Lecture 10a: Cloud Computing

1. What is Cloud Computing?
2. X as a Service
3. Key Challenges
4. Developing for the Cloud
WHAT IS CLOUD COMPUTING?

A style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet. (Wikipedia)
Why is it called *Cloud*?

- services provided on virtualised resources
- virtual machines spawned on demand
- location of services no longer certain
- similar to *network cloud*
Flavours of Cloud Computing:

http://www.mazikglobal.com/blog/cloud-computing-stack-saas-paas-iaas/
WHAT IS CLOUD COMPUTING?
KEY CHARACTERISTICS OF CLOUD COMPUTING

SP 800-145. The NIST Definition of Cloud Computing:

① On-demand, self-service
   - get resources (CPU, storage, bandwidth etc),
   - automated: as needed, right now!

② Network access
   - services accessible over the network, standard protocols

③ Pooled resources
   - provider: multi-tenant pool of resources
   - dynamically assigned and reassigned per customer demand

④ Elasticity
   - Scalability: rapidly adjust resource usage as needed

⑤ Measured service
   - monitor resource usage
   - billing for resources used
BENEFITS

Flexibility:
- Flexible provisioning
- Add machines on demand
- Add storage on demand

Effort:
- Low barrier to entry
- Initial effort: no need to spec and set up physical infrastructure
- Continuing effort: no need to maintain physical infrastructure
Cost:

- Low initial capital expenditure
- Avoid costs of over-provisioning for scalability
- Pay for what you use

in “Developing and Extending Applications for Windows Azure with Visual Studio”
Reliability:

- Redundancy
- Trust reliability of provider
- Data backups
- What happens when provider goes down?
- What about Security? Privacy?
Public vs Private Clouds?

Public: open services available to everyone

Private: owned, operated, and available to specific organisation

Is this still cloud computing?

Hybrid: system uses some private cloud services and some public cloud services.

http://blog.nskinc.com/IT-Services-Boston/bid/32590/Private-Cloud-or-Public-Cloud
INFRASTRUCTURE AS A SERVICE: IaaS

Service provider provides:

➔ Server and network hardware
➔ Virtual machines
➔ IP addresses
➔ Services to manage VMs (create, start, stop, migrate)
➔ Optional: storage, database, synchronisation, communication

Client provides:

➔ OS and OS environment
➔ Web server, DBMS, etc.
➔ Middleware
➔ Application software
Challenges – Client:

- Transparency (naming, redirection)
- Scalability: replication and load balancing decisions
- Synchronisation and coordination
- Security
- Fault tolerance
- Software maintenance and sys admin

Challenges – Provider:

- Hardware provisioning and maintenance
- Load management
- IP address management, DNS management
- Infrastructure fault tolerance
- Monitoring, logging, billing
- Storage
EXAMPLE 1: AMAZON WEB SERVICES (AWS)

- Elastic Compute Cloud (EC2)
- Simple Storage Solution (S3)
- Simple DB
- Simple Queue Service

http://vmtoday.com/2013/07/introduction-to-amazon-web-services-aws/
Elastic Compute Cloud (EC2):

- Instances: virtual cores, memory, storage
  - instance types (cpu, memory, net, storage options):
    - t, m, c, p, g, x, r, i, d
    - micro, small, medium, large, xlarge, ...

- Cost:
  - free tier: limited instances, free CPU hours
  - on-demand: $0.007 - $39 per hour
  - reserved: 1-3 years, discounted, fixed cost

- Launch Amazon Machine Image (AMI) on instances
- Preconfigured or custom images
Using EC2
1. Grab your credit card and create an account (10 min). Open the EC2 Dashboard.

2. Select where you want to create your virtual machine (called ‘instance’)

3. Hit this button!
4. Select a machine image (called AMI) to use
   - Various OS (Major distros, Ubuntu, Windows, ...)
     - Some from official providers, some from 3rd parties
   - Many pre-configured images
     - E.g. with Apache + MySQL, Windows + SQL Server, etc.
   - You can ‘save’ your instance to create your own AMI
5. Determine the amount of resources to allocate. Price varies, e.g:
- t2.micro: USD 0.0146/hour (Linux) USD 0.0192/hour (Win)
- t2.medium: USD 0.0584/hour (Linux) USD 0.0764/hour (Win)
- m5.large: USD 0.12/hour (Linux) USD 0.212/hour (Win)
Additional costs for other software (e.g. SQL Server)
6. Done! (<5 minutes in total)
   - Set SSH key, configure firewall, etc.
   - Each machine has a randomly assigned public IP address and DNS name, e.g.:
     `ec2-54-252-240-203.ap-southeast-2.compute.amazonaws.com`
SSH from a desktop

7. Connect to the new virtual machine
   - Just SSH to the address
   - Use appropriate username and keypair
   - You have root or sudo access

You’re in Amazon’s Datacenter in Sydney!
If you need Windows, launch a Windows instance and remote-desktop to it.

