Cloud Computing

Winter Term 2023/2024

Practical Assignment No. 1

Due: 19.11.2023 11:55 pm

For this assignment, you will explore a cloud computing system from a user perspective. The primary goal is to gain insight into performance characteristics of virtualization by benchmarking a cloud provider, different virtualization and isolation techniques, and comparing and evaluating the benchmark results, in the process familiarizing yourself with Google Cloud and the respective tools as an example of today's Infrastructure-as-a-Service clouds. The secondary goal is to gain an understanding of isolation features provided by modern Linux kernels.

1. Prerequisites

1.1. Student groups

We assume you already formed groups of 5 students on the course website in ISIS.

1.2. Unix Platform

Work on a Unix-like platform to solve the assignments. Linux is recommended, Mac OS works as well. If you use Windows privately, work in a virtual machine (either locally, such as VirtualBox, or use one of the Cloud platforms).

1.3. Google Cloud

- Visit this link to retrieve your GCP credit coupon:
  - Enter your TUB Email address
  - Each student can retrieve one coupon
  - Each coupon is worth 50$ and valid until 17.10.2024
  - You must redeem your coupon until 17.02.2024 (thereafter they expire)
  - Since you are working in groups: We recommend not redeeming all coupons at once. Use one up (as group) and redeem the next if required.
  - **Do not spend all google cloud credits on this assignment. You will need them for the following assignments as well.**
- You will receive an email with further instructions on how to create an account and redeem your coupon
- You can optionally check if you are eligible for additional free trial credit at https://cloud.google.com/free/
- Study the documentation of the Google Cloud SDK and learn how to set up and use the command line tools
- Configure your Google Cloud SDK that it can access your GCP account, i.e. commands like the following should work:

- `gcloud compute instances list`

Note: You might run out of credits. To prevent this situation, do not redeem all credits at once and remember to **shut down your VMs when you do not use them!**

Do not spend all google cloud credits on this assignment. You will need them for the following assignments as well.

2. Virtual Machines

Write a well-commented Shell script that prepares and starts multiple virtual machines on the GCP platform. In future assignments, you will be allowed to use automation tools such as Ansible or Terraform.

- You can first manually experiment with the **`gcloud`** commands and make notes, before finally combining the commands into a script.
- Each command in the script should be commented, make sure the script is clean and not cluttered with unnecessary text.
- After writing the script, test it: delete all running VMs, run the script, and check that the result is satisfactory.
- Do **not** include any private information such as your access key or secret key in your submission (if necessary, replace them with dummy strings).
- All requirements listed below should be covered by at least one command.

General requirements:

1. Generate a local SSH key pair and name the output files `id_rsa` and `id_rsa.pub`. Make sure to specify a valid username as the comment in the public key.

Requirements for GCP:

1. Prepare a modified copy of the public key as described in the GCP documentation. Use shell commands like `echo`, `cat` and file redirection (">"). Note: the username in the prepared key must match the username specified with `ssh-keygen`.
2. Upload the public key into your project metadata.
3. Create a Firewall rule that allows incoming ICMP and SSH traffic. The rule must apply only for VMs with the tag "cc".
4. Launch three GCP instances, each with a different machine type: *n1-standard-4, n2-standard-4, c3-standard-4*. This should happen in a loop. For each instance:
   - Add the tag "cc".
   - Use image for "Ubuntu Server 22.04".
   - Enable nested virtualization (See here and/or here).
   - Resize the VM disk volume size to 100 GB.

Outputs:

- Script **create-gcp-instances.sh**
  - Containing all commands for preparing and starting your VMs, including comments explaining what the commands do.

