Mass Storage Structure

RAID

Liu yufeng

Fx_yfliu@163.com

Hunan University

CRUX: HOW TO MAKE A LARGE, FAST, RELIABLE DISK

How can we make a large, fast, and reliable storage system? What are the key techniques? What are trade-offs between different approaches?

RAID (Redundant Array of Inexpensive Disks)

- **Use multiple disks** in concert to build a **faster**, **bigger**, and more **reliable** disk system.
  - RAID just looks like a big disk to the host system.

- Advantage
  - **Performance** & **Capacity**: Using multiple disks in parallel
  - **Reliability**: RAID can tolerate the loss of a disk.

RAIDs provide these advantages transparently to systems that use them.

RAID Interface

- When a RAID receives I/O request,
  1. The RAID **calculates** which disk to access.
  2. The RAID **issue** one or more **physical I/Os** to do so.

- RAID example: A mirrored RAID system
  - Keep <u>two copies</u> of each block (each one on a separate disk)
  - Perform <u>two physical I/Os</u> for every one logical I/O it is issued.

- Externally, a RAID **looks like a disk**: a group of blocks one can read or write.

RAID Internals

- RAID内部包括:
  - A microcontroller
    - RAID通常包括一个微控制器，运行固件以指导RAID 的操作。
  - Volatile memory (such as DRAM)
    - 包括 DRAM 这样的易失性存储器，在读取和写入时缓冲数据块。
  - Non-volatile memory
    - 在某些情况下，还包 括非易失性存储器，安全地缓冲写入
  - 可能包含专用的逻辑电路，来执行奇偶校验计算

How To Evaluate A RAID

- We evaluate each RAID design along **three axes**.
- 第一个方面是容量（capacity）。在给定一组 N 个磁盘的情况下，RAID 的客户端可用的容量有多少?
- 第二个方面是可靠性（reliability）。给定设计允许有多少磁盘故障?
- 第三个方面是性能（performance）。性能有点难以评估，因为它在很大程度上取决于磁盘阵列提供的工作负载。.

