Web Server Modelling - Individual Project Proposal

Andrew Ferrier, ajf98

There are various issues to be decided upon when creating a web serving environment – whether to buy many small machines and connect them together in a 'farm' or to buy one large server, what kind of web server software to install, how many concurrent serving processes to run on each physical machine, how to configure the interactions between the web server and other software such as databases, etc.

My idea and project proposal is to construct a software program that allows one to model web servers and networks of connected web servers (web server farms), along with simulated requests and replies from clients of the web servers, and to visualise this graphically on the screen with measurements of average response times, volume of data being transferred, clients being handled in parallel, etc., so that one can get 'realtime'[1] feedback on changes to the system. This would enable a systems administrator to get a much better idea of how changes to the system would impact upon performance, cost, etc.

I would partly use this program to write my report, describing various conclusions that I had reached on the performance and viability of various configurations, although of course the point of the program would be that it would allow the user to test a near-infinite set of possible (very specific) configurations, and hence I could only come to some general conclusions.

The idea would attempt to overcome the inherent problems of already existing software performing similar functions:

- ?? They do not simulate the system; they require physical servers and clients for distributed testing – very expensive and inflexible!
- ?? They do not allow such a great deal of interaction and visualisation as I envisage – i.e. 'realtime' graphs and displays, etc. of the progress of the system.
- ?? They do not allow modelling of more than one server.
- ?? They do not simulate varying network conditions.

Ability to Extend

I believe that one advantage of this proposal is that it allows the flexibility to produce a simple program, and then to build on that progressively. The system has scope for being made as complex as there is time to allow. The following are ideas for ways to extend the project from a simple one to a more complex one, hopefully a high-rating one. These are only a sampling of ideas – more extensions would be possible I feel.

---

[1] *Note: in putting 'realtime' in quotes I mean that the simulation would not necessarily have time running at the same speed as actual clock time, but that any changes made to the parameters of the simulated system would become apparent in the measured statistics immediately since the system could be 'running' as the parameters were being altered.*

?? HTTP requests could be modelled in a basic way initially (e.g. the same time to process for each request), and extended to fit a statistical distribution later on that models more accurately the kinds of requests users really make, or possibly a distribution constructed from real, measured, data.
?? A basic static HTTP server could be modelled at first, with modelling of dynamic content and a back-end database and/or application server being added in later.
?? Reliability testing features could be added in; such that, for example, certain packets of data were randomly 'dropped', to see how this would affect performance and reliability. A simple model of how HTTP and TCP/IP work would be incorporated, with the possibility of modelling 'new' fictional protocols to investigate other possibilities.
?? There is the possibility to extend the system to model other networks of connected machines, such as mail servers, for example. Another system I might like to model would be IBM MQSeries, which I worked on during my industrial placement – it's a messaging and queuing product and hence might be suitable for modelling also. These are of course 'store and forward' based server architectures, rather than transaction based ones, so some thought would be required on the best way to do this – this extension is more ambitious I feel.

Possible Implementation

I would initially propose that the system would be very suitable for modelling in Java, as it seems sensible that an object-oriented and threaded language such as that would be quite appropriate for simulation, but I would have to investigate this further.