

FROM INTRODUCTION TO PRACTICE

Lesson 4.1: OceanBase Database Migration and Synchronization

Peng Wang
OceanBase Global Technical Evangelist



Agenda



- **Lesson 4.1**
 - **Introduction to OceanBase and MySQL Compatibility**
 - **Introduction to OMS Migration and Synchronization**
 - **OBLogProxy Introduction**
- **Lesson 4.2**
 - **OBDUMPER/OBLOADER Introduction**
 - **Introduction to SQL Command Export**
 - **Introduction to Other Ecological Components**
 - **Comparison of Components**

OceanBase and MySQL Compatibility

Data Types

**Character Set
&
Collation**

Index

**Partition
Support**

Other



Compatibility Comparison

Data Type Compatibility Comparison

- Type Not Currently Supported

Type	MySQL	OceanBase	Description
SERIAL	Support	Not supported	SERIAL is an alias for BIGINT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE

- String Type Comparison

Type	MySQL 8	OceanBase 4.2
CHAR	255 characters	256 characters
VARCHAR	65535 characters (Actually around 16383)	262144 characters
BINARY	255 bytes	256 bytes
VARBINARY	65535 bytes (Actually around 65532)	1048576 bytes
TINYBLOB	255 bytes	255 bytes
BLOB	65535 bytes	65536 bytes
MEDIUMBLOB	16777215 bytes	16777216 bytes
LONGBLOB	4294967295 bytes (4GB)	536870911 bytes
TINYTEXT	255 bytes	255 bytes
TEXT	65535 bytes	65536 bytes
MEDIUMTEXT	16777215 bytes	16777216 bytes
LONGTEXT	4294967295 characters (4GB)	536870911 bytes

Character Set Compatibility Comparison

- The Following Charset and Collation Are Supported:

Character Set	Collation	Description
utf8mb4	utf8mb4_general_ci	Use common sorting rules
utf8mb4	utf8mb4_bin	Use binary collation
utf8mb4	utf8mb4_unicode_ci	Use a collation based on the Unicode Collation Algorithm (UCA)
binary	binary	Use binary collation
gbk	gbk_chinese_ci	Use Chinese language sorting rules
gbk	gbk_bin	Use binary collation
utf16	utf16_general_ci	Use common sorting rules
utf16	utf16_bin	Use binary collation
utf16	utf16_unicode_ci	Use a collation based on the Unicode Collation Algorithm (UCA)
gb18030	gb18030_chinese_ci	Use Chinese language sorting rules
gb18030	gb18030_bin	Use binary collation

Character Set	Collation	Description
latin1	latin1_swedish_ci	Use Swedish/Finnish collation
latin1	latin1_bin	latin1 uses binary collation
gb18030_2022	gb18030_2022_bin	Use binary collation
gb18030_2022	gb18030_2022_chinese_ci	Use pinyin collation rules. Case-insensitive. The default collation for this character set in MySQL mode
gb18030_2022	gb18030_2022_chinese_cs	Use pinyin sorting rules. Case sensitive
gb18030_2022	gb18030_2022_radical_ci	Use radical stroke sorting rules. Not case sensitive
gb18030_2022	gb18030_2022_radical_cs	Use radical stroke sorting rules. Case sensitive
gb18030_2022	gb18030_2022_stroke_ci	Use the stroke order. Not case sensitive
gb18030_2022	gb18030_2022_stroke_cs	Use the stroke order. Case sensitive

Index Comparison

- The Following Index Types Are Not Currently Supported

Index Type	Index Data Structure	MySQL	OceanBase
Index Extension	B-tree	Support	Not supported
Descending Index	B-tree	Support	Not supported
Full-text Index	B-Tree	Support	Not supported
HASH Index	B-Tree	Support	Not supported
LOCK Option	/	Support	Not supported
Index Merge	B-Tree	Support	Not supported

Partition Compatibility Comparison

OceanBase	MySQL
The OceanBase database supports primary partitioning, templated and non-templated secondary partitioning	MySQL database does not support non-templated secondary partitions
The secondary partitioning of OceanBase database supports Hash, Key, Range, Range Columns, List and List Columns partitioning	The secondary partitioning of the MySQL database only supports Hash partitioning and Key partitioning
Truncate partition without locking the table	Truncate partition locks the table

Note: For more information on compatibility, please visit the official website: <https://en.oceanbase.com/>

其他兼容性对比



Procedural Language >

- OceanBase is compatible with most PL functions of MySQL databases:
 - ✓ Data types
 - ✓ Stored procedures
 - ✓ Custom functions
 - ✓ Triggers
 - ✓ Exception handling
- In addition, OceanBase database-specific MySQL PL system packages, including DBMS_RESOURCE_MANAGER, DBMS_STATS, DBMS_UDR, DBMS_XPLAN, and DBMS_WORKLOAD_REPOSITORY, etc



SQL_MODE >

Currently, in OceanBase V4.2.1, all SQL_MODE are supported.



Backup and Recovery >

- Cluster-level backup and recovery are not supported. Because OceanBase is a multi-tenant architecture, it only promotes tenant-level backup and recovery
- Cold backup is not supported
- Does not support validity verification of backup data
- Backup and recovery of partial databases within tenants are not supported

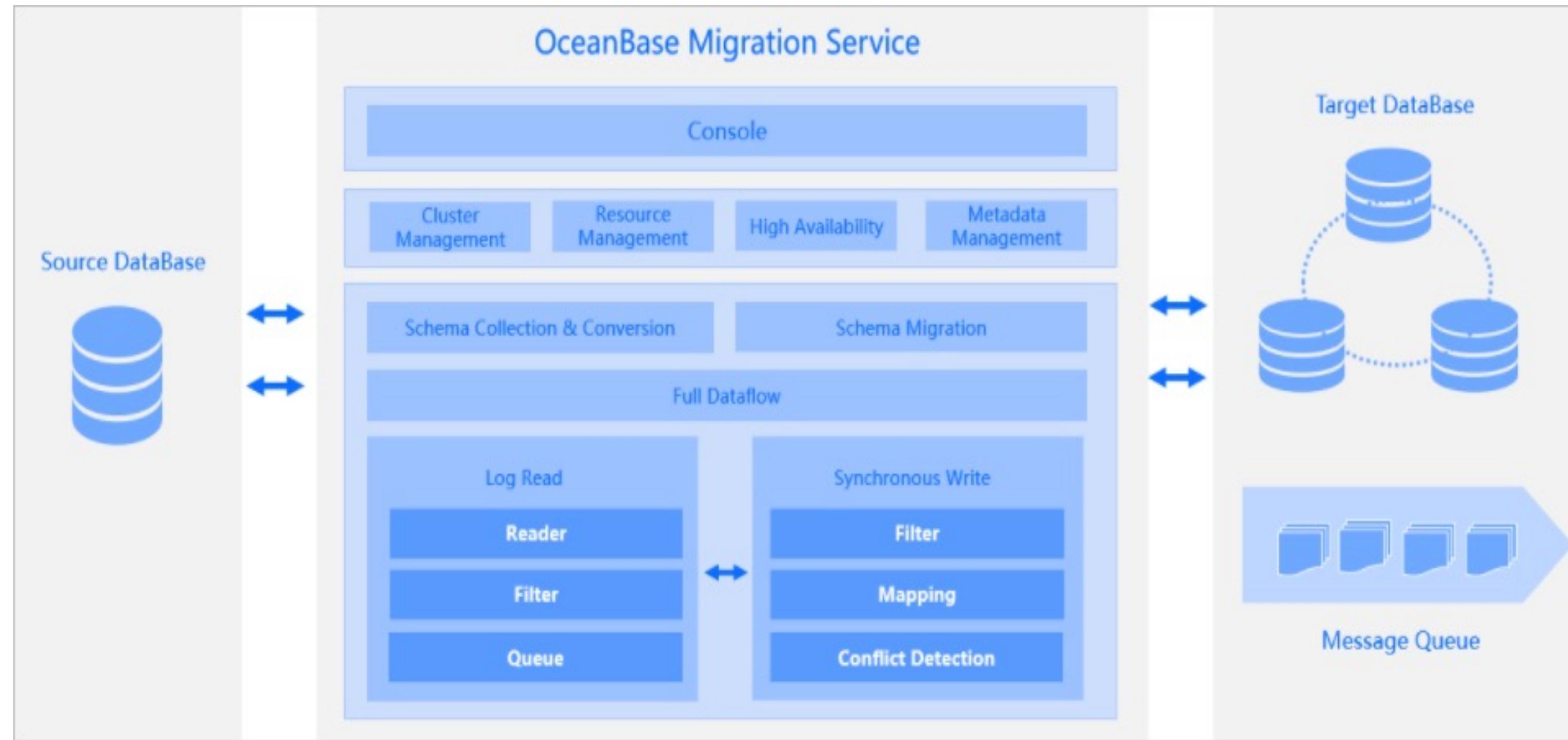
Agenda



- **Lesson 4.1**
 - Introduction to OceanBase and MySQL Compatibility
 - **Introduction to OMS Migration and Synchronization**
 - OBLogProxy Introduction
- **Lesson 4.2**
 - OBDUMPER/OBLOADER Introduction
 - Introduction to SQL Command Export
 - Introduction to Other Ecological Components
 - Comparison of Components

OMS Introduction

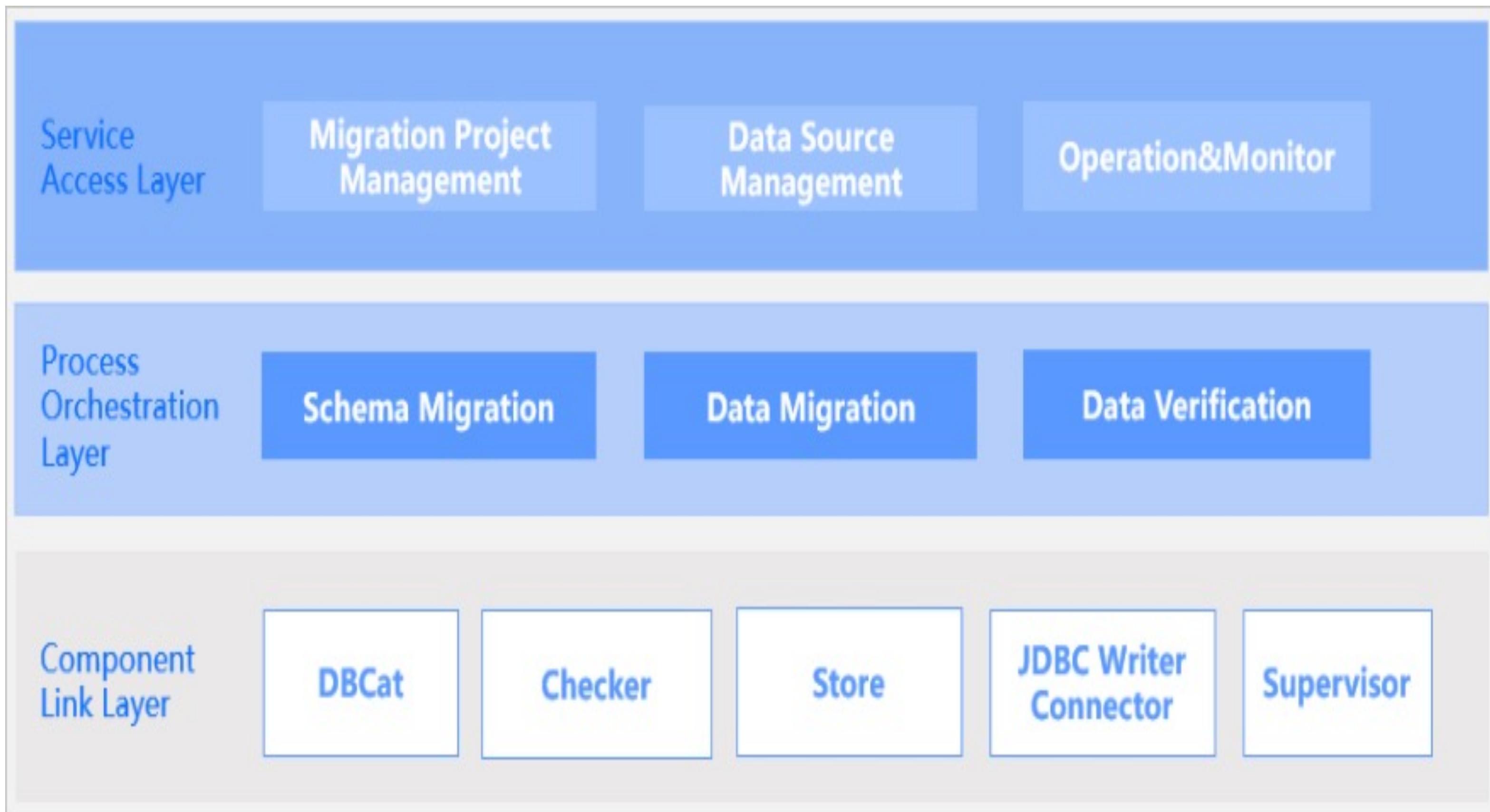
- OceanBase Migration Service (OMS) is a service provided by the OceanBase database that supports data interaction between homogeneous or heterogeneous data sources and the OceanBase database. It can migrate existing data online and synchronize incremental data in real-time.



