固态磁盘可以加强随机读写的性能。
相比机械磁盘固态磁盘支持更好的并发。
固态磁盘更容易损毁。

SSD PCI-E SSD

适用于存在大量随机 IO 的场景。
使用于解决单线程负载的 IO 瓶颈。

SSD 易损耗，单线程，因此使用与从服务器上。

centos 内核相关参数（/etc/sysctl.conf）

innodb 在 5.7 之后支持全文索引和空间函数。

阻塞和死锁：
阻塞是锁与锁之间的相互等待而产生的，比如一个资源加了互斥锁，一个事务的访问的同时其他事务需要阻塞等待该事务释放锁。
死锁：死锁是因为多个事务之间相互占用对方的资源而产生的。

Myisam 存储引擎：
索引在内存当中，数据在 OS 中。
InnoDB 存储引擎：
索引和数据都在内存当中。
磁盘的配置和选择：
1.使用传统机器硬盘。
2.使用 RAID 增强传统机器硬盘的性能。
3.使用固态存储 SSD 和 PCIe 卡。
4.使用网络存储 NAS 和 SAN。

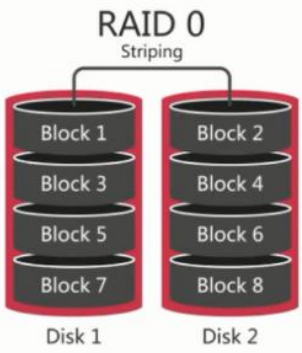
传统机器硬盘取数据的过程：
1.移动磁头到磁盘表面上的正确位置。
2.等待磁盘旋转，使的所需的数据在磁头之下。
3.等待磁盘旋转过去，所有所需的数据都被磁头读出。

使用 RAID 增加传统机器硬盘的性能：
RAID：全称磁盘冗余队列的简称
简单来说 RAID 的作用就是可以把多个容量较小的磁盘组成一组容量更大的磁盘，并提供数据冗余来保证数据完整性的技术。

RAID0 是最早出现的 RAID 模式，也称为数据条带。是组建磁盘阵列中最简单的一种形式，只需要 2 块以上的磁盘即可，成本低，可以提高整个磁盘的性能和吞吐量。RAID0 没有提供冗余或错误修复能力，但是实现成本是最低的。

