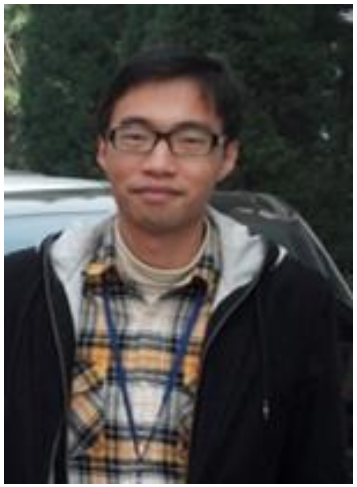

SH'OUG
SHANGHAI ORACLE USERS GROUP
上海ORACLE用户组

SH'OUG

MySQL高可用方案比较

by 汪伟华, Jun 2015

汪伟华



- 8年Oracle相关开发及数据库运维经验
  (Oracle DB, MySQL, Oracle Apps)
- MySQL OCP

ORACLE
Certified Professional
MySQL 5.6 Database
Administrator

SH'OUG
SHANGHAI ORACLE USERS GROUP
上海ORACLE用户组

Parnassus
诗 檀

- E-mail：biot.wang@parnassusdata.com

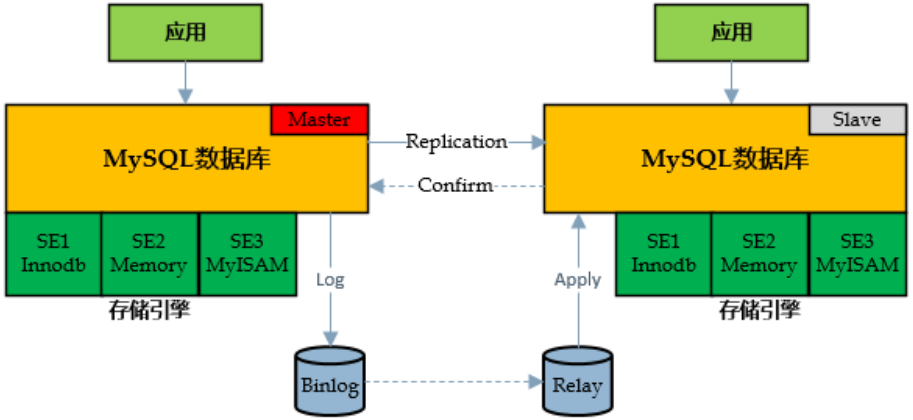
议程

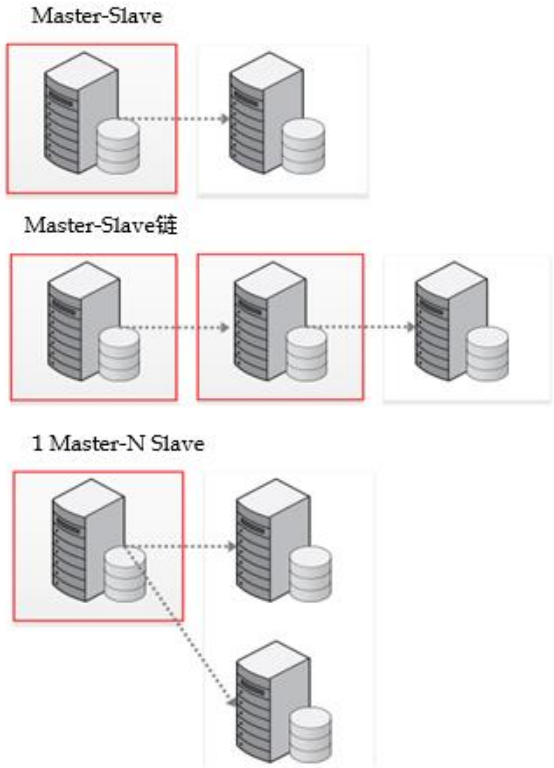
- 多样的MySQL HA架构

多样的MySQL HA架构

- MySQL replication
  - 可能会丢失部分数据 (几秒), 可靠性一般
  - 需要两倍存储空间
  + 可扩展
- DRBD/Pacemaker/Corosync/Linux(过去的DRBD/heartbeat/Linux架构)
  - 受到SYNC模式的性能影响
  - 需要两倍存储空间（且备机不可用）
  - 不可扩展 (仅主 + 镜像)
  - 可扩大LVM，不过需要主备同时扩大，同时需要设置用以识别扩大的空间
  + 高可靠性
- LVS+MySQL Cluster
- 相关架构特点对比:
  http://dev.mysql.com/doc/mysql-ha-scalability/en/ha-overview.html
- 其他架构(MySQL Fabric, Heatbeat+共享存储, MHA架构)等

