MySQL 管理和架构介绍

平安健康的 MySQL 开发，测试，以及产品环境介绍

目前所有系统均为 mysql 分为 office（开发）测试，线上网段　目前测试线上均为杭州机房网段。

开发/10.0.128 网段　测试 业务+性能/10.128.240 网段　线上/10.128.6 网段

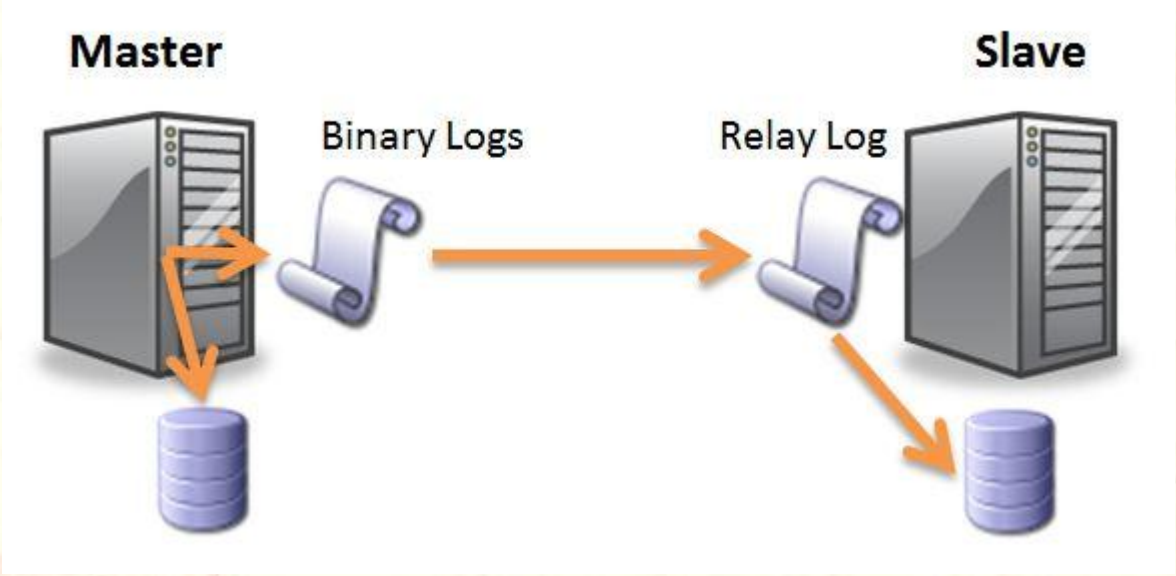
给个例子：

目前 user 库的线上为 10.128.6.x 网段 使用 TDDL 做水平拆分,分布在两台物理机器　测试网段为 10.128.240.x 网段 没有使用 TDDL 全部库均在一台 DB 上　开发网段为 10.0.128.x 处于逐渐淘汰状态。

目前所有 mysql 版本均为 5.6.17 所有开发测试环境均需要与线上统一,遗留的历史问题如 diamond 数据库后续也会升级到 5.6 版本

MySQL 的主从模式介绍以及变种

目前主要采用 M-S 架构 即 master 通过逻辑 SQL 语句复制到 slave 机器 , slave 目前均为备份机器 没有提供查询服务。



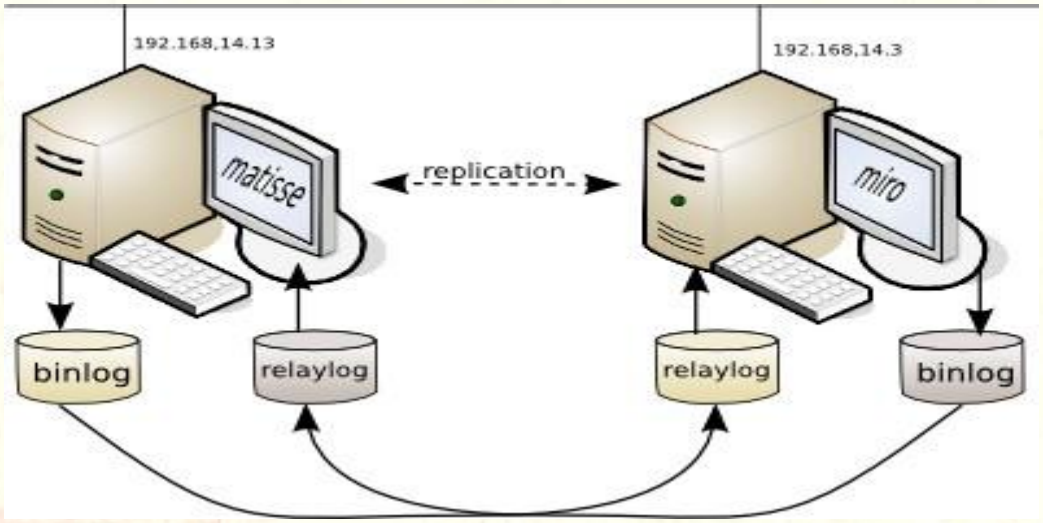
具体过程为 master 操作写入本地 binlog -> 通过 dump 进程读入到 slave 本地 relay log -> slave 回放 relay log 完成复制。