The OMS Community Edition mainly includes:

- Management console: one-stop migration scheduling
- DBCat: Data object acquisition and conversion component
- Incremental pull component Store, incremental synchronization component Incr-Sync, full import component Full-Import, and full verification component Full-Verification
- Basic service components: cluster management, resource pool management, high availability components, metadata management, and other components to ensure efficient scheduling and stable operation of the migration module

Introduction to OMS Layering Functions



- Service Access Layer
- Process Orchestration Layer
- Component Layer
 - **DBCat:** Core components of OMS Community Edition structure migration
 - **Store:** Incrementally pull component
 - **Full-Import:** Full Import Component
 - **Incr-Sync:** Incremental Sync Component
 - **Full-Verification:** Full verification component
 - **Supervisor:** It is responsible for monitoring the above components

Component Terminology Introduction

Type	Terminology	Description
Control Components	Ghana	OMS Community Edition console
	CM	OMS Community Edition cluster management service
	Supervisor	The OMS community version proxy service is responsible for receiving and executing instructions issued by the cluster management service
Incremental Parsing Component	Store	It is used to pull and parse database transaction logs and store the parsed log records in a unified format. Downstream can consume incremental data from the Store consistently without worrying about the differences in log formats between different databases
	LogProxy	OceanBase database log agent
Framework Components	Incr-Sync	Incremental Synchronization Component: requests incremental data from the incremental pull component, filters data based on the synchronization object configured by the user, and then synchronizes log records to the target instance while ensuring transaction timing and transaction consistency
	Full-Import	Full Import Component: responsible for the migration of existing data in source library table objects and the synchronization of some incremental data
	Full-Verification	Full Validation Component: responsible for performing full field validation on row records in the migration table and generating correction statements for inconsistent data
Other	MetaDB	The OceanBase database service is deployed based on containers and has a built-in OceanBase database agent for direct external access. MetaDB can store metadata for products such as OCP, ODC, and OMS.
	InfluxDB	A container-based time series database service for storing monitoring data of the OMS Community Edition data transmission project

Limitation of Use

- In the data migration and data synchronization functions, the applicable versions of OceanBase Community Edition and other data terminals are as follows

Type	Data Migration	Data Synchronization
OceanBase Community Edition	V3.1.0–CE、V3.1.1–CE、V3.1.2–CE、V3.1.3–CE、V3.1.4–CE、3.1.5–CE、V4.0.0–CE、V4.1.0–CE、V4.2.0–CE、V4.2.1–CE、V4.2.2–CE	V3.1.0–CE、V3.1.1–CE、V3.1.2–CE、V3.1.3–CE、V3.1.4–CE、3.1.5–CE、V4.0.0–CE、V4.1.0–CE、V4.2.0–CE、V4.2.1–CE、V4.2.2–CE
Other data terminals	<ul style="list-style-type: none">MySQL: V5.5、V5.6、V5.7、V8.0MariaDB: V10.2TiDB: V3.x、V4.x、V5.x、V6.x、V7.xPostgreSQL: V10.x、V12.xGreenPlum: 4HBase: 1.2.0–cdh5.15.2	<ul style="list-style-type: none">Kafka: V0.9、V1.0、V2.xRocketMQ: V4.7.1

OMS Deployment Requirements

- Environmental Requirements
 - Install Docker, the version required is not less than V1.13
 - Prepare a database to store the metadata of OMS Community Edition. It can be MySQL or OceanBase. 2C/4G is enough
- Standard resource specifications for basic services and management

System Resources	Basic Services	Control	Sum
CPU	1C	4C	5C
Memory	1G	7G	8G
Disk	100G	100G	200G

- Database to database migration and synchronization

System Resources	Store	Incr-Sync	Full-Import	Sum
CPU	4C	2C	2C	8C
Memory	8G	8G	4G	20G
Disk	Depends on the production speed of incremental logs on the database side and the storage time of incremental data in OMS Community Edition	5G	5G	>10G

OMS Deployment Requirements

- Data synchronization from database to big data

System Resources	Store	Incr-Sync	Full-Import	Sum
CPU	4C	2C	2C	8C
Memory	8G	4G	4G	16G
Disk	Depends on the production speed of incremental logs on the database side and the storage time of incremental data in OMS	5G	5G	>10G

- Deployment Type
 - Single-node deployment
 - Multi-node deployment in a single region
 - Multi-node deployment in multiple regions
- InfluxDB
 - If you need OMS Community Edition to collect and display historical monitoring data, please deploy InfluxDB time series database

Single Node Deployment Demonstration

1. Download Installation Package

【 OceanBase official website - Enterprise Services - Software and Tools Download 】 <https://en.oceanbase.com/softwarecenter>

Migration Tools

oblogproxy V2.0.2 ▾
Provide data migration services between OceanBase Community Edition and heterogeneous MySQL database
[↓ X86 version- el7 ▾](#) [↓ ARM version- el7 ▾](#)

OceanBase Loader and Dumper V4.3.1 ▾
Data export tool tailored for OceanBase Community Edition
[↓ X86 version](#)

OceanBase Migration Service V4.2.4_CE ▾
Provide data migration services between OceanBase Community Edition and heterogeneous MySQL database
[↓ X86 version](#)

2. Load Image

```
# sudo docker load -i oms_4.2.2-ce.tar.gz
```

Single Node Deployment Demonstration

3. Get the Deployment Script

```
# sudo docker run --name oms-config-tool <OMS_IMAGE> bash && sudo docker cp oms-config-tool:/root/docker_remote_deploy.sh . && sudo docker rm -f oms-config-tool
```

4. Modify the Configuration File

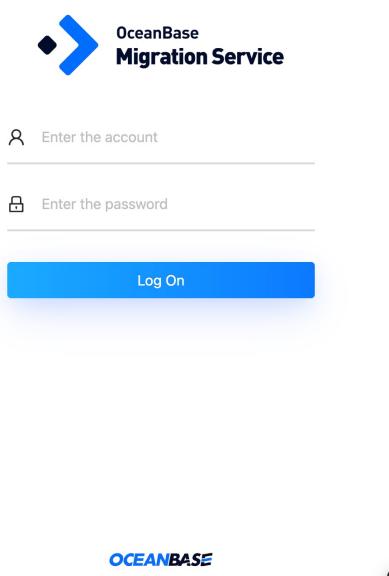
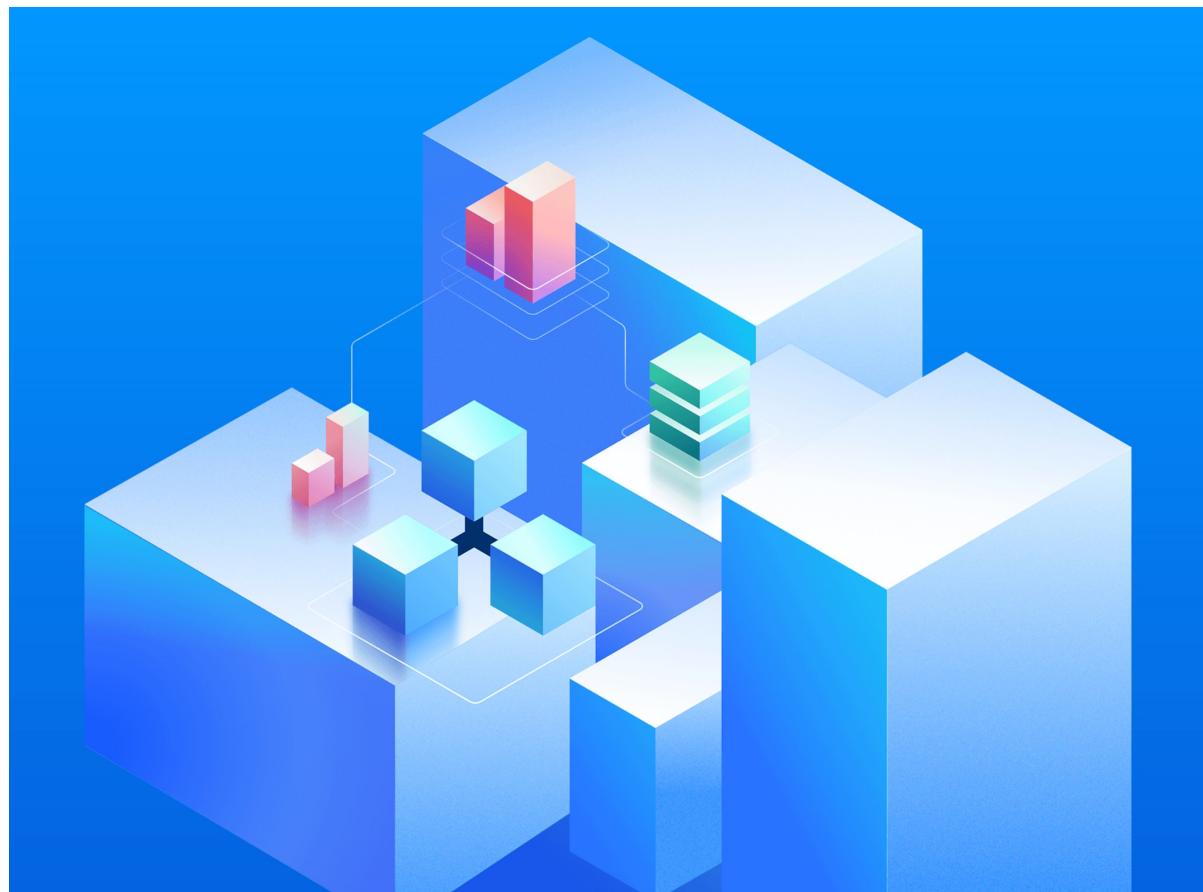
```
oms_meta_host: xxx.xxx.xxx.1
oms_meta_port: 2883
oms_meta_user: oms_meta_user
oms_meta_password: *****
drc_rm_db: oms_rm
drc_cm_db: oms_cm
drc_cm_heartbeat_db: oms_cm_heartbeat
cm_url: http://xxx.xxx.xxx.2:8088
cm_location: 100
cm_region: cn-anhui
cm_region_cn: 安徽
cm_is_default: true
cm_nodes:
  - xxx.xxx.xxx.2
tsdb_service: 'INFLUXDB'
tsdb_enabled: true
tsdb_url: 'xxx.xxx.xxx.4:8086'
tsdb_username: username
tsdb_password: *****
ghana_server_port:xxxxx
nginx_server_port:xxxxx
cm_server_port:xxxxx
supervisor_server_port:xxxxx
sshd_server_port:xxxxx
```

Single Node Deployment Demonstration

5. Execute Deployment

```
# sh docker_remote_deploy.sh -o <OMS_Container_Directory> -c <config.yaml> -i <IP> -d <OMS_IMAGE>
```

6. Login Page



7. Check Status

```
[root@localhost ~]# supervisorctl status
Nginx                  RUNNING  pid 1509, uptime 0:06:59
oms_console             RUNNING  pid 1514, uptime 0:06:49
oms_drc_cm               RUNNING  pid 1556, uptime 0:06:39
oms_drc_supervisor       RUNNING  pid 1958, uptime 0:06:16
sshd                   RUNNING  pid 1948, uptime 0:06:18
```

Configuration Details - Part I

Parameter	Description	Required
oms_meta_host	The IP address of the meta-repository, which can be MySQL or OceanBase Community Edition	Required
oms_meta_port	The port number of the meta-repository	Required
oms_meta_user	The user name for the meta-repository	Required
oms_meta_password	The user password for the meta-repository	Required
drc_rm_db	The database name for the Administration Console	Required
drc_cm_db	The name of the cluster management service's Metabase	Required
drc_cm_heartbeat_db	The name of the heartbeat library for the cluster management service	Required
cm_url	<p>The address of the OMS Community Edition cluster management service. For example: http://xxx.xxx.xxx.1:8088</p> <p>Note:</p> <p>When deploying a single node, it is usually configured as the current OMS Community Edition machine IP. Please Do Not Use http://127.0.0.1:8088</p> <p>The access address of the OMS Community Edition console is: the IP address of the host where OMS is deployed: 8089. For example, http (https)://xxx.xxx.xxx.1:8089. 8088 is the program port responsible for scheduling, and 8089 is the port of the web page address. Port 8088 must be used here.</p>	Required
cm_location	Region code, the value range is [0,127]. Each region is assigned a number, and users can choose on their own	Required

Configuration Details - Part II

Parameter	Description	Required
cm_region	Region, for example, cn-jiangsu.	Optional
cm_region_cn	The Chinese identifier of a region. For example, Jiangsu.	Optional
cm_nodes	A list of machine IP addresses for the OMS Community Edition cluster management service.	Required
cm_is_default	Indicates whether the OMS community edition cluster management service is used by default.	Optional, default value is true
tsdb_enabled	Indicates whether the indicator reporting function (monitoring capability) is enabled, including true or false.	Optional, default value is false
tsdb_service	Indicates the type of time series database, including INFLUXDB and CERESDB.	Optional, default value is CERESDB
tsdb_url	The machine address where InfluxDB is deployed. When tsdb_enabled is true, modify this parameter according to the actual environment.	Optional
tsdb_username	The user name of the time series database. When tsdb_enabled is true, modify this parameter according to the actual environment. After deploying the time series database, you need to manually create a time series database user and fill in the user name and password.	Optional
tsdb_password	The password of the time series database. When tsdb_enabled is true, please modify this parameter according to the actual environment.	Optional

Configuring Data Sources

OMS

New Data Source

Data Source Management

What is a data source?

Data sources are the connection information of the databases that are to be migrated or synchronized, including IP addresses, port numbers, and account information, etc. Before or during a data migration/synchronization project, you must create data sources for the source and destination databases respectively. You can directly use created data sources as the source and the destination.

How to create a data source

Data Sources

<input type="checkbox"/> Data Source Identifier	Data Source Type	Host IP address:port	Database username	Added At	Actions
<input type="checkbox"/> target-ob	OB_MySQL_CE	127.0.01:2881	root	2024-11-07 17:40:29	Test Connection Copy Edit Delete
<input type="checkbox"/> Source-MySQL	MySQL	172.16.23.241:3306 (Primary)	root (Primary)	2024-11-07 14:39:02	Test Connection Copy Edit Delete

2 in total 10 / page ▾

Remarks(optional)

After you specify the endpoint information, click Test Connection to test whether the endpoint can be connected.

Cancel Test Connection OK

Configuring Migration Tasks

OMS

- Overview
- Data Migration**
- Data Synchronization
- Data Source Manag...
- OPS & Monitoring
- System Managem... ^
- Permission Manag...
- Alert Management
- Associated OCP Clu...
- System Parameters

Data Migration / MySQL_OB_OMS_test

< MySQL_OB_OMS_test • Running

Stop View Objects View Component Monitoring More

Basic Information

ID: np_682s5z01hkxs Migration Type: 结构迁移+全量迁移+增量同步+全量校验 Alert Level: Low Protection

Created By: root Created At: 2024-11-08 17:34:58 Full Migration Concurrency: Stable

View Details: MySQL » OceanBase MySQL-C... View Details

Migration Details

Pre-check Schema Migration Full Data Migration Incremental Synchronization(3 seconds) Full Verification Forward Switchover

Status: Monitoring Description of Latency Calculation Current Timestamp: 2024-11-08 17:37:56 DDL/DML Statistics:

Latency: 3 seconds (13seconds ago 更新)

Synchronization Object Statistics

Incremental Synchronization Performance

Enter a search keyword.

