What is…
Google App Engine?

Wesley Chun
Developer Advocate, Google
http://code.google.com/appengine
@wescpy / @Google / @App_Engine

Cloud Computing
What is it (besides being buzzword-compliant)?
What is 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) Forum May 2010
http://csrc.nist.gov/groups/SNS/cloud-computing/forum-workshop_may2010.html

Cloud Computing concept... it's been around longer than you think

"In 1984, John Gage and Bill Joy of Sun Microsystems were credited with saying, "The network is the computer," one of the most memorable slogans of the information age. This concept, based on the tenet of making computing resources available to all users irrespective of location, as long as they are connected to the network, forms the basis of the Internet as we know it, and foretold the advent of cloud computing."

Garry Metcalf, Analysys Mason, Sep 2009
The Benefits of Cloud Computing

• Reduced Cost  
  o Incremental payment based on usage
• Elastic Storage  
  o Data storage can easily grow with business
• High Automation  
  o Platform updates/patches updated automatically
• Flexibility  
  o Resources can be changed on-demand
• More Mobility  
  o Universal Accessibility
• Shift towards Innovation  
  o Less maintenance, can focus on product innovation

Bottom line: More focus on business!

The shift towards Cloud Computing

"By 2012, 20% of Global 2000 enterprises will be using public cloud services, up from under 5% in 2009."

Yefim Natis - Gartner Cloud Computing Analyst, AADI Dec 2009
What is App Engine?

- App Engine is a platform
- You build & test your app
- Then upload your app to Google
- App Engine runs everything
- No need to worry about machines, network, storage, scalability, etc.
Challenges building web apps

What keeps you up at night?
DIY Hosting means hidden costs

• Idle capacity
• Software patches & upgrades
• License fees
• Lots of maintenance
• Traffic & utilization forecasting
• Upgrades
Easy to start
Easy to scale
Easy to maintain

We do the dirty work…

Google App Engine

“We wear pagers so you don’t have to”
The Components
of Google App Engine

Components

Language Runtimes
Web-based Admin Console
SDK
Scalable Infrastructure
Components

Scalable Infrastructure

Linux
GFS
Bigtable
Hardware

Components

Python, Java
& Go Runtimes

Batteries Included

Alternative API/Runtimes also available (more later)
What are programming languages?

Various dialects that let humans give instructions to computers

• Some are easier than others
• Some are special-purpose
• Adults: Java, C/C++, Python, Ruby, PHP
• Kids: Scratch, Alice, Python, BASIC

Getting the job done

The king of ease-of-use

• Extremely rapid development
• Very low barrier of entry
• Simple yet robust syntax
• Rich library of packages/modules
• App Engine’s first language API
The king of enterprise development

Java

- Ubiquitous in Enterprise computing
- Adheres to Java servlet standard
- Rich library of packages/modules
- Eclipse Plug-in support
- Alternative language support

The new kid on the block

- Best of both worlds
  - The power & reliability of a statically-type language
  - The benefits of dynamically-typed languages
  - Built-in concurrency
  - Compiled but w/garbage collection & run-time reflection
- Flexible alternative to Python & Java
Java compliant Servlet Standard

Ensuring Portability

Java standards

<table>
<thead>
<tr>
<th>JSR-164</th>
<th>JSR-220, JSR-243</th>
<th>Java SE</th>
<th>JSR-019</th>
<th>JSR-107</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java Servlet</td>
<td>JDO / JPA</td>
<td>java.net.URL</td>
<td>javax.mail</td>
<td>javax.cache</td>
</tr>
</tbody>
</table>

Web App Container | Datastore API | URL Fetch | Mail API | Memcache |

Low level App Engine APIs

Java Plug-n-Play

Google Plugin for Eclipse
Extended Language support through JVM

- Java
- Scala
- JRuby (Ruby)
- Groovy
- Quercus (PHP)
- Rhino (JavaScript)
- Jython (Python)

Components

Web-based Admin
Application Monitoring

App Engine Dashboard

App Engine Monitoring

App Engine Health History
Components

- SDK
- Run Locally
- Easy Deploy
- Manage Versions
- APIs

Case Studies
Serving our Users
App Engine growth

How many ______ ? By the numbers

100,000+ Active developers per month
200,000+
Active apps per week

1,500,000,000+
Page views per day
Developers who know that App Engine…

…scales for social web and mobile apps
…and grows with you and your app

Social networking at scale

>62M Users

- 3.6MM DAUs on FB
- 1.9MM DAUs on MS
- Add Orkut, Bebo, Hi5, Friendster, Hyves, Ning...

