MySQL最酷和最新功能

Ivan Ma （马楚成）

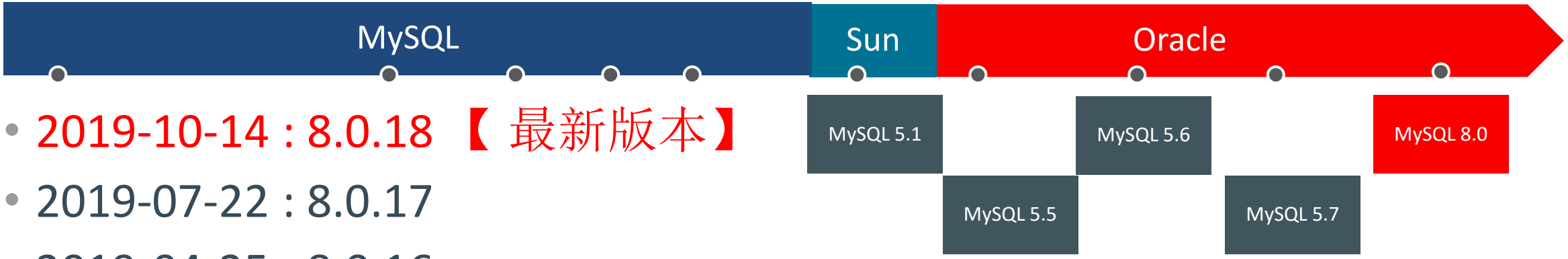
20191214

深圳[*3306π*]技术大会

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

最新版本： MySQL 8.0.18



- 2019-10-14 : 8.0.18 【 最新版本】
- 2019-07-22 : 8.0.17
- 2019-04-25 : 8.0.16
- 2019-02-01 : 8.0.15 / 2019-01-21 : 8.0.14
- 2018-10-22 : 8.0.13
- 2018-07-27 : 8.0.12
- 2018-04-19 : 8.0.11 General Availability

Agenda

MySQL InnoDB Cluster Basics

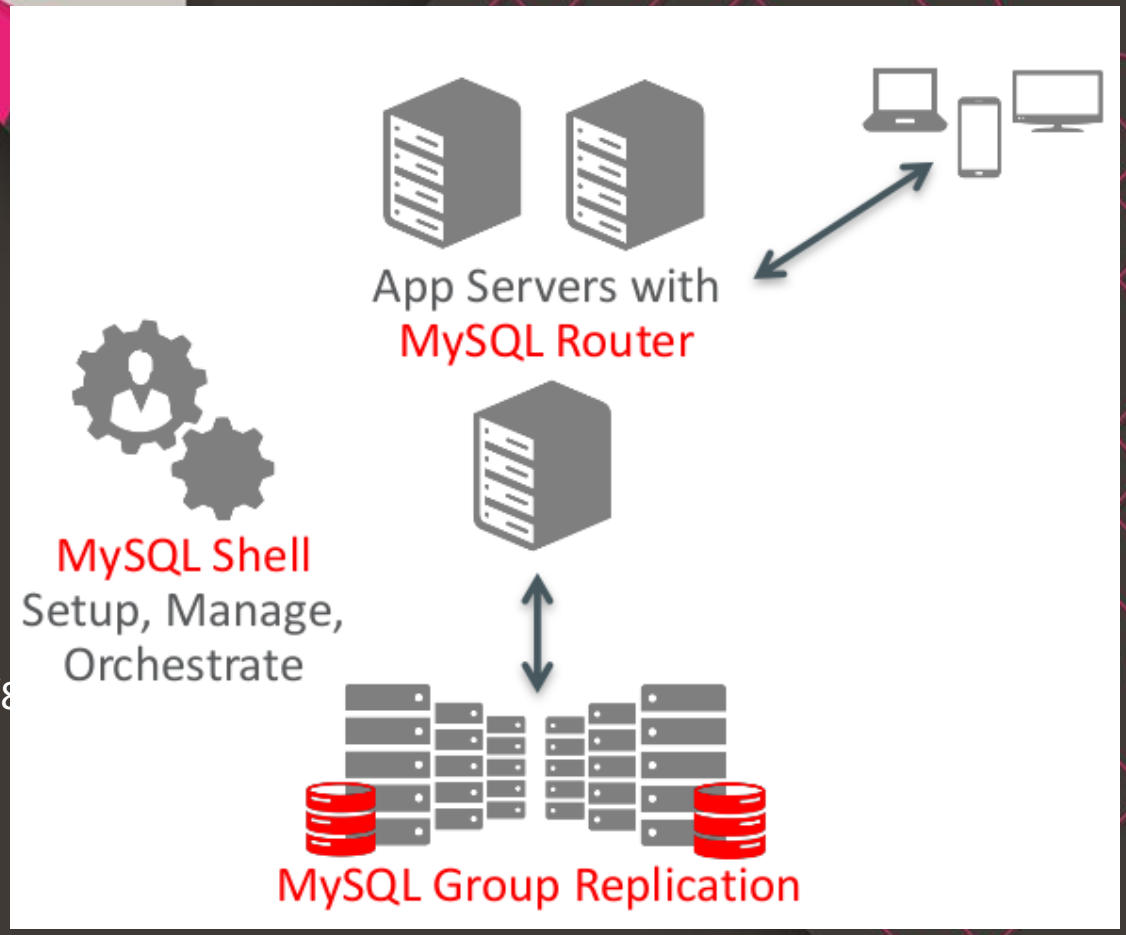
Deployment Example

DeepDive - InnoDB Cluster Configuration

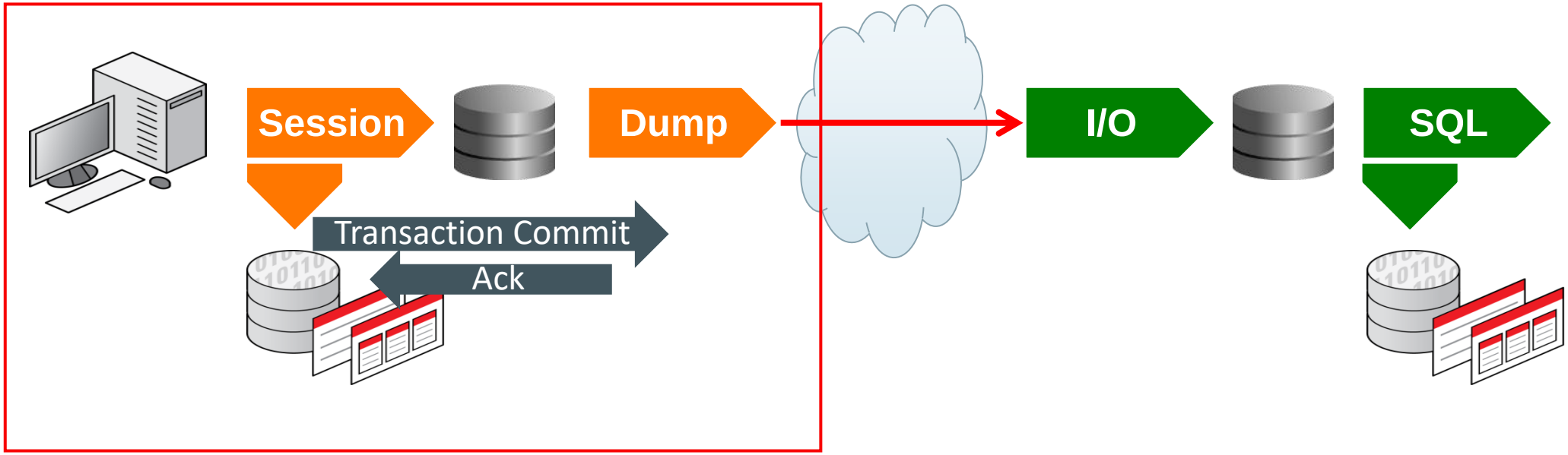
MySQL 8.0.18 – 好东西 !

