NETWORK ORACLE SERVER MONITORING

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

M.Sc (APPLIED SCIENCE - COMPUTER TECHNOLOGY)

OF BHARATHIAR UNIVERSITY

Submitted by

K.Thiyagu
Reg. No – 0137Q0059

Guided by

Mr. G.S.Nanda Kumar, B.E, Lecturer,

Department of Computer Science and Engineering

Kumaraguru College of Technology
(Affiliated to Bharathiar University)

Coimbatore – 641 006

APRIL 2003

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to Bharathiar University)
Department of Computer Science and Engineering

Coimbatore – 641006

CERTIFICATE

This is to certify that the project work entitled

"NETWORK ORACLE SERVER MONITORING"

Done by

K.Thiyagu
Reg. No – 0137Q0059

Submitted in partial fulfillment of the requirement for the award of the degree of M.Sc (Applied science - Computer Technology) of Bharathiar University.

S. Thangaraj

Professor and HOD

Sunanda Kumari 02/05/03

Internal Guide

Submitted to University Examination held on 10.05.2003

R. Dinesh

Internal Examiner (10/5/03)
[ap/cse/kct]

External Examiner

Seshasayee
Paper and
Boards Limited

ISO 9001/14001 CERTIFIED

Regd Office & Works : Pallipalayam, Namakkal District
Erode - 638007, Tamilnadu, India. Ph :91-4288-240221 To 240228
Fax : 91- 4288 - 240229 E-mail : edoff@spbltd.com

Pers/Project work 2003-04-11

CERTIFICATE

This is to certify that Mr. **K. Thiyagu, M.Sc(AS-CT)** student of Kumaraguru College of Technology, Coimbatore-641006, has done a project work as a part of his curriculam on **Network Oracle Server Monitoring,** in our Organisation from **02-12-2002** to **11-04-2003.**

for Seshasayee Paper and Boards Limited

(K.RAMASAMY)
Deputy Manager(Welfare)

DECLARTION

I hereby declare that the project entitled "**NETWORK ORACLE SERVER MONITORING**" is successfully done at Seshasayee Paper and Board Limited, Erode and submitted to **Kumaraguru College of Technology**, Coimbatore afflicted to Bharathiar University as the project work of **M.Sc (APPLIED SCIENCE - COMPUTER TECHNOLGY)**, is a record of original work done by me during my period of study in Kumaraguru College of Technology, Coimbatore – 641 006, under the supervision and guidance of **Mr. G.S.Nanda Kumar B.E,** *Lecturer, Dept of CSE, Kumaraguru College of Technology, Coimbatore*. And this project work has not formed the basis of award of any Degree / Diploma / Associate ship / Fellowship or similar title any candidate of any university

Name : K.Thiyagu

Reg. No : 0137Q0059

Signature:

ACKNOWLEDGEMENT

I would like to express my gratitude to our beloved Principal **Dr.K.K.Padmanabhan**, Ph.D., Kumaraguru College of Technology, Coimbatore, for his constant encouragement throughout my course.

I wish to thank **Dr. S.Thangaswamy**, Ph.D., Head, Department of Computer science, Kumaraguru College of Technology, Coimbatore, for his invaluable guidance and suggestions that encouraged me to complete this project successfully.

I admit my heart full thanks to my course co-coordinator, **Mr. R.Dinesh**, M.S., Assistant Professor, Department of Computer Science, Kumaraguru College of Technology, Coimbatore, for being supportive throughout the tenure of my project.

I express my deep sense of gratitude to **Mr.G.S.Nanda Kumar**, B.E., Lecturer, Department of Computer Science, Kumaraguru College of Technology, Coimbatore, for her valuable guidance and constructive suggestion during the course of this project.

My heartfelt thanks and gratitude to **Mr. K.Ramasamy**, Deputy Manager, Seshasayee **P**aper and **B**oards Limited, Erode, for granting me permission to undertake the project work.

I also take this opportunity to extend my sense of gratitude to all the faculty members, non-teaching staffs of the Computer science Department, K.C.T, Coimbatore, for their guidance and co-operation rendered throughout my course.

SYNOPSIS

SYNOPSIS

Our Project involves in the creation of a tool, which helps the network database administrator in monitoring his networked database. The database taken into consideration is that of an Oracle backend. Certain mission critical parameters are constantly monitored and subsequently recursive actions are taken based upon the feedback. The values are plotted graphically for better visual comprehension. The set of values to be plotted are selected by the administrator based upon his dynamic requirements.

The basic protocol used by network administrators is the SNMP - Simple Network Management Protocol. The SNMP manager queries the SNMP agents for values of the various parameters. The critical values received in response, are analyses to determine the threshold limit, which varies according to the network environment. Preventive actions are then initiated if required.

The SNMP manager designed by us in this project identifies all the Oracle Servers configured as an SNMP agent within the network. The database administrator then queries the identified agents for their status. The manager initiates conversation through Port 161, which has been reserved for SNMP communication. The procedures to be followed during initiation and the start of a communication session between a manager and the agent are strictly adhered to.

On successful identification of the agent within the network, the administrator can view its general status and specify parameters to be calculated and monitored. The subsequent OIDs are passed on to the specified agent and queried for a result. The parameters are graphically monitored if specified.

TABLE OF CONTENTS

PAGE NO

1.0 Company Profile

1.0. Company Profile

M/s Seshasayee Paper and Boards ltd., was promoted by Seshasayee Brothers Private Limited. In 1950's India saw the birth of many newspaper mills. However it was only in 1960 India claimed to have a paper mill of its own. M/s Seshasayee Paper and Boards limited , SPB as the company is shortly called, located on the banks of river cauvery was established in June 1960, supported by over 13000 share holders and with a rated capacity to produce 20000 tones of pulps and paper per annum.

The company was the recipient of ISO 9001 and 14001 award for its consistent performance on all fronts.

This factory was established in technical collaboration with M/s Persons and Whitmore South Asia Inc., USA, which is the export arm of M/s Blacjclawson Inc., USA, the first ever in the world to cook bamboo and bagasse in the digester.

Owing to continuous short supply of bamboo, the mill started using various types of hardwood and softwood in its furnish and at present the mixture of bamboo and wood is in the ratio 10:9 such as percentage of wood in the furnish can truly be claimed as technological break through. Then SPB began to grow, expand and diverse its capacity in 1969as 35,000 tones of pulp and paper per annum.

In 1976, the paper mill embarked upon its second expansion of the project to increase its capacity to 60,000 tones per annum. In 1981, the company bagged a prestigious contract to render technical consultancy for installation of massive news print of a project based on bagasse for the TamilNadu government involving the capital outlay Rs.200 crores.

The third expansion of the company from 60,000 tones to 1, 15,000 tones is consummated in the year 2000. The company bags the First prize for environmental protection by the TamilNadu pollution control board for the year 1995-1996.

There are more than 1000 employees working in various departments of the Organization, working for the office automation jobs. All the process including the production, automation, inventory, payroll, stores, accounting, purchase, etc., are computerized.

There are more than 60 nodes connected in the network, and also more than 20 personnel computers are available in the organization situated at various departments. “Windows 2000” and “Unix” is the main server used by the organization to serve all the nodes in various departments of the concern and “Oracle” is the database management system used by the organization.

HIGHLIGHTS

Incorporation of the Company – 1960

Commencement of Production – 1962

First Expansion - 1969

Second Expansion – 1978

Third Expansion -2000

ISO 9001 – 1996

ISO 14001 – 2000

Factory Area (acres) – 101

Colony Area (acres) – 161

2.0 AN INSIGHT

The Simple Network Management Protocol (SNMP) is acknowledged as the published, open standard for heterogeneous network management applications.

Designed primarily for database, network, and system administrators, Oracle SNMP Support integrates the management of Oracle products into a number of existing, widely-used management systems. This feature enables key Oracle products running anywhere on an enterprise's network to be located, identified, and monitored by a management station running at a centrally located node, in much the same way and using much the same tools as traditionally have been used to monitor the activity of the network itself. It thereby integrates the tasks of database and of network administrators, enabling both to use some of the same tools and to better integrate their tasks. Tools using SNMP traditionally provide powerful features for monitoring network components. Oracle extends this power to enable SNMP monitoring of some of its own products.

2.1 WORKING OF SNMP

SNMP (Simple Network Management Protocol) is a standard Internet protocol enabling certain nodes in a network, the management stations or managing nodes, to query other network components or applications for information concerning their status and activities. Such a query is known as an SNMP poll. The items that can be so polled are called managed elements. Traditionally, managed elements were limited to network components such as bridges and routers, but recently the definition has been extended to include mission-critical applications such as databases. The software used by a management station is called a management framework or management platform. The management framework uses the SNMP protocol to request information from agents on the nodes being managed, and those agents send back the appropriate responses. The agents can also, independently of the framework, transmit messages called traps to well-known addresses in response to specific events. This is done to enable quick and possibly automatic reactions to the specific conditions that the traps indicate.

All requests sent to a given network node are handled by the same master agent. This agent redirects the requests to the appropriate managed elements on the node, in some cases using subagents. The protocol used for this is not yet standardized and is not SNMP.

Figure 1-1 shows the components of a management station and of a sample managed node.

