# 【IO】IO系统性能之一：衡量性能的几个指标

Linux操作系统　作者：xysoul_云龙　时间：2021-12-30 09:39:00　2175　0

作为一个数据库管理员，关注系统的性能是日常最重要的工作之一，而在所关注的各方面的性能中， IO性能却是最令人头痛的一块，面对着各种生涩的参数和令人眼花缭乱的新奇术语，再加上存储厂商的忽悠，总是让我们有种云里雾里的感觉。本系列文章试图从基本概念开始对磁盘存储相关的各种概念进行综合归纳，让大家能够对IO性能相关的基本概念，IO性能的监控和调整有个比较全面的了解。

在这一部分里我们先舍弃各种结构复杂的存储系统，直接研究一个单独的磁盘的性能问题，藉此了解衡量IO系统性能的各个指标以及之间的关系。需要注意的是，本文探讨的仅限于磁盘IO性能，网络IO性能不考虑在内

## 1. 几个基本概念

在研究磁盘性能之前，我们必须先了解磁盘的结构，以及工作原理。不过在这里就不再重复说明了，关于 磁盘结构 和 工作原理 的信息可以参考维基百科上面的相关词条————Hard disk drive(英文)和硬盘驱动器(中文)

### 1） 读写IO(Read/Write IO)操作

磁盘是用来给我们存取数据用的，因此当说到IO操作的时候，就会存在两种相对应的操作， 存数据 时候对应的是写IO操作， 取数据 时候对应的是读IO操作。

### 2）单个IO操作

当控制磁盘的控制器接到操作系统的 读IO操作 指令的时候，控制器就会给磁盘发出一个读数据的指令，并同时将要读取的数据块的地址传递给磁盘，然后磁盘会将读取到的数据传给控制器，并由控制器返回给操作系统，完成一个读IO的操作。同样的，一个写IO的操作也类似，控制器接到 写IO操作 的指令和要写入的数据，并将其传递给磁盘，磁盘在数据写入完成之后将操作结果传递回控制器，再由控制器返回给操作系统，完成一个写IO的操作。单个IO操作指的就是完成一个写IO或者读IO的操作。

### 3）随机访问(Random Access)与顺序访问(Sequential Access)

随机访问指的是本次IO所给出的扇区地址和上次IO给出的扇区地址相差比较大，这样的话磁头在两次IO操作之间需要作比较大的移动操作才能重新开始读/写数据。相反的，如果当次IO给出的扇区地址与上次IO结束时的扇区地址一致或者是接近的话，那么磁头就能很快的开始这次IO操作，这样的多个IO操作称为连续访问。因此，尽管相邻的两次IO操作在同一时刻发出，但如果他们请求的扇区地址相差很大的话也只能称为随机访问，而非顺序访问。

4）**顺序IO模式(Queue Mode)与并发IO模式(Burst Mode)**

磁盘控制器可能会一次对磁盘组发出一连串的IO指令，如果磁盘组一次只能执行一个IO指令时称为顺序IO; 当磁盘组能同时执行多个IO指令时，称为并发IO。并发IO只能发生在由多个磁盘组成的磁盘上，单块磁盘只能一次处理一个IO命令。

## 2. 单个IO的大小(IO Chunk Size)

熟悉数据库的人都会有这么一个概念，那就是数据库存储有个基本的块大小(Block Size)，不管是SQL Server还是Oracle，默认的块大小都是8KB，就是数据库每次读写都是以8KB为单位来进行的。那么对于数据库应用发出的固定8KB大小的单次读写，到了磁盘这一层面会是怎么样的情况呢？ 就是对于读写磁盘来说，单个IO操作操作数据的大小是多少呢， 是不是也是一个固定的值?