Object Name	Source Database	Destination Database	Update	Delete	Insert	Truncate	Change Su
No data							m

OCEANBASE

Managing Migration Tasks

The screenshot shows the OceanBase OMS Data Migration interface. On the left, a sidebar menu includes 'OMS' (selected), 'Overview', 'Data Migration' (selected), 'Data Synchronization', 'Data Source Management', 'OPS & Monitoring' (with a dropdown arrow), and 'System Management'. The main area displays a migration task named 'MySQL_OB_OMS_test' which is 'Running'. The task details include:

- Basic Information:** ID: np_682s5z01hkxs, Migration Type: 结构迁移+全量迁移+增量同步+全量校验, Alert Level: Low Protection, Created By: root, Created At: 2024-11-08 17:34:58, Full Migration Concurrency: Stable.
- Migration Details:** A sequence of steps: Pre-check (green checkmark), Schema Migration (green checkmark), Full Data Migration (green checkmark), Incremental Synchronization(1 second) (blue circle with question mark), Full Verification (green checkmark), and Forward Switchover (grey circle). The 'Forward Switchover' step is highlighted with a blue border.
- Attention:** A tooltip message: "Forward switchover (the abstraction and standardization steps of the conventional system cutover process) will not switch over the business application connections, but is a task flow that needs to be executed in data migration tasks of OMS in order for application switchover." It lists three points:
 1. OMS automatically deletes the additional columns and unique indexes on which the migration depends, and adds CHECK constraints.
 2. You must manually activate the triggers and foreign keys in the destination, and add the database objects that OMS does not support, such as functions and stored procedures.
 3. If you have selected Reverse Incremental Migration, you must manually disable the triggers and foreign keys in the source.
- Status:** Not Started.
- Data Preview:** Shows a folder icon and the text 'No data'.

Managing Migration Tasks

OMS

- Overview
- Data Migration**
- Data Synchronization
- Data Source Management
- OPS & Monitoring
- System Management

OCEANBASE

View Component Monitoring

Component Monitoring Table (Source)

Component Type	Component ID
Store	172.16.24.108-z01hkxs_682s7008
Incr-Sync	172.16.24.108-z01hkxs_682s5z01hkxs0006
Full-Import	172.16.24.108-z01hkxs_682s5z01hkxs005
Full-Verification	172.16.24.108-z01hkxs_682s5z01hkxs0006

Component Monitoring Table (Destination)

Component Type	Component ID
Object Name	

View Logs

store.log **congo.log** **connector/connector.log**

Find Download Copy Refresh

```

113
114 [2024-11-08 17:35:30.650] ../../../../deps/thirdparty/easy/src/io/easy_connection.c:780(tid: 7f8c21ffb700)[22676] Accepted connect request from client(127.0.0.1:40014).
115 [2024-11-08 17:35:30.650] ../../../../deps/thirdparty/easy/src/io/easy_connection.c:735(tid: 7f8c21ffb700)[22676] Ralarm timer on connection, conn(127.0.0.1:44712_127.0.0.1:40014_22_0x7f8c100013c0 tp=0 t=1731058530650353-0 s=0 r=0 io=0/0 sq=0), detect_interval(20.000000), ack_timeout(60000), ralarm_count(1).
116 [2024-11-08 17:35:30.650] ../../../../deps/thirdparty/easy/src/io/easy_connection.c:904(tid: 7f8c21ffb700)[22676] Connection established on server side, conn(127.0.0.1:44712_127.0.0.1:40014_22_0x7f8c100013c0 tp=0 t=1731058530650353-0 s=0 r=0 io=0/0 sq=0), cb(0x7f8c3458c030).
117 [2024-11-08 17:35:30.650] ../../../../deps/thirdparty/easy/src/io/easy_connection.c:774(tid: 7f8c21ffb700)[22676] Failed to do accept.
118 Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual loading of the driver class is generally unnecessary.
119 listen port is 36666
120 build_tsc_timestamp: use_tsc=1 scale=419
121 build_tsc_timestamp: use_tsc=1 scale=419
122 2024-11-08 17:36:49 (ConnectionManager.cpp:245): Connection Manager handle socket: 23
123 2024-11-08 17:36:49 (ReadyConnection.cpp:462): FD: 23, Recv identification: 1, use id: No, use encryption: No
124 2024-11-08 17:36:49 (MessageIDUtil.h:13): Max message id: 2147352575
125 2024-11-08 17:36:49 (CompressConnection.cpp:55): Single connection with no messageid transfer
126 2024-11-08 17:36:49 (CompressConnection.cpp:88): Single Conn Srcaddr info: 172.16.24.108:17003, FD: 23
127 2024-11-08 17:36:49 (CompressConnection.cpp:93): Single Conn Dstaddr info: 172.16.24.108:41510, FD: 23
128 2024-11-08 17:36:49 (BatchsBuf.h:60): Init buf size: 1048576
129 2024-11-08 17:36:49 (BatchsBuf.h:60): Init buf size: 1048576
130 2024-11-08 17:36:49 (BatchsBuf.h:60): Init buf size: 1053696
131 2024-11-08 17:36:49 (ConnectionManager.cpp:364): New group generated, identification is: 1, sub fd: 23
132 2024-11-08T17:45:49.981+0800: 620.678: [GC (Allocation Failure) 2024-11-08T17:45:49.981+0800: 620.678: [ParNew: 838912K->28897K(943744K), 0.0542334 secs] 838912K->28897K(1992320K), 0.0543878 secs] [Times: user=0.07 sys=0.03, real=0.06 secs]
133 2024-11-08T20:13:43.006+0800: 9493.703: [GC (Allocation Failure) 2024-11-08T20:13:43.006+0800: 9493.703: [ParNew: 867809K->15702K(943744K), 0.0245715 secs] 867809K->15702K(1992320K), 0.0246711 secs] [Times: user=0.06 sys=0.00, real=0.02 secs]
```

Viewing the Task Log - Each Component Log

Console Logs

- The default directory for Console logs is /home/admin/logs/ghana/Ghana

CM Logs

- The default directory for CM component logs is /home/admin/logs/cm/log

Store Logs

- The default directory for the Store component log is /home/ds/store/store{storeport}/log
- You can also view the Store component log in the OMS Community Edition console

Full–Import/Incr–Sync Logs

- The default directory for Full–Import/Incr–Sync component logs is /home/ds/run/{component ID}/logs
- You can also view the Full–Import/Incr–Sync component logs in the OMS Community Edition console

Full–Verification Logs

- The directory of the Full–Verification component log is /home/ds/run/{name}/logs
- You can also view the Full–Verification component log in the OMS Community Edition console

Supervisor Logs

- Find the location of the Supervisor component log according to the logging.path item in the /home/ds/supervisor/config/drc.properties file

Incr-Sync/Full-Import Tuning

Query Metrics Information

Enter the directory of the corresponding task, /home/ds/run/{component ID}, and execute the following command

```
./connector_utils.sh metrics

2024-03-18 10:58:36 INFO
2024-03-18 10:58:35.812
SOURCE: [RPS:0.7, IOPS:0.0M, delay:2812ms]
SINK: [RPS:0.0, TPS:0.0, IOPS:0.0M, delay:1710730715812ms]
SINK_TIME: [execute_time:0.0ms/record, commit_time:0.0ms/batch]
SINK_SLOW_ROUTES:
SINK_THREAD: 0/4
DISPATCHER: wait record:0, ready batch:0, shardTime:nullms/record
forward_slot0 batchAccumulate: 0, recordAccumulate: 0
queue_slot1 batchAccumulate: 0, recordAccumulate: 0
heap:458M/1945M, noHeap:63M/488M, threadCount:21, cpu:0.263, sysCpu:5.022
ParNew(count:0, cost:0) ConcurrentMarkSweep(count:0, cost:0)
```

Incr-Sync/Full-Import Tuning

Diagnostics Incr-Sync/Full-Import

Go to the directory of the corresponding task: /home/ds/run/{component ID}, and execute the command

- `./connector_utils.sh diagnose -s 'YYYY-MM-DDTHH:mm:ss' -e 'YYYY-MM-DDTHH:mm:ss'`
- Without a time parameter, the default analysis is within 10 minutes. If `-e` is not specified, it means from the `-s` time to the current time

```
./connector_utils.sh diagnose

[Metrics]
TPS: [last:345,avg:277.28,p99:911.00]
RPS: [last:106,avg:257.08,p99:968.00]
IOPS: [last:2KB,avg:21.33KB]
EXECUTE_TIME: [last:34ms,avg:220.44ms,p99:783.00ms]
SINK_DELAY: [last:19ms,avg:260.31ms,p99:819.00ms]
SOURCE_DELAY: [
    source_subtopic2_source_delay: [last:702ms,avg:525.00ms,p99:986.00ms]
    source_subtopic1_source_delay: [last:14ms,avg:490.69ms,p99:973.00ms]
]
QUEUE_BATCH_ACCUMULATE: [
    frame_queue_slot_1.batchAccumulate: [last:420,avg:496.00,p99:975.00]
    frame_queue_slot_2.batchAccumulate: [last:310,avg:470.05,p99:975.00]
]
JVM-MEM: heap:34.28M/3641M, noHeap:19.38M/0M]
THREAD: [count:4, sink:14/16]
CPU: [last:17,avg:27.95,p99:62.00]
[Pref]
sink block: true
youngGc: true
[Suggest]
config[coordinator.shuffleMinBatchSize]:20 to 40
config[coordinator.shuffleMaxBatchSize]:40 to 80
jvm to: -Xmx4096m
```

Incr-Sync/Full-Import Tuning

GC time is too long

Here, the GC time is too long, which means that the Young GC is more than 300ms per second, and there is a Full GC every second.

Go to the directory of the corresponding task, /home/ds/run/{component ID}, and execute the command

```
/opt/alibaba/java/bin/jstat -gcutil `cat task.pid` 1s
```

- Increase JVM memory: adjust coordinator > connectorJvmParam to `-Xms12g -Xmx16g`.

This is just an example, you need to refer to the memory adjustment of the current machine. If the previous parameters exist, you can remove `-Xmn`.

- Sync && Full: Reduce the value of coordinator > bridgeQueueSize. The default value is 256, which can be reduced to 32
- Synchronize Kafka: Set the value of the sink > lingerMs parameter to 1
- Set the coordinator > throttleMemoryBound parameter (Unit: bytes) to limit the memory. It is recommended to set this parameter to 1/4 of the maximum memory.

For example, if the maximum heap memory is 16G, the value is $16 * 1024 * 1024 * 1/4 = 4294967296$

- If the dispatcherClassName configuration item in conf/coordinator.json or conf_new/coordinator.json is ShuffleRecordDispatcher, you can change the following configuration items of the coordinator :
 - maxRecordCapacity = 1000 controls the total number of dispatcher queues. The default is based on shuffleMinBatchSize *(shuffleBucketSize*1.5) = 3840
 - Set the value of the shuffleBucketSize parameter to 32. Reduce the number of batch buckets, the default is 128
 - Set the value of the shuffleFlushIntervalMs parameter to 10 to speed up the push to the Sink end
 - Added sink > workerNum parameter, the default value is 16, and the maximum value can be adjusted to 64

Task Release

OMS

Overview Data Migration Data Synchronization Data Source Management OPS & Monitoring System Management

Migrate Data

What is data migration?

The data migration feature helps you migrate data between homogeneous or heterogeneous data sources in business scenarios such as database upgrade, cross-instance data migration, database splitting, and database scaling. If you only need to synchronize data between two databases, go to [Data Synchronization](#) and perform operations as prompted.

[Quick start](#) | [Limits](#)

Migration Tasks

MySQL_OB_OMS_test Stopped Start Release More

ID: np_682s5z01hkxs Task Type: MySQL » OB_MySQL_CE [?](#)

Created At: 2024-11-08 17:34:58

Labels: [Tag](#)

Pre-check Schema Migration Full Data Migration Incremental Synchronization Full Verification Forward Switchover

1 in total 1 10 / page

Typical scenario introduction

- Batch data import (such as PoC, etc.)
- When the memory is insufficient, the flow may be limited, resulting in longer import time
- When the Memtable dump is not fast enough, Out Of Memory may be reported and the import may fail
- The source side has a huge table

Solution Value

- Bypass Memtable and write directly to storage, reducing unnecessary overhead and improving write performance
- Bypassing Memtable, tenant memory size is decoupled from data import volume
- Data writing efficiency is greatly improved

The screenshot shows the OceanBase Management System (OMS) interface under the 'Data Migration' tab. The main configuration area is divided into three sections: Schema Migration, Full Data Migration, and Incremental Synchronization.

- Schema Migration:** Shows character set mappings: latin1 to latin1 and latin1_swedish_ci to latin1_swedish_ci.
- Full Data Migration:** Configuration details include:
 - Concurrency Speed: Stable (selected), Normal, Fast, Custom.
 - Read Concurrency: 4.
 - Write Concurrency: 4.
 - JVM Memory: 2 ~ 4 GB.
 - Processing Strategy When Records Exist in Target Object: Stop Migration (selected).
 - Whether to Allow Post-indexing: Allow (selected).
 - Writing Method: SQL (selected, highlighted with a red box).
- Incremental Synchronization:** Configuration details include:
 - Concurrency Speed: Stable (selected), Normal, Fast, Custom.
 - Read Concurrency: 4.
 - Write Concurrency: 4.
 - JVM Memory: 2 ~ 4 GB.
 - Incremental Record Retention Time: 120 Hours.

