**一、选择Percona Server、MariaDB还是MYSQL**

1、Mysql三种存储引擎

MySQL提供了两种存储引擎：MyISAM和 InnoDB，MySQL4和5使用默认的MyISAM存储引擎。从MYSQL5.5开始，MySQL已将默认存储引擎从MyISAM更改为InnoDB。

MyISAM没有提供事务支持，而InnoDB提供了事务支持。

XtraDB是InnoDB存储引擎的增强版本，被设计用来更好的使用更新计算机硬件系统的性能，同时还包含有一些在高性能环境下的新特性。

2、Percona  Server分支

Percona Server由领先的MySQL咨询公司Percona发布。

Percona Server是一款独立的数据库产品，其可以完全与MySQL兼容，可以在不更改代码的情况了下将存储引擎更换成XtraDB。是最接近官方MySQL Enterprise发行版的版本。

Percona提供了高性能XtraDB引擎，还提供PXC高可用解决方案，并且附带了percona-toolkit等DBA管理工具箱，

3、MariaDB

MariaDB由MySQL的创始人开发，MariaDB的目的是完全兼容MySQL，包括API和命令行，使之能轻松成为MySQL的代替品。

MariaDB提供了MySQL提供的标准存储引擎，即MyISAM和InnoDB，10.0.9版起使用XtraDB（名称代号为Aria）来代替MySQL的InnoDB。

4、如何选择

综合多年使用经验和性能对比，首选Percona分支，其次是MariaDB，如果你不想冒一点风险，那就选择MYSQL官方版本。

**二、常用的MYSQL调优策略**

**1、硬件层相关优化**

修改服务器BIOS设置

选择Performance Per Watt Optimized(DAPC)模式，发挥CPU最大性能。

Memory Frequency（内存频率）选择Maximum Performance（最佳性能）

内存设置菜单中，启用Node Interleaving，避免NUMA问题

**2、磁盘I/O相关**

使用SSD硬盘

如果是磁盘阵列存储，建议阵列卡同时配备CACHE及BBU模块，可明显提升IOPS。

raid级别尽量选择raid10，而不是raid5.

**3、文件系统层优化**

使用deadline/noop这两种I/O调度器，千万别用cfq

使用xfs文件系统，千万别用ext3；ext4勉强可用，但业务量很大的话，则一定要用xfs；

文件系统mount参数中增加：noatime, nodiratime, nobarrier几个选项（nobarrier是xfs文件系统特有的）；

**4、内核参数优化**

修改vm.swappiness参数，降低swap使用率。RHEL7/centos7以上则慎重设置为0，可能发生OOM

调整vm.dirty\_background\_ratio、vm.dirty\_ratio内核参数，以确保能持续将脏数据刷新到磁盘，避免瞬间I/O写。产生等待。

调整net.ipv4.tcp\_tw\_recycle、net.ipv4.tcp\_tw\_reuse都设置为1，减少TIME\_WAIT，提高TCP效率。

**5、Mysql参数优化建议**

建议设置default-storage-engine=InnoDB，强烈建议不要再使用MyISAM引擎。

调整innodb\_buffer\_pool\_size的大小，如果是单实例且绝大多数是InnoDB引擎表的话，可考虑设置为物理内存的50% -70%左右。

设置innodb\_file\_per\_table = 1，使用独立表空间。

调整innodb\_data\_file\_path = ibdata1:1G:autoextend，不要用默认的10M,在高并发场景下，性能会有很大提升。

设置innodb\_log\_file\_size=256M，设置innodb\_log\_files\_in\_group=2，基本可以满足大多数应用场景。

调整max\_connection（最大连接数）、max\_connection\_error（最大错误数）设置，根据业务量大小进行设置。

另外，open\_files\_limit、innodb\_open\_files、table\_open\_cache、table\_definition\_cache可以设置大约为max\_connection的10倍左右大小。

key\_buffer\_size建议调小，32M左右即可，另外建议关闭query cache。

mp\_table\_size和max\_heap\_table\_size设置不要过大，另外sort\_buffer\_size、join\_buffer\_size、read\_buffer\_size、read\_rnd\_buffer\_size等设置也不要过大。