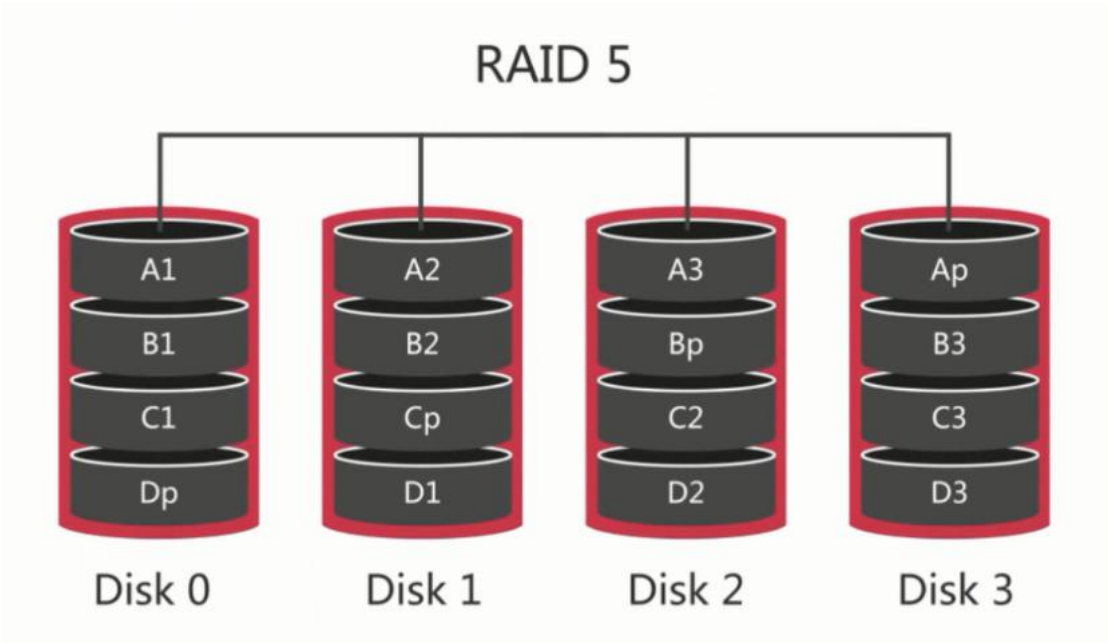
常用的RAID组别——RAID 0

RAID 0是最早出现的RAID模式，也称之为数据条带。是组建磁盘阵列中最简单的一种形式，只需要2块以上的硬盘即可，成本低，可以提高整个磁盘的性能和吞吐量。RAID 0没有提供冗余或错误修复能力，但是实现成本是最低的

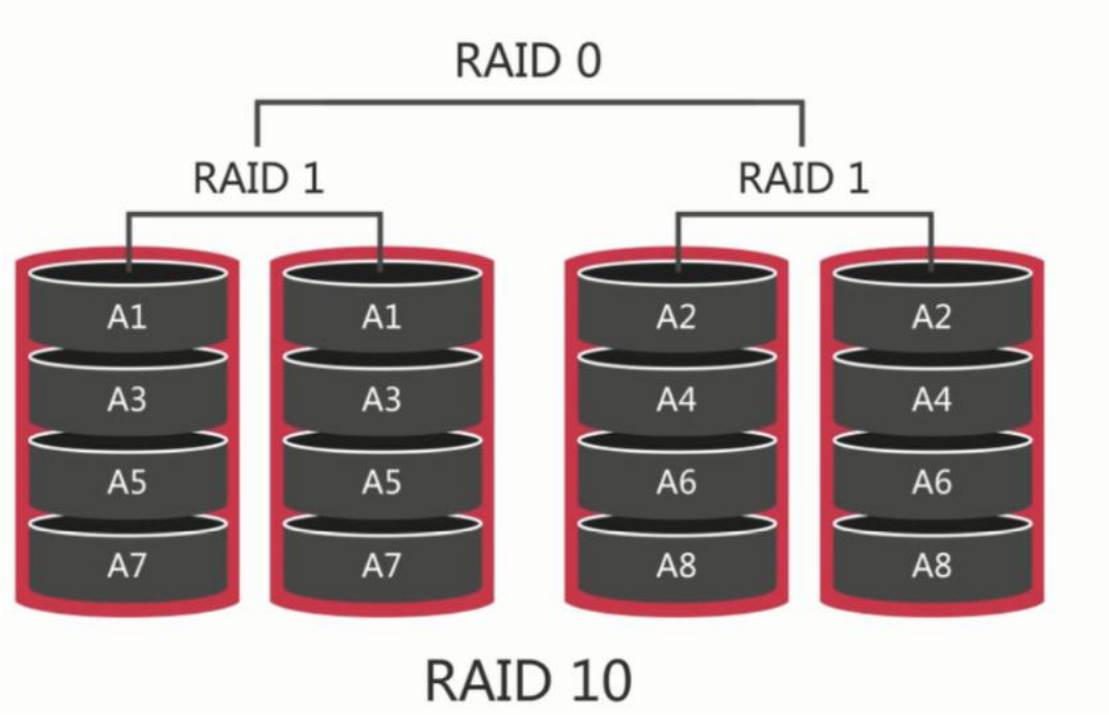


RAID1 又称磁盘镜像，原理是把一个磁盘的数据镜像到另一个磁盘上，也就是说数据在写入一块磁盘的同时，会在另一块闲置的磁盘上生成镜像文件，在不影响性能情况下最大限度的保证系统的可靠性和可修复性。

RAID5 又称为分布式奇偶校验磁盘阵列。通过分布式奇偶校验把数据分散到多个磁盘上，这样如果任何一个盘数据失效，都可以从奇偶校验块中重建。但是如果两块磁盘失效，则整个卷的数据都无法恢复。



RAID10 又称分片的镜像。它是对磁盘先做 RAID1 之后对两组 RAID1 的磁盘再做 RAID0，所以对读写都有良好的性能，相对于 RAID5 重建起来更简单，速度也更快。



