



OST

Eastern Switzerland
University of Applied Sciences

Distributed Systems (DSy)

Introduction – Redundancy / Fault Tolerance

Thomas Bocek

20.02.2026

Learning Goals

- Distributed systems add complexity. Avoid complexity!
- Why do we need distributed systems?
 - 1) Scaling (if one machine is not enough)
 - 2) Location (to move closer to the user)
 - 3) Redundancy / Fault-tolerance (HW will fail eventually)

Distributed Systems Motivation

- Why Distributed Systems – Fault-tolerance
 - Any hardware will crash eventually
 - Soft errors vs hard errors
- Source: Bad Pin Connections, Incorrect RAM Timings, Clock Issues, RAM Design Flaws, CPU/RAM/Motherboard Integrated Logic Defects, DRAM Cell Amplification Errors, Cosmic Rays [[link](#)]
- Solar flares, [Coronal mass ejection](#), Solar proton events, Background radiation



- Random bit flips in memory
 - [1990](#): “Computers typically experience about one cosmic-ray-induced error per 256 megabytes of RAM per month”
 - [Google study 2009](#): more than 8% of DIMMs affected by errors per year
 - [2007](#): 44 reported memory errors (41 ECC and 3 double bit) on ~1300 nodes during a period of about 3 months

Distributed Systems Motivation

- Facebook DRAM Study (2015) [[link](#)]
- "Memory Errors in Modern Systems: The Good, The Bad, and The Ugly" (ASPLOS 2015) [[link](#)]
- Google / ETH Zürich – Rowhammer still an issue (2025): [[link](#)]
- [Cosmic rays](#) "may" be blamed for an electronic voting error in Belgium (2003)
 - Bit flip in electronic voting machine
 - Added 4096 extra votes to one candidate
 - Candidate more votes than were possible

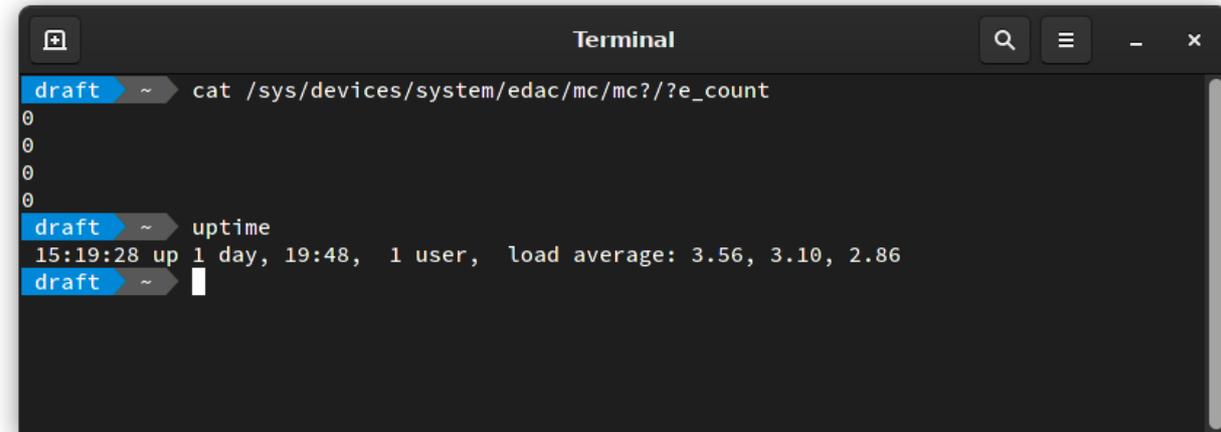
- Error-correcting code memory (ECC)
- [Hamming Code](#), correct 1 bitflip / detect 2 bitflips

Bit position	1	2	3	4	5	6	7
Encoded data bits	p1	p2	d1	p4	d2	d3	d4
Parity bit	p1	✓		✓		✓	✓
	p2		✓	✓			✓
	p4				✓	✓	✓

- Used for Servers, not (yet) used for consumer products – [good idea?](#)
- Consumer: DDR5 has on-die ECC, but weaker than traditional ECC
 - Corrects errors inside the chip (at rest), but not on the data bus (in transit)

Distributed Systems Motivation

- Influencing factors
 - Sensitivity of each transistor, number of transistors on the microchip, altitude, floor level
- Mars Rover?
 - Cassini reported 280 bitflip/day [link] – max 890 due to solar proton event - TMR with ~300MB RAM
- [Radiation-tolerant FPGAs](#) → TMR
 - Triple Modular Redundancy — same operation runs 3 times in parallel, majority vote determines correct result (2 of 3 must agree)
- Double bit-flips unlikely?
 - Jaguar super computer with 360TB ECC RAM
 - Double bitflip → [happened every 24h](#)
- Check your HW



```
Terminal
draft ~ cat /sys/devices/system/edac/mc/mc??e_count
0
0
0
0
0
draft ~ uptime
15:19:28 up 1 day, 19:48, 1 user, load average: 3.56, 3.10, 2.86
draft ~
```

