Why use MySQL

by royalwzy

个人介绍

王朝阳(Oracle ACE)

＊数据库技术爱好者：获有Oracle 10g/11g OCM, Oracle 10g/11g/12c OCP, MySQL 5.6 Database Administrator, MCITP/MCTS DBA(MSSQL2008), RHCE, Java Programmer等认证。

＊数据库技术使用者：+5年数据库使用及管理经验，精通异构数据库(Oracle/MySQL/MSSQL/NoSQLs)和异构操作系统(Linux/Unix/Windows)之间的数据同步；擅长数据库的设计和调优；通过大数据技术进行用户行为性分析和智能推荐；部署MySQL的集群和高可用环境，并实现Oracle到MySQL的数据同步和迁移。

＊数据库技术研究者：独立开发mysqlexp工具(用于从MySQL数据库自定义导出数据)、mysqlclone工具(用于克隆MySQL数据库)和sqluldr工具(用于从Oracle数据库导出文本数据)，并进行后续的更新和完善。

＊数据库技术分享者：作为Oracle ACEA、Oracle Young Expert、ACOUG/SHOUG/OCMU核心成员、社区专家，经常参与技术类交流活动；担任云和恩墨高级讲师、尚观科技Oracle讲师进行专业的培训工作，向更多数据库爱好者传播数据库知识；活跃于OTN和ITPUB等论坛，并长期维护和更新自己的数据库技术交流博客(royalwzy.com)。

主要内容

- MySQL介绍
- MySQL分布式数据库应用
- MySQL高可用技术介绍
- MySQL相关工具介绍

MySQL介绍

- MySQL历史背影介绍。
- MySQL的发版机制。
- MySQL对甲骨文的意义。
- MySQL vs Oracle。
- 如何成为一名优秀的MySQL DBA人员。

MySQL的历史

• 1996.05：MySQL1.0发布。
• 1996.10：MySQL3.11.1发布。
• 1998.01：瑞典的MySQL AB公司开发了BDB(Berkeley DB)引擎，因为它支持事务，所以Mysql也开始支持事务了。至此才是我们一直认为的MySQL数据库。
• 2000.04：MySQL对旧的存储引擎进行整理，命名为MyISAM。
• 2001：MySQL4.0发布。Heikiki Tuuri向MySQL提出建议，希望能集成他们的存储引擎InnoDB，这个引擎同样支持事务处理，还支持行级锁。
• 2003.12：MySQL5.0发布，也是用的比较多对5.x版本，增加了视图和存储过程等特性。
• 2008.08：MySQL5.1发布。
• 2008.01.16：Sun收购MySQL。
• 2009.04.20：Oracle以每股9.50美元，74亿美元的总额收购Sun。
• 2010.04.22：MySQL5.5发布，也是变化比较大的一次版本升级。半同步复制等特性。
• 2013.02.05：MySQL5.6.10 GA发布。(最近的稳定版本是5.6.21(2014.09.23)。预览版本是5.7.4(2014-03-31)。)
• 2013.06.18：Oracle修改MySQL授权协议，移除了GPL许可证。

MySQL的发版机制

- GPL(General Public License)
  1.MySQL Community Edition
  2.MySQL Cluster Community Edition
  3.MariaDB/Percona Server
- Commercial
  1.MySQL Classic Edition
  2.MySQL Standard Edition
  3.MySQL Enterprise Edition
  4.MySQL Cluster CGE
  5.MySQL Embedded (OEM/ISV)

MySQL对Oracle的意义?

• MySQL 代表了Oracle 所提供的同类产品中最出色的,面向基于Web 的应用程序的数据 库解决方案,它也是嵌入式数据库的不错选择。

• 因此,MySQL 使Oracle 的产品更为完整,是对Oracle DB 的有力补充。

• Oracle 大力投资MySQL 的原因是为了提供可驱动下一代 Web,移动和嵌入式应用程序的MySQL 解决方案。

MySQL vs Oracle

- 功能方面：

1.Oracle：无论 OLAP 还是 OLTP，无论是锁定机制还是事务支持，无论是内置函数还是外部可扩展功能，都非常强大。

2.Mysql：更擅长OLTP业务，但由于不支持Hash Join/Sort-Merge Join，以及分析函数相对较少，所以OLAP方面功能相对欠缺，但支持基本的事务以及锁定机制。

- 性能－写入：

1.Oracle：需要记录 Redo Log 且保证每次事务都fsync到物理磁盘以保证事务安全，连续写；数据的写入大多是在内存中完成，后台进程进行内存到磁盘的定期批量刷新，随机写为主。

2.Mysql：InnoDB引擎(Oracle拥有)与Oracle类似；MyISAM 引擎无事务所以没有事务日志到磁盘的fsync问题，但由于其表锁的原因，并发比较弱。但是总体看来和 Oracle 相差不大。

