Georgia Tech Computer Science

CS2200
Systems and Networks
Spring 2022

Datacenters

Alexandros (Alex) Daglis
School of Computer Science
Georgia Institute of Technology
alexandros.daglis@cc.gatech.edu

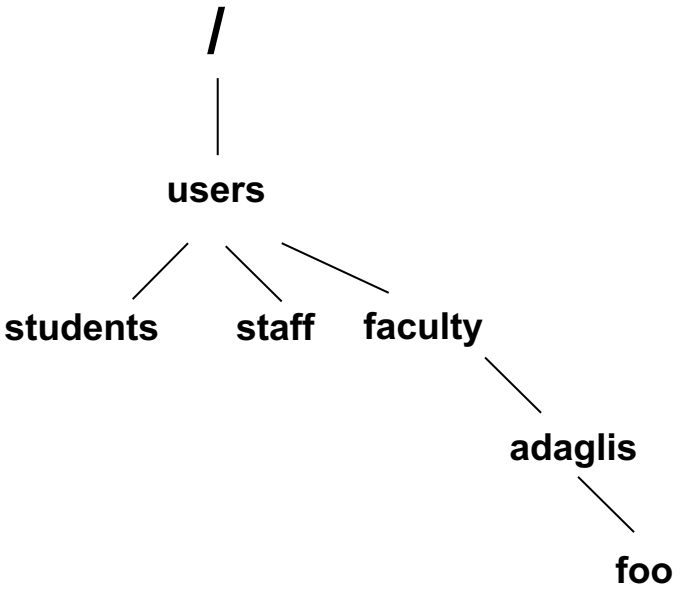
Agenda

- File Systems Wrap-up
- Datacenters

Allocation Strategy	File represen-tation	Free list mainte-nance	Sequential Access	Random Access	File growth	Allocation Overhead	Space Efficiency
Contiguous	Contiguous blocks	complex ✗	Very good ✓	Very good ✓	messy ✗	Medium to high ✗	Internal and external fragmentation ✗
Contiguous With Overflow	Contiguous blocks for small files	complex ✗	Very good for small files ✓	Very good for small files ✓	OK ✓	Medium to high ✗	" ✗
Linked List	Non-contiguous blocks	Bit vector ✓	Good but dependent on seek time ✓	Not good ✗	Very good ✓	Small to medium	Excellent ✓
FAT Partitioning not good for user	"	FAT ✓	" ✓	Good but dependent on seek time ✓	" ✓	Small ✓	" ✓
Indexed	"	Bit vector ✓	" ✓	" ✓	limited ✗	" ✓	" ✓
Multilevel Indexed Small Files ☹	"	Bit vector ✓	" ✓	" ✓	good ✓	" ✓	" ✓
Hybrid	"	Bit vector ✓	" ✓	" ✓	" ✓	" ✓	" ✓

Unix file system

- Metadata in the i-node
  - Access rights: U G O
  - User name (uid)
  - Group name (gid)
  - Size
  - Modification date
  - Data/index block addrs



"d" if directory

User Grou Othe

Links

Name (in directory)

```
-rw-r--r--   1 adaglis  faculty 475  Apr 21   2022 foo
```

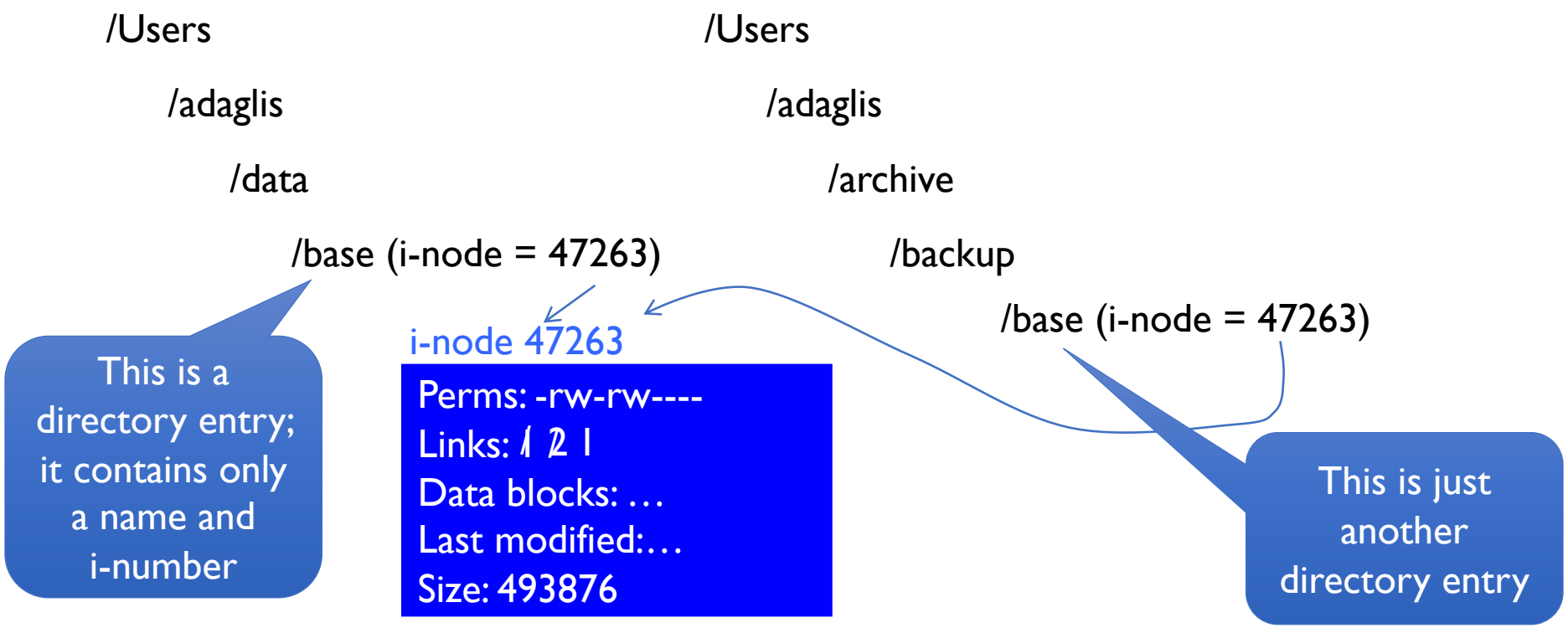
Links

- Remember that Unix-style file system separates the directory entry from the i-node; this means more than one directory entry can index the same i-node.
- When you create a file, you initialize an i-node and create a directory entry; this sets the link count in the i-node to 1
- You can also create additional links to the file with the link() system call or *ln* command; this increments the link count in the i-node for each link; this is called a "hard link"
- Curiously, there is no such thing as a primary or secondary link; both directory entries are names for the file and neither link has precedence over the other
- There is no system call to "remove" a file; you can only unlink a directory entry
- The file is deleted from the file system only when its link count reaches zero; you remove a link with the unlink() system call or the *rm* command
- Even better, each i-node also has an in-use count which counts the number processes that have the file open; the i-node is not deallocated until the link count AND the in-use count both reach zero, so you can have an open file that has an i-node but no directory entry and hence no name in the file system

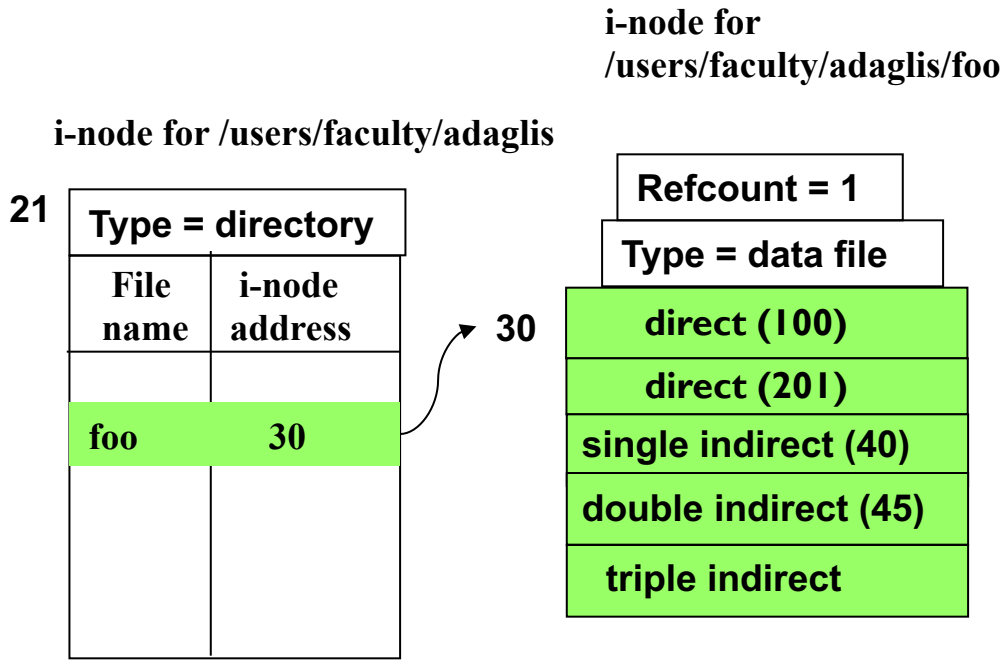
Symbolic links

- Unix also provides a way to create file aliases; these are called symbolic links
- A symbolic link occupies an i-node (contrast with a hard link doesn't take an additional i-node)
- The i-node is marked as a symbolic link and the data blocks contain a path name instead of user data
- When a symbolic link is encountered while following a path name, the path is replaced by the contents of the symbolic link and the search continues by following the path in the symbolic link
- The file system doesn't promise that symbolic links point to anything; the presence of a symbolic link pointing to a file name doesn't prevent that file from being unlinked
- Symlinks work across file systems contrasted with hard links that do not

