

C436H Coursework: Load Testing

Deadline: Monday 5 March 2018, 17:00. Electronic submission of PDF report on CATE.

Submission: Individual or in groups of 2 students.

Feedback: Feedback session on 8 March 2018, 11:00-13:00.

Goals: 1) Learn the *response time law* for interactive workloads. 2) Run performance tests against a JPetStore instance on Microsoft Azure to gain experience on load testing and cloud computing.

Coursework: The coursework consists of two parts. In all parts you should use Apache JMeter with the `jmeter-load-test.jmx` test plan provided with the coursework. Your objective is to write a report to address the requirements of Part 1 and Part 2. The total coursework weight in the course is 10%.

What to submit. You are asked to submit on CATE a brief PDF report describing your findings. There are no space limits, but it is recommended to aim at 2-3 pages, with font size 11.

Please make sure that you do **not** delete your VMs or PaaS web apps before your coursework is returned to you. You can stop them, but please do not delete them in case we need to check something.

Marking. We will reward with higher grades submissions with a clear presentation of the experimental methodology and results, and with reflections on the outcomes you have seen in the experiments. If you anything surprised you compared to what you initially expected, please comment on that too.

Getting technical help. If you face technical problems related to Azure or JMeter, please contact your GTA, Salvatore Dipietro (s.dipietro14@imperial.ac.uk).

Background: Response Time Law

Load testing tools such as JMeter cyclically send requests to an IT system and record QoS metrics such as mean response time (R) and throughput (X) of requests. Assume that the total time during which the system was tested *in the steady phase* is T and that the number of clients in the steady phase is N . Also, suppose that a client that issues a request is blocked until receiving a response. Moreover let the think time Z be the mean time between a client receiving a response and submitting the next request.

The *response time law* states that as the steady phase duration T grows asymptotically large, it is

$$R = N/X - Z \quad (1)$$

The response time law may be seen as an application of Little's law, which is a standard result of queueing theory¹. The formula holds under fairly general conditions, provided that T is large, so that the difference between issued requests and obtained responses is negligible.

Part 1 (weight: 40%)

Initially, have an in-depth look to the test plan to familiarize with it. It is recommended that you use the help feature in the JMeter GUI to clarify what individual test plan elements do. Then you are asked to address the following requirements:

Q1) Run JMeter with $N = 8$ clients and $Z = 0$ against your JPetStore VM (Basic A1 size, UK South region) for 15 minutes or more, until you see a performance degradation of the *View Order* requests. Include in the report your results for throughput and response times. Can you explain the progressive increase of mean response times of the *View Order* requests?

Q2) Now disable the *View Order* request (in the GUI, you can do this by right-clicking on *View Order*). Repeat the test as in **Q1** but for $N = 1, 2, 4, 8, 16$ and $Z = 0$. Present your results for throughput and response times. Check to what extent the response time law (1) is verified in the experiment by comparing R and $N/X - Z$ for each test. If you face problems in matching the law, try to repeat the experiment on a different machine, preferably located at another site (e.g., at home instead than in the labs).

¹http://homes.cs.washington.edu/~lazowska/qsp/Images/Chap_03.pdf

Part 2 (weight: 60%)

The goal of this part is to explore the sensitivity of results to the experimental setup.

Q3) Pick one of the following problems and report your experimental findings:

- A) *Sensitivity to geographical placement.* Fix N and $Z = 0$, e.g. as in **Q1**. Study how test results change with the geographical location of the VM (e.g., East US, West US, ...).
- B) *Sensitivity to service model.* Rerun **Q2** against a deployment of JPetStore on PaaS (see Azure tutorial). Compare the maximum achievable throughput of the IaaS VM with the PaaS web app instance.
- C) *Sensitivity to measurement point.* Rerun **Q2** using deployments of JMeter: i) inside the JPetStore VM; (ii) another Azure VM. (To run JMeter, use the command line script on CATE.)
- D) *Sensitivity to autoscaling.* Fix N and $Z = 0$, e.g. as in **Q1**. Setup a PaaS autoscaling rule on Azure and show how the response time change before, during, and after the trigger of the scale up action.
- E) *Creative option.* Propose to the lecturer a different problem with a level of challenge comparable to the ones above.

Each of the above problems carries the same maximum score and you can pick the one you like best.

Guidelines

- *Test duration.* Run the tests for no less than 15 minutes, and the longer the better.
- *Inconsistent results.* It is best that you run an experiment twice to check that your results are consistent. If you are affected by inconsistent results, try running JMeter from the command line. You may also test in the evening (less network traffic).
- *Running JMeter from the command line.* You can find on CATE a Linux bash script to run JMeter from the command line, if you need to. If you installed JMeter in /homes/mylogin/apache-jmeter-3.3, start ./runjmeter.sh from /homes/mylogin after putting in the same directory jmeter.properties (unzip the archive released with the coursework) and jpetstore-load-test.jmx. Before running the script make sure that you **change the URL variable** in ./runjmeter.sh to point to the IP address of your JPetStore.
- *Tomcat restart.* It is often a good idea to restart Tomcat before carrying out a new experiment. See step 4.13 in the Azure tutorial. You need to use bash bin/catalina.sh stop to stop Tomcat and bash bin/catalina.sh start to restart it after waiting a few seconds.
- *Is JPetStore up?* JMeter will run even if JPetStore is down, so before running an experiment, make sure that JPetStore is up. Use an incognito browser window to check (to avoid cookie caching).
- *Exporting results to CSV.* If you want, you can store the measurements in a CSV file that you can import in Excel or MATLAB for plotting. A CSV file is automatically generated by the command line scripts, while in the GUI you can use the *View Results in Table / Write results to file* feature. In the JMeter GUI you can export the plots by right clicking on the figure that you want to save.