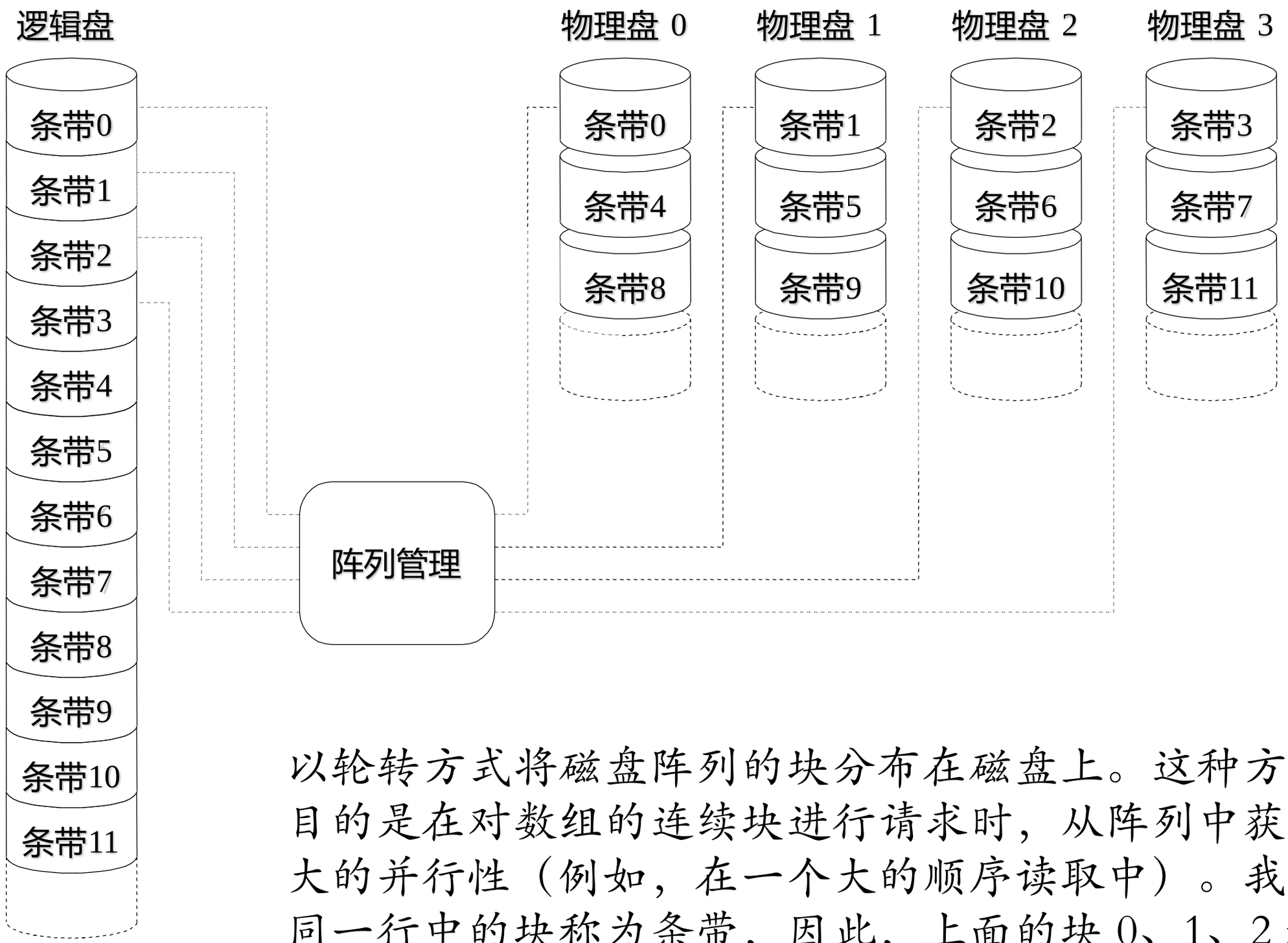
考虑 3 个重要的 RAID 设计：RAID 0 级（条带化），RAID 1 级（镜像）和 RAID 4/5 级（基于奇偶校验的冗余）。

RAID Level 0: Striping

- The first RAID level is actually **not a RAID level** at all, in that there is **no redundancy**.

- However, RAID level 0, or **striping** as it is better known, serves as an excellent **upper-bound** on performance and capacity and thus is worth understanding.

RAID0



以轮转方式将磁盘阵列的块分布在磁盘上。这种方法的目的是在对数组的连续块进行请求时，从阵列中获取最大的并行性（例如，在一个大的顺序读取中）。我们将同一行中的块称为条带，因此，上面的块 0、1、2 和 3 在相同的条带中。

- RAID Level 0 is the simplest form as **striping** blocks.
  - **Spread the blocks** across the disks in a round-robin fashion.
  - No redundancy
  - Excellent performance and capacity

Disk 0	Disk 1	Disk 2	Disk 3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

RAID-0: Simple Striping

- 下面例子中，在每个磁盘上放置两个 4KB 块，然后移动到下一个磁盘。因此， 此 RAID 阵列的大块大小（chunk size）为 8KB，因此条带由 4 个大块（或 32KB）数据组成:

Disk 0	Disk 1	Disk 2	Disk 3	
0	2	4	6	chunk size:
1	3	5	7	2 blocks
8	10	12	14	
9	11	13	15	

Striping with a Bigger Chunk Size

补充：RAID 映射问题

在研究 RAID 的容量、可靠性和性能特征之前，我们首先提出一个问题，我们称之为映射问题（the mapping problem）。这个问题出现在所有 RAID 阵列中。简单地说，给定一个逻辑块来读或写，RAID 如何确切地知道要访问哪个物理磁盘和偏移量？

对于这些简单的 RAID 级别，我们不需要太多复杂计算，就能正确地将逻辑块映射到其物理位置。以上面的第一个条带为例（大块大小= 1 块= 4KB）。在这种情况下，给定逻辑块地址 A，RAID 可以使用两个简单的公式轻松计算要访问的磁盘和偏移量：