MySQL vs Oracle

- 性能－简单查询：

1.Oracle：在高并发场景下，由于其在事务控制实现方面的优势，以及多进程的机制，显示出了相对明显的优势。

2.Mysql：在并发数不是太高的前提下（50以下），相对于 Oracle 没有显示出弱势，甚至部分存储引擎下还有一些优势，但是随着并发数的增加，就逐步体现出了与 Oracle 的差距，这与其基于线程的机制也有一定关系。

- 性能－复杂查询：

1.Oracle：具有丰富的统计信息，CBO优化器相对于 Mysql 来说也先进很多，加上对 Hash Join 的全面支持及丰富的分析函数，无论是从功能还是性能角度来说，多表 Join 都是 Oracle 的优势所在。

2.Mysql：其查询优化器所能参考的统计信息相对较少，优化器深度也比 Oracle 少很多，所以在多表 Join 的时候出现执行计划异常并不少见。此外，不支持 Hash Join 的天生缺陷也让其 Join 能力大打折扣。

MySQL vs Oracle

- 扩展能力：

1.Oracle：由于其极高的一致性要求，造成架构上的不少限制，导致其扩展成本相对高很多。

2.Mysql：原生分布式架构的优势在于并发支持，但延时问题一直被广为诟病。所以大部分场景下是人肉进行分布式拆分，但其Replication特性加上对一致性的约束相较Oracle弱，使其架构灵活性很高。

- 可维护性：

1.Oracle：这一点上 Oracle 具有非常大的优势，无论是性能调优、监控、备份，数据存储、安全、集群维护方面，Oracle都提供了非常丰富的工具和解决方案。

2.Mysql：原生工具相对简单，但由于其开源的特性，有一些第三方开发的工具支持，但总提仍然比 Oracle 少了很多。

MySQL vs Oracle

- 安全性：

1.Oracle：严格的数据隔离级别实现，强大的审计功能，支持用户组合角色，严格的身份验证机制。

2.Mysql：只有InnoDB引擎支持简单的事务，没有审计功能，不支持角色，没有权限回收功能，内置的身份验证机制，没有基于回滚的恢复功能，不支持在线的表修改操作（比如：类似于ALTER TABLE或CREATE TABLE一类的操作都是非事务性的.它们会提交未提交的事务，并且不能回滚也不能做灾难恢复.Schame被保存在文件系统上，这一点与它使用的存储引擎无关。）

- 商业支持：

1.Oracle：商业软件，全套的商业服务支持。完善的文档手册支持，几乎涵盖所有功能。

2.Mysql：有商业服务支持，但文档过于简单，包含的内容也比较少。

- 软件成本：

1.Oracle：License费用比较高，企业版55W/2Core，之后每年都需要付合同价22%的服务费。

2.Mysql：有开源也有商业版本，但价格相对非常便宜。

那么问题来了！

- Why Use MySQL!!!

如何成为一名优秀的MySQL DBA人员?

- 如何成为一名DBA?
- 如何成为一名Mysql DBA?
- 如何成为一名优秀的DBA?
- 如何成为一名优秀的MySQL DBA?
- 考证是否真的有用?

MySQL分布式数据库应用

- What?
- Why?
- How?

What?

- 本质就是数据切分。

Why?

- 解决数据库容量问题。
- 解决单台机器性能压力的最优选择。
- 减小了宕机的概率（高可用为:90%，1-(1-90%)^n）。
- 极大提高访问速度和并发量（特别是锁）。
- 任何机制的引入带来的优点的同时会带来很多的缺点，我们只使用优点即可。

How?

- 水平切分
  1.范围
  2.哈希
  3.映射

- 垂直切分
  1.热数据
  2.冷数据

注意事项!

- 定义数据路由规则（DRR）。
- 添加分布式数据访问层（DDAL）。
- 通过反范式化避免关联。
- 避免跨库事务，可以通过消息队列达到事务最终一致性。
- 引入集群和负载均衡概念。

MySQL高可用技术介绍

- MySQL Replication
- MySQL Clustering
- MySQL Proxy
- DRBD
- Galera
- Percona XtraDB Cluster

从哪些方面考虑HA

- 什么是SLA?

