MySQL NDB Cluster 的实践

强昌金

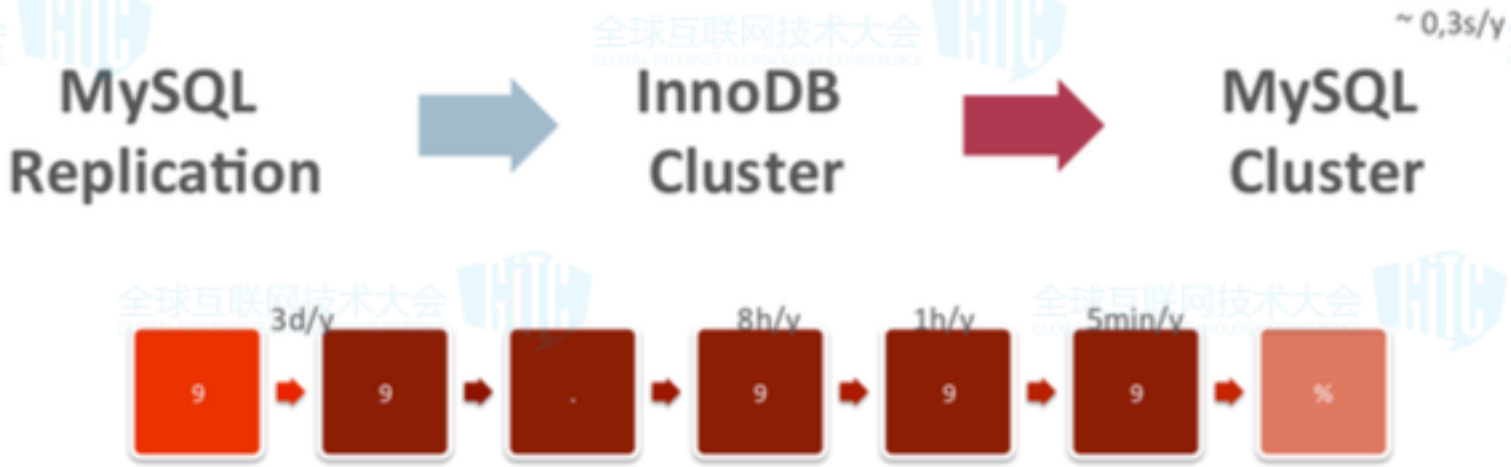
目录

关于我

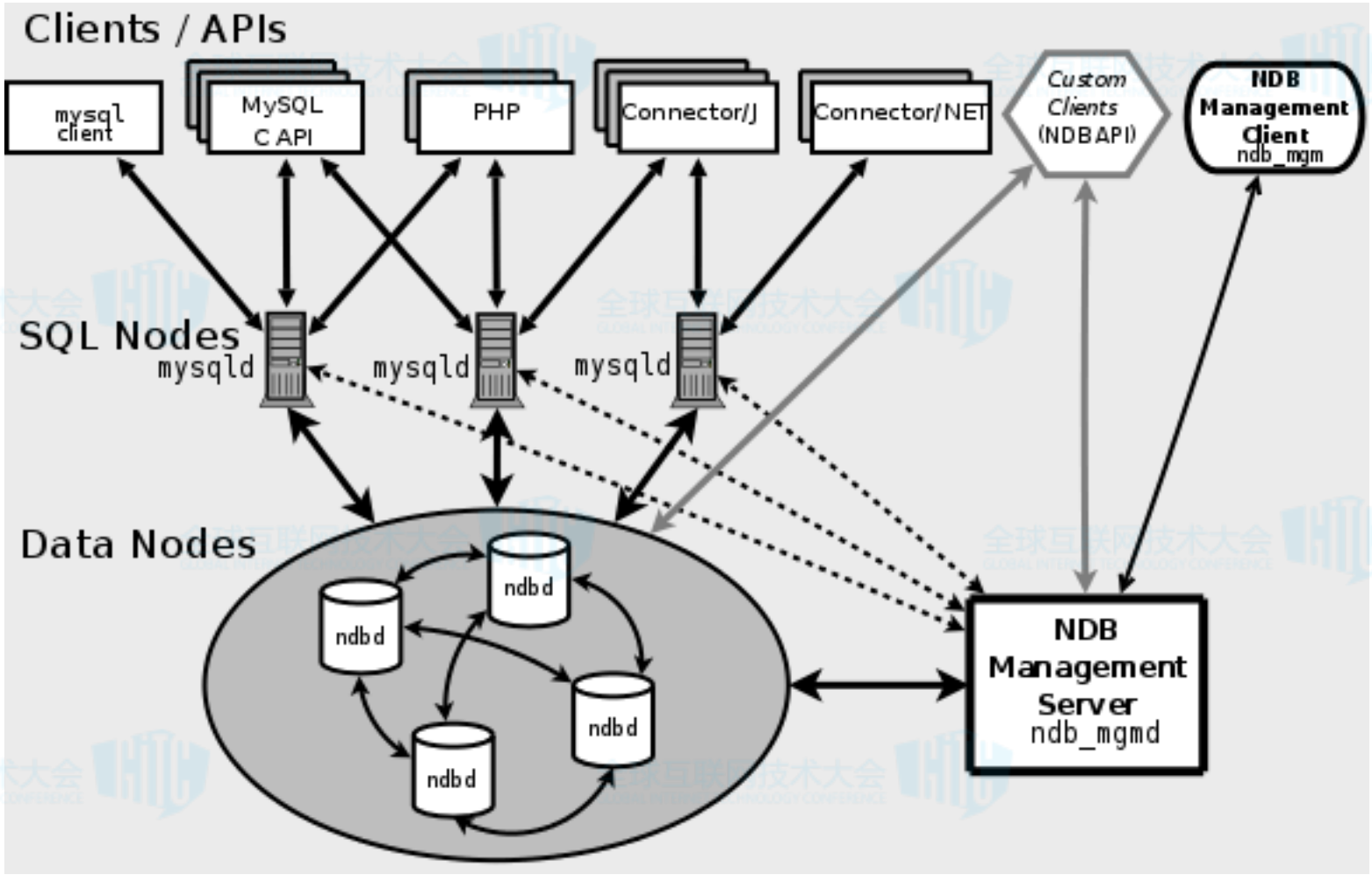
- 陌陌 DBA(2014-2015)
- 去哪儿 高级DBA(2015-2017)
- 去哪儿网数据库管理平台
- 《MySQL 运维内参》
- CRUG(Redis中国用户组)副主席
- MySQL OCP
- 自由职业:提供MySQL/Redis咨询与技术支持