磁盘=A % 磁盘数

偏移量 =A / 磁盘数

请注意，这些都是整数运算（例如，4/3 = 1 而不是 1.33333…）。我们来看看这些公式如何用于一个简单的例子。假设在上面的第一个 RAID 中，对块 15 的请求到达。鉴于有 4 个磁盘，这意味着我们感兴趣的磁盘是（14 % 4 = 2）：磁盘 2。确切的块计算为（14 / 4 = 3）：块 3。因此，应在第三个磁盘（磁盘 2，从 0 开始）的第四个块（块 3，从 0 开始）处找到块 14，该块恰好位于该位置。

Chunk Sizes

- Chunk size mostly affects performance of the array
- **Small chunk size**
  - 较小的大块意味着许多文件将跨多个磁盘进行条带化，增加了对单个文件的读取和写入的并行性。但是，跨多个磁盘访问块的定位时间会增加，因为整个请求的定位时间由所有驱动器上请求的最大定位时间决定
- **Big chunk size**
  - 较大的大块大小减少了这种文件内的并行性，因此依靠多个并发请求来实现高吞吐量。但是，较大的大块大小减少了定位时间

Determining the “best” chunk size is hard to do.

Most arrays use larger chunk sizes (e.g., 64 KB)

RAID Level 0 Analysis

N : the number of disks

- **Capacity** → RAID-0 is perfect.
  - Striping delivers N disks worth of useful capacity.

- **Performance** of striping → RAID-0 is excellent.
  - All disks are utilized often in parallel.

- **Reliability** → RAID-0 is bad.
  - Any disk failure will lead to data loss.

Evaluating RAID Performance

- Consider two different performance metrics.
- 单请求延迟
- RAID 的稳态吞吐量，即许多并发请求的总带宽.
- 假设有两种类型的工作负载：顺序（sequential）和随机（random）。
- 对于顺序的工作负载，我们假设对阵列的请求大部分是连续的。
    - 例如，一个请求（或一系列请求）访问 1MB 数据，始于块（B），终于（B+1MB），这被认为是连续的。
- 对于随机工作负载，我们假设每个请求都很小，并且每个请求都是到磁盘上不同的随机位置。
    - 一些重要的工作负载（例如数据库管理系统（DBMS）上的事务工作负载）表现出这种类型的访问模式，因此它被认为是一种重要的工作负载。
- 对于顺序访问，磁盘以最高效的模式运行，花费很少时间寻道并等待旋转，大部分时间都在传输数据。对于随机访问，情况恰恰相反：大部分时间花在寻道和等待旋转上，花在传输数据上的时间相对较少。

Evaluating RAID Performance Example

- 我们假设磁盘可以在连续工作负载下以 S MB/s 传输数据，并且在随机工作负载下以 R MB/s 传输数据。一般来说，S 比 R 大得多。
- 给定以下磁盘特征， 计算 S 和 R。假设平均大小为 10MB 的连续传输，平均为 10KB 的随机传输。另外，假设以下磁盘特征： 平均寻道时间 7ms ，平均旋转延迟 3ms ，磁盘传输速率 50MB/s 。
- 要计算 S，我们需要首先计算在典型的 10MB 传输中花费的时间。首先，我们花 7ms 寻找，然后 3ms 旋转。最后，传输开始。10MB / 50MB/s =1/5s，即 200ms 的传输时间。 因此，对于每个 10MB 的请求，花费了 210ms 完成请求。

$$S = \frac{\text{数据量}}{\text{访问时间}} = \frac{10\text{MB}}{210\text{ms}} = 47.62\text{MB/s}$$

类似地计算 R。寻道和旋转是一样的。然后我们计算传输所花费的时间，即 10KB /50MB/s= 0.195ms。

$$R = \frac{\text{数据量}}{\text{访问时间}} = \frac{10\text{KB}}{10.195\text{ms}} = 0.981\text{MB/s}$$

RAID-0 Analysis

- For capacity, it is perfect: given N disks, striping delivers **N disks worth of useful capacity**.

