CSCC09F
Programming on the Web

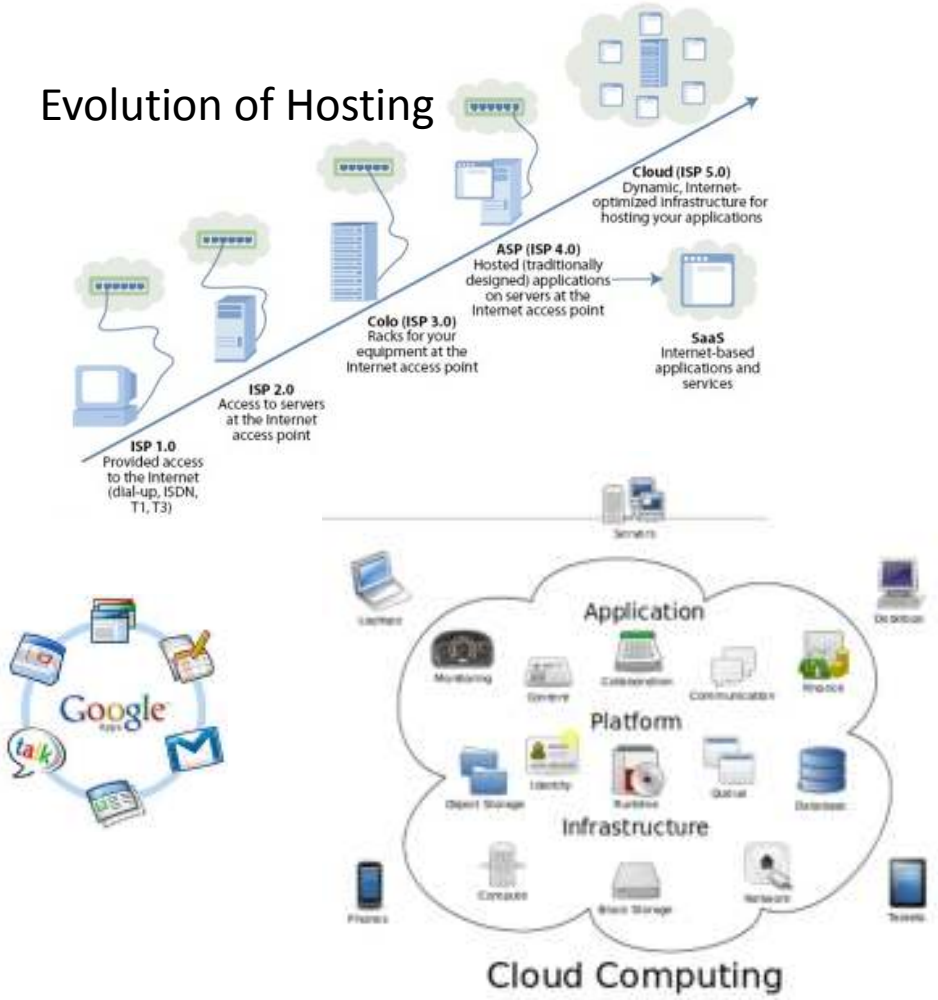
Deploying Apps to the Cloud

What is a "cloud"?

- Software and hardware operating in data-centers
- Public cloud: cloud used to provide utility computing services, e.g.:
  - Amazon EC2: Amazon data-centers, Xen, EC2 APIs and administrative interface
  - Google AppEngine: Google data centres, GFS, AppEngine APIs, administrative interface …
  - Batch processing software: MapReduce, Hadoop, Pig, Dryad
- Private cloud: datacenters not available for rental, e.g. operated for exclusive use of large corporation

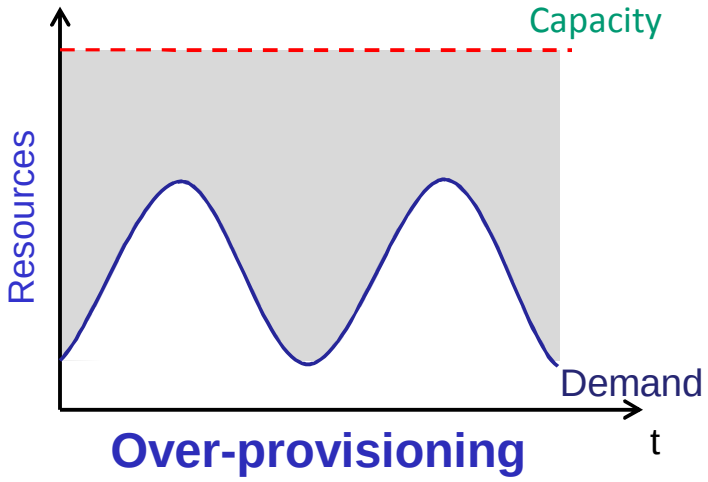
The Cloud: Network-Attached "*"

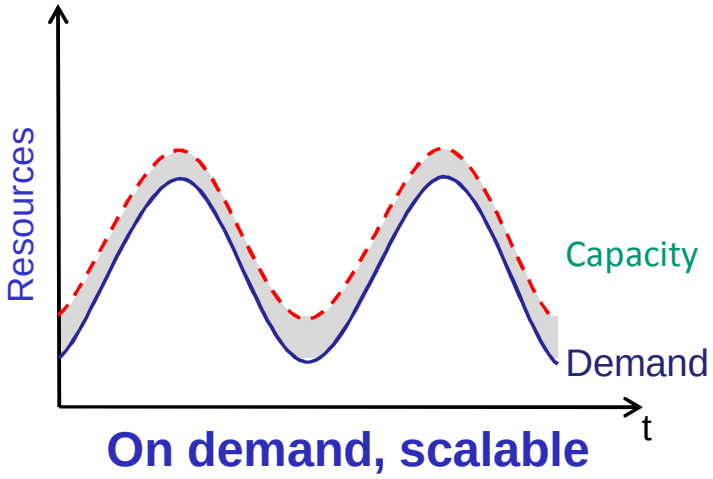
- Concepts:
  - SaaS (Software as a Service)
    - e.g. Google apps, salesforce.com, MS Office 365
  - Paas (Platform as a Service)
    - e.g. Amazon EC2, S3; Google AE, iCloud, Dropbox

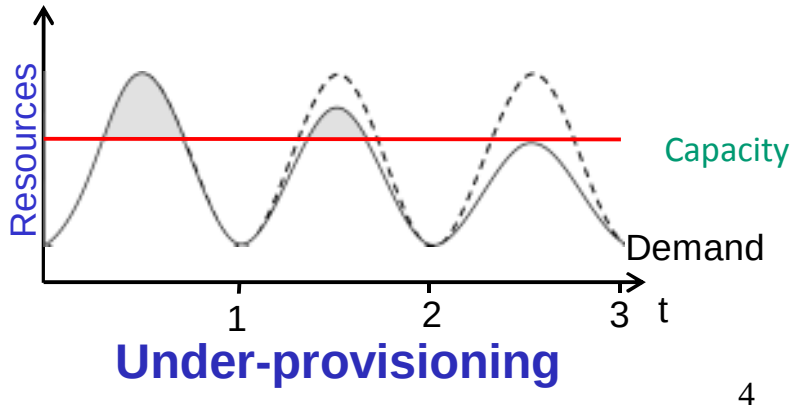


Cloud Computing: Mitigates risks

- typical average utilization 5%-20%
- examples of utilization spikes:
  - Animoto (video slide-show service) demand surged from 50 servers to 3500 servers in 3 days
  - Black Friday sales







Example: Amazon's Elastic Compute Cloud (EC2)

- Rent virtual-machine (VM) instances to run your software
- Monitor and increase/decrease the number of VM's as demand changes
- How to use:
  - create an Amazon Machine Image (AMI): applications, libraries, data and associated settings
  - upload AMI to Amazon S3 (simple storage service)
  - use Amazon EC2 Web service to configure security and network access
  - choose OS, start AMI instances
  - monitor & control via Web interface or APIs

Example: Amazon EC2

- Characteristics:
  - **Elasticity**: increase or decrease capacity within minutes
    - monitor and control via EC2 APIs
  - Completely controlled: root access to each instance
  - Flexible: choose your OS, software packages…
    - Redhat, Ubuntu, openSuse, Windows Server,…
    - small, large, extra large instances
  - Reliable: Amazon datacenters, high availability and redundancy
  - Secure: Web interface to configure firewall settings
- Cost:
  - CPU: small instance, $0.10 per hour for Linux, $0.125 per hour for Windows (1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor)
  - Bandwidth: in $0.10, out $0.17 per GB
  - Storage: $0.10 per GB-month, $0.10 per million I/O requests
  - FREE: one year – 750hrs (Linux/Windows), 600MB RAM, 30GB* disk, 15GB network

Google AppEngine (GAE)

- Develop Web app on your local machine, and then upload to Google. GAE will handle the rest for you …
- Steps to use GAE:
  - Download AppEngine SDK (Windows, Linux, MacOS)
  - Develop your program locally (Python, Java, Go)
    - e.g. set of Python programs,
      input = request URL (a REST Web service request),
      output = return message (e.g. HTML)
    - debug locally
  - Register for a Google application id
  - Submit your application to Google

GAE: Hello World - local setup

- Creating a Simple Request Handler

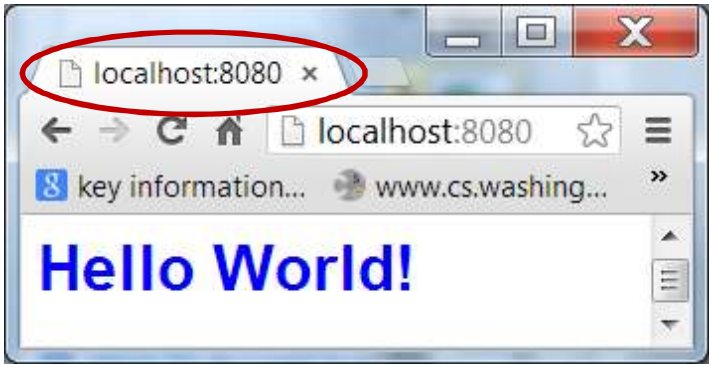
  Create file **hello_gae.py**:

```
print 'Content-Type: text/html'
print ''
print '<h1 style="color:blue">Hello World!</h1>'
```

- Map url to handler (similar to web.xml)

  Edit configuration file app.yaml

```
application: c09hello
version: 1
handlers:
- url: /.*
  script: hello_gae.py
```

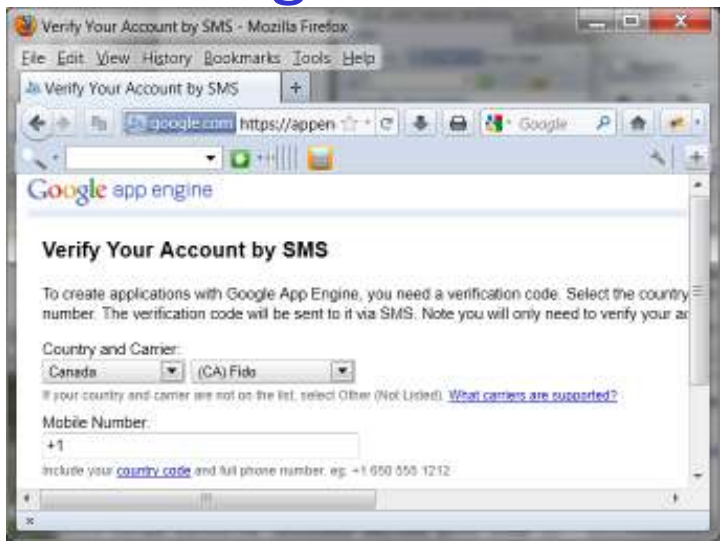


- Data storage:
  - Distributed file system
  - Store using AppEngine API, retrieve using GQL
- Debug locally: http://localhost:8080/

GAE Deployment

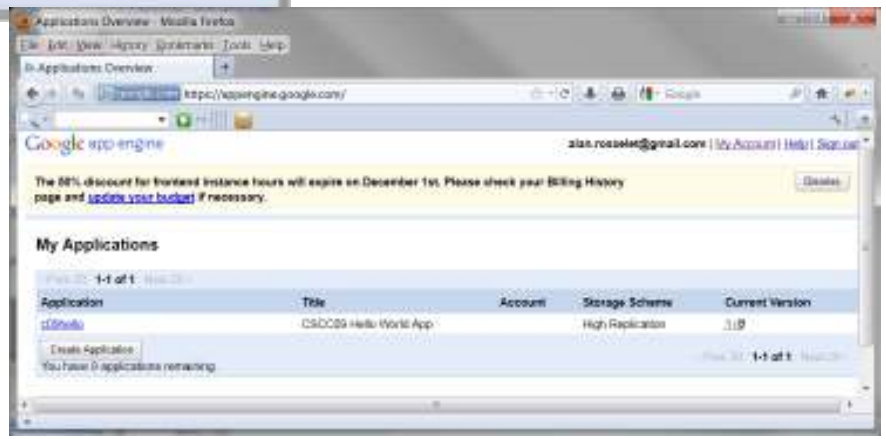
- Register for an application ID
  - http://appengine.google.com
  - verification code sent to your mobile
- Uploading the Application
  - **`appcfg.py update helloworld/`**
  - enter Google username and password at the prompts
  - **`http://application-id.appspot.com`**
- Manage using Administration Console
  - set up domain name
  - invite other people to be developers
  - view error logs, traffic logs
  - switch between different versions

Registration

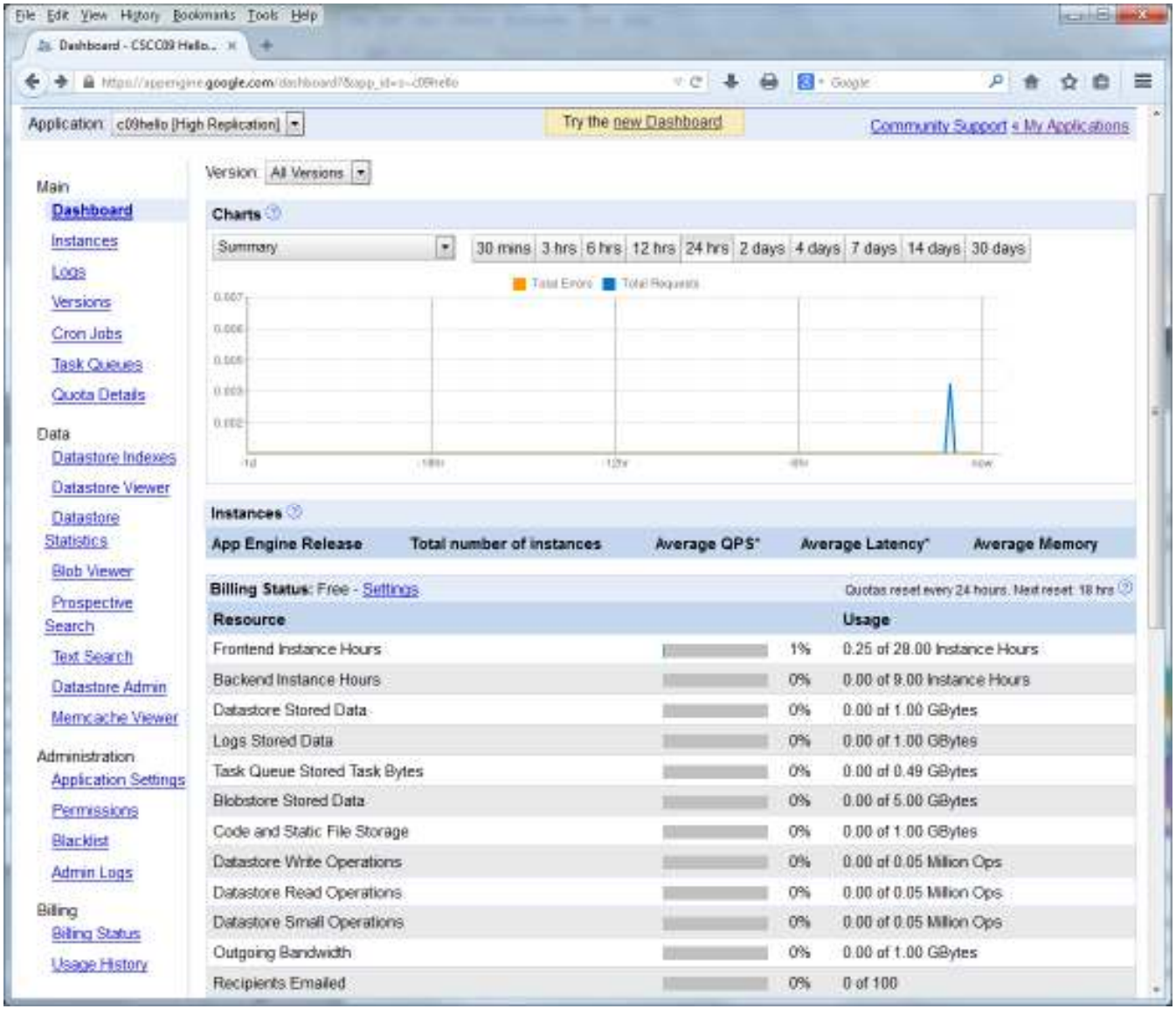


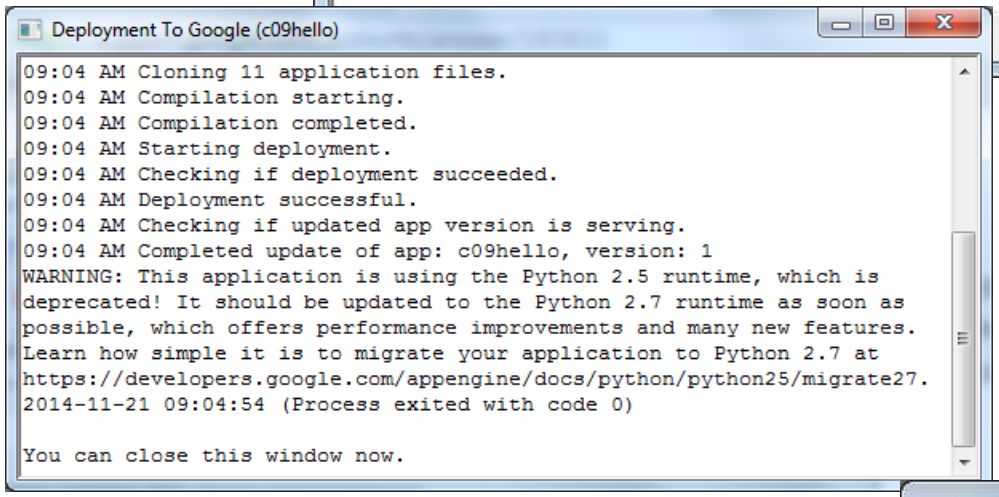
Upload

Manage

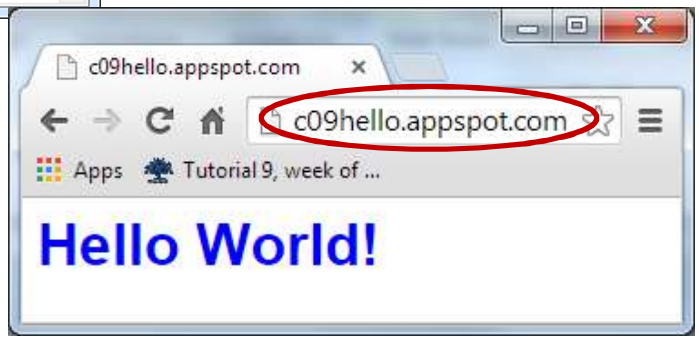


Dashboard


Application c09hello [High Replication]
Try the new Dashboard
Community Support « My Applications
Main
Dashboard
Instances
Logs
Versions
Cron Jobs
Task Queues
Quota Details
Data
Datastore Indexes
Datastore Viewer
Datastore Statistics
Blob Viewer
Prospective Search
Text Search
Datastore Admin
Memcache Viewer
Administration
Application Settings
Permissions
Blacklist
Admin Logs
Billing
Billing Status
Usage History
Version All Versions
Charts
Summary
30 mins 3 hrs 6 hrs 12 hrs 24 hrs 2 days 4 days 7 days 14 days 30 days
Instances
App Engine Release
Total number of instances
Average QPS*
Average Latency*
Average Memory
Billing Status: Free - Settings
Quotas reset every 24 hours. Next reset: 18 hrs
Resource
Usage
Frontend Instance Hours 1% 0.25 of 28.00 Instance Hours
Backend Instance Hours 0% 0.00 of 9.00 Instance Hours
Datastore Stored Data 0% 0.00 of 1.00 GBytes
Logs Stored Data 0% 0.00 of 1.00 GBytes
Task Queue Stored Task Bytes 0% 0.00 of 0.49 GBytes
Blobstore Stored Data 0% 0.00 of 5.00 GBytes
Code and Static File Storage 0% 0.00 of 1.00 GBytes
Datastore Write Operations 0% 0.00 of 0.05 Million Ops
Datastore Read Operations 0% 0.00 of 0.05 Million Ops
Datastore Small Operations 0% 0.00 of 0.05 Million Ops
Outgoing Bandwidth 0% 0.00 of 1.00 GBytes
Recipients Emailed 0% 0 of 100



GAE: Deployment



GAE Summary

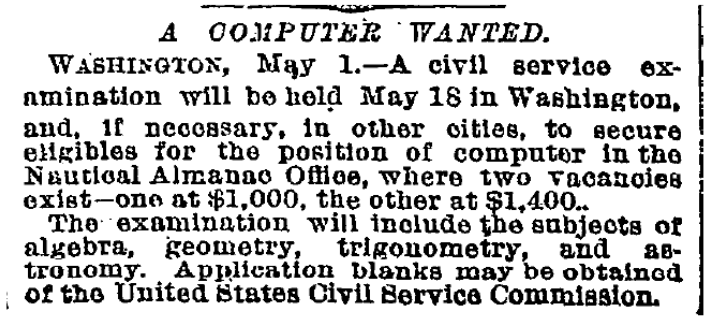
- Characteristics
  - easy to get started, low administrative overhead
  - scales automatically
  - reliable – always there, even if left dormant for months
  - easy to integrate with Google user service: request login…
- Cost:
  - can set daily quota
  - CPU hour: 1.2 GHz Intel x86 processor

Resource	Unit	Unit cost	Free (daily)
Outgoing Bandwidth	gigabytes	$0.12	10GB
Incoming Bandwidth	gigabytes	$0.10	10GB
CPU Time	CPU hours	$0.10	46 hours
Stored Data	gigabytes per month	$0.15	1GB (all)

What's new with Cloud Computing

- Illusion of unlimited computing resources
- Elimination of an up-front commitment by users
- Ability to use and pay on demand
- Lack of tangible product, ownership, privacy?
- Cloud Computing vs P2P?
  - both take advantage of remote resources
  - P2P: does not use clouds (datacenters), peers do not get paid (at least not in $$), lower reliability
- Cloud Computing vs "Grid" Computing?
  - Both use clouds, but Grid Computing requires commitment, shared resources, based on common interests. E.g. University research-grid

Example Cloud "Killer Apps"



A COMPUTER WANTED.

WASHINGTON, May 1.—A civil service examination will be held May 18 in Washington, and, if necessary, in other cities, to secure eligibles for the position of computer in the Nautical Almanac Office, where two vacancies exist—one at $1,000, the other at $1,400.

The examination will include the subjects of algebra, geometry, trigonometry, and astronomy. Application blanks may be obtained of the United States Civil Service Commission.

- Mobile and Web applications
  - mobile devices: low memory & computation power
- Extensions of desktop software
  - Matlab, Mathematica, other CPU-intensive tasks
- Batch processing / MapReduce
  - Washington Post: 200 EC2 instances (1,407 server hours), processed 17,481 pages (low-quality non-searchable PDF images) of Hillary Clinton's travel documents 30-min/page with desktop OCR software; 9 hours on EC2 ($144.62)
  - New York Times used 100 Amazon EC2 instances + Hadoop application to convert 4TB of raw TIFF images (digitized articles from 1851-1980) into 11K article PDFs in 24 hours ($240)