OST
Eastern Switzerland
University of Applied Sciences

Distributed Systems & Blockchain (DS1)

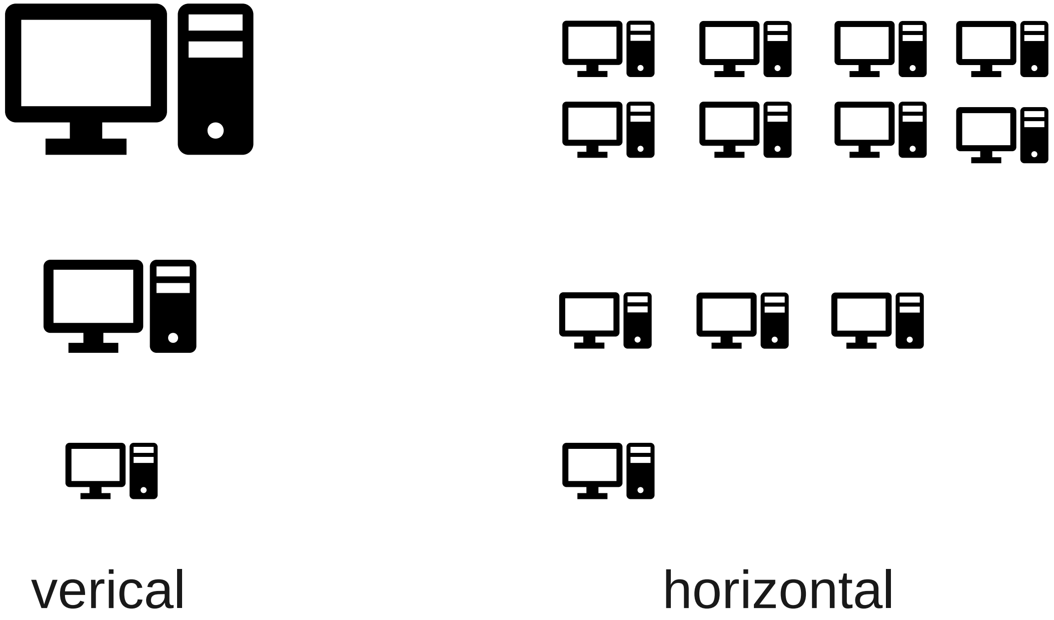
Introduction

Thomas Bocek

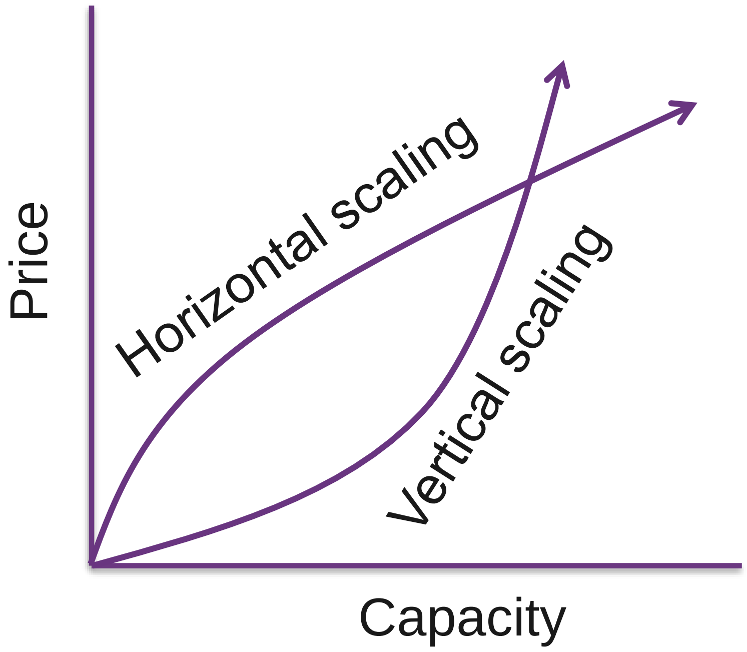
28 February 2021

Distributed Systems Motivation

- Why Distributed Systems
  - Scaling
    - Vertical (scale up), more memory, faster CPU
    - Horizontal (scale out), more machines
    - Apple has 75’000 Apache Cassandra nodes storing 10 petabytes of data in 2015 [source]



- Economics
  - Initially scaling vertically is cheaper, until you max out HW
  - Current x86 max: 64 cores (AMD)



