6.033 Spring 2021

Lecture #14: Reliability

building reliable systems from unreliable components

Katrina LaCurts | lacurts@mit.edu | 6.033 2021

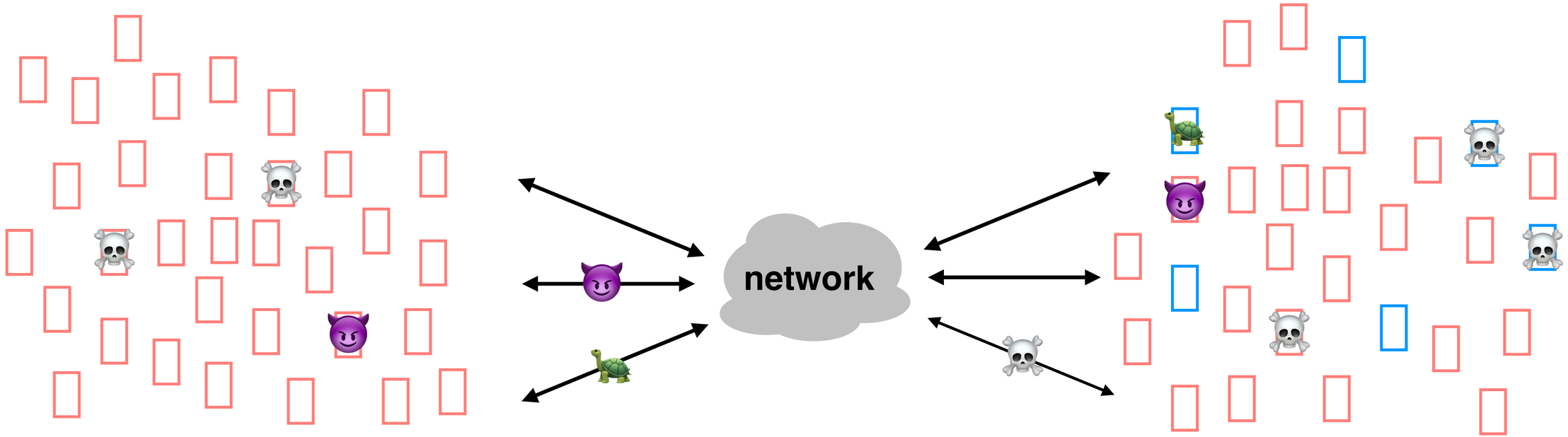
6.033 in the news

"We've identified two separate fiber breaks in our network, impacting services for Spectrum customers in Maine and New Hampshire," Heidi Vandenbrouck said in an email Monday night. "These separate breaks have impacted our redundant path, which normally serves as backup, when a break or damage is incurred in a part of the network."

https://www.pressherald.com/2021/04/05/spectrum-restores-internet-service-after-outage-affects-thousands-across-maine-and-new-hampshire/

scalability: how does our system behave as we increase the number of machines, users, requests, data, etc.?

fault-tolerance/reliability: how does our system deal with failures (☠️)? machines crashing, network links breaking, etc.



security: how does our system cope in the face of targeted attacks (😈)?

performance: how do we define our performance requirements, and know if our system is meeting them? what do we do if performance is subpar (🐢)?

how do our design and implementation choices affect people and communities? who makes those choices?

how to build reliable systems

how to build reliable systems

1. **identify** all possible faults; decide which ones we're going to handle

how to build reliable systems

1. **identify** all possible faults; decide which ones we're going to handle

2. **detect**/**contain** faults

how to build reliable systems

1. **identify** all possible faults; decide which ones we're going to handle

2. **detect**/**contain** faults

3. **handle** faults ("recover")

how to build reliable systems

1. identify all possible faults; decide which ones we're going to handle
2. detect/contain faults
3. handle faults ("recover")

how to measure success

how to build reliable systems

1. **identify all possible faults; decide which ones we're going to handle**
2. **detect/contain faults**
3. **handle faults ("recover")**

how to measure success

there are many things we could measure, but we will typically focus on availability: what fraction of time is the system up and available to use

how to build reliable systems

1. **identify all possible faults; decide which ones we're going to handle**
2. **detect/contain faults**
3. **handle faults ("recover")**

how to measure success

there are many things we could measure, but we will typically focus on availability: what fraction of time is the system up and available to use

today we'll focus on handling disk failure

how to build reliable systems

1. **identify all possible faults; decide which ones we're going to handle**
2. **detect/contain faults**
3. **handle faults ("recover")**

how to measure success

there are many things we could measure, but we will typically focus on availability: what fraction of time is the system up and available to use

today we'll focus on handling disk failure

we want to make the disk as reliable as possible so that we don't lose data; we especially don't want to lose data when the machine fails (which is what will start happening next week)

