RAID

Redundant Array of Independent Disks

Niv Dayan

RAID Addresses Three Problems

Our database size exceeds one drive and we need more storage

RAID Addresses Three Problems

Our database size exceeds one drive and we need more storage

A drive fails, and we need to recover its data

RAID Addresses Three Problems

Our database size exceeds one drive and we need more storage

A drive fails, and we need to recover its data

We want to overcome the limits of one storage device speed

Origin from 1988

A Case for Redundant Arrays of Inexpensive Disks (RAID)

David A. Patterson, Garth Gibson, and Randy H. Katz

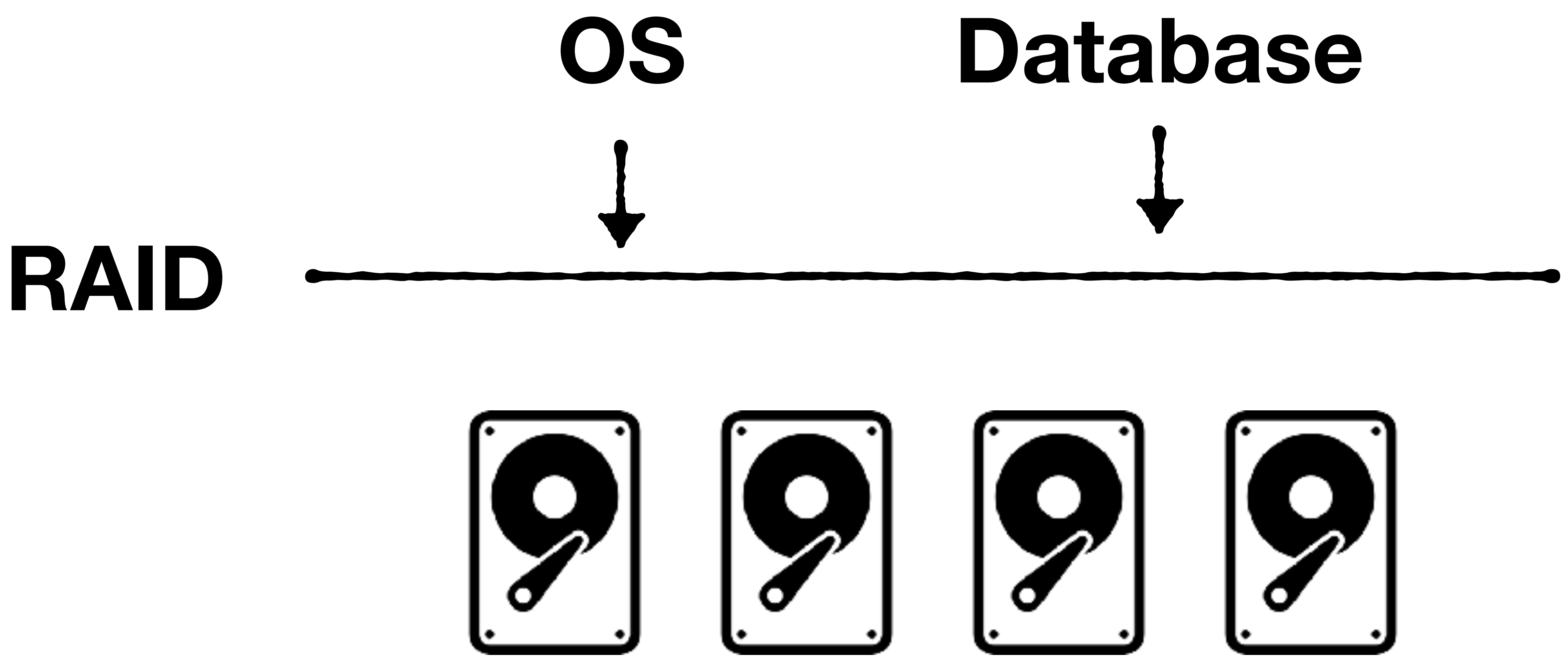
Computer Science Division
Department of Electrical Engineering and Computer Sciences
571 Evans Hall
University of California
Berkeley, CA 94720
(pattrsn@pepper.berkeley.edu)

Abstract

Increasing performance of CPUs and memories will be squandered if not matched by a similar performance increase in I/O. While the capacity of Single Large Expensive Disk (SLED) has grown rapidly, the performance improvement of SLED has been modest. Redundant Arrays of Inexpensive Disks (RAID), based on the magnetic disk technology developed for personal computers, offers an attractive alternative to SLED, promising improvements of an order of magnitude in performance, reliability, power consumption, and scalability.