MySQL InnoDB Cluster Basics

https://dev.mysql.com/doc/refman/8



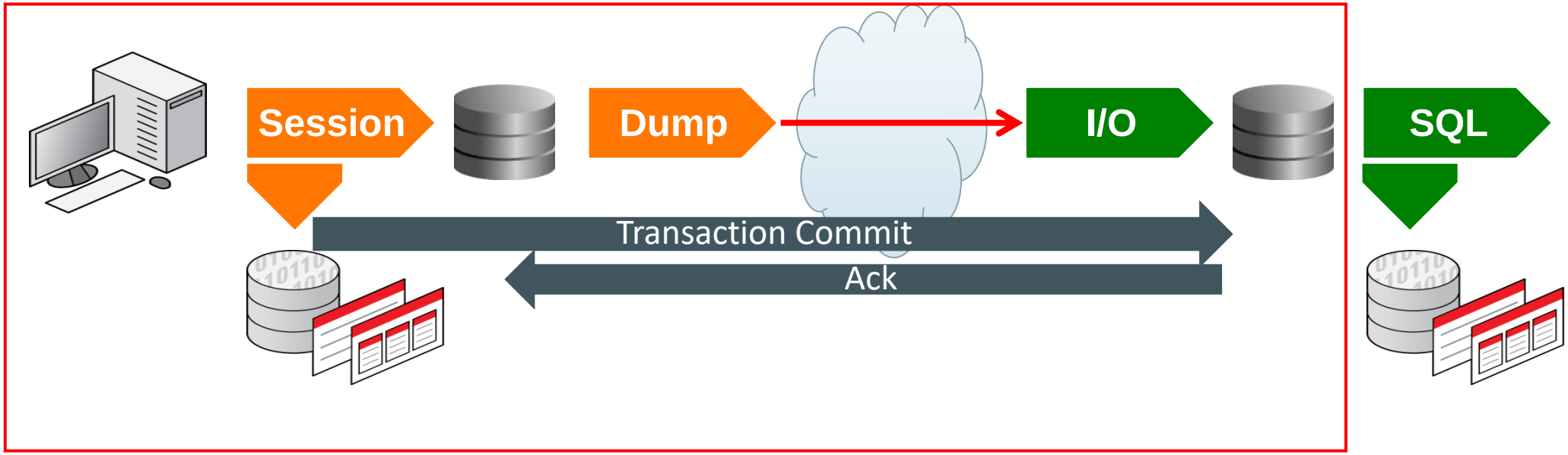
MySQL Replication –异步复制



- “DATA”在提交到存储引擎之前先提交到binlog
- “COMMIT” 在本地完成

- “DUMP”线程从binlog读取事件并将其传输到SLAVE服务器
- I/O线程读取复制事件，并将其存储到RELAY日志中
- SQL线程：读取RELAY日志并将其写入存储引擎

MySQL Replication – 半同步复制

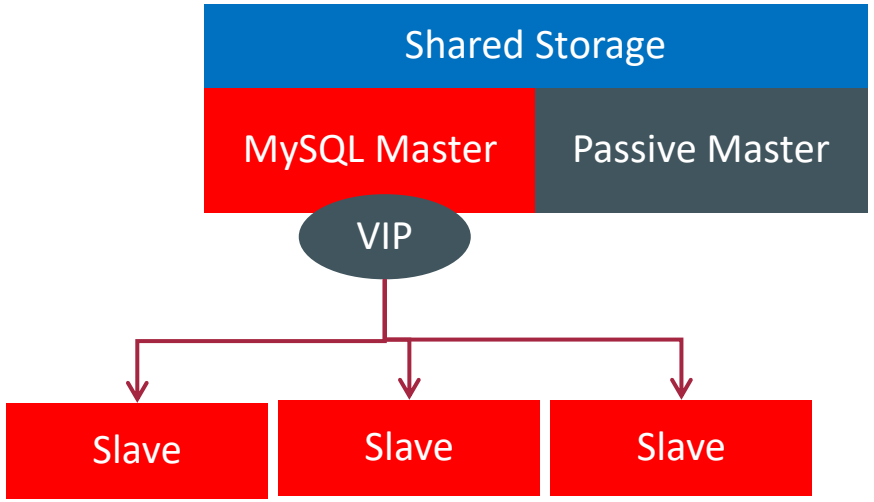


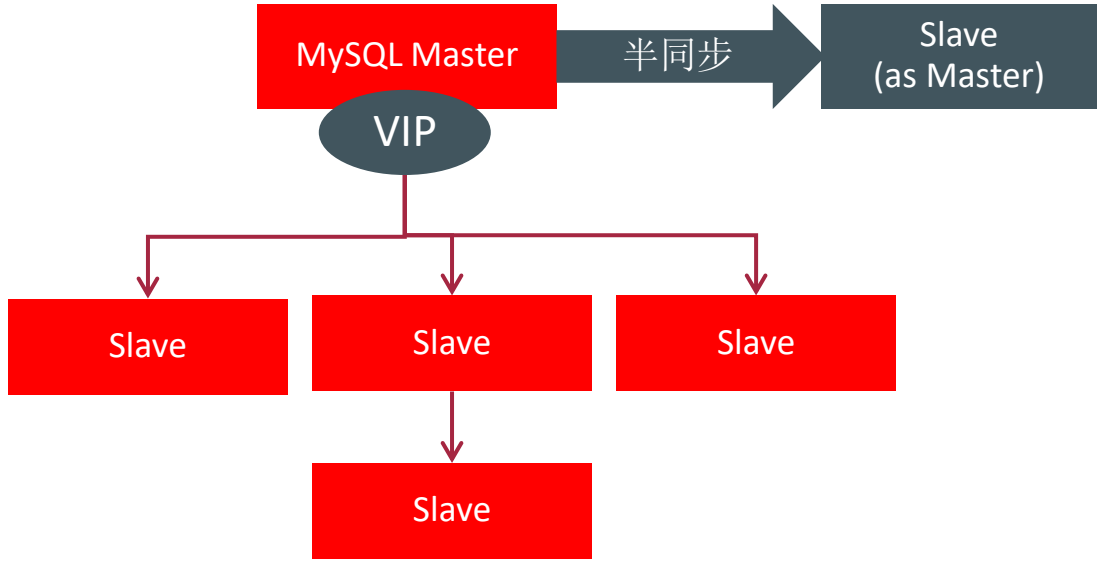
- “DATA”在提交到存储引擎之前先提交到binlog
- “DUMP”线程从binlog读取事件并将其传输到SLAVE服务器
- I/O线程读取复制事件，并将其存储到RELAY日志中
- “COMMIT”在提交完成

