EC/CS 528: Cloud Computing

Overview of Cloud Computing

Instructor: Alan Liu
Announcements

• Piazza Discussion:
  • piazza.com/bu/fall2022/eccs528
  • Access code: cloudcomputing

• User account and skill survey:
  • Group and project matchings.
  • Prepare multiple project choices.
  • Q&A.

• New project update:
  • MLOps with Databricks in Public Clouds
How to start the project?

• Meet weekly with your mentor
  • Schedule a weekly meeting time
  • Record the meeting; mentors talk fast, being able to replay what they said can be super valuable

• Each person should say:
  • What have accomplished since last meeting?
  • What are you going to accomplish by next one?
  • Are they blocked?

• Don’t be blocked until weekly meeting:
  • Set up mechanism to ask quick questions to each other, and to mentor, e.g., slack

• Remember you are a team
Building the “cloud” from scratch - spec and buy
Then receive and assemble...

then you have to run it...
Issues?

[What do you think?]

- People and skills
  - $N$ areas of expertise $= O(N)$ people
- Scaling?
Why is Cloud Computing transformative?

• Major change in computation is managed and used:
  – Economics of central utility: Price of computers, Operational efficiency, Location (e.g., cheap power, distribution), Co-location other customers, Utilization shared capacity, shared services (e.g., DR)
  – “As with the factory-owned generators that dominated electricity production a century ago, today's private IT plants will be supplanted by large-scale, centralized utilities.” -- Nicholas Carr

• Availability of massive capacity on demand; elastically scale up and down:
  – Startups don’t need to be acquired by Google or MS: a startup won’t get money today to buy HW.
  – What happens when massive HPC becomes available to everyone?

• Gets rid of key impediments for developing & distributing SW
  – Avoids need for broad HCL, OS support, … many highly specialized software products…
Cloud in a nutshell

• On-demand access
• Economies of scale

All computing will move to the cloud
This is really nothing new…

Original vision of Utility/grid computing:

"If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility... The computer utility could become the basis of a new and important industry."

When was this statement from?

Why now?
Layers of Cloud

- **Infrastructure as a Service (IaaS):** AWS, Azure, OpenStack, MOC…
- **Platform as a Service (PaaS):** Salesforce’s Force.com, Google App engine, AWS, MSFT Azure
- **Software as a Service (SaaS):** Hosted applications: Gmail, Facebook, Google docs, eBay
Motivation for using cloud

- Cloud is not inexpensive today
  - 2-20x more expensive than local

- Administrators do not come in fractional units; if you are small cheaper

- Offers elasticity: can deal with massive fluctuations on demand

- Offers huge variety of services:
  - cloud provider can afford to amortize cost over a huge number of customers
Examples

• Microsoft’s Azure
• Amazon’s AWS
• Google’s Cloud Services
Remember this?

Host it R us.

Host 4 Less
AirBnB Example

• Success of market depends on network of renters and landlords;
  — starts really small
AirBnB

https://aws.amazon.com/solutions/case-studies/airbnb/

- 2010 – 24 EC2 instances, 300 GB of data
- 2015 – 1000 EC2 instances, 50 TBytes data

- Grew up entirely on AWS, no data center, no capital purchases, no racking/stacking, no acquisition networking…
  - 5-person operations team
  - Piggyback on AWS for external network, availability zones

- Rapid growth easily accommodated.
Coursera

• Massive on-line courses from Stanford, Duke…
• Went from 0 to 3.2 million users in first year
• Accessed from around the world
• Spikes common, e.g., 75% increase in load in 5 minutes
Example Architecture

Elastic Load Balancing

EC2
Autoscaling

RDS

Red Shift

Cloud Front

S3
Technology discussed

• EC2 & Elastic Load Balancing & EC2 Autoscaling – increase/decrease number of servers as needed.
• Relational Database Service (RDS) – managed service set up DB, patching, read-only replicas, across regions, backups automatically, snapshots
• Cloud Front – CDN, moved from 500 msec to 50msec average latency
• Red Shift – Data warehouse
Layers of data center

Hardware level:
How do you build cloud-scale systems?
Layers of data centers

“Operating system”: How do you manage and run cloud applications? What about file systems?
Frameworks:
How do you write a distributed application?
Above the file system:
How do you manage and work with structured data?
Layers

Networking:
How do the parts of a cloud-scale system talk to each other?
Top-down view of the course

- A computer
  - Tasks/processes
  - Scheduling Systems
  - File systems
  - High level computation frameworks
  - (Semi) Structured Data
  - Networking

- DC Building
  - Racks
  - Servers
  - Internal Network

- Berkely view of cloud computing
  - VMs, Containers, etc
  - VM/Container schedulers
  - Distributed File System

- Datacenter as a computer
  - Distributed Computation
  - Distributed Data Base & Streaming
  - Software Defined Networking
Top-down view of the course

