AppScale:
Bringing Portability and Simplicity to Cloud-based Application Development and Deployment

Chandra Krintz
Professor
Computer Science Department
UC Santa Barbara

SEDE Keynote
Sep 26, 2013
Web and Mobile App Development

- “Service-ize” digital assets and IP
- Accessible everywhere, all the time (BYOD)
Web and Mobile App Development

- “Service-ize” digital assets and IP
- Accessible everywhere, all the time (BYOD)
- **AppScale** is a cloud platform that simplifies this process
  - Part of exploding trend: developer APIs & plug/play services
AppScale

Cloud Infrastructure (as a Service)
- Applications
- Runtimes
- Security & Integration
- Databases
- Servers
- Virtualization
- Server HW
- Storage
- Networking

Your IT, Devs, DevOps

Cloud Platform (as a Service)
- Applications
- Runtimes
- Security & Integration
- Databases
- Servers
- Virtualization
- Server HW
- Storage
- Networking

Managed by:
- Vendor software (public) e.g. AWS
- PaaS software (on-premise)

- Vendor software (public) e.g. Google App Engine, Azure
- PaaS software (on-premise)
Development & Deployment Simplified

- AppScale decouples innovation from common services
  - Automatically manages and scales apps + service ecosystem
AppScale

Development & Deployment Simplified

- AppScale decouples innovation from common services
  - Automatically manages and scales apps + service ecosystem
AppScale

Development & Deployment Simplified

- AppScale decouples innovation from common services
  - Automatically manages and scales apps + service ecosystem
  - Access scalable services via well-defined de facto standard APIs

- User Credentialing
- Data storage & Processing (NoSQL, SQL, …)
- Security & Authentication
- Messaging & Communications
- Monitoring & Logging
- Web Hosting & Serving
- Innovation

AppScale decouples innovation from common services by
automatically managing and scaling apps and service ecosystem. Access scalable services via well-defined de facto standard APIs.

Developer Innovation
AppScale

Development & Deployment Simplified

- AppScale decouples innovation from common services
  - Automatically manages and scales apps + service ecosystem
  - Access scalable services via well-defined de facto standard APIs
    - Starting point: Google App Engine Services

UserCredentialing

Data storage & Processing (NoSQL, SQL, ...)

Security & Authentication

Monitoring & Logging

Web Hosting & Serving

Messaging & Communications

Innovation

AppScale automatically manages and scales apps and services, allowing developers to focus on innovation. It provides scalable services via well-defined APIs, starting with Google App Engine Services. This simplifies development and deployment, ensuring that innovation is decoupled from common services.
Google App Engine

- Google studied its devs for 10+ years
  - Best practices made available via a public PaaS
  - De facto standard in web/mobile services development
    - Vastly simplifies web and mobile dev and deployment
- Since Fall 2011
  - 3M+ active apps, 300K+ devs, billions of page views daily
    - Wide variety of SMBs and enterprise use
  - Half of ALL IP addresses touch a GAE service weekly
Google App Engine

- Google studied its devs for 10+ years
  - Best practices made available via a public PaaS
  - De facto standard in web/mobile services development
    - Vastly simplifies web and mobile dev and deployment
- Since Fall 2011
  - 3M+ active apps, 300K+ devs, billions of page views daily
    - Wide variety of SMBs and enterprise use
  - Half of ALL IP addresses touch a GAE service weekly

- Limitations of public-only cloud options
  - Lock-in
  - Privacy
  - Loss of control
  - Cost
AppScale Mirrors Google App Engine

Your App Here

[Diagram showing the mirroring process from Google App Engine to AppScale]
AppScale Mirrors Google App Engine

On-premise
Behind your firewall
 Everywhere

NO CODE REWRITE

Your App Here

AppScale
Open Source and Plug & Play

AppScale

MapReduce
NoSQL
Search

cloudera
Hortonworks
Hadoop

HD:
HBase

Cassandra
Hypertable

Apache
Solr
Lucene

Elasticsearch

Eucalyptus
CloudStack
OpenStack

Java
Spring

Python
Django

No Code
Rewrite
Control DevOps vs NoOps

- **DevOps**
  - Developer participation in customization
    - Platform, services
    - Deployment, configuration, scaling

- **NoOps**
  - Platform hides the details of its implementation
  - Automates everything

- AppScale enables *LeanOps*
  - You *choose* the balance that is most efficient and productive for you, your devs, your organization
## AppScale Software Stack

<table>
<thead>
<tr>
<th>API Layer</th>
<th>Task Q (RabbitMQ)</th>
<th>XMPP &amp; channel (ejabberd, strophejs)</th>
<th>Blobstore (Tornado)</th>
<th>memcached</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing (Nginx and HAProxy)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AppController and AppServer(s)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datastore Support Layer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(API, adaptors, plug-ins, automatic config/deploy/scale support)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hadoop</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datastore (HBase or Hypertable)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datastore (Cassandra, MySQL Cluster, MongoDB, MemcacheDB, Redis, ...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HDFS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Installing & Deploying AppScale

- Get the image
  - Download from http://download.appscale.com
  - Or make your own image (10GB+ recommended -- DB data)
    - And install from github (appscale, appscale-tools)

- Get the tools (github wiki: Installing-the-AppScale-Tools)
  - On Mac OSx or use the VM itself

- Deploy an AppScale cloud (2GB+ memory for instances)
  - IaaS deploy: automatic using registered image (EC2/GCE/Euca)
  - Cluster deploy: You start the VMs and specify their IPs
Virtualized Deployment

$ appscale init cluster
#edit the AppScalefile to update ips_layout
$ appscale up
#status displays while cloud starts up

... The status of your AppScale instance is at the following URL: http://192.168.30.4/status

$ git clone https://github.com/AppScale/sample-apps.git
$ appscale deploy sample-apps/python/guestbook

... Your app can be reached at the following URL: http://192.168.30.4/apps/guestbook

$ appscale down

... Successfully shut down your AppScale deployment.
$ appscale init cloud
#edit the AppScalefile to update min/max nodes and add creds
$ appscale up
#status displays while cloud starts up
... The status of your AppScale instance is at the following URL:
http://192.168.30.4/status
$ git clone https://github.com/AppScale/sample-apps.git
$ appscale deploy sample-apps/python/guestbook
... Your app can be reached at the following URL: http://192.168.30.4/apps/guestbook

$ appscale down
... Successfully shut down your AppScale deployment.
Welcome to AppsCake!

AppsCake makes deploying AppScale a piece of cake!

AppsCake currently supports deployment strategies over Amazon EC2, Eucalyptus and virtual machines.

Learn more about AppScale>

Choose deployment infrastructure:
- Cluster
- Cloud

Subscribe

Do you love Google App Engine? Do you love open source?
Subscribe to AppScale news and updates to stay tuned on great product releases.
Go to appscale.com/
STOCHSS: Simplifying Large-Scale Stochastic Simulation and Data Analysis

In collaboration with L. Petzold, Uppsala Univ., & D. Gillespie

- Deploy small jobs locally
- Experiment and explore
  
  Run Locally

- Visualize/download results
  
- Deploy large jobs, store large datasets
  - Trade-off time and monetary cost
  - On-premise or public clouds

  Run Via Cloud
Integrated Development Environment

- Build model(s) - multiphysics
  - Scale it up to increasing levels of complexity
- Explore the parameter space
- Seamlessly deploy the appropriate computing resources as needed
- Building on powerful tools:
  - StochKit2, URDME
- Available for Mac and Linux at www.stochss.org
  (V1.0 supports well-mixed simulation)
Create new species

<table>
<thead>
<tr>
<th>Name</th>
<th>Initial value</th>
<th>Unit</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Species

<table>
<thead>
<tr>
<th>Name</th>
<th>Initial value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>molecules</td>
</tr>
<tr>
<td>C</td>
<td>10</td>
<td>molecules</td>
</tr>
<tr>
<td>Da</td>
<td>1</td>
<td>molecules</td>
</tr>
<tr>
<td>Da_prime</td>
<td>0</td>
<td>molecules</td>
</tr>
<tr>
<td>Dr</td>
<td>1</td>
<td>molecules</td>
</tr>
<tr>
<td>Dr_prime</td>
<td>0</td>
<td>molecules</td>
</tr>
<tr>
<td>Ma</td>
<td>0</td>
<td>molecules</td>
</tr>
<tr>
<td>Mr</td>
<td>0</td>
<td>molecules</td>
</tr>
<tr>
<td>R</td>
<td>10</td>
<td>molecules</td>
</tr>
</tbody>
</table>

Showing 1 to 9 of 9 entries
New StochKit2 Ensemble
Generate a new ensemble using StochKit2. Not sure what this means?

Name: xian_ensemble
The name that will be used to reference the ensemble.

Required Parameters
Simulation time 0 to 100 store state every 0.1 t.u.

Realizations 100000
Number of realizations.

Optional Parameters
Run Locally  Run via Cloud
In Summary, AppScale Is...

- **Open source**
  - Distributed **runtime system** for scalable web & mobile service deployment

- **Easy installation** and deployment on
  - Laptop, on-premise cluster/datacenter, public/private clouds

- **A dev/deployment model that helps you innovate faster**
  - API-compatible w/ **services** from GAE, GCE, AWS, Azure, others
    - Run 3M+ active GAE apps over AppScale without modification
    - Learn from, use, re-purpose 2K+ open source GAE apps

- **StochSS** – Science + Cloud simplified
  - A service-oriented approach for large scale scientific simulations and data analytics

- **Frictionless tenancy**
  - Deploy everywhere, over any service implementation, without changing your app. Move easily when need dictates.
Students and Collaborators

- AppScale Student Researchers and Visitors!
  - Current: Stratos Dimopoulos, Chris Horuk, Hiranya Jayathilaka, Alex Pucher
  - Past: V. Arora, M. Baranski, C. Bunch, N. Canumalla, J. Chohan, N. Chohan, A. Gupta, S. Hedge, M. Hubert, J. Kupferman, P. Lakhina, Y. Li, Y. Nomura (Fujitsu), K. Prakasam, S. Sundaram

- Collaborators
  - L. Petzold, A. Hellander, R. Wolski (Eucalyptus)
  - UCSB Computational Science and Engineering Research Group

- Support
  - Google, IBM Research, NSF, NIH

http://www.cs.ucsb.edu/~racelab
Thank You!

Try it today and send us your feedback:
http://www.appscale.com
@appscalecloud
#appscale on IRC

Chandra Krintz
ckrintz@cs.ucsb.edu
@ckrintz