- SQL线程： 读取RELAY日志并将其写入存储引擎

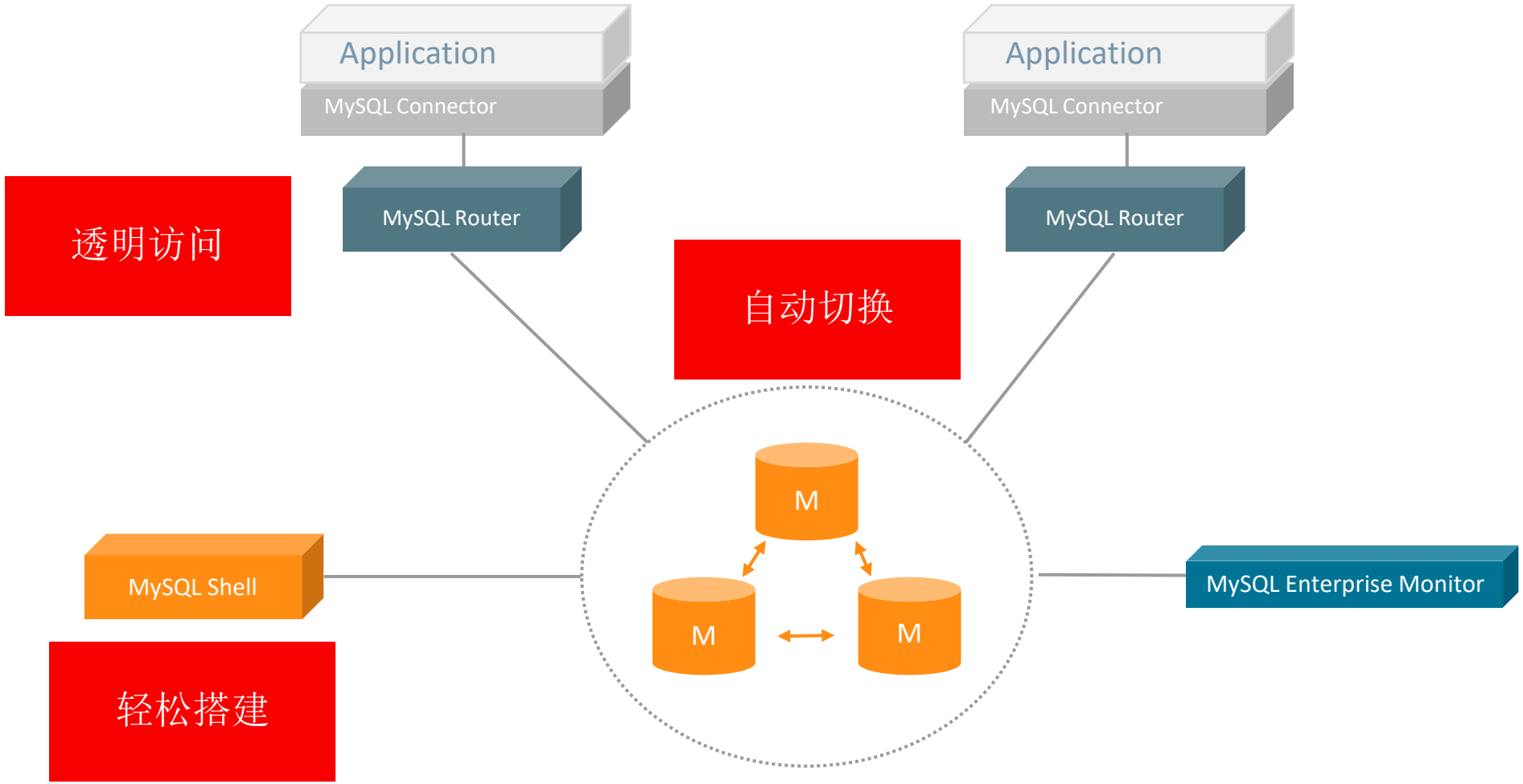
MySQL复制 (异步/ 半同步）

- Master & Slave （1 + 1）/ 主从 （1 + M）
- Master → Slave → Slave （1 + 1 + M …) / (1 + M + M …)





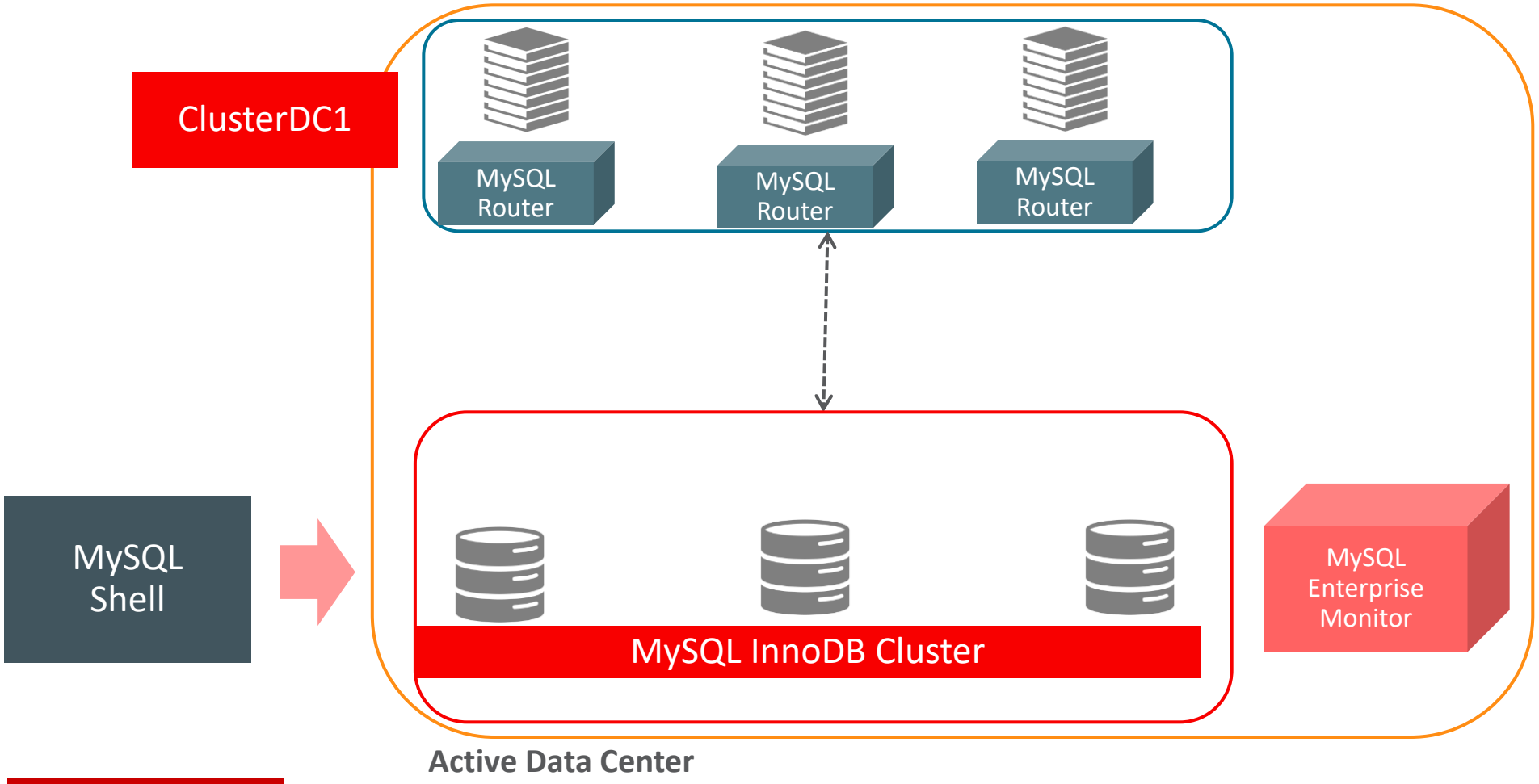
MySQL InnoDB Cluster:架构



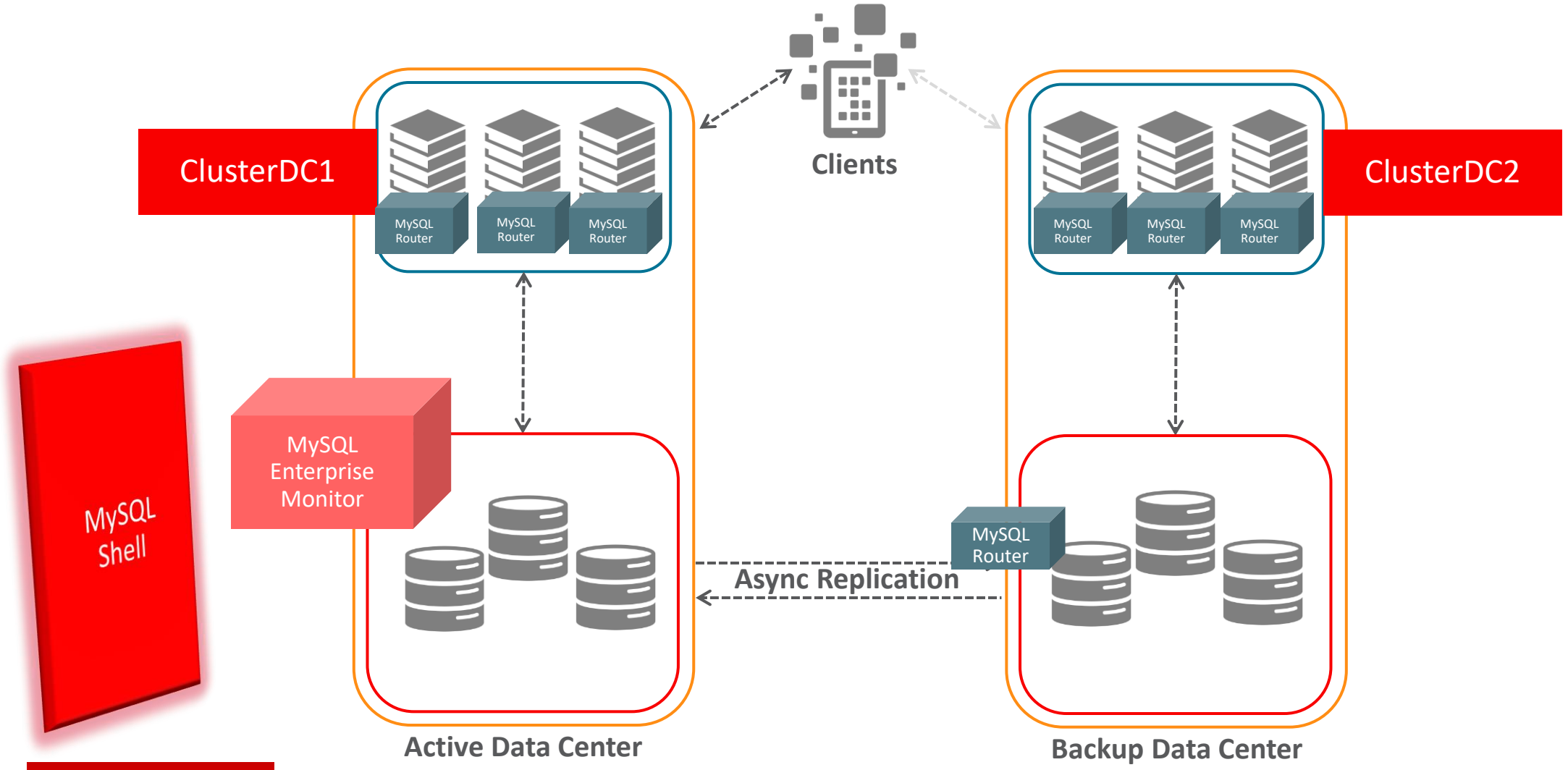
Deployment Examples
MySQL InnoDB Cluster

