,...	1981890

14.1.69. `ndbSqlAddr` & `ndbSqlUser` & `ndbSqlPass`

Description of parameter:

Property	Value
Parameter value	ndbSqlAddr
Visible or not	Yes
Description of parameters	NDB SQL-end IP address
Default value	localhost:3329
Whether Reload is valid or not	No
Min Compatible Version	2.5.2

Property	Value
Parameter value	ndbSqlUser
Visible or not	Yes
Description of parameters	NDB SQL front-end username
Default value	root
Whether Reload is valid or not	No
Min Compatible Version	2.5.2

Property	Value
Parameter value	ndbSqlPass
Visible or not	Yes
Description of parameters	NDB SQL front-end password
Default value	root
Whether Reload is valid or not	No
Min Compatible Version	2.5.2

Parameter Setting:

ndbSqlAddr, ndbSqlUser, ndbSqlPass are supporting parameters: ndbSqlAddr is physical address of NDB SQL node; ndbSqlUser and ndbSqlPass are respectively username and password for connecting NDB SQL node.

```
<property name="ndbSqlAddr">localhost:3329</property>

<property name="ndbSqlUser">root</property>

<property name="ndbSqlPass">root</property>
```

14.1.70. ndbSqlDataAddr**Description of parameter:**

Property	Value
Parameter value	ndbSqlDataAddr
Visible or not	Yes
Description of parameters	IP address and port receiving NDB SQL connection
Default value	127.0.0.1:3327
Whether Reload is valid or not	No
Min Compatible Version	2.5.2

Parameter Setting:

The connection from NDB SQL to compute node, that is the communication port from the server IP and NDB SQL where the compute node resides to the compute node, the default value is 127.0.0.1:3327.

```
property name="ndbSqlDataAddr">127.0.0.1:3327</property>
```

14.1.71. ndbSqlMode

Description of parameter:

Property	Value
Parameter value	ndbSqlMode
Visible or not	Yes
Description of parameters	Mode used by NDB SQL node (NDB execution mode: none: Forbidden NDB function, the default value; local: NDB SQL server and compute node server are on the same computer)
Default value	none
Whether Reload is valid or not	No
Min Compatible Version	2.5.2

Parameter Setting:

none: the default value, representing Forbidden NDB function; local: NDB SQL server and compute node server are on the same computer, execute NDB logic if meeting NDB condition.

```
<property name="ndbSqlMode">none</property>
```

14.1.72. ndbSqlVersion & ndbVersion

Description of parameter:

Property	Value
Parameter value	ndbSqlVersion
Visible or not	Yes
Description of parameters	NDB SQL Version Number
Default value	5.7.24
Whether Reload is valid or not	No
Min Compatible Version	2.5.2

Property	Value
Parameter value	ndbVersion
Visible or not	Yes
Description of parameters	NDB Engine Version Number
Default value	7.5.12
Whether Reload is valid or not	No
Min Compatible Version	2.5.2

Parameter Setting:

ndbSqlVersion and ndbVersion are of corresponding relation, and please refer to official MySQL document for the specific corresponding relation. ndbSqlVersion is 5.7.24 by default, and ndbVersion is 7.5.12 by default. The NDB Engine Version supported by the current compute node is 7.5.4 and above, and if to use NDB version, it's required that the data source version must be 5.7.16 and above.

```
<property name="ndbSqlVersion">5.7.24</property>
<property name="ndbVersion">7.5.12</property>
```

14.1.73. operateMode

Description of parameter:

Property	Value
Parameter value	operateMode
Visible or not	No
Description of parameters	Operating mode of compute node
Default value	0
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.6

Parameter Setting:

server.xml 中 operateMode 参数配置如下：

```
<property name="operateMode">0</property><!-- Operating mode, 0: normal mode, 1: performance mode, 2: debug mode-->
```

Role of parameter:

The parameter controls the operating mode of compute nodes, 0 is normal mode, 1 is performance mode, 2 is debug mode. In normal mode, no other parameters or functions will be changed. In performance mode, the functions related to the following parameters will be forced to disable, while in debug mode, the functions related to the following parameters will be forced to enable.

recordSql

recordSQLSyntaxError

recordCrossDNJoin

recordUNION

recordSubQuery

recordDeadLockSQL

recordLimitOffsetWithoutOrderby

recordSQLKeyConflict

recordSQLUnsupported

recordMySQLWarnings

recordMySQLErrors

recordHotDBWarnings

recordHotDBErrors

recordDDL

recordSQLIntercepted

recordAuditlog

recordSQLForward

recordSqlAuditlog

operateMode is a hidden parameter, and the default mode is normal mode, that is, operateMode=0. When the compute node is started, the corresponding log will be output in hotdb.log as follows:

```
2020-09-11 10:15:34.701 [INFO] [MANAGER] [sNIOExecutor-1-4] cn.hutox.hotpu.hotdb.a(3259) - current sysconfig params:SystemConfig: [VIP=null, adaptiveProcessor=true, aggressive=false, allowRwWithoutReadConsistentId=false, autoIncrement=1, badConnAfterContinueGet=true, badConnAfterFastCheckAllIdle=true, bakUrl=null, bakUrl=l=jdbc:mysql://192.168.160.210.202:3316/hotdb_config?useUnicode=true&characterEncoding=UTF-8, charset=utf8, checkConnLastUsedTime=3000, checkConnValid=true, checkKConnValidTimeOut=500, checkKsParamInterval=6000000, checkUpdate=true, checkVPIPeriod=500, clientFoundRows=false, clusterExecutionTimeOuts=20000, clusterHeartbeatTimeOutMs=5000, clusterHost=192.168.210.203, clusterName=HotDB-Cluster, clusterNetwork=192.168.210.0/24, clusterPacketTimeOutMs=5000, clusterPort=3326, clusterSize=3, clusterStartedPacketTimeOutMs=5000, config_file=null, cryptMandatory=false, dataNodeIndexCheckPeriod=120, deadlockCheckPeriod=0, defaultMaxLimit=10000, defaultSortedLimitSize=2000, drBakUrl=jdbc:mysql://192.168.200.116:3006/hotdb_config?useUnicode=true&characterEncoding=UTF-8, drBakUrl=jdbc:mysql://192.168.200.1:3306/hotdb_config?useUnicode=true&characterEncoding=UTF-8, enableSleep=false, enableCursor=false, enableFlowControl=true, enableHeartbeat=false, enableLatencyCheck=false, enableListener=false, enableSSL=false, enableSleep=false, enableSubquery=true, enableUnsafe=false, enableWatchdog=true, enableXA=true, errorsPermittedInTransaction=true, frontConnectionTxIsolationLevel=2, frontWriteBlockTimeOut=10000, globalUniqueConstraint=false, haMode=1, haModeHost=192.168.210.202:3325, ioinBatchSize=10000, joinCacheSize=32, joinLoopSize=1000, joinSize=1000000, joinable=true, keyStore=/server.jks, latencyCheckPeriod=500, managerPort=3325, masterSourceInitWaitTimeOut=300, maxAllowedPacket=64M, maxConnections=5000, maxIdleTimeForTransaction=864000000, maxJoinSize=1000K, maxLatencyForRWSplit=10000, maxLotInSize=1000, query=20000, maxQsRecordLength=4000, maxUserConnections=4900, maxWritingToNetTimes=120000, operateMode=0, parkPeriod=100000, pingLogCleanPeriod=3, pingLogCleanPeriodUnit=1, pingPeriod=3600, prefetchBatchInit=100, prefetchBatchMin=10, prefetchValidTimeOut=10000, processorBufforChunk=16384, processorBufforPool=163840000, processorCheckPeriod=1000, processorExecutor=4, processors=16, readOnly=false, recordAuditLog=true, recordCrossDbJoin=false, recordDdl=false, recordDeadlockSql=true, recordHotBWs=true, recordHotBWsWarn=false, recordLimitOffsetWithoutOrderBy=false, recordMySQLErrors=false, recordMySQLOptions=false, recordMySQLOptionsWarn=false, recordMySQLOptionsWarns=false, recordSQLForward=false, recordSQLIntercepted=false, recordSQLKeyConflict=false, recordSQLSyntaxErrors=false, recordSQLUnsupported=true, recordsSql=false, recordSqlAuditLog=false, recordSubQuery=false, recordUnionID=false, routeByRelativeCol=false, serverId=2, serverPort=3323, showAllAffectedRowsInGlobalTable=false, skipDatatypeCheck=false, skipBdNodesCheck=false, socketBacklog=1000, sqlTimeout=3600, ssUseSsM=false, statisticsUpdatePeriod=0, strategyForRWSplit=0, switchByLoginFailure=false, switchOverTimeOutForTrans=3000, timerExecutor=4, timestampProxy=0, txIsolation=2, url=jdbc:mysql://192.168.210.202:3316/hotdb_config?useUnicode=true&characterEncoding=UTF-8, usingIO=0, version=0.15, versionComment=HotDB-2.5.6 HotDB Server by Hutox Tech, waitConfigGSyncFinish=false, waitForSlaveInFailure=true, waitSyncFinishAtStartup=true, weightForSlaveNWSplit=50]
```

In normal mode, the compute nodes will start according to the parameter configuration of server.xml, and not be affected by operateMode.

When it is set to performance mode, that is, modifying server.xml, adding operateMode=1 parameter configuration, then make it work by executing reload @@config in 3325 port, and the compute node will output the corresponding information in hotdb.log as follows:

In performance mode, the compute node will initially force to disable the parameters that affect the performance of compute nodes, for example:

recordSql=false,recordSQLSyntaxError=false,recordCrossDNJoin=false,recordUNION=false,recordSubQuery=false,recordDeadLockSQL=false,recordLimitOffsetWithoutOrderby=false,recordSQLKeyConflict=false,recordSQLUnsupported=false,recordMySQLWarnings=false,recordMySQLErrors=false,recordHotDBWarnings=false,recordHotDBErrors=false,recordDDL=false,recordSQLIntercepted=false,recordAuditlog=false,recordSQLForward=false,recordSqlAuditlog=false, even these parameters are configured to true in server.xml.