此过程中 可能出现瓶颈的地方为 本地 binlog 的写入 ,dump 读取的速率,网络的传输延迟,relay log 的写入,以及 relay log 的 apply 速率。

另外的一些复制架构：

双 MM 架构：

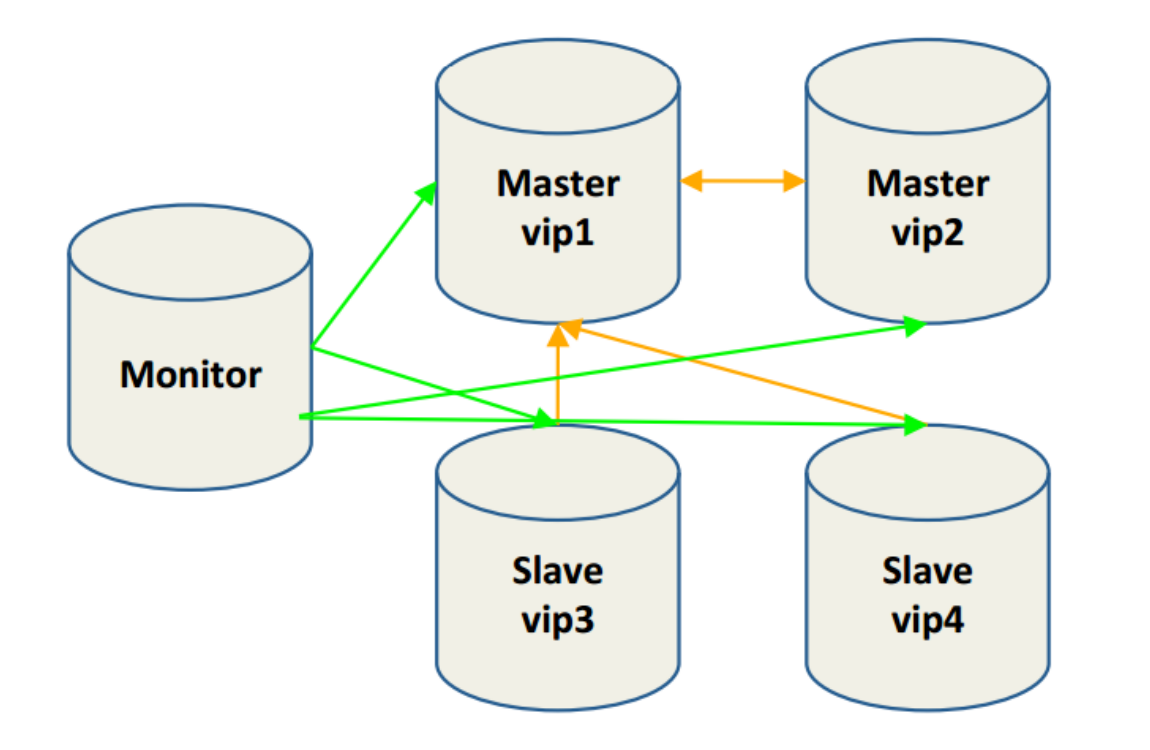
两边同为 master 通过 PK offset 控制不相互冲突 ， 对于 M-S 架构的优势为 两台 DB 均对外提供服务。



两台 mysql 互写 binlog 各自维护自己的 local relay-log。可以配合 keepalived 等第三方 HA 软件来实现 MySQL 高可用。

