Report: Homework 3 - IAAS Cloud Computing

Jan Schlenker

April 16, 2015

Instructor:	Dipl.-Ing. Dr. Simon Ostermann
Parts solved of the sheet:	Tasks 1-5
Total points:	15

1 How to run the programme

First of all extract the archive file `homework_3.tar.gz`:

```
$ tar −xzf homework_3.tar.gz
$ cd homework_3
```

Afterwards move/copy the binary files `gm` and `povray` to the `bin/` directory and the files `scherk.args`, `scherk.ini` and `scherk.pov` to the `inputdata/` directory:

```
$ cp <gm−file−path> <povray−file−path> bin/
$ cp <scherk−files−dir>/scherk∗ inputdata/
```

At last change the settings and run the main programme, e.g.:

```
$ ./main.sh 3 t2.micro
```

2 Programme explanation

The files of the the programme are structured as follows:

- The `main.sh` script just executes `launchInstances.sh` and `cloudRender.sh`
- The `launchInstances.sh` script contains the programme to launch multiple AWS instances
- The `cloudRender.sh` script contains the programme to do the povray worklflow on the AWS instances
- The `settings` file contains the settings for the main programme. These should be adjusted by the executor.
- The `bin` directory contains the binaries `povray` and `gm` which will be copied to the AWS instances

- The `inputdata` directory contains the necessary files for the `povray` binary which will be copied to the AWS instances

Below is the programme explanation task by task:

- **Task 1:** Only thing to mention here is that port 22 should be available. This can easily be done by creating a Security Group with the AWS-console and set an appropriate inbound type.
- **Task 2:** To copy the files the ssh command can be used. For that a key is necessary, which can be generated with the AWS-console as well.
- **Task 3:** The `launchInstances.sh` script basically uses the `aws ec2 run-instances` command with the `--count` flag to start as many instances as whished. The `--security-group-ids` flag is important for ssh use later on. Afterwards the povray files are copied to each instance.
- **Task 4:** The `cloudRender.sh` script reads the `scherk.ini` file to get the amount of frames and renders these frames on the different instances.
- **Task 5:** Both the `launchInstances.sh` and `cloudRender.sh` contain timestamps to measure the instance launching, execution and instance termination time.

3 Results

Measurements were made for the instance types `t2.micro`, `m3.large`, `c4.xlarge` and `c3.2xlarge`. For all measurements the image was the default amazon image, the frame number was 64 and the number of instances was 3. Table 1 shows the measurement results, where T_L represents the launching, T_E the execution and T_T the termination time.

The launching time and the termination time are nearly the same for each measurement. The execution time decreases with the number of CPUs. Of course the sequential part of the programme (e.g. copy files over network, create gif etc.) limits the speedup.

Instance type	**CPUs**	**T_L in s**	**T_E in s**	**T_T in s**	**Speedup**	**Efficency**
t2.micro	1	113,41	277,35	35,15	-	-
m3.large	2	102,31	240,10	35,41	1,16	0,58
c4.xlarge	4	109,01	139,21	35,20	1,99	0,50
c3.2xlarge	8	110,10	110,12	35,01	2,52	0,31

Table 1: Measurements