Cloud WorkBench

Infrastructure-as-Code Based Cloud Benchmarking

Joel Scheuner, Philipp Leitner, Jürgen Cito, Harald Gall

University of Zurich, Switzerland
{lastname}@ifi.uzh.ch
Motivation (1)

Infrastructure-as-a-Service (IaaS)
Motivation (1)

Infrastructure-as-a-Service (IaaS)

Google Compute Engine

Amazon Web Services EC2

Microsoft Azure

GoGrid

Rackspace

Joyent
Motivation (1)

Infrastructure-as-a-Service (IaaS)

>22 Instance Types

Google Compute Engine

Amazon EC2

Microsoft Azure

GOGRID

Rackspace

Joyent
Performance Variations

Figure 6: Disk Write Bandwidth (KB/Sec)

Motivation (2)

Performance Variations

BlueLock 4.47E+03
ElasticHosts 1.16E+03
Amazon AWS 2.54E+04

Figure 6: Disk Write Bandwidth (KB/Sec)


Even for services with the same specification!
Motivation (3)

• Benchmark (i.e. performance test)

• Problems
  • Time-consuming
  • Error-prone

Define

Prepare

Execute
Cloud Experiment Automation


Expertus  (3) D. Jayasinghe, J. Kimball, S. Choudhary, T. Zhu, and C. Pu. **An Automated Approach to Create, Store, and Analyze Large-scale Experimental Data in Clouds.** In 14th IEEE Int. Conf. on Information Reuse and Integration (IRI), pp. 357–364, August 2013.
Related Work

Cloud Experiment Automation


Our approach: Provisioning via Infrastructure-as-Code
How can existing IaaS cloud benchmarks be described in a modular and portable manner?
Research Questions

I
How can existing IaaS cloud benchmarks be described in a modular and portable manner?

II
How can such benchmarks be periodically scheduled and executed in cloud environments in a fully reproducible way, and without manual interaction?
Benchmark Definition
Benchmark Definition

Cloud VM Configuration

Benchmark Definition * 1..* Cloud VM Configuration

```ruby
config.vm.provider :aws do |aws|
  aws.region = "eu-west-1"
  aws.ami = "ami-896c96fe"
  aws.instance_type = "t1.micro"
end
```
Benchmark Definition

```
config.vm.provider :aws do |
  aws.region = "eu-west-1"
  aws.ami = "ami-896c96fe"
  aws.instance_type = "t1.micro"
end
```

# Update package index
include_recipe 'apt'
# Install benchmark via package manager
package 'sysbench'
Architecture Overview

Experimentation

Upload Configuration

Provisioning Service
Architecture Overview

Experimenter

Cloud WorkBench Server

Configure and Execute Benchmark

Upload Configuration

Manage VMs

Provider API

IaaS Providers

Provisioning Service
Architecture Overview

Cloud WorkBench Server

IaaS Providers

Provisioning Service

Experimenter

Configure and Execute Benchmark

Upload Configuration

Manage VMs

Orchestrate Execution

Submit Results

Fetch Configuration
Implementation (1)
Implementation (1.1)

```ruby
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.provider :google do |google, override|
    google.image = "debian-7-wheezy-v20140619"
    google.machine_type = "n1-standard-1"
    google.zone = "europe-west1-a"
  end

  config.vm.provision "chef_client", id: "chef_client" do |chef|
    chef.add_recipe "fio-benchmark@0.3.0"
    chef.json = {
      "fio" => {
        "metric_definition_id" => "seq. write",
        "config" => {
          "size" => "10m",
          "refill_buffers" => "1"
        }
      }
    }
  end
end
```
Implementation (1.3)

Schedule

15 5,17 ***

— 15 minutes past 5am and 5pm of every day

Metrics

seq. write
KB/s (ratio)

cpu
model name (nominal)
Implementation (2)
Implementation (2.1)
Case Study

Sequential Disk Write Speed

- m1.small + SSD Storage
- m1.small + HDD Storage
- t1.micro + SSD Storage
- t1.micro + HDD Storage

Graph showing the sequential disk write speed over time for different storage configurations.
Conclusion

- Cloud WorkBench: Open source cloud experiment automation tool
- Infrastructure-as-Code for cloud benchmarking
- Easily configurable
- Reproducibly executable
Cloud WorkBench

Open Source

https://github.com/sealuzh/cloud-workbench

Benchmark Execution

1. **Trigger Execution**
   - CWB Server
   - Acquire Resources
   - Provision VM

2. **Fetch VM Configurations**
   - Cloud VM
   - Apply VM Configurations

3. **VM Provisioning Completed**
   - Provider API
   - Start Benchmark Run
   - Notify Benchmark Completed

4. **Run Benchmark**
   - Cloud VM
   - Postprocess Results

5. **Submit Metric(s)**
   - Provider API

6. **Notify Postprocessing Completed**
   - CWB Server

7. **Release Resources**
   - Experimenter / Scheduler
Future Work

- Support additional cloud providers
  (Currently: Amazon EC2 + Google Compute Engine)

- Integrate statistical analysis capabilities

- Support the entire benchmarking lifecycle via a single web tool
  (define, execute, analyze)