答案是不确定。首先操作系统为了提高IO的性能而引入了文件系统缓存(File System Cache)，系统会根据请求数据的情况将多个来自IO的请求先放在缓存里面，然后再一次性的提交给磁盘，也就是说对于数据库发出的多个8K数据块的读操作，有可能放在一个磁盘读IO里就处理了。

还有对于有些存储系统也是提供了缓存(Cache)的，接收到操作系统的IO请求之后，也是会将多个操作系统的IO请求合并成一个来处理。不管是操作系统层面的缓存还是磁盘控制器层面的缓存，目的都只有一个，提高数据读写的效率。因此，每次单独的IO操作大小都是不一样的，它主要取决于系统对于数据读写效率的判断。

当一次IO操作大小比较小的时候，我们称为小的IO操作。比如说1K、4K、8K这样的；当一次IO操作的数据量比较大的时候，我们称为大IO操作，比如32K、64K甚至更大。

在我们说到块大小(Block Size)的时候，通常我们会接触到多个类似的概念，像我们上面提到的那个在数据库里面的数据最小管理单元， Oracle称之为块(Block)，大小一般为8K， SQL Server称之为页(Page)，一般大小也为8K。

在文件系统里面，我们也能碰到一个文件系统的块(Block)。在现在很多的Linux系统中都是4K，它的作用其实跟数据库里面的块/页是一样的，都是为了方面数据管理。但是说道单次IO的大小，跟这些块的大小都是没有直接关系的，在英文里单次IO大

小通常被称为IO Chunk Size，不会说成IO Block Size的。

## 3. IOPS(IO per Second)

IOPS是指IO系统每秒所执行IO操作的次数，是一个重要的用来衡量系统IO能力的一个参数。对于单个磁盘组成的IO系统来说，计算它的IOPS不是一件很难的事情，只要我们知道了系统完成一次IO所需要的时间的话，我们就能推算出系统IOPS来。

现在我们就来推算一下磁盘的IOPS，假设磁盘的转速(Rotational Speed)为15K RPM，平均寻道时间为5ms，最大传输速率为40MB/s（这里将读写速度视为一样，实际会差别比较大）。

对于磁盘来说一个完整的IO操作是这样进行的： 当控制器对磁盘发出一个IO操作命令的时候，磁盘的驱动臂(Actuator Arm)带读写磁头(Head)离开着陆区(Landing Zone，位于内圈没有数据的区域)，移动到要操作的初始数据块所在的磁道(Track)的正上方，这个过程被称为寻道(Seeking),对应消耗的时间被称为 `寻道时间`(Seek Time)； 但是找到对应磁道还不能马上读取数据，这时候磁头要等到磁盘盘片(Platter)旋转到初始数据块所在扇区(Sector)落在读写磁头正上方之后才能开始读写数据，在这个等待盘片旋转到可操作扇区的过程中消耗的时间称为 `旋转延时`(Rotational Delay); 接下来就随着盘片的旋转，磁头不断的读/写相应的数据块，直到完成这次IO所需要操作的全部数据，这个过程称为数据传送(Data Transfer)，对应的时间称为 `传送时间`(Transfer Time)。完成这三个步骤之后，一次IO操作也就完成了。

```
Total = 寻道时间 + 旋转延迟 + 传送时间
```

在我们看硬盘厂商的宣传单的时候，我们经常能看到3个参数，分别是平均寻址时间、盘片旋转速度以及最大传送速度，这三个参数就可以提供给我们计算上述三个步骤的时间。

1） 第一个寻址时间，考虑到被读写的数据可能在磁盘的任意一个磁道，既有可能在磁盘的最内圈(寻址时间最短），也可能在磁盘的最外圈（寻址时间最长），所以在计算中我们只考虑平均寻址时间，也就是磁盘参数中标明的那个平均 `寻址时间`,这里就采用当前最多的 `15K RPM` 硬盘的5ms。

