Cloud Computing – An Introduction

Dr. Basem Suleiman

Web Application Engineering
School of Computer Science and Engineering
UNSW
The Internet - Web Applications

https://support.office.com/en-us/
The Internet – Web Applications

• Web Applications drive the Internet
  – Email Applications
  – Social Media/Network
  – E-commerce Applications
  – Web and data Services
  – Business processes

• What makes such applications functional?
  – Hardware (Servers, Networks, Storage), Application and System Software, Databases system and software, etc.

Sources: https://www.tipsandtricks-hq.com/ecomerce/tag/buy-button-image
Web Hosting – Traditional Models

• Collection of servers and computing devices that are networked together and co-located into a single facility
• Servers can be configured and set up with appropriate systems and application software
• Major online companies have their own data centres, Google, eBay, Amazon
• An enterprise cloud have private data center

http://www.zdnet.com/i/story/60/98/011892/ebay-topaz-room1.jpg
Cloud Computing – Enabling Technologies

Cloud computing is the result of the evolution and adoption of existing technologies and paradigms

• **Virtualization**
  • A software that separates a physical computing device into one or more virtual devices, each of which can be easily used and managed to perform computing tasks

• **Autonomic computing**
  • Automation of the process through which a user can provision resources on-demand
  • Minimal user involvement, the automated process reduces costs and potential human errors

• **Service-Oriented Computing**
  • All resources in cloud computing model are provided as services
  • Use of the well-established standards and best practices gained in the domain of SOA to allow global and easy access to cloud services in a standardized way
So, What’s Cloud Computing?

“Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

(National Institute of Standards and Technology (NIST), USA).

• Note: there are many definitions and classification of cloud computing, it’s characteristics and models
• NIST definition and models are widely adopted in research and industry
Cloud Computing – Essential Characteristics

On-demand self-service
• Computing capabilities, such as server and network storage, are provisioned as needed automatically without requiring human interaction with each service 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

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 demand

NIST definition of cloud computing http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
Cloud Computing – Essential Characteristics

Rapid elasticity
• Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand
• To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated 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)

NIST definition of cloud computing http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
Cloud Computing – Service Models

Cloud Consumers / Clients

Software as a Service (SaaS)

Platform as a Service (PaaS)

Infrastructure as a Service (IaaS)

Source: https://commons.wikimedia.org/w/index.php?curid=18327835
Infrastructure as a Service (IaaS)

- Obtain basic computing resources such as processing, storage, networking, and associated software as services

- Enabled by virtualisation - running fully-functional independent virtual instances on top of the actual physical infrastructure
  
  - Processing - Virtual machines - VMWare, Xen, etc.
  - Storage - Virtual Hard Disk over Distributed File Systems - NFS, GFS, Storage Pools, etc.

- Examples: Amazon, Rackspace, GoGrid (public and private cloud resources)
Amazon Web Services

- Amazon Elastic Compute Cloud (EC2) - Computing (VMs)
- Amazon Elastic Block Store (EBS) - Persistent block-level storage for EC2 VMs
- Amazon RDS - Cloud-based RDBMS service
- Amazon Simple Storage Service (S3) - Storage Volumes
- Amazon CloudFront - Content delivery network that uses S3 to store objects in edge locations.
- Amazon Elastic LoadBalancer, CloudWatch
- Amazon DynamoDB - NoSQL database
- And many more ...
Platform as a Service (PaaS)

- Allow users to develop and deploy applications on to a vendor's platform using specific API/libraries that abstract the actual process.
- Develop application in cloud-based environment.
- Hardware details are hidden from the user - vendor takes care of the infrastructure (could be IaaS).
- User has control over the hosting configurations and how much application-level resources (objects, sessions, database tables) to consume.
- Pricing is also dependent on the application size.
- Examples: Google App Engine, Microsoft Azure Services, Heroku, etc.
Platform as a Service (PaaS)

• Allow users to develop and deploy applications on to a vendor's platform using specific API/libraries that abstract the actual process
• Develop application in cloud-based environment
• Hardware details are hidden from the user - vendor takes care of the infrastructure (could be IaaS)
• User has control over the hosting configurations and how much application-level resources (objects, sessions, database tables) to consume
• Pricing is also dependent on the application size
• Examples: Google App Engine, Microsoft Azure Services, Heroku, etc.
Software as a Service (SaaS)

- Full-featured applications are delivered to the customer over the Internet
- No need to install the application locally
- Subscription model instead of upfront license and install model
- Some developer use APIs for integrating SaaS apps into internal apps
- Examples: Salesforce.com (CRM), Zoho (Collaboration),
Cloud Computing – Deployment Models

**Private Cloud:** The cloud infrastructure is owned by a service provider who makes it available to the general public for commercial purposes.

