Cloud Computing
Winter Term 2018/2019

Practical Assignment No. 1

Due: 29.11.2018 23:59

The primary goal of this assignment is to gain insight into performance characteristics of virtualization by benchmarking different cloud providers and comparing and evaluating the benchmark results. The secondary goal is to familiarize yourself with Amazon EC2 and Google Cloud, and the respective tools as an example of today's Infrastructure-as-a-Service clouds.

1. Prerequisites

1.1. Student groups
- We assume you already formed groups of 3-4 students on the course website in ISIS (link in the "Assignments" section).

1.2. Amazon AWS Educate
Create and Amazon AWS Educate Starter account in order to work with the cloud platform and perform further tasks.

- Go to https://www.awseducate.com/registration and go through the registration process. **Important**:
  - On Step 2, enter your TU Berlin student email address
  - On Step 3, select the bottom checkbox for a "AWS Educate Starter Account"
  - You will receive a confirmation email, follow the link in the email and finish the application process.
- The application process usually completes within a few hours, but might take longer in individual cases. Since only one account is required per group, all group members can attempt to register in order to speed up the process.
- AWS Documentation: Using the Amazon EC2 online documentation you can find out how to setup and use the API and AMI tools

Note: You are free to create a full Amazon AWS Educate account (not Starter). This will require a credit card number, which might be charged in the case that you use up the 100$ credit that you receive per semester. In return, you will gain access to many additional features and resources that are quite interesting to play with! We recommend you try out these features, however in this case it will be ESPECIALLY IMPORTANT to **shut down your VMs** when you are done working with them, otherwise your credit card might be charged!

1.3. Google Cloud

- Create at GCP account at https://cloud.google.com/
- Check your ISIS group forum for a GCP coupon code and redeem it at: https://console.cloud.google.com/billing/redeem
- Google requires you to enter Credit Card details, which is only used to verify that you are a real person
- You can optionally check if you are eligible for additional free trial credit at https://cloud.google.com/free/
- Study the GCP documentation and learn how to use the dashboard and command line tools

Note: Although the AWS Educate Starter account is free and you have coupons for the Google Cloud Platform, you might run out of credits. Remember to **shut down your VMs when you do not use them!**

2. Virtual Machines

- Prepare one VM instance in EC2 and GCE, respectively, which you can access remotely and where you can execute benchmarks (see below).
- Use the *command line tools* of the respective platform to create the instance and all required resources (keypair, security group, etc). Make sure to keep a listing of the shell commands you execute, as they are part of your submission (see section „Submission Deliverables"). Remember **not** to include any private information such as your access key or secret key in your submission (replace them with dummy strings).
- At least the following steps are required:
  1. Configure CLI tools (exclude the commands for this in the submission)
  2. Generate a local SSH key pair
  3. Upload the public key (gcloud requires manual modifications)
  4. Make sure the VM is accessible via ICMP and SSH (configure security groups in AWS and firewall rules in gcloud)
  5. AWS only: create a subnet, and optionally a VPC
  6. Create the instance
  7. Ping the public IP of your instance (exclude from submission)
  8. Connect to your instance via SSH (exclude from submission)

Hints:
- Shut down the virtual machine until you start working on the assignments below
- We suggest you use one of Amazon's official Linux images, but if you want to use a different one you may do so. Amazon Linux is based on RedHat Linux

Outputs:
- Files **listing-ec2.txt** and **listing-google.txt**
  - Containing all commands you used for preparing your VMs, including comments explaining what the commands do

3. Prepare Performance Benchmarks

Write benchmarking scripts for the main system resources CPU, memory and disk.

- The scripts take no parameters and each script must run for 10-20 seconds while benchmarking the respective system resource.
- In case you obtain multiple measurements during that time, use the median of those values as the final output.
- Except for the disk benchmarks, your scripts are not allowed to use external tools (exception: `gcc`). All files required by your benchmarks must be submitted together with the scripts.
- Afterwards the script must output one single floating point value (e.g. 14.533)
- The interpretation of the resulting value depends on the benchmarked resource and used tool.