服务器硬件对性能的影响

CPU

- 64位的CPU一定要工作在64位的系统下
- 对于并发比较高的场景CPU的数量比频率重要
- 对于CPU密集性场景和复杂SQL则频率越高越好

CentOS系统参数优化

内核相关参数 (/etc/sysctl.conf)

net.core.somaxconn=65535

net.core.netdev_max_backlog=65535

net.ipv4.tcp_max_syn_backlog=65535

CentOS系统参数优化

内核相关参数 (/etc/sysctl.conf)

net.ipv4.tcp_fin_timeout = 10

net.ipv4.tcp_tw_reuse = 1

net.ipv4.tcp_tw_recycle = 1

CentOS系统参数优化

内核相关参数 (/etc/sysctl.conf)

net.core.wmem_default =87380

net.core.wmem_max =16777216

net.core.rmem_default =87380

net.core.rmem_max =16777216

CentOS系统参数优化

内核相关参数 (/etc/sysctl.conf)

net.ipv4.tcp_keepalive_time = 120

net.ipv4.tcp_keepalive_intvl = 30

net.ipv4.tcp_keepalive_probes = 3

内核相关参数 (/etc/sysctl.conf)

kernel.shmmax = 4294967295

Linux内核参数中最重要的参数之一，用于定义单个共享内存段的最大值。

CentOS系统参数优化

内核相关参数 (/etc/sysctl.conf)

kernel.shmmax = 4294967295

注意：1.这个参数应该设置的足够大，以便能在一个共享内存段下容纳下整个的Innodb缓冲池的大小。

CentOS系统参数优化

内核相关参数 (/etc/sysctl.conf)

kernel.shmmax = 4294967295

注意：2.这个值的大小对于64位linux系统，可取的最大值为物理内存值-1byte，建议值为大于物理内存的一半，一般取值大于Innodb 缓着冲池的大小即可，可以取物理内存-1byte。

CentOS系统参数优化

内核相关参数 (/etc/sysctl.conf)

kernel.shmmax = 4294967295

vm.swappiness = 0

Linux系统内存交换区：

如果我们使用free-m在系统中查看可以看到类似下面的内容其中swap就是交换分区。

CentOS系统参数优化

内核相关参数 (/etc/sysctl.conf)

kernel.shmmax = 4294967295

vm.swappiness = 0

当操作系统因为没有足够的内存时就会将一些虚拟内存写到磁盘的交换区中这样就会发生内存交换。

CentOS系统参数优化

增加资源限制 (/etc/security/limit.conf)

* soft nofile 65535
* hard nofile 65535

} 加到limit.conf文件的末尾就可以了

*	表示对所有用户有效
soft	指的是当前系统生效的设置
hard	表明系统中所能设定的最大值
nofile	表示所限制的资源是打开文件的最大数目
65535	就是限制的数量

CentOS系统参数优化

磁盘调度策略 (/sys/block/devname/queue/scheduler)

```
cat /sys/block/sda/queue/scheduler
noop anticipatory deadline [cfq]
```

CentOS系统参数优化

磁盘调度策略 (/sys/block/devname/queue/scheduler)

noop(电梯式调度策略)

NOOP实现了一个FIFO队列，它像电梯的工作方法一样对I/O请求进行组织，当有一个新的请求到来时，它将请求合并到最近的请求之后,以此来保证请求同一介质。NOOP倾向饿死读而利于写，因此NOOP对于闪存设备、RAM及嵌入式系统是最好的选择。

CentOS系统参数优化

磁盘调度策略 (/sys/block/devname/queue/scheduler)

deadline (截止时间调度策略)

Deadline确保了在一个截止时间内服务请求，这个截止时间是可调整的，而默认读期限短于写期限。这样就防止了写操作因为不能被读取而饿死的现象，Deadline对数据库类应用是最好的选择。

CentOS系统参数优化

磁盘调度策略 (/sys/block/devname/queue/scheduler)

anticipatory (预料I/O调度策略)

本质上与Deadline一样，但在最后一次读操作后，要等待6ms，才能继续进行对其它I/O请求进行调度。它会在每个6ms中插入新的I/O操作，而会将一些小写入流合并成一个大写入流，用写入延时换取最大的写入吞吐量。AS适合于写入较多的环境，比如文件服务器 ，AS对数据库环境表现很差。