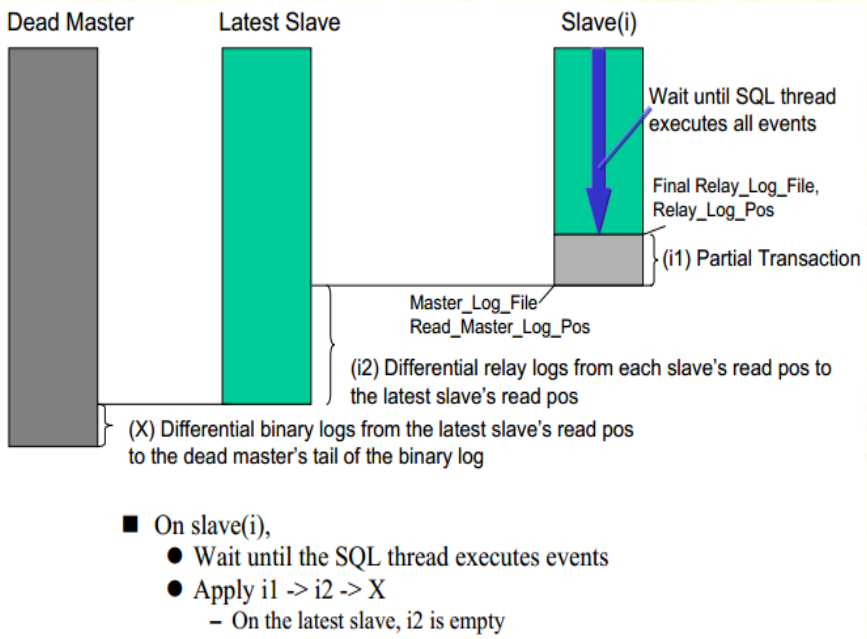
Google MMM 架构

通过 Monitor 节点实现两个 master 节点的监控,此架构兼顾了 M-M 以及 HA 的自动切换，通过 VIP 的切换实现对业务的透明。



具体过程：如果 master 1 down 掉　首先 remove 掉 vip1,show master status ,Change master to POS . 改变 slave 3 slave 4 的 master 为 master 2　online vip1 在 master 2 上。

