An Information Technology Management Reference Architecture Implementation

Designed for the Sun Internet Data Center Mail and Messaging Reference Architecture

By Edward Wustenhoff

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 USA
650 960-1300

Part No. 816-6725-10
Revision A

An Information Technology Management Reference Architecture Implementation

This article is the fifth in a series of articles by Edward Wustenhoff on the data center management infrastructure. The focus of this article is on the implementation of the management infrastructure.

It is a follow-up article on the “An information Technology Management Reference Architecture” article published earlier by Edward Wustenhoff and the Sun BluePrints™ group. It describes the implementation of IT management reference architecture in the Authorized iForce℠ Ready Center (iFRC) program that displays an IDC Mail and Messaging Architecture. The iFRC program is a Sun program that provides reference implementations and proof of concepts to assist our customers in avoiding common pitfalls.

This article describes the technical aspects and details of the management and organization (M&O) architecture deployment. Conceptually, the IDC Mail and Messaging Reference Architecture implementation consists of two major components. The first component of the implementation is the Message Server (Sun™ ONE) that represents the mission-critical business application. The second major component of this implementation is the systems management tools that are implemented to monitor, maintain, and report on availability and SLA adherence of the business application. This is the M&O architecture.

This article describes the implementation of these tools and their interactions, and discusses how they are combined into a systems management and Service Level Agreement (SLA) reporting solution. This article provides topological views of the management implementation, individual tool implementations, combined tools implementations, and details of the essential integration points between the tools.

Limitations

This article does not describe the people and process aspects of IT management. Further, this article is not intended to be a detailed implementation reference. It is to be viewed as a proof of concept, showing the structure and essential functionality to be addressed in an M&O architecture. Also, the M&O architecture is not designed with a focus on availability and performance.

System Topology

FIGURE 1 shows the implementation of the M&O architecture. Note how the hierarchy matches the management architecture that was discussed in "Concepts, IT Management Framework" in the previous article. Also, note that the actual implementation does not include anything outside the box that indicates the IT infrastructure management framework.

FIGURE 2 presents the basic topology of the actual system environment. This diagram is organized so that there is a distinction between the managed systems and the systems that are to perform the systems management function.

Note that the systems pictures *do not* represent the actual system types that were deployed. All of the management systems were Netra™ X1 servers.

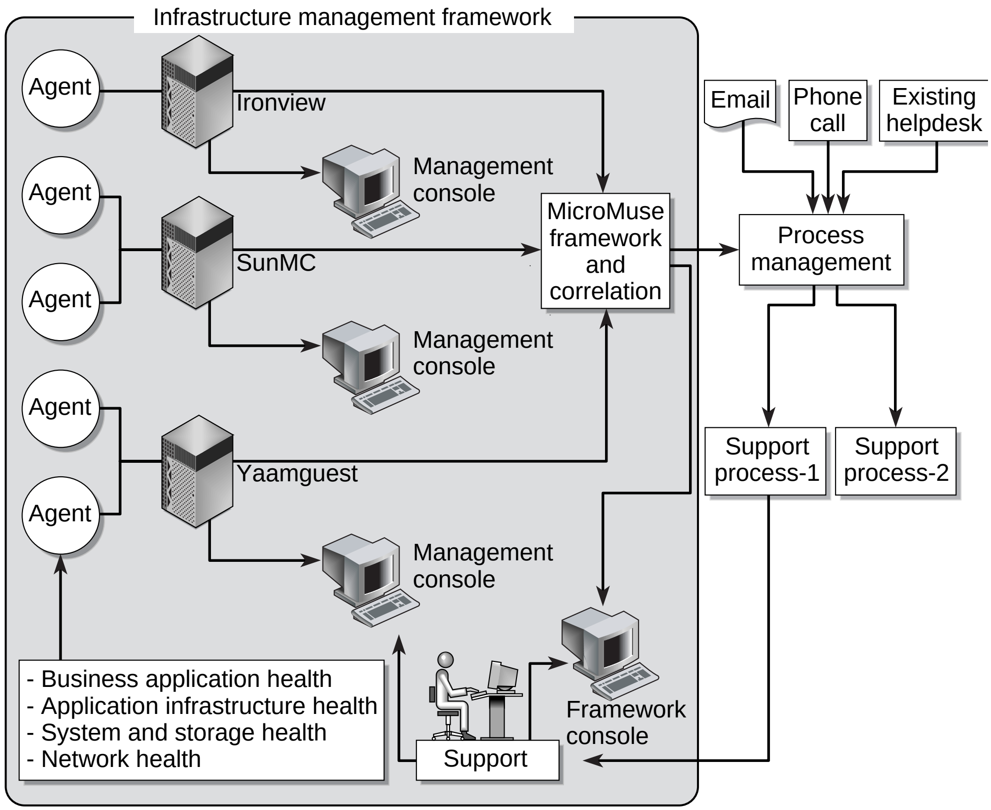

FIGURE 1 IT Management Architecture

Notice in FIGURE 2 that individual systems are implemented to support the server components of the management tools being implemented. This particular configuration was chosen primarily for logistical reasons and would not likely be practical in a production environment. The exact layout of an implementation is determined by many factors including size of machine, speed of network connections, port conflicts between products, and limiting the impact of outages.

With this basic system configuration established, the following sections detail the implementation of the systems management tool set used in the iFRC program and the manner in which they are interfaced to each other. The final section in this article describes the combined tools implementation in order to present a comprehensive view of the environment.