**Public Cloud:** The cloud infrastructure is owned by an organization which operates it only for its internal use.

**Hybrid Cloud:** Combination of private and public clouds. E.g. resources sourced from public clouds whenever demand exceeds the capacity of a private cloud (cloud bursting)

NIST definition of cloud computing http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

https://commons.wikimedia.org/w/index.php?curid=6089457
Cloud Computing – Business Models

### Table 3: Classification of pricing models and cloud server offering types

<table>
<thead>
<tr>
<th>Pricing model</th>
<th>Offering type</th>
<th>Commitment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per-use</td>
<td>On-demand servers ($ per hour use)</td>
<td>Nil</td>
</tr>
<tr>
<td></td>
<td><em>Examples:</em> Amazon on-demand and spot instances, Rackspace cloud servers,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Terremark vCloud (per-hour)</td>
<td></td>
</tr>
<tr>
<td>Subscription</td>
<td>Dedicated servers (upfront $ per time period)</td>
<td>Short-term (less than 6 months) and</td>
</tr>
<tr>
<td></td>
<td><em>Examples:</em> GoGrid dedicated servers (monthly), Joyent Smart-Machines (monthly),</td>
<td>Long-term (1–3 years)</td>
</tr>
<tr>
<td></td>
<td>Rackspace servers (monthly)</td>
<td></td>
</tr>
<tr>
<td>Prepaid per-use</td>
<td>On-demand servers ($ per hour use deducted from prepaid credit)</td>
<td>Nil</td>
</tr>
<tr>
<td></td>
<td><em>Examples:</em> ElasticHosts hourly-burst cloud servers, GoGrid cloud servers (hourly),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Joyent SmartMachines (daily)</td>
<td></td>
</tr>
<tr>
<td>Subscription+per-use</td>
<td>Dedicated servers (upfront $ per month/year) + on-demand instances $ per hour use</td>
<td>Short-term (less than 6 months) and Long-term (1–3 years)</td>
</tr>
<tr>
<td></td>
<td><em>Examples:</em> ElasticHosts monthly cloud servers + hourly usage, Joyent monthly</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SmartMachines + daily usage, Amazon reserved instances (1 or 3 years)</td>
<td></td>
</tr>
</tbody>
</table>

On understanding the economics and Elasticity Challenges of deploying business applications on public cloud infrastructure

Pricing Models of AWS cloud services
Data-Intensive Application

- In 2007, The New York Times decided to make all public domain articles from 1851 - 1922 available free of charge
- 11 million articles from 1885 - 1980 - each of which is composed of TIFF images that have to be combined – hugely compute and data-intensive
- Solution - Use Amazon S3 to store the article data (4 TB) and EC2 machines to generate the PDFs which were saved back to S3 from where they are served
- Use Hadoop (open-source Map-Reduce implementation) for programming

Cloud Computing - Case Study

Data-Intensive Application

• In 2007, The New York Times decided to make all public domain articles from 1851 - 1922 available free of charge

• 11 million articles from 1885 - 1980 - each of which is composed of TIFF images that have to be combined – hugely compute and data-intensive

• Solution - Use Amazon S3 to store the article data (4 TB) and EC2 machines to generate the PDFs which were saved back to S3 from where they are served

• Use Hadoop (open-source Map-Reduce implementation) for programming

• 100 EC2 instances + Hadoop + 24 hours = Job Done!

• There are many high-performance data-intensive applications being deployed on cloud now - bioinformatics, data mining, image and video processing

Elasticity – Cloud Computing
Elasticity - Hosting Vs. IaaS Cloud

- On-demand computing resources – e.g., servers, storage
- Efficient use of resources – pay per usage time (pay-as-you-go)

What is Elasticity (Auto-Scaling)?

- The ability of a system to dynamically adapt its underlying computing infrastructure resources in response to variable workload changes over time

- Can be at SaaS, PaaS, IaaS

- IaaS Elasticity
  - Adding/removing virtual or physical servers
  - Increasing/decreasing CPU, memory and storage capacity by adding/removing additional hardware components to existing machines
  - Increasing/decreasing network speed and number of IP addresses
  - Increasing/decreasing amount of data transfer and number of data operations/requests of cloud resources

- Manual (user interface) vs. automated means (APIs)
  - Auto-scaling
IaaS Elasticity and Web Applications

- Key characteristics of Web applications
  - Highly transactional - business value
  - Commodity - Variable workload patterns

- IaaS cloud
  - On-demand computing resources/services – pay-as-you-go
Case Study – Animoto AWS Elasticity

• Animoto an online video service, makes it easy to make and share videos in just a few minutes

• The company launched in 2007 using its own servers, but moved to AWS for additional capacity

• When Animoto integrated with Facebook in 2008, attracting 750,000 new users in 3 days, it used AWS to handle the load.