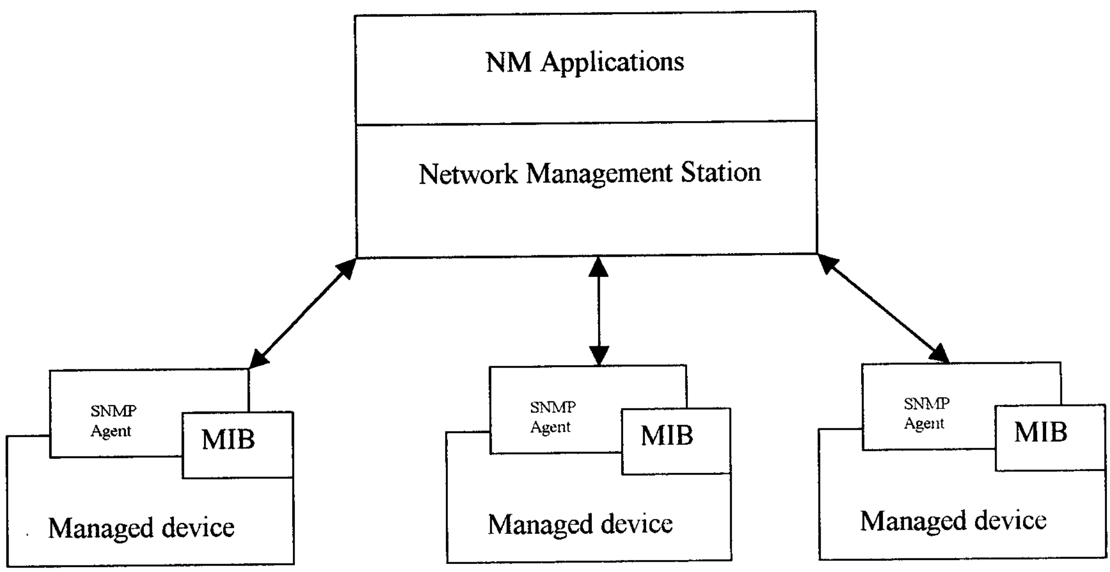


Fig. 1.1 – THE SNMP ARCHITECTURE

SNMP Components:

1. Network Management Station (NMS)
2. Management Agent
3. Management Information Base (MIB)
4. SNMP Protocol

2.2 THE COMPONENTS OF SNMP

The components shown in Figure 1-1 are explained in the sections that follow.

2.2.1 MANAGEMENT STATION

The management station refers to a node from which managed elements are monitored using the SNMP protocol. Typically, it is a stand-alone workstation that is on the same network or internet work as the managed elements.

2.2.2 MANAGEMENT FRAMEWORK

At the management station, the management framework uses SNMP to request management information from other nodes. The framework collects, graphs, and possibly acts on that SNMP data, and saves some or all of it in a repository for historical analysis and reporting. Management frameworks include many tools and options. In addition to directly requesting information from managed nodes, frameworks typically use daemons to alert them when a managed node has sent a trap in response to a specific set of conditions. The traps also can be used to trigger management applications.

2.2.3 MANAGEMENT APPLICATION

The management applications are the tools integrated with the management framework to accomplish more specialized network or database tasks. These applications contain virtually all of the sophisticated logic associated with network management. A customized management application can work with one or more frameworks (on different management stations) or run independently. Because

Oracle SNMP Support is equally accessible to any type of provider, there are many different ways that applications can utilize it. A fundamental management application, often shipped by default along with the management framework, is one that is capable of discovering the network topology and collecting some basic identification information about each discovered network entity or service. Such an application, for instance, may discover all hosts in a subnet along with their vendor, location, and status. Using this information, the management application can subsequently build up logical maps of the environment.

2.2.4 MANAGED NODE

The managed node is a platform, such as a UNIX server, on which elements to be monitored reside. In Figure 1-1, two managed elements -- an Oracle7 or Oracle8 server and Oracle Names are located on the managed node.

2.2.5 MASTER AGENT

The master agent is the process on a managed node that accepts queries, also called "polls", from the management framework and communicates with the elements to be managed in order to answer the query. It also can send SNMP traps independently in response to specific conditions. Only one master agent can exist on each managed node. Any node that does not have an agent will not be able to respond to SNMP requests, but this does not prevent other nodes on the network from doing so. In other words, it is not necessary that every node in a network be able to respond to SNMP, although this is normally desirable. The master agent may be either monolithic or extensible. If it is monolithic, it communicates directly with the elements to be managed. Although such an agent can manage multiple elements on the same node, the set of elements that it can manage is fixed when the agent is created, because the monolithic agent itself is responsible for interfacing to the managed elements. If, on the other hand, the master agent is extensible, it will use a specific subagent for each element it has to manage. That subagent is then responsible for interfacing to the element. In this scenario, new subagents can register with the master agent at any time, so new managed elements can be added dynamically. Some operating systems supply only monolithic agents. In this case, Oracle provides a master agent that can effectively treat that monolithic agent as a subagent, enabling new managed elements to be added to the node dynamically.

2.2.6 SUBAGENT

The subagent is a process that receives queries for a particular managed element from the master agent and sends back the appropriate answers to the master agent. One subagent exists for each managed element residing on the managed node (with the exception that a single subagent can handle multiple Oracle database instances on the same node). In Figure 1-1, one subagent is dedicated to the Oracle7 or Oracle8 server application and another subagent is dedicated to the Names application. The subagent(s) and master agent communicate using a multiplexing protocol dictated by the master agent. There is no standard protocol for this connection, and, while a few protocols are widely used, none is a designated standard.

Notice that the subagent for the Oracle7 and Oracle8 servers is a separate process that communicates with the server through SQL*Net (using the IPC protocol). The Oracle Names subagent, on the other hand, is embedded in the application software itself. Both of these approaches are acceptable, as the specific means the subagents use to extract SNMP values are opaque to the master agent and to the framework.

2.2.7 ORACLE ENTERPRISE MANAGER

Oracle Enterprise Manager is a system management tool which provides an integrated solution for managing a heterogeneous environment. It combines a graphical console, agents, common services, and tools to provide an integrated, comprehensive systems management platform for managing Oracle products.

Oracle Enterprise Manager does not use SNMP directly. Instead, it communicates with the gent over SQL*Net using Transparent Network Substrate (TNS) connections. The agent listens to SNMP requests and passes them on to Oracle Enterprise Manager.

2.2.8 INTELLIGENT AGENTS

The agents are intelligent processes running on remote nodes in the network. An agent resides on the same node as the service it supports. However, the agent can support more than one service on a particular node. For example, if two databases are installed on one machine, a single agent can support both databases. The agents perform such tasks as running jobs and monitoring events. They are also responsible for handling SNMP requests, if SNMP is supported on the agent's platform. The agents support SNMP so applications can communicate directly with the agent using SNMP protocol on supported platforms. The agents provide access to Oracle's database Management Information Base (MIB) variables. Although the agent supports SNMP, the Oracle Enterprise Manager Communication Daemon uses TNS to communicate with the agent. Therefore, with Oracle Enterprise Manager, you can submit jobs or events that access Oracle MIB variables through the Daemon even when the database resides on a platform that does not support SNMP.

2.2.9 MANAGEMENT INFORMATION BASE (MIB)

A management information base (MIB) is a text file, written in ASN.1 notation, which describes the variables containing the information that SNMP can access. The variables described in a MIB, which are also called MIB objects, are the items that can be monitored using SNMP. There is one MIB for each element being monitored. Each monolithic or subagent consults its respective MIB in order to learn the variables it can retrieve and their characteristics. The encapsulation of this information in the MIB is what enables master agents to register new Subagents dynamically -- everything the master agent needs to know about the subagent is contained in its MIB. The management framework and management applications also consult these MIBs for the same purpose. In Figure 1-1, two MIBs exist, one for the Oracle Server and one for Oracle Names. MIBs can be either standard (also called public) or proprietary (also called private or vendor).

The actual values of the variables are not part of the MIB, but are retrieved through a platform-dependent process called "instrumentation". The concept of the MIB is very important because all SNMP communications refer to one or more MIB objects. What are transmitted to the framework are, essentially, MIB variables and their current values.

All MIBs are, in fact, part of one large hierarchical structure, with leaf nodes containing unique identifiers, data types, and access rights for each variable and the paths providing classifications. There is a standard path structure that includes branches for private subtrees.

Each leaf of this tree provides the following information about one MIB variable:

- the variable's name
- the Object Identifier (OID) of the variable

- the variable's data type
- the access rights associated with the variable
- a textual description of the meaning of the variable

The variable's name is intended to be descriptive, whereas the OID is a number that describes the path taken through the tree to reach that variable. For example, the variable named sysContact, identified by the OID 1.3.6.1.2.1.1.4 (meaning iso.org.dod and so on), is a read-write string variable that contains contact information about the administrator of the underlying system.

All objects contained under the mgmt branch are considered standard and are tightly regulated by the Internet Engineering Task Force (IETF). For example, the standard RDBMS MIB lives under the mgmt branch and is supported by all relational database servers that claim to be SNMP-enabled. Oracle further adds its own MIB objects under the private branch to increase the manageability of its products. The following MIBs are specific to Oracle Services and are found under the {private.enterprise.oracle} branch:

- the Oracle Database MIB
- the Oracle Listener MIB
- the Oracle Names MIB

2.3 HOW SNMP COMMUNICATIONS ARE PERFORMED

Since most management information does not demand reliable delivery SNMP packets are transmitted from one node to a well-known address of another node, but no verification of successful delivery is made. The penalty for the light weight, connectionless SNMP communication is paid by the management applications, which need to verify that SNMP transactions get completed successfully within a reasonable amount of time. If SNMP packets get lost in the network, the application cancels the associated transaction and possibly re-initiates it.

The most popular SNMP implementation uses the User Datagram Protocol (UDP) over the Internet Protocol (IP), although implementations also exist over other protocols, such as Novell's Internet work Packet Exchange (IPX) and Apple's AppleTalk.

2.4 SNMP SUPPORT ON ORACLE

Oracle SNMP Support is provided for the following products:

- Oracle7 Server, release 7.2 (compatible back to 7.0), and higher, on certain platforms
- two networking services (Network Listener and Oracle Names, bundled with SQL*Net release 2.2 and higher)
- Oracle Enterprise Manager

On many platforms, an SNMP master agent is provided directly along with the operating system.