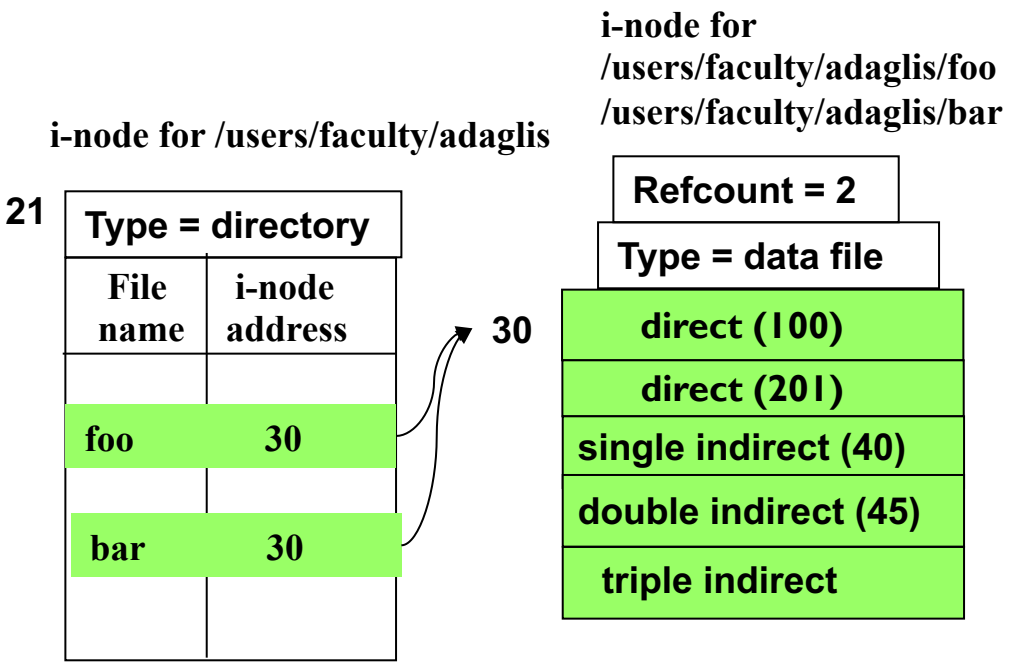
Unix file and links



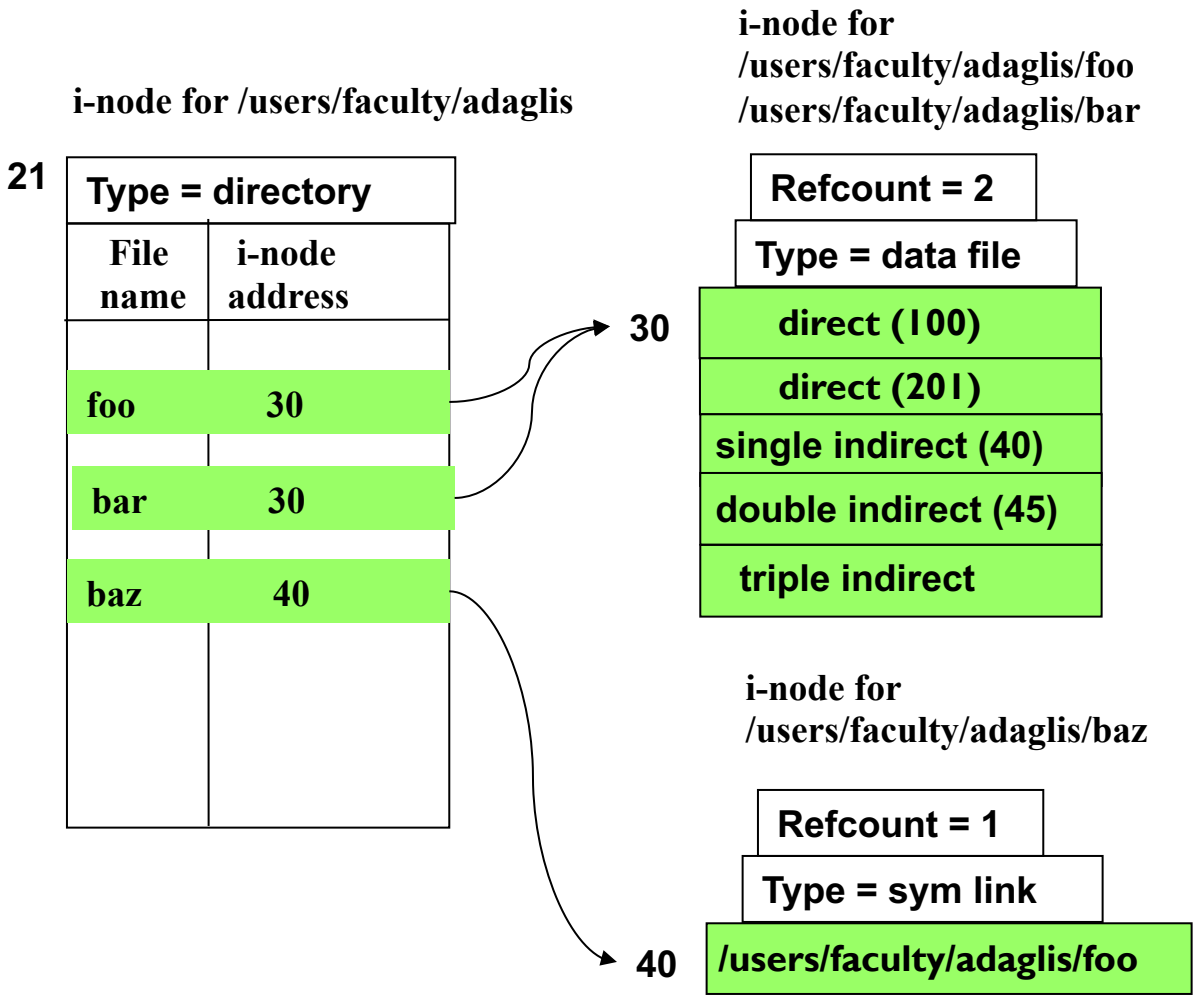

```
ln /Users/adaglis/data/base /Users/adaglis/archive/backup/base
rm /Users/adaglis/data/base # remember this calls unlink()
```



Reference count on files

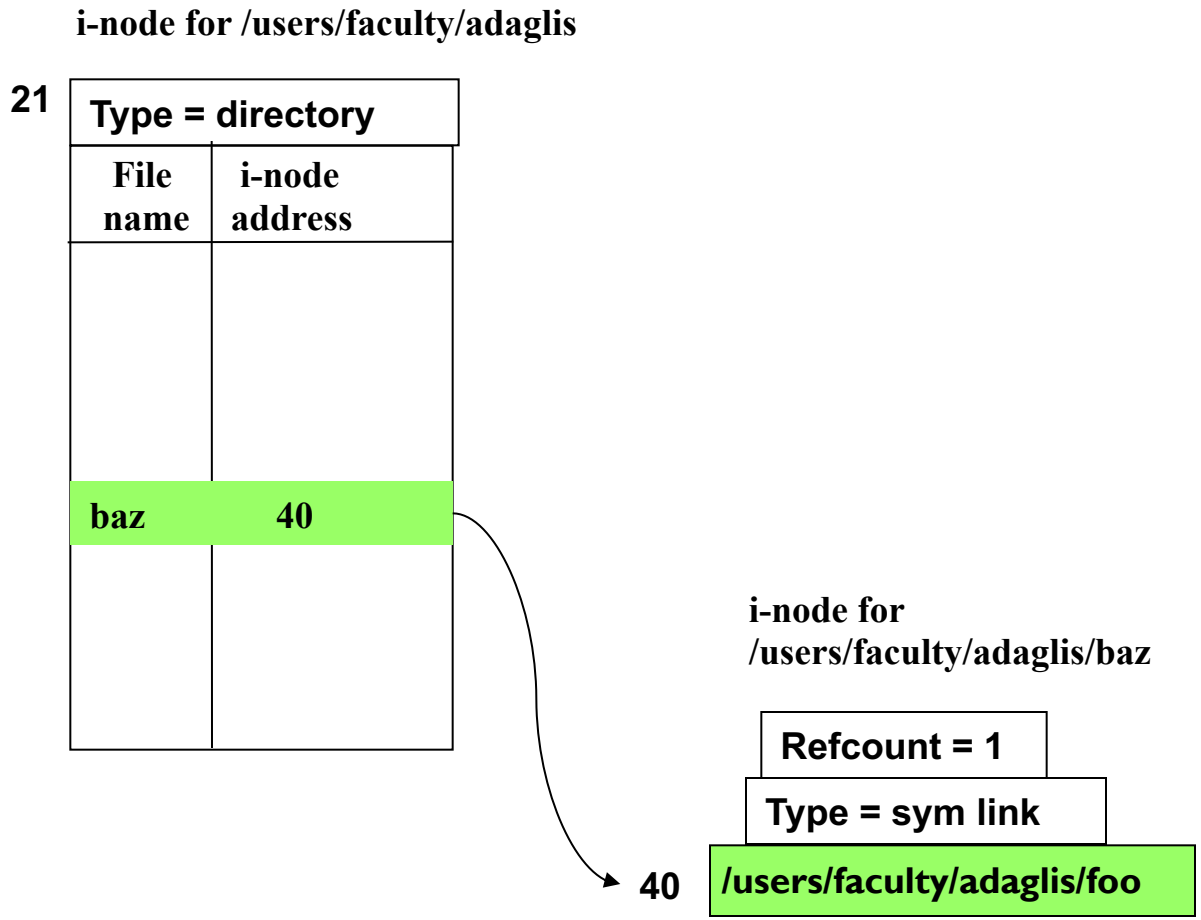

i-node for /users/faculty/adaglis
21
Type = directory
File name
i-node address
foo
30
bar
30
30
i-node for /users/faculty/adaglis/foo
/users/faculty/adaglis/bar
Refcount = 2
Type = data file
direct (100)
direct (201)
single indirect (40)
double indirect (45)
triple indirect

Hard links


i-node for /users/faculty/adaglis
21
Type = directory
File name
i-node address
foo 30
bar 30
baz 40
i-node for /users/faculty/adaglis/foo /users/faculty/adaglis/bar
30
Refcount = 2
Type = data file
direct (100)
direct (201)
single indirect (40)
double indirect (45)
triple indirect
i-node for /users/faculty/adaglis/baz
40
Refcount = 1
Type = sym link
/users/faculty/adaglis/foo