MySQL HA

MySQL High-Availability & Scale-Out Solutions



MySQL NDB Cluster 架构



MySQL NDB Cluster 概述

- Network DataBase
- In-Memory
- Shared-Nothing
- Auto-Sharding
- High-Availability
- Data-Persisnce
- Read/Write consistency/Scalability

MySQL NDB Cluster— API

SQL

- Industry standard
- Joins & complex queries
- Relational model

Memcached

- simple to use API
- key/value
- driver for many languages
- ideal as e.g. PHP proxy

mod_ndb

- REST/JSON
- HTML
- using Apache

ClusterJ

- simple to use Java API
- Web & telco
- Object Relational Mapping
- native & fast access to MySQL Cluster

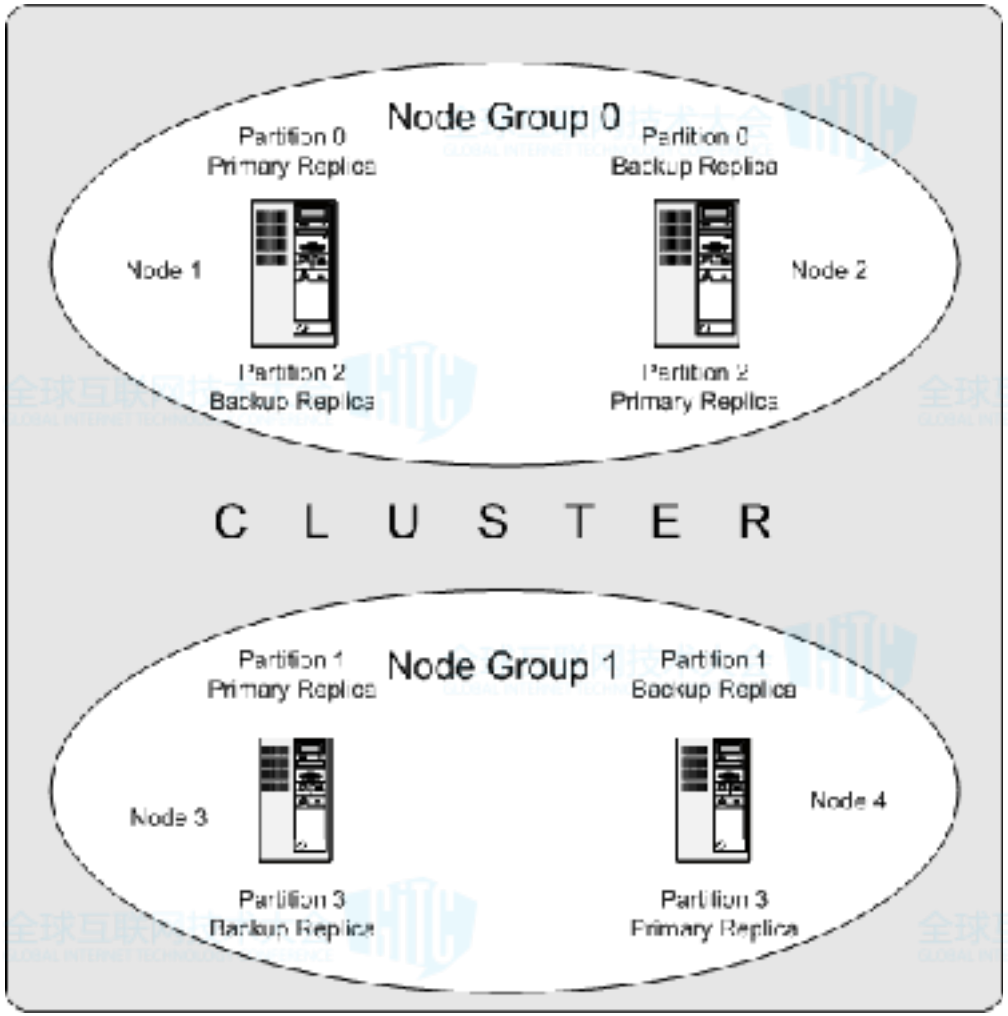
C++

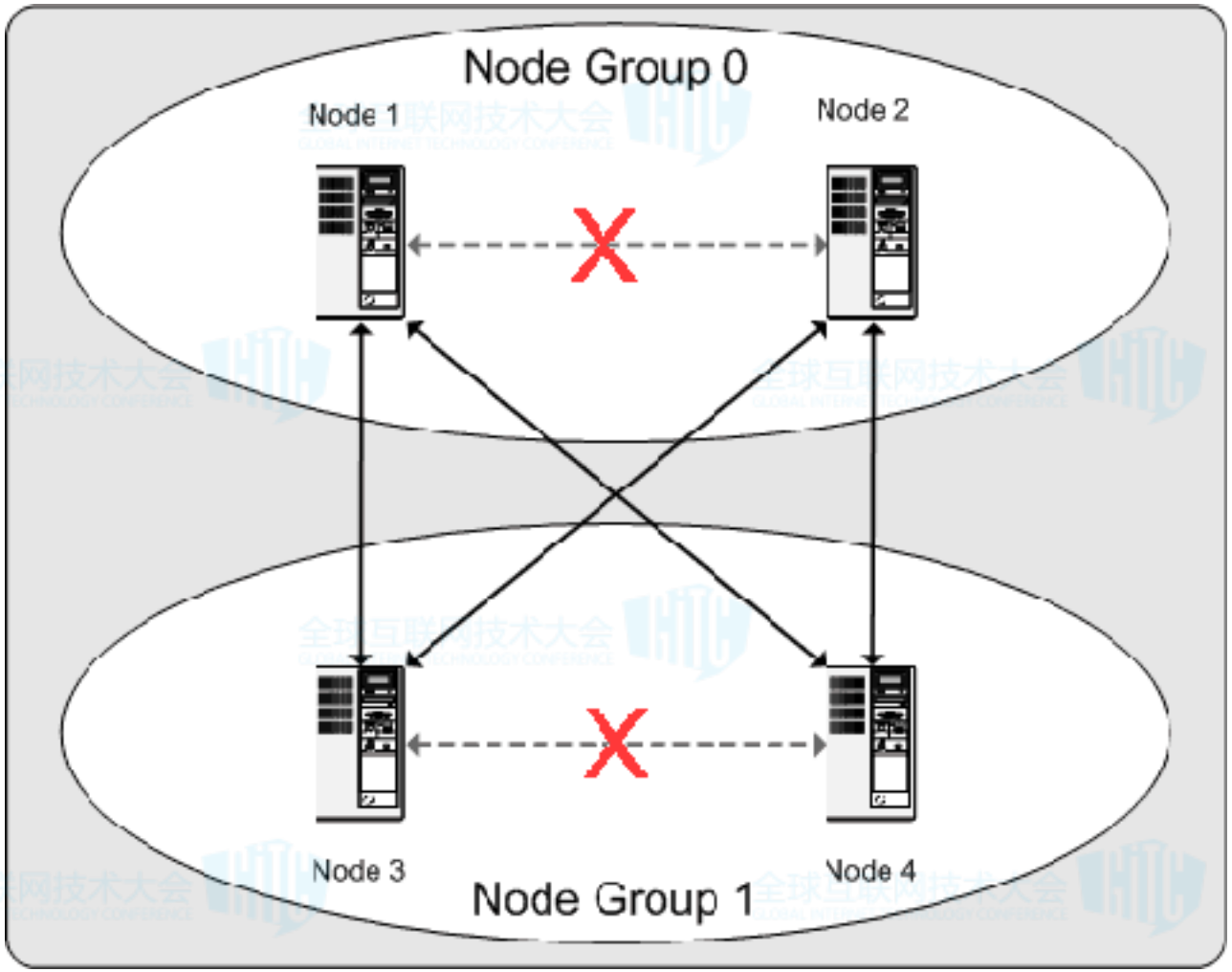
- experienced developer
- ultra low latency / real-time