2.4.1 CONFIGURING SNMP FOR ORACLE

This section discusses the general procedures for configuring SNMP for Oracle databases and Oracle ENTERPRISE MANAGER.

2.4.2 CONFIGURING SNMP ON WINDOWS NT

To configure the Oracle SNMP support on a managed node, follow the procedure outlined below:

Specify the port where the master agent is listening.

The port is specified in the TRANSPORT section of the MASTER.CFG file located in the $ORACLE_HOME\network\admin directory.

For example, add the following section to the file:

```
TRANSPORT   ordinary     SNMP
OVER UDP SOCKET
AT PORT 161"
```

Specify the authentication in the COMMUNITY section of the MASTER.CFG file.

```
COMMUNITY public
ALLOW ALL OPERATIONS
USE NO ENCRYPTION
```

Continue to Step 3 if the Encapsulator is to be used.

Specify an unused port where the encapsulated agent, Microsoft SNMP Service, should be listening. Microsoft SNMP Service typically uses port 1161.

The port is specified in the SERVICES file located in the NT_HOME\SYSTEM32\DRIVERS\ETC directory.

For example, make sure that you have the following line in the file:

```
Snmap                 1161/udp          snmp
```

Edit the Encapsulator configuration file, ENCAPS.CFG, located in the ORACLE_HOME\PEER\ADMIN directory to specify which non-PEER master agents are to be encapsulated.

You must add at least an AGENT entry, including MIB-subtrees manageable by NMS, for the encapsulated master agent.

For example, you should have a section in the file. See the example below:

```
AGENT AT PORT 1161 WITH COMMUNITY public
SUBTREES 1.3.6.1.2.1.1,
1.3.6.1.2.1.2,
1.3.6.1.2.1.3,
1.3.6.1.2.1.4,
1.3.6.1.2.1.5,
1.3.6.1.2.1.6,
1.3.6.1.2.1.7,
1.3.6.1.2.1.8,
1.3.6.1.4.1.77
FORWARD ALL TRAPS;
```

2.5 BENEFITS OF ORACLE SNMP SUPPORT

The primary benefits of Oracle SNMP Support include the following:

- The monitoring of key Oracle products is quickly integrated into any management framework based upon SNMP.
- These Oracle products are located, identified, and monitored in real time across enterprise networks of any size.
- Administrators see standard Oracle icons that represent Oracle products in a network map. This map is dynamically customizable. In fact, administrators can define and customize various network maps for different purposes.
- Administrators see the current status of Oracle products, as shown by several status variables that are defined for each product in a management information base (MIB), or they can select which elements to view on the basis of their status.
- Administrators can anticipate exceptional conditions by defining thresholds and alerts, to respond to special situations as soon as they occur or to enable automatic responses.

- Administrators can more readily determine key characteristics of Oracle objects, such as database size, number of users, and activity level.
- Administrators can store and analyze historical data that has been obtained through SNMP.

Strictly speaking, Oracle SNMP Support is intended more for monitoring Oracle products than for managing them. Oracle SNMP Support is invaluable for tracking the status of an entire network of Oracle applications -- first, to verify normal operations, and second, to spot and react to potential problems as soon as they are detected. However, for purposes of investigating and ameliorating some problems, other Oracle tools such as Oracle Server Manager may be more appropriate. This is because Oracle SNMP Support is designed to query status, but not to change system parameters, whereas other tools are designed to set or tune system parameters. Oracle does not support using SNMP to change, as opposed to query, system parameters primarily because the security that SNMP currently can provide is not considered adequate.

3.0 COMPARITIVE ANALYSIS

NetPol - A manager for an Oracle Server configured for SNMP service is unique of its kind. Though SNMP monitoring tools are available, largely for monitoring passive networked elements like routers and bridges, this tool is designed specifically for monitoring the Oracle server configured with SNMP utility.

This customized tool designed with an appealing visual interface, based upon SNMP concepts for M/s Seshasayee Paper and Boards Limited, helps the database administrator in monitoring the server dynamically rather than ping the server at specific intervals for values. Besides, the tool can be customized according to his requirements.

The tool conceals within its simplicity the complexities of the initiation, start & end of an SNMP session (for monitoring the critical parameters for their threshold limit) with the Agent via the use indigenously developed component. The Component Object Model (COM) – based set of ActiveX controls are supported in more development environments than ever before. Our ActiveX (ATL) controls and sample applications plug into all popular development environments. Some of the advantage include: viz:-

- Multi-tier architecture that separates all processing into 3 layers—no competing product offers this level of flexibility and modularity
- Formatting tier converts bytes into objects

- Upper-layer protocol tier (Telnet, FTP, SMTP, etc.) implements communication protocols

- Winsock tier handles all communication tasks
- Blocking behavior is included, so you can write applications using blocking or non-blocking operation. Scripting languages such as VB and JavaScript are now fully supported.
- Superior Object Model maximizes your effectiveness in your development environment—collection properties provide superior list handling, formatting objects make it easier to manipulate complex structures (like MIME and SNMP messages), and the use of enumerated types and objects makes your job as simple as it should be.

The controls provide concise programming models for the protocols and contain events to help monitor lengthy operations, and the formatting objects transform data between network bytes and COM objects with which your application can easily work.

4.0 SYSTEM ANALYSIS

The objective of the project is to design a manager application, which monitors the oracle server configured with SNMP service. This has been realized by dividing the entire project into 3 modules each of which has been explained in detail, below:

4.1. SNMP OID ANALYSIS

During this phase the parameters to be monitored were thoroughly studied. Object Identifiers (OIDs) are objects whose values are to be monitored *(stored within MIB (Management Information Base))*. In order that the Oracle server is monitored, the server should be configured with the SNMP facility.

SNMP recognizes 3 categories of objects: - SNMP Manager, SNMP Agent, MIB. SNMP manager is that tool that monitors any agent through queries. The SNMP agent responds to the SNMP manager with the parameter values. The critical values to be monitored are specified using SNMP OIDs.

4.1.1 READING THE MIB VARIABLE DESCRIPTIONS

The MIB variables monitored are described in specific terms. Each of the attribute of the MIB Variable is described in detail:-

4.1.1.1 VARIABLE NAME

Syntax

Maps to the SYNTAX element of SNMP MIB definition, Version 2.

Max-Access

Maps to the MAX-ACCESS element of SNMP MIB definition, Version 2.

Status

Maps to the STATUS element of SNMP MIB definition, Version 2.

Explanation

Describes the function, use and precise derivation of the variable. (For example, a variable might be derived from a particular configuration file parameter or performance table field.) When appropriate, incorporates the DESCRIPTION part of the MIB definition, Version 2.

Typical Range

Describes the typical, rather than theoretical, range of the variable. For example, while integer values for many MIB variables can theoretically range up to 4294967295, a typical range in an actual installation will vary to a lesser extent. On the other hand, some variable values for a large database can actually exceed this "theoretical" limit (a "wraparound"). Specifying that a variable value typically ranges from 0 to 1,000 or 1,000 to 3 billion will help the third-party developer to develop the most useful graphical display for the variable.

Significance

Describes the significance of the variable when monitoring a typical installation. Alternative ratings are Very Important, Important, less important, or not normally used. Clearly, the DBA will want to monitor some variables more closely than others. However, which variables fall into this category can vary from installation to installation, depending on the application, the size of the database, and on the DBA's objectives. Nevertheless, assessing a variable's significance relative to the other variables in the MIB can help third-party developers focus their efforts on those variables of most interest to the most DBAs.

Related Variables

Lists other variables in this MIB, or other MIBs implemented by Oracle, that relate in some way to this variable. For example, the value of this variable might derive from that of another MIB variable. Or perhaps the value of this variable varies inversely to that of another variable. Knowing this information, third-party developers can develop useful graphic displays of related MIB variables.

Suggested Presentation

Suggests how this variable can be presented most usefully to the DBA using the management application: as a simple value, as a gauge, or as an alarm, for example.

4.1.2 MIB CLASSIFICATION

4.1.2.1 NETWORK SERVICES MIB

The public Network Services MIB (as described in RFC 1565) contains generic variables designed to apply to all types of network service applications. Oracle has implemented those variables of this MIB that are relevant to two of the Oracle services:

- Oracle7 Server
- Oracle Names

Specific values for these variables are retrieved from a variety of sources including configuration files, tables internal to the network service application, and from the master agent itself. The Listener does not make use of any Network Services MIB variables.

4.1.2.2 PUBLIC RDBMS MIB

The public RDBMS MIB is the proposed standard MIB for relational databases that has been defined by the IETF Working Group. This MIB allows fo database discovery, identification of the database, and characterization of database size and activity level.

The public RDBMS MIB includes management variables that are both common to all RDBMS and independent of vendor. While this MIB makes a clear distinction between the database and its server, at this time the Oracle database subagent only recognizes the standard Oracle configuration of one database to one server. Thus, it does not account for the Oracle Parallel server or gateways. The information in this MIB is mostly retrieved from dynamic performance tables (V$ tables) and the INIT.ORA configuration file of the Oracle7 and Oracle8 Servers.

4.1.2.3 PRIVATE ORACLE DATABASE MIB

The private Oracle Database MIB contains additional RDBMS statistics that are specific to the Oracle7 architecture. Like the public RDBMS MIB, the private Oracle Database MIB derives most of its information from dynamic performance tables of the database and from the INIT.ORA database configuration file.

The private Oracle Database MIB is platform-independent. Thus, one Oracle7 or Oracle8 Server MIB applies, whether supported on MVS or Windows or SCO. This allows a DBA to use one MIB, regardless of the number of platforms being managed.