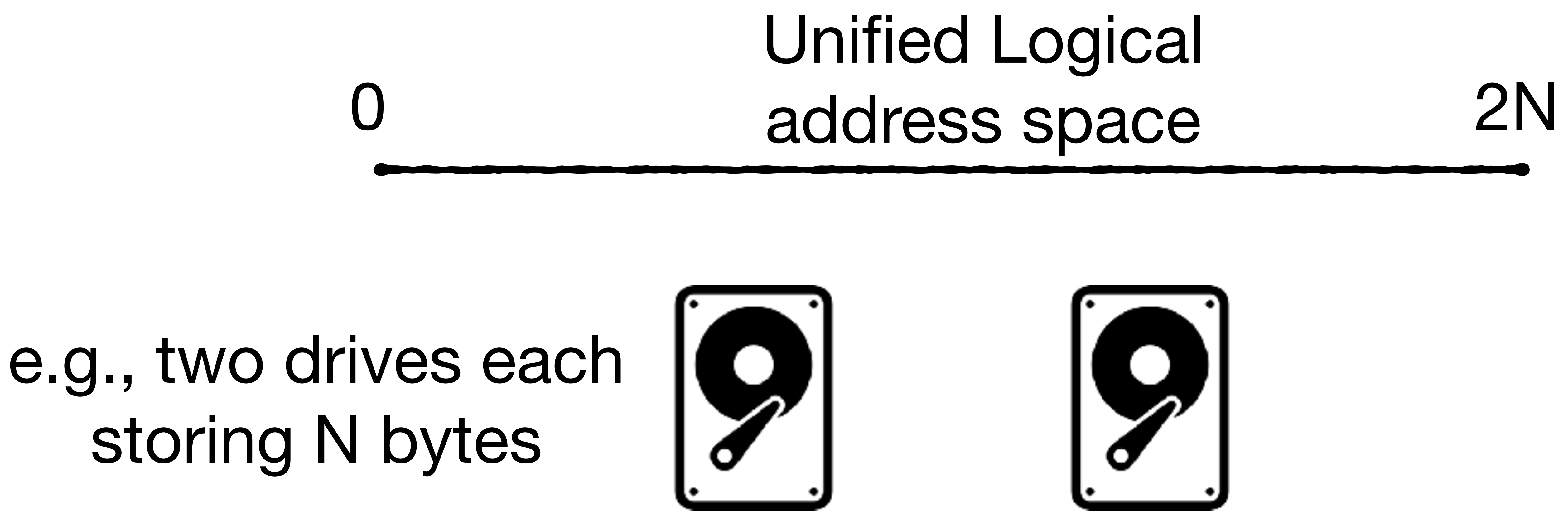
This paper introduces five levels of RAIDs, giving their relative cost/performance, and compares RAIDs to an IBM 3380 and a Fujitsu Super Eagle.

