CISC 3320
C28b Mass Storage: Reliability and Efficiency

Hui Chen

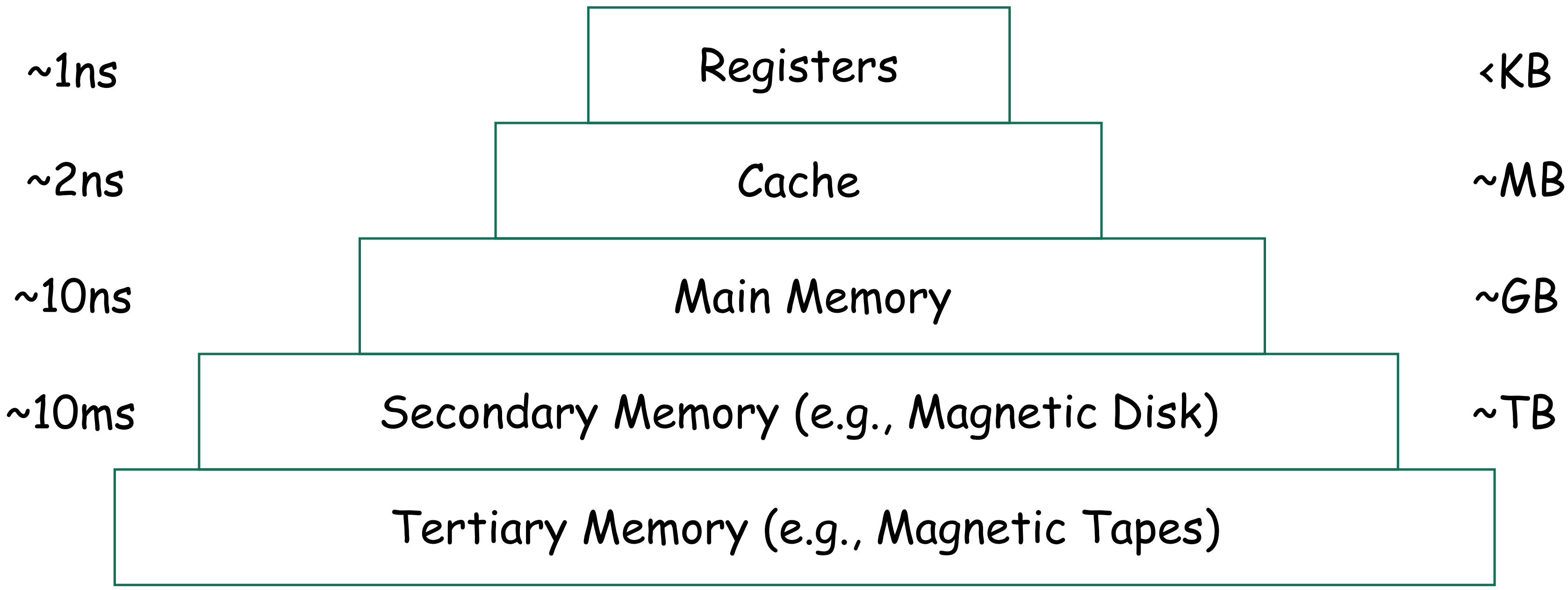
Department of Computer & Information Science

CUNY Brooklyn College

Acknowledgement

- These slides are a revision of the slides provided by the authors of the textbook via the publisher of the textbook

Memory Hierarchy



Outline

- Reliable and Efficiency
  - Redundancy and parallelism
  - RAID Structure

Mass Storage: Design Goals

- Function
  - They need to work, read & write
- Reliability
  - Murphy's law
    - "Anything that can go wrong will go wrong"
  - How do we make it appearing reliable?
- I/O Efficiency (performance)
  - We need it to be "fast"

Disk Failures and Data Loss

- Mean time between failures (MTBF)
  - The statistical mean time that a device is expected to work correctly before failing, see an example.
- Mean time to repair
  - Exposure time when another failure could cause data loss
- Mean time to data loss based on above factors (Why? See next)

Redundancy

- Mirroring: duplicate disk drive
  - Example: two physical disk drives are presented as a logical disk drive (mirrored volume)
- Disk failure
  - Failure of one physical disk does not result in data loss when the failed physical disk is replaced in time
- Data loss
  - Failure of two physical disk drives (at the same time, or before replacement of the first failed disk)
- Redundancy can reduce chances of data loss (greatly)

