Department of Computer Science, Electrical and Space Engineering Luleå University of Technology

Cloud Services

Lab 5 Report: Option 1 - Cloud Benchmarking

Author:
Hanna Ogbazghi
Elyas Khorasani
Yacine Rabehi

Supervisors:
Karan Mitra

20 January 2022

Contents

1 Download and Installation of Apache JMeter 1

2 Lab 4 Webpage Performance Test 2

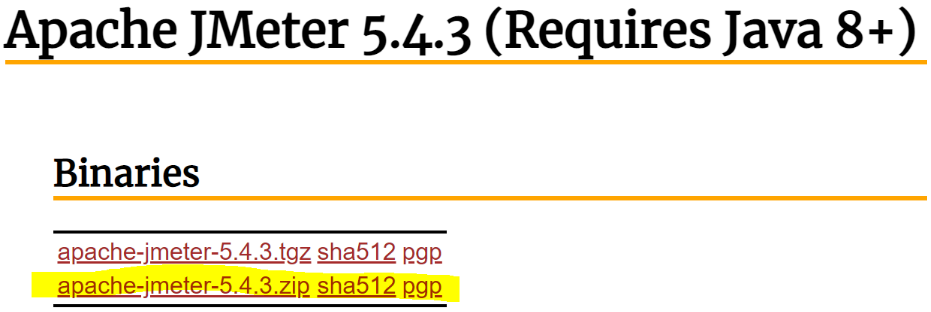
2.1 Building a Test Plan 2

2.2 Results and Analysis 4

1 Download and Installation of Apache JMeter

No concrete installation is needed to run Apache JMeter, the only requirement is having a Java version 8 or higher. To use Apache JMeter, an option is downloading the binaries in zip format from the the following link: https://jmeter.apache.org/download_jmeter.cgi

Figure 1 shows said file.



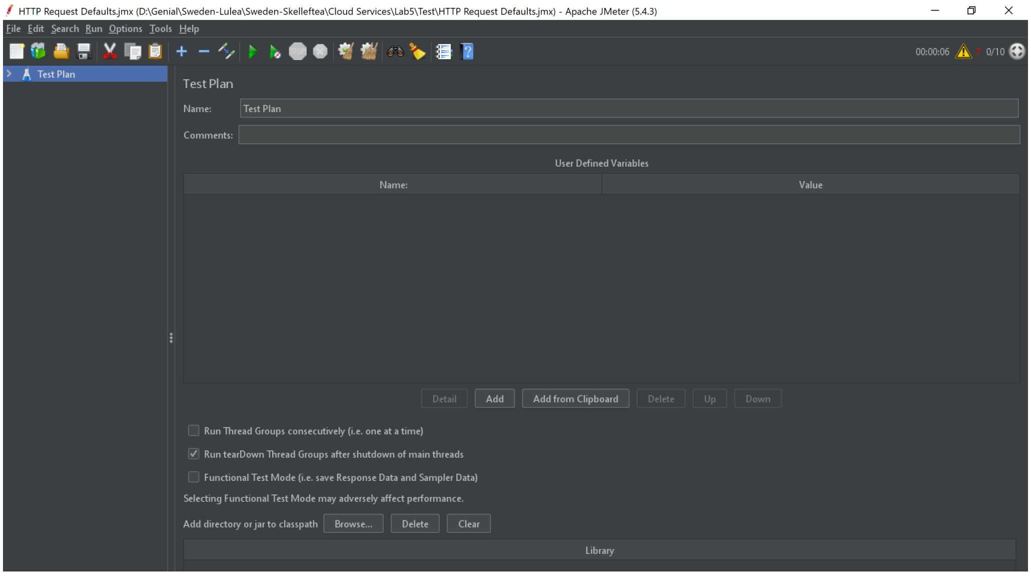
Apache JMeter 5.4.3 (Requires Java 8+)

Binaries

apache-jmeter-5.4.3.tgz sha512 pgp
apache-jmeter-5.4.3.zip sha512 pgp

Figure 1: Apache JMeter Binaries

After extracting the files, we run the file "jmeter.bat" which is located in the "bin" folder. This prompts the welcome screen for Apache JMeter GUI. Figure 2 illustrates this.