MySQL NDB Cluster—Data Nodes

- Data Nodes
  - 数据存储
  - 分片
  - 自动复制
  - 扩展性

Data Nodes(Node Groups、Replicas、Partitions)





MySQL NDB Cluster—SQL Nodes

- SQL Nodes
  - SQL Access to data
  - Round-robin app access
  - Geographic Replication
  - Scale out for performance

MySQL NDB Cluster—Managerment Nodes

- Management Nodes
  - Distribute configuration data
  - Control Cluster membership
  - Run backup
  - Arbitrator in case of network partitioning

MySQL NDB Cluster—安装与配置

- SQL Nodes
  - 安装MySQL实例
  - 修改my.cnf配置文件
- Data Nodes
  - 安装Data Node软件
  - 修改my.cnf配置文件
- Management Nodes
  - 安装Management Node
  - 修改my.cnf配置

```
[mysql_cluster]
ndb-connectstring=192.168.1.2

[mysqld]
ndbcluster
ndb-connectstring=192.168.1.2
```

```
[mysqld]
ndbcluster
ndb-connectstring=192.168.1.2:1186

[mysql_cluster]
ndb-connectstring=192.168.1.2:1186
```

```
[ndbd default]
NoOfReplicas=2
DataMemory=80M
IndexMemory=18M

[ndb_mgmd default]
PortNumber=1186
DataDir=/data/mysql/multi/3307

[ndb_mgmd]
NodeId=1
hostname=192.168.1.2


[ndbd]
ServerPort=1186
HostName = 192.168.1.3
DataDir=/data/mysql/multi/3306/data

[ndbd]
ServerPort=1186
HostName = 192.168.1.4
DataDir=/data/mysql/multi/3306/data


[mysqld]
HostName=192.168.1.2

[mysqld]
HostName=192.168.1.7
```

MySQL NDB Cluster—节点启动类型

- 启动类型
  - Initial start
  - System restart
  - Node restart
  - Initial node restart

MySQL NDB Cluster—ndbinfo库

- 仲裁相关
  - arbitrator_validity_*
- 参数配置
  - config_params 与 config_values
- 节点信息
  - config_nodes、membership、nodes
- CPU统计信息
  - cpustat_*
- NDB数据字典
  - dict_obj_info 与 dict_obj_types
- 磁盘写入情况
  - disk_write_speed_*
- 命中率问题
  - diskpagebuffer
- LOG使用情况
  - logbuffers 和 logspaces
- 重启信息
  - restart_info
- NDB表分布
  - table_*

MySQL NDB Cluster—Programs

21.4.1 **ndbd** — The NDB Cluster Data Node Daemon
21.4.2 **ndbinfo_select_all** — Select From ndbinfo Tables
21.4.3 **ndbmtd** — The NDB Cluster Data Node Daemon (Multi-Threaded)
21.4.4 **ndb_mgmd** — The NDB Cluster Management Server Daemon
21.4.5 **ndb_mgm** — The NDB Cluster Management Client
21.4.6 **ndb_blob_tool** — Check and Repair BLOB and TEXT columns of NDB Cluster Tables
21.4.7 **ndb_config** — Extract NDB Cluster Configuration Information
21.4.8 **ndb_cpcd** — Automate Testing for NDB Development
21.4.9 **ndb_delete_all** — Delete All Rows from an NDB Table
21.4.10 **ndb_desc** — Describe NDB Tables
21.4.11 **ndb_drop_index** — Drop Index from an NDB Table
21.4.12 **ndb_drop_table** — Drop an NDB Table
21.4.13 **ndb_error_reporter** — NDB Error-Reporting Utility
21.4.14 **ndb_import** — Import CSV Data Into NDB
21.4.15 **ndb_index_stat** — NDB Index Statistics Utility
21.4.16 **ndb_move_data** — NDB Data Copy Utility
21.4.17 **ndb_perror** — Obtain NDB error message information
21.4.18 **ndb_print_backup_file** — Print NDB Backup File Contents
21.4.19 **ndb_print_file** — Print NDB Disk Data File Contents
21.4.20 **ndb_print_frag_file** — Print NDB Fragment List File Contents
21.4.21 **ndb_print_schema_file** — Print NDB Schema File Contents
21.4.22 **ndb_print_sys_file** — Print NDB System File Contents
21.4.23 **ndb_redo_log_reader** — Check and Print Content of Cluster Redo Log
21.4.24 **ndb_restore** — Restore an NDB Cluster Backup
21.4.25 **ndb_select_all** — Print Rows from an NDB Table
21.4.26 **ndb_select_count** — Print Row Counts for NDB Tables
21.4.27 **ndb_setup.py** — Start browser-based Auto-Installer for NDB Cluster
21.4.28 **ndb_show_tables** — Display List of NDB Tables
21.4.29 **ndb_size.pl** — NDBCLUSTER Size Requirement Estimator
21.4.30 **ndb_top** — View CPU usage information for NDB threads
21.4.31 **ndb_waiter** — Wait for NDB Cluster to Reach a Given Status