Gigya...

gigya Socialize
...scales for event-based websites

Official Royal Wedding Website hosted on App Engine

On Wedding day, served:
- Up to 2000 requests per second
- 15 million pageviews
- 5.6 million visitors

http://goo.gl/F1SGc
…scales for enterprise-targeted cloud apps

Not all apps user-facing or web-based!!

- Need backend server processing? Want to build your own?
- Go cloud with App Engine!
- No UI needed for app to talk to App Engine, just need HTTP or XMPP
- Great place for user info e.g., high scores, contacts, badges, etc.
- Better UI: move user data off phone & make universally available
…scales for web and mobile gaming

Features and Futures
Services and APIs now... and coming soon
### Rich set of App Engine services/APIs

- **Memcache**
- **Datastore**
- **URL Fetch**
- **Mail**
- **XMPP**
- **Task Queue**
- **Images**
- **Blobstore**
- **User Service**

---

### In Review: 4+ years of New Features & Continuous Innovation

<table>
<thead>
<tr>
<th>Year</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2008</strong></td>
<td><strong>App Engine with Python runtime launch</strong>, Django 0.96, Memcache API, Images API, Logs export, Batch write/delete, HTTPS support, System status dashboard page, Detailed Quota page in Admin Console</td>
</tr>
<tr>
<td><strong>2009</strong></td>
<td><strong>Java runtime launch</strong>, Billing, Remote API &amp; Shell, Bulkloader, Cron jobs, Key-only queries, Task Queues, Django 1.0 &amp; 1.1, XMPP API, Receiving Email, Blobstore API</td>
</tr>
<tr>
<td><strong>2010</strong></td>
<td>Datastore cursors, Async URLfetch, Appstats profiling, DoS filter, Eventual consistency, OpenID, OAuth, Namespaces, high-performance image service, Instances console, Datastore Admin with bulk entity delete &amp; copy, Channel API(?)</td>
</tr>
<tr>
<td><strong>2011</strong></td>
<td><strong>Go runtime launch</strong>, High Replication datastore, Bulk entity copy b/w apps, 10-minute URLfetch, Backends (long-term servers), Pull Queues, Geolocation headers, WebP images, ProtoRPC, SDK HRD, Channel presence, Django 1.2</td>
</tr>
</tbody>
</table>
Fees & Usage

- Three classes of service
  - Free
  - Paid ($9 app/month)
  - Premium ($500/month, unlimited apps)

- Quota
  - Current preview/beta quotas (now till late 2011)
    - http://code.google.com/appengine/docs/quotas.html
  - Official product pricing/quotas (late 2011)

- 99.95% SLA for paid service classes
- Operational support for Premium service class

---

App Engine Roadmap

- App Engine out-of-preview/official support
- SSL access on non-appspot.com domains
- Improved datastore import/export backup/restore
- MapReduce
- Full-text Search over Datastore
- Python 2.7
- Improved monitoring and alerting
- Raise request/response size limits for some APIs
- See code.google.com/appengine/docs/roadmap.html
What is “vendor lock-in?”

- **What is it?**
  - Systems which inherently make it **very difficult or impossible to migrate** (data and/or logic) to other similar/competitive systems

- **Applies to App Engine?**
  - Yes & no.
  - Don’t get something for nothing.
  - Goal: take advantage of Google’s infrastructure
  - Price: need to write against Google APIs. Make sense?

**Fighting “lock-in” with App Engine**

- **“Must use” Google APIs**
  - Not necessarily

- **Web frameworks**
  - App Engine comes with webapp
  - But can use Django, web2py, Tipfy, Bottle

- **Datastore API**
  - Can use Django ORM (with django-nonrel)

- **Datastore bulkloader**
  - Easily up- or download your data

- **Open source backend systems**
  - AppScale (UCSB)
  - TyphoonAE
Google Apps Integration
App Engine apps in your Apps domain

Google Apps + your apps

Our Google Apps

Your custom applications

Google's scalable serving architecture
Google Apps integration

http://appid.appspot.com/

Google Apps

http://yourapp.yourdomain.com/

Add to Apps domain

Add Domain

Domain Name: [example.com]

Note: You must sign up for Google Apps to register this domain or place that you already have it.

Sign up for Google Apps → Learn more

You have requested that the "example.com" service be added to your domain

Please accept the Google Apps Engine terms and conditions to continue.

Google App Engine terms and conditions

Law of the Google App Engine terms and conditions require that you agree to the terms of service available at https://developers.google.com/appengine/terms. By accepting the terms and conditions below, you confirm that you have reviewed and agreed to the terms and conditions and that you are a legal entity entitled to enter into this Agreement.

I accept the terms and conditions of service.

Google App Engine

Agreement: I accept. Continue to add this service. Learn more

By accepting "Accept, Continue to add this service," you are agreeing to the Google App Engine terms and conditions. You can make custom mail addresses for this service at any time in the service settings page.

Advance this service → Cancel
Integrated into Apps suite

Service settings

Email - Active
http://mail.google.com/almazianicolai.com

Calendar - Active
http://calendar.almazianicolai.com

Sites - Active
http://sites.google.com/almazianicolai.com

my-app-id - Active
https://my-app-id.appspot.com
http://my-app-id.mazianicolai.com

All services are ready

Chat: active
Users can sign in by downloading Google Talk

Docs: active
http://docs.almazianicolai.com

Mobile
Get Google Apps on your mobile device

Moderator: Google Labs - Active
http://gp.almazianicolai.com

Getting Started

Python Hands-on Workshop
App Engine online resources

http://code.google.com/appengine
  Downloads, docs, forums, FAQ, etc.
  Various Google API codelabs
http://code.google.com/p/googleappengine
  Source code, issues, wiki, etc.
http://code.google.com/eclipse
  Eclipse plug-in for Java users
http://appengine.google.com
  Login, app/account management
http://googleappengine.blogspot.com
  Official Google App Engine blog

Also check http://stackoverflow.com

Hello World

Linux, MacOS, etc. command-line:

  $ dev_appserver.py helloworld # run dev svr
  $ appcfg.py update helloworld # deploy live

Windows GUI (also avail for Mac):

![App Engine GUI Launcher](image)
Project Contents

- **app.yaml** – main configuration file
- **index.yaml** – automatically generated to index your data
- **main.py** – your main application "controller" code goes here

Hello World configuration (app.yaml)
Hello World (main.py)

```python
from google.appengine.ext import webapp
from google.appengine.ext.webapp import util

class MainHandler(webapp.RequestHandler):
    def get(self):
        self.response.out.write('Hello world!')

application = webapp.WSGIApplication([
    ('/', MainHandler),
], debug=True)

util.run_wsgi_app(application)
```

Testing Your Install

```bash
$ cd google_appengine
~/Desktop/google_appengine/helloworld
$ dev_appserver.py helloworld
INFO    2009-03-04 17:51:22,354 __init__.py]
.
.
.

• Can start development server in 2 ways
  • Command-line above
  • Launcher user interface for Macs & PCs
```
Hello World (development server)

By default, your apps hosted at: http://APP-ID.appspot.com
Creating applications

Modify app.yaml for upload to production

```yaml
application: helloworld
version: 1
runtime: python
api_version: 1

handlers:
- url: .*
  script: main.py
```
Uploading applications

$ cd google_appengine
~/Desktop/google_appengine/helloworld
$ appcfg.py update helloworld
Scanning files on local disk.
Initiating update.
Email: ...

• Can upload to production in 2 ways
  • Command-line above
  • Launcher user interface for Macs & PCs

Check it out

Hello world!
Summary: Why App Engine?

- Launch quickly
  - Familiar development tools
  - Fast app uploads & version management
- Innovate continuously
  - Many APIs enable continuous addition of new features
- Serve reliably
  - Google infrastructure, administration console
- Scale seamlessly
  - Elastic usage
- Distribute widely
  - Web, mobile app backend, Google Apps Marketplace

Thank You

Questions?

@wescpy / @Google / @App_Engine
Google App Engine
Update

Wesley J. Chun
wesc-api@google.com
@wescpy
OSCON (Portland, OR)
July 2011

About the Speaker

- Software engineer by profession
  - Currently at Google (cloud products)
- Course instructor: teaching Python since 1998
  - Private Corporate Training & Public Courses
- Community volunteer
  - User groups: BayPIGgies and SF Python Meetup
  - Other: Tutor mailing list, Python conferences
- Author/co-author (books, articles, blog, etc.)
  - *Core Python Programming* ([2009,]2007, 2001)
  - *Python Web Development with Django* (2009)
I Teach

I Write
About The Talk

- Recent Updates to Google App Engine
- Update on existing services
- New features in App Engine 1.5.x
- Leaving preview mode
- Miscellaneous
Go as third language runtime

- Released SDK for Mac & Linux
  - Windows port in the works
- Uses tweaked Python SDK tools
- Launched for production 1.5.2 (Jul 2011)
- [http://blog.golang.org/2011/05/go-and-google-app-engine.html](http://blog.golang.org/2011/05/go-and-google-app-engine.html)
Update on High Replication Datastore

- Master/Slave (M/S) original App Engine datastore
- High Replication Datastore (HRD) uses Paxos algorithm
- 3x the cost of M/S
- Slightly slower (distributed writes)
- Most queries eventually consistent
- Need ancestor queries to ensure strong consistency
- Well-received

HRD as default

- Price reduction to discourage M/S (no longer 3x)
- Delivered ~99.999% uptime since launch in January
- M/S price increases when App Engine leaves Preview
- Working on better tools
  - To make migration easier
  - To estimate updated cost
Backends feature (servers)

- Long-running servers
- Publicly-addressable
- No more 30s or 10m deadline

Pull Queues

- Task Queues vs. Pull Queues
  - Tasks Queues take & execute work
  - Work is "pushed" to them
  - Must create & execute normal tasks
  - Execute at a given rate
- Pull queues offer more flexibility
Use case

- More flexible and open
- Can have App Engine app create tasks
- Use external app to process those tasks
- Or vice versa
- Work can be created or consumed outside App Engine
- External access via REST API
- Allows for 2-way producer-consumer model

Other App Engine 1.5(.0)

- Mail API updates
- Source code downloading
  - Owner or uploader
App Engine 1.5.1

- Geolocation header
  - X-AppEngine-country
- Presence support in Channel API
- WebP support in Images API
- Create RPC services with ProtoRPC
- High Replication in SDK

App Engine 1.5.2

- Production: adjustable scheduler parameters
- Advanced query planning
- Namespaced datastore statistics
- Tasks
  - New Task Queue details page
  - Pull Queue task size bump to 1MB
  - Pull Queue task lease extensions
The Future

Leaving Preview mode

App Engine out-of-preview

- Later this year
- Full productization
- Formal pricing model
- http://googleappengine.blogspot.com/2011/05/year-ahead-for-google-app-engine.html
Original launch

- http://googleblog.blogspot.com/2008/04/developers-start-your-engines.html
- Growth stats since launch
  - 100,000+ developers use App Engine every month
  - 200,000+ active apps
  - 1,500,000,000+ page views per day
- Company recognizes these "Googley" numbers

What does leaving preview mean?

- Long-term commitment by Google to App Engine
- Long-term commitment by App Engine to users!
- It's a real product now
- Comes with "real" product stuff
- http://googleappengine.blogspot.com/2011/05/year-ahead-for-google-app-engine.html
Usage Types/Tiers

- Free Apps
  - Same as before but more restrictive
- Paid Apps
  - 99.95% SLA
  - $9/app/month
- Premier Accounts
  - $500/month
  - Unlimited apps

Instances

- CPU-hours replaced by Instance-hours (IH)
  - 1 instance running for 1 hour
  - Instances similar to instances in Admin Console today
  - Different cost models
    - Pay-as-you-go
    - Weekly
APIs

- Currently charged as CPU hours
- Will be charged per operation

Leaving preview features

- 99.95% SLA
  - [http://code.google.com/appengine/sla.html](http://code.google.com/appengine/sla.html)
- New ToS
- New Pricing
  - [http://www.google.com/enterprise/appengine/appengine_pricing.html](http://www.google.com/enterprise/appengine/appengine_pricing.html)
- FAQ
  - [https://groups.google.com/group/google-appengine/browse_thread/thread/a1bfa432e0c002a7](https://groups.google.com/group/google-appengine/browse_thread/thread/a1bfa432e0c002a7)
  - [https://groups.google.com/group/google-appengine/browse_thread/thread/1e5ba8835fdadc7c](https://groups.google.com/group/google-appengine/browse_thread/thread/1e5ba8835fdadc7c)
Google I/O Updates

Experimental feature updates

- Google I/O talks
- Full-text Search
  - Full-text search service comes to App Engine
  - Other queries: numeric, geo, date search capabilities, etc.
- MapReduce
  - Mapper has been available for awhile
  - Working on completing shuffle and reduce
Other Google cloud activity

- Google Storage launch
  - http://googlecode.blogspot.com/2011/05/google-storage-for-developers-open-to.html
- Google Prediction launch
  - http://googlecode.blogspot.com/2011/05/google-prediction-api-helps-all-apps-to.html

Recent+Upcoming Events

- Oct 18-20: Python training course, San Francisco
  - http://cyberwebconsulting.com
- Sep-Oct: PyCon Argentina & Python Brasil
  - Buenos Aires and São Paolo
- Jul 25-29 O’Reilly Open Source (OSCON), Portland
  - http://oscon.com
- Jul 11-13 ACM CSTA CS&IT Conference, New York
  - http://www.csitsymposium.org
- Jun 20-25 EuroPython, Florence
  - http://europython.eu
- May 8-10: Google I/O, San Francisco
  - http://google.com/io