HTTP Request Defaults.jmx (D:\Genial\Sweden-Lulea\Sweden-Skelleftea\Cloud Services\Lab5\Test\HTTP Request Defaults.jmx) - Apache JMeter (5.4.3)

Test Plan

Name: Test Plan

Comments:

User Defined Variables

Name:	Value

□ Run Thread Groups consecutively (i.e. one at a time)

☑ Run tearDown Thread Groups after shutdown of main threads

□ Functional Test Mode (i.e. save Response Data and Sampler Data)

Selecting Functional Test Mode may adversely affect performance.

Add directory or jar to classpath

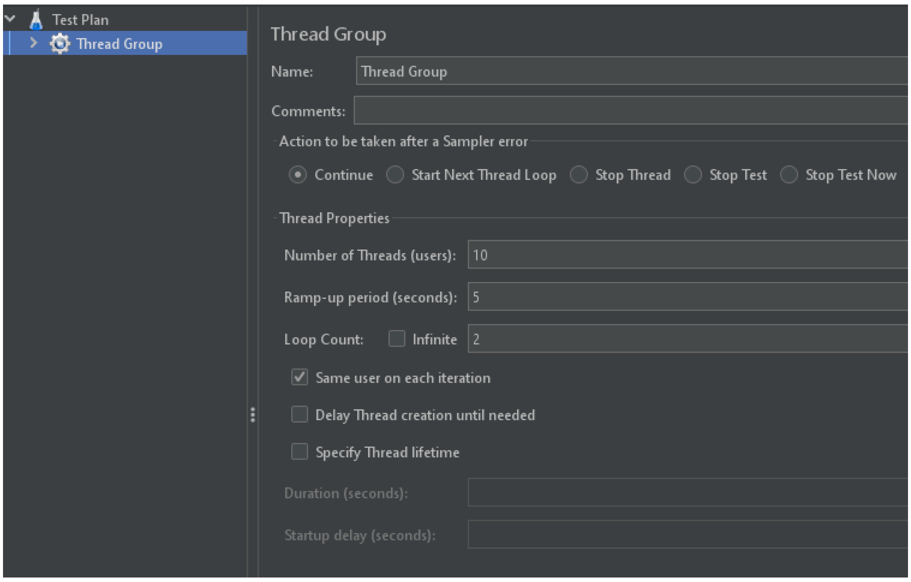
Figure 2: Apache JMeter GUI front screen

2 Lab 4 Webpage Performance Test

2.1 Building a Test Plan

To build a Test Plan, we follow the next steps as described in chapter 4 of the user's manual (https://jmeter.apache.org/usermanual/build-web-test-plan.html):

1. **Creating a Thread Group**
   This is done by right clicking on the Test Plan and selecting Add>Threads>Thread Group. This prompts the screen illustrated in 3.

   Thread Group

   Name: Thread Group

   Comments:

   Action to be taken after a Sampler error

   ● Continue ○ Start Next Thread Loop ○ Stop Thread ○ Stop Test ○ Stop Test Now

   Thread Properties

   Number of Threads (users): 10

   Ramp-up period (seconds): 5

   Loop Count: □ Infinite 2

   ☑ Same user on each iteration

   □ Delay Thread creation until needed

   □ Specify Thread lifetime

   Duration (seconds):

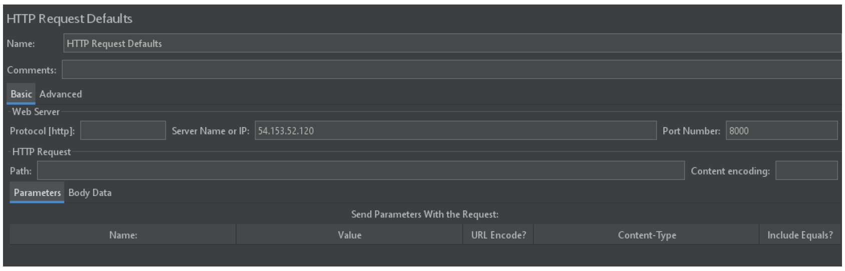
   Startup delay (seconds):

   Figure 3: Thread Group

   We then specify the number of users, the ramp-up period (which is the time to take to run all the users) and the loop count.
   For our test, we vary the parameters such that, number of users = [10,50,100,200,400], ramp-up period(s) = [1,5,10] and loop count = 10.

2. **Adding Default HTTP Request Properties**
   As the name suggests, we setup here some default parameters that would be taken for all HTTP requests. We do this by right clicking on Thread Group, then Add>Config element> HTTP request defaults.
   In our case, those would be: IP = 54.153.52.120 and Port = 8000. The relevant screen is shown in figure 4.