Data Synchronization

- OceanBase Migration Service (OMS) Community Edition supports real-time data synchronization between OceanBase Community Edition and Kafka and RocketMQ. OMS Community Edition can be applied to business scenarios such as real-time data warehouse construction, data query, and report diversion
- The data synchronization project from the source to the target mainly supports incremental data synchronization scenarios, writing DML changes that occur from the source to the target in real time. The minimum granularity of synchronization supported by OMS Community Edition is table level and the maximum is tenant level
- Data format description, supporting serialization to control the message format of data synchronization to the target end
 - Default JSON message format
 - Canal JSON message format
 - Dataworks JSON message format
 - SharePlex JSON message format
 - DefaultExtendColumnType JSON message format
 - Debezium JSON message format
 - DebeziumFlatten JSON message format
 - Maxwell JSON message format

Agenda



- **Lesson 4.1**
 - Introduction to OceanBase and MySQL Compatibility
 - Introduction to OMS Migration and Synchronization
 - **OBLogProxy Introduction**
- **Lesson 4.2**
 - **OBDUMPER/OBLOADER Introduction**
 - Introduction to SQL Command Export
 - Introduction to Other Ecological Components
 - Comparison of Components

OBLogProxy Introduction

OBLogProxy is OceanBase's incremental log proxy service. It can establish a connection with OceanBase and read incremental logs, providing change data capture (CDC) capabilities for downstream services

- **OBLogProxy Features**

OBLogProxy has two modes: Binlog mode and CDC mode

- **Binlog Mode**

Binlog mode enables OceanBase to be compatible with MySQL binlog. It supports real-time synchronization with existing MySQL binlog incremental parsing tools, allowing these tools to seamlessly transition to OceanBase

- **CDC Mode**

CDC mode is used to solve data synchronization. In CDC mode, OBLogProxy can subscribe to data changes in the OceanBase database and synchronize these data changes to downstream services in real time, achieving real-time or quasi-real-time replication and synchronization of data.

OBLogProxy Installation

Method 1: RPM Installation

1、Download and install

You can download the OBLogProxy installation package from the OceanBase official website (<https://www.oceanbase.com/softwarecenter>) or GitHub repository (<https://github.com/oceanbase/oceanbase>)

2、Upload to the server and execute the installation command

```
rpm -i oblogproxy-{version}.{arch}.rpm
```

The default location for project installation is /usr/local/oblogproxy

Method 2: Source Code Construction

1、Prerequisites

Install CMake, V3.20 and above is required

2、Install Dependencies

```
yum install -y git wget rpm rpm-build gcc gcc-c++ make glibc-devel glibc-headers libstdc++-static binutils zlib zlib-devel bison flex java-11-openjdk
```

3、Source code compilation

```
git clone https://github.com/oceanbase/oblogproxy.git  
cd oblogproxy  
mkdir build  
cd build  
cmake ..  
cmake --build . -j 8
```

Configuration Files - Part I

FROM INTRODUCTION TO PRACTICE

The default configuration file of OBLogProxy is placed in conf/conf.json

Fields	Default Value	Description
service_port	2983	Service listening port
encode_threadpool_size	8	Encoding thread pool size
encode_queue_size	20000	Encoding queue size
max_packet_bytes	67108864	Maximum packet size
record_queue_size	20000	Data send queue size
read_timeout_us	2000000	Batch read timeout period (Unit: μ s)
read_fail_interval_us	1000000	Batch read failure cycle (Unit: μ s)
read_wait_num	20000	Batch read waiting number
send_timeout_us	2000000	Batch sending timeout period (Unit: μ s)
send_fail_interval_us	1000000	Batch sending failure cycle (Unit: μ s)
check_quota_enable	false	Enable quota checking
check_clog_enable	true	Enable clog inspection
command_timeout_s	10	Command execution timeout (Unit: μ s)
log_quota_size_mb	5120	Log file total size threshold (Unit: MB)
log_quota_day	7	Log file retention time threshold (Unit: d)
log_gc_interval_s	43200	Log file cleanup cycle (Unit: s)
log_level	2	Log level, default is info level
log_flush_strategy	1	Log refresh policy
log_flush_level	2	Log refresh level
log_flush_period_s	1	Log refresh cycle (Unit: s)
log_max_file_size_mb	1024	Maximum size of a single log file (Unit: MB)

Fields	Default Value	Description
log_retention_h	360	Log file retention period (Unit: h)
oblogreader_path_retain_hour	168	oblogreader subprocess directory retention time (Unit: h)
oblogreader_lease_s	300	oblogreader subprocess start detection time (Unit: s)
oblogreader_path	/usr/local/oblogproxy/run	oblogreader subprocess directory root path (absolute path)
bin_path	/usr/local/oblogproxy/bin	Project executable program path (absolute path)
oblogreader_obcdc_ce_path_template	/usr/local/oblogproxy/obcdc/obcdc-ce-%d.x-access/libobcdc.so	oblogreader depends on obcdc library path template (absolute path)
allow_all_tenant	true	Allow subscriptions for all tenants
auth_user	true	Enable user authentication
auth_use_rs	false	Enable user authentication through root server
auth_allow_sys_user	true	Allow subscription to system tenants
ob_sys_username	""	System tenant username ciphertext (must be configured by yourself)
ob_sys_password	""	System tenant password ciphertext (must be configured by yourself)
counter_interval_s	2	Counter period (Unit: s)
metric_enable	true	Enable indicator collection (resource consumption, CPU, memory, network)

Configuration Files - Part II

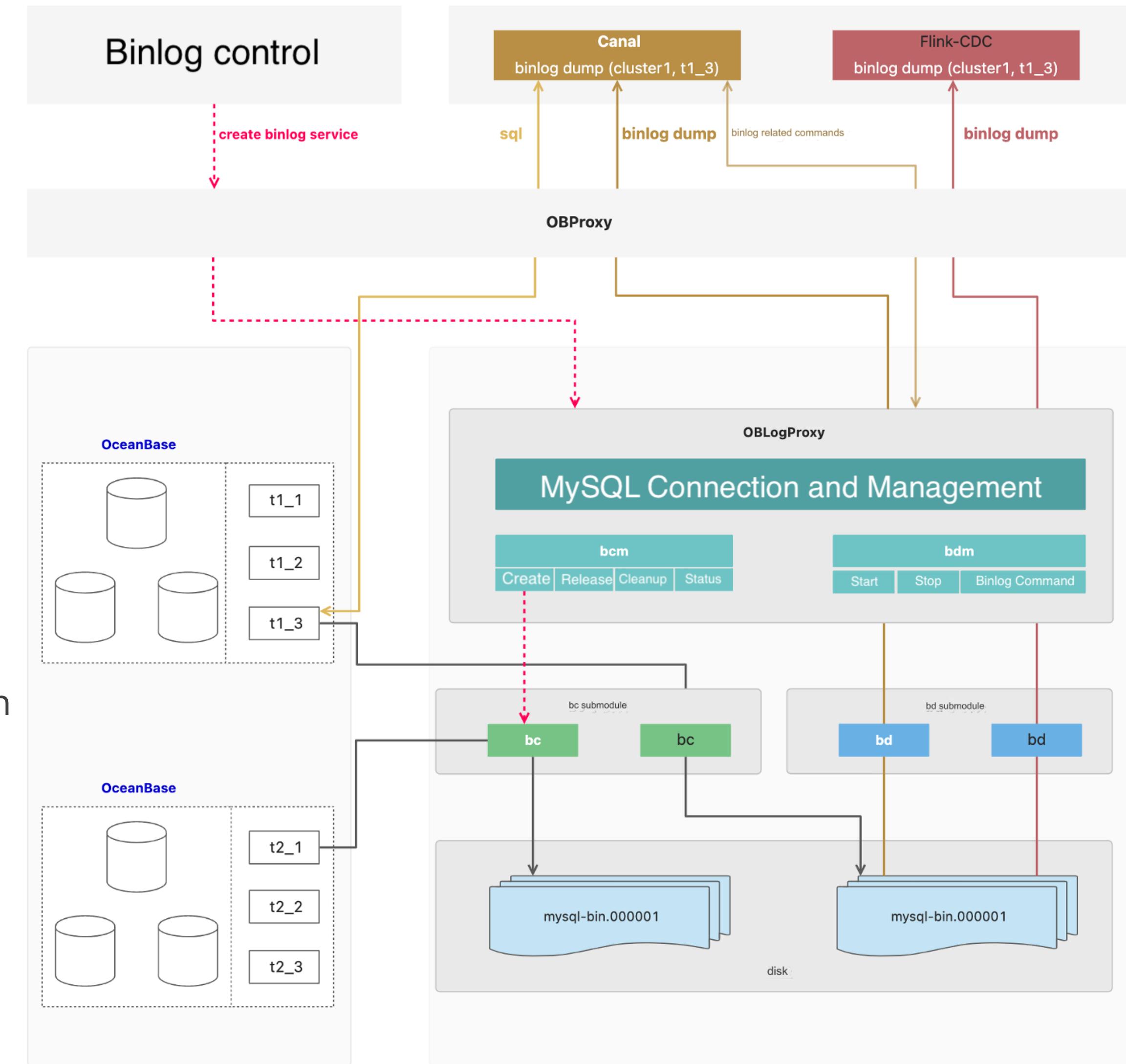
FROM INTRODUCTION TO PRACTICE

Fields	Default Value	Description	Fields	Default Value	Description
metric_interval_s	10	Index collection cycle	binlog_ignore_unsupported_event	true	Ignore events that do not support conversion
debug	false	Print debug information	binlog_max_event_buffer_bytes	67108864	Binlog event conversion buffer size
verbose	false	Print Details	binlog_mode	false	Enable binlog mode
verbose_packet	false	Print packet details	table_whitelist	""	Binlog conversion whitelist
verbose_record_read	false	Print record read details	binlog_nof_work_threads	16	Thread pool size for non-BC scheduling requests
readonly	false	Read-only mode	binlog_bc_work_threads	2	BC Scheduler Thread Pool Size
count_record	false	Number of records	binlog_max_file_size_bytes	524288000	The size of a single binlog file
channel_type	plain	Link Type	binlog_convert_timeout_us	10000	Binlog conversion timeout (Unit: μ s)
tls_ca_cert_file	""	CA certificate file path (absolute path)	binlog_checksum	true	Enable binlog checksum
tls_cert_file	""	Server-side signature certificate path (absolute path)	binlog_heartbeat_interval_us	1000000	Binlog heartbeat event sending cycle (Unit: μ s)
tls_key_file	""	Private key path on the server side (absolute path)	binlog_gtid_display	true	Display GTID information
tls_verify_peer	true	Enable OBLogClient authentication	binlog_ddl_convert	true	Enable DDL conversion (OceanBase DDL will be converted to MySQL DDL after enabling it)
liboblog_tls	false	Enable TLS for communication with ObServer	binlog_memory_limit	3G	Binlog memory limit, default is 3G
liboblog_tls_cert_path	""	ObServer related certificate file path (absolute path)	binlog_working_mode	storage	Binlog working mode, default is storage
binlog_log_bin_basename	/usr/local/oblogproxy/run	Binlog service directory root path (absolute path)	binlog_recover_backup	true	Enable binlog backup and recovery
binlog_obcdc_ce_path_template	/usr/local/oblogproxy/obcdc/obcdc-ce-%d.x-access/libobcdc.so	binlog depends on obcdc library path template (absolute path)	wait_rotate_ready_max_try	1000	Maximum number of retries while waiting for rotation to be ready

Introduction to Binlog Mode

Binlog mode was created to be compatible with MySQL binlog, and supports existing MySQL binlog ecosystem tools to synchronize with OceanBase. Existing MySQL binlog ecosystem tools can be smoothly switched to the OceanBase database. Binlog mode is consistent with MySQL 5.7 and has the same functions:

- Supports Position mode subscription. In Position mode, the subscriber is based on the specified binlog file name and offset, also known as binlog position. In this mode, subscription is based on physical position.
- Supports GTID mode subscription. GTID (Global Transaction ID) is a mechanism for uniquely identifying and tracking transactions in a distributed environment. In GTID mode, each transaction has a unique global transaction ID that is not affected by changes and offsets in the binlog file. In this mode, subscription is based on logical position.



Comparison between Binlog mode and MySQL binlog compatibility

MySQL binlog related commands

Command	Support
show master status	support
show binary logs	support
show binlog events	support
purge binary logs	support
reset master	Not supported
show binlog server	support

MySQL binlog replication protocol

Protocol	Support
com_binlog_dump	support
com_binlog_dump_gtid	support
com_register_slave	support

MySQL-related system variables

System Variables	OceanBase Support Notes
binlog_format	row
binlog_row_image	full
binlog_checksum	crc32
binlog_rows_query_log_events	off
log_bin	on
server_id	An integer greater than 0. The value remains unchanged after the tenant is created.
server_uuid	This value should remain unchanged after the tenant is created.

Note: OceanBase's DDL syntax differs somewhat from MySQL, as OceanBase includes extended syntax that may not be parsed by MySQL-compatible tools.

To address this, OceanBase offers compatibility support. We recommend setting init_sql in OBProxy to enable _show_ddl_in_compat_mode. Once enabled, OceanBase's show 'create table output' will be fully compatible with MySQL syntax.

Binlog Mode Usage Limitations

Functional limitations

- Binlog mode does not support the extended semantics for `enum` and `set` types implemented by OceanBase. For example, the number of set definitions supports more than 64, supports `duplication`, and `enum` supports inserting undefined data.
- Binlog mode does not support `varchar(65536)` definition.
- Binlog mode does not support `gis` type.

Binlog Mode Configuration

Prerequisites

Before using the Binlog mode function, you need to confirm the version information of OceanBase and OBProxy

Configure OBProxy

1. Configure the OBLogProxy service address for OBProxy, and use the sys or proxysys tenant account to connect to OBProxy

```
mysql -h'ip' -P 2883 -u'user'@sys#'cluster' -D oceanbase -A -p'password'
```

2. Configure the binlog service address binlog_service_ip, the address format is ip:port

```
alter proxyconfig set binlog_service_ip='ip:port';
```

3. Enable binlog service enable_binlog_service

```
alter proxyconfig set enable_binlog_service='True';
```

4. Set init_sql, turn on show_ddl_in_compat_mode

```
alter proxyconfig set init_sql='set _show_ddl_in_compat_mode = 1;';
```

5. Verify that the configuration is correct

```
show proxyconfig like 'config_name';
```

Binlog Mode Configuration

Configure OBLogProxy

1. The default directory of OBLogProxy project is /usr/local/oblogproxy, and the default configuration file is placed in conf/conf.json
2. The following configuration needs to be modified in conf/conf.json

```
"binlog_mode": true  
"ob_sys_username": ""  
"ob_sys_password": ""
```

- (Optional) OBLogProxy requires the configuration of the user's username and password. The user must be a user of the **sys tenant** of OceanBase to connect. **Note** that the username here should not contain the cluster name or tenant name, and must have read permissions for the OceanBase database under the sys tenant.
- First, get the encrypted username and password

```
./bin/logproxy -x username  
./bin/logproxy -x password
```
- Then save the results to the ob_sys_username and ob_sys_password configuration items in the conf/conf.json file respectively

Run OBLogProxy

3. Start the OBLogProxy service

```
cd /usr/local/oblogproxy && ./run start
```

4. Create a binlog service and connect to OBLogProxy, where 2983 is the default port of OBLogProxy

```
mysql -h'ip' -P 2983
```

- After OBLogProxy is started, you need to send a SQL command to OBLogProxy to create a binlog service.

```
CREATE BINLOG [IF NOT EXISTS] FOR TENANT `cluster`.`tenant` [TO USER `user` PASSWORD `pwd`] [FROM `timestamp`]
WITH CLUSTER URL `cluster_url`;
```

- The user and pwd configuration items here are optional. If the ob_sys_username and ob_sys_password configuration items are not configured in the conf/conf.json file, you need to specify user and pwd
- cluster_url is the current OceanBase config_url, which can be queried using the following SQL statement

```
show parameters like '%config_url%';
```
- The timestamp is the start time of creating the binlog service, in microseconds. If the timestamp is not specified, the current time will be used by default.