When it is debug mode, the compute node will output the corresponding information in hotdb.log, as follows:

```
2020-09-10 10:19:03.912 [INFO] [NIOExecutor-1-4] cn.hotpu.hotdb.a[3259] - current sysconfig params: SystemConfig: {VIP=null, adaptiveProcessor=true, aggressive=false, allowRWWithoutReadConsistentInXA=0, autoIncrement=1, badConnAfterContinueGet=true, badConnAfterFastCheckAllIdle=true, bakUrl=null, bakUrls=jdbc:mysql://192.168.210.202:3316/hotdb_config?useUnicode=true&characterEncoding=UTF-8, charset=utf8, checkConnLastUsedTime=3000, checkConnValid=true, checkConnValidTimeOutMs=500, checkParamInterval=600000, checkUpdate=true, checkVIPPeriod=300, clientFoundRows=false, clusterElectionTimeoutMs=2000, clusterHeartbeatTimeoutMs=5000, clusterHost=192.168.210.203, clusterName=HotDB-Cluster, clusterNetwork=192.168.210.204, clusterPacketTimeoutMs=5000, clusterPort=3326, clusterStartedPacketTimeoutMs=5000, configIGB=false, cryptHandshake=false, defaultNodeIndexCheckPeriod=120, deadlockCheckPeriod=0, defaultMaxLimit=10000, defaultSortedLimitSize=2000, drBakUrl=jdbc:mysql://192.168.200.1:3306/hotdb_config?useUnicode=true&characterEncoding=UTF-8, drUrl=jdbc:mysql://192.168.200.1:3306/hotdb_config?useUnicode=true&characterEncoding=UTF-8, enableCursor=false, enableHeartbeat=false, enableLatencyCheck=false, enableLowControl=true, enableUnsafe=false, enableWatchdog=true, enableXA=true, errorsPermittedInTransaction=true, frontConnectionTxIsolationLevel=2, frontWriteBlockTimeOut=10000, globalUniqueConstraint=false, haMode=1, haModeHost=192.168.210.202:3325, haState=master, heartbeatPeriod=2, heartbeatTimeOutMs=500, highCostSqlConcurrency=32, idcId=1, idleTimeOut=0, inHeapEntrySize=20000, joinBatchSize=10000, joinCacheSize=32, joinLoopSize=1000, joinSize=1000000, joinable=true, keyStore=/server.jks, latencyCheckPeriod=600, managerPort=3325, masterSourceInitWaitTimeOut=300, maxAllowedPacket=64M, maxConnections=5002, maxIdleTransactionTimeOut=86400000, maxJoinSize=1000K, maxLatentInSubQuery=20000, maxSqlRecordLength=4000, maxUserConnections=4900, maxWritingToNetTimes=120000, operateMode=2, parkPeriod=100000, pingLogCleanPeriod=3, pingLogCleanPeriodUnit=1, pingPeriod=3600, prefetchBatchMax=10000, prefetchBatchMin=10, prefetchValidTimeOut=10000, processorBufferChunk=16384, processorBufferPool=163840000, processorCheckPeriod=1000, processorExecutors=4, processors=16, readOnly=false, recordAuditLog=true, recordCrossDNJoin=false, recordDDL=false, recordDeadLockSQL=true, recordHotDBErrors=true, recordHotDBWarnings=true, recordLimitOffsetWithoutOrderby=false, recordMySQLErrors=false, recordMySQLWarnings=false, recordSQLForward=false, recordSQLIntercepted=false, recordSQLKeyConflict=false, recordSQLSyntaxError=false, recordsSQLUnsupported=true, recordSQL=false, recordSqlAuditLog=false, recordSubQuery=false, recordUNION=false, routeByRelativeCol=false, serverId=2, serverPort=3323, showAllAffectedRowsInGlobalTable=false, skipDatatypeCheck=false, skipPbbNodesCheck=false, socketBacklog=1000, sqlTimeout=3600, ss1UseSM4=false, statisticsUpdatePeriod=0, strategyForRWSplit=0, switchByFailover=false, switchOverTimeOutForTrans=3000, timestampProxy=0, txIsolation=2, url=jdbc:mysql://192.168.210.202:3316/hotdb_config?useUnicode=true&characterEncoding=UTF-8, usingAIO=0, version=8.0.15, versionComment=HotDB-2.5.6 HotDB Server by Hotpu Tech, waitConfigFinish=false, waitForSlaveInFailover=true, waitSyncFinishAtStartup=true, weightForSlaveRWSplit=50}
2020-09-11 10:19:03.912 [INFO] [INIT] [$NIOExecutor-1-4] cn.hotpu.hotdb.a[2433] - using config datasource in start up:[id:-1,nodeId:-1 192.168.210.202:3316/hotdb_config status:1,charset:utf8]
2020-09-11 10:19:03.915 [INFO] [INIT] [$NIOExecutor-1-4] cn.hotpu.hotdb.a[873] - master config datasource [id:-1,nodeId:-1 192.168.210.202:3316/hotdb_config status:1,charset:utf8] connect success.
2020-09-11 10:19:03.978 [INFO] [INIT] [$NIOExecutor-1-4] cn.hotpu.hotdb.a[3499] - The operateMode is set to 2(debug mode), we will run in the debug mode, and the system parameter will be automatically set to: recordSql=true,recordSQLSyntaxError=true,recordCrossDNJoin=true,recordUNION=true,recordSubQuery=true,recordSQLKeyConflict=false,recordSQLUnsupported=true,recordMySQLWarnings=true,recordMySQLErrors=true,recordHotDBWarnings=true,recordHotDBErrors=true,recordLimitOffsetWithoutOrderby=true,recordSQLIntercepted=true,recordMySQLSyntaxError=true,recordSQLForward=true,recordSQLAuditLog=true
```

In debug mode, the compute node will force to enable the parameters related to the debug function, for example:

recordSql=true,recordSQLSyntaxError=true,recordCrossDNJoin=true,recordUNION=true,recordSubQuery=true,recordDeadLockSQL=true,recordLimitOffsetWithoutOrderby=true,recordSQLKeyConflict=true,recordSQLUnsupported=true,recordMySQLWarnings=true,recordMySQLErrors=true,recordHotDBWarnings=true,recordHotDBErrors=true,recordDDL=true,recordSQLIntercepted=true,recordAuditLog=true,recordSQLForward=true,recordSqlAuditLog=true, even if the parameters are configured to false in server.xml. It should be noted that in debug mode, the compute node will generate more log files, so it is necessary to pay attention to the remaining available disk space to prevent the log files from occupying the disk and causing the compute node service downtime.

14.1.74. parkPeriod

Description of parameter:

Property	Value
Parameter value	parkPeriod

Property	Value
Visible or not	Yes
Description of parameters	Thread Dormancy Period at the idle time of Message System (ns)
Default value	100000
Min value	1000000
Max value	1000
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.3

Parameter Setting:

The parkPeriod parameter in Server.xml is set as follow:

```
<property name="parkPeriod">100000</property>
```

Role of parameter:

This parameter is used for adjusting sleep time of cost message queue thread at the idle time of message queue of internal thread communication.

14.1.75. pingLogCleanPeriod

Description of parameter:

Property	Value

Parameter value	pingLogCleanPeriod
Visible or not	Hidden
Description of parameters	Ping Log Clean Period, 3 by default
Default value	3
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.9

Parameter Setting:

pingLogCleanPeriod parameter configuration in Server.xml is configured as follow:

```
<property name="pingLogCleanPeriod">3</property><!-- Ping Log Clean Period, 3 by default -->
```

Role of parameter:

pingLogCleanPeriod parameter is 3 by default, with the optional unit being Hour, Day and Month, which shall be decided by the other parameter pingLogCleanPeriodUnit. This parameter is mainly used to control the clean period of the data stored in configDB at the time of ping check, and delete the data before specified time periodically every day.

14.1.76. pingLogCleanPeriodUnit

Description of parameter:

Property	Value
Parameter value	pingLogCleanPeriodUnit

Visible or not	Hidden
Description of parameters	Unit of ping log clean period, 2 by default, 0: Hour, 1: Day, 2: Month
Default value	2
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.9

Parameter Setting:

pingLogCleanPeriodUnit parameter configuration in Server.xml is configured as follow:

```
<property name="pingLogCleanPeriodUnit">2</property><!-- Unit of ping log clean period, 2 by default, 0: Hour, 1: Day, 2: Month -->
```

Role of parameter:

pingLogCleanPeriodUnit parameter is 2 by default, meaning that the unit of ping log clean period is Month, besides, the option 0 means Hour and 1 means Day. This parameter is mainly used to control the unit of ping log clean period, and is in support use with the pingLogCleanPeriod parameter.

14.1.77. pingPeriod

Description of parameter:

Property	Value
Parameter value	pingPeriod
Visible or not	Hidden

Description of parameters	ping server period, Unit: s, 3600s by default, min 300s
Default value	3600
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.9

Parameter Setting:

pingPeriod parameter configuration in Server.xml is configured as follow:

```
<property name="pingPeriod">3600</property><!-- ping server period, Unit: s, 3600s by default, min 300s -->
```

Role of parameter:

pingPeriod parameter is 3600 by default, Unit: s, which is mainly used to control ping check period. It's by default to ping a round of IP address of all servers connected with the HotDB Server per hour, for example, client server, configDB server, data source server, etc. Min 300s (that is 5mins) could be configured to trigger a round of check. If the previous round of check is not completed in an hour, then this round of check shall be abandoned directly.

In the detection process, for a certain IP address, the program will automatically use 10 packets (64 byte), 10 packets (65000 byte), which are pinged every 1 second. When the network quality is found to be failed, the interval of ping will be shortened to once per minute, and the criteria of failure judgment are as follows:

- If 64-byte packets in the same IDC are not all lost, when the average delay is greater than 1 ms or the max delay is greater than 2 ms, or there is a packet loss, the time, ping type, average delay, max delay and packet loss rate will be recorded into the ConfigDB hotdb_ping_log. If 65000-byte packets are not all lost, when the average delay is greater than 3 ms, or the max delay is greater than 5 ms, or there

is packet loss, the time, ping type, average delay, max delay, packet loss rate will be recorded into the ConfigDB hotdb_ping_log table.

- If 64-byte packets across the IDCs are not all lost, when the average delay is greater than 10 ms or the max delay is greater than 20 ms, or there is a packet loss, the time, ping type, average delay, max delay and packet loss rate will be recorded into the ConfigDB hotdb_ping_log. If 65000-byte packets are not all lost, when the average delay is greater than 15 ms, or the max delay is greater than 30 ms, or there is packet loss, the time, ping type, average delay, max delay, packet loss rate will be recorded into the ConfigDB hotdb_ping_log table.

14.1.78. prefetchBatchInit

Description of parameter:

Property	Value
Parameter value	prefetchBatchInit
Visible or not	Yes
Description of parameters	The initial value of the auto-incremental batch size
Default value	100
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.4

Parameter Setting:

```
<property name="prefetchBatchInit">100</property>
```

Role of parameter:

The initial value of the auto-incremental batch size. If the initial value is set as 100, the range difference of the prefetch interval is 100 by default. For example, if the prefetch starts from 123, the prefetch interval is [123, 223].

The initial value can be configured within the upper and lower limits of the auto-incremental batch size actually configured (prefetchBatchMax and prefetchBatchMin).

The default range is [10, 10000].

14.1.79. prefetchBatchMax

Description of parameter:

Property	Value
Parameter value	prefetchBatchMax
Visible or not	Yes
Description of parameters	The upper limit of the auto-incremental batch size
Default value	10000
Min value	10
Max value	100000
Whether Reload is valid or not	Yes

Property	Value
Min Compatible Version	2.5.4

Parameter Setting:

```
<property name="prefetchBatchMax">10000</property>
```

Role of parameter:

The upper limit of the auto-incremental batch size. If 1000 is set, the maximum value of the range difference of each prefetch interval is 1000. For example, if the prefetch starts from 123, the maximum value in the prefetch interval is not more than 1123, that is, the values are within [123,1123].

14.1.80. prefetchBatchMin

Description of parameter:

Property	Value
Parameter value	prefetchBatchMin
Visible or not	Yes
Description of parameters	The lower limit of the auto-incremental batch size
Default value	10
Min value	2

Property	Value
Max value	1000
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.4

Parameter Setting:

```
<property name="prefetchBatchMin">10</property>
```

Role of parameter:

The lower limit of the auto-incremental prefetch batch size. If 100 is set, the minimum value of the range difference of each prefetch interval is 100. For example, if the prefetch starts from 123, the maximum value in the prefetch interval is not less than 223, that is, the next prefetch batch starts from 223 at least, and the next prefetch batch [$\geq 223, 223 + \text{prefetch batch size}$].

14.1.81. prefetchValidTimeout

Description of parameter:

Property	Value
Parameter value	prefetchValidTimeout
Visible or not	Yes

Property	Value
Description of parameters	The valid timeout time of prefetch(seconds)
Default value	10
Min value	3
Max value	86400
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.4

Parameter Setting:

```
<property name="prefetchValidTimeout">30</property>
```

Role of parameter:

The valid timeout time of prefetching the auto-incremental batch. When set as 0, it means that the auto-incremental batch is not abandoned due to the timeout. For example, if 30 seconds is set, the prefetch range is 1-100. If it is more than 30 seconds, the unused value is no longer used.

14.1.82. processorExecutor

Description of parameter:

Property	Value
Parameter value	processorExecutor

Property	Value
Visible or not	Yes
Description of parameters	Number of threads of processors
Default value	4
Min value	2
Max value	8
Whether Reload is valid or not	No
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="processorExecutor">4</property><!-- Number of threads of processors -->
```

Role of parameter:

This parameter is used for setting number of threads of processors in internal thread pool model of the compute node. The parameter [adaptiveProcessor](#) is Enabled by default, and when enabled, the compute node will make Automatic Adaptation to Max processorExecutor.

Log in to 3325 port, execute the show @@threadpool; command, and then you could view the current number of processorExecutor.

14.1.83. Processors

Description of parameter:

Property	Value
Parameter value	processors
Visible or not	Yes
Description of parameters	Number of processors
Default value	8
Min value	4
Max value	128
Whether Reload is valid or not	No
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="processors">8</property><!—Number of processors -->
```

Role of parameter:

This parameter is used for setting number of threads in internal thread pool model of the compute node. The parameter [adaptiveProcessor](#) is Enable by default, and when enabled, the compute node will make Automatic Adaptation to the max number of processors.