3. **Adding HTTP Requests**
   Here, we create as many HTTP requests as needed by right clicking on Thread Group then Add>Sampler>HTTP request. In our case, we have one such



HTTP Request Defaults

Name: HTTP Request Defaults

Comments:

Web Server

Protocol [http]: Server Name or IP: 54.153.52.120 Port Number: 8000

HTTP Request

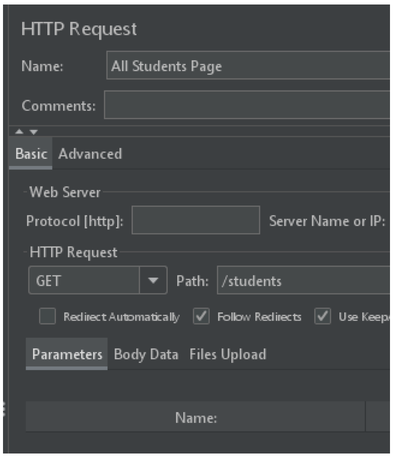
Path: Content encoding:

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?

Figure 4: HTTP default settings

request. We specify the HTTP request as "GET" and the path as "/students" (as can be seen in figure 5).
We note that there is no need to specify the address since it was configured in default settings in the previous step.



HTTP Request

Name: All Students Page

Comments:

Web Server

Protocol [http]: Server Name or IP:

HTTP Request

GET Path: /students

□ Redirect Automatically ☑ Follow Redirects ☑ Use Keep

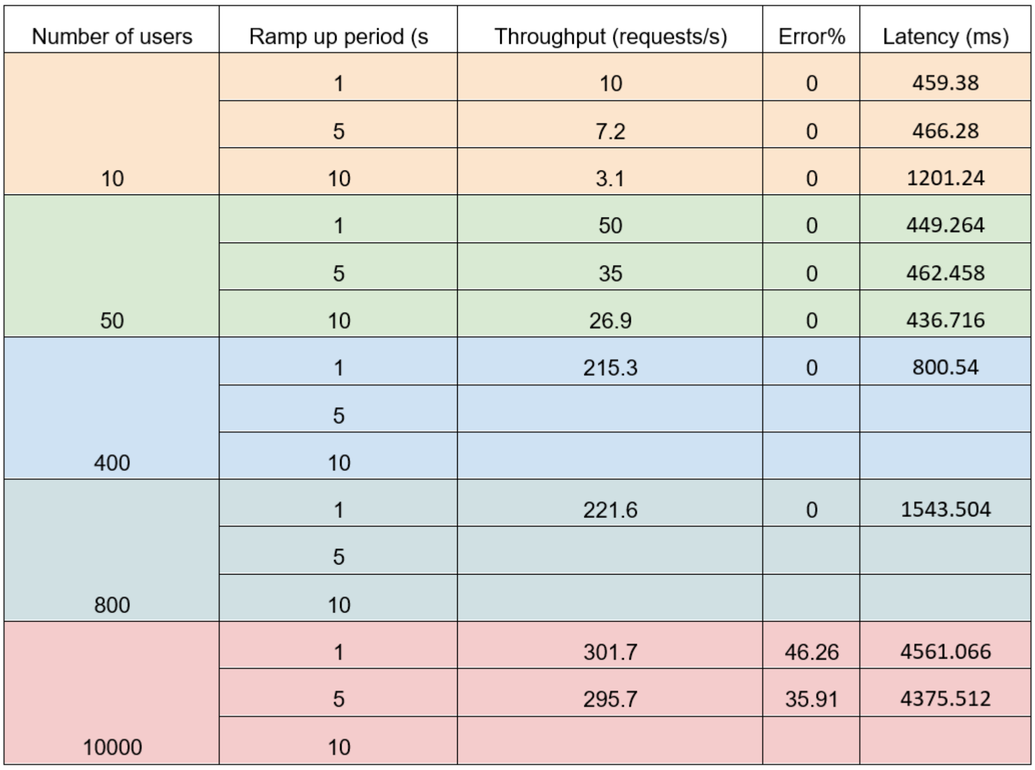
Name:

Figure 5: HTTP request configuration for getting names of all students registered

4. **Adding a Listener to View/Save the Test Results**
In this step, we select the wanted type of report. We do this by right clicking on the Thread Group, then Add>Listener>X. In our case we select the two options of "Summary Report" and "View Results in Table". From the first option, we are interested in the two variables of **Throughput(requests/second** and **Error %**. From the second option, we are interested in the variable of **Latency**.
We note that there is no option for average latency, so we will calculate that using excel.