https://dev.mysql.com/doc/refman/8.0/en/mysql-innodb-cluster-production-deployment.html

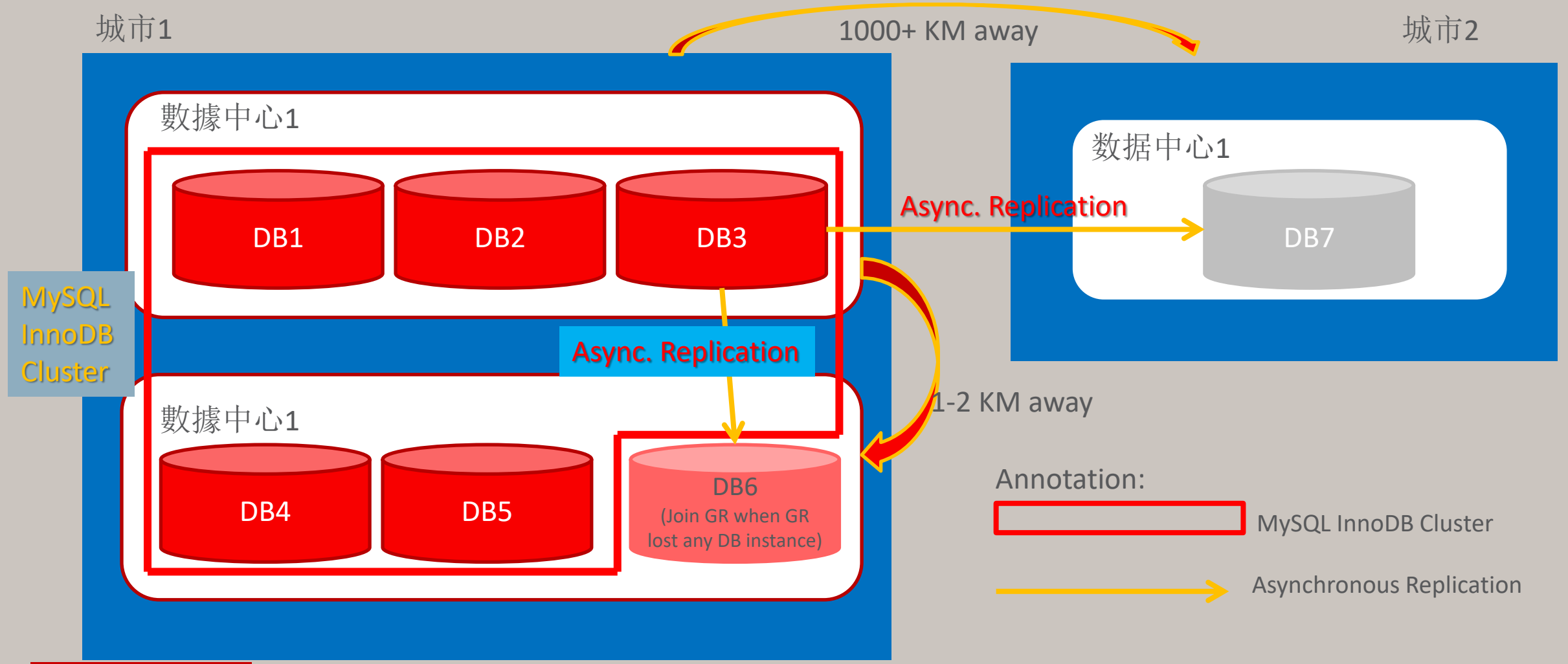
Single Data Center



跨地域异步复制



两地三个数据中心



探究MySQL InnoDB集群配置

MySQL InnoDB Cluster

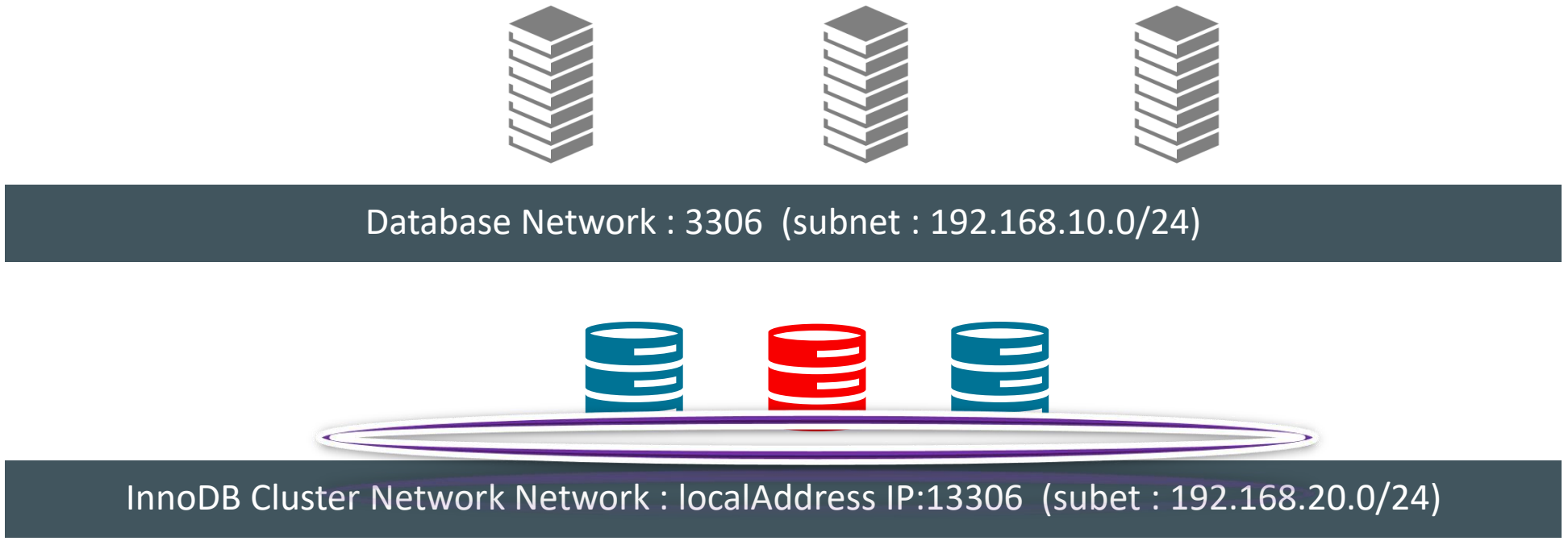
MySQL Server – Persisted Variables	默认值	试试看
group_replication_consistency	EVENTUAL	BEFORE_ON_PRIMARY_FAILOVER
group_replication_ip_whitelist group-replication-local-address	AUTOMATIC EMPTY	set to be the subnet of the PRIVATE IP
group_replication_member_expel_timeout	0 [NETWORK RELIABILITY]	set to the value of 30 (seconds)
group_replication_autorejoin_tries	- [NETWORK RELIABILITY]	set to 12 (5 mins interval for each retry)
group_replication_unreachable_majority_timeout	0 [NETWORK RELIABILITY]	Please set a value - the timeout value that the application will wait in the access minority (when there is network partition happening). e.g. (for 2 minutes wait time ➔ 120)
group_replication_member_weight	50 (for all nodes)	e.g. Configured for Node1,Node2,Node3 as 40,50,60 respectively
group-replication-exit-state-action	ABORT_SERVER / READ_ONLY	if 8.0.18 [OFFLINE_MODE]
report_host		To be defined for the hostname/ip Interface with Application

