<https://baike.baidu.com/item/IO/5918?fr=aladdin>

I/O输入/输出(Input/Output)，分为IO设备和IO接口两个部分。 在POSIX兼容的系统上，例如[Linux](https://baike.baidu.com/item/Linux" \t "_blank)系统，I/O操作可以有多种方式，比如DIO(Direct I/O)，AIO(Asynchronous I/O，异步I/O)，Memory-Mapped I/O(内存映射I/O)等，不同的I/O方式有不同的实现方式和性能，在不同的应用中可以按情况选择不同的I/O方式。

**中文名**

输入/输出

**外文名**

IO

**I/O输入/输出**

分为IO设备和IO接口两个部分

**效    果**

提高缓存

**注意事项**

缓存和磁盘阵列提高存储IO性能

## 目录

1. 1 [简介](https://baike.baidu.com/item/IO/5918?fr=aladdin#1)
2. 2 [安装与接线](https://baike.baidu.com/item/IO/5918?fr=aladdin#2)
3. 3 [提高缓存](https://baike.baidu.com/item/IO/5918?fr=aladdin#3)
4. ▪ [高速缓存(Cache)](https://baike.baidu.com/item/IO/5918?fr=aladdin#3_1)
5. ▪ [RAID0](https://baike.baidu.com/item/IO/5918?fr=aladdin#3_2)

## 简介

[编辑](javascript:;)

I/O输入/输出(Input/Output)，分为IO设备和IO接口两个部分。

在POSIX兼容的系统上，例如Linux系统，I/O操作可以有多种方式，比如DIO(Direct I/O)，AIO(Asynchronous，I/O 异步I/O)，Memory-Mapped I/O(内存映设I/O)等，不同的I/O方式有不同的实现方式和性能，在不同的应用中可以按情况选择不同的I/O方式。

输入输出I/O流可以看成对字节或者包装后的字节的读取就是拿出来放进去双路切换；实现联动控制系统的弱电线路与被控设备的强电线路之间的转接、隔离，以防止强电窜入系统，保障系统的安全；

与专线控制盘连接，用于控制重要消防设备（如消防泵、喷淋泵、风机等），一只模块可控制一台大型消防设备的启、停控制；

插拔式结构，可像安装探测器一样先将底座安装在墙上，布线后工程调试前再将切换模块插入底座。易于施工、维护；

通过无源动合接点或切换AC220V电压作为回答信号。

确认灯动作灯—红色，回答灯—绿色；动作时，动作灯[常亮](https://baike.baidu.com/item/%E5%B8%B8%E4%BA%AE" \t "_blank)、回答灯常亮。

接点负载AC250V/3A、DC30/V7A启动为一组常开/常闭触点、停止为一组常开触点。

## 安装与接线

[编辑](javascript:;)

安装孔距为65mm，用2只M4螺钉或A4自攻钉固定在安装位置。

端子1接多线盘启动端；端子2接多线盘停止端；

端子3接多线盘回答端；端子4接电源地G；

端子5、6为停止命令对应的常开触点输出；

端子11、12接220V回答信号；

端子13、14为启动命令对应的常开触点输出；端子14、15为启动命令对应的常闭触点输出；

触点输出均为无源。

端子16接24V电源正极；

应用（接专线控制盘）

注意事项：可使用AC220V或无源闭合信号作为回答反馈信号。

JBF-151F/D只有1个回答输入，它是启动1的回答。

JBF-151F/D启动发出后可提供一组常开或常闭[触点](https://baike.baidu.com/item/%E8%A7%A6%E7%82%B9" \t "_blank)，停止命令输出时只输出一对常开点。

## 提高缓存

[编辑](javascript:;)

