**BORANG PENGESAHAN STATUS TESIS**

JUDUL: SYSTEM AND PERFORMANCE MONITORING SYSTEM

SESI PENGAJIAN: 2005/2006

Saya NOR AZITA BT ABU HANIFAH

mengaku membenarkan tesis (PSM/~~Sarjana~~/~~Doktor Falsafah~~) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Kolej Universiti Teknikal Kebangsaan Malaysia.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)

   _____ SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

   _____ TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

   _____ TIDAK TERHAD

(TANDATANGAN PENULIS)

Alamat tetap: 49 Jalan Pegaga, Tmn Merdeka
Batu Berendam, 75350 Melaka.
Tarikh: 28 April 2006

(TANDATANGAN PENYELIA)

Nama Penyelia: Cik Zakiah Ayop

Tarikh: 28 April 2006

# SYSTEM AND PERFORMANCE MONITORING SYSTEM

NOR AZITA BTE ABU HANIFAH

This report is submitted in partial fulfillment of the requirements for the Bachelor of Information and Communications Technology (Computer Networking)

FACULTY OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
KOLEJ UNIVERSITI TEKNIKAL KEBANGSAAN MALAYSIA

# DECLARATION

I hereby declare that this project report entitled

**SYSTEM AND PERFORMANCE MONITORING SYSTEM**

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT: _____ Date: 28 APRIL 2006

(NOR AZITA BT ABU HANIFAH)

SUPERVISOR: _____ Date: 28 APRIL 2006

(MISS ZAKIAH BT AYOP)

# DEDICATION

To my beloved lecturers, family and friends...

## ACKNOWLEDGEMENT

I would like to express my gratefulness to participation of the following organization or any individuals who contribute in my Projek Sarjana Muda program and the thesis project. I would like to begin my thanksgiving to my supervisor Miss Zakiah Ayop who had thought and guided me through this thesis project development.

Last but not least, I send my special thanks to the lecturers of faculty of Information Technology and Communication who really doing hard to give explanation and opinion about this thesis project.

## ABSTRACT

System and performance monitoring system (SPM) is a client/server application, which enables user to view, monitor and take control of other application on different host. Using this system, network administrator will be able to get detailed view of all the running processes or applications in client desktop and kill any applications that are deviant and cause harm to other clients in the network environment. SPM uses Remote Frame Buffer (RFB) protocol that connects the client to the server. SPM server can either allow or not allow clients to access some application through its own desktop. It also helps administrators to monitor and observe clients activities through their desktop. SPM is developed using Java technology on the Windows platform, which is the ideal platform for network computing.

# ABSTRAK

*System and Performance Monitoring tool (SPM)* adalah aplikasi klien/pelayan yang membenarkan pengguna untuk melihat, memerhati dan mengawal aplikasi dari *host* lain. Sistem ini akan digunakan oleh pengurus rangkaian untuk mendapatkan gambaran sebenar semua proses/aplikasi yang berlaku di *desktop* klien dan menamatkan aplikasi yang merbahaya dan boleh merosakkan klien-klien yang lain di dalam rangkaian. SPM menggunakan protokol *Remote Frmae Buffer (RFB)* untuk menghubungkan klien kepada pelayan. Pelayan SPM boleh sama ada membenarkan atau melarang klien untuk mengakses sebahagian aplikasi yang mendatangkan kerosakan kepada pengguna lain melalui *desktop* pelayan. *System and Performance Monitoring tool (SPM)* dibangunkan dengan menggunakan teknologi Java di Windows platform yang sesuai untuk aplikasi *network computing.*

# TABLE OF CONTENTS

# LIST OF TABLE

# LIST OF DIAGRAM

# CHAPTER I

# INTRODUCTION

This chapter will define the subject of this project. It will outline the scientific purposes and objectives for the project performed and give a sufficient background to understand the rest of the project and what has been done previously. This chapter will describe in detail about project background, problem statement, objective, scope, project significance, expected results, and conclusion.