4.1.2.4 OTHER ORACLE PRIVATE MIBS

In addition to the private database MIB, Oracle has defined private MIBs for the following products:

- Oracle Network Listener
- Oracle Names
- Oracle Enterprise Manager

4.1.3 INTERPRETING SNMP OIDS

This section briefly covers how SNMP object identifiers (OIDs) are assigned, as it pertains to the MIBs that Oracle has implemented. Specifically, this section covers the following topics:

- interpreting Oracle OIDs
- interpreting OIDs for Oracle-implemented public MIBs
- interpreting OIDs for instances of private Oracle MIB variables

For more information on SNMP OIDs, see any of the standard SNMP texts listed in the preface.For easy reference, tables listing the object identifiers for each object of a given Oracle-implemented MIB can be found in the appendix covering that MIB.

4.1.3.1 INTERPRETING ORACLE OIDS

The SNMP standard (RFC 1442) specifies that an object identifier (OID) be used to uniquely identify each object. An OID is a sequence of elements that indicates a *hierarchical* organization of identifiers. These elements take the form of a series of "dotted" integers, similar in format to an Internet address.

An example OID for a private Oracle MIB variable follows:

1.3.6.1.4.1.111.4.1.7.1.1

In this OID:

- The first element (1) refers to the *iso* object
- The second element (3) refers to the *org* object
- The third element (6) refers to the *dod* object

- The fifth element (4) refers to the *private* object
- The sixth element (1) refers to the *enterprises* object
- The seventh element (111) refers to the Oracle object

Each of the elements listed above are assigned by entities outside Oracle. Thus, all objects within the Oracle ID space (that is, those objects to which Oracle has assigned OIDs), share the root OID 1.3.6.1.4.1.111.

Continuing use of the above example OID for illustration, Oracle has defined the elements of its private ID space as follows:

- The eighth element (4) is the MIB service identifier. Oracle has defined the following MIB service identifiers:

  4 - Oracle Private Database MIB

  5 - Oracle Listener MIB

  6 - Oracle Names MIB

  7 - Oracle MultiProtocol Interchange MIB

- The ninth element (1) is the MIB object identifier. Oracle has defined the following MIB object identifiers for each service:

  1 - SNMP variable

  2 - SNMP trap

- The tenth element (7) is the MIB table identifier (oraDbConfigTable, in this case).
- The eleventh element (1) is the table entry identifier. This element can be considered as a place holder and is always 1.
- The twelfth element (1) is the leaf object identifier (identifying a particular variable in the table).

4.1.3.3 INTERPRETING OIDS FOR INSTANCES OF ORACLE MIB VARIABLES

Because variables in Oracle's MIBs are defined in tables, there can be multiple instances of a single variable. If, for example, there are two Oracle databases running on a given managed node, each database will have its own value for MIB variables such as ApplInboundAssociations, rdbmsDbName, and rdbmsSrvInfoDiskReads.

Not all variables are indexed on a per-service basis, as described above. It is also possible for a variable to have many instances for a single database. For example, while rdbmsSrvParamEntry describes a single database configuration parameter, that same managed node will have many instances of rdbmsSrvParamName.

To uniquely identify the multiple instances of these variables, each MIB table is indexed by one for more variables which, together, uniquely identify the rows of the table. (These index variables are conceptually equivalent to the primary key of a relational database table.) To refer to a particular instance of a variable, concatenate the variable's OID with the values of the index variables of the MIB table to which the variable belongs.

For instance, rdbmsDbName is defined within the rdbmsDbTable, which is indexed by the variable rdbmsDbIndex. For example, assume two databases are running on a host, one with SNMP index 2, the other with SNMP index 4. Then the name of the first database can be specified by concatenating the OID for rdbmsDbName (1.3.6.1.2.1.39.1.1.1.4) with the appropriate value of rdbmsDbIndex (2), or 1.3.6.1.2.1.39.1.1.1.4.2. Similarly, the name of the second database is the value of 1.3.6.1.2.1.39.1.1.1.4.4.

If a table is indexed by more than one variable, add the appropriate value of each index variable to the end of the OID, in the order they are listed in the table's MIB definition INDEX clause, separated by dots. The size (oraDbDataFileSizeAllocated, or 1.3.6.1.4.1.111.4.1.3.1.3) of the fifth data file (5) of the second database on the above host (whose rdbmsDbIndex is 4) is the value of 1.3.6.1.4.1.111.4.1.3.1.3.4.5. The anatomy of the OID is thus studied and parameters with respect to Oracle analysed. The critical parameters required to be monitored for the apt functioning of an Oracle Server are understood and the permutations and combinations of the OIDs researched.

4.2. THE MANAGER APPLICATION

This Phase involved the creation of the graphical interface tool - an SNMP manager, which identifies the Oracle servers, configured with SNMP support within the network the administrator can choose between parameters displayed, according to the dynamic requirements of the network traffic. This option lends the administrator a helping hand in efficiently monitoring the system.

Each of the parameters checked spawns its own function, passes its own list of OIDs required for calculating the effective value of the parameter. The values of the parameters are then numerically displayed via a numerical gauge. The numerical display counter updates the value displayed, in specific intervals of time known as the measurement interval. The functions specified are specific to the agent and may be repeated under different instances. Thus, it was decided to create a manager component tool, which engulfed all the complexities of internal assembly level coding required during the establishment of a communication session between a manager and an agent. The component so developed is called the manager component. The component-based architecture brings about a rigid and structural framework to the system design. The probes activated during specific time intervals measure the server performance and evaluate it using the specified parameters. The run time values are then displayed for numerical reference.

4.2.1 FEATURES OF NETPOL- THE SNMP MANAGER TOOL

NetPol has been so designed that it initiates a communication session between the specified SNMP agent and the manager. The database administrator can specify the Host-Id of the Oracle server agent he would like to monitor.When specified, NetPol verifies whether the agent is present within the network and is active. If not, NetPol suspends the search and notifies the absence of the agent within the network.

Upon addition of the agent, the DbAdministrator is given the option of verifying its general properties through the General Properties tab. He could specify the parameters to be monitored via the specifications tab. Each of the checked parameters spawns its own list of SNMP OIDs and the values are calculated according to the formulae specified. The communication via port 161 is initialized, so that the specified Oracle Agent can be polled for the checked parameters.

When the numerical display tab is clicked. The entire port is blocked for the probe interval of 10000Mseconds during which the values are obtained. The measurement interval is clocked at 30seconds i.e. the Agent is probed for values for every 30 seconds. The values so obtained are then displayed in the numerical gauge. The values obtained after poll are updated in the database. The database consists of uniform tables for each of the parameters. The updating interval and the measurement interval are the same, as specified within the Oracle Installation guide. The parameters probed are explained in detail in the next section.

4.2.2 PARAMETER SPECIFICATIONS

1. Block Changes per Transaction = $\frac{\text{oraDbSysDbBlockChanges}}{\text{oraDbSysUserCalls}}$

This ratio measures the amount of Data Manipulation Language (DML) work that each transaction performs. Creating or dropping indexes impacts this value, because changes to index blocks increment it.

2. Block Get Rate = $\frac{\text{(oraDbSysConsistentGets+oraDbSysDbBlockGets)}}{\text{time unit}}$

This ratio determines the block get rate. The block get rate is a basic measure of the rate at which the application system references the database. The time unit typically used in this ratio is one second.

3. Block Visits per Transaction = $\frac{\text{(oraDbBlockGets+oraDbSysConsistentGets)}}{\text{OraDbSysUserCommits}}$

This ratio measures the work database load imposed per transaction; if it is moving Independently, then this strongly indicates that there has been a change in the application workload.

4. Cache Hit Ratio = $\frac{\text{(oraDbSysConsistentGets + oraDbSysBlockGets - oraDbSysPhysReads)}}{\text{(oraDbSysConsistentGets + oraDbSysBlockGets)}}$

This ratio measures the effectiveness of the buffer cache. The normally acceptable range is 70 - 85%.

5. Call Rate = $\frac{\text{(oraDbSysRecursiveCalls + oraDbSysUserCalls)}}{\text{time unit}}$

This ratio measures the work demand rate being placed on the instance from all work sources. It should be noted, however, that this rate may not be directly comparable across application system version changes where row at a time loop constructs have been recoded as set operations or vice versa. Use of the array interface will also affect this ratio.

6. Calls per Transaction = $\frac{\text{oraDbSysUserCalls}}{\text{oraDbSysUserCommits}}$

This ratio measures the number of client requests made per transaction. Calls per transaction can be used to detect changes in the application, or in the ways in which it is being used. This value may rise sharply as ad hoc queries increase.

6. Changed Block Ratio = $\frac{\text{oraDbSysDbBlockChanges}}{\text{(oraDbSysBlockGets + oraDbSysConsistentGets)}}$

This ratio measures the balance between queries and DML within this database application. Changes in this ratio indicate and/or quantify changes in indexation or application usage.

7. LibraryCacheMissRatio = $\frac{\text{(oraDbLibraryCachePins - oraDbLibraryCacheReloads)}}{\text{oraDbLibraryCachePins}}$

If this ratio begins to rise, then resource usage can be expected to increase. A rising library cache miss ratio may be due to wider use of application functionality causing more SQL statements and stored procedures to be active than had previously been the case.

8. Transaction Rate = oraDbSysUserCommits

The transaction rate is a basic measure of application work, and would be calibrated in transactions per second (tps) for a typical OLTP benchmark. Administrators should be particularly concerned if a fall in this value is associated with a rise in the number of connected users or vice versa. Changes in application structure or work patterns can also distort this figure.

9. User Call Rate = oraDbSysUserCalls

This rate measures the work demand rate being posed by client side applications running under the instance. It should be noted, however, that this may not be directly comparable across application system version changes where code has been moved from client to server side or vice versa.

10. User Rollback Ratio $= \frac{\text{oraDbSysUserRollbacks}}{(\text{oraDbSysUserCommits} + \text{oraDbSysUserRollbacks})}$

The user rollback ratio indicates the rate at which application transactions are failing. Rolling back a transaction uses significant resources, and would seem to indicate that all of the resources expended in executing the transaction have been wasted.

