



## GS04: Tools and Environments

### Lab Session 9: Performance Testing with JMeter

The aim of this lab session is to gain experience in performance and load testing. Performance and Throughput requirements are very dominant requirements for many software architectures. This is particularly the case when the load is uncertain because an application will be used by the general public through Internet access channels. In this lab session we will explore the use of Apache's JMeter, developed as part of the Jakarta project. JMeter allows the definition of performance test cases, their execution under a given load profile and visualization of key performance indicators, such as latency and throughput. We will use the GZ04 web application that we already used in Lab Session 8. You might wish to refer to the JMeter documentation at <http://jakarta.apache.org/jmeter/usermanual/index.html>.

#### Defining a JMeter Test Case

Launch JMeter from the All Programs menu. Then create a test plan that consists of a thread group that you repeat just once and chose only one concurrent thread initially. Inside the thread group create two HTTP requests. The first one should create a new UCAS Application at <http://pizza.cs.ucl.ac.uk:8082/UcasApply.jsp>. The second request should delete the application you created at <http://pizza.cs.ucl.ac.uk:8082/UcasDelete.jsp>. For both HTTP requests you will need to add a number of parameters. You can identify which parameters these need to be by viewing the source of the Java Server Page. When you think you have configured the two HTTP requests correctly, add a listener to view the results as a tree. This will allow you to see which HTTP requests were sent and what the servers responses looked like after you have run the test.

#### Measuring Response Time / Latency

Once you have convinced yourself that your HTTP requests are working correctly, replace the Tree Listener with a Graph Results listener. This will allow you to see the performance measurement summary as a graph. Then increase the number of times you want the test case to be repeated (for example to 500) and run the result. Interpret the results. Human users will not notice delays that are smaller than about 100 milliseconds.

#### Measuring Response Time under Load / Throughput

The test you conducted above measured the performance of a response of the server that was otherwise idle (assuming that none of your fellow students executed a test at the same time). You can also test the behaviour of the server under load. To do this increase the number of threads to, for example 10 or 15 and add a ramp up period of 10-15 seconds so that you can see the effect that adding new concurrent threads has on the response time.