衡量性能的几个指标的计算中我们可以看到一个15k转速的磁盘在随机读写访问的情况下IOPS竟然只有140左右，但在实际应用中我们却能看到很多标有5000IOPS甚至更高的存储系统，有这么大IOPS的存储系统怎么来的呢?这就要归结于各种存储技术的使用了，在这些存储技术中使用最广的就是高速缓存(Cache)和磁盘冗余阵列(RAID)了，本文就将探讨缓存和磁盘阵列提高存储IO性能的方法。

### 高速缓存(Cache)

在当下的各种存储产品中，按照速度从快到慢应该就是内存>闪存>磁盘>磁带了，然而速度越快也就意味着价格越高，闪存虽然说是发展势头很好，磁盘的速度无疑是计算机系统中最大的瓶颈了，所以在必须使用磁盘而又想提高性能的情况下，人们想出了在磁盘中嵌入一块高速的内存用来保存经常访问的数据从而提高读写效率的方法来折中的解决，这块嵌入的内存就被称为高速缓存。

说到缓存，到操作系统层，再到磁盘控制器，还有CPU内部，单个磁盘的内部也都存在缓存，所有这些缓存存在的目的都是相同的，就是提高系统执行的效率。

当然在这里我们只提跟IO性能相关的缓存，与IO性能直接相关的几个缓存分别是文件系统缓存(File [SyST](https://baike.baidu.com/item/SyST" \t "_blank)em Cache)、磁盘控制器缓存(Disk [CON](https://baike.baidu.com/item/CON" \t "_blank)troller Cache)和磁盘缓存(Disk Cache,也称为Disk Buffer)，不过当在计算一个磁盘系统性能的时候文件系统缓存也是不会考虑在内的，我们重点考察的就是磁盘控制器缓存和磁盘缓存。

不管是控制器缓存还是磁盘缓存，他们所起的作用主要是分为三部分：缓存数据、预读(Read-ahead)和回写(Wri[te](https://baike.baidu.com/item/te)-back)。

**缓存数据**

首先是系统读取过的数据会被缓存在高速缓存中，这样下次再次需要读取相同的数据的时候就不用在访问磁盘，直接从缓存中取数据就可以了。当然使用过的数据也不可能在缓存中永久保留的，缓存的数据一般那是采取LRU算法来进行管理，目的是将长时间不用的数据清除出缓存，那些经常被访问的却能一直保留在缓存中，直到缓存被清空。

**预读**

预读是指采用预读算法在没有系统的IO请求的时候事先将数据从磁盘中读入到缓存中，然后在系统发出读IO请求的时候，就会实现去检查看看缓存里面是否存在要读取的数据，如果存在(即命中)的话就直接将结果返回，这时候的磁盘不再需要寻址、旋转等待、读取数据这一序列的操作了，这样是能节省很多时间的;如果没有命中则再发出真正的读取磁盘的命令去取所需要的数据。

缓存的命中率跟缓存的大小有很大的关系，理论上是缓存越大的话，所能缓存的数据也就越多，这样命中率也自然越高，当然缓存不可能太大，毕竟成本在那儿呢。如果一个容量很大的存储系统配备了一个很小的读缓存的话，这时候问题会比较大的，因为小缓存缓存的数据量非常小，相比整个存储系统来说比例非常低，这样随机读取(数据库系统的大多数情况)的时候命中率也自然就很低，这样的缓存不但不能提高效率(因为绝大部分读IO都还要读取磁盘)，反而会因为每次去匹配缓存而浪费时间。

执行读IO操作是读取数据存在于缓存中的数量与全部要读取数据的比值称为缓存命中率(Read Cache Hit Radio)，假设一个存储系统在不使用缓存的情况下随机小IO读取能达到150IOPS，而它的缓存能提供10%的缓存命中率的话，那么实际上它的IOPS可以达到150/(1-10%)=166。