2) 第二个旋转延时，和寻址一样，当磁头定位到磁道之后有可能正好在要读写扇区之上，这时候是不需要额外延时就可以立刻读写到数据，但是最坏的情况却是要磁盘旋转整整一圈之后磁头才能读取到数据，所以这里我们也考虑的是平均旋转延时，对于 `15K RPM` 的磁盘就是(60s/15K)/2 = 2ms。(磁盘转速单位一般为： 转/每分钟)

3） 第三个传送时间，磁盘参数提供给我们的最大传输速度，当然要达到这种速度是很有难度的，但是这个速度却是 `纯读写磁盘` 的速度，因此只要给定了单次IO的大小，我们就知道磁盘需要花费多少时间在数据传送上，这个时间就是IO Chunk Size /Max Transfer Rate。

# 4. IOPS计算公式

现在我们就可以得出这样的计算单次IO时间的公式：

```
IO Time = Seek Time + 60 sec/Rotational Speed /2 + IO Chunk Size / Transfer Rate
```

于是我们可以这样计算出：

```
IOPS = 1/IO Time = 1/(Seek Time + 60 sec / Rotational Speed / 2 + IO Chunk Size / Transfer Rate)
```

> 注：IOPS全称为Input/Output Operations Per Second，大意是硬盘每秒的读写次数。一个硬盘的随机读取IOPS主要由其主控和接口决定。在测试硬盘随机读取性能上，大部分软件会使用4KB大小的数据区块作为测试基准。以希捷酷鱼510系列固态硬盘为例，由于其搭载的主控支持NVMe 1.3协议，所以在4KB随机读取上可达到100096 IOPS，写入上也高达896000 IOPS。

对于给定不同的IO大小，我们可以得出下面的一系列数据：

1) **IO大小为4K**

```
4K (1/7.1 ms = 140 IOPS) 5ms + (60sec/15000RPM/2) + 4K/40MB = 5 + 2 + 0.1 = 7.1
```

2) **IO大小为8K**

```
8k (1/7.2 ms = 139 IOPS) 5ms + (60sec/15000RPM/2) + 8K/40MB = 5 + 2 + 0.2 = 7.2
```

3) **IO大小为16K**

```
16K (1/7.4 ms = 135 IOPS) 5ms + (60sec/15000RPM/2) + 16K/40MB = 5 + 2 + 0.4 = 7.4
```

4) **IO大小为32K**

```
32K (1/7.8 ms = 128 IOPS) 5ms + (60sec/15000RPM/2) + 32K/40MB = 5 + 2 + 0.8 = 7.8
```

5) **IO大小为64K**

```
64K (1/8.6 ms = 116 IOPS) 5ms + (60sec/15000RPM/2) + 64K/40MB = 5 + 2 + 1.6 = 8.6
```

从上面的数据可以看出，当单次IO越小的时候，单次IO所耗费的时间也越少，相应的IOPS也就越大。

上面我们的数据都是在一个比较理想的假设下得出来的，这里的理想情况就是磁盘要花费平均大小的 `寻址时间` 和平均的 `旋转延时`，这个假设其实是比较符合我们实际情况中的随机读写。在随机读写中，每次IO操作的寻址时间和旋转延时都不能忽

略不计，有了这两个时间的存在，也就限制了IOPS的大小。现在我们考虑一种相对极端的顺序读写操作，比如说在读取一个很大的存储连续分布在磁盘上的文件，因为文件的存储分布是连续的，磁头在完成一个读IO操作之后，不需要重新的寻址，也不需要旋转延时，在这种情况下我们能得到一个很大的IOPS值，如下：

```
4K (1/0.1 ms = 10000 IOPS) 0ms + 0ms + 4K/40MB = 0.1 8k (1/0.2 ms = 5000 IOPS) 0ms + 0ms + 8K/40MB = 0.2 16K (1/0.4 ms = 2500 IOPS) 0ms + 0ms + 16K/40MB = 0.4 32K (1/0.8 ms = 1250 IOPS) 0ms + 0ms + 32K/40MB = 0.8 64K (1/1.6 ms = 625 IOPS) 0ms + 0ms + 64K/40MB = 1.6
```

