BY CRAIG OLSON

Tackling the Problems and Pitfalls of Multiple-Vendor Platforms

Problems that span multiple platforms are a source of frustration for both users and administrators. As more systems are interconnected and those interconnections grow in complexity, the chances for transient abnormalities increase. To meet these growing challenges, system administrators need access to increasingly sophisticated tools.

ORGANIZATIONS commonly use different computer systems to meet individual application needs. In many cases, each of the different systems represent a different platform, vendor and operating system. In this resulting mixed environment, administering systems can be a challenging task.

When unexpected situations arise, system administrators looking for a resolution often face one of two undesirable scenarios. Either they have to operate tools on several different systems while trying to address separate aspects of one problem, or they must try to use one global tool with access to several different systems that can address multiple aspects at once.

The first alternative is quite precise, but can be rather awkward. The second solution offers a broader scope of the situation, but often sacrifices the amount of detailed information. Depending upon the situation, either choice may fail to provide a timely solution. The key is to match the tools available with the problems that arise and to recognize that for some problems you may not yet have the tool that matches.

Most systems administrators are familiar with the situations illustrated in the following scenarios. What may not be as widely known are the available management strategies and practical methodologies for evaluating, selecting and blending these solutions.

SCENARIO #1: THE FORGOTTEN PASSWORD

You are a system administrator for the dozen or so servers that house your company's accounting, inventory, shipping and support applications. Right now, you are working on a report for your manager on department staffing levels. Just as you are in the middle of entering numbers into a spreadsheet, a user walks into your cube and says, "I've forgotten my password."

Slightly exasperated, you roll your chair over to an x-term and fire up the appropriate utility. You ask the user for his user ID, but keep getting an error when you try to bring up the user record. After a couple of tries to make sure you spelled it right, you realize this isn't the system the user is trying to access. You roll over to another x-term and try again; same result.

"Which server are you trying to use," you ask. "I need to get into SAP," is the reply. After you explain that SAP is an application, not a server, the user still doesn't know which server. You continue to move from one administration terminal to another while explaining that some of the servers are named after planets and the rest are named after the seven dwarves. The user is still drawing a blank. As you work through each server's administration interface, you list each of the server names to no avail. Finally, it turns out to be the last system you try — and not even one that has an SAP application on it. Never mind; you reset the password and the user goes away, possibly happy. Now you can get back to your report, if you can remember where you left off.

Forgotten, misplaced or disabled passwords are typical security issues. The solution seems simple: Find the user's account and re-enable the password. What might be simple for one system, however, becomes more complex with each additional system. Unless you are lucky enough to work in a single vendor environment, for every system that is in use there is likely to be a different means for administering that system's security.

The first task, then, is to identify the system on which the password needs attention. Quite often, this can be a challenge in itself, as the administrator and user are often at very different levels of computer competency. Language, terminology and vocabulary all work against the situation when one is looking from the hardware out and the other is looking from the end user's monitor in. Software interactions can often occur behind the scenes, creating situations where the user is actually logged into different systems at different times, with the distinctions between servers being far from obvious to the casual user.

If passwords were only lost occasionally, the problem would seem simpler. However, security and access issues always need timely attention. Business growth usually results in an increase in users, systems and security concerns. Often, these concerns result in time limits on passwords, requiring users to change them at regular intervals. Large numbers of users changing passwords on large numbers of systems will almost guarantee that some user will misplace some password for some system at an inconvenient time for the system administrator who must drop everything to retrieve it. What is needed, then, is a way to shift the burden of activity away from the administrator and toward something else. The possible solutions for this something else range from very direct to very complex.

The Direct Approach

An example of a direct approach is cooperative security. This approach allows security information, such as passwords, to be managed cooperatively across multiple systems. One example is the Network Information System (NIS) feature on UNIX systems. NIS, in theory, allows one system to share information with other systems on the network. To put this to practical use in this case, we would use NIS to share the password file among all the UNIX systems on our network. When a user changes his password on one system, NIS allows that change to take place automatically on other systems without any extra user interaction.

NIS is not particularly easy to set up and may take some time to configure properly on each system. However, the trade-off of time spent doing the setup once vs. time spent repeatedly hunting down lost passwords should be easily justified. While it is true that in the past some sites may have had problems using NIS, these problems are in the past. Recent operating system releases for most vendors include a stable and mature version of NIS.

Complex Solutions

At the other end of the spectrum are the complex solutions. These bring technology to bear on the issue of security. The weak link in the password issue is the user's ability to remember it. If that is the problem, then the potential solution is to get rid of passwords completely.

The key is to match the tools available with the problems that arise and to recognize that for some problems you may not yet have the tool that matches.

For companies looking for a more high-tech solution, the emerging field of biometrics may hold the answer. While there are many possible security solutions in the general category of biometrics, the most promising may be the concept of the fingerprint reader. In this system, small touch pads are installed next to each computer terminal. To log on to a system the user places a finger on the touch pad that is connected to the security server. It scans the user's fingerprint, and if a match is made, the user is admitted to the appropriate system. No passwords are needed. While this method requires an investment in time and money to select and deploy, the return on investment will be in enhanced security and simplified administration.