Redundancy and Data Loss: Factors

- Mean time between failure (MTBF) of a single disk drive and many disk drives
  - Example
    - MTBF of a single disk drive: 1,000,000 hours
    - 100 disk drives: 1,000,000/100 = 10,000 hours = 416.7 days
- Mean time to repair (MTTR): time required to replace the failure disk drive
- Mean time to data loss: time required to have a data loss (the second disk also failed before the failed one is repaired)

Redundancy and Data Loss: Example

- For two-disk mirroring case (Disk A and Disk B)
  - MTBF = 1,000,000 hours
  - MTTR = 10 hours
  - Data loss
    - Disk A failed first, and then disk B failed
    - Disk B failed first, and them Disk A failed
  - Mean time to data loss: failure of the mirrored volume (the 2nd disk drive also failed before the 1st could be replaced) : $1{,}000{,}000^2$ / (2 x 10) =5×10^{10} hours = 5.7×10^6 years!

Practical Consideration

- Disk failures are not independent
- Example
  - Large number of disk drive failures can be the result of a power failure and a tremor of a minor earthquake
  - Manufacturing defects in a batch of disk drives can lead to correlated failures
  - The probability of failure grows as disk drives age
- Mean time to data loss is smaller

Parallelism and Performance

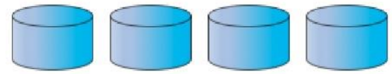
- Observation
  - Duplicating disk drives doubles the rate at which read requests can be handled
- Data stripping
  - Splitting data across multiple disk drives
  - Bit-level stripping
    - Splitting the bits of each byte across multiple disk drives
      - Example: using an array of 8 disks (or a factor of 8 or a multiple of 8)
  - Block-level stripping
    - Splitting blocks of a file across multiple disk drives
- Benefits
  - Increase the throughput of multiple small accesses (page accesses)
  - Reduce the response time of large accesses

RAID Structure

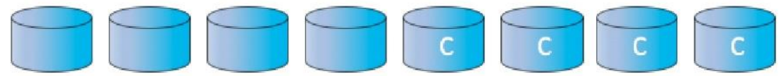
- **RAID** – **redundant array of inexpensive disks**
  - multiple disk drives provides reliability via **redundancy**
- Increases the **mean time to data loss**
- Frequently combined with **NVRAM** to improve write performance
- Several improvements in disk-use techniques involve the use of multiple disks working cooperatively

RAID Levels

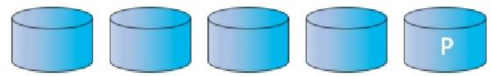
- Disk **striping** uses a group of disks as one storage unit
- RAID is arranged into six different levels
- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data
  - **Mirroring** or **shadowing** (**RAID 1**) keeps duplicate of each disk
  - Striped mirrors (**RAID 1+0**) or mirrored stripes (**RAID 0+1**) provides high performance and high reliability
  - **Block interleaved parity** (**RAID 4, 5, 6**) uses much less redundancy
- RAID within a storage array can still fail if the array fails, so automatic **replication** of the data between arrays is common
- Frequently, a small number of **hot-spare** disks are left unallocated, automatically replacing a failed disk and having data rebuilt onto them



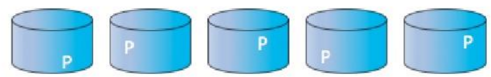
(a) RAID 0: non-redundant striping.



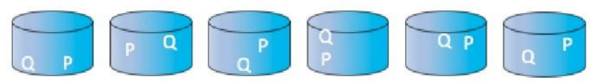
(b) RAID 1: mirrored disks.



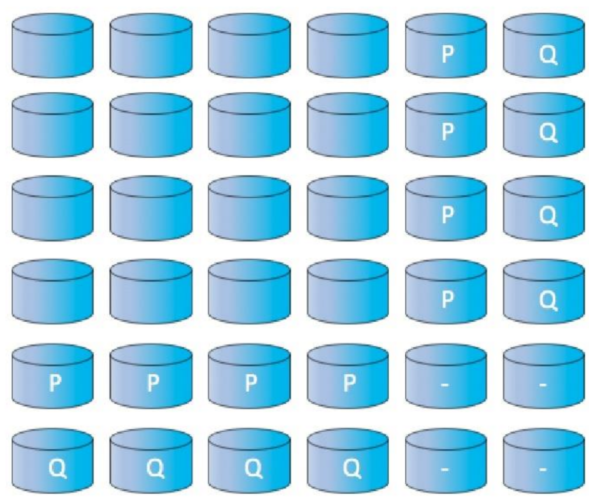
(c) RAID 4: block-interleaved parity.



(d) RAID 5: block-interleaved distributed parity.