CentOS系统参数优化

磁盘调度策略 (/sys/block/devname/queue/scheduler)

echo deadline > /sys/block/sda/queue/scheduler

echo <schedulername > /sys/block/devname/queue/scheduler
如 echo deadline > /sys/block/sda/queue/scheduler

文件系统对性能的影响

EXT3/4系统的挂载参数 (/etc/fstab)

data=writeback | ordered | journal

noatime，nodiratime

/dev/sda1/ext4　noatime,nodiratime,data=writeback 1 1

MySQL常用存储引擎之MyISAM

MySQL5.5之前版本默认存储引擎

临时表

在排序、分组等操作中，当数量超过一定的大小之后，由查询优化器建立的临时表

MySQL常用存储引擎之MyISAM

特性

- 并发性与锁级别
- 表损坏修复

check table tablename

repair table tablename

MySQL常用存储引擎之MyISAM

特性

- MyISAM表支持的索引类型

支持全文索引，也支持对text,blob字段的前500个字符建立
前缀索引。

- MyISAM表支持数据压缩

命令行：myisampack

MySQL常用存储引擎之MyISAM

限制

版本< MySQL5.0时默认表大小为4G

如存储大表则要修改MAX_Rows 和 AVG_ROW_LENGTH

MySQL常用存储引擎之MyISAM

限制

版本< MySQL5.0时默认表大小为4G

如存储大表则要修改MAX_Rows 和 AVG_ROW_LENGTH

版本> MySQL5.0时默认支持为256TB

MySQL常用存储引擎之MyISAM

适用场景：

- 非事务型应用
- 只读类应用
- 空间类应用

MySQL常用存储引擎之Innodb

Innodb使用表空间进行 数据存储

innodb_file_per_table

ON:独立表空间:tablename. ibd

OFF:系统表空间:ibdataX

MySQL常用存储引擎之Innodb

系统表空间和独立表空间要如何选择

比较：

- 系统表空间无法简单的收缩文件大小
- 独立表空间可以通过optimize table命令收缩系统文件

MySQL常用存储引擎之Innodb

系统表空间和独立表空间要如何选择

比较：

- 系统表空间会产生IO瓶颈
- 独立表空间可以同时向多个文件刷新数据

MySQL常用存储引擎之Innodb

系统表空间和独立表空间要如何选择

建议：

- 对Innodb 使用独立表空间

表转移的步骤

把原来存在于系统表空间中的表转移到独立表空间中的方法

步骤：

1.使用mysqldump导出所有数据库表数据

2.停止MySQL服务，修改参数，并删除Innodb相关文件

3.重启MySQL服务，重建Innodb系统表空间

4.重新导入数据

MySQL常用存储引擎之Innodb

系统表空间和独立表空间要如何选择

Innodb 数据字典信息

Undo 回滚段

MySQL常用存储引擎之Innodb

Innodb存储引擎的特性

- Innodb是一种事务性存储引擎
- 完全支持事务的ACID特性
- Redo Log 和 Undo Log

持久日志　　回滚日子

```
mysql> show variables like 'innodb_log_buffer_size';
+------------------------+----------+
| Variable_name          | Value    |
+------------------------+----------+
| innodb_log_buffer_size | 16777216 |
+------------------------+----------+
1 row in set (0.00 sec)
```

```
mysql> show variables like 'innodb_log_files_in_group';
+---------------------------+-------+
| Variable_name             | Value |
+---------------------------+-------+
| innodb_log_files_in_group | 2     |
+---------------------------+-------+
1 row in set (0.00 sec)
```

MySQL常用存储引擎之Innodb

Innodb存储引擎的特性

- Innodb支持行级锁
- 行级锁可以最大程度的支持并发
- 行级锁是由存储引擎层实现的

MySQL常用存储引擎之Innodb

什么是锁

- 锁对主要作用是管理共享资源的并发访问
- 锁用于实现事务的隔离性

MySQL常用存储引擎之Innodb

锁的类型

- 共享锁（也称读锁）
- 独占锁（也称写锁）

	写锁	读锁
写锁	不兼容	不兼容
读锁	不兼容	兼容

```
mysql> lock table myinnodb write;
Query OK, 0 rows affected (0.00 sec)
```