Distributed Systems Motivation

Horizontal Scaling

+ Lower cost with massive scale

+ Easier to add fault-tolerance

+ Higher availability

- Adaption of software required

- More complex system, more components involved

Vertical Scaling

+ Lower cost with small scale

+ No adaption of software required

+ Less administrative effort

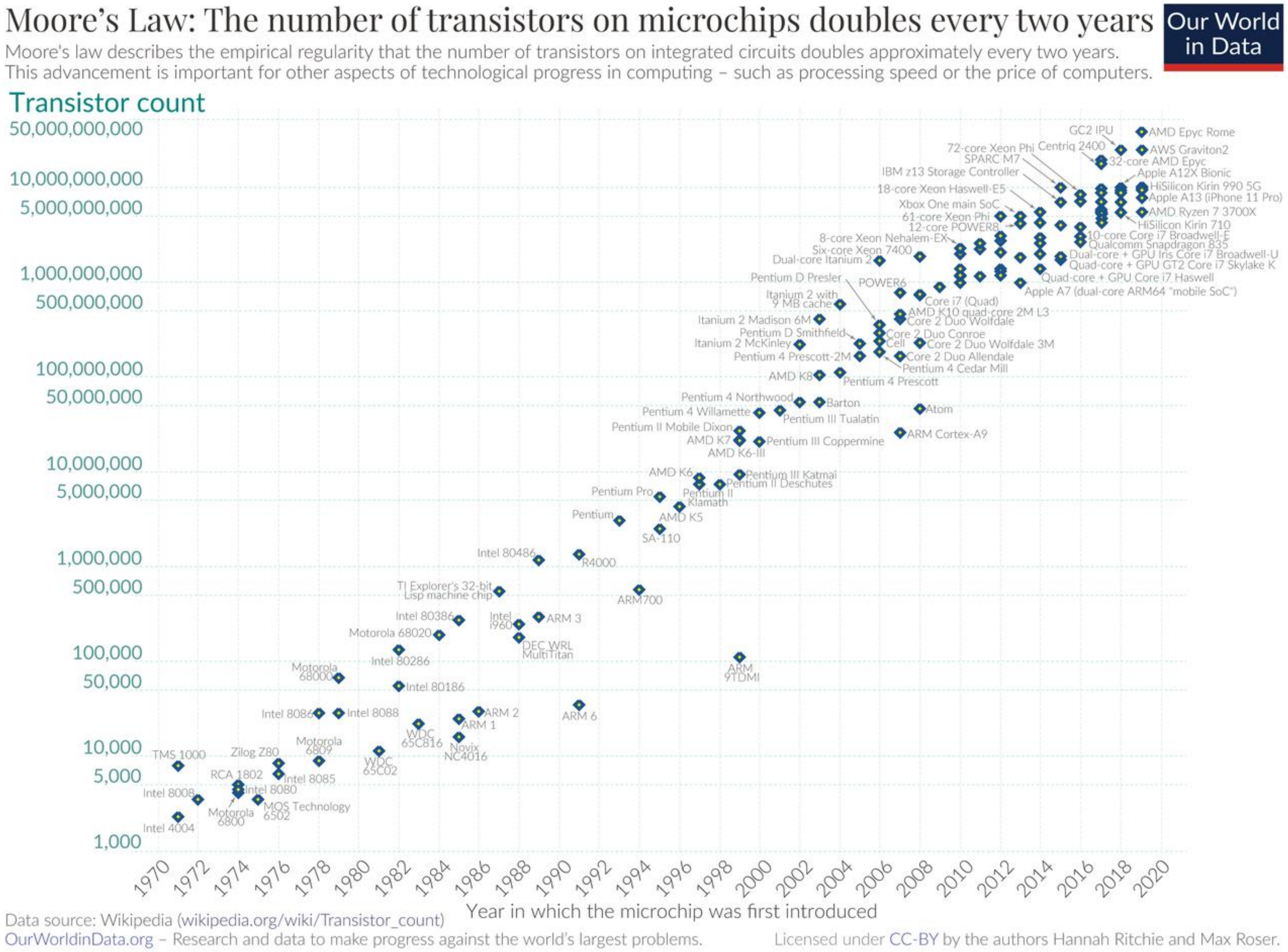
- HW limits for scaling

- Risk of HW failure causing outage

- More difficult to add fault-tolerance

Vertical Scaling Performance

- Moore’s Law – nr. of transistors doubles every 2 years (other predictions, doubling chip performance every 18 month)
- Dead in 2025? Or 2045?
- Forbes 1995: "The price per transistor will bottom out sometime between 2003 and 2005. From that point on, there will be no economic point to making transistors smaller. So Moore's Law ends in seven years."
- AMD Ryzen, 64 cores ~40b transistors
- Graphcore C2 IPU for AI ~24b transistors
- Apple M1 ~16b