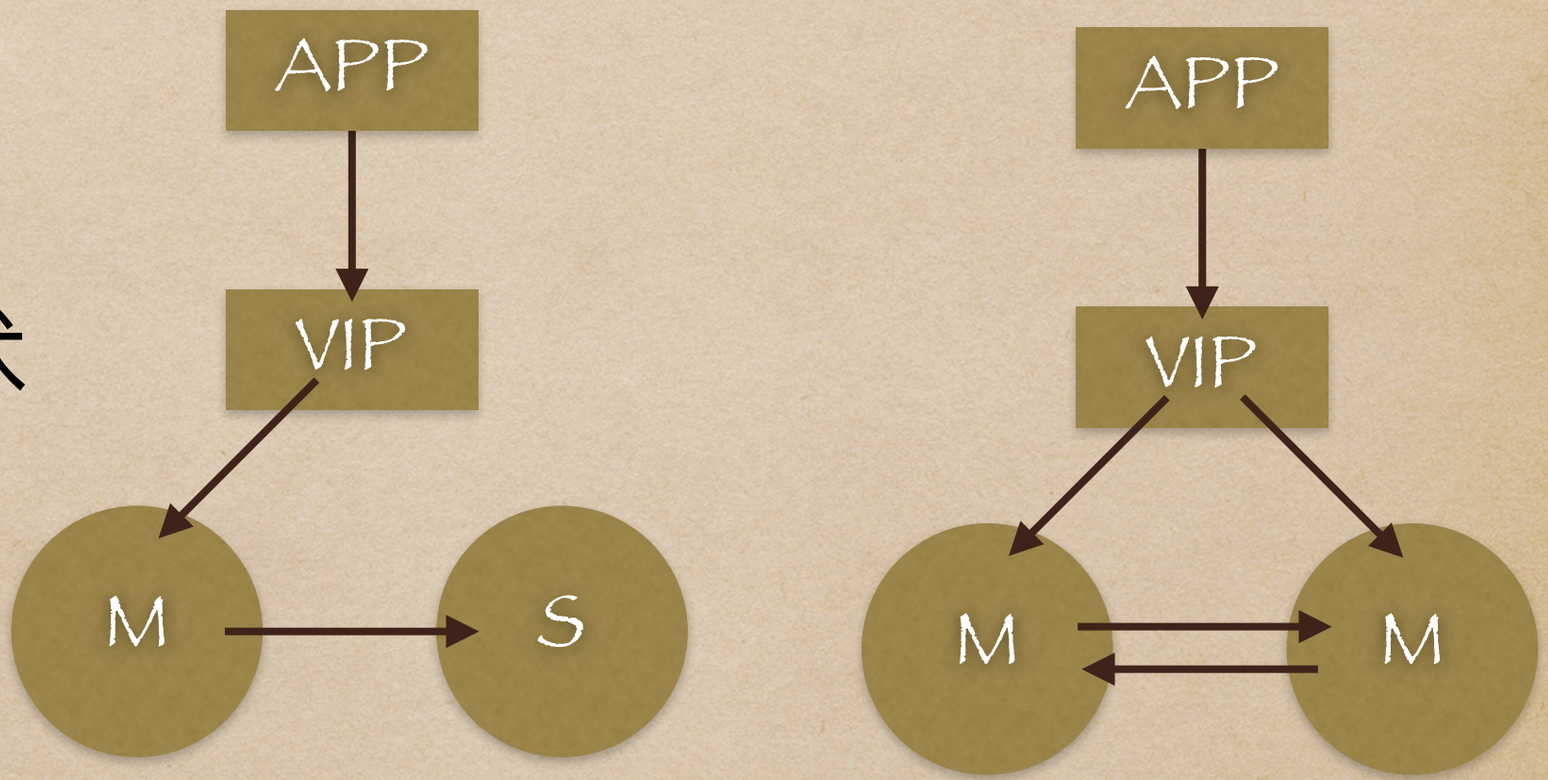
99.9999%	<32s
99.999%	<5.3mins
99.99%	<53mins
99.9%	<8.7hours
99%	<87.6hours

- 公司一年的利润是多少?
- 为了提高SLA等级需要投入多少?

MySQL HA-MySQL Replication

- MySQL复制的架构设计
  1.M-S
  2.M-M

- 常用的技术
  1.MMM
  2.MHA



MySQL HA-MySQL Replication

- 优点

  1.配置比较简单。

  2.可以有多台从库，实现读写分离。

- 缺点

  1.M-S架构还是单节点。

  2.M-M架构的话需要解决数据冲突和修复的问题。

  3.Failover需要依赖MHA/MMM脚本。

  4.投票选举机制太复杂。

MySQL HA-MySQL Replication

- WTF

Multi-Master Replication Manager for MySQL

NOTE: By now there are a some good alternatives to MySQL-MMM. Maybe you want to check out Galera Cluster which is part of MariaDB Galera Cluster and Percona XtraDB Cluster.

MMM (**M**ulti-**M**aster Replication **M**anager for MySQL) is a set of flexible scripts to perform monitoring/failover and management of MySQL master-master replication configurations (with only one node writable at any time).

The toolset also has the ability to read balance standard master/slave configurations with any number of slaves, so you can use it to move virtual IP addresses around a group of servers depending on whether they are behind in replication.

The current version of this software is stable, but the authors would appreciate any comments, suggestions, bug reports about this version to make it even better. Current version 2.0 development is led by Pascal Hofmann. If you require support, advice or assistance with deployment, please contact Percona or Open Query.