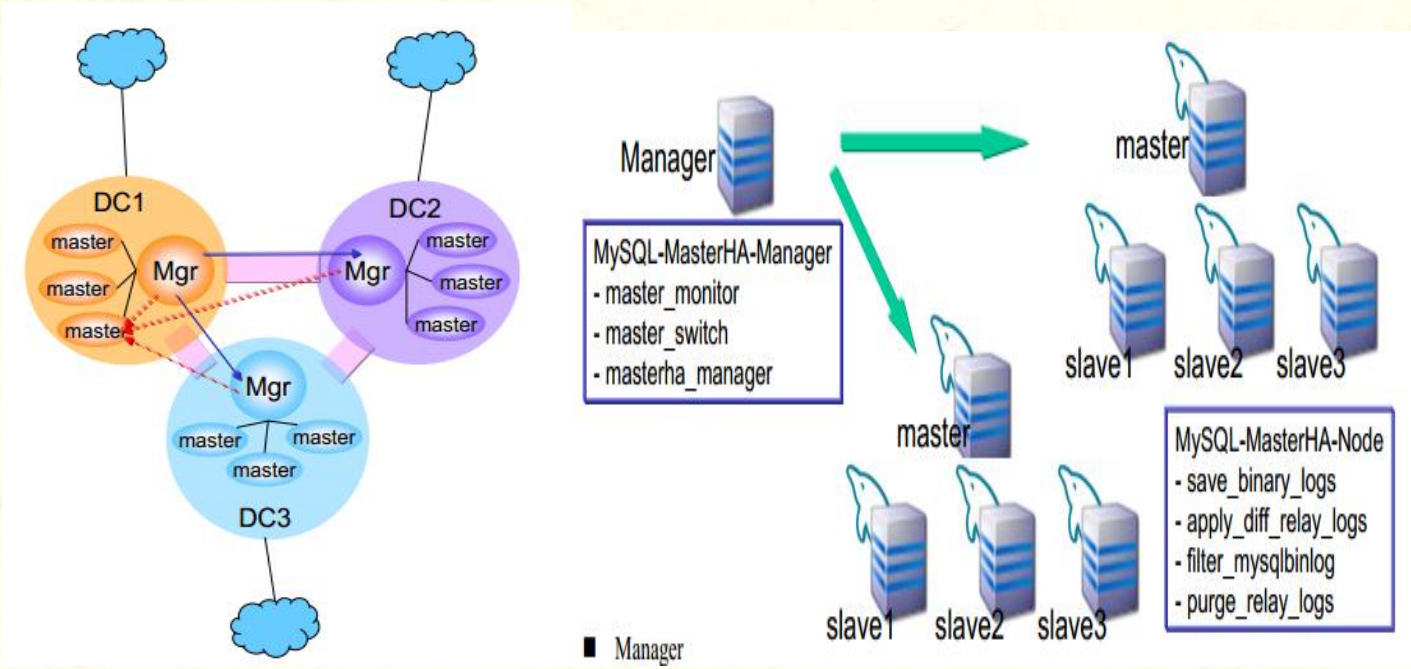
MHA 架构 -- Now we used



具体步骤： 首先在备选 slavei 上执行完有所 event 从最近的 latest slave apply log - recover slavei, 从 master 上的 binlog read 成一个 single file (gap between latest slave and master) - recover slavei 在 recover 的过程中同时 parallel recover slaves。

MHA 的优势,可以自动 recover slave 自动 change master VIP 的转移也是随 master 的转变而迁移.对业务透明。 可以集中化管理

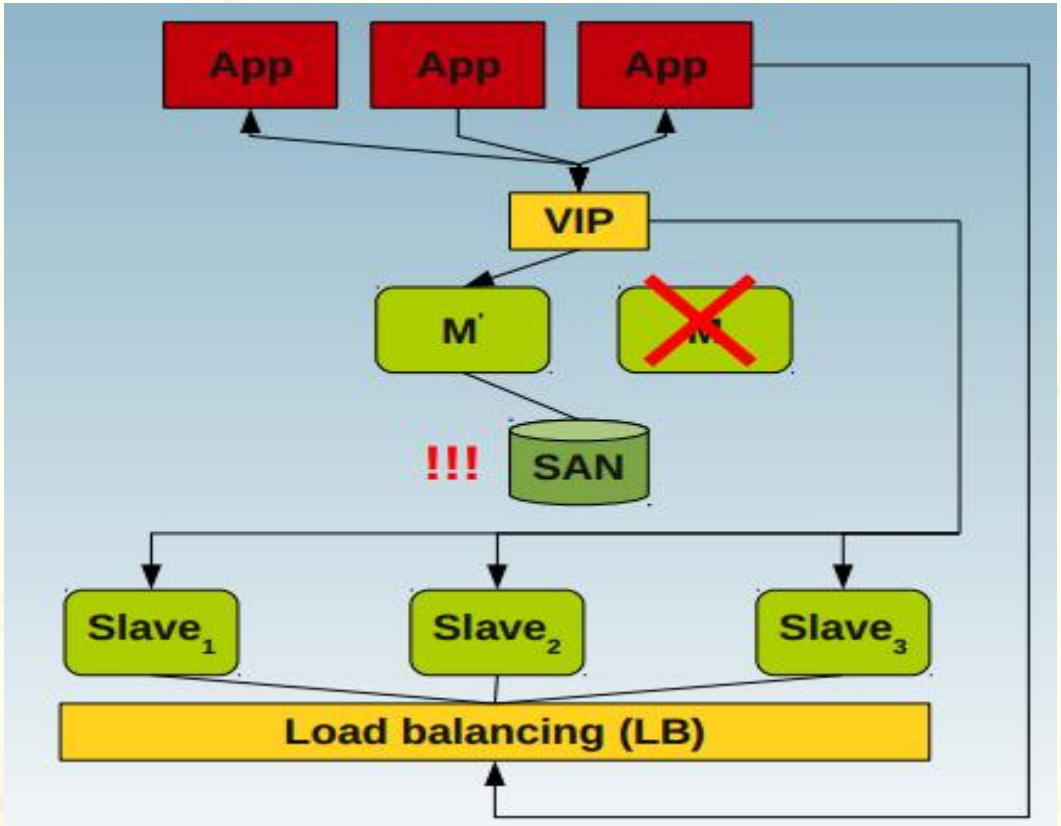
可以在一个 IDC 部署多个 manager 监控集群 (one manager node multi server db) 也可以在多个 IDC 部署不是多个 manager



