Introduction to Amazon Web Services

Thilina Gunarathne
Salsa Group, Indiana University.
With contributions from Saliya Ekanayake.
Introduction

• Fourth Paradigm – Data intensive scientific discovery
  – DNA Sequencing machines, LHC

• Commercial Cloud Platforms
  – Amazon Web Services
  – Microsoft Azure Platform
  – Google AppEngine
Cloud Computing

• On demand computational services over web
  – Spiky compute needs of the scientists
• Horizontal scaling with no additional cost
  – Increased throughput
• Cloud infrastructure services
  – Storage, messaging, tabular storage
  – Cloud oriented services guarantees
  – Virtually unlimited scalability
Amazon Web Services

• Compute
  – Elastic Compute Service (EC2)
  – Elastic MapReduce
  – Auto Scaling

• Storage
  – Simple Storage Service (S3)
  – Elastic Block Store (EBS)
  – AWS Import/Export

• Messaging
  – Simple Queue Service (SQS)
  – Simple Notification Service (SNS)

• Database
  – SimpleDB
  – Relational Database Service (RDS)

• Content Delivery
  – CloudFront

• Networking
  – Elastic Load Balancing
  – Virtual Private Cloud

• Monitoring
  – CloudWatch

• Workforce
  – Mechanical Turk
Amazon Web Services

• Compute
  – Elastic Compute Service (EC2)
  – Elastic MapReduce
  – Auto Scaling

• Storage
  – Simple Storage Service (S3)
  – Elastic Block Store (EBS)
  – AWS Import/Export

• Messaging
  – Simple Queue Service (SQS)
  – Simple Notification Service (SNS)

• Database
  – SimpleDB
  – Relational Database Service (RDS)

• Content Delivery
  – CloudFront

• Networking
  – Elastic Load Balancing
  – Virtual Private Cloud

• Monitoring
  – CloudWatch

• Workforce
  – Mechanical Turk
Demo Application

• Job queue based embarrassingly parallel application execution
  – BLAST, Monte Carlo simulations, many image processing applications, parametric studies

• Cap3 – Sequence Assembly*
  – Assembles DNA sequences by aligning and merging sequence fragments to construct whole genome sequences

• Executable available at
  http://seq.cs.iastate.edu/cap3.html

• Demo programs
  – http://salsahpc.indiana.edu/tutorial/apps/aws/

Sequence Assembly in the Clouds

**Cap3** parallel efficiency

**Cap3** – Per core per file (458 reads in each file) time to process sequences
Cost to assemble to process 4096 FASTA files*

- **Amazon AWS total : 11.19 $**
  - Compute 1 hour X 16 HCXL (0.68$ * 16) = 10.88 $
  - 10000 SQS messages = 0.01 $
  - Storage per 1GB per month = 0.15 $
  - Data transfer out per 1 GB = 0.15 $

- **Azure total : 15.77 $**
  - Compute 1 hour X 128 small (0.12 $ * 128) = 15.36 $
  - 10000 Queue messages = 0.01 $
  - Storage per 1GB per month = 0.15 $
  - Data transfer in/out per 1 GB = 0.10 $ + 0.15 $

- **Tempest (amortized) : 9.43 $**
  - 24 core X 32 nodes, 48 GB per node
  - Assumptions : 70% utilization, write off over 3 years, including support

* ~ 1 GB / 1875968 reads (458 reads X 4096)
Security Credentials

• Access Keys
  – Making a REST or Query API request
  – JAVA SDK for S3, SQS, SimpleDB

• EC2 Key Pairs
  – Launching/connecting to EC2 instances

• X.509 Certificate
  – SOAP API
  – Command line tools
AWS Toolkit for Eclipse

• Open source plug-in for Eclipse
• AWS Java SDK
  – Java API for AWS services
• Amazon SimpleDB management
  – Configure, edit, query
• Amazon EC2 management
  – Deploy, debug, manage
Installing AWS Toolkit in Eclipse

• Installing
  – Java 1.5 or higher
  – Eclipse 3.5 or higher (Java EE distribution recommended)
  – http://aws.amazon.com/eclipse
Simple Storage Service (S3)

