Collecting, cataloguing and searching performance information of Cloud resources.

Olaf Elzinga
<table>
<thead>
<tr>
<th>Price</th>
<th>Memory</th>
<th>Processor</th>
<th>Disk</th>
<th>Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>$5/mo</td>
<td>512MB</td>
<td>1 Core</td>
<td>20GB</td>
<td>1TB</td>
</tr>
<tr>
<td>$10/mo</td>
<td>1GB</td>
<td>1 Core</td>
<td>30GB</td>
<td>2TB</td>
</tr>
<tr>
<td>$20/mo</td>
<td>2GB</td>
<td>2 Core</td>
<td>40GB</td>
<td>3TB</td>
</tr>
<tr>
<td>$40/mo</td>
<td>4GB</td>
<td>2 Core</td>
<td>60GB</td>
<td>4TB</td>
</tr>
<tr>
<td>$80/mo</td>
<td>8GB</td>
<td>4 Core</td>
<td>80GB</td>
<td>5TB</td>
</tr>
</tbody>
</table>

Pricing in USD. Excludes any applicable tax.

Source: https://www.digitalocean.com/pricing/
Research question

How can an automated cloud benchmark tool test any given application component to maintain a cloud performance catalogue?
State of the art review

Requirements for the automatic cloud benchmark tools:

- Publicly available
- Open-source
- Maintained
- Support for private and public IaaS providers
## Related work

<table>
<thead>
<tr>
<th></th>
<th>Custom benchmarks</th>
<th>Schedule</th>
<th>Provider support</th>
<th>Catalogue result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud WorkBench [1]</td>
<td>Yes</td>
<td>Yes</td>
<td>Only public</td>
<td>No</td>
</tr>
<tr>
<td>CloudBench [2]</td>
<td>No</td>
<td>No</td>
<td>Public and private</td>
<td>No</td>
</tr>
</tbody>
</table>


Technical gaps

- Catalogue the collected results
- Ability to add providers
- Possibility to add custom benchmarks
Requirements

Requirements for the users:

- Easy to use
- Fully automatic and possible to scheduling benchmarks
- Custom benchmarks to test different performance attributes
- Catalogue results

Requirements for developers:

- Modular in design
Cloud Performance Collector
Cloud Performance Collector: modules

● Provider module
  ○ Provision VM
  ○ Release VM

● Deploy and run module
  ○ Installing, configuring and running the benchmarks

● Result module
  ○ Parse the useful parts of the benchmark output
Cloud Performance Collector: workflow

User

Cloud Performance Collector

Cloud provider API

VM

Cloud catalogue

Test scenario 1

provision VM

ok (IP)
deploy software

software is successfully deployed
run benchmark
results

release VM

successfully destroyed

filter result

Catalogue
Cloud Performance Collector: prototype

- CLI
- Provider modules written in bash
- Installing, configuring and running the benchmarks via Ansible [1]
- Benchmarks as Dockerfile
- Scheduling via crontab

Execution example:

- bash modules/providers/geni/geni nictaXL

[1] https://www.ansible.com
Experimental setup

ExoGeni:

- University of Amsterdam (UvA)
- National ICT Australia (NICTA)
- Raytheon BBN Technologies (BBN)
### Experimental setup: ExoGeni resources

<table>
<thead>
<tr>
<th>Type</th>
<th>CPU</th>
<th>Memory</th>
<th>SSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>1 vCPU</td>
<td>3 GB</td>
<td>25 GB</td>
</tr>
<tr>
<td>L</td>
<td>2 vCPU</td>
<td>6 GB</td>
<td>50 GB</td>
</tr>
<tr>
<td>XL</td>
<td>3 vCPU</td>
<td>12 GB</td>
<td>100 GB</td>
</tr>
</tbody>
</table>
Experimental setup: questions

- Will VM instances with the same specifications and image from the same provider give similar performance?
- Will the same VM instance with the same workload provide a constant level of performance over time?
- Will a given application component perform the same compared to the synthetic benchmarks?
Experiment 1:

- Measure the difference in performance between different VMs with the same image
- Using a different VM instance every 2 hours
- Measured 24 times (every hour)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Component</th>
<th>Metrics</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sysbench</td>
<td>CPU</td>
<td>Calculate the primeness of 100,000 numbers</td>
<td>Duration (sec)</td>
</tr>
<tr>
<td>Stream</td>
<td>Memory</td>
<td>Triad $A[i] = B[i] + \text{scalar} \times C[i]$</td>
<td>Throughput MB/s</td>
</tr>
<tr>
<td>iozone</td>
<td>Disk</td>
<td>Read and write 64Kb using a file of 2GB</td>
<td>Throughput MB/s</td>
</tr>
</tbody>
</table>
Experiment 1: results CPU (sysbench)
Experiment 1: results memory (STREAM)
Experiment 1: results disk I/O read and write (iozone)
Experiment 2:

- Using the same VM instance for every benchmark
- Use the same benchmark tools as experiment 1
- Measured 24 times (every hour)
Experiment 2: results CPU (sysbench) & memory (STREAM)
Experiment 2: results disk I/O read and write (iozone)
Experiment 3:

- Using docker container with the application Montage
  - An astronomical image mosaic engine
- Measuring how long it takes to create the astronomical image
- Measured 24 times (every hour)
Experiment 3: results Montage
Conclusion

- Performance can vary between different VMs within an ExoGeni rack
- The same VM instance perform similar over time
- Largest instance is not always the right choice
- Problems provisioning VMs and suddenly were unreachable (UvA rack)
Future work

- Test it with a larger amount of applications
- Test the network performance of resources
- Design the cloud catalogue
Questions?