FS20

VERTEILTE SW-SYSTEME (VSS)

Introduction

T. Bocek

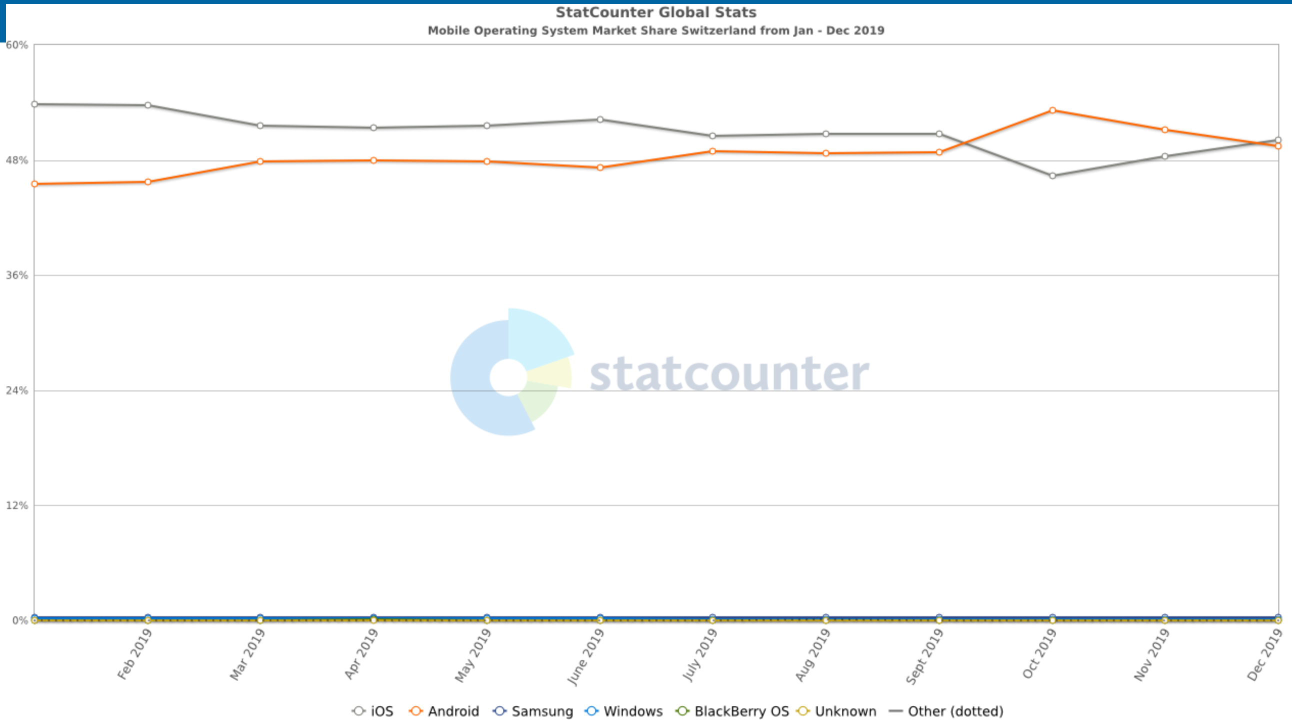
thomas.bocek@hsr.ch

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Distributed Systems - In the *News*

- **03.02.2020 [OpenPush - A Free, Decentralized Push Messaging Framework for Android]**
  - Current situation: Android (~87% market share, CH:50/50) - Firebase Cloud Messaging (FCM)
  - Push Messaging Frameworks can coordinate/maintain connections
    - Firebase Cloud Messaging (FCM) – not open
    - Your app needs to integrate with FCM, and bundle with unfree software – F-droid!
    - Privacy concerns
  - Radio is battery intensive (Wifi - 3/4/5G)
    - Only send/receive when necessary!
  - Optimize battery lifetime vs. instant notifications
  - App has own Push Framework – difficult to maintain an open connection while app runs in background (workaround visible in status menu)



- WIP
  - https://gitlab.com/Bubu/pushclient
  - https://gitlab.com/Bubu/pushserver

Distributed Systems - In the *News*

- **16.02.2020 [IOTA cryptocurrency shuts down entire network after wallet hack]() – [status page]()**
  - 12.2.2020, 17:55 warning about Trinity wallet
    - 18:20 shutdown of Coordinator
  - 13.2.2020, investigation ongoing, 10 users affected, root cause unknown
  - 14.2.2020, checking dependencies (electron), no protocol breach
    - 16:50, root cause found: Trinity Wallet (no details given)
  - 15.2.2020, working on reverting stolen funds
  - 16.2.2020, fix ready
  - 17.2.2020, new version of Trinity Wallet released → migrate tokens if you used Trinity after 17.12.2019

- "To address the minority of cases in which unauthorized token transfers were made out of users' wallets, a third step is needed. We will perform a global snapshot of the network that will, **pending community validation**, enable us to bring stolen tokens back to the affected users."

→ Blockchain immutability?

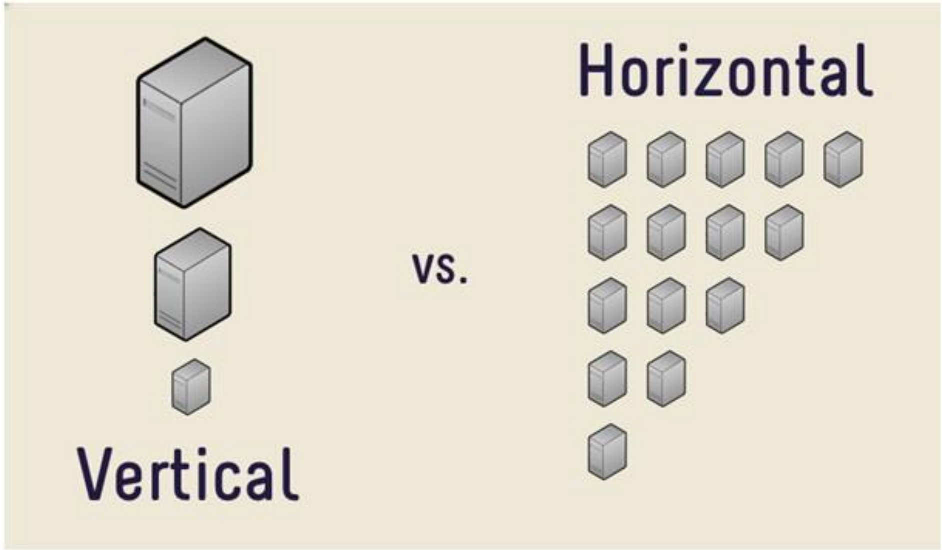
- KYC procedure to get back stolen tokens
- After this process, Coordinator is restarted

URL: b.socrative.com/student/

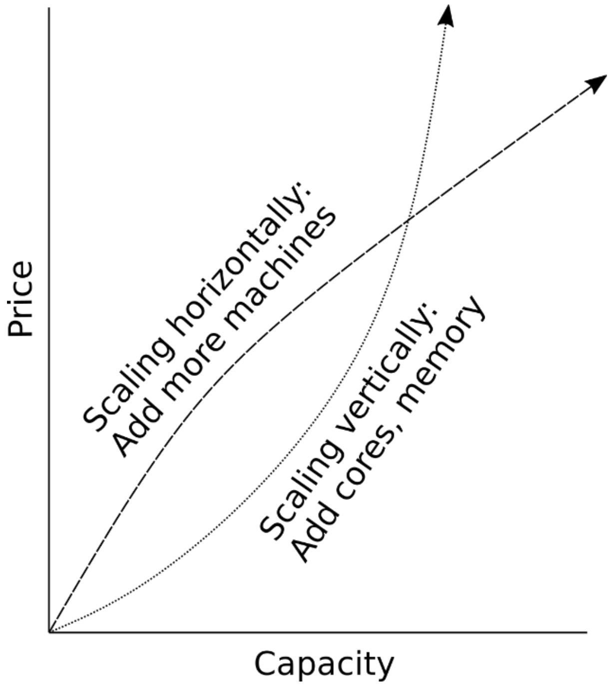
Room: HSR

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



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



Distributed Systems Motivation

Horizontal Scaling

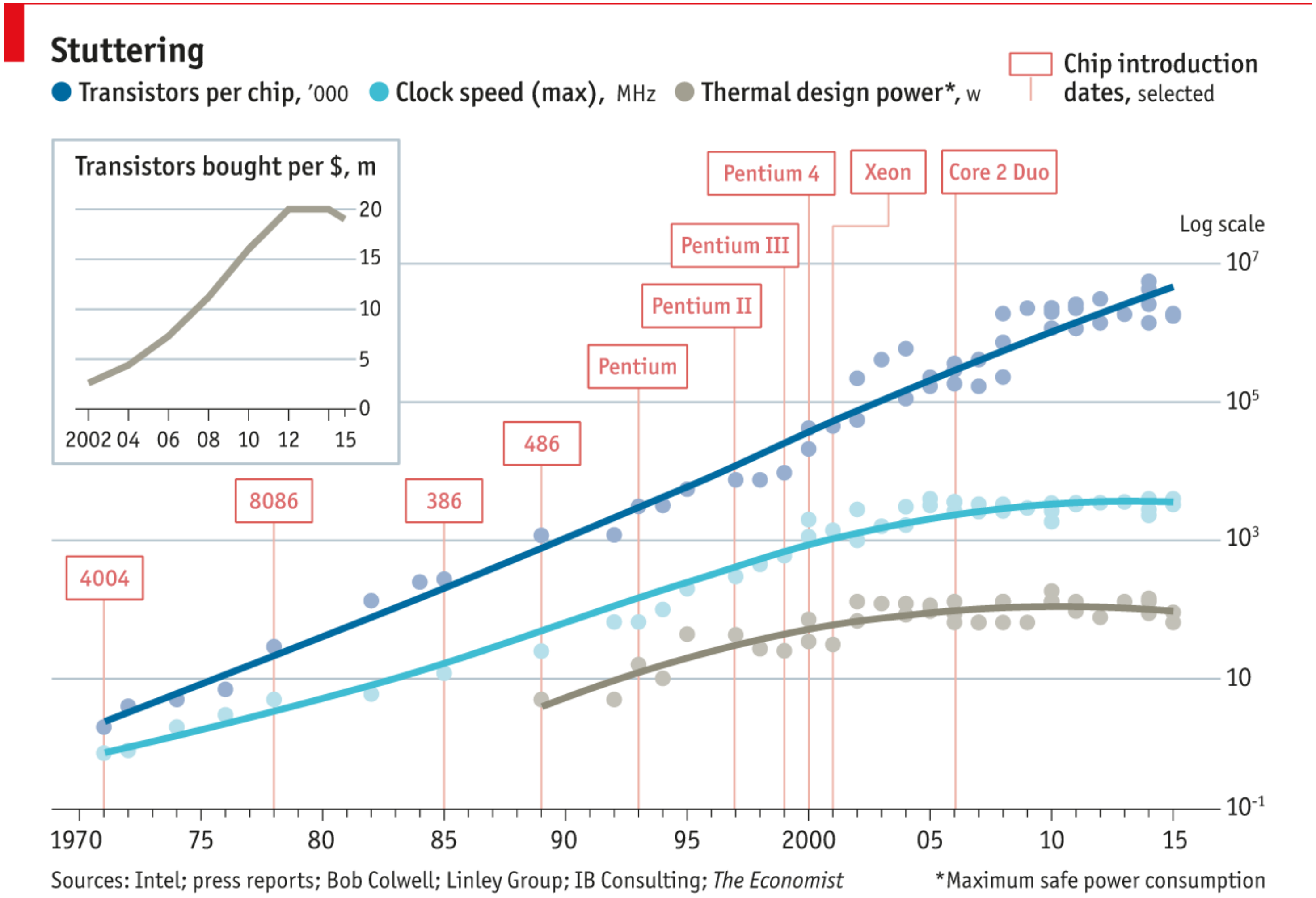
- + Lower cost with massive scale
- + Easier to add fault-tolerance
- + Higher availability
- - Adaption of software required
- - More complex system, more components involved
- - More power consumption

Vertical Scaling

- + Lower cost with small scale
- + No adaption of software required
- + Less administrative effort
- + Less power consumption
- - Risk of HW failure causing outage
- - More difficult to add fault-tolerance
- - HW limits for scaling

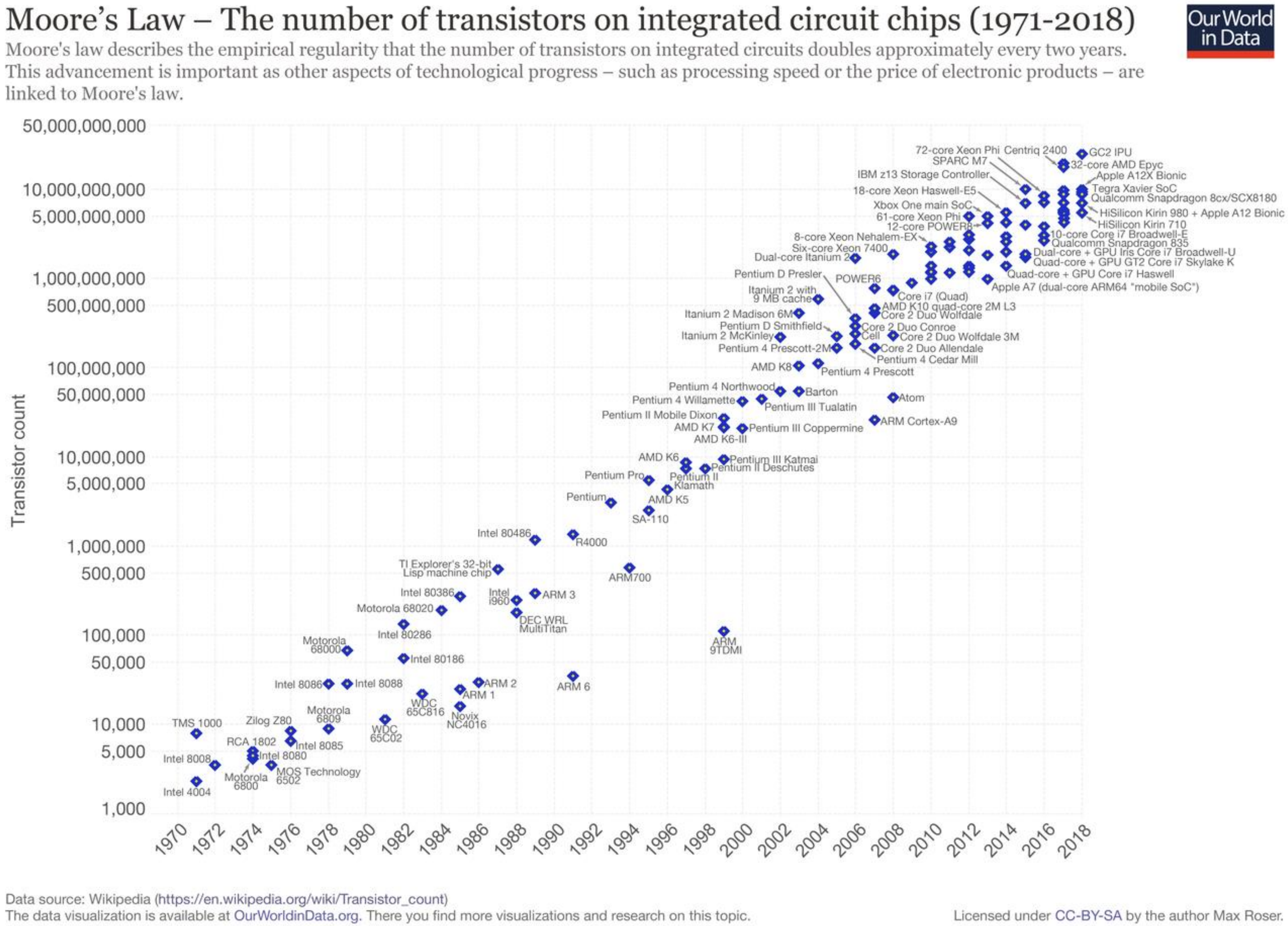
Vertical Scaling Performance

- **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."**



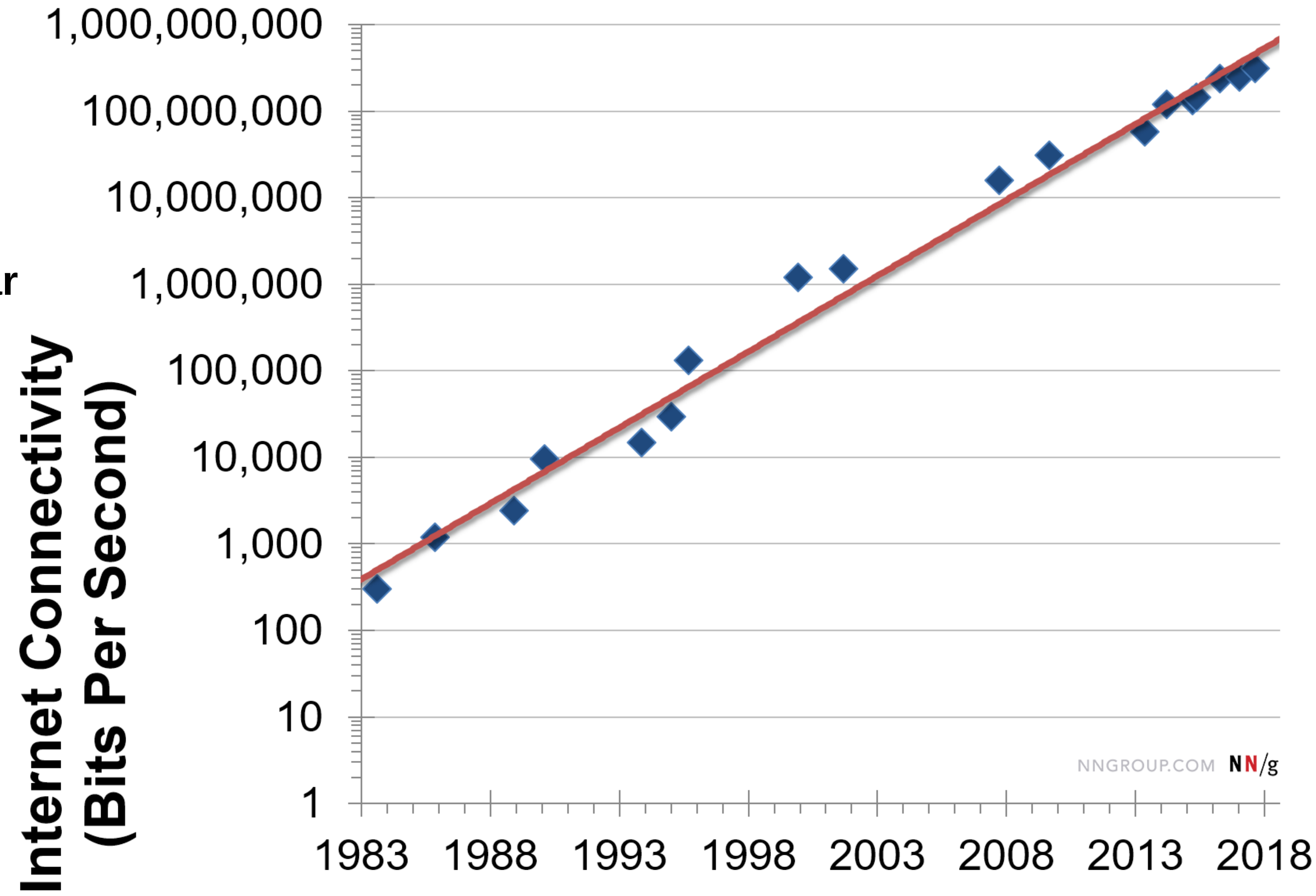
Vertical Scaling Performance

- AMD Ryzen, 64 cores ~40b transistors
- Graphcore C2 IPU for AI ~24b transistors
- Nvidia GV100 ~21b transistors



Vertical Scaling Performance

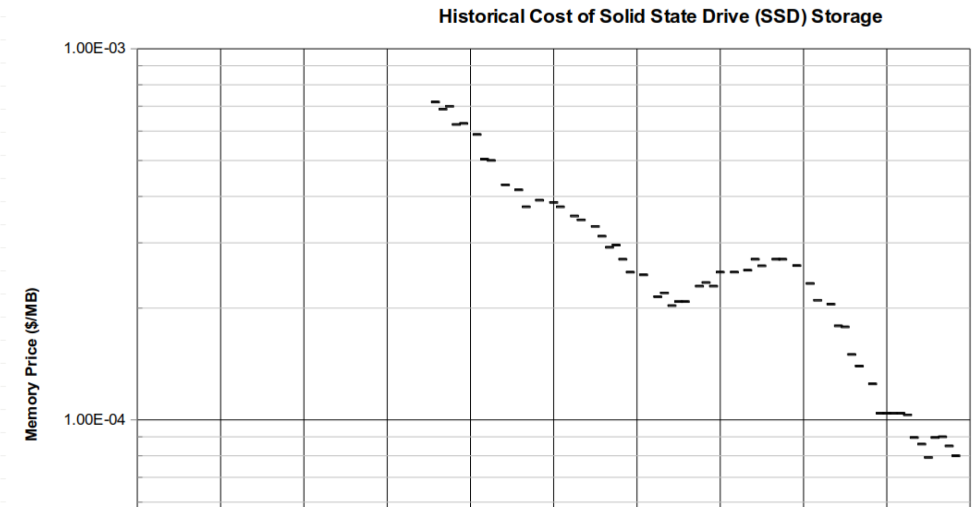
- **Nielsen's Law: a high-end user's connection speed grows by 50% per year**
- **[Bandwidth grows slower than computer power]**
  1. Telecoms companies are conservative
  2. Users are reluctant to spend much money on bandwidth
  3. The user base is getting broader
- **Optimize for bandwidth**
- **[Zmap]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

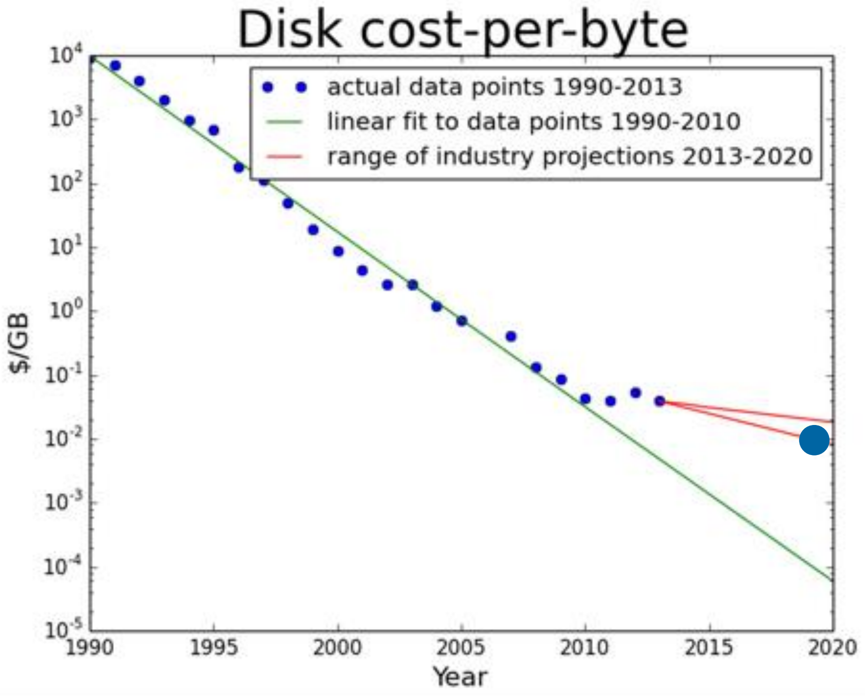


		Annualized Growth Rate	Compound Growth Over 10 Years
Nielsen's law	Internet bandwidth	50%	57×
Moore's law	Computer power	60%	100×

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 important
- **Cloud – Dropbox, Spotify**
  - Streaming



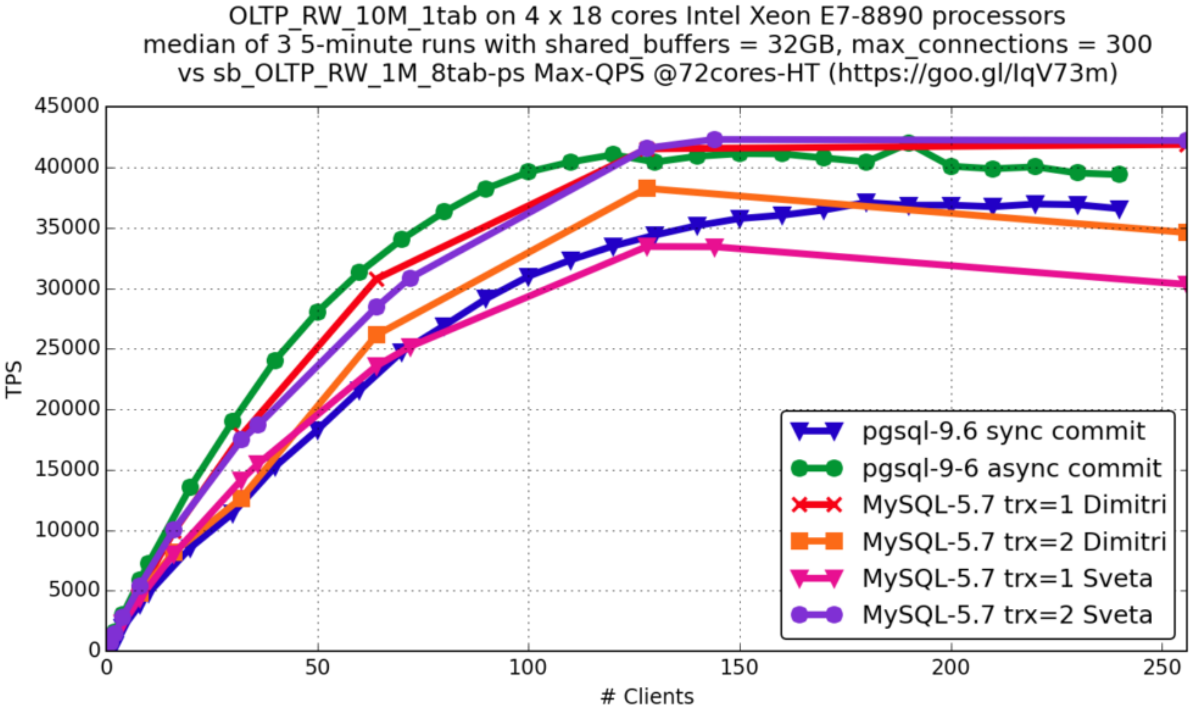


http://blog.dshr.org/2016/05/the-future-of-storage.html

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
- **Horizontal scaling = distributed system**

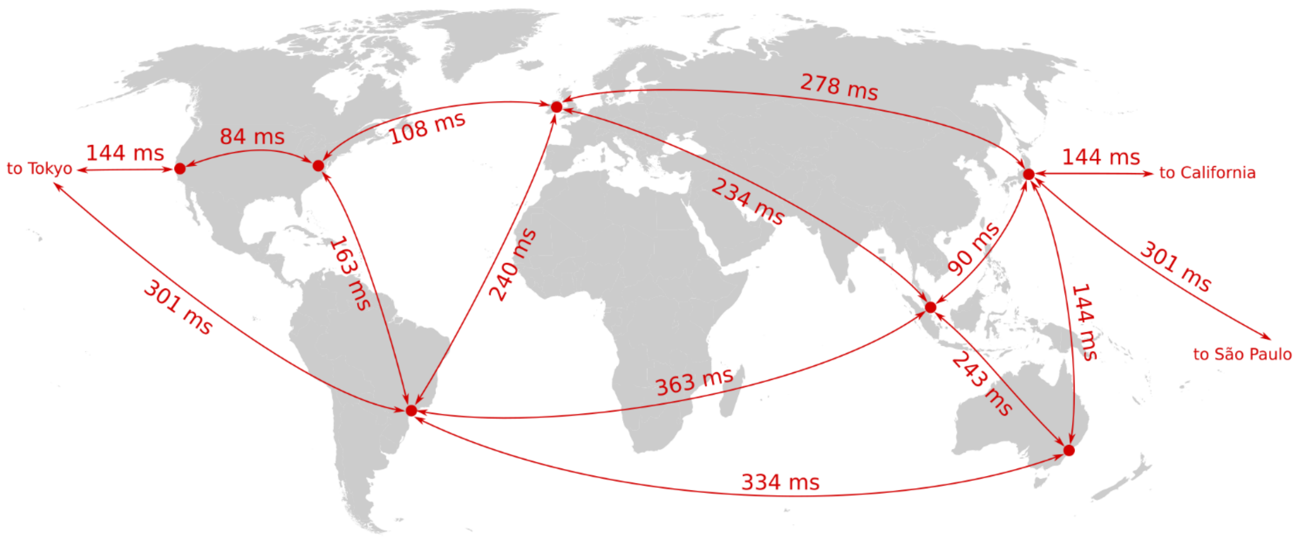


https://www.percona.com/blog/2017/01/06/millions-queries-per-second-postgresql-and-mysql-peaceful-battle-at-modern-demanding-workloads/

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

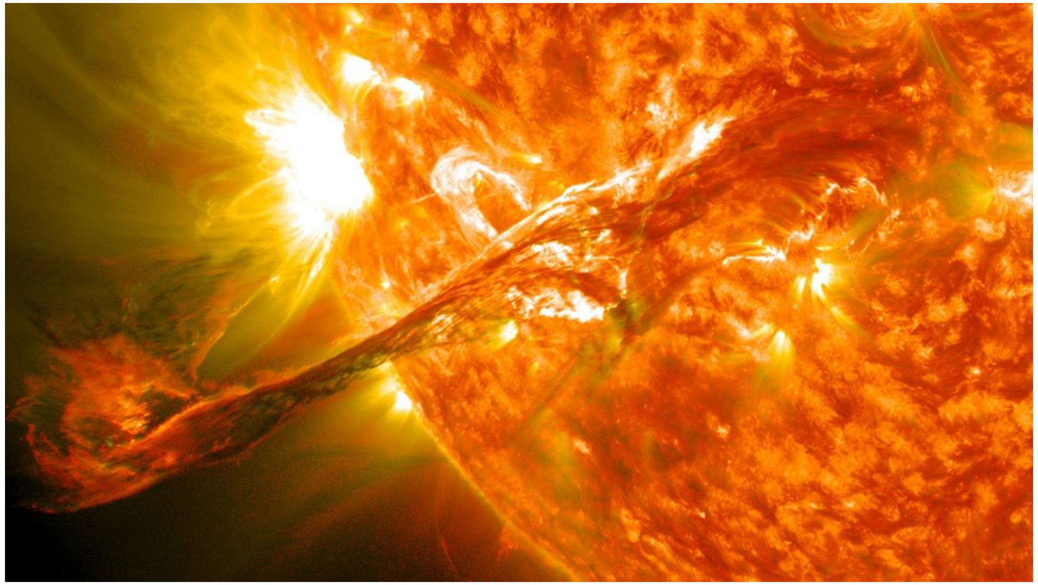


- **New protocols can decrease RTT**
  - Upcoming lecture
- **Place services closer to user**
  - Sometimes latency of 278ms is unacceptable
  - Gaming / Esports:
    - Human reaction time 200ms
    - Total from keypress to display:
      - Thinkpad 13 ChromeOS: 70ms
      - Lenovo X1 carbon 2016: 150ms
    - TV output lag ~15-30ms (random TV)
    - Keyboard 15-60ms
- **CDN: Content delivery network**
  - Place your images, sites, scripts close to your users

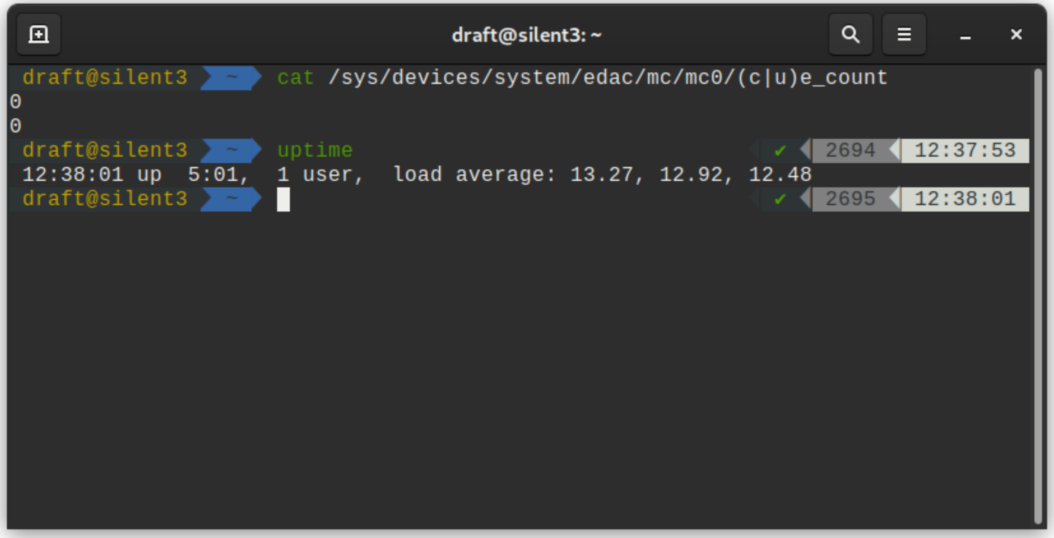
Distributed Systems Motivation

- **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: 1-5 bit errors per 8GB of RAM per hour
  - 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



- **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**



- **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,

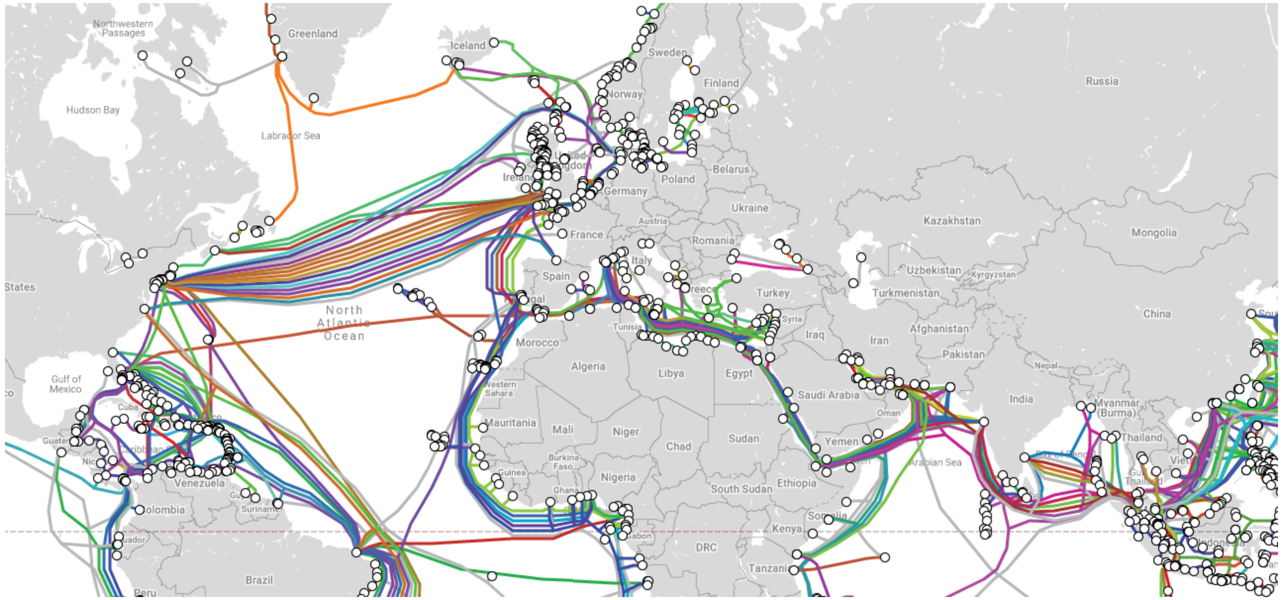
Bitsquat Domain	Original Domain
ikamai.net	akamai.net
aeazon.com	amazon.com
a-azon.com	amazon.com
amazgn.com	amazon.com
microsmft.com	microsoft.com
micrgsoft.com	microsoft.com

- **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

- **~06.01.2020** **Cable Cut Causes Yemen Internet Outage**
- **09.01.2020** **African Countries Affected by FALCON Undersea Cable Cut**
- **17.01.2020** **Gabon, Congo, DRC and Cameroon Experience Internet Disruption**
- **05.02.2020** **Repairs to submarine cables rescheduled**
  - 22.12.2019 Undersea cables' breakdown affects Internet connections in Vietnam
- **06.02.2020** **Google and Facebook turn their backs on undersea cable to China**
  - First undersea cable directly connecting Hong Kong and US (Dr Peng Telecom & Media Group → works closely with Huawei)

- **Submarine Cable Map**



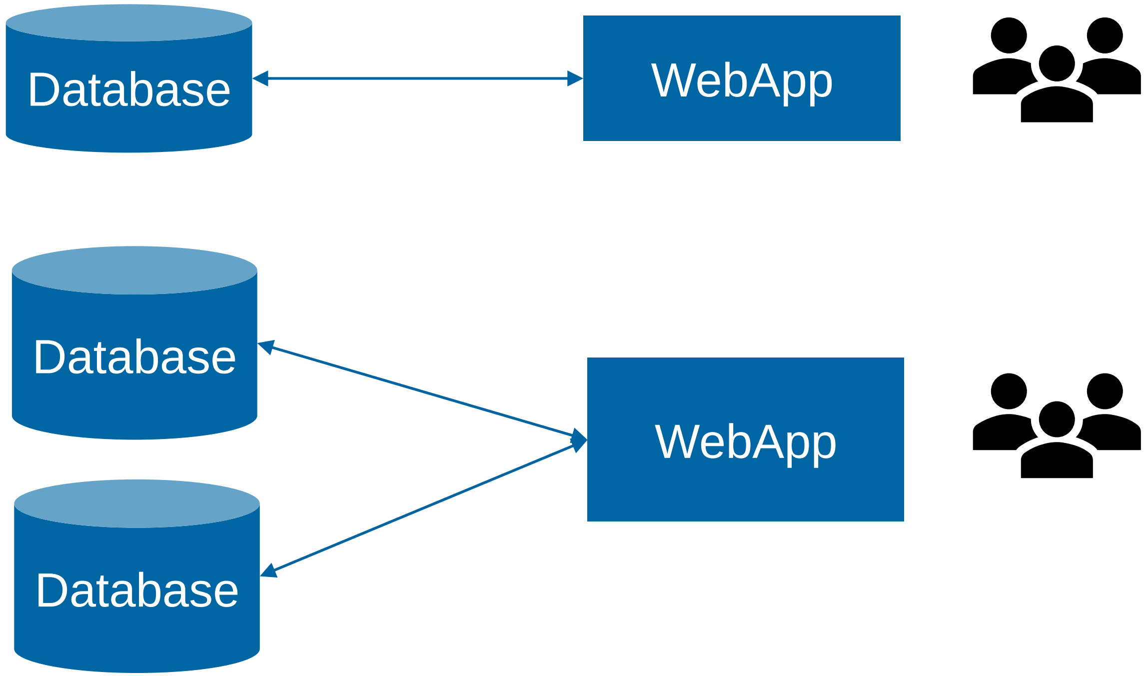
URL: b.socrative.com/student/

Room: HSR

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
- Need protocols TCP/UDP
  - RPC (gRPC, custom)
- Latency matters, Google uses internally gRPC
  - Companies tend to use standards
    LinkedIn's Kafka – own protocol
- Network is under control or client/server → no NAT issues

"Fully" Decentralized Systems

- Mechanisms that work well:
  - Consistent hashing (DHTs)
  - Flooding/broadcasting - Bitcoin
- Need protocols TCP/UDP
  - RPC (gRPC, custom)
- Often custom protocols, BitTorrent (µTP)
  - If it's a hobby project, write your own protocol, its much more fun!
- 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, only in fully distributed systems)
  - Proof of stake – Leader election, PBFT protocols, (distributed systems advanced)
  - 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

URL: b.socrative.com/student/

Room: HSR

Questions?

Dr. Thomas Bocek thomas.bocek@hsr.ch