Cloud Computing

Ed Lazowska
Bill & Melinda Gates Chair
Paul G. Allen School of Computer Science & Engineering
University of Washington

July 2017
Personal computing

- Web browser
- Office applications
- Math and science
- Databases and storage
- Email
Cloud email accessed through the browser
With the cloud provider’s domain name ...
Or with your own ...
Why not office applications too?
Why not everything else?
Consider ...

- Essentially infinite capacity
- You pay for exactly what you use (instantaneous expansion and contraction)
- No capital costs
- 1,000 processors for 1 day costs the same (or less) as 1 processor for 1,000 days (totally revolutionary!)
- 7x24x365 operations support, auxiliary power, redundant network connections, geographical diversity
- Someone else does backup; someone else handles software updates
- Sharing and collaboration are easy
- It continuously gets faster and less expensive (vs. purchased equipment) - AWS just instituted its 44th price cut!
Instantaneous expansion, effectively without limit

Animoto: EC2 Instance Usage

Credit: Werner Vogels, Amazon.com
Amazon Elastic Compute Cloud (EC2)

- Today: $0.04 per hour for a single monster computer
  - 4 cores of 2.4 GHz Intel Xeon E5-2676 v3 (Haswell) running Linux
  - 16 GB memory
  - EBS storage at 750 Mbps
This includes

- Purchase + replacement
- Housing
- Power
- Operations support
- Reliability
- Security
- Instantaneous expansion and contraction
Then, of course, there's what you can get for free!

Amazon EC2 Pricing

Free Tier*

As part of AWS's Free Usage Tier, new AWS customers can get started with Amazon EC2 for free. Upon sign-up, new AWS customers receive the following EC2 services each month for one year:

- 750 hours of EC2 running Linux/Unix Micro instance usage
- 750 hours of EC2 running Microsoft Windows Server Micro instance usage
- 750 hours of Elastic Load Balancing plus 15 GB data processing
- 30 GB of Amazon EBS Standard volume storage plus 2 million IOs and 1 GB snapshot storage
- 15 GB of bandwidth out aggregated across all AWS services
- 1 GB of Regional Data Transfer
A datacenter has 50 - 250 containers
A container has 1,000 - 2,000 servers
A server has two processors, 2 disks, tons of memory, battery backup
Processors are chosen for power efficiency, not performance
Isn’t this just timesharing?

- Many hundreds of machines are involved in a single Google search request (remember, the web is 400+TB)
  - There are multiple clusters (of thousands of computers each) all over the world
  - DNS routes your search to a nearby cluster
A cluster consists of Google Web Servers, Index Servers, Doc Servers, and various other servers (ads, spell checking, etc.)

These are cheap standalone computers, rack-mounted, connected by commodity networking gear.
Within the cluster, load-balancing routes your search to a lightly-loaded Google Web Server (GWS), which will coordinate the search and response.

The index is partitioned into “shards.” Each shard indexes a subset of the docs (web pages). Each shard is replicated, and can be searched by multiple computers - “index servers.”

The GWS routes your search to one index server associated with each shard, through another load-balancer.

When the dust has settled, the result is an ID for every doc satisfying your search, rank-ordered by relevance.
The docs, too, are partitioned into “shards” - the partitioning is a hash on the doc ID. Each shard contains the full text of a subset of the docs. Each shard can be searched by multiple computers - “doc servers”

The GWS sends appropriate doc IDs to one doc server associated with each relevant shard

When the dust has settled, the result is a URL, a title, and a summary for every relevant doc
Meanwhile, the ad server has done its thing, the spell checker has done its thing, etc.

The GWS builds an HTTP response to your search and ships it off.

Many hundreds of computers have enabled you to search 400+TB of web in ~100 ms.
Enormous volumes of data
Extreme parallelism
The cheapest imaginable components
  - Failures occur all the time
  - You couldn’t afford to prevent this in hardware
Software makes it
  - Fault-Tolerant
  - Highly Available
  - Recoverable
  - Consistent
  - Scalable
  - Predictable
  - Secure
The Paxos algorithm is central to building web-scale systems.

Paxos went unpublished and unused for a decade!