MySQL replication





- 全局事务ID (GTID)
- Crash-Safe多线程Slaves
- Group Commit
- Replication Checksums
- Binlog API

如何量化理解More Read, More Slaves这句话

MySQL replication 对于那些需要处理频繁读和非频繁写的系统非常有益。
在建设前评估需备库数量和性能提高比率，你必须了解你的查询模式，并以主库和备库的读写关系为基准进行判断。

我们会用了一个简单计算模型来显示你从 replication 中可以获得的系统提升。

首先统计每秒相对的读(read)和写(write)次数。
假设当前系统负载中包含 10%写和 90%读，而我们已经确定读写基准关系为读的次数为 1200 减去写次数的两倍。换言之，系统可以在无写操作的情况下达到每秒 1200 次读，其平均写时间
比平均读时间慢 2 倍, 两者的关系是线性的。 假设主库和备库性能相同，且我们使用 1 主 N 备。那么，对每个服务器(主库或备库):

如何量化理解More Read, More Slaves这句话

reads = 1200 - 2 * **writes**
reads = 9 * **writes** / (**N** + 1) (读被分布在各个服务器上, 写则被replicate 到多个备库上)
9 * **writes** / (**N** + 1) + 2 * **writes** = 1200
writes = 1200 / (2 + 9/(**N** + 1))

最后的等式表明在 N 个备库的情况下和写之间的关系，通过分析可得出以下结论：

- 如果 N=0（意味着无 replication），系统每秒可处理约 1200/11=109 次写。
- 如果 N=1，可得到每秒为 184 次写。
- 如果 N=8，每秒写提升至 400 次。
- 如果 N=17，每秒写为 480 次。

最终 N 趋于无穷大，我们可获得约 600 次写每秒。系统性能提升约 5.5 倍。
然而，仅 8 台备库的情况下，我们就能获得大约 4倍的性能提升了。

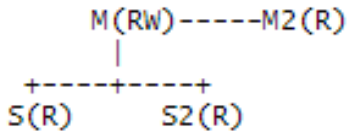
如何量化理解More Read, More Slaves这句话

虽然模型很理想化。然而，通过回答以下问题应该能帮助你判断大致增加多少 replication 会对你的系统性能有一定的提高：

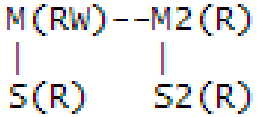
- 系统的读/写比率是多少？
- 如果你降低读负载，服务器可更多承贷多少写负载？
- 你的网络带宽足够多少台备库使用？

基于MySQL replication之后的更多架构变化

明确理解了主从读写关系后，高可用中的数据一致性问题被提上日程



此中架构被广泛采纳。
这种架构的问题在于解决一致性。
Failureover中产生的M2不一致，将导致S和S2无法进行Replication



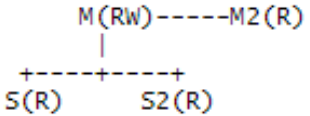
这种架构最大的劣势是三层管理(M->M2->S2)，一旦M2出现问题，S2就不再可用，恢复需要对M2和S2两个节点进行恢复。这种架构基本大家都不会用。