- [What can happen](#): e.g., expr segfaults

Distributed Systems Motivation

- HDD break [\[link\]](#), SSDs wear out
 - SSDs consist of NAND cells with a limited lifetime
 - An SSD disk has spare NAND that are used when cells break
 - `smartctl -a /dev/xyz`
- SLC, MLC, TLC, QLC [\[link\]](#), [\[link\]](#)
 - SLC: 50'000 - 100'000 write/erase cycles
 - MLC: 3'000 - 10'000 – write/erase cycles
 - TLC: 1'000 - 3'000 – write/erase cycles
 - QLC: 100 - 1'000 – write/erase cycles
- Data loss on unpowered SSDs [\[link\]](#)
 - 100% → no spare used, My old laptop was at 92%
 - When value is down at 0% disk capacity degrades
 - E.g., Samsung 4TB drive uses QLC [\[link\]](#)
 - Write 100 times the same 4kb file, and cells are broken?
 - Wear leveling: distribute write and erase operations across all memory cells
 - If wear leveling goes wrong: Samsung 990 Pro [\[link\]](#)
 - Caching with SLC → files / cells that are frequently changed, store on SLC, once they don't change that often move to MLC/TLC/QLC

```
≡ START OF SMART DATA SECTION ≡
SMART overall-health self-assessment test result: PASSED

SMART/Health Information (NVMe Log 0x02)
Critical Warning:          0x00
Temperature:              49 Celsius
Available Spare:           100%
Available Spare Threshold: 10%
Percentage Used:          17%
Data Units Read:          511,224,474 [514.7 TB]
Data Units Written:       764,591,703 [391 TB]
Host Read Commands:      7,756,128,132
Host Write Commands:     4,042,594,169
Controller Busy Time:    17,710
Power Cycles:             462
Power On Hours:          23,008
```

Distributed Systems Motivation

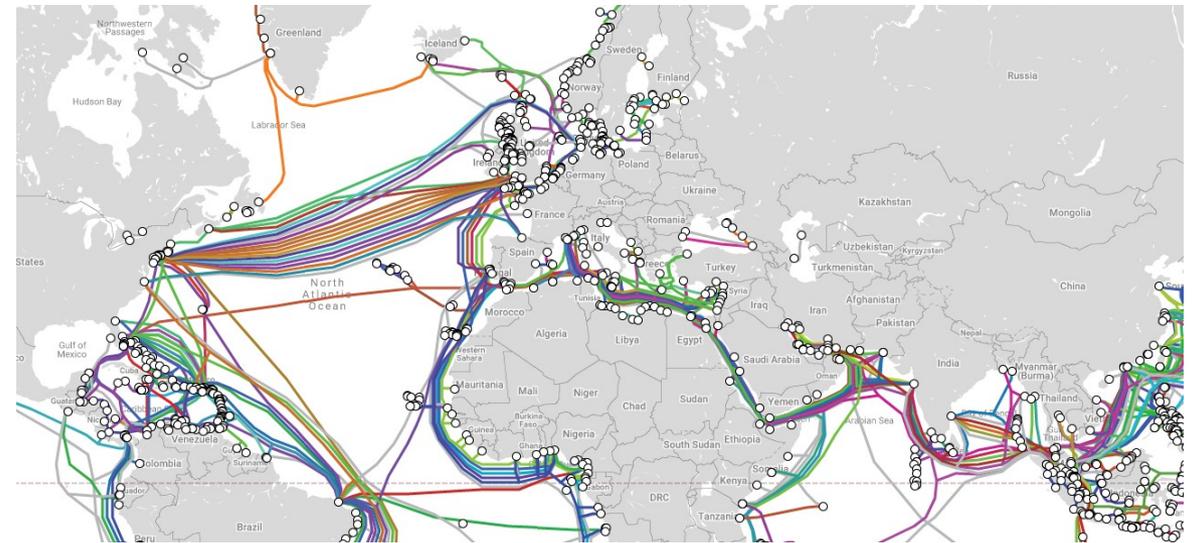
- Random bit flips in memory
 - Artem Dinaburg (2011): registered domains with single bit error
 - "59 unique IPs per day made HTTP requests to my 32 bitsquat domains"
 - Most users from China (more bitflips on Chinese machines?)
 - Project Bitfl1p: larger replication with hundreds of domains (2015)
 - Register names with single bit error, e.g,
- Idea: if bitflip happens, it may happen for DNS names in your memory
 - Over 1 million DNS queries per 24 hours

Bitsquat Domain	Original Domain
ikamai.net	akamai.net
aeazon.com	amazon.com
a-azon.com	amazon.com
amazgn.com	amazon.com
microsmft.com	microsoft.com
micrgsoft.com	microsoft.com

Fault Tolerance

- Seacables provide 99% of data connectivity [\[link\]](#)
- Network outages happens [\[link\]](#)
 - 22.01.2026: Indigo West Subsea Cable Repairs Underway [\[link\]](#)
 - 11.11.2025: Repairs to Broken Subsea Broadband Cable for Tiree Island Set to Begin [\[link\]](#)
 - 24.10.2025: Pakistan ISP Says Its Internet Services Will Be Fully Restored Across Country by Monday [\[link\]](#)
 - 15.10.2025: Internet Services Restored After Repair of Submarine Cable: PTCL [\[link\]](#)

- [Submarine Cable Map](#)



Distributed Systems Motivation

- Hardware will fail eventually
 - Bit flips in memory can occur
 - SSDs will wear out over time
 - Network cables can be damaged
 - Not a question of "if" but "when"
- Distributed Systems provide a solution
 - Multiple machines provide redundancy
 - When one machine fails, others can take over
- Load can be redistributed among remaining machines
- System continues to function despite individual failures
- Trade-off consideration
 - Distributed systems add complexity
 - Use only when benefits outweigh the added complexity
 - Redundancy and fault tolerance must justify the complexity