Other Binlog Operations

To execute the following SQL, you need to connect directly to OBLogProxy. You can connect to OBLogProxy using the following command:

```
mysql -h'ip' -P 2983
```

- Release the binlog service

```
DROP BINLOG [IF EXISTS] FOR TENANT `cluster`.`tenant`;
```

- Clean up binlog logs

- Clean up by time

```
PURGE BINARY LOGS BEFORE '2023-03-01 00:00:00' [FOR TENANT `cluster`.`tenant`];
```

- Clean up by file

```
PURGE BINARY LOGS TO 'mysql-bin.000001' [FOR TENANT `cluster`.`tenant`];
```

- Get binlog service status

- Get all binlog service status

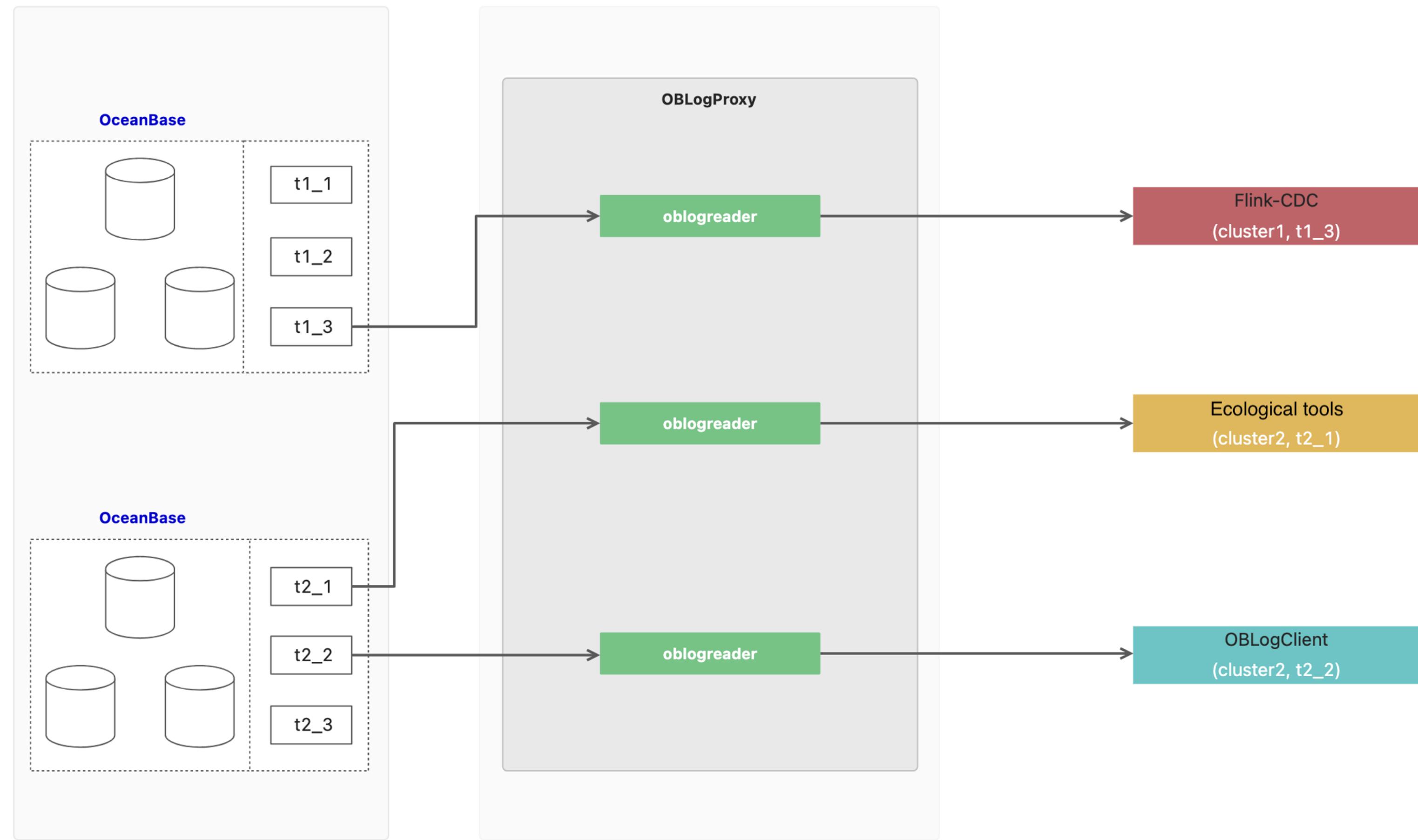
```
SHOW BINLOG STATUS;
```

- Get the binlog service status of a specified tenant

```
SHOW BINLOG STATUS [FOR TENANT `cluster`.`tenant`];
```

Introduction to CDC Mode

CDC mode is used for real-time data synchronization. In CDC mode, OBLogProxy can subscribe to data changes in the OceanBase database (needs to cooperate with OBLogClient to complete data subscription) and synchronize these data changes to downstream services in real-time, realizing real-time or quasi-real-time replication and synchronization of data.



Using CDC Mode

Prerequisites

Before using the Binlog mode function, you need to confirm the version information of OceanBase and OBProxy

Configure OBLogProxy

The default directory of OBLogProxy project is /usr/local/oblogproxy, and the default configuration file is placed in conf/conf.json

1. The following configuration needs to be modified in conf/conf.json

```
"ob_sys_username": ""  
"ob_sys_password": ""
```

- (Optional) OBLogProxy requires the configuration of the user's username and password. The user must be a user of the sys tenant of OceanBase to connect. Note that the username here should not contain the cluster name or tenant name, and must have read permissions for the OceanBase database under the sys tenant.
- First, get the encrypted username and password

```
./bin/logproxy -x username  
./bin/logproxy -x password
```
- Then save the results to the ob_sys_username and ob_sys_password configuration items in the conf/conf.json file respectively

Run OBLogProxy

2. Start the OBLogProxy service

```
cd /usr/local/oblogproxy && ./run start
```

3. Before using OBLogClient subscription, you need to make sure that the Maven dependency is installed.

```
<dependency>
  <groupId>com.oceanbase</groupId>
  <artifactId>oblogclient-logproxy</artifactId>
  <version>x.y.z</version>
</dependency>
```

OBLogClient Reference Code

```
ObReaderConfig config = new ObReaderConfig();
// Set the OceanBase root server address list, the format is (can support multiple, separated by ':') ip1:rpc_port1:sql_port1;ip2:rpc_port2:sql_port2
config.setRsList("xxx.xxx.xxx.1:2882:2881;xxx.xxx.xxx.2:2882:2881");
// Set username and password (non-system tenant)
config.setUsername("r***");
config.setPassword("****");
// Set the startup location (UNIX timestamp, unit s), 0 means starting from the current time.
config.setStartTimestamp(0L);
// Set the subscription table whitelist, wild: tenant.db.table, '*' means wildcard.
config.setTableWhiteList("sys.*.*");

// Specify the OBLogProxy service address and create an instance.
LogProxyClient client = new LogProxyClient("xxx.xxx.xxx.1", 2983, config);
// Add RecordListener
client.addListener(new RecordListener() {
    @Override
    public void notify(LogMessage message){
        // Process the message
    }

    @Override
    public void onException(LogProxyClientException e) {
        // Handle errors
        if (e.needStop()) {
            // Unrecoverable exception, need to stop Client
            client.stop();
        }
    }
});
// start up
client.start();
client.join();
```

Thank You!



OceanBase Official website:
<https://oceanbase.github.io/>



GitHub Discussions:
<https://github.com/oceanbase/oceanbase/discussions>



FROM INTRODUCTION TO PRACTICE

Lesson 4.2: Oceanbase Database Migration and Synchronization

Peng Wang
OceanBase Global Developer Evangelist



Agenda



- **Lesson 4.1**
 - Introduction to OceanBase and MySQL Compatibility
 - Introduction to OMS Migration and Synchronization
 - OBLogProxy Introduction
- **Lesson 4.2**
 - **OBDUMPER/OBLOADER Introduction**
 - Introduction to SQL Command Export
 - Introduction to Other Ecological Components
 - Comparison of Components

OBLOADER Introduction

What is OBLOADER

- Client tool developed in Java language
- Applicable only to OceanBase database
- Import database object definition files and table data files in storage media into OceanBase database
- Compatible with CSV format files exported by client tools such as mysqldump and Mydumper
- OBLOADER specifically optimizes data import performance, has built-in multiple data preprocessing functions, automatic fault tolerance to ensure data import stability, and provides relatively rich monitoring information, so that users can observe the performance and progress of data file import in real-time

Product Features

- Supports importing database object definitions and table data from local disks, Apache Hadoop, Aliyun OSS or Amazon S3
- Supports importing SQL-Format files exported by mysqldump
- Supports importing data files in standard CSV, Insert SQL, ORC, Parquet and other formats
- Supports rich data cleaning functions
- Supports multiple error handling strategies
- Supports logical segmentation of original data files before importing to give full play to CPU multi-core performance
- Supports encryption of sensitive parameters specified in the command line. Including: database account password, cloud storage account key

OBDUMPER Introduction

What is OBDUMPER

- Client tool developed in Java language
- Only applicable to OceanBase database
- This tool can be used to export objects and table data defined in the OceanBase database to storage media in a specified file format
- Users who want to use OBDUMPER for logical backup can directly integrate this tool into the database operation and maintenance system.
(Note: Incremental backup is not supported)
- Compared with client-side export tools such as mysqldump, OBDUMPER has the following significant advantages:
 - Fast data export capability, designed with multiple data query strategies, greatly improving export performance
 - Rich data exchange capabilities, supporting exporting table data in multiple formats to multiple storage media
 - Powerful data processing capabilities, compressing, encrypting, desensitizing, preprocessing, etc. data before exporting

Product Features

- Supports exporting database object definitions and table data to local disk, Aliyun OSS and Amazon S3.
- Supports exporting table data to files in formats such as CSV, Insert SQL, ORC, and Parquet.
- Supports specifying partition names and only exporting data in the specified table partition.
- Supports specifying global filtering conditions and only exporting data that meets the conditions.
- Supports configuring data preprocessing rules and performing preprocessing such as conversion and desensitization on data before export.
- Supports specifying SCN or TIMESTAMP and only exporting historical snapshot data at valid transaction points or time points.
- Supports exporting data from OceanBase backup copies. (Note: different from backup clusters)
- Supports specifying custom query statements and only exporting the result set of the query statement.
- Supports exporting globally consistent data through the latest snapshot version without locking the table.
- Supports encryption of sensitive parameters specified in the command line. Including: database account password, cloud storage account key

Precautions - Part I

- Please refer to the [RFC 4180](#) specification for the standard CSV format. It is recommended that the [RFC 4180](#) specification be strictly followed when importing
- When importing and exporting large amounts of data, modify the memory parameters of JVM in the running script to improve performance
- In this case, the object names, data file names, and rule file names specified in the command-line parameters must match exactly. Since MySQL defaults to lowercase, if you need to distinguish between uppercase and lowercase, enclose the table name in brackets ([]). For example: --table '[test]' means the test table, and the file name format is test.group.sequence.suffix; --table '[TEST]' means the TEST table, and the file name format is TEST.group.sequence.suffix. Among them, group represents the subtask number (determined by the program according to the subtask segmentation strategy), sequence represents the file roll number (when the file size exceeds --block-size, it will roll), and suffix represents the file extension
- When importing, OBLOADER can recognize the file name format of "table name" + file extension. You can use the `--file-regular-expression` command line option to implement custom file search rules through regular expressions
- When database objects have dependencies (e.g., foreign key relationships between tables or triggers depending on sequences), it's best to import them in the correct dependency order. Using options like `--all` or `--table '*'` does not guarantee the precise import order

Precautions - Part II

- Tables without primary keys do not support breakpoint resuming yet
- Before using OBLOADER in OceanBase 3.2.4 and later versions, ensure the open_cursors system configuration is set to a higher value to avoid import failures. After completing the data import, reset it to its original value. For example:

```
ALTER SYSTEM SET open_cursors = 65535;
```

- When importing DDL, note the difference between the `--mix` and `--ddl` options:
 - The `--ddl` option is designed for files containing a single DDL statement.
 - The `--mix` option does not have this restriction.
- OBLOADER supports the following file formats:
 - DDL file: The content in the file only contains DDL statements, not table data.
 - CSV file: The standard CSV format that complies with the [RFC 4180](#) specification.
 - SQL file: The content in the file only contains INSERT SQL statements, and the data does not wrap.
 - ORC file: The format that complies with the standard Apache ORC, and zstd compression is used by default.
 - Parquet file: The format that complies with the standard Apache Parquet, and zstd compression is used by default.
 - MIX file: The content in the file contains DDL, DML, and any other statements that comply with the SQL standard.
 - POS file: The format defined by a fixed byte length, and fixed character length is not currently supported.
 - CUT file: Data columns are separated by single or multiple characters, with no delimiter. Different from the standard CSV format
- For other information, please refer to the official website documentation

Preparation Before Use

- Operating Environment

Environment	Require
System version	Supports Linux/macOS/Windows 7 and later versions
Java Environment	Please install Oracle JDK 1.8.0_3xx and configure the JAVA_HOME environment variable
Character Set	UTF-8 file encoding is recommended
JVM parameters	Please edit the bin/obloader and bin/obdumper scripts to modify the JVM memory parameters to avoid JVM running out of memory

- OBLOADER Runtime Permissions

- When using OBLOADER to connect to the OceanBase database to import data, the account connected to the database needs to have the execution permissions of commands such as CREATE, SELECT, INSERT, and UPDATE.
- Before importing data, the user needs to have the query permission of the OceanBase database.