SCENARIO #2: THE SLUGGISH NETWORK

You've just picked up where you left off on that report on department staffing for your manager. The telephone rings. With a sigh, you answer it. "Why is the network so slow?" a user asks. You glance up at a screen showing packet and error rates for the network; there is nothing obviously wrong. "It looks all right to me," you reply. "What seems to be the problem?"

"It's just slow. I've been waiting and waiting, but none of the commands I've started ever get done," the user replies.

After more conversation, you discover that the user is trying to read a file on one server and create a new copy on another. When the first request appeared to take too long, the user tried again. When that request also failed to complete, he kept trying again and again. When that still did not produce the desired results, the user called you to complain about the network.

It's possible that there is a problem somewhere, but it may not be the network. Trying to locate the exact problem involves at least three components: the system from which the file is being copied; the network across which it's copying; and the system to which it is being copied. Unfortunately, the problem could be in any one of the three and, as administrator, you have to check all three. In most organizations with multiple platforms and vendors, this means multiple system tools and locating the problem could be a headache.

First, you use one tool to do a general health check on the system where the copy originates. You need to determine why the user's COPY command did not complete. Is the system unusually busy processing some high priority task or handling a high number of disk requests? No, everything appears normal.

Following the path of the request, next you use another tool to examine the network(s) that are carrying the data for the copy. This may be as simple as checking statistics for a single LAN, or as complex as interrogating several network segments, routers and bridges. In the complex case, you may need to use several different tools to get a complete picture. As with the system, you are looking for something abnormal — a high number of errors, for example — or for signs of congestion. Your analysis indicates no particular problem with the network. It is being used, but the level of activity seems normal.

Finally, you use yet another tool to look at the second system, the target of the new copy, and conduct an examination similar to the one on the first system. Unfortunately, the results are also similar; you cannot find anything operating out of the ordinary. Perplexed, you call the user back to verify the system names, the response sounds all too familiar: "What? Oh, that? The problem went away right after I called you. Whatever you did worked."

Of course, you didn't do anything. What you did was put effort into a time-consuming task complicated by the different architectures involved and the transient nature of the problem. There is no way to pinpoint the trouble until it happens again. Maybe the next time you'll catch the glitch before it goes away. For now, there is nothing more to do but get back to the report you still haven't finished.

Problems that span multiple platforms and disappear before they can be examined are a source of frustration for both users and administrators. As more systems are interconnected and those interconnections grow in complexity, the chances for transient abnormalities increase. To meet these growing challenges, system administrators need access to increasingly sophisticated tools.

The first set of sophisticated tools allow a shift from reaction to action. In reactive mode, the tools used collect system activity data operate only on request — usually after a problem is reported. In active mode, the tools collect system activity information continuously. As the data is collected, the tools also compare selected statistics to normal behavior. When a statistic's value crosses a threshold into the realm of abnormal behavior, the tool raises an alert condition.

Using a system tool with this type of active notification offers two advantages. First, it provides the basis for preventative diagnosis — looking for the causes of abnormal situations before the impact to the user causes the system administrator's telephone to ring. Second, it provides an immediate system summary when problems are reported — a single place to quickly check a system's vital signs.

The shift from reactive monitoring to active reporting can be made one system at a time, minimizing the impact on systems, organizations and personnel. While this may still leave a variety of tools by platform, the benefit comes in the number of problems avoided or resolved.

A more complex version of this approach is to shift into active mode while integrating all the information collected into a single interface. The Network Management Console provides a single point of control solution for all system and network operations in the company. Network Management Consoles, and there are several providers on the market, involve small agent software packages that run on each system, probes that interrogate the network, and a central host that aggregates the information and reports it within a single interface.

These products also offer many built-in threshold and alerting features that could have identified the user's problem in this scenario before the user noticed the delay. However, while overall management consoles have a great breadth of information for identifying a problem, they do not always provide enough detail regarding the situation to actually resolve the issue.

In some cases, there is no alternative but to fall back on the specific tool suited for the specific system to locate and solve the specific problem. However, while global sophistication may not yet provide all the answers, the move from reactive problem hunting to active problem avoidance will reduce the number of questions.

> **While global sophistication may not yet provide all the answers, the move from reactive problem hunting to active problem avoidance will reduce the number of questions.**

SCENARIO #3: THE UNRESPONSIVE SYSTEM

You've just about finished the department report when your manager walks in. You show her a couple of elegantly produced graphs of your results. "Never mind that," she says, "I just got out of a staff meeting. The order-processing department says they've been getting a bad response time the past three weeks, and it's only getting worse. Figure out why."

This problem is different from the other two because it is chronic — it has been building over time. The other two are acute — they happen in a matter of seconds. This one is going to make you think in a different dimension.