<table>
<thead>
<tr>
<th>Task/Process</th>
<th>Scheduling Systems</th>
<th>File Systems</th>
<th>High level computation frameworks</th>
<th>(Semi) Structured Data</th>
<th>Networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC Building</td>
<td>Racks</td>
<td>Internal Network</td>
<td>VMs, Containers, etc</td>
<td>VM/Container schedulers</td>
<td>Distributed File System</td>
</tr>
<tr>
<td>Tasks/processes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Distributed Computation</td>
</tr>
<tr>
<td>Scheduling Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Distributed Data Base &amp; Streaming</td>
</tr>
<tr>
<td>File systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Software Defined Networking</td>
</tr>
</tbody>
</table>

Berkely view of cloud computing

- Datacenter as a computer
- Xen
- Container-OS

Xen Container - OS

Berkely view of cloud computing

A computer

Software Defined Networking

Networking

High level computation frameworks

(Semi) Structured Data

Networking

File systems

Distributed Data Base & Streaming

Distributed Computation

Distributed File System

VM/Container schedulers

VMs, Containers, etc

Internal Network

Servers

Racks

DC Building

A computer
Top-down view of the course

- A computer
- Tasks/processes
- Scheduling Systems
- File systems
- High level computation frameworks
- (Semi) Structured Data
- Networking

- DC Building
- Racks
- Servers
- Internal Network
- VMs, Containers, etc
- VM/Container schedulers
- Distributed File System
- Distributed Computation
- Distributed Data Base & Streaming
- Software Defined Networking

- Berkely view of cloud computing
- Datacenter as a computer
- MGHPC
- Xen
- Container-OS
- Google File Sys
- MapReduce
# Top-down view of the course

<table>
<thead>
<tr>
<th>A computer</th>
<th>DC Building</th>
</tr>
</thead>
<tbody>
<tr>
<td>Racks</td>
<td></td>
</tr>
<tr>
<td>Servers</td>
<td></td>
</tr>
<tr>
<td>Internal Network</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tasks/processes</th>
<th>VMs, Containers, etc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduling Systems</td>
<td>VM/Container schedulers</td>
</tr>
<tr>
<td>File systems</td>
<td>Distributed File System</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>High level computation frameworks</th>
<th>Distributed Computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Semi) Structured Data</td>
<td>Distributed Data Base &amp; Streaming</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Networking</th>
<th>Software Defined Networking</th>
</tr>
</thead>
</table>

- Berkely view of cloud computing
- Datacenter as a computer
- MGHPC
- Xen
- Container-OS
- OpenLambda
- Borg
- Berkeley serverless
- Google File Sys
- MapReduce
- Top-down view of the course
Top-down view of the course

- DC Building
- Racks
- Servers
- Internal Network
- VMs, Containers, etc
- VM/Container schedulers
- Distributed File System
- Distributed Computation
- Distributed Data Base & Streaming
- Software Defined Networking
- Tasks/processes
- Scheduling Systems
- File systems
- High level computation frameworks
- (Semi) Structured Data
- Networking

Berkely view of cloud computing

Datacenter as a computer

MGHPCC

Xen

Container-OS

OpenLambda

OpenShift

Borg

Mesos

Berkeley serverless

Google File Sys

MapReduce

Dapper
Top-down view of the course

- A computer
- Tasks/processes
- Scheduling Systems
- File systems
- High level computation frameworks
- (Semi) Structured Data
- Networking

DC Building
- Racks
- Servers
- Internal Network

VMs, Containers, etc
- VM/Container schedulers
- Distributed File System

Distributed Computation
- Distributed Data Base & Streaming
- Software Defined Networking

Datacenter as a computer
- Berkely view of cloud computing
- MGHPC

Xen
- Container-OS
- OpenLambda
- OpenShift

Borg
- Mesos
- Berkeley serverless

Google File Sys
- MapReduce

BigTable
- Dynamo

Dapper

MGHPC

OpenShift

Dapper
Top-down view of the course

1. A computer
   - DC Building
   - Racks
   - Servers
   - Internal Network

2. Tasks/processes
   - VMs, Containers, etc
   - VM/Container schedulers
   - Distributed File System

3. Scheduling Systems
   - Distributed Computation

4. File systems
   - High level computation frameworks
   - (Semi) Structured Data

5. Networking
   - Software Defined Networking

6. Berkely view of cloud computing
   - Datacenter as a computer
   - MGHPC

   - Xen
   - Container-OS
   - OpenLambda
   - OpenShift

   - Borg
   - Mesos
   - Berkeley serverless

   - Google File Sys

   - MapReduce
   - Spark
   - Hive

   - BigTable
   - Dynamo

   - Dapper
Top-down view of the course

- A computer
  - DC Building
  - Racks
  - Servers
  - Internal Network

- Tasks/processes
  - VMs, Containers, etc
  - VM/Container schedulers
  - Distributed File System

