RAID

- Redundant Array of Independent Disks
- A set of physical disk drives viewed by the OS as a single logical drive
- Data are distributed across the physical drives. May improve performance.
- Redundant disk stores parity information. Recoverability, reliability.

RAID Levels

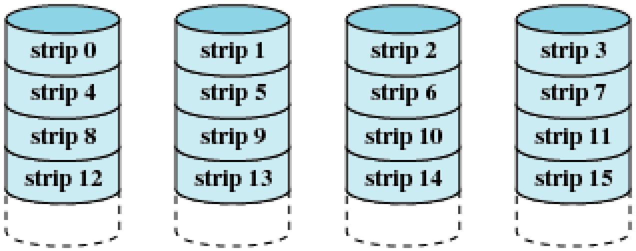
Table 6.3 RAID Levels

Category	Level	Description	Disks Required	Data Availability	Large I/O Data Transfer Capacity	Small I/O Request Rate
Striping	0	Nonredundant	N	Lower than single disk	Very high	Very high for both read and write
Mirroring	1	Mirrored	$2N$	Higher than RAID 2, 3, 4, or 5; lower than RAID 6	Higher than single disk for read; similar to single disk for write	Up to twice that of a single disk for read; similar to single disk for write
Parallel access	2	Redundant via Hamming code	$N + m$	Much higher than single disk; comparable to RAID 3, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
	3	Bit-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
Independent access	4	Block-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 5	Similar to RAID 0 for read; significantly lower than single disk for write	Similar to RAID 0 for read; significantly lower than single disk for write
	5	Block-interleaved distributed parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 4	Similar to RAID 0 for read; lower than single disk for write	Similar to RAID 0 for read; generally lower than single disk for write
	6	Block-interleaved dual distributed parity	$N + 2$	Highest of all listed alternatives	Similar to RAID 0 for read; lower than RAID 5 for write	Similar to RAID 0 for read; significantly lower than RAID 5 for write

Note: N = number of data disks; m proportional to log N

RAID 0 (Non-redundant)

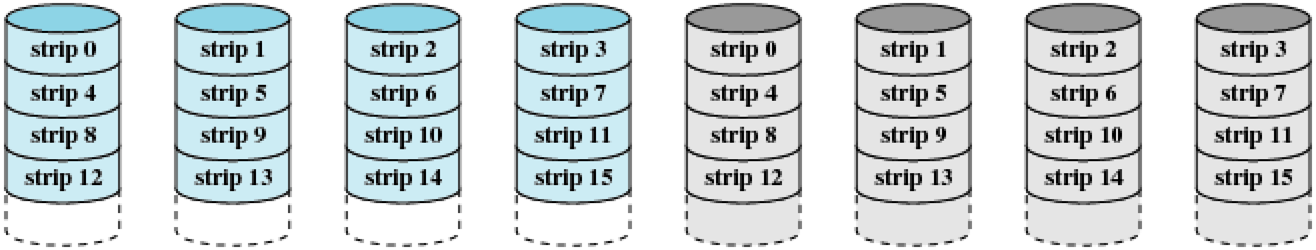
- The logical disk is divided into strips, mapped round robin to consecutive physical disks
- Improve performance in disk read/write
- Not fault tolerant



(a) RAID 0 (non-redundant)

RAID 1 (Mirrored)

- Each disk is mirrored by another disk
- Good performance if the hardware supports concurrent read/write to the mirrored pair
- Reliable, but expensive



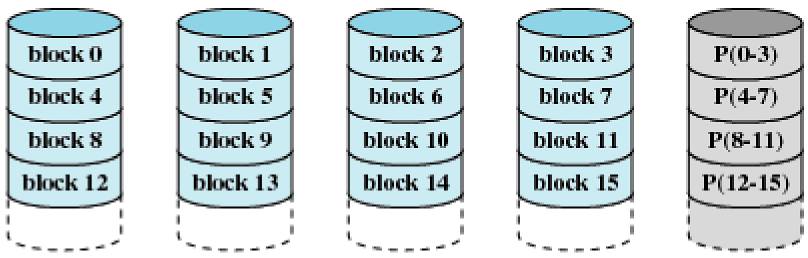
(b) RAID 1 (mirrored)