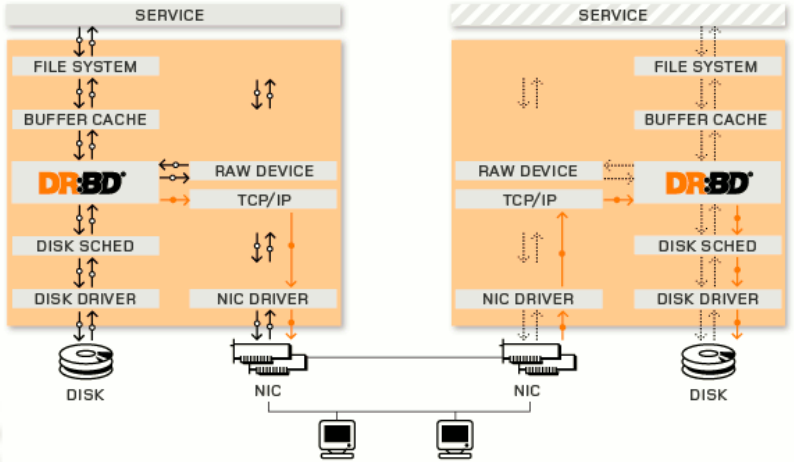
DRBD/Pacemaker/Corosync/Linux

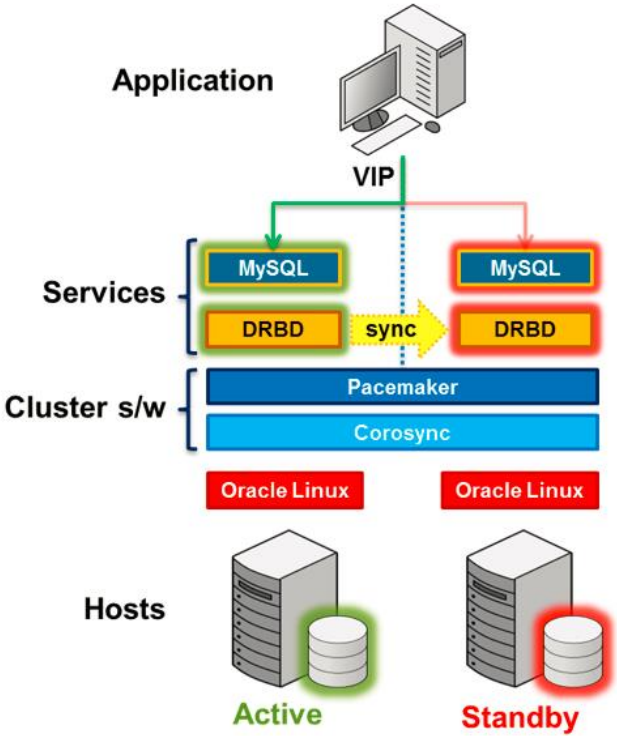
- Corosync控制主备资源切换

(通过corosync或者heartbeat提供的资源信息，
pacemaker通过resource agent来管理资源，比如start、stop)