Moore’s Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

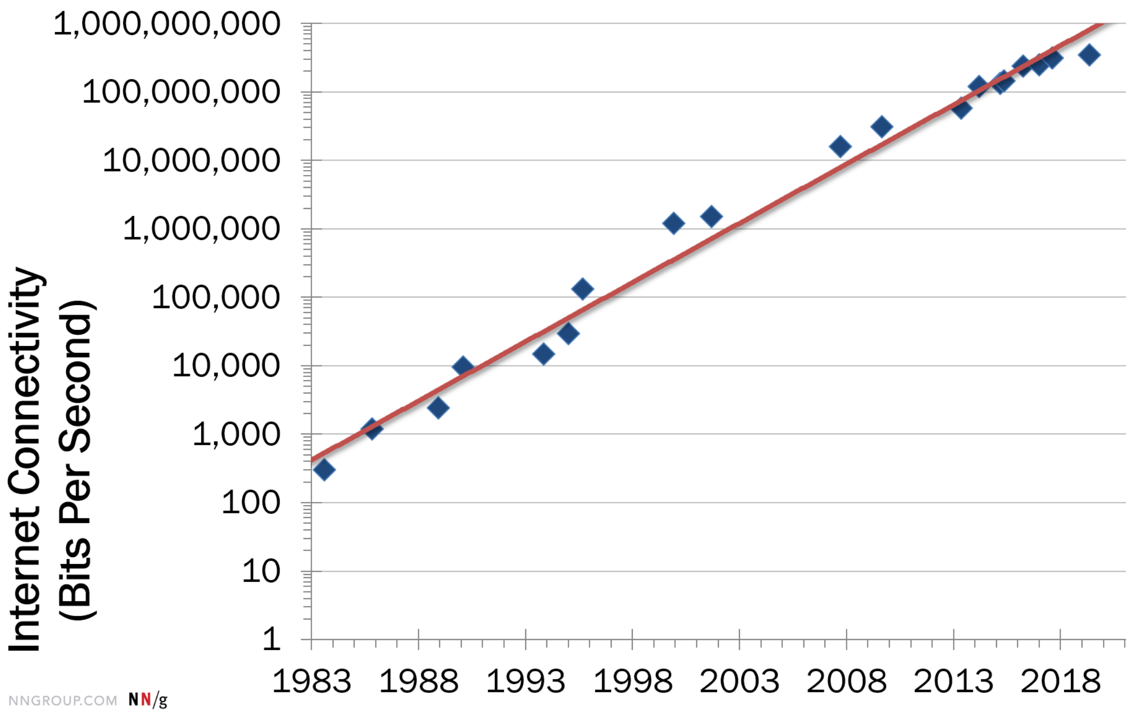
Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world’s largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

Vertical Scaling Performance

- Nielsen’s Law: a high-end user’s connection speed grows by 50% per year
- Bandwidth grows slower than computer power
  1. Telecoms companies are conservative
  2. Users are reluctant to spend much money on bandwidth
  3. The user base is getting broader
- Optimize for bandwidth
- Zmap complete scan of the IPv4 address space in under 5 minutes

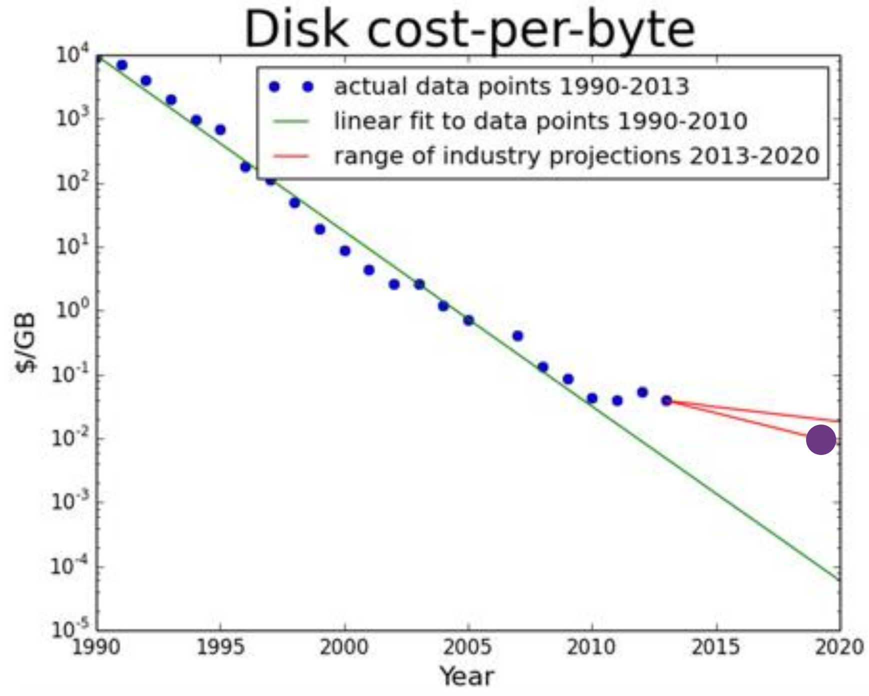


https://www.nngroup.com/articles/law-of-bandwidth/

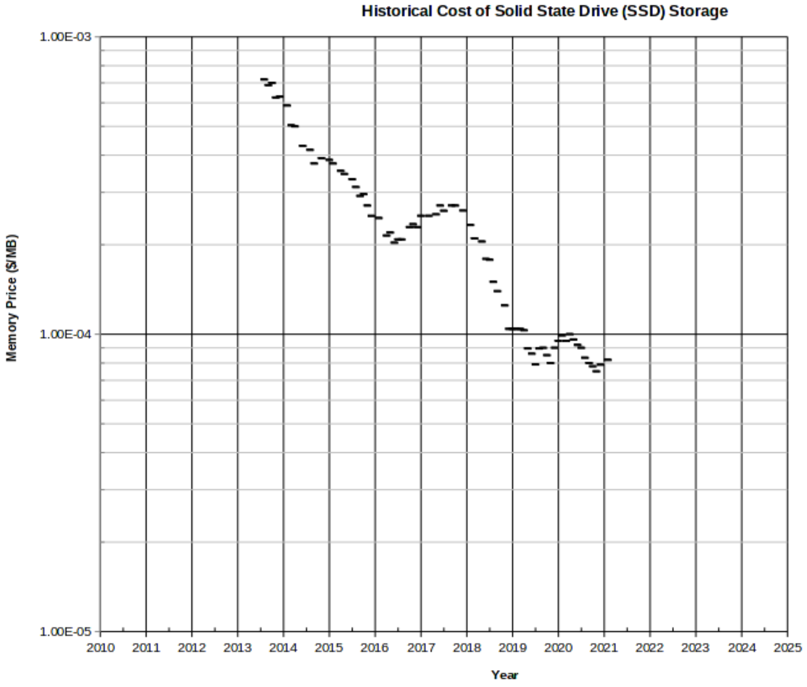
		Annualized Growth Rate	Compound Growth Over 10 Years
Nielsen's law	Internet bandwidth	50%	57×
Moore's law	Computer power	60%	100×