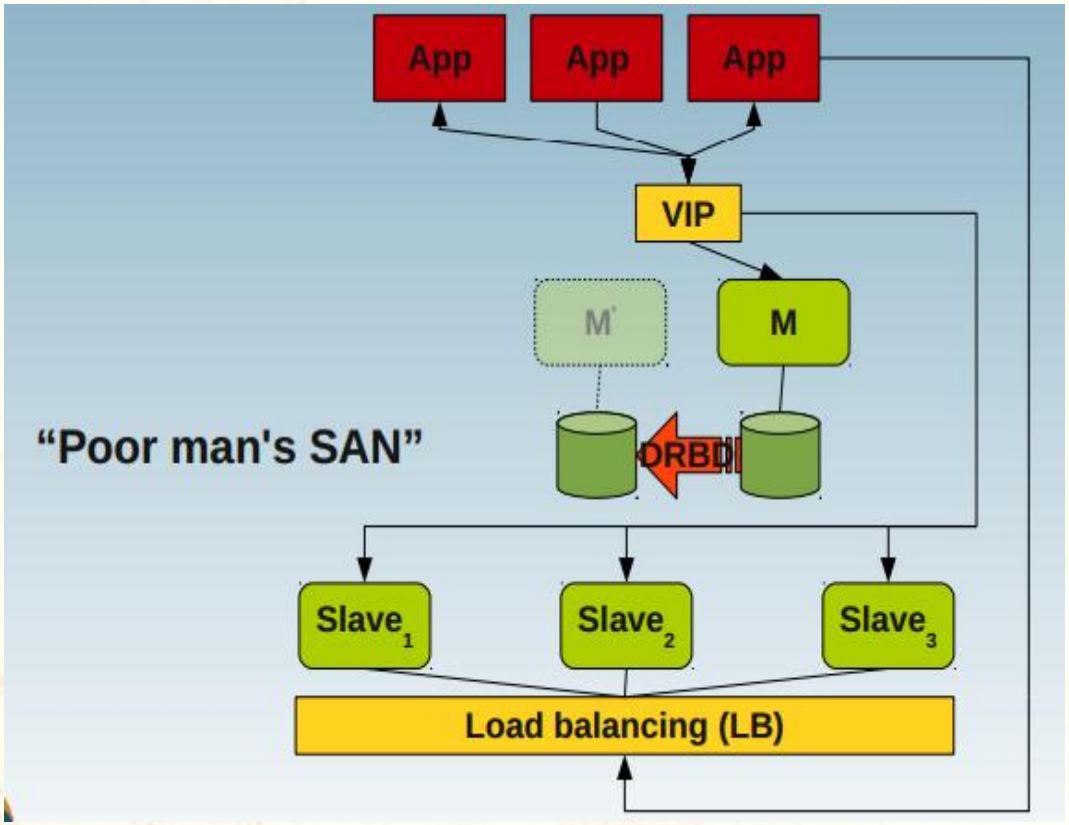
MHA 目前来说是一个最好的解决方案,很多公司在使用 例如 Facebook DeNA Booking YHD Alibaba 等。

Other HA solutions:

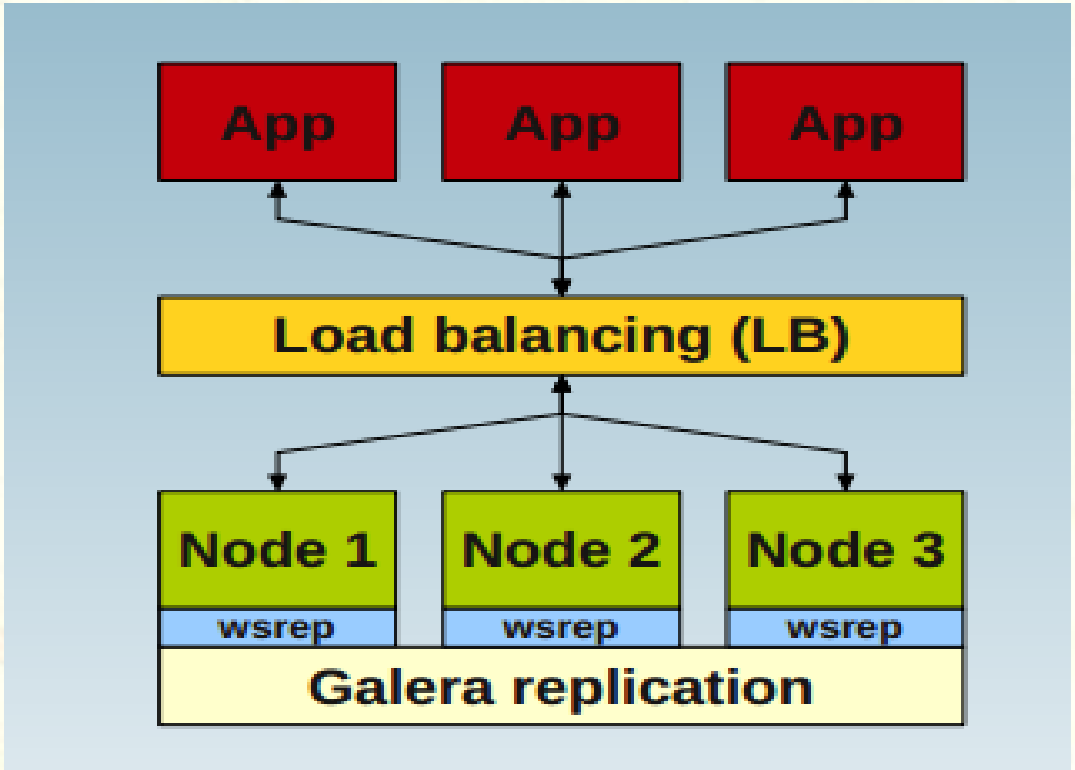
1. Shared disk with SAN　　优势：不会发生数据丢失　劣势： SAN 硬件成本 以及存储硬件成本，网络的复杂性。



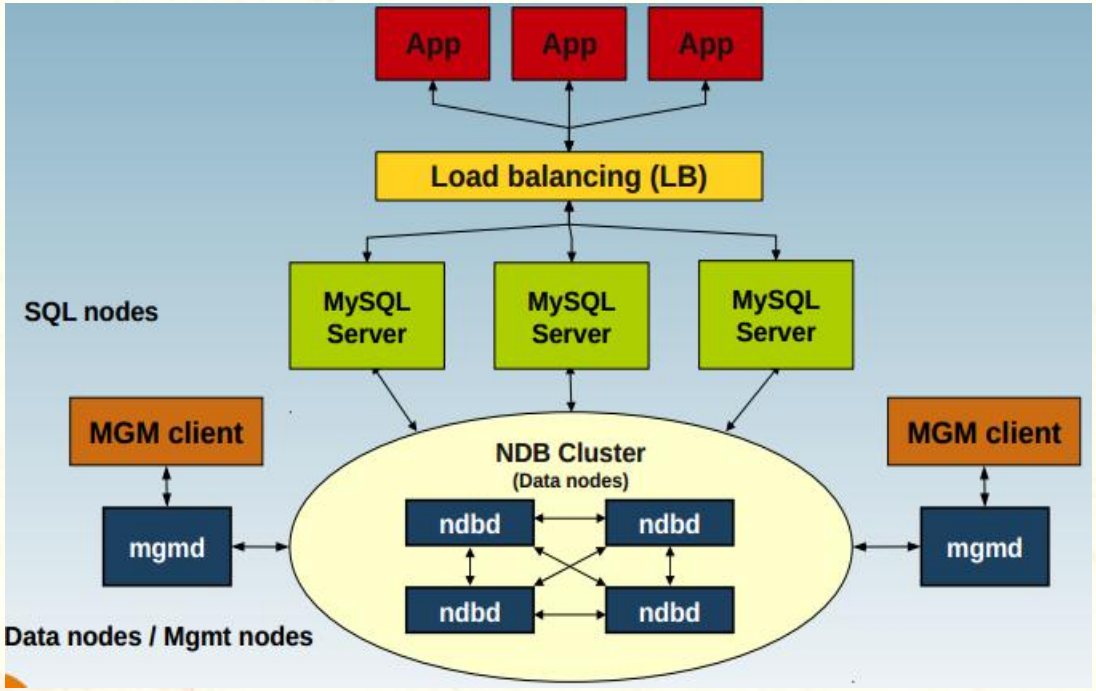
2. DRDB Block replication　优势：更低的复制延迟　劣势：不能做负载均衡 复杂的配置 不可控