MySQL NDB Cluster—在线添加节点与节点组

- 修改(ndb_mgm)配置文件、添加新的节点信息
- 关闭所有的Management Nodes、使用reload启动，加载新配置
- 重启已有的Data Nodes
- 重启所有的SQL Nodes
- 启动新的Data Nodes(--initial)
- 创建新的Node Group
- 重新分配集群中的数据(ALTER TABLE *** ALGORITHM=INPLACE REORGANIZE PARTITION)

MySQL NDB Cluster—Backup

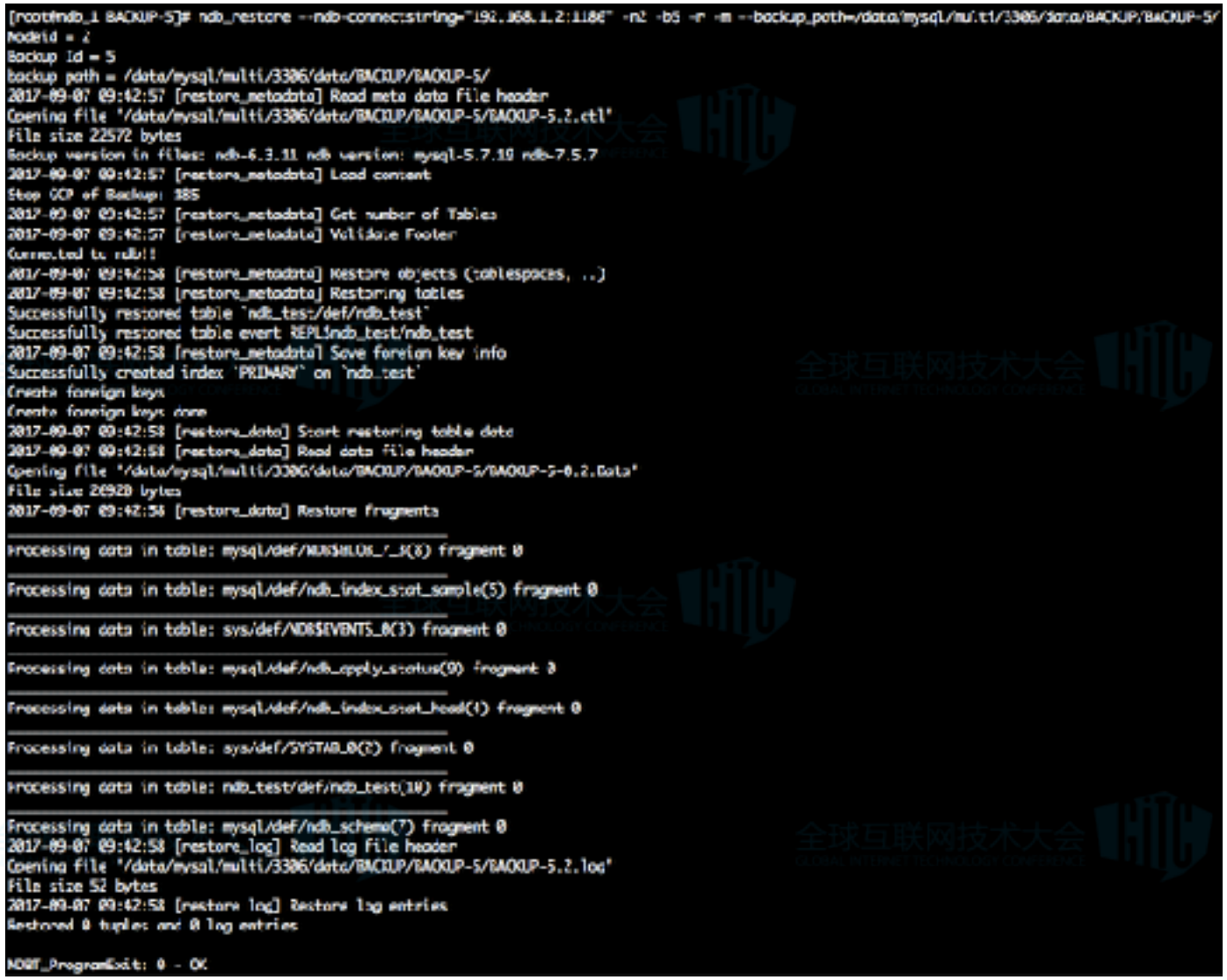
- 备份内容
  - Metadata
  - Table records
  - Transaction log

```
START BACKUP [backup_id] [wait_option] [snapshot_option]

wait_option:
WAIT {STARTED | COMPLETED} | NOWAIT

snapshot_option:
SNAPSHOTSTART | SNAPSHOTEND
```