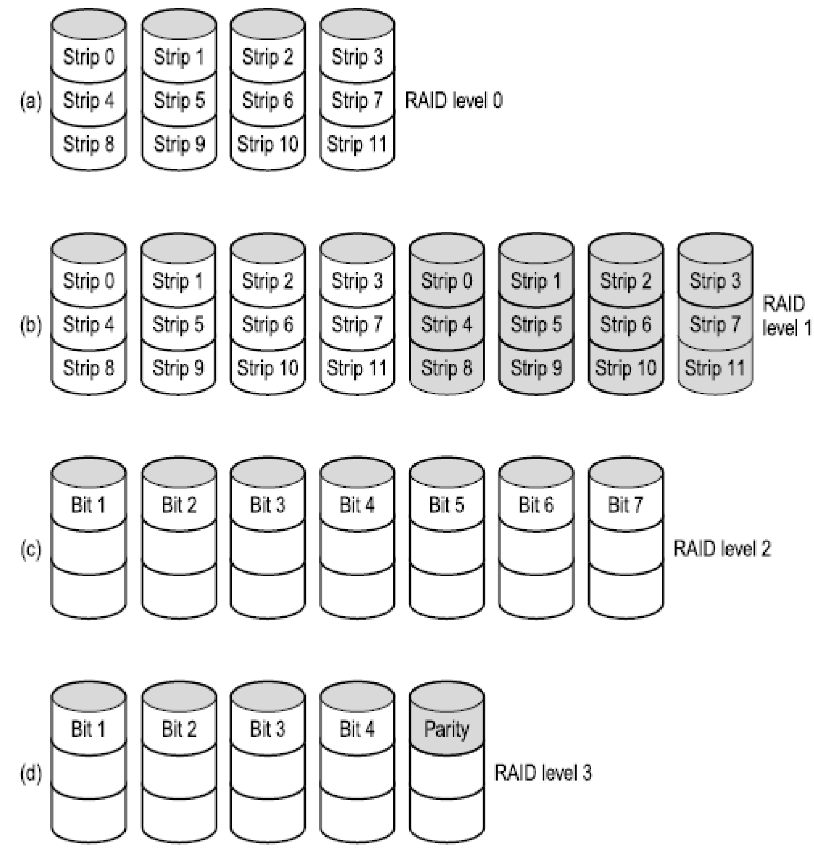
(e) RAID 6: P + Q redundancy.



(f) Multidimensional RAID 6.

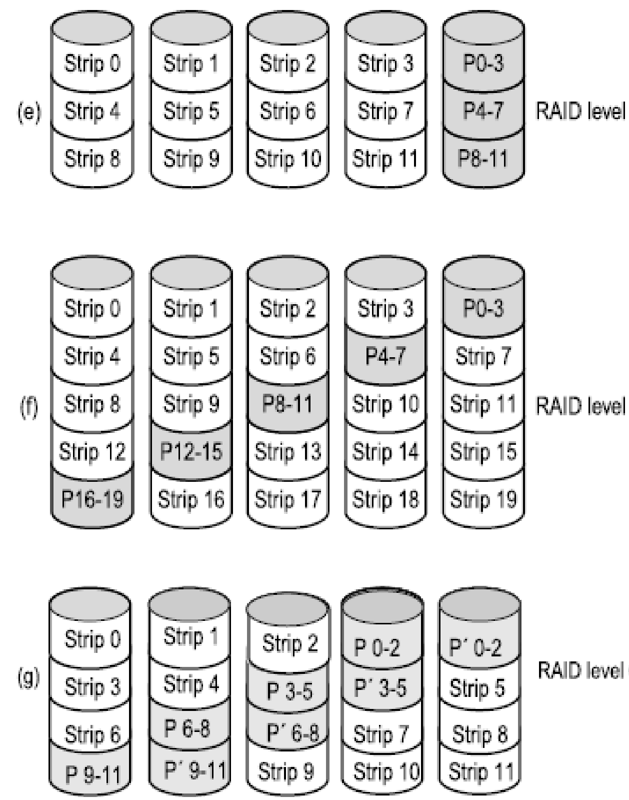
RAID Level 0 - 3

- Level 0 (not a true RAID): Stripping only
- Level 1: Mirror + stripping
- Level 2: Memory-style error-correction-code (ECC) organization
  - Example
    - Split a byte into 4-bit nibbles
    - Add Hamming code (3 bits) to each
    - One bit per drive onto 7 disk drives
- Level 3: Bit-interleaved parity organization
  - Compute a parity bit
  - Driver detects error, a parity bit is sufficient to correct error (unlike memory)
- Backup and parity drives are shown shaded.