相比第一组数据来说差距是非常大的，因此当我们要用IOPS来衡量一个IO系统性能的时候，我们一定要说清楚是在什么情况下的IOPS，也就是要说明读写的方式 以及单次IO的大小，当然在实际当中，特别是在OLTP的系统，随机的小IO读写是最有说服力的。

下面给出一个目前机械硬盘与固态硬盘的读写速度的一个参考：

1） **机械硬盘**

- 5400转笔记本硬盘平均读写速度大致在60-90MB这个区间
- 7200转台式机硬盘大致在130-190MB区间，10000转的西数黑盘也在这个区间内
- 10000转和15000转台式机硬盘数据不详

2） **固态硬盘**

固态硬盘读写速度与容量成正比，目前市售的至少300MB+

1TB固态硬盘普遍500MB+

2013新Mac Pro采用PCIE连接方式的SSD可以达到700MB左右

# 5. 传输速度/吞吐率

现在我们要说的传输速度（另一个常见的说法是吞吐率）不是磁盘上所表明的最大传输速度或者说理想传输速度，而是磁盘在实际使用的时候从磁盘系统总线上流过的数据量。有了IOPS数据之后，我们是很容易就能计算出对应的传输速度来的。

```
Transfer Rate = IOPS * IO Chunk Size
```

还是上面的那一组IOPS数据，我们可以得出相应的传输速度如下：

```
4K: 140 * 4K = 560K / 40M = 1.36% 8K: 139 * 8K = 1112K / 40M = 2.71% 16K: 135 * 16K = 2160K / 40M = 5.27% 32K: 116 * 32K = 3712K / 40M = 9.06%
```

从上面可以看出，实际上的传输速度是很小的，对总线的利用率也是非常的小。

这里一定要明确一个概念，那就是尽管上面我们使用IOPS来计算传输速度，但是实际上传输速度和IOPS是没有直接关系的。在没有缓存的情况下，它们共同的决定因素都是对磁盘系统的 访问方式 以及 单个IO大小 。对磁盘进行随机访问的时候，我们可以利用IOPS来衡量一个磁盘系统的性能，此时的传输速度不会太大； 但是当对磁盘进行连续访问时，此时的IOPS已经没有了参考的价值，这个时候限制实际传输速度却是磁盘的最大传输速度。因此在实际的应用当中，只会用IOPS来衡量小IO的随机读写性能，而当要衡量大IO连续读写的性能的时候，就要采用传输速度而不能是IOPS了。

## 6. IO响应时间

最后来关注一下能直接描述IO性能的IO响应时间。IO响应时间也被称为IO延时(IO Latency)。IO响应时间就是从操作系统内核发出的一个 读 或者 写 的IO命令到操作系统内核接收到IO回应的时间，注意不要和单个IO时间混淆了。单个IO时间仅仅指的是IO操作在磁盘内部处理的时间，而IO响应时间还要包括IO操作在IO等待队列中所花费的等待时间。

计算IO操作在等待队列里面消耗的时间有一个衍生于 利托氏定理 (Little's Law)的排队模型 M/M/1模型 可以遵循，由于排队模型算法比较复杂，到现在还没有搞太明白(如果有谁对 M/M/1模型 比较精通的话，欢迎给予指导），这里就罗列一下最后的结果，还是上面计算的IOPS数据来说：

```
8K IO Chunk Size (135 IOPS, 7.2 ms)
135 => 240.0 ms
105 => 29.5 ms
75 => 15.7 ms
45 => 10.6 ms
64K IO Chunk Size(116 IOPS, 8.6 ms)
135 => 没响应了……
105 => 88.6 ms
75 => 24.6 ms
45 => 14.6 ms
```