- 备份文件
  - BACKUP-backup_id.node_id.ctl
  - BACKUP-backup_id.node_id.data
  - BACKUP-backup_id.node_id.log

MySQL NDB Cluster—Restore

- 命令：ndb_restore
- 重要参数
  - restore_meta
  - restore_data
  - backup_path
  - backup_id
  - ndb-connectstring
  - nodeid
  - include-database
  - include-tables
  - restore_epoch



MySQL NDB Cluster—Rolling Restart

- 滚动重启的原因
  - 配置修改
  - NDB Cluster的升级或者降级
  - 更改节点所在的主机
  - 系统重启或集群重启

MySQL NDB Cluster—Rolling Restart

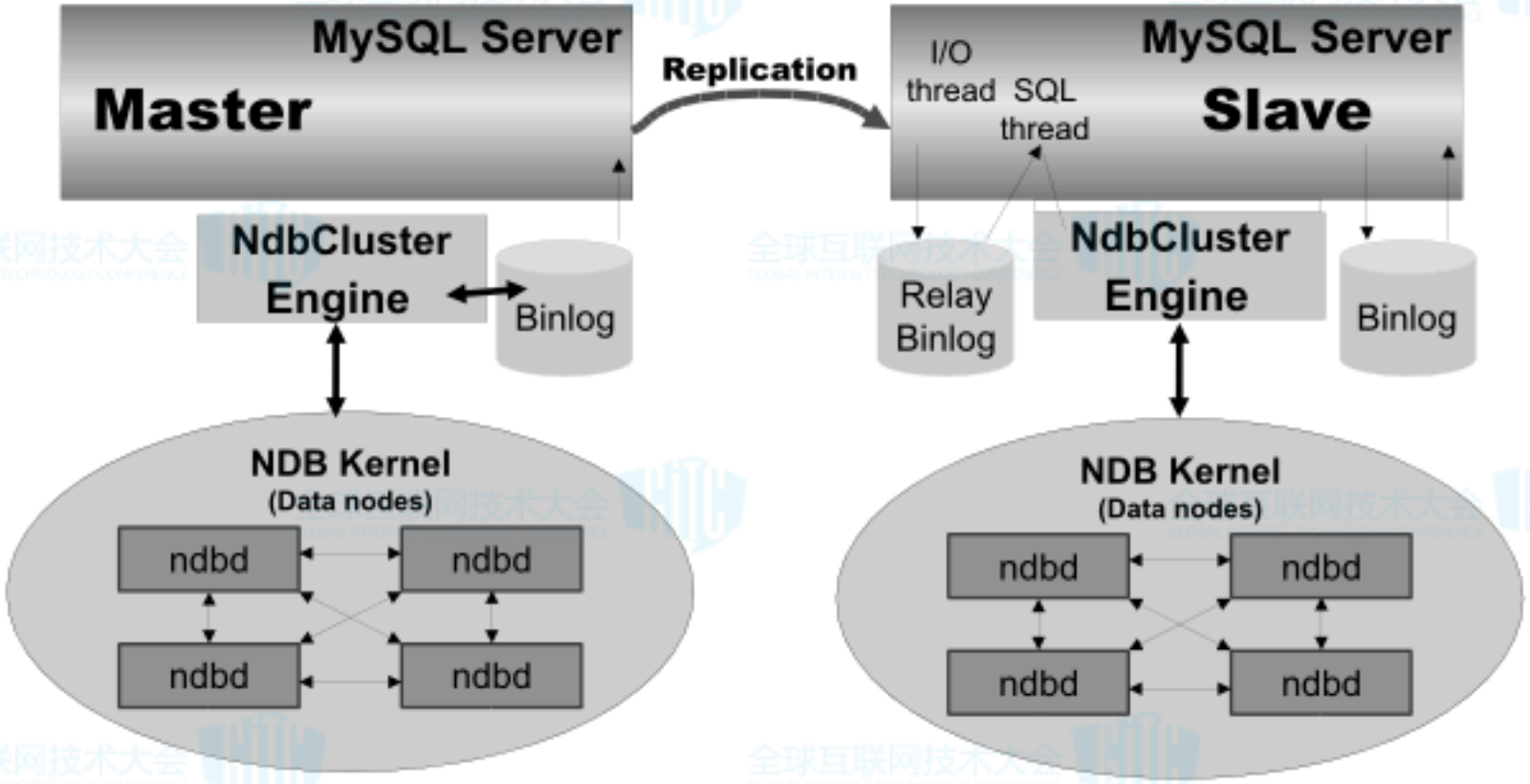
- 滚动重启的过程
  - 关闭、修改配置、重启所有的Management Node
  - 关闭、修改配置、重启所有的Data Node
  - 关闭、修改配置、重启所有的SQL Node

