Storage → Fault Tolerance → Multi-Processing

Topics:
- Faults, Errors, Failures
- Reliability + Availability
- MTTF (Service Accomplishment)
- MTTR (Service Interruption)
- Fault Detection + Recovery
- N-Module Redundancy
- RAID 0, 1, 4, + 5

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. Dependability is about: "can we expect the Delivered Service to match the Specified 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. (aka, the module is no longer "dependable")

System "Modules" are the larger components of a computer system: processor, memory, etc

Faults, Errors, Failures

escalation ↓

Fault = module deviates from specified behavior Also known as "Latent Error". See definition below. A "Fault" is the event that results in an Error. The Cause to the effect.

Error = actual behavior differs from expected behavior Also known as "Activated Fault" or "Effective Error". See definition below.

Failure = system deviates from specified behavior

A Fault CAN (but doesn't always) produce an error, and an error CAN (but doesn't always) produce a Failure.

A Failure cannot occur without being caused by an Error, and an Error cannot occur without being connected to a Fault.

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. Latent error = "potential" error that can occur

Effective Error = a latent error that has occurred, an activated fault. Effective error = a "realized" latent error

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" is quantitative while "Dependability" is qualitative.

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

A typical measure for "Reliability" is Mean Time To Failure (MTTF).

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

Availability = (Service Accomplishment) / [(Service Accomplishment) + (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. (MTTF)

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.

Remember:
- MTTF measures how long the system is expected to be in "Service Accomplishment" mode.
- MTTR measures how long the system is expected to be in the "Service Interruption" mode.

Availability = MTTF/ (MTTF + MTTR)

The denominator (MTTF + MTTR) = total time
So Availability = (Service Accomplishment Time) / (Overall Time)

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 ex: no coffee allowed in the server room.

Fault Tolerance

Prevent faults from becoming failures. ~~Use ECC (error correction code)~~ ex: Redundancy. For example, ECC is used on memory reads to ensure that the data is valid/detect if a bit is incorrectly flipped and if so, correct it.

Speed Up Repair (availability is improved)
ex: Spare Hard Drive in a drawer

Fault Tolerance Techniques

Recovery Technique

Checkpointing - used for transient and intermittent faults

This is a "system recovery technique", but it needs an "error detection" technique, like 2-Way Redundancy.

-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

Detection Technique

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

bc 2-Way Redundancy is just an error detection technique, but it doesn't provide us with any Recovery mechanism. How would we know which one of the two is the correct outcome? We can use Checkpointing as the "system recovery technique" once we detect the error.

Detection & Recovery Technique

3-way Redundancy Can serve as both a "Detection" & "Recovery" technique

-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

this is a more general, redundancy-based approach

N = number of modules

"DMR" N=2→Dual Module Redundancy - ~~detects but does not correct 1 faulty module~~ Can guarantee Detection (but not Correction) of 1 Faulty Module.

"TMR" N=3→Triple Module Redundancy - corrects 1 faulty module Can guarantee Detection & CORRECTION of 1 Faulty Module.

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

An example of 5 Module Redundancy is the Space Shuttle. 5 Computers are performing the same thing and they vote on the results. If 1 wrong result in a vote, we can still continue with normal operation. If there are 2 wrong votes, it'll abort the mission, but this isn't considered a failure because 3 computers still outvote the 2. We abort because if an additional computer fails (causing 3 wrong results), we can no longer recover.

Fault Tolerance for Memory and Storage

Error Detection and Error Correction Codes

"DMR" & "TMR" are overkill for Memory & Storage. The techniques listed here would work better in this case. With "Error Detection" and "Error Correction Codes", we store extra information that allows us to detect and/or correct one or more bits of error.

-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

If a Fault flips a bit, then the Parity no longer matches the data. It can only DETECT an ODD # of BIT ERRORS (1, 3, 5, etc.). If there are an even # of bit errors (ex: 2 bit errors), then the parity bit will the same and the detection will fail. Parity also does not tell you WHICH bit failed, so it doesn't provide any Recovery mechanism either.

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

Ex: "SECDED" (Single Error Correction, Double Error Detection) - Can detect and correct any 1 bit flip. If there are 2 bit flips, we can detect it but cannot correct it. Used in DRAM modules.

-RAID: Redundant Array of Independent Disks

RAID

Redundant Array of Independent Disks

-several disks are used in place of one disk

Several disks play the role of 1 disk. They can either pretend to be a LARGER disk, OR they can pretend to be a MORE RELIABLE disk, OR a MIX of both.

