PERFORMANCE EVALUAT ION OF A WEB BASED SYSTEM CASE STUDY: LAMP BASED LEARNORG MOODLE

Sulochana Jayashamalee Suoriyaarachchi

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

This dissertation was submitted to the Department of Computer Science and Engineering of the University of Moratuwa in partial fulfillment of the requirements/or the Degree of Master of Science in Computer Science

Department of Computer Science and Engineering
University of Moratuwa
Sri Lanka

February 2010

96423

Abstract

Web based applications are widely deployed around the world for everyday activities of an average person ranging from simple entertainment to complex social, economic, political, educational and scientific tasks. LAMP that abbreviates the combination of Linux, Apache, My SQL and PHP is a popular set of technologies on which most of the web applications are deployed. Although LAMP based web applications arc deployed in millions, the question is whether the intended purposes of these applications are fulfilled satisfactorily from the end user's point of view. The response time and the server resource utilization are the most noteworthy yardsticks using which performance is quantified

This study proposes a proper performance evaluation procedure and recommends an appropriate set of tools and techniques that can be used for the same. The typical method of evaluating performance is to monitor only the server side resource utilization. Many popular tools report the server resource utilization as average values over a period of few minutes whereas most of the user interactions span only for a few seconds. These average values may indicate that the servers are functioning smoothly, while the users may be suffering from poor response from the server. In contrast, this study proposes that while the response time at the user's end is being monitored, the server resources must also be tracked and analyzed.

The case study of LeamOrg- Moodle is used to exemplify the proposed procedure and how the same can be extended. The popular Belief of network always being the bottleneck was not supported by the empirical results of the study. The results obtained for the system under study revealed that the memory can also be a resource bottleneck.

University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

DECLARATION

The work included in this report was done by me, and only by me, and the work has not been submitted for any other academic qualification at any institution.

Candidate: S. J Sooriyaarachchi

Signature:

[illegible]

Date: 08/03/2010

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

I certify that the declaration above by the candidate is true to the best of my knowledge and that this report is acceptable for evaluation for the MSc Research Project

Supervisor: Vishaka Nanayakkara

Signature:

Date: 08/03/2010

UOM Verified Signature

Acknowledgement

I would like to thank the Department of Computer Science and Engineering of the University of Moratuwa for giving me the opportunity to conduct this research and granting me the necessary funds and resources.

My supervisor, Ms. Vishaka Nanayakkara should be thanked for her untiring efforts in going through my work and guiding me throughout the project.

I would like to extend my thanks to Mr. Shantha Fernando for proposing this useful project idea and for his initial guidance given specially in defining the scope. I am grateful to Mr. Shantha Fernando for his initiation and commitment in implementing the e-learning system at the University of Moratuwa, which became the case system in this research.

I would like to thank Dr. Chandana Gamage for motivating me to publish a part of this work at an international conference and Prof. Gihan Dias for pursuing me to complete this work.

A special thank should be extended to my fiancé, Nalin for bearing with me in stressful situations and encouraging me during the research. I would like to thank my family members for all the support given throughout the research. I would like to thank all my friends, especially Sarves for all the support and for encouraging me.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Table of Contents

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

List of Figures

UNIVERSITY OF MORATUWA LIBRARY SRI LANKA

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

List of Tables

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

List of Abbreviations

AB	-	Apache Benchmark
CMS	-	Content Management System
CORBA	-	Common Object Request Broker Architecture
CPU	-	Central Processing Unit
DAV	-	Distributed Authoring and Versioning
DNS	-	Domain Name Service
FTP	-	File Transfer Protocol
HTML	-	Hypertext Markup Language
HTTP	-	Hypertext Transfer Protocol
I/O	-	Input Output
ID	-	Identifier
IMAP	-	Internet Message Access Protocol
LAMP	-	Linux+Apache+MySQL+PHP (or Perl or Python...)
LDAP	-	Lightweight Directory Access Protocol
MIB	-	Management Information Base
MIME	-	Multipurpose Internet Mail Extensions
MOODLE	-	Modular Object Oriented Learning Environment
MRTG	-	Multi Router Traffic Grapher
ODBC	-	Open Database Connectivity
PDF	-	Portable Document Format
PHP	-	Hypertext Preprocessor scripting language
POP	-	Post Office Protocol
RRD	-	Round Robin Database
RTT	-	Round Trip Time
SLA	-	Service Level Agreement
SNMP	-	Simple Network Management Protocol
SQL	-	Structured Query Language
SUT	-	System under Test
TCP	-	Transmission Control Protocol
WSLT	-	Web Server Load Tool

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

1 Introduction

1.1 Background

World Wide Web is no longer a read-only system for the users. Presently, it is a read-write system with user interactions.

Almost all day-to-day activities of an average person ranging from simple entertainment to complex social, economic, political, educational and scientific tasks are expected to be enhanced, enabled and expanded using the web. Various e-disciplines such as e-commerce, e-governance, e-learning and e-channelling have become buzz words in the society.

Among these, e-Learning is of prime importance in all aspects of formal and informal education. Not only the educational institutions such as schools and universities, but also the other institutions in the industry rely on e-learning over the web as a promising mechanism for training and education.

Although the applications are deployed over the web in large numbers, it is questionable that their intended purposes or Service Level Agreements (SLA) are met satisfactorily. The performance of such systems in terms of responsiveness, availability and reliability as well as the resource utilization has not received comparable attention in contrast to their wide deployment.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Hence, this study focuses on the performance of a web based system, which is based on a set of popular technologies. Therefore, the outcomes of the study will be applicable for many such applications based on similar technologies.

1.2 Case Study

Moodle is a popular open source Content Management System (CMS). LearnOrg-Moodle in the University of Moratuwa, Sri Lanka is a Moodle instance running on Apache on Linux with MySQL as its backend. In other words, LearnOrg-Moodle is a LAMP based system similar to millions of such systems deployed worldwide.

LAMP refers to the web development and deployment platform which comprises Linux, Apache, MySQL and PHP. Though "P" can refer to other server side scripting languages such as Perl or Python, here it refers to PHP due to the simple reason that Moodle application is developed in PHP.

LearnOrg-Moodle is crucial in the Blended Mode Education System practised in the Faculty of Engineering of the University of Moratuwa. It hosts a number of courses offered by the faculty especially for the first year students. The system provides dynamic content and interactive services to a user base of over 3000 students and about 200 staff members. High demand for scaling the system in an environment where the resources are limited implies that there should be optimal capacity planning. Furthermore, in the existing system there had been user complaints about system's responsiveness and unavailability with simultaneous user access. Recently another Moodle system was introduced to the system, which is referred to as Mihindu-Moodle, to handle first year students' requirements. This system caters to around 1000 users.

It is required to find out the acceptable responsiveness and the system's limits at overloading conditions in order to quantitatively and qualitatively determine actual resource requirements of the system.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

To generalize the problem, the objective here is to extract facts about the performance of a web application based on a systematic performance evaluation.

1.3 Research Problem

Performance analysis of the systems was not a prime focus of the system administrators in the university due to various reasons such as the small scale of the systems, the lack of staff and cost of analysis.

Whenever a performance issue is reported the administrators would monitor servers using typical tools such as *top* and MRTG and analyze various logs. Until the time of this research, the server administrators of the university were not well aware of the tools and techniques, which would be better suited for performance monitoring and what aspects should be monitored.

Hence, this study seeks answers to the following research questions:

Question 1. What is the proper approach of performance evaluation of a web application?

Question 2. What tools and techniques are appropriate for the selected approach?

Question 3. What aspects affect the performance of LAMP based web applications?

1.4 Scope

The study identifies the workload patterns on the system mentioned in section 1.2, evaluates the performance of the existing system by applying identified types of workload, identifies the performance bottlenecks, proposes enhancements to eliminate the bottlenecks, and gives recommendations for growth of the system.

The outcome of this work is based on the system performance metrics such as response time and resource utilization.

The qualitative factors such as the look-and-feel and the navigating efficiency of the Moodle application were not considered.

Moodle application was viewed from the point of view of resource consumption and interactions with the rest of the architectural components in the LAMP platform. Hence most of the business logic of Moodle was assumed to be a black box.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Similarly all the architectural components of the system were viewed in terms of the service rates and capacities so that the aspects such as security were not considered.

1.5 Thesis Outline

This thesis is structured as follows: Chapter 2 describes the literature related to the underlying concepts and technologies. Chapter 3 explains the methodology followed in conducting the research. Chapter 4 describes the system under study in detail based on the findings of the system identification. Chapter 5 describes the workload characteristics, and the tools and techniques for applying the workload on the system. Chapter 6 explains the experimental design. Chapter 7 illustrates the results of the performance tests. Chapter 8 gives an analysis and a discussion of the results given in Chapter 7. Finally, Chapter 9 concludes the thesis by summarising the findings and giving recommendations.

2 Literature Review

2.1 Moodle

MOODLE, which stands for Modular Object Oriented Distributed Learning Environment, originated with the objective of incorporating pedagogical features missing in then-existed learning environments in 1998. Moodle is based on social constructivist pedagogy.

According to Moodle philosophy, *constructivism* is the point of view, which advocates the philosophy of people actively constructing new knowledge as they interact with their environment [1]. Social Constructivism extends this idea to a group of people collaboratively creating a small culture of shared artefacts with shared meaning.

Moodle brings this pedagogy on to the web, which generates a huge amount of user interaction with the system via activities such as forums, wikis, quizzes, assignments, and chats. Hence web user satisfaction is of prime importance in the case of learning through Moodle.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

UNIVERSITY OF MORATUWA LIBRARY SRI LANKA

2.2 Web User Satisfaction

Much research has been conducted in the area of web user satisfaction often based on e-commerce sites. Gehrke and Turban present five determinants of user satisfaction regarding e-commerce sites [2]. These five determinants are page loading speed, business contents, navigation efficiency, security and marketing/customer focus, in the order of importance respectively. Page load speed, delay, response time and such speed related parameters are highlighted in several studies with respect to user satisfaction. The delay experienced by web users is elaborated in [3]. According to this study, a web experience consists of several episodes and each episode consists of wait experiences and interactions. As far as wait experiences are concerned duration of the wait, uncertainty of waiting times, information provided to the user about the wait, the point at which the wait occurs within an episode and the waiting time with respect to the user-expectation are important aspects. The study shows that more than the absolute waiting time the user perception of waiting with respect to his expectation has larger impact. Further the uncertainty of the waiting time is an important factor rather than waiting time itself [3].

Hence, the simple assumption that "faster is better" in providing quality of service in web based systems is not logical. In order to engineer a web based system, there should be an SLA that satisfies the user and also that is feasible for the system to maintain. SLA should promise specific and quantifiable parameters of *Quality of Service.* Quality of service will be represented in terms of parameters that a user can feel and that make sense for decision makers in administering the system. In order to make sure the user satisfaction is met by adhering to the SLA, the parameter values specified in the SLA should be related to users' experience.

2.3 Performance Tuning and Capacity Planning

Achieving user satisfaction has always been a challenge with the technical issues arising from lager demand and limited resources. Therefore, performance tuning and capacity planning are important missions to meet these challenges in any system. Interactive web applications are no exception.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Systems need to be fine tuned to provide the agreed level of service. Capacity planning should also be done such that SLA of the system is met. According to [4] typical questions that arise in capacity planning are;

1. What are the maximum load levels that the system will be able to handle in the production environment?
2. What would the average response time, throughput and resource utilization be under the expected workload?
3. How would performance change if load is increased? Does the system scale?
4. Which components have the largest effect on the overall system performance and are they potential bottlenecks?
5. What hardware and software resources are needed to guarantee that SLAs are met?

Furthermore according to [4] approaches to seek answers for above questions are often ad-hoc, experts' opinion or rules of thumb. Often a rule of thumb for performance gain has been the over-provision of resources which leads the institutions nowhere. But the systematic approach will be to do a performance analysis and based on that form the predictions.

Performance evaluation involves multi-disciplinary skills such as mathematical, statistical, analytical, communication and data representational skills.

Systematic approach of performance evaluation involves the following steps [5]:

- Stating goals and defining the system boundaries
- Identifying services and outcomes of the system
- Selecting performance metric
- Listing parameters
- Selecting factors to study
- Selecting evaluation technique
- Selecting workload
- Designing experiments
- Analyzing and Interpreting Data
- Presenting results
- Repeating

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

As per above process, the evaluator needs a thorough understanding about the system throughout the process. The following sections are dedicated to give an insight into the technologies and concepts related to the system under study, which basically is a LAMP based system providing dynamic and interactive web experience for e-learning.

2.4 *Web Application Architecture*

Web applications are typically multi-tier, often 3-tier consisting of a front-end web server layer, an application server layer, and a back-end database layer. The front-end layer accepts client HTTP requests and serves content from the file system and those generated by application server as shown in Figure 2-1.

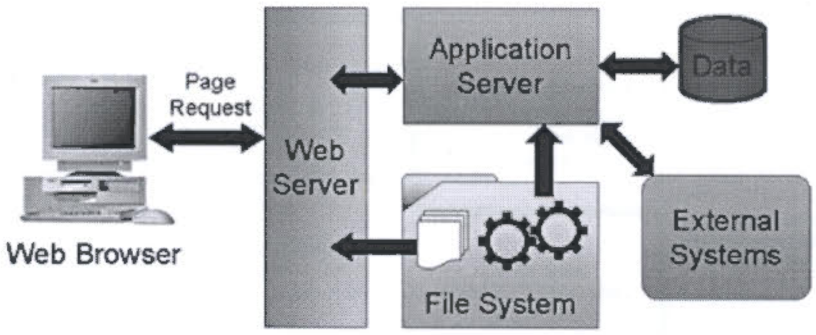


Figure 2-1: Canonical Web architecture [7]

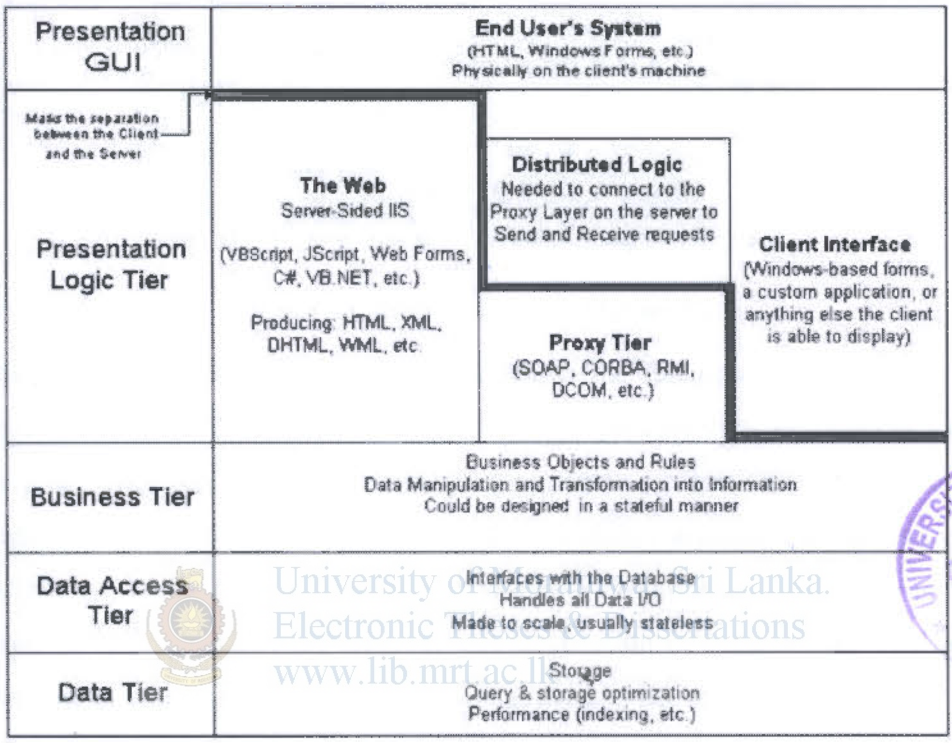
The application server layer handles the business logic and computes the information needed for constructing the requested pages. The back-end database layer stores the necessary data for generating dynamic content. There is no one-to-one mapping between the multi-tier logical layers and physical architectures [6].

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

According to [7] there are significant architectural differences owing to different mechanisms that tie the elements shown in Figure 2-1 together. Some examples for architectural decisions are:

- How to preserve the user's session where the communication from the browser to the web server is generally stateless
- Placement of the application's business logic: models such as the accordingly thin client model, the fat client/thin server model and models where it is distributed are available. However, most systems today tend to push business logic to the server side.
- Whether the communication from the logic to the data should be stateless or stateful

- Method of connecting to the application's persistence data: how the illusion of objects is given to the user while data continues to live in relational tables, whether the connection from the system's business logic to its data is manifested via a mechanism such as JDBC or via messaging.


Presentation GUI
End User's System
(HTML, Windows Forms, etc.)
Physically on the client's machine
Marks the separation between the Client and the Server
Presentation Logic Tier
The Web
Server-Sided IIS
(VBScript, JScript, Web Forms, C#, VB.NET, etc.)
Producing: HTML, XML, DHTML, WML, etc.
Distributed Logic
Needed to connect to the Proxy Layer on the server to Send and Receive requests
Proxy Tier
(SOAP, CORBA, RMI, DCOM, etc.)
Client Interface
(Windows-based forms, a custom application, or anything else the client is able to display)
Business Tier
Business Objects and Rules
Data Manipulation and Transformation into Information
Could be designed in a stateful manner
Data Access Tier
Interfaces with the Database
Handles all Data I/O
Made to scale, usually stateless
Data Tier
Storage
Query & storage optimization
Performance (indexing, etc.)

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

UNIVERSITY OF MORATUWA
LIBRARY
SRI LANKA

Figure 2-2: Typical N-tier model [8]

A typical N-tier architecture is described in [8] as shown in Figure 2-2. As per [8] the presentation logic tier provides end user with the interface to the application by transforming the output of the business tier into a usable and readable format. The proxy tier facilitates the distributed computing by acting on behalf of the distributed logic layer. The data access tier is a reusable interface to the database which does not contain business rules or data manipulation logics.

As far as hardware technologies, software technologies and interactions among components are concerned web based systems are increasingly becoming complex. When understanding such complicated systems viewing them from a single view point is not sufficient. A 4+1 model is given in [7] that can be used to understand web applications.

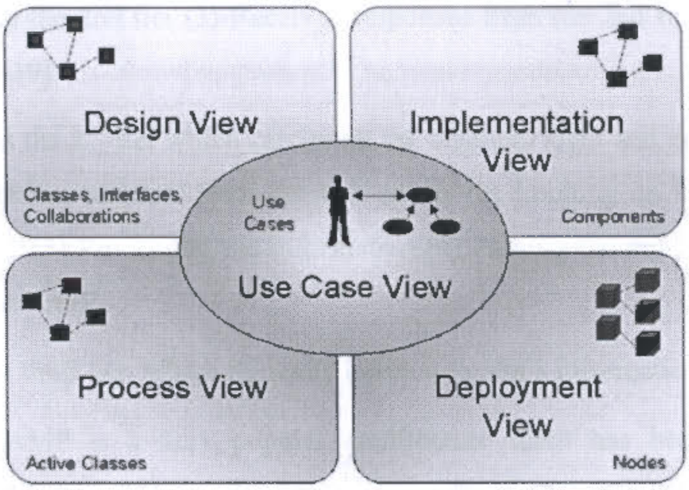


Figure 2-3: 4+1 model view by Philippe [7]

The design view as shown in Figure 2-3 captures the functional requirements of the system or the services provided to the end user. The process view encompasses threads and processes that form concurrency and synchronization mechanisms of the system. This view captures mostly the performance and the scalability aspects. The implementation view addresses the configuration management of the system while the deployment view encompasses system's hardware topology and installation of components. The use case view encompasses the behaviour of the system as seen by the end users [7].

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Modelling 3-tier web applications is also advantageous in aspects such as capacity planning, overload control, performance management, and resource provisioning [9]. Study presented in [9] proposes such a model using queuing network theory. The researchers also have built a test bed to measure the model parameters, based on industry server components and TPC-W benchmark.

2.5 LAMP

LAMP architecture can also be considered as 3-tier where Apache acts as the front-end web server or the 1st tier, PHP engine acts as the 2nd tier, and the MySQL database acts as the 3rd tier.

Apache as the 1st tier performs three basic functionalities: (1) Receives requests from the clients, serve for static web requests (2) At the same time forwards complex dynamic content requests to the 2nd tier (3) Receives responses from the 2nd tier and sends them back to the clients [9].

PHP engine forms the 2nd tier which carries all the business logic and performs functions such as: (1) Receiving requests from the web server (2) Looking up information in the database or 3rd tier (3) Processing the information (4) Passing the processed information back to the web server [9].

MySQL forms the third tier, which basically keeps web site's information stored [9].

"Even though LAMP is a very popular architecture there has been little work to characterize and henchmark the architecture, especially at an application level, but there has heen a substantial amount of work done to analyse the performance of some other web applications" [10].

2.5.1 Apache Web Server

Modern web servers such as Apache are capable of operating in both multi-threaded and multi-process modes. In multi threaded mode Apache is structured as a pool of worker threads to handle HTTP requests. A worker thread processes the request until it completes and then accepts a new request. In the thread pool model, the threads are pre-created at the start up [9].

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

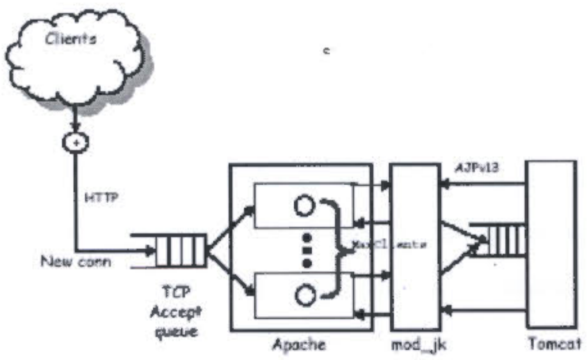


Figure 2-4: Apache multi-threaded architecture [9]

As shown in Figure 2-4, the Apache server's multi-threaded architecture is modelled as a multi-station queuing centre where each station represents a worker thread. The requests are waiting at the TCP accept queue until it is assigned to a worker thread. The requests processed at the Apache server are then forwarded to the middle tier which in this case is a Tomcat Server [9]. But the model can be used for the LAMP environment by replacing Tomcat server by PHP engine.