RAID divides data along with redundancy on multiple disks



Enables larger data, better performance, and recovery of a drive failure

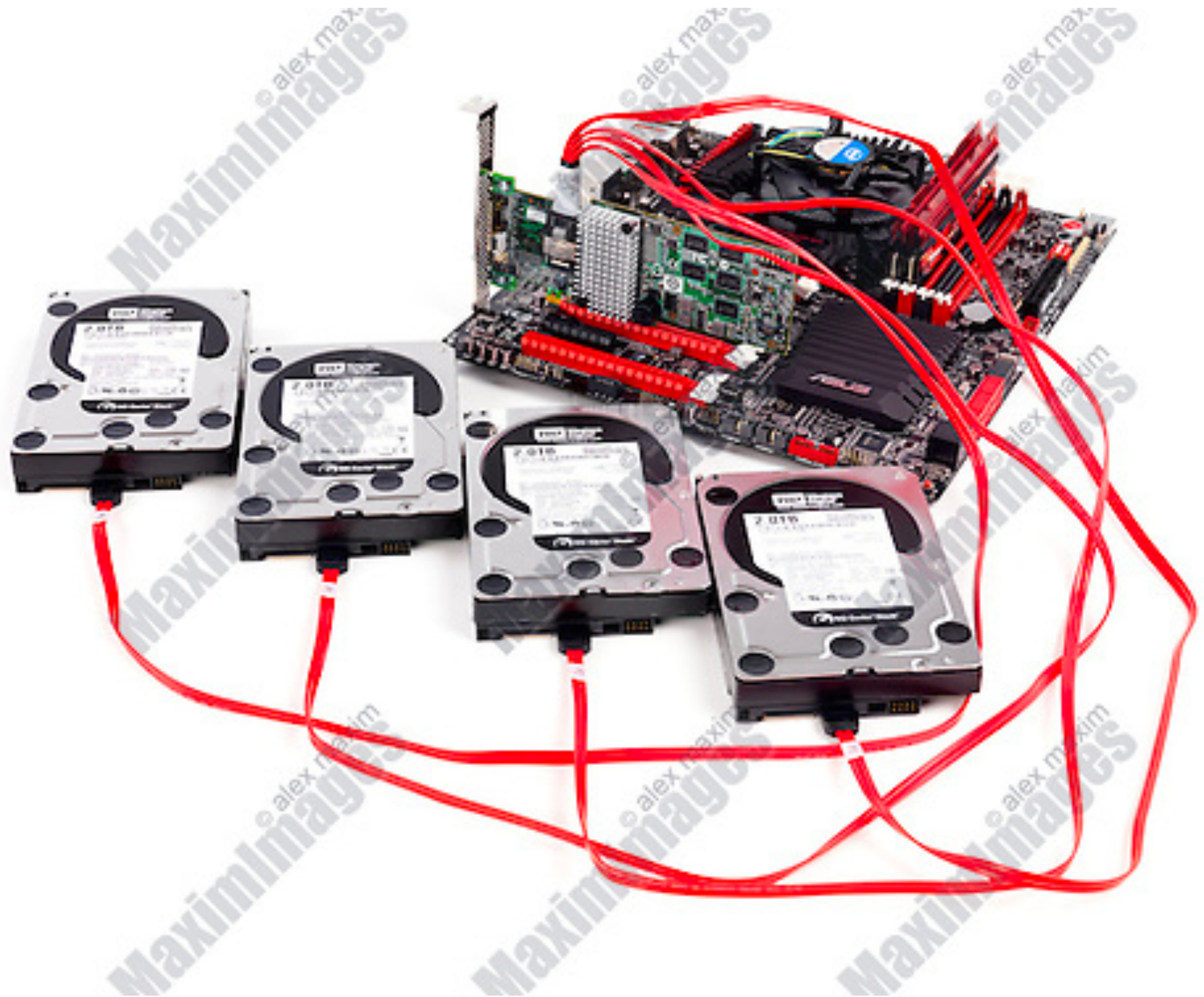
Expose a larger logical address space to OS