Vertical Scaling Performance

- Kryder's Law: disk density doubling every 13 month
- «Soon hard drives will migrate into phones, still cameras, PDAs, cars and everyday appliances»
  https://www.scientificamerican.com/article/kryders-law/ , Aug. 2005
- User behavior changed
  - SSD, speed is importan
- Cloud – Dropbox, Spotif
  - Streaming



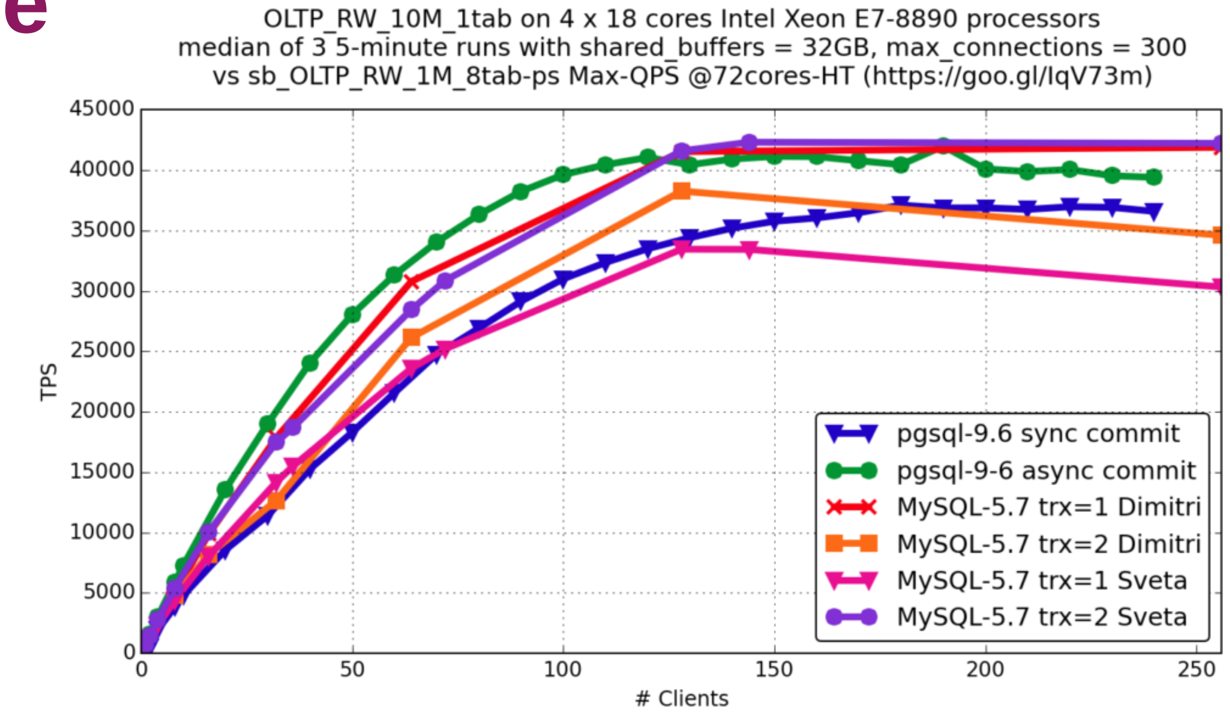
http://blog.dshr.org/2016/05/the-future-of-storage.html



Source: https://jcmit.net/flashprice.htm

Vertical Scaling Performance

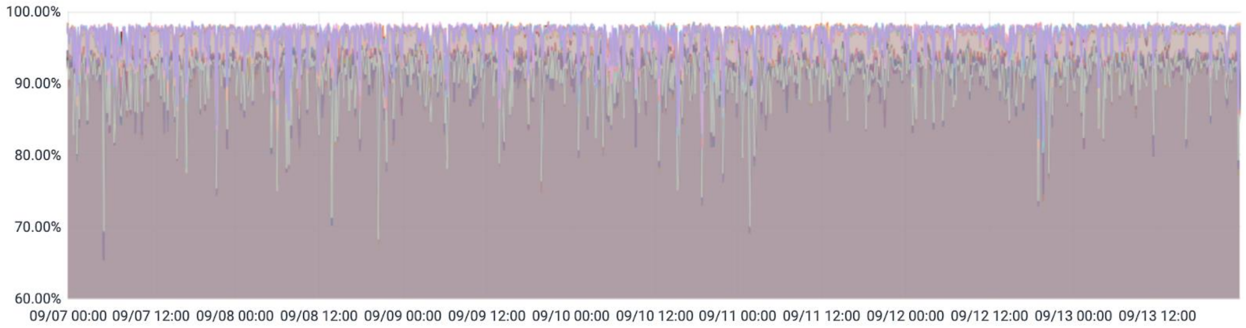
- Vertical scaling
  - HW today is fast!
    - Database benchmark with a fast machine in 2017:
    - 1mio QPS read
    - 0.04mio QPS read/write
  - Best principle for small and simple applications!
  - Simple website with a few DB calls is not HW intensive
    - But: ML, Gaming (cloud gaming) are HW intensive