There are two series of Apache servers which differ in capability and architecture, namely Apache 1.3 series and Apache 2.0 series.

Apache 1.3: This is a process based server, which forks several child processes at start up to achieve stability. This has performance penalty due to cost of process creation and context switching. Since the processes are isolated they cannot share code, data or system resources.

Apache 2.0: This series has major improvements over the former. The followings are some important ones.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

- Ability to be configured as process based, thread based or mixture of two models. Inception of threads are advantageous over processes due to the fact that they are lightweight, they can share code, data and resources, and increased scalability of the server. Compared to process based model the disadvantage here is if a thread misbehaves it can corrupt data or code of other threads.
- Support for many protocols such as FTP, POP3, etc other than HTML. Thus supports for dynamic content generation and authentication.

Regardless of the version, the modularity of Apache is an important feature, that is Apache comes with a number of modules bundled with the server as well as a number of third party modules are available. The user can easily enable or disable the modules according to the requirements.

Authentication modules allow authentication against backend database as well as against plain text files. There are modules for access control and secure data communication as well.

Performance and scalability are achieved via important modules that control throttling. Throttling is the slowing down of content-delivery based on some criteria such as type of the file, client IP address and bandwidth limits. Furthermore, load-balancing can be achieved by modules such as mod_rewrite and reverse proxy so that load is distributed among several backend servers. Reverse proxy is a web server placed in front of other servers to offload certain tasks from backend servers. Seamless redirection of HTTP requests can be done targeting at under-utilized servers. This provides fine grain, per request load balancing. Furthermore, mod_backhand allows seamless redirection.

Modules such as mod_deflate and several other filtering modules allow compression of content thus saving bandwidth.

Publishing or simply managing and uploading content is provided by protocols such as DAV (Distributed Authoring and Versioning).Virtual hosting allows for hosting many sites in single server [11].

UNIVERSITY OF MORATUWA LIBRARY SRI LANKA

2.5.2 PHP

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

PHP is a server side, cross platform, HTML embedded scripting language. The engine that runs PHP is the middleware that generates dynamic contents in the 3-tier LAMP applications.

PHP's modular design provides modules for; Database connectivity for popular databases such as Oracle, MS-SQL server, ODBC interface, MySQL, mSQL, PostgreSQL and so on, XML support, File transfer (eg: FTP), HTTP, Directory support (eg: LDAP), Mail support (eg: IMAP, POP3), PDF document generation, CORBA, SNMP…etc.

Here the focus is more on PHP's interaction with MySQL database and Apache web server.

PHP acts as a module in Apache server itself and shares the same process address space of the web server. Hence doesn't have any inter-process communication overhead with the server. Disadvantage of this architecture is in the situations when the web server is the bottleneck there is no chance to offload PHP middle tier into a separate machine.

Performance and scalability are achieved via important modules that control throttling. Throttling is the slowing down of content-delivery based on some criteria such as type of the file, client IP address and bandwidth limits. Furthermore, load-balancing can be achieved by modules such as mod_rewrite and reverse proxy so that load is distributed among several backend servers. Reverse proxy is a web server placed in front of other servers to offload certain tasks from backend servers. Seamless redirection of HTTP requests can be done targeting at under-utilized servers. This provides fine grain, per request load balancing. Furthermore, mod_backhand allows seamless redirection.

Modules such as mod_deflate and several other filtering modules allow compression of content thus saving bandwidth.

Publishing or simply managing and uploading content is provided by protocols such as DAV (Distributed Authoring and Versioning).Virtual hosting allows for hosting many sites in single server [11].

UNIVERSITY OF MORATUWA LIBRARY SRI LANKA

2.5.2 PHP

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

PHP is a server side, cross platform, HTML embedded scripting language. The engine that runs PHP is the middleware that generates dynamic contents in the 3-tier LAMP applications.

PHP's modular design provides modules for; Database connectivity for popular databases such as Oracle, MS-SQL server, ODBC interface, MySQL, mSQL, PostgreSQL and so on, XML support, File transfer (eg: FTP), HTTP, Directory support (eg: LDAP), Mail support (eg: IMAP, POP3), PDF document generation, CORBA, SNMP…etc.

Here the focus is more on PHP's interaction with MySQL database and Apache web server.

PHP acts as a module in Apache server itself and shares the same process address space of the web server. Hence doesn't have any inter-process communication overhead with the server. Disadvantage of this architecture is in the situations when the web server is the bottleneck there is no chance to offload PHP middle tier into a separate machine.

Whenever HTTP server comes across a PHP tag the PHP interpreter module is invoked. PHP scripts are then taken over by the interpreter module and executed.

PHP uses native code database driver, but the database interface is considered to be ad hoc.

2.5.3 MySQL

MySQL is also a multi-threaded server where a thread cache is used instead of thread pool model. Threads in a thread cache are managed in dynamic fashion and they are not pre-created at start up. When the number of threads needed to serve the requests exceeds the thread cache size new threads will be created. However, only the number of threads equal to the thread cache size is reused and maintained alive [9].

In [9] MySQL database server is modeled as a multi stationed queuing centre which is load-dependent, but for the simplification the number of stations is considered to he the averaged number of worker threads at a run.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

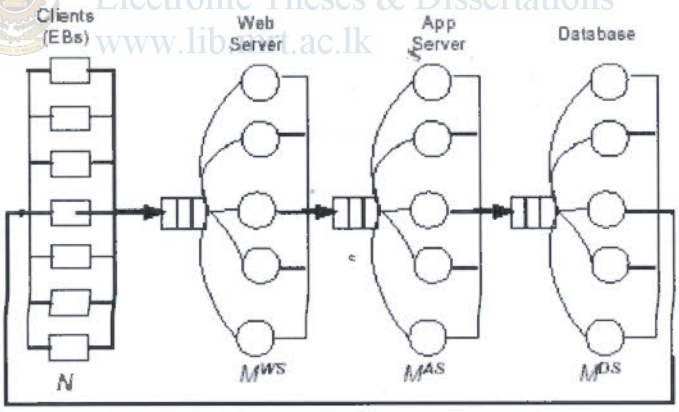


Figure 2-5: Queuing network model of 3-tiered Web service architecture [9]

Overall model for the 3-tier architecture proposed in [9] is a closed queuing network as shown in Figure 2-5, where N is the number of clients accessing the site, MWS, MAS and MDS are the number of worker threads at the web server, the application server and the database server respectively. Moreover, the average service times at each tier DWS, DAS and DDS are also model parameters. Other than that think time, Z is also considered in the model. Think time is defined in [12] as the time between displaying the requested results and issuing of a new request.

2.6 Performance Parameters

Many studies focus on time-related parameters such as response time, delay and page load time. Throughput is another widely used performance parameter. According to [12] response time is the speed of service from user's point of view whereas the throughput is that from system's point of view. System capacity is also a commonly evaluated parameter. In addition, access failures and error rates are also considered. A comprehensive set of parameters are presented in [13] : Total number of HTTP requests, HTTP requests per second, Number of good HTTP responses (200 OK), Number of bad HTTP responses (non-200 OK), Total HTML received [MB], Average HTML traffic [Mbit/s], Minimum HTTP response time [ms], Maximum HTTP response time [ms], Minimum number of HTTP connections, Maximum number of HTTP connections, Update SQL queries per second, Non-update SQL queries per second, Total SQL queries per second.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

However, Andreolini et al. highlights the importance of considering granularity level of performance evaluation so that the performance parameters must be selected accordingly [6]. Granularity levels given in the study are system level, node level, hardware resource level, software component level, process level, and function level listed in the order of varying from coarse-grain to fine-grain respectively.

2.7 Workload Characterization

Workload provides a compact description of the load by means of quantitative and qualitative parameters and functions [14].

Choice of workload model to test an e-commerce site is a problem by itself [6]. This observation will not change much regarding e-learning sites.

Typical web browsing workload model is oriented to define the number and the size of embedded objects and think time [6].

TPC-W is identified as the only complete benchmarking model available for e-commerce sites in several studies including [6]. There are little or no similar studies and benchmarking standards found regarding e-learning sites. Nevertheless, the work in e-commerce will give an idea as to how a workload model can be selected for an e-learning site. Moreover Andreolini et al. in [6] explains a TPC-W such as workload model which incorporates two scenarios, namely browsing and buying as well as the percentages of static and dynamic content.

In order to characterize the workload it is essential to capture user behaviour. Kotsis and Taferner in [12] capture user behaviour using log files and summarize the behaviour as sequence of user requests and the think times between requests. There are some interactions specified in the TPC-W benchmark specification [15].

Web interaction of users is often viewed in terms of sessions. A session is the period during which a user is active on Internet followed by a silent period as considered in [16]. According to [17] a session contains *temporally* and *logically* related request sequences from the same client. As claimed in [18] session consists of interdependent requests, hence the session based synthetic workload must reflect inter-request dependencies.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

There is a separate area of research called Web Usage Mining, which is the process of analysing web browsing behaviour. It is a three-phase process comprising: data preparation, pattern discovery and pattern analysis [19]. Data in this case is contained in the access logs. Logs record the web accesses sequentially according to timestamps. There are many technical issues involved in data preparation phase. User identification, session identification, caching issue, and page-view identification are major issues that must be addressed during the data preparation phase. Users are typically identified by means of unique IP addresses in Web access. Sessions are usually separated by setting a threshold value for the time duration between consecutive accesses [19]. However, threshold based mechanisms are error prone and clustering techniques are applied instead. A. Bianco et.al suggests a 3-step algorithm derived using clustering methods [16]. Hence, threshold based methods and clustering techniques can be considered as two of several methods found in literature for identifying sessions.

2.8 Workload Generation

Emulation is an important aspect in performance evaluation, capacity planning and workload characterization. Emulators are supposed to mimic realistic workload [20].

Client emulators are used to generate workload according to the workload model derived at the phase of workload characterization.

The benchmarking tools provide for generating workload by setting values for workload parameters. There are in-built benchmarking tools coming with LAMP software bundle for example AB (Apache Benchmark) tool in Apache server [21]. HTTPerf is another well-known tool for workload generation which can run a web session [22]. AutoBench is a tool which is implemented as a wrapper based on HTTPerf for automating the execution of HTTPerf. This tool has the capacity to repeatedly execute HTTPerf and get statistics in the form of a spreadsheet.

2.9 Workload Parameters

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

In [23], some workload parameters that are used to monitor the response time are the number of clients, page type (*static*, *cgi*), pages per connection, n/w drop rate, http version(1.0, 1.1) , RTT(ms), connections per sec, SYN drop rate and how long to run the workload. It will be beneficial to look at workload parameters of available workload generators. Simic et al. in [13] mentions about a workload generator for web servers called Web Server Load Tool (WSLT) which is said to simulate real user behaviour. The paper also lists few parameters found in WSLT such as increment step in terms of the number of connections, polling interval, maximum number of connections, time to run, and the number of browsers. The paper also mentions another commercial application used for database server benchmarking Quest Software's Benchmark Factory for Databases.

When issuing HTTP workload on a web server it is important to consider HTTP's interaction with TCP. For example, Heidemann in [24] shows that the interaction between TCP slow-start implementation and the HTTP's MIME data transfer mechanism in a particular experimental set up has caused server to wait for a long delay till an acknowledgement comes from the client.

2.10 Performance Monitoring

Monitoring is identified in [5] as the first and a key step in performance evaluation. Monitoring can be categorized as passive and active. Monitoring a system while it is being subjected to a synthetic workload is referred to as active monitoring. If the monitoring is done while real load is in progress without applying monitor's own load it is referred to as passive monitoring.

Olshefski et al. explains several monitoring mechanisms that can be used to monitor response time as perceived by the clients: 1. Periodically measuring the response time by means of geographically distributed set of monitors, 2. Instrumenting existing web pages with client side scripting which is a 'post-connection' approach, 3. Tracking the servicing of requests at the server, 4. Reconstructing response time using network packet traces [23].

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Other than monitoring the responses, the server side resource monitoring is also important. SNMP (Simple Network Monitoring Protocol) is a popular application protocol for polling servers for various counters that record resource utilizations such as memory available, swap space, and CPU time of users and system. This information is available in MIB (Management Information Bases) [25].

2.11 Experimental Design

There are two well accepted experimental models in web application benchmarking: virtual client testing model and the record-replay authoring model [18].

Proper experimental design should be devised such that maximum information can be gathered on the minimum number of experiments. It is essential to separate effects of factors from those of random variations.

The following terminology is important in systematic experiment design as given in [5].

- Response variable: outcome of experiment
- Factors: variables that affect the response variable. Also called predictor variables or predictors
- Levels: values that a factor can take, also called treatment
- Primary factors: factors selected for getting quantified
- Secondary factors: factors that are not selected for getting quantified
- Replications: the number of repetitions of experiments.
- Experimental unit: any entity used for an experiment

Designing an experiment involves specifying the number of experiments, deciding the factor-level combination for each experiment and selecting the number of replications.

Simple, full factorial and fractional factorial are three major types of experimental design. One factor is varied at a time in the simple design whereas all the combinations of all the levels of factors are considered in the full factorial design. Only a set of factors is considered in the fractional factorial design. A special case of fractional factorial design with two levels from each factor is denoted as $2^n r$ where n is the number of factors and r is the number of replication [5].

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

2.12 Evaluation Techniques

There are three main techniques for performance evaluation: analytical modelling, simulation and measurement. Analytical modelling can be used at any stage of a project, with a lesser amount of time, without the need for tools and instruments than analysts, but the achievable accuracy is low due to assumptions and simplifications. The evaluations done with analytical models must be verified by simulations or measurements. Simulation can also be used at any stage of the project, with heavy use of computer languages and tools, and can achieve moderate accuracy. This needs verification by measurement or analytical modelling. Measurement can be used only after the post-prototype stage, with heavy use of instrumentation, however the accuracy can vary. Results should be verified by analytical modelling or simulation [5].

LIBRARY
UNIVERSITY OF MORATUWA, SRI LANKA
MORATUWA

3 Methodology

The research methodology consists of identifying the system and workload, selecting and configuring testing tools, designing experiments to evaluate resource utilization at identified types of workloads, performing experiments and gathering data, and finally analysing and presenting data in order to draw conclusions as to how well the system performs at normal and peak loads and its scalability.

3.1 *Workload Identification*

The complaint which was heard unofficially from time to time about this system was that it takes a long time to load pages.

Poor page load times were experienced especially in instances where 100 students were sitting for an online quiz. In addition to this delay, during quizzes, database errors were experienced at the point of which final submissions were made by around 100 concurrent users.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Apart from complaints an important question asked by decision makers and users is that how many users the system can handle concurrently in different situations such as interactive tasks such as quizzes.

These abstract ideas had to be converted into test cases with quantifiable observations. The test cases so derived are given in Chapter 7.

A survey was used to capture workload characteristics from users' point of view whereas system logs were analysed to capture system workload characteristics.

3.1.1 Survey

A survey questionnaire was distributed among 130 students selected from different batches who were frequent users of LearnOrg-Moodle. This survey aimed at capturing user perception and user behaviour regarding the usage of Moodle. Another objective was to have a quantitative and qualitative evidence of the above mentioned complaints. The perception and behaviour vary according to the level of familiarity of the user with the

96423

system. The performance may vary due to different access technologies, and hardware and software in client computer. Hence the survey comprised of questions to achieve the following goals and the sub goals:

G1). To gather demographic information of sample population

a. Familiarity with Moodle

b. Frequency of usage

c. What activities in Moodle are used

G2). To identify access technologies and client side technologies

a. Type of connection

b. Type of browser

c. Other applications in the machine

d. Hardware details of the client machine

G3). To get user perception ahout responsiveness of the Moodle

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

a. Uncertainty of delay

b. Acceptable delay

c. Waiting times while working with Moodle

G4). To gather user complaints

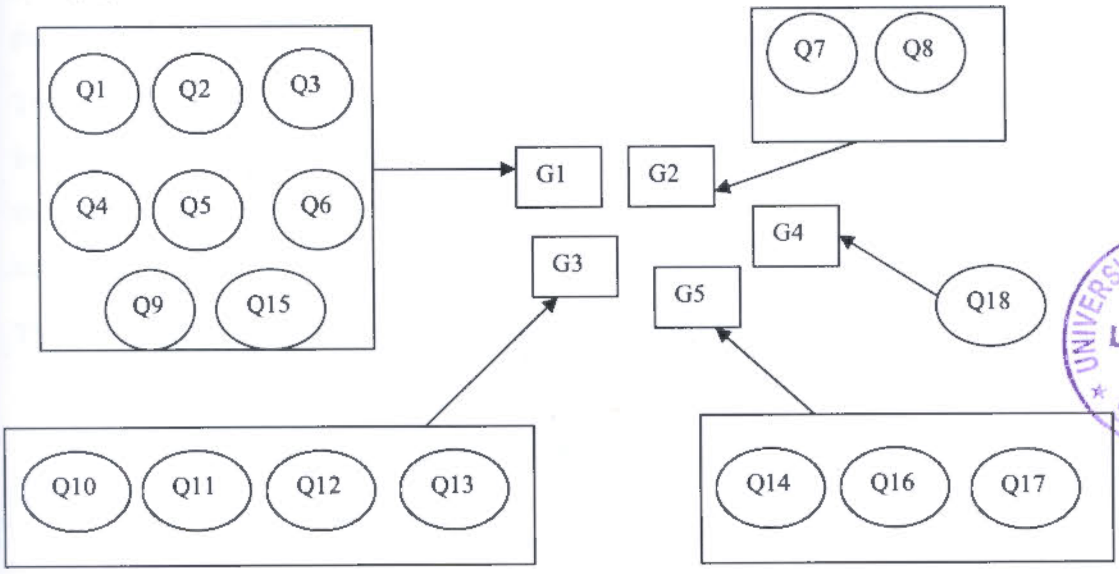
a. Errors in connecting to Moodle

G5). To capture user behaviour when working with Moodle

a. Submission pattern of online quizzes

h. Behaviour at non responsive pages

Survey questionnaire is attached in Appendix. Relevance of questions with respect to survey goals is as follows:



UNIVERSITY OF MORATUWA LIBRARY SRI LANKA

Figure 3-1: Mapping of survey goals and questions

Results of the survey are analysed under the System Identification in Chapter 4.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

3.1.2 Log Analysis

LAMP systems keep records of workload traces from the point at which a request appears at the web server and until the response gets back to the client. Httpd access logs and mysqld query logs are two such examples. Moodle records user related details such as requested URLs and time of request in its mdl_log table. The httpd access logs were analysed and the results are elaborated in Chapter 5.

3.2 System Identification

System identification phase aimed at finding out system parameters.

LearnOrg-Moodle runs on a medium sized LAN consisting of several dedicated servers such as DNS, Proxy, and a number of intermediate networking devices. It was required to limit the scope of the system by defining a boundary which includes most important components only. The contribution of the network nodes to the overall response time in accessing Moodle was analysed by undertaking a preliminary test as follows. Moodle was accessed from one of the LANs within the department network and the packets of the communication were captured at the client machine. By considering the protocol of the packets and the timestamps, the communications were broken down to several steps.

This study resulted in removing DNS lookup and the propagation delay over network from the response time due to their negligible effect. Hence the system was confined to the LAMP based server machine and this will be referred hereafter as the SUT (System under Test).

The following steps were carried out to investigate further into the SUT.

- Defining the system boundaries
- Identifying services and outcomes of the system
- Selecting performance metric
- Listing parameters
- Selecting factors to study

3.3 System Monitoring

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

There are two types of monitoring involved in this study. One is the resource monitoring of the server and the other is the monitoring of the communication between server and the client.

In order to identify the loading condition in terms of resource utilization of the web server, there had to be a monitoring mechanism. Initially, to address this requirement MRTG [26] was used by configuring it to monitor resources. It was observed that MRTG can show readings averaged for periods of 5 minutes and the expected granularity level for monitoring resources according to the user interactions was required to be finer than what MRTG could offer. Therefore RRDtool [27] was selected as the resource monitor.

Both MRTG and RRDtool use SNMP as the underlying protocol to get the information related to system resources such as CPU, memory and I/O. Read only access to SNMP MIBs is sufficient and also advisable for security reasons. Though SNMP is not advisable to be enabled in an operational server for security reasons, here the server is monitored by isolating it from the operational environment. Therefore it does not make a huge security issue. However the newer versions of SNMP come with data encryption and other security enhancements so that it will be harmless to use the method mentioned here on the operational system as well.

The workload generating tool mentioned in Chapter 5 provides facilities to record the parameters such as response time and content of the replies involved with the communication between the server and the test system.

The resource utilization and the communication had to be synchronized so that the changes in the server resource utilizations can be explained with respect to changes made in the applied workloads by the test system. Resources were monitored remotely from the same client that monitored the response times. Therefore the timestamp of the client was used for both types of readings.

3.4 Performance Testing

Throughout the design and implementation of performance experiments a test bed which mimics the actual system was used. This environment was implemented on a set of machines to which all administrative privileges were available.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The technique devised for performance testing was to replay the recorded workload traces in controlled manner. This can be considered as a simulation technique. The workload was recorded such that it exercises different types of workload that were identified at the workload characterization phase.

A separate test environment was implemented with a *test-server* and a *test-client-system*. *Test-server* was installed in a low-end machine but with the operating system and software as similar as possible to the operational Moodle server. This was to design experiments without jeopardizing the operational system.

Test-server has the following system specifications:

Processor: Intel P4 2.8 GHz, Cache 1024 KB

RAM: 512 MB

Linux: version 2.6.9-5.ELsmp

Moodle: version 1.8+ (a copy of the Mihindu Moodle)

Test-client-system was installed with needed tools for workload recording, replaying and monitoring. This system facilitated development of test scripts. Workload recording was done using a Perl script that was written with the Perl modules available at CPAN repository as discussed in Chapter 5. Workloads were replayed changing parameters such as number of sessions and rate using a command line tool called *HTTPerf* and this is also discussed in Chapter 5. Resource monitoring was done using SNMP. SUT was polled from the Test-client-system and the MIB values were stored in a RRD (Round Robin Database) as discussed in Chapter 6. RRDTool was found out be a flexible tool for manipulating this database and it provided facilities to graph MIB values related to resources of SUT such as CPU, memory and I/O.

