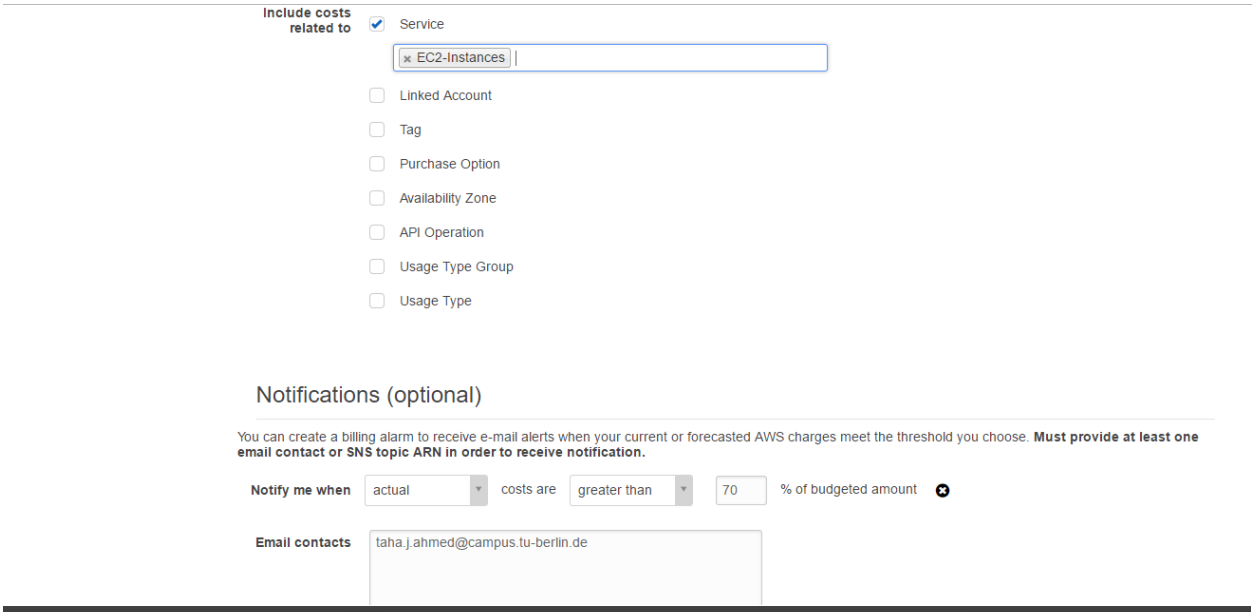
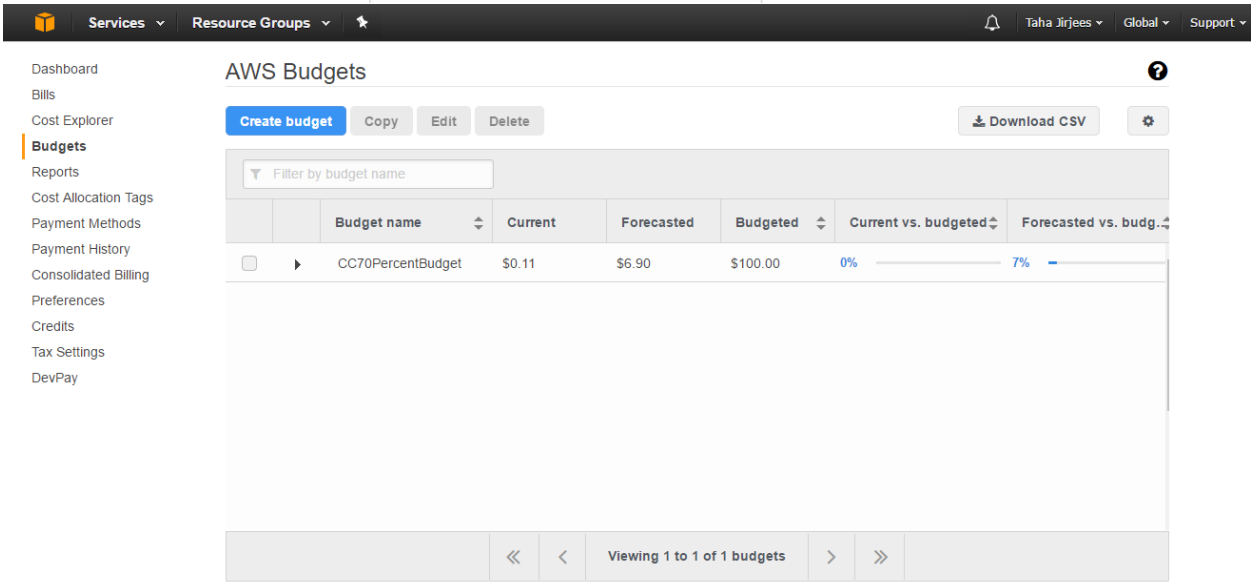
Clound Computing Assignment 1