• Internet Data Storage
  – Reliable, Simple, Scalable, and Inexpensive

• Three Concepts
  – Buckets
    • Analogous to a folder with no nesting
    • URL accessible
    • Option to enforce geographical constraints
  – Objects
    • Actual data stored in buckets, e.g. PDF, Video, etc.
    • Up to 5 gigabytes
    • Unlimited number of objects
    • Retrievable via HTTP, HTTPS, or BitTorrent
    • Private, public or selectively for users
  – Keys
    • Unique key to identify each object in a bucket
Simple Storage Service (S3)

- **Access Logs**
  - Option to enable logs for buckets

- **Pricing**
  - Data storage
    - 0.15$ per GB for first 50TB to 0.055$ per GB for over 5000TB
  - Data transfer in
    - 0.1$ per GB (free till Nov, 2010)
  - Data Transfer out
    - 0.15$ per GB up to 10TB to 0.08$ per GB for over 150TB
  - Requests
    - PUT, COPY, POST, LIST -> 0.01 $ per 1000 requests
    - Others -> 0.01$ for 10,000 requests

- **Reduced Redundant Storage**
  - 2/3 of the storage cost
Using S3 as the Data Storage

- S3 management console
- Uploading the input data to S3
- Downloading/uploading files (s3 objects) programmatically
- Run Sample
  - AWSStepOne eclipse project
AWS Import/Export

• Accelerates Moving Large Scale Data
  – In to and out of AWS using portable storage
  – Utilized Amazon’s high-speed internal network
  – Often faster than Internet upload/download for large data
• Simple Steps
  – Prepare a portable storage device
  – Request AWS with S3 bucket, key, and shipping address
  – Receive an ID, digital signature, an AWS shipping address
  – Identify and authenticate storage device with digital signature
  – Ship it and wait for Amazon to ship it back 😊
• Data migration, content distribution, offsite backup, disaster recovery, direct data interchange
Simple Queue Service

- Reliable and Scalable Distributed Messaging Framework
  - Create, store, and retrieve text messages (up to 8 KB)
  - Eventual consistency
- Messages
  - Stored until retrieved or four days
  - MessageID, ReceiptHandle, MD5OfBody, Body
- Queues
  - Possible to create unlimited number of queues
- Concerns
  - Queue order, i.e. FIFO, is not guaranteed
  - Message deletion in a queue is not guaranteed
  - Querying a queue is not guaranteed to return all messages
  - Guarantee at least once delivery, but not exactly once
Simple Queue Service

• **Visibility Timeout**
  – When received, the message will be locked in the queue for a given time
  – Message reappears when the lock “expires”, unless deleted by the earlier recipient

• **Access through REST as well as SOAP API’s**

• **Queue sharing**

• **Pricing**
  – 0.01$ for 10,000 requests
  – Data transfer in
    • 0.10$ per GB after Nov, 2010
  – Data transfer out
    • 0.15$ per GB up to 10TB TO 0.08$ per GB over 150 TB
Using the Queue to Schedule Jobs

• Queue Operations
  – CreateQueue
  – putMessage
  – getMessage
    • visibility time out
  – deleteMessage

• Fault tolerance

• Run sample
  – AWSSampleTwo Eclipse project
Simple Notification Service (SNS)

• Notification Service
  – Scalable, flexible, and cost-effective
  – Topic based publishing
  – Multiple protocol support, e.g. HTTP, email, etc.
  – Eliminates polling through push mechanism

• Simple Steps
  – Create a topic
    • Identify subject or event type
  – Set policies
    • Publisher/subscriber limiting, protocol, etc.
  – Add subscribers
  – Publish message
SimpleDB

- Non-relational data store
  - No need to pre-define schema

- Dataset Indexing and Querying Framework
  - Highly available, scalable, secure, and fast
  - Store and retrieve structured data
  - Eventual consistency
    - Optional consistent reads
  - No transactions
    - Conditional puts/deletes
      - Condition based on existing value
SimpleDB

• Domains
  – Containers to store and query structured data
    • Analogous to a spreadsheet
  – No cross domain querying

• Items
  – Individual objects within domains
    • Analogous to a row in worksheet
    • Contains attributes with values; similar to columns and cells
SimpleDB

• Limitations
  – Domain size, domains per AWS account, Attributes, etc.