This test environment was used as the prototype throughout the study and the results being analyzed in Chapter 6 onwards pertain to this test environment, not to the operational environment.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

4 System Identification

This chapter gives the details of the actual operational system from which the decisions are taken with regard to the parameters of importance for the implementation of the test environment.

In order to identify parameters that affect performance from the time user sends a request to the SUT till the response appears at the interface to the client, the system is viewed in the following viewpoints in this chapter: design, deployment, implementation, process and users. These views are in accordance with the 4+1 model discussed in Chapter 2.

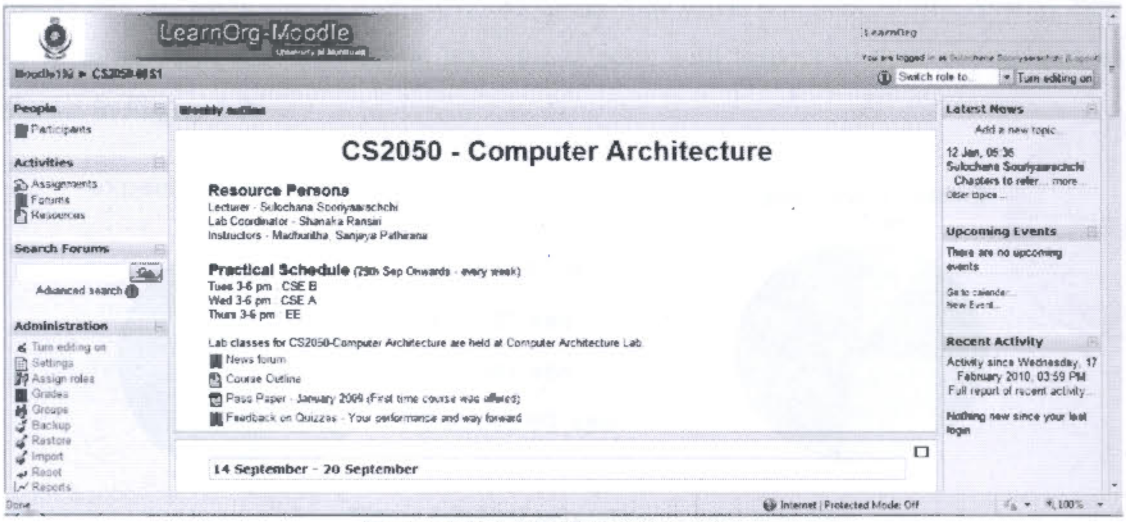
4.1 *Design*

Moodle which is designed based on constructivism provides tools for peer interaction and collaboration such as forums, wikis and chats. It provides an academic environment that can hold many courses and materials in different formats including video. One particularly important activity in Moodle is online quiz. Further the online assignment submission is a widely used facility.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The users need to get authenticated in order to access the materials and to participate in activities. Authentication is done mainly against an external database.

Given in Figure 4-1 is a course page of a subject offered over the Moodle:



CS2050 - Computer Architecture

Resource Persons
Lecturer - Sulochana Sooriyaarachchi
Lab Coordinator - Shanaka Ransiri
Instructors - Madhushitha, Sanjaya Pathirana

Practical Schedule (29th Sep Onwards - every week)
Tues 3-6 pm CSE B
Wed 3-6 pm CSE A
Thurs 3-6 pm EE

Lab classes for CS2050-Computer Architecture are held at Computer Architecture Lab
News forum
Course Outline
Pass Paper - January 2009 (First time course was offered)
Feedback on Quizzes - Your performance and way forward

14 September - 20 September

12 Jan, 05:36
Sulochana Sooriyaarachchi
Chapters to refer... more...
Older topics...

There are no upcoming events

Activity since Wednesday, 17 February 2010, 03:59 PM
Full report of recent activity...
Nothing new since your last login

Figure 4-1: A Moodle course page

The application demands users to authenticate themselves against the database after which a session is maintained. Sessions can be maintained in Moodle by two means; using cookies or cookieless sessions with PHP session IDs.

The dynamic content of the courses is queried from the *moodle* database by relevant PHP scripts.

The function provided to the end user in terms of the learning experience by Moodle includes: providing a course page with links to various learning materials, learning activities such as assignment submission, wiki, quizzes, and forums.

4.2 Users

Users of the system are the students around 1000 in number from all the engineering courses in the university. Additionally, there are about 50 people playing the role of teacher who develop content in the courses and manage the courses. But the future demands may be even bigger classes with distant mode accesses.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The performance issue is mainly felt and caused by the student users. It was observed that it takes long time even to login to a Moodle course page when around 100 students concurrently sit for quizzes. The database errors occurred at the final submission of quizzes.

According to the survey mentioned under Methodology in Chapter 3, it was observed that nearly half the users expect Moodle to have page load times less than 10 sec. Nearly, 50% of them perceive that the system response is poor with respect to their expectation as shown in Figure 4-2.

UNIVERSITY OF MORATUWA
LIBRARY
SRI LANKA

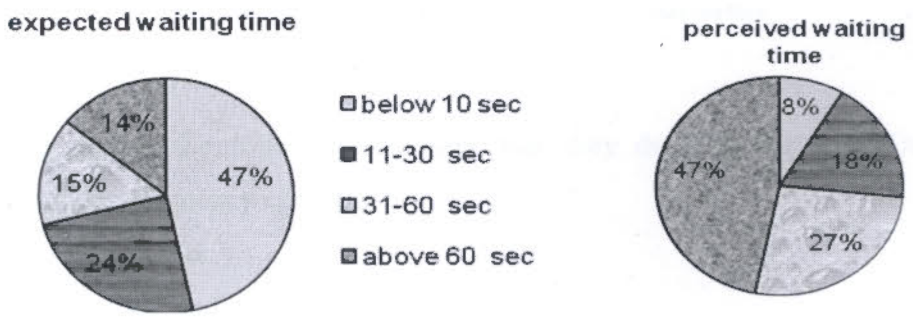


Figure 4-2: User perception on page load time

It was observed that the students submit the quizzes at the last moment in bulk. The usage habit in this regard was also surveyed. It shows that approximately 70 % of the users do last minute submission where they click *Submit all and finish* option in Moodle quizzes as per Figure 4-3.

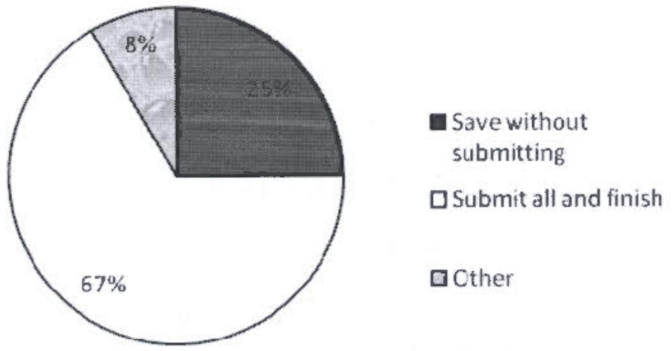


Figure 4-3: User behaviour - Quiz snbmission in Moodle

Another important behaviour is that nearly 70% of the users start interacting with the system as soon as the desired portion of the page is loaded without waiting for the whole page to load as shown in Figure 4-4.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

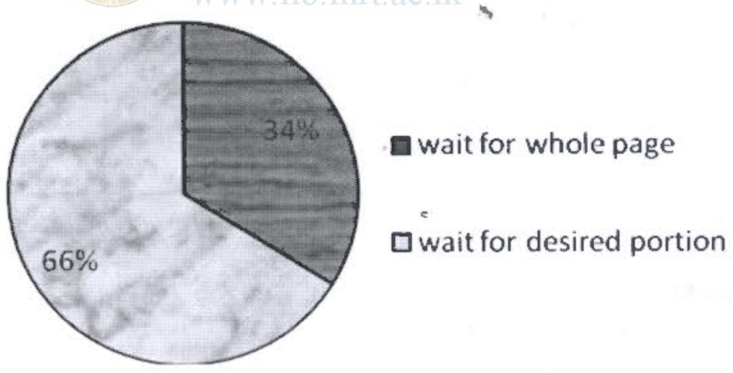


Figure 4-4: User behaviour - Waiting for interaction

The users ranked the following activities that they do with the LearnOrg-Moodle according to the frequency of usage:

a. View pages
b. Post text in forum
c. Post text in wiki

d. Upload files

e. Online quiz

f. Open link to pdf file

g. Open link to web pages

h. Download files

i. Chat

A point-scheme was used in order to derive a common ranking of activities according to how frequently the students used them. Rank 1 through 10, were allocated with points 10 through 1 so that the top ranking operations get the highest points, see Table 4-1.

Table 4-1: Ranking of activities in Moodle by how frequently those are used

Activity	1	2	3	4	5	6	7	8	9	10	Aggregated Points	General rank
download file	470	387	136	35	24	5	4	0	0	0	1061	1
View	690	54	48	49	24	10	8	3	4	0	890	2
open pdf	190	288	216	84	42	10	8	0	2	0	840	3
upload file	120	108	136	119	168	55	24	12	0	0	742	4
open web	140	99	152	210	72	30	0	3	2	0	708	5
post to forum	20	90	104	77	120	100	36	9	0	0	556	6
online quiz	20	54	56	42	30	60	76	33	10	0	381	7
post to wiki	0	27	32	28	6	40	60	66	10	1	270	8

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Users were also asked to rank the operations done on the LearnOrg-Moodle according to the time consumption.

Similar to the analysis described above regarding Table 4-1, a common ranking for the operations were derived as shown in Table 4-2 according to time consumption.

Top most rank corresponds to the most time consuming operation.

According to Table 4-2, it is observed that file handling is the most important operation and the users think that it is quite frequently used and the most time consuming operation. This is a high level statement which is not conclusive enough without the type and sizes of files that the users deal with. Hence next level of analysis of user behaviour was done using system logs giving emphasis in file sizes and types they accessed which is discussed in Chapter 5.

Table 4-2: Ranking of operation in Moodle by time consumption

Most time consuming	1	2	3	4	5	6	7	8	Aggregated Points	General Rank
download file	490	180	56	56	6	10	12	9	819	1
Login	200	144	152	112	36	35	20	18	717	2
upload file	190	324	88	21	60	0	8	6	697	3
load course	140	144	152	119	48	25	24	9	661	4
post to forum	40	45	96	98	84	70	8	6	447	5
submit online quiz	80	45	128	14	60	50	8	3	388	6
post to wiki	10	27	56	28	48	25	52	12	258	7
post in chat	0	36	16	63	24	20	24	45	228	8

The above results show that the user behaviour depends on the composition of the courses in the LearnOrg-Moodle where this system is mainly used to keep lecture materials online in pdf, ppt, doc, odt, formats. Only few courses are observed to have video materials. Among interactive activities forums are quite frequent in many courses. Chats are not at all used as an activity under courses currently. Also the students rarely use chat for their interaction since more sophisticated popular chat applications are available outside Moodle.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

4.3 Deployment

Mihindu-Moodle is installed as a LAMP system with the following hardware, OS and Software specifications:

Specification of server machine:

CPU: 2 Dual, Intel(R) Xeon(TM) CPU 3.20GHz

2 MB Level 1 Cache per processor

RAM: 4 GB

Swap: 8 GB

Software:

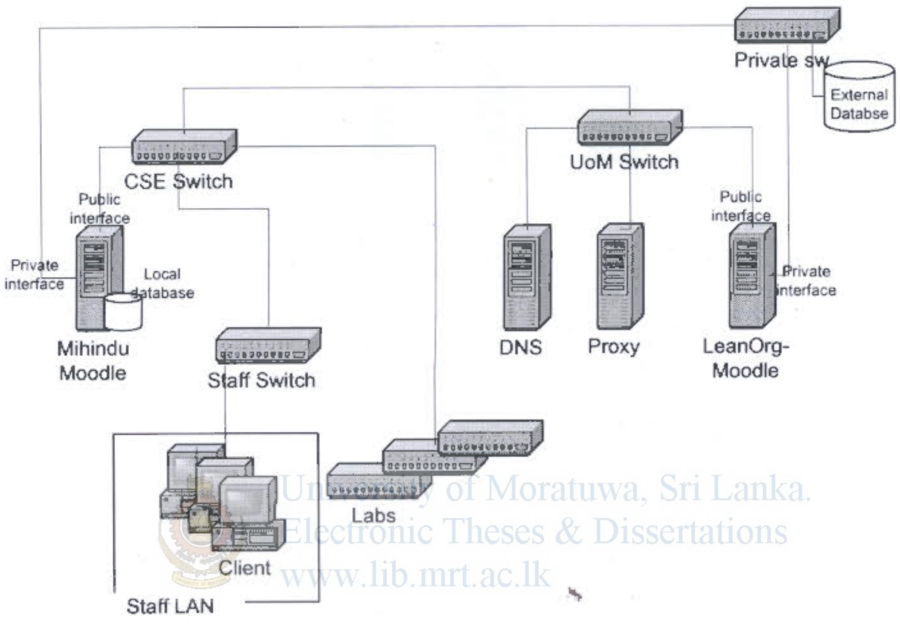
Moodle 1.8+

Apache 2.0.52

PHP 4.3.9.

MySQL 4.1.20

Redhat Enterprise Linux 4.1



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 4-5: System Implementation - Connectivity

The system implementation is given in Figure 4-5. The most common sequence of operations that user does is

- Opening Moodle home page
- Log in
- Entering to a course
- Log out

In order to have an approximated estimation of the time consumption, this sequence of operations was initiated by a client residing within the Staff LAN in FF and IE browsers and packets were captured by performing it 10 times. Objective was to identify the response times of each operations or events within operations.

When the packets were analyzed for above application level operations, system level activities such as DNS lookup and TCP connection establishment were also considered separately.

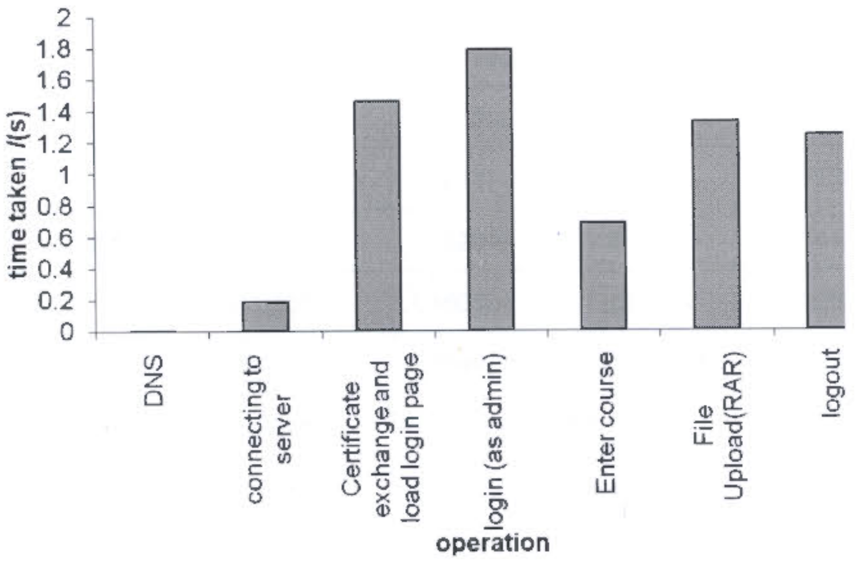


Figure 4-6: Response time of common set of operations

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Accordingly, the DNS lookup time as well as the time taken to establish the connection to the Mihindu-Moodle server is comparably small as shown in Figure 4-6. These depend almost all on the network. Hence the network is assumed not to be the bottleneck for the time being. So the study concentrates on:

- Web server
- Database server
- Interactions of Moodle application with the resources

Moodle application is also assumed to be a black box so that only the interaction of the application with the system resources is considered.

Since time consumption in file handling of the Mihindu-Moodle is a concern from the users' point of view it was also studied using packet analysis. When uploading files of type such as *pdf*, *ppt*, or *doc* they open up in the browser which need not be considered in the response time. Hence archive file was selected. File sizes of 1.4 MB, 2.8 MB, 4.1 MB, 5.6 MB, and 6.9 MB were used and each was uploaded for times both in FF and IE and the observations are given in Table 4-3.

Table 4-3: Sigle user file uploads

	Time taken to upload in Mihindu-Moodle (in sec)				
Trial	**1.4MB**	**2.8MB**	**4.1MB**	**5.6 MB**	**6.9 MB**
1	1.349416	1.206145	1.294106	1.795957	2.010419
2	1.002326	1.020135	1.145812	1.293164	1.447202
3	0.872929	1.001548	1.12827	1.293544	1.49106
4	0.939397	1.008756	1.162046	1.272042	1.43447
5	0.857115	1.010766	1.160584	1.254743	1.420931
6	1.182489	1.184056	1.354903	1.432675	1.967309
7	1.018446	1.218626	1.304202	1.544348	2.144001
8	0.950801	1.102283	1.413927	1.470495	1.900801
9	0.946849	1.124129	1.416137	1.52547	1.969046
10	1.184282	1.200273	1.639126	1.435222	1.704565
Mean	**1.030405**	**1.1076717**	**1.3019113**	**1.431766**	**1.7489804**
Variance	0.025104013	0.008313026	0.026248564	0.027863075	0.078848469

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

UNIVERSITY OF MORATUWA, SRI LANKA
LIBRARY

On average to upload a file of size in the range 1 to 7 MB in the Mihindu-Moodle it takes around 1.35 sec.

Since this coarse grain analysis is done using only a single client machine it does not capture the effect of concurrent users. Hence it is required to emulate the real users, and this is investigated in detail in Chapter 5.

4.4 Process

Process captures the sequence of operations from the time a request comes from a client over the network till the response gets back to the client.

- TCP connection establishes between server and the client
- Web server listens for HTTP requests

- Translation of URI into filename, check the access privileges, validating user id in the HTTP request, authorizing the user
- Determine the MIME type of the requested object
- Processing the requested object
- Send the data to the client
- Log the request

Processing of the object is done here in PHP scripts in the Moodle software. The dynamic content is generated by querying the MySQL database. The queries can be logged at the point of arrival at the server as well as after the query is committed.

Since both Apache web server and the MySQL database server are multi-threaded/process applications they spawn more threads or processes as the number of simultaneous requests increases.

There are caching mechanisms available at various stages of the process both implemented by Apache and MySQL. Also Linux maintains caching and memory management techniques as well as processor scheduling techniques.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The parameters that govern the resource and process management are found in configuration files and those can take different values. Httpd directives in Apache configuration file, PHP configurations, MySQL variables and Linux settings comprise the system parameters. Considering all the parameters is impractical within the scope of this project. Therefore the server parameters are kept constant and only the workload parameters are changed.

4.5 Conclusion

Since the impact of the network was observed to be negligible, it was decided that the network should be left out from the test bed. Hence the test bed was developed in such a way that it consists of the server and one client machine directly connected to the server. The connection was a 100 Mbps Ethernet so that its bandwidth was large enough compared to actual system.

The study was purposely biased towards the client's view of the system so that the monitoring happens at the client machine and it is done at the interaction of the client with the server's front end: that is the Apache web server. Consequently, the database interactions were considered to be transparent to the user.

The need for emulation of actual human users was emphasized in the above analysis in order to answer very practical questions such as "how many quiz users can your Moodle installation handle". The workload was determined in a manner that it enables to capture the Moodle functionalities ranked above by the users. Hence more effort was put on deriving a better mechanism to emulate real human users and generate workload in controlled manner.

Tests were designed such that they capture the concurrency aspect of workload. The single physical client machine in the test bed was configured to emulate multiple users. Also the file handling was an aspect of the system that needs testing in general, apart from the Moodle functions.

Accordingly the following test goals were decided:

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

- To find the maximum number of human users that will overload the system
- To find what Moodle activity causes the system to be overloaded most
- To find what user behaviours cause overloading

To address above cases it was needed to identify what observation shows the overloading condition. The server resources and response time the client experience were selected as the factors to be monitored, so that overloading conditions and clients' view of the system can be decided.

5 Workload

5.1 Workload Characterization

As mentioned in the above chapter it was observed that the loading according to Moodle activities gives a realistic meaning to the workload from the users' point of view. This makes it easy to map number of real users to the load on the system.

According to the conclusions derived in Chapter 4 the study considers only the front end or the Apache web server as the point of applying workload. Hence, the loading happens in terms of HTTP requests.

The workload of HTTP requests can have two forms: 1. Request oriented workload 2. Session oriented workload.

In real world scenario the workload appears as web sessions, especially with regard to applications such as Moodle. However, the workload applied on Moodle corresponds to the following categories and at times request oriented workload will also be useful as explained below:

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

1. View pages: many Moodle sessions consist of sequence of page views. The pages are partly static and partly dynamic. The workloads should distinguish between static page views and dynamic page views.
2. Post text in forum and in wiki
3. Uploading and downloading files: this can be tested as the file handling capability of the server and that of the Moodle application separately. Other than that the type of file and the size of file are important parameters in this workload.
4. Online quiz: workload in this category may vary according to the type of questions, whether the questions are shuffled or not, if the questions span across multiple pages or not, how they are submitted

In this study a **Moodle session** is considered as the sequence of HTTP requests sent by a single Moodle account from the time of login till the end of desired Moodle operation.

The HTTP requests are supposed to accompany with POST data and think time between requests.

Concurrency of sessions is a special factor to be considered when applying session oriented workload. Statistical properties of inter-session arrival time are important in testing a site such as Moodle. Some number of sessions applied at once and the same number applied with an inter-session arrival time may give different results and both situations may be possible in real world. Arrival of sessions may also be at uniform rate or bursty.

5.2 Tools and Techniques

There were two techniques in generating Moodle sessions.