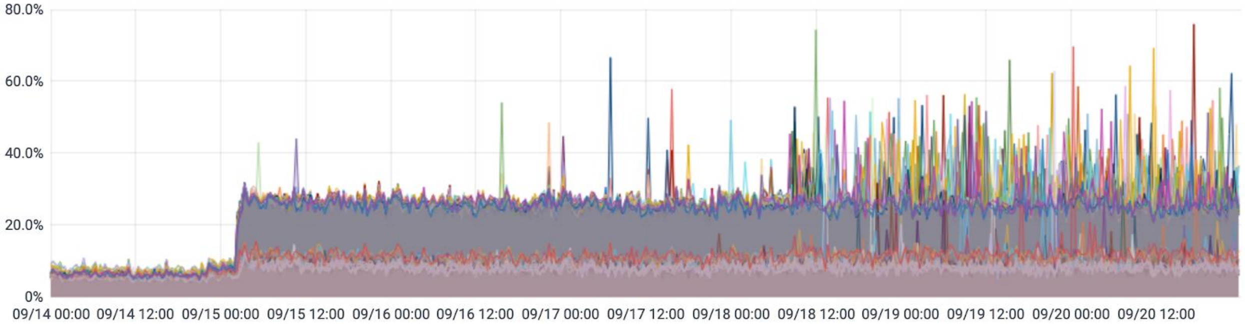


https://www.percona.com/blog/2017/01/06/millions-queries-per-second-postgresql-and-mysql-peaceful-battle-at-modern-demanding-workloads/

Vertical Scaling Performance

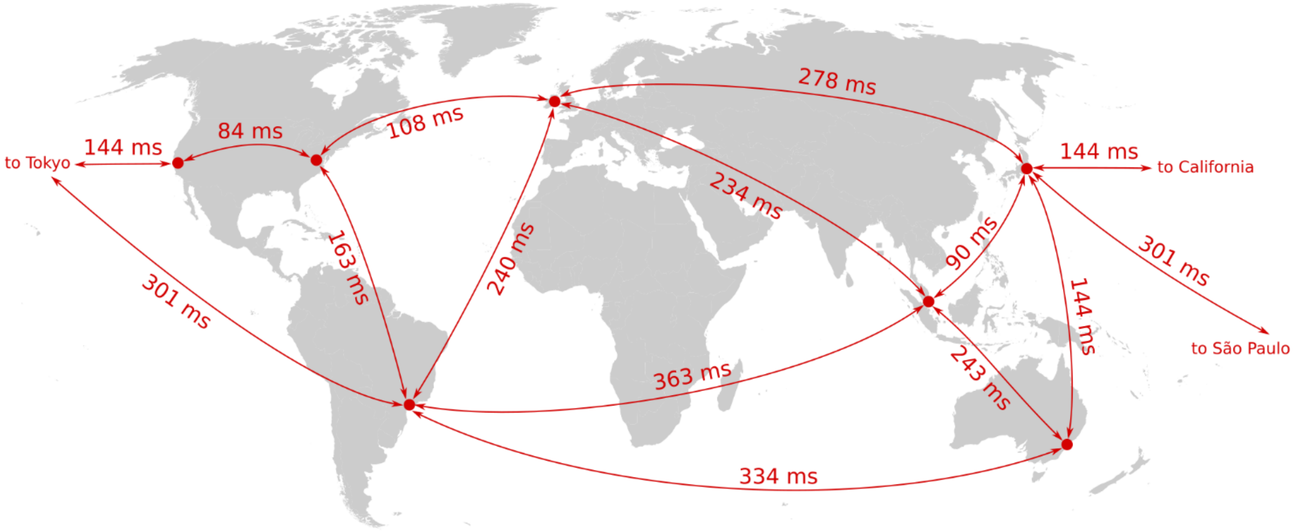
- Let's Encrypt
  - 21.01.2021: The Next Gen Database Servers Powering Let's Encrypt
    - Providing certificates for 235m websites
    - “A database is at the heart of how Let’s Encrypt manages certificate issuance” - 1 single MariaDB
    - “We run the CA against a single database in order to minimize complexity” – Some read operations at replicas, one server for writes
    - 2x Xeon 24-cores running at 90%
    - Upgrade to 2x64 Epyc, on 15.09, running at 25%
      - Query 3 times faster
      - SATA → NVMe - IO from 500MB/s to 3 GB/s





Distributed Systems Motivation

- Why Distributed Systems
  - Location
    - Everything gets faster, latency stays
    - Physically bounded by the speed of light