MySQL NDB Cluster—Rolling Restart

RESTART TYPE:				
Cluster Configuration Change	**Cluster Software Upgrade or Downgrade**	**Change on Node Host**	**Cluster Reset**	
A. Management node (ndb_mgmd) processes...				
1. Stop all ndb_mgmd processes 2. Make changes in global configuration file(s) 3. Start all ndb_mgmd processes	1. Stop all ndb_mgmd processes 2. Replace each ndb_mgmd binary with new version 3. Start ndb_mgmd processes	1. Stop all ndb_mgmd processes 2. Make desired changes in hardware, operating system, or both 3. Start all ndb_mgmd processes	(OR) 1. Stop all ndb_mgmd processes 2. Start all ndb_mgmd processes	Restart all ndb_mgmd processes (optional)

B. For each data node (ndbd) process...					
(OR)		1. Stop ndbd 2. Replace ndbd binary with new version 3. Start ndbd	1. Stop ndbd 2. Make desired changes in hardware, operating system, or both 3. Start ndbd	(OR)	
1. Stop ndbd 2. Start ndbd	Restart ndbd			1. Stop ndbd 2. Start ndbd	Restart ndbd
C. For each SQL node (mysqld) process...					
(OR)		1. Stop mysqld 2. Replace mysqld binary with new version 3. Start mysqld	1. Stop mysqld 2. Make desired changes in hardware, operating system, or both 3. Start mysqld	(OR)	
1. Stop mysqld 2. Start mysqld	Restart mysqld			1. Stop mysqld 2. Start mysqld	Restart mysqld

MySQL NDB Cluster—集群间的复制



MySQL NDB Cluster—搭建复制

- 备份数据
  - ndb_mgm(M)>START BACKUP
- 将备份数据传送到从库的所有NDB节点
- 创建数据库
- 从库上RESET SLAVE
  - mysql(S)>RESET SLAVE;
- 所有从节点的Data Node恢复数据
  - ndb_restore -c “slave_ndb_mgm:port” -n * -b * -r -m --backup_path=***
  - ndb_restore -c “slave_ndb_mgm:port” -n * -b * -r -e --backup_path=***
- 找点
- 开启复制

MySQL NDB Cluster—相关表

- ndb_binlog_index

```
mysql> SHOW CREATE TABLE `ndb_binlog_index` \G
*************************** 1. row ***************************
       Table: ndb_binlog_index
Create Table: CREATE TABLE `ndb_binlog_index` (
  `Position` bigint(20) unsigned NOT NULL,
  `File` varchar(255) NOT NULL,
  `epoch` bigint(20) unsigned NOT NULL,
  `inserts` int(10) unsigned NOT NULL,
  `updates` int(10) unsigned NOT NULL,
  `deletes` int(10) unsigned NOT NULL,
  `schemaops` int(10) unsigned NOT NULL,
  `orig_server_id` int(10) unsigned NOT NULL,
  `orig_epoch` bigint(20) unsigned NOT NULL,
  `gci` int(10) unsigned NOT NULL,
  `next_position` bigint(20) unsigned NOT NULL,
  `next_file` varchar(255) NOT NULL,
  PRIMARY KEY (`epoch`,`orig_server_id`,`orig_epoch`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 STATS_PERSISTENT=0
```

- ndb_apply_status

```
mysql> SHOW CREATE TABLE `ndb_apply_status` \G
*************************** 1. row ***************************
       Table: ndb_apply_status
Create Table: CREATE TABLE `ndb_apply_status` (
  `server_id` int(10) unsigned NOT NULL,
  `epoch` bigint(20) unsigned NOT NULL,
  `log_name` varchar(255) CHARACTER SET latin1 COLLATE
  `start_pos` bigint(20) unsigned NOT NULL,
  `end_pos` bigint(20) unsigned NOT NULL,
  PRIMARY KEY (`server_id`) USING HASH
) ENGINE=ndbcluster DEFAULT CHARSET=latin1
```

MySQL NDB Cluster—复制必要条件

- 复制需求
  - 基于行复制(binlog-format=[ROW|MIXED])
  - server-id唯一
  - 版本兼容
  - API使用问题

MySQL NDB Cluster—集群间复制的限制

- 主从连接断开、需要人工介入
  - 主SQL Node与SQL Node断开
  - 主SQL Node与主集群中的Data Node断开
  - 主SQL Node无响应，被集群丢弃
- 不支持GTID
- 不支持多线程复制

MySQL NDB Cluster—Single Replication Channel & Two Replication Channels

- Single Replication Channel
  - 两个节点(M-S)
- Two Replication Channels
  - 4个节点(M、M’、S、 S’)
  - 建立一对主从关系(M-S)