Looks to the OS like one drive, though consists of many

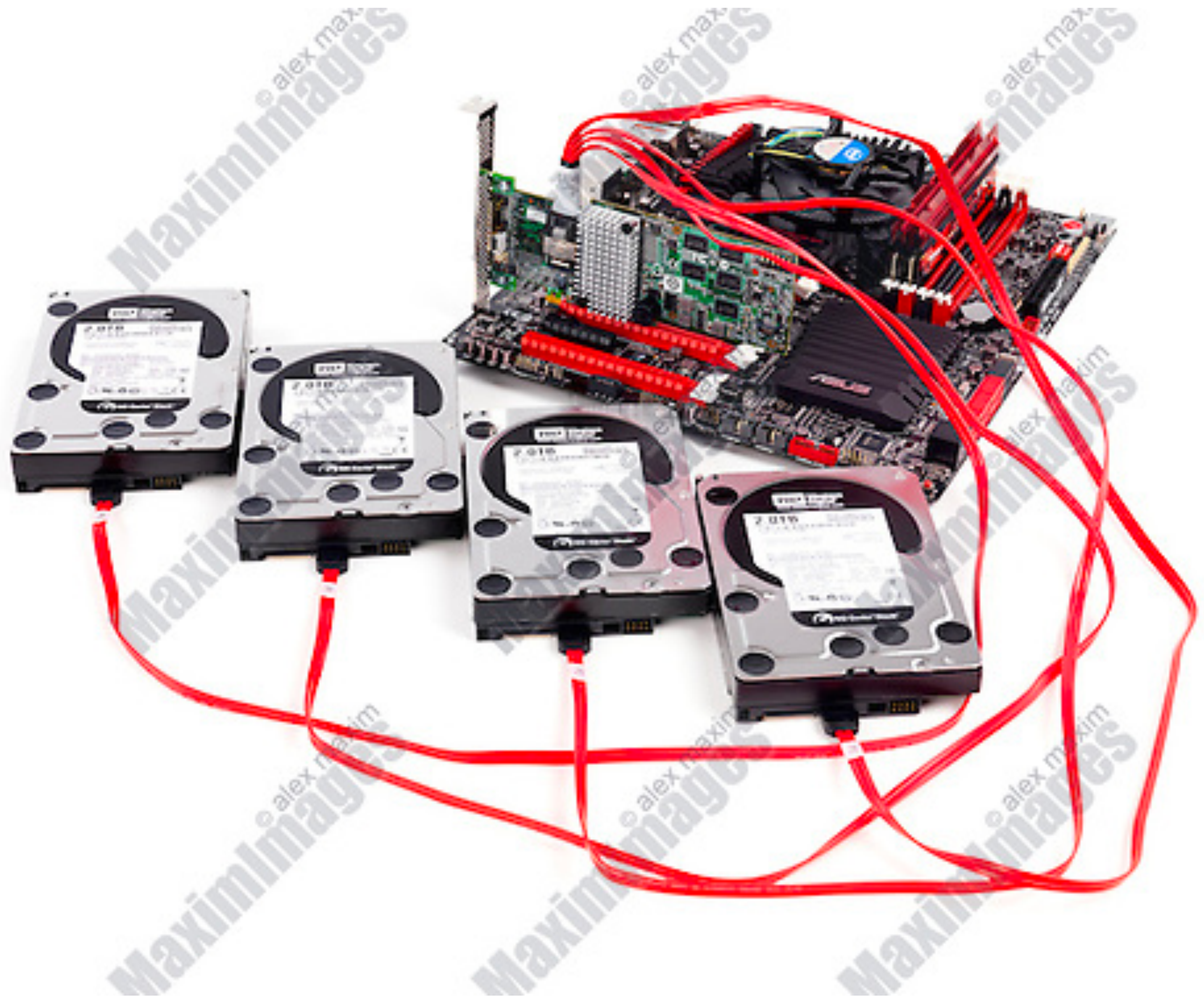
Can be implemented in…

Hardware



Can be implemented in…

Hardware Or Software



There are many RAID designs, but we'll only cover five

RAID 0 **RAID 1** **RAID 0+1** **RAID 4** **RAID 5**