(1) Extracting sessions from the past access traces

(2) Recording sample sessions

5.2.1 Extracting Sessions from Logs:

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

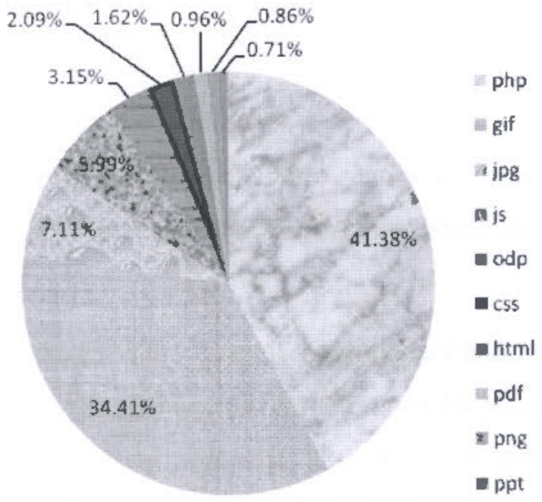
Several tools were tested for log analysis, namely; Webalizer, Xlogan, Awstat and Web Log Filter. They extract information in bulk such as total numher of hits made to a page and file sizes of mostly requested pages. Existing tools give information such as daily, hourly, monthly usages in terms of hits and file sizes. They do not go into user by user analysis.

Hence it was required to develop my own script for the purpose of user session level data analysis. Though users could be separated hy IP address it does not represent users realistically. It demands more web mining techniques to extract realistic sessions.

A Perl script was developed to extract the requests by IP address and then to plot the timestamps against each request. An attempt was made to use a clustering technique for identifying the sessions by considering the inter-request times. Since the method given in next section is simpler and suffices for the scope of this project the web mining was not performed any further.

5.2.2 Workload Characteristics from Logs

Data extracted using Awstat tool was important for identifying the types of files that were requested from the operational e-learning environment and the percentage of requests in which each type of file was requested. The results are given in Figure 5-1 and Figure 5-2.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 5-1: File types requested

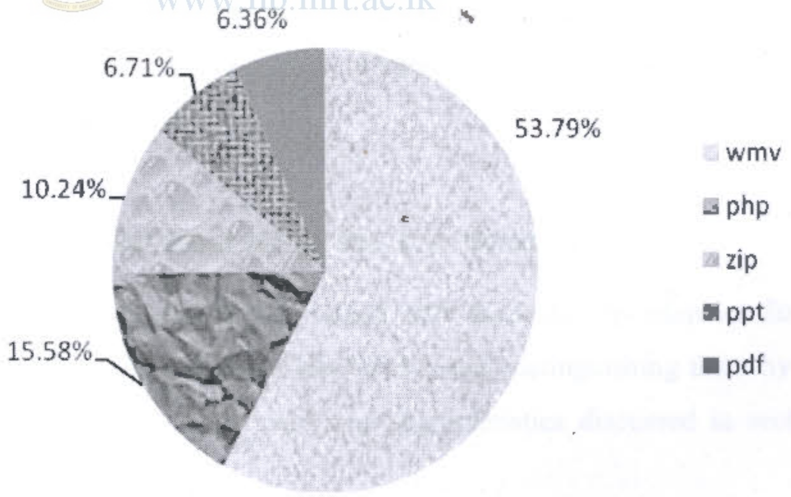


Figure 5-2: Bandwidth usage by file types

According to the analysis of traffic of the past 15 months, *wmv* account for 54% of the total bandwidth, while the *php* files account for 15%, and *zip* files for 10%. *ppt* files and *pdf* files have occupied equally around 6% of bandwidth each.

When the numbers of requests are concerned *php* files are the most requested as they account for around 40% of the requests. *gif* images account for about 35% and *jpg* images are 7%. It was observed that due to large number of requests for *odp* files in few months and without any requests in rest of the months still the averaged percentage of *odp* became 3%. This shows that depending on the interest of the course creator the type of files being requested can change immensely.

Though a large amount of bandwidth is consumed by *wmv* files the number of requests accounted for them is a small amount such as 0.1% of the total requests. This proves that the video files are the largest ones that are requested from the system. Therefore the synthetic workloads that represent large files are advisable to be of *wmv* type. Small files should be *gif* files.

When *php* files are concerned they are affected by database accesses. Parameters such as the types of queries, and the size and the collation of tables may affect the database access performance. Hence the types of *php* pages should be selected from Moodle site such that a special consideration is given to SQL queries they incur. The query logs in MySQL provides for monitoring the database interactions. However, in this research, MySQL operations are considered transparent to the user and the point of interaction considered is the front end web server only. Therefore, this part of the work was considered outside the scope and was not addressed.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

5.2.3 Recording and Replaying Sample Session

This approach provides sessions that reflect inter-requests dependencies far better than the extractions of logs. The sessions can be recorded distinguishing them by the Moodle application functions so that the workload characteristics discussed in section 5.1, are adhered to.

Hence it was decided to record the sample user sessions and to replay them in a controlled manner to emulate Moodle users.

5.2.4 Tools for regenerating HTTP workloads

Tools must apply the transcripts of workload traces, and output the response times and resource utilization. The commonly cited tools for replaying HTTP traffic in industry are Apache Benchmark, JMeter, HTTPerf, and Siege.

Apache Benchmark: In-built with Apache server coming with the RHEL CDs that were used for the SUT. It's a command line tool. It supports POST data so that authentication is testable. Limitation is that more than one URL cannot be applied at a time.

Siege: This is a command line tool. It supports sending cookies but does not receive cookies.

JMeter: This can run custom scripts of HTTP sessions with cookies. Uses more resources to run the tool than others

HTTPerf: Works with cookies. It can act on a sequence of URLs at a time. HTTPerf was found to be a light weight, open source tool for playing web sessions. It is a command line tool, which was developed by Mosberger from HP company [22].

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

5.2.5 HTTPerf and Workload Recorder

HTTPerf is one of the robust among the few tools that have session based measurement capability. It has more flexibility to set parameters such as inter-session arrival times as well as the inter-request arrival time within sessions. The ability of HTTPerf to sustain itself while executing a very larger number of sessions is another advantage compared to the tools such as WebStone and SPECWeb. The developers of HTTPerf also have taken into account the synchronization and concurrency in distributed systems such as web based systems. Network I/O statistic from HTTPerf helps verifying that network is not saturated. Also the error statistics help determining validity of results and also they help detecting the overloading conditions.

A sample *httperf* command used to play web sessions is given in Figure 5-3. *Wsesslog* is the option that provides the facility to run web sessions that is stored in a text file. This particular session file is of the format given in Figure 5-4.

```
#httperf --server mihindu.uom.lk --port 80 --wsesslog 1,1,sessionfile --add-headers "application/x-www-form-urlencoded" --print-reply > outfile
```

Figure 5-3: A sample httperf command

According to Figure 5-4 the *think* parameter gives the inter-request time within a session.

For recording this session in the above format the Perl script given below was used. This script is written using third party modules such as *HTTP::Proxy*, *HTTP::Recorder::Httperf* obtained from the well known CPAN repository. This script runs as a proxy and records the session between a web browser and the Moodle server.

```
/moodle18/login/index.php method=POST
contents="MoodleSession=2fa9e34da80f9527245f8db2df955c66&MoodleSession=2fa9e34da80f9527245f8db2df955c66&username=moodleadmin&password=xxxxxxx&testcookies=1" think=102881
        /moodle18/login/index.php?MoodleSession=2fa9e34da80f9527245f8db2df955c66 method=GET think=3
        /moodle18/theme/standard/styles.php?MoodleSession=2fa9e34da80f9527245f8db2df955c66 method=GET think=2
        /moodle18/theme/formal_white/styles.php?MoodleSession=2fa9e34da80f9527245f8db2df955c66 method=GET
        /moodle18/lib/javascript-static.js?MoodleSession=2fa9e34da80f9527245f8db2df955c66 method=GET
        /moodle18/lib/javascript-mod.php?MoodleSession=2fa9e34da80f9527245f8db2df955c66 method=GET
        /moodle18/lib/overlib.js?MoodleSession=2fa9e34da80f9527245f8db2df955c66 method=GET
        /moodle18/lib/cookies.js?MoodleSession=2fa9e34da80f9527245f8db2df955c66 method=GET
        /moodle18/lib/ufo.js?MoodleSession=2fa9e34da80f9527245f8db2df955c66 method=GET
        /moodle18/theme/formal_white/logo_small.jpg method=GET
```

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

UNIVERSITY OF MORATUWA LIBRARY SRI LANKA

Figure 5-4: A sample session file

The proxy setting in the browser was made to the machine where the recorder script is running and the port should be set to the value in the script.

```
#!/usr/bin/perl
        use HTTP::Proxy;
        use HTTP::Recorder::Httperf;
        my $proxy = HTTP::Proxy->new();
        # create a new HTTP::Recorder::Httperf object
        my $agent = new HTTP::Recorder::Httperf;
        # set the log file (optional)
        $agent->file("/tmp/myfile");
        # set HTTP::Recorder as the agent for the proxy
        $proxy->agent( $agent );
        # start the proxy
        $proxy->port('3128');
        $proxy->start();
        1;
```

Figure 5-5: Workload recording script

5.3 Conclusion

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

According to the survey results given in Chapter 4 and the log analysis given in section 5.2.2 of this chapter, the following types of sessions were selected as workload types:

1. Page view
    a. Static text page
    b. Moodle front page
    c. Login and view a course page
2. Quiz session: a quiz with 10 MCQ questions. The sequence of interactions is given in Figure 5-6.

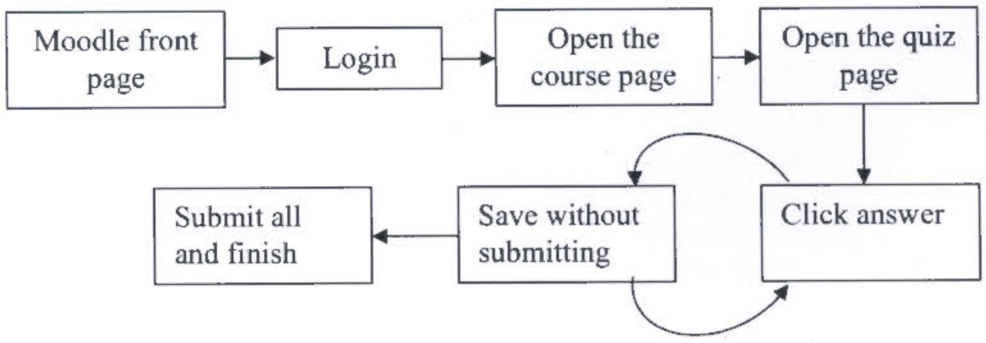


Figure 5-6: Sequence of interactions in a quiz session

3. File upload/download request

   Large file – a wmv file of size 53.5 MB of a 1-hour video lecture

   Small file – a gif file of size 15.5 KB

   The file handling within Moodle and outside Moodle was separately looked at.

It was decided that the Perl script mentioned above should record the sessions and these sessions should be replayed using HTTPerf.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

6 Experimental Design

The question of interest in the experiments in this study are 'how does the number, inter arrival time distributions and the type of sessions affect the resource utilization of the server and the response time seen at the client'.

6.1 *Outputs from the Experiments*

Resource utilization of the SUT is an important output so that resource monitoring is critical. SNMP MIBs of the test server given in Table 6-1 were polled remotely from a directly connected machine.

Table 6-1: Resource monitoring parameters and their meanings

MIB	**Description (based on /usr/share/snmp/mibs/UCD-SNMP-MIB.txt)**	**Units**
memAvailSwap	The amount of swap space currently unused or available	kB
memAvailReal	The amount of real/physical memory currently unused or available	kB
memTotalFree	The total amount of memory free or available for use on this host. This value typically covers both real memory and swap space or virtual memory.	kB
memShared	The total amount of real or virtual memory currently allocated for use as shared memory. This object will not be implemented on hosts where the underlying operating system does not explicitly identify memory as specifically reserved for this purpose.	kB
memBuffer	The total amount of real or virtual memory currently allocated for use as memory buffers.This object will not be implemented on hosts where the underlying operating system does not explicitly identify memory as specifically reserved for this purpose.	kB

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

memCached	The total amount of real or virtual memory currently allocated for use as cached memory. This object will not be implemented on hosts where the underlying operating system does not explicitly identify memory as specifically reserved for this purpose	kB
ssSwapIn	The average amount of memory swapped in from disk, calculated over the last minute	
ssSwapOut	The average amount of memory swapped out to disk, calculated over the last minute	
ssIOSent	The average amount of data written to disk or other block device, calculated over the last minute. This object has been deprecated in favour of 'ssIORawSent(57)', which can be used to calculate the same metric, but over any desired time period	
ssIOReceive	The average amount of data read from disk or other block device, calculated over the last minute. This object has been deprecated in favour of 'ssIORawReceived(58)', which can be used to calculate the same metric, but over any desired time period	kB
ssCpuUser	The percentage of CPU time spent processing user-level code, calculated over the last minute. This object has been deprecated in favour of 'ssCpuRawUser(50)', which can be used to calculate the same metric, but over any desired time period.	
ssCpuSystem	The percentage of CPU time spent processing system-level code, calculated over the last minute. This object has been deprecated in favour of 'ssCpuRawSystem(52)', which can be used to calculate the same metric, but over any desired time period.	
ssCpuIdle	The percentage of processor time spent idle, calculated over the last minute. This object has been deprecated in favour of 'ssCpuRawIdle(53)', which can be used to calculate the same metric, but over any desired time period	

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The major server resources in concern are CPU, Memory and I/O. Due to the complexity of monitoring all the possible parameters, it was decided that a set of parameters to be selected such that major resources are covered as listed in Table 6-1. *ssSwapOut,* *ssSwapIn, ssIOSent,* and *ssIOReceived,* represent I/O operations. *ssCpuUser, ssCpuSystem,* and *ssCpuIdle* represent CPU utilization. *memTotalFree, memShared, memCached, memBuffer, memAvailReal* and *memAvailSwap* are important as memory related parameters. Hence these few parameters were selected to suffice the scope of this research.

Monitored data was stored in a Round Robin Database. According to [28] a Round Robin Database consists of a fixed number of data items and a pointer to the current element as shown in Figure 6-1.

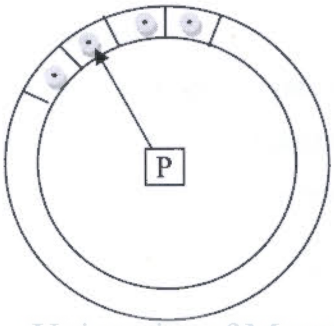


Figure 6-1: Round Robin Database structure

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Round Robin Archives (RRA) carry variables that store different data items. For each parameter being monitored a separate RRA was defined in the RRD. The number of elements of the RRA should be large enough to store the number of readings taken for each parameter. When all the elements of the round are filled, the new data will replace the initially inserted values. This way the database does not grow indefinitely so that no additional maintenance is needed.

RRDTool is a command line tool working on Unix environments, for manipulating RRDs. It provides facilities for creating, populating, and even calculating formula on data in the RRDs. It also provides graphing facilities.

RRDTool allows the data item to be inserted into RRA, indexed by a timestamp. In RRDtool's terminology, *step* is the time duration between two data elements when the *RRA* is updated. *Heartbeat* is the timeout for receiving a data item.

Httperf output given below represents the client perceived performance parameters:

- Number of initiated TCP connections
- Number of requests sent
- Number of replies received
- Rate of initiating connections (as well as the initiation time per connection)
- Maximum number of connections initiated simultaneously
- Lifetime of the successful connections (time from the initiation till closing of connections on which at least one request was completed successfully)
- Rate at which requests are issued
- Average size of the HTTP requests
- Response rate and the number of response rate samples collected at every 5 sec intervals. For better statistics HTTPerf recommends to have at least 30 samples so that tests are recommended to be run for more than 150 sec.
- Response time between sending of the last byte of the request till receiving the first byte of response.
- Response size
- Number of replies baving each HTTP status code range.
- Number of times a connection failed due to client timeout.
- Number of times a connection failed due to socket level timeout.
- Number of times connection attempt failed due to server refusing
- Number of times a connection failed due to server resetting
- Number of times HTTPerf went out of file descriptors (fd-unavail). This happens when client is overloaded. Also number of times the client's file descriptor table was full (ftab-full), Number of times the client run out of TCP port numbers (addrunavail), also number of other unknown errors with error numbers (other)

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Main client-perceived output is selected to be the *reply time* given as an output by HTTPerf.

6.2 *Workload Configuration*

If the sources of variability are not properly controlled then the experimental bias may occur. Therefore, sources of variability had to be identified to improve the precision of results. Replication improves the confidence level of the results.

The prime cause of the variations of the utilization must be the workload being applied. Hence the following control experiment was also conducted.

Too many factors with too many levels cause complex experimental design. Therefore the configuration of the system is kept constant to figure out the effect of workload related factors.

The default Apache configurations such as the followings were kept as they are;

- Timeout (maximum time that the server waits for a client's response) – 120 s
- KeepAlive (if persistent connections are supported) – On
- MaxKeepAliveRequest (maximum number of requests per connection) – 100
- KeepAliveTimeout (maximum allowable time between two requests over a persistent connection) – 15 s

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Input parameters of HTTPerf:

- hog: to use as many TCP connections as needed without limiting to the ephemeral ports that are 1024 through 5000.
- num-calls: total number or requests to be issued on each connection. For this value to be greater than 1 the server must support persistent connections.
- num-conns: total number of connections to create
- rate: the fixed rate at which the connections are created. 0 results in creating connections sequentially.
- period: inter arrival time of sessions, this can be set to deterministic/exponential (Poisson)/ uniform distributions
- timeout: the amount of time HTTPerf is willing to wait for a response from the server.
- wsesslog: the session file and the number of times it should be run are given in this option

The following test cases were performed to identify the effect of workload parameters on response time.

Case 1: A simple html file was requested. The system configurations were fixed and the workload configurations were changed according to Table 6-2.

Table 6-2: workload configuration for case 1

Variable	Factor	Level -1	Level 1
A	hog	no	yes
B	num-calls	1	100
C	rate	0	1000
D	num-conns	10	1000

All combinations of the above levels were tested and the $2^4 3$ design with regression model was used for the experiment.

The variation due to B is 81.4% and variation due to errors is **6.46%**

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Case 2: Effect of number of replication on the variation of result

The same test mentioned in case 1 was run with 10 replications.

Percentage variation due to errors was **5.5%** .

As far as Test 1 and Test 2 are concerned the reduction in effect of errors was not justifiable enough to run 10 replications compared to the computation cost incurred by that many replications. Therefore, three replications were decided to be enough for the an experiment.

Case 3: This experiment was aimed at identifying a workload that gives stable outputs at all the replications of a simple test.

Variances of the response times with respect to each test are as in Table 6-3. When the rate and the number of connections are large the response times were varying enormously among the 10 replications. Number of concurrent connections can be suspected to be responsible for this huge variation. This is mainly due to the fact that the other parameters do not contribute to this variation in a major proportion.

Table 6-3: Observations for case 3

hog	Num-calls	Rate	Num-conn	variance of response time	Variance of Concurrent connections
yes	100	1000	10	0.03	0
yes	100	1000	1000	30.37	46.54
yes	100	0	10	0	0
yes	100	0	1000	0	0
yes	1	1000	10	0	0
yes	1	1000	1000	838.11	6494.1
yes	1	0	10	0	0
yes	1	0	1000	0	0
no	100	1000	10	0.01	0
no	100	1000	1000	7.79	4
no	100	0	10	0	0
no	100	0	1000	0	0
no	1	1000	10	0	0
no	1	1000	1000	693.33	10211.88
no	1	0	10	0	0
no	1	0	1000	0	0

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Case 4: Concurrency of connections Vs response time

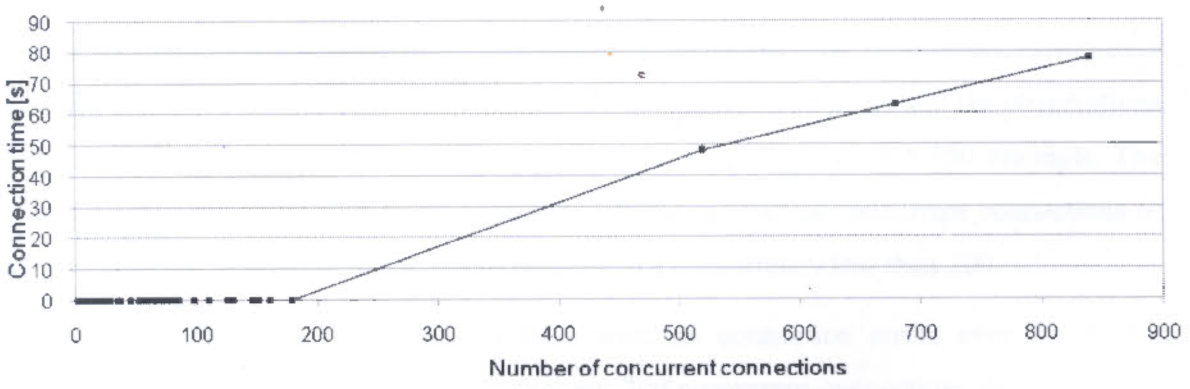
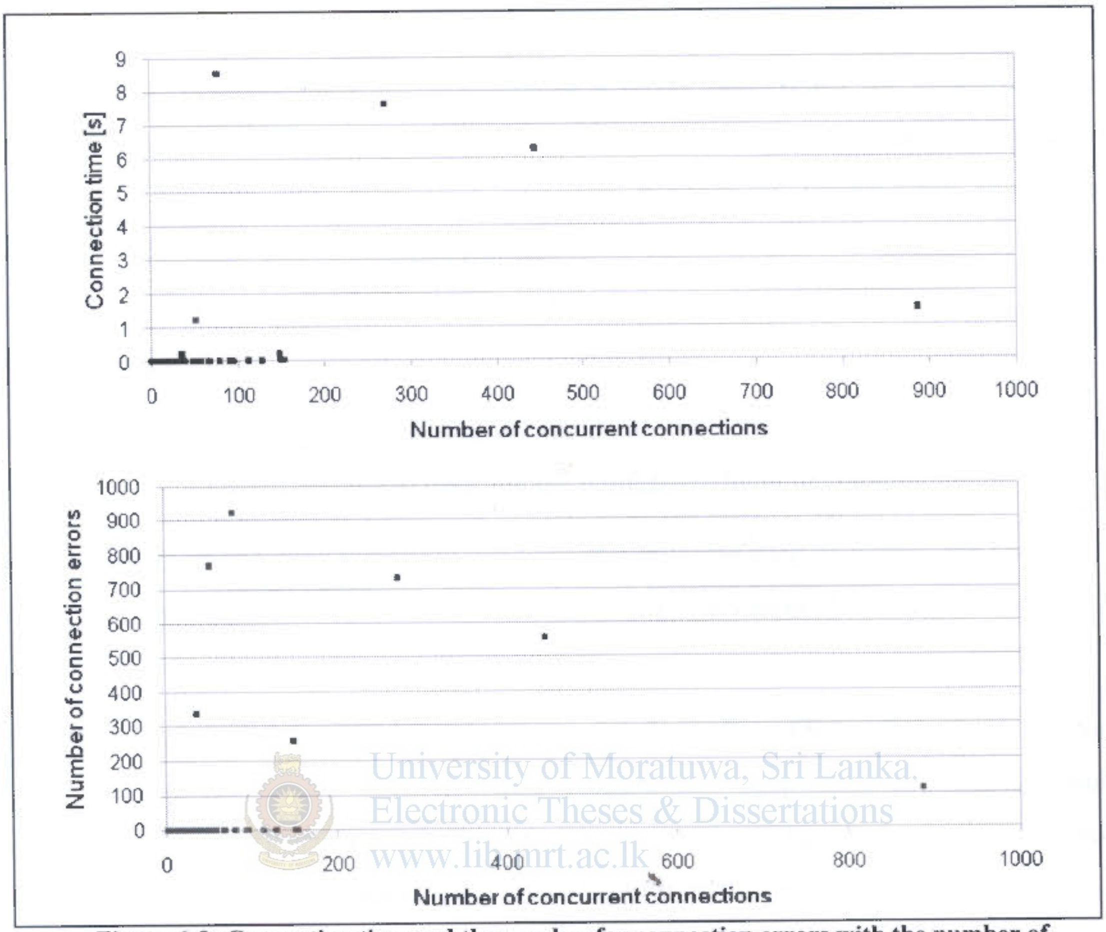


Figure 6-2: Variation of the connection time with the number of connections with hog option.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 6-3: Connection time and the number for connection errors with the number of concurrent connections when the *hog* option is disabled

An HTTP request was applied on the web server emulating 1000 simultaneous users over 1000 exclusive connections. This command was repeated 100 times and the observations showed that the number of concurrent connections varied in all the 100 attempts. The connection times are shown in Figure 6-2 with the number of concurrent connections of each attempt. Almost all the attempts resulted in a concurrency less than 200.

