

Grails in the Cloud

Google App Engine, EC2 and Cloud Foundry

Cloud Computing Service Categories

- Infrastructure as a service
 - ▶ On-demand virtual machines
- Platform as a Service
 - ▶ Write your application to a Cloud API
- Software as a Service
 - ▶ Renting software as an end-user (SalesForce, GMail)

Cloud Hosting Options for Grails

- Google App Engine (PaaS)
 - ▶ Use Google's Java App Engine w/Grails
- Amazon EC2 (IaaS)
 - ▶ Configure an EC2 VM for Java & Go!
- Cloud Foundry (IaaS)
 - ▶ Management layer atop Amazon EC2 - simple deployment
- VMWare vSphere (IaaS)
 - ▶ VMWare's private cloud technology



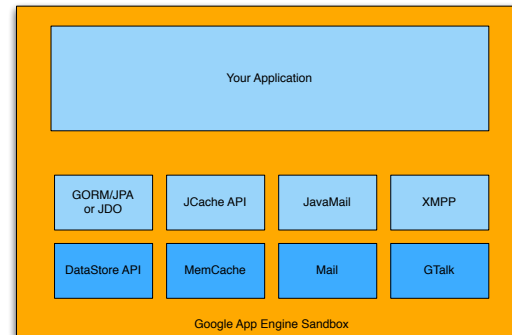
VMWare's vSphere will be a cloud computing platform

PaaS - Google App Engine

- A platform hosted on Google's network
- Language / Platform options
 - ▶ Python
 - ▶ Java
- The Java platform supports Java Web Applications
 - ▶ Google places constraints on Java APIs

Google's Sandbox Restrictions

- Your application cannot directly
 - ▶ Write to the file system
 - ▶ Open a network socket
 - ▶ Spawn a thread or sub-process
 - ▶ Make system calls
 - ▶ Use a standard Relational Database
- JVM Calls are subject to a Whitelist
 - ▶ Any classes/methods outside of the whitelist will fail to execute
- Use of Google's Mail, cache, XMPP interfaces via APIs



- ▶ These restrictions force you into specific choices, and therefore platform lock-in
- ▶ You really have to design your app around AppEngine
- ▶ Not easy to migrate existing apps

Database Restrictions

- Must use Google's DataStore
 - ▶ This is not a relational database
 - ▶ But can be fronted by JDO, JPA
 - ▶ Not all of JPA supported
- Major restrictions
 - ▶ Index definition
 - ▶ Entity Groups and Transactions, OCC
 - ▶ Relationship restrictions
 - ▶ No join queries



Google asks you to switch from a read (query) to a write (pre-calculate) mentality.

It is better to flatten out a relationship if you are going to call up related data on a regular basis.

Writing is much slower than reading.

Transactions cannot span entity groups...

Why Host in AppEngine?

- Zero-cost hosting until threshold
 - ▶ Thresholds are listed on the App Engine site
 - <http://code.google.com/appengine/docs/quotas.html>
 - Access to GMail, Image Service, domain accounts, XMPP, etc...
- Instant scalability
 - ▶ App Engine provides scalability on demand
 - ▶ You are using their underlying services (BigTable, etc)
 - ▶ Tradeoff: vendor lock-in

Grails and AppEngine

- AppEngine supports Groovy and Spring MVC
- The app-engine plugin provides support for App Engine on Grails
 - ▶ Installs the JPA Container
 - ▶ Creates the app-engine deployment scripts
- Currently in progress, with many workarounds and issues
 - ▶ Expect this to improve over next 6 months



Current restrictions are strange

- only 12 applications per account
- cannot delete an application
- you can register for another account, but you must have a unique cell phone # for each account
- your application gets passivated if unused, so some write cron jobs to ping the application

JPA/JDO in Grails

- Must use JPA or JDO instead of Grails domains
 - ▶ Recommend JPA as most compatible with Grails
 - ▶ Place JPA classes in grails-app/domain
 - ▶ Note - you must create your domain classes in packages
- To use JPA with GORM API, install a plugin
 - ▶ `grails install-plugin gorm-jpa`

Demonstration - App Engine & Grails

App Engine Summary

- You can host Grails applications for free, but...
 - ▶ Beware of vendor lock-in, a big issue on App Engine
 - ▶ You must enable billing to go past thresholds
 - ▶ You have to modify your approach and re-write your data layer
 - ▶ Not all plugins will work



- 1,000 file maximum in your deployment
- Most plugins won't work due to filesystem, network, thread restrictions
- JPA doesn't support owned One to Many

Groovy Alternative: Gaelyk

- Framework written by Guillaume Laforge & Marcel Overdijk
- Uses simple "Groovlets" and a direct abstraction of the DataStore API
- Much easier to write and deploy
- Closer to Python AppEngine code
- More information
 - ▶ <http://gaelyk.appspot.com/>



Infrastructure as a Service

Infrastructure Options

- Any platform that hosts Java EE will host Grails
 - ▶ Watch minimum RAM requirements
- Example: Amazon EC2
 - ▶ Easily scale from one to dozens of servers
 - ▶ Implement in any app server and database you'd like
 - ▶ Use EC2 directly or CloudFoundry
- Tradeoff : Price -vs- agility

Amazon Elastic Compute Cloud (EC2)

- A vast network of computers
- You create / use Amazon Machine Images (AMIs) to run Virtual Machines
- Many VMs exist with Java out of the box...
 - ▶ Java "Webapps in a Box" Apache Tomcat on Port 80, MySQL, Java 6
 - EC2 AMI: ami-aa5abac3
 - Reference:
www.javawebappsinabox.com/tutorial.html



This VM has Java 1.6, Apache Tomcat and MySQL all ready to go

Demo - Amazon EC2

But it gets easier!

- Use SpringSource Cloud Foundry
 - ▶ Built by Chris Richardson, EC2 consultant
 - ▶ Company Acquired by SpringSource
- Simple deploy to pre-built EC2 instances
 - ▶ Just supply schema & war
 - ▶ Choose your topology
 - ▶ CloudFoundry will deploy based on your settings

CloudFoundry Demo

In Summary...

- AppEngine is free (to a point), however you are locked in to a vendor solution
- EC2 is very flexible, infinitely configurable, but you need infrastructure expertise
- CloudFoundry (SpringSource tool) hides EC2 complexity and provides web-driven deployment to EC2



Thank you