RAID 0 - Pure striping

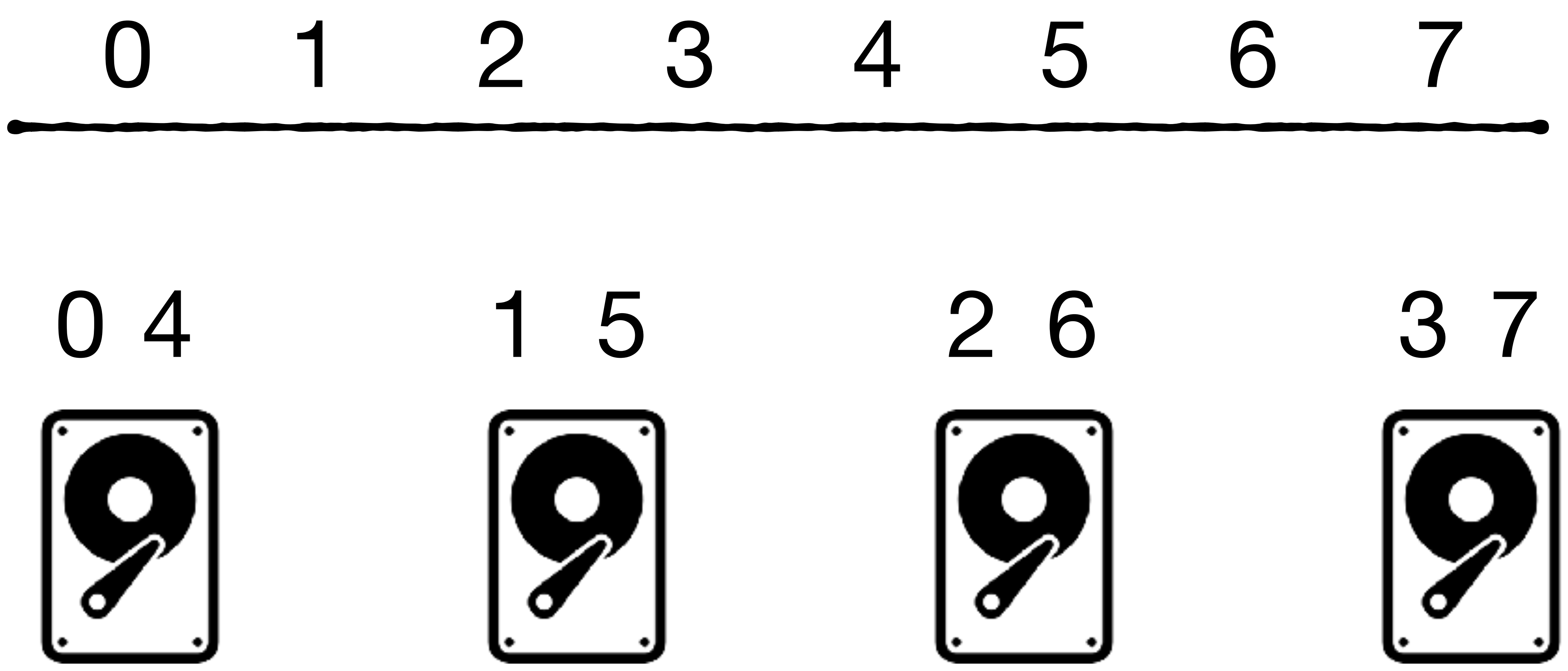
Stripe data in the logical address

Each number represents
a 4 KB block address

RAID 0 - Pure striping

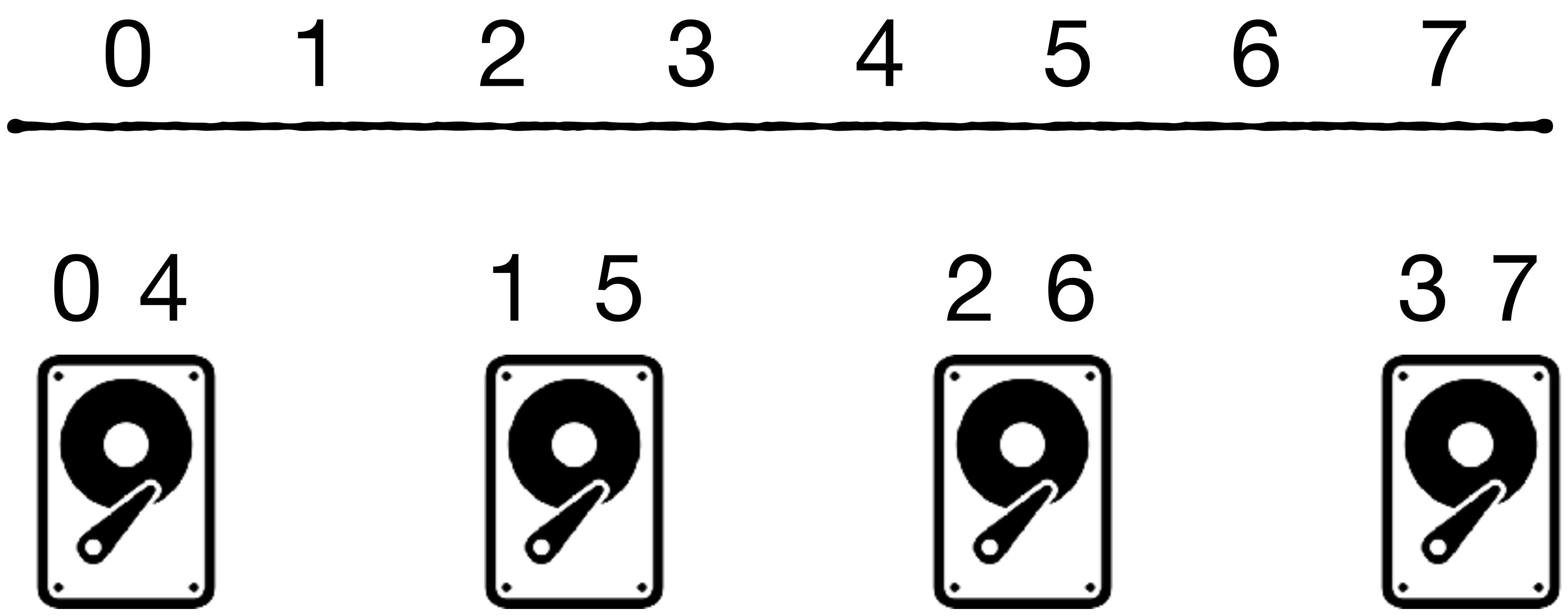
Stripe data in the logical address

Each number represents