MySQL常用存储引擎之Innodb

Innodb状态检查

show engine innodb status

MySQL常用存储引擎之CSV

文件系统存储特点

数据以文本方式存储在文件中

.CSV文件存储表内容

.CSM文件存储表的元数据如表状态和数据量

.frm文件存储表结构信息

MySQL常用存储引擎之CSV

特点

- 以CSV格式进行数据存储
- 所有列必须都是不能为NULL的
- 不支持索引
- 可以对数据文件直接编辑
  保存文本文件内容

```
1,"aaa","bbb"
2,"ccc","ddd"
3,"eee","fff"
```

MySQL常用存储引擎之CSV

适用场景

适合做为数据交换的中间表

MySQL常用存储引擎之Archive

文件系统存储特点

- 以zlib对表数据进行压缩,磁盘I/O更少
- 数据存储在ARZ为后缀的文件中

MySQL常用存储引擎之Archive

Archive存储引擎的特点

- 只支持insert 和 select 操作
- 只允许在自增ID列上加索引

MySQL常用存储引擎之Memory

文件系统存储特点

也称HEAP存储引擎，所以数据保存在内存中

MySQL常用存储引擎之Memory

功能特点

- 支持HASH索引和BTree索引
- 所有字段都为固定长度 varchar(10)=char(10)
- 不支持BLOG和TEXT等大字段

MySQL常用存储引擎之Memory

功能特点

- Memory存储引擎使用表级锁
- 最大大小由max_heap_table_size参数决定

默认16MB，对已经存在的表是不生效的

```
mysql> create index idx_c1 on mymemory(c1);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> create index idx_c2 using btree on mymemory(c2);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

MySQL常用存储引擎之Memory

容易混淆的概念

Memory存储引擎表

VS

临时表
- 系统使用临时表
  - 超过限制使用Myisam临时表
  - 未超限制使用Memory表
- create temporary teble 建立的临时表

MySQL常用存储引擎之Memory

使用场景

- 用于查找或者是映射表，例如邮编和地区的对应表
- 用于保存数据分析中产生的中间表
- 用于缓存周期性聚合数据的结果表

Memory数据易丢失，所以要求数据可再生

MySQL常用存储引擎之 Federated

特点

- 提供了访问远程MySQL服务器上表的方法
- 本地不存储数据，数据全部放到远程服务器上
- 本地需要保存表结构和远程服务器的连接信息

MySQL常用存储引擎之 Federated

如何使用

默认禁止，启用需要在启动时增加federated参数

mysql://user_name[:password]@host_name[:port_num]/db_name/tbl_name

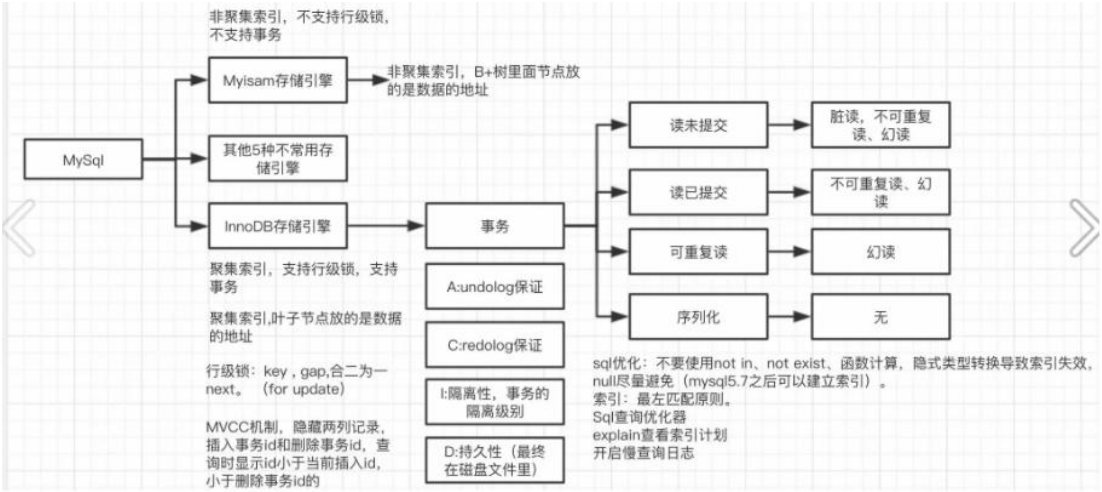
```
mysql> grant select,update,insert,delete on remote.remote_fed to fred_link@'127
.0.0.1' identified by '123456';
Query OK, 0 rows affected (0.13 sec)
```

