Fault Tolerance

Dependability and failures are real life concerns. Understanding how to detect and recover from failures is the topic of this lesson. RAID 0 - RAID 6 are discussed.

Dependability

Dependability = quality of a delivered service that justifies relying on the system to provide the service.
Specified Service = the expected system behavior
Delivered Service = the actual system behavior

System Modules have an ideal expected behavior. When the behavior deviates from the ideal the system to no longer provides the expected service.

Faults, Errors, Failures

Fault = module deviates from specified behavior
Error = actual behavior differs from expected behavior
Failure = system deviates from specified behavior

Faults, Errors, and Failures Example

Example: An ADD function works in all cases, except the case 5 + 3 = 7.

Latent Error = an error that occurs only when a specific task is performed.
Effective Error = a latent error that has occurred, an activated fault.
Failure = when the system deviates from the expected behavior.

A fault is needed to produce an error, but not every fault becomes an error.
For example: if the ADD function is never asked to produce the answer to 5 + 3, this is a fault that is never activated, so there is no error.

There can be an error in a system, and never have a failure.
For example: The ADD function produces the answer 5 + 3 = 7, but the answer is never used. This is an error that is not a failure.

Reliability and Availability

Reliability can be measured.
To measure reliability consider the system to be in one of two states

- Service accomplishment - normal state, providing the service expected
- Service interruption - the service not being provided

Reliability = a measure of the continuous service accomplishment

Mean Time To Failure (MTTF) = how long will the system provide service before the next service interruption.

Availability measures service accomplishment as a fraction of overall time.

For example: if a system provides service for one year and has service interruption for one year.

The availability for the system is: 50%
The reliability for the system is 1 year.

If the system provides one month of service, then one month of service interruption for two years.

The availability is 50%
the reliability is 1 month

Mean Time to Repair (MTTR) when a service interruption occurs, how long until service is restored.

Availability = MTTF/ (MTTF + MTTR)

Kinds of Faults

Faults Classified by Cause:

Hardware Fault - hardware components fail to perform as designed
Design Faults - software bugs, hardware design
Operation Fault - operator and user mistakes
Environmental Fault - fire, power failure, etc

Some faults may not result in errors.

Fault Classified by Duration

Permanent Fault - cannot not be corrected
Intermittent Fault - recurring fault
Transient Fault - fault occurs and does not occur again

Improving Reliability and Availability

Fault Avoidance

Prevent faults from occurring

Fault Tolerance

Prevent faults from becoming failures. Use ECC (error correction code)

Speed Up Repair (availability is improved)

Fault Tolerance Techniques

Checkpointing - used for transient and intermittent faults

-The state of the system is periodically saved

-When an error is detected the system is restored to the correct state

If checkpointing and system restore takes too long, then this considered a service interruption

2-Way Redundancy

-Two modules do the same work

- outcomes are compared

-Roll back if the outcomes are different

This method requires a system recovery technique

3-way Redundancy

-3 or more modules do the same work

-if the outcomes are different, the majority wins

This method is expensive, but it can tolerate a fault in one module.

N-Module Redundancy

N = number of modules

N=2 Dual Module Redundancy - detects but does not correct 1 faulty module

N=3 Triple Module Redundancy - corrects 1 faulty module

N=5 Five Module Redundancy - detects and corrects up to 2 modules

Fault Tolerance for Memory and Storage

Error Detection and Error Correction Codes

-Parity: add one extra bit it is the XOR of the data bits.

If a bit is in error, the parity bit will detect it

-ECC: this code can detect and correct a single bit. It can also detect two bit errors, but it cannot fix them.

-RAID: Redundant Array of Independent Disks

RAID

Redundant Array of Independent Disks

-several disks are used in place of one disk

-each disks detects errors using ECC

Goal of RAID - better performance

- Read/Write accomplishment even when there is a bad sector or when a disk fails.

RAID 0

Uses striping to improve performance.
RAID 0 takes 2 disks and makes it look like 1 disk.
The advantages of this are:
-Twice the data throughput of a single disk
-Less queuing delay
Disadvantage
-Reliability is worse

RAID 0 Reliability

f = failure rate for a single disk

For a single disk MTTF = 1/f
N Disks in RAID 0

RAID 1 Mirroring

A second disk is a copy of the first disk.
The write performance is the same as for 1 disk.
The read performance is twice the performance of one disk.

Tolerates any faults that affects 1 disk.

RAID 1 Reliability

2 Disks in RAID 1
MTTDL = (MTTF / 2) + MTTF

RAID 1 Reliability if Disks are Replaced

$MTTDL = (MTTF_1 / 2) * (MTTF_1 / MTTR_1)$

RAID4

The disks are block interleaved.
For N disks in RAID4
-N-1 disks contain striped data like RAID0
-1 disk has parity blocks

A damaged disk that cannot be corrected with ECC can be reconstructed by using the parity bit and the data values in the other disks.

RAID4 is a more general technique that mirroring.
A write to a RAID4 must write to the required disk and to the parity disk
A read just reads the required disk

RAID4 Performance and Reliability

Read Performance - the same throughput as N-1 disks

Write Performance - ½ throughput of 1 disk. This is the primary reason why RAID5 is used.

MTTF -

-if all the disks are operational: (MTTF of a single disk) / N

-if a disk fails, and NO repair is done:

(MTTF of a single disk) / N + ((MTTF of a single disk) / (N-1))

-if the repair is done:

(MTTF of a single disk) / N * (MTTF of a single disk) / (N-1) / (MTTR of 1 disk)

The MTTF of RAID4: (MTTF of 1 disk * MTTF of 1 disk) / (N * (N-1)*(MTTR of 1 disk)

RAID4 Write

When doing a write, the data and parity bit must both be written.

To update the parity bit without reading all the data bits:

-XOR the old data with the new data

-XOR this result with the parity bit

-The final result of this XOR is the new parity bit

The parity disk will be a bottleneck because every read and write must access the parity bit.

RAID 5

Distributed block-interleaved parity

The parity is spread throughout all the disks, there is no dedicated parity disk.

Read Performance = N * Throughput of one disk

Write Performance = N/4 * Throughput of 1 disk

RAID 6

Two Parity Blocks/ Group - similar to RAID 5 but with two parity bits per group

RAID 6 can still work with two failed stripes

The two parity blocks are different -

-one is a parity bit

-second is a check block

If one disk fails - use the parity bit

If two disks fail - use equations to reconstruct the data

The overhead is much higher than for RAID 5 and the probability of a second disk failing before the first failed disk is repaired is very small.