- OBDUMPER Runtime Permissions

- When using OBDUMPER to connect to the OceanBase database to export data, the account connected to the database needs to have the execution permissions of commands such as CREATE and SELECT.
- Before exporting data, the user needs to have the query permission of the oceanbase database.

Download And Run The Tool

1. Please download the derivative tool package at <https://www.oceanbase.com/softwarecenter>
2. After unzipping the package, go to the directory where the running script is located.

```
# Windows  
cd {ob-loader-dumper}/bin/windows  
# Linux or macOS  
cd {ob-loader-dumper}/bin
```

3. Execute the following statement to view the help of command line options

- Run OBLOADER

```
# Windows  
call obloader.bat --help  
# Linux or macOS  
../obloader --help
```

- Run OBDUMPER

```
# Windows  
call obloader.bat --help  
# Linux or macOS  
../obloader --help
```

Import and Export Examples

Option Classification

Command line options are divided into basic options and advanced options

- Basic options: Common options, including connection options (connection database mode), functional options (file format, database object type, storage path) , and other options
- Advanced options: including functional options (timestamp format, table/column blacklist and whitelist filtering, error handling) and performance options

OBLOADER Import Example

```
$ ./obloader -h xx.x.x.x -P 2883 -u test@mysql#cluster_a -p ***** -D USERA --csv --table '*' -f /output
```

–h, –P, –u, –p, –D: connection options; --csv: the file format option; --table: the database object type option; –f: the storage path.

OBDUMPER Export Example

```
$ ./obdumper -h xx.x.x.x -P 2883 -u test@mysql#cluster_a -p ***** -D USERA --csv --table '*' -f /output
```

–h, –P, –u, –p, –D: connection options; --csv: the file format option; --table: the database object type option; –f: the storage path.

Bypass Import

OBLOADER 4.2.6 and later versions support bypass import of data.

Bypass import key parameters	Description
--rpc-port= <i>rpc_port_num</i>	Used to connect to the OBServer RPC port. This option is used with --direct, --parallel to connect to the OBServer RPC port to import data in bypass import mode.
--direct	Used to specify bypass import mode. This option is used with --rpc-port, --parallel
--parallel= <i>parallel_num</i>	Used for parallelism when loading data during bypass import. This option is used with --rpc-port, --direct

Precautions

- OBLOADER bypass import mode does not currently support binary data types
- OBLOADER bypass import mode supports connecting OBServer and ODP. Corresponding version requirements:
 - When connecting to OBServer: OBServer version must be 4.2.0 or later
 - When connecting to ODP: The ODP version must be 4.1.3 or later, and the OBServer version must be 4.2.1 or later

Import Performance Tuning

Command Line Option Tuning

- For wide tables or long column values, reduce the parameter value of the --batch option.
- Indexes affect the performance of data import. Except for primary keys and unique keys, ordinary indexes are delayed until the data import is completed.
- When the machine load and network are low, adjust the parameter value of the --thread option as appropriate

Virtual Machine Parameter Tuning

Modify the virtual machine parameters in the import script to 60% of available physical memory. Default value: **-Xms4G -Xmx4G**

```
vim bin/obloader
JAVA_OPTS="$JAVA_OPTS -server -Xms4G -Xmx4G -XX:MetaspaceSize=128M -XX:MaxMetaspaceSize=128M -Xss352K"
```

Database Kernel Tuning

The performance of importing data will be severely affected by the tenant's incremental memory write speed

- When the incremental memory is insufficient, the database will trigger a compaction or dump. Merges consume performance, so try not to trigger them
- When the incremental memory usage reaches the tenant speed limit threshold, the import performance will also decrease
- When the incremental memory usage is full, data import may fail easily. It is recommended that the tenant speed limit threshold be higher than 90.

The setting of dump-related parameters is related to the size of tenant memory and write speed, and needs to be tuned according to actual conditions. The kernel-related tuning parameters are as follows.

Import Performance Tuning

Parameter	Default Value	Description
set global ob_sql_work_area_percentage=20;	5	The percentage of memory used during SQL execution. Value range: [0, 100].
set global max_allowed_packet=1073741824;	130023424	The maximum size of a network packet that the server can receive.
alter system set freeze_trigger_percentage=30;	3.x Default 70 4.x Default 20	Used to set the tenant memory usage threshold that triggers global freeze. major_freeze_trigger_percent = major_freeze trigger threshold / memstore capacity, where memstore capacity is calculated by the configuration item memstore_lmt_percent. Calculation formula: memstore_lmt_percent = memstore_limit/min_memory. Value range: [1, 99].
alter system set minor_freeze_times=500;	4.x Deprecated	Used to set the number of small merges that trigger a global merge. When the value is 0, it means turning off small merges. When the memory exceeds the preset limit, a minor freeze or major freeze will be triggered. This parameter refers to the number of times a minor freeze is triggered between two consecutive major freezes. 0 means prohibiting automatic triggering of minor freeze. Value range: [0, 65536].
alter system set minor_compact_trigger=16;	2	It is used to control the threshold for triggering layer dumps to push down to the next layer. When the total number of Mini SSTables in the layer reaches the set threshold, all SSTables will be pushed down to the next layer to form a new Minor SSTable
alter system set merge_thread_count=32;	4.x Deprecated	Used to set the number of threads for daily merge work. When the value of this configuration item is 0, the number of merged work processes is calculated as min{10,cpu_cnt*0.3}, where cpu_cnt is the number of system CPUs. After modifying the dynamic parameters, there is no need to restart and it will take effect immediately. Value range: [0, 256].
alter system set minor_merge_concurrency=48;	4.x Deprecated	Used to set the number of concurrent threads for small merges. Value range: [0, 64].
alter system set writing_throttling_trigger_percentage=100;	60	Set the server memory current limit threshold (it is recommended to disable server current limit)

Export Performance Tuning

Command Line Option Tuning

Virtual Machine Parameters	Default Value	Description
--thread	CPU * 2	The concurrent number of export threads is adjusted according to the utilization of database system resources
--page-size	1000000	Specify the size of the task slice and adjust it according to the utilization of database system resources

Virtual Machine Parameter Tuning

- Modify the virtual machine parameters in the export script to 60% of available physical memory.

Default value: `-Xms4G -Xmx4G`

```
vim bin/obdumper
JAVA_OPTS="$JAVA_OPTS -server -Xms4G -Xmx4G -XX:MetaspaceSize=128M -XX:MaxMetaspaceSize=128M -Xss352K"
```

Database Kernel Tuning

- When exporting consistent data, it is recommended to manually trigger a merge before exporting the data, and then re-export the data after the merge is successful

Data Processing

Defining Control Files

```
lang=java
(
    Column_name Byte_offset (optional) "Preprocessing_function" (optional) Mapping_definition (optional),
    Column_name Byte_offset (optional) "Preprocessing_function" (optional) Mapping_definition (optional),
    Column_name Byte_offset (optional) "Preprocessing_function" (optional) Mapping_definition (optional),
    Column_name Byte_offset (optional) "Preprocessing_function" (optional) Mapping_definition (optional)
);
```

- Column name:** The name of the field in the database table structure. OBLOADER is not case-sensitive. If you need to distinguish between upper and lower case columns, put the column name in square brackets ([]) or backticks (``). For example: [c1] means column c1, and [C1] means column C1
- Byte offset position:** supports two declaration methods, absolute offset, and relative offset. Only **--pos** format data supports defining byte offset position
 - Absolute offset:** position(start, end), start and end represent the starting position (bytes) and ending position (bytes) of the field respectively. If you need to specify the column length to ignore a certain section of bytes when importing into the database table, you can use the special keyword **_FILLER** instead of the actual column name and mark the part of the data. This operation will create an anonymous column that will be parsed by OBLOADER but not associated with any existing columns in the table. This keyword is particularly suitable for ignoring irrelevant data in the file, such as padding bytes at the end of a row. Example: _filler position(5:10), which means skipping bytes 5 to 10
 - Relative offset:** position(length), where length indicates the length of the field (bytes)
- Preprocessing function:** Configure a preprocessing function for the specified column in the control file to preprocess the imported data
- Mapping definition:** column position of preprocessed data in the imported data file

Note: • The naming convention of the control file is **<table name>.ctrl**

- One control file corresponds to one table. When preprocessing data from multiple tables, you need to create multiple control files under the **ctl-path** path, and the control file name must be the same as the corresponding table name to be imported
- Use the **--ctl-path** option to specify the absolute path to the control file

Data Processing

Preprocessing Function: When defining a control file, users can configure a corresponding preprocessing function for each column

Precautions

- The formal parameters of any preprocessing function can only reference the value of the current column, and cross-column references are not supported yet. For example: c14 "concat(c15, '_suffix')" is not supported, only c15 "concat(c15, '_suffix')" is supported
- The preprocessing functions LPADB() and RPADB() are suitable for characters such as letters, numbers, and Chinese characters. Truncation may occur when processing Emoji
- The preprocessing function REPLACE() behaves differently in the derivative tool than in Oracle and MySQL. Its syntax is closer to Oracle and its implementation is closer to MySQL
- The preprocessing function NVL() refers to the corresponding built-in function in MySQL, which distinguishes between empty characters and NULL in its implementation
- Date and time-related preprocessing functions (SYSTIMESTAMP, TMSFMT() and TO_TIMESTAMP()) are only accurate to milliseconds. In addition, the server clock on which the tool is running must be the same as the database server clock
- The preprocessing functions LPAD(char,length[,pad_string]) and RPAD(char,length[,pad_string]) refer to the corresponding built-in functions in MySQL. The length parameter is the total length of the return value that will be displayed on the terminal. In most character sets, this will be the number of characters in the return value. However, in some multibyte character sets, the displayed length of the string may be different from the actual number of characters in the string, so this function is not safe when processing multibyte values

Data Processing

Conditional Expressions: When defining a control file, you can use conditional expressions to perform simple logical and arithmetic operations to achieve more complex data processing capabilities.

Conditional Expression Syntax

--Simple Case expression

```
CASE <expr> WHEN <constant> THEN [constant | expr] ELSE [constant | expr] END;
```

--Search Case expression

```
CASE WHEN <condition> THEN [constant | expr] ELSE [constant | expr] END;
```

The parameter **<expr>** is a conditional operation expression. The parameter value is as follows:

Conditional Expression	Return Type	Description
is [not] null	boolean	Perform empty or non-empty judgment. Example: case when col1 is null then 'Y' else 'N' end;
not <condition>	boolean	Negates a truth conditional expression. Example: case when not col1 is null then 'N' else 'Y' end;
[not] in (string_list)	boolean	Perform relational judgment. Example: case when col1 in ('1','3') then 'Y' else 'N' end;

Data Processing Examples

```

lang=java
(
    c1 "lower(c1)" map(1),
    c2 "ltrim(c2)" map(2),
    c3 "rtrim(c3)" map(3),
    c4 "substr(c4,0,5)" map(4),
    c5 "trim(c5)" map(5),
    c6 "upper(c6)" map(6),
    c7 "nanvl(c7,'0')" map(7),
    c8 "replace(c8,'a','A')" map(8),
    c9 "nvl(c9,'null')" map(9),
    c10 "length(c10)" map(10),
    c11 "lpad(c11,5,'x')" map(11),
    c12 "rpad(c12,5,'x')" map(12),
    c13 "convert(c13,'utf-8','gbk')" map(13),
    c14 "concat(c14, '_suffix')" map(14),
    c15 "none" map(15),
    c16 "systimestamp" map(16),
    c17 "constant('1')" map(17),
    c18 "lpadb(c18,5,'x')" map(18),
    c19 "rpadb(c19,5,'x')" map(19),
    c20 "case when length(trim(c20))<18 then 'Y' else 'N' end" map(20),
        -- The value of the c20 column is matched with the conditional truth value. If it is true, the value of the corresponding column is returned.
    c21 "case length(trim(c21)) when '1' then 'one' when '2' then 'two' else 'unknown' end" map(21),
        -- The value of the c21 column is matched with the condition, and if the match is successful, the value of the corresponding column is returned
    C22 "SYSDATE" map(22),
    C23 "MASK(C23)" map(23),
        -- The value of column c22 is the current date
        -- The value of the c23 column is anonymized. This is only valid for the column name. The uppercase and lowercase letters and numbers in the column are replaced with the default anonymized characters (default anonymized characters: uppercase X, lowercase x, number n)
    C24 "MASK_FIRST_N(C24,'A','a','b',3)" map(24),
        -- The value of the c24 column specifies the desensitized characters of uppercase and lowercase letters and numbers (the default N is 0, starting from the first character)
    C25 "MASK_LAST_N(C25,'A','a','b',3)" map(25),
        -- The value of the c25 column specifies the desensitized characters of uppercase and lowercase letters and numbers (the default N is 0, starting from the last character)
    C26 "MASK_SHOW_FIRST_N(C26,'A','a','b',3)" map(26),
    C27 "MASK_SHOW_LAST_N(C27,'A','a','b',3)" map(27),
    C28 "REVERSE(C28)" map(28),
)

```

-- Convert the letters in the values of column c1 to lowercase
-- The values in column c2 are truncated with spaces starting from the left
-- The value of column c3 is truncated with spaces starting from the right
-- The first position of the value in column c4 is a string of 5 characters long
-- The values of column c5 are truncated with spaces on both sides
-- Convert the letters in the values of column c6 to uppercase
-- The value of column c7 is validated as numeric, and 0 is returned if it is not a numeric value
-- Replace a with A in the value of column c8
-- The value of column c9 is checked to be null. If it is null, a null string is returned.
-- The length of the value of column c10 is calculated
-- Append a 5-byte string 'x' to the left of the value of column c11
-- Append a 5-byte string 'x' to the right of the value of column c12
-- The value of the c13 column is converted from gbk to utf-8 character encoding
-- The value of column c14 is concatenated with the constant
-- The value of column c15 is not processed in any way and the value of the corresponding column is returned directly
-- The value of the c16 column is not processed in any way and the timestamp of the current cluster is directly returned.
-- The value of column c17 is not processed and the constant 1 is returned.
-- Append a 5-byte (single) character 'x' to the left of the value of column c18
-- Append a 5-byte (single) character 'x' to the right of the value of column c19
-- The value of the c20 column is matched with the conditional truth value. If it is true, the value of the corresponding column is returned.
-- The value of the c21 column is matched with the condition, and if the match is successful, the value of the corresponding column is returned
-- The value of column c22 is the current date
-- The value of the c23 column is anonymized. This is only valid for the column name. The uppercase and lowercase letters and numbers in the column are replaced with the default anonymized characters (default anonymized characters: uppercase X, lowercase x, number n)
-- The value of the c24 column specifies the desensitized characters of uppercase and lowercase letters and numbers (the default N is 0, starting from the first character)
-- The value of the c25 column specifies the desensitized characters of uppercase and lowercase letters and numbers (the default N is 0, starting from the last character)
-- The value of the c26 column specifies the number of characters not to be masked (the default N is 0, starting from the first character)
-- The value of the c27 column specifies the number of characters not to be masked (the default N is 0, starting from the last character)
-- The value of column c28 has the characters in reverse order

