CS 683 Emerging Technologies
Fall Semester, 2006
Doc 27 Amazon Web Services
Dec 5, 2006

References

Amazon Web Services,

http://www.amazon.com/gp/browse.html?node=3435361

http://www.mturk.com/mturk/welcome

http://www.amazon.com/gp/browse.html?node=16427261

Ruby AWS::S3 library, http://amazon.rubyforge.org/

Python SQS & S3 library http://code.google.com/p/boto/

Rich Internet Applications, wikipedia

Web 2.0 site, Monkey Bites, http://blog.wired.com/monkeybites/2006/09/web_20_champion.html?entry_id=1551386

Mechanical Turk

http://www.mturk.com/mturk/welcome

Submit tasks that people can perform for you

Can submit via web page, SOAP or REST

When some performs the task, you pay them a fee you set

Amazon Simple Storage Service

http://www.amazon.com/gp/browse.html?node=16427261

On-line storage
Files up to 5Gb

Pricing
- $0.15 per GB-Month of storage
- #0.20 per GB of data transferred

Supports
- SOAP
- REST
- Bittorrent downloads

Ruby Library

http://amazon.rubyforge.org/

S3 - Basics

Buckets	Object (file)
Container for objects(files)	Data & Metadata
Bucket names must be unique	Belongs to a bucket
Bucket creator controls access to bucket	Key (name) unique in bucket
	Metadata name-value pairs content type date modfied Can add own metadata

AWS::S3 Ruby Library

http://amazon.rubyforge.org/

```
Bucket.create('jukebox')
music_bucket = Bucket.find('jukebox')
#Add a file
file = 'black-flowers.mp3'
  S3Object.store(
    file,
    File.open(file),
    'jukebox'
  )

music_bucket.objects       #list of objects in bucket
Service.buckets            #list of bucket
file = music_bucket['black-flowers.mp3']
```

Developer has

Access Key

Secret Access Key

Amazon Simple Queue Service (SQS)

Queue for storing messages between computers

Used to transfer date between parts of distributed application

Amazon is willing to act as a reliable server

Cost

- $0.10 per 1,000 messages
- $0.20 per GB of data transferred

Python SQS Interface

http://code.google.com/p/boto/

```
>>> from boto.connection import SQSConnection
>>> conn = SQSConnection('<aws access key>', '<aws secret key>')

>>> q = conn.create_queue('myqueue')
>>> resultSet = conn.get_all_queues()

>>> from boto.message import Message
>>> m = Message()
>>> m.set_body('This is my first message.')
>>> rs = q.write(m)

>>> rs = q.get_messages()
>>> message = rs[0]
>>> message.get_body()
>>> q.delete_message(message)
```

Amazon Elastic Compute Cloud

Amazon rents time on machines

Provides root access

Each machine at least:

- 1.7Ghz x86 processor
- 1.75GB RAM
- 160GB local disk

SOAP/REST interface

Use S3 for storage

Cost $0.10 per intsance-hour

Rich Internet Applications (RIA)

Web applications

Features and functionality of desktop applications

JavaScript
Adobe Flash
Adobe Flex
Windows Presentation Foundation
Java applets
Java Web Start
OpenLaszlo
Curl

Adobe Apollo

Cross-operating system runtime (Mac, Windows)

Desktop RIA

Use Flash/Flex and/or HTML/Javascript to develop app

Due 2007

Some Web 2.0 Sites

http://blog.wired.com/monkeybites/2006/09/web_20_champion.html?entry_id=1551386

Flickr
del.icio.us
Local.Live.com
GMail
Google Calender
Google Spreadsheets
Digg
YouTube
Yelp
BlinkList
Bloglines
Basecamp
Writely

Dimewise
Kayak
Spurl
LibraryThing
Last.fm
NetVibes

Mashups

See wikipedia's mashup page

http://en.wikipedia.org/wiki/Mashup_%28web_application_hybrid%29

ET

The End