Log in to 3325 port, execute show @@threadpool; command, and then you could view the number of current processors.

14.1.84. readOnly

Description of parameter:

Property	Value
Parameter value	readOnly
Visible or not	No
Description of parameters	readOnly mode or not
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.8

Parameter Setting:

```
<property name="readOnly">false</property><!-- readOnly mode or not -->
```

Role of parameter:

It is used for setting the current compute node as readonly mode. In readonly mode, compute node only receives DQL (SELECT statement) operation, and SET command row and SHOW type operations, and refuses to execute DDL(CREATE TABLE/VIEW/INDEX/SYN/CLUSTER statement), DML(INSERT, UPDATE, DELETE) and DCL (GRANT, ROLLBACK [WORK] TO [SAVEPOINT], COMMIT) modification operation commands

Notices: This parameter is still provided for single compute node service, and does not allow multiple compute nodes to provide service simultaneously, that is, it does not allow enabling multiple compute nodes and providing external service at the same time

Enable status:

```
mysql> drop table customer;
ERROR 1289 (HY000): Command not allowed in Read-Only mode.
```

14.1.85. recordAuditlog

Description of parameter:

Property	Value
Parameter value	recordAuditlog
Visible or not	No
Description of parameters	Record audit log
Default value	true
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.0

Parameter Setting:

recordAuditlog parameter configuration in Server.xml is configured as follow:

```
<property name="recordAuditlog">true</property><!--Record audit log -->
```

Role of parameter:

recordAuditlog enables audit log or not. This parameter is used for controlling whether to record the management port operation information or not. When the audit log is enabled, you could view management port operation record via Event->Audit log on the management platform.

14.1.86. recordCrossDNJoin

Description of parameter:

Property	Value
Parameter value	recordCrossDNJoin
Visible or not	No
Description of parameters	Record cross-node JOIN in log
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.7

Parameter Setting:

recordCrossDNJoin parameter configuration in Server.xml is configured as follow:

```
<property name="recordCrossDNJoin">false</property>
```

Role of parameter:

recordCrossDNJoin records cross-node join statement

Create Table:

```
account table auto_crc32sharding, sharding key id, node 1
```

```
borrower table auto_modsharding, sharding key id, node 2
```

Execute as follow:

```
mysql> SELECT * FROM account a JOIN borrower b;
```

View /logs/sql.log of compute node installation directory

```
2018-05-22 16:17:11.607 [INFO] [CROSSDNJOIN] [$NIOExecutor-6-2] JoinVisitor(4947) – SELECT * FROM
account a JOIN borrower b
```

14.1.87. recordDDL

Description of parameter:

Property	Value
Parameter value	recordDDL
Visible or not	No
Description of parameters	Record DDL statement in log
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.7

Parameter Setting:

recordDDL parameter configuration in Server.xml is configured as follow:

```
<property name="recordDDL">false</property>
```

Role of parameter:

recordDDL log records DDL statement, execute the following statement:

```
mysql> create table abc(id int);
```

View /logs/sql.log of compute node installation directory

```
2018-05-23 14:23:52.697 [INFO] [HOTDBWARNING] [$NIOExecutor-6-2] ServerConnection(2368) – sql:  
create table abc(id int), warning: {Create table without Primary Key and unique key}
```

```
2018-05-23 14:23:52.698 [INFO] [DDL] [$NIOExecutor-6-2] ServerConnection(123) – sql: create table abc(id  
int)
```

14.1.88. recordDeadLockSQL

Description of parameter:

Property	Value
Parameter value	recordDeadLockSQL
Visible or not	No
Description of parameters	The log records the statement triggering deadlock
Default value	true
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.7

Parameter Setting:

recordDeadLockSQL parameter configuration in Server.xml is configured as follow:

```
<property name="recordDeadLockSQL">true</property>
```

Role of parameter:

recordDeadLockSQL log records the statement triggering deadlock:

1. Create deadlock scenario

2. View /logs/hotdb.log of compute node installation directory:

```
2018-05-23 14:54:30.865 [INFO] [DEADLOCK] [$NIOREACTOR-1-RW] am(-1) – sql: INSERT INTO
table2000 VALUES (3); error response from MySQLConnection [node=4, id=277, threadId=133815,
state=borrowed, close=false, autocommit=false, host=192.168.220.102, port=3309, database=db249,
localPort=15332, isClose:false, toBeClose:false], err: Lock wait timeout exceeded; try restarting transaction, code:
1205
```

14.1.89. recordHotDBErrors

Description of parameter:

Property	Value
Parameter value	recordHotDBErrors
Visible or not	No
Description of parameters	The log records error message returned by HotDB
Default value	true
Whether Reload is valid or not	Yes

Property	Value
Min Compatible Version	2.4.8

Parameter Setting:

recordHotDBErrors parameter configuration in Server.xml is configured as follow:

```
<property name="recordHotDBErrors">true</property>
```

Role of parameter:

recordHotDBErrors log records error message returned by compute node

For example: when executing Create statement by user without create privilege, the prompt is as follow:

```
2018-06-04 10:43:07.316 [INFO] [HOTDBERROR] [$NIOExecutor-3-0] ServerConnection(155) – sql: create
table a001(id int), err: [CREATE] command denied to user 'jzl' to logic database 'TEST_JZL'
```

14.1.90. recordHotDBWarnings

Description of parameter:

Property	Value
Parameter value	recordHotDBWarnings

Property	Value
Visible or not	No
Description of parameters	The log records the warning message returned by the compute node
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.7

Parameter Setting:

recordHotDBWarnings parameter configuration in Server.xml is configured as follow:

```
<property name="recordHotDBWarnings">false</property>
```

Role of parameter:

recordHotDBWarnings log records the warning message returned by the compute node, execute as follow:

```
create table abc(id int);
```

View /logs/sql.log of the compute node installation directory

```
2018-05-23 14:23:52.697 [INFO] [HOTDBWARNING] [$NIOExecutor-6-2] ServerConnection(2368) – sql:  
create table abc(id int), warning: {Create table without Primary Key and unique key}
```

```
2018-05-23 14:23:52.698 [INFO] [DDL] [$NIOExecutor-6-2] ServerConnection(123) – sql: create table abc(id int)
```

14.1.91. recordLimitOffsetWithoutOrderby

Description of parameter:

Property	Value
Parameter value	recordLimitOffsetWithoutOrderby
Visible or not	No
Description of parameters	The log records the limit statement without orderby
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.7

Parameter Setting:

recordLimitOffsetWithoutOrderby parameter configuration in Server.xml is configured as follow:

```
<property name="recordLimitOffsetWithoutOrderby">false</property>
```

Role of parameter:

recordLimitOffsetWithoutOrderby log records the limit statement without orderby.

Execute as follow:

```
mysql> select * FROM account a WHERE a.Branch_name IN(SELECT b.Branch_name FROM branch b ) limit 1,3;
```

View /logs/sql.log of compute node installation directory

```
2018-05-23 14:05:14.915 [INFO] [LIMITOFFSETWITHOUTORDERBY] [$NIOExecutor-6-l]
SubqueryExecutor(97) - sql: select * FROM account a WHERE a.Branch_name IN(SELECT b.Branch_name
FROM branch b) limit 1,3

2018-05-23 14:05:14.922 [INFO] [LIMITOFFSETWITHOUTORDERBY] [$NIOExecutor-2-3] BaseSession(97)
- sql: SELECT A.`Balance`, A.`Branch_name`, A.`Account_number`, A.`account_date` FROM account AS a
WHERE a.Branch_name IN (UNHEX('4272696768746F6E'), UNHEX('4272696768746F6E'),
UNHEX('526564776F6F64'), UNHEX('50657272797269646765'), UNHEX('50657272797269646765'),
UNHEX('526564776
F6f64'), NULL) LIMIT 1 , 3
```

14.1.92. recordMySQLErrors

Description of parameter:

Property	Value
Parameter value	recordMySQLErrors
Visible or not	No
Description of parameters	The log records error message returned by MySQL
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.7

Parameter Setting:

recordMySQLErrors parameter configuration in Server.xml is configured as follow:

```
<property name="recordMySQLErrors">false</property>
```

Role of parameter:

recordMySQLErrors log records error message returned by MySQL.

Execute as follow:

```
msyql> select form;
```

View /logs/hotdb.log of compute node installation directory

```
2018-05-23 14:38:55.843 [INFO] [MYSQLERROR] [$NIOREACTOR-7-RW] MySQLConnection(56) – sql:  
select form, error response from MySQLConnection [node=4, id=223, threadId=118551, state=borrowed,  
close=false, autocommit=true, host=192.168.220.103, port=3309, database=db249, localPort=27007, isClose:false,  
toBeClose:false], err: Unknown column 'form' in 'field list', code: 1054
```

14.1.93. recordMySQLWarnings

Description of parameter:

Property	Value
Parameter value	recordMySQLWarnings
Visible or not	Hidden
Description of parameters	The log records the warning message returned by MySQL
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.7

Parameter Setting:

recordMySQLWarnings parameter configuration in Server.xml is configured as follow:

```
<property name="recordMySQLWarnings">false</property>
```

Role of parameter:

recordMySQLWarnings log records the warning message returned by MySQL.

Execute as follow:

```
mysql> update account set Account_number="$!\\\"##";
```

View /logs/sql.log of compute node installation directory,

```
2018-06-12 10:52:07.011 [INFO] [MYSQLWARNING] [|$NIOREACTOR-3-RW] showwarningsHandler(79) -
sql: UPDATE account SET Account_number = '$!\\\"##', warning from MySQLConnection [node=2, id=78814,
threadId=75272, state=running, closed=false, autocommit=false, host=192.168.200.51, port=3309, database-
```

db249, localPort=13317, isclose:false, toBeclose:false], warning: Data truncated for column 'Account_number' at row 1, code: 1265

2018-06-12 10:52:07.012 [INFO] [MYSQLWARNING] |[\$NIOREACTOR-3-RW] showwarningsHandler(79) - sql: UPDATE account SET Account_number = '*\$!\\\'##', warning from MySQLConnection [node=2, id=78814, threadId=75272, state=running, closed=false, autocommit=false, host=192.168.200.51, port=3309, database=db249, localPort=13317, isclose:false, toBeclose:false], warning: Data truncated for column 'Account_number' at row 2, code: 1265

2018-06-12 10:52:07.012 [INFO] [MYSQLWARNING] |[\$NIOREACTOR-3-RW] showwarningsHandler(79) - sql: UPDATE account SET Account_number = '*\$!\\\'##', warning from MySQLConnection [node=3, id=55313, threadId=166, state=running, closed=false, autocommit=false, host=192.168.200.52, port=3309, database=db249, localPort=13317, isclose:false, toBeclose:false], warning: Data truncated for column 'Account_number' at row 1, code: 1265

2018-06-12 10:52:07.013 [INFO] [MYSQLWARNING] |[\$NIOREACTOR-3-RW] showwarningsHandler(79) - sql: UPDATE account SET Account_number = '*\$!\\\'##', warning from MySQLConnection [node=3, id=55313, threadId=166, state=running, closed=false, autocommit=false, host=192.168.200.52, port=3309, database=db249, localPort=13317, isclose:false, toBeclose:false], warning: Data truncated for column 'Account_number' at row 2, code: 1265

14.1.94. recordSql

Description of parameter:

Property	Value
Parameter value	recordSql
Visible or not	Yes
Description of parameters	Make statistics of SQL execution condition or not
Default value	false
Whether Reload is valid or not	Yes

Property	Value
Min Compatible Version	2.4.3

Parameter Setting:

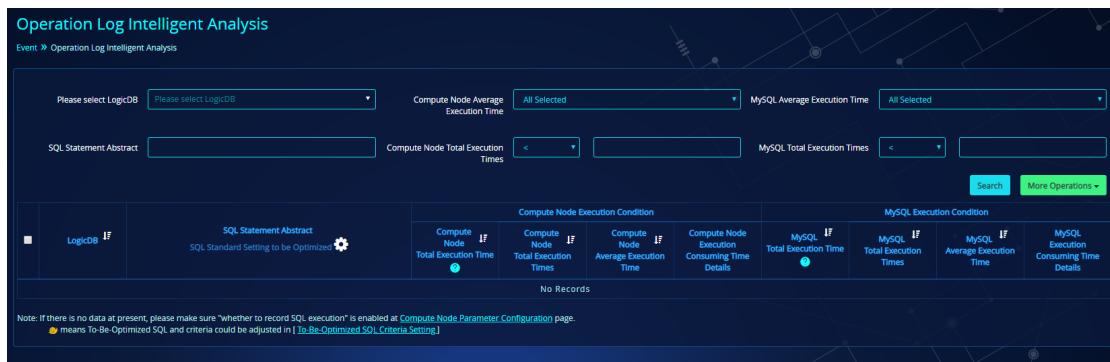
<property name="recordSql">false</property>!-- Make statistics of SQL execution condition or not, Yes: true,
No: false -->

Role of parameter:

Make statistics of SQL execution condition or not.