## 1.1 Project Background

The thesis "System and Performance Monitoring server " (SPM) contributes from originality projects of System Monitoring Protocol server (SMP). SPM is an emerging technology that allows administrator to be able to monitor and remote control a system and the performance of server or workstations. The need of a project persists, as every time precautions are taken, administrator has to input workforce and time to get information about equipment. In addition, they are supported to make a report after taking precautions. In some cases, administrator is not able to pick out unauthorized Software of each PC. Networking section probably will run a call center. If there is a breakdown in one of users' sites, the call center find it through their own network forum or by phone call. The troubleshooter will try to understand advance what the problem is over the phone conversation and then prepares parts

needed. He gets access to the computer and settles the trouble. As a final step, he submits a troubleshooting report to the customer's person in charge. But for some cases, after receiving a breakdown call, it is hard to find out the cause of the breakdown, especially, when the user has changed the hardware. It might be also cause by of the difference level between PC operators, yet checking over the phone call will results in vain.

Therefore, maintenance personnel have to visit users' site in most cases and this requires time and cost. Through these obstacles, the current server which is a Real-time and interactive PC inventory management software has offered only a system monitoring function such as remotely control client computers, checking software license compliance, tracking hardware modifications and upgrade requirements and manage all PC inventory issues. So this project indeed will follow a case study and improvement of existing system as the trend recent years is user demand for smooth system access and stable system performance.

## 1.2 Problem Statements

The investigation conducted in this thesis is being done through a case study on the current application of System Monitoring Protocol server. As mentioned earlier, one of the benefits of SMP server is improving a help desk PC level with remote controlling which is the main important function in monitoring other servers or hundreds of network workstations. However, with this application an administrator would only notice the problem from user's site through the their call. As a solution, in order to save up the money and time of making a phone call, an enhancement need to be done to this system. There will be alert properties for administrator to set up performance that includes CPU usage, CPU average usage, and disk usage. If the received server or workstations performance data meet the defined condition, an administrator will be notified by

methods of sending an Email- SPM server will send an email to the defined email address via SMTP server. There is also a security problem in the current project which remote controlling functionality does not need a connection password. In the other words, in case of accidental anyone without permission could control and see clients' private data. In other cases, this system also does not have a log or history file for any operation done in remote controlling task. Any changes or task being done on workstations can be track later if any issues come up after the changes or operation made. Finally, we can say that these statements of problems have directly influenced the motive of the project and they are applicable for project to find a solution in improving system through those specific cases.

## 1.3 Objectives

In order to find solution that satisfies all of end users demand one will have to solve a number of different topics based on general network monitoring problem. Performance, scalability, reliability and security are some of these topics, which are required to be able to satisfy all possible solutions to implement. This thesis will thus try to find solutions that fulfill the objective of:

i. To create a network monitoring tools for network administrator to control client activities.
ii. Show the client desktop by remotely control function.
iii. System, which can capture all the clients' activities and the program that users used currently.
iv. System, which server could send an instant message to client in network.
v. Capture data of day, date, time and client IP address each time of connection between server and client is made.
vi. System that allow an only an authorize people to operate the remote control function.

## 1.4 Scope

This system monitoring tools can be use by network administrator to manage and control the entire network. The system is only monitoring but not analyzing the network. This tool observes the clients activities and gave action if network policies are not being followed. It can also determine the PC name, IP address and other information of the nodes in the network. It can capture the client current activities and all the process and application that are running by the clients. Administrator can also view the client desktop remotely.

The user of this project is network administrator, system administrator, network engineer or anyone who is involved in network management. This project also applies to any machine that uses Windows Operating System, which is based on NT system and server should be run with Java jdk 1.5.0.

This project also focused on LAN environment and could be used on KUTKM network environment. For all above facilities, it is very appropriate not only for network administrators but also to the user of entire network.