3. Prepare Performance Benchmarks

We want to benchmark different machine types, virtualization technologies, and their characteristics. For this, we will use the `sysbench` tool to run rudimentary benchmarks on our VMs. Write a benchmarking Shell script called `benchmark.sh` that uses the `sysbench` command to perform the 4 benchmarks described below. It should initially check if the tool is already installed, and if not, install it, e.g., with:

```
sudo apt update && sudo apt install -y sysbench
```

The script must be executed without parameters and output a *single line* on the standard output that contains all measurements in CSV format, prefixed by the Unix epoch timestamp when the measurements were started:

```
1573143062,1058.52,9890.61,132.06,414.23
```

On the standard error output, the script can print arbitrary output.
The benchmarked resources are:

- CPU speed (events/s)
- Memory access (MiB/s)
  - Use block size of 4KB and a total size of 100TB (tera bytes)
- Random-access disk read speed (MiB/s)
- Sequential disk read speed (MiB/s)
  - For both disk tests, use 1 file with size 1GB, and make sure sysbench uses ***direct*** disk access to avoid caching effects.

Each benchmark must be executed for 60 seconds. The results of all 4 benchmarks must be collected in the script, then printed as one line, as shown above. Except for `sysbench`, the script is only allowed to use commands that are available in the default Ubuntu images used (don't install any additional packages with `apt` or similar tools).
Make sure that every command in your script is commented.

Hints:

- The sysbench tool outputs a lot of information that we do not need. Use typical Shell tools such as **`grep`** and **`awk`** to obtain only the necessary piece of output from each `sysbench` invocation.

Outputs:

- Script **`benchmark.sh`**

4. Prepare Virtualization Technologies

Write a shell script that prepares each GCP VM for the usage of further virtualization technologies:

1. Install QEMU (and the KVM kernel module) on each GCP VM. Study the QEMU documentation on how to work with disk images. Obtain a VM image in the `qcow2` format with Ubuntu 22.04 installed. It is recommended to obtain an Ubuntu cloud image. When working with a cloud image, you must configure cloud-init (see for example this guide and these examples). The guest OS needs to be able to access the internet and you need SSH access into the guest to exchange files and run commands. You are free to use QEMU manually or use a management tool such as Libvirt. Give the VMs access to all CPU cores of the respective host and ensure sufficient memory.
2. Install Docker on each GCP VM. Study the Docker documentation and get to know the Docker command line tool.
3. Write a Dockerfile that builds a container image based on Ubuntu 22.04 that can execute your benchmark script without any further files or other changes. When starting, the container should automatically execute the benchmark script, output the results, and exit.
4. Make sure you have the two VMs (KVM and Qemu) running idle in the background, ready to accept SSH connections. Copy the benchmark script to each VM, so that you can later on execute it via ssh.
5. At the end of your script, each GCP VM should be prepared such that a docker container can be started as well as the two guest VMs are usable. This is a precondition for the actual execution of experiments.

Outputs:

- Script **prepare-experiments.sh**
  - Prepares each GCP VM for the experiments to conduct by installing necessary virtualization technologies. It contains all required commands and sufficient code documentation.
- **Dockerfile**

5. Execute Performance Benchmarks

The final step is to execute the benchmark script on the following platforms:

- Natively on each GCP VM
- In a Docker container on each GCP VM
- Virtualization with hardware support (Qemu +KVM) on each GCP VM
- Virtualization with dynamic binary translation (Qemu) on each GCP VM

Execute the prepared benchmark script at least 48 times on each platform and GCP VM: Every 30 minutes, trigger the execution of the benchmark script on each GCP VM and its associated platforms. After 48 executions = 24 hours, gather the individual results (the results must be collected in one CSV per GCP VM and platform). In the end, each CSV file should have at least 2*24=48 entries.

Write a script `execute-experiments.sh` that automates the outlined steps. In summary, it should:

- Prepare the 4 result files by writing a CSV header into them (file names and CSV header see below)
- Run the benchmark on the native system and collect the results
- Run the Docker image and collect the results
  - Hint: do not use the --tty/-t option, since it merges the stdout and stderr streams
- Run the KVM benchmark via ssh and collect the results
- Run the Qemu benchmark via ssh and collect the results

To ease the experiment execution, use cron for automation. Use the `crontab` and `echo` commands to add a single entry to the Crontab of your user on each GCP VM. The entry should execute the `execute-experiments.sh` script and hence trigger experiment execution and result gathering.