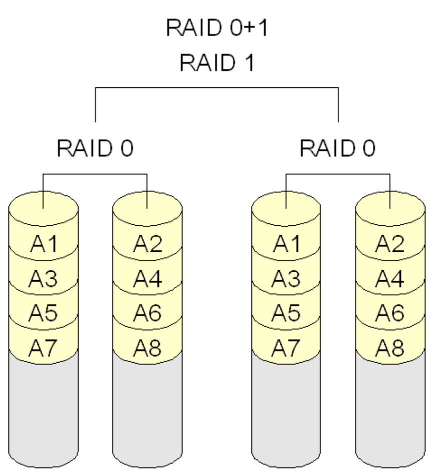
RAID Level 4 - 6

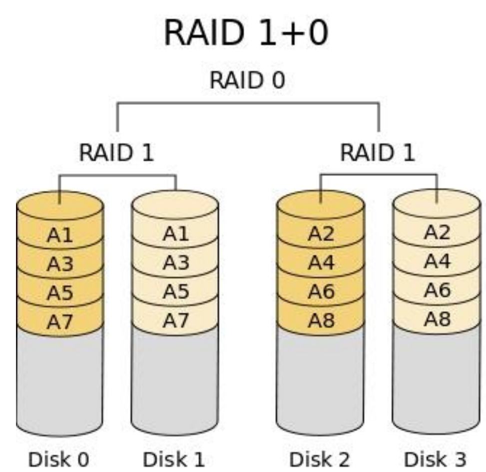
- Level 4: block-interleaved parity organization
  - Stripping
  - Compute parity for blocks
  - One disk for parities
- Level 5: block-interleaved distributed parity
  - Stripping
  - Compute parity for blocks
  - Parities are distributed
- Level 6: P+Q redundancy scheme
  - Extra redundant information to guard multiple disk failures
- Backup and parity drives are shown shaded.



RAID 0+1 and 1+0

- Combination of RAID levels 0 and 1
- RAID 0+1: stripping and then mirroring
- RAID 1+0: mirroring and then stripping





Selecting RAID Level

- RAID 0: high performance, data loss is not critical
- RAID 1: high reliability with fast recovery
- RAID 0+1 & 1+0: both performance and reliability
  - Expense: 2 for 1
- RAID 5:
  - Often preferred for large volumes of data
- Required number of disks?

Other Features

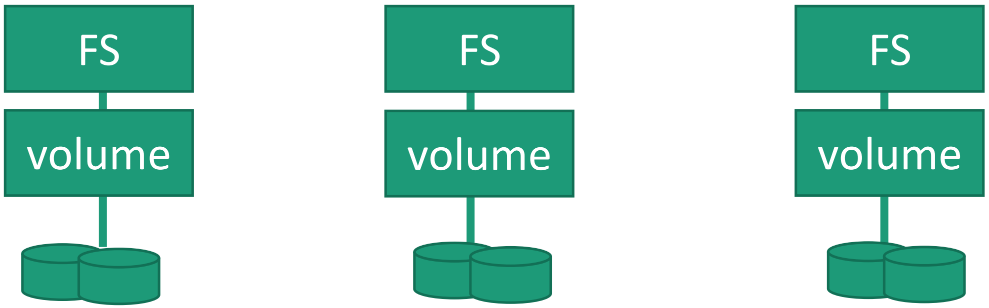
- Regardless of where RAID implemented, other useful features can be added
- **Snapshot** is a view of file system before a set of changes take place (i.e. at a point in time)
  - More in Ch 12
- Replication is automatic duplication of writes between separate sites
  - For redundancy and disaster recovery
  - Can be synchronous or asynchronous
- Hot spare disk is unused, automatically used by RAID production if a disk fails to replace the failed disk and rebuild the RAID set if possible
  - Decreases mean time to repair

Questions

- Reliability and performance
- Performance via parallelism
- Reliability via redundancy
- RAID
  - Which level to use? How many disks?