4.3 OFFLINE GRAPHICAL REPRESENTATION

The Graphical Representation tool plots the values of the parameters stored within the database. The tool has been so designed as to calculate the range of the values from the stored tables dynamically, i.e. the range of values referred to within the x-axis differs according to the range of the period specified for the particular parameter.

The graph plots a range of values for a single parameter at any instance. The host for which the values are to be plotted is to be selected from the list provided. The updation interval remains the same for all the parameters i.e. 30 secs. The offline tool incorporates a database scheduler for cleaning up the database. The scheduled activity is updated within the Windows .INI log file for referance by the system. Thereby, the database administrator can schedule the record deletion activity according to his needs. He could also customise the scheduling activity.

5.0 SYSTEM DESIGN

This Chapter describes the design architecture of NetPol - the manger tool for oracle Server configured with SNMP service.

5.1 FUNCTIONAL BLOCK DIAGRAM

NetPol comprises of a Listener component which receives the Trap Signals from the Oracle Agent a Trap signal is an urgent signal sent by the agent in response to a sudden change in the network. The signal consists of parameter values which are to be reset within the manager. The parameter values are specific to the agent.

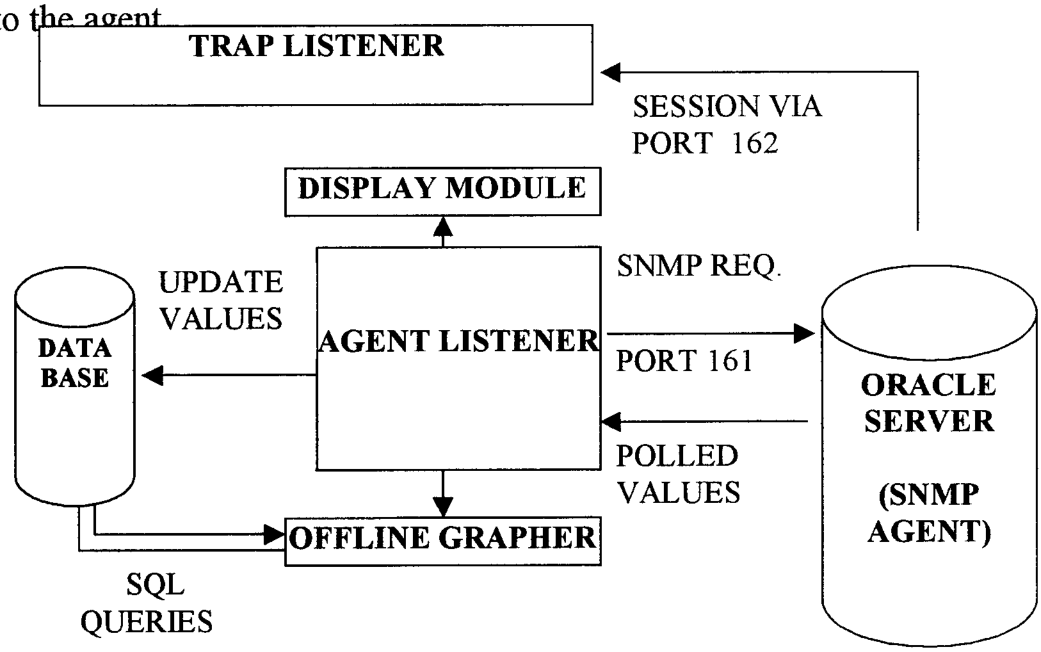


Fig. 4.1

Illustrates the functional block diagram of the manager.

Besides the Trap Listener the tool comprises of an Agent Listener module. This is the module of NetPol which communicates with the Agent at specified intervals. Each and every communication will involve initiating, opening & closing the session with the specified agent. The session complexities are handled by the Listener with flair.

The Display unit displays the values passed on by the Listener. Though communication is one-way most of the time, session turns to be full-duplex when the display unit commands the listener to pause communication.

The database block is used to store the values obtained in response. This block is explained in detail in the next section. The Graph is an offline display unit which plots the values from the database based on the specified parameters. This offline graphical unit helps in comparison of values of the monitored agents. The scheduler is that block that is used to schedule the cleaning activity of the stored database values.

5.2 DATABASE DESIGN

The database maintained by NetPol comprises of a unique relationship between Each of the tables specified for each of the 8 parameters being monitored. The tables are referenced by unique index values. The primary key values for each of the tables are the HostID which is unique and does not carry a null value.

The parameters monitored include:

1. Block Changes Per Transaction 2.Block Visits per Transaction 3.Cache Hit Ratio 4. Calls per Transaction 5.Library Cache Miss Ratio 6.Transaction Rate 7.User Call Rate 8. User Roll Back Ratio 9.Changed Block Ratio 10.Block Get Rate

The tables maintained for each of the parameters are of the similar design.

5.3 SEQUENCE SPECIFICATIONS

The event history of NetPol can at best be explained with snapshots of important events that take place. This section details the event sequence specifications of such important events.

5.3.1 ADDITION OF AGENT WITHIN NETWORK

Whenever an agent is specified to be added into the list of the agents to be monitored, NetPol will have to validate the existence of the specified agent within the network. The existence of an agent within the network is found out by a broadcast message send by the manager to all the nodes within the network (addressed as 255.255.255.255) asking it to identify itself. The string values are then compared. During the broadcast session the entire Port 161 reserved for SNMP communication is blocked for a period of 10 seconds.

5.3.2 AGENT MISSING IN NETWORK

If the agent specified for addition within the network is missing the manager intimates the administrator of the viewpoint. The session releases the blockage after a period of 10 seconds. The Agent can be reported as missing even when there isn't any communication from the agent within the stipulated time period due to unprecedented congestion in the network.

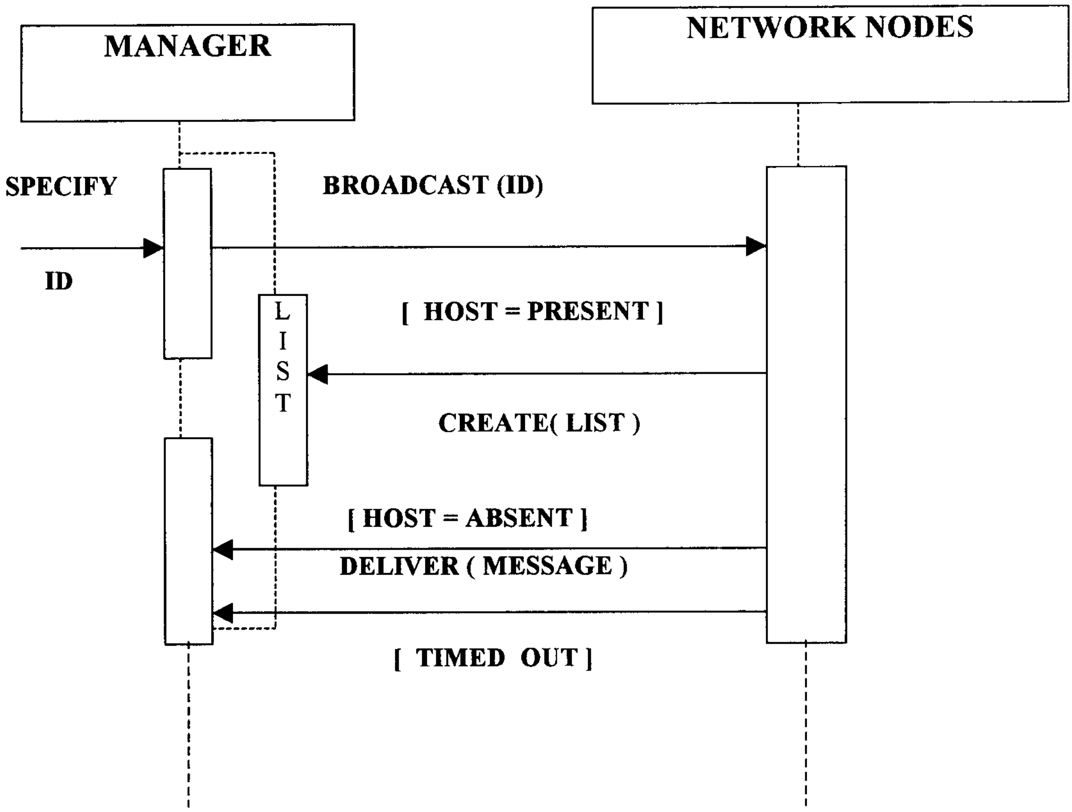


Fig 4.2
Explains the event sequence.

5.3.3 IDENTIFYING THE AGENT

The Agent being monitored can be identified through the general properties tab. The parameters used to specify the general properties of the agent include: - SysName, SysLocation, SysUptime, SysDowntime, SysId & SysSpecification. The values are obtained at run time by passing the respective OIDs to the agent clicked, within the list of the agents to be monitored.