3. Gelera replication 优势：负载均衡 多主节点读写 没有延迟 劣势：写多份 block 更多的 lock 问题 DDL 问题 不稳定



4. MySQL cluster NDB 优势: 数据分片存储 劣势 : 查询的限制 （根据 fragment） join 困难 目前不稳定。



另外的架构 主要为第三方 tools 的使用例如：keepalived+ M-S ，heartbeat, tungsten 复制 等这里不再详述。

MySQL 的常用工具介绍

线上 MySQL 采用 xtrabackup 备份，另外 master slave 架构中的 slave 也会充当 cold backup 的角色。使用 python 脚本 调用 xtrabackup 进程在线备份。

目前 mysql 迁移主要使用如下工具：

逻辑导入导出： mysqldump, mysqldumper, mysqlcopy

物理备份恢复： xtrabackup innodb-hotbackup（商业）

Binlog 备份 --- 可用于多种架构设计 ： binlog-server (SkySQL) mysqlbinlog-remotebackup

http://blog.booking.com/mysql_slave_scaling_and_more.html

https://mariadb.com/blog/mariadb-replication-maxscale-and-need-binlog-server/

我们现在的备份策略 -

1. 通过读取字典表自定义备份数据库 （每天一次全备份）

```
-rw-r--r-- 1 root root     4393529 Aug  3 15:01 xtrade_fullbak_2014-08-03-15-00-09_10.128.6.22.3308.tar.gz
-rw-r--r-- 1 root root   324605709 Aug  3 15:02 xtrade_fullbak_2014-08-03-15-00-09_10.128.6.22.3310.tar.gz
-rw-r--r-- 1 root root 21177196973 Aug  3 17:31 xtrade_fullbak_2014-08-03-16-00-13_10.128.6.22.3309.tar.gz
-rw-r--r-- 1 root root    28579588 Aug  4 12:01 xtrade_fullbak_2014-08-04-12-00-22_10.128.6.22.3306.tar.gz
-rw-r--r-- 1 root root     9598166 Aug  4 12:00 xtrade_fullbak_2014-08-04-12-00-22_10.128.6.28.3306.tar.gz
-rw-r--r-- 1 root root     5860888 Aug  4 13:01 xtrade_fullbak_2014-08-04-13-00-25_10.128.6.30.3306.tar.gz
-rw-r--r-- 1 root root   179132089 Aug  4 14:02 xtrade_fullbak_2014-08-04-14-00-29_10.128.6.22.3307.tar.gz
-rw-r--r-- 1 root root     5408089 Aug  4 14:01 xtrade_fullbak_2014-08-04-14-00-29_10.128.6.28.3307.tar.gz
-rw-r--r-- 1 root root     5203351 Aug  4 14:01 xtrade_fullbak_2014-08-04-14-00-29_10.128.6.32.3306.tar.gz
-rw-r--r-- 1 root root     4393536 Aug  4 15:01 xtrade_fullbak_2014-08-04-15-00-32_10.128.6.22.3308.tar.gz
-rw-r--r-- 1 root root   337303677 Aug  4 15:03 xtrade_fullbak_2014-08-04-15-00-32_10.128.6.22.3310.tar.gz
-rw-r--r-- 1 root root 21602516669 Aug  4 17:34 xtrade_fullbak_2014-08-04-16-00-36_10.128.6.22.3309.tar.gz
```

