

A decorative header featuring four overlapping spheres: a green one on the left, and blue, red, and yellow ones on the right, all partially cut off by the top edge of the slide.

Google App Engine

A conceptual overview



A decorative header featuring four overlapping spheres: a green one on the left, and blue, red, and yellow ones on the right, all partially cut off by the top edge of the slide.

Cloud computing

Your code. Our machines



Different kinds of clouds: Instance based

You are aware of every instance or 'machine'

You are largely responsible for the interactions between them

You install and maintain the software that runs on the machines

You get (almost) complete control over the entire system

Example: Amazon EC2



Different kinds of clouds: Fabric based

You don't get allocated a set of machines

You deploy to an 'environment' which controls how many machines are used

Someone else takes care of the machines, OS and system maintenance

Example: App Engine



Different kinds of clouds: Fabric based

You have limited control over the system

You have limited responsibility for the system

Trade flexibility for scalability (down as well as up) and ease of use



What is Google App Engine?

Fabric based cloud

Secure sandboxed multi-language web app environment
distributed across multiple Google datacenters

What does that mean?



Sandboxed

Restricted execution environment to protect you from other users and vice versa

- No threads
- No native code
- No filesystem access
- No sockets
- Must respond to requests within 30 seconds

Shared-nothing web apps

Web apps:

- Receive HTTP request and send back HTTP response

Shared-nothing:

- Each HTTP request sees a clean environment
- Each HTTP request can go to an arbitrary machine
- No state from previous requests is available unless it was persisted to the Datastore or/and Memcache
 - Sessions API in Java fakes it using Datastore and Memcache
 - Various Python libraries can also let you fake it



Shared-nothing web apps

Minimal App Engine web app (in Python) :

- app.yaml file
 - Indicate language, URL and a handler for that URL
- Python file with handler code

Java requires:

- The folder structure for a WAR file
- web.xml file for URL mapping
- appengine-web.xml file for configuring the application
- One servlet to handle the requests



Language specific sandboxes

JVM sandbox

Java 5 and 6 with implementations of:

- JDO
- JPA
- JavaMail
- java.net

JRuby, Clojure, Scala, etc are all supported



Language specific sandboxes

Python sandbox

Version 2.5.2

Data model classes

Simple web app framework

Popular libraries like Django also supported



Quotas, constraints and billing

File Size limits

500 QPS

Limits on number of incoming and outgoing requests (millions per day)

Above certain limits then you have to pay



Why App Engine?

Cheap

Easy scalability

Ease of use

Local development environment with full source code

Ever increasing set of platform services provided by Google

Freemium



Platform services

Distributed Datastore:

- Based upon Bigtable
- Not an RDBMS
- Stores schema-less entities making data migration easier
- Entities have a 'kind' and queries can retrieve all entities with a given 'kind'
- Query on the values of properties of entities
- Supports transactions on an 'entity group'
- GQL: Not SQL but a similar query language

Platform services

Memcache

- Distributed non-persistent in-memory cache

UrlFetch

- Library that fetches content across the web on your behalf since you're not allowed to open sockets
- HTTP and HTTPS
- Synchronous or Asynchronous

Platform services

Mail

- Send email with attachments

Image manipulation

- Rotate
- Resize
- Flip
- Crop

Platform services

Authentication using Google Accounts

- Optional
- People don't have to sign up to use your app
- You can restrict your app to users within a particular domain
- You can find out the user's email address and a permanent id
- Simple and secure API that eliminates an entire class of security problems



Platform services

Scheduled tasks

- Basically cron
- Configure up to 20 of these in the cron.yaml
- Specify a url with a handler that will be invoked at the time you indicate
- Easily tested

Platform services

Task queues

- Experimental feature
- Python only (for the moment)
- A Task handler is a web hook
- You add tasks (URLs with parameters) onto a task queue
- Tasks are effectively delayed HTTP requests
- AppEngine invokes your URL at some point in the future
- Task handlers run just like any other request handler
- Lets you escape the limits of request-response
- Background tasks and complex processing become possible
- Feedback wanted



Platform services

Remote API

- Lets you remotely control your application by sending it HTTP requests
- Local shell built on top of it
- Bulk data migration, import and export can be done this way

More services to come: <http://code.google.com/appengine/docs/roadmap.html>

- XMPP
- Large files
- Datastore dump and restore
- Others...



Resources

- <http://code.google.com/appengine/>
- <http://groups.google.com/group/google-appengine/web/google-app-engine-open-source-projects>
- <http://code.google.com/appengine/casestudies.html>
- <http://appgallery.appspot.com/>
- <http://googleappengine.blogspot.com/>

A decorative header featuring four overlapping spheres: a green one on the far left, and a blue, red, and yellow one overlapping each other towards the right. A thin black horizontal line is positioned below the spheres.

Questions?

The Google logo, consisting of the word "Google" in its signature multi-colored font (blue, red, yellow, green, blue, red) with a trademark symbol (TM) to the right.

Google™