Example of a single CSV file:

```
time,cpu,mem,diskRand,diskSeq
1573143062,1058.52,9890.61,132.06,414.23
1573143092,1057.51,9891.62,133.07,415.24
1573143122,1056.50,9892.63,134.08,416.25
...
```

Additional tasks:

- After taking all benchmarks, use the `plot.py` Python script provided on ISIS to generate a plot for your benchmarks. If the script completes without errors, your files have the correct format. The python script assumes that you saved your results in a particular format:

  `[n1|n2|c3]-[native|docker|kvm|qemu]-results.csv`

- Answer the questions below in a well-formatted, readable text.
- Add the crontab commands to the end of the `prepare-experiments.sh` script. Hint: You can envision the following overall workflow: First, we create the GCP VMs using the `create-gcp-instances.sh`. Afterward, we copy the `benchmark.sh` script and the `execute-experiments.sh` script to each GCP VM. Eventually, we execute `the prepare-experiments.sh` script, which installs the necessary tools and configures the cron jobs, thereby starting the experiment execution on each GCP VM.

Hints:

- When executing long-running experiments in a public cloud, remember to keep an eye on your credits. You should have more than enough credit to run these benchmarks, but make sure to shut down your VMs when you don't use them.

Outputs:

- Script **execute-experiments.sh**
- 1 plot generated by `plot.py`:
  - **`all-results-plot.png`**
- Text with answers to questions
  - Please write the answers into the text field within the submission of assignment 1 or submit a .txt or .pdf file with the answers. Other file formats will not be accepted.
  - Please use max. 200 words per question

CPU benchmark questions:

1. Shortly describe how sysbench performs CPU benchmark. What does the resulting events/s value represent?
2. Look at the plots of your long-term measurements. Do you see any seasonal changes?

Memory benchmark questions:

1. Shortly describe, how `sysbench` measures memory performance.
2. How would you expect virtualization to affect the memory benchmark? Why?

Disk benchmark questions:

1. Shortly describe how `sysbench` performs the disk benchmarks.
2. Compare the results for the two operations (sequential, random). What are the reasons for the differences?

General Benchmark questions:

1. Look at your benchmark results. Are they consistent with your expectations regarding the different virtualization platforms? Explain your answer. What are the main reasons for the differences between the platforms? Answer these questions for all benchmarks:
    a. CPU
    b. Memory
    c. Random disk access
    d. Sequential disk access
2. Look at your benchmark results. Are they consistent with your expectations regarding the different machine types? Explain your answer. What are the main reasons for the differences between the platforms? Answer these questions for all benchmarks:
    a. CPU
    b. Memory
    c. Random disk access
    d. Sequential disk access

5. Submission Deliverables

Submit your solution on the ISIS platform as individual files. Please submit ONLY the necessary files and use exactly the given file names!
Submit the text with answers to the questions directly in the submission text field or upload a .txt or .pdf file.

Expected submission files:

- `create-gcp-instances.sh`
  - **1 script file, 10 points**
- `benchmark.sh`
  - **1 script file, 8 points**
- `prepare-experiments.sh`
  - **1 script file, 12 points**
- `Dockerfile`
  - **1 script file, 5 points**
- `execute-experiments.sh`
  - **1 script file, 10 points**
- `all-results-plot.png`
  - **1 image file, 5 points**
- Text with answers to questions
  - **6 x 1 points (small questions), 2 * 2 points (questions)**

Total points: 60

Final Warning: Please make sure your VMs are shut down when you are not using them! Make good use of your credits :-)

Do not spend all google cloud credits on this assignment. You will need them for the following assignments as well.