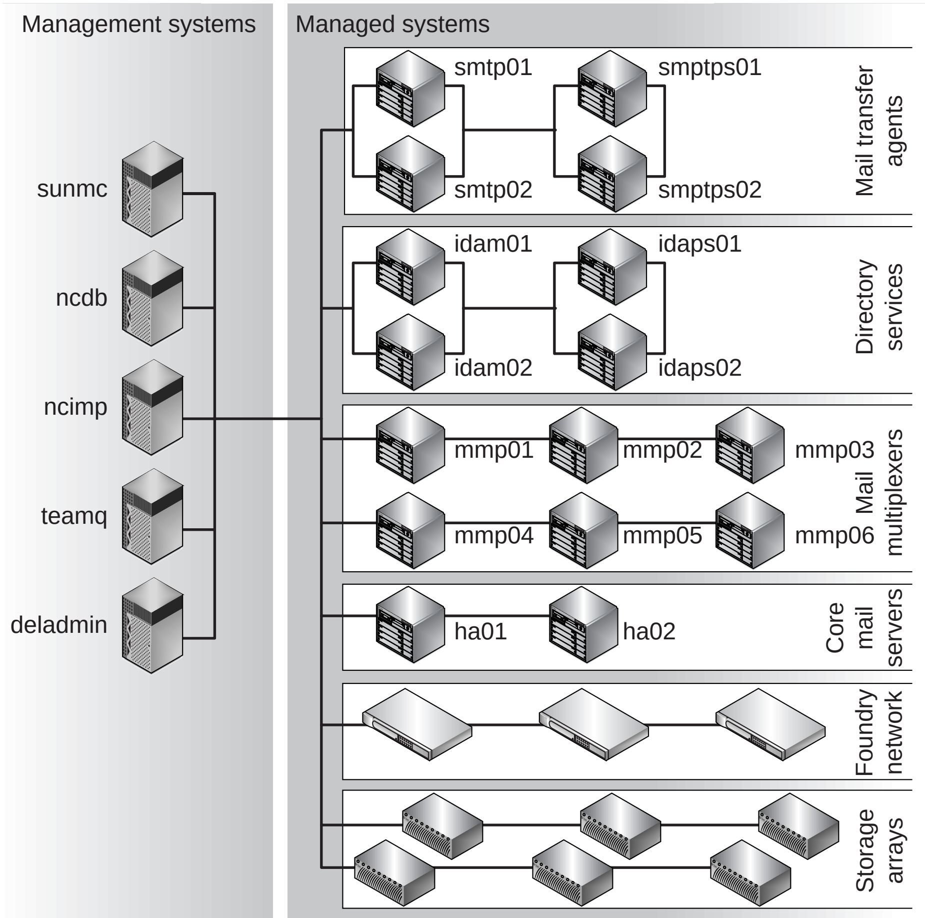

FIGURE 2 iFRC Management Configuration of the IDC Mail and Messaging Reference Architecture Deployment

Tools Selection

Referring to the requirements described in the previous article, the following products were deployed to facilitate the different views.

- **MicroMuse Netcool** provides the NOC view that shows all alerts in all components. These components have their own views:

- **MicroMuse ISM** provides the SLA view that reports the status of the messaging application against a predefined .
- **MicroMuse Netcool** and **ISM** provides the applications infrastructure SLA view that reports on the status of the software components that support the application. Lightweight Directory Access Protocol (LDAP), Network Time Protocol (NTP) and domain name service (DNS) are included in this.

- **Sun™ Management Center (SunMC**) software is the core of the system administration view that reports on the status and supports the administration of all Sun computer and storage systems.
- **TeamQuest** is responsible for the performance view that monitors the status of the system and networks as far as they relate to being able to handle the processing load.
- **Foundry's Ironview** with **MicroMuse's Netcool** provide the network view that reports on network alerts and supports the administration of the network components.

Using the same organization as described in "Implementation Requirements, Management at All Layers" of the previous article, TABLE 1 summarizes the managed aspects per tool by layer.

TABLE 1 Tools Distribution by Layer

Layer	iForce implementation	Fault	Config	Accounting	Performance	Security
Business application	Message Server 5.1 (Sun One)	SunMC / Micromuse	TBD	TBD	TeamQuest Micromuse	TBD
	MMP	SunMC / Micromuse	TBD	TBD	TeamQuest Micromuse	TBD
	MTA	SunMC / Micro-Muse	TBD	TBD	TeamQuest Micromuse	TBD
Application infrastructure	Directory Server 4.13 (Sun™ One)	Micromuse	TBD	TBD	TeamQuest Micromuse	TBD
	DNS	Micromuse	TBD	TBD	TeamQuest Micromuse	TBD
	Firewall	Micromuse	TBD	TBD	TeamQuest	TBD
	NTP	Micromuse	TBD	TBD	TeamQuest Micromuse	TBD
Computing and storage platform	Netra™ T1 server	SunMC	SunMC	TBD	TeamQuest	TBD
	Netra 1405 server	SunMC	SunMC	TBD	TeamQuest	TBD
	Sun Fire™ 6800 server	SunMC	SunMC	TBD	TeamQuest	TBD

TABLE 1 Tools Distribution by Layer

Layer	iForce implementation	Fault	Config	Accounting	Performance	Security
	Netra X1 server	SunMC	SunMC	TBD	TeamQuest	TBD
	Sun StorEdge™ T3 array	SunMC	SunMC	TBD	TeamQuest	TBD
	Sun Enterprise™ A1000 hardware	SunMC	N/A	TBD	TeamQuest	TBD
	Sun SAN switches	SunMC	TBD	TBD	TeamQuest	TBD
	Sun™ Cluster 3.0 software	SunMC	SunMC	TBD	TeamQuest	TBD
	Solaris™ Operating Environment (Solaris OE)	SunMC	SunMC	TBD	TeamQuest	TBD
Network infrastructure	Foundry NetIron	Foundry	Foundry	TBD	TeamQuest	TBD
	Foundry ServerIron	Foundry	Foundry	TBD	TeamQuest	TBD
	Foundry BigIron	Foundry	Foundry	TBD	TeamQuest	TBD
Facilities infrastructure	iForce Lab	N/A	N/A	N/A	N/A	N/A

Tools Implementation Details

The following sections describe the details of the tools implemetation.

IronView

The IronView product set monitors and manages the networking equipment from Foundry Networks. IronView provides a console for direct access to the network topology. Its main function is the administration of the network topology. It was implemented on the `deladmin` server.

The Foundry Network devices issue SNMP traps directly to Netcool for the Network Operations Center (NOC) and SLA processing. In this implementation, IronView is installed co-resident with the Message Server (Sun ONE) delegated administrator services because the IronView processing requirements are very low.

Sun Management Center 3.0

The SunMC 3.0 software is an element manager that details and monitors the operation of Sun servers, peripherals, and applications. In the context of the iFRC program, SunMC is used primarily to provide low-level monitoring of the application servers to report on hardware and operating system stresses and faults.

Loren Pearce from SunPS collected and described the following details on Sun Management Center 3.0.

SunMC 3.0 has a three-tier architecture. The first tier includes the “agents” that collect information and alerts from the individual systems being monitored. The second tier is a server component that collects information from the monitoring agents and reports this information to the clients. This tier also provides automation capabilities.

The third tier is the client tier that is usually provided by the server, console, and an agent component all executing on one system. Additional agents are deployed to the application servers for monitoring of these systems. The console is viewed using Sun Ray™ appliance displays. While this is not unusual, it is not necessary to combine the three components of SunMC 3.0 on one system.