You’re in Amazon’s Datacenter in Sydney!
8. Terminate (decommission) or stop (shutdown/hibernate) instances when they are not in use
   - Instances cost you by time – not by actual resource usage
   - Consider using a script to stop instances at a convenient time (say midnight)
   - Restart instances manually when you next need them.
### 9. Check the cost in near real-time
- Hours to run virtual machines
- Network in/out
- VPN
- Disk access
- # of request made to services
- ...
Regions and Availability Zones:

- 99.95% availability per service region
- Regions: geographically dispersed, independent
- Availability zones: contained in Regions
- Availability zones: isolated from failures in other zones, but connected

Elastic IP addresses:

- IP address associated with account
- Dynamic remapping to specific instances
  - instance has *private IP address* and *public IP address*
  - *Elastic IP* can be mapped (and re-mapped) to private IP

Elastic Load Balancing:

- Distributes traffic across instances
- Monitors ‘health’ of instances: customisable
- Routes to healthy instances
Auto Scaling:

- Automatically start or stop new instances
- User-defined conditions
  - manual (minimum group size), schedule
  - instance health, CloudWatch input

https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html
Security:

- **Infrastructure Security**
  - Data centre physical security
  - Software and hardware maintenance
  - Monitoring and Testing (automatic and manual)

- **Application Security**
  - API access control (access keys)
  - Firewall settings for instances (security groups)
  - Virtual Private Cloud (VPC): private or public subnetworks
  - Encrypted storage support
  - Logging
STORAGE

Elastic Block Store:

- Network Attached Storage (NAS) (servers with disks)
- Block level storage volumes
- Mounted as block device (e.g. disk) on an instance
- Physical Servers and Disks shared by customers (no caching, competing for disk and net IO)
- Replicated in Availability zone
- Cost: per GB/per month
Simple Storage Service (S3):

- Buckets: store objects
  - Can be placed in specific regions
- Objects: data and metadata
  - metadata: key-value pairs describing the object
  - identified by key (unique within a bucket)
  - versioned
- Consistency:
  - highly replicated
  - eventual consistency, no locking
  - atomic object update
- Access control
Snapshots:
- Point in time copy of EBS volume
- Stored in S3
- Differential
- Can be used to bootstrap image

Simple Database Service (SimpleDB):
- Non-relational database: key-value
- Partitioned into *domains*
- Consistency
  - highly replicated
  - eventual consistency
- Typical uses: logging, indexing S3 data
- Erlang!
- Replaced by DynamoDB
Simple Queue Service (SQS):

- Message-queue oriented communication service
- Persistent, asynchronous messaging
- At-least once delivery guarantee
- No ordering guarantee
- Access control

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/
Platform as a Service

Service provider provides:

- Hardware infrastructure
- OS and platform software (middleware)
- Distributed storage management
- Load balancing, replication, migration
- Management and Monitoring services

Client provides:

- Application
Challenges – Client:

- Learn new API and environment
- Follow API
- Optimise to limits of API and platform
- Security for own app

Challenges – Provider:

- Transparency (naming, redirection)
- Scalability: replication and load balancing decisions
- Synchronisation and coordination
- Security
- Fault tolerance
- Monitoring
- Software maintenance and sys admin
Example 2: App Engine

- Various development languages (Python, Java, PHP, Go)
- ... and runtime environments
- Storage based on Big Table
- Optimisation via Memcache
- Lots of APIs
- Per use billing
- Transparent scaling
1. Create an account (5 min). GAE offers a large amount of quota for free

2. Create a new project
3. Write an application using GAE’s framework

4. Deploy your application on GAE!
sup, World!

5. Running application.

Scale up/down, load balancing, replication, database management, ... many services are provided by GAE.
6. Check your resource usage (CPU, storage, # of API calls, ...)
Pay only when usage exceeds the free quota
SOFTWARE AS A SERVICE

Service provider provides:

- Hardware infrastructure
- OS and platform software (middleware)
- Distributed storage management
- Load balancing, replication, migration
- Management and Monitoring services
- Application

Client provides:

- Data
Challenges – Client:
- Learn new application
- Deal with potential restrictions
  - Web interface, restricted functionality
  - No offline access, no local storage

Challenges – Provider:
- Transparency (naming, redirection)
- Scalability: replication and load balancing decisions
- Synchronisation and coordination
- Security
- Fault tolerance
- Monitoring
- Software maintenance and sys admin
- Application development and maintenance
KEY CHALLENGES OF CLOUD COMPUTING

Scalability:

➤ Datacentre vs Global
➤ Partitioning
  • Services and Data
➤ Replication

Consistency:

➤ Dealing with consequences of CAP Theorem
➤ Dealing with un-usability of eventual consistency
Reliability:

- SLA (Service Level Agreement): guarantees given by provider
  - How reliable are the guarantees?
  - What is the consequence if they aren’t met?

- Redundancy and Replication
  - within same provider (e.g. Availability Zones, Regions, etc.)
  - migration across providers

- Geographically distributed architecture
Design for failure: Chaos Monkey

- test how well system deals with failure
- regularly and randomly kill system services
Security and Privacy:

➔ External threats
  • Denial of Service
  • Infrastructure or platform service compromise
  • SaaS compromise: data theft

➔ Co-located threats: other customers
  • Isolation: but, covert channels, bugs in isolation

➔ Privacy: data collected by providers
  • IaaS and PaaS providers: encryption only helps a bit
  • SaaS providers: at mercy of service provider
  • Governments and others: where is your data stored or processed? Which laws apply?
DEVELOPING FOR THE CLOUD

Examples from Amazon:

WEB APPLICATION HOSTING

http://aws.amazon.com/architecture/