Sym links

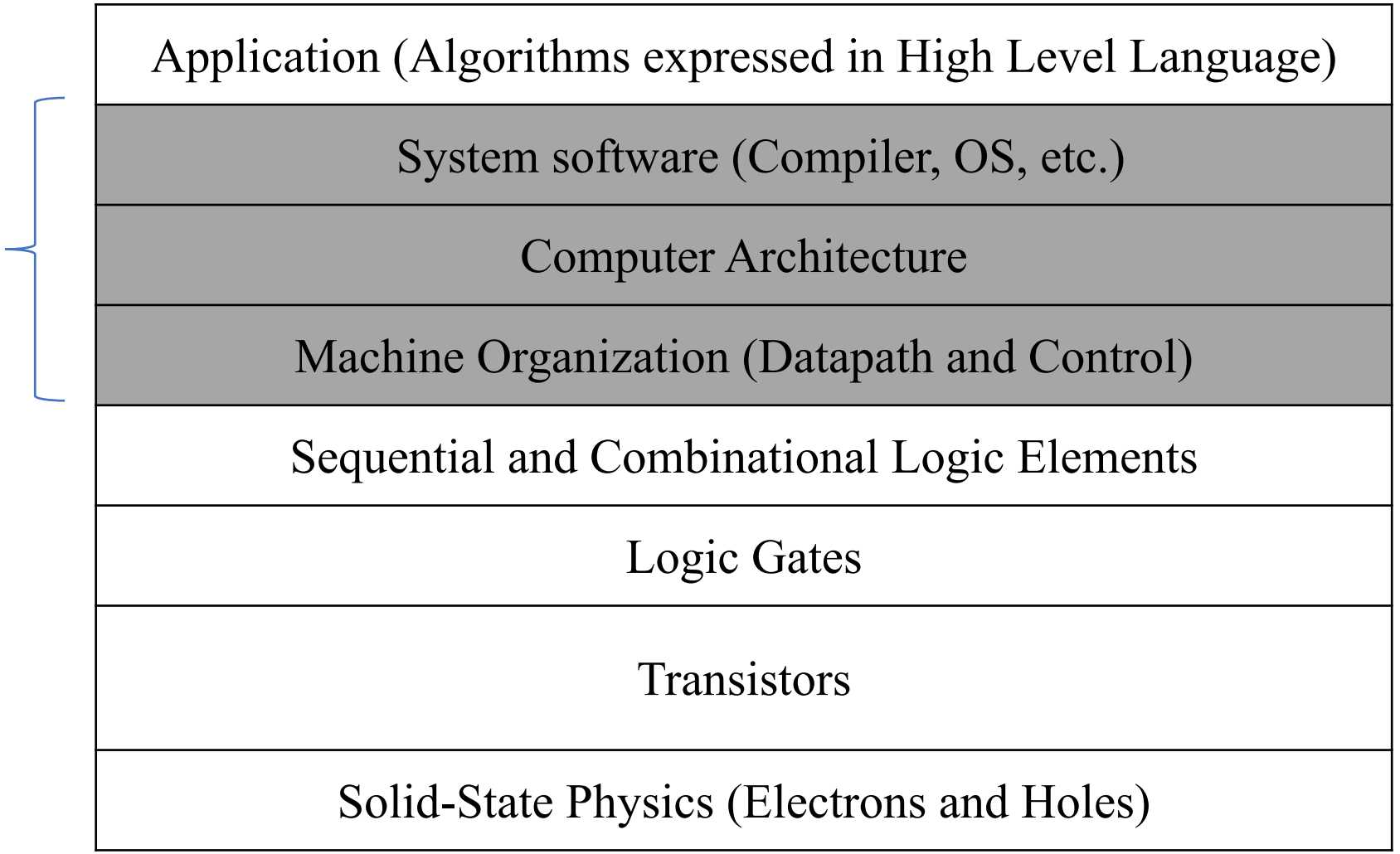
After deleting foo and bar

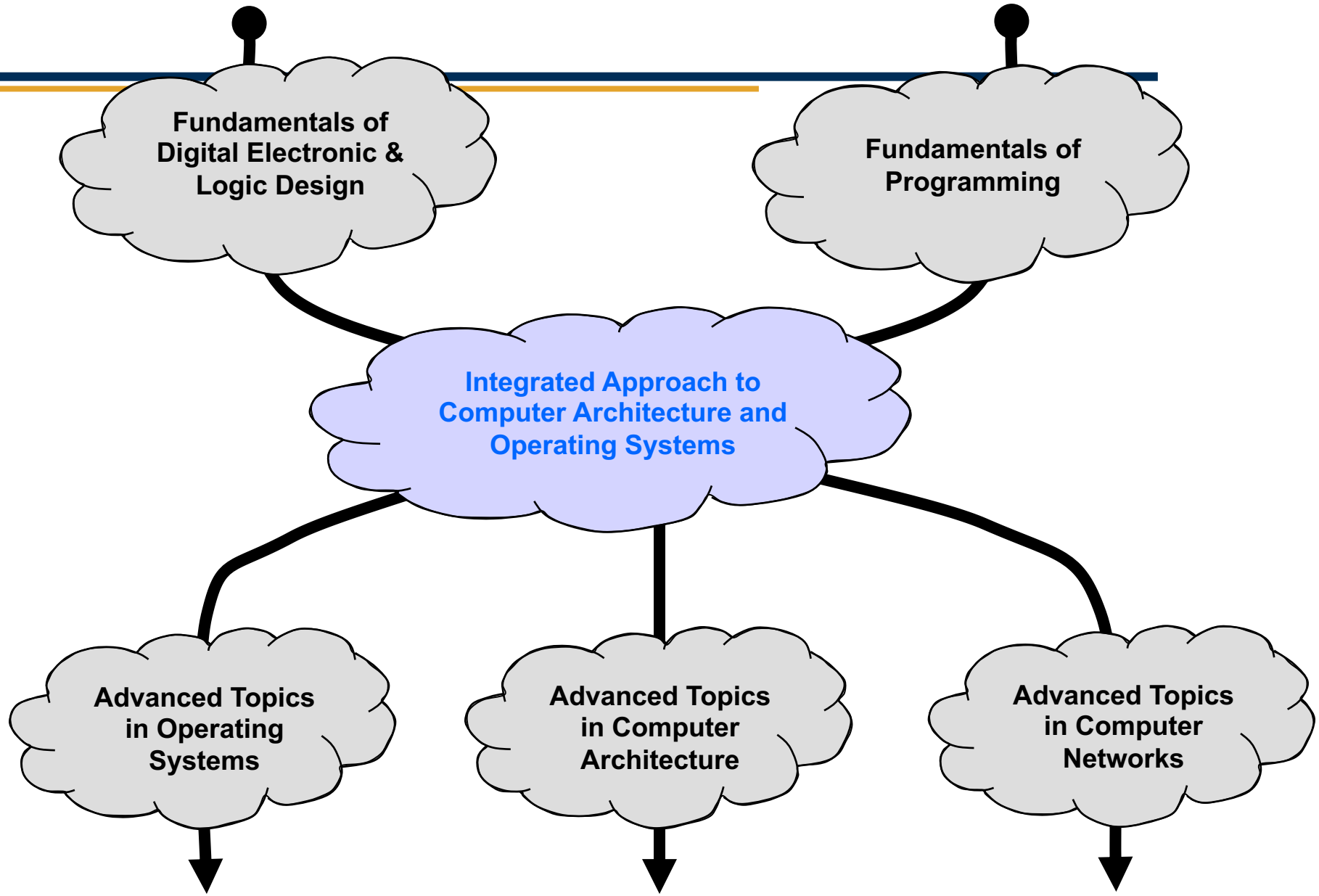


Attribute	Meaning	Elaboration
Name	Name of the file	Attribute set at the time of creation or renaming
Alias	Other names that exist for the same physical file	Attribute gets set when an alias is created; system such as Unix provide explicit commands for creating aliases for a given file; Unix supports aliasing at two different levels (physical or hard, and symbolic or soft)
Owner	Usually the user who created the file	Attribute gets set at the time of creation of a file; systems such as Unix provide mechanism for the file's ownership to be changed by the superuser
Creation time	Time when the file was created first	Attribute gets set at the time a file is created or copied from some other place
Last write time	Time when the file was last written to	Attribute gets set at the time the file is written to or copied; in most file systems the creation time attribute is the same as the last write time attribute; Note that moving a file from one location to another preserves the creation time of the file
Privileges •Read •Write •Execute	The permissions or access rights to the file specifies who can do what to the file	Attribute gets set to default values at the time of creation of the file; usually, file systems provide commands to modify the privileges by the owner of the file; modern Linux and Windows file systems such as ext4 and NTFS also provide an access control list (ACL) to give more granular access to different users
Size	Total space occupied on the file system	Attribute gets set every time the size changes due to modification to the file

Unix command	Semantics	Elaboration
touch <name>	**Create a file with the name <name>**	Creates a zero byte file with the name <name> and a creation time equal to the current wall clock time
mkdir <sub-dir>	**Create a sub-directory <sub-dir>**	The user must have write privilege to the current working directory (if <sub-dir> is a relative name) to be able to successfully execute this command
rm <name>	**Remove (or delete) the file named <name>**	Only the owner of the file (and/or superuser) can delete a file
rmdir <sub-dir>	**Remove (or delete) the sub-directory named <sub-dir>**	Only the owner of the <sub-dir> (and/or the superuser) can remove the named sub-directory
ln –s <orig> <new>	**Create a name <new> and make it symbolically equivalent to the file <orig>**	This is name equivalence only; so if the file <orig> is deleted, the storage associated with <orig> is reclaimed, and hence <new> will be a dangling reference to a non-existent file
ln <orig> <new>	**Create a name <new> and make it physically equivalent to the file <orig>**	Even if the file <orig> is deleted, the physical file remains accessible via the name <new>
chmod <rights> <name>	**Change the access rights for the file <name> as specified in the mask <rights>**	Only the owner of the file (and/or the superuser) can change the access rights
chown <user> <name>	**Change the owner of the file <name> to be <user>**	Only superuser can change the ownership of a file
chgrp <group> <name>	**Change the group associated with the file <name> to be <group>**	Only the owner of the file (and/or the superuser) can change the group associated with a file
cp <orig> <new>	**Create a new file <new> that is a copy of the file <orig>**	The copy is created in the same directory if <new> is a file name; if <new> is a directory name, then a copy with the same name <orig> is created in the directory <new>
mv <orig> <new>	**Renames the file <orig> with the name <new>**	Renaming happens in the same directory if <new> is a file name; if <new> is a directory name, then the file <orig> is moved into the directory <new> preserving its name <orig>
cat/more/less <name>	**View the file contents**	

What did we learn in this course?




Fundamentals of Digital Electronic & Logic Design
Fundamentals of Programming
Integrated Approach to Computer Architecture and Operating Systems
Advanced Topics in Operating Systems
Advanced Topics in Computer Architecture
Advanced Topics in Computer Networks

Where do I go from here?

- CS 3210 – OS Design
- CS 3220 – Processor Design
- CS 3251 – Networking
- CS 3240/4240 – Compilers
- CS 4290/6290 – Advanced/High Performance Computer Architecture
- CS 4210/6210 – Advanced OS
- CS 6250 – Computer Networks

Reminder: CIOS is open

- Your feedback helps me improve cs2200!
  - Your future peers will be thankful
  - What worked well? Suggestions for improvement?

- CIOS participation incentive:

  Entire class gets **1% bonus** to total grade if we reach **95% participation**

  As of this morning, completion rate was 52%

Agenda

- File Systems Wrap-up
- Datacenters

What is a Datacenter?

"A large group of networked computer servers typically used by organizations for the remote storage, processing, or distribution of large amounts of data."

- *Oxford dictionary*

"Datacenters are buildings where multiple servers and communication gear are co-located because of their common environmental requirements and physical security needs, and for ease of maintenance."

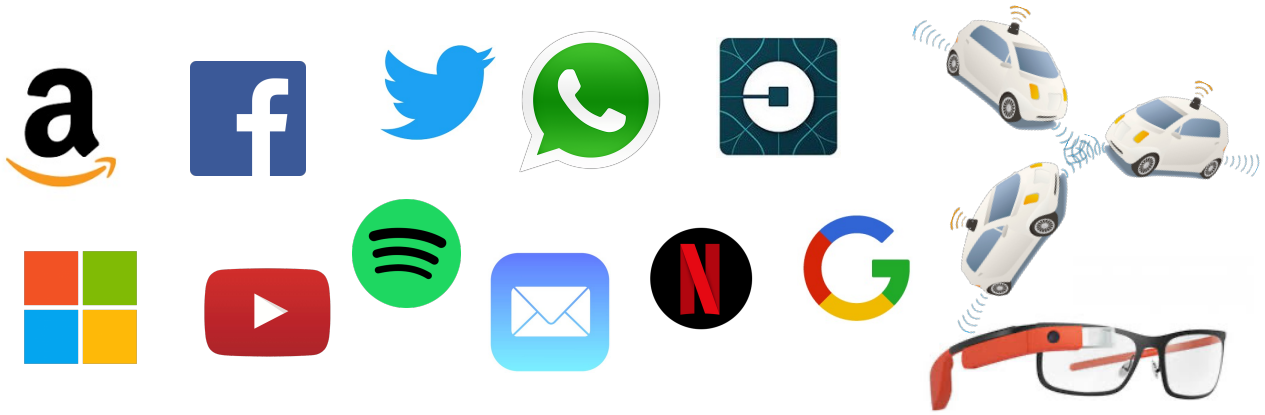
- Barroso* et al., *The Datacenter as a Computer*

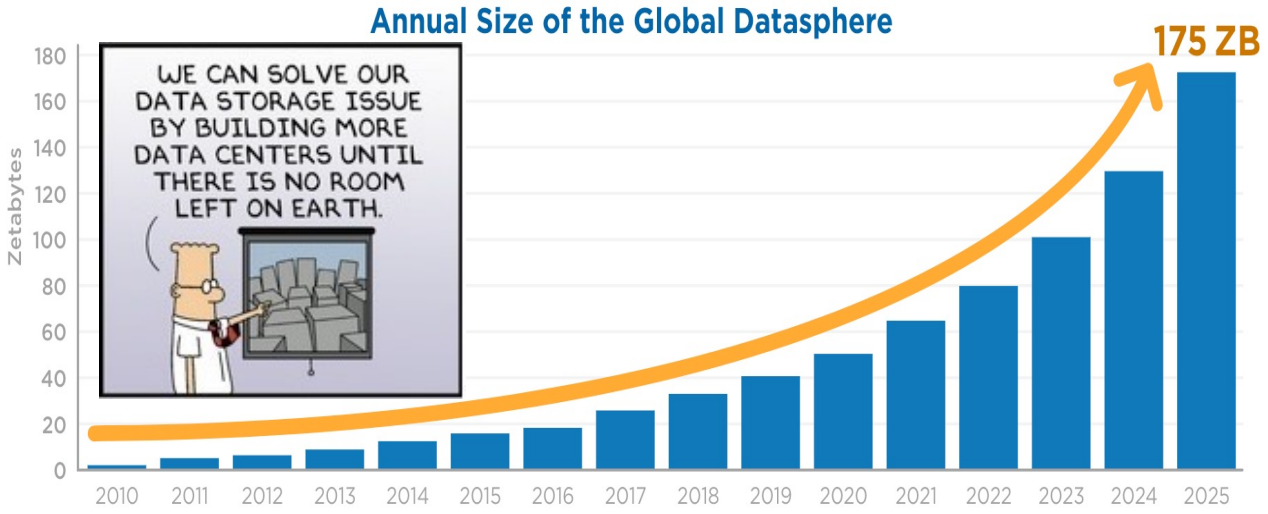
Term originated in the 90s with the advent of client-server architecture