- For reliability, striping is also perfect, but in the bad way: **any disk failure will lead to data loss**.

- Finally, performance is excellent: all disks are utilized.

- For steady-state throughput, throughput equals N multiplied by S (**N · S MB/s**) . For a large number of random I/Os, we can again use all of the disks, and thus obtain **N · R MB/s**.

RAID Level 1: Mirroring

- 对于镜像系统，我们只需生 成系统中每个块的多个副本。当然，每个副本应该放在一个单独的磁盘上。通过这样做， 我们可以容许磁盘故障。
- 在一个典型的镜像系统中，我们将假设对于每个逻辑块，RAID 保留两个物理副本。

Disk 0	Disk 1	Disk 2	Disk 3
0	0	1	1
2	2	3	3
4	4	5	5
6	6	7	7

Simple RAID-1: Mirroring

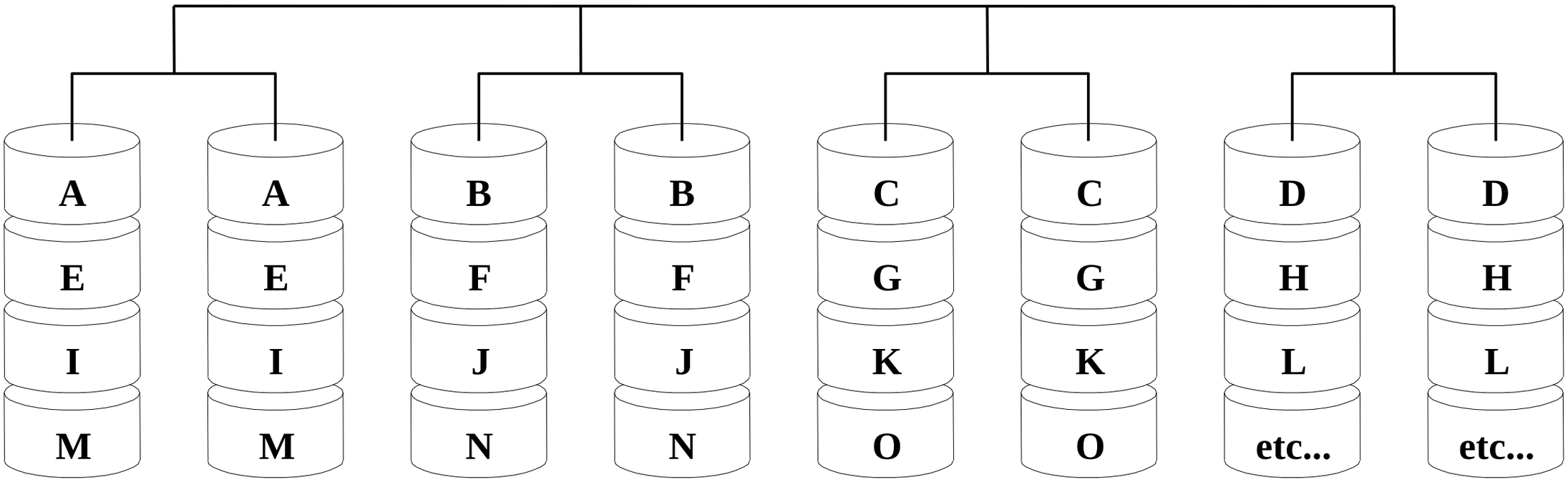
磁盘 0 和磁盘 1 具有相同的内容，而磁盘 2 和磁盘 3 也具有相同的内容。数据在这些镜像对之间条带化。

上面的安排是常见的安排，有时称为 RAID-10（或 RAID 1+0），因为它使用镜像对（RAID-1），然后在其上使用条带化（RAID-0）.目前， 我们的讨论只是假设上面布局的镜像。