CPU Benchmark
LINPACK is a popular CPU benchmark in the supercomputing community. Derive the implementation of your benchmark script for CPU compute power from the `linpack.sh` script provided on ISIS.

Memory Benchmark
Create a benchmarking script for memory operations based on the `memsweep.sh` script provided on ISIS.

Disk Benchmarks
Benchmark sequential (Bytes/s) and random (Operations/s) read and write access to the virtualized disk. For sequential read/write use the `dd` tool, for random access use the `fio` tool. Install these tools on your VMs.

Outputs:
- Scripts:
    - `measure-cpu.sh`
    - `measure-mem.sh`
    - `measure-disk-random.sh`
    - `measure-disk-sequential.sh`
- All files required by your scripts

4. Execute Performance Benchmarks

Execute the prepared benchmarks on the following two Cloud platforms. Each benchmark must be executed once every hour for two days. This results in:

- 2 VMs * 48 hours * 4 benchmarks = 384 total measurements

You can implement the regular invocation of the benchmark any way you like, using programming languages like Python or techniques like `cron`. The result of the regular invocations must be 4 CSV (comma separated values) files (one per benchmark) that contain the collected values and the Unix timestamp of when each measurement was done.

Example of such a CSV file:

```
time,value
1542717365,677.4445
1542717366,746.3566
1542717367,144.6778
1542717368,143.5634
1542717369,435.7880
...
```

After taking all benchmarks, look at your results and answer the questions below. Plotting the results (e.g. with Excel) might be helpful.

Hint:
- When executing long-running experiments in a public cloud, remember to keep an eye on your credits. You might have to shut down your VM between repeated measurements. Use the Amazon/Google APIs to start and stop your instances automatically.

Outputs:
- Eight CSV files:
  - `[ec2|google]-[cpu|mem|disk-sequential|disk-random].csv`
  - For the two platforms, covering the 4 benchmarks:
    - sequential read/write Bytes/s
    - random read/write operations
    - CPU benchmark
    - Memory benchmark
- Text file `answers.txt`
  - Including all the answers for the questions below. Please use max. 200 words per question

CPU benchmark questions:
1. Look at `linpack.sh` and `linpack.c` and shortly describe how the benchmark works.
2. Find out what the LINPACK benchmark measures (try Google). Would you expect paravirtualization to affect the LINPACK benchmark? Why?
3. Look at your LINPACK measurements. Are they consistent with your expectations? If not, what could be the reason?

Memory benchmark questions:
1. 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?
2. Look at your memsweep measurements. Are they consistent with your expectations? If not, what could be the reason?

Disk benchmark questions:
1. Look at the disk measurements. Are they consistent with your expectations? If not, what could be the reason?
2. Compare the results for the two operations (sequential, random). What are reasons for the differences?

5. Submission Deliverables

Use the OpenSubmit system running at http://wally175.cit.tu-berlin.de to submit your solutions. Logging into the OpenSubmit platform requires a Gitlab account at https://gitlab.tubit.tu-berlin.de, which you can access with your TUB account. Every group member must register on the OpenSubmit platform, but only one member should submit the final solution and select the other group members as co-authors. Your solution will be validated automatically, you can resubmit your solution multiple times until it passes the tests.

Expected contents of your solution zip file:

- `listing-[amazon|google].txt`
  - **2 listing files, 10 points each**
- `answers.txt`
  - **7 points**
- `measure-[cpu|mem|disk-sequential|disk-random].sh`
  - **4 script files, 6 points each**
- `[ec2|google]-[cpu|mem|disk-sequential|disk-random].csv`
  - **8 CSV files, 3 points each**
- `Any additional files required by your benchmark scripts`

Total points: 75

Final Warning: Please make sure your VMs are shut down when you are not using them! :-)