## 1.5 Project Significance

This project is started to help administrator to monitor the responsiveness of the servers and workstations on their sites. This would probably be benefit to those administrators as the project has considered the monitoring system performance and the effects of configuration changes on system. The thesis has been choosing to be configured on windows platforms, as it is more stable and more familiar to most users. Besides, functions in this system are easy to operate as it has been displayed in figured button that will easily user to click the appropriate function. In many situations of organization management, an action, correction, or changes made should be writing down and following by report. Furthermore, in SPM monitoring system environment, a SMP log is important in order to track SMP remote control to see a history about when it

was run against a client machines and took the responsibility. So in this case, a log history would provide precious information if there is any problem occurred.

## 1.6 Conclusion

The purpose of this thesis is to design, implement and evaluate a system performance monitoring in a new trend of networking technology that responsive to perform an effective monitoring services in such aspects. In this thesis, various application in performance monitoring systems are discussed, in order to make sure an effectiveness and full reliance of the system. The objective of the system performance evaluation of the project is to see the capability of the system, number of clients and the number agents that can be supported. Next activity will preview to the literature review and project methodology.

# CHAPTER II

# LITERATURE REVIEW AND PROJECT METHODOLOGY

This chapter will demonstrate past study related to the topic of the project. Each problem or issues discuss through the project will come out with a preposition solution and hypothesis to the problem. An extraction, analysis, drawing conclusion are made from sources such as books, journals, technical reports, web-page, e-books, anonymous references, proceeding conferences, and CD-ROMs. This is important as they imply and make use of other people works and research.

## 2.1 Introduction

Chapter two of this thesis examines several of the other tools available to perform various performance monitoring tasks. In order to help establish the reasoning behind the development of the System Performance Monitor (SPM), the similarities and differences of each tool are highlighted. As the survey of other tools continued, it became evident that in order to provide the desired network analysis, a custom tool would have to be designed. This project would develop an inexpensive, customizable network-monitoring tool that could then be modularly extended to perform a variety of interaction tasks.

## 2.2 Fact and finding

Network management is not a new field. The International Organization for Standardization (ISO) first started working on the simple network management protocol (SNMP) in the mid-1980s. Since then, numerous open-source and commercial tools have been developed. Some of these tools are designed for a specific functionality, such as just collecting network data. Other tools both collect and analyze network data making them better suited for stand-alone operation. The majority of the tools are designed for smaller networks where the network bandwidth stays below 100 Mbps. Tools capable of monitoring larger networks exist as well, however they are often only available through commercial vendors at a considerable cost. The following tool has at least one type of network management functionality comparable to a function in the System Performance Monitoring tool.

### 2.2.1 Software Monitoring Systems

Software monitoring systems are implemented by inserting an extra set of instruction into the target system to cause data capture. Both the triggering and recording phases of program monitoring are accomplished by executing the inserted code, and the recorded data is often stored in the working memory of the target system. Since an execution of the instrumentation code uses the computing power and working memory of the target system, software monitoring systems may result in an unacceptable performance penalty of the target program, and possibly their execution behavior is also affected. The potential advantages of software monitoring systems are their flexibility, and that no additional hardware is required for their implementation. Without using hardware support, the dilemma of finding a balance between minimizing interference due to monitoring and recording sufficient information about the execution behavior of a target program always exists. Limiting instrumentation on the one hand may provide inadequate measurement detail, whereas excessive instrumentation, on the other hand,

may perturb the target system to an unacceptable degree [Joyce, J., Lomow, G., Slind, K.(1987). Monitoring Distributed Systems, 2nd edition. The U.S.A: ACM Trans.Pp 121-150.] In program monitoring systems, the pre-defined events of interest are ordered according to their time of occurrence and they are replayed in the same order during visualization stage. In order to timestamp the events, a clock support is necessary. Since there is no hardware support for software monitoring systems, they rely on the target system's clock(s) and hence the instrumentation code must have an access to the target system's clock to timestamp events with the clock's readings.