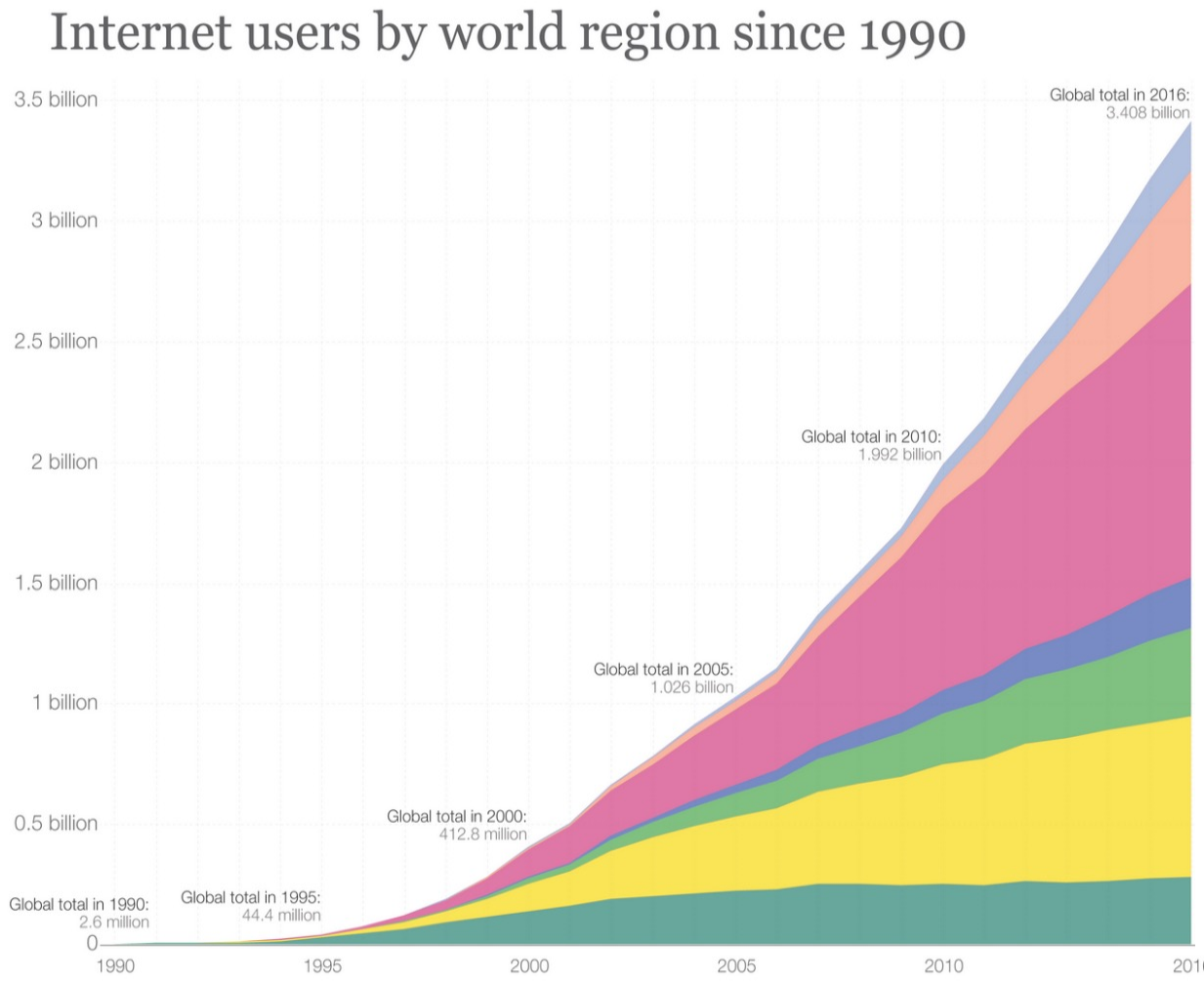
**Recipient of 2020 Eckert-Mauchly award*
"for pioneering the design of warehouse-scale computing and driving it from concept to industry"

Growth of Large-Scale Internet Services

Driven by growing demands: more services, more data, more users

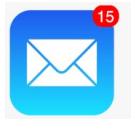






Some Mind-Boggling Stats

In the duration of this lecture, there will have been:

13 billion  sent

38 million 

340 million 

340PB of internet traffic 

- 360 million video views
- 2.5 years worth of video uploaded
  (in 2010, the same statistic was "only" 3.5 months)

Service Families Powered by Datacenters

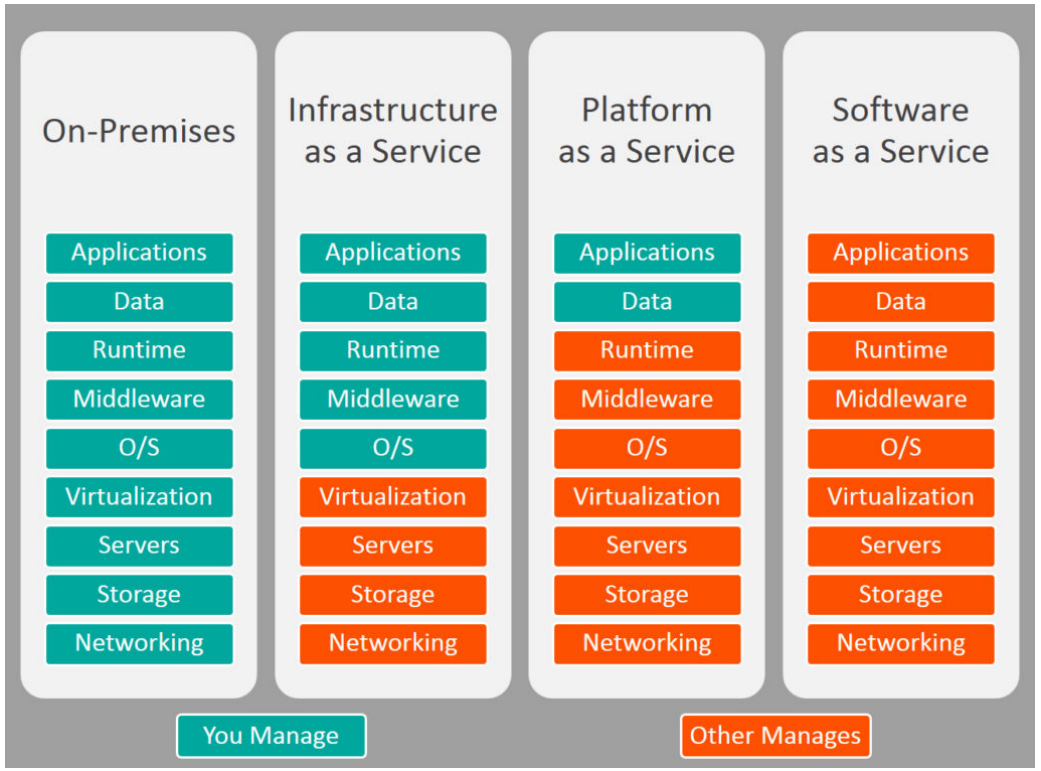
IaaS – Infrastructure as a Service
- E.g., AWS EC2, Microsoft Azure

PaaS – Platform as a Service
- E.g., Facebook

SaaS – Software as a Service
- E.g., Google docs, Gmail

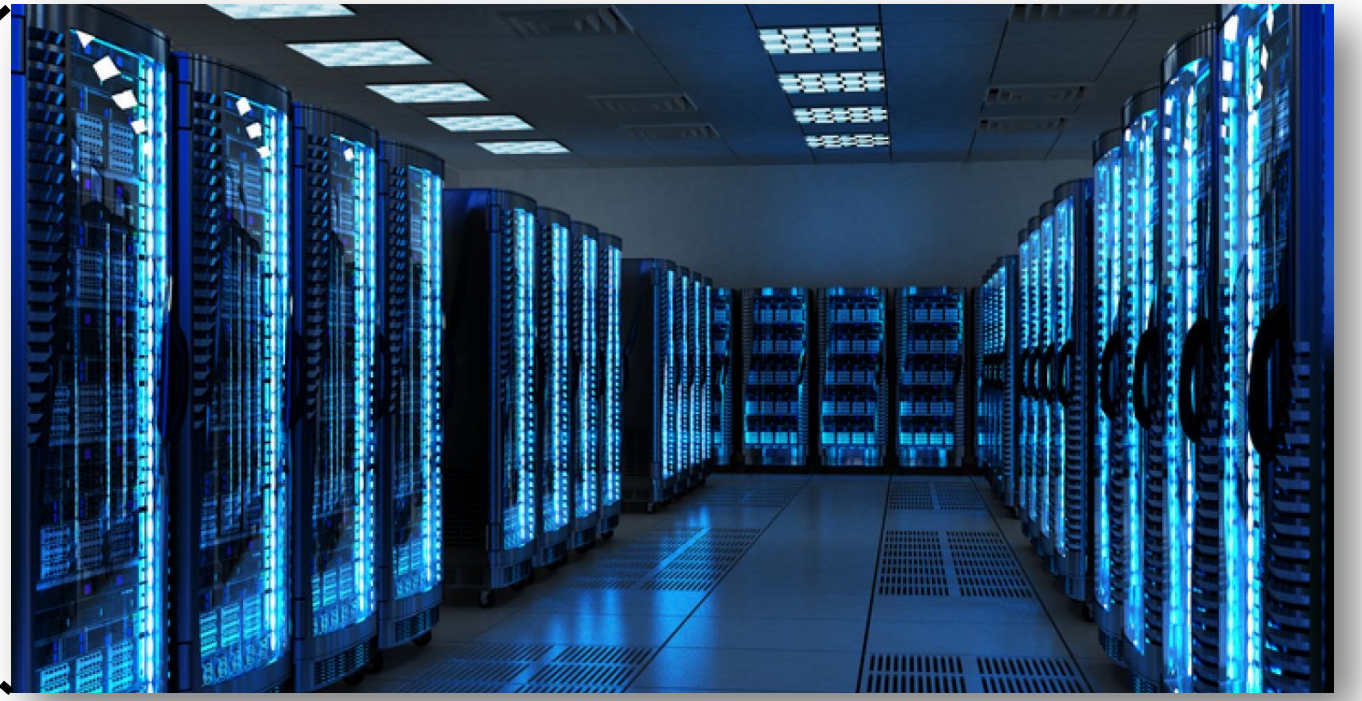
FaaS – Function as a Service
- AWS Lambda, Google Cloud Function

CaaS – Container as a Service
- Container orchestration platform – e.g., Google Kubernetes, Docker Swarm



Datacenters Are the World's Largest Computers

a.k.a., "warehouse-scale computers"



- 10-100x football field
- 10^4-10^5 servers
- 10^6 cores
- 10^{17} bytes
- 10s of MW