-each disks detects errors using ECC

Each of the Disks can independently detect when they have an error, and so in the overall RAID scheme, we can tell which of the disks has an error.

~~Goal of RAID - better performance~~

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

Goals of RAID:
- Better Performance
- (Better Reliability). We want normal read/write accomplishment (aka "service accomplishment"), even when there are bad sectors on some disks, or if an entire disk fails.

Not all RAID techniques improve both of these goals, some only improve performance and some only improve reliability.

On a single disk, there are multiple tracks. However, the problem is that while the head is positioned to read a certain track (ex: track 0), it cannot read another track (ex: track 1). Also we'd need to serialize access to the tracks (as they're stacked sequentially), so if we wanted to read tracks 0, 17, & 27, we'd need to figure out the order in which to move the head assembly.

RAID 0 takes two disks and makes them look like a single disk by splitting the tracks across the disks, which we'd call "Stripes" (ex: Disk A has "stripe" 0, 2, 4, etc. while Disk B has "stripe" 1, 3, 5, etc.). So "track" is the name for a "physical"/actual ring on a disk, while "stripe" is a "logical" track in a RAID 0 configuration. So stripe 1 in the example would be located on Track 0 on Disk B.

RAID 0 IMPROVES PERFORMANCE, REDUCES RELIABILITY

Stripe 0	Stripe 1
Stripe 2	Stripe 3
Stripe 4	Stripe 5

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 = # of Failures per Disk per Second. Should be a very small (< 1) number. We'd assume the Failure Rate is constant over time.

For Disk, MTTF is also known as Mean Time To Data Loss (MTTDL). For a SINGLE disk, MTTDL = MTTF

For a single disk $MTTF_1 = 1/f_1$

$MTTDL_1 = MTTF_1$

N Disks in RAID 0 <- For RAID 0 with "N" disks, the failure rate is:

$$f_N = (N)(f_1) = (N)\left(\frac{1}{MTTF_1}\right)$$

$$MTTF_N = \frac{MTTF_1}{N}$$

<- For "N" disks, this is the mean time that 1 of them will fail. This means that in a RAID 0, the more disks (greater "N") reduces the MTTF. Ex: 2 Disks in RAID 0 will have half the MTTF as 1 disk.

RAID 1 Mirroring SLIGHTLY IMPROVES PERFORMANCE, DRAMATICALLY IMPROVES RELIABILITY

A second disk is a copy of the first disk.

The write performance is the same as for 1 disk. (2 writes occur simultaneously)

The read performance is twice the performance of one disk.

Tolerates any faults that affects 1 disk.

Ex: if there's a bad sector on Disk A, then we can just read the corresponding/mirrored sector on Disk B.

RAID 1 Reliability

Remember: $f_N = (N)(f_1)$

2 Disks in RAID 1

Note: This assumes that when the first disk fails, we don't fix or replace it!

MTTDL = (MTTF / 2) + MTTF

Explained: Both of the 2 Disks in RAID 1 are okay until one disk fails at: $\frac{MTTF_1}{2}$

Then, the remaining disk lives on for: $MTTF_1$

So combined, the MTTDL for 2 Disks in RAID 1 = $\left(\frac{MTTF_1}{2}\right) + MTTF_1 = MTTDL_{RAID1-2}$

RAID 1 Reliability if Disks are Replaced (as soon as possible)

As we can see above ^, adding an extra Disk in RAID 1 only gives us an additional 1.5x the MTTF of a single disk. However, that assumes that after the first disk failure, we ignore it and don't do anything. Realistically, we'd detect the first failure and immediately try to replace it, so the extra 0.5x of a single disk-MTTF buys us some time to replace the failed disk.

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

$$MTTDL_{RAID1-2} = \frac{MTTF_1}{2} * \frac{MTTF_1}{MTTR_1} = \frac{(MTTF_1)^2}{2(MTTR_1)}$$

MTTF is usually significantly larger than MTTR, which means that our MTTF/MTTDL for 2 Disks in RAID 1 is MUCH larger than the MTTF/MTTDL of a single disk (if we replace after the first failure).

RAID4

The disks are block interleaved.

For N disks in RAID4 For N disks in RAID4, (N-1) Disks are used together like RAID0 and then 1 Disk has the parity blocks (meant to protect the data contained in the other disks).