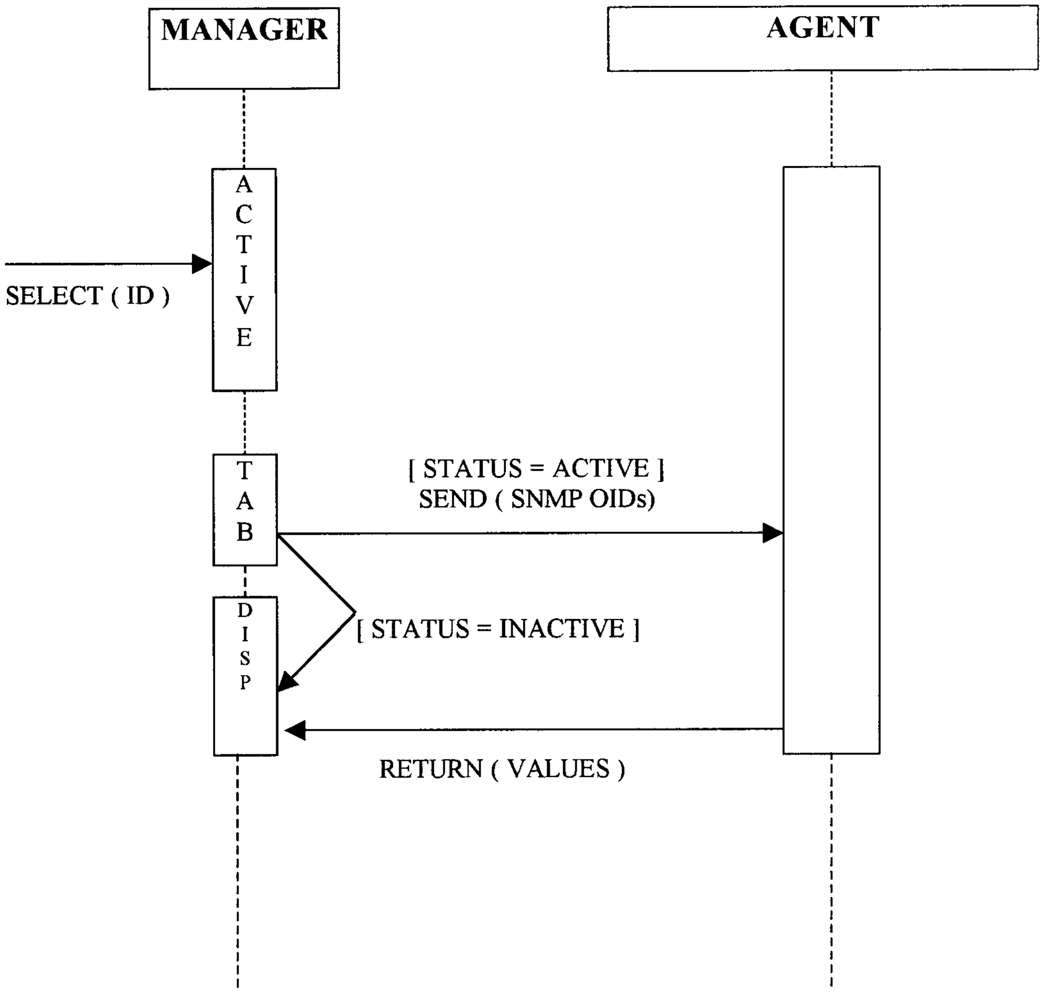


Fig. 4.3

An Instance Event Sequence Diagram

5.3.4 PARAMETER SPECIFICATION

The parameters to be monitored for the specified agent are too checked at the list of options available. On submission, the parameters cannot be rechecked. The manager application queries the agent for the specified parameter values and waits for a period of 10seconds. The values obtained in return are displayed in the numerical gauge. Simultaneously, the values are updated within the database for future reference. Incase the values aren't returned within the stipulated time period, the session is closed and the manager intimates the administrator that the agent is

Busy. The communication session is opened when the parameters are specified and closed either by the manager or by the administrator. The updation and the measurement intervals remain the same for the Oracle Database server.

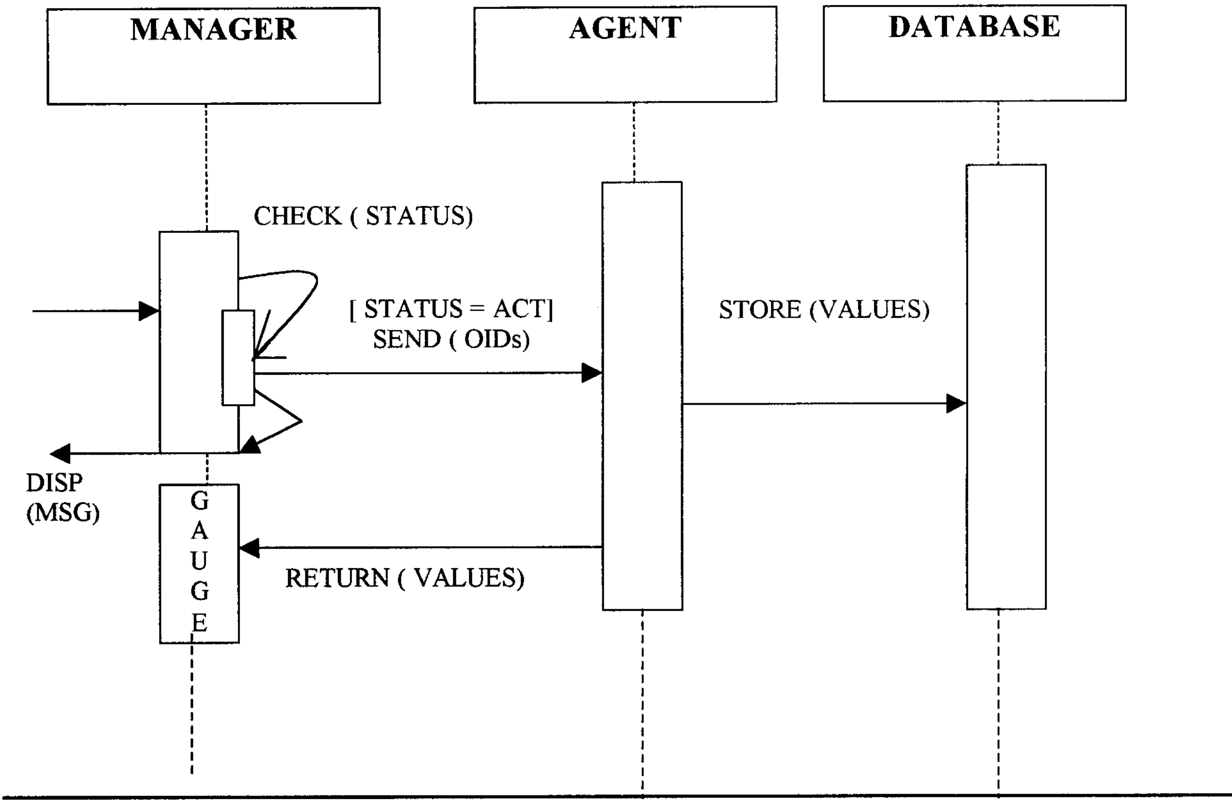


Fig. 4.4

An Instance Event Seq. Dig.

5.3.5 NUMERICAL DISPLAY

The Numerical Display gauge displays the values obtained as a result of the parameters polled at the agent. The values are refreshed by every 10 seconds (the measurement interval).When the administrator pauses the application, the agent listener temporarily suspends the session, to be resumed later-on. During this period values aren't updated at the database but the ADODC connection remains open.

5.3.6 OFFLINE DISPLAY

The Offline graphical representation unit plots values from the database. The values to be fetched from the database are specified by the SQL query. The primary key is the Hosted whose value is unique. Based upon the HostID selection the date, time & parameter to be plotted are to be selected.

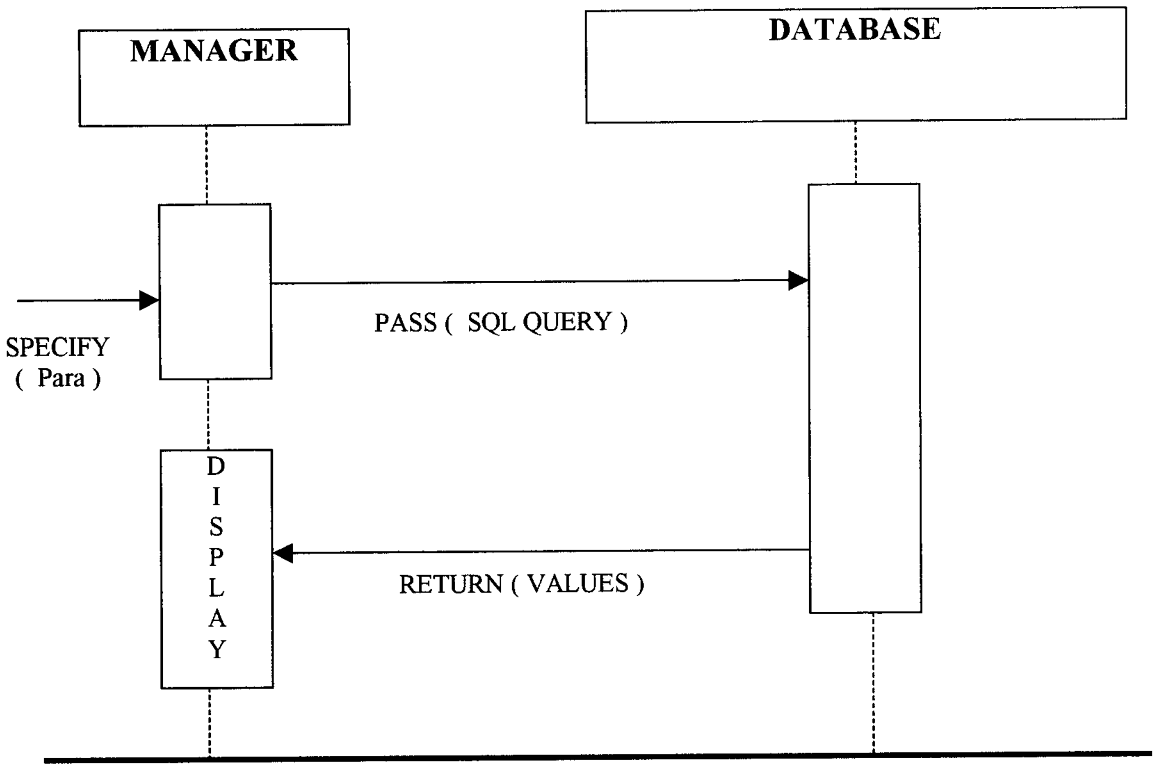


Fig 4.4

An Instance Event Sequence Diagram

6.0 SYSTEM TEST

NetPol was tested for its intended usage before its complete installation. The testing phase which lasted for a period of 2 weeks concentrated at the functional aspects of the entire software. The application was tested for its integrity and environmental compatibility.

