MODELLING & SIMULATION OF A VIRTUAL CAMPUS
A Case Study Regarding the Open University of Catalonia

Angel A. Juan, Pau Fonseca, Joan M. Marquès, Xavi Vilajosana
Dep. of Computer Science, Open University of Catalonia, Rambla Poblenou, 156, 08018, Barcelona, Spain

Javier Faulin
Dep. of Statistics and OR, Public University of Navarre, Navarre, Spain

Keywords: Modelling & Simulation, Distributed Systems, Network Implementation Choices.

Abstract: In this paper we present a case study regarding the modelling and simulation of a real computer system called Castelldefels. This system gives support to the Virtual Campus of the Open University of Catalonia (UOC), an online university that offers e-learning services to thousands of users. After analyzing several alternatives, the OPNET software was selected as the convenient tool for developing this network-simulation research. The main target of the project was to provide the computer system's managers with a realistic simulation model of their system. This model would allow the managers: (i) to analyze the behaviour of the current system in order to discover possible performance problems such as bottlenecks, weak points in the structure, among others, and (ii) to perform what-if analysis regarding future changes in the system, including the addition of new Internet-based services, variations in the number and types of users, changes in hardware or software components, etc.

1 INTRODUCTION

In order to analyze computer systems and networks performance, both analytical and simulation methods can be used. Analytical methods are based upon mathematical analysis that characterizes a network as a set of equations. This approximation usually implies considering several restrictive assumptions, which tend to be not very realistic, since networks are complex systems formed up by hardware and software (protocols, applications, queuing policies, etc.). Alternatively, simulation techniques can also be used to model in detail the dynamic nature of real computer networks (Law, 2006) (Banks et al., 2001). Simulation allows engineers to test different network designs, even before the network physically exists, and to perform what-if analysis with models of the already existent networks without exposing them to failures or inoperative periods.

The Open University of Catalonia (UOC) is an online university located at Barcelona (Spain) with more than 37,000 community members, including students from Spain and Latin America, professors, and managers among others. With this amount of potential intranet users, performance fine-tuning of the computer system that gives support to the UOC Virtual Campus becomes the most important task for system managers. For that reason, a team formed by managers, professors and students started the so-called Castelldefels Project. The main objective of this project is to improve the system performance levels and, consequently, to increase the quality of the service offered to users of the UOC Virtual Campus. This is carried out by selecting appropriate values for configuration parameters such as network topology, hardware devices, queuing and balancing policies, protocols, etc. (Kurose and Ross, 2005) (Peterson and Davie, 2003).

The computer system makes use of load balancing mechanisms, which allow a convenient load distribution (requests from distinct users) among different available servers. Two dedicated hardware devices perform this load balancing task. While one of the load balancers is operating, the other is in stand-by status. Thus, maintenance tasks can be done without having to stop the web service and, moreover, this service will be available even if

A. Juan A., Fonseca P., M. Marquès J., Vilajosana X. and Faulin J. (2008).
MODELLING & SIMULATION OF A VIRTUAL CAMPUS - A Case Study Regarding the Open University of Catalonia.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - ISAS*, pages 187-190
DOI: 10.5220/0001671501870190

the active load balancer fails. These load balancers are responsible for assuring readiness, at any moment, of web services (HTTP, HTTPS, FTP, SMTP and other proprietary applications). Different load assignment policies can be set up in the balancer. Thus, the balancer device decides which available server will pay attention to an incoming user request based on one of the following selection criteria: random criterion, sequential criterion, the least-loaded-server criterion, or the smallest-response-time-server criterion. In order to check if a concrete server is operative, the load balancer carries out regular tests on the servers. This monitoring process can also be configured with different optional parameters, such as elapsed time between two consecutive tests.

2 SYSTEM DESCRIPTION

When a user opens a web browser and types the University URL, the user request is balanced among several portal servers and a portal page is loaded into the user browser. To complete that process, the client request has already passed through the frontier routers, crossed the firewalls, and arrived to the load balancers, where a new session has been settled down with an available portal server.

Once the user has introduced her login and password and they have been validated in the database, she gets access to the Virtual Campus. This means that her request has been balanced and it has finally arrived to an available front-end server. There are about 25 front-ends servers in the Castelldefels system. Between the load balancers and the front-ends there are also two more hardware devices: a web accelerator and two application firewalls.

3 SIMULATION WITH OPNET

There are several discrete-event simulation programs specially designed for network simulation. One of these programs is the open-source OMNeT++ (Varga, 2001). After some preliminary studies, though, we decided to use the proprietary software OPNET for three reasons: (i) it seemed to be the most widely tested, used and documented software in the network simulation area (Chang, 1999) (Aboelela, 2003) (Brown and Christianson, 2004) (Qadan and Guizani, 2005), (ii) it offers an outstanding library of network devices, and (iii) a free license for academic and research use was available from the software developer.

OPNET is a complete package composed by several modules. It can be scaled from Local Area Networks (LANs) to Wide Area Networks (WANs) formed up by thousands of workstations.

