Cloud and Big Data

Course Objective

- **Graduate level course on Cloud Computing**
  - Focus is on learning and building extremely large scale systems and applications leveraging Cloud
    - Building blocks and design patterns in designing backend of typical Internet Scale application
  - Learn concepts as well as hands-on experience by using real cloud and cloud technologies.
  - Three key objectives: le oud to build applications, build scalable intelligent syst

- **We shall learn cloud technologies by usi s and services - Amazon AWS, Google Cloud, Hadoop/Spark, Kafka, Elastic, Dynamo etc.**
- **Required background**
  - Programming experience with one of the following Java/Python, web services basics
  - Operating Systems concepts, networking concepts would help you understand more

--> If you are not familiar with web services, take a look at materials on any web application design technologies.

What would you learn in this course…

- HowTo
  - How to use a Cloud as a compute node?
  - How to use cloud to design an Internet scale application?
  - How to process a very large amount of data?
  - How to build your own cloud using open source?
- Concepts: Building Bloc
  - Virtualization, Containe
  - Peta-byte scale storage systems
  - Event and messaging systems (Kafka)
  - noSQL datastore (Cassandra, mongo, Dyna
  - Elastic Search
  - Compute in a cluster
  - Intelligent AI applications
  - …
- Case studies with real systems/cloud
- Compute Cloud, Storage Cloud, Data Cloud

Main Modules

- **Cloud Platform and Programming**
  - Basic cloud concepts
  - Hands-on experience with Amazon AWS Cloud
  - Virtualization as an enabling technology
  - Virtualization vs Containers vs Serverless
  - ***Build a Web application leveraging cloud***

- **Building Blocks in an Extremely Large Scale Application**
  - Scalable data store and noSQL database
  - Message Queues: Kafka
  - Unstructured data and queries: Elastic Search
  - In-memory data store
  - devOps: Containers, micro-
  - ***Build a scalable applicati***

- **Private Cloud**
  - Understand key concepts for building a cloud
  - Use Openstack cloud management stack
  - devops/chef/puppet for private cloud automation
  - ***Build your own cloud***

- **Big Data Computing Platform and Programming**
  - Hadoop eco-system, and batch data processing & storage
  - MapReduce, Hive, Hbase
  - *Spark and Spark Streams*
  - ***Intelligent Real-time system design using Spark***

Tentative Syllabus/Lectures

- Intro to Cloud: IaaS, PaaS, SaaS cloud, AWS, GCP, Azure Cloud (but we focus on building using AWS)
- Designing a web application using cloud
- Virtualization as Cloud Enabling technology; Virtualization vs Containers
- Building Private Cloud (OpenStack)
- DevOps in a Cloud and
- Designing Extremely La
  - Message Queue (Kafka)
  - Event Notification
  - Scalable no-SQL
  - Lambda architecture
  - Indexing and searching unstructured data (Elastic Search)
- Computing in a Cluster
  - Hadoop/MR
  - Spark based compute model
- Use cases: Designing Intelligent Services in a Cloud - we will use a variety of AWS ML and Google ML APIs to design interesting use case

Tentative Course Schedule

Date	Topic	Reading List
09/08	Intro to Cloud	
09/15	Cloud Programming	GFS
09/22	Designing Scalable Web Application	BigTable
09/29 [A1]	Designing Web Scale Applications	Kafka
10/06	Message Queues and Logs	Cassandra, DynamoDB
10/13	noSQL da	duce
10/20 Quiz1 [A2]	Container	
10/27	Cluster Computer: Spark	
11/03	Spark Data Frames	
11/10 [A3]	Spark Advanced	spanner
11/17	Private Cloud	
11/24	Intelligent Systems	
12/01 Quiz 2	Advanced Topics	
12/08	Advanced Topics	
12/15	Final Demo	

Course Material

- **Lecture Notes**
  - Each lecture will have a theme topic. Lecture slides will be provided for each lecture. Additional reference materials will be specified.

- **Reading List**
  - A set of landmark papers in the area of large scale systems
  - You submit a paper su uestions.

- **Three programming Assignments**
- **A final Course project**
- **Reference Texts**
  - AWS in Action
  - Elastic Search in Action
  - Kafka Definitive Guide
  - Hadoop: The Definitive Guide
  - Learning Spark

Grading and requirements

- 2 Quizzes -- 25%
- Assignments – 35% grade
  - 3 homework stressed on technologies and programming

- Course project -- 40% grade
  - Students may team up

- Submission process – everything to be don works and Github

Project: Learn how to innovate in this space

- **Objective is to learn how to innovate in this space**
- **Four phases to your project**
  1. Concept and business idea
  2. Technology viability and architecture
  3. Execution planning and prototyping
  4. Demo, socialization and review
- **Few suggestion**
  - Don't procrastinate – you get A+ (and earn millions!)
  - Form your team carefully – asking, interview ates. Float around some ideas,, kick the tire. Take a look at lot of rec are bought by Google, Apple, FB, Amazon etc. **Take a look at beta.list**
  - Cloud + Social + Mobile is a good recipe for a perfect storm

What you need to do soon

- Get account on few popular clouds
  - Amazon AWS (EC2, S3)
  - Google Cloud Platform, Google Storage
  - We are working with Amazon to get free accounts

- Course Project
  - Substantial portion of y roject
  - I will provide a set of project categories that y from or come up with your own. But each project category will have a se eed to be demonstrated
  - You need to have a team and a project propo 5:00pm

What is Cloud?