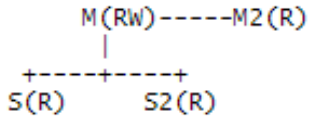
- DRBD类似主机热备
- 主备库不共享
- VIP



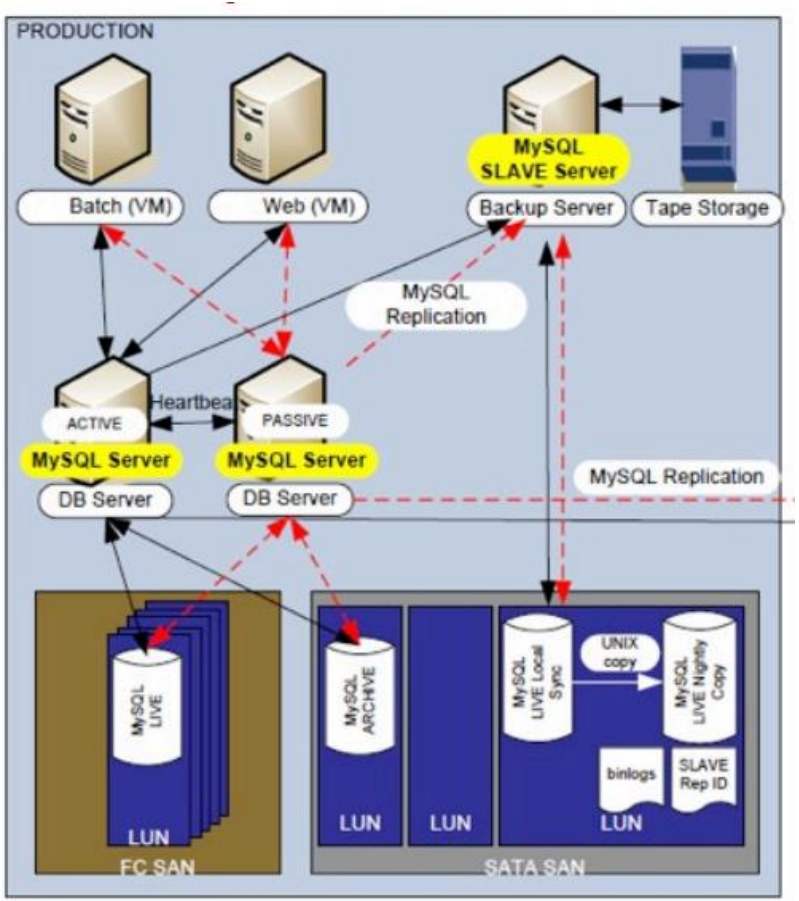




共享存储架构



- Heartbeat+共享存储HA架构
  - Heartbeat控制资源
  - 共享存储
    - LUN's accessible from two servers
    - ext3 – 仅mount到活动的节点
    - no LVM - LVM is not clustered
  - Virtual IP / VIP
  - MySQL 5.0实例运行在一个节点上
    - read-write数据必须为InnoDB
    - read-only数据可以是MyISAM
  - 优缺点
    - 共享一个数据文件
    - 一旦主库down, 切换备库恢复时间长(10min多分钟+)



Oracle ACE

OakTable.net

此架构来自**Alex Gorbachev**在IOUG Collaborate的演讲

备份策略

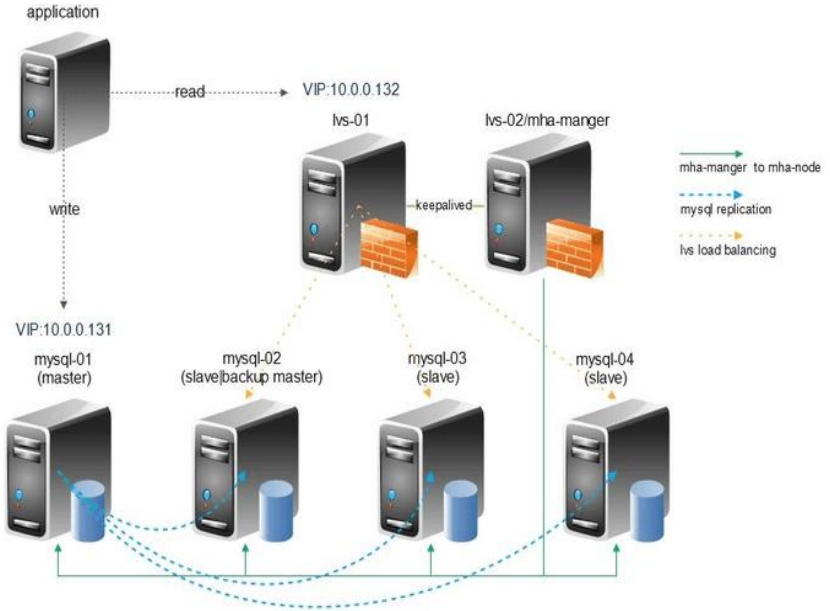
- 分为LIVE和ARCHIVE
  - LIVE - InnoDB 200-500GB
  - ARCHIVE - MyISAM 2 TB
- LIVE备份 - on slave
  - FLUSH ... WITH READ LOCK
  - 停止slave SQL thread
  - LVM snapshot 或RSYNC
  - 或者使用mysqldump, mysqlbinlog进行备份转储
- 恢复
  - LIVE first as a whole instance
  - ARCHIVE later - it's MyISAM