a 4 KB block address



RAID 0 - Pure striping

Stripe data in the logical address



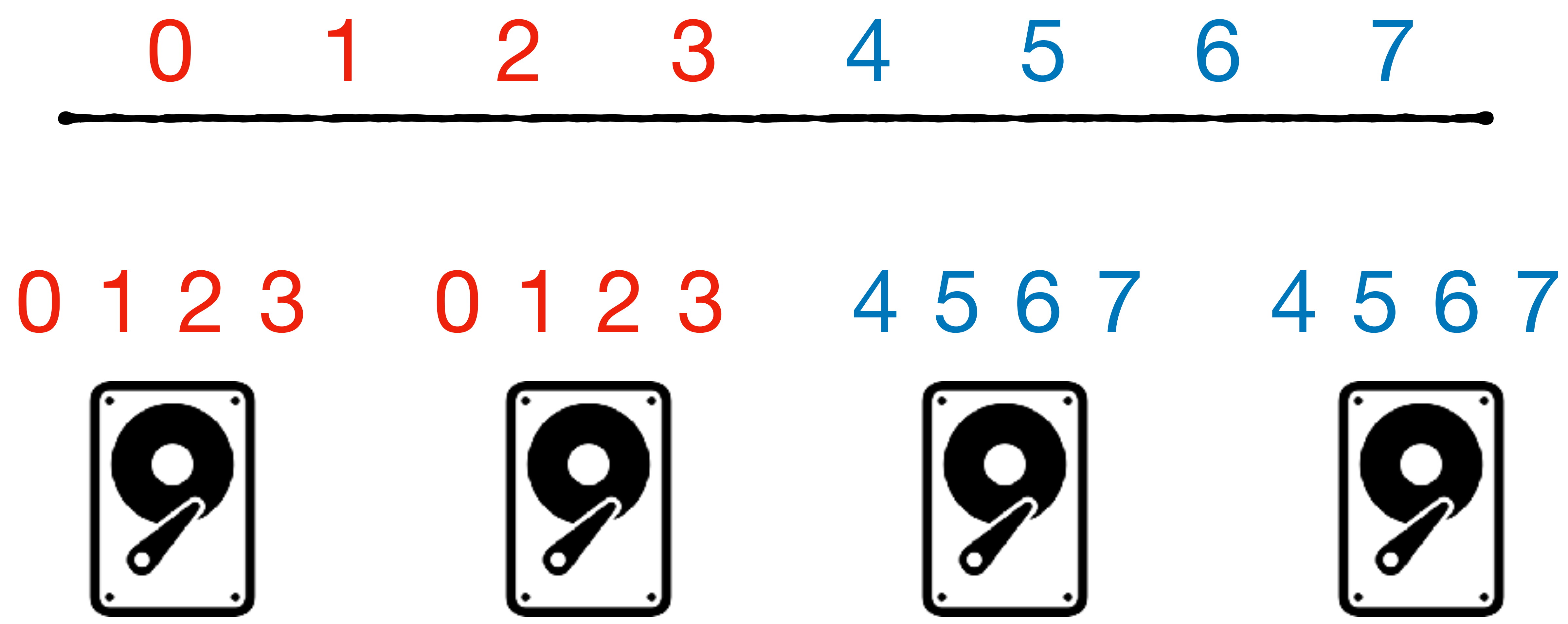
1. Much faster sequential writes and reads
2. Also improvement for random writes and reads due to load balancing
3. No redundancy. If one disk fails, we lose data.

