Quick Overview

- Objectively compare and contrast Google Compute Engine with Amazon’s EC2
- Using scripts in Linux, and some common benchmarks, provide statistics detailing the performance
- Compare the results between Google Compute Engine and Amazon’s EC2 and determine the highest performing and cost effective option that will suit the needs of a large range of workloads
  - Bag of tasks
  - No workflows at this point
Researched various instance types
Became familiar with Amazon EC2 dashboard, and how to easily set up new instances
Used ‘fog’ to automate creating instances and running benchmarks to make the whole process faster and more effective
Using Putty (ssh), connected to Amazon EC2 instance in order to run commands
Installed and ran most of the benchmark software needed
  - Fixed compiling and run-time errors
List of Benchmarks

- CacheBench – memory
- LmBench – bandwidth and latency
- Bonnie++ – hard drive and file system
- Phoronix Test Suite – large collection of benchmarks
- Crafty – Chess Engine for CPU performance
- dcreaw – times RAW to PPM image conversion
- eSpeak – Speech Synthesizer
- HMMer – Timed HMMer Search
- John the Ripper – Password Cracker
- OpenSSL – speed test
- Sudokut – Sudoku Solver
### Amazon EC2

<table>
<thead>
<tr>
<th>Instance Type</th>
<th>Cores</th>
<th>Memory (GB)</th>
<th>Local Disk (GB)</th>
<th>Price ($/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1.small</td>
<td>1</td>
<td>1.7</td>
<td>160</td>
<td>0.06</td>
</tr>
<tr>
<td>m1.large</td>
<td>2</td>
<td>7.5</td>
<td>850</td>
<td>0.24</td>
</tr>
<tr>
<td>m1.xlarge</td>
<td>4</td>
<td>15</td>
<td>1690</td>
<td>0.48</td>
</tr>
<tr>
<td>c1.medium</td>
<td>2</td>
<td>1.7</td>
<td>350</td>
<td>0.145</td>
</tr>
<tr>
<td>c1.xlarge</td>
<td>8</td>
<td>7</td>
<td>1690</td>
<td>0.58</td>
</tr>
</tbody>
</table>

### Google Compute Engine

<table>
<thead>
<tr>
<th>Instance Type</th>
<th>Cores</th>
<th>Memory (GB)</th>
<th>Local Disk (GB)</th>
<th>Price ($/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1-standard-1-d</td>
<td>1</td>
<td>3.75</td>
<td>420</td>
<td>0.132</td>
</tr>
<tr>
<td>n1-standard-2-d</td>
<td>2</td>
<td>7.5</td>
<td>870</td>
<td>0.265</td>
</tr>
<tr>
<td>n1-standard-4-d</td>
<td>4</td>
<td>15</td>
<td>1770</td>
<td>0.53</td>
</tr>
<tr>
<td>n1-highcpu-2-d</td>
<td>2</td>
<td>1.80GB</td>
<td>870GB</td>
<td>$0.16</td>
</tr>
<tr>
<td>n1-highcpu-8-d</td>
<td>8</td>
<td>7.20GB</td>
<td>2 x 1770GB</td>
<td>$0.65</td>
</tr>
</tbody>
</table>
You are using the following Amazon EC2 resources in the US West (Oregon) region:

- 1 Running Instance
- 1 Volume
- 3 Key Pairs
- 0 Placement Groups
- 0 Elastic IPs
- 2 Snapshots
- 0 Load Balancers
- 2 Security Groups

Optimize your resources' cost, performance and security with AWS Trusted Advisor

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance

Note: Your instances will launch in the US West (Oregon) region

Service Health

Service Status:

- US West (Oregon):
  This service is operating normally

Availability Zone Status:

- us-west-2a:
  Availability zone is operating normally
- us-west-2b:
  Availability zone is operating normally
- us-west-2c:
  Availability zone is operating normally

Scheduled Events

US West (Oregon):
No events
**t1.micro Instance**

Sudokut 0.4:
pts/sudokut-1.0.0
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 4 Minutes
  Started Run 1 @ 17:06:50
  Started Run 2 @ 17:10:15
  Started Run 3 @ 17:16:06 [Std. Dev: 29.34%]
  Started Run 4 @ 17:20:07 [Std. Dev: 25.78%]
  Started Run 5 @ 17:25:43 [Std. Dev: 22.82%]
  Started Run 6 @ 17:30:08 [Std. Dev: 20.49%]

Test Results:
  200.85880494118
  349.38766288757
  238.80797696114
  334.39287686348
  262.40996313095
  305.86973690987

Average: 281.95 Seconds

---

**m1.large Instance**

Sudokut 0.4:
pts/sudokut-1.0.0
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 29 Minutes
  Started Run 1 @ 16:23:50
  Started Run 2 @ 16:24:38
  Started Run 3 @ 16:25:23 [Std. Dev: 0.84%]
  Started Run 4 @ 16:26:00
  Started Run 5 @ 16:26:35
  Started Run 6 @ 16:27:10

Test Results:
  44.265406131744
  43.555793046951
  43.728178977966

Average: 43.85 Seconds

---

*using a t1.micro instance*
#!/usr/bin/ruby
require 'rubygems'
require 'fog'

# create a connection
connection = Fog::Compute.new(
  :provider => 'AWS',
  :aws_access_key_id => 'AKIAJ5X5MN54Z777IE3A',
  :aws_secret_access_key => 'GRjDOMpgRKpzqHyjM6hzXVhuHlAv2u4IlM3hhhwUN'
)

# create a new instance
server = connection.servers.create

# connect to instance
connection.servers

# determine wait time of bootup
server.wait_for { ready? }
Future Work...

- Gather all data
  - Using same image with all the installed benchmarks
  - Run all 10 benchmarks approximately 10 times on each instance (10 instances total)
  - That’s approximately 1000 individual tests!

- Provide statistics over all the data

- Evaluate the statistics and data between Google Compute Engine and Amazon EC2 to determine highest performing and cost effective option that will suite the needs of a large range of workloads