RAID1＋0

- RAID1+0
  - 先镜像后分块

先做RAID1，然后再做RAID0，因此RAID1+0允许坏多个盘，只要不是一对磁盘坏就可以。

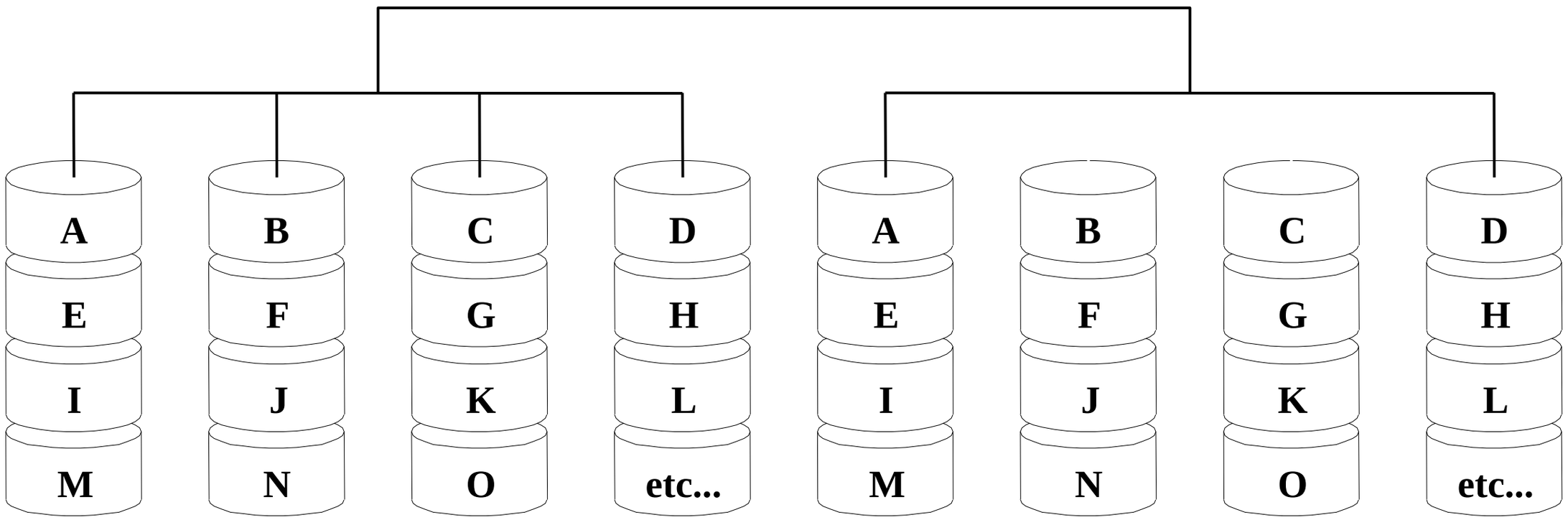


RAID 0和RAID 1分别用于增强存储性能（RAID0条带）和数据安全性（RAID1镜像），而RAID0+1和RAID1+0兼顾了RAID0和RAID1的优点，它在提供RAID1一样的数据安全保证的同时，也提供了与RAID0近似的存储性能。

RAID0＋1

- RAID0+1
  - 先分块后镜像

先做两个RAID0，然后再做RAID1，因此RAID0+1允许坏多个盘，但只能在坏在同一个RAID0中，不允许两个RAID0都有坏盘。



在RAID 0+1技术中，当一块物理磁盘出现故障将导致整个虚拟磁盘损失，因此相当于块中物理磁盘的有效故障。如果其它块物理磁盘有一块丢失，数据将发生丢失。虽然从原理上可以从剩余磁盘数据中重建，但目前市场上的RAID控制器都不能做到数据完全恢复。

Disk 0	Disk 1	Disk 2	Disk 3
0	0	1	1
2	2	3	3
4	4	5	5
6	6	7	7

- 从镜像阵列读取块时，RAID 有一个选择：它可以读取任一副本。例如，如果对 RAID 发出对逻辑块 5 的读取，则可以自由地从磁盘 2 或磁盘 3 读取它。

- 在写入块时，不存在这样的选择：RAID 必须更新两个副本的数据，以保持可靠性。但请注意，这些写入可 以并行进行。例如，对逻辑块 5 的写入可以同时在磁盘 2 和 3 上进行。