Lvs+KeepAlived+MHA+MySQL架构

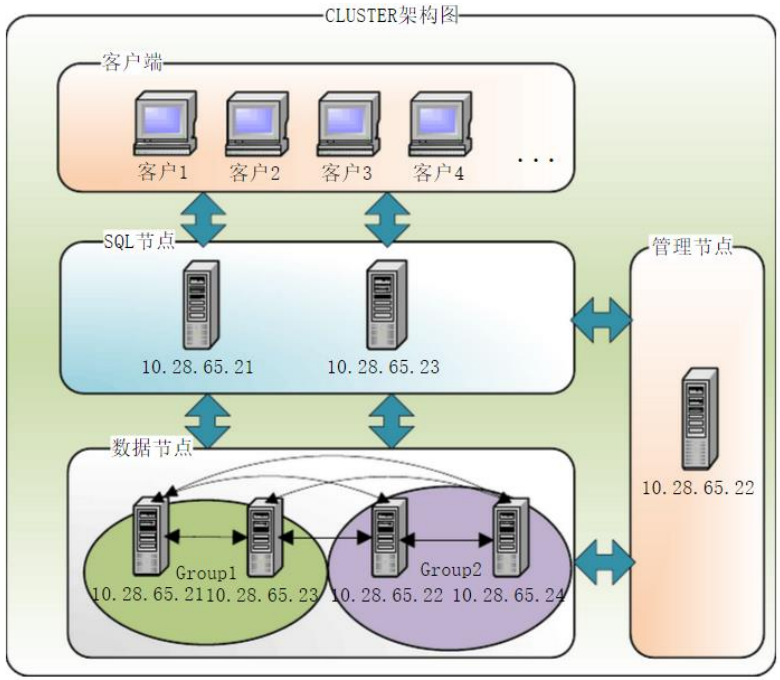
- Lvs+KeepAlived+MHA+MySQL架构
  - 1）从宕机崩溃的Master保存二进制日志事件（binlogevent）；
  - 2）识别含有最新更新的Slave；
  - 3）应用差异的中继日志(relaylog)到其他Slave；
  - 4）应用从Master保存的二进制日志事件；
  - 5）提升一个Slave为新的Master；
  - 6）使其他的Slave连接新的Master进行复制；

优缺点

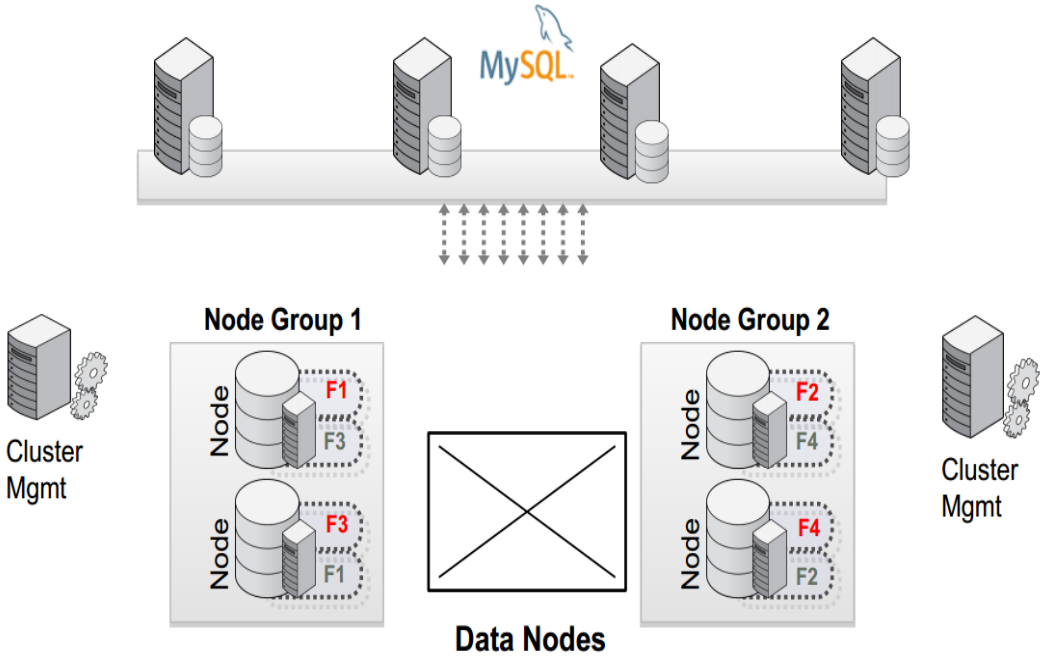
- 需要至少半同步复制以降低数据丢失风险
- 故障切换速度快(0~30s)
- MHA能最大程度保证数据一致性
- 注意：MySQL服务挂了，但是可以从服务器拷贝二进制。但如果硬件宕机或者SSH不能连接，不能获取到最新的binlog日志，如果复制出现延迟，会丢失数据。



