Warehouse-Scale Computers
datacenter pictures
datacenter pictures: servers racks

Facebook datacenter, Prineville, Oregon; via OregonLive
datacenter pictures: servers

- (6) N+2 Fans
- Optional Remote Heatsink for high wattage CPUs
- Dual 3Φ PSU with Battery
- DDR4 DIMMs
- Next Gen CPUs
- Up to (3) FHHL PCIe x16 Cards
- Universal Motherboard
- Up to (8) M.2 NVMe SSDs
- 50G Networking
- 945 mm
- 441 mm

image: Open Compute Project (proposed 2016)
datacenter pictures: cooling
Mechanical Penthouse

Air mixing section - Return air / outside air / filter corridor

Evaporative cooling / humidification corridor
Data Suite

- Hot aisle containment – ductless return
- Cold aisle pressurization – ductless supply
datacenter pictures: backup power
datacenter pictures: battery room

image: NOAA Center for Weather and Climate Prediction (at University of Maryland)
datacenter pictures: battery cabinet
datacenter pictures: TOR switch
**datacenter sizes**

tens to hundreds of megawatts

tens of thousands to hundreds of thousands of servers
Money, Money, Money

Case B Cost Distribution

- Server Power: 14.3%
- Server OPEX: 1.5%
- Server Interest: 3.8%
- Server Amortization: 29.5%
- PUE Overhead: 11.5%
- DC Amortization: 18.5%
- DC Interest: 10.3%
- DC OPEX: 10.6%
Kinds of cost

Operational:
- power — e.g. cheap hydroelectric
- maintainence — replacement equipment, etc.
- people — sysadmins

Capital
- buying/renting building + cooling + backup power
- buying servers and replacing them when they become outdated

Common metric — cost per Watt
Datacenter Applications

“the web”/interactive:
latency matters
reliability matters
“free” parallelism — independent (mostly) requests

“batch”:
throughput matters
use ‘spare’ capacity from interactive stuff
Varying demand

Urdaneta et al, “Wikipedia workload analysis for decentralized hosting”
Datacenter applications: consolidation/unpredictability

used

allocated

portion of CPU

portion of memory

0 5 10 15 20 25

time (days)
Datacenter versus Supercomputer

both *purpose-built*

different kinds of applications

datacenters tend to be more continuously upgraded
DC v SC: Goals

datacenter: focus on **cost-performance**

**scale-out**: more servers, not bigger machine

  bigger individual machines are less efficient per dollar
  want to use most mass-produced hardware

**consolidation** — run multiple applications together

more software modifications to use worse servers
DC v SC: Network

highly optimized datacenter network latency:

supercomputer network latency:
often less than ten microseconds round-trip
Datacenter Topology: historical

traditional datacenter topology:

image: Al Fares et al, “A Scalable, Commodity Data Center Network Architecture”
Datacenter Topology: four-post

image: Farrington and Andreyev, “Facebook’s data center network architecture”
Datacenter topology: Clos (1)

image: Singh et al, “Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google’s Datacenter Network”
Datacenter topology: Clos (2)

Aggregation Block (32x10G to 32 spine blocks)

image: Singh et al, “Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google’s Datacenter Network”
Datacenter Topology: Clos (3)

Scale-out

image: Greenberg, “SDN for the Cloud” (Microsoft)
DC v SC: Servers

very similar!

mass-produced, usually superscalar processors

usually high-power CPUs

... but not the most expensive
Server Balance

want to maximally use all server resources

balance CPU, memory, storage (disk or SSD)

depends on what applications you run
“wimpy” servers

another proposal: cheap, low-power servers at much higher density
DC v SC: Storage

storage on normal servers
  less networking required
  computations use local (fast) storage

separate storage racks
  flat storage hierarchy, more convenient to program
DC v SC: Reliability

supercomputer: usually more reliable/expensive components

supercomputer: failures — reboot it all

datacenter: expect failures

datacenter: failures — work around broken component
DRAM errors

uncorrectable: approx. 0.03% of servers per month

Meza et al, “Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of Trends in the field”
Hard Drive failures

AFR(%) over time for different drive populations:

- **Low**
- **Medium**
- **High**

**3-Month**
**6-Month**
**1 Year**
**2 Year**
**3 Year**
**4 Year**
**5 Year**

Pinheiro et al, “Failure Trends in a Large Disk Drive Population”
trading for software complexity

redundancy — handle failures
means having backup copies of everything

lots of applications per server — scheduling

slower network — compute close to data
energy efficiency

also a problem for supercomputers, etc.

but optimized much more heavily in the datacenter era
old datacenter efficiency

- IT Equipment: 50%
- Cooling: 25%
- Air Movement: 12%
- UPS/Power Distribution: 10%
- Ancillary: 3%
PUE — $\frac{\text{total power}}{\text{IT equipment power}}$

servers and networking equipment

modern large datacenter: $< 1.2$

before attention to this problem, PUEs of 2 or more were common
Achieving high non-IT efficiency

airflow — don’t mix hot/cold air

increased ambient temperature

cooling efficiency
  evaporative cooling
  better climates

power: increased electrical efficiency, e.g.:
  avoid AC/DC conversions
  distributed UPS
  get server power supplies that accept utility voltage
server efficiency

not especially well studied

similar losses from in-server power supplies, etc.

energy efficiency of components varies a lot
power-capping

underprovision cooling, power distribution, etc.

limit what runs on servers to stay under actual maximum
power proportionality problem (1)
power proportionality problem (2)
power proportionality problem (3)
power-saving modes (1)

what about “sleep” modes?
  save a lot of power
  take milliseconds to seconds to start/end

servers need to be available continuously (e.g. stored data)

10% utilized server might be doing some work in every second

not enough time to really save power
power-saving modes (2)

processors have lower frequency/voltage modes

problem: doesn’t save power in proportion to performance lost

problem: things other than processors use power
whack-a-mole in power usage

keep finding things which keep machine from sleeping for long times

keep finding components that use power continuously

tedious engineering problem
the datacenter for rent

public clouds — selling datacenter resources

e.g. Amazon Web Services

one way to deal with lower utilization
datacenter futures

started with: servers $\equiv$ desktop

trend now: beefier servers

(revisiting old ‘supercomputers’??)