CSE 4/521
Introduction to Operating Systems
Lecture 21 – Mass Storage Structure III
(RAFT Structure, Stable-Storage Implementation)
Sumer 2018
Overview

Objective:
• To discuss operating-system services provided for mass storage, such as RAID

• RAID Structure
• Stable-Storage Implementation
Recap

• Disk Scheduling Algorithms:
  • FCFS
  • SSTF
  • SCAN (variant C-SCAN)
  • LOOK (variant C-LOOK)
Questions

1. Compare **efficiency**, **resource requirements** of SCAN and LOOK. (Easy)

2. Is SSTF an “**optimal**” disk scheduling algorithm? (Medium)

3. What is the fundamental **information available** to disk manager that memory manager does not have? (Intuitive)
Overview

• RAID Structure
• Stable-Storage Implementation
Overview

• RAID Structure

• Stable-Storage Implementation
RAID Structure

• RAID – redundant array of inexpensive independent disks
  • multiple disk drives provides reliability via redundancy
• Increases the mean time to failure
• Mean time to repair – exposure time when another failure could cause data loss
• Mean time to data loss based on above factors
• If mirrored disks fail independently, consider disk with 100,000 mean time to failure and 10 hour mean time to repair
  • Mean time to data loss is $100,000^2 / (2 \times 10) = 500 \times 10^6$ hours, or 57,000 years!
• Frequently combined with NVRAM to improve write performance
• Several improvements in disk-use techniques involve the use of multiple disks working cooperatively
RAID Structure

• Disk striping uses a group of disks as one storage unit
• RAID is arranged into 6 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
RAID Structure:
RAID Levels

(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 2: memory-style error-correcting codes.

(d) RAID 3: bit-interleaved parity.

(e) RAID 4: block-interleaved parity.

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

(g) RAID 6: P + Q redundancy.
RAID Structure: RAID 0

- Data is divided into blocks and is spread in a fixed order among all the disks in the array

- also known as disk striping

- + improves read and write performance via parallel access

- - does not provide any fault tolerance
RAID Structure: RAID 1

- All data written to the primary disk is written to the mirror disk
- provides a redundant, identical copy of all data
- also known as disk mirroring
- + provides fault tolerance
- + also generally improves read performance (but may degrade write performance)
- - doubles the storage requirement
RAID Structure: RAID 2

- Uses memory-style error correcting code (ECC) that employs disk-striping strategy that breaks a file into bits and spreads it across multiple disks
- Used Hamming Code method requires three extra disks for four data disks
- Provides fault tolerance
  - Can both detect & recover from single bit failures
  - Can detect but not correct double bit failures
RAID Structure:
RAID Level 3

• Uses **bit-interleaved parity**
• + requires only **one disk for parity** for 4 data disks
• - suffers from a write bottleneck, because all parity data is written to a single drive
• - RAID 2 & 3 cannot serve multiple requests simultaneously
RAID Structure: RAID Level 4

- Similar to RAID level 3, but it employs striped data in much larger blocks or segments (block-interleaved parity).
- RAID level 4 suffers from a write bottleneck (due to parity disk).
- Not as efficient as RAID level 5, because (as in RAID level 3) all parity data is written to a single drive.
RAID Structure: RAID Level 5

- Known as striping with parity (block-interleaved distributed parity)
- the most popular RAID level, replaced RAID 3 & 4
- similar to level 4 in that it stripes the data in large blocks across all the disks in the array
- It differs in that it writes the parity across all the disks
- The data redundancy is provided by the parity information
- The data and parity information are arranged on the disk array so that the two are always on different disks
RAID Structure: RAID Level 6

- Stores extra redundant info to recover from multiple disk failures (P+Q redundancy)
- would need 2 additional disks for each 4 data disks
  - more reliability versus less data space
- uses Reed-Solomon error correcting code

![RAID 6 Diagram](diagram.png)
RAID Structure: RAID (0+1) and (1+0)

- Combination of RAID 0 and 1
- Better performance and reliability, but doubles the disk storage requirement
RAID Structure: 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)
- **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
RAID Structure: Extensions

• RAID alone does not prevent or detect data corruption or other errors, just disk failures
• 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
RAID Structure:
ZFS Checksums All Metadata and Data
RAID Structure:
Traditional and Pooled Storage

(a) Traditional volumes and file systems.

(b) ZFS and pooled storage.
Overview

• RAID Structure
• Stable-Storage Implementation
Stable-Storage Implementation

• **Write-ahead log scheme** requires stable storage
• Stable storage means **data is never lost** (due to failure, etc)
• To implement stable storage:
  • Replicate information on more than one nonvolatile storage media with independent failure modes
  • Update information in a controlled manner to ensure that we can recover the stable data after any failure during data transfer or recovery
• Disk write has 1 of 3 outcomes
  1. **Successful completion** - The data were written correctly on disk
  2. **Partial failure** - A failure occurred in the midst of transfer, so only some of the sectors were written with the new data,
  3. **Total failure** - The failure occurred before the disk write started, so the previous data values on the disk remain intact
Stable-Storage Implementation

• If failure occurs during block write, recovery procedure restores block to consistent state
  • System maintains 2 physical blocks per logical block and does the following:
    1. Write to 1st physical
    2. When successful, write to 2nd physical
    3. Declare complete only after second write completes successfully

Systems frequently use NVRAM as one physical to accelerate
Credits for slides

Silberschatz, Galvin and Gagne

Tevfik Kosar