On the surface, you know the problem should be easy to solve. If the situation has lasted for weeks, then there is little chance that it is going to elusively slip away while you are trying to find an answer. On the other hand, what you need to be able to do is compare the performance of the order-processing application now with the performance sometime before three weeks ago since they were satisfied with response time last month. This month, it's a different story. If you can just find what is different between the two points in time, you will have the answer.

It sounds simple, but depending upon the tools you have installed, actually solving the problem may not be. If, for example, you have no historical information about the performance of the system before three weeks ago, nothing short of time travel will clear things up immediately. In addition, since it has been suggested that the trend is toward increasingly poor response, you need to focus on the future as well.

While time travel may be a thing of the future, it does have practical applications in system management. The situation described in this scenario requires two solutions: one to identify what happened to response time in the past and another to identify how to avoid a similar condition in the future. Looking into the past can be accomplished by continuously saving historical information on system performance. This may require a shift in perspective similar to the move to active from reactive. In this case, the shift is from monitoring performance to recording performance.

Many products display system performance information. This is useful for examining the current state of a system. Some of these products also record and preserve performance information for later analysis. The value of these more comprehensive tools is immense when faced with the need to examine previous system behavior. There is no substitute for information not collected.

Organizations moving toward active system resource management should weigh the future value of historical analysis when making product selections. Given the availability of historical performance information, the problem of identifying the source of the slow response time becomes a matter of comparing the past and present information.

The actual cause is likely to be found in one of two areas. Perhaps some application-side usage changed when, for example, a new version of software was installed, resulting in a marked increase in system utilization. Alternatively, the normal growth of system activity may have pushed a critical system resource, like the processor, over the edge of its utilization curve, resulting in increased resource contention and overall poorer performance.

The first situation is an easy one. The second involves a more analytical approach. Likewise, examining future impact also requires an analytical approach. In both cases, you need a tool that takes a more abstract view of system behavior. Typically, these products are called capacity planners, modelers or forecasters.

DELVING INTO THE TOOLBOX

In many cases, the aura of mystery and complexity around these products can intimidate the faint-hearted. This aura may be deceiving, and the value of bringing an analytical approach to system administration may be well worth the investment in breaking through the barrier. The trick is, again, to find one that runs on as many of the installed platforms as possible, so only one interface needs to be learned and day-to-day management is less complex.

Using a capacity planning tool as part of normal systems analysis can provide better insight into the causes of system behavior and response time problems in particular. For example, the analytical approach can show the relationship between system component usage and application response time more clearly than an online monitor can. The ability to anticipate the impact of application growth, or the benefit of configuration changes, make the use of a planning tool worth the effort.

Irrespective of the situation, when choosing a solution it is vital to first review the scope of the problem and decide to what extent it needs to be solved. Often the biggest obstacle to overcome is taking the time to start looking for a solution. If a good portion of a typical day is spent fighting fires, it is difficult to find time for quiet analysis of potential fire fighting or fire suppression tools.

When this is the case, it may be a good idea to reduce the scope of the problem you are trying to solve. A simple issue may very well be resolved at no cost with freeware. However, accept that while public domain software, shareware, or so-called "lite" versions of commercial products may be a good starting point, they are not always the end solution. Often these solutions represent a lower technology than for-pay products.

Often the biggest obstacle to overcome is taking the time to start looking for a solution.

Low-tech tools usually do not interact with other software and, in mixed platform environments, a handful of clever but unrelated tools may introduce more confusion than conclusions. If you understand these limitations, using low-cost methods while seeking a long-term solution will still be beneficial. Consider the time saved by the interim solution as time needed to find an end solution.

As your organization matures, you may find that larger/broader products become more attractive. This may lead you in the direction of an umbrella tool, like a management console, where other products fit "underneath" and cooperatively interact to increase the tool's usefulness. When building your suite of management tools, it may pay to keep in mind the potential for using components to build a more comprehensive solution.

It's also a good idea to anticipate problems. Take into account that while some solutions are very expensive, they are also very effective — they can resolve the immediate problem and even prevent other problems from occurring down the line, as in the unresponsive system scenario.

Other important factors to take into consideration when purchasing any software solution are the product's target market, system impact, learning curve and, of course, cost. Select tools where the product's intended audience matches the size of your organization, the number of terminals and users, and factors that may push its limits, such as the amount of guest log-on activity.

It is also important to understand the potential impact of a tool on the activity of your system. For example, purchasing a tool to report and record performance that creates a performance problem itself is not going to be an effective solution. Likewise, your organization's personnel turnover rate should define the amount of time you can afford to train people to use the software. If you have a high turnover rate, avoid complex solutions that require an investment in training.

CONCLUSION

It can seem to be a daunting assignment to find the right solution and upgrade your techniques and tools. However, remember that part of the problem with administering complex multiple-vendor systems is that the time that could be spent finding methods to make the job easier is often spent performing the tasks for which the methods are needed. With a modest investment in time and analysis, the payback can be rewarding. ts

Craig Olson is product manager for TeamQuest Corporation, Clear Lake, Iowa.