A Strategy for Managing Performance

John Brady, Sun Professional Services

Sun BluePrints™ OnLine—December 2002

http://www.sun.com/blueprints

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

Part No. 817-0836-10
Revision 1.0, 12/16/02
Edition: December 2002

A Strategy for Managing Performance

Computer systems play an important role in business, and with the appropriate attention paid to the compute environment, compute costs are minimized and compute equipment utilization is maximized. The result is a direct benefit to the bottom line. An important and often neglected area of compute environment management revolves around the tasks of managing computer system performance. It's easy to understand why this task is neglected, because poor performance and identifying its causes can be ambiguous and time consuming. What's needed is a strategy for managing performance.

This article offers suggestions for developing a strategy for managing performance. The strategy is high level so it can be applied to almost any compute environment. In addition, a variety of performance related products are listed.

This article is intended for CIOs, CTOs, IT managers, service delivery managers, and anyone interested in managing performance of computer systems.

Performance Versus Availability

The goal of the compute environment is to deliver adequate and reliable services to support the company and customers. Anything that adversely affects availability of the computer systems is immediately felt and quick remedies are sought. Various figures exist for the amount of lost revenue from an hour of downtime of critical computer systems in different industries. For example, a financial brokerage house can lose as much as $6.5 million for each hour of downtime.[1] To avoid this agony, extraordinary amounts of money are spent for equipment and personnel to avoid downtime and increase availability.

But what about the less obvious performance decline? Poor performance of computer systems has a negative impact on company revenues along the same lines as downtime. As with downtime, there is a negative effect either through lost opportunities (customers or orders turned away), or through lost revenue (inability to ship or bill). A comparative cost figure can be derived for performance problems; a five-percent decrease in performance of a critical application translates to just over an hour of lost processing in a 24-hour period. For the same brokerage house, a five percent performance decline can also result in millions of lost dollars. If the loss were due to downtime, a remedy would be immediately implemented, but for poor performance, without a strategy to detect and correct the decline, the impact on revenue might go unchecked indefinitely.

What is Performance Management?

Performance management is the measurement, analysis, and optimization of computer resources to provide an agreed-upon level of service. The focus is on service delivery.

To understand the concepts of performance management there are a few definitions and concepts that are helpful as described in the following sections.

1. Source: Contingency Planning Research & Strategic Research Corporation.

Performance Management Spectrum

Managing computer system performance is not binary (you do it, or you don't), but rather a series of what the Gartner Group calls maturity levels as follows:

- Chaotic—No consistent use of performance tools.
- Reactive—Event consoles are used.
- Proactive—Performance monitoring and historical tools are used.
- Service—Capacity planning practices are followed.
- Value—An IT/business metric is established (performance is managed with a direct link to revenue).

Each of these levels builds on the one before, adding more management and control of the underlying systems, and providing more benefit back to the business. Start by developing the infrastructure to be reactive, move towards the proactive level, and build further over time.

Monitoring Levels

Measurements for throughput, latency, and utilization need to be made at all levels of the stack:

- Hardware (processor, memory, disk, network)
- Operating system
- Database and middleware
- Application

Each level of the stack must be monitored because there might be inefficiencies at any of the levels, which do not appear as inefficiencies at other levels. You can only identify and resolve these situations when you have data available for all levels.

For example, an ORACLE® database that is missing an index on a table will adversely affect performance at the database level. If you only monitor performance at the hardware and operating system level, all the statistics (disk access, CPU utilization, and so on) will appear normal, and the problem will not be detected.

Performance Management Products

There are a variety of products available to support performance management strategies.

For the operating system level on Sun systems, the Solaris™ operating environment automatically keeps track of many activity counters in the kernel, and these can be extracted in a number of ways. Two utilities, each of which provides lists of statistics about various system activities are as follows:

- `sar`—gives information on overall system performance over time (not on individual processes).
- `system accounting`—gives information on the total resource usage of completed processes.

For a complete performance management strategy, you should use third party products that hook into the system data and provide extra functionality. There are a number of products available, all of which target some of the many different requirements for performance management. Many of these are point products, addressing only one or two of the requirements of performance management, but there are a few comprehensive products that cover all the requirements.

Currently, BMC PATROL and TeamQuest products (see "References" on page 11) are particularly comprehensive on the Solaris platform, covering all the major requirements of performance management, which include:

- System activity measurement tools
- Workload breakdown information
- Measurement of higher software layers (for example, ORACLE and SAP)
- Browser-based publication of standard reports
- Customizable thresholds, alerts, and alarms
- Features to model system behavior
- Ability to identify bottlenecks
- Facilities for capacity planning for future growth

There are other performance management products available that address specific needs of different situations. Some specialize in one of the areas, or are specific to an application software environment (such as ORACLE, SAP, or the Web).

For all of the functionality and scope, these performance management tools are relatively inexpensive. Prices vary, but tend to be less than one percent of the list price of the computer system they run on. This is a small price to pay to ensure that the valuable systems are operating at maximum efficiency.

Performance Management Strategy

Performance management strategy starts with gathering activity data for all levels of the system stack. The data is used to check against the current system behavior and to estimate future performance.