In addition to monitoring and reporting on the application servers, SunMC 3.0 agents are deployed on the systems management servers in order to monitor their operation and alert the systems administrator to any problems within the monitoring complex.

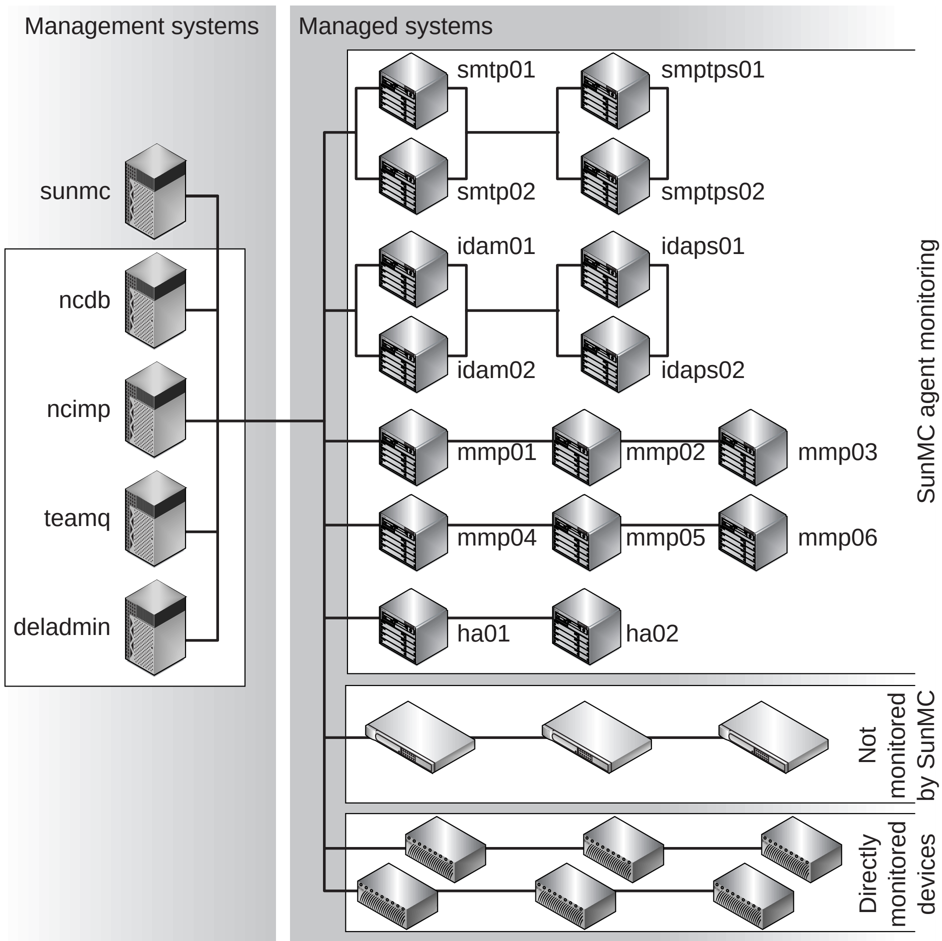

FIGURE 3 SunMC 3.0 Configuration

TABLE 2 details how the various components of SunMC 3.0 are distributed throughout the system topology.

TABLE 2 SunMC 3.0 Component Distribution

Component	System	Port(s)	Notes
SunMC server	`sunmc`	162	163
164	165	166	167
168	2099		SunMC console
`sunmc`	Variable		SunMC agent

TABLE 2 SunMC 3.0 Component Distribution

Component	System	Port(s)	Notes
sunmc	161	SunMC agent replaces standard snmp daemon	
ncdb	161		
ncimp	161		
teamq	161		
smtpr01	161		
smtpr02	161		
smtps01	161		
smtps02	161		
ha01	161		
ha02	161		
ldaps01	161		
ldaps02	161		
ldapm01	161		
ldapm02	161		
mmp01–mmp06	161		

The Netcool SunMC probe collects alerts for reporting to the NOC and the SLA manager. This probe acts as a client to SunMC to capture the system level events and forwards them to the Netcool database.

SunMC also provides investigation and alert management through its console. The console is made available as part of the system administration view so that a system operator can obtain additional details pertaining to outages detected in the environment. With this information, the system administrator can determine the correct course of action for resolving the operational impact.

TeamQuest

TeamQuest performance software monitors the performance metrics of the environment against the best practices as defined in the Sun Professional Services[SM] (SunPS) program performance tuning and capacity planning methodology, outlined in the “Implementation Requirements” section of the previous article. When performance target violations are detected, alerts are sent to Netcool in the form of SNMP traps that are then processed and forwarded to the NOC and SLA views.

TeamQuest also provides capacity planning and performance tuning capabilities.

This section describes the details of how the iFRC implemented certain aspects of performance management. It was provided by Scott Johnson, Senior Engineer and Joe Rich, Manager Corporate Technical Relations, from TeamQuest Corporation, who provided the technical expertise and tools.

TeamQuest View and TeamQuest Model were the two products used in the iFRC to provide a mechanism to capture and report on the performance and capacity of the systems in the architecture.

TeamQuest View is a comprehensive, analytical tool that reports system performance and helps locate bottlenecks. Performance data is collected from several different sources on the system and stored into a local database for real-time or historical analysis.

TeamQuest Model provides predictive analysis required for long-term capacity planning. This tool, using data retrieved from the architecture systems, was used to build models of the systems and applications. These models were validated against the benchmark data and were used to experiment with “what-if” questions relating to the number of users and the effects of equipment changes.

TeamQuest View

TeamQuest View consists of a server-based component (framework) that is responsible for (among other things) performance data collection and storage and threshold violation notification. TeamQuest View also has a Motif-based client GUI (TQView) which, when connected over TCP/IP to the server component, produces graphical displays of both real-time and historical performance data. In the iFRC, the client GUI with associated reports is located on a separate server with automatic login capability from any one of several management stations connected to a SunRay server. Upon login, TQView automatically establishes a connection to the desired architecture server and produces a series of predefined performance reports (operational views). The reports consist of a number of statistics that were selected from baseline performance monitoring metrics as defined in the SunPS performance tuning and capacity planning methodology. FIGURE 4 is one example, showing important CPU metrics, the top 10 busiest disks, the number of RPC client calls per second and the buffer read and write cache hit percentages. In addition to these reports, TeamQuest provides a large number of pre-defined reports and also supports ad hoc report generation.