When the *hog* option was enabled there were no connection errors even for higher concurrencies such as 800 users. After about 200 concurrent connections the connection time keeps growing when the concurrency increases. This should be due to the buffering of connections when a large number of connection requests, is coming in for the server.

Without *hog* option, the connections ended up in errors even when the concurrency was less than 200.

The responses are reliable when *hog* option is used so that hereafter all tests are run with *hog* option enabled. The number of concurrent connections varies in a large range when a large number of connections are requested at a larger rate. When the concurrency of connections changes, the connection times also change within the replications of the same workload. Therefore, the connection rate and number of connections should be maintained at a number lower than 200 in this case.

6.3 Conclusions

According to the above discussion *hog* option should be kept ON and *three replications* are enough for each experiment to improve the confidence level of the readings.

Server configurations should be kept constant and only the type of workloads should be varied. Common workload parameters should be kept at values that incur less error as discussed above.

SNMP MIBs selected from the list in Table 6-1 are *ssSwapOut, ssSwapIn, ssIOSent, ssIOReceived, ssCpuUser, ssCpuSystem, ssCpuIdle, memTotalFree, memShared, memCached, memBuffer, memAvailReal* and *memAvailSwap* because they cover major resources such as CPU, Memory and IO utilization sufficiently for the scope of this research.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

7 Performance Tests and Results

7.1 *Case 1- Background Processes and Load*

Normal load was observed without starting the web server initially. The monitoring was started from the moment the SNMP daemon was started by login into the SUT from a remote laptop soon after booting the SUT. The monitored parameters are *ssSwapOut, ssSwapIn, ssIOSent, ssIOReceived, ssCpuUser, ssCpuSystem, ssCpuIdle, memTotalFree, memShared, memCached, memBuffer, memAvailReal* and *memAvailSwap*. The interpretations of these terms are given in Table 6-1. Parameters related to swapping, namely *ssSwapOut, ssSwapIn*, and *memAvailSwap* as well as *memShared* did not show any variation so that the graphs are not given here.

According to Figure 7-1 I/O, CPU, Interrupts and Contexts vary in a transient manner for a period of around 10 minutes.

After the time 09:50 the apache server was manually started and the change to the resources can only be observed in the graphs related to *Real Memory*, namely *memCached, memBuffered* and *memAvailReal*. Hence it can be concluded that Apache is a memory-intensive process. Swap space is not used in this situation. I/O and CPU usage are levelling to a constant value after the transient period, regardless of the event of starting the Apache server.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

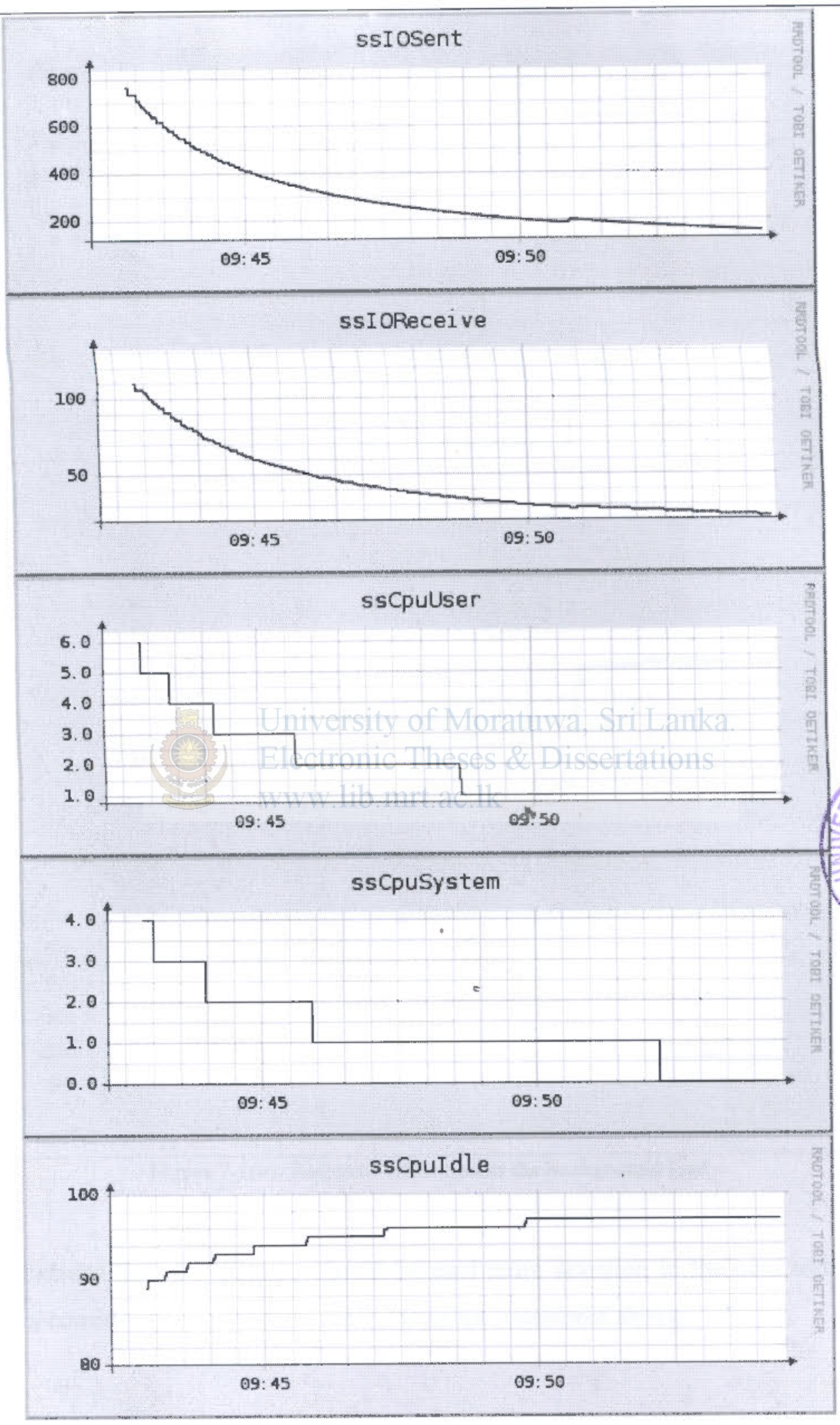


Figure 7-1(a): Resource utilization at the background load

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

UNIVERSITY OF MORATUWA
LIBRARY
SRI LANKA

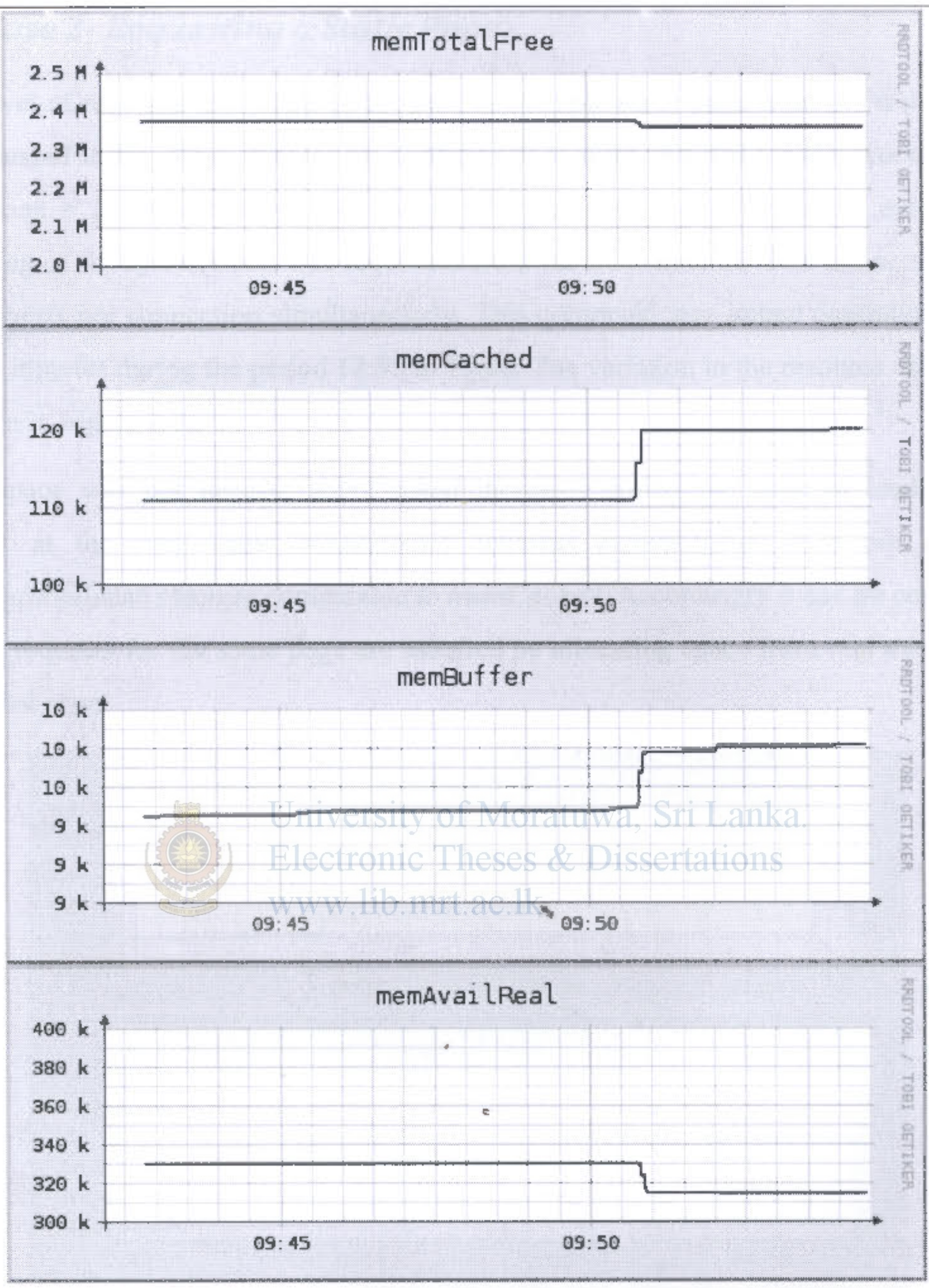


Figure 7-1(b): Resource utilization at the background load

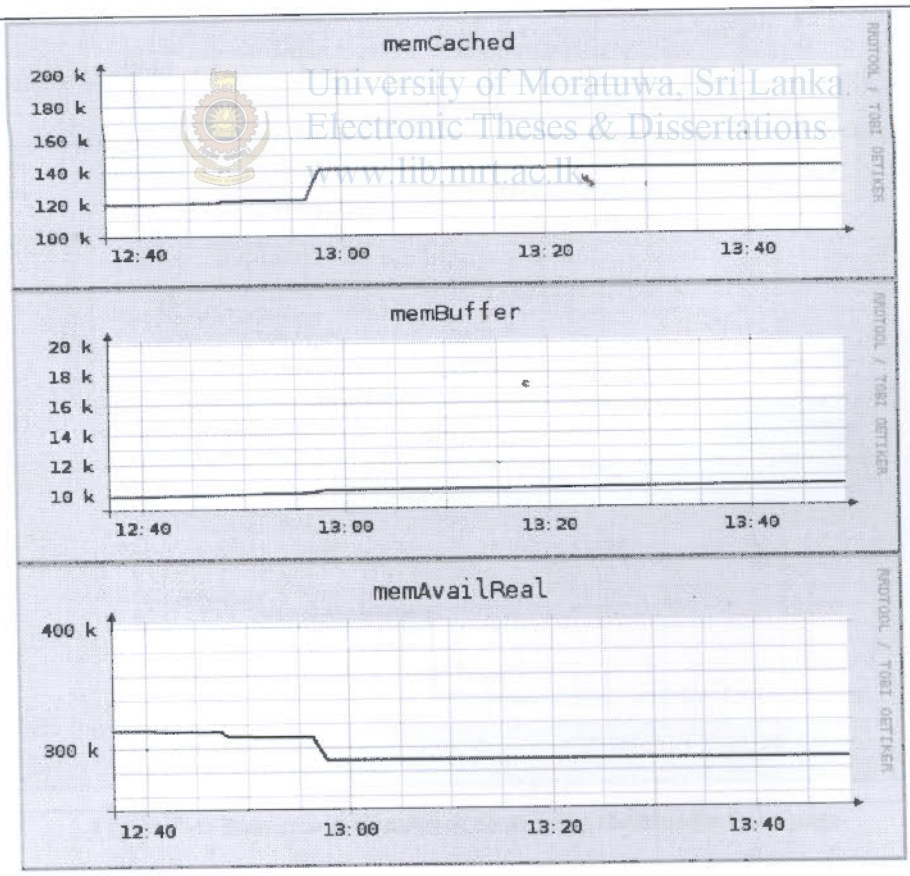
University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

It was decided that the memory should be paid more attention in the next tests unless there is a considerable variation in other resources considered above.

7.2 *Case 2- Requesting a Static Page*

Next level of readings was taken by requesting a simple static page from the web server. As discussed in the above chapter the configurations were selected for this workload as: num-conns = 10, num-calls = 100, rate = 1000 and hog = on. Hence, the index page consisting of 85 bytes of text was requested from the SUT over 10 TCP connections, as 100 requests per connection simultaneously. This command was issued continuously for about 2 minutes during the period 12:55 to 13:00. The variation in the resource utilization is shown in Figure 7-2.

Swap space was not needed. *memCached* increases during the time of loading and remains at the new value continuously whereas *memBuffered* does not change. *memAvailReal* also changes comparable to *memCached*. Accordingly it can be concluded that the requests for the same page are satisfied by allocating space from real memory to cache that page.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

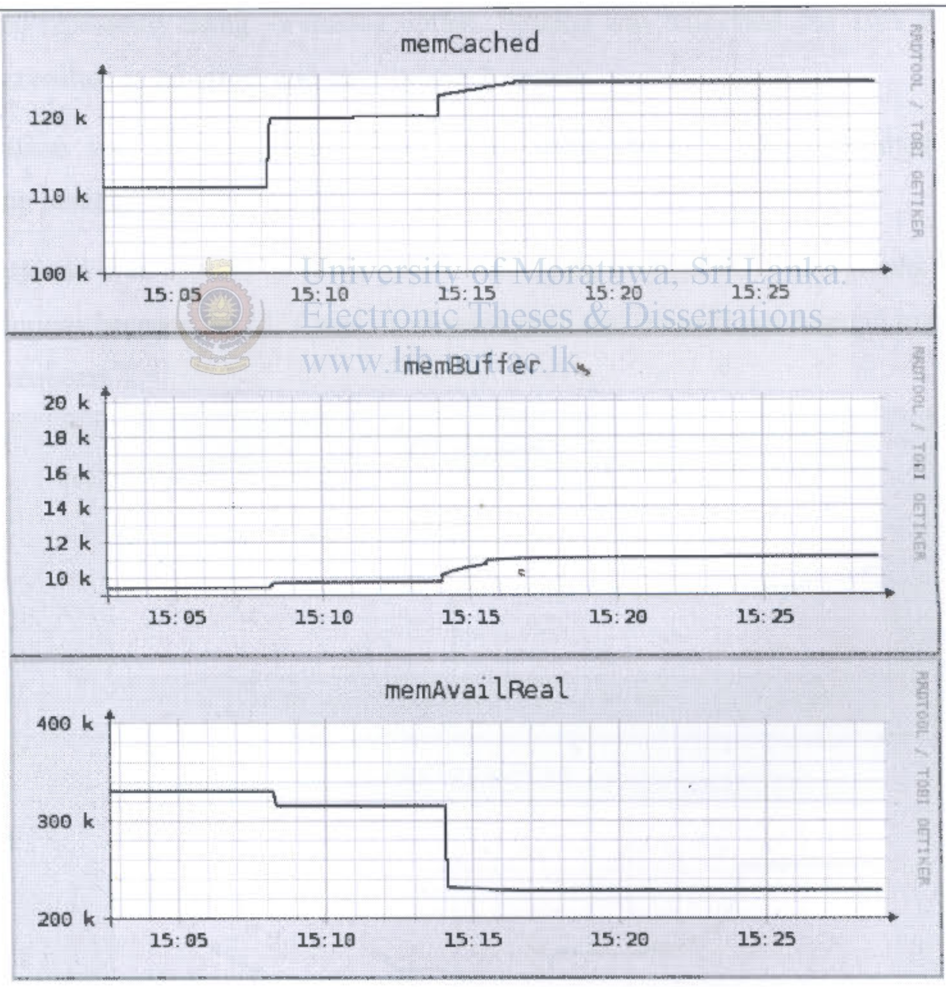
Figure 7-2: Static page requested 1000 times simultaneously

7.3 *Case 3- Requesting the Moodle Front Page*

The same workload configuration as in case 2 above was applied except for the particular page that was requested. This time the Moodle Front Page was requested. This resulted in the session given in Figure 7-3 as recorded by the workload recorder script.

```
/moodle18 method=GET think=21
    /moodle18/ method=GET
/moodle18/login/index.php?MoodleSession=aec6e553aa5b2bd3d0a551dc
6524f005 method=GET
```

Figure 7-3: Recorded session for requesting the Moodle front page



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 7-4: Resource utilization at requesting the Moodle front page

SUT was restarted before applying the load to reset the resource utilizations after the interactions between the remote laptop in session recording and other maintenance tasks. The time taken to reach a steady state after the reboot shall be called resetting time hereafter. However, it is needed to wait for the transient period before applying the load, which is a lesser time duration compared to the resetting time. Transient and resetting are not much visible in memAvailReal and this is the parameter in concern very often in the following cases.

The swap space and the shared memory were not in use in this case also, so that the relevant graphs are omitted in Figure 7-4. As shown in Figure 7-4 at time 15:08, the Apache server was started after the transient period. At 15:14 the front page was requested repeatedly using –wsesslog option. Session was requested one after the other and this resulted in no errors and gave proper response.

The session was then requested 1000 times simultaneously which resulted in the following results.