Arif Gömleksiz, Muhammad Talal Saleem, Musadiq Raees, Syed Muhammad Hani, Taha Jirjees Ahmed

AWS Budget Screenshots

AWS Budget Screenshots to follow:





AWS Instance Preparation and Launch

```
### Configure User Access

aws configure
AWS Access Key ID [None]: **********************************
AWS Secret Access Key [None]: **********************************
Default region name [None]: eu-central-1
Default output format [None]: json

### Create Security Group

aws ec2 create-security-group --group-name devenv-sg
--description "security group for development environment in EC2"


### Authorize ssh access/ security group ingress

aws ec2 authorize-security-group-ingress --group-name devenv-sg
--protocol tcp --port 22 --cidr 0.0.0.0/22

### Create KeyPair

aws ec2 create-key-pair --key-name devenv-key
--query 'KeyMaterial' --output text > devenv-key.pem



### Launch Instance

aws ec2 run-instances --image-id ami-ba68bad5
--security-group-ids sg-d082f3bb --count 1
--instance-type m3.medium --key-name devenv-key
--query 'Instances[0].InstanceId'
```

OpenStack VM Preparation and Launch

Loading Environment Variable

```
#!/bin/bash
export OS_USERNAME=cc17-group45
export OS_PASSWORD=EbLlxp2BtV
export OS_TENANT_NAME=cc17-group45
export OS_AUTH_URL=http://cloud.cit.tu-berlin.de:5000/v2.0
```

Launch VM

```
### Call Environemnt Variables
```

```
source openrc
//Add key

ssh-keygen -t rsa -b 4096 -C "y"
openstack keypair create --public-key ~/CC-1/CCkey.pem.pub CCkey

### Adding ICMP and SSH rule

openstack security group rule create default --protocol tcp
--dst-port 22:22 --remote-ip 0.0.0.0/0
openstack security group rule create --protocol icmp default

### Launch VM Instance
openstack server create --flavor 604de11c-3222-4902-8523-11cc61b5b485
--image 11f6b8aa-31df-4b66-8b42-5ee9760c47ba
--key-name CCKey --security-group default CC_test

### Adding floating ip

openstack floating ip list
openstack server add floating ip CC_test 10.200.1.204

# SSH Instance
ssh -i ~/Downloads/cloud.key ubuntu@10.200.1.204
```