Innodb群集内网络和外网络

- 内网/专用网络 - 交换数据

  group_replication_ip_whitelist

  group-replication-local-address



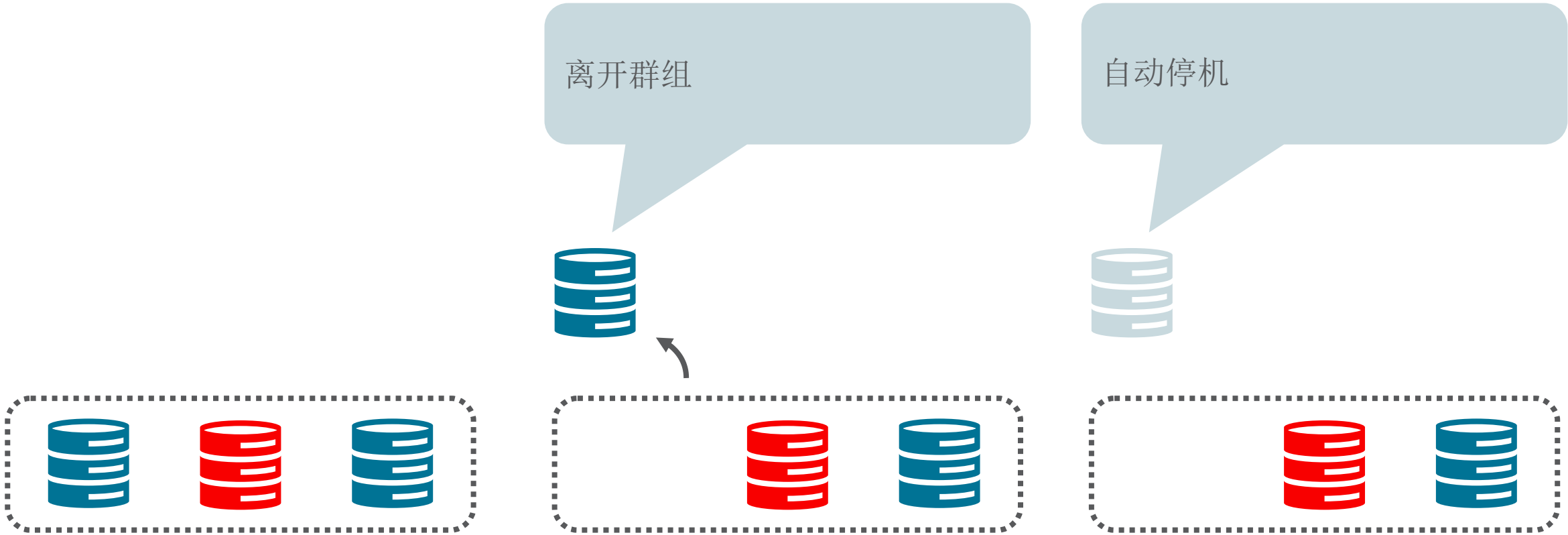
```
mysqlsh --uri gradmin:grpass@primary:3310 -e "

var x = dba.createCluster('mycluster',
{exitStateAction:'OFFLINE_MODE',
        consistency:'BEFORE_ON_PRIMARY_FAILOVER',
        expelTimeout:30,
        memberSslMode:'REQUIRED',
        ipWhitelist:'192.168.56.0/24',
        localAddress:'node1:13310',
        clearReadOnly:true,
        interactive:false,
autoRejoinTries:120,
        memberWeight:80
        })
x = dba.getCluster()
print(x.status())
“
```

```
mysqlsh --uri gradmin:grpass@primary:3310 -e "
x = dba.getCluster()
x.addInstance('gradmin:grpass@node1:3320',
{exitStateAction:'OFFLINE_MODE',
        recoveryMethod:'incremental',
        localAddress:'node2:13310',
        autoRejoinTries:120,
        memberWeight:70
        })
print(x.status())
"
```

```
mysqlsh --uri gradmin:grpass@primary:3310 -e "
x = dba.getCluster()
x.addInstance('gradmin:grpass@node1:3320',
{exitStateAction:'OFFLINE_MODE',
        recoveryMethod:'incremental',
        localAddress:'node3:13310',
        autoRejoinTries:120,
        memberWeight:60
        })
print(x.status())
"
```

退出群组时： 中止服务器 （停机）

SET GLOBAL group_replication_exit_state_action = ABORT_SERVER



dev.mysql.com/doc/refman/8.0/en/group-replication-options.html#sysvar_group_replication_exit_state_action

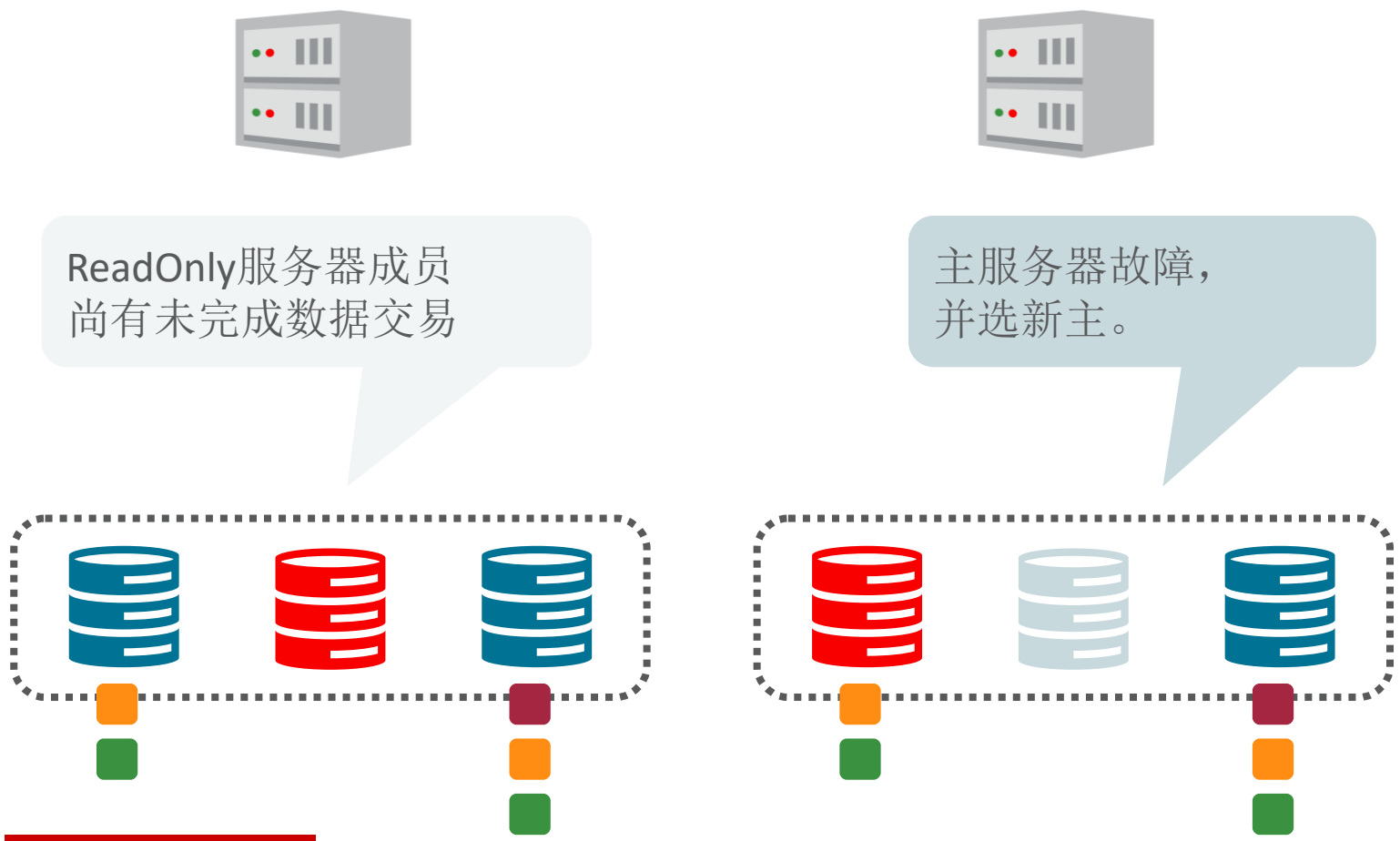
group_replication_consistency = BEFORE_ON_PRIMARY_FAILOVER

