AppScale

Chandra Krintz
Computer Science Dept.
Univ. of California, Santa Barbara

KDT Mind Meld
Mar. 5, 2012
AppScale

• A cloud platform
  ■ Distributed system that provides a complete runtime stack
    ▸ Upload (Python, Java, Go, ...) programs and web services
  ■ Providing scalable program-level abstractions via well-defined interfaces
    ▸ Different storage options, user management, tasking, messaging, ...
    ▸ Automates configuration, distributed deployment

• Executes over
  ■ Virtualized cluster resources (requires manual VM instantiation)
  ■ Infrastructure-as-a-Service (automatically)
    ▸ Amazon EC2, Eucalyptus, any IaaS with euca2ools support
  ■ Isolation at either/both the VM and process level
What AppScale Is/Does and Why

- Open source
  - Facilitate research into the next generation of
    - Cloud runtime systems, services, applications, technologies
  - Engender a community of users
  - Leverage and integrate wide variety of popular open source technologies
    - HTTP proxies, applications servers, load balancers, databases, multimedia, communications, distributed locking, messaging, programming models (mapreduce)...
    - Emerging research: StochKit, KDT, ...
What AppScale Is/Does and Why

- Open source
- Automate configuration/deployment of distributed apps
  - Broaden participation in use of cloud systems
  - Increase programmer productivity: cloud/web/distributed apps
    - Reduce the overhead of using popular distributed open source technologies
What AppScale Is/Does and Why

- Open source
- Automate configuration/deployment of distributed apps
- Portability for cloud applications
  - Provide a universal set of APIs
  - Over different cloud fabrics
    - Without application modification
    - “Write Once, Run Anywhere (WORA)” for the cloud
  - Eliminate *lock-in* to any single public cloud vendor
    - Encourage cost competition
  - Investigate and facilitate hybrid cloud use
What AppScale Is/Does and Why

- Open source
- Automate configuration/deployment of distributed apps
- Portability for cloud applications
- For a wide range of applications and application domains
  - Using different programming languages & programming models
  - Multiple domains:
    - Web service based
    - Computationally intensive (HPC)
    - Data analytics
What AppScale Is/Does and Why

• Open source
• Automate configuration/deployment of distributed apps
• Portability for cloud applications
• For a wide range of applications and application domains
• Mirror public cloud technology “standard”
  • Scalable, efficient, fault resilient, and elastic
  • API compatibility with “standard”
    • Google App Engine (GAE)
    • Engender user community from an existing one
    • Real applications written by others to evaluate
• Portable API for popular cloud services
  • Data access/storage, mail/messaging, task execution, monitoring
  • VM management (coordination with IaaS layer)
Google App Engine (GAE)

private, enterprise data, Google apps

SDC

GAE Application (Python, Java, Go)

Locally via SDK or at MyApp.appspot.com

GAE Application (Python, Java, Go)

- Images
- URL Fetch
- Services
- Cron
- Memcache
- Blob store
- Tasks
- Users
- IM
- Mail
- Protobuf Data APIs

Data Store
GAE: Upload to Google Public Cloud

GAE Application (Python, Java, Go)

appcfg.py update MyApp/ → MyApp.appspot.com

Free w/ quotas
Pay for additional scale:
- CPU, BW, emails, data
- BigTable/MegaStore (HRD):
  - Automatic scaling
  - High availability

SDC
Administrator Console

private, enterprise data, Google apps

MyApp.appspot.com

GAE app users via the Internet

Google App Engine (GAE)

Images
URL Fetch
Blob Store
Users
Mail
IM
Tasks
Cron
Mem
Cache
Data store

MyApp.appspot.com

private, enterprise data, Google apps

appscale
From GAE to AppScale

Google App Engine SDK

GAE Application (Python, Java, Go)

Images
URL Fetch
Blob Store
Users
Mail
IM
Tasks
Cron
Mem Cache
Data Store

Open-source Google App Engine Software Development Kit (SDK)