RAID-1 Analysis
Capacity and Reliability

- 从容量的角度来看，RAID-1 价格昂贵。在镜像级别=2 的情 况下，我们只能获得峰值有用容量的一半。因此，对于 N 个磁盘，镜像的有用容量为 N/2。

- 从可靠性的角度来看，RAID-1 表现良好。它可以容许任何一个磁盘的故障。

RAID-1 Analysis - Performance

- 从单个读取请求的延迟角度来看，我们可以看到它与单个磁盘 上的延迟相同。 RAID-1读取任何一个副本都可以。

- 在完成写入之前， 需要完成两次物理写入。这两个写入并行发生，因此时间大致等于单次写入的时间。然而， 因为逻辑写入必须等待两个物理写入完成，所以它遭遇到两个请求中最差的寻道和旋转延迟，因此（平均而言）比写入单个磁盘略高。

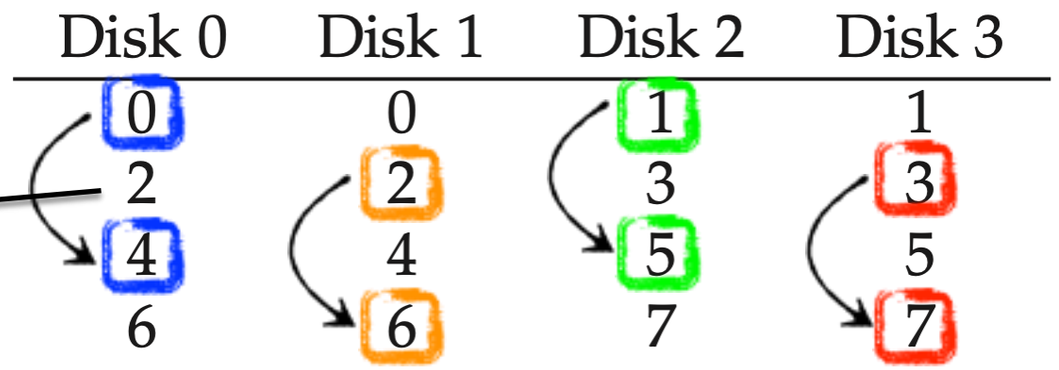
RAID-1 Analysis - Performance

- 顺序写入磁盘时，每个逻辑写入必定导致两个物理写入。例如，当我们写入逻辑块 0时，RAID 在内部会将它写入磁盘 0 和磁盘 1。因此，我们可以得出结论，顺序写入镜像阵列期间获得的最大带宽是 (N/2) · S MB/s，即峰值带宽的一半。也就是说连续写入时，无法并行。

- The **sequential read** will only obtain a bandwidth of (N/2) · S MB/s. Why?

- Imagine we need to read blocks 0, 1, 2, 3, 4, 5, 6, and 7. Let’s say we issue the read of 0 to disk 0, the read of 1 to disk 2, the read of 2 to disk 1, and the read of 3 to disk 3. We continue by issuing reads to 4, 5, 6, and 7 to disks 0, 2, 1, and 3, respectively.

实际上，每个磁盘都会接收到每个其他块的请求，当它在跳过的块上旋转时，不会为客户提供有用的带宽。因此，每个磁盘只能提供一半的峰值带宽。因此，连续读出时，无法并行



RAID-1 Analysis - Performance

- 随机读取是镜像 RAID 的最佳案例。在这种情况下，我们可以在所有磁盘上分配读取数 据，从而获得完整的可用带宽。因此，对于随机读取，RAID-1 提供 N · R MB/s。

- 每个逻辑写入必须变成两个物理写入，因此在所有磁盘都将被使用的情况下，客户只会看到可用带宽的一半，因此随机写的性能: (N/2) · R MB/s.

