探索 MySQL 高可用架构之 MHA(概念和架构篇)

什么是高可用性?

很多公司的服务都是 24 小时*365 天不间断的。比如 Call Center。这就要求高可用性。再比如购物网站，必须随时都可以交易。那么当购物网的 server 挂了一个的时候，不能对业务产生任何影响。这就是高可用性。

如何处理 failover?

解释 failover，意思就是当服务器 down 掉，或者出现错误的时候，可以自动的切换到其他待命的服务器，不影响服务器上 App 的运行。

以 MySQL 为例,什么样的架构才能保证其高可用性呢?

MySQL replication with manual failover

同步数据是采用 MySQL replication 的方法，在 MySQL 分表分块到主从已经解释。简单的说就是从库根据主库的日志来做相应的处理，保证数据的一致。通常还配合 MySQL Proxy 或 Amoeba 等进行读写分离减少服务器压力。

manual failover，显然当 Master 挂掉时，利用本方式是需要手动来处理 failover，一般来说是将 slave 更改为 server。

Master-Master with MMM manager(Multi-Master Replication Manager)

同步数据的方式是 Multi-Master Replication Manager，在 MySQL 分表分块到主从解释，多主多从的设置，是一个 loop 环形，每个 DB 既是前一个 DB 的 Slave 又是后一个的 Master。优势就在于，一个 Master 挂掉，也还可以继续 DB 操作。每个 DB 都可以进行读写，分散压力。

Heartbeat/SAN

处理 failover 的方式是 Heartbeat，Heartbeat 可以看成是一组程序，监控管理各个 node 间连接的网络。当 node 出现错误时，自动启动其他 node 开始服务。Heartbeat 必须解决的一个问题就是 split brain，在网络中的一个 node down 掉后，每个 node 都会认为其他 node down 掉并尝试开始服务，因为产生数据冲突。

通过 SAN 来共享数据

SAN：Storage Area Network，是一种 LAN 来处理大数据量的传输，提供了计算机和存储系统之间的数据传输。各个计算机组成的集群可以通过 SAN 共享存储。

Heartbeat/DRBD

处理 failover 的方式依旧是 Heartbeat。

同步数据使用 DRBD：Distributed Replicated Block Device(DRBD)是一个用软件实现的、无共享的、服务器之间镜像块设备内容的存储复制解决方案。和 SAN 网络不同，它并不共享存储，而是通过服务器之间的网络复制数据。

MySQL Cluster

MySQL Cluster 也是由各个 DB node 组成一个 cluster，在这个 cluster 中由网络连接。可以自由的增减 node 的个数来对应数据库压力。

MySQL 高可用性大杀器之 MHA

MHA(Master High Availability)目前在 MySQL 高可用方面是一个相对成熟的解决方案，它由日本 DeNA 公司 youshimaton(现就职于 Facebook 公司)开发，是一套优秀的作为 MySQL 高可用性环境下故障切换和主从提升的高可用软件。在 MySQL 故障切换过程中，MHA 能做到在 0~30 秒之内自动完成数据库的故障切换操作，并且在进行故障切换的过程中，MHA 能在最

大程度上保证数据的一致性，以达到真正意义上的高可用。

该软件由两部分组成：MHA Manager(管理节点)和 MHA Node(数据节点)。MHA Manager 可以单独部署在一台独立的机器上管理多个 master-slave 集群，也可以部署在一台 slave 节点上。MHA Node 运行在每台 MySQL 服务器上，MHA Manager 会定时探测集群中的 master 节点，当 master 出现故障时，它可以自动将最新数据的 slave 提升为新的 master，然后将所有其 他的 slave 重新指向新的 master。整个故障转移过程对应用程序完全透明。

在 MHA 自动故障切换过程中，MHA 试图从宕机的主服务器上保存二进制日志，最大程度的保证数据的不丢失，但这并不总是可行的。例如，如果主服务器 硬件故障或无法通过 ssh 访问，MHA 没法保存二进制日志，只进行故障转移而丢失了最新的数据。使用 MySQL 5.5 的半同步复制，可以大大降低数据丢失的风险。MHA 可以与半同步复制结合起来。如果只有一个 slave 已经收到了最新的二进制日志，MHA 可以将最 新的二进制日志应用于其他所有的 slave 服务器上，因此可以保证所有节点的数据一致性。

目前 MHA 主要支持一主多从的架构，要搭建 MHA,要求一个复制集群中必须最少有三台数据库服务器，一主二从，即一台充当 master，一台充当备用 master，另外一台充当从库，因为至少需要三台服务器，出于机器成本的考虑，淘宝也在该基础上进行了改造，目前淘宝 TMHA 已经支持一主一从。