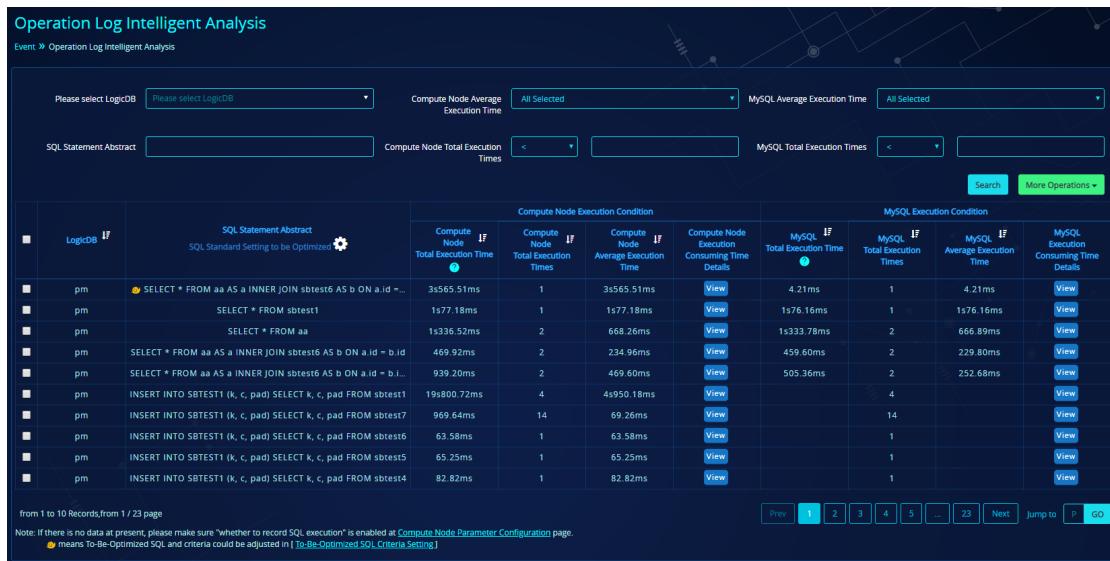
1. View via the Slow Query Log Analysis page on the management platform

- OFF status



The screenshot shows the 'Operation Log Intelligent Analysis' page. At the top, there are dropdown menus for 'Please select LogicDB' and 'MySQL Average Execution Time'. Below these are several input fields for 'SQL Statement Abstract' and 'Compute Node Total Execution Times'. The main area contains a table with columns for 'LogicDB IF', 'SQL Statement Abstract', and various execution time metrics. A note at the bottom states: 'Note: If there is no data at present, please make sure "whether to record SQL execution" is enabled at Compute Node Parameter Configuration page. ⚡ means To-Be-Optimized SQL and criteria could be adjusted in [To-Be-Optimized SQL Criteria Setting]'.

- ON status



The screenshot shows the same 'Operation Log Intelligent Analysis' page but with data populated. The table now lists many rows of SQL statements, each with a 'LogicDB IF' column indicating if it's optimized. The columns include 'SQL Statement Abstract', 'Compute Node Total Execution Time', and various MySQL execution time metrics like 'Total Execution Time', 'Average Execution Time', and 'Consuming Time Details'. Each row has a 'View' button next to it. A note at the bottom says: 'from 1 to 10 Records, from 1 / 23 page. Note: If there is no data at present, please make sure "whether to record SQL execution" is enabled at Compute Node Parameter Configuration page. ⚡ means To-Be-Optimized SQL and criteria could be adjusted in [To-Be-Optimized SQL Criteria Setting]'.

2. View statistics of SQL execution via server configDB

```
mysql> select * from hotdb_query_records order by db_id limit 1\G
*****
1. row*****  
  
    id: 2  
  
  db_id: 1  
  
   type: SELECT  
  
  query: SELECT COUNT(*) FROM union b  
  
total_hotdb_time: 67934  
  
total_mysql_time: 52105  
  
      hms1: 0  
  
     hms10: 1  
  
    hms300: 1  
  
      hs1: 0  
  
     hs3: 0  
  
    hs10: 0  
  
   hs60: 0  
  
  hs600: 0  
  
 hs600p: 0  
  
  mms1: 0  
  
 mms10: 1  
  
mms300: 1  
  
   ms1: 0  
  
   ms3: 0  
  
  ms10: 0  
  
  ms60: 0  
  
 ms600: 0
```

```
ms600p: 0

htime24: 67934

htime48: 67934

mtime24: 52105

mtime48: 52105

hcount24: 2

hcount48: 2

mcount24: 2

mcount48: 2

return_rows: 2

last update time: 2018-05-29 11:04:31.000000

crc: 321944166562

1 row in set (0.00 sec)
```

14.1.95. recordSqlAuditlog

Description of parameter:

Property	Value
Parameter value	recordSqlAuditlog
Visible or not	No
Description of parameters	Record SQL audit log.
Default value	false

Property	Value
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.5

Parameter Setting:

The parameter recordSqlAuditlog in Server.xml is configured as false by default:

```
<property name="recordSqlAuditlog">false</property>
```

Role of parameter:

If the parameter is set as true, DDL、DML、DQL and other operations will be recorded in logs/extra/sqlaudit/ of the compute node installation directory.

For example, execute DDL on the server of compute node and view the log output.

```
{"affected_rows":"0","command":"CREATE TABLE `t_sharding_01` (\n`id` int(10) NOT NULL AUTO_INCREMENT,\n`name` varchar(50) NOT NULL,\n`age` int(3),\nPRIMARY KEY (`id`)\n) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4","connection_id":"44","end_time":"2020-04-27 14:58:34.769","failed_reason":"","host":"127.0.0.1","ip":"127.0.0.1","log_id":9524067900080128,"logic_db":"CXD_DB","matched_rows":"0","port":3323,"query_rows":0,"sql_subtype":"CREATE","sql_type":"DDL","status":1,"time":2020-04-27 14:58:34.736,"user":cxd@%}
```

Note: the log is output in the format of json. Special characters such as double quotation marks are escaped with \.

The meaning of some keys in json is as follows:

sql_type: the type of SQL currently executed, including DDL/DML/DQL/OTHER.

sql_subtype: the subtype of SQL currently executed, among which DDL includes CREATE/ALTER/DROP/TUNCATE/RENAME; DQL includes SELECT; DML includes UPDATE/DELETE/INSERT/REPLACE/LOAD; OTHER includes SET/PREPARE/TRANSACTION/SHOW.

ip: the IP address of the client executing SQL.

time: time to execute SQL.

user: the user (including the host name) who connects to the compute node and executes SQL

host: connects to the host value specified by the compute node.

logic_db: connects to the LogicDB used by the compute node to execute SQL.

connection_id: the front-end connection ID used to execute SQL.

command: the statement that specifically executes SQL (the original SQL statement).

query_rows: the number of data rows returned (mainly reflected in the SELECT operation).

affected_rows: the number of rows affected by SQL execution.

matched_rows: the number of rows matched by SQL execution.

status: whether the SQL execution status is success or failure. 0 is for failure and 1 is for success.

failed_reason: the reason why SQL execution failed.

end_time: end time of SQL execution.

14.1.96. recordSQLIntercepted

Description of parameter:

Property	Value
Parameter value	recordSQLIntercepted
Visible or not	No
Description of parameters	The log records the intercepted statement
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.7

Parameter Setting:

recordSQLIntercepted parameter configuration in Server.xml is configured as follow:

```
<property name="recordSQLIntercepted">true</property>
```

Role of parameter:

recordSQLIntercepted log records the intercepted statement; configuration of intercepted statement is in Middleware management platform->Safety->SQL firewall.

View /logs/sql.log of compute node installation directory

```
2018-06-01 14:17:45.669 [INFO] [SQLINTERCEPTED] [$NIOExecutor-1-2] g(-1) – sql: DELETE FROM
sql_intercept_tab, user:zy, ip: 192.168.200.45, db: TEST_JZL, intercepted by filewall: not allowed to execute
delete without where expression
```

14.1.97. recordSQLKeyConflict

Description of parameter:

Property	Value
Parameter value	recordSQLKeyConflict
Visible or not	No
Description of parameters	The log records the statement with Primary Key conflict and violating foreign key constraint

Property	Value
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.7

Parameter Setting:

recordSQLKeyConflict parameter configuration in Server.xml is configured as follow:

```
<property name="recordSQLKeyConflict">false</property>
```

Role of parameter:

recordSQLKeyConflict log records the statement with Primary Key conflict and violating foreign key constraint.

Executed as follow:

Create Table:

```
mysql> CREATE TABLE `vtab001` (`id` int(11) NOT NULL,`name` varchar(255) DEFAULT NULL,PRIMARY KEY (`id`));
```

Execute Insert statement once:

```
mysql> insert into vtab001 values(1,'aaa');
```

Execute again, to violate the Primary Key Constraint:

```
mysql> insert into vtab001 values(1,'aaa');
```

View /logs/sql.log of compute node installation directory

```
2018-06-01 14:09:47.139 [INFO] [SQLKEYCONFLICT] [$NIOREACTOR-1-RW] MySQLConnection(65) – sql:  
insert into vtab001 values(1,'aaa'), error response from MySQLConnection [node=1, id=19, threadId=121339,  
state=borrowed, closed=false, autocommit=true, host=192.168.220.102, port=3306, database=db249,  
localPort=56158, isclose:false, toBeClose:false], err: Duplicate entry '1' for key 'PRIMARY', CODE: 1062
```

14.1.98. recordSQLSyntaxError

Description of parameter:

Property	Value
Parameter value	recordSQLSyntaxError
Visible or not	No
Description of parameters	The log records statement with Syntax error
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.7

Parameter Setting:

recordSQLSyntaxError parameter configuration in Server.xml is configured as follow:

```
<property name="recordSQLSyntaxError">false</property>
```

Role of parameter:

recordSQLSyntaxError records statement with Syntax error.

For example:

```
mysql> SELECT * FROM;
```

View /logs/sql.log of compute node installation directory

```
2018-05-22 16:12:42.686 [INFO] [SQLSYNTAXERROR] [$NIOExecutor-6-3] ServerConnection(671) -
SELECT * FROM
```

14.1.99. recordSQLUnsupported

Description of parameter:

Property	Value
Parameter value	recordSQLUnsupported
Visible or not	No
Description of parameters	The log records the statement not supported
Default value	true
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.7

Parameter Setting:

recordSQLUnsupported parameter configuration in Server.xml is configured as follow:

```
<property name="recordSQLUnsupported">true</property>
```

Role of parameter:

recordSQLUnsupported log records the unsupported statement.

For example:

```
Create Table: mysql> CREATE TABLE `vtab001` (`id` int(11) NOT NULL,`name` varchar(255) DEFAULT NULL,PRIMARY KEY (`id`));
```

```
Execute the statement not supported by HotDB for the time being: mysql> select * into vtab001_bak from vtab001;
```

View /logs/sql.log of compute node installation directory

```
2018-05-22 14:19:54.395 [INFO] [SQLUNSUPPORTED] [$NIOExecutor-6-2] ServerConnection(110) – sql:  
select * into vtab001_bak from vtab001
```

14.1.100. recordSubQuery

Description of parameter:

Property	Value
Parameter value	recordSubQuery
Visible or not	No
Description of parameters	The log records Subquery
Default value	false
Whether Reload is valid or not	Yes

Property	Value
Min Compatible Version	2.4.7

Parameter Setting:

recordSubQuery parameter configuration in Server.xml is configured as follow:

```
<property name="recordSubQuery">false</property>
```

Role of parameter:

recordSubQuery log records Subquery.