Animoto's Facebook scaling story on Amazon's Elastic Compute Cloud
Cloud-based & Application-based SLA*

Types of Scaling – Scaling Architecture

(a) Horizontal Scaling

(b) Vertical Scaling
Horizontal vs. Vertical Scaling

• Horizontal (Scale-out and Scale-in)
  – More computing resources (e.g., servers)
  – Reliable – fail-over scenario
  – Fully automated
  – Growing management complexity

• Vertical (Scale-up and Scale-down)
  – More powerful computing resources – bigger servers
  – Single point of failure
  – Human intervention
  – Reasonable management overhead
# Horizontal vs. Vertical Scaling – Trade-offs

<table>
<thead>
<tr>
<th>Scaling strategy</th>
<th>Server scaling costs</th>
<th>Monitoring costs</th>
<th>Application’s availability and reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal scaling</td>
<td>24 hrs/w × $0.085/hr × 6 servers × 52 weeks = $636.48/yr</td>
<td>Defining and configuring 7 metrics for 6 servers Costs: $3.5 per server/mo × 6 servers × 12 months = $252</td>
<td>Highly available—no single point of failure Highly likely reliable—quick recovery time</td>
</tr>
<tr>
<td>Vertical scaling</td>
<td>24 hrs/w × $0.68 × 1 server × 52 weeks = $848.64</td>
<td>Defining and configuring 7 metrics for 1 server Costs: $3.5 per server/mo × 1 server × 12 months = $42</td>
<td>Low availability—single point of failure Highly likely unreliable—long recovery time</td>
</tr>
<tr>
<td>Hybrid scaling</td>
<td>(24 hrs/w × $0.085/hr × 3 servers × 52 weeks) + (24 hrs/w × $0.34 × 1 server × 52 weeks) = $742.48</td>
<td>Defining and configuring 7 metrics for 4 servers Costs: $3.5 per server/mo × 4 servers × 12 months = $168</td>
<td>Improved availability—no single point of failure Improved reliability—medium recovery time</td>
</tr>
</tbody>
</table>

*Suleiman et. al. On Understanding The Economics and Elasticity Challenges of Deploying Business Applications On Public Cloud Infrastructure, JISA 2011*
IaaS Elasticity (Auto-Scaling) Services

• AWS Elastic Load Balancing

• GoGrid’s Infrastructure and RAM Scaling

• Rackspace Cloud Monitoring and AWS CloudWatch

• AWS and Rackspace Auto Scaling

• RightScale cloud management platform
## Elasticity Rules - Examples

<table>
<thead>
<tr>
<th>IaaS Service</th>
<th>Auto Scaling Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Web Services (AWS)</td>
<td>If the application latency seen by the load balancer is greater than 800 millisecond for 9 minutes continuously, scale out the application tier</td>
</tr>
<tr>
<td>Rackspace Auto Scale</td>
<td>If average memory utilization across all web servers is greater than 85%, then scale out the web servers</td>
</tr>
</tbody>
</table>
Elasticity (Auto-Scaling) Rules

• Rule-based mechanism

  – Monitor certain resources/application metrics

  – Determine when to trigger adding releasing computing resources

  – Determine how much computing resources to add/release

  – Choose appropriate values for the core thresholds and parameters
Structure of Elasticity Rules

Monitor `<MetricName>` every `<T>`  

```
// Evaluate Elasticity Condition (EC) every `<T_i>`

IF `<MetricName>` {Comparison Operator} `<M_θ>` FOR `<T_w>`
```

```
// Execute Elasticity Action (EA) when EC is satisfied

Change `<ResourceName>` by `<P>`

Wait for `<T_c>`
```
Auto-scaling Rules – Example

Monitor CPU Utilization (CPUUtil) every 1 min. interval

IF CPUUtil > 80% FOR 7 minutes
  Add 1 server of small capacity //Scale-out
  Wait 5 consecutive 1 min. intervals

IF CPUUtil < 30% FOR 10 minutes
  Remove 1 server of small capacity //Scale-in
  Wait 7 consecutive 1 min. interval
AWS Elasticity (Auto-Scaling)

- Elastic Load Balancer
- Auto-scaling group
- Server Config. Image
- Associated with
- Auto-scaling rules
  - Scale-in
  - Scale-out
- Triggers
- Metric Alarm
- Metrics

Cloud Computing, Dr. Basem Suleiman
Guest Lecture 2
Architecting for the Cloud
Dr. Adenen Guabtni