Parity strip

- Computed and updated at write, verified at read
- Every write results in two read and two write of strips
- A corrupted strip can be recovered

To compute the parity strip...
P(0-3) := b0 ⊕ b1 ⊕ b2 ⊕ b3

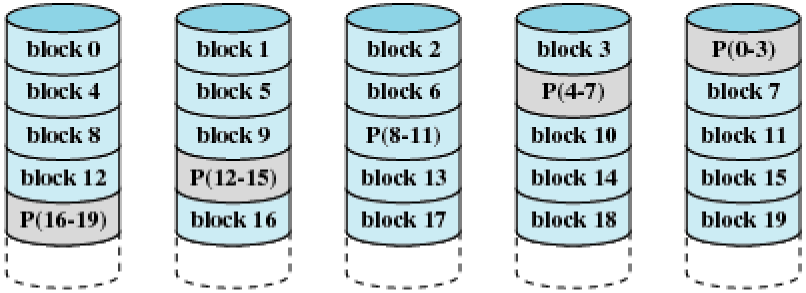
To recover the block 0...
b0 = *P(0-3)* ⊕ b1 ⊕ b2 ⊕ b3



(e) RAID 4 (block-level parity)

RAID 5 (Block-level distributed parity)

- Having all parity strips on one disk may make it a bottleneck. Instead, we can distribute the parity strips among the disks
- If a single disk fails, the system can regenerate the data lost
- Reliable. Good performance with special hardware



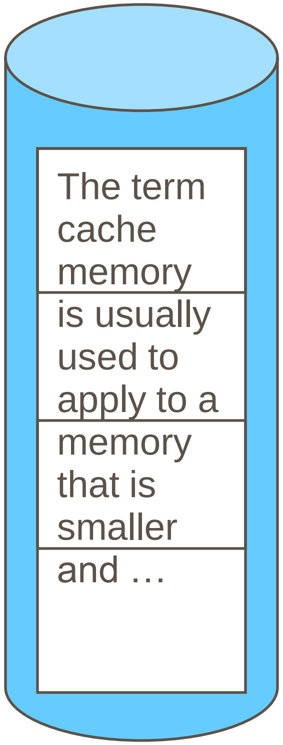
(f) RAID 5 (block-level distributed parity)

Table 6.4 RAID Comparison

Level	Advantages	Disadvantages	Applications
0	I/O performance is greatly improved by spreading the I/O load across many channels and drives No parity calculation overhead is involved Very simple design Easy to implement	The failure of just one drive will result in all data in an array being lost	Video production and editing Image Editing Pre-press applications Any application requiring high bandwidth
1	100% redundancy of data means no rebuild is necessary in case of a disk failure, just a copy to the replacement disk Under certain circumstances, RAID 1 can sustain multiple simultaneous drive failures Simplest RAID storage subsystem design	Highest disk overhead of all RAID types (100%) – inefficient	Accounting Payroll Financial Any application requiring very high availability
2	Extremely high data transfer rates possible The higher the data transfer rate required, the better the ratio of data disks to ECC disks Relatively simple controller design compared to RAID levels 3, 4, & 5	Very high ratio of ECC disks to data disks with smaller word sizes— inefficient Entry level cost very high— requires very high transfer rate requirement to justify	No commercial implementations exist/not commercially viable
3	Very high read data transfer rate Very high write data transfer rate Disk failure has an insignificant impact on throughput Low ratio of ECC (parity) disks to data disks means high efficiency	Transaction rate equal to that of a single disk drive at best (if spindles are synchronized) Controller design is fairly complex	Video production and live streaming Image editing Video editing Prepress applications Any application requiring high throughput
4	Very high Read data transaction rate Low ratio of ECC (parity) disks to data disks means high efficiency	Quite complex controller design Worst write transaction rate and Write aggregate transfer rate Difficult and inefficient data rebuild in the event of disk failure	No commercial implementations exist/not commercially viable
5	Highest Read data transaction rate Low ratio of ECC (parity) disks to data disks means high efficiency Good aggregate transfer rate	Most complex controller design Difficult to rebuild in the event of a disk failure (as compared to RAID level 1)	File and application servers Database servers Web, e-mail, and news servers Intranet servers Most versatile RAID level
6	Provides for an extremely high data fault tolerance and can sustain multiple simultaneous drive failures	More complex controller design Controller overhead to compute parity addresses is extremely high	Perfect solution for mission critical applications