- -N-1 disks contain striped data like RAID0
- -1 disk has parity blocks

Stripe 0	⊕	Stripe 1	⊕	Stripe 2	=	Parity 0,1,2
Stripe 3	⊕	Stripe 4	⊕	Stripe 5	=	Parity 3,4,5

⊕ XOR

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. If 1 disk fails, we can use the other Stripes (from the same row) and the Parity of the row to get back the lost Stripe. Ex: Lose Stripe 3, use Stripe 4, 5, and "Parity 3,4,5" to reconstruct.

RAID4 is a more general technique that mirroring. Mirroring is RAID4 with N = 2. The Parity is just copy (mirror) of the original data.

A write to a RAID4 must write to the required disk and to the parity disk

A read just reads the required disk

RAID4 Capacity = (N - 1) * (Capacity of 1 disk)

RAID4 Performance and Reliability

Read Performance - the same throughput as N-1 disks Only reading from Data disks, not the Parity disk. Read throughput = (N-1)*(throughput of a single disk)

Write Performance - ½ throughput of 1 disk. This is the primary reason why RAID5 is used. B/c writes require a read of old Parity, update it, and then write it back. So 2 accesses for every write.

MTTF -

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

2a. -if a disk fails, and NO repair is done: This MTTF is NOT better than that of a Single Disk. So you never want to do this option. Do option 2b instead.

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

2b. -if the repair is done: $= \left(\frac{MTTF_1}{N}\right)\left[\frac{\left(\frac{MTTF_1}{N-1}\right)}{MTTR_1}\right]$

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

2b = The MTTF of RAID4: (MTTF of 1 disk * MTTF of 1 disk) / [(N * (N-1)*(MTTR of 1 disk)] $= \frac{(MTTF_1)^2}{(N)(N-1)(MTTR_1)}$

Side-note, this is also the MTTF of RAID5!

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 XORing the old data with the new data will tell us which bits of the data have changed.
- -XOR this result with the parity bit A flip in the Data (previous step) will result in a flip in the Parity, which gives us the new Parity
- -The final result of this XOR is the new parity bit

So we have to do 2 reads per write: the stripe that we're trying to overwrite (to get the old data) and the parity.

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

Read is better than RAID4 bc all N disks in RAID5 can read data. Write is better than RAID4 bc RAID4 write is always limited to HALF the throughput of 1 disk.

RAID5 is ALWAYS better than RAID4. Same cost, better throughput!

RAID 5

Data Capacity = (N - 1) * (Capacity of 1 Disk) <-- This is the same as a RAID4 Capacity.

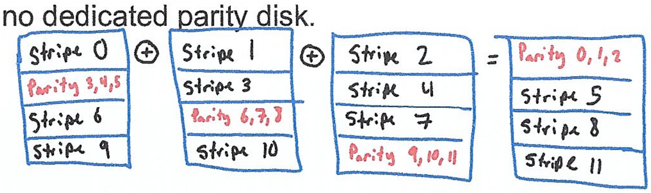
Distributed block-interleaved parity (Similar to RAID4, but the parity is spread among all disks instead of all on 1 disk) Reliability of RAID5 is the SAME as RAID4.

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

Here, the 4 is because of the 4 accesses we have to do: 1 read of old data, 1 read of old parity, 1 write of new data, & 1 write of new parity



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

RAID6 has 2x more overhead than RAID5 (we pay 2x the cost for increased reliability). Write overhead goes from 4 accesses (RAID4 & RAID5) to 6 accesses (RAID6). RAID6 is only really good for the scenario where if 1 disk fails and there's a good chance that a second disk will also fail before the 1st failed disk is replaced.

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.

RELIABILITY AND AVAILABILITY QUIZ

- HARD DISK
  - WORKS FINE FOR 12 MONTHS
  - BREAKS (CAN'T SPIN), TAKES 1 MONTH TO REPLACE MOTOR
  - WORKS FINE FOR 4 MONTHS
  - BREAKS (CAN'T MOVE HEADS), TAKES 2 MONTHS TO UNSTUCK
  - WORKS FINE FOR 14 MONTHS
  - BREAKS (HEAD BROKEN), WOULD TAKE 3 MONTHS TO FIX
  - THROW AWAY, BUY NEW DISK

