Report: Homework 5 - Advanced Programming

Jan Schlenker

May 7, 2015

Instructor:	Dipl.-Ing. Dr. Simon Ostermann
Parts solved of the sheet:	Cloud task
Programming language:	Java
Library used:	jclouds
Total points:	15

1 Requirements

- Java 1.7
- Maven

2 How to run the programme

First of all extract the archive file `homework_5.tar.gz`:

```
$ tar -xzf homework_5.tar.gz
$ cd homework_5
```

Afterwards move/copy the povray files `povray`, `gm` and `scherk.pov`to the `povray/` directory:

```
$ cp <gm-file-path> <povray-file-path> <scherk.pov-file-path> povray/
```

Now use maven to compile the sources and build a jar with dependencies:

```
$ mvn compile assembly:single
```

At last run the created jar, where `<instances>` is the number of instances which should be started and ¡frames¿ is the number of frames which should be rendered:

```
$ java -jar target/ec2-cloud-renderer-1.0.0-SNAPSHOT-jar-with-dependencies.jar <
      aws_access_key_id> <aws_secret_access_key> <instances> <frames>
```

3 Programme explanation

The files of the the programme are structured as follows:

- The `src` directory contains the source file
- The `povray` directory contains the binaries `povray` and `gm` and the povray file `scherks.pov` which will be copied to the amazon instances
- The `pom.xml` file contains information about the project and configuration details used by Maven to build the project
- The `results` directory will be generated during runtime and contains the animated gif-file

The source file `src/main/java/de/yarnseemannsgarn/ec2_cloud_renderer/App.java` basically uses the `jclouds Compute API` to create as many instances as given by the user. Fixed paramaters are the Instance type (`t1.micro`) and the location (`us-west-1`). To enable ssh connections to the instances the `SshjSshClientModule` is used. For parallel rendering the programme uses the standard `Thread` class of Java. The locale machine collects the rendered pictures and runs the `gm` script to produce the gif file.

4 Results

Measurement were made for 128 frames with 1, 2, 4, 8 and 16 instances. Table 1 shows the measurement results, T_R is the copy and render time.

Instances	$\mathbf{T}_K$ in s	**Speedup**	**Efficency**
1	0	0	0
2	0	0	0
4	0	0	0
8	0	0	0
16	0	0	0

Table 1: Measurements