• Pricing
  – Free tier
    • 25 machine hours, 1 GB storage
  – Machine utilization
    • 0.14$ per machine hour
  – Data transfer in
    • 0.10$ per GB after Nov, 2010
  – Data transfer out
    • 0.15$ per GB up to 10TB TO 0.08$ per GB over 150 TB
  – Structured storage
    • 0.25$ per GB per month
Using the SimpleDB for monitoring & metadata storage

• Operations
  – CreateDomain
  – ReplaceableItem List
  – batchPutAttributes

• Run sample
  – AWSSampleThree Eclipse project

• Check the Eclipse SimpleDB management view
Relational Database Service (RDS)

- Relational Database as-a-service
  - Full capabilities of MySQL database
  - Easy deployment, managed, secure, scalable, and reliable
- Simple Steps
  - Use AWS Management Console/API to launch a database instance (DB Instance)
  - Connect to DB Instance with any MySQL supported tool
  - Monitor through Amazon CloudWatch
- Features
  - Automated backups
  - DB snapshots
  - Multi-AZ deployments
    - Enhanced availability though multiple availability zones
SimpleDB vs RDS

• SimpleDB
  – No administrative burden at all
  – Scales up/down automatically
  – Highly available
    • No downtime
  – No joins, no transactions
  – Flexible

• RDS
  – Existing applications that require relational database
  – Need to decide the scaling decisions
    • How much storage, what size instance, etc
Elastic Compute Service

• Lease Linux as well as Windows VM’s
  – 32 bit as well as 64 bit VM’s
  – Pay as you go
    • Just a credit card to get going
  – Dynamically scale up/down
  – Increase throughput by horizontal scaling for the same cost
  – ‘root’ access to VM’s

• Pre-configured, template images
  – Create AMI to store customized images
Elastic Compute Service

• Purchasing options
  – On demand
  – Reserved
    • One time fee + usage
  – Spot
    • Bit for unused EC2 capacity
    • Sometimes going 33% of the price of on demand
  – Cluster compute instances

• Elastic IP addresses
**Elastic Compute Service**

### Pricing

- Standard, High-memory, High-CPU, cluster

<table>
<thead>
<tr>
<th>Instance Type</th>
<th>Memory</th>
<th>EC2 compute units</th>
<th>Actual CPU cores</th>
<th>Cost per hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large</td>
<td>7.5 GB</td>
<td>4</td>
<td>2 X (~2Ghz)</td>
<td>0.34$</td>
</tr>
<tr>
<td>Extra Large</td>
<td>15 GB</td>
<td>8</td>
<td>4 X (~2Ghz)</td>
<td>0.68$</td>
</tr>
<tr>
<td>High CPU Extra Large</td>
<td>7 GB</td>
<td>20</td>
<td>8 X (~2.5Ghz)</td>
<td>0.68$</td>
</tr>
<tr>
<td>High Memory 4XL</td>
<td>68.4 GB</td>
<td>26</td>
<td>8X (~3.25Ghz)</td>
<td>2.40$</td>
</tr>
<tr>
<td>Cluster 4XL</td>
<td>23 GB</td>
<td>33.5</td>
<td>*</td>
<td>1.60$</td>
</tr>
</tbody>
</table>

* 2 x Intel Xeon X5570, quad-core “Nehalem” architecture
Sequence Assembly Performance with different EC2 Instance Types

![Chart showing performance comparison]

- Amortized Compute Cost
- Compute Cost (per hour units)
- Compute Time
GTM Interpolation performance with different EC2 Instance Types

- EC2 HM4XL best performance. EC2 HCXL most economical. EC2 Large most efficient.
HPC in AWS

• Newest announcement
  – Cluster compute instances

• Features
  – Ability to group them in to clusters
  – Low latency full duplex 10 Gbps between instances
  – Published processor architecture
  – Hardware virtual machine

• Limitations
  – No spot or reserved instances
  – No Auto scaling
CloudWatch

• Monitor Amazon Cloud Resources
  – EC2 instances, EBS volumes, Elastic Load Balancers, and RDS database instances
  – Insight to resource utilization, performance, and demand patterns
  – Exposed through Amazon Management Console, API, command line tools
• Pay only for monitoring EC2 instances
• Enables AutoScaling for EC2 instances
  – Dynamically add/remove instances based on CloudWatch metrics
• Pricing
  – 0.015$ per instance hour
Auto Scaling