MySQL HA-MySQL Clustering

- MySQL Clusing的工作原理

1.基于存储引擎方式的高可靠性方案，是事务安全的，实时复制数据，可用于需要高可靠性及负载均衡的场合。

2.由N个节点构成，每个节点均运行着多种进程（包括MySQL服务器，NDB Cluster的数据节点，管理服务器）。

- 常用技术

MySQL + NDB-Cluster Engine

MySQL HA-MySQL Clustering

- 优点

1.Share Nothing。每个组件有自己的内存和磁盘，不存在单点故障。

2.可用于负载均衡 / 高可靠性场景。

3.高伸缩性。

- 缺点

1.随着数据库的变大，对内存的需求变得更大，因此成本很高。

2.维护成本高。

3.复杂查询性能受限。

MySQL HA-MySQL Proxy

- MySQL Proxy的工作原理

1.是一个处于客户端和MySQL服务端之间的简单程序，用于监测、分析或改变它们的通信。

2.相当于一个中间层代理，负责将前台应用的连接请求转发给后台的数据库，并且通过使用lua脚本，可以实现复杂的连接控制和过滤，从而实现读写分离和负载平衡。

3.对于应用来说MySQL Proxy是完全透明的。

MySQL HA-MySQL Proxy

- 优点

1.用于负载均衡、故障转移、查询分析、查询过滤和数据修改等等

2.读写分离：主数据库处理事务性查询，让从库处理SELECT查询。数据库复制被用来把事务性查询导致的变更同步到集群中的从库。

- 缺点