LVS + MySQL NDB Cluster架构



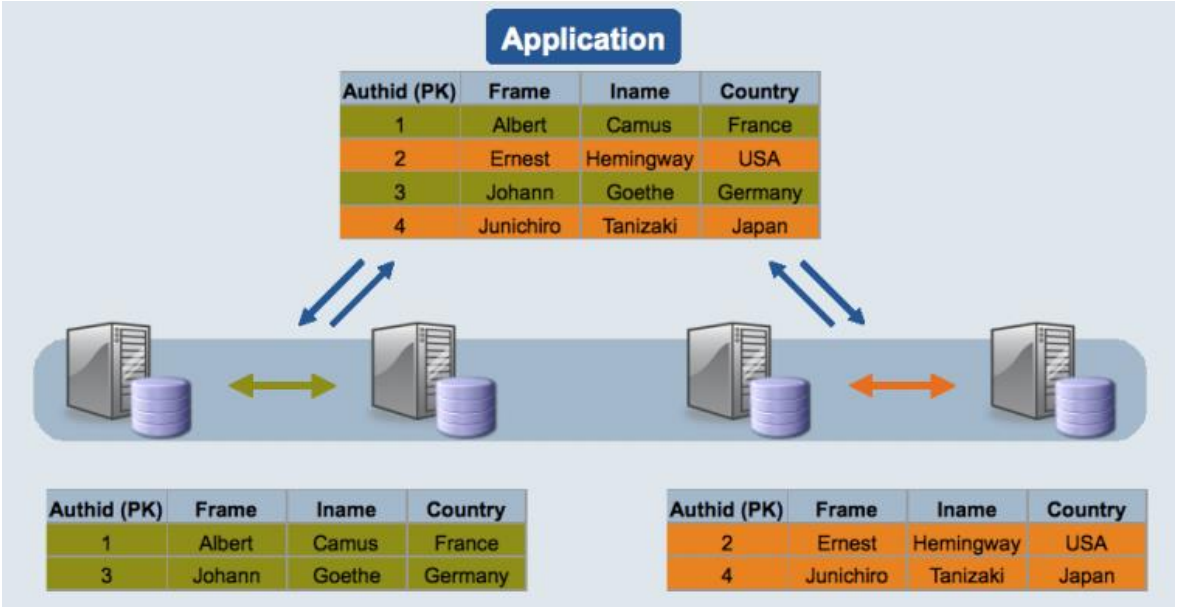
MySQL NDB Cluster – Shared Nothing



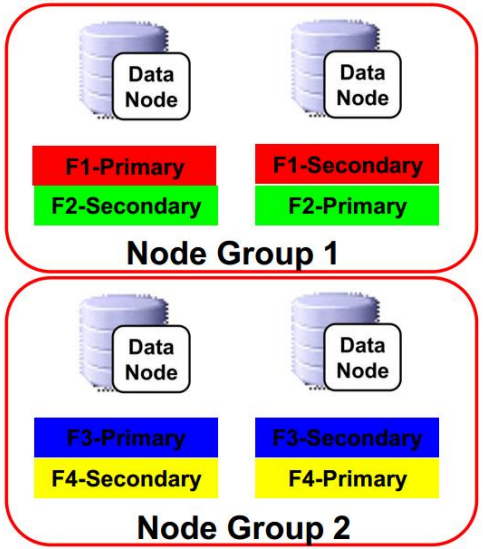
优点：

- 分布式、无共享架构:
  集群中的每个节点都是冗余的，可以放在单独的主机上，从而确保在发生进程、硬件或网络故障时的持续可用性。
- 无单点故障
- 同步复制
- 自动故障切换
- 多站点集群