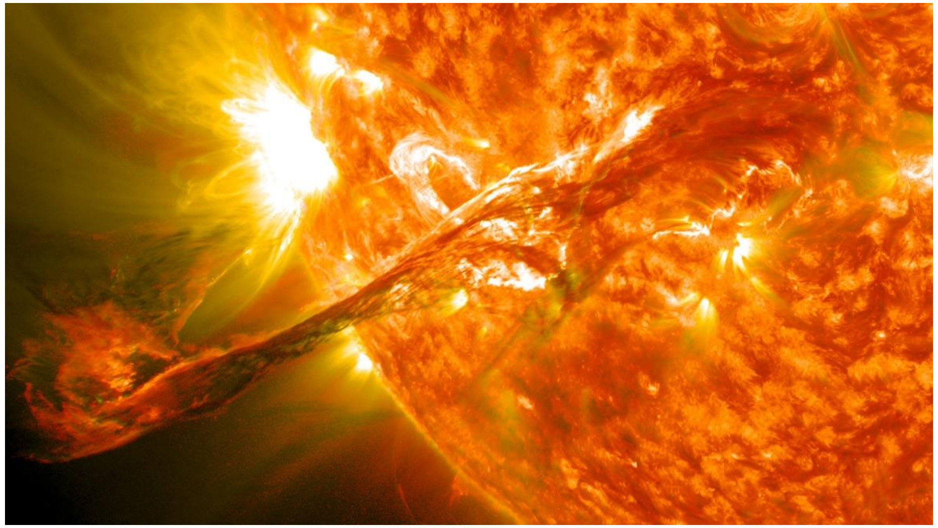
https://www.inkandswitch.com/local-first.html

- New protocols can decrease #RT
  - Upcoming lecture
- Place services closer to user
  - Sometimes latency of 310ms is unacceptable
    - ping sydney.edu.au
  - Gaming / Esports:
    - Human reaction time 200ms
    - Total from keypress to display:
      - Thinkpad 13 ChromeOS: 70ms
      - Lenovo X1 carbon 2016: 150ms
    - TV output lag ~15-30ms (random TV)
    - Keyboard 15-60ms
- CDN: Content delivery network
  - Place your images, sites, scripts close to your users

Distributed Systems Motivation

- Why Distributed Systems
  - Fault-tolerance
    - Any hardware will crash eventually
- 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 affectedby errors per year
  - 2007: 44 reported memory errors (41 ECC and 3 double bit) on ~1300 nodes during a period of about 3 month
- Source
  - Cosmic rays
    - Solar flares, Coronal mass ejection, Solar proton events, Background radiation

- 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

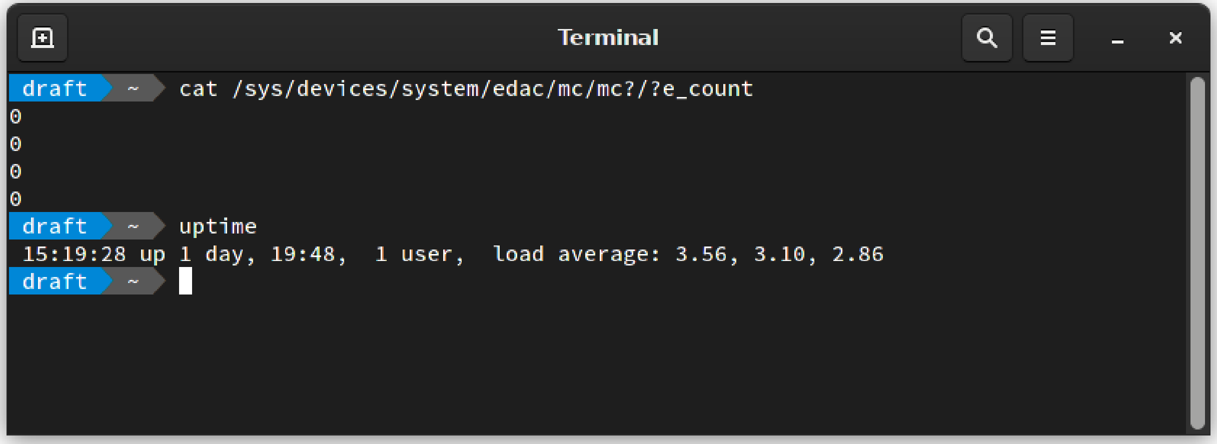


https://en.wikipedia.org/wiki/Solar_flare

Distributed Systems Motivation

- Influencing factors
  - Sensitivity of each transistor, number of transistors on the microchip, altitude
  - Smaller transistors leading to an increased sensitivity per transistor, but smaller cells make smaller targets
- Mars Rover? (Cassini reported 280 bitflip/day – max 890 with ~300MB)
  - Radiation-tolerant FPGAs → TMR
- Error-correcting code memory
  - Uses TMR or Hamming Code, correct 1 bitflip / detect 2 bitflips
  - Used for Servers
  - Not used for consumer products