Performance Benchmarks

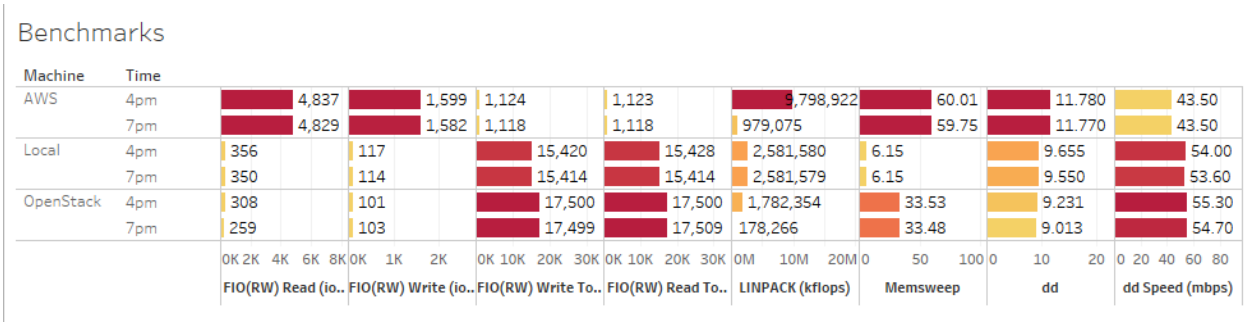


Figure 1: Benchmark Scores

Disk Benchmarks

Q) Look at the disk measurements. Are they consistent with your expectations. If not, what could be the reason?

We expected I/O operations in paravirtualized to be slower than full virtualization or dedicated accessed systems. The measurements are consistent with the understanding as OpenStack performs better than AWS and local 500HDD is better then paravirtualized systems.

Q) Based on the comparison with the measurements on your local hard drive, what kind of storage solutions do you think the two clouds use?

AWS medium m3 instance that we used in this assignment uses 128SDD (solid state type drive) while OpenStack and local system uses HDD. This proves that for direct accessed disk copy/move operations powerful HDDs' paravirutized systems performs better than paravirtualized HDDs.

CPU Benchmark

Q) Look at linpack.sh and linpack.c and shortly describe how the benchmark works.

Linpack gives result in floating point operations per second (FLOPS). It performs linear algebra computing operations on systems.

Q) Find out what the LINPACK benchmark measures (try Google). Would you expect paravirtualization to affect the LINPACK benchmark? Why?

The Linpack Benchmark is a measure of a computer's floating-point rate of execution. It is determined by running a computer program that solves a dense system of linear equations.The linpack benchmark lists the performance in Mflop/s of a number of computer systems.Since these operations require dedicated access to host's resources paravirtualization will have a negative effect on performance of virtualized systems.

Q) Look at your LINPACK measurements. Are they consistent with your expectations? If not, what could be the reason?

Linpack measurements are consistent with our understanding that locally it performs better than from virtualized systems. Local system with the following configuration performed 200% better then medium AWS and openstack instances: Intel(R) CoreT i7-4500U CPU @ 1.80GHz

Memory Benchmark

Q) Find out how the memsweep benchmark works by looking at the shell script and the C code. Would you expect virtualization to affect the memsweep benchmark? Why?

Memsweep measures the required time to write and clean heap memory from different locations.The locations are chosen such that a cache miss occurs and data is loaded direct from memory. Since the hypervisor needs to communicate at HW level for memory requests we get considerably better results for memsweep in no virtualized systems

Q) Look at your memsweep measurements. Are they consistent with your expectations. If not, what could be the reason?

The measurements verify that locally the time for memsweep is considerably lower than paravirtualized machines. These measurements show that paravirtualization affects memory access time.

Benchmarking Methodologies

dd:

```
#!/bin/sh touch testfile.txt,results.txt echo "Sequential Write"
echo "Calculating throughput" >> results.txt dd
if=/dev/zero of=~/testfile.txt bs=512MB count=1 oflag=direct
conv=fdatasync 2>> results.txt
echo "Calculating Latency" >> results.txt
dd if=/dev/zero of=~/testfile.txt bs=512MB count=1000 oflag=dsync
2>> results.txt
```

fio:

```
#!/usr/bin/env bash echo "Random Read and Write" fio --randrepeat=1
--ioengine=libaio --direct=1 --gtod_reduce=1 --name=test
--filename=test --bs=4k --iodepth=64 --size=28M
--readwrite=randrw --rwmixread=75 --output=outputRW.txt
echo "Random Read" fio --randrepeat=1
--ioengine=libaio --direct=1 --gtod_reduce=1
--name=test1 --filename=test1 --bs=4k --iodepth=64
--size=28M --readwrite=randread
--output=outputR.txt echo "Random Write"
fio --randrepeat=1 --ioengine=libaio
--direct=1 --gtod_reduce=1 --name=test2
--filename=test2 --bs=4k --iodepth=64 --size=28M
--readwrite=randwrite --output=outputW.txt
```