For example:

```
mysql> select * FROM account a WHERE a.Branch_name IN(SELECT b.Branch_name FROM branch b );
```

View /logs/sql.log of compute node installation directory

```
2018-05-23 13:56:11.714 [INFO] [SUBQUERY] [$NIOExecutor-6-0] SubqueryExecutor(169) – select * FROM account a WHERE a.Branch_name IN(SELECT b.Branch_name FROM branch b )
```

14.1.101. recordUNION

Description of parameter:

Property	Value
Parameter value	recordUNION
Visible or not	No
Description of parameters	The log records UNION
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.7

Parameter Setting:

recordUNION parameter configuration in Server.xml is configured as follow:

```
<property name="recordUNION">false</property>
```

Role of parameter:

recordUNION records UNION statement.

For example:

```
mysql> SELECT * FROM trends UNION  SELECT * from trends_uint;
```

View /logs/sql.log of compute node installation directory

```
2018-05-23 13:30:27.156 [INFO] [UNION] [$NIOREACTOR-5-RW] UnionExecutor(162) - SELECT * FROM
trends UNION  SELECT * from trends_uint
```

14.1.102. routeByRelativeCol

Description of parameter:

Property	Value
Parameter value	routeByRelativeCol
Visible or not	No
Description of parameters	It does not include the route via Secondary Index Field at the time of sharding key
Default value	false
Whether Reload is valid or not	No
Min Compatible Version	2.5.2

Parameter Setting:

routeByRelativeCol parameter configuration in Server.xml is configured as follow:

```
<property name="routeByRelativeCol">false</property><!-- It does not include the route via Secondary Index
Field at the time of sharding key -->
```

Role of parameter:

This function is OFF by default, that is, it does not route via Secondary Index Field when the sharding key is not included, and it is ON after being modified as true. After enabled, it supports to locate to the specific node via Query Secondary Index and distribute the SELECT Query statement to specified node only instead of all nodes when the SELECT Query statement does not include sharding key but includes Unique Constraint Field.

14.1.103. serverId

Description of parameter:

Property	Value
Parameter value	serverId
Visible or not	Yes
Description of parameters	Cluster node number 1-N (number of nodes), unique in cluster
Default value	1
Whether Reload is valid or not	No
Min Compatible Version	2.5.0

Parameter Setting:

serverId parameter configuration in Server.xml is configured as follow:

```
<property name="serverId">1</property><!-- Cluster node number 1-N (number of nodes), unique in cluster -->
```

Role of parameter:

It is used for dividing connection communication ID among the nodes in the cluster, and this Parameter Setting shall be set from 1 continuously without repetition, and in case of repetition, the cluster will start abnormally.

14.1.104. service port & management port

Description of parameter:

Property	Value
Parameter value	service port
Visible or not	Yes
Description of parameters	service port
Default value	3323
Whether Reload is valid or not	No
Min Compatible Version	2.4.3

Property	Value
Parameter value	management port
Visible or not	Yes
Description of parameters	management port
Default value	3325
Whether Reload is valid or not	No
Min Compatible Version	2.4.3

Role of parameter:

Service port is used to log in to the compute node to execute relevant statements, and its use is similar to that of MySQL.

management port is used to monitor compute node service information and monitoring statistical information, and you could view by executing relevant commands.

14.1.105. showAllAffectedRowsInGlobalTable

Description of parameter:

Property	Value
Parameter value	showAllAffectedRowsInGlobalTable
Visible or not	Yes
Description of parameters	Whether Global Table IDU statement shows total number of AffectedRows in all nodes
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Role of parameter:

When showAllAffectedRowsInGlobalTable Parameter is set as true, Global Table will execute insert,delete,update related SQL statements, and the result will show total number of affected nodes.

For example: Global Table join_c06_ct associates 8 nodes; after executing this SQL statement, 1 row of the actual data will be updated; when this Parameter is set as true, the result will show that the affected number is 8 (that is: number of updated

rows*number of affected nodes).

```
mysql> delete from join_us06_ct where id = 8;
```

Query OK, 8 rows affected (0.01 sec)

```
mysql> update join_us06_ct set e = 'y' where id =7;
```

Query OK, 8 rows affected (0.04 sec)

Rows matched: 8 Changed: 8 Warnings: 0

```
mysql> insert into join_us06_ct values (8,6,1.3,1.4,'y','u',now(),now(),2017);
```

Query OK, 8 rows affected (0.01 sec)

Records: 8 Duplicates: 0 Warnings: 0

When this Parameter is set as false, it will only show number of rows affected, as prompted below:

```
mysql> update join_us06_ct set e = 'm' where id =4;
```

Query OK, 1 rows affected (0.10 sec)

Rows matched: 1 Changed: 1 Warnings: 0

14.1.106. skipDatatypeCheck

Description of parameter:

Property	Value
Parameter value	skipDatatypeCheck

Property	Value
Visible or not	No
Description of parameters	Control whether to skip checking of column data type in table structure
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.5

Parameter Setting:

skipDatatypeCheck parameter in Server.xml

```
<property name="skipDatatypeCheck">true</property>
```

Role of parameter:

When executing Create and Alter statements on Middleware server, whether to check if there is double, float, real data type or not in non-sharding key.

For example:

```
mysql> alter table skipDatatypeCheck add(phone double(10,3));
```

skipDatatypeCheck=false:

```
mysql> alter table skipDatatypeCheck add(phone double(10,3));
```

```
ERROR 1064 (HY000): Column type:'DOUBLE' is forbidden, you could change column:'PHONE' to type  
'DECIMAL'
```

skipDatatypeCheck=true:

```
mysql> alter table skipDatatypeCheck add(phone double(10,3));
```

```
Query OK, 0 rows affected (0.23 sec)
```

14.1.107. socketBacklog

Description of parameter:

Property	Value
Parameter value	socketBacklog
Visible or not	No
Description of parameters	service port Socket backlog
Default value	1000
Min value	1000
Max value	4000
Whether Reload is valid or not	No
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="socketBacklog">1000</property><!-- service port Socket backlog (Unit: Ge) -->
```

Role of parameter:

Service port socket requires a certain period of time to process socket connection of the Client and has a queue to store the Client Socket haven't been processed in time; the

capacity of the queue is backlog. If the queue has been fully occupied by Client socket, ServerSocket will refuse the new connections to guarantee enough queue capacity, thus there will be no connection which can't be connected due to small queue capacity.

14.1.108. sqlTimeout

Description of parameter:

Property	Value
Parameter value	sqlTimeout
Visible or not	Yes
Description of parameters	Sql Execution Timeout (S)
Default value	3600
Min value	1
Max value	28800
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.3

Role of parameter:

This is the max time of compute node from sending SQL to data source to receiving SQL execution result (including single-node and cross-node). If exceeding the set time, timeout will be prompted.

When SQL execution time exceeds the set time, there will be prompt as follow:

```
mysql> select a.* , b.* , c.* from customer_auto_3 a join customer_auto_1 b on a.postcode=b.postcode join
customer_auto_2 c on a.provinceid=c.provinceid where c.provinceid in (12,15) and b.province !='anhui' group by
a.postcode order by a.birthday,a.provinceid,b.birthday,c.postcode limit 1000;
```

ERROR 1003 (HY000): query timeout, transaction rollbacked automatically and a new transaction started
automatically

14.1.109. sslUseSM4

参数说明:

Property	Value
参数值	sslUseSM4
是否可见	否
参数说明	是否支持国密算法
默认值	否
Reload 是否生效	是
最低兼容版本	2.5.5

参数设置:

server.xml 中 sslUseSM4 参数如下配置:

```
<property name="sslUseSM4">true</property><!--是否支持国密算法 -->
```

参数作用:

在 server.xml 中打开 enableSSL 和 sslUseSM4 开关，可以使客户端访问计算节点的过程处于国密验证的加密状态。

```

<property name="ndbSqlUser"><root></property>!-- NDB SQL前端用户名(NDB SQL node user name) -->
<property name="ndbSqlPass"><root></property>!-- NDB SQL前端密码(NDB SQL node password) -->
<property name="ndbSqlDataAddr">127.0.0.1:3327</property>!-- 接收NDB SQL连接的IP地址和端口(NDB Data node address) -->
<property name="failoverAutoresetslave">false</property>!-- 故障切换时, 是否自动重置主从复制关系(Autoreset replication relationship) -->
<property name="prefetchBatchMax">10000</property>!-- 序列号预取自增长批次大小的上限, 最小10, 最大1000000(Maximum batch size of sequence pre-->
<property name="prefetchBatchMin">10</property>!-- 序列号预取自增长批次大小的下限, 最小2, 最大1000(Minimal batch size of sequence pre-->
<property name="prefetchBatchInit">100</property>!-- 序列号预取自增长批次的初始值, 在配置的上下限范围内(Initial batch size of sequence pre-->
<property name="prefetchValidTimeout">10</property>!-- 序列号预取自增长批次的超时废弃时间(单位: 秒, 0为永不废弃, 最小3, 最大86400)(Batch-->
<property name="enableListener">false</property>!-- 启用Listener模式(Enable Listener mode or not)-->
<property name="enableCalcite">true</property>!-- 启用Calcite插件(Enable Calcite plugin or not)-->
<property name="enableSSL">true</property>!-- 是否开启SSL连接功能(Enable SSL connection or not)-->
<property name="sslUseSM4">true</property>!-- 是否开启SM4加密套件(Enable SM4 encryption or not)-->
<property name="keyStore"></server></property>!-- 用于TLS连接的数据证书.jks文件的路径(Path to the data certificate .jks file for T-->
<property name="keyStorePass">BB5A70F75DD5FEB214A5623D171CEEB</property>!-- 用于TLS连接的数据证书.jks文件的密码(Password of the da-->
<property name="dropTableRetentionTime">0</property>!-- 被删除表保留时长, 默认为0, 不保留(The retention period of the deleted table. 0-->
</system>

```

该功能对于用户来说只能通过抓包查看, 示例: 抓包可见 TLS 握手包中存在 HotDB Server 国密 SM4 定义的加密套件编号: 0xff01, 说明 SM4 加解密套件已生效。

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000813	192.168.210.225	192.168.210.225	TLSv1.1	144	Ignored Unknown Record
5	0.000818	192.168.210.225	192.168.210.225	TCP	66	48630 → 3323 [ACK] Seq=1 Ack=79 Win=131072 Len=0 TSval=1896499745 TSecr=1896499745
6	0.010215	192.168.210.225	192.168.210.225	TLSv1.1	102	Ignored Unknown Record
7	0.010221	192.168.210.225	192.168.210.225	TCP	66	3323 → 48630 [ACK] Seq=79 Ack=37 Win=32732 Len=0 TSval=1896499754 TSecr=1896499754
8	0.106327	192.168.210.225	192.168.210.225	TLSv1.1	170	Client Hello
9	0.106335	192.168.210.225	192.168.210.225	TCP	66	48630 → 3323 [ACK] Seq=79 Ack=141 Win=32628 Len=0 TSval=1896499850 TSecr=1896499850
10	0.107904	192.168.210.225	192.168.210.225	TLSv1.1	1724	Server Hello, Certificate, Server Hello Done
11	0.114688	192.168.210.225	192.168.210.225	TCP	66	48630 → 3323 [ACK] Seq=141 Ack=1737 Win=262144 Len=0 TSval=1896499859 TSecr=1896499852
12	0.130219	192.168.210.225	192.168.210.225	TLSv1.1	333	Client Key Exchange
Handshake Type: Client Hello (1) Length: 95 Version: TLS 1.1 (0x0302) Random: 5f16a07d73b0f5a1e61a3217bfea36791284907427cf57d6... Session ID Length: 0 Cipher Suites Length: 32 Cipher Suites (16 suites)						
Cipher Suite: Unknown (0xff01) Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035) Cipher Suite: TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA (0xc005) Cipher Suite: TLS_ECDH_RSA_WITH_AES_256_CBC_SHA (0x000f) Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039) Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x0038) Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f) Cipher Suite: TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA (0xc004) Cipher Suite: TLS_ECDH_RSA_WITH_AES_128_CBC_SHA (0x000e) Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033) Cipher Suite: TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x0032) Cipher Suite: TLS_EMPTY_RENEGOTIATION_INFO_SCSV (0x00ff) Compression Methods Length: 1 Compression Methods (1 method)						
4	0.000813	192.168.210.225	192.168.210.225	TLSv1.1	144	Ignored Unknown Record
5	0.000818	192.168.210.225	192.168.210.225	TCP	66	48630 → 3323 [ACK] Seq=1 Ack=79 Win=131072 Len=0 TSval=1896499745 TSecr=1896499745
6	0.010215	192.168.210.225	192.168.210.225	TLSv1.1	102	Ignored Unknown Record
7	0.010221	192.168.210.225	192.168.210.225	TCP	66	3323 → 48630 [ACK] Seq=79 Ack=37 Win=32732 Len=0 TSval=1896499754 TSecr=1896499754
8	0.106327	192.168.210.225	192.168.210.225	TLSv1.1	170	Client Hello
9	0.106335	192.168.210.225	192.168.210.225	TCP	66	48630 → 3323 [ACK] Seq=79 Ack=141 Win=32628 Len=0 TSval=1896499850 TSecr=1896499850
10	0.107904	192.168.210.225	192.168.210.225	TLSv1.1	1724	Server Hello, Certificate, Server Hello Done
11	0.114688	192.168.210.225	192.168.210.225	TCP	66	48630 → 3323 [ACK] Seq=141 Ack=1737 Win=262144 Len=0 TSval=1896499859 TSecr=1896499852
12	0.130219	192.168.210.225	192.168.210.225	TLSv1.1	333	Client Key Exchange
> Internet Protocol Version 4, Src: 192.168.210.225, Dst: 192.168.210.225 > Transmission Control Protocol, Src Port: 3323, Dst Port: 48630, Seq: 79, Ack: 141, Len: 1658 > Transport Layer Security > TLSv1.1 Record Layer: Handshake Protocol: Multiple Handshake Messages Content Type: Handshake (22) Version: TLS 1.1 (0x0302) Length: 1653 > Handshake Protocol: Server Hello Handshake Type: Server Hello (2) Length: 81 Version: TLS 1.1 (0x0302) Random: 5f16a07d1e70b55ecb577ae0dba07f44a70a83b309af4... Session ID Length: 32 Session ID: 5f16a07d736cd4693a36fe575339cb4d476d3af3ed04abb7... Cipher Suite: Unknown (0xff01) Compression Method: null (0) Extensions Length: 9 > Extension: renegotiation_info (len=1) > Extension: extended_master_secret (len=0) > Handshake Protocol: Certificate Handshake Type: Certificate (11) Length: 1560 Certificates Length: 1557 > Certificates (1557 bytes) > Handshake Protocol: Server Hello Done						

14.1.110. statisticsUpdatePeriod

Description of parameter:

Property	Value
Parameter value	statisticsUpdatePeriod
Visible or not	Yes
Description of parameters	Command Statistics Persistence Period
Default value	0 Non-persistence
Min value	0
Max value	3600000
Whether Reload is valid or not	N for v.2.4.5 Y for v.2.4.7 and above
Min Compatible Version	2.4.3

Role of parameter:

This is the period of command statistics information persistence to the configDB.

If set as 0, the program will exit due to abnormality, and there will be no persistence; if the configured value is greater than 0, then periodical persistence could be made to the database, and could also be accumulated upon restart.

When executing SQL statement on Client, relevant commands will be counted into configDB. When set as 0, none will be counted into configDB.