Agenda



- **Lesson 4.1**
 - **Introduction to OceanBase and MySQL Compatibility**
 - **Introduction to OMS Migration and Synchronization**
 - **OBLogProxy Introduction**
- **Lesson 4.2**
 - **OBDUMPER/OBLOADER Introduction**
 - **Introduction to SQL Command Export**
 - **Introduction to Other Ecological Components**
 - **Comparison of Components**

Several Commands

- SELECT INTO OUTFILE
- LOAD DATA
- INSERT INTO SELECT

SELECT INTO OUTFILE Export

SELECT INTO OUTFILE Export

SELECT INTO OUTFILE statement is a common way to export data. SELECT INTO OUTFILE statement can limit the fields to be exported, which is a good way to meet the needs of some scenarios where primary key fields do not need to be exported. It is a very convenient way to import and export data when used with LOAD DATA INFILE statement to import data

Grammar

```
SELECT column_list_option
  INTO OUTFILE file_route_option
    [format_of_field_option]
    [start_and_end_option]
  FROM table_name_list
  [WHERE where_conditions]
  [GROUP BY group_by_list [HAVING having_search_conditions]]
  [ORDER BY order_expression_list]

  column_list_option:
    column_name[,column_name]...

  file_route_option:
    '/path/file'
    | 'oss://$PATH/$FILENAME/?host=$HOST&access_id=$ACCESS_ID&access_key=$ACCESSKEY'

  format_of_field_option:
    {FIELDS | COLUMNS}
    [TERMINATED BY 'string']
    [[OPTIONALLY] ENCLOSED BY 'char']
    [ESCAPED BY 'char']

  start_and_end_option:
    LINES
    [STARTING BY 'string']
    [TERMINATED BY 'string']
```

SELECT INTO OUTFILE Export

Parameter	Required	Description	Example
column_list_option	Y	Export column options. If you want to select all data, you can use *. column_name: column name.	SELECT col1,col2,col3 ...
file_route_option	Y	Select the export file path, support exporting to Alibaba Cloud OSS Description Since Alibaba Cloud OSS has a file size limit, files larger than 5 GB will be split into multiple files when exported to OSS, each file is smaller than 5 GB	... INTO OUTFILE '/home/admin/student.sql' ...
format_of_field_option	N	Export field format options. Specify the format of each field in the output file, specified by the FIELDS or COLUMNS clause <ul style="list-style-type: none"> TERMINATED BY: Used to specify the symbol between field values. For example, TERMINATED BY ',' specifies a comma as the mark between two field values ENCLOSED BY: A symbol used to specify the enclosing field value. For example, ENCLOSED BY '\"' means that the character value is enclosed between double quotes. If the OPTIONALLY keyword is used, the specified character enclosing is used only for string type values ESCAPED BY: Used to specify an escape character. For example, ESCAPED BY '*' specifies an asterisk (*) as the escape character to replace the default escape character (\) 	... TERMINATED BY ',' ENCLOSED BY '\"' ...
start_and_end_option	N	The start and end character options for exporting data lines. Specify the start and end characters of each line in the output file, set by the LINES clause <ul style="list-style-type: none"> STARTING BY: Specifies the character at the beginning of each line TERMINATED BY: Specifies the end character of each line 	... LINES TERMINATED BY '\n' ... Indicates that a line will end with a newline character
FROM table_name_list	Y	Specifies the object to select data from	... FROM tbl1,tbl2 ...
WHERE where_conditions	N	Specify the filter conditions, and the query results will only include data that meets the conditions	... WHERE col1 > 100 ...
GROUP BY group_by_list	N	Specifies the grouping field, usually used with aggregate functions. Description All columns after the SELECT clause that do not use aggregate functions must appear after the GROUP BY clause	... GROUP BY col1,col2 ...
HAVING having_search_conditions	N	Filters each group of data after grouping. The HAVING clause is similar to the WHERE clause, but the HAVING clause can use cumulative functions (such as SUM, AVG, etc.)	... HAVING SUM(col1) < 160 ...
ORDER BY order_expression_list	N	Specifies that the result set displays the query results in ASC or DESC format according to one or more columns. If ASC or DESC is not specified, ASC is used by default <ul style="list-style-type: none"> ASC: indicates ascending order DESC: indicates descending order 	... ORDER BY col1,col2 DESC ...

SELECT INTO OUTFILE Export Example

1、 Create table tbl1 in the test database of tenant mysql001 and insert data

2、 Set the exported file path

- Set the system variable `secure_file_priv` to configure the path that can be accessed when importing or exporting files

3、 The export file name is `tbl1.sql`; specify a comma as a marker between two field values; use the " character to wrap string type values; use a newline character as the end marker

```
obclient [test]> CREATE TABLE tbl1(col1 INT PRIMARY KEY,col2
varchar(128),col3 INT);
Query OK, 0 rows affected
```

```
obclient [test]> INSERT INTO tbl1
VALUES(1,'one',80),(2,'two',90),(3,'three',100);
Query OK, 3 rows affected
Records: 3  Duplicates: 0  Warnings: 0
```

```
obclient [test]> SELECT * FROM tbl1;
+----+-----+-----+
| col1 | col2 | col3 |
+----+-----+-----+
| 1 | one | 80 |
| 2 | two | 90 |
| 3 | three | 100 |
+----+-----+
3 rows in set
```

```
obclient [test]> SELECT * INTO OUTFILE
'/home/admin/tbl1.sql'
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY ""
LINES TERMINATED BY '\n'
FROM tbl1;
Query OK, 3 rows affected
```

```
[xxx@xxx /home/admin]# cat tbl1.sql
1,"one",80
2,"two",90
3,"three",100
```

LOAD DATA Import

LOAD DATA Import

- LOAD DATA syntax, users import data from the outside into the database. Currently supports importing from the server, client, and OSS files

Precautions

- LOAD DATA statement is not allowed for tables with triggers
- To import data from an external file, you need the FILE privilege and the following settings:
 - When loading server-side files, you need to set the system variable `secure_file_priv` in advance to configure the path that can be accessed when importing or exporting files
 - When loading local files on the client, you need to add the option "`--local-infile[=1]`" when starting the MySQL/OBClient client to enable the function of loading data from the local file system

LOAD DATA Process

1. Parsing files: OceanBase database reads the data in the file according to the file name entered by the user, and decides to parse the data in the input file in parallel or serially according to the specified parallelism
2. Distribute data: Since OceanBase is a distributed database, the data of each partition may be distributed on different OBServer nodes. LOAD DATA will calculate the parsed data and determine which OBServer node the data needs to be sent to
3. Insert data: When the target OBServer node receives the data, it executes the INSERT operation locally to insert the data into the corresponding partition

LOAD DATA Import

Grammar

```
LOAD DATA
[/*+ PARALLEL(N) load_batch_size(M) APPEND */]
| [/*+ PARALLEL(N) direct(bool, int) */]
[REMOTE_OSS | LOCAL] INFILE 'file_name'
[REPLACE | IGNORE]
INTO TABLE table_name
[ {FIELDS | COLUMNS}
  [TERMINATED BY 'string']
  [[OPTIONALLY] ENCLOSED BY 'char']
  [ESCAPED BY 'char']
]
[LINES
  [STARTING BY 'string']
  [TERMINATED BY 'string']
]
[IGNORE number {LINES | ROWS}]
[(column_name_var
  [, column_name_var] ...)]
```

Parameter Parsing - Part I

Parameter	Description
parallel(N)	The parallelism of loading data, N defaults to 4
load_batch_size(M)	Specify the batch size for each insert. The default value of M is 100. The recommended value range is [100,1000]
APPEND	Use Hint to enable bypass import, which means that space is allocated directly in the data file and data is written. APPEND Hint is equivalent to using direct(false, 0) by default, and can also realize the function of collecting statistics online (GATHER_OPTIMIZER_STATISTICS Hint)
direct	Use Hint to enable bypass import. The bool parameter in direct(bool, int) indicates whether the given CSV file needs to be sorted. True indicates that sorting is required; int indicates the maximum number of error rows to be tolerated
REMOTE_OSS LOCAL	Optional <ul style="list-style-type: none"> REMOTE_OSS is used to specify whether to read data from the OSS file system LOCAL is used to specify whether to read data from the client's local file system. If the LOCAL parameter is not used, data will be read from the file system on the server (OBServer node)
file_name	Specifies the path and file name of the input file. file_name has the following format: <ul style="list-style-type: none"> Import file in OBServer node or client: <code>/\$PATH/\$FILENAME</code> Import the file on OSS: <code>oss://\$PATH/\$FILENAME/?host=\$HOST&access_id=\$ACCESS_ID&access_key=\$ACCESSKEY</code>
REPLACE IGNORE	If a unique key conflict occurs, REPLACE means overwriting the conflicting row, and IGNORE means ignoring the conflicting row. LOAD DATA uses the table's primary key to determine whether the data is duplicated. If the table does not have a primary key, there is no difference between the REPLACE and IGNORE options. By default, when encountering duplicate data, LOAD DATA will record the erroneous data in the log file

Parameter Parsing - Part II

Parameter	Description
table_name	The name of the table to import data into, supporting partitioned tables and non-partitioned tables
FIELDS COLUMNS	<p>Specify the format of the field</p> <ul style="list-style-type: none"> • ENCLOSED BY: Set the modifier of the exported value • TERMINATED BY: Set the terminator of the exported column • ESCAPED BY: Set the characters ignored by the exported value
LINES STARTING BY	Specify line start character
LINES TERMINATED BY	Specify line terminator
IGNORES number { LINES ROWS }	<p>Ignore the first few lines, LINES means the first few lines of the file, and ROWS means the first few lines of data specified by the field separator. By default, the fields in the input file are mapped to the columns in the table one by one. If the input file does not contain all the columns, the missing columns will be filled by default according to the following rules:</p> <ul style="list-style-type: none"> • Character type: Empty string • Numeric type: 0 • Date type: 0000-00-00
column_name_var	Specify the column names to import

LOAD DATA Example

1. Server-side File Import

```
LOAD DATA /*+ PARALLEL(4) APPEND */  
INFILE '/home/admin/a.csv'  
INTO TABLE t;
```

2. Import Data from Client (Local) Files

```
# Open a terminal or command prompt window and enter the following command to start the client  
obclient --local-infile -hxxx.xxx.xxx.xxx -P2881 -uroot@mysql001 -p***** -A -Dtest
```

```
# In the client, execute the LOAD DATA LOCAL INFILE statement to load the local data file  
LOAD DATA LOCAL INFILE '/home/admin/test_data/tbl1.csv' INTO TABLE tbl1 FIELDS TERMINATED BY ',';
```

3. Importing Data from OSS Files

```
LOAD DATA /*+ direct(true,1024) parallel(16) */ REMOTE_OSS INFILE 'oss://antsys-  
oceanbasebackup/backup_rd/xiaotao.ht/lineitem2.tbl?host=***.oss-  
cdn.***&access_id=***&access_key=***' INTO TABLE tbl1 FIELDS TERMINATED BY ',';
```

INSERT INTO Grammar

Grammar

```
INSERT INTO target_table_name[(target_col_name[, target_col_name] ...)]
SELECT [(source_col_name[, source_col_name] ...)]
FROM source_table_name
[WHERE expr];
```

Parameter Explanation

Parameter	Description
target_table_name	Target table for data migration
target_col_name	The column name of the target table. If you want to update all column data of the target table, you can leave the column name blank
source_col_name	The column name of the source table. Select the column to be migrated. If you want to select all data, you can use * to indicate. Note: The number of selected columns must be consistent with the number of columns in the target table
source_table_name	Source table for data migration
WHERE expr	The filter condition for migrating data. If it is not filled, all rows selected by SELECT will be migrated

INSERT INTO Example

```
obclient [test]> SELECT * FROM tbl1;
+----+----+----+
| id | name | age |
+----+----+----+
| 1 | ab  | 8 |
| 2 | bc  | 18 |
| 3 | cd  | 14 |
| 4 | de  | 19 |
| 5 | ef  | 6 |
| 6 | fg  | 15 |
+----+----+----+
6 rows in set
```

```
obclient [test]> DESC tbl2;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| col1  | int(11) | YES  |    | NULL    |    |
| col2  | int(11) | YES  |    | NULL    |    |
+-----+-----+-----+-----+-----+
2 rows in set
```

INSERT INTO Example

```
obclient [test]> SELECT * FROM tbl2;  
Empty set  
  
obclient [test]> INSERT INTO tbl2 SELECT id,age FROM tbl1 WHERE age > 10;  
Query OK, 4 rows affected  
Records: 4  Duplicates: 0  Warnings: 0  
  
obclient [test]> SELECT * FROM tbl2;  
+----+----+  
| col1 | col2 |  
+----+----+  
|  2 |  18 |  
|  3 |  14 |  
|  4 |  19 |  
|  6 |  15 |  
+----+----+  
4 rows in set
```