RAID Level 4: Saving Space With Parity

- We now present a different method of adding redundancy to a disk array known as **parity**.

- Parity-based approaches attempt to use less capacity and thus overcome the **huge space penalty** paid by mirrored systems.

- They do so at a cost, however: performance.

- In a five-disk RAID-4 system, we might observe the following layout:

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
4	5	6	7	P1
8	9	10	11	P2
12	13	14	15	P3

- To compute parity, we need to use a mathematical function that enables us to withstand the loss of any one block from our stripe. It turns out the simple function **XOR** does the trick quite nicely.

C0	C1	C2	C3	P
0	0	1	1	XOR(0,0,1,1) = 0
0	1	0	0	XOR(0,1,0,0) = 1

- How do we apply XOR to a bunch of blocks to compute the parity?

- RAID 在每个磁盘上放置了 4KB（或更大）的块。如何将 XOR 应用于一堆块来计算奇偶校验？事实证明这很容易。只需在数据块的每一位上执行按位 XOR。

- Given 4 blocks 0010, 1001, 1100, 1001, how to get the parity block?

表 38.6　　将 XOR 用于块

Block0	Block1	Block2	Block3	Parity
00	10	11	10	11
10	01	00	01	10

RAID-4 Analysis - Capacity

- From a capacity standpoint, RAID-4 uses 1 disk for parity information for every group of disks it is protecting. Thus, our useful capacity for a RAID group is **N-1**.

RAID-4 Analysis - Reliability

- Reliability is also quite easy to understand: RAID-4 tolerates **1 disk failure** and **no more**. If more than one disk is lost, there is simply no way to reconstruct the lost data.

RAID-4 Analysis - Performance

- 连续读取性能可以利用除奇偶校验磁盘以外的所有磁盘，因此可提供（N−1）·S MB/s（简单情况）的峰值有效带宽。

- 要理解顺序写入的性能，我们必须首先了解它们是如何完成的。将大块数据写入磁盘时，RAID-4 可以执行一种简单优化，称为全条带写入（full-stripe write）。例如，设想块 0、1、2 和 3 作为写请求的一部分发送到 RAID

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
4	5	6	7	P1
8	9	10	11	P2
12	13	14	15	P3

- RAID 可以简单地计算 P0 的新值（通过在块 0、1、2 和 3 上执行 XOR），然后将所有块（包括奇偶块）并行写入上面的 5 个磁盘（在图中以灰色突出显示）。因此，全条带写入是 RAID-4 写入磁盘的最有效方式。

- 一旦我们理解了全条带写入，计算 RAID-4 上顺序写入的性能就很容易。有效带宽也是 $(N-1) \cdot S$ MB/s。

- 随机读取的性能。从表中还可以看到，一组 1 块的随机读取将分布在系统的数据磁盘上，而不是奇偶校验磁盘上。因此，有效性能是：（N−1）· R MB/s。

Random write performance for RAID-4

- 随机写入，在写入数据的同时必须更新奇偶校验值。如何正确并有效地更新它？
- 第一种称为加法奇偶校验（additive parity），要求我们做以下工作。为了计算新奇偶校验块的值，并行读取条带中所有其他数据块（在本例中为块 0、2 和 3），并与新块（1）进行异或。结果是新的校验块。为了完成写操作，你可以将新数据和新奇偶校验写入其各自的磁盘，也是并行写入。
- 在较大的 RAID 中，需要大量的读取来计算奇偶校验。

Random write performance for RAID-4 (Cont.)

- **Method 2:减法奇偶校验（subtractive parity）方法**

C0	C1	C2	C3	P
0	0	1	1	XOR(0,0,1,1)=0

  - Update C2(old) → C2(new)
    1. Read in the old data at C2 (C2(old)=1) and the old parity (P(old)=0)
    2. Calculate P(new): $P(new) = (C2(old)\ XOR\ C2(new))\ XOR\ P(old)$
  - If C2(new)==C2(old) → P(new)==P(old)
  - If C2(new)!=C2(old) → Flip the old parity bit

```
P(new) = (C(old) XOR C(new)) XOR P(old)
```

- We can express this whole mess neatly with XOR as it turns out (if you understand XOR, this will now make sense to you):

```
P(new) = (C(old) XOR C(new)) XOR P(old)
```

- For this performance analysis, let us assume we are using the subtractive method.