```
mysql> use local
Database changed
mysql> CREATE TABLE `remote_fed` (
    ->   `id` int(11) NOT NULL AUTO_INCREMENT,
    ->   `c1` varchar(10) NOT NULL DEFAULT '',
    ->   `c2` char(10) NOT NULL DEFAULT '',
    ->   PRIMARY KEY (`id`)
    -> ) engine=federated connection='mysql://fred_link:123456@127.0.0.1:3306/r
emote/remote_fed';
Query OK, 0 rows affected (0.02 sec)
```

MySQL常用存储引擎之 Federated

使用场景

偶尔的统计分析及手工查询



MySQL服务器参数

MySQL获取配置信息路径

- 命令行参数
  mysqld_safe --datadir=/data/sql_data
- 配置文件
  mysqld --help --verbose | grep -A 1 'Default options'
  /etc/my.cnf /etc/mysql/my.cnf /home/mysql/my.cnf
  ~/.my.cnf

MySQL服务器参数

MySQL配置参数的做用域

- 全局参数

  set global 参数名=参数值;

  set @@global.参数名 :=参数值;

- 会话参数

  set [session] 参数名=参数值;

  set @@session.参数名 := 参数值;

MySQL服务器参数

内存配置相关参数

- 确定可以使用的内存的上限
- 确定MySQL的每个连接使用的内存

  sort_buffer_size　　join_buffer_size

  read_buffer_size　　read_rnd_buffer_size

MySQL服务器参数

如何为缓存池分配内存

Innodb_buffer_pool_size

总内存-（每个线程所需要的内存*连接数)-系统保留内存

key_buffer_size

```
select sum(index_length)
from information_schema.tables where engine='myisam'
```

MySQL服务器参数

Innodb I/O相关配置

- Innodb_log_file_size
- Innodb_log_files_in_group
- 事务日志总大小 =
  Innodb_log_files_in_group*Innodb_log_file_size
- Innodb_log_buffer_size

MySQL服务器参数

Innodb I/O相关配置

- Innodb_flush_log_at_trx_commit

  0:每秒进行一次log写入cache，并flush log到磁盘
  1[默认]: 在每次事务提交执行log写入cache，并
  flush log到磁盘
  2[建议]: 每次事务提交,执行log数据写入到cache，每
  秒执行一次flush log到磁盘

MySQL服务器参数

Innodb I/O相关配置

- Innodb_flush_method=O_DIRECT
- Innodb_file_per_table = 1
- Innodb_doublewrite = 1

MySQL服务器参数

MyISAM I/O相关配置

delay_key_write

OFF:每次写操作后刷新键缓冲中的脏块到磁盘

ON:只对在键表时指定了delay_key_write选项的表使用延迟刷新

ALL:对所有MYISAM表都使用延迟键写入

MySQL服务器参数

安全相关配置参数

expire_logs_days 指定自动清理binlog的天数

max_allowed_packet 控制MySQL可以接收的包的大小

skip_name_resolve 禁用DNS查找

MySQL服务器参数

安全相关配置参数

sysdate_is_now 确保sysdate()返回确定性日期

read_only 禁止非super权限的用户写权限

skip_slave_start 禁用Slave自动恢复

MySQL服务器参数

安全相关配置参数

sql_mode 设置MySQL所使用的SQL模式

- strict_trans_tables
- no_engine_subtitution
- no_zero_date
- no_zero_in_date
- only_full_group_by

MySQL服务器参数

其它常用配置参数

- sync_binlog 控制MySQL如何向磁盘刷新binlog
- tmp_table_size 和 max_heap_table_size
  控制内存临时表大小
- max_connections 控制允许的最大连接数

什么影响了性能

数据库设计对性能的影响

- 过分的反范式化为表建立太多的列
- 过分的范式化造成太多的表关联
- 在OLTP环境中使用不恰当的分区表
- 使用外键保证数据的完整性