- Double bit-flips unlikely?
  - Jaguar super computer with 360TB ECC RAM
  - Double bitflip → happened every 24h
- Check your HW



```
draft  ~  cat /sys/devices/system/edac/mc/mc?/?e_count
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

- Random bit flips in memory
  - Bitsquatting: DNS Hijacking without exploitation (2011)
  - Register names with single bit error, e.g,

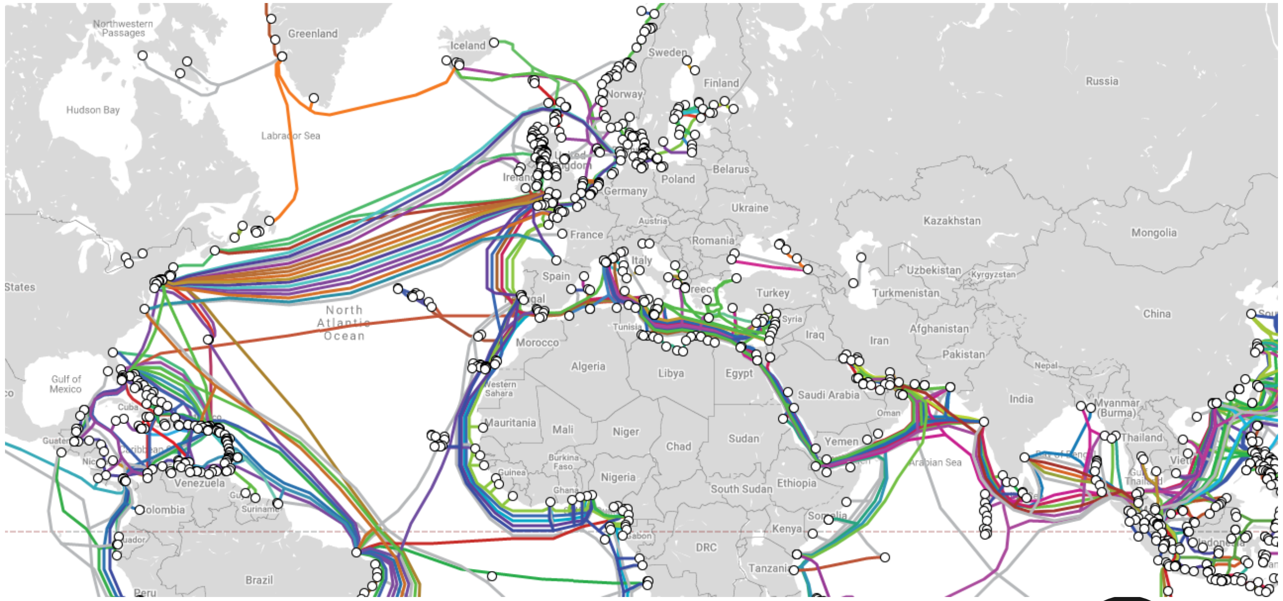
Bitsquat Domain	Original Domain
ikamai.net	akamai.net
aeazon.com	amazon.com
a-azon.com	amazon.com
amazgn.com	amazon.com
microsmft.com	microsoft.com
micrgsoft.com	microsoft.com

- Idea: if bitflip happens, it may happen for DNS names in your memory
  - “59 unique IPs per day made HTTP requests to my 32 bitsquat domains”

- Key findings
  - Finding 1: Bit-errors can be exploited via DNS
  - Finding 2: Not all bit-errors are created equal
    - bit-error in PC vs. bit-error in proxy
  - Finding 3: Mobile and embedded devices may be more affected than traditional hardware

Fault Tolerance

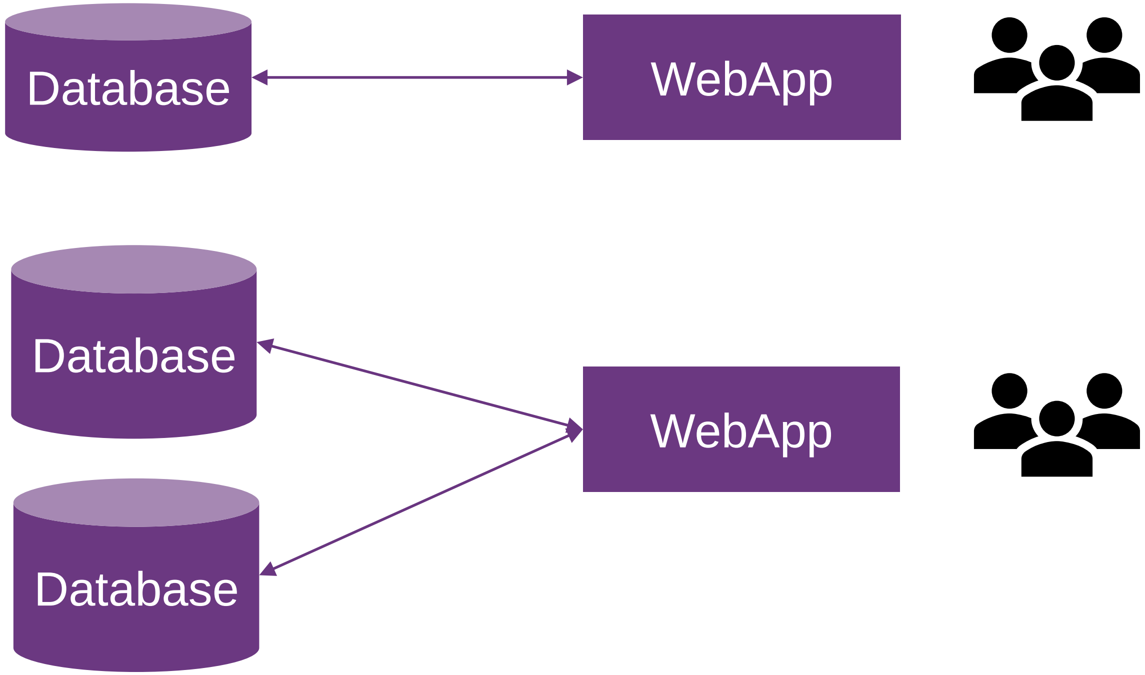
- 18.02.2021: Pakistan Experiencing Second Cable Fault In A Week
  - One seacables broke near Egypt, Internet at lower speed in Pakistan
- 25.01.2021: Sea-Me-We-5 to Undergo Repairs This Week
  - Internet may be slow in Bangladesh on 31.01.2021. Bangladesh connected with 2 undersea cables, one needs to be fixed
- 11.01.2021: Two international undersea optical cables, IA and APG, had problems
  - Vietnam affected
- 23.04.2020: Streit zwischen Init7 und UPC: Internet-Verbindung wieder besser
  - Traffic zwischen UPC und Init7 ueber USA geleitet → 140ms delay
- Submarine Cable Map



Distributed Systems Categorization

- It is useful to classify distributed systems as either tightly coupled, meaning that the processing elements, or nodes, have access to a common memory, and loosely coupled, meaning that they do not [reference]
  - In this lecture, distributed systems ≈ loosely coupled
- A homogeneous system is one in which all processors are of the same type; a heterogeneous system contains processors of different types
  - In this lecture, distributed systems ≈ heterogeneous system
- Small-scale system: WebApp + database vs. large-scale with more than 2 machines
  - In this lecture, often distributed systems ≈ large-scale system

- Decentralized vs. distributed
  - Decentralized ~ distributed in the technical sense, but not owned by one actor



Distributed Systems Definition

Definition: A distributed system in its simplest definition is a group of computers working together as to appear as a single computer to the user

Distributed Systems Categorization

- Another classification
- CAP theorem - states that a distributed data store cannot simultaneously be consistent, available and partition tolerant
  - Consistency—Every node has the same consistent state
  - Availability— Every non-failing node always returns a response
  - Partition Tolerant—The system continues to be consistent even when network partitions
- With network partition - choose between consistency and availability
  - Is a system AP or CP?
- Blockchain and CAP
  - Both, if you wait for n blocks, CP, if you don't AP
- Cassandra AP
  - But can be configured CP

Distributed Systems Categorization

“Controlled” Distributed Systems

- 1 responsible organization
- Low churn
- Examples:
  - Amazon DynamoDB
  - Client/server
- “Secure environment”
- High availability
- Can be homogeneous / heterogeneous

“Fully” Decentralized Systems

- N responsible organizations
- High churn
- Examples:
  - BitTorrent
  - Blockchain
- “Hostile environment”
- Unpredictable availability
- Is heterogeneous

Distributed Systems Categorization

“Controlled” Distributed Systems

- Mechanisms that work well:
  - Consistent hashing (DynamoDB, Cassandra)
  - Master nodes, central coordinator
- Network is under control or client/server → no NAT issues

“Fully” Decentralized Systems

- Mechanisms that work well:
  - Consistent hashing (DHTs)
  - Flooding/broadcasting - Bitcoin
- NAT and direct connectivity huge problem

Distributed Systems Categorization

“Controlled” Distributed Systems

- Consistency
  - Leader election (Zookeeper, Paxos, Raft)
- Replication principles
  - More replicas: higher availability, higher reliability, higher performance, better scalability, but: requires maintaining consistency in replicas
- Transparency principles apply

“Fully” Decentralized Systems

- Consistency
  - Weak consistency: DHTs
  - Nakamoto consensus (aka proof of work)
  - Proof of stake – Leader election, PBFT protocols
    Is Bitcoin eventually consistent?
    - Some argue no, some argue it has even stronger guarantees
- Replication principles apply to fully decentralized systems as well
- Transparency principles apply

Transparency in distributed systems

- Distributed system should hide its distributed nature
  - Location transparency – users should not be aware of the physical location
  - Access transparency - users should access resources in a single, uniform way
  - Migration, relocation transparency – users should not be aware, that resource have moved
  - Replication transparency – users should not be aware about replicas, it should appear as a single resource
  - Concurrent transparency – users should not be aware of other users
  - Failure transparency – users should be aware of recovery mechanisms
  - Security transparency – users should be minimally aware of security mechanisms
- More/other transparencies here, here, here
  - Depends on the context

Fallacies of Distributed Computing

- 8 fallacies to consider

1. The network is reliable
   – Submarine cables
2. Latency is zero
   – Ping to Australia is ~300ms
3. Bandwidth is infinite
   – What is faster? Send a bike courier with an 8TB disk, that arrives 10h later, or send the data with a 1Gibt/s link? 8 * 1000 * 8 / (10 * 60 * 60) = 1.7Gbit/s
4. The network is secure
   – Assume someone is listening. Don’t send sensitive data over the network
5. Topology doesn't change
   – Ping to Australia, request can take different route than reply
6. There is one administrator
   – Sometimes your route goes from one company to another rival company (UPC, Init7)
7. Transport cost is zero
   – Someone build and maintains the network
8. The network is homogeneous
   – From fiber to wifi to cable, server, desktop, mobile