6.1 INETGRATION TESTING PHASE

This phase concentrated at testing the bottom-line functioning of the software. The various modules developed separately were tested for their functionality after their integration into a complete composite software unit. Besides testing the modules separately, each of the modules was analysed trough static and dynamic testing. Static testing involved native code analysis while dynamic testing involved testing the modules with test data developed after analysis of the dynamic network environment.

The code was tested for its response to the following scenarios

6.1.1 CASE STUDIES:

6.1.1.1 An agent missing

6.1.1.2 Specified agent suddenly crashes

6.1.1.3 Specified agent doesnot reply

The specified scenarios are run time situations which may occur due to network congestion. The software gracefully exited from all the bottle-necks. If timed – out (Case 2 & 3), the agent requests the administrator for further course of action. In all the other cases the software determines a course of action by itself. All the situations in which conditions 2 & 3 can arise were thoroughly tested to prevent the software being stalled at any point of its execution.

6.1.1.4 Unspecified database entry, null input in database, database connectivity

The database was tested for its robustness. By specifying test data speculating various conditions. The database entries were verified through integrity checks and enforcement of constraints.

6.1.1.5. Scheduler scheduled activities

The database scheduler was checked for its functioning at various scheduled times, its suspension and revoking procedures were put to strenuous tests. The robustness of the software was put to test through test data specifications.

6.2 ENVIRONMENT TESTING

This phase involved testing the software for its functioning in the prescribed environment. The software was installed in a network running on a Windows NT platform. The installation package was tested for its working in the above platform. The components developed were installed as DLLs in the native system and locally tested.

CONCLUSION

7.0 CONCLUSION

NetPol is an attempt towards creation of an easy to comprehend visual interface that monitors the SNMP agent during run-time. Besides taking into consideration the practical working environment of a database administrator, NetPol has been designed for better future enhancements.

The Graphical display engine of NetPol which plots the values offline has a definite scope for being incorporated as an online one. The online unit can be clubbed with the numerical gauge to provide the Db Administrator a better on-screen display. The graphical display tool, at the moment plots the values of a single parameter at a time. This provides another arena for improvement where-in the tool can be coded to plot values of specified parameter of multiple hosts, for comparison. Besides, the tool can be redesigned to plot values of multiple parameters of a specific SNMP agent.

Though the DB Administrator can specify a number of agents (Oracle Servers configured with SNMP service) at a single instant, to be added on within the list of agents to be monitored, NetPol V-1 monitors only a single agent at an instant. To monitor the other agents, a separate instance of NetPol will have to be initialized. An improvised version could therefore include the facility where-in a single instance of NetPol monitors all the listed SNMP Oracle Agents.

The next version of NetPol would incorporate the above mentioned modifications. Besides, SNMP agents other than the Oracle Server, configured with the SNMP utility would be added on after a thorough research on its dynamic requirements, its native ratios, critical parameters, and their respective SNMP OIDs.

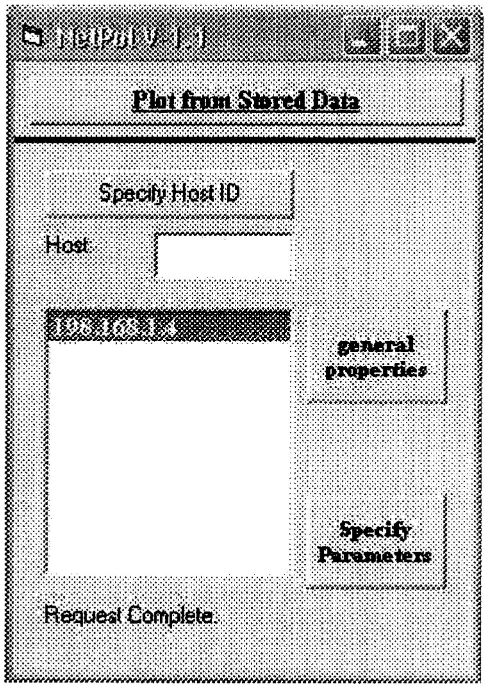
This would make the NetPol - (Network Police) - a boon to the Network Admistrator, true to its name.

DATABASE DESIGN

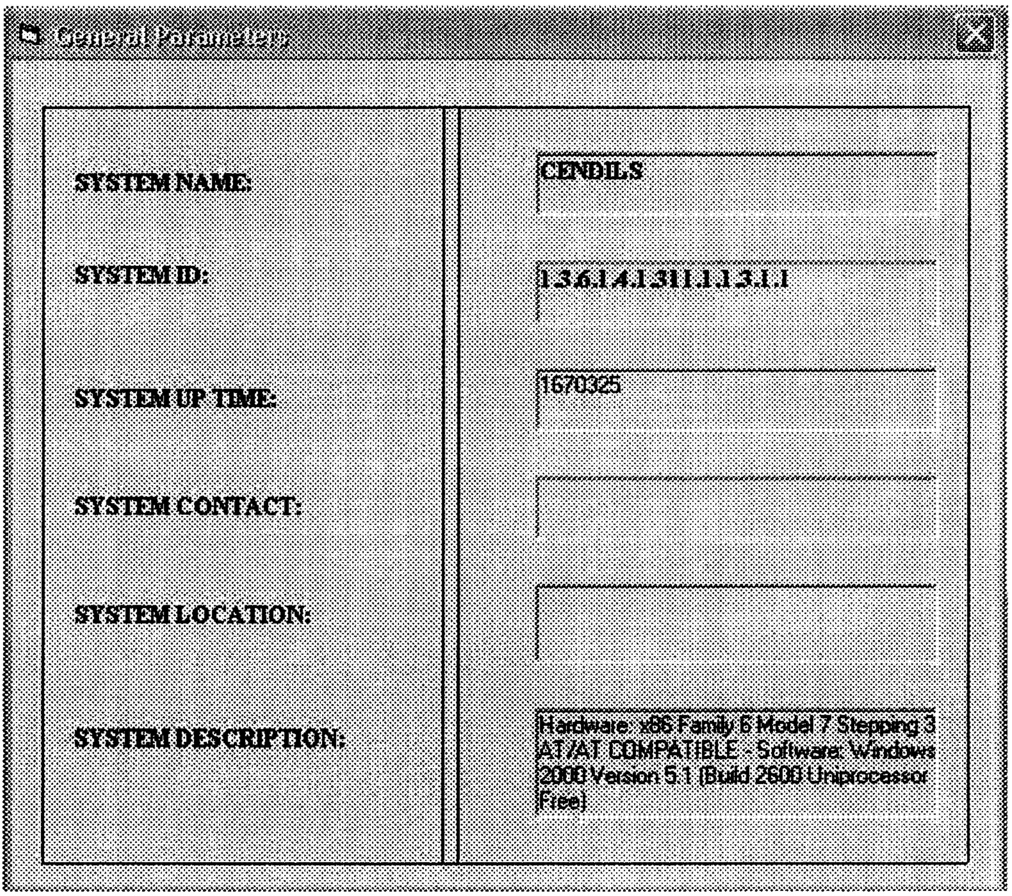
FIELDS	TYPE	DESCRIPTION
HOSTID	VARIABLE	SPECIFIES THE ORACLE AGENT BEING MONITORED
DATE	DATE	SPECIFIES THE DATE AT WHICH AGENT WAS MONITORED
HOUR	INTEGER	HOUR VALUE EXTRACTED FROM THE TIME$
MINUTE	INTEGER	MINUTE VALUE EXTRACTED FROM THE TIME$
SNMP VARIABLE	LONG INTEGER	CONTAINS VALUE OF THE SNMP VARIABLE BEING MONITORED