MTTF: $\boxed{10}$ MONTHS   MTTR: $\boxed{2}$ MONTHS

AVAILABILITY: $\boxed{83.33}$ %

MTTF = (12+4+14) / 3 = 10

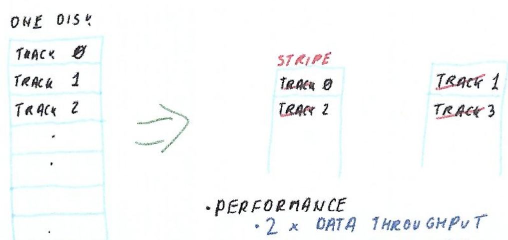
MTTR = (1 + 2 + 3) / 3 = 2

Availability = 10 / (10 + 2) = 0.8333 * 100 = 83.33

FAULT CLASSIFICATION QUIZ

- PHONE GETS WET, HEATS UP, EXPLODES
  - GOT WET IS A transient FAULT (BY DURATION) AND environmental FAULT (BY CAUSE)
  - WAS SUPPOSED TO PREVENT ITSELF FROM OPERATING WHEN WET, HEATING UP IS A RESULT OF A permanent FAULT (BY DURATION) AND design FAULT (BY CAUSE)

RAID Ø: STRIPING TO IMPROVE PERFORMANCE



- PERFORMANCE
  - 2 x DATA THROUGHPUT
  - LESS QUEUING DELAY
- RELIABILITY - WORSE THAN 1 DISK

RAID Ø RELIABILITY

- f - FAILURE RATE FOR A SINGLE DISK
  - FAILURES/DISK/SECOND
- SINGLE-DISK $MTTF = \frac{1}{f}$ $MTTDL_1 = MTTF_1$ (MEAN TIME TO DATA LOSS)

- N DISKS IN RAID Ø
  $f_N = N \cdot f_1 \implies MTTF_N = MTTDL_N = \frac{MTTF_1}{N}$

  2 DISKS $\implies MTTF_2 = \frac{MTTF_1}{2}$

RAID 0 QUIZ

- RAID 0 ARRAY WITH FOUR DISKS
- ONE DISK: 200 GB, 10 MB/s THROUGHPUT, MTTF = 100,000 h

OUR RAID 0 ARRAY:

- CAN STORE 800 GB
- THROUGHPUT 40 MB/s
- MTTF 25000 h

RAID 1 RELIABILITY

- f - FAILURE RATE FOR A SINGLE DISK
  - FAILURES/DISK/SECOND
- SINGLE-DISK $MTTF = \frac{1}{f}$ $\quad MTTDL_1 = MTTF_1$ MEAN TIME TO DATA LOSS
- 2 DISKS IN RAID 1

$f_N = N \cdot f_1 \Rightarrow$ BOTH DISKS OK UNTIL $\frac{MTTF_1}{2}$

REMAINING DISK LIVES ON FOR $MTTF_1$

$MTTDL_{RAID1-2} = \frac{MTTF_1}{2} + MTTF_1$ ASSUMES NO DISK REPLACED

- BUT WE DO REPLACE FAILED DISKS! MTTF?

RAID 1 RELIABILITY IF FAILED DISKS REPLACED?

- f - FAILURE RATE FOR A SINGLE DISK
  - FAILURES/DISK/SECOND
- SINGLE-DISK $MTTF = \frac{1}{f}$
- BOTH DISKS OK FOR $\frac{MTTF_1}{2}$
- DISK FAILS, HAVE ONE OK DISK FOR $MTTR_1$
- BOTH DISKS OK AGAIN $\Rightarrow \frac{MTTF_1}{2}$

$MTTDL_{RAID1-2} = \frac{MTTF_1}{2} \times \frac{MTTF_1}{MTTR_1}$

$MTTR_1 << MTTF_1$

PROBABILITY OF SECOND DISK FAILING DURING $MTTR_1$

$= \frac{MTTR_1}{MTTF_1}$

RAID 1 QUIZ

- RAID1 ARRAY WITH TWO DISKS
- ONE DISK: 200 GB, 10 MB/s THROUGHPUT, MTTF = 100,000 h
- WE REPLACE FAILED DISK, MTTR = 24 h
- OUR TWO-DISK RAID 1 ARRAY HAS
  - DATA CAPACITY 200 GB
  - THROUGHPUT 13.33 MB/s (50% RD, 50% WR)
  - MTTF = 208333333 h

Why is throughput 13.33 MB/s and NOT 15?
With RAID1, we get a write throughput of 10 MB/s and a read throughput of 20 MB/s.
For any given second, we will be spending 1/3 of a second on the reads and 2/3 of a second on the writes- this way we get to do the same number of reads and writes per second (it's just that the reads are twice as fast as the writes).

