dd命令的conv=fsync,oflag=sync/dsync

# dd

dd命令是一个非常强大的命令，对于一些比较底层的问题，使用dd命令往往可以得到出人意料的效果。我们可以用它来测试磁盘的读写性能。之前一直以为他只能测试块设备，但是今天看到一个文章说他同时是可以测试文件系统的（IOzone也是可以测试文件系统跟块设备，但IOmeter是不能用来测试文件系统的）。

而对于dd命令，我们常用到的两个设备就是 /dev/null /dev/zero ，因为避免覆盖此文主题，所以对该特殊设备 见这里说明：http://blog.csdn.net/menogen/article/details/38060003

dd有有些参数是挺难理解的，今天用了两个小时才弄明白了设置conv=conv=fsync,oflag=sync/dsync,后两者比较好区分，前两者不好区分

我们知道 使用dd来测试硬盘读写速度只能提供一个大概的测试结果，而且是连续IO 而不是随机IO ，理论上文件规模越大，测试结果越准确。理论上bs越大，所测得性能越高

# 如何真正写磁盘

dd if=/dev/zero of=test bs=64k count=16k　这个是不准确的，因为命令结束的时候数据还没有真正写到磁盘上去，因为对磁盘的写，我们一般是先写到了缓存就返回了。

我们来看dd的帮助页面对于一些参数的解释

the FLAG 参数（完整的看手册哦，这里假设你是知道iflag跟oflag的）

-dsync  
   Use synchronized I/O for data. For the output file, this forces a physical write of output data on each write. For the input file, this flag can matter when reading from a remote file that    has been written to synchronously by some other process. Metadata (e.g., last-access and last-modified time) is not necessarily synchronized.

-sync    likewise, but also for metadata

the CONV参数  
   -fsync   
  Synchronize output data and metadata just before finishing. This forces a physical write of output data and metadata

dsync跟sync比较好理解，前者是只同步写数据，sync同时写元数据

但是感觉dsync与 -fsync怎么感觉有些一样？ 网上的说法是  dd if=/dev/zero of=test bs=64k count=4k oflag=dsync　这个可以当成是模拟数据库插入操作，所以很慢，但还是没太明白。

后来自己认真的抠了这英文用词， conv=fsync  Synchronize output data and metadata just before finishing 意思也就是在dd命令结束前同步data和metadata，那就是不是每一次写都同步一次咯，也就是如果我们在dd命令中写了100次，他可能是等到最后的时候才把他们同步到磁盘。而oflag=dsync是说Use synchronized I/O for data. For the output file, this forces a physical write of output data on each write，注意这里边用词  a physical write of output data on each write,那就是他是每一次写都得等到这一次写写到了磁盘才进行下一个写，也就是如果我们使用dd写100次，他每次写都是写到磁盘后才进行下一次写的。所以这样当然要比conv=fsync慢一些吧。那么自己感觉如果只是写一次的话，两者应该是差别不大的，后来做了下小实验，证实确实是这样的。



在第一个图中，我们只写1块，然后使用oflag=sync与conv=fsync 测出来一个是32.1kb/s 一个是37.8kb/s 差别不大。但是下一个我写1000个，conv=fsync就明显的比oflag=dsync/sync快很多了，所以觉得上面自己扣的英文的理解应该是正确的。

所以在用dd做读或者写的时候，应该要注意自己的使用场景，如果需要将数据写入磁盘的话

dd if=/dev/zero of=test bs=64k count=16k  是不准确的，

而 dd if=/dev/zero of=test bs=64k count=16k conv=fsync 比较准备，他在dd结束前会写到磁盘，

而dd if=/dev/zero of=test bs=64k count=4k oflag=dsync或者sync 是真正的每写一次就写一次磁盘，所以其实可以听到磁盘啪啪啪的响的。

# dd如何绕开cache

如果要规避掉文件系统cache,直接读写,不使用buffer cache，需做这样的设置  
iflag=direct,nonblock  
oflag=direct,nonblock  
iflag=cio  
oflag=cio

direct 模式就是把写入请求直接封装成io 指令发到磁盘。

非direct 模式，就把数据写入系统缓存，然后就认为io 成功，并由操作系统决定缓存中的数据什么时候被写入磁盘