Bypass Import

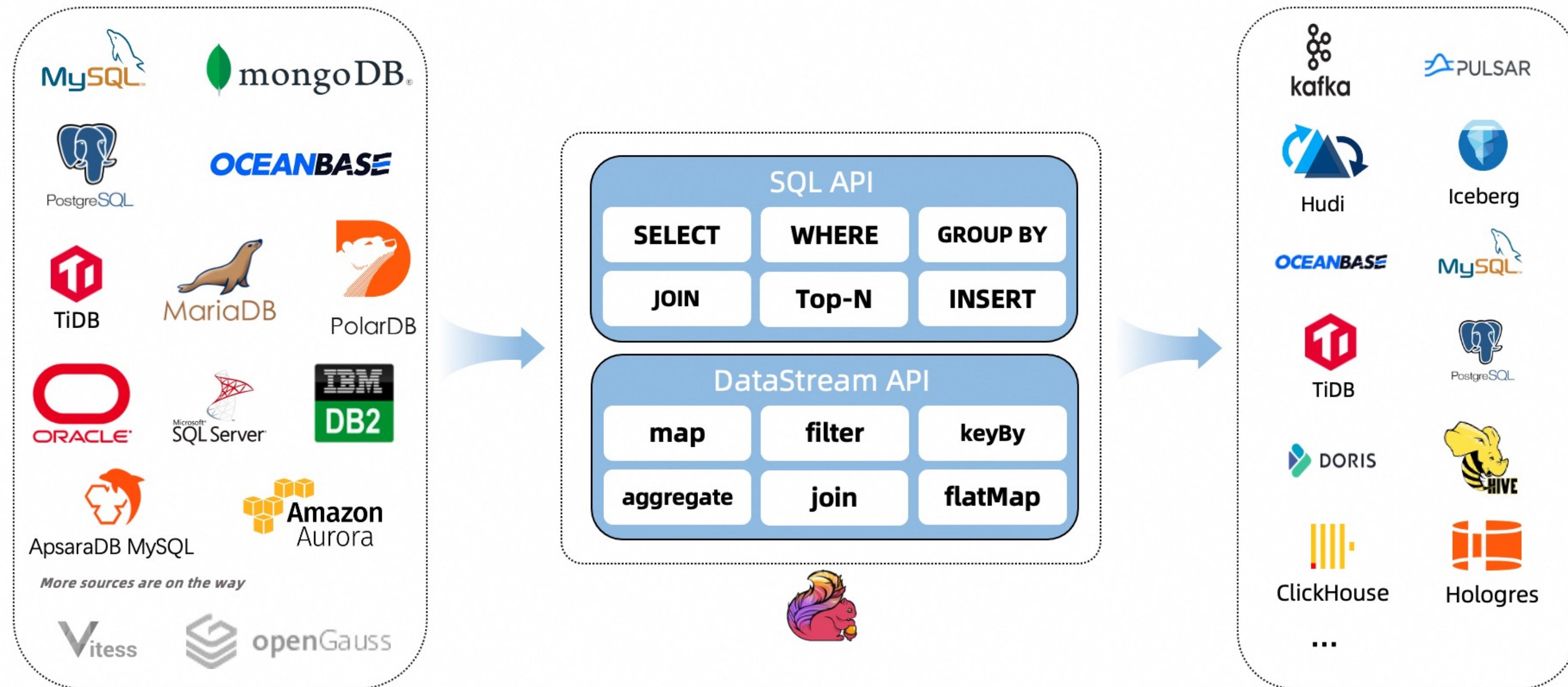
- LOAD DATA /*+ direct(bool, int) */
- LOAD DATA /*+ APPEND */
- INSERT /*+ append */ INTO SELECT

Agenda



- **Lesson 4.1**
 - Introduction to OceanBase and MySQL Compatibility
 - Introduction to OMS Migration and Synchronization
 - OBLogProxy Introduction
- **Lesson 4.2**
 - OBDUMPER/OBLOADER Introduction
 - Introduction to SQL Command Export
 - **Introduction to Other Ecological Components**
 - Comparison of Components

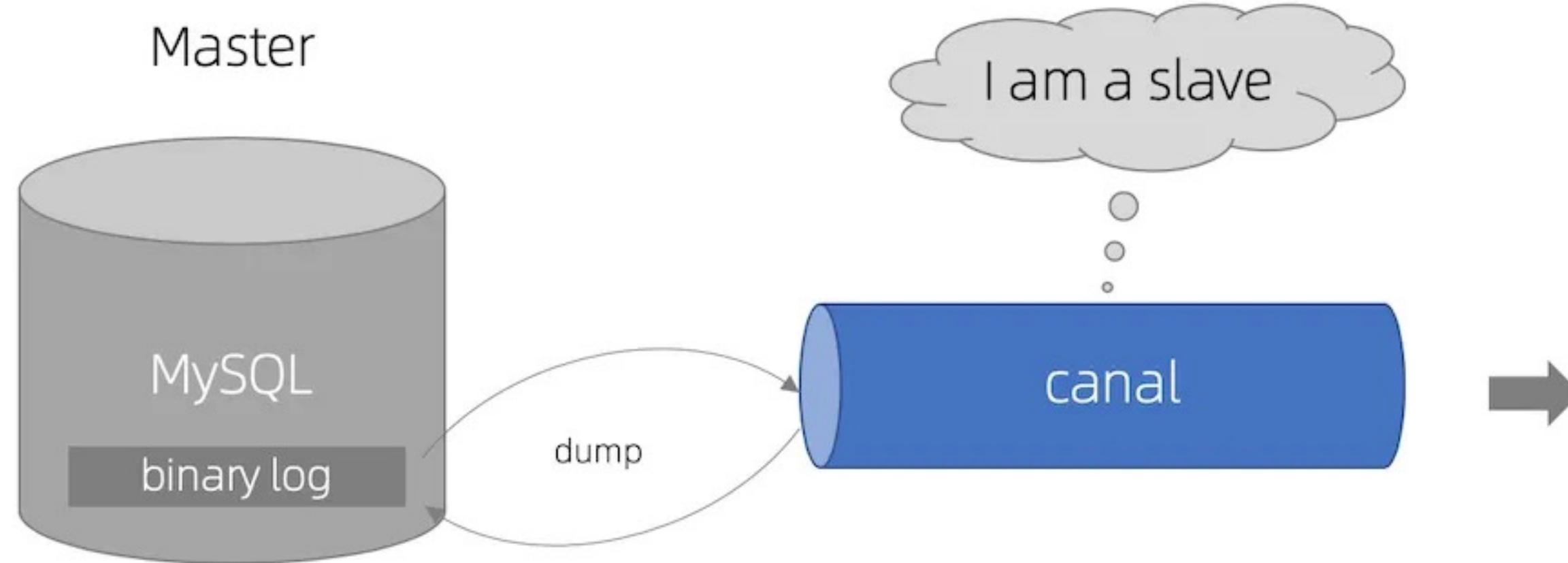
Flink CDC



MySQL migration to OceanBase case: <https://en.oceanbase.com/docs/common-oceanbase-database-1000000001103629>



Canal

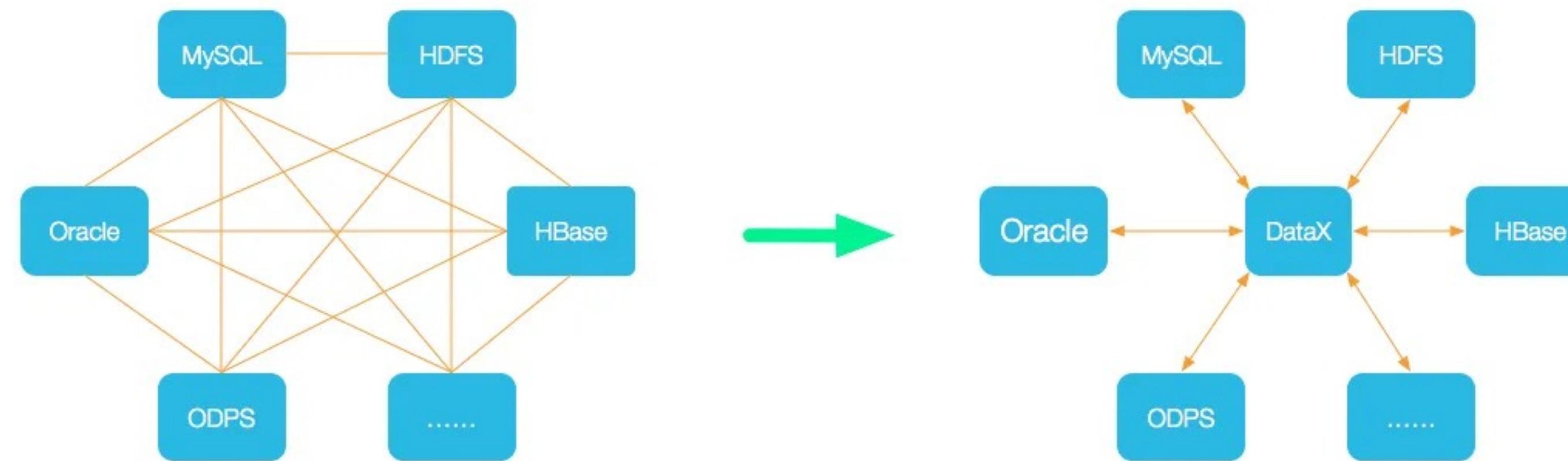


...

MySQL migration to OceanBase case: <https://en.oceanbase.com/docs/common-oceanbase-database-1000000001103636>



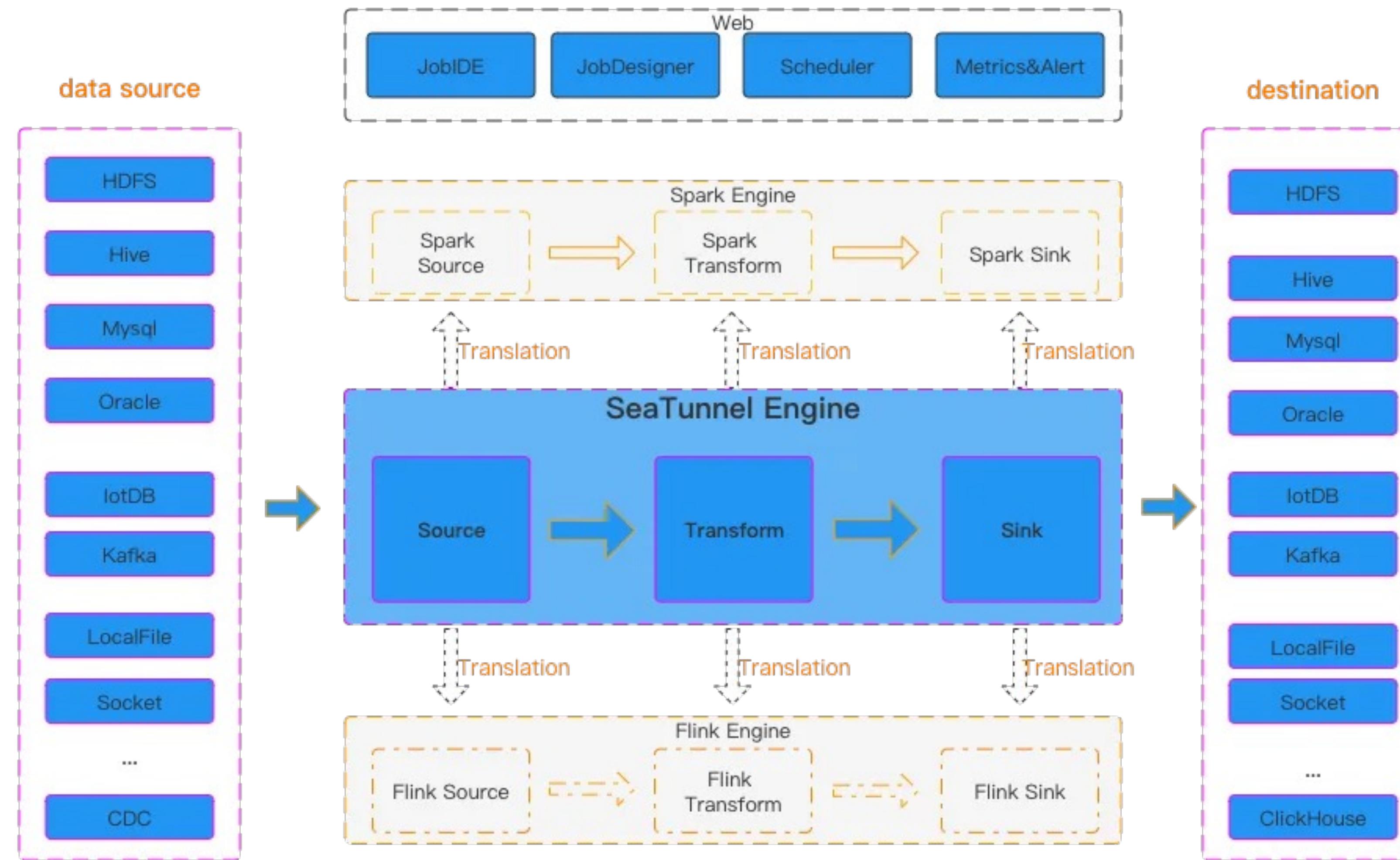
DataX



MySQL migration to OceanBase case: <https://en.oceanbase.com/docs/common-oceanbase-database-10000000001103634>



SeaTunnel



Agenda



- **Lesson 4.1**
 - **Introduction to OceanBase and MySQL Compatibility**
 - **Introduction to OMS Migration and Synchronization**
 - **OBLogProxy Introduction**
- **Lesson 4.2**
 - **OBDUMPER/OBLOADER Introduction**
 - **Introduction to SQL Command Export**
 - **Introduction to Other Ecological Components**
 - **Comparison of Components**

Comparison of Various Components

Migration plan	Schema Migration	Full data migration	Incremental data migration	Data Verification	Supported data sources
OMS	Support	Support	Support	Support	OceanBase, MySQL, MariaDB, PostgreSQL、GreenPlum, Hbase, TiDB, Kafka, RocketMQ
OBLogProxy	Support	Not supported	Support	Not supported	OceanBase, MySQL Binlog ecological tools, CDC ecological tools, OBLogClient
OBLOADER & OBDUMPER	Support	Support	Not supported	Not supported	OceanBase
SQL command migration	Support	Support	Not supported	Not supported	Mainstream databases, SQL text, CSV files, etc
DataX	Not supported	Support	Not supported	Not supported	More, see the official documentation for specific ranges
Canal	Support	Support	Support	Not supported	More, see the official documentation for specific ranges
Flink cdc	Support	Support	Support	Not supported	More, see the official documentation for specific ranges
SeaTunnel	Support	Support	Support	Not supported	More, see the official documentation for specific ranges

Thank You!



OceanBase Official website:
<https://oceanbase.github.io/>



GitHub Discussions:
<https://github.com/oceanbase/oceanbase/discussions>