• Automatically Scale Up/Down EC2 Capacity
  – Conditions are set based on CloudWatch metrics
  – Seamlessly handles demand spikes and drops
  – Consumed through API/command line tools

• Common Uses
  – Automatically scaling EC2 fleet
    • Close follow up of the demand curve
  – Maintaining EC2 fleet at a fixed size
    • Keep healthy EC2 instance number constant
  – Auto scaling with Elastic Load Balancing
    • Efficient load balancing

• Pricing
  – Free with CloudWatch
Deploying the Application in EC2

• Launching instances
  – Spot instances
  – Security groups

• Log-in to instances

• Public AMI for this demo
  – ami-af0ae1c6
  – You need to fill you keys 😊
AMI

• Amazon Machine Images
• Installing the program
• Saving AMI
Run the Program

• Launch the workers
• Run the Driver program
• Monitor using CloudWatch
Elastic MapReduce

- MapReduce as-a-service
  - Utilizes Apache Hadoop, Amazon EC2, and Amazon S3
- Simple Steps
  - Develop MapReduce program
    - Many language support, e.g. Pig, Java, Ruby, C++, etc.
  - Upload data to S3
  - Create and monitor “job flow” through AWS Management Console/command line/API
- Pros
  - Reliable, secure, elastic, and easy
  - Third party tools
  - Seamless integration with EC2, S3
- Cons
  - No tweaking of Hadoop
  - Only supports Hadoop MapReduce framework
EMR bucket names

• S3N Native File System for Hadoop
  – Bucket names should not contain underscores “_”
  – Bucket names should be between 3 and 63 characters long
  – Bucket names should not end with a dash

• Tips for EMR
  – Include at least 3 slashes in the paths
    • S3n://wc-input/
  – Do not use an existing bucket for output
  – More tips
Running WordCount using EMR

• Upload data to S3
  – Create a logs folder
• Create job flow
• Debugging & logging
• Monitoring using Lynx
• Download output
Elastic Block Store (EBS)

• Data you save in the running instance are not persistent
• Block level storage volumes
• Off the instance persistent storage
• Ideal for applications like databases
• Pricing
  – 0.10 $ per GB per month provisioned
  – 0.10 $ per million I/O requests
Elastic Load Balancing

• Automatic Distribution of Incoming Traffic
  – Distribute across single or multiple Availability Zones
  – Avoid routing to unhealthy EC2 instances
  – Session affinity load balancing
  – Metrics reported by CloudWatch
  – Auto scale capacity
  – Greater fault tolerance
Virtual Private Cloud (VPC)

• Secure and Seamless Bridge
  – Between a company’s IT infrastructure and AWS cloud
  – Isolated AWS compute resources via VPN
  – Extend existing management capabilities to cloud resources, e.g. security, firewalls, etc.

• Features
  – Bridge with encrypted VPN connection
  – Add EC2 instances to VPC
  – Route traffic between VPC and Internet over VPN to examine/monitor data flow

• Pricing
  – 0.05$ per VPN connection per hour
  – Data transfer out – 0.15$ per GB to 0.08$ per GB
CloudFront

- Content Delivery as-a-service
  - Delivers static and streaming content
  - Global network of edge locations
    - US, Europe, Hong Kong/Singapore, Japan
  - Automatic routing of objects to nearest edge location
  - Reliable, scalable, and fast

- Simple Steps
  - Store the original versions of files in a S3 bucket
  - Create a distribution and register the bucket
  - Use the distribution’s domain name to as an access point
Mechanical Turk

• Marketplace for Human Intelligence Work
  – Access a virtual community of on-demand workers
  – Programmatically access marketplace
  – Define Human Intelligence Tasks (HITs)
    • Identifying objects in an image, transcribing audio, etc.
  – Load HITs to marketplace
  – Qualify workforce
    • Enable qualification tests for tasks requiring special skills
  – Pay only for accepted work/output
  – Retrieve results via service API
Thank You!

• Questions? 😊
Acknowledgments

• Prof. Geoffrey Fox, Dr. Judy Qui, Saliya Ekanayake, Tak-Lon Wu (Stephen) and the Salsa group

• Dr. Ying Chen and Alex De Luca from IBM Almaden Research Center

• Virtual School Organizers