For advanced research, OPNET Modeler offers advanced tools for model design, simulation, data mining and analysis. Using this software, it is possible to edit the source code of any hardware device included in the OPNET library of components, which is provided by hardware manufacturers such as Cisco or 3Com. OPNET Modeler is based in a three-level design hierarchy: (1) a network model, where networks and sub-networks are defined; (2) a node model, where node's (hardware devices) internal structure is defined; and (3) a processes model, where internal node states and functioning can be defined by using C/C++ programming (Svensson and Popescu, 2003).

4 MODELLING THE SYSTEM

At this stage of our project, we have focused on the partial modelling of the Castelldefels architecture, focusing ourselves on the Campus network, which is the infrastructure mainly used by students and professors. Additionally, we have reduced somewhat the model size of the Campus network, assuming that it has fewer servers and devices than the real system has (25 front-ends, 3 mail servers, one backup load balancer and up to 5,000 concurrent users requesting HTTP, FTP and email services).

We have distributed our model in three levels. The first level represents the WAN, which is modelled by a set of geographical nodes. All these nodes are connected to one special node, the Castelldefels node, using an IP (Internet) cloud. At a second level, we found the details of each node, including the Castelldefels one (Figure 1).

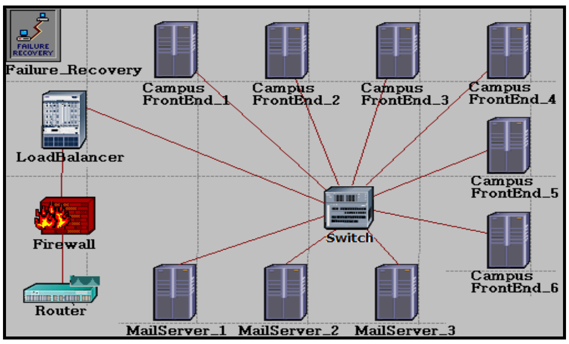

Figure 1: Second Level - The Castelldefels node.

Each of the remaining subnodes is a representation of a Metropolitan Computer Network (MAN), which aggregates several LANs. These MANs configure a third level in our model (Figure 2).

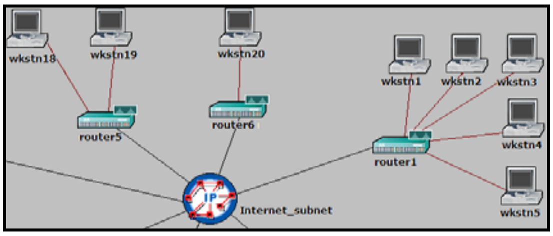

Figure 2: Third Level – Modelling the MANs.

The main components of our model are:

- Applications: Each application defines a service that can be executed in a workstation or in a server. It also defines the load of the system. To emulate the behaviour of the Campus application –Web-based front-ends with file transfer and email support–, we defined three OPNET applications: HTTP (Heavy Browsing), FTP (Low Load) and Email (Medium Load)
- Profiles: A profile is a group of applications to be used by some type of users such as students, professors, etc. Each profile also defines the statistical distributions for the simulation engine. At this stage, only one general profile –with support for all applications– was defined.
- Servers: There were 9 servers to execute applications: 6 front-end servers to emulate the basic Campus activity using HTTP and FTP, and 3 email servers. All servers were directly connected to the load balancer.
- Load balancer: It is a device that decides which server will pay attention to the next user request. Servers are assigned to balanced applications. Each application is balanced using its own policy: Round-Robin or sequential for HTTP and FTP, and Number-of-connections for email. The Round-Robin policy assigns each incoming request to the next server in the sequence, while the Number-of-connections policy assigns tasks according to the number of connected users.
- Frontier router: All former devices were connected to a router through a firewall, which restricts the supported applications. The router was the single connection to the Internet.
- User networks: several networks were set up to emulate students' activity. Each network may have a different number of users, from one to several hundreds, and it was connected to Internet through a gateway. Services were directly requested to the load balancer.

Even when some simplification assumptions were made in this initial model, we expected to obtain interesting results that guide us in the development of more detailed models.

5 SIMULATION RESULTS

After implementing our model in the OPNET Modeler software, we have carried out several simulation experiments regarding parameters such as balancing policies, traffic load, response time, CPU utilization, or failure-recovery time, among others.

The experiments allow us to study different scenarios (what-if analysis) depending on different configurations of the Castelldefels computer system, which helps its managers to get a better understanding of the system behaviour and, which may be even more important, to be able to predict changes in the system performance due to changes in the system configuration: number of servers, traffic loads, balancing policies, maintenance policies, etc. For instance, Figure 3 shows a graphic that predicts the response-time of the FTP application depending on the balancing policy (Minimum load versus Round Robin) and on the number of concurrent clients (500 or 1,000).