The workload was applied at about 15:45. According to Figure 7-5 the graphs became discontinuous because the SUT was overloaded such that it could not respond to even the SNMP requests.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

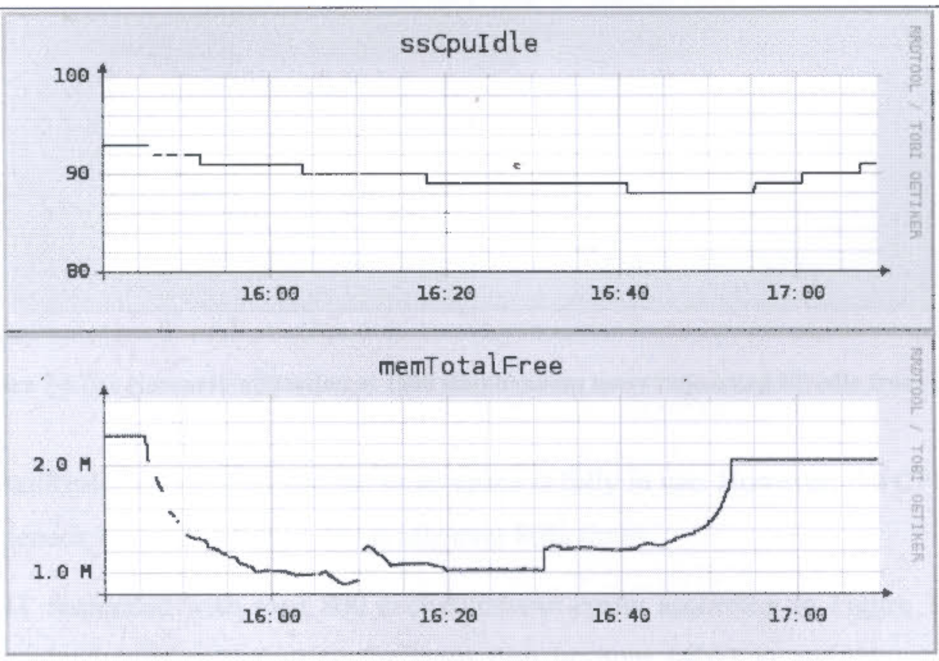
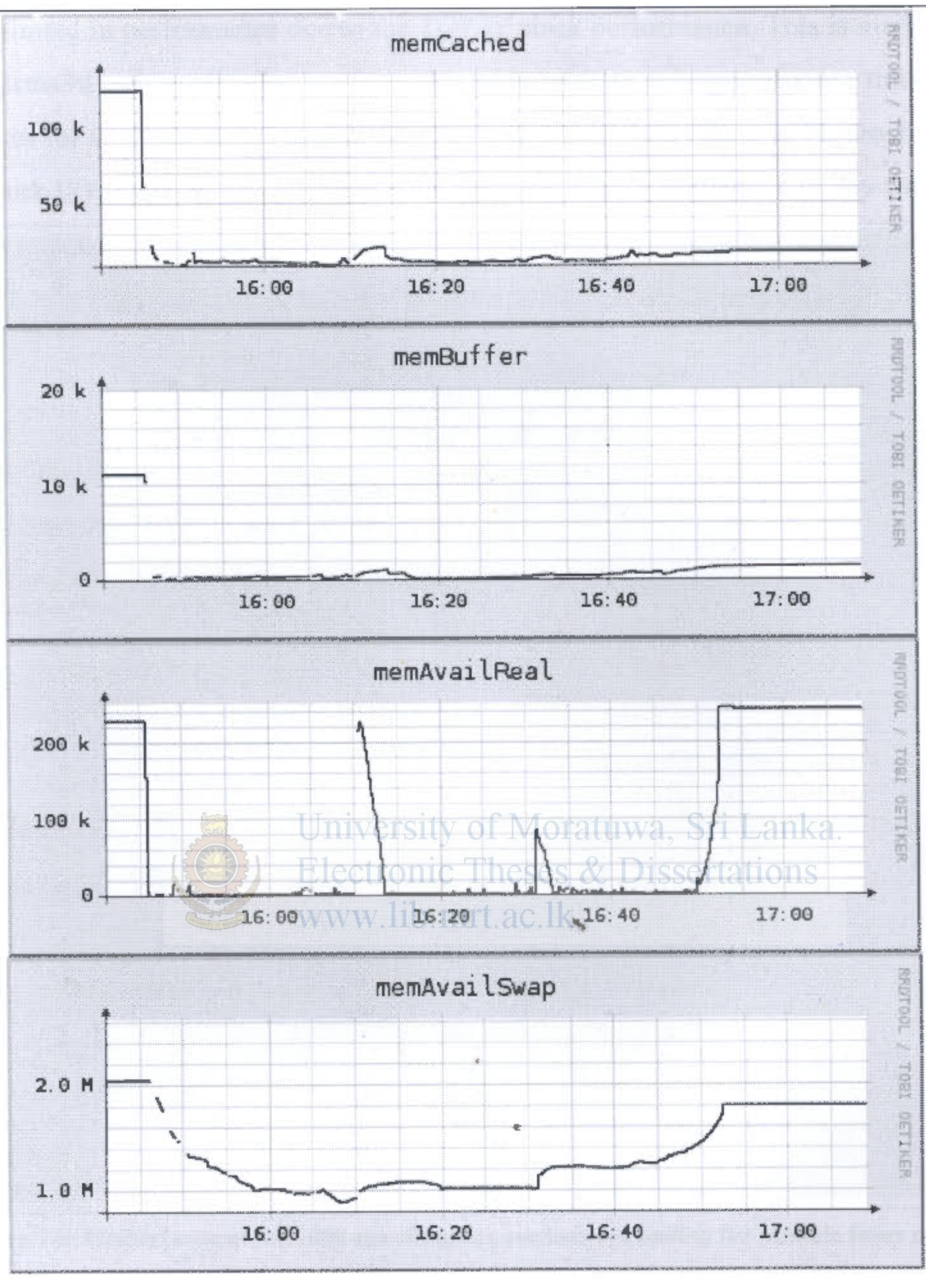


Figure 7-5 (a): Resource utilization at 1000 simultaneous users requesting Moodle front page



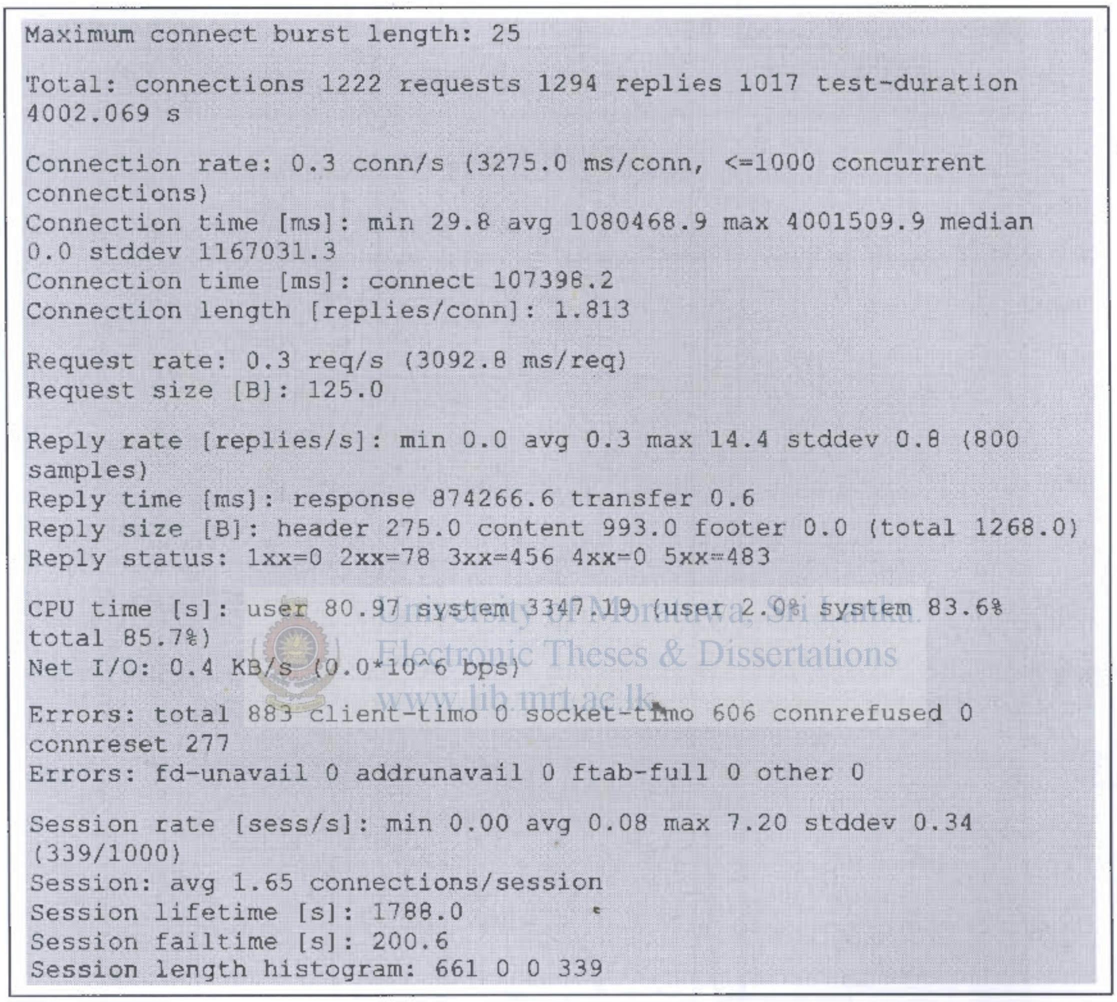
University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 7-5 (b): Resource utilization at 1000 simultaneous users requesting Moodle front page

memAvailReal drops to zero and the swap apace is fully in use. However, the CPU is not the bottleneck because still the CPU is idle over 80% times.

The SUT responded with over 800 socket-timeout-errors according to Figure 7-6. The total workload kept on executing for more than an hour (4002 s) and this should be because the SUT had to wait for socket timeouts. Therefore, it can be concluded that the

SUT is limited in performance due to the TCP/IP stack performance. This is visible to the user in terms of web application performance issue with an average response time of over 10 minutes for such a simple request. Another important observation in Figure 7-6 is that the network I/O is almost 0%. Therefore the network is not a bottleneck in any way.



```
Maximum connect burst length: 25

Total: connections 1222 requests 1294 replies 1017 test-duration
4002.069 s

Connection rate: 0.3 conn/s (3275.0 ms/conn, <=1000 concurrent
connections)
Connection time [ms]: min 29.8 avg 1080468.9 max 4001509.9 median
0.0 stddev 1167031.3
Connection time [ms]: connect 107398.2
Connection length [replies/conn]: 1.813

Request rate: 0.3 req/s (3092.8 ms/req)
Request size [B]: 125.0

Reply rate [replies/s]: min 0.0 avg 0.3 max 14.4 stddev 0.8 (800
samples)
Reply time [ms]: response 874266.6 transfer 0.6
Reply size [B]: header 275.0 content 993.0 footer 0.0 (total 1268.0)
Reply status: 1xx=0 2xx=78 3xx=456 4xx=0 5xx=483

CPU time [s]: user 80.97 system 3347.19 (user 2.0% system 83.6%
total 85.7%)
Net I/O: 0.4 KB/s (0.0*10^6 bps)

Errors: total 883 client-timo 0 socket-timo 606 connrefused 0
connreset 277
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

Session rate [sess/s]: min 0.00 avg 0.08 max 7.20 stddev 0.34
(339/1000)
Session: avg 1.65 connections/session
Session lifetime [s]: 1788.0
Session failtime [s]: 200.6
Session length histogram: 661 0 0 339
```

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 7-6: Httperf output for 1000 simultaneous sessions requesting for Moodle front page

The response time was measured by requesting the same session for 1 time, 10 times and 100 times as shown in Table 7-1 in addition to the afore mentioned experiment with 1000 sessions.

Table 7-1: Response time with the number of concurrent sessions

Number of times	1	10	100	100 (deterministic arrival)	1000
Response time (ms)	1291.3	2016.5	30505.0	23470.3	874266.6 (with errors)

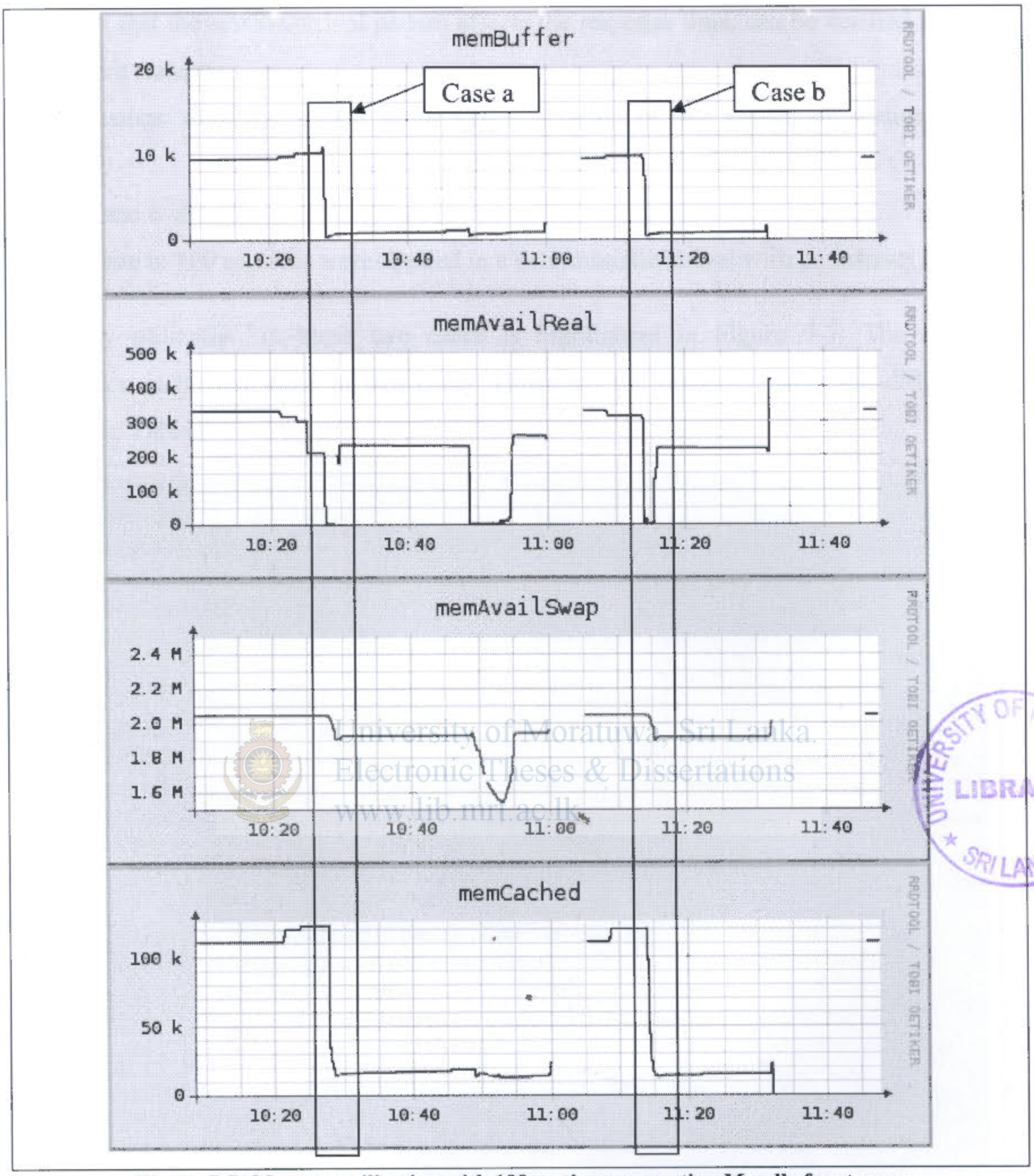


Figure 7-7: Memory utilization with 100 sessions requesting Moodle front page

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

UNIVERSITY OF MORATUWA
LIBRARY
SRI LANKA

It can be noted that when the number of concurrent sessions increases all the users experience poor response time. This can be solved by having a control over the session arrival pattern.

The fact that the session arrival pattern affects the response time, can be verified from the following test case:

100 sessions were run within about 120 s duration in two modes of session arrival namely,

Case a: All 100 sessions were applied once

Case b: 100 sessions were applied in a deterministic arrival with an interval of 0.1 s.

Memory utilization in these two cases is highlighted in Figure 7-7. The resource utilization is almost same in two cases. However, the response times are different as shown in Table 7-1.

```
/moodle18 method=GET think=25
    /moodle18/ method=GET
/moodle18/login/index.php?MoodleSession=3e74b89dcdc0aa64bdefbc80613d
9876 method=GET
/moodle18/login/index.php method=POST
contents="username=wingperf&password=wingperf&testcookies=1"
think=11
/moodle18/ method=GET think=4
/moodle18/theme/formal_white/logo.jpg method=GET think=3
    /moodle18/pix/spacer.gif method=GET
    /moodle18/pix/i/course.gif method=GET
    /moodle18/pix/t/switch_minus.gif method=GET
    /moodle18/mod/forum/icon.gif method=GET
    /moodle18/calendar/overlib.cfg.php method=GET
/moodle18/course/view.php?id=42 method=GET think=3
/moodle18/pix/i/users.gif method=GET think=2
    /moodle18/pix/help.gif method=GET
    /moodle18/mod/assignment/icon.gif method=GET
    /moodle18/mod/choice/icon.gif method=GET
    /moodle18/mod/data/icon.gif method=GET
    /moodle18/mod/resource/icon.gif method=GET
    /moodle18/mod/wiki/icon.gif method=GET
    /moodle18/pix/i/grades.gif method=GET
    /moodle18/pix/f/pdf.gif method=GET
    /moodle18/pix/i/one.gif method=GET
    /moodle18/pix/f/excel.gif method=GET
    /moodle18/pix/f/web.gif method=GET
    /moodle18/pix/f/zip.gif method=GET
    /moodle18/pix/f/image.gif method=GET
    /moodle18/pix/f/word.gif method=GET
```

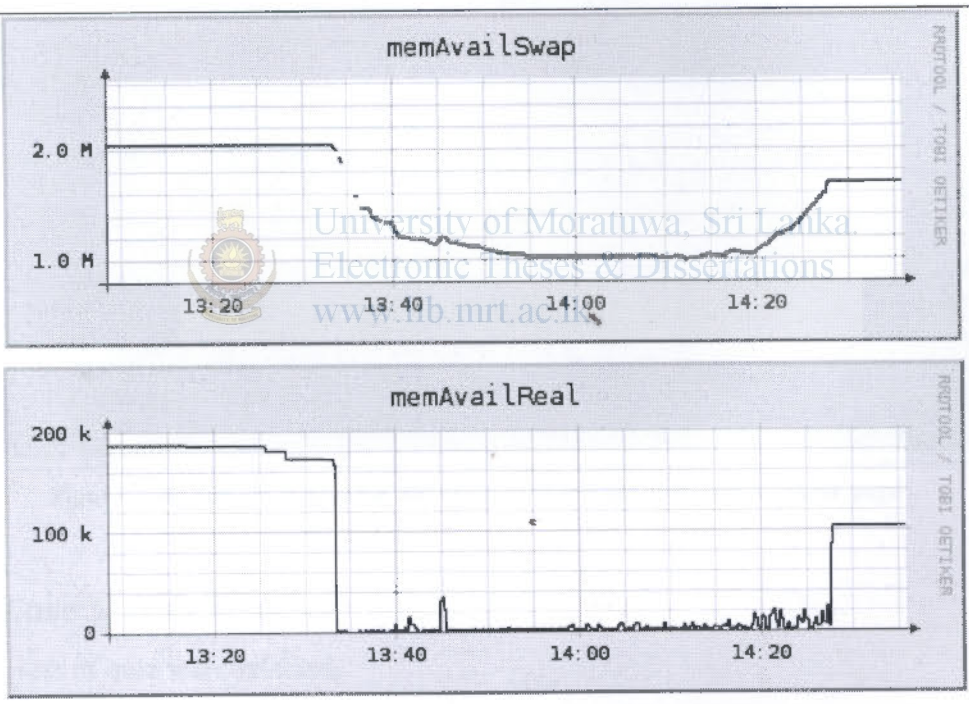
University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 7-8: Session to login and open the course page

7.4 *Case 4- Session to Login and Open the Course*

A session that login and opens a course page is as shown in Figure 7-8.