### 2.2.2 Hardware Monitoring Systems

In hardware monitoring systems, a hardware device is attached to bus of the target system to passively snoop the bus and detect a set of pre-defined signals. Triggering takes place on a specific combination of the pre-defined signals. Data recording is carried out by hardware, and the recorded data is stored in a separate memory independent of the monitored system. The primary advantage of hardware monitoring systems is that their interference with the execution of the target system is minimal since the monitoring system shares no computing resource of the target system. Although such devices can be designed to have minimal or no perturbation effect on the target system, their main drawback is that they generally provide limited low-level information about the execution behavior of the target system [Haban, D., Wybranietz, D. (1990). A hybrid Monitor for Behavior and Performance Analysis of Distributed Systems. The U.S.A: McGraw-Hill. Pp 197-211.] Simple snooping of system buses, or probes connected to the processor's memory ports or I/O channels do not provide sufficient information about the target system. To collect valuable run-time information, hardware-monitoring systems often use sophisticated features of hardware. Another drawback of hardware monitoring systems is that the desired signals may not be accessible as integrated circuits technique advances and more functions are built on chips [Tsai, J. P., Bi, Y., Smith, R. (1996). Distributed Real-Time Systems Monitoring, Visualization, Debugging, and Analysis. The Netherlands: John Wiley & Sons. Inc.Pp

122-127.] Plattner [Plattner, B. (1984). Real-time Execution Monitoring. The U.S.A: IEEE Trans. Pp 756-764.] proposes a hardware monitoring system for monitoring single-processor real-time systems. In Plattner's system, a hardware device called a listener is attached to the bus of the target processor and a separate storage space called a phantom memory is used to mirror the contents of the memory of the target system in real-time. A monitoring process is employed to access all information from the phanthom memory. This implementation of program monitoring obviously does not interfere with the execution of the target system since it uses no resource of the target system. The main drawback of this system is the extra cost of constructing the phanthom memory.

### 2.2.3 Hybrid Monitoring Systems

Hybrid monitoring systems are attractive compromise between the intrusive software monitoring systems and the expensive non-intrusive hardware monitoring systems. They utilize both software and hardware approaches to program monitoring, and to minimize perturbation due to monitoring by allowing the hardware to perform the majority of the monitoring task. Hybrid monitoring systems insert instrumentation code into the target system to detect the occurrences of pre-defined events of interest. Data recording is carried out by hardware, and the collected data is saved in a separate memory independent of the memory of the target system. Hybrid monitoring systems use two different triggering approaches: *memory mapped* and *co-processor monitoring* [Tsai, J. P., Bi, Y., Smith, R. (1996). Distributed Real-Time Systems Monitoring, Visualization, Debugging, and Analysis. The Netherlands: John Wiley & Sons. Inc.Pp 141-145.] In memory mapped monitoring, a set of pre-defined addresses are used to trigger data recording. The monitoring unit is mapped onto the memory addresses with each address representing an event. In co-processor monitoring approach, the co-processor instructions are used to trigger event recording. The recording unit acts as a co-processor that executes the monitoring instructions. To invoke recording of data

pertinent to events of interest, the co-processor instruction is sent by the target processor to the monitoring unit. Haban and Wybranietz's DTM (distributed test methodology) [Haban, D., Wybranietz, D. (1990). A hybrid Monitor for Behavior and Performance Analysis of Distributed Systems. The U.S.A: McGraw-Hill. Pp 354-361.] system uses the hybrid monitoring approach to monitor program execution and to collect information pertinent to the events of interest. The main idea in the DTM monitoring system is that the target system detects significant events and these events are processed and displayed by a dedicated hardware. The DTM monitoring system is a typical example of hybrid monitoring system that employs a memory mapped monitoring approach discussed in the previous paragraph. Based on finding about these three types of monitoring system, the thesis will implement the hybrid monitoring system. We use hybrid monitoring, because the technique will combine advantages of both software monitoring and hardware monitoring in our project later.