从上面的数据可以看出，随着系统实际IOPS越接近理论的最大值，IO响应时间会成非线性的增长，越是接近最大值，响应时间就变得越大，而且会比预期超出很多。一般来说，在实际的应用中有一个70%的指导值，也就是说在IO读写的队列中，当队列大小小于最大IOPS的70%的时候，IO的响应时间增加会很小，相对来说让人比较能接受，一旦超过70%，响应时间就会戏剧性的暴增，所以当一个系统的IO压力超出最大可承受压力的70%的时候就是必须要考虑调整或升级了。

另外补充说一下这个70%的指导值也适用于CPU响应时间，这也是在实践中证明过得。一旦CPU超过70%，系统将会变得受不了的慢。很有意思的东西。

# 7. 附: Linux内核文件系统block与硬盘sector关系

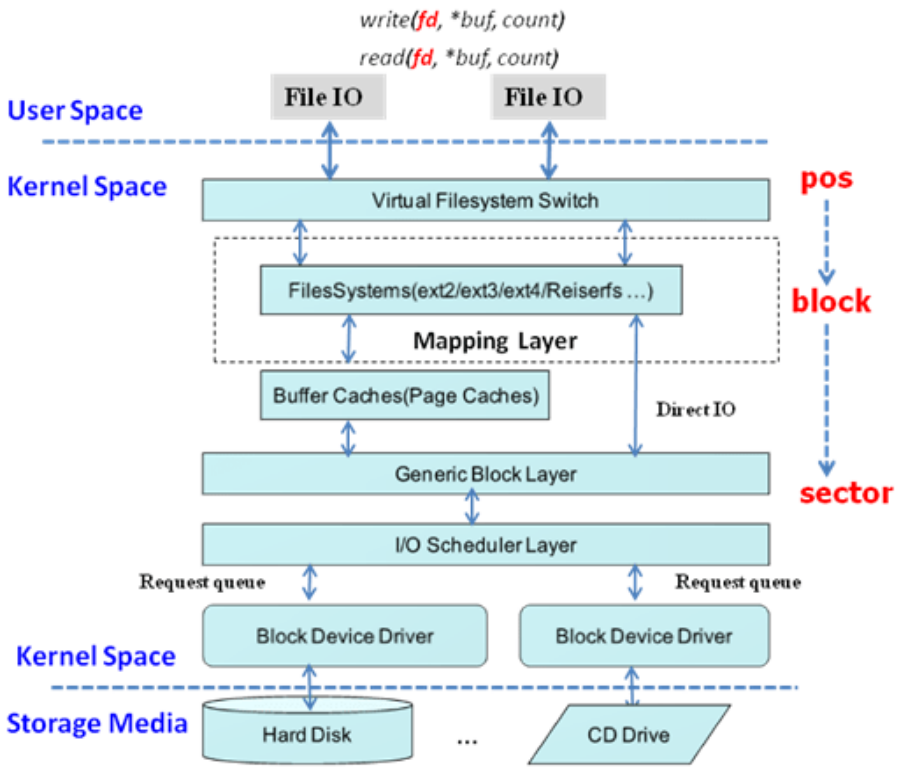
在系统运行过程中，有时会遇到下面打印信息，报告读写某个扇区错误

```
kernel: end_request: I/O error, dev sdg, sector 2252148039 kernel: end_request: I/O error, dev sdc, sector 3297222879
```

1、这个扇区（sector）的含义是什么？和硬盘上的sector是一回事吗？

2、Sector和文件系统中的Block有什么关系？

3、而在我们上层应用读写的是文件内偏移量pos，pos与block/Sector之间有什么关系？




文件偏移量pos，是针对文件本身而言，即文件内的偏移。

Block是文件系统上的概念，一般文件系统block大小为4K。

Sector是磁介质硬盘最小单元，一般为512字节。

Block值一般与sector值是不相等的.

**[参看]:**

1. IO系统性能之一：衡量性能的几个指标
2. 如何让linux服务器磁盘io性能翻倍
3. 磁盘性能评价指标—IOPS和吞吐量