2/3 of a second spent on writes. Reads take half as long as writes, so half of 2/3 is 1/3. Together, they'll make 3/3 = 1 second.

It is NOT 15 MB/s, because that assumes that for half a second we are reading, and for the other half a second we are writing. But if that was the case, then we'd end up reading TWICE the amount of data than we would for our writes. So if the workload is 50/50 in terms of accesses, then it is NOT 50/50 in terms of time spent on reads and time spent on writes. Reads are faster, thus we spend less time on them.

Example of Solving RAID Throughput

Lets say we have 5 MBs to write and another 5 MBs to read. How long would it take for us to do this? The write would take us 0.5 seconds (5 MB / 0.5 s = 10 MB / 1 s). The read would take us 0.25 seconds since we can do the reads twice as fast as the writes in RAID 1. So for 10 MBs total (with 5 MBs of write and 5 MBs of read), it would take us 0.5 + 0.25 = 0.75 seconds.
10 MBs / 0.75 seconds = 13.33 MBs / 1 sec

Another method of solving it:

We want to Read & Write the same amount of data: X MB
We know the Read Throughput, R_t = 20 MB/s
The Write Throughput, W_t = 10 MB/s
We can describe the Read/Write Throughput relationship as: R_t = 2W_t

So we want to know, in 1 second, how many MBs we can read & write (if the amount we read & write are the same). In other words:

$$\frac{X\,MB}{R_t} + \frac{X\,MB}{W_t} = 1\ second$$

This makes sense because the units of R_t and W_t are MB/s, so all the units match up. We want to solve for X. Replacing R_t and W_t with their values:

$$\frac{X\,MB}{10\frac{MB}{s}} + \frac{X\,MB}{20\frac{MB}{s}} = 1\ second$$

Or

$$\frac{2X\,MB}{20\frac{MB}{s}} + \frac{X\,MB}{20\frac{MB}{s}} = 1\ second$$

$$\frac{3X\,MB}{20\frac{MB}{s}} = 1\ second$$

$$\frac{3X\,MB}{20\frac{MB}{s}} = 1\ second$$

$$3X\,MB = 20\left(\frac{MB}{s}\right) * second$$

Seconds cancel out

$$3X\,MB = 20\,MB$$

X = 20/3 = 6.6667

Remember that X is both the throughput for the reads and writes INDIVIDUALLY. We need to combine them to get the total throughput:

Total Throughput = 2X = 2 * (20/3) = 13.3333

Solving RAID 1 MTTF

MTTDL = (MTTF_1 / 2) * (MTTF_1 / MTTR_1)
= (100,000 / 2) * (100,000 / 24) = 208,333,333

RAID4 QUIZ

- RAID4 ARRAY WITH FIVE DISKS
- ONE DISK. 200 GB, 10 MB/s THROUGHPUT, MTTF = 100,000 h
- WE REPLACE FAILED DISK, MTTR = 24 h
- OUR FIVE-DISK RAID4 ARRAY HAS
  - DATA CAPACITY 800 GB
  - THROUGHPUT 888 MB/s (50% RD, 50% WR)
  - MTTF = 20833333 h

RAID 4 Capacity = (Capacity of 1 Disk)(N − 1) = (200 GB)(5 − 1) = 800 GB

Throughput

RAID4 Read Throughput = R_t = (N−1)(Throughput of 1 Disk) = (5−1)(10 MB/s) = 40 MB/s

RAID4 write Throughput: W_t = 1/2 (Throughput of 1 Disk) = (1/2)(10 MB/s) = 5 MB/s

$$\left.\begin{array}{l} X_w = \text{Amount of data to write} \\ X_r = \text{Amount of data to read} \\ X_T = \text{Total data} = X_w + X_r \end{array}\right\} MB$$

Since we have 50% Read + 50% write, then:

$$X_w = \frac{X_T}{2} \;\;+\;\; X_r = \frac{X_T}{2} \;\;+\;\; X_w = X_r$$

We can describe the data throughput in 1 second with the following eqn:

$$\frac{X_w\, MB}{W_t\, \frac{MB}{s}} + \frac{X_r\, MB}{R_t\, \frac{MB}{s}} = 1s$$

$$\frac{X_w\, \cancel{MB}}{W_t\, \frac{\cancel{MB}}{s}} + \frac{X_r\, \cancel{MB}}{R_t\, \frac{\cancel{MB}}{s}} = 1s$$

$$\frac{X_w}{W_t}s + \frac{X_r}{R_t}s = 1s$$, where $\frac{X_w}{W_t}$ + $\frac{X_r}{R_t}$ are the time we spend on writes + reads, respectively