**回写**

要先说一下，用于回写功能的那部分缓存被称为写缓存(Write Cache)。在一套写缓存打开的存储中，操作系统所发出的一系列写IO命令并不会被挨个的执行，这些写IO的命令会先写入缓存中，然后再一次性的将缓存中的修改推到磁盘中，这就相当于将那些相同的多个IO合并成一个，多个连续操作的小IO合并成一个大的IO，还有就是将多个随机的写IO变成一组连续的写IO，这样就能减少磁盘寻址等操作所消耗的时间，大大的提高磁盘写入的效率。

读缓存虽然对效率提高是很明显的，但是它所带来的问题也比较严重，因为缓存和普通内存一样，掉电以后数据会全部丢失，当操作系统发出的写IO命令写入到缓存中后即被认为是写入成功，而实际上数据是没有被真正写入磁盘的，此时如果掉电，缓存中的数据就会永远的丢失了，这个对应用来说是灾难性的，目前解决这个问题最好的方法就是给缓存配备电池了，保证存储掉电之后缓存数据能如数保存下来。

和读一样，写缓存也存在一个写缓存命中率(Write Cache Hit Radio)，不过和读缓存命中情况不一样的是，尽管缓存命中，也不能将实际的IO操作免掉，只是被合并了而已。

控制器缓存和磁盘缓存除了上面的作用之外还承当着其他的作用，比如磁盘缓存有保存IO命令队列的功能，单个的磁盘一次只能处理一个IO命令，但却能接收多个IO命令，这些进入到磁盘而未被处理的命令就保存在缓存中的IO队列中。

**RAID(Redundant [Array](https://baike.baidu.com/item/Array" \t "_blank)Of Inexpensive Disks)**

如果你是一位数据库管理员或者经常接触[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "_blank)，那对RAID应该很熟悉了，作为最廉价的存储[解决方案](https://baike.baidu.com/item/%E8%A7%A3%E5%86%B3%E6%96%B9%E6%A1%88" \t "_blank)，RAID早已在服务器存储中得到了普及。在RAID的各个级别中，应当以RAID10和RAID5(不过RAID5已经基本走到头了，RAID6正在崛起中，看看这里了解下原因)应用最广了。下面将就RAID0，RAID1，RAID5，RAID6，RAID10这几种级别的RAID展开说一下磁盘阵列对于磁盘性能的影响，当然在阅读下面的内容之前你必须对各个级别的RAID的结构和工作原理要熟悉才行，这样才不至于满头雾水，推荐查看wikipedia上面的如下条目：RAID，[Standard](https://baike.baidu.com/item/Standard) RAID levels，Nested RAID levels。

### RAID0

RAID0将数据条带化(striping)将连续的数据分散在多个磁盘上进行存取，系统发出的IO命令(不管读IO和写IO都一样)就可以在磁盘上被并行的执行，每个磁盘单独执行自己的那一部分请求，这样的并行的IO操作能大大的增强整个存储系统的性能。假设一个RAID0阵列有n(n>=2)个磁盘组成，每个磁盘的随机读写的IO能力都达到140的话，那么整个磁盘阵列的IO能力将是140\*n。同时如果在阵列总线的传输能力允许的话RAID0的吞吐率也将是单个磁盘的n倍。

其他**RAID区域**

RAID1镜像磁盘，使用2块硬盘，一般做系统盘的镜像，读IO为一块硬盘的IO，写IO为2块硬盘的IO。

RAID10既能增加IO的读写性能又能实现数据的冗余,使用盘的数量为2的倍数且要大于等于4，且硬盘空间相同，这样的缺点是要实现IO扩展就必须增加相应的硬盘数量，实现同样的性能硬盘成本要成倍增长。允许不同硬盘数据的任何一块丢失。

RAID3拿出单独一块盘做奇偶校验盘，做到数据的冗余

这种情况下允许一块硬盘损坏。由于磁盘的任何数据发生改变都会重新对校验盘进行改写，所以过多的写操作会成为整个系统的瓶颈，此种RAID级别只能用于对读请求相对较高，写请求不多的环境。RAID3已基本淘汰，一般用RAID5技术替代。