MySQL NDB Cluster – 分片



ID	Capital	Country	UTC	
1	Copenhagen	Denmark	2	Partition 1
2	Berlin	Germany	2	
3	New York City	USA	-5	Partition 2
4	Tokyo	Japan	9	
5	Athens	Greece	2	Partition 3
6	Moscow	Russia	4	
7	Oslo	Norway	2	Partition 4
8	Beijing	China	8	



MySQL NDB Cluster

缺点：

- 对需要进行分片的表需要修改引擎Innodb为NDB，不需要分片的可以不修改。
- NDB的事务隔离级别只支持Read Committed，即一个事务在提交前，查询不到在事务内所做的修改；而Innodb支持所有的事务隔离级别，默认使用Repeatable Read，不存在这个问题。
- 外键支持：外键性能有问题（因为外键所关联的记录可能在别的分片节点中），所以建议去掉所有外键。
- Data Node节点数据会被尽量放在内存中，对内存要求大，如果内存不够用会导致性能大幅下降。

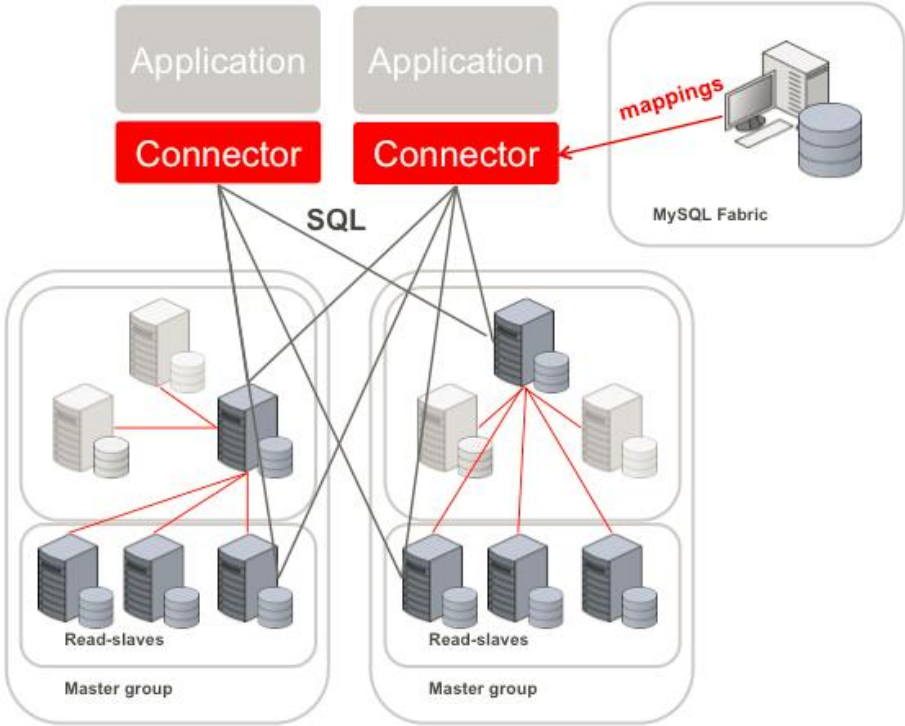
优点：

- 可将数据分布于多地
  – 在多地进行同步复制(Synchronous replication)和自动故障切换(auto-failover)
- 是一个无冲突处理的Active-Active双活方案

MySQL Fabric

- MySQL Fabric
  - 2014年年中发布的新解决方案
  - 需要使用新的Connector API应用接口来访问
  - MySQL Fabric Node管理整个MySQL Farm
  - 开源并基于MySQL Replication
  - 可自动分片和主备切换

  优缺点：
  - 自增长键不能作为分片的键；
  - 事务及查询只支持在同一个分片内，事务中更新的数据不能跨分片，查询语句返回的数据也不能跨分片。
  - 当前为止还没有成熟的实际实施案例研究。



Using, Learning & Sharing

SHOUG

Let's Leverage Oracle Together


SH'OUG
SHANGHAI ORACLE USERS GROUP
上海ORACLE用户组