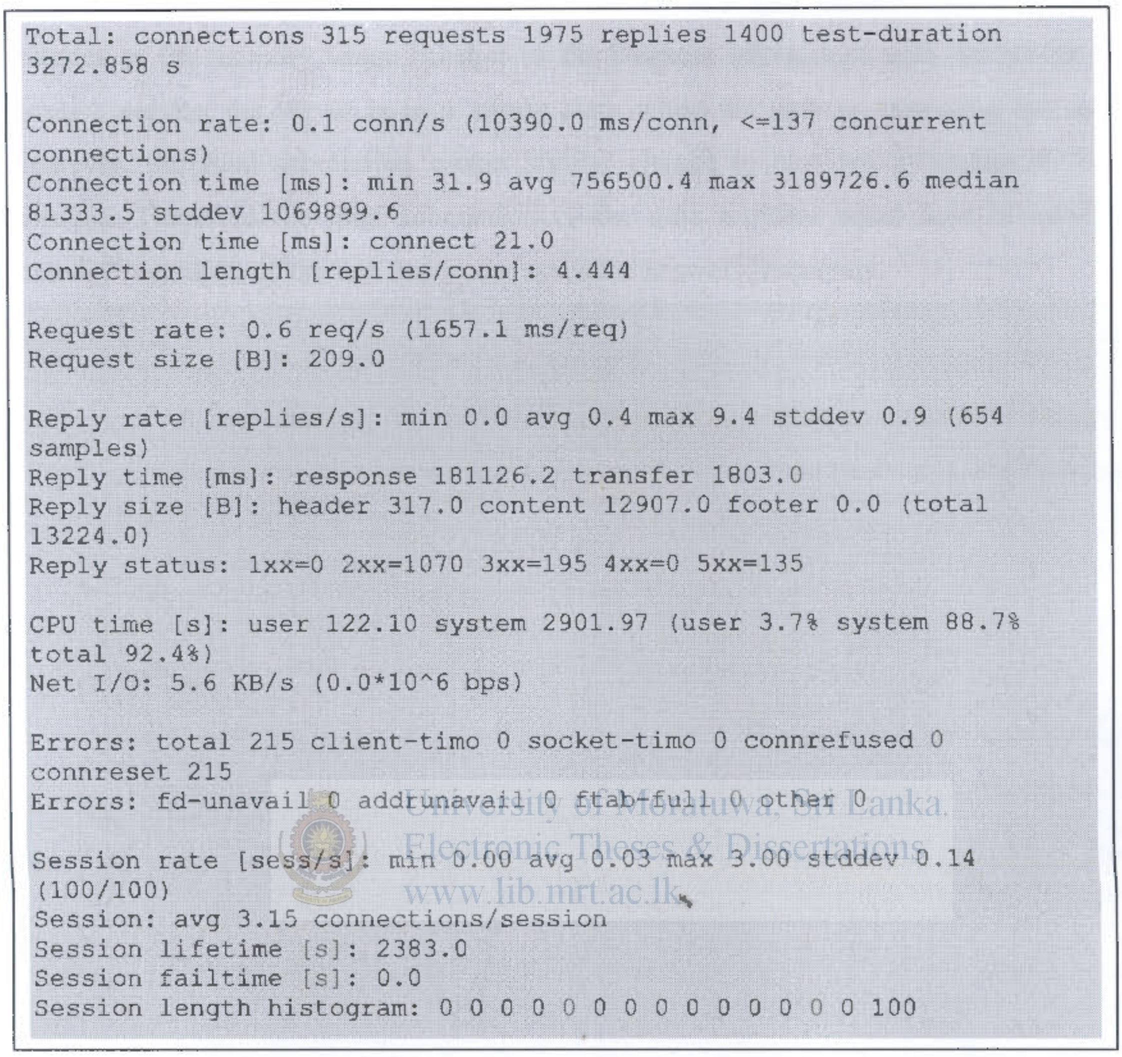
100 numbers of this session were applied simultaneously and it resulted in the memory utilization shown in Figure 7-9. The duration of a single session is about 35 seconds when all the think times in Figure 7-8 are added. However, a session has taken about 180 s on average to complete, according to the *HTTPerf* output given in Figure 7-10. The responses for each request will also take some time to arrive at the client, but that should not cause this much of a difference between session duration. There are about 200 connection reset errors that shows the response is not so healthy.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 7-9: Memory ntilization at 100 sessions that login and view a course page

Memory utilization in Figure 7-9 shows that the available memory has dropped to almost zero as well as the swap space is used largely. Therefore the memory bottleneck in the SUT can be suspected to cause a poor response time in this case also.



```
Total: connections 315 requests 1975 replies 1400 test-duration
3272.858 s

Connection rate: 0.1 conn/s (10390.0 ms/conn, <=137 concurrent
connections)
Connection time [ms]: min 31.9 avg 756500.4 max 3189726.6 median
81333.5 stddev 1069899.6
Connection time [ms]: connect 21.0
Connection length [replies/conn]: 4.444

Request rate: 0.6 req/s (1657.1 ms/req)
Request size [B]: 209.0

Reply rate [replies/s]: min 0.0 avg 0.4 max 9.4 stddev 0.9 (654
samples)
Reply time [ms]: response 181126.2 transfer 1803.0
Reply size [B]: header 317.0 content 12907.0 footer 0.0 (total
13224.0)
Reply status: 1xx=0 2xx=1070 3xx=195 4xx=0 5xx=135

CPU time [s]: user 122.10 system 2901.97 (user 3.7% system 88.7%
total 92.4%)
Net I/O: 5.6 KB/s (0.0*10^6 bps)

Errors: total 215 client-timo 0 socket-timo 0 connrefused 0
connreset 215
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

Session rate [sess/s]: min 0.00 avg 0.03 max 3.00 stddev 0.14
(100/100)
Session: avg 3.15 connections/session
Session lifetime [s]: 2383.0
Session failtime [s]: 0.0
Session length histogram: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 100
```

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 7-10: httperf output for 100 sessions that login and open a course page

7.5 *Case 5- Quiz Session*

Two *level*s of quiz were selected.

Level-1: answer all 10 questions, and submit all and finish at once

Level-2: answer 1 or 2 questions at a time and save, and finally submit all and finish.

As shown in Figure 7-11 the session becomes longer in terms of the number of files when the user frequently saves the answers. This causes more interaction with the Moodle server during the quiz.

Memory utilization in Figure 7-12 shows that Level-2 quiz session causes frequent changes in the memory usage because of the frequent interactions with the server. In Level-1 session the server is in a steady state while the user is answering the quiz. However, the final submission causes similar change in memory utilization in both sessions. Therefore the final submission of the quiz acquires equal hold of server's memory regardless of the fact that a user saves the answers frequently.

Level 1	Level 2
/moodle18 method=GET think=21	/moodle18 method=GET think=21
/moodle18/ method=GET	/moodle18/ method=GET
/moodle18/login/index.php?MoodleSes	/moodle18/login/index.php?MoodleSes
sion=x method=GET	sion=y method=GET
/moodle18/theme/standard/styles.php	/moodle18/theme/standard/styles.php
?MoodleSession=x method=GET	?MoodleSession=y method=GET
/moodle18/theme/formal_white/styles	/moodle18/lib/javascript-
.php?MoodleSession=x method=GET	static.js?MoodleSession=y
/moodle18/lib/javascript-	method=GET
static.js?MoodleSession=x	/moodle18/theme/formal_white/styles
method=GET	.php?MoodleSession=y method=GET
/moodle18/lib/javascript-	/moodle18/lib/javascript-
mod.php?MoodleSession=x method=GET	mod.php?MoodleSession=y method=GET
/moodle18/lib/overlib.js?MoodleSess	/moodle18/lib/overlib.js?MoodleSess
ion=x method=GET	ion=y method=GET
/moodle18/lib/cookies.js?MoodleSess	/moodle18/lib/cookies.js?MoodleSess
ion=x method=GET	ion=y method=GET
/moodle18/lib/ufo.js?MoodleSession=	/moodle18/lib/ufo.js?MoodleSession=
x method=GET	y method=GET
/moodle18/login/index.php	/moodle18/login/index.php
method=POST	method=POST
contents="MoodleSession=x&username=	contents="MoodleSession=y&username=
wingperf&password=wingperf&testcook	wingperf&password=wingperf&testcook
ies=1" think=5	ies=1" think=6
/moodle18/?MoodleSession=x	/moodle18/?MoodleSession=y
method=GET think=4	method=GET think=3
/moodle18/calendar/overlib.cfg.php?	/moodle18/calendar/overlib.cfg.php?
MoodleSession=x method=GET	MoodleSession=y method=GET
/moodle18/course/view.php?id=45&Moo	/moodle18/course/view.php?id=45&Moo
dleSession=x method=GET think=2	dleSession=y method=GET think=2
/moodle18/mod/quiz/view.php?id=366&	/moodle18/mod/quiz/view.php?id=366&
MoodleSession=x method=GET think=4	MoodleSession=y method=GET think=5
/moodle18/mod/quiz/attempt.php?id=3	/moodle18/mod/quiz/attempt.php?id=3
66&MoodleSession=x method=GET	66&MoodleSession=y method=GET
think=5	think=5
/moodle18/mod/quiz/attempt.php?id=3	/moodle18/mod/quiz/attempt.php?id=3
66 method=POST	66 method=POST
contents="MoodleSession=x&quizpassw	contents="MoodleSession=y&quizpassw
ord=vba123" think=5	ord=vba123" think=5
/moodle18/mod/quiz/timer.js?MoodleS	/moodle18/mod/quiz/timer.js?MoodleS
ession=x method=GET	ession=y method=GET
/moodle18/mod/quiz/attempt.php	/moodle18/mod/quiz/attempt.php
method=POST	method=POST
contents="MoodleSession=x&q=101&que	contents="MoodleSession=y&q=101&que
stionids=2533%2C2534%2C2527%2C2526%	stionids=2530%2C2535%2C2527%2C2528%

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

```
2C2528%2C252$
/moodle18/mod/quiz/review.php?attem
pt=6044&MoodleSession=x method=GET
think=3
/moodle18/mod/quiz/view.php?q=101&M
oodleSession=x method=GET
```

```
2C2529%2C253$
/moodle18/mod/quiz/attempt.php
method=POST
contents="MoodleSession=y&q=101&que
stionids=2530%2C2535%2C2527%2C2528%
2C2529%2C253$
/moodle18/mod/quiz/attempt.php
method=POST
contents="MoodleSession=y&q=101&que
stionids=2530%2C2535%2C2527%2C2528%
2C2529%2C253$
/moodle18/mod/quiz/attempt.php
method=POST
contents="MoodleSession=y&q=101&que
stionids=2530%2C2535%2C2527%2C2528%
2C2529%2C253$
/moodle18/mod/quiz/attempt.php
method=POST
contents="MoodleSession=y&q=101&que
stionids=2530%2C2535%2C2527%2C2528%
2C2529%2C253$
/moodle18/mod/quiz/attempt.php
method=POST
contents="MoodleSession=y&q=101&que
stionids=2530%2C2535%2C2527%2C2528%
2C2529%2C253$
/moodle18/mod/quiz/attempt.php
method=POST
contents="MoodleSession=y&q=101&que
stionids=2530%2C2535%2C2527%2C2528%
2C2529%2C253$
/moodle18/mod/quiz/review.php?attem
pt=6045&MoodleSession=y method=GET
think=3
/moodle18/mod/quiz/view.php?q=101&M
oodleSession=y method=GET
/moodle18/mod/quiz/review.php?attem
pt=6045&MoodleSession=y method=GET
/moodle18/mod/quiz/view.php?q=101&M
oodleSession=y method=GET
```

University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 7-11: Two levels of quiz sessions

UNIVERSITY OF MORATUWA
LIBRARY
SRI LANKA

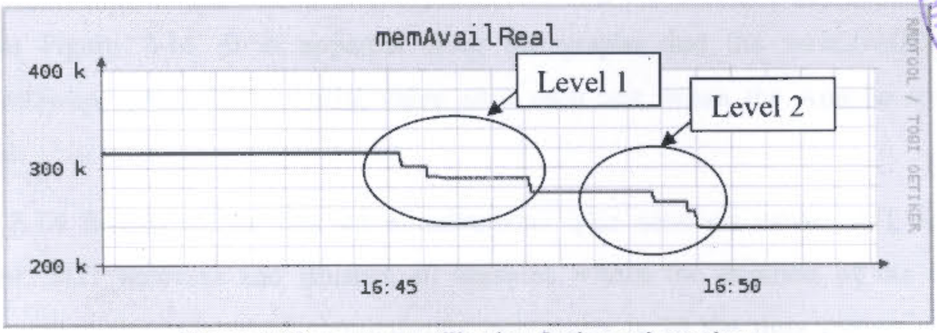
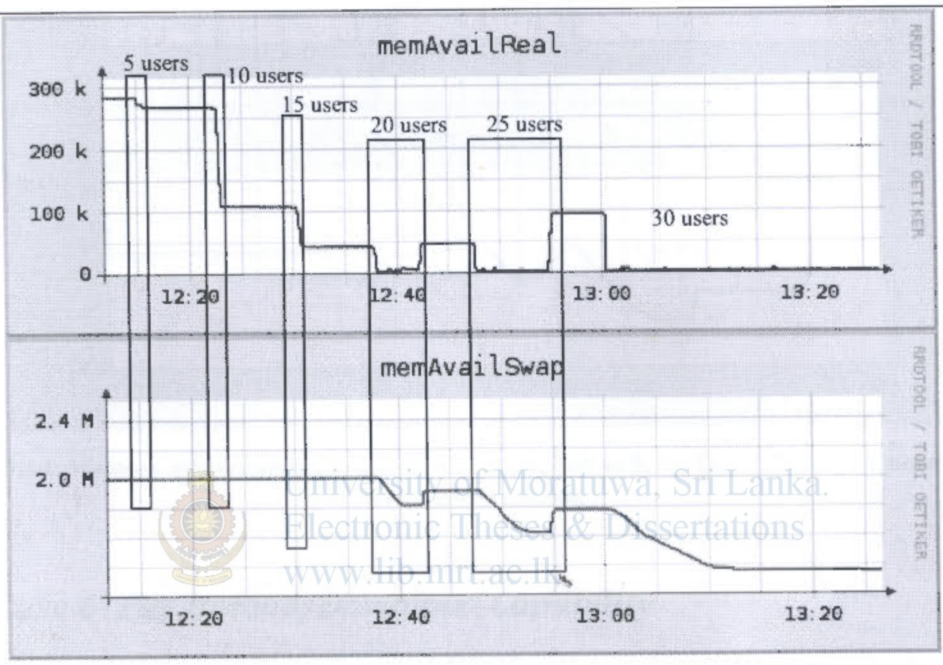


Figure 7-12: memory utilization during quiz sessions

Level-2 quiz session was run as 5, 10, 15, 20, 25, and 30 simultaneous users interleaved by 5 minute intervals without resetting the SUT by any means.

According to Figure 7-13, the *realMemAvail* does not return to initial value after each test. Though the tests are supposed to be independent, the memory utilization of previous test has affected the consecutive tests. Therefore, when the number of users is 20 the SUT starts swapping.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 7-13: Memory utilization at quiz session run as 5, 10, 15, 20, 25, and 30 simultaneous users

memAvailReal could be reset only by restarting the Apache service without rebooting the SUT. That way the resetting time could be reduced to a fraction of a second. Then the same series of tests was run with the action of resetting in between tests. The results are given in Figure 7-14. It is apparent from the graphs that the *memAvailReal* and *memAvailSwap* return to the initial value after each test, when the Apache service is restarted.

Figure 7-14 further shows that 25 simultaneous quiz sessions cause SUT to swap. However, SUT recovers and finishes all sessions within the duration of the quiz. In contrast, when the number of simultaneous quiz users is 30 the quiz cannot complete within the allocated time. Although the SUT responds and technically it is operating, user satisfaction and the desired user-task are not fulfilled. Therefore, this SUT cannot handle over 30 simultaneous quiz users from the users' point of view.

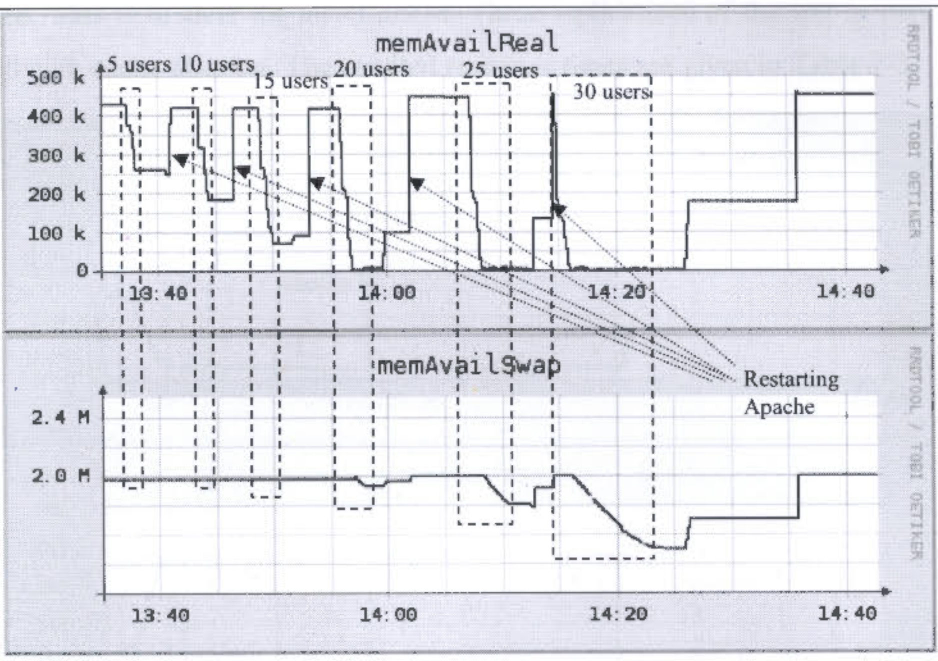


Figure 7-14: Memory utilization when quiz session is applied as 5, 10, 15, 20, 25 and 30 simultaneous users with resetting

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

7.6 *Case 6- File Upload/Download Capability*

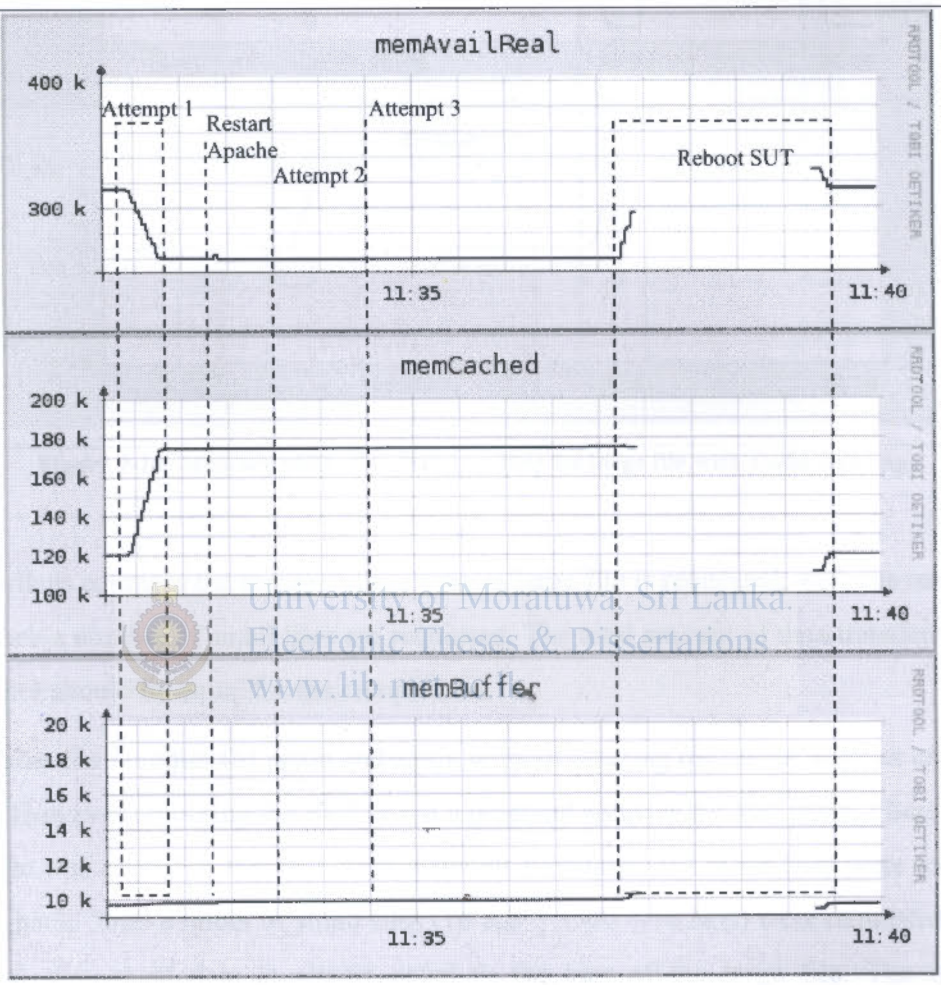
The *large file* was requested outside Moodle, (directly from web root) three times and the response times are given below:

Table 7-2: Response times for requesting the large file

Attempt	1	2	3
Response time	20.014 s	4.874 s	4.941 s

Response was over a single connection and as one request per connection.

According to Figure 7-15, *memAvaiReal* decreases only at the first attempt of requesting the file. *memCached* increases by an amount similar to this reduction. Next attempts, namely attempt2 and attempt3 must have been satisfied from the cache. That should be why the response times in attempt 2 and 3 are equal to each other and much less than that in attempt 1 as shown in Table 7-2. Cache does not reset only by restarting the Apache service. It resets by rebooting the SUT as shown in Figure 7-15. Hence, such page requesting experiments should be followed by reboot of the SUT when these are replicated. This is to clear the *memCached*. Three replications of the test in case 6 were executed with cache clearing. The resulted response times are given in Table 7-3.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 7-15: Memory utilization when the large file was requested without resetting the cache

Table 7-3: Response times for requesting the large file with cache clearing

Attempt	1	2	3
Response time	19.944 s	20.219 s	19.947 s
Net I/O	2736.0 KB/s (22.4*10^6 bps)	2698.8 KB/s (22.1*10^6 bps)	2735.6 KB/s (22.4*10^6 bps)

Response was over a single connection and as one request per connection.

The memory utilization shows similar variation in all the three attempts in Figure 7-16.

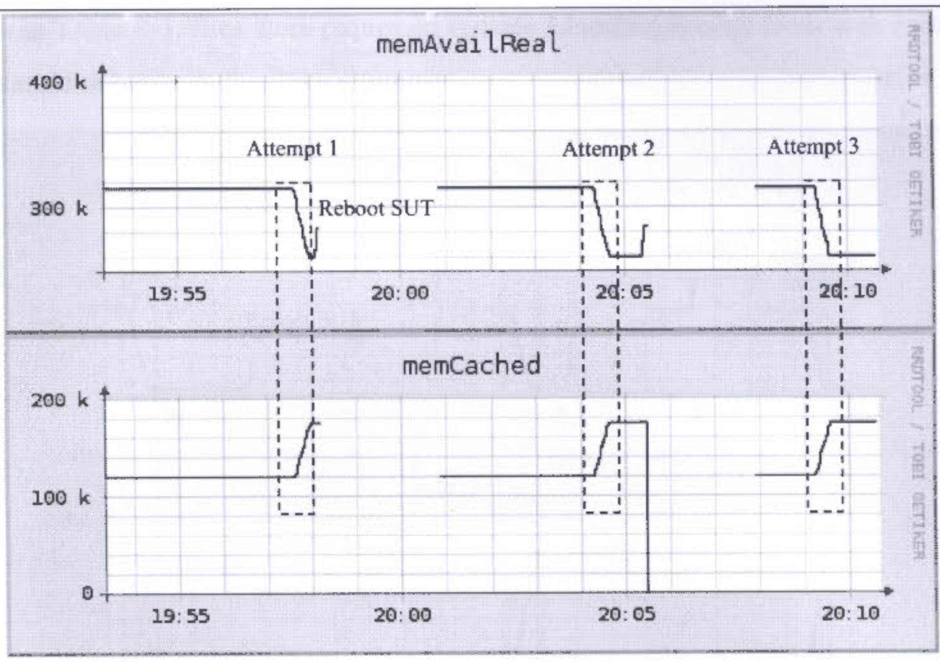


Figure 7-16: Memory utilization in requesting the large file with cache clearing

It is worth to compare the response time when a large file is requested, with the response time when a number of small files are requested. The total amount of data transferred in both cases should be equal for the comparison.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

If the same file is requested again and again without clearing the cache, it gives erroneous results. However rebooting the SUT is not a practical solution for this issue. Therefore the small file was copied at the SUT with different filenames and those files were requested back to back. 3500 number of small files (16 KB * 3500 = 56 MB) were requested so that the total amount of data is almost equal to the size of the large file. The test was replicated three times. The response times are given in Table 7-4.