RAID 0 **RAID 1** RAID 0+1 RAID 4 RAID 5

RAID 1 - Mirroring

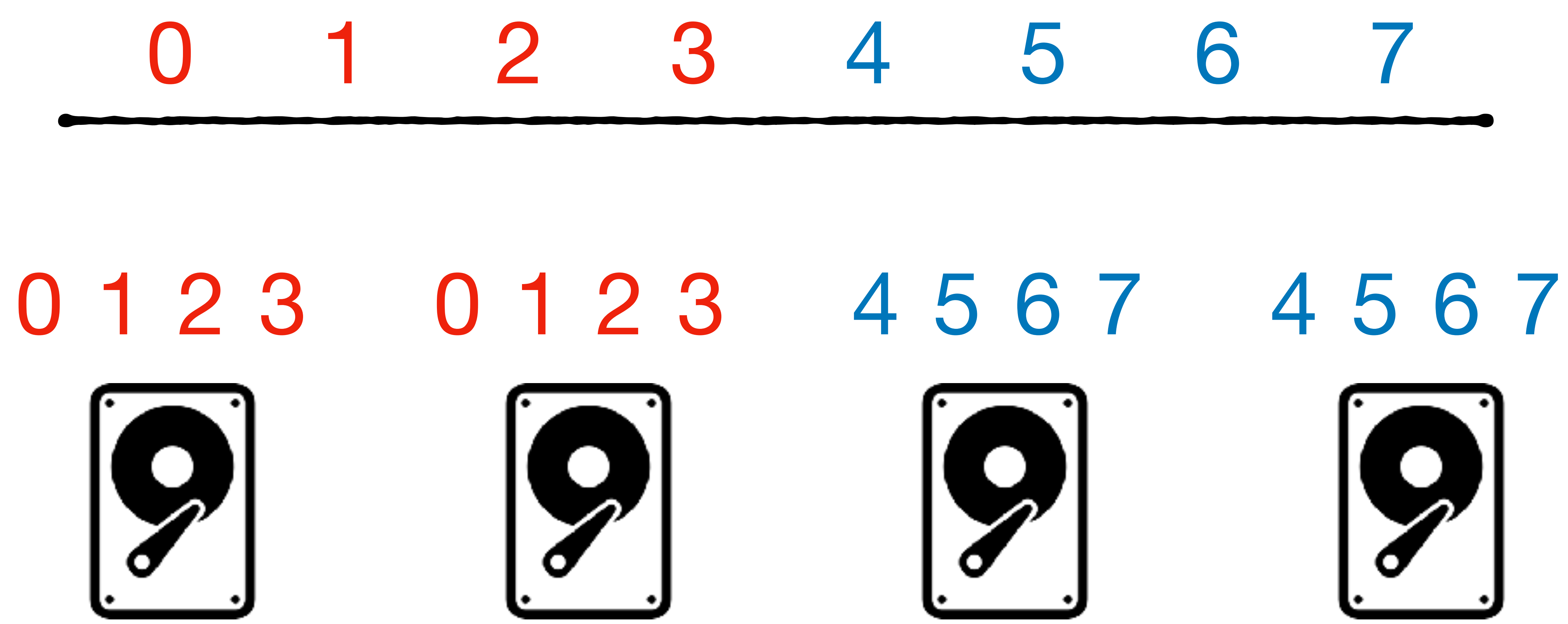
Each drive has one mirror

Each number represents a 4 KB block address



RAID 1 - Mirroring

Each drive has one mirror



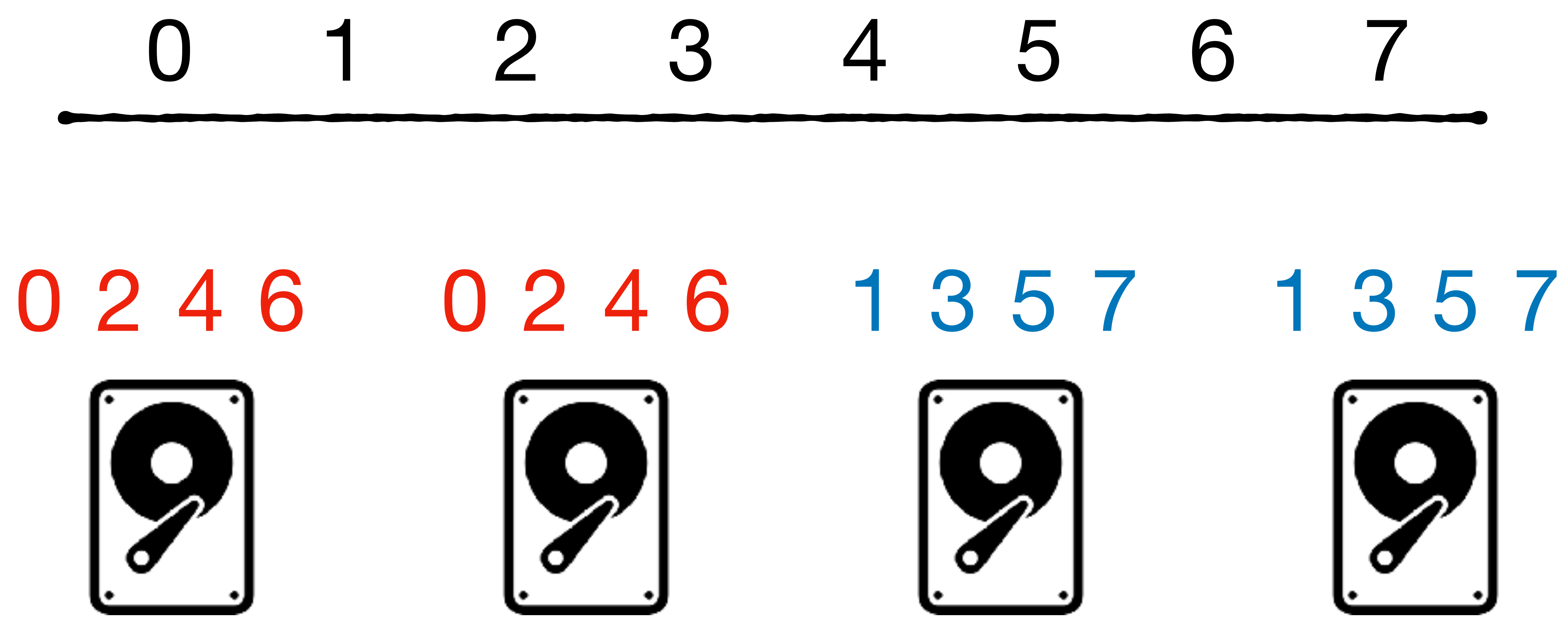
1. Slower writes as they must make 2 copies
2. Faster reads as we have a choice to read from a non-busy drive
3. Allows recovery of a disk but costs 50% of storage capacity