1.官方方版本性能低，多年不更新。

2.Proxy本身身是瓶颈。

3.需要使用Lua语言。

MySQL HA-DRBD

- DRBD的工作原理

  1.Distributed Replicated Block Device。

  2.是一种块设备，类似于网络RAID1的功能。

  3.当数据写入本地文件系统时，数据还将会被发送到网络中另一台主机上以相同的形式记录在一个文件系统中。

  4.本地(主节点)与远程主机(备节点)的数据可以保证实时同步，当本地系统出现故障时，远程主机上还会保留有一份相同的数据，从而达到HA的目的。

- 常用技术

  MySQL + DRDB + Heartbeat

MySQL HA-DRBD

- 优点

  1.对数据文件进行保护。

  2.没有集群软件通病，如脑裂（不要自动启动）。

- 缺点

  1.没有负载均衡。

  2.浪费资源：备机目前还不能提供读。

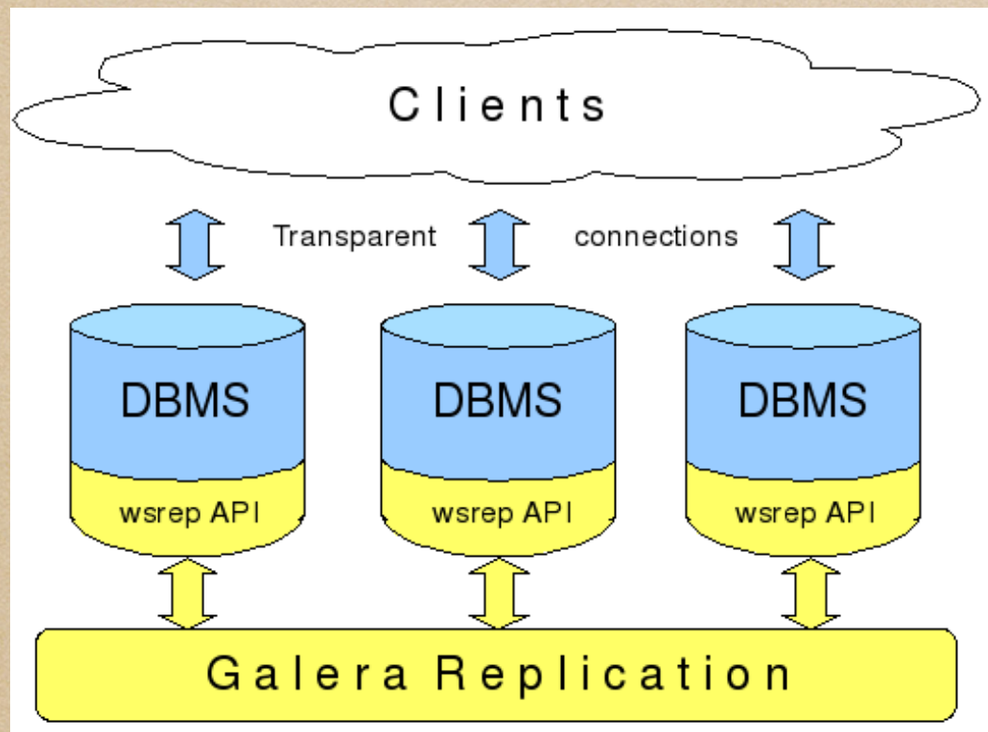
  3.恢复成本高：无法代替备份，可能会造成坏数据块。

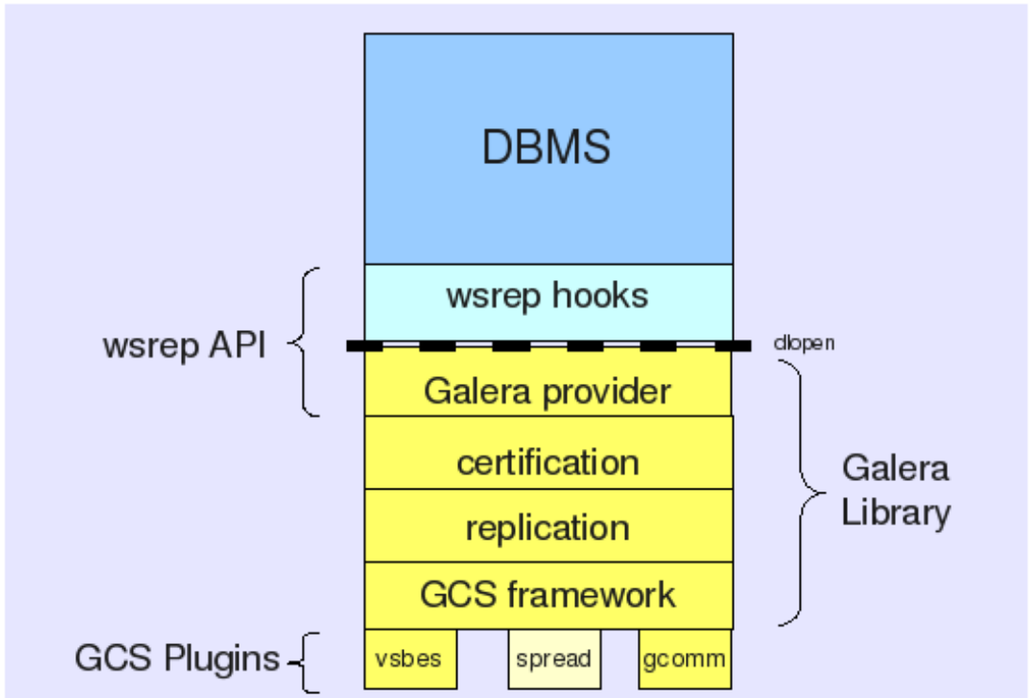
  4.故障转移时间成本高：无法做到秒级以内。

  5.对于MyISAM表用处不大：DRDB主要是针对InnoDB引擎使用。

  6.增加写操作负担：主要是InnoDB设置为innodb_flush_log_at_trx_commit=1导致许多小的写入和fsync()调用，则DRBD同步将会比较缓慢。

