Toward an Integrated Monitoring Scheme

Tom Roney, National Center for Supercomputing Applications, University of Illinois, Urbana, IL

ABSTRACT: *The National Center for Supercomputing Applications (NCSA) employs a virtual operator (V-Oper) on UNIX operating systems. V-Oper is a monitoring tool developed at NCSA for the detection of potential system and application problems, and displays warning messages for administrative attention. It is interactive to provide automated implementation of counter measures against reported problems. V-Oper and all other tools used to monitor NCSA systems are being collected to run under a single software package, the Computer Associates' Unicenter TNG. This paper will discuss the various means by which NCSA effectively pilots a production environment, and the progression toward an integrated monitoring scheme.*

1 Introduction

The National Center for Supercomputing Applications (NCSA), in its new role as the Leading Edge Site in the National Computational Science Alliance (also NCSA), is funded for the purpose of prototyping a national information infrastructure—a grid that will connect supercomputers across the United States using high-speed networks. This will increase both the accessibility and the capability of supercomputing technologies. The idea is to integrate resources as we do today with clustered systems. But here we are speaking in terms of clustering different architectures between different sites.

The national grid will have to be monitored as a single entity. Regardless of the specifics, such as what exactly will be monitored, it is important that a computer site on the grid has the ability to provide system and site status as needed. In other words, it is important to develop an integrated monitoring scheme where all monitoring information is stored in a single repository. In this way, any and all requested data that could possibly be useful for monitoring the national grid could readily be made available to and retrieved from a single source, from each site on the grid.

The importance of an integrated monitoring scheme is equal in measure to the difficulty of the task. It may take years for NCSA to fully integrate its monitoring tools. In the early days of the supercomputing industry, operators and system administrators developed their own monitoring tools. Today, the development of system monitoring tools is an industry in its own right. Billions of dollars are being traded annually for the promise of a comprehensive monitoring tool that will provide a single interface between the system operators and the systems they monitor.

Operators used to outnumber the systems they monitored. In the early 1990's, NCSA's operations staff consisted of three to four full-time employees on both the second and third shifts, and five full-time employees on the first shift. Their attention was focused on two systems, the Cray-II and the Cray-YMP. The operations staff has decreased in size over the years, while the number of computers they monitor has increased dramatically. However, sophisticated monitoring tools are balancing this scale in favor of the operator.

2 Development of the OS-Log

It has always been the case at NCSA that whenever there is the slightest hint of a problem on one of the systems, it is as if someone yelled, "FIRE!" People come out of the woodwork, so to speak, to put those fires out, regardless of the time of day. Operators, administrators, supervisors, engineers, and even users get involved.

Everyone gets involved, and everyone is very careful with details. So careful, in fact, that we used to write on paper every keystroke that was entered at the system console as root user. An e-mail message would then go out to those who needed to know about system changes, reporting any action taken as root user. The positive side effect of this was that the people involved monitored not only the systems, but each other as well.

Over time this practice proved so effective that it developed into what is called the OS-Log (Operating System Log). The OS-Log is simply a file online every monitored computer. It serves as a journal for all system changes. Anytime a change is made to any monitored system, the person making that change must document that change in that system's OS-Log. Cutting and pasting is sometimes necessary to document a change while, at other times, a brief note will suffice. Three times a day a

cron job gathers whatever new information has been posted to the OS-Log, and sends it via e-mail to people who need to know—operators, administrators, supervisors, engineers, and in some cases users.

The OS-Log, in addition to keeping everyone involved, offers several other huge benefits. For example, it serves to help diagnose system problems, as sometimes it is the case that what has last changed on a system is found to be the cause of a problem. Another benefit is its effectiveness when used as a training tool for the novice operator, or as refresher material for the seasoned administrator. Even the seasoned administrator can use help with those problems that can resurface after years of once occurring, their resolutions having been forgotten.

It might be surprising to note that even now there is an OS-Log of sorts for the grid:

Error! Bookmark not defined.

From that page:

> *The objectives of this repository are to aid in the dissemination of information, provision of training material, as well as to serve as a forum for idea-exchange and experience-sharing amongst users of all levels. Readers who feel that they have experiences on the Origin 2000 that could benefit others are encouraged to contribute to this Page*.