- Allows users to request computing/storage resources through web interfaces
- You do not need to own or install or manage these resources.
- Pay as you go - Resources on-demand
- Elastic: Use as much as you want or as less as you want
  - Users can assume infin resources are available.
  - Users can request reso release/remove resources when they don't need.
- Compute and storage resources are now tr are entities. You get access to such resources programmatically cal hardware anymore!
- So what are the Clouds! Where are the Cloud?

- Read this paper: http://cacm.acm.org/magazines/2010/4/81493-a-view-of-cloud-computing/fulltext

Why Cloud?

- You can get as many as 1000 machines for an hour for a few dollars to run a complex application!
- You don't need to manage, maintain or fix any machines!
- You can use as little as 1 machine or as many as 10000 machines depending on what your current needs are!
- Two key focus: on-dem

Essential Characteristics

- *On-demand self-service.* A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
- *Broad network access.* Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
- *Resource pooling.* The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer deman ence in that the customer generally has no control or knowledge ov curces but may be able to specify location at a higher level of abs ier). Examples of resources include storage, processing, memory, network bandwidth, and v
- *Rapid elasticity.* Capabilities can be rapidly and elastical me cases automatically, to quickly scale out and rapidly released to quickly scale in , the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
- *Measured Service.* Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

Service Models

- *Cloud Software as a Service (SaaS).* The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.
- *Cloud Platform as a Servic* onto the cloud infrastructur programming languages and tools supported by t manage or control the underlying cloud infrastruc systems, or storage, but has control over the dep hosting environment configurations. the consumer is to deploy plications created using consumer does not twork, servers, operating s and possibly application
- *Cloud Infrastructure as a Service (IaaS).* The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

Deployment Models

- *Private cloud.* The cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise.
- *Community cloud.* The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise.
- *Public cloud.* The cloud infr general public or a large industry group and is owne ervices.
- *Hybrid cloud*. The cloud infrastructure is a comco ore clouds (private, community, or public) that remain unique entities gether by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).

Berkeley View of Cloud Definition

- IaaS → SaaS Provider -→ SaaS User

Source: Above the Clouds: A Berkeley View of Cloud Computing

Different types of utility model

- IaaS Cloud (Amazon EC2)
  - Low level of computing resource abstraction
  - Provides a (virtual) machine to users
  - Makes it hard for IaaS providers to support automatic scaling, failover etc.
- Google AppEngine
  - Targeted at web applic
  - Enforces an application
  - Clean separation betwe storage tier
  - Benefit: makes it possible to handle auto-scali over/high availability
- Microsoft Azure
  - Applications need to be written using .NET libraries
  - More flexible than Google AppEngine
  - Able to provide some automated scaling
  - Between Application framework and hardware virtual machines

Different Cloud Offerings: A Layered Perspective

- Higher the stack, less control but more automation for user
- Lower the stack, more control but more responsibility for user

Example Clouds and Usage Scenario

- IaaS
  - Amazon EC2, Rackspace
- PaaS
  - Google AppEngine
  - Microsoft Azure
- SaaS
  - salesforce.com
- Roll your own
  - Open Source software stack
    - Open Nebula
    - Eucalyptus
    - Openstack

- Machine level abstraction
  - User requests a machine with desired CPU, mem, disk possibly with a preconfigured OS and software
  - IaaS Cloud provides a virtual server with (minimal) pre-installed software such as OS
- Platform level abstraction
  - User writes application using PaaS defined interfaces atform to support the deployment and his application
- User installs and adapts to build own Cloud

Cloud Computing Economics

- Three useful usage scenarios
  - Load varying with time
  - Demand unknown in advance
  - Batch analytics that can benefit from huge number of resources for a short time duration

- Why pay-as-you-go model makes sense economically even if cost is higher than buying a server and depreciating the h/w
  - Extreme elasticity
  - Transference of risk (of

Source: Above the Clouds: A Berkeley View of Cloud Computing

Top obstacles and opportunities for Cloud

Source: Above the Clouds: A Berkeley View of Cloud Computing

IaaS Cloud Example: Amazon EC2

- Amazon EC2 provides public IaaS Cloud
- User uses a portal to request a machine with specific resource
  - CPU, memory, disk space
  - Pre-built OS and possibly middleware

PaaS Cloud: Google App Engine

- PaaS model
- Provides a platform to host web applications
- App Engine SDK for programming (Python and Java support)
- A set of primitives (datastore, URL fetch, memcache, JavaMail, Images, authentication..)
- User focuses on developing th
- Once deployed, scaling, availa ne platform

Let's use a IaaS Cloud (Amazon EC2)

- http://aws.amazon.com/console/
- Amazon EC2 console based provisioning demo

Traditional vs Cloud-based Application

Various Methods to Access AWS

Amazon AWS console (EC2 view)

- User logs in with AWS credentials

User launches request instance → a list of prebuilt stack is provided

- AWS shows a list of available pre-built base software stack (**called Virtual Appliances**) user may request to add to the machine

User can choose the resource size (CPU, mem choices)

- Instance request wizard guides through resource choices

User specifies security/access configurations

AWS provisions an instance and returns user credentials

Next Week

- Account setup and testing
  - Sign up for AWS account. Sign up for AWS EC2 and S3 services.
  - Create a micro instance with Amazon Linux stack with appropriate keys and access control using AWS portal. SSH into the instance you created.
  - Read Chapter 1 and 3 f
- Assignment 0
  - Building Modern Web Application (Just compl y following this link: https://aws.amazon.com/getting-started/hand rn-app-fargate-lambda-dynamodb-python/

Some additional links

- Hands-on Tutorials on AWS: https://aws.amazon.com/getting-started/hands-on/
- https://aws.amazon.com/solutions/case-studies/
- http://aws.amazon.com/awscredits