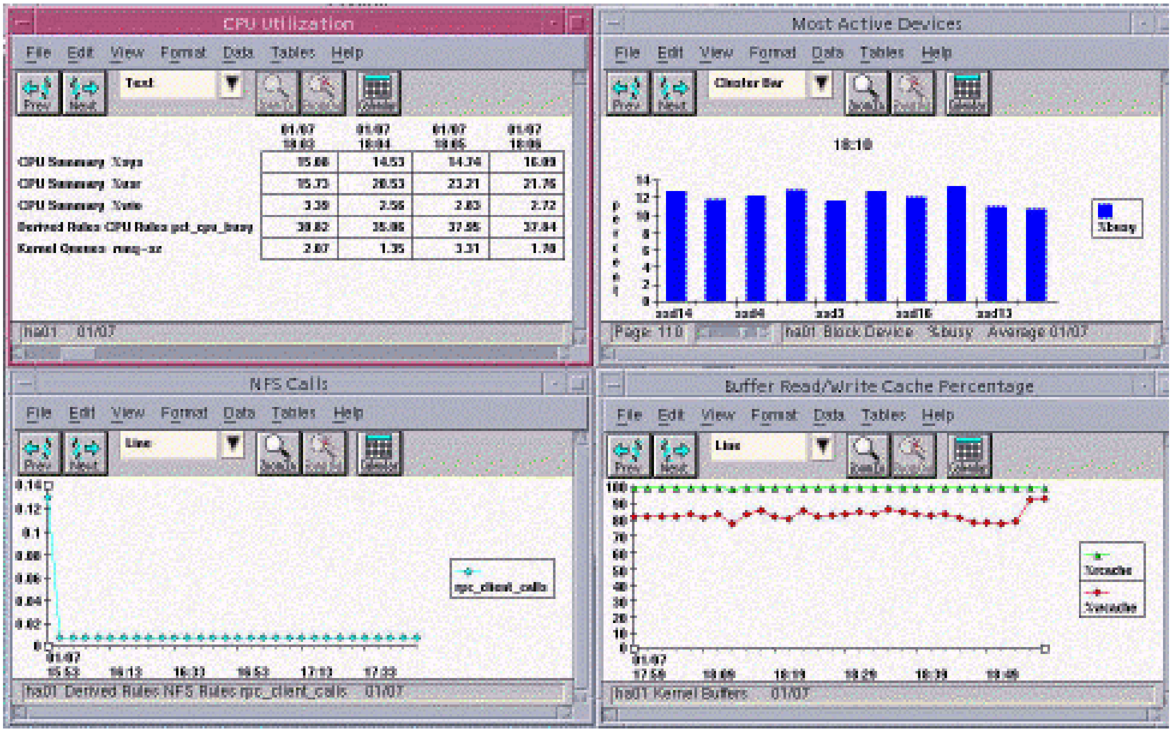

FIGURE 4 Operational View Example of the Message Server (Sun ONE) from TeamQuest View

Threshold Violation Detection and Notification

The TeamQuest Framework component samples data at a preset interval and stores that data into its performance database. default sampling rate of one minute is implemented. As the data is collected, TeamQuest automatically checks the values of the statistics against any defined threshold levels. A set of thresholds, based on baseline performance monitoring metrics as defined in the SunPS performance tuning and capacity planning methodology, was established for all of the servers in the architecture.

When a threshold criterion is met, an alert is issued. The default action is to log the event into the performance database in the alarm log. In addition, the iFRC team chose to send all alerts through an SNMP trap to the MicroMuse Netcool Omnibus console. MicroMuse assisted in the integration of the TeamQuest View client as a tool under Netcool so that it could be launched in context to the server that issued the alert. In this way, a report showing a graph of the statistic in violation can automatically be generated. FIGURE 5 and FIGURE 6 show drilling down from an alert event sent to Netcool to the statistic in violation on the server under stress.

FIGURE 5 MicroMuse/TeamQuest View Integration

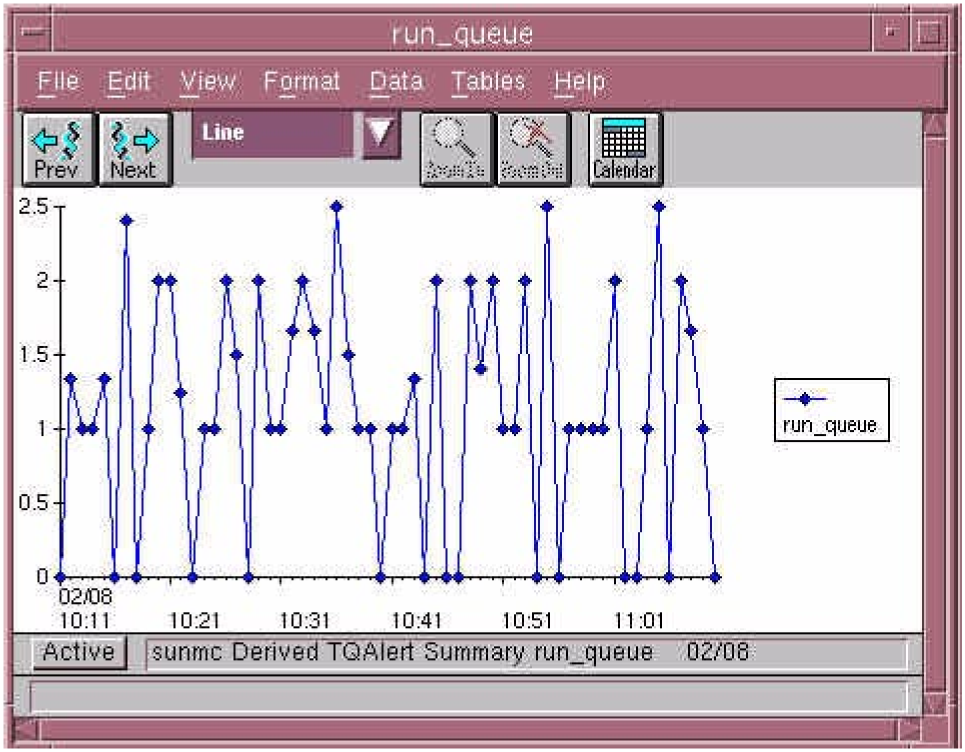

FIGURE 6 TeamQuest View Report as a Result of the “Start in Context” by MicroMuse

Application Level Resource Reporting

When monitoring, tuning, and predicting capacity of systems, identifying the critical applications (or workloads) of the system is imperative. SLAs are usually created for these critical applications, and thus dictate which workloads must be monitored.

TeamQuest provides a mechanism, through identification of the processes that comprise an application, to track the resource consumption of that group of processes. Through a definitional language, the user builds process groups that are called workloads.