Table 7-4: Response time when 3500 small files were requested

Attempt	1	2	3
Response time	30.628 s	31.788 s	30.883 s
Net I/O	1805.3 KB/s (14.8*10^6 bps)	1739.5 KB/s (14.2*10^6 bps)	1790.4 KB/s (14.7*10^6 bps)

Same as in Table 7-3, files were requested outside Moodle (directly from web root). SUT was restarted in between the three attempts.

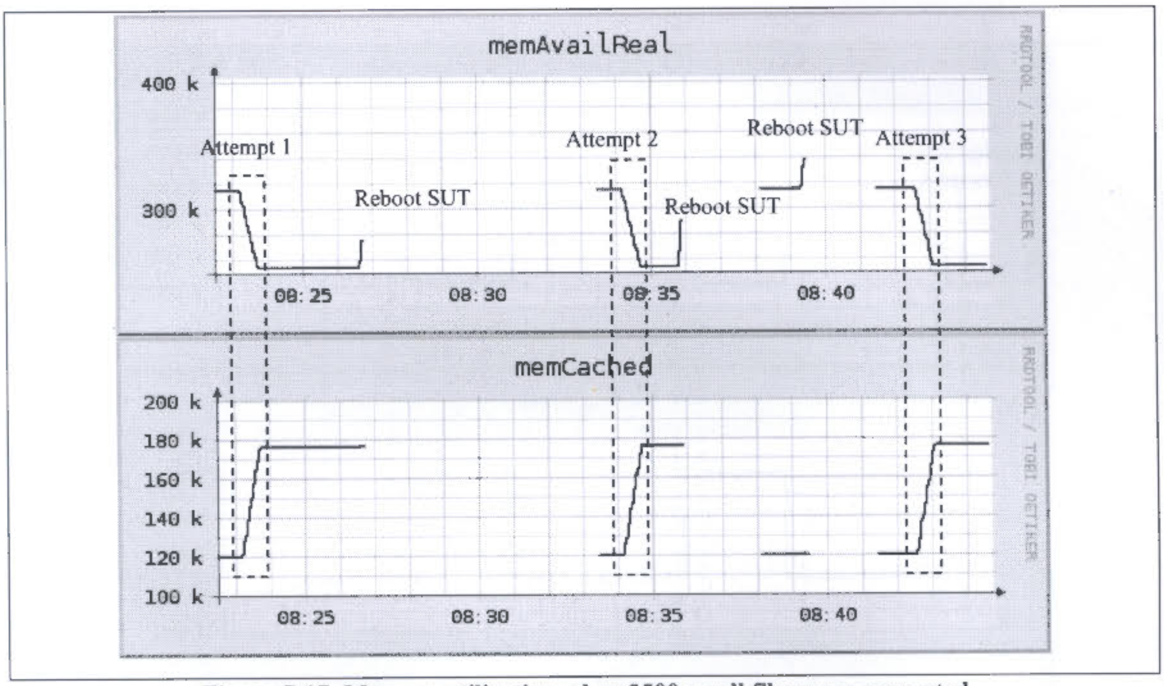


Figure 7-17: Memory utilization when 3500 small files were requested

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The large file was uploaded through Moodle three times. The file upload session consists of the sequence of steps as in Figure 7-18.

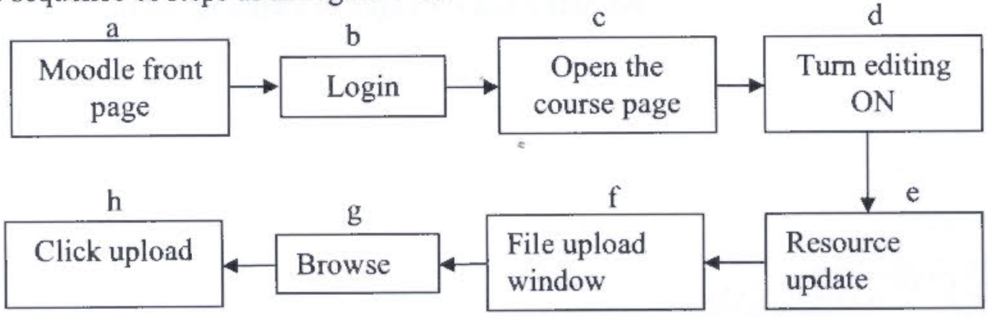


Figure 7-18: Sequence of interactions of a file uploading session

Steps (a) through (f) are performed only once. Steps (g) and (h) were repeated three times. In between these three times the file was deleted at the SUT.

According to Figure 7-19, it has taken about 30 seconds for uploading the *large file* to Moodle. This duration is closer to the time taken to download the same file outside the Moodle. Therefore the Moodle application does not incur considerable overhead in handling file transfers. The transfer time is determined by the action of processing the file for transmission at the two ends.

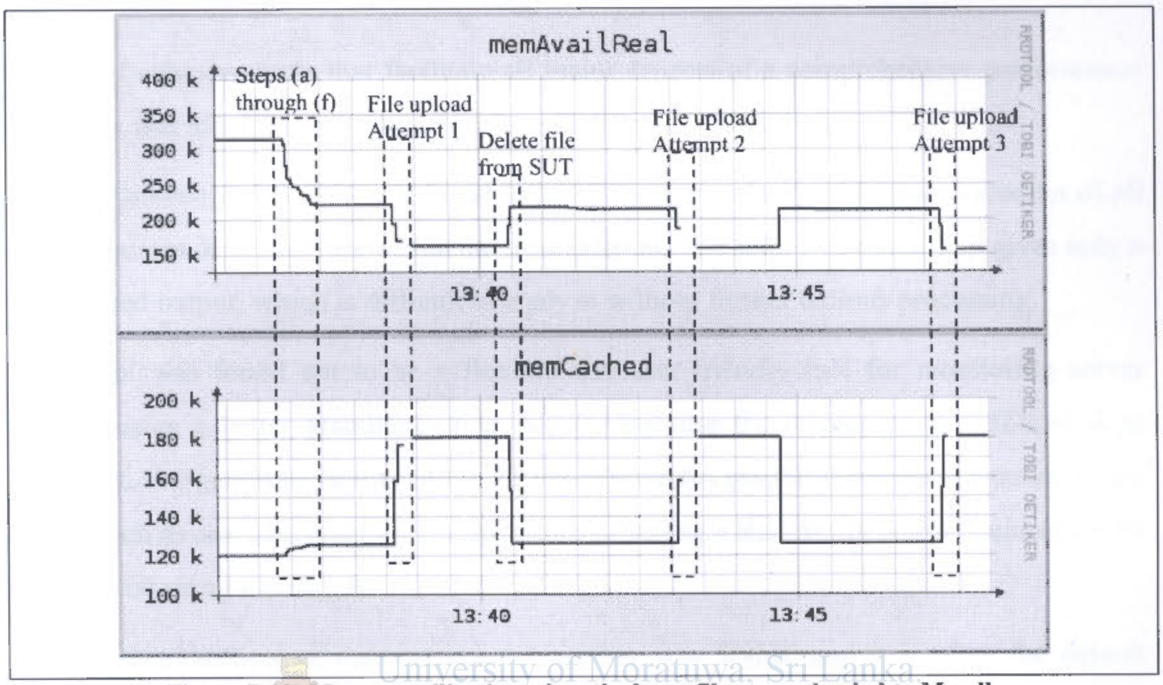


Figure 7-19: Memory utilization when the large file was uploaded to Moodle

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The network bandwidth is not a bottleneck because the network I/O is less than 25% of the total bandwidth, as shown in Table 7-3 and Table 7-4.

UNIVERSITY OF MORATUWA
LIBRARY
SRI LANKA

8 Analysis and Discussion

8.1 *Analysis of the Methodology*

A set of cohesive tools that facilitate all major aspects of a comprehensive performance analysis was identified in order to achieve the research objectives.

The tools such as MRTG were found to he of poor granularity to capture variation of all the resources of the servers within the time intervals less than 5 minutes. *Top* gives only a text based output, which is difficult to analyze without further tedious processing.

RRDtool was found out to be a flexible and user-friendly tool for monitoring server resources in a better granularity than MRTG because the resolution of RRDtool is in seconds. The graphing facility of the RRDtool provides graphical output in contrast to the tools such as *top*. The graphs were updated by running a shell script and it could easily be automated using a *cron* job.

Underlying protocol of the resource monitoring was SNMP and it is often the default choice in available resource monitoring tools including commercial tools. SNMP is a powerful protocol and it provides wide variety of information of the systems even at process level. SNMP utilities such as *net-snmp* provide easy and quick ways to poll the server and obtain the information related to server resources.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Performance testing of the system before deployment has not received sufficient attention. Therefore, synthetic workload generation was not in concern. However, this research introduced HTTPerf and several Perl modules available freely on the web, as a very appropriate solution for this purpose. The biggest strength of this combination of tools is the ahility to record real interaction between the clients and a server, and replay the interactions. Therefore the real environment is mimicked in the synthetic workload.

8.2 Analysis of the Test Results

The findings from the test results discussed in Chapter 7 can be summarized as follows:

For all the types of workloads such as Quiz sessions, Page views and especially when these workloads are simultaneously applied on the SUT, memory became the bottleneck. CPU and I/O were never a bottleneck.

It was the memory utilization that showed direct correlation to the clients' interaction. Every communication between the client and the SUT was visible in the memory graphs.

The SUT never became unavailable even when it was hosting 1000 concurrent users as in case 3 in Chapter 7. But the response time was in minutes (10 min) and that was an unacceptable level of response. Unless the response time from the point of view of the user was considered, this situation would not have been captured as a performance problem. It was only the swap utilization that showed a possible performance issue. However, as in case 4 (Figure 7-14), the swapping does not always imply an unacceptable response time at the client side. Therefore it is quite important to measure the response time at the client's end especially when multiple users are active on the server. This should be performed at least before deploying these types of systems by simulating users as in the process given in this study.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Other than the memory bottleneck, it was obvious from the case 3, when 1000 concurrent users were active the TCP/IP stack of the server was also a bottleneck. Various timeouts in TCP connections causes long delays when the requested number of connections could not be handled by the server. This delay is experienced by the user as a performance issue in the web application.

The network bandwidth provided between the client and the SUT was 100 Mbps and the network utilization never reached this amount.

9 Conclusion

9.1 Research Outcomes

The objectives of the research as mentioned in Chapter 1 were to arrive at conclusions about the performance of a web based system. Three research questions made the basis for the research.

The first question was what the proper approach was to reach the aforementioned objective. The systematic approach of a performance evaluation includes these major steps: workload characterization, system identification, workload generation, experimental design, resource and response monitoring, and data representation and analysis. It was essential to monitor the responses of both the server and the client ends, and correlate the user activities with the variations of the server resources.

The second research question was what tools and techniques are appropriate for the above approach. Workload characterization can be achieved by questionnaires given to the users directly and by log analysis. Recording real user interactions gives the best synthetic workloads. These recorded workloads had to be enriched with identified workload characteristics. There are free open source tools available for log analysis and workload recording.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The third research question was what aspects affected the performance of the system under study. Memory is the most critical resource for the LAMP based LearnOrg-Moodle in its current configuration in the test bed. It was observed that swapping at the server could warn about a possible performance issue, but that had to be verified by observing the response time at a client's end. Network or the CPU could not be regarded as performance limitation factors.

9.2 *Recommendations and Further Research*

A service level agreement should be implemented between the system administrators and the service receivers whether the system is small scale or large scale. It does not require expensive commercial tools to benchmark the systems and to give an estimation of the capability of the system. The open source tools freely available in the web can be combined to carry out a systematic performance evaluation. Although there are systematic approaches for a performance evaluation they are often limited to theory. However these procedures should be applied in real for the smooth functionality of a system.

Resource monitoring is a very critical stage of a performance evaluation task. Therefore, proper tools should be selected so that the resources of the server can be monitored in seconds because the human interactions occur generally in durations of seconds. User's point of view of the system responses is quite important and the server resource utilization alone cannot detect performance issues. Response time is a reasonable candidate for capturing how fast the user gets the response from the server, but these times must be measured simultaneously while a large number of users are accessing the system. This is a very essential test to be conducted specially before deploying the system. In such pre-deployment tests synthetic workload becomes an unavoidable choice. If the system is benchmarked at a point of time that past records are available, then the recorded workloads can be modified and randomized by including statistical properties of the past records.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The tools and techniques discussed in the previous chapters can be extended and incorporated into a single framework as further improvements for the research outcomes. This has already being implemented as a final year project at the Department of Computer Science and Engineering.

Since the framework and systematic procedure are in place, various tests can now be devised by changing the software and hardware configurations of web based systems and also comparisons can be made between different web application platforms.

References

[1]. Moodle Community. "Philosophy." Internet: http://docs.moodle.org/en/Philosophy. May. 24, 2005 [Accessed: Mar. 10, 2008]

[2]. Dave Gehrke and Efraim Turban. "Determinants of successful website design: relative importance and recommendations for effectiveness," in *Proc. 32nd Hawaii International Conference on System Sciences*, 1999, p. 5042.

[3]. Benedict G.C. Dellaert and Barbara E. Kahn. "How tolerable is delay? Consumers' evaluations of internet web sites after waiting." *Journal of Interactive Marketing*, vol. 13, no. 1, pp. 41-54, 1999.

[4]. Samuel Kounev and Alejandro Buchmann. "Performance modelling of distributed e-business applications using queueing petri nets," in *Proc. 2003 IEEE International Symposium on Performance Analysis of Systems and Software*, 2003, pp 143-155.

[5]. Raj Jain. *Art of Computer Systems Performance Analysis Techniques for Experimental Design Measurements Simulation and Modelling*. NY: Wiley-Interscience, 1991.

[6]. Mauro Andreolini, Michele Colajanni, Riccardo Lancellotti, and Francesca Mazzoni. "Fine grain performance evaluation of e-commerce sites." *ACM SIGMETRICS Performance Evaluation Review*, vol. 32, no. 3, pp 14-23, Dec. 2004.

[7]. Grady Booch. "The architecture of Web applications." Internet: http://www.ibm.com/developerworks/ibm/library/it-booch_web/, Jun. 2001 [Accessed: April 2008]

[8]. Robert Chartier. "Application Architecture: An N-Tier Approach - Part 1." Internet: http://www.15seconds.com/issue/011023.htm, Oct. 23, 2001 [Accessed: April 2008]

[9]. Xue Liu, Jin Heo, and Lui Sha. "Modelling 3-tiered web applications," in *Proc. the 13th IEEE International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems*, 2005, pp 307-310.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

[10]. UV Ramana. "Some experiments with the performance of LAMP architecture," in *Proc. The Fifth International Conference on Computer and Information Technology*, 2005, pp 916-921.

[11]. Daniel Lopez Ridruejo. "Apache Overview HOWTO." Internet: http://tldp.org/HOWTO/Apache -Overview-HOWTO-2.html#ss2.1, Oct. 10, 2002 [Accessed: Jan. 20, 2008]

[12]. Gabriele Kotsis, and Lukas Taferner. "Performance comparison of web-based database access," in *Proc. International Symposium on Distributed Computing and Applications to Business, Engineering, and Science*, 2002, pp 360–364.

[13]. Dragan Simic, Srecko Ristic, and Slobodan Obradovic, "Measurement of the achieved performance levels of the WEB applications with distributed relational database." *Facta universitatis - series: Electronics and Energetics*, vol. 20, no. 1, pp. 31-44, April 2007.

[14]. G. Pallis, A. Vakali, L. Angelis, and M.S. Hacid. "A study on Workload Characterization for a Web Proxy Server," in *Proc. the 21st IASTED International MultiConference on Applied Informatics*, 2003, pp. 779-784.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

[15]. C. Amza, et al., "Bottleneck characterization of dynamic web site benchmarks," Rice University, Houston, Texas, Tech. Rep. TR02-391, 2002.

[16]. A. Bianco, G. Mardente, M. Mellia, M. Munaf`o, and L. Muscariello, "Web User Session Characterization via Clustering Techniques." *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp 405-416, April 2009.

[17]. Jordi Guitart, David Carrera, Vicenc Beltran, Jordi Torres and Eduard Ayguade. "Session-Based Adaptive Overload Control for Secure Dynamic Web Applications," in *Proc. the 2005 International Conference on Parallel Processing*, 2005, pp. 341-349.

[18]. Diwakar Krishnamurthy, Jerome A. Rolia and Shikharesh Majumdar. "A Synthetic Workload Generation Technique for Stress Testing Session-Based Systems." *IEEE Transactions on Software Engineering*, vol. 32 , no. 11, pp. 868-882, Nov. 2006.

[19]. Magdalini Eirinaki, "Web Mining: A Roadmap," Department of Informatics, Athens University of Economics and Business, Tech. Rep. IST/NEMIS, 2004.

[20]. Hani Jamjoom, Chang-Hao Tsai, Kang G. Shin and Sharad Singhal. "Eve: A measurement-centric emulation environment for adaptive internet servers," in *Proc. the 2007 spring simulation multiconference*, 2007, pp. 17-24.

[21]. Apache Software Foundation. "Apache HTTP Server Project." Internet: http://httpd.apache.org/, Jan. 2004 [Accessed: Feb. 2008]

[22]. D. Mosberger and T. Jin. "HTTPerf: A Tool for Measuring Web Server Performance." *Performance Evaluation Review*, vol. 26, no. 3, pp. 31-37, Dec. 1998.

[23]. David P. Olshefski, Jason Nieh, and Dakshi Agrawal, "Inferring client response time at the web server." *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 1, pp. 160-171, June 2002.

[24]. John Heidemann. "Performance Interactions between P-HTTP and TCP Implementations." *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 65-73, April 1997.

[25]. J. Case, M. Fedor, M. Schoffstall and J. Davin. "RFC1157 – Simple Network Management Protocol." Internet: http://www.ietf.org/rfc/rfc1157.txt, May 1990 [Accessed: Jan. 2009]

[26]. Tobi Oetiker. "Tobi Oetiker's MRTG - The Multi Router Traffic Grapher." Internet: http://oss.oetiker.ch/mrtg/, Nov. 19, 2008 [Accessed: Mar. 2009]

[27]. Tobi Oetiker. "About RRDtool." Internet: http://oss.oetiker.ch/rrdtool/, April 2009 [Accessed: May 2009]

[28]. Fine Connection, "Accessing the RRD," Internet: http://www.fineconnection.nl /files/manual/Help_Accessing_the_RRD.html, April 2007 [Accessed: May 2009]

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Appendix

University of Moratuwa, Sri Lanka
Faculty of Engineering
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MSc in Computer Science, 2007 Batch,

This questionnaire is part of a Research Project. Your effort to give correct and precise information is highly appreciated. Please underline the correct choice whenever there are multiple choices given.

Q1. Current Academic Year and Semester:.......................................

Q2. Field of specialization:...

Q3. How many hours per day you spend on web browsing on average?...................

Q4. What web sites you visit most often?
....
....
....

Q5. How frequently do you use University LMS (LearnOrg-Moodle) ?
a. .... times a day
b. .... times a week

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Q6. During what time of the day do you use LearnOrg-Moodle often (Please color the slots)?

6-8 am							
8-10 am							
10-12 am							
12-2 noon							
2-4 pm							
4-6 pm							
6-8 pm							
8-10 pm							
10-12 pm							
	Mon	Tues	Wed	Thur	Fri	Sat	Sun

Q7. From where do you access LearnOrg-Moodle most often?
a. From ....(Level 1, Level 2, SMART, CSE, CIT.....etc) lab of ....(CSE, ENTC, EE....etc) Dept
b. From outside university via
   i. Dial up over landline
   ii. Dial up over CDMA
   iii. ADSL
   iv. Broadband wireless

v. Other (Please specify how you connect)

:...

Please give connection speeds against the selected technologies above (if possible)

Q8. What is the web browser you often use to work with LearnOrg-Moodle?

....

Q9. Rank following operations in LearnOrg-Moodle according to how frequently you use them

Most frequent = rank 1

a.	View pages	
b.	Post text in forum	
c.	Post text in wiki	
d.	Upload file	
e.	Online quiz	
f.	Open link to pdf file	
g.	Open link to web pages	
h.	Download files	
i.	Chat	

Other (please specify and give the rank too): ....

Q10. Rank the followings giving rank 1 to the most time consuming operation you have experienced in LearnOrg-Moodle

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

a.	Login	
b.	Load a course page	
c.	Post to a forum	
d.	Post to a wiki	
e.	Upload a file of size	
f.	Submit an online quiz	
g.	Download a file of size	
h.	Post into a chat	

Other (please specify and give the rank):............................

UNIVERSITY OF MORATUWA LIBRARY SRI LANKA

Q11. How long would you tolerate to wait till a LearnOrg-Moodle page gets loaded?
.... min/ sec

Q12. What is the longest time duration you had to wait for a response of LearnOrg-Moodle (give rough value):
.... min/ sec

Q13. Time to load LearnOrg-Moodle pages is very poor/ poor/ average/ good/ very good compared to fastest web sites I have often browsed.

Q14. I will wait to work till
a. whole page get loaded on my browser
b. only the desired part of the page starts appearing

Q15. How many online tests have you faced on LearnOrg-Moodle?......................

Q16. When doing online tests, how would you finish the test?

a. Click "submit all and finish" at last moment
b. Click "Save without submitting" each page while doing the test and let automatically submit when time is over

Q17. If a page goes non responsive when I click on a button/ link

a. I would wait for some time (>1 min) without doing anything
b. I would click the button/browser's refresh button again after few (>10) seconds
c. I would keep on clicking the button/browser's refresh button several times
d. I would close the browser

Q18. How often you get error in loading LearnOrg-Moodle pages

a. Very often
b. ......% of times I use it
c. Rarely
d. Never up to now

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

RECEIVED
16 FEB 2011
LIBRARY
University of Moratuwa, Sri Lanka