2.2 Results and Analysis

We run the tests for all the combination of variables described in the "Adding Default HTTP Request Properties" of the previous section.
The results are summarized and displayed in figure 6.
NB: some results are missing due to an access problem to the instance caused by the huge amount of previous requests.



Number of users	Ramp up period (s	Throughput (requests/s)	Error%	Latency (ms)
10	1	10	0	459.38
	5	7.2	0	466.28
	10	3.1	0	1201.24
50	1	50	0	449.264
	5	35	0	462.458
	10	26.9	0	436.716
400	1	215.3	0	800.54
	5			
	10			
800	1	221.6	0	1543.504
	5			
	10			
10000	1	301.7	46.26	4561.066
	5	295.7	35.91	4375.512
	10			

Figure 6: Results Summary

In order to analyse the obtained results, we plot some graphs and make some tables illustrating:
- The effects of number of users.
- The effects of ramp-up period.
These graphs and tables are shown in figures 7, 8 and 9.

From figure 7, we can deduce the effect of the increase of the number of users, which is:
- Increasing the throughput.
- Increasing the Latency.
- Increasing the possibility of errors (no errors till 10000 users).

From figure 8, we can deduce the effect of the ramp-up period, which is:
- Decreasing the throughput.

- Almost no effect on the Latency (slight difference noticed).

Saturation point

Figure 9 gives us a hint on the saturation point for our application. We can observe that starting from 10000 users, which produce a throughput of around 300 requests/s, we have 46% and 36% request errors for ramp-up periods of 1 and 5 seconds.