2.12.1 Annualized Failure Rate (AFR) and Mean Time Between Failures (MTBF)

The product shall achieve an Annualized Failure Rate - AFR - of 0.73% (Mean Time Between Failures - MTBF - of 1.2 Million hrs) when operated in an environment that ensures the HDA case temperatures do not exceed 40°C. Operation at case temperatures outside the specifications in Section 2.9 may increase the product Annualized Failure Rate (decrease MTBF). AFR and MTBF are population statistics that are not relevant to individual units.

AFR and MTBF specifications are based on the following assumptions for business critical storage system environments:

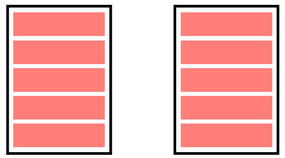
- 8,760 power-on-hours per year.
- 250 average motor start/stop cycles per year.
- Operations at nominal voltages.
- Systems will provide adequate cooling to ensure the case temperatures do not exceed 40°C. Temperatures outside the specifications in Section 2.9 will increase the product AFR and decrease MTBF.

Nonrecoverable read errors	1 per 10^{15} bits read, max
Annualized Failure Rate (AFR)	0.73% (nominal power, 40°C case temperature)

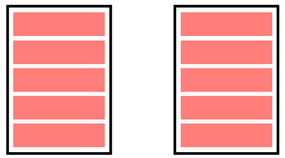
https://www.seagate.com/staticfiles/support/disc/manuals/NL35%20Series%20&%20BC%20ES%20Series/Barracuda%20ES.2%20Series/100468393f.pdf