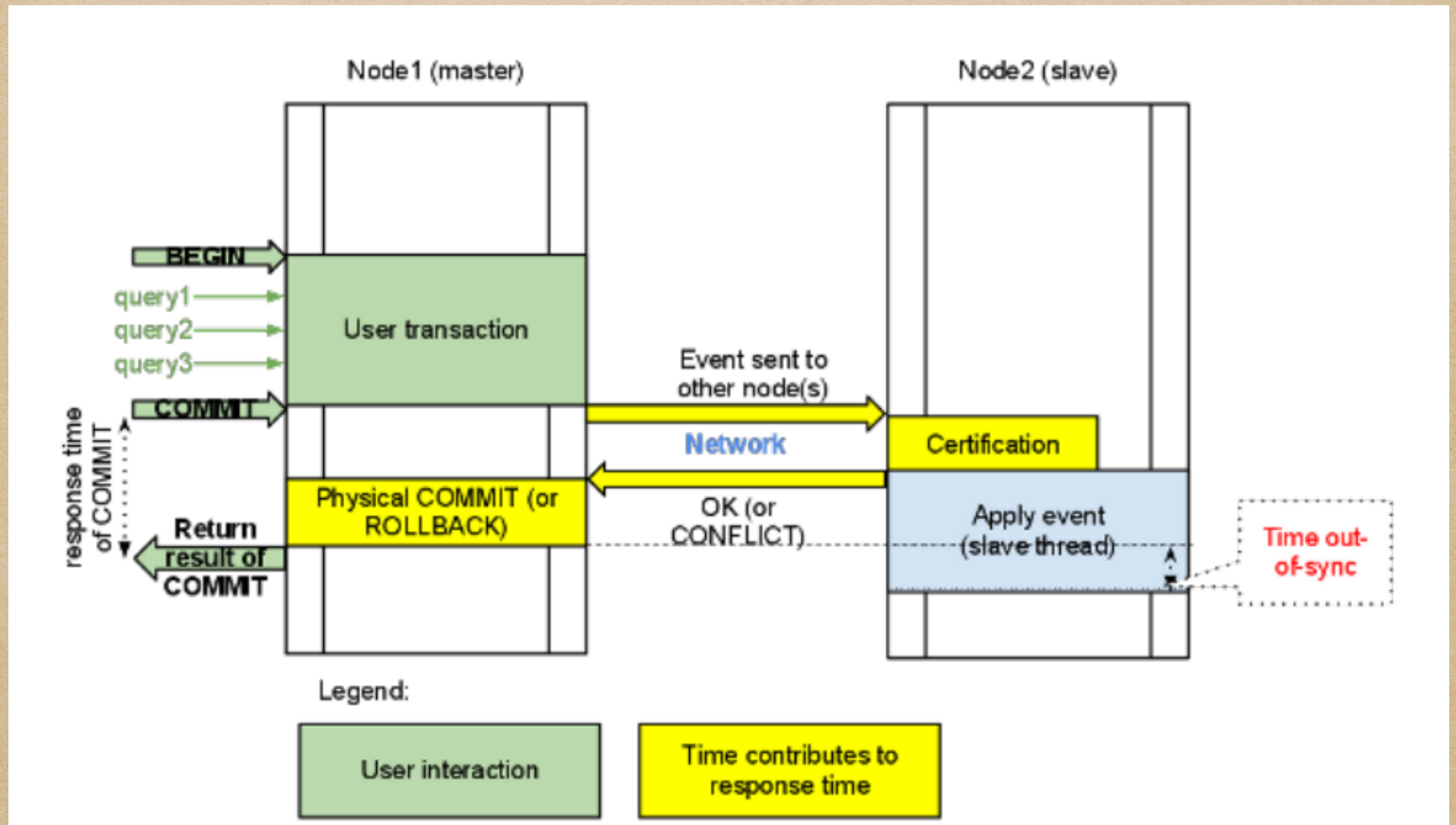
MySQL HA-Galera-复制架构



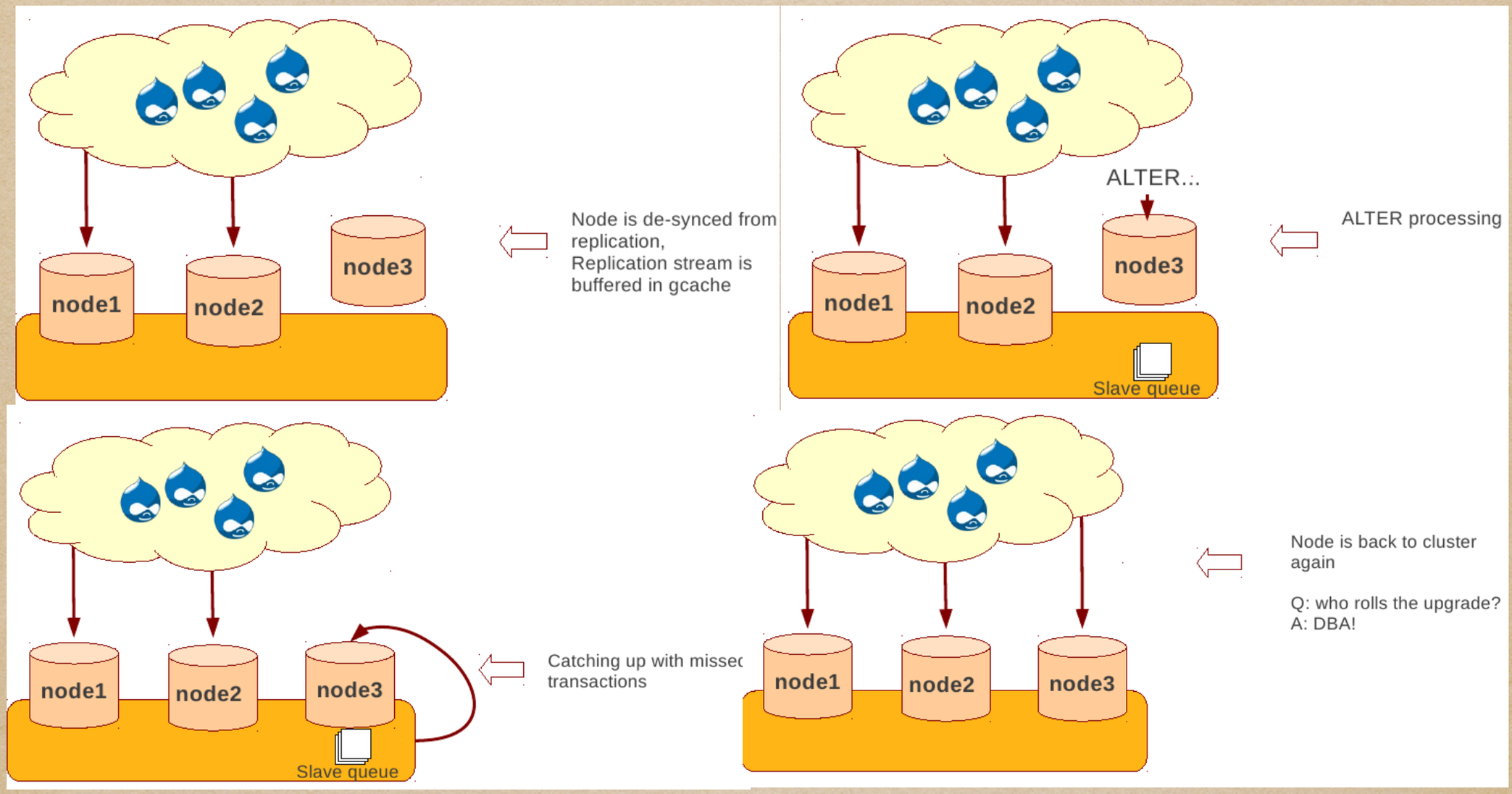


- 完成的事务在集群内的广播。
- 应用从其它节点接收并全局验证通过的事件到本地。
- 集群内通信，节点存活的检测。
- 多点写入时的锁冲突检测机制。
- 等待队列中事务的并发提交。

MySQL HA-Galera



MySQL HA-Galera-DDL



MySQL HA-Galera

- 优点
  1.多主多活同步复制架构。
    1.同步复制：事务要么在所有节点提交或不提交。
    2.多主复制：可以在任意节点进行写操作。
  2.节点自动配置：成员的添加和移除是自动的。
  3.行级别并行复制。
  4.IST & RSU特性。
    1.IST(Incremental State Transfer)增量状态传输。
    2.RSU(Rolling Schema Update)旋转更新架构。

- 缺点
  1.加入新节点，开销大。需要复制完整的数据。
  2.所有的写操作都将发生在所有节点上。
  3.有多少个节点就有多少重复的数据。
  4.只支持InnoDB引擎。

MySQL HA-Galera-Why InnoDB is Choosen!

- 数据库必须支持事务。

- 复制事件要能够原子性执行。

- 复制事件必须全局有序。

Galera vs MySQL Replication

- 分布式系统的CAP理论
  1.C－一致性：所有节点的数据一致。
  2.A－可用性：一个或多个节点失效，不影响服务请求。
  3.P－分区容忍性：节点间的连接失效，仍然可以处理请求。

- 任何一个分布式系统，需要满足这三个中的两个
  1.MySQL Replication: 可用性和分区容忍性。
  2.Galera: 一致性和可用性。

MySQL HA-Percona XtraDB Cluster

- 封装了Galera，更易用。
- 完全兼容MySQL和Percona Server。
- Percona的支持以及活跃的社区交流。

HA环境下也需要做备份

- 冷备
- 热备
- 逻辑备份
- 物理备份

MySQL相关工具介绍

- MYSQLEXP
- MYSQLCLONE

MYSQLEXP-What is MYSQLEXP?

- MYSQLEXP is free, simple and efficient. It's used to export data from mysql database.

- You can execute a sql statement explicitly or read it from sql file, yet you can flexibly specify the field delimiter, and what field enclosed by.

MYSQLEXP-How to use it?

```
royalwzy@tools$ ./mysqlexp
mysqlexp Version 2.1.0.0, for OS X, by Wang Zhaoyang(royalwzy.com)
<sonne.k.wang@gmail.com>

This tool is used to export data through sql statement.
Usage: mysqlexp [OPTIONS] [database]
 -?, --help                Display this help and exit.
 -I, --help                Synonym for -?.
 -h, --host=name           Connect to host, [localhost] for default.
 -u, --user=name           User for login, [root] for defualt.
 -p, --password=name       Password to use when connecting to server, [] for
default.
 -D, --database=name       Database to use, [mysql] for default.
 -P, --port=#              Port number to use for connection, [3306] for default.
 -S, --socket=name         The socket file to use for connection.
 -c, --charset-set=name    The charset to use for connection, [utf8] for default.
 -v, --verbose             Write more information, [0] for default.
 -V, --version             Output version information and exit.
 -H, --header              Whether display header or not.
 -d, --delimiter=symbol    Field delimiter, [,] for default.
 -E, --enclose=symbol      What field enclose by.
 -e, --sql=statement       Sql statement to execute.
 -s, --sqlfile=name        The file to store sql statement, instead of reading
from command line.
 -f, --dumpfile=name       Write data to the dump file, [mysqlexp.csv] for
default.
 -b, --feedback=nums       How many rows to feedback, [500000] for default.
```

MYSQLEXP-Quick Examples!