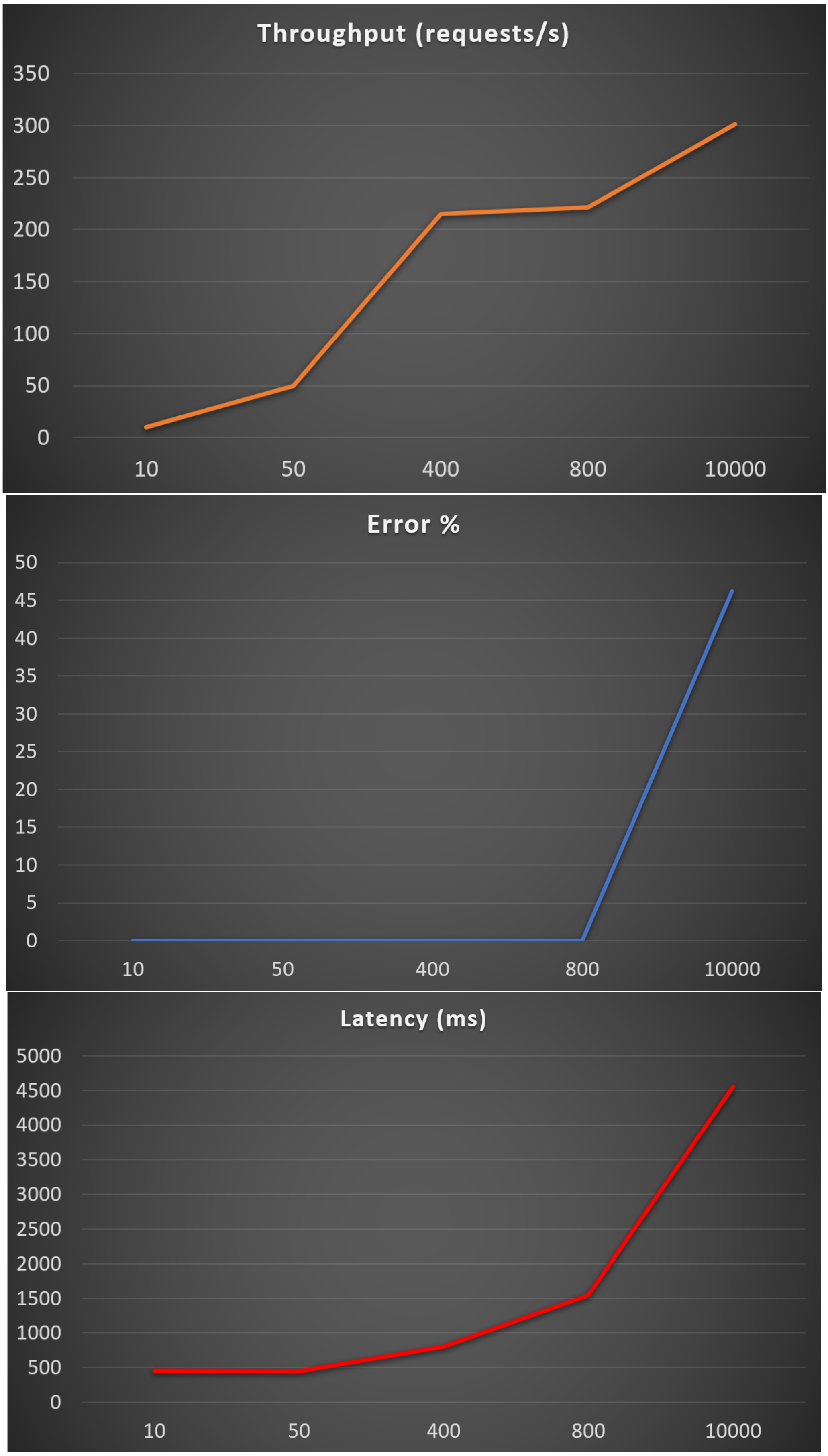
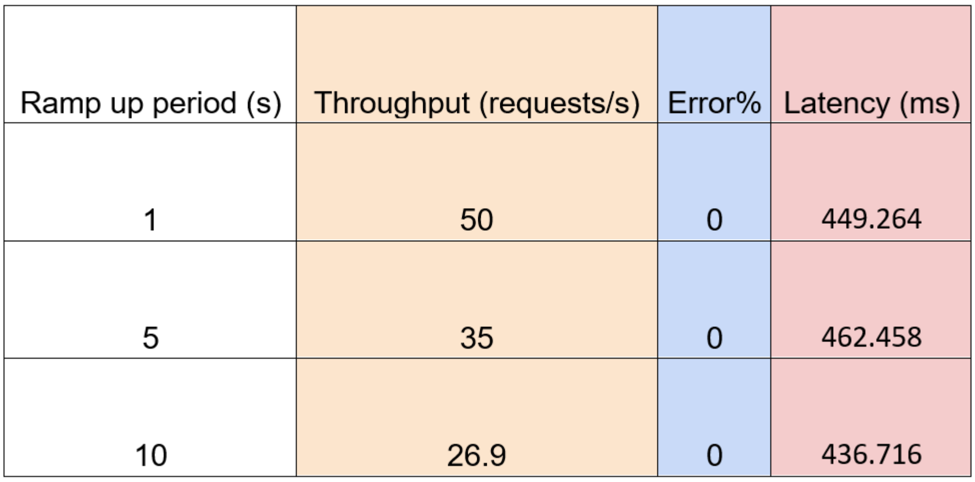
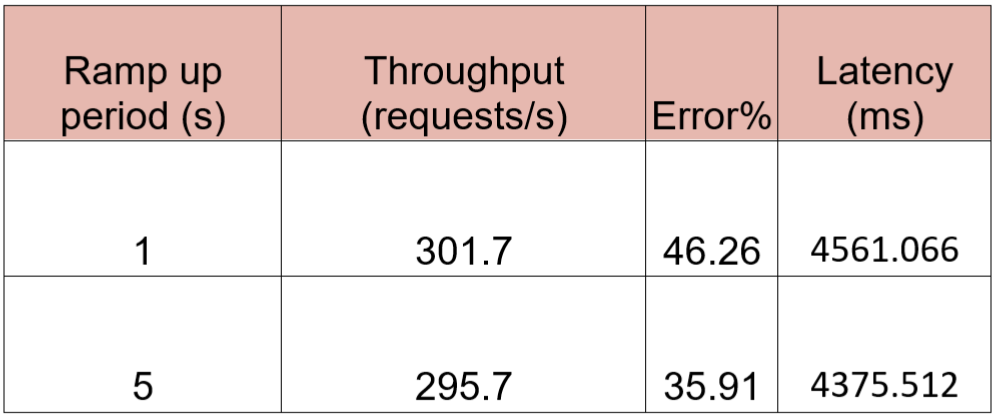


Figure 7: Effect of number of users



Ramp up period (s)	Throughput (requests/s)	Error%	Latency (ms)
1	50	0	449.264
5	35	0	462.458
10	26.9	0	436.716

Figure 8: Effect of ramp-up period



Ramp up period (s)	Throughput (requests/s)	Error%	Latency (ms)
1	301.7	46.26	4561.066
5	295.7	35.91	4375.512

Figure 9: Saturation