RAID 0 RAID 1 **RAID 0+1** RAID 4 RAID 5

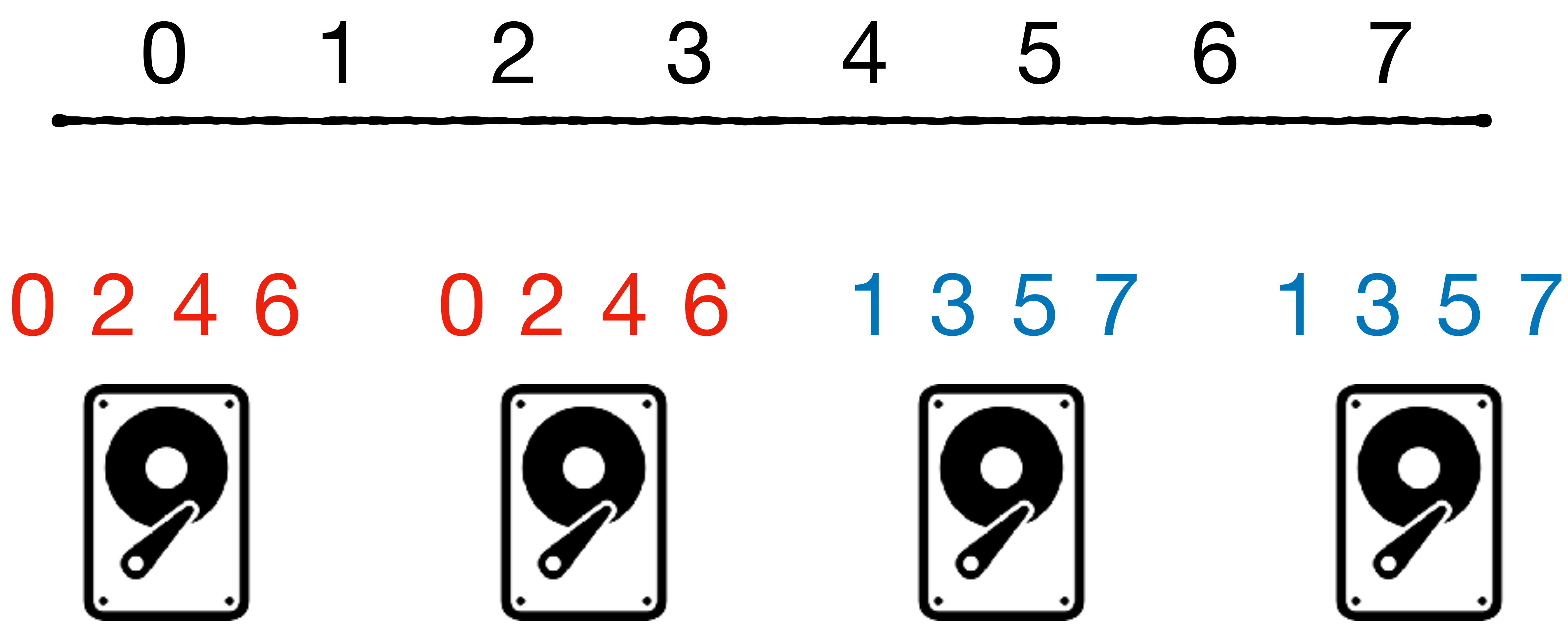
RAID 0+1 - Striping and Mirroring

We stripe and mirror data at the same time

Each number represents
a 4 KB block address



RAID 0+1 - Striping and Mirroring



1. Faster sequential reads and writes as they are more distributed
2. Writes still require making two copies, and reads still have flexibility
3. Still requires 50% of storage capacity

RAID 0 RAID 1 RAID 0+1 **RAID 4** **RAID 5**

For next time

☺