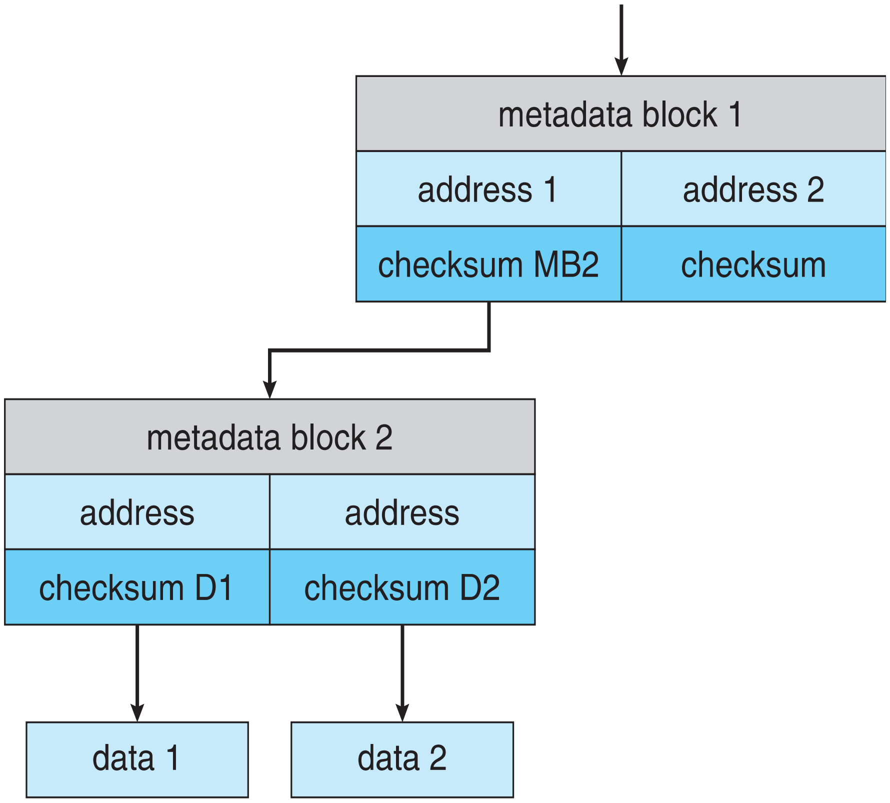
Limitation of RAID

- RAID alone does not prevent or detect data corruption or other errors, just disk failures
- RAID is not flexible
  - Present a disk array as a volume
  - What if a file system is small, or large, or change over time?



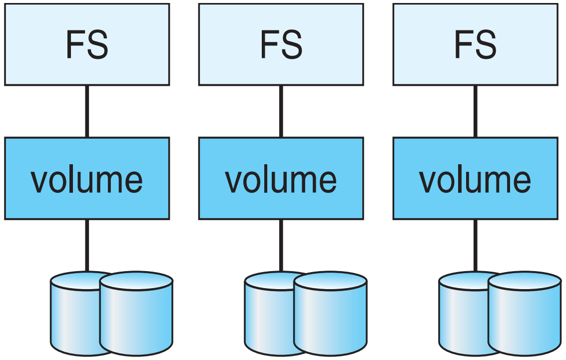
Extensions: Solaris ZFS

- Solaris ZFS adds **checksums** of all data and metadata
- Checksums kept with pointer to object, to detect if object is the right one and whether it changed
- Can detect and correct data and metadata corruption
- ZFS also removes volumes, partitions
  - Disks allocated in **pools**
  - Filesystems with a pool share that pool, use and release space like `malloc()` and `free()` memory allocate / release calls

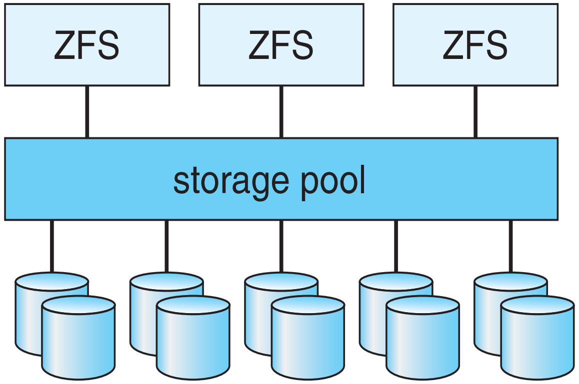


ZFS checksums all metadata and data

Traditional and Pooled Storage



(a) Traditional volumes and file systems.



(b) ZFS and pooled storage.

Object Storage

- General-purpose computing, file systems not sufficient for very large scale
- Another approach – start with a storage pool and place objects in it
  - Object just a container of data
  - No way to navigate the pool to find objects (no directory structures, few services
  - Computer-oriented, not user-oriented
- Typical sequence
  - Create an object within the pool, receive an object ID
  - Access object via that ID
  - Delete object via that ID
- Object storage management software like Hadoop file system (HDFS) and Ceph determine where to store objects, manages protection
  - Typically by storing N copies, across N systems, in the object storage cluster
  - Horizontally scalable
  - Content addressable, unstructured

Questions?

- Limitation of RAID?
- ZFS?
- Object storage