Your local cluster resources (virtualized and/or managed via Eucalyptus) or Amazon EC2
<table>
<thead>
<tr>
<th>APIs</th>
<th>GAE</th>
<th>AppScale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datastore</td>
<td>BigTable/Megastore</td>
<td>Cassandra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HBase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hypertable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MemcacheDB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MongoDB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MySQL Cluster</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Redis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Voldemort</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Voldemort</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Amazon SimpleDB</td>
</tr>
<tr>
<td></td>
<td>GAE</td>
<td>AppScale</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Datastore</td>
<td>BigTable/Megastore</td>
<td>Open Source Alternatives</td>
</tr>
<tr>
<td>Blobstore</td>
<td>Proprietary</td>
<td>Tornado + Datastore</td>
</tr>
<tr>
<td>Memcache</td>
<td>Proprietary Memcache</td>
<td>memcached</td>
</tr>
<tr>
<td>XMPP</td>
<td>Google Talk</td>
<td>ejabberd</td>
</tr>
<tr>
<td>Channel API</td>
<td>Google Talk</td>
<td>ejabberd and Strophe.js</td>
</tr>
<tr>
<td>Images</td>
<td>Picassa</td>
<td>Google SDK</td>
</tr>
<tr>
<td>URL Fetch</td>
<td>Proprietary</td>
<td>Google SDK</td>
</tr>
<tr>
<td>Task Queues, Cron</td>
<td>Proprietary</td>
<td>RabbitMQ</td>
</tr>
<tr>
<td>Mail</td>
<td>Google Mail</td>
<td>Sendmail</td>
</tr>
<tr>
<td>MapReduce</td>
<td>Task Queues, Pipeline</td>
<td>Task Queues, Hadoop</td>
</tr>
<tr>
<td>VMInstance Control</td>
<td>--</td>
<td>Euca2ools</td>
</tr>
<tr>
<td>Cloud configuration</td>
<td>--</td>
<td>Neptune</td>
</tr>
<tr>
<td>HPC and data analytics</td>
<td>--</td>
<td>Neptune enabling MPI, UPC, Erlang, X10, StochKit, KDT, ...</td>
</tr>
</tbody>
</table>
Neptune

- Domain-specific programming language and runtime
  - Extensions to the Ruby programming language
- Automating deployment of non-GAE (non-web-service) application code
- Facilitates deployment of apps from other app domains
  - HPC          Data analysis          Graph processing
  - Written in any programming language
    - Python, Java, Go, Ruby, C/C++, Erlang, ...
  - Supported toolkits (Neptune *types*)
    - MPI, UPC, X10, StochKit, KDT
    - MapReduce (Hadoop, Hive)

- Runtime integrates with and leverages AppScale
Using Neptune

- Deploy an AppScale cloud; place code
- On any machine with Ruby and Neptune (gem) write/run job specification
  - Storage (code and data) can be local filesystem, AppScale datastore, Google Bigtable, Amazon S3, Walrus (Eucalyptus), ...

```ruby
neptune :type => :mpi,

:code => "/code/ring",

:nodes_to_use => 32,

:output => "/output/ring"
```
Using Neptune

- Deploy an AppScale cloud; place code
- On any machine with Ruby and Neptune (gem) write/run job specification
  - Storage (code and data) can be local filesystem, AppScale datastore, Google Bigtable, Amazon S3, Walrus (Eucalyptus), ...

```ruby
neptune :type => :kdt,

:code => "/code/Graph500.py",

:nodes_to_use => 16,

:output => "/output/graph500"
```
## AppScale Software Stack

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ejabberd (xmpp &amp; channel support)</td>
<td>Supports instant messaging and channels</td>
</tr>
<tr>
<td>Blobstore</td>
<td>Provides storage solution for AppScale</td>
</tr>
<tr>
<td>Task Q</td>
<td>Manages task queues, ensuring efficient processing and task execution</td>
</tr>
<tr>
<td>memcached</td>
<td>Caches frequently accessed data to improve performance</td>
</tr>
<tr>
<td>Routing (Nginx and HAProxy)</td>
<td>Handles incoming requests and manages load balancing</td>
</tr>
<tr>
<td>AppController and AppServer(s)</td>
<td>Coordinates application deployment and manages server instances</td>
</tr>
<tr>
<td>Datastore Support Layer (API, plug-in, automatic configuration...)</td>
<td>Supports various datastores, including HBase and Hypertable</td>
</tr>
<tr>
<td>Datastore (HBase or Hypertable)</td>
<td>Distributed database system for large-scale applications</td>
</tr>
<tr>
<td>HDFS</td>
<td>High-Performance Distributed File System for large-scale data storage</td>
</tr>
<tr>
<td>Neptune runtime</td>
<td>Virtual machine runtime environment for managing AppScale services</td>
</tr>
<tr>
<td>HPC/Analytics Toolkits</td>
<td>Provides tools for high-performance computing and analytics</td>
</tr>
</tbody>
</table>

AppScale Software Stack Diagram:

- **Hadoop (Hadoop)**
  - **Datastore (HBase or Hypertable)**
  - **HDFS**
  - **Routing (Nginx and HAProxy)**
  - **AppController and AppServer(s)**
  - **Datastore Support Layer (API, plug-in, automatic configuration and deployment support)**
    - **Datastore (Cassandra, MySQL Cluster, Voldemort, MongoDB, MemcacheDB, Redis, ...)**
    - Neptune runtime
    - HPC/Analytics Toolkits
AppScale Deployment