**三、   MYSQL常见的应用架构分享**

**1、主从复制解决方案**

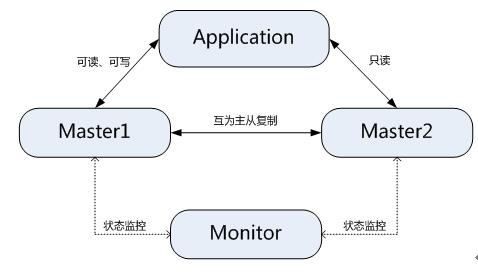
这是MySQL自身提供的一种高可用解决方案，数据同步方法采用的是MySQL replication技术。MySQL replication就是从服务器到主服务器拉取二进制日志文件，然后再将日志文件解析成相应的SQL在从服务器上重新执行一遍主服务器的操作，通过这种方式保证数据的一致性。

为了达到更高的可用性，在实际的应用环境中，一般都是采用MySQL replication技术配合高可用集群软件keepalived来实现自动failover，这种方式可以实现95.000%的SLA。

[](http://s5.51cto.com/wyfs02/M01/7D/68/wKioL1bnzqXTpra9AAA4W5Syh94610.jpg)

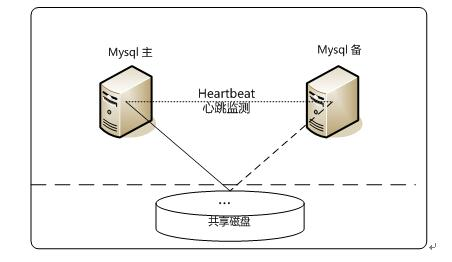
**2、MMM/MHA高可用解决方案**

MMM提供了MySQL主主复制配置的监控、故障转移和管理的一套可伸缩的脚本套件。在MMM高可用方案中，典型的应用是双主多从架构，通过MySQL replication技术可以实现两个服务器互为主从，且在任何时候只有一个节点可以被写入，避免了多点写入的数据冲突。同时，当可写的主节点故障时，MMM套件可以立刻监控到，然后将服务自动切换到另一个主节点，继续提供服务，从而实现MySQL的高可用。

[](http://s2.51cto.com/wyfs02/M00/7D/6B/wKiom1bnzinxyNbJAAAwafwrvME471.jpg)

**3、Heartbeat/SAN高可用解决方案**

在这个方案中，处理failover的方式是高可用集群软件Heartbeat，它监控和管理各个节点间连接的网络，并监控集群服务，当节点出现故障或者服务不可用时，自动在其他节点启动集群服务。在数据共享方面，通过SAN（Storage Area Network）存储来共享数据，这种方案可以实现99.990%的SLA。

[](http://s2.51cto.com/wyfs02/M02/7D/68/wKioL1bnzsbSfpPAAAAtC9WNeBM583.jpg)

**4、Heartbeat/DRBD高可用解决方案**

此方案处理failover的方式上依旧采用Heartbeat，不同的是，在数据共享方面，采用了基于块级别的数据同步软件DRBD来实现。

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。

[](http://s1.51cto.com/wyfs02/M00/7D/6B/wKiom1bnzkrB-vhLAAAxkEsaZh8093.jpg)

**四、MYSQL经典应用架构**

**[](http://s4.51cto.com/wyfs02/M00/7D/68/wKioL1bnzuahF40iAABcNZGGaqI867.jpg)**

其中：

         Dbm157是mysql主，dbm158是mysql主的备机，dbs159/160/161是mysql从。

MySQL写操作一般采用基于heartbeat+DRBD+MySQL搭建高可用集群的方案。通过heartbeat实现对mysql主进行状态监测，而DRBD实现dbm157数据同步到dbm158。

读操作普遍采用基于LVS+Keepalived搭建高可用高扩展集群的方案。前端AS应用通过提高的读VIP连接LVS，LVS有keepliaved做成高可用模式，实现互备。

         最后，mysql主的从节点dbs159/160/161通过mysql主从复制功能同步mysql主的数据，通过lvs功能提供给前端AS应用进行读操作，并实现负载均衡。