For the IDC Mail and Messaging Architecture, a common set of workload definitions is created and loaded into the TeamQuest performance database on each system.

Note – To enhance the data collection, Solaris OE process accounting was enabled on all of the systems. Thus when a sample was taken, both the running process activity and the completed process activity was merged to provide the best capture ratio.

Note – TeamQuest reduction processing was deactivated so that each and every process captured was recorded into the performance database.

As TeamQuest process-level data was collected, it was compared to the workload definitions and deposited into the appropriate workload container. This allowed the iFRC team to do several very important functions:

- Monitor (and alert on) the resource consumption by workload.
- Provide the basic operational entities for the capacity planning exercises described later. In addition to the prestored workload definitions, TeamQuest has the capability to reprocess the raw process data after the fact into workloads, thus allowing for "tuning" and further refinement of the workloads. TABLE 3 lists the workloads and their definitions for the MAIL servers.

TABLE 3 Sample and Workload Definitions

Workload name	Description	Definition
Management Services	Processes associated with TeamQuest Performance software or SunMC monitoring agents	command = `/tq.*/` or command = esd
Cluster	Processes associated with Sun Cluster	fullcmd = `/.*cluster.*/`
Stored	Message store maintenance program (deadlock checks, message deletion, and so forth)	command = `stored` and login = `mailsrv`
Post Office Protocol (POP)	Processes that handle POP3 messages to/from MMP servers	command = `popd` and login = `mailsrv`
(Internet Mail Access Protocol (IMAP)	Processes that handle IMAP4 messages to/from MMP servers	command = `imapd` and login = `mailsrv`
Simple Mail Transfer Protocol (SMTP)	Processes that handle the MTA queue for (SMTP) messages	login = `mailsrv` and ((command = `tcp_smtp_server`) or (command = `ims_master`))
Other Message Server (Sun ONE)	Any other processes associated with the Message Server (Sun ONE) application	login = `mailsrv`
Other	Any processes not explicitly included in the preceding workloads	

FIGURE 7 is a screen display of one of the mail servers showing the overall CPU utilization along with the same utilization broken out by course application workloads and finally by a finer granularity set of workloads. It is easy to see how much of the resource each application is consuming and, within the Message Server 5.1 (Sun ONE) application, which specific mail protocol is using the most resources (POP).

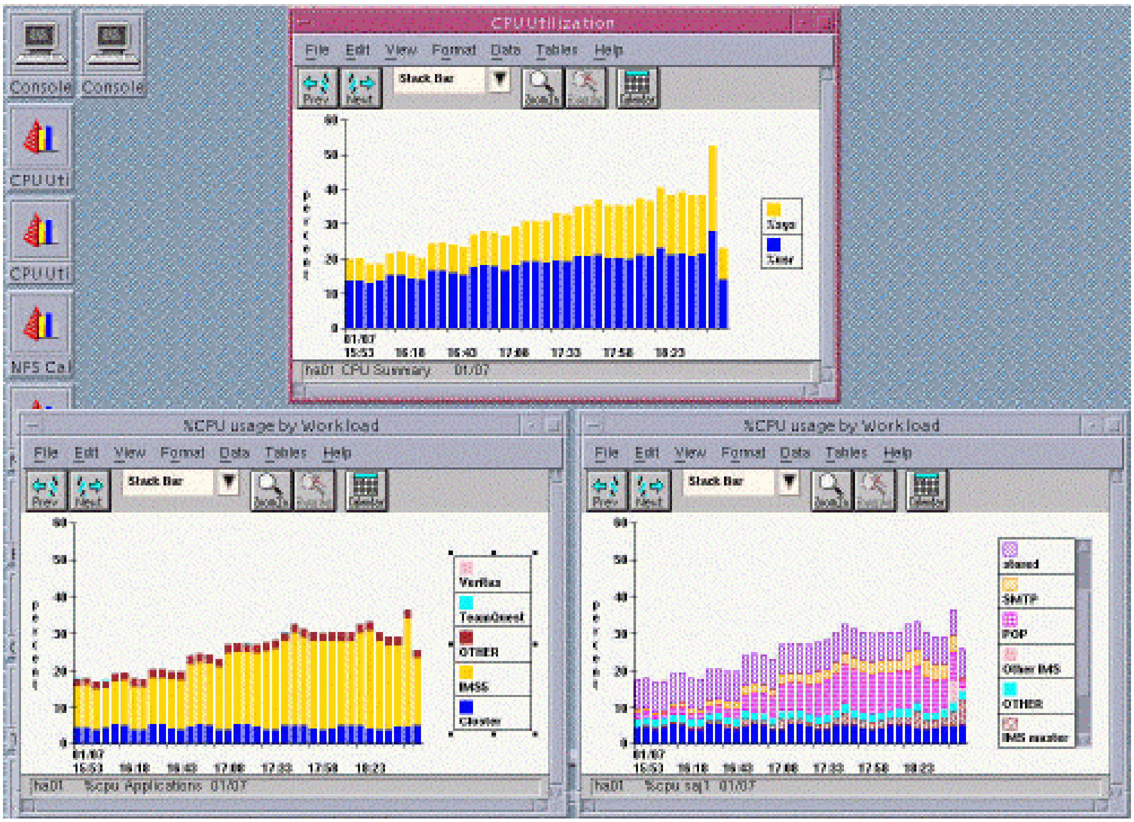

FIGURE 7 Overall CPU Utilization By Workload

Capacity Planning

As mentioned earlier, TeamQuest Model is the capacity planning tool that was used in this study. TeamQuest Model applies operational analysis principles to measured computer system performance data to build baseline performance models. You can then modify the baseline models to predict system performance when configurations and/or workloads change.

The models are known as *queuing network models* (QNMs). A QNM consists of a set of resources (CPU, disks, terminals, and so forth) that can be visited by customers from one or more workloads. Workloads can have unique intensities (arrival rates or populations), priorities, and average service demands (visits and service times per visit) at resources. Solving a QNM produces estimates of throughputs, response times, queue lengths, and utilizations for each workload at each resource, as well as estimates of overall workload throughputs and response times.

Results are presented in spreadsheet like displays and charts. For example, components of response time charts allow performance analysts to quickly assess impacts of queuing delays at resources on workload response times.

TeamQuest Model provides two methods to solve QNMs—approximate mean value analysis (MVA) and simulation. Approximate MVA is normally used, since it is much faster than simulation. Approximate MVA iterates solving for response times, throughputs and queue lengths until convergence occurs for all workloads at all resources. Approximate MVA is the method that was used to model the messaging IDC reference architecture systems.