- Scheduling Systems
  - Distributed Computation

- File systems
  - Distributed Data Base & Streaming

- High level computation frameworks
  - Software Defined Networking

- (Semi) Structured Data

- Networking

- Berkely view of cloud computing
  - Datacenter as a computer
  - MGHPC

- Networking
  - Xen
  - Container-OS
  - OpenLambda
  - OpenShift

- Software Defined Networking
  - Borg
  - Mesos
  - Berkeley serverless
  - Google File Sys
  - Flat Datacenter
  - CEPH

- Distributed Data Base & Streaming
  - MapReduce
  - Spark

- (Semi) Structured Data
  - Hive

- Networking
  - BigTable
  - Dynamo

- Software Defined Networking
  - Kafka

- Berkely view of cloud computing
  - Dapper
Top-down view of the course

- A computer
  - DC Building
  - Racks
  - Servers
  - Internal Network
- Tasks/processes
  - VMs, Containers, etc
  - VM/Container schedulers
  - Distributed File System
- Scheduling Systems
- File systems
- High level computation frameworks
- (Semi) Structured Data
- Networking
- Software Defined Networking
- Berkely view of cloud computing
- Datacenter as a computer
- MGHPC
- OpenLambda
- OpenShift
- Xen
- Container-OS
- Borg
- Mesos
- Google File Sys
- Berkeley serverless
- Flat Datacenter
- CEPH
- MapReduce
- Spark
- Hive
- BigTable
- Dynamo
- Spanner
- Kafka
- Memoryv @ Facebook
- Dapper
Top-down view of the course

A computer
- DC Building
  - Racks
  - Servers
  - Internal Network
- Tasks/processes
- Scheduling Systems
- File systems
- High level computation frameworks
- (Semi) Structured Data
- Networking

VMs, Containers, etc
- VM/Container schedulers
- Distributed File System

Distributed Computation

Distributed Data Base & Streaming

Software Defined Networking

Berkely view of cloud computing
- Datacenter as a computer
  - MGHPC
- Xen
- Container-OS
- OpenLambda
- OpenShift
- Open Stack
- Borg
- Mesos
- Berkeley serverless
- Google File Sys
- Flat Datacenter
- CEPH
- MapReduce
- Spark
- Tensorflow
- Hive
- BigTable
- Spanner
- Dynamo
- Kafka
- Memoryv @ Facebook
- Dapper
Top-down view of the course

A computer

Tasks/processes

Scheduling Systems

File systems

High level computation frameworks

(Semi) Structured Data

Networking

DC Building

Racks

Servers

Internal Network

VMs, Containers, etc

VM/Container schedulers

Distributed File System

Distributed Computation

Distributed Data Base & Streaming

Software Defined Networking

Berkely view of cloud computing

Datacenter as a computer

MGHPCC

Xen

Container-OS

OpenLambda

OpenShift

Open Stack

Borg

Mesos

Berkeley serverless

Google File Sys

Flat Datacenter

CEPH

MapReduce

Spark

Tensorflow

Hive

BigTable

Dynamo

Kafka

Jupiter rising

Memoryv @ Facebook

Dapper
Transformation

- Transformed how SW is developed:
  - continuous deployment; changes tested with real customers
  - example Facebook failure last year
  - massive advantage over waterfall
- It’s all about distributed applications
  - change from pets to cattle
  - care about 99th% tail latency
  - stateless servers
  - huge set of higher level services: Containers as a Service, Functions as a Service, Analytics as a Service...
The challenges

• Monoculture from security perspective

• Emerging oligopoly:
  — Lack of competition limits sources innovation
  — Price is outrageously expensive

• Effort to lock in users: e.g., networking

• Big brother…, or perhaps just Giants whose incentives are not aligned with privacy and marketplace; Consider Facebook
The Datacenter as a Computer

An Introduction to the Design of Warehouse-Scale Machines – 2nd Edition

Luiz André Barroso, Jimmy Clidaras, Urs Hölzle
Traditional “wide-area” networks
Elements of data center
Storage assumptions

• Storage distributed across all machines
• Software like GFS distributes, versus NAS appliance
  — Redundancy even if rack level failure
  — Multiplex server resources (NIC/enclosure/power)
  — Exploits cheap desktop disks
• Typically network oversubscribed
  — E.g., 32 * 40Gig links nodes, 4 *100Gig Links up
Storage Hierarchy

One Server
- DRAM: 16 GB, 100 ns, 20 GB/s
- Disk: 2 T B, 10 ms, 200 MB/s
- Flash: 128 GB, 100 us, 1 GB/s

Local Rack (80 servers)
- DRAM: 1 TB, 300 us, 100 MB/s
- Disk: 160 TB, 11 ms, 100 MB/s
- Flash: 20 TB, 400 us, 100 MB/s

Cluster (30 racks)
- DRAM: 30 TB, 500 us, 10 MB/s
- Disk: 4.80 PB, 12 ms, 10 MB/s
- Flash: 600 TB, 600 us, 10 MB/s
Latency, bandwidth & capacity
Inside a data center
Self-introduction

Q&A