Substitution:

$$\frac{\frac{X_T}{2}}{W_t}s + \frac{\frac{X_T}{2}}{R_t}s = 1s$$

$$\frac{X_T}{2W_t}s + \frac{X_T}{2R_t}s = 1s$$

$$\frac{X_T}{2(5)}s + \frac{X_T}{2(40)}s = 1s$$

$$\frac{X_T}{10}s + \frac{X_T}{80}s = 1s$$

$$\frac{8X_T}{80}s + \frac{X_T}{80}s = 1s$$

$$\frac{9X_T}{80}\cancel{s} = 1\cancel{s}$$

$$X_T = \frac{80}{9} = 8.88$$

So our total throughput in 1 second is 8.88 MB/s

$$\text{MTTF (RAID4)}: \frac{(MTTF_1)^2}{(N)(N-1)(MTTR_1)} = \frac{(100{,}000\,h)^2}{(5)(5-1)(24\,h)} = \frac{(10{,}000{,}000{,}000\,h^{\cancel{2}})}{(480\,\cancel{h})} = 20{,}833{,}333.33\,h$$

RAID5 QUIZ

- RAID5 ARRAY WITH FIVE DISKS
- ONE DISK: 200 GB, 10 MB/s THROUGHPUT, MTTF = 100,000 h
- WE REPLACE FAILED DISK, MTTR = 24 h
- OUR FIVE-DISK RAID5 ARRAY HAS
  - DATA CAPACITY 800 GB
  - THROUGHPUT 20 MB/s (50% RD, 50% WR)
  - MTTF = 20833333 h

RAID 5 Capacity = (Capacity of 1 Disk)(N − 1) = (200 GB)(5 − 1) = 800 GB

Throughput

RAID5 Read Throughput = R_t = (N)(Throughput of 1 Disk) = (5)(10 MB/s) = 50 MB/s

RAID 5 write Throughput: W_t = $\left(\frac{N}{4}\right)$(Throughput of 1 Disk) = $\left(\frac{5}{4}\right)$(10 MB/s) = 12.5 MB/s

$$\left.\begin{array}{l} X_w = \text{Amount of data to write} \\ X_r = \text{Amount of data to read} \\ X_T = \text{Total data} = X_w + X_r \end{array}\right\} MB$$

Since we have 50% Read + 50% write, then:

$$X_w = \frac{X_T}{2} \;\;+\;\; X_r = \frac{X_T}{2} \;\;+\;\; X_w = X_r$$

We can describe the data throughput in 1 second with the following eqn:

$$\frac{X_w\, MB}{W_t\, \frac{MB}{s}} + \frac{X_r\, MB}{R_t\, \frac{MB}{s}} = 1s$$

$$\frac{X_w\, \cancel{MB}}{W_t\, \frac{\cancel{MB}}{s}} + \frac{X_r\, \cancel{MB}}{R_t\, \frac{\cancel{MB}}{s}} = 1s$$

$$\frac{X_w}{W_t}s + \frac{X_r}{R_t}s = 1s$$, where $\frac{X_w}{W_t}$ + $\frac{X_r}{R_t}$ are the time we spend on writes + reads, respectively.

Substitution:

$$\frac{\frac{X_T}{2}}{W_t}s + \frac{\frac{X_T}{2}}{R_t}s = 1s$$

$$\frac{X_T}{2W_t}s + \frac{X_T}{2R_t}s = 1s$$

$$\frac{X_T}{2(12.5)}s + \frac{X_T}{2(50)}s = 1s$$

$$\frac{X_T}{25}s + \frac{X_T}{100}s = 1s$$

$$\frac{4X_T}{100}s + \frac{X_T}{100}s = 1s$$

$$\frac{5X_T}{100}\cancel{s} = 1\cancel{s}$$

$$X_T = \frac{100}{5} = 20$$

So our total throughput in 1 second is 20 MB/s

$$\text{MTTF (RAID}\overset{5}{\cancel{4}}\text{)}: \frac{(MTTF_1)^2}{(N)(N-1)(MTTR_1)} = \frac{(100{,}000\,h)^2}{(5)(5-1)(24\,h)} = \frac{(10{,}000{,}000{,}000\,h^{\cancel{2}})}{(480\,\cancel{h})} = 20{,}833{,}333.33\,h$$