主故障转移上的数据一致性

Primary Secondary

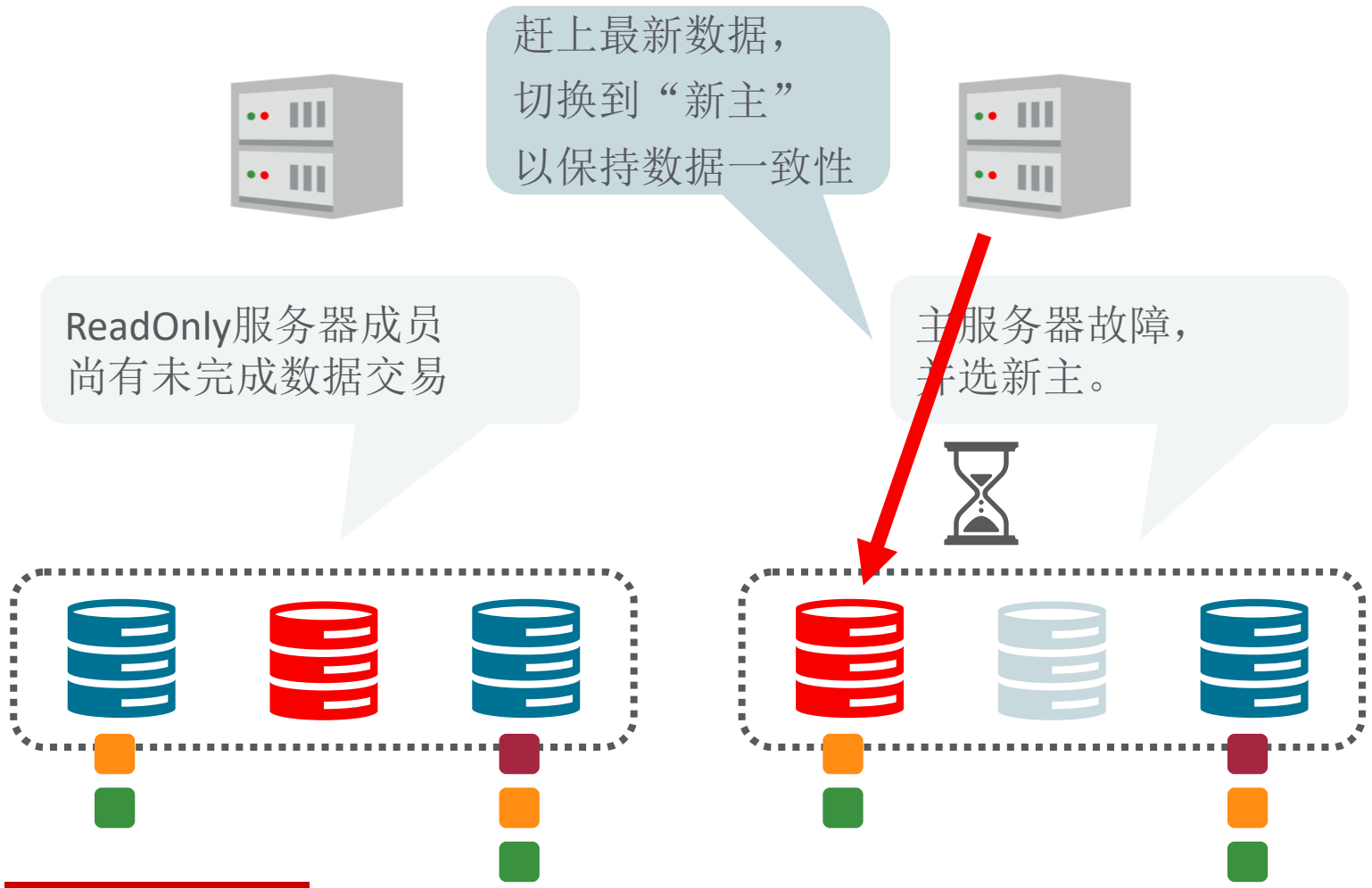
group_replication_consistency = BEFORE_ON_PRIMARY_FAILOVER

主故障转移上的数据一致性



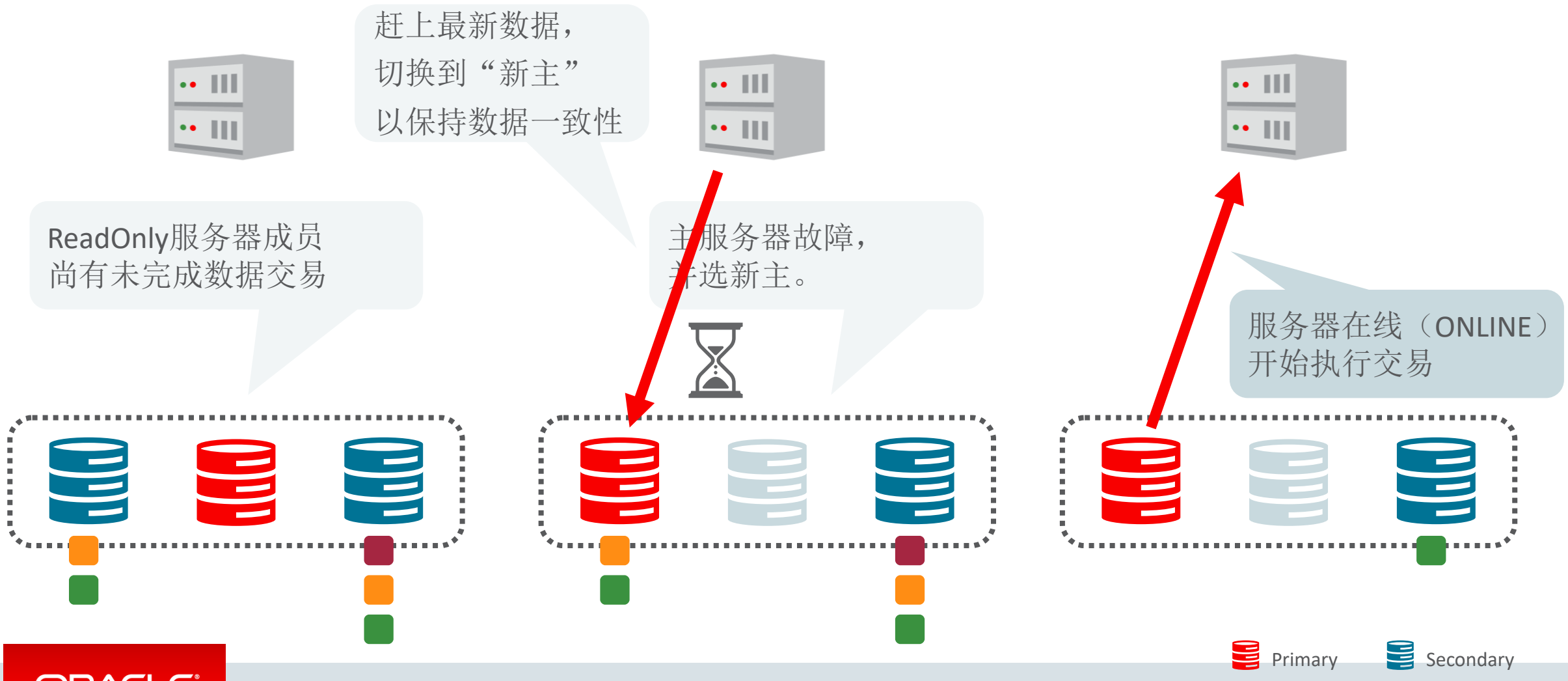
group_replication_consistency = BEFORE_ON_PRIMARY_FAILOVER