1.by sql statement:

```
royalwzy@tools$ ./mysqlexp -hlocalhost -uroot -p*** -S "/data/mysql/mysql.sock" -Dtest -H -d"," -E"\"" -e "SELECT * FROM bank_cnaps" -f "/tmp/dumpfile.csv" -b 20000
2014-12-01 16:43:24.693194 INFO:        0 row  exported.
2014-12-01 16:43:24.722960 INFO:    20000 rows exported.
2014-12-01 16:43:24.752173 INFO:    40000 rows exported.
2014-12-01 16:43:24.781532 INFO:    60000 rows exported.
2014-12-01 16:43:24.810474 INFO:    80000 rows exported.
2014-12-01 16:43:24.839623 INFO:   100000 rows exported.
2014-12-01 16:43:24.868831 INFO:   120000 rows exported.
2014-12-01 16:43:24.895926 INFO:   138444 rows exported.
```

2.by sql file:

```
vi /tmp/mysqlexp.sql
SELECT * FROM bank_cnaps
royalwzy@tools$ ./mysqlexp -hlocalhost -uroot -p*** -S "/data/mysql/mysql.sock" -Dtest -H -d"," -E"\"" -s "/tmp/mysqlexp.sql" -f "/tmp/dumpfile.csv"
2014-12-01 16:47:19.663454 INFO:        0 row  exported.
2014-12-01 16:47:19.869040 INFO:   138444 rows exported.
```

MYSQLCLONE-What is MYSQLCLONE?

MYSQLCLONE is free, simple and efficient. It's used to transfer a mysql database to the other.

MYSQLCLONE-How to use it?

```
royalwzy@tools$ ./mysqlclone
mysqlclone Version 1.2.0.0, for OS X, by Wang Zhaoyang(royalwzy.com)
<sonne.k.wang@gmail.com>

This tool is used to clone database.
Usage: mysqlclone [OPTIONS] [database]
 -?, --help                Display this help and exit.
 -h, --help                Synonym for -?.
 --src-host=name           Source connect to host.
 --src-user=name           Source user for login.
 --src-password=name       Source password to use when connecting to server.
 --src-port=#              Source port number to use for connection, [3306] for default.
 --src-socket=name         The source socket file to use for connection.
 --src-db=name             Source databases to clone, cann't be system databases.
 --dst-host=name           Destination connect to host.
 --dst-user=name           Destination user for login.
 --dst-password=name       Destination password to use when connecting to server.
 --dst-port=#              Destination port number to use for connection, [3306] for
default.
 --dst-socket=name         The destination socket file to use for connection.
 --dst-db=name             Database names are remapped to another database, cann't be system
databases.
 --dst-bin-log             Whether generate binlog or not , [false] for default.
 -m, --mode                Which mode to transfer data, can only be load or insert, [load]
for default.
 -d, --drop-src-db         Whether drop source database or not, [FALSE] for default.
 -D, --no-data             No row information, [FALSE] for default.
 -E, --events              Events included, [FALSE] for default.
 -R, --routines            functions and procedures included, [FALSE] for default.
 -x, --lock-all-tables     Locks all tables across all databases, [Lock the table to be
read] for default.
 -v, --verbose             Write more information, [0] for default.
 -V, --version             Output version information and exit.
```

MYSQLCLONE-Quick Examples!

1.LOAD mode:
royalwzy@tools$ ./mysqlclone --src-host=localhost --src-user=root --src-db=test --dst-host=localhost --dst-user=root --dst-db=test1
2014-12-05 21:05:00 INFO: database test started to clone, and remapped to test1.
2014-12-05 21:05:00 INFO: - table [t1] was transferred, Records: 0 Deleted: 0 Skipped: 0 Warnings: 0
2014-12-05 21:05:00 INFO: - table [t2] was transferred, Records: 3 Deleted: 0 Skipped: 0 Warnings: 0

2.SCHEMA only:
royalwzy@tools$ ./mysqlclone --src-host=localhost --src-user=root --src-db=test --dst-host=localhost --dst-user=root --dst-db=test1 -R -E -D
2014-12-05 21:06:08 INFO: database test started to clone, and remapped to test1.
2014-12-05 21:06:08 INFO: - table [t1] was transferred.
2014-12-05 21:06:08 INFO: - table [t2] was transferred.
2014-12-05 21:06:08 INFO: - procedure [pr_test] was transferred.
2014-12-05 21:06:08 INFO: - procedure [pr_time] was transferred.
2014-12-05 21:06:08 INFO: - function [fn_test] was transferred.
2014-12-05 21:06:08 INFO: - event [event_test] was transferred.

3.INSERT mode:
royalwzy@tools$ ./mysqlclone --src-host=localhost --src-user=root --src-db=test --dst-host=localhost --dst-user=root --dst-db=test1 --mode=INSERT
2014-12-05 21:04:16 INFO: database test started to clone, and remapped to test1.
2014-12-05 21:04:16 INFO: - table [t1] was transferred, [0] row succeed and [0] row failed.
2014-12-05 21:04:16 INFO: - table [t2] was transferred, [3] rows succeed and [0] row failed.

Q&A