Baseline models of several representative benchmarks were created, including an enterprise model, built by including a representative system from each server class in the messaging system.

Since actual transaction completion rates were not available for the user activity, a synthetic transaction rate was constructed based on the email sender test profile and the number of users. FIGURE 8 shows the throughput of the two major workloads for the enterprise. The throughput rate is the transactions per second based on the synthetic transaction definition discussed previously.

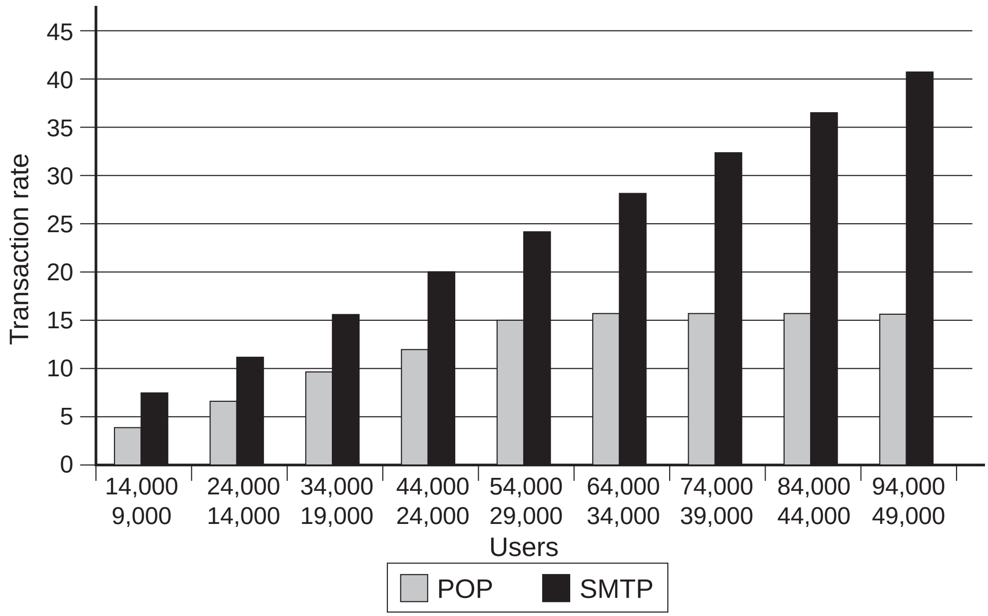

FIGURE 8 Major Workload Throughput

The SMTP workload throughput increases as the number of email senders increases. However, the throughput for the POP workload increases steadily only to the 54000/29000 user level and reaches a plateau by the 64000/34000 user level. This indicates that something is constraining the activity of the POP workload.

Further analysis indicates that the constraint occurred in the MMP server. FIGURE 9 shows that the CPU for mmp01 is about 95 percent utilized and is most likely the constraining factor.

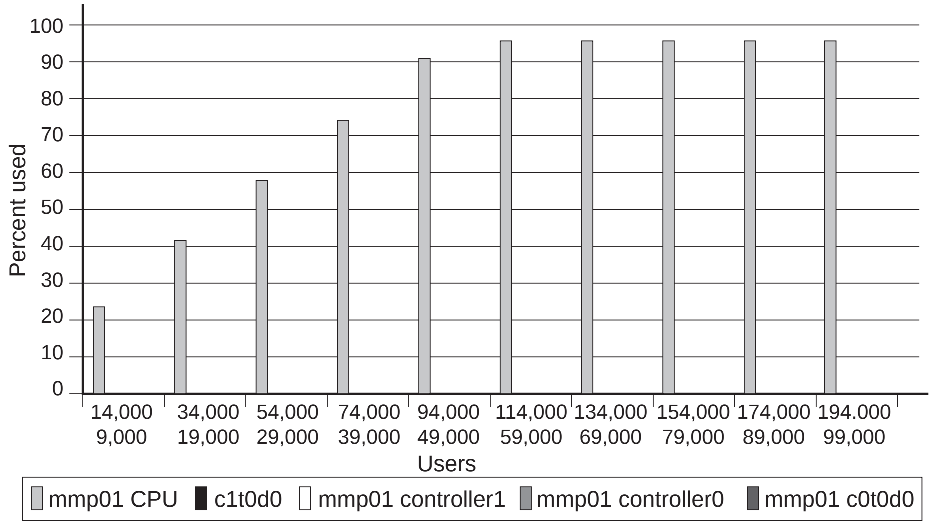

FIGURE 9 Five Most Active Resources for System `mmp01`

In the benchmark, the configuration consisted of six MMP servers, which would appear to be able to support about 8000 email readers. The models suggest that to support the 94,000 email reader level would require 12 Netra T1 systems.

Since TeamQuest Model allows you to play "what-if" scenarios both on population growth and on configuration changes, you can see the results of the same workload run on a two-CPU or a four-CPU Netra t1405. (The relative performance characteristics of the CPUs were provided by SunPS).

With the MMP servers upgraded to a two-CPU Netra t1405, the enterprise is able to adequately process the work. FIGURE 10 shows the throughput for both the POP and SMTP workloads steadily increasing as the number of users increases.