Here we are sharing information among developers, those who need to know—again to include operators, administrators, supervisors, engineers, and users. And because this repository is so similar in nature to the OS-Log, it will not be of any surprise to find that it will take some people a long time to get used to contributing to and referring to the repository for the Origin2000. Like the OS-Log, this repository at Boston University will serve as a diagnostic tool. For some people it will serve as a training tool, while for others it will serve as a reference for research and development.

3 Development of the Virtual Operator

Just as the OS-Log evolved from an old policy (to write on paper every keystroke entered at the system console), V-Oper developed along a similar vein. In the days when operators outnumbered the systems they monitored, each operator would create their own ad hoc scripts to help with some specific monitoring task. The script might wrap a simple *ps* command and then beep, send e-mail, or flash a banner message—anything to get the attention of the operator about some anomaly.

As NCSA grew in size and complexity, the operators had to streamline their operation. They were monitoring more systems, more processes, more users, different architectures, and all this amidst constant growth and change.

When the operators pooled their resources, small ad hoc scripts turned into larger and more robust scripts. However, a typical script was mostly a bulky framework of environment variables and declarations, checks and balances for errors, also incorporating some kind of alarm mechanism, and generating a log of its activity. Only a few lines of the script contributed to the actual monitoring need. For the sake of efficiency, the bulky framework common to the many monitoring scripts should stand apart from and be shared by them.

The many monitoring scripts resulted in alarms from just as many sources. There were too many voices yelling fire, sometimes making it difficult for an operator to know that the alarm pending should be taking priority over the alarm being addressed.

The goal of NCSA's operators has always been to discover and correct problems before the user community could be effected by them. The ideal would be to have the ability to investigate and correct the course of even the potential problem. In other words, the goal has been to muster more navigating control of the systems, to be able to effectively pilot the systems through a carefully monitored environment.

A virtual operator was needed within the NCSA computing environment. (Think of HAL in 2001: A Space Odyssey). A virtual operator could provide the necessary framework and run all the monitoring scripts throughout the computing environment, and report to the operations staff through a single interface.

The solution was to build V-Oper—short for Virtual Operator. V-Oper was first developed as a non-interactive bourne shell script. Its task was to display, on a single screen, messages generated from all monitored systems. Today, V-Oper is a PERL script that communicates and interacts with the operations staff through an xterm session. There is nothing very fancy about it. Its library of monitoring functions perform tasks such as checking:

- load average
- disk usage
- remote mounts

- root processes
- status of batch queues
- status of backups
- status of security monitors

V-Oper also keeps records for trend analysis, and supplies the operator with helpful hints and possible actions to take against reported problems. It is capable of event management, whereby it could respond automatically to a given event. For instance, if a daemon was found to be missing from the process table, V-Oper could restart the daemon.

V-Oper is not designed to handle all of the monitoring needs of NCSA's computing environment. NCSA employs a number of other monitoring tools from outside sources. Therefore, to elaborate on the inner workings of V-Oper is not in keeping with the scope of this paper, which is to discuss NCSA's move toward a totally integrated monitoring scheme. If there is interest in knowing more about V-Oper, please contact **Error! Bookmark not defined.**.

4 Other Monitoring Tools

Tripwire, developed at Purdue University, monitors changes in a UNIX file system. NCSA runs a long and a short version of Tripwire to monitor the security status of its computing environment. The long version runs once a day against thousands of files. The short version runs every hour against only the most crucial files.

The short version of Tripwire allows the operator to quickly know when the status of a crucial system file has changed. In some cases, the short version of tripwire will report to the operator a change that an administrator has made to a system file, before the administrator has had a chance to document the change in the OS-Log! The administrator receives a call from the operator who is trying to verify that the change was made by a friendly source. The administrator is always quite impressed by such a call.

Swatch, a *simple watcher* developed by Stephen Hansen and Todd Atkins, is another monitoring tool that has proved to be extremely useful. NCSA configures its monitored hosts so that messages generally written to the hosts' local syslog files are rerouted to the syslog file on a designated syslog host. Swatch runs on that one host, scanning the local syslog file which now contains all the messages from all monitored hosts, for alerts from TCP-Wrappers, Kerberos, and still other monitoring tools.

SunNet Manager by Sun Microsystems, and Performance Co-Pilot by Silicon Graphics are also used by NCSA's operation staff to help monitor and diagnose NCSA's computing environment. Visual tools such as these make it much easier for operators to understand what is happening with the computing environment.

What has been covered so far does not exhaust the list of monitoring tools employed at NCSA. The networking staff is separate from the operations staff, and has their own need for monitoring tools that the operations staff does not require. The same is true for the security staff.

Some tools, are sometimes not recognized as monitoring tools. The NCSA Helpdesk, which is staffed mostly by the operation staff, employs Remedy's *Action Request System* software to channel and track helpdesk requests. Users will call or write the helpdesk to inform the operators about a problem with their office workstations. The helpdesk is, for all intents and purposes, the best monitoring tool provided for the users at this time. The operations staff cannot monitor the users' workstations under the current monitoring scheme.

NCSA is home to hundreds of scientists and staff members, all of whom will have workstation problems to report at some time. It would be nice if we could run a basic monitoring program on the users' workstations. The helpdesk would then be notifying the users of problems detected on the users' workstations, not the users notifying the helpdesk. It would be really nice if we could also inform the users of causes and solutions for the problems on their workstations. And it would be especially nice if we could inform the users, too, that the operators have already taken steps to correct the problems. Of course, the ultimate would be that the problems are fixed automatically, by a very sophisticated monitoring mechanism.

There are times when the helpdesk will hear from users regarding a *monitored* system, such as a file server. They want to report the problem to the helpdesk. This is perfectly understandable and even desirable, as it is symptomatic of user involvement. However, this is one problem resulting in many voices yelling, "FIRE!"

If a user has a problem accessing files from a remote site, it should be possible for a user to enter a web site and probe for information regarding the status of the file server at that remote site. Is the system having software problems? Is the system's load excessive? Is the system down and, if so, what is the expected uptime? There

should be some means by which the user can determine the status of any system at any site on the national grid.

Keep in mind that the request for status information could be from several sources—a user, an administrator, or a grid-monitoring program. For this to happen with efficiency, all system information at a given site on the national grid must reside in a single repository.

5 Grid Monitoring

There is a big push these days by software developers to develop sophisticated monitoring tools, as some industries are investing billions of dollars annually for the means by which a very large and complex computer network can be monitored and controlled from a single remote site. Take the airline industry for example. Here, is a serious monitoring situation. The industry's standard is roughly "zero tolerance for error." To not meet this standard means that lives could be lost.

Scenarios of complex monitoring situations are not hard to come by. There are many industries bearing enormous responsibilities, and relying heavily upon their international computer networks for support.

So grid technology, in terms of system monitoring schemes, is not new. However, it would be wrong to think in terms of porting this technology to our industry. We have to port our industry to that scheme! This is exactly what people do in the business sector after having purchased the monitoring software of their dreams—they port their business to the monitoring software.

Both buyers and developers understand that this is a very arduous task. There are cases in which buyers have spent millions of dollars and years of hard work porting their business to the monitoring software, only to give up [1]. The software developers are working with industry leaders to simplify the task of moving toward an integrated monitoring scheme.

6 The Deployment of CA-Unicenter TNG

Computer Associates International, Inc. (CA) is a world leader in mission-critical business software. CA-Unicenter TNG (Unicenter) is their solution to the complexities of monitoring large computer networks. NCSA is committed to the task of integrating its many monitoring needs within the Unicenter architecture, and has designated a Unicenter Deployment Team to work with CA to resolve the difficulties associated with the monitoring and management of very large, complex, distributed computing environments.

6.1 Architecture of Unicenter

The Unicenter architecture is supported by a communication protocol, the Common Communication Interface (CCI), using TCP/IP sockets, remote procedure calls, SQL named pipes, SNMP, or a combination thereof. SNMP, for Simple Network Management Protocol, is used to monitor and control networked resources. CCI also provides encryption of all Unicenter's event message traffic.

The architecture of Unicenter consists of:

- *WorldView*—A fully integrated user interface, and the Common Object Repository (CORE).
- *Agents*—The means to monitor and control all aspects of a business enterprise.
- *Enterprise Management*—Facilities that provide the means to manage information technology (IT) functions on machines throughout the enterprise.

6.2 Unicenter WorldView and CORE

The WorldView provides a library of ready-to-use two- and three-dimensional maps, and a two-dimensional graphical representation of the logical structure of an enterprise. It ties things together at the level of human perception and understanding, making resources more realistic and manageable.

Besides the maps of the WorldView, Unicenter provides browsers for:

- Class of common objects
- Object instances
- Topology of logical structure
- Link between objects

The Class Browser provides a comprehensive look at the classes and their properties. The Object Browser lists objects derived from each class. The Topology Browser displays the parent-child relationships between objects as they appear on the map. The link browser keeps track of connections among objects.

Whereas the WorldView ties entities together at the level of human perception, the object repository ties entities together at code level by:

- describing the structure of information
- explaining object relationships and interaction rules
- mapping instance data to an underlying database
- synchronizing information on events

It includes:

- class definitions
- managed objects
- policies
- topology
- status information

In essence, the object repository provides both the common services and storage facilities enabling management functions on all levels of the enterprise.

Building the Common Object Repository and populating it with object data that describes an enterprise is performed automatically by a process called Discovery. The Discovery service detects resources on an IT infrastructure and populates the Common Object Repository with managed objects that represent these resources and their relationships. A new resource added to the network can be discovered automatically, and a message then generated to inform the operations staff so that it cannot go undetected by them.

These aspects of the Unicenter architecture—the WorldView and the Common Object Repository—make it possible to create business oriented points-of-view, called Business Process Views. Business Process Views provide the logical connection between enterprise management and information technology. We can customize views of our enterprise. We can look at collections of common objects, such as all routers, or all routers in Building A only.

6.3 Unicenter Agents

Agents gather information about resources. An important feature offered by agents is their ability to "instrument" a resource so that specific information about that resource can be gathered and the resource managed. For example, suppose we are interested in just a few specific values in a large database. An agent can be written that monitors those values and notifies us when the values meet certain criteria. Unicenter includes several pre-packaged agents, but also provides the facilities to create agents of our own. The Agent Factory is part of a software developer's kit that is provided with the full distribution of the Unicenter software.

There are many agents available:

- The *operating-system agents* monitor such things as the utilization of CPU, memory, swap space, file system space, file system i-node number, message queue space.
- The *Log Agents* monitor events logged to a file, such as syslog.
- The *Process Agents* monitor the existence or absence of processes and services.
- The *Performance Agents* monitor system resources for performance bottlenecks.
- Agents are available for specific applications (for example: Lotus Notes).

The agent's role is to update the Common Object Repository, and to maintain a private local cache—referred to as an object store. The agent creates and forwards SNMP traps. When an agent detects an event, its WorldView icon will turn red if it is a critical event, or yellow for a warning. This change in color begins at the agent level, and propagates up to the host level, changing the color of the host's icon. Then, still propagating up to the subnet level, the subnet's icon changes color, and the propagation does not stop until it reaches the highest level. The operator, by drilling down through the different levels, will find the agent reporting the problem.

6.4 Enterprise Management

The Enterprise Management model for distributed systems management allows for implementation of policy-based management throughout an IT infrastructure in a centralized fashion. For example, we can direct important system, network and application event messages to a centralized location, and optionally respond automatically as a matter of policy to messages that would otherwise require human intervention. Policies are a set of rules that outline how and when enterprise management tasks are performed. Once policies are defined, Enterprise Management automatically follows those policies.

Enterprise Management provides many functions:

- Event Management
- Problem Management
- Workload Management

- Performance Management
- Automated Storage Management (ASM)
- ASM Media Management
- Security Management
- Print / Spool Management
- Report Management

We will take a look at a few of these that relate most to the topics already discussed in this paper.

6.4.1 Event Management

We have already touched on the Event Management function, which enables us to define policies that identify important activity and respond to them automatically. An Event Management Console serves as the interface to the Event Management function. It provides a complete view of the ongoing event processing across the entire enterprise. It is a visual window into the message traffic of events, commands, and messages monitored by the Event Management function.

Events, commands, and messages can be:

- Translated
- Categorized
- Suppressed
- Routed to remote servers
- Enhanced to draw attention to them on the Event Management Console
- Displayed in a separate window on the Event Management Console
- Consolidated to one central display
- Filtered based on server, user, machine, and workstation
- Used to execute programs automatically in response to an event
- Used to issue commands either on a local server or a remote server
- Used to trigger the automatic opening of problem incidents via Machine Generated Problem Tracking (MGPT), a feature of Problem Management

6.4.2 Problem Management

The Problem Management function provides a mechanism for the management of day-to-day problems and questions, to continuously improve the reliability of a computing environment. It requires the accurate identification of the cause of a problem as it relates to specific hardware, software, and/or procedural errors.

There are three major areas within Problem Management:

- Component definitions
- Problem definitions
- Machine-Generated Problem Tracking

Component definitions are used to define a systems' configuration, including hardware and software, as well as non-computer related equipment such as environmental systems (air conditioning, heating), telecommunication components, security systems, and more. Component definitions let you record warranty and maintenance information for each component, establish parent / child relationships, and track the movement of components.

Problem definitions are entered into Problem Management manually by the operator, and automatically by the Machine-Generated Problem Tracking (MGPT) facility. A problem is any reported incident that requires investigation and action. MGPT provides for the automatic opening of problem tickets (records) based on activity monitored by the Event Management function.

Problem Management ties in with Event Management and helpdesk software. The benefit of this is that we can create problem-tracking policies to identify problem conditions on individual hosts, within applications, and on the network. This could, for instance, provide information that may be used to assess the impact of the failure of one component on another.

6.4.3 Workload Management

As system usage levels and complexities increase, it becomes more difficult to keep computer systems running efficiently. The Workload Management function controls crucial operations, such as:

- scheduling jobs
- monitoring job sequence
- monitoring job failure
- adhering to time requirements
- sharing and distributing jobs across machines where there are sufficient resources for them to run

We can schedule jobs using calendars, which is called predictive scheduling. Or, we can schedule jobs through

the use of triggers and actions, referred to as event-based scheduling.

6.4.4 Performance Management

Performance agents collect both real-time and historical data from the operating system. The Performance Management suite provides four graphical applications:

- Performance Scope
- Performance Trend
- Performance Configuration
- Chargeback

These GUI applications provide not only an online, real-time window into the performance of the systems across an enterprise, but also enable us to examine the historical performance of our systems over many days, weeks or months, allowing us to spot performance bottlenecks, problematical trends, etc.

Using the Performance Scope we can:

- Identify performance bottlenecks
- Identify components which are consuming excessive resources
- See the effect on performance of changing system parameters
- Trace back to when a problem first occurred

Using the Performance Trend we can:

- Determine which servers are heavily loaded
- See the patterns of activity and use of applications and servers
- Identify problematic trends
- Investigate the impact of moving applications and users to other servers
- Determine the effect of running Workload at different times

The configuration structure of the Performance components allows us to perform component configuration from a central point of control, using GUIs. All component configuration information is held in a central location, and is distributed to remote components as necessary.

The Chargeback component provides a method of attributing resource accounting data obtained from heterogeneous platforms across the enterprise to "Real World" charge groups. This enables us to identify the use each user is making of each distributed system.

7 Summary

We have looked at NCSA's current monitoring scheme, which incorporates a variety of monitoring tools. We have discussed the importance of integrating these tools. We have learned that the private business sector has a lot of experience in the development and deployment of monitoring software suitable for very large, complex, distributed computing environments.

NCSA is moving toward an integrated monitoring scheme with the help of Computer Associates International, Inc. (CA), a leader in mission-critical business software. There is an ongoing effort between NCSA and CA to resolve the difficulties associated with the move to an integrated monitoring scheme.

8. Acknowledgments

I would like to thank Computer Associates International, Inc. for their contribution to this paper. Much of what appears in section 6 was taken right out of their CA-Unicenter_TNG_Concepts_Guide [2], with their permission. I owe a debt of gratitude to the other members of the Unicenter Deployment Team: Ral Geis, Cameron Ninham, and Randall K. Sharpe. Not only did they contribute by proofreading and amending the rough drafts, they never complained about the time I spent on this paper, away from the Unicenter project.

9. References

[1] InformationWeek (1998). **Enterprise Management Disillusionment**. [On-line]. Available: //www.techweb.com/wire/story/TWB19980218S0012 [1998, Feb 18].

[2] Computer Associates International, Inc. (1997). **Unicenter TNG: Concepts Guide.** Islandia, NY: Author.