```
mysql> use test_ct
```

Database changed

```
mysql> select * from tid;
```

Empty set (0.03 sec)

对象		hotdb_server_statistics @...				
	开始事务	备注	筛选	排序	导入	导出
type	statistics					
1	{"TEST_CT":{}}					
2	{"TEST_CT":{"TID":0}}					

14.1.111. strategyForRWSplit

Description of parameter:

Property	Value
Parameter value	strategyForRWSplit
Visible or not	Yes
Description of parameters	Enable Read/write splitting or not
Default value	0
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Role of parameter:

When the parameter is set as 0, it means not to enable Read/write splitting, and all will be read from the host.

```
mysql> select * from ss order by id;
```

```
+-----+-----+
```

```
| id | a      |
```

```
+-----+-----+
```

```
| 1 | master |
```

```
| 2 | master |
```

```
| 3 | master |
```

```
| 4 | master |
```

```
| 5 | master |
```

```
+-----+-----+
```

```
5 rows in set (0.00 sec)
```

When the parameter is set as 1, it means that splittable Read requests are sent to all available data sources, and according to the set read proportion of the Slave, read the Slave or Master.

```
mysql> select * from ss;
```

```
+-----+-----+
```

```
| id | a      |
```

```
+-----+-----+
```

```
| 1 | master |
```

```
| 2 | master |
```

```
| 3 | master |
```

```
| 4 | slave  |
```

```
| 5 | slave  |
```

```
+-----+-----+
```

```
5 rows in set (0.00 sec)
```

When the parameter is set as 2, it means that splittable Read requests are sent to available Slave data sources, and all Read requests beyond the transaction are sent to the Slave data source, while the Read requests within the transaction are sent to the Master data source.

➤ Beyond transaction:

```
mysql> select * from ss;
```

```
+-----+-----+
```

```
| id | a      |
```

```
+-----+-----+
```

```
| 1 | slave |
```

```
| 2 | slave |
```

```
| 3 | slave |
```

```
| 4 | slave |
```

```
| 5 | slave |
```

```
+-----+-----+
```

```
5 rows in set (0.00 sec)
```

➤ Within transaction:

```
mysql> select * from ss order by id;
```

```
+-----+-----+
```

```
| id | a      |
```

```
+-----+-----+
```

```
| 1 | master |
```

```
| 2 | master |
| 3 | master |
| 4 | master |
| 5 | master |
+----+-----+
5 rows in set (0.00 sec)
```

When the parameter is set as 3, it means that the Read requests in transaction before Write occurs are sent to available Slave data sources. Read requests beyond the transaction are sent to available Salve data sources.

```
mysql> begin
Query OK, 0 row affected (0.00 sec)
```

```
mysql> select * from ss;
```

```
+----+-----+
| id | a      |
+----+-----+
| 4 | slave |
| 5 | slave |
| 600004 | write |
| 600007 | write |
| 600013 | write |
| 1 | slave |
| 2 | slave |
| 3 | slave |
```

```
+-----+-----+
```

```
8 rows in set (0.00 sec)
```

```
mysql> insert into ss values(null,'write');
```

```
Query OK, 0 row affected (0.01 sec)
```

```
mysql> select * from ss;
```

```
+-----+-----+
```

```
| id | a |
```

```
+-----+-----+
```

```
| 1 | master |
```

```
| 2 | master |
```

```
| 3 | master |
```

```
| 600014 | write |
```

```
| 4 | master |
```

```
| 5 | master |
```

```
| 600004 | write |
```

```
| 600007 | write |
```

```
| 600013 | write |
```

```
+-----+-----+
```

```
9 rows in set (0.00 sec)
```

For details, please refer to [Read/write splitting](#).

14.1.112. switchByLogInFailover

Description of parameter:

Property	Value
Parameter value	switchByLogInFailover
Visible or not	No
Description of parameters	When failover, control whether to determine switch priority by Master_Log_File position of various data sources under the node or not
Default value	false
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.5

Parameter Setting:

```
<property name="switchByLogInFailover">false</property><!--When failover, select switch priority according to Read_Master_Log_Pos -->
```

Role of parameter:

True status: When failover, give priority to determining the switch priority via the Standby Slave synchronization speed, and the specific shall be determined by position of Master_Log_File and Read_Master_Log_Pos, and the switch with quicker synchronization speed will be taken in priority; if all Slave Read_Master_Log_Pos positions are the same, then match according to the set priority

False status: switch according to failover rule of the user

Note: Manual Switch operation is not under control of this parameter

14.1.113. switchoverTimeoutForTrans

Description of parameter:

Property	Value
Parameter value	switchoverTimeoutForTrans
Visible or not	Yes
Description of parameters	When making Manual Switch, the old transaction Wait Commit Timeout (ms)
Default value	3000
Min value	1800000
Max value	0
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Parameter Setting:

switchoverTimeoutForTransParameter in Server.xml is set as below:

```
<property name="switchoverTimeoutForTrans">3000</property>
```

Role of parameter:

At the time of manual Master/Slave switch, check whether there is old transaction timeout or not.

That is: Before manual execution of Master/Slave switch, enable Non-commit of execution transaction, then execute Manual Switch, and commit transaction with

Timeout, and then transaction commit succeeded. If the timeout time is exceeded, the front-end connection will be disconnected, and transaction will roll back automatically.

For example:

1. Set “switchoverTimeoutForTrans”Timeout as 36000ms
2. Enable transaction to execute Insert operation, make manual execution of Master/Slave switch, and commit transaction within 36000ms. Commit succeeded as follow:

```
mysql> begin;  
  
Query OK, 0 rows affected (0.00 sec)  
  
  
mysql> insert into TEST_001 values(1);  
  
Query OK, 0 rows affected (0.00 sec)  
  
  
mysql> commit;  
  
Query OK, 0 rows affected (0.00 sec)  
  
  
mysql> select * from TEST_001;  
+----+  
| id |  
+----+  
| 1 |  
+----+  
1 row in set (0.01 sec)
```

After committing the transaction, it's queried that id=1

3. Enable the transaction to execute Insert operation, make manual execution of Master/Slave switch. If the transaction is not committed within 36000 ms, then due to Commit timeout, the transaction will roll back as follow:

```
mysql> begin;  
  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> insert into TEST_001 values(2);  
  
Query OK, 0 rows affected (0.00 sec)
```

Execute Query statement one minute later:

```
mysql> select * from TEST_001;  
  
ERROR 2013 (HY000): Lost connection to MySQL server during query
```

ERROR 2016 (HY000): MySQL server has gone away

No connection. Trying to reconnect...

Connection id: 40672

Current database: test_jzl

Query after re-login, and it's found that the transaction is not committed:

```
mysql> select * from TEST_001;  
  
+----+  
| id |  
+----+  
| 1 |  
+----+  
  
1 row in set (0.01 sec)
```

14.1.114. timerExecutor

Description of parameter:

Property	Value
Parameter value	timerExecutor
Visible or not	Yes
Description of parameters	Number of threads of timers
Default value	4
Min value	2
Max value	8
Whether Reload is valid or not	No
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="timerExecutor">4</property><!-- Number of threads of timers -->
```

Role of parameter:

The parameter [adaptiveProcessor](#) is enabled by default, and when enabled, the compute node will make Automatic Adaptation to the Max timerExecutor. Log in to 3325 port, execute show @@threadpool; command, and you could view the number of current timerExecutor.

14.1.115. timestampProxy

Description of parameter:

Property	Value
Parameter value	timestampProxy
Visible or not	Yes
Description of parameters	TimeProxy mode
Default value	0
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.1

Parameter Setting:

When timestampProxy parameter is 0, it means auto mode, and when the compute node checks that the time difference of data source is greater than 0.5, it will be the auto Proxy of the Global Time Function. If less than 0.5, it only make Proxy of the time function of the Global Table, high-accuracy time stamp and cross-node statement.

```
<property name="timestampProxy">0</property>
```

When the parameter is set as 1, it means global_table_only, only in Global Table mode; the compute node only make Proxy of the time function of Global Table.

```
<property name="timestampProxy">1</property>
```

When the parameter is set as 2, it means all, in Global mode, and the compute node will make Proxy of the Global Time Function. **Role of parameter:**

```
<property name="timestampProxy">2</property>
```

Role of parameter:

This parameter is used for Complete Global Proxy of the table with on update current_timestamp property or SQL with Time Function, and for solving the problem of data abnormality and inter-node data inconsistency due to insert or update operation when the table has this property. If timestampProxy is set as 0 and the difference is great, or is set as 2, it will greatly influence execution speed and efficiency of all update statements.