MySQL NDB Cluster — Failover with NDB Cluster Replication

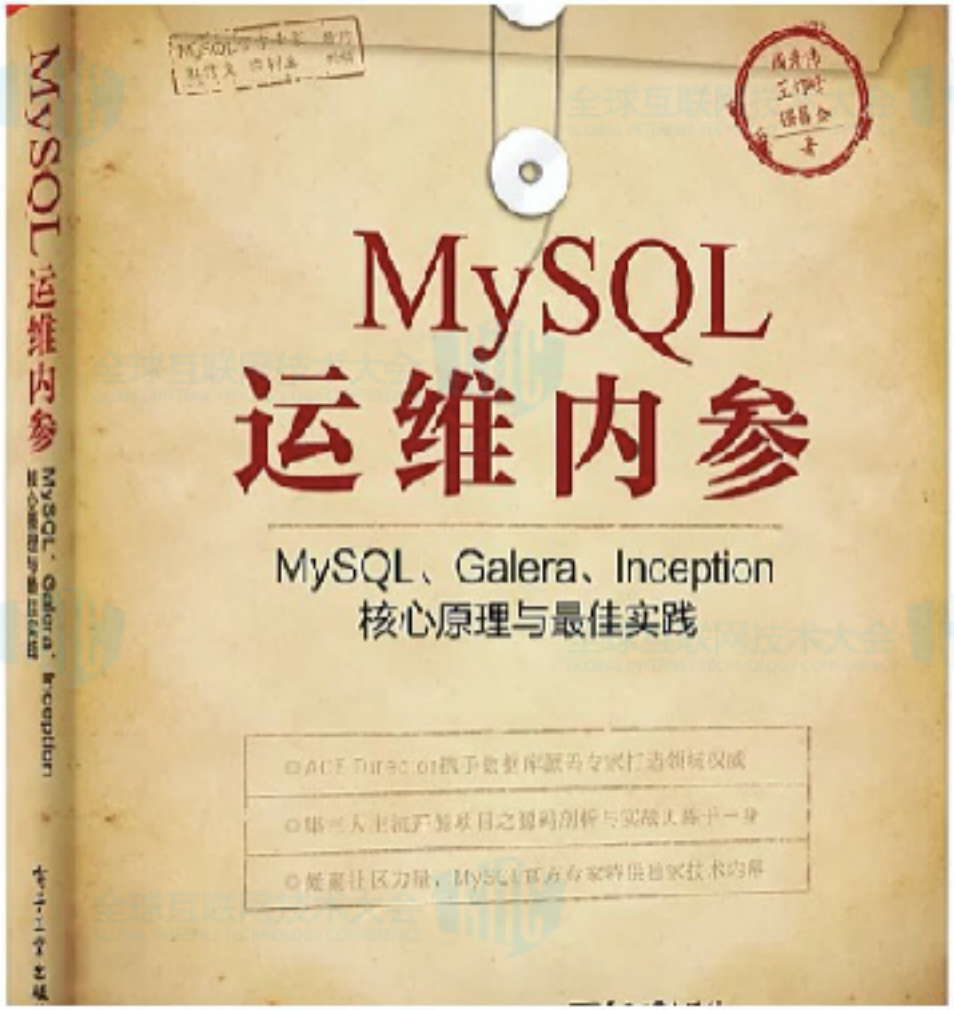
- 获取最新的GCP(S)
  - SELECT max(epoch) as latest_gcp FROM mysql.ndb_apply_status;
- 获取切换的binlog的位置(M’)
  - SELECT SUBSTRING_INDEX(next_file,'/',-1) as bin_file , next_position as position FROM mysql.ndb_binlog_index WHERE epoch = latest_gcp ORDER BY epoch ASC LIMIT 1;
- 切换主从关系(S’)
  - CHANGE MASTER TO MASTER_LOG_FILE=‘bin_file’, MASTER_LOG_POS=position

MySQL NDB Cluster—限制

21.1.6.1 Noncompliance with SQL Syntax in NDB Cluster
21.1.6.2 Limits and Differences of NDB Cluster from Standard MySQL Limits
21.1.6.3 Limits Relating to Transaction Handling in NDB Cluster
21.1.6.4 NDB Cluster Error Handling
21.1.6.5 Limits Associated with Database Objects in NDB Cluster
21.1.6.6 Unsupported or Missing Features in NDB Cluster
21.1.6.7 Limitations Relating to Performance in NDB Cluster
21.1.6.8 Issues Exclusive to NDB Cluster
21.1.6.9 Limitations Relating to NDB Cluster Disk Data Storage
21.1.6.10 Limitations Relating to Multiple NDB Cluster Nodes

广宣篇

- 内容简介
  - MySQL
  - Inception
  - Galera Cluster
  - MySQL Group Replication
  - InnoDB 内核
  - 运维经验总结



长按或扫描二维码
查看详情

微店

THANKS!