RAID 1

mirroring

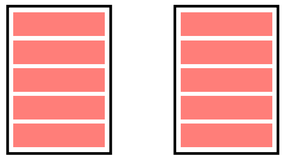


RAID 1
mirroring



+ can recover from single-disk failure

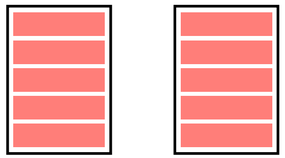
RAID 1
mirroring



+ can recover from single-disk failure

− requires 2N disks

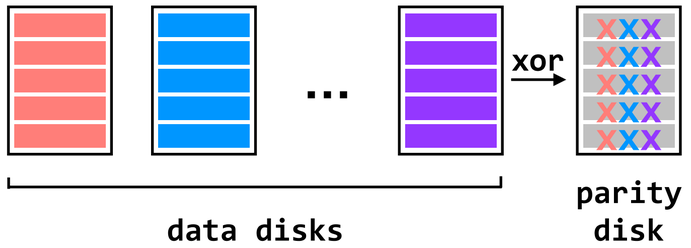
RAID 1
mirroring



+ can recover from single-disk failure

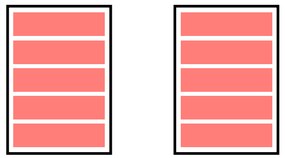
− requires 2N disks

RAID 4
dedicated parity disk



sector i of the parity disk
is the xor of sector i
from all data disks

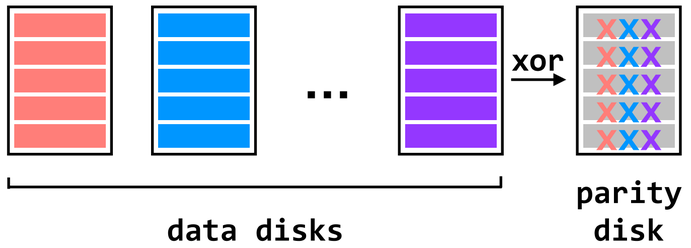
RAID 1
mirroring



+ can recover from single-disk failure

− requires 2N disks

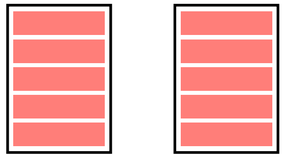
RAID 4
dedicated parity disk



sector i of the parity disk is the xor of sector i from all data disks

+ can recover from single-disk failure

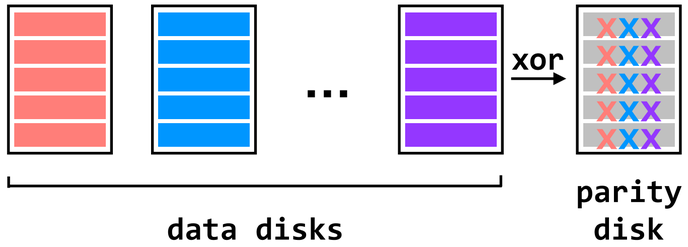
RAID 1
mirroring



+ can recover from single-disk failure

− requires 2N disks

RAID 4
dedicated parity disk

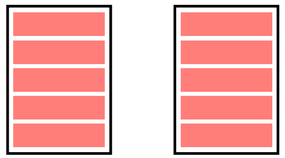


sector i of the parity disk is the xor of sector i from all data disks

+ can recover from single-disk failure

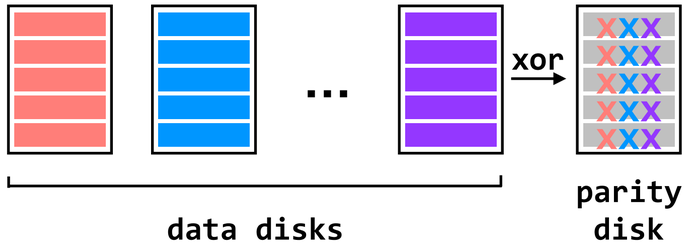
+ requires N+1 disks

RAID 1
mirroring



+ can recover from single-disk failure
− requires 2N disks

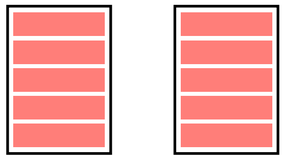
RAID 4
dedicated parity disk



sector i of the parity disk is the xor of sector i from all data disks

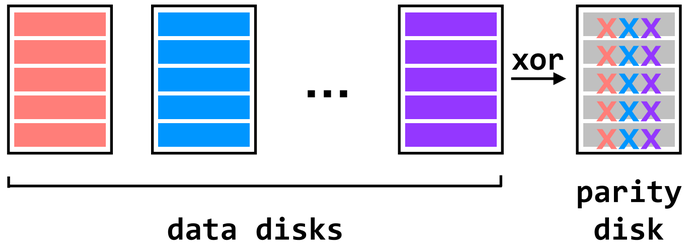
+ can recover from single-disk failure
+ requires N+1 disks
+ performance benefits if you stripe a single file across multiple data disks

RAID 1
mirroring



+ can recover from single-disk failure
− requires 2N disks

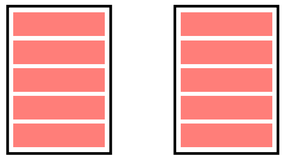
RAID 4
dedicated parity disk



sector i of the parity disk is the xor of sector i from all data disks

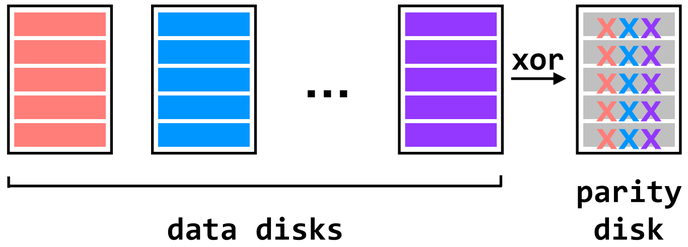
+ can recover from single-disk failure
+ requires N+1 disks
+ performance benefits if you stripe a single file across multiple data disks
− all writes go to the parity disk

RAID 1
mirroring



+ can recover from single-disk failure
− requires 2N disks

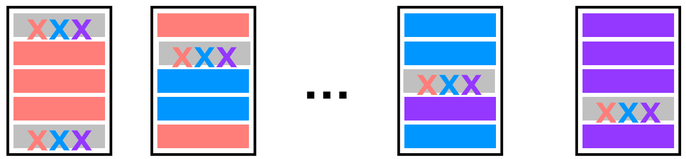
RAID 4
dedicated parity disk



sector i of the parity disk is the xor of sector i from all data disks

+ can recover from single-disk failure
+ requires N+1 disks
+ performance benefits if you stripe a single file across multiple data disks
− all writes go to the parity disk

