

# Web Server Performance Simulation — Progress Report (version 1.0.0.1)

Andrew Ferrier

May 21, 2002

This brief document describes my progress on my individual project so far, and explains my plans for finishing the project.

My individual project involves writing software to simulate web server systems, then using that software to come to some conclusions about the best way to set up and configure web server systems. I won't explain the objectives in any more detail here but more information can be found in my outsourcing report (which can be found at <http://www.new-destiny.co.uk/andrew/project/>), and also includes information on my original timetable for the project.

So far I have completed more than what I had planned to do by this date. I had planned to have a working simulation system up-and-running, with a complete XML Schema or DTD, and have full input capability working (the last item complete by the end of February). I have completed the following items:

- I have a working simulation system which supports all the features described for the input and execution portions of the outsourcing report, including server threading, but with the exception of the 'experimental' (empirical) and 'Poisson' distributions, which I am planning on adding shortly. The following additional features and changes have also been made:
  - Instead of 'magic' routing as originally described in the outsourcing report, I have implemented user-specified routing. This was for two main reasons:
    - \* It enables the user to choose which routes to use for each combination of two system objects. Moreover, different routes can be used for each direction.
    - \* I realised that, particularly for a system with a large number of system objects (which I would like the software to be able to simulate), pre-calculating ideal routes would be a computationally difficult and possibly impossible task. Even if I did manage to implement such a system, it would require a lot of complex extra code.
  - As a consequence of the above user-specified routing mechanism, I have also added TTL (time-to-live) values to messages, to ensure that mistakes in the user routing specification will not cause the system

to run away as messages will expire. As the software is currently implemented, messages that expire output a warning.

- Some tags and attributes are slightly different from those in the original outsourcing report specification to ensure that XML Schema defaults can be added, and to make future extensions of the input file structure easier. The changes are syntactic but they don't really change what the user can specify. They will be documented in the final report.
  - A 'drop' attribute has been added to the network node tag which allows for the network node to randomly drop packets.
  - The system outputs brief progress reports whilst it is executing, every 10 seconds, to let the user know that there is still progress, and give an estimate of the time to completion.
  - The network nodes and servers print out queue statistics: the mean length in the queue, and the mean time in the queue, and the variance of these quantities.
- I have an XML Schema which is used to check the input to the program as it currently stands. There are a few extra validation issues to be dealt with — ensuring some sensible defaults are set, etc., but in general, the Schema is complete and will only be changed when further features are added to the simulation to ensure that it compares with the input to the program.
  - The program has full input support for all of its current features, with XML Schema validation. Some work needs to be done to increase the reliability of this because it appears that some Java XML parsing engines do not fully support XML Schema validation, and thus the program needs to fail gracefully if Schema validation cannot be performed. But input of well-formed input documents currently works correctly.

At the moment I am working on the following items:

- I am ensuring that it is possible for the user of the simulation program to determine when the simulation has reached the equilibrium point of the system, and restart measurements from that point, to ensure that measurements are accurate and meaningful. To this end, I'm initially adding a data dumping facility to the program to enable the user to determine, by plotting the data, when equilibrium is reached. I will add a feature so that the user can specify the amount of (virtual) time that has to pass before measurements are started — the user will have to run the simulation once, find the equilibrium start point, and run the simulation again with the start point specified.

The data dumping feature should also be useful for other purposes so that the user can analyse the changes in the simulation system over time. Currently, the user can specify the (virtual) time period inbetween data dumps.

Later on, I would like to add a feature so that the system can automatically detect when equilibrium is reached. However, this will require some careful thought to ensure that the system does not reach false conclusions.

- I am studying my usage of the Java random generator. Up till now, I have merely used the standard random number generator, but it has been pointed out to me that this may not be adequate as the random number generator may have inadequately short periods. This is something I need to study further and amend if necessary.
- I am improving the way in which my multiple server threads are modelled by implementing a simulation of single-processor thread switching, using a ‘time-token’ based approach. This approach is experimental, and I will document it’s progress later.
- I am adding support for the Poisson and ‘empirical’ distributions to the simulation.

My plans for completion of the project roughly match those in my original timetable in the outsourcing report, and are as follows:

- I first plan to deal with the items that I am currently working on, that I have mentioned above.
- Then I intend on packaging an early version of the software and making it available to various people for testing and comment, modifying it accordingly in response to their comments.
- I then plan on enhancing the output capabilities of the simulation, so that the data dumping output is comprehensive and as user-configurable as necessary, and the user can dump all the data from the system, including that specified in the outsourcing report. At the same time as I’m doing this, I also intend to start using the software in a controlled manner to try to answer some of the questions that I originally posed in my outsourcing report. I plan to do this by half way through April.
- Then I plan on wrapping up my creation of the software: trying to minimise the number of bugs, ensuring that it is fully documented, etc. I also intend to complete my studies using the software to answer some server sizing questions. I plan to complete this by the end of May.
- I then plan to complete the final report, documenting what I have done and the conclusions that I’ve come to.