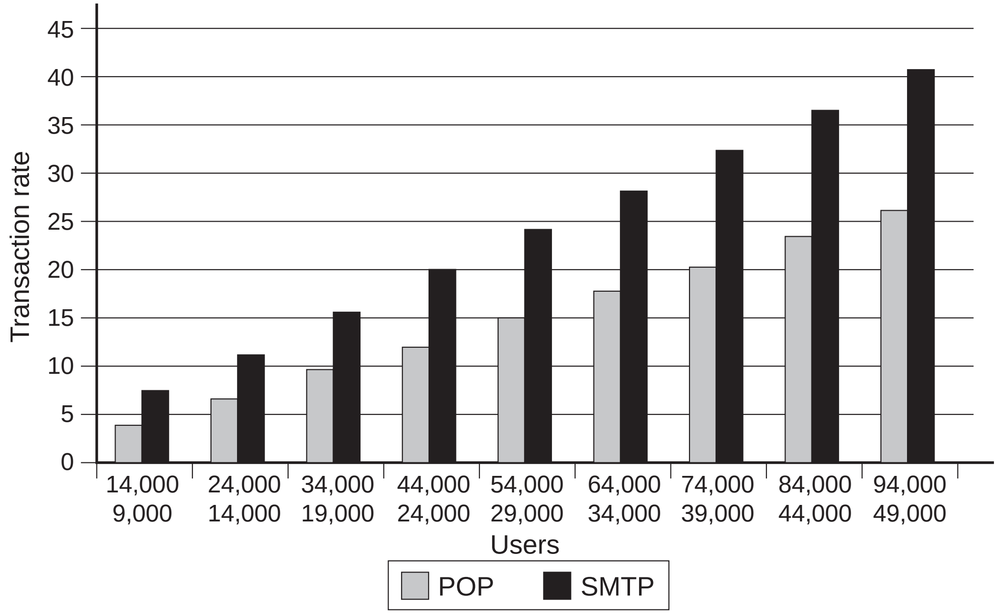

FIGURE 10 POP and SMTP Throughputs on MMP Servers as Two-CPU Netra t1405 Servers

The active resource utilization for each system indicates that there are not any bottlenecks in any of the systems (MMP – 8 percent, messaging – 39 percent, MTA – 18 percent). However, if we continue our modeling exercise, and double the number of active users, the MMP server becomes saturated. Then, increasing the hardware on the MMP server is necessary. By modeling a number of "what-if" scenarios, we arrive at a four-CPU configuration that adequately supports the doubling of the population. FIGURE 11, FIGURE 12, and FIGURE 13 show the active resource utilization for each system type, supporting up to a 194000/99000 user level.

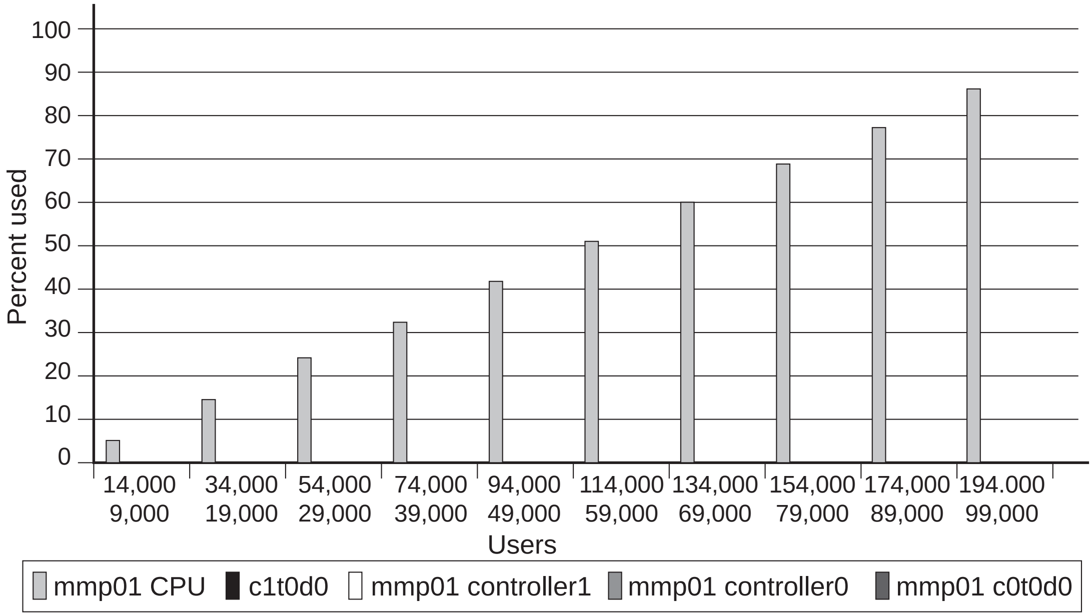

FIGURE 11 `mmp01` Active Resource Utilization (4-CPU Netra t1405 Server)

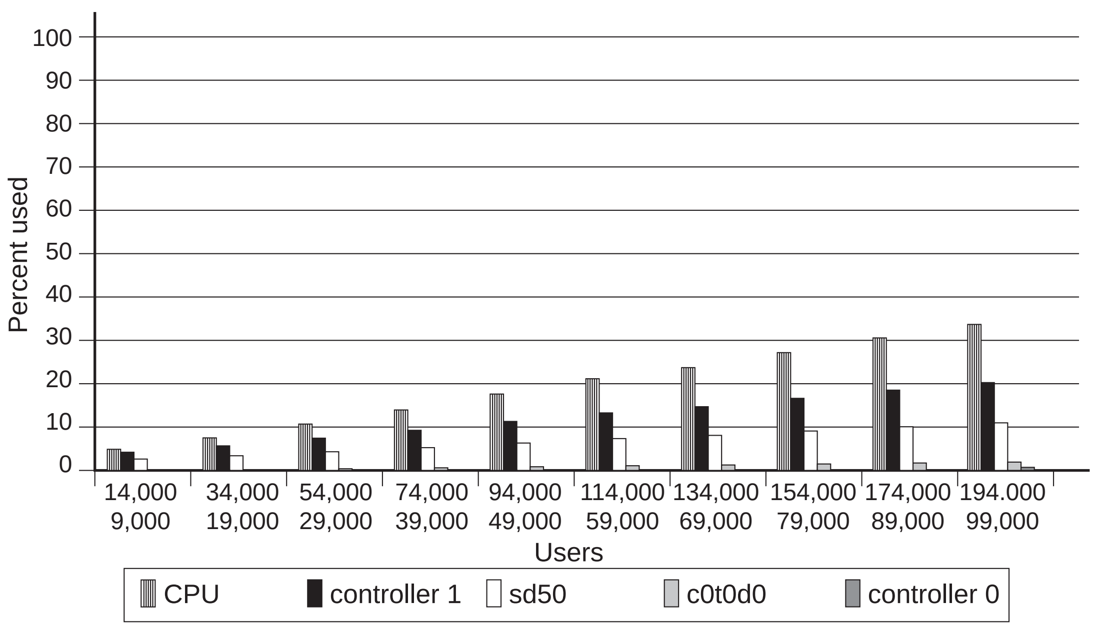

FIGURE 12 `smtp01` Active Resource Utilization (4-CPU Netra t1405 Server)