RAID 5
spread out the parity

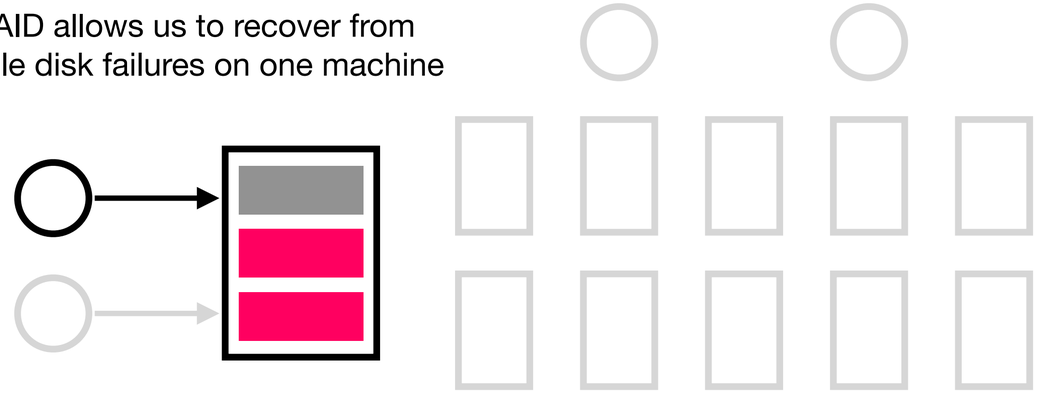


+ can recover from single-disk failure
+ requires N+1 disks
+ performance benefits if you stripe a single file across multiple data disks
+ writes are spread across disks

our goal is to build **reliable systems from unreliable components**. we want to build systems that serve many clients, store a lot of data, perform well, all while keeping availability high

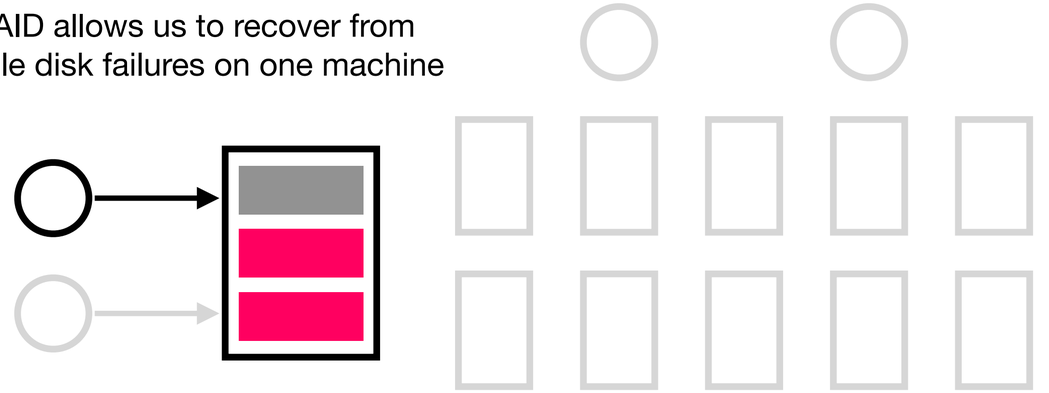
our goal is to build **reliable systems from unreliable components**. we want to build systems that serve many clients, store a lot of data, perform well, all while keeping availability high

RAID allows us to recover from single disk failures on one machine



our goal is to build **reliable systems from unreliable components**. we want to build systems that serve many clients, store a lot of data, perform well, all while keeping availability high

RAID allows us to recover from single disk failures on one machine



the high-level process of dealing with failures is to identify the faults, detect/contain the faults, and handle the faults. in lecture, we will build a set of abstractions to make that process more manageable

this slide — which we'll start and end on in each lecture — will get more involved as that goes along

systems have faults. we have to take them into account and build reliable, **fault-tolerant systems**. reliability comes at a cost — there are tradeoffs between reliability and simplicity, reliability and performance, etc.

our main tool for improving reliability is **redundancy**. one form of redundancy is **replication**, which can be used to combat many things including disk failures (important, because disk failures mean lost data).

RAID replicates data across disks on a single machine in a smart way. RAID 5 protects against single-disk failures while maintaining good performance

one can extend the ideas in RAID to protect against multiple disk failures. RAID 6 does this, for example