SCREEN FORMAT

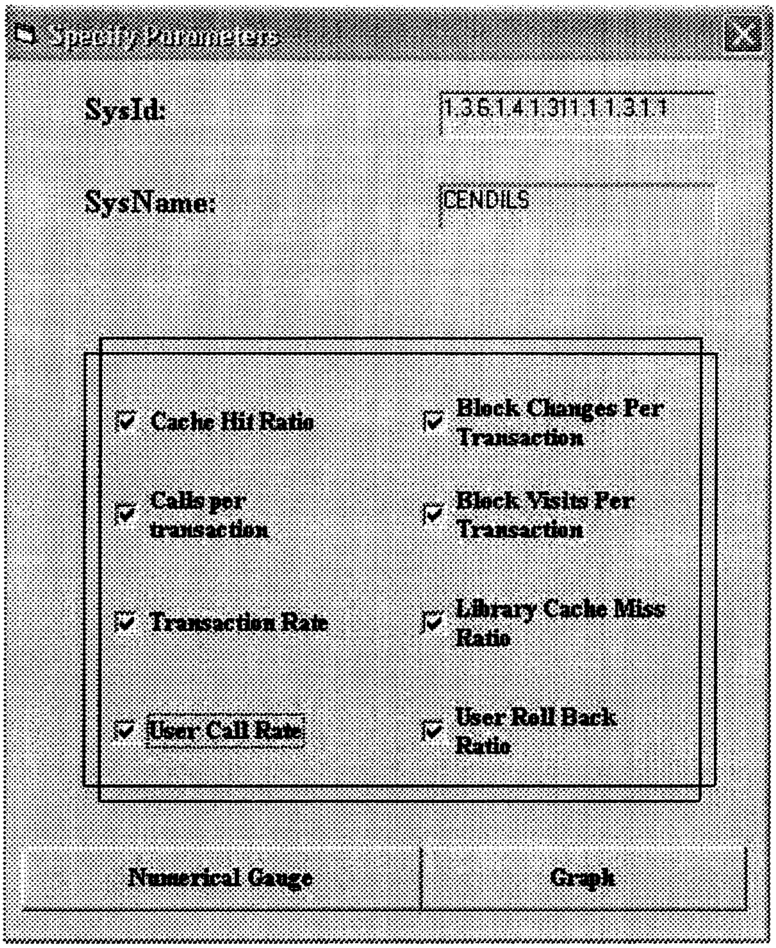
First Page



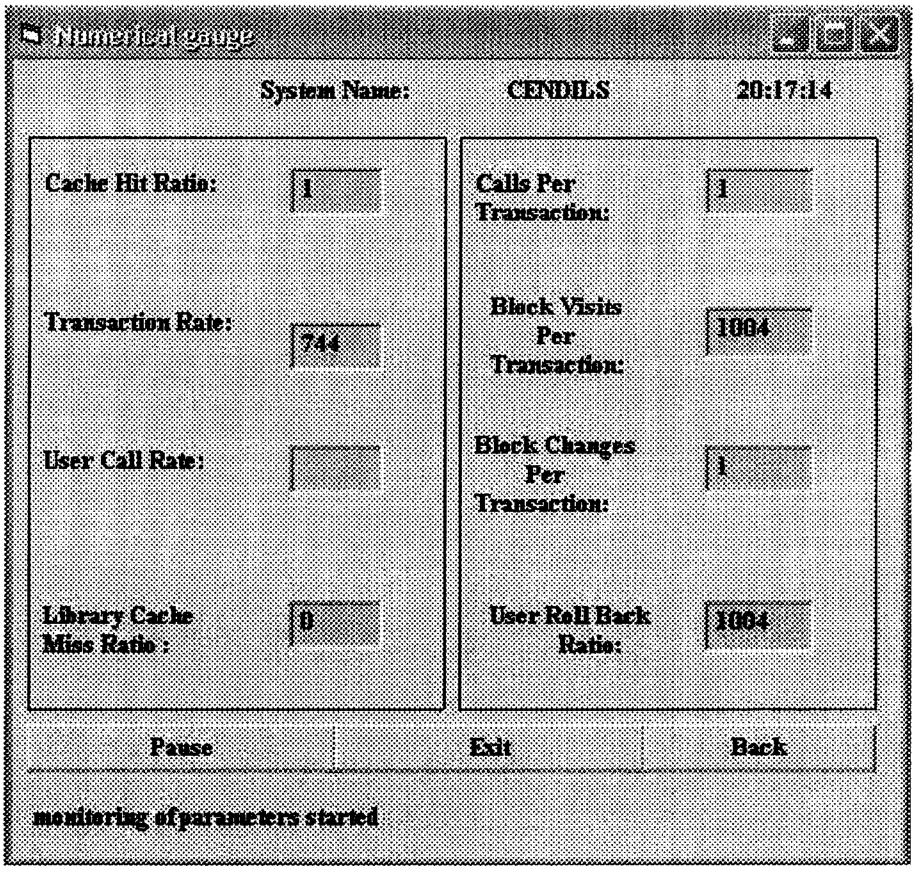
General Parameters



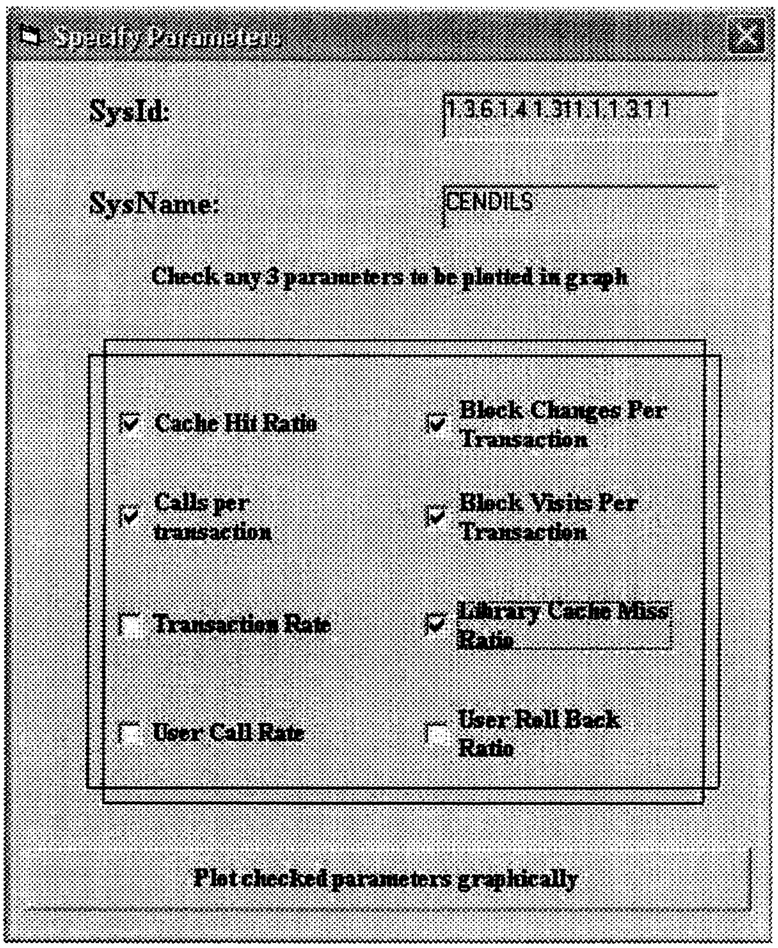
Specify Parameters



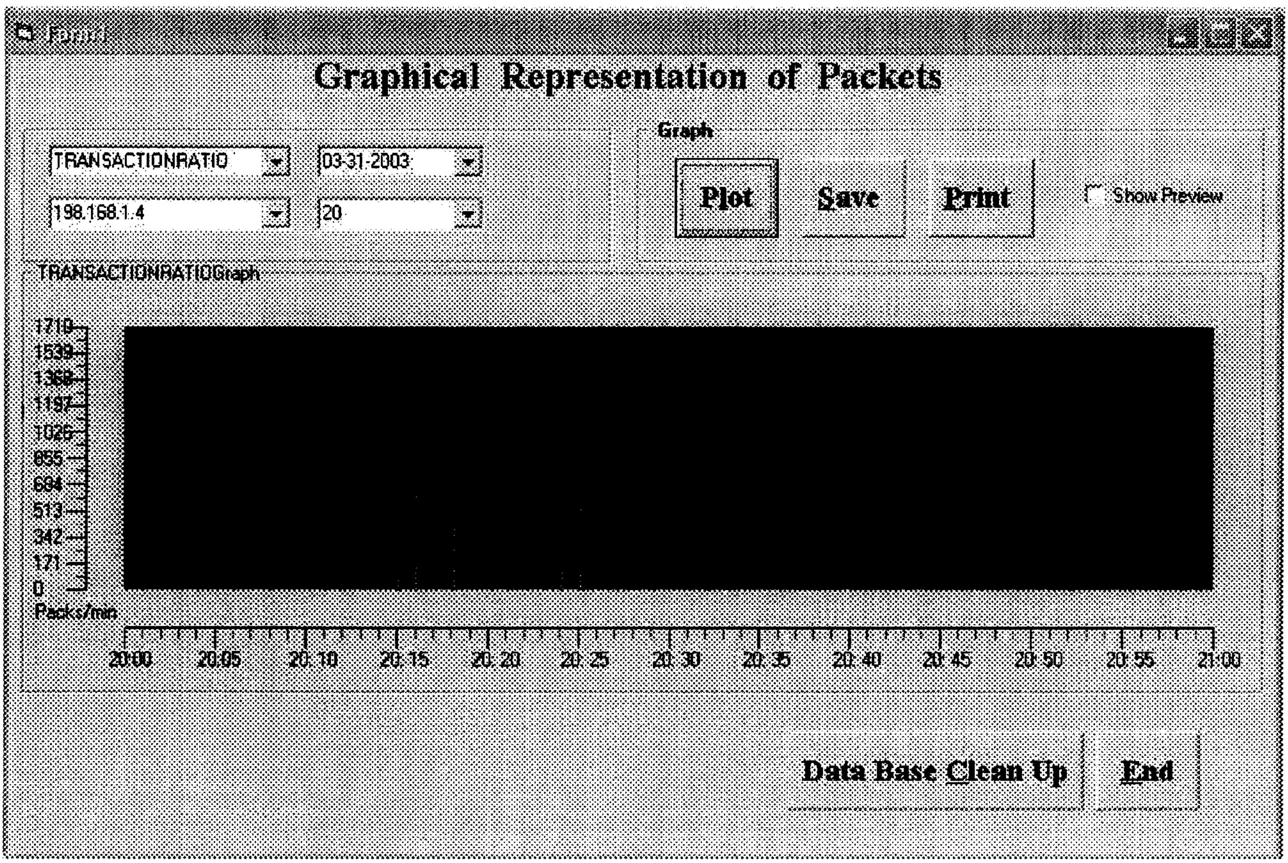
Numerical Display



Specify Parameters



Graphical Display



BIBLIOGRAPHY

1. Sean J. Harnedy,"Total SNMP: Exploring the Simple Network Management Protocol", Second Edition, Prentice Hall, 1997.

2. William Stallings, "SNMP, SNMPv2, and RMON: The Practical Network Management", Second edition, Addison-Wesley, 1996.

3. RFC 1442, Structure of Management Information for version 2 of the Simple Network Management Protocol.

4. RFC 1565, The Network Services Monitoring MIB.

5. Visited www.oradoc.com/snmp/main.html

6. Visited www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm