HYBRID CLOUD SUPPORT FOR LARGE SCALE ANALYTICS AND WEB PROCESSING
Overview

- Google App Engine (GAE)
- GAE Analytics Libraries
- AppScale
- Hybrid
  - Data synchronization
  - Hive Analytics
- Evaluation
- Conclusion
Google App Engine
Google App Engine

- **Platform-as-a-Service**
  - Developers focus on their applications
  - Test locally then deploy on Google’s infrastructure

- **Language Support**
  - Python 2.5 and 2.7
  - Java
  - Go

- **500,000 existing apps**

- **Auto-scaling, pay-as-you-go**
  - Web requests, background tasks, and storage
# Google App Engine APIs

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datastore</td>
<td>Key/Value object storage</td>
</tr>
<tr>
<td>Memcache</td>
<td>Distributed caching service</td>
</tr>
<tr>
<td>Blobstore</td>
<td>Storage of large files</td>
</tr>
<tr>
<td>Channel</td>
<td>Long lived JavaScript connections</td>
</tr>
<tr>
<td>Images</td>
<td>Simple image manipulation</td>
</tr>
<tr>
<td>Mail</td>
<td>Sending and receiving email</td>
</tr>
<tr>
<td>Users</td>
<td>Login service with Google Accounts</td>
</tr>
<tr>
<td>Task Queues</td>
<td>Background tasks</td>
</tr>
<tr>
<td>URL Fetch</td>
<td>Resource fetching with HTTP request</td>
</tr>
<tr>
<td>XMPP</td>
<td>XMPP-compatible messaging service</td>
</tr>
</tbody>
</table>
Google App Engine

- Restrictions to enforce scalability and security
  - Limited query support
  - Runtime restrictions
  - No socket access, no file system access
  - White-list of libraries
Data Analytics in GAE
Task Queues

- Background tasks of 10 minutes
  - `taskqueue.add(url=’/path/to/my/worker’)`
- Task names to prevent fork bombs
- Task Queue Chaining
  - Splitting up large background jobs is the burden of the developer
- GAE Analytics libraries abstract away the Task Queue (TQ)
 Fantasm

- Based on the Task Queue
  - Uses memcache and DB to manage state
- State machine driven
  - Specified in YAML
- Iterate over a large dataset
- Fan-in to join data
Pipeline

- Task Queue based
- Chains tasks into a workflow

```python
class Add(pipeline.Pipeline):
    def run(self, a, b):
        return a + b

class Multiply(pipeline.Pipeline):
    def run(self, a, b):
        return a * b

class LinearFunc(pipeline.Pipeline):
    def run(self, x, slope=1, offset=0):
        # y = m*x + b
        mx = yield Multiply(x, slope)
        yield Add(mx, offset)
```
GAE MapReduce

- Built on top of GAE infrastructure
  - Not Google’s internal MapReduce or Hadoop
- Parallel processing and reductions on large datasets
- Map across a particular type of object
  - Must scan the entire type (no subsets)
- Multiple MR jobs can be linked with Pipeline
AppScale

- Private PaaS with GAE API compatibility
  - Application portability
  - Engenders a developer community
- Distributed and fault tolerant API implementations
  - Leverages open source and new software systems
- Supports Python, Java, and Go languages
- Infrastructure agnostic
  - One virtual machine with all components
  - KVM, Xen, EC2, Eucalyptus, Openstack, etc
- Datastore agnostic
  - Cassandra, HBase, Hypertable, etc
The AppScale Stack

<table>
<thead>
<tr>
<th>Python GAE Server</th>
<th>Java GAE Server</th>
<th>Go GAE Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>eJabberD (xmpp &amp; channel API)</td>
<td>Blobstore server</td>
<td>memcacheD</td>
</tr>
<tr>
<td>Routing (HAProxy and Nginx)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AppController</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cassandra</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
GAE Analytics in AppScale

- Task Queue with RabbitMQ
Analytic Libraries in GAE

- Learning curve for libraries
- Analytics must be part of application code
  - Can introduce bugs
  - Can disrupt the user experience
  - May require significant code and time to implement
AppScale and GAE Hybrid
Hybrid Solution

- Connect GAE to AppScale
  - Scale of GAE for OTAP
  - Flexibility of AppScale for OLAP
- Contributions
  - Datastore mirroring
  - Hive queries
Datastore Library

- Datastore library for asynchronous updates
  - Best Effort (BE) with async URL Fetch
  - Eventual Consistency (EC) with Task Queue
  - No transaction support
Hive Support in AppScale

- Run SQL statements which translate to Hadoop MR jobs
- We provide the mapping interface from GAE data to Hive queries

Data Scientist
## AppScale Stack with Hive Support

<table>
<thead>
<tr>
<th>Python GAE Server</th>
<th>Java GAE Server</th>
<th>Go GAE Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>eJabberD (xmpp &amp; channel API)</td>
<td>Blobstore server</td>
<td>memcached</td>
</tr>
<tr>
<td><strong>Routing (HAProxy and Nginx)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AppController</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hive</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hadoop</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CassandraFS</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Hive Queries

- Simple, short, and fast on-demand queries
- Offline processing of online data
- Does not impact user experience
- Does not introduce bugs or code bloat
- No ETL, data processing in-place

```
SELECT COUNT(*) FROM appid_kind;
```
Measurements
Cross Cloud Measurements

Async -> GAE -> TaskQueue

EC2 Region A

EC2 Region B
Cross Cloud Data Transfer

Round-trip Time and Bandwidth Between a GAE Application and Different EC2 Regions
Cross Cloud Task Queue Delay

Round-trip time from multiple regions to a deployed GAE application with task queue delay
### Analytic Execution Time

<table>
<thead>
<tr>
<th>Operation</th>
<th>Fantasm (GAE)</th>
<th>Pipeline (GAE)</th>
<th>MapReduce (GAE)</th>
<th>Hive (AppScale)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregate</td>
<td>11334.59</td>
<td>98.34</td>
<td>377.70</td>
<td>20.94</td>
</tr>
<tr>
<td>Grep</td>
<td>10360.40</td>
<td>98.89</td>
<td>227.57</td>
<td>10.69</td>
</tr>
<tr>
<td>Join</td>
<td>10147.75</td>
<td>159.96</td>
<td>256.40</td>
<td>23.41</td>
</tr>
<tr>
<td>Subset</td>
<td>78.28</td>
<td>3.81</td>
<td>237.75</td>
<td>20.66</td>
</tr>
<tr>
<td>Wordcount</td>
<td>10977.50</td>
<td>222.14</td>
<td>840.71</td>
<td>21.54</td>
</tr>
</tbody>
</table>

Processing 100K Entities (5 trials)
Cost Analysis

- Data synchronization
  - Bandwidth: $0.12/GB
  - Data backup

- Benefits
  - Programmer productivity
  - Ad-hoc analysis
Related Work

- Private PaaS offerings
  - Red Hat’s OpenShift
  - VMWare’s CloudFoundry

- GAE Compatible
  - TyphoonAE

- MapReduce
  - Amazon Elastic MapReduce
  - Mesos framework
Thanks

Check out AppScale at:

appscale.cs.ucsb.edu