主故障转移上的数据一致性



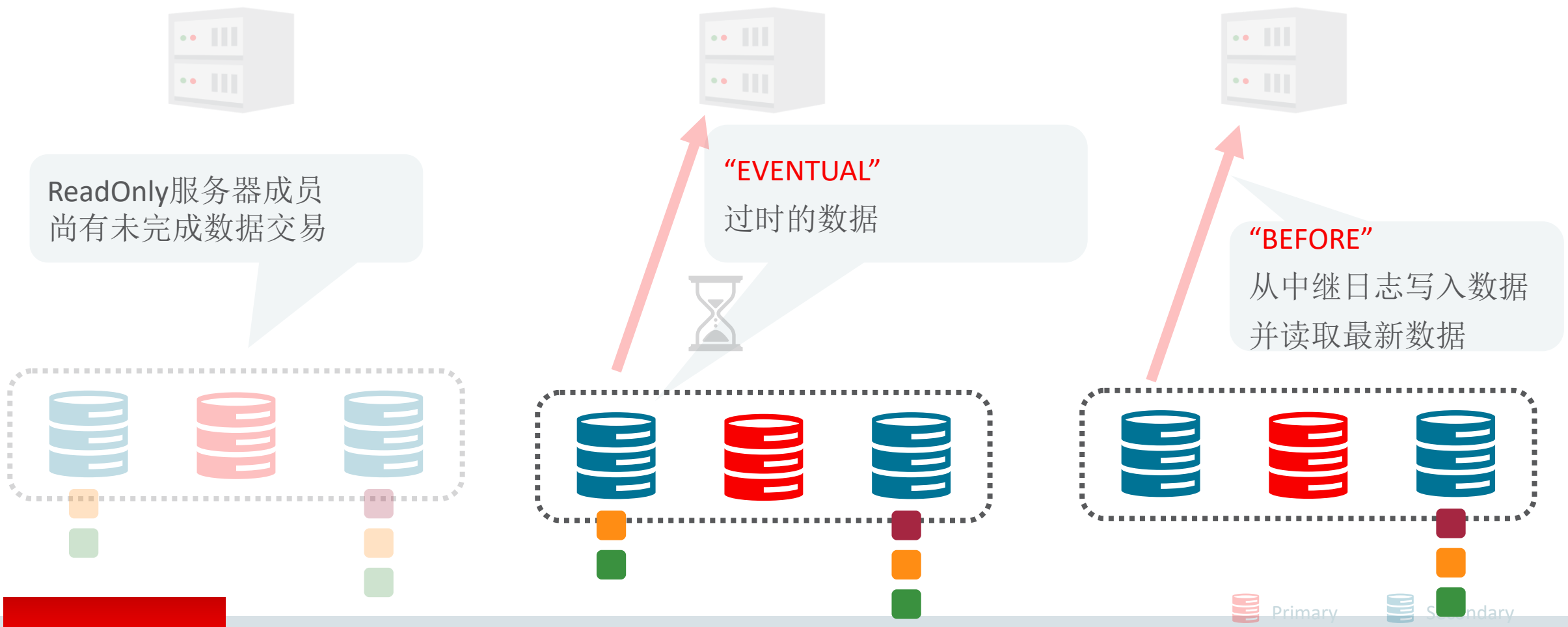
Primary Secondary

主故障转移上的数据一致性



数据一致性 : for Read

SET SESSION group_replication_consistency = BEFORE



数据一致性设置

SET SESSION group_replication_consistency = BEFORE

SET SESSION group_replication_consistency = AFTER

SET SESSION group_replication_consistency = BEFORE_ON_PRIMARY_FAILOVER

SET SESSION group_replication_consistency = BEFORE_AND_AFTER

DEFAULT : SET SESSION group_replication_consistency = EVENTUAL

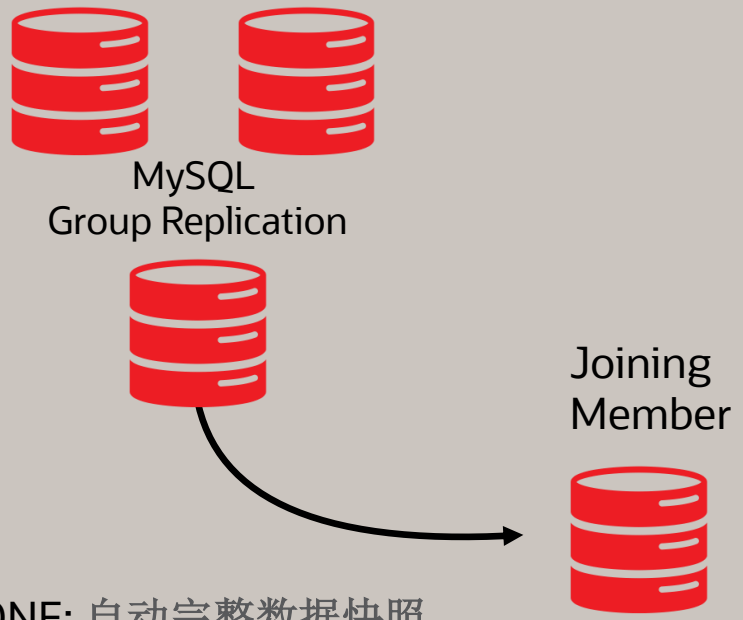
网络可靠性

- **group_replication_member_expel_timeout**
  - 默认值：5秒，如果没有响应，该服务器将被剔除！
    如果网络偶尔中断， 该怎么办？
    大事务正在运行，在”expel”情况下，才会进行选主！
- group_replication_autorejoin_tries
  - 对于发生故障的服务器（例如由于网络问题），无需人工干预即可重新加入
  - 每5分钟重试一次
- group_replication_unreachable_majority_timeout
  - 如果主服务器正在处理事务，但网络出现故障–节点将卡住 (default : 0)
  - E.g. 120 seconds : Hold up for 2 minutes

MySQL InnoDB Cluster

使用CLONE插件自动配置节点

NEW!



CLONE: 自动完整数据快照置备

```
Clone based state recovery is now in progress.

NOTE: A server restart is expected to happen as part of the clone process. If the
server does not support the RESTART command or does not come back after a
while, you may need to manually start it back.

* Waiting for clone to finish...
NOTE: 10.175.165.243:3320 is being cloned from 10.175.165.243:3310
** Stage DROP DATA: Completed
** Clone Transfer
    FILE COPY  ##################################==============   68%  In Progress
    PAGE COPY  ==================================================    0%  Not Started
    REDO COPY  ==================================================    0%  Not Started
```

关于MySQL Replication配置

MySQL Server		
slave-parallel-type	DATABASE	LOGICAL_CLOCK
slave-parallel-workers	not defined	say 2 threads or more
slave_preserve_commit_order	not defined	ON
binlog-format		ROW
binlog-checksum		NONE
gtid-mode enforce-gtid-consistency log-slave-updates		Turn ON
master-info-repository relay-log-info-repository		TABLE
transaction-write-set-extraction		XXHASH64

MySQL路由器配置示例

MySQL Router

MySQL Router		
max_connections max_connect_errors	Defaults as max_connections=512 max_connect_errors=100	Changed to Higher Value!!!
log	Default setting is INFO	Do you need initial setting of DEBUG for checking and turn it INFO for normal operation
[logger] sinks	Not specified	(e.g.) sinks=filelog,**eventlog,syslog**
use_gr_notifications (New in 8.0.17)	Default = 0	1 : Enable notification group_replication/membership/quorum_loss, group_replication/membership/view, group_replication/status/role_change, and group_replication/status/state_change.

MySQL Router

- The File System MUST not be FAT32/FAT/exFAT…

- MySQL Router will check for the privilege setting for key files which it must NOT be owned by ‘everyone’

```
PS E:\tempdata\myrouter1> & 'C:\Program Files\mysql\MySQL Router 8.0\bin\mysqlrouter.exe' -c
.\mysqlrouter.conf
PID 4260 written to 'e:/tempdata/myrouter1\mysqlrouter.pid'
Error: Invalid keyring file access rights (Everyone has full access rights).
PS E:\tempdata\myrouter1>
```

MySQL Router – Log (On Linux)

- **Logrotation via SIGHUP**
- Sending a SIGHUP signal to the router process will now close and reopen the logfile.
  - e.g.
    - # mv mysqlrouter.log mysqlrouter-`date`.log
    - # kill -SIGHUP <pid of the MySQL Router>
- It allows the integration with the logrotate to rotate and compress the Router's logfiles.