14.1.116. url & username & password**Description of parameter:**

Property	Value
Parameter value	url
Visible or not	Yes
Description of parameters	configDB address
Default value	jdbc:mysql://127.0.0.1:3306/hotdb_config
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Property	Value
Parameter value	username
Visible or not	Yes
Description of parameters	configDB username

Property	Value
Default value	hotdb_config
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Property	Value
Parameter value	password
Visible or not	Yes
Description of parameters	configDB password
Default value	hotdb_config
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Role of parameter:

url, username and password are supporting parameters: url is the configDB route storing compute node configuration information; username and password are username and password for connecting this physical database. This configDB is used for storing configuration information.

```
<property name="url">jdbc:mysql://192.168.200.191:3310/hotdb_config</property><!-- Master configDB address
-->

<property name="username">hotdb_config</property><!-- Master configDB username -->

<property name="password">hotdb_config</property><!-- Master configDB password -->

<property
```

This username and password shall be created in MySQL instance, and could log in to this configDB only after being granted with privilege. Both the username and password could be customized.

```
mysql> grant select,insert,update,delete,super on *.* to hotdb_config_9@'%' identified by 'hotdb_config_9';
```

```
Query OK, 0 row affected (0.00 sec)
```

```
root@127.0.0.1:(none) 5.7.19-HotDB-2.5.2 04:20:08> show grants;
```

```
+-----+ | Grants for root@localhost | +-----+
```

```
| GRANT ALL PRIVILEGED ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
```

```
| GRANT PROXY ON "@" TO 'root'@'localhost' WITH GRANT OPTION | +-----+
```

```
1 row in set (0.07 sec)
```

```
root> mysql -uhotdb_config_9 -photdb_config_9 -h127.0.0.1 -P3306
```

```
mysql: [Warning] Using a password on the command line interface can be insecure.
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

```
Your MySQL connection id is 16323
```

```
Server version: 5.7.19-HotDB-2.5.2 HotDB Server by Hotpu Tech
```

```
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

When configDB connection fails after enabling the compute node, the compute node will reconnect at interval of 3s, until connection still fails after more than 30 minutes of retry, then interrupt Enable operation.

The last packet set successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.

2018-06-12 15:25:56.789 [ERROR] [INIT] [main] HotdbConfig(275) – no available config datasources. retry in 3 seconds.

14.1.117. usingAIO

Description of parameter:

Property	Value
Parameter value	usingAIO
Visible or not	No
Description of parameters	Use AIO or not, Yes
Default value	0
Whether Reload is valid or not	No
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="usingAIO">0</property><!--Use AIO or not, Yes: 1, No: 0 -->
```

When the parameter is set as 0, the compute node uses NIO, and marks opposition between AIO and NIO

Role of parameter:

It's used for setting whether the current compute node enables AIO or not

AIO: Asynchronous Input/Output, and the server realization mode is to create a thread for a valid request, and all I/O requests of the Client are firstly completed by OS and then informed to the server application to enable thread for processing, IO mode is applicable to the architecture with many connections and long connections (heavy operation). Since at present, AIO hasn't been completed on Linux and optimization of AIO by compute node is far inferior to NIO, therefore, it's not recommended to enable this parameter.

```
root> tail -n 300 hotdb.log | grep 'aio'

2018-06-01 13:51:18.961 [INFO] [INIT] [main] j(-1) – using aio network handler

2018-06-01 13:52:19.644 [INFO] [INIT] [main] j(-1) – using aio network handler
```

14.1.118. version

Description of parameter:

Property	Value
Parameter value	version
Visible or not	No
Description of parameters	The version number shown to the public by the compute node
Default value	Synchronize with result of the compute node show @@version, for example:5.6.29-HotDB-2.5.1

Property	Value
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Role of parameter:

The version number shown by the compute node to the public, which could be modified by Custom, and it could specify relevant connection protocol of inferior versions.

```
<property name="version">5.6.1</property><!—Version Number -->
```

When log in to MySQL instance, you could view corresponding version number:

```
root> mysql -uct -pct -h127.0.0.1 -P2473

mysql: [Warning] Using a password on the command line interface can be insecure.

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 30

Server version: 5.6.1-HotDB-2.4.7 HotDB Server by Hotpu Tech

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
root@127.0.0.1:(none) 5.7.19-HotDB-2.5.2 04:20:14> select version();
```

```
+-----+
```

```
| VERSION()           |
+-----+
| 5.6.1-HotDB-2.4.7 |
+-----+
1 row in set (0.03 sec)
```

Note: When min version number of all data source is inferior to or equals to Max version number supported by the compute node, then among all data sources, show the min version number to the public; when version number of the data source exceeds the Max version number supported by the compute node, then show a complete version number of the Max protocol version supported by the compute node to the public, and currently, 5.7.18 is supported at the highest. When version number of all data sources is bigger than 5.7.18, then this parameter will change the version number, otherwise, the min version among the data sources will be taken directly.

14.1.119. **versionComment**

Description of parameter:

Property	Value
Parameter value	versionComment
Visible or not	No
Description of parameters	Version comment of compute node.
Default value	(None)
Whether Reload is valid or not	Yes
Min Compatible Version	2.5.5

Parameter Setting:

versionComment, the version comment of compute node for external display, can be customized and used with the parameter Version. If the parameter value is a different string, you can replace the original version comment with the configured string; if you do not want to display any comment, you can configure the parameter value as a space (and the actual display is a space); if no value is configured, HotDB-2.5.5 HotDB Server by Hotpu Tech will be displayed by default (2.5.5 is the version number of the compute node itself).

For example:

Null: <property name="versionComment"></property>, connect to the compute node:

```
[root@hotdb]# mysql -uroot -proot -P3323 -h192.168.210.49
```

mysql: [Warning] Using a password on the command line interface can be insecure.

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 235

Server version: 5.7.23 HotDB-2.5.3 HotDB Server by Hotpu Tech

.....

A space: <property name="versionComment"></property>, connect to the compute node:

```
[root@hotdb]# mysql -uroot -proot -P3323 -h192.168.210.49
```

mysql: [Warning] Using a password on the command line interface can be insecure.

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 235

Server version: 5.7.23

.....

A customized string: <property name="versionComment">hotpu</property>, connect to the compute node:

```
[root@hotdb]# mysql -uroot -proot -P3323 -h192.168.210.49
```

mysql: [Warning] Using a password on the command line interface can be insecure.

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 235

Server version: 5.7.23 hotpu

.....

Note: the status result after connection and the prompt when the client is connecting the compute node will both be displayed according to the version comment.

.....

```
root@192.168.210.49:(none) 5.7.23 08:41:42> status;
```

```
-----  
mysql Ver 14.14 Distrib 5.7.21, for linux-glibc2.12 (x86_64) using EditLine wrapper
```

Connection id: 444

Current database:

Current user: root@192.168.210.49

SSL: Not in use

Current pager: stdout

Using outfile: "

Using delimiter: ;

Server version: 5.7.23 hotpu

Protocol version: 10

Connection: 192.168.210.49 via TCP/IP

.....

14.1.120. VIP & checkVIPPeriod

Description of parameter:

Property	Value
Parameter value	VIP
Visible or not	Yes
Description of parameters	Virtual IP address
Default value	Null
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.8

Property	Value
Parameter value	CheckVIPPeriod
Visible or not	Yes
Description of parameters	Check VIP Period
Default value	500ms
Min value	10ms
Max value	1000ms
Whether Reload is valid or not	Yes

Property	Value
Min Compatible Version	2.4.8

VIP and checkVIPPPeriod are supporting parameters; VIP is set as Keepalived virtual IP, and checkVIPPPeriod is used for controlling check frequency of virtual IP. When the compute node enables VIP check, if the compute node in slave status finds existence of VIP, then it will execute online automatically; if compute node in master status finds that VIP does not exist, then it will auto offline. This group of parameters are applicable to Compute Node High Availability environment, and it's recommended making configuration under compute node Master/Slave node environment, and shall be set as actual virtual IP of the current Keepalived. If not set or in case of set error, there will be no processing, while this parameter could be ignored in single compute node

Parameter Setting:

VIP Parameter of Server.xml is set as IP of Keepalived; CheckVIPPPeriod is the check period, and the unit: ms

```
<property name="VIP">192.168.220.106</property><!--           virtual IP (Null if not filled or if the format is
not IPv4)          -->
<property name="checkVIPPPeriod">500</property><!--           virtual IP check period (If VIP is valid, check
VIP period, unit: ms)          -->
```

View configuration script of Keepalived:

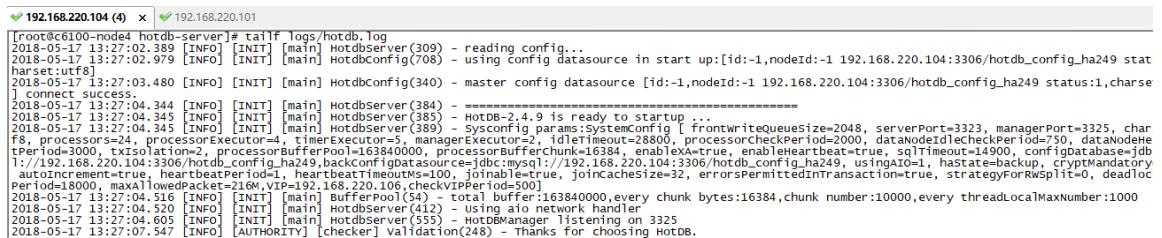
```
cat /etc/keepalived/keepalived.conf
```

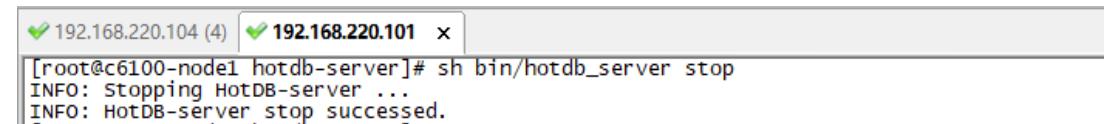
Determine corresponding IP:

```
virtual_ipaddress {
    192.168.220.106/24 dev bond0 label bond0:1
}
```

Role of parameter:

It is in Compute Node High Availability environment, and under the condition of change in root password, when making high availability switch, it will make switch by checking according to existence mode of VIP, in order to avoid switch failure after changing the password

A terminal window titled '192.168.220.104 (4)' showing log output from the HotDB server. The logs indicate the server is reading config files, initializing, and starting up. It shows various configuration parameters like port numbers, chunk sizes, and connection details.

A terminal window titled '192.168.220.104 (4) 192.168.220.101' showing the command 'sh bin/hotdb_server stop'. The logs show the server stopping successfully.

14.1.121. waitConfigSyncFinish

Description of parameter:

Property	Value
Parameter value	cryptMandatory
Visible or not	No
Description of parameters	When enabled, wait for configDB synchronization or not

Property	Value
Default value	false
Whether Reload is valid or not	No
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="waitConfigSyncFinish">true</property><!-- When enabled, wait for configDB synchronization  
or not -->
```

Role of parameter:

It is used for setting whether to wait for configDB synchronization or not when enabled.