• Available as
  ■ A virtual machine image
    ‣ Eucalyptus compatible: precluding our need to support all Linux distributions and virtualization layers
  ■ An Amazon EC2 AMI
  ■ Open source (automated AppScale image installation)

• Tool set for command-line cloud deployment
  ■ Inspired by AWS tools for instance management
    ‣ Run, describe, terminate instances
  ■ Automates deployment & configuration of all components

<table>
<thead>
<tr>
<th>AppScale software stack (all AppScale components)</th>
<th>AppScale Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System (Ubuntu Lucid)</td>
<td></td>
</tr>
</tbody>
</table>

AppScale VM image (on disk, prior to instantiation)
AppScale roles (specify via a YAML config file, or use defaults)

- AppController, AppLoadBalancer, AppServer, AppDB
- Cloud status monitor, ZooKeeper, Neptune runtime/agent, TaskQ master/agent, logging master/agent
- Custom, Open (for fast elasticity)
AppScale Cloud Deployment
Summary

• AppScale is an open source distributed execution platform for a wide range of applications
  ■ Automatically deploys complex distributed technologies
  ■ Investigate different domains, programming systems, cloud technologies and services
• API-compatibility with GAE
  ■ GAE apps run over AppScale without modification
• IaaS-portable API/deployment for HPC and data analytics
• Hybrid cloud support
  ■ Automatic use of different public clouds and their services
• Many research directions to investigate
  ■ Runtime systems, OS/virtualization, dist computing, optimization
  ■ Cloud (cost models/estimation, scaling/elasticity, hybrids)
Thanks!

- Students and Visitors!
  - Vaibhav Arora, Maciej Baranski, **Chris Bunch**, Navyasri Canumalla, Jovan Chohan, **Navraj Chohan**, Nupur Garg, Anand Gupta, Shashank Hedge, Matt Hubert, Jonathan Kupferman, Puneet Lakhina, Yiming Li, Nagy Mostafa, Yoshihide Nomura (Fujitsu), Kowshik Prakasam, Raviprakash Ramanujam, Andres Riofrío, Sujay Sundaram, Bing Wei, Michal Weigel

- Support
  - Google, IBM, NSF, NIH

http://www.cs.ucsb.edu/~ckrintz
http://appscale.cs.ucsb.edu/
AppScale Cloud Default Deployment

- Head Node
- Slave Node
- Slave Node
- Slave Node
AppScale Cloud Default Deployment

- Authentication
- App/Task distribution
- System coordination
- Fault handler

Application and Task server DB

Application and Task server DB

Application and Task server DB
• **AppScale roles**
  - AppController (AC), AppLoadBalancer (ALB), AppServer (AS), AppDB (DBM, DBS)
  - Cloud status/monitors (CS), ZooKeeper (ZK), Neptune server (NS), TaskQ master/agent (TQM, TQA)

Custom, Open
AppScale Hybrid Clouds

Public cloud - Private cloud
Private cloud - Public cloud
Private cloud K - Private cloud J
Public cloud (zone A) - Public cloud (zone B)
Public cloud vendor X - Public cloud vendor Y
AppScale Hybrid Clouds

Public cloud  - Private cloud
Private cloud  - Public cloud
Private cloud K - Private cloud J
Public cloud (zone A) - Public cloud (zone B)
Public cloud vendor X - Public cloud vendor Y
AppScale Hybrid Clouds

Diagram showing a hybrid cloud setup with layers for App/Task Server, DB, Load Balancer, and Task Queue.
AppScale Hybrid Clouds
AppScale Hybrid Clouds

- Ld Bal Task Q
- App/Task Server
- MPI Job
- S3/Walrus
- DB
- App/Task Server
- DB
AppScale Interoperability & Hybrid Support

- GAE applications: over AppScale or GAE
- Datastore support via database support layer
  - AppScale native (plugins), AWS SimpleDB, AWS S3, GAE Bigtable
- Task Q support
  - AppScale native (rabbitMQ), AWS SQS, Microsoft Azure, GAE
- Task agent support
  - Web service and background (non-GAE) process on-premise
  - Python, Ruby, Java, Go, R, C/C++
  - Python, Java, Go & over AWS over GAE
- High-performance toolkits (not GAE compatible)
  - MPI, UPC, X10, StochKit, KDT
  - MapReduce (Hadoop), Hive support (offline analytics)