官方介绍：https://code.google.com/p/mysql-master-ha/

本次架构实现功能

a.一主库，三个从库(其中 1 个为备主)，实现 ABBB 复制

b.使用 Atlas 实现读写分离，主库和备主库接收写操作，从库接收读操作

c.使用 Mha 实现现有架构的高可用

d.使用 keepalived 实现 vip 的漂移

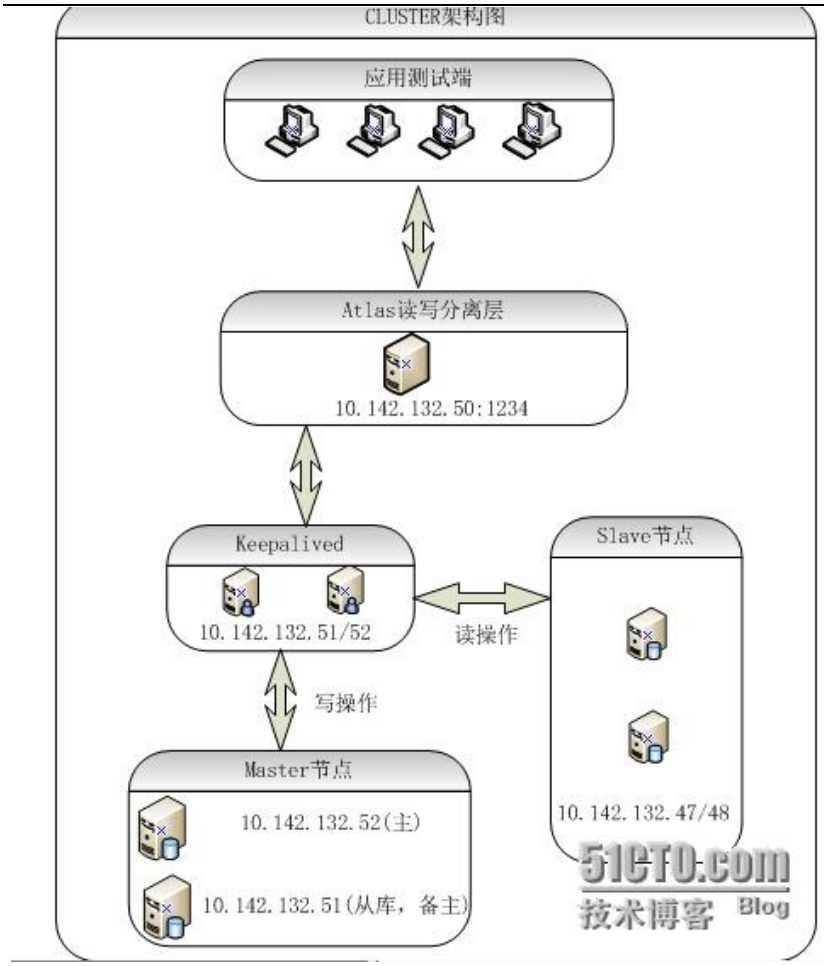
e.手工编写 shell，修复 Mha 的不足

e1.修复当 AB 故障切换一次后，mha-manager 会自动退出

e2.修复原主库，出问题后，修复后不能自动加入现有 AB 集群

e3.关于 relay log 的清除

本次实现架构图



本次架构主机划分

服务器名	IP 地址	虚拟 IP（VIP）
Master Server1	10.142.132.51	10.142.132.49
Backup Server1(备主)	10.142.132.52	10.142.132.49
Backup Server2	10.142.132.47	无
Backup Server3	10.142.132.48	无
Mysql Proxy	10.142.132.50	无

软件版本

项	项值
操作系统	RedHat release 5.4
Mysql 版本	mysql-5.6.17.tar.gz
Atlas	Atlas-2.2.1.el5.x86_64.rpm
keepalived	keepalived-1.2.2.tar.gz
Mha Manager	mha4mysql-manager-0.53.tar.gz
Mha Node	mha4mysql-node-0.53.tar.gz

安装路径

机器 IP	安装路径	路径说明
Master Server1	/app/mysql	Mysql 安装路径
	/etc/keepalive	Keepalive 配置路径
Backup Server1(备主)	/app/mysql	Mysql 安装路径
	/etc/keepalive	Keepalive 配置路径
Slave Server1	/app/mysql	Mysql 安装路径
Slave Server2	/app/mysql	Mysql 安装路径
Mysql Proxy	/usr/local/mysql-proxy	Mysql 读写分离器安装路径