Architectural Overview

Network pools resources together

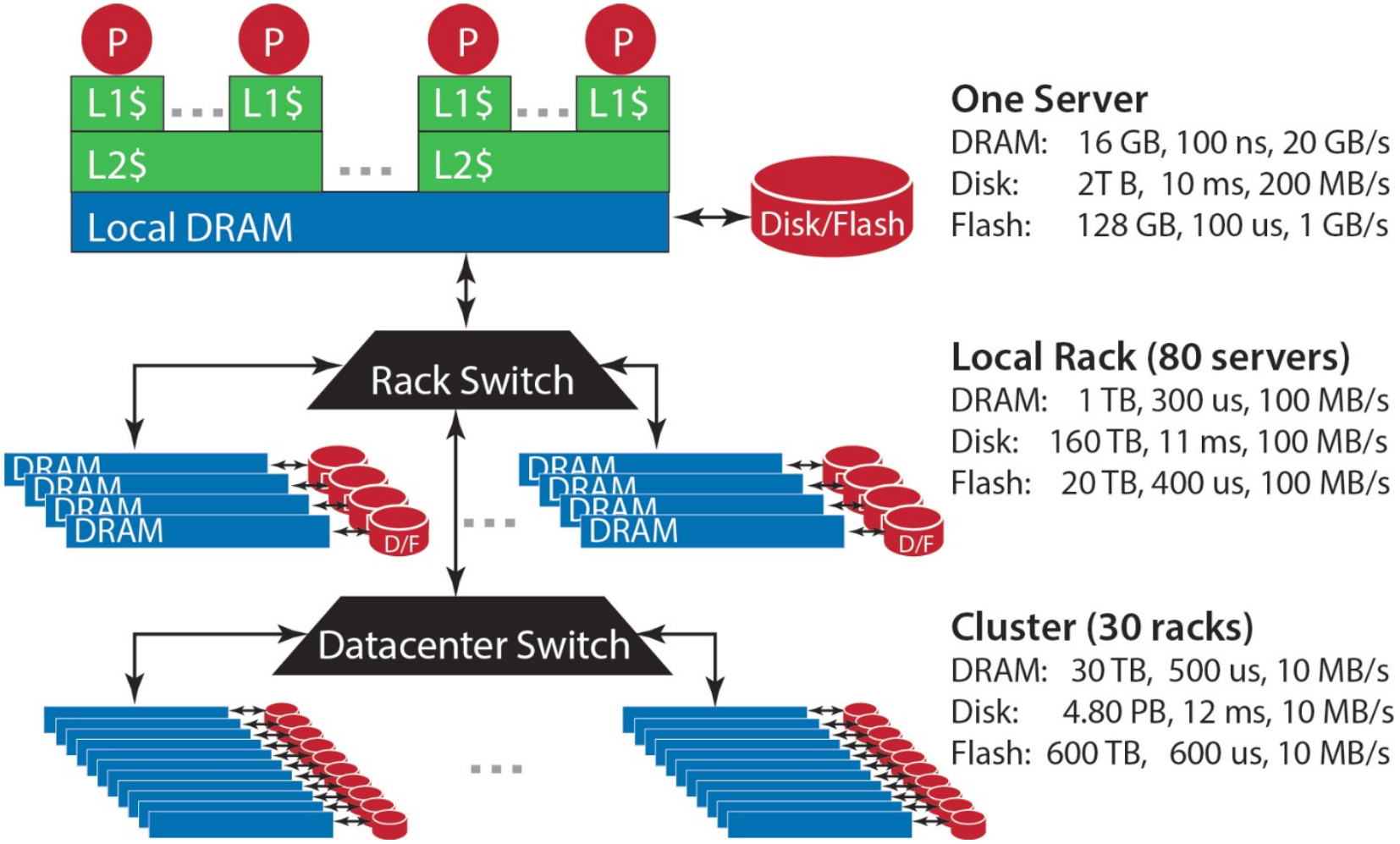
Scalability via Hierarchical Designs

Hierarchy in

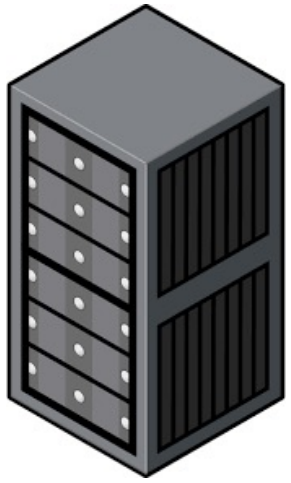
- Memory
- Storage
- Network

Affects

- Data placement
- Job scheduling
- Performance variability
- Cost



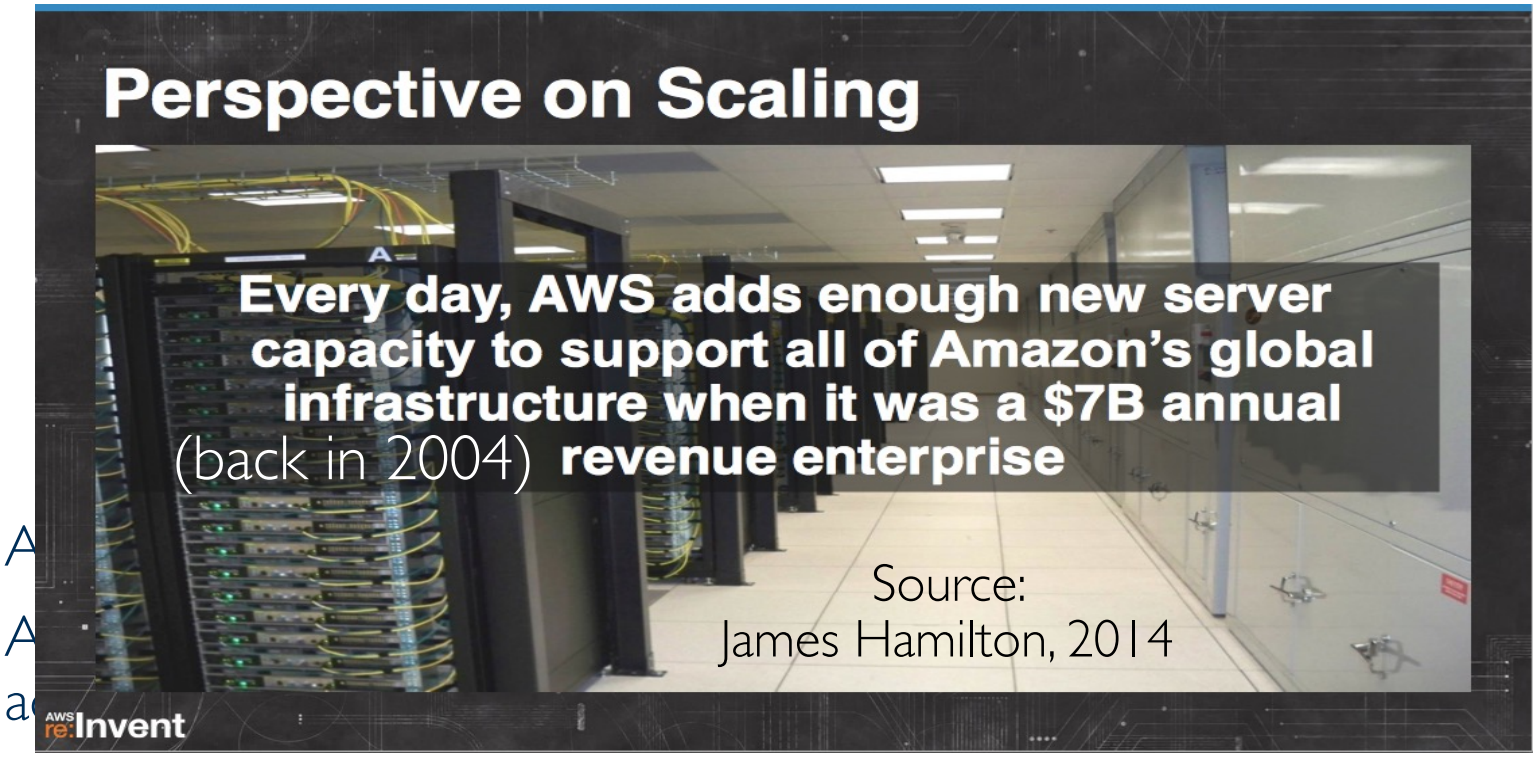
Datacenter Scaling



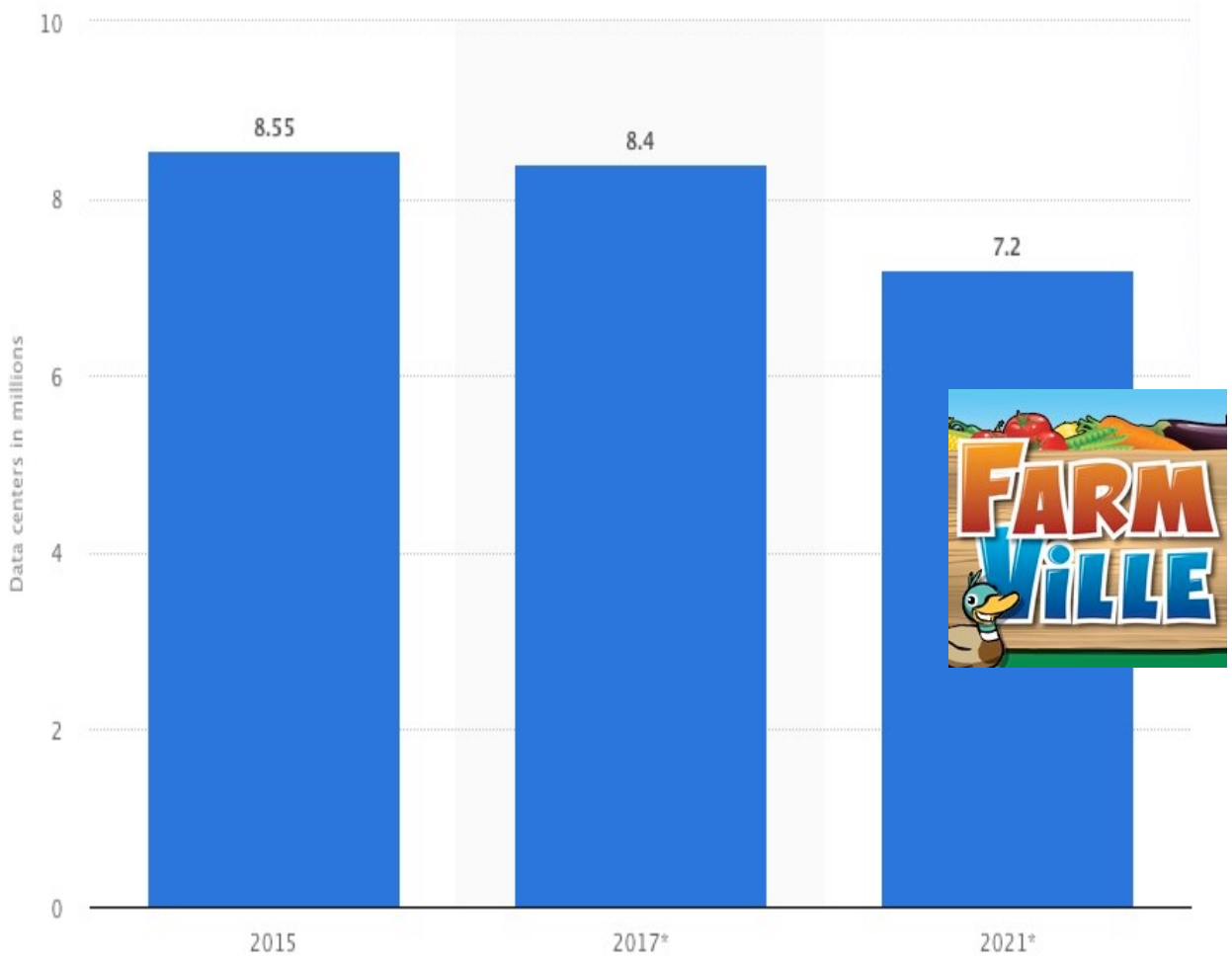
=

- Amazon in 2004
- $7B revenue
- **Total** hardware resources

- A
- A
a



Datacenter Scaling



- Number of datacenters dropping?!
- But average size is growing!
  → more hyperscalers