- 使用减法方法。对于每次写入，RAID 必须执行 4 次 物理 I/O（两次读取和两次写入）。

- 在一个随机写的时候，数据盘和校验盘可以并行读和写，先读再写，所以是R/2

Small-write problem

- The parity disk can be a **bottleneck.**
  - Example: update blocks 4 and 13 (marked with *)

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
*4	5	6	7	+P1
8	9	10	11	P2
12	*13	14	15	+P3

Writes To 4, 13 And Respective Parity Blocks.

RAID-4 throughput under random small writes is $(\frac{R}{2})$ MB/s (*terrible*).

- Disk 0 and Disk 1 can be accessed in parallel.
- Disk 4 prevents any parallelism.

可以并行访问数据磁盘，奇偶校验磁盘也不会实现任何并行

Raid4 延迟

- RAID-4 中的 I/O 延迟。你现在知道，单次读取（假设没有失败）只映射到单个磁盘，因此其延迟等同于单个磁盘请求的延迟。
- 单次写入的延迟需要两次读取，然后两次写入。读操作可以并行进行，写操作也是如此，因此总延迟大约是单个磁盘的两倍。（有一些差异，因为我们必须等待两个读取操作完成，所以会得到最差的定位时间，但是之后，更新不会导致寻道成本，因此可能是比平均水平更好的定位成本。）

RAID Level 5: Rotating Parity

- To address the small-write problem RAID-5 was introduced. RAID-5 works almost **identically to RAID-4**, except that it **rotates** the parity block across drives.

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
5	6	7	P1	4
10	11	P2	8	9
15	P3	12	13	14
P4	16	17	18	19

RAID-5 With Rotated Parity

RAID-5 Analysis

- Capacity — N : the number of disks
  - The useful capacity for a RAID group is $(N-1)$.
- Reliability
  - RAID-5 tolerates 1 disk failure and no more.

RAID-5 Analysis (Cont.)

N : the number of disks

- Performance
  - Sequential read and write (N-1)*S 和Raid4一致
  - Random read : a little better than RAID-4
    - RAID-5 can utilize all of the disks. (N*R)
  - Random write : $\frac{N}{4} * R$ MB/s
    - The factor of four loss is cost of using parity-based RAID.

RAID-4 的随机写入性 能明显提高。想象一下写入块 1 和写入块 10，这将变成对磁盘 1 和磁盘 4（对于块 1 及其奇偶校验）的请求以及对磁盘 0 和磁盘 2（对于块 10 及 其奇偶校验）的请求。因此，它们可以并行进行。事实上，我们通常可以假设，如果有大量的随机请求，我们将能够保持所有磁盘均匀忙碌。如果是这样的话，那么我们用于小写入的总带宽将是 N/4 · R MB/s。4 倍损失是由于每个 RAID-5 写入仍然产生总计 4 个 I/O 操 作，这就是使用基于奇偶校验的 RAID 的成本。

RAID Comparison: A Summary

N : the number of disks
D : the time that a request to a single disk take

	RAID-0	RAID-1	RAID-4	RAID-5
Capacity	N	N/1	N-1	N-1
Reliability	0	1 (for sure) $\frac{N}{2}$ (if lucky)	1	1
Throughput				
Sequential Read	N・S	(N/2) ・S	(N-1) ・S	(N-1) ・S
Sequential Write	N・S	(N/2) ・S	(N-1) ・S	(N-1) ・S
Random Read	N・R	N・R	(N-1) ・R	N・R
Random Write	N・R	(N/2) ・R	$\frac{1}{2}$ R	$\frac{N}{4}$ R
Latency				
Read	D	D	D	D
Write	D	D	2D	2D

RAID Capacity, Reliability, and Performance