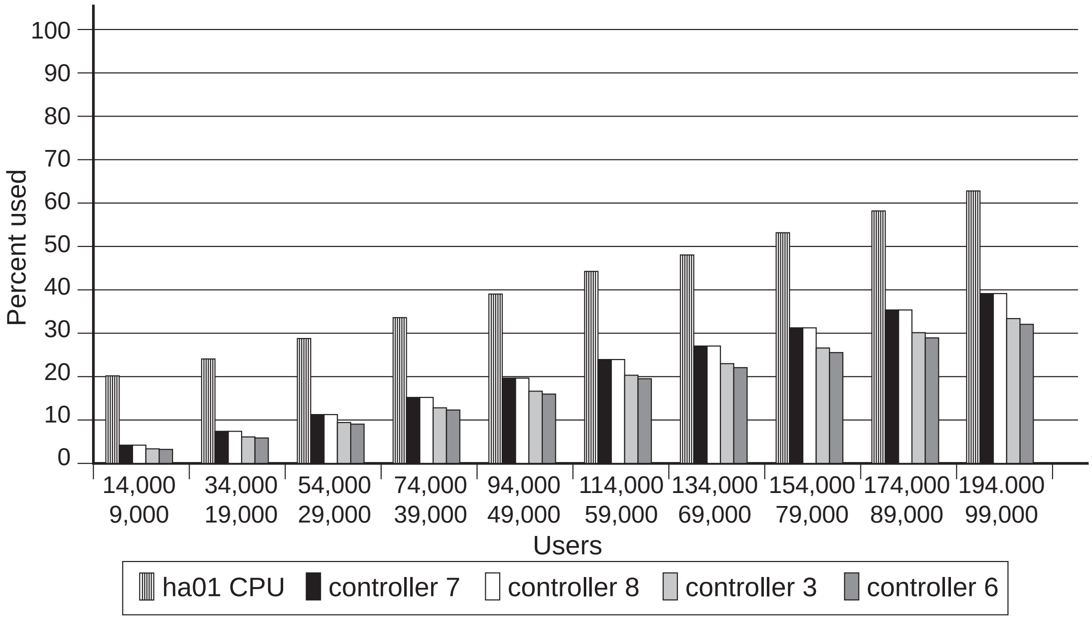

FIGURE 13 `ha01` Active Resource Utilization (12-CPU Sun Fire 6800 Server)

The `ha01` system, being one of the message store systems, is handling the mail demands of the increased user population easily.

Through the TeamQuest Performance software, the iFRC team could provide accurate server sizing information based on the various end-user usage requirements.

Micromuse

The Micromuse Netcool product suite is implemented to collect events and information from the other system management tools, consolidate this information, assess the impact on the environment, and produce service level reports and information against the application set. This is accomplished using the following Micromuse products

- **Object Server:** The Netcool Object Server is an in-memory database that provides event collection, filter design, and view services. The Object Server provides the foundation upon which the other products in the Netcool suite operate. In the iFRC configuration, redundant object servers are installed in order to provide fault tolerance to the collection system.

- **Internet Service Monitors:** The Micromuse Internet Service Monitors (ISM) allow monitoring of 17 different protocols and constructing a simulated transaction. They can be set up in a distributed master/slave relationship, and do not require agents on the remote host that provides the service being monitored. Using the ISMs, you can monitor the Message Server 5.1 (Sun ONE) functionality (LDAP, SMTP, POP, and IMAP) and report their availability for SLA monitoring purposes.
- **Impact:** The Micromuse Impact product allows super-imposition of business logic onto the data that has been captured by probes and monitors and is in the Micromuse Object Server. Impact also allows sophisticated event correlation over time, execution of tests to verify events, event consolidation and event enrichment to name a few possibilities.
- **Visionary:** The Micromuse Visionary product is an analysis tool that utilizes weighted measurements of various SNMP collected values to assert certain fault conditions exist. It is an attempt to be an engineer in a box. Visionary uses weighted MIB expressions to define the state of something measurable via SNMP (BGP health, Host status, and so forth). You can create your own MIB expressions to be used in the polling and analysis.
- **WebTop:** The Micromuse WebTop product is a two-directional Java™ interface into the Netcool Object Server Event List. WebTop can be configured to provide a one-directional (read-only) view to events pertinent to a specific customer.

Netcool Topology

As previously mentioned, the Netcool Object server has been installed on two servers in the IDC Mail and Messaging Reference Architecture implementation in order to provide some level of redundancy to the Netcool system. These Object servers are connected to each other using a bidirectional gateway. With this gateway, a Virtual Object Server is created with the two “real” Object Servers as the participants. Clients need only reference the name of the Virtual Object Server. Object Server operations are performed by one of the two real Object Servers and then synchronized with the other. This scheme provides fault tolerance for the object server.

Monitoring of the Message server 5.1 (Sun ONE) application environment is accomplished using probes provided with the Micromuse suite of products. These probes collect information from the management tools, normalize the information into a standard format, and then inject the resulting record into the database. Probes installed in this environment are:

Syslog: The `syslog` probe monitors the `syslog` output of the servers and reports potential problems and failures to the SLA manager and the NOC.

mttrapd: The `mmtrapd` facility listens for and handles traps. In this implementation, this facility is receiving traps from TeamQuest and normalizing the information. From these traps, the following information is collected:

- System name that generated alarm
- User text added to alarm
- TeamQuest severity
- Sampled value
- Normal/Previous/Current threshold values
- Timestamp of alarm occurrence (from TeamQuest)
- User text description line 1
- User text description line 2
- User text description line 3
- Current TeamQuest database name and hostname where TeamQuest detected the alarm, main alert bucket, subgroup1, subgroup2, subgroup3, application statistics

SunMC: The SunMC probe connects to the SunMC 3.0 server in order to receive SunMC events and passes filtered and normalized information to the Netcool object server. The contents of the specific record passed back to the Netcool object server are dependent upon the type of alert that was generated by SunMC.

The Netcool components are distributed in the iFRC as follows:

TABLE 4 Netcool Component Distribution

Server	Component	Notes
Ncdb	Object Server	Part of virtual object server
	ISM	Monitors Message Server (Sun ONE) applications
	mttrapd	Gets traps from TeamQuest
	syslog probe	
	Process Automation	
	License Server	
Ncimp	Object Server	Part of virtual object server
	syslog probe	
	Bidirectional Gateway	Combines with ncdb to create a virtual object server
	Process Automation	
SunMC	SunMC probe	
	Process Automation	Monitors SunMC probe
	License Server	With SunMC probe license

Netcool Console Services

The Netcool implementation provides a NOC view of the environment that is a consolidation of the alerts and information from the other monitoring components. This is a standard feature of Netcool and is provided on the NOC console for Netcool.

In addition to the NOC view, Netcool Impact generates a Service Level Management (SLM) view of the environment. By monitoring the information being generated by the other monitoring tools and the ISMs, Impact can determine how service levels on the application are affected and generate alarms on the SLM view.

In cases where there are performance impacts to the monitored applications, you have the ability to interrogate live TeamQuest graphs and reports directly from the Netcool SLM view. This is accomplished by creating a customized menu tool on the Netcool display and linking that directly to the TeamQuest agent that sent the alarm.