Zynga Ditches Data Center Plans For AWS

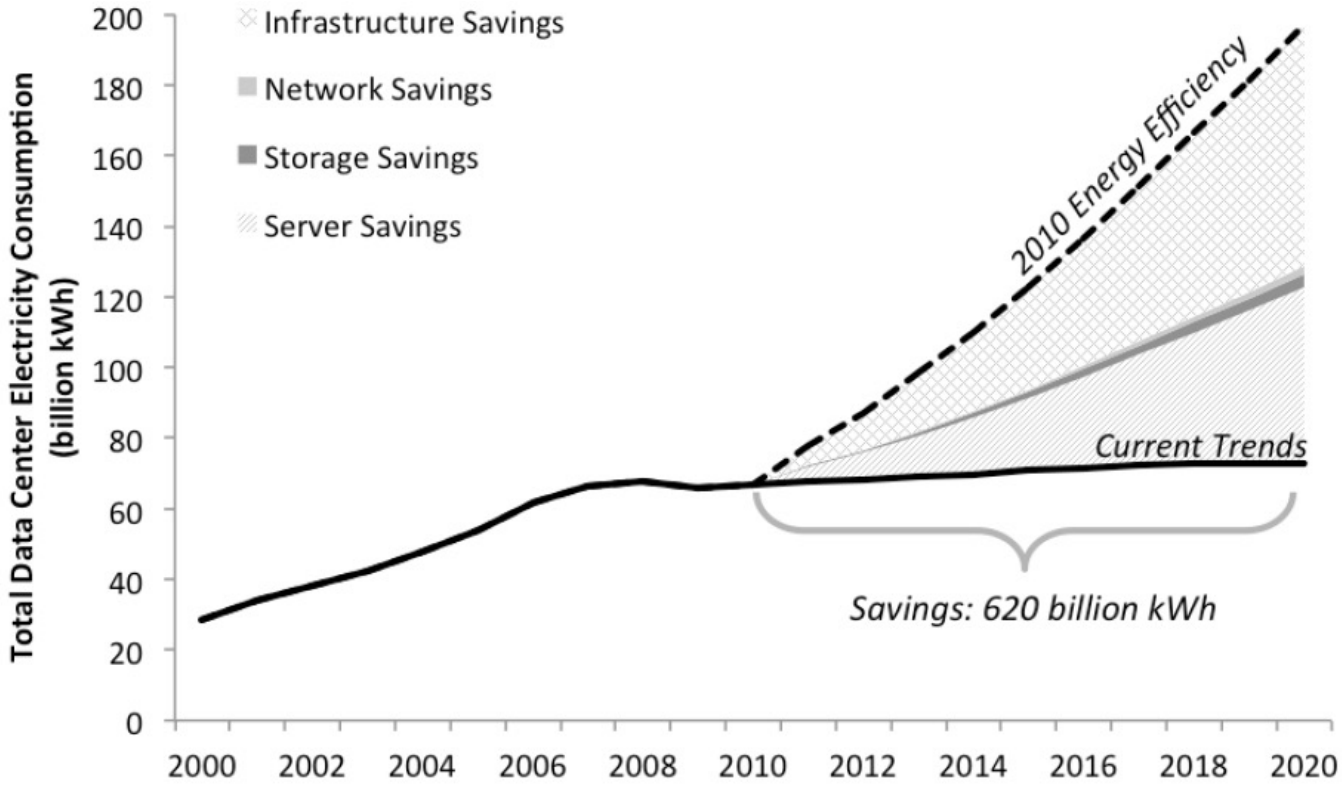
The casual gaming company spent over $100 million on data centers as it shifted away from Amazon Web Services. The company has come full circle, back to relying on AWS

Datacenter Power

2% of world's electricity

Dramatic recent improvements

- Focus on better energy efficiency
- Annual growth rate dropped
  15% (2000-2005) → 5% → 3%
- More hyperscalers → Improved efficiency



A hyperscaler's datacenter

The same Microsoft datacenter today

A Microsoft datacenter a couple of years back

Key Challenges of Scale

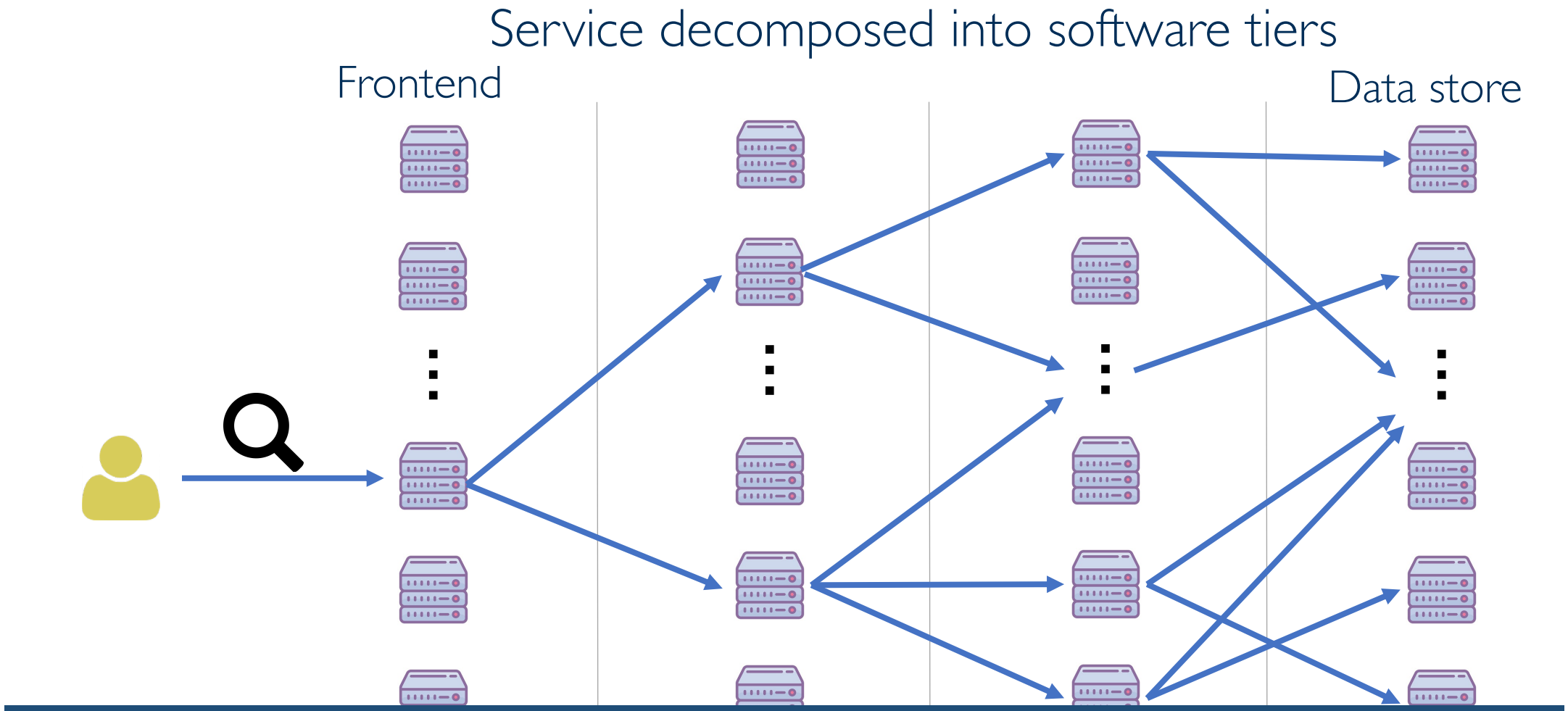
Massive scale enabled by using commodity parts at high volume

Challenges:

- Fault tolerance: With an MTTF of 10 years and 10k disks, ~3 disk failures **per day**
- Availability & performance predictability: services come with guarantees of 3+ 9s

General approach: Leverage redundancy and develop sophisticated software techniques for proper management

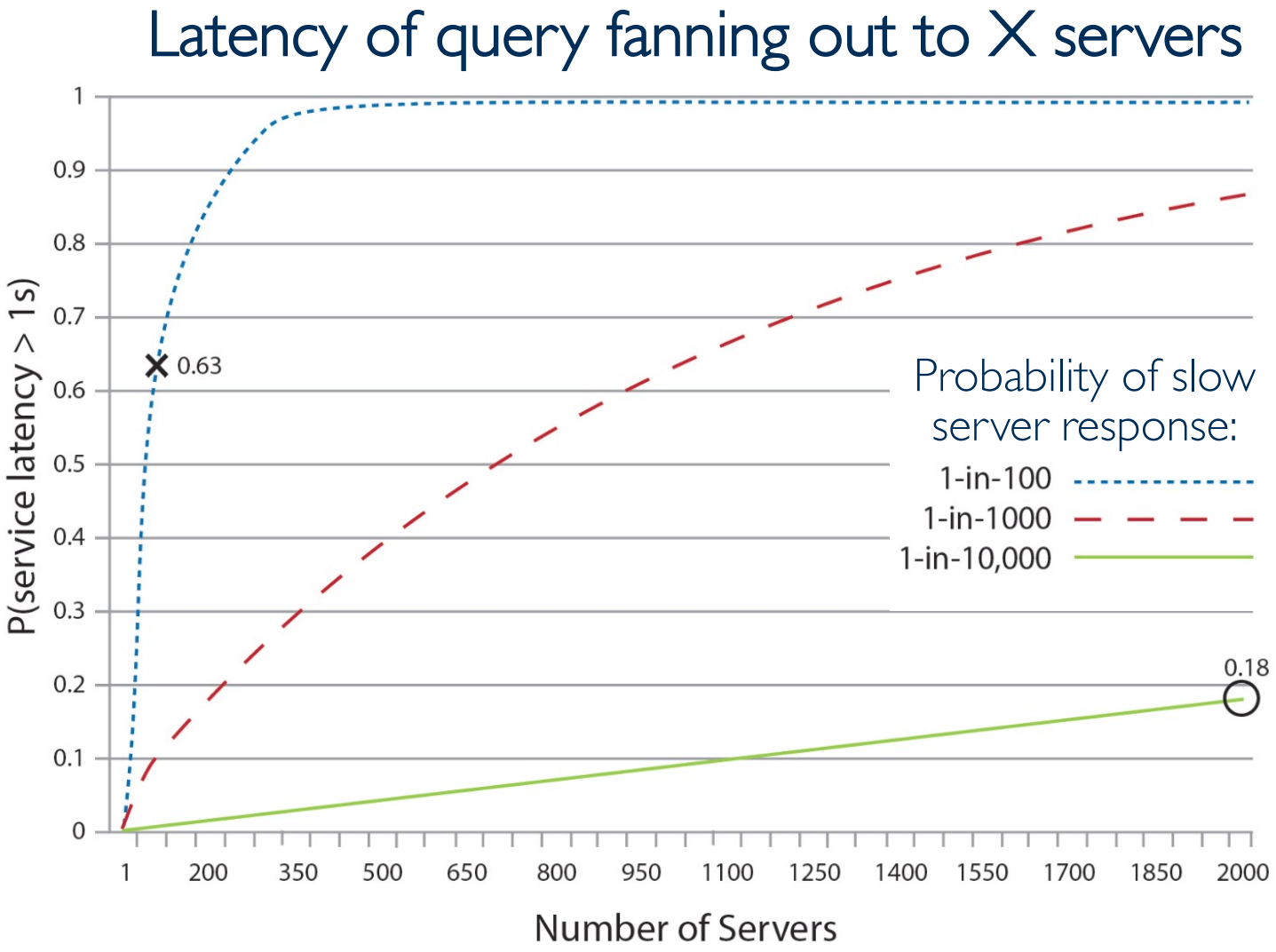
High-level View of Service Deployment



Fan-out enables scalability, but introduces special challenges

The *Tail at Scale* Challenge

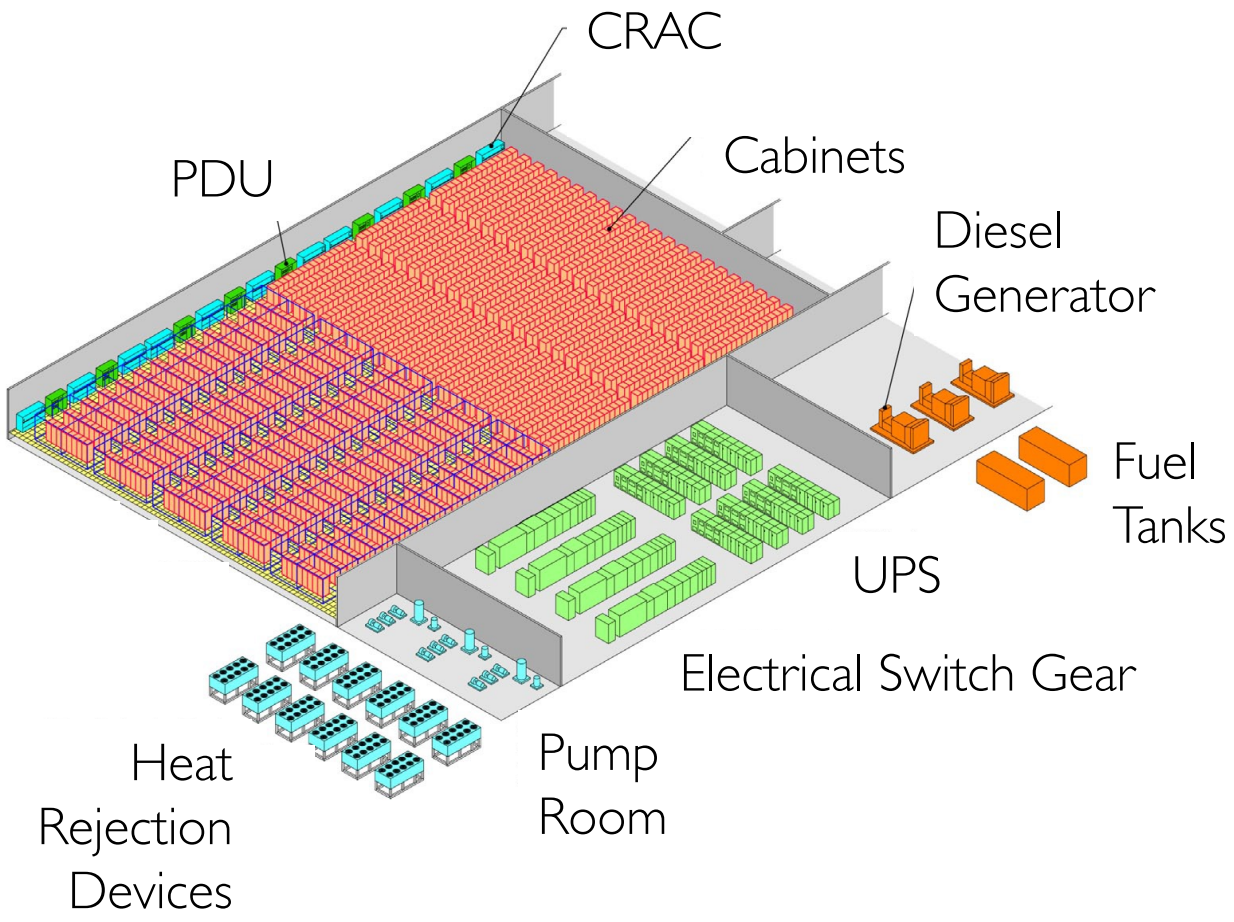
- Single query spreads to 100s-1000s of servers
- One straggler server can affect whole query
- With 1% prob of slow response, a query fanning out to 100 servers will be slow 63% of the time!



Datacenter Considerations Extend Far Beyond Computing

Numerous non-computing concerns

- Power delivery
- Cooling
- Form factor considerations
- Datacenter economics – TCO
- …



Datacenter Sustainability

Based on MICRO 2020 Keynote talk by Bobbie Manne (Microsoft)

Datacenter Energy & CO2

205TWh of electricity/year used

Equivalent to 1 year of:

31.3M cars

31,292 wind turbines

189.3M acres of forestland (Texas is ~172M acres)

145M metric tons CO2

Enlightening Numbers

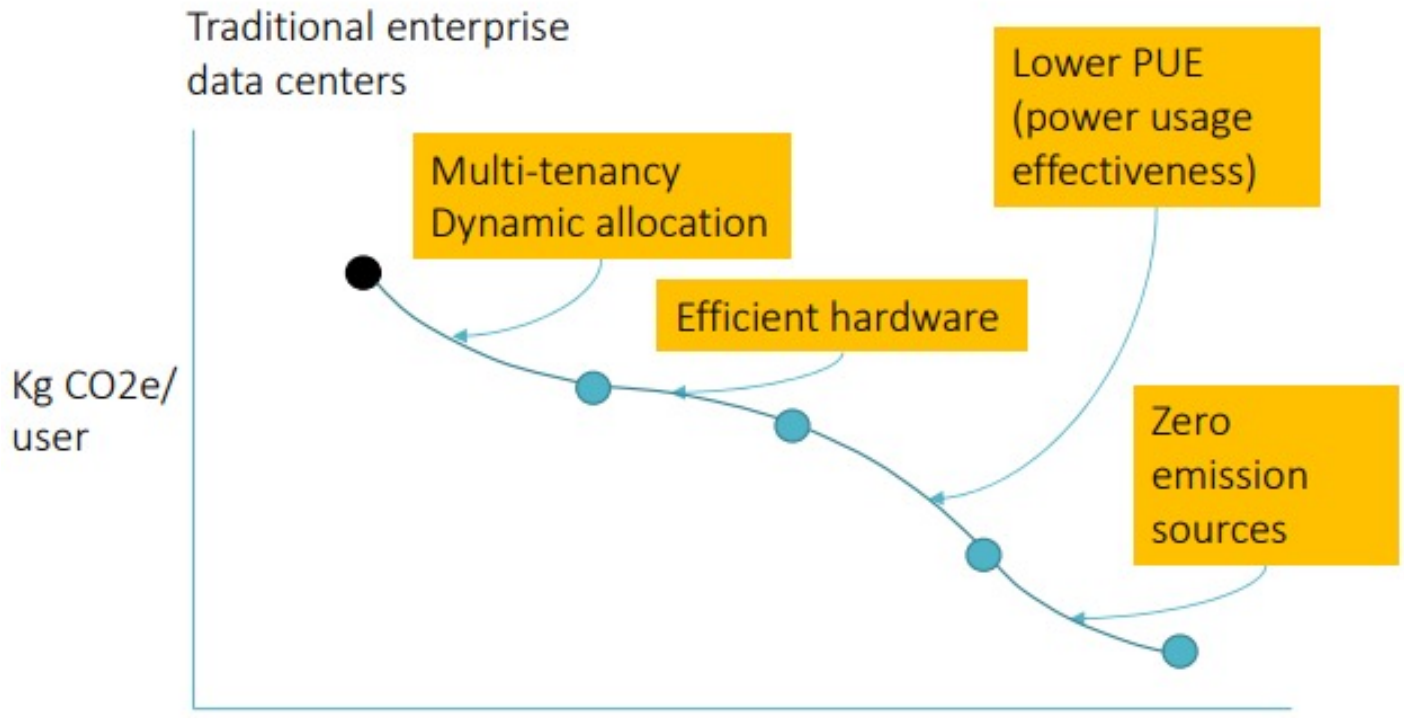
- Total DC electricity = 31M cars

- 1 monthly Facebook user = driving a car 0.25 miles
  - Total Facebook users = 54k cars

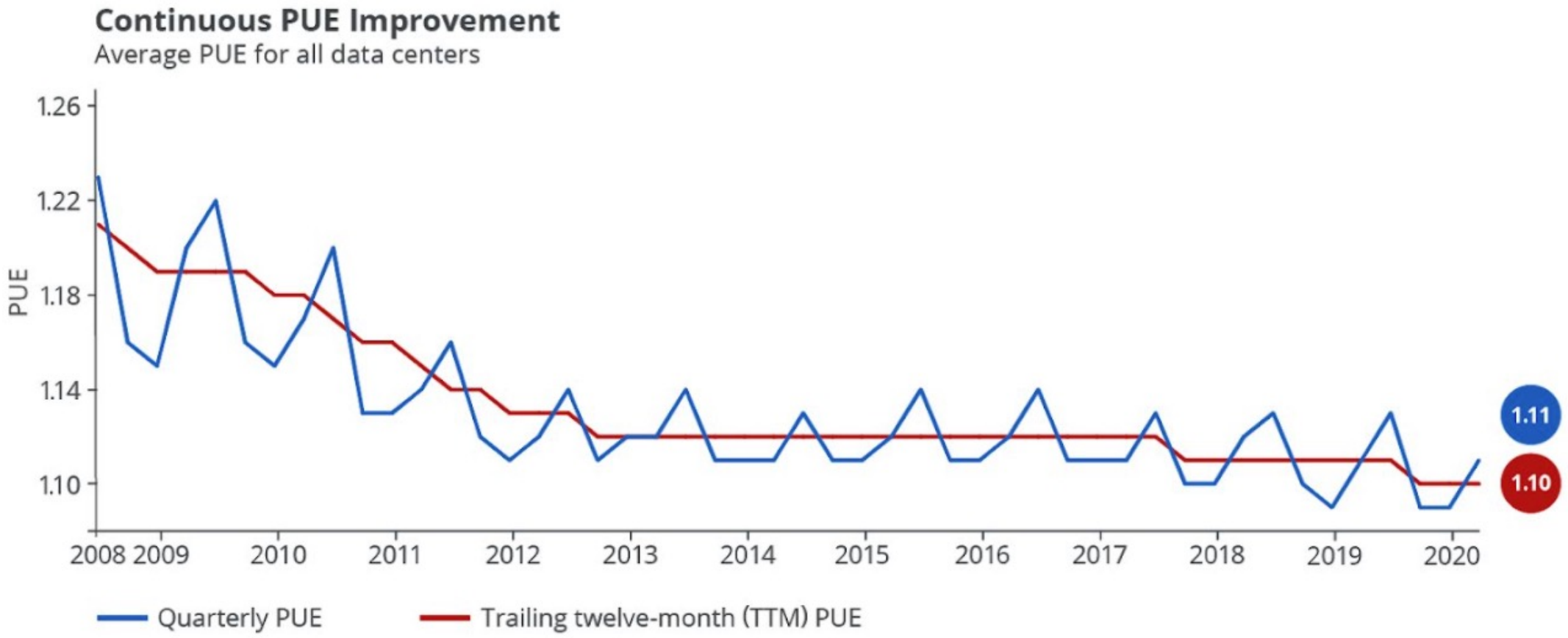
- Netflix accounts for 1.1M-2.2M cars
  - 165M hours of daily streaming

Efficiencies of Scale

- Benefits of hyperscalers
  - 2x increase in users and 12x increase in traffic, only 6% increase in energy



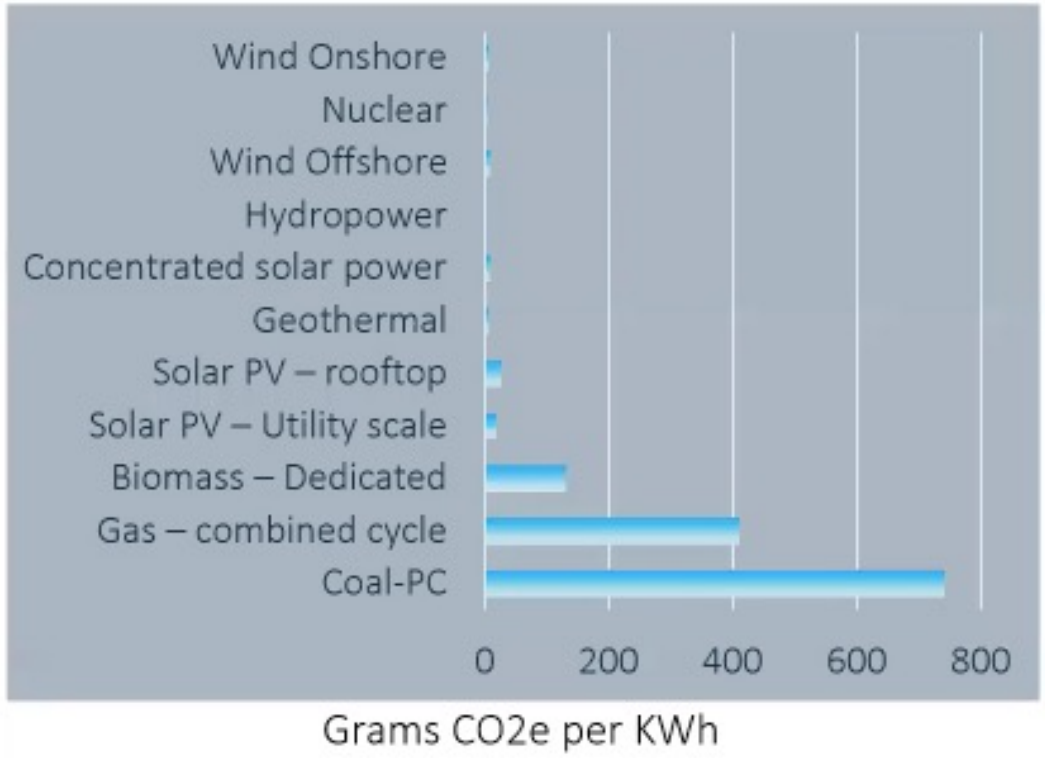
PUE of Google datacenters over time

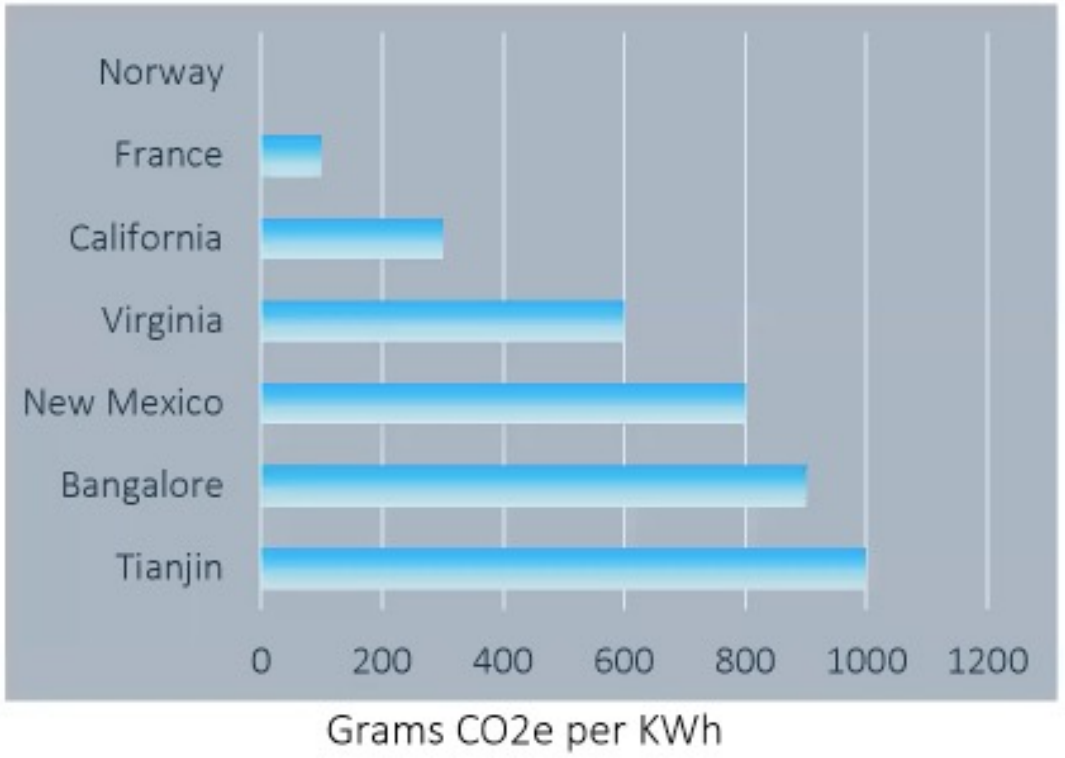


Think of the end-to-end system

- Not just the datacenter – but the devices as well
  - E.g., for Netflix, both DC and receiving device
- Not just the running energy – much more than servers
  - "Embodied" and "use" carbon
  - Buildings and components in buildings
  - Every step involves energy consumption and emissions
- Need to consider each component's lifecycle, defining amortization period

Not all electricity is created equal





- Source of energy defines corresponding emissions
- Wide range of carbon intensity

Emission-aware workload placement

- Not just locality/utilization/latency/performance criteria
  - Placement decision may dictate carbon footprint
  - One more source of variability: carbon intensity of currently available energy source

- Consider shifting work in space and time to use type of energy with better carbon intensity

Sustainability is a multi-faceted topic

Carbon emissions are one aspect of it → directly tied to global warming

- Tied to energy consumption
- Energy is spent in many forms - not just electricity

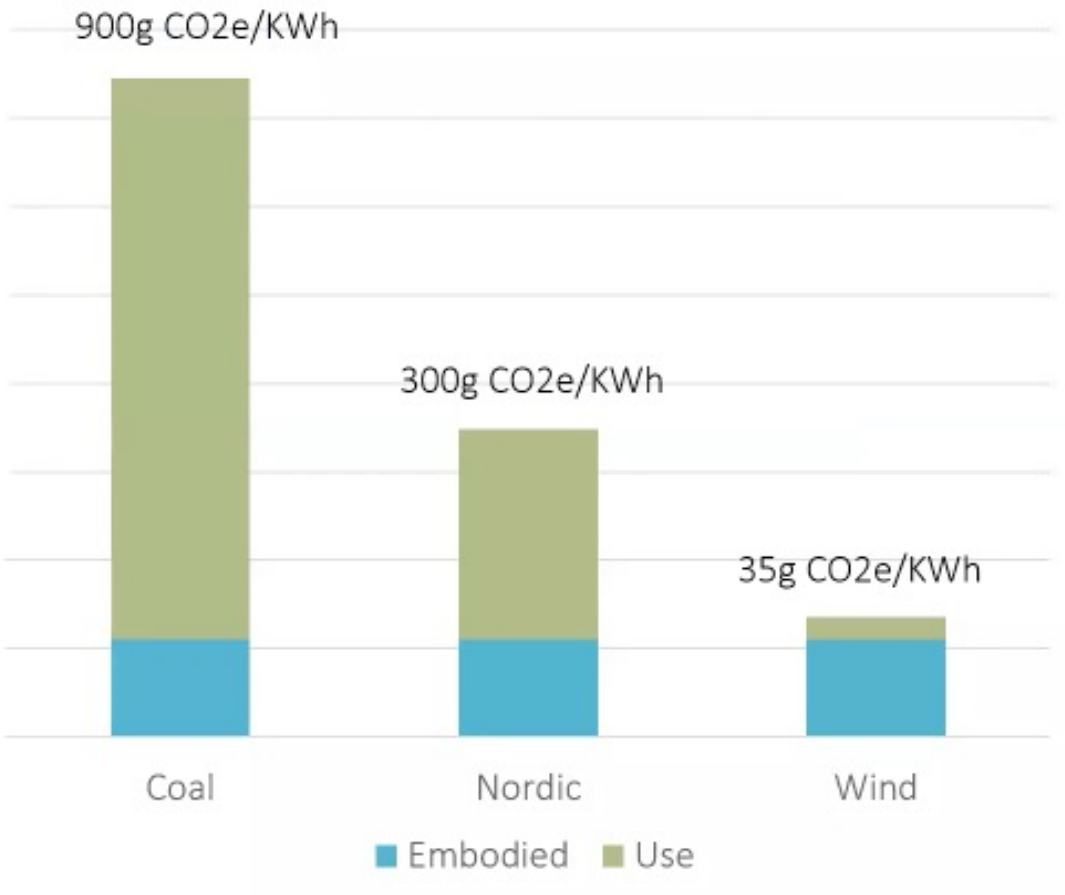
Apple 2018:

- 25.2M metric tons CO_2 emissions
- 74% manufacturing, 19% use
- ~1/3 of all CO_2 emissions from ICs

275 Million phones in US

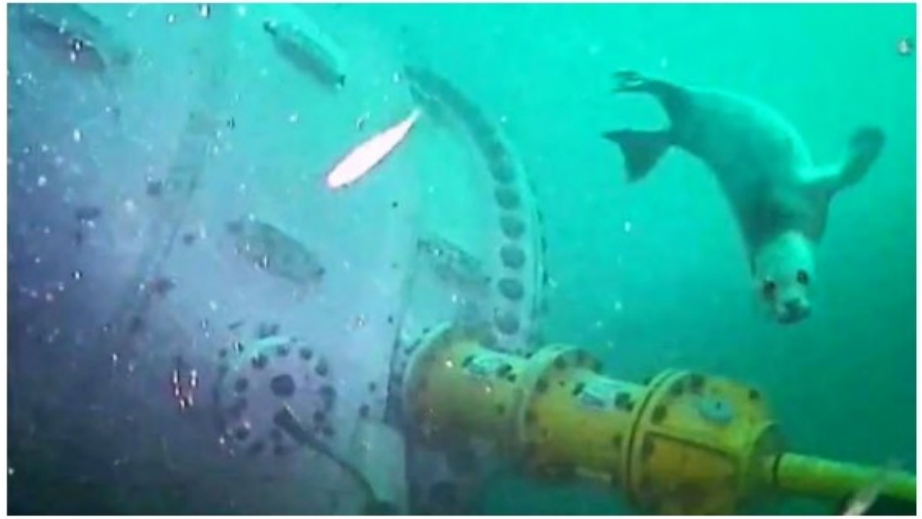
- Increasing use from 28 to 36 months would save 1M mTons CO_2 emissions annually

Embodied vs Use Carbon Emission



Microsoft's Project Natick

- Mini datacenter submerged in ocean
- 1/8th failure rate



Liquid Immersion

- Microsoft project Zissou
  - Liquid boiling point around 50 C
  - zero water consumption: doesn't cause water to evaporate in external cooling infra
  - Enables safe overclocking
  - Higher density → can lower embodied carbon