2. 通过 mariadb 汇总的备份

http://www.vmcd.org/2014/05/multi-master-rep-in-mysql-db/

3. 通过 slave 充当备份 （后期可能用于读负载均衡）

平安健康的 MySQL 线上架构介绍

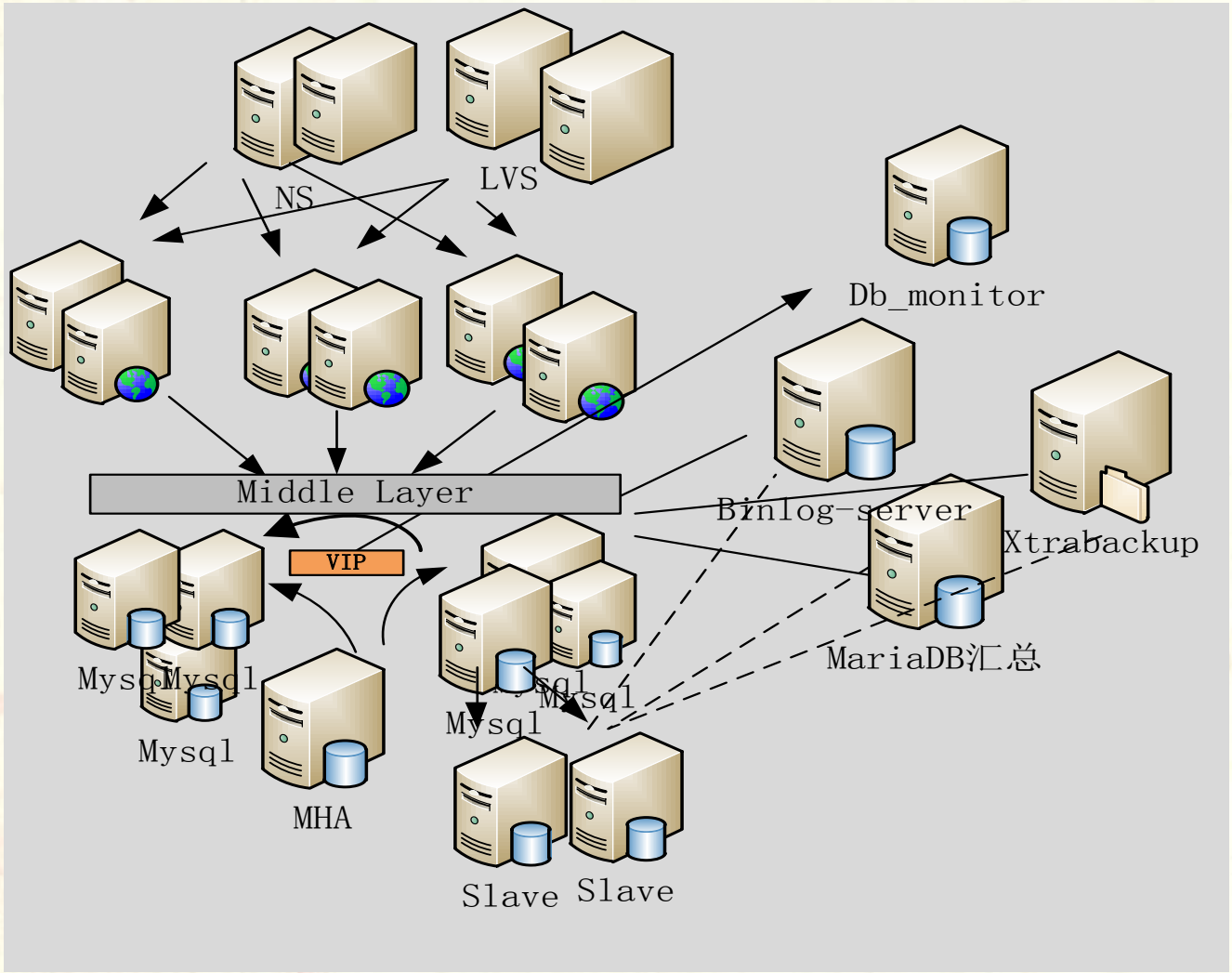
目前我们采用 MHA 作为 MySQL 高可用架构,目前最重要的两套数据库 user,pay 分别采用 TDDL,钆账功能 保证数据的可靠性 LVS 替代 haproxy NS 作为 7 层负载均衡 IM 业务，NS7 层负载 WEB 业务。

我们将来要实现的架构:

采用 backuped binlog 作为第三方 binlog 源 MHA 已经支持 保证在主 master 机器起不来的情况下 不丢失 binlog 。

MHA 采用第三方路由方式（已经实现）,在跨 IDC 的应用中可以防止 MHA master 节点误判。

底层的 slave 分布式可以考虑采用 binlog-server 实现,以减小网络传输量（当然这个架构需要大量的 slave 情况下才有明显的作用） 总体架构为：TDDL+ MHA + binlog-backup + monitor


LVS
NS
Db_monitor
Middle Layer
Binlog-server
Xtrabackup
VIP
MariaDB汇总
Mysql
Mysql
Mysql
Mysql
MHA
Slave
Slave

关于中间层：

目前使用 TDDL+Tair 水平拆库策略 即：APP --> 水平拆分到 10 个 database --> 每个 DB 10 张 table.
Cache 层由 tair 实现， 目前冗余量为 10 亿,基本满足了业务需求。另外关于 MySQL 的特殊性，需要另外一些系统以及硬件的辅助。

附 tair + TDDL 架构图：