For a finding we find the similarities and difference of two SMP monitoring tool and Performance monitoring tool (PMP), which are from the publisher Eastbow L.A.B. (http://www.eastbow.com/eng/smp_intro.htm). Basically a SMP tool is automatic PC inventory management software where user can examine hundreds or thousands of client PCs' hardware and software automatically, track hardware modifications and upgrade requirements hardware, detect pirated software, remotely control client computers, and manage all PC inventory issues. Then PMP is a performance monitor for remote/multiple servers. User can monitor the real-time CPU; memory and disk usage status from the several networked remote servers and can be notified for the specific server status. PMP provides email notification and automatic/manual remote command execution. PMP also provides HTML reporting and user can share it with their staff. We found that this two type of tool can be combined in one application, as it will bring more functionality to user. The application to be built is more to SMP monitoring tool. We will just use some function in PMP monitoring tool in order to enhance the level of help desk.

## 2.3 Project Methodology

Software development is always asked what methodology to use. The implication is that one particular methodology must be superior to all others. However, most development methodologies can be basically categorized as either a waterfall or iterative. This project chooses the waterfall approach that emphasizes a structured progression between defined phases. Each phase consists of a definite set of activities and deliverables that must be accomplished before the following phase can begin. The waterfall model abstracts the essential activities of software development and lists them in their most primitive sequence of dependency. This projects development rarely follow such a model literally, mainly because the model can and is applied to itself recursively, yielding an almost fractal fabric of actual activity. The verbs of waterfall development are 'analyze', 'design', 'code' and 'test' [Royce, J., Winston, W. (1970). Managing the Development of Large Software Systems, 2nd edition. The U.S.A: Wageningen press.Pp 76-82.] In addition, strengths of the waterfall approach include the ease of analyzing potential changes, the ability to coordinate large distributed teams, predictable budgets, and the relatively small amount of time required from subject matter experts. Typically waterfall methodologies result in a project schedule with 20-40% of the time budgeted for the first two phases, 30-40% of the time to the programming, and the rest allocated to testing and implementation time. So, this project will include a rigidly detailed set of procedures and controls to cover everything from the types of communications to use in various situations, to authorizing and tracking change orders, to the specific ways that defects are logged, communicated and re-tested.

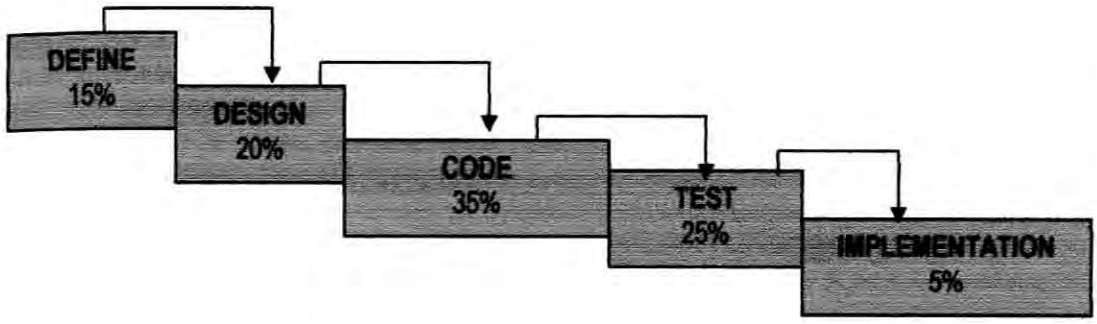


**Figure 2.1** Waterfall methodology sequence

Figure 2.1 shows a waterfall methodology structures a project into distinct phases with defined deliverables from each phase.

**i. Define phase:**

Identify project objective and constraint: The overall design process is initiated by identifying the project needs. The first step is to understand the project objectives to be met and the business constraints that will limit the project. To begin this process is to find out about the project operates in. Information on the competition, market conditions, products, and services are gathered. Next meet with the customer. At this meeting we go over the material gathered above to confirm what we have found out to develop the system. Then, we get a concise statement as to the goals of the project. What problems are to be solved? What new capabilities and functions are to be added? And what must happen for this project to be a success? Determine what will happen if the project is a failure. Is this an absolutely critical project function or just a limited test of some idea we really do not expect to work out? At this point we should have a good understanding of the basis for the project. The immediate output from identifying the project objective and constraint is a set of monitoring goals, which will initiates the technical system design process in the next phase.