Block-oriented disk

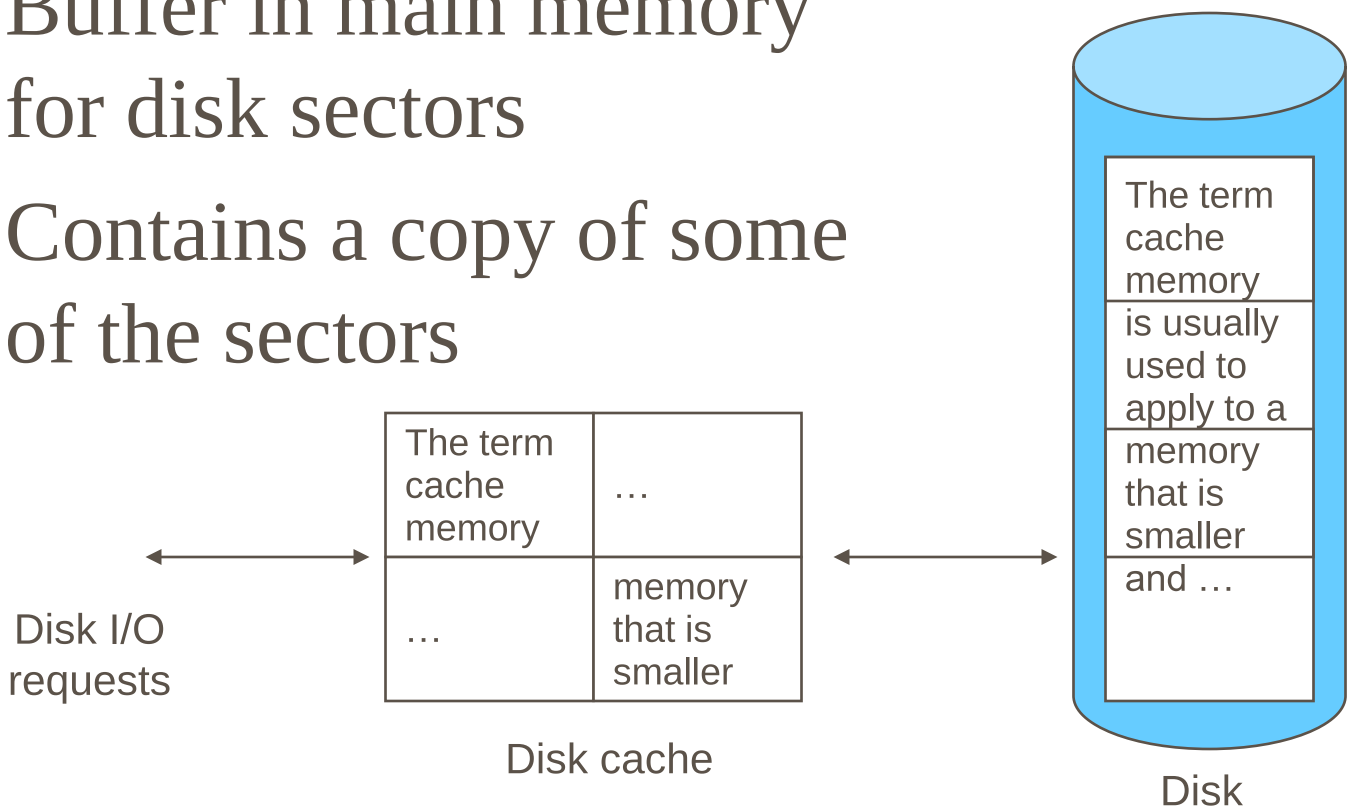
- Disk is block-oriented. One sector is read/written at a time.
- In PC, a sector is 512 byte




```
while (!feof(F)) {
  // read one char
  fscanf(F, "%c", &c);
  …
}
```

Disk Cache

- Buffer in main memory for disk sectors
- Contains a copy of some of the sectors



Disk Cache, Hit and Miss

- When an I/O request is made for a particular sector, the OS checks whether the sector is in the disk cache.
  - If so, (cache hit), the request is satisfied via the cache.
  - If not (cache miss), the requested sector is read into the disk cache from the disk.