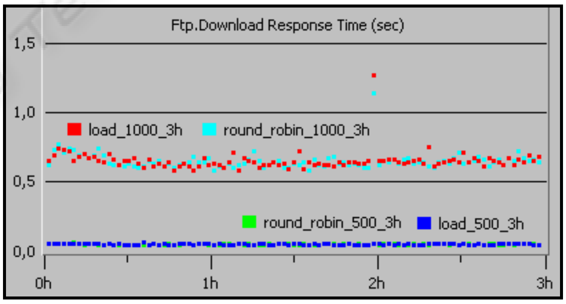

Figure 3: Response time for the FTP application.

According to Figure 3, it seems clear that response times do not significantly depend on the selected balancing policy. Instead, they mainly depend on the number of concurrent users. In fact, this happens to be the most determining factor over the response-time parameter. Other factors, such as the number of servers in the system, seem not to be as decisive, since the CPUs of the available servers are not being pushed to their full capacity (Figure 4). Similar conclusions are reached for the HTTP and e-mail applications.

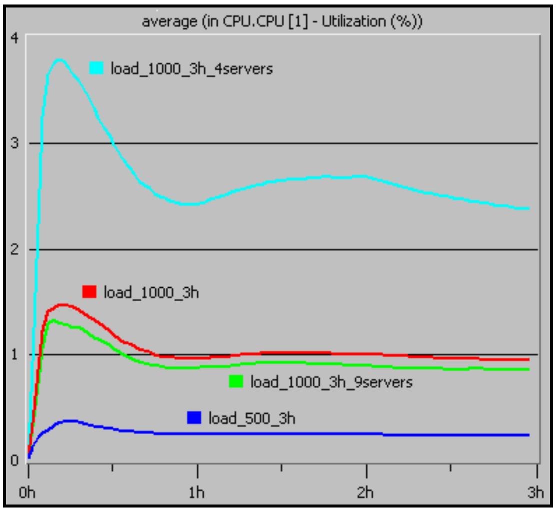

Figure 4: CPU utilization for server 2.

As the previous examples may suggest, once the model is implemented the number of simulation experiments that can be performed to compare different scenarios is almost unlimited, since managers can change the system configuration in every possible way including: number of clients, number of dedicated servers, balancing policy, users profiles, applications load, individual server failures and recoveries, etc. Furthermore, our simulation model soon results in an invaluable tool for the Castelldefels system managers, which have the ultimate responsibility over the quality of the service offered by the UOC Virtual Campus.

6 ACADEMIC APPLICATIONS

It is expensive to set up a physical networking lab for university students. Moreover, even if one university made that investment, such labs would have significant limitations when dealing with different possible scenarios (what-if analysis). In fact, it is virtually impossible to cover, using a physical network, the wide diversity of existing technologies and configurations.

For that reason, use of discrete-event simulation, as a methodology to confront network design and fine-tuning problems, is not only interesting in the professional arena but also in the academic one (Theunis et al. 2003). The current level of computer software and hardware allows the efficient application of simulation-based methods and algorithms to network analysis, allowing a major comprehension of networks' internal functioning process. Using simulation, students are able to analyze alternative scenarios and designs both for LANs and WANs.

7 CONCLUSIONS

When studying networks performance, simulation techniques offer clear advantages over analytical ones, such as: (a) the opportunity of creating models which faithfully reflect the real network characteristics and behaviour, and (b) the possibility of obtaining additional information about the system internal functioning.

We have used OPNET to develop a model of the computer system that gives support to the UOC Virtual Campus. The model allows us to experiment with some what-if scenarios and to test how the system will perform under different configurations. Performing what-if analysis on the simulation model before implementing changes in the real system may avoid unexpected problems in the system day-to-day behaviour.

REFERENCES

Aboelela, E., 2003. *Network Simulation Experiments Manual*. Morgan Kaufmann

Banks, et al., 2001. *Discrete-Event System Simulation*. Prentice-Hall

Brown, K., Christianson, L., 2004. *OPNET Lab Manual*. Prentice Hall.

Chang, X., 1999. Network simulations with Opnet. In *Proceedings of the 1999 Winter Simulation Conference*, pp. 307-314

Kurose, J., Ross, K., 2005. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley

Law, A., 2006. *Simulation Modeling and Analysis*. McGraw-Hill

Peterson, L., Davie, B., 2003. *Computer Networks. A Systems Approach*. Morgan Kaufmann

Qadan, O., Guisan, M., 2005. *OPNET Lab Manual*. John Wiley & Sons

Svensson, T., Popescu, A., 2003. *Development of laboratory exercises based on OPNET Modeler*. Master thesis, June 2003. Blekinge Institute of Technology, Sweden.

Theunis, J., et al., 2003. Advanced Networking Training for Master Students Through OPNET Projects. In *Proceedings of the 2003 OPNETWORK Conference*, Washington D.C., USA

Varga, A., 2001. The OMNeT++ Discrete Event Simulation System. In *Proceedings of the European Simulation Multiconference (ESM'2001)*. June 6-9, Prague, Czech Republic.