On Windows
sinks=eventlog

MySQL 8.0.18

Replication 改进

- MySQL InnoDB Cluster / GR
  - OFFLINE_MODE (ADMIN PORT)
  - TLSv1.3 Support with OpenSSL 1.1.1
- Replication with privilege Checks
  - In particularly useful for Multi-source (channel) Replication to allow restricted security applier
    - db1_channel (using priv check user db1_user)
    - db2_channel (using priv_check user db2_user)

权限检查for SLAVE Applier

Master	Slave – Channel1	Slave – Channel 2
	mysql> CREATE USER 'rpl_applier_dbuser1'@'localhost';	mysql> CREATE USER 'rpl_applier_dbuser2'@'localhost';
	mysql> GRANT REPLICATION_APPLIER,SESSION_VARIABLES_ADMIN ON *.* TO 'rpl_applier_dbuser1'@'localhost'; mysql> GRANT CREATE,INSERT,DELETE,UPDATE ON db1.* TO 'rpl_applier_dbuser1'@'localhost';	mysql> GRANT REPLICATION_APPLIER,SESSION_VARIABLES_ADMIN ON *.* TO 'rpl_applier_dbuser2'@'localhost'; mysql> GRANT CREATE,INSERT,DELETE,UPDATE ON db2.* TO 'rpl_applier_dbuser2'@'localhost';
	mysql> STOP SLAVE SQL_THREAD FOR CHANNEL ‘ch1’; mysql> CHANGE MASTER TO PRIVILEGE_CHECKS_USER = 'rpl_applier_dbuser1'@'localhost'; mysql> START SLAVE SQL_THREAD FOR CHANNEL ‘ch1’;	mysql> STOP SLAVE SQL_THREAD FOR CHANNEL ‘ch2’; mysql> CHANGE MASTER TO PRIVILEGE_CHECKS_USER = 'rpl_applier_dbuser2'@'localhost'; mysql> START SLAVE SQL_THREAD FOR CHANNEL ‘ch2’;
	mysql> SELECT Channel_name, Privilege_checks_user FROM performance_schema.replication_applier_configuration; +----------------------+---+ | Channel_name | Privilege_checks_user | +----------------------+---+ | ch1 | 'rpl_applier_dbuser1'@'localhost' | +----------------------+---+ 1 row in set (0.00 sec)	mysql> SELECT Channel_name, Privilege_checks_user FROM performance_schema.replication_applier_configuration; +----------------------+---+ | Channel_name | Privilege_checks_user | +----------------------+---+ | ch2 | 'rpl_applier_dbuser2'@'localhost' | +----------------------+---+ 1 row in set (0.00 sec)

InnoDB Cluster –选主

- Primary member election（Since 8.0.17)
  - 最低成员版本为先
  - 权重较高 （member weight)
  - server uuid的顺序

MySQL Shell 8.0.18

- **Python 3 Migration**
  - Python 2.7 – EOL by end of the year 2019
  - Minimum Support Version : 3.4.3
    - Bundled with 3.7.4
- Built-in Thread Reports
- \edit : External Editor

Ease of Use in MySQL Shell

- built-in \show command:
  - *\show threads， \show thread*

```
MySQL  localhost:3307  JS  \show threads -o tid,cid,time,state,info
+-----+-----+----------+---------------------------------+-----------------------
| tid | cid | time     | state                           | info
+-----+-----+----------+---------------------------------+-----------------------
| 108 | 65  | 00:28:00 | NULL                            | insert into test.user v
| 110 | 67  | 00:27:10 | Waiting for table metadata lock | drop table user
| 118 | 75  | 00:00:00 | executing                       | SELECT json_object('cid
+-----+-----+----------+---------------------------------+-----------------------
```

  - *External Editor*

```
Type '\help' or '\?' for help; '\quit' to exit.
MySQL  JS  \edit function sample()
```

```
function sample() {
  var text = "This function was created in an external editor";
  println(text);
}
```

MySQL 8.0.18

- SQL – Hash Join
  - For example SELECT * FROM t1 JOIN t2 ON t1.col1 = t2.col1;
    不需要任何索引来执行，并且在大多数情况下比当前的算法更有效。
- Before 8.0.18, Join : Nested-Loop Join or Block nested Loop
  - Nested-Loop : Inner table to be READ many times (e.g. Outer : 100, Inner : 100 ➔ 10,000 次)
  - Block Nested loop : Reduce the # of READ for Inner Table / Inner table to match a BLOCK of rows from Outer Table(e.g. Outer : 100, Inner : 100, BLOCK : 10 ➔ 10 x 100 ➔ 1,000 次)
  - Batched Key Access (BKA) : Similar to Block Nested Loop
- 新功能 8.0.18, for eq_ref or ref join types and no indexes
  - To build hash table for the values from outer table and read inner table to match the rows in Hash table
  - Less I/O, Read Only once in INNER table

innodb_idle_flush_pct

- *InnoDB:*
  - *Add new option to control write IOPs when idle* (WL#13115) - option innodb_idle_flush_pct
  - which controls write IOPs when InnoDB is idle.
  - The purpose is to reduce write IO for longer life of the flash storage.
- This feature is based on a contribution from Facebook, see bug#88566.

Security

- Only OpenSSL
  - *Remove support for wolfSSL and yaSSL from the MySQL codebase*
- Random Password
  - CREATE USER user IDENTIFIED BY RANDOM PASSWORD
  - ALTER USER user IDENTIFIED BY RANDOM PASSWORD
  - SET PASSWORD [FOR user] TO RANDOM.

8.0.18 变量的变化

- TDE – New Hashicorp
- Compression
- Security
- Replication
- InnoDB – Flush

https://dev.mysql.com/doc/refman/8.0/en/added-deprecated-removed.html

MySQL Enterprise Backup

- MEB 8.0.18 : MySQL Enterprise Backup通过MySQL的页面跟踪功能，支持更快的增量备份
  set --incremental=page-track.
  - https://dev.mysql.com/doc/mysql-enterprise-backup/8.0/en/backup-incremental-options.html
    - --incremental[={page-track|full-scan|optimistic}]
- Pre-requisite :
  - Base Backup while the page track is enabled
    - INSTALL COMPONENT "file://component_mysqlbackup";
    - SELECT mysqlbackup_page_track_set(true);
  - mysqlbackup needs enough memory to process all the tracked pages in memory.
    - A rough guideline is as follows: the default value of 300 [MB] for the --limit-memory option allows **mysqlbackup** to handle about 600GB of changed data.

MySQL Enterprise Backup Process

https://docs.oracle.com/cd/E17952_01/mysql-enterprise-backup-8.0-en/meb-backup-process.html

InnoDB data files
Redo logs
Relay logs except for the log fliles currently in use

Backup Lock – block DDL operations

Scan Innodb tables for DDL changes since the start of the backup an make the changes accordingly

FLUSH TABLES tbl_name, [, tbl_name] .. WITH READ LOCK

Collect logging-related data like the current innoDB LSN, binary log position, GTID, replication master or slave status …

The rest of the files are copied (redo log which are not copied, metadata files, etc…)

BACKUP lock - RELEASED

The latest info from binary or relay log file in use is coied

RELEASE the LOCKS from non-InnoDB Tables

8.0.17

- Log Archiving
  - Online Backup – MySQL Enterprise Backup
  - Start Archive Log so to catch up data during the backup
- Clone feature in MySQL 8.0.17
  - Important feature in MySQL InnoDB Cluster

MySQL 8.0.17

- Automatic Provisioning via Cloning
  - Clone is a “Database Snapshot”
  - Ease of Use, Takes away and surpasses Gallera
  - *Includes integration on distributed recovery*
    - start the group replication process in a new server and automatically clone the data from a donor with very little effort
- JSON
  - MVI – Multi-Value Indexes
    - Make it possible to index JSON arrays
  - Schema

Server Platforms

- Added support for EL8
  - Highlights
    - Added ARM64 support
    - More container / virtualization support
    - Improved security

Router

- Rest API
  - Monitoring
  - Health
  - Metadata
  - Routing
- Change notification
  - Notified about most of the cluster changes asynchronously, almost immediately

Shell

- Clone support
  - Can configure how *Cluster*.addInstance() behaves
  - Letting cloning operations proceed in the background
  - Show different levels of progress in MySQL Shell
- New – shell extensions
- Upgradechecker fixes.
- Easiest way to get started is
  - https://github.com/mzinner/mysql-shell-ex

Connectors - X Dev API

- Supports indexing array fields
- Send connection attributes
- OVERLAPS and NOT OVERLAPS operators for expressions on JSON arrays or objects

MEB

- MEB now uses mysql redo archive for backups
  - No longer issue with race condition on redo
- Now with support for encrypted redo log files

8.0.16

- mysqld – automatic upgrade
  - No more mysql_upgrade

THANK YOU

Q&A