Turn on the switch, in case of latency between configDB master/slave, it shall wait for the master/slave configDB to catch up with the Slave configDB, and maintain that the configDB data currently used is the latest data, only in this way could it be enabled

OFF status: Enable successfully when there is latency between master/slave configDB

```
2018-06-01 16:21:14.958 [INFO] [INIT] [main] j(-1) - reading config...
```

```
2018-06-01 16:21:15.170 [info] [INIT] [main] a(-1) - using config datasource in start up:[id:-1,nodeId:-1  
127.0.0.1:3306/hotdb_config_249 status:1;charset:utf8]
```

```
2018-06-01 16:21:15.518 [info] [INIT] [main] a(-1) - master config datasource [id:-1,nodeId:-1  
127.0.0.1:3306/hotdb_config_249 status:1;charset:utf8] connect success.
```

```
2018-06-01 16:21:16.892 [info] [INIT] [main] j(-1) -
```

```
2018-06-01 16:21:16.893 [info] [INIT] [main] j(-1) - HotDB-2.4.9 is ready to startup ...
```

```
2018-06-01 16:21:16.894 [info] [INIT] [main] j(-1) - Sysconfig params:SystemConfig  
[ frontwriteQueueSize=2048, service port=9993, management port=999S, charset=utf8, processors=8,  
processorExecutor=4, timerExecutor=4, managerExecutor=2, idleTimeout=28800, processorcheckPeriod=1000,  
dataNodeIdleCheckPeriod=120, dataNodeHeartbeatPeriod=3000, txIsolation=2, processorBufferPool=163840000,
```

```
processorBufferchunk=16384, enableXA=false, enableHeartbeat=true, sqlTimeout=42100,
configDatabase=jdbc:mysql://l27.0.0.1:3306/hotdb_config_249,backConfigDatasource=jdbc:mysql://l27.0.0.1:3306
/botdb_config_249, usingAIO=0, hastate=master, cryptMandatory=false, autoIncrement=true, heartbeatPeriod=1,
heartbeatTimeoutMs=100, joinable=true, joincachesize=4, errorsPermittedInTransaction=true,
strategyForRWSplit=3, deadlockCheckPeriod=0, maxAllowedPacket=64M,viP=null,checkVIPPeriod=l600]

2018-06-01 16:21:17.210 [info] [INIT] [main] BufferPool(-1) - total buffer:163840000,every chunk
bytes:16384,chunk number:10000,every threadLocalMaxNumber:10000

2018-06-01 16:21:17.216 [INFO] [INIT] [main] j(-1) - usinq aio network handler
```

ON status:

It could be enabled only after waiting for master/slave synchronization

```
2018-07-12 14:28:52.019 [INFO] [INIT] [$NIOREACTOR-9-RW] XAInitRecoverHandler(125) – wait for config
datasource synchronizing...
```

14.1.122. waitForSlaveInFailover

Description of parameter:

Property	Value
Parameter value	waitForSlaveInFailover
Visible or not	Yes
Description of parameters	In failover, whether to wait for the Slave to catch up with replication or not
Default value	true
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="waitForSlaveInFailover">true</property><!-- In failover, whether to wait for the Slave to catch up with replication or not -->
```

Role of parameter:

It's used for setting in failover, whether to wait for the Slave to catch up with replication or not

ON status:

When the Slave has replication latency, it can't switch to the Slave, and the compute node will keep checking, and could make switch only after waiting for replication synchronization

```
mysql> show @@latency;

+-----+-----+-----+
| dn | info | latency |
+-----+-----+-----+
| 4 | 192.168.200.51:3310/phy248 | 192.168.200.51:3310/phy248 | 0 ms |
| 5 | 192.168.200.51:3311/phy248 | 192.168.200.51:3311/phy248 | 0 ms |
| 6 | 192.168.200.51:3312/phy248 | 192.168.200.51:3312/phy248 | 19582 ms |
| 7 | 192.168.200.51:3313/phy248 | 192.168.200.51:3313/phy248 | 0 ms |
+-----+-----+-----+
4 rows in set (0.02 sec)
```

In the log you can see the prompt of no longer using the failed Master data source, and the data source without replication synchronization will not be enabled

```
2018-06-08 10:36:47.921 [INFO] [FAILOVER] [Labor-1552] j(-1) - slave_sql_running is Yes in :[id:178,nodeId:6
192.168.200.52:3312/phy248 status:1,charset:utf8] during failover of datanode 6

2018-06-08 10:36:48.417 [WARN] [HEARTBEAT] [$NIOConnector] m(-1) - datasoruce 6
192.168.200.51:3312/phy248 init heartbeat failed due to: Get backend connection
failed:java.net.ConnectException:connection refused
```

```
2018-06-08 10:36:48.418 [WARN] [HEARTBEAT] [$NIOConnector] m(-1) - datasoruce 6
192.168.200.51:3312/phy248 init heartbeat failed due to: Get backend connection
failed:cn.hotpu.hotdb.h.l:java.net.connectException: connection refused

2018-06-08 10:36:48.918 [WARN] [HEARTBEAT] [$NIOConnector] m(-1) - datasoruce 6
192.168.200.51:3312/phy248 init heartbeat failed due to: Get backend connection failed:j
ava.net.ConnectException: connection refused

2018-06-08 10:36:48.918 [WARN] [HEARTBEAT] [$NIOConnector] m(-1) - datasoruce 6
192.168.200.51:3312/phy248 init heartbeat failed due to: Get backend connection
failed:cn.hotpu.hotdb.h.l:java.net.connectException: connection refused

2018-06-08 10:36:48.982 [INFO] [FAILOVER] [Labor-1552] j(-1) - masterLogFile:mysql-
bin.000518,readMasterLogFile:mysql-bin.000518,readMasterLogPos:384545127,execMaster
LogPos:384512435,relayLogFiTe:mysql-relay-bin.000002,relayLogPos:
248414,secondBehindMaster:19,execLogchanged:true in slave: MySQLConnection [node=6, id=140, threadId=3
15945, state=borrowed, closed=false, autocommit=true, host=192.168.200.52, port=3312, database=phy248,
localPort=64694, isClose:false, toBeClose:false]
```

OFF status:

When the master/slave data source has replication latency, it could switch to the Slave directly, without waiting for replication synchronization.

```
2018-06-08 16:19:22.864 [INFO] [FAILOVER] [Labor-1852] bh(-1) – switch datasource:6 for datanode:6
successfully by Manager.
```

Special note: the effect of master_delay on switching is adjusted in 2.5.6 and above.

When the parameter waitForSlaveInFailover

(In failover, whether to wait for the Slave to catch up with replication or not) is enabled, if the delay setting is detected during switching, the setting will be automatically cancelled before catching up with the replication. After the switching, the delay setting will be restored. If it is still greater than 10s after cancelling the delay setting, switching will not be allowed, and the previously set value of master_delay will be restored.)

14.1.123. waitSyncFinishAtStartup

Description of parameter:

Property	Value
Parameter value	waitSyncFinishAtStartup
Visible or not	Yes
Description of parameters	When enabled, wait for synchronization of the Master data source or not
Default value	true
Whether Reload is valid or not	No
Min Compatible Version	2.4.3

Parameter Setting:

```
<property name="waitSyncFinishAtStartup">true</property><!-- When enabled, wait for synchronization of the Master data source or not -->
```

Role of parameter:

When enabled, wait for synchronization of the Master data source or not. Turn on the switch, wait for master/slave data source to make replication synchronization when enabling the compute node, so as to guarantee data consistency of master/slave data source

Precondition:

Under the condition of Master data source latency, enabling compute node will prompt that the replication synchronization of the current data source has not been completed, and it shall provide service only after replication synchronization completes.

Turn on the switch: When enable the compute node, wait for replication synchronization of master/slave data source, so as to guarantee data consistency of master/slave data source

```
2018-06-01 17:15:12.990 [info] [INIT] [$NIOREACTOR-3-RW] k(-1) - masterLogFile:mysql-bin.000667,relayMasterLogFile:mysql-bin.000667,readMasterLogPos:4668659,execMasterLogPos:4555931,relayLogFile:mysql-relay-bin.000004,relayLogPos: 2121597,secondBehindMaster:90,execLogchanged:true in server:MySQLConnection [node=3, id=41, threadId=l7054, state=running, closed=false, autocommit=true, host=192.168.200.52, port=3310, database=db249, localPort=18965, isClose:false, toBeClose:false]

2018-06-01 17:15:12.990 [info] [INIT] [$NIOREACTOR-3-RW] k(-1) - masterLogFile:mysql-bin.000667,relayMasterLogFile:mysql-bin.000667,readMasterLogPos: 4669275,execMasterLogPos:4555931,relayLogFile:mysql-relay-bin.000004,relayLogPos: 2121597,secondBehindMaster:90,execLogchanged:true in server:MySQLConnection [node=3, id=50, threadId=l7084, state=running, closed=false, autocommit=true, host=192.168.200.52, port=3310, database=db249, localPort=20329, isClose:false, toBeClose:false]

2018-06-01 17:15:12.990 [info] [INIT] [$NIOREACTOR-3-RW] k(-1) - masterLogFile:mysql-bin.000667,relayMasterLogFile:mysql-bin.000667,readMasterLogPos: 4670199,execMasterLogPos: 4557471,relayLogFile:mysql-relay-bin.000004,relayLogPos: 2122521,secondBehindMaster:90,execLogchanged:true in server:MySQLConnection [node=3, id=41, threadId=l7054, state=running, closed=false, autocommit=true, host=192.168.200.52, port=3310, database=db249, localPort=18965, isClose:false, toBeClose:false]
```

Turn off the switch: No other abnormalities, the compute node could be enabled directly

```
2018-06-01 16:21:14.958 [INFO] [INIT] [main] j(-1) - reading config...

2018-06-01 16:21:15.170 [info] [INIT] [main] a(-1) - using config datasource in start up:[id:-1,nodeld:-1 l27.0.0.1:3306/hotdb_config_249 status:1;charset:utf8]

2018-06-01 16:21:15.518 [info] [INIT] [main] a(-1) - master config datasource [id:-1,nodeld:-1 l27.0.0.1:3306/hotdb_config_249 status:1;charset:utf8] connect success.

2018-06-01 16:21:16.892 [info] [INIT] [main] j(-1) -
=====
```

```

2018-06-01 16:21:16.893 [info] [INIT] [main] j(-1) - HotDB-2.4.9 is ready to startup ...

2018-06-01 16:21:16.894 [info] [INIT] [main] j(-1) - Sysconfig params:SystemConfig
[ frontwriteQueueSize=2048, service port=9993, management port=9995, charset=utf8, processors=8,
processorExecutor=4, timerExecutor=4, managerExecutor=2, idleTimeout=28800, processorCheckPeriod=1000,
dataNodeIdleCheckPeriod=120, dataNodeHeartbeatPeriod=3000, txIsolation=2, processorBufferPool=163840000,
processorBufferChunk=16384, enableXA=false, enableHeartbeat=true, sqlTimeout=42100,
configDatabase=jdbc:mysql://127.0.0.1:3306/hotdb_config_249,backConfigDatasource=jdbc:mysql://127.0.0.1:33
06/hotdb_config_249, usingAIO=o, hastate=master, cryptMandatory=false, autoIncrement=true,
heartbeatPeriod=1, heartbeatTimeoutMs=100, joinable=true, joinCacheSize=4, errorsPermittedInTransaction=true,
strategyForRWSplit=3, deadlockCheckPeriod=0, maxAllowedPacket=64M, VIP=null, checkVIPPeriod=1600]

2018-06-01 16:21:17.210 [info] [INIT] [main] BufferPool(-1) - total buffer:163840000,every chunk
bytes:16384,chunk number:10000,every threadLocalMaxNumber:1000

2018-06-01 16:21:17.216 [info] [INIT] [main] j(-1) - using aio network handler

```

14.1.124. weightForSlaveRWSplit

Description of parameter:

Property	Value
Parameter value	weightForSlaveRWSplit
Visible or not	Yes
Description of parameters	Read Proportion of the Slave, 50 by default (percentage)
Default value	50
Whether Reload is valid or not	Yes
Min Compatible Version	2.4.4

Parameter Setting:

weightForSlaveRWSplitParameter in server.xml is set as 50

```
<property name="weightForSlaveRWSplit">50</property>
```

Role of parameter:

weightForSlaveRWSplit and strategyForRWSplit are supporting parameters, and only when Read/write splitting strategy is 1 (separatable Read requests are sent to all available data sources), could Read Proportion of the Slave be meaningful. If the Slave latency exceeds threshold value of readable Standby Slave, read the Active Master by default

Under master/slave condition: read proportion of the Slave is 50% by default

Under one-master and multi-slave condition (such as: one master and double slaves): read proportion of the Master is 50%, read proportion of Slave A is 25%, and read proportion of Slave B is 25%

For example: Active Master mark: name= Master

```
mysql> select * from vrab001;
```

id	name
1	Master
2	Master
3	Master
4	Master

Slave mark: name= Slave

```
mysql> select * from vrab001;
```

```
+-----+-----+
```

```
| id | name |
```

```
+-----+-----+
```

```
| 1 | slave |
```

```
| 2 | slave |
```

```
| 3 | slave |
```

```
| 4 | slave |
```

```
+-----+-----+
```

Execute select Query operation for several times, read 50% of master/slave respectively

15. Appendix

15.1. Notices for HotDB Server

15.1.1. Recommendation on JDBC version

It's recommended that the JDBC version should be mysql-connector-java-5.1.27.jar, 8.0 could be compatible at the highest.

15.1.2. Recommendation on JAVA database connection pool

It's recommended using proxool-0.9 for connection pool

15.1.3. Reserved field of database design

Compute node could specify data node according to DNID, therefore, the DNID Field name is reserved field of the database (do not use this Field name in table structure).

The compute node judges whether the data source is available or not via operating the hotdb_heartbeat table of the data source, therefore, hotdb_heartbeat serves as reserved field of Table Name.