You can analyze the historical data to determine the average values for the most important system statistics. You can use the historical data to determine the range of acceptable deviation from this normal value. These normal values can be used as threshold values to test the current system activity for signs of significant deviations.

The historical data can be analyzed for trends over time. Is system utilization or workload increasing, or is it steady? Are there any cycles within these changes –monthly or yearly? These simple trends can be extrapolated forward to predict future workload requirements.

As performance data is collected, it can be checked against the established thresholds determined earlier. Any significant deviations can be reported and investigated. This investigation uses the performance data that is being collected to drill down further to establish the cause of the anomalies.

The volume of data gathered about the system activity can be used to build a model of how the system behaves. Using the expected changes from the trend analysis, and other anticipated changes in workload, the future performance of the system can be estimated using this model. If there are any shortcomings in the results, the model can be changed in various ways to show what needs to be done to meet performance targets.

The following sections outline a performance management strategy by dividing the activities into two broad categories –gathering data and taking action.

Gathering Data

The following four steps define the basic concepts of performance management in a way that moves your management strategy from chaotic or reactive to the proactive or service level. To achieve and maintain these higher levels, you must continue to implement these steps, repeating them continually to stay in control.

1. **Define Goals**

   Before you can check if any system is performing optimally, you first need to know what you expect it to be doing. Ultimately, this is to satisfy the business requirements for which the application was deployed.

   These requirements should be expanded further and form part of the Service Level Agreement (SLA), which defines the nature of the service that the system and application is expected to deliver. Then you need to define a number of key performance indicators (KPIs).

   The SLA is an end-user oriented document, using standard business terminology. KPIs are specific definitions of quantities to be measured, that together show that the SLA has been met. The SLA mentions the general targets of the system. The KPIs define exactly what to measure, how to measure it, and the allowed values. The KPIs are unambiguous so that there are no disagreements over what the target is, what is being measured, and what the system is achieving.

   For more information on the role of the SLA in the data center, how to create one, and how to define KPIs, refer to the Sun BluePrints™ article titled "Service Level Agreement in the Data Center" (for the URL to this article, see "References" on page 11).

2. **Define Workload**

   With nothing else done, you might measure activity on a system and get a result of something like 80% CPU utilization. What is using 80% of the CPU? Further investigation is required to determine what is consuming the resources on the system.

   Structure the work on the system into well-defined workloads. Record resource consumption against the defined workloads. Workloads normally correspond to sets of processes (running programs), and it is straightforward to gather the process information and translate it into workload data.

   Workload breakdown is important for problem analysis, and for future capacity planning under changed or new workloads.

3. **Measure Performance**

   When you implement the systems and the application software, you need to collect and store measurement data on what each system is doing.

   There are three primary types of measurements:

   - Latency—The elapsed time taken to perform one unit of work. From an end-user perspective, this is the response time of the application. But you can also measure the latency of individual actions and tasks within the system.
   - Throughput—A measure of the amount of work done in a period of time. Usually this is expressed as a number of transactions completed in a period of time. Throughput is a measure of the quantity of service, while latency is a measure of the quality of service.

- Utilization—The amount of a fixed capacity resource that was used. Most often this is either for a resource that remains consumed until released, such as memory, or for a resource that has a renewable capacity for work for a given time period, such as CPU utilization. Normally, this is expressed as a percentage of the available capacity.

These measurements need to be taken at all levels of the stack (hardware, operating system, database and middleware, and application levels) that make up the environment within each system.

In collecting and storing these performance metrics, you automatically create a baseline of recorded activities which contributes to defining what is normal or typical usage. Save this data because you can use it later for comparison purposes.

4. Analyze, Model, and Deduce

With enough data collected about activity on each system, use the data to create a model of what each system is doing. Although this can be done manually, this is time consuming and awkward, and is accomplished much easier with software that is designed for this purpose (see “Performance Management Products” on page 4).

The performance management software employs various mathematical techniques to represent the computer system and the flow of work through it. This can be done with enough accuracy that the calculated behavior of the system is very close to the actual system behavior.

A common technique used is that of queuing networks. Each resource in the computer (CPU, memory, disk, network, and so on) is represented as a *server* with a finite capacity for processing, and a *queue* for holding requests in front of it. These resources are interconnected to represent the computer system.

As work comes into the system, each resource server processes the respective requests. If requests arrive at a greater rate than the resource can process, then the requests sit in the given queue. In such situations, the total elapsed time for each request increases significantly as the requests first have to wait before being processed.

By changing characteristics of each server, such as the processing capacity or the number of server entities, or by changing the arrival rate of work, different scenarios can be modeled and their performance evaluated.

Taking Action

Take the appropriate action based on the results gathered.

Correct Performance Problems

If you identified any performance problems while gathering performance data, take the time to fix the performance problems at this time. The goal is to identify the root cause of the performance problem. This can be done by using any preferred analysis technique, or by using the model created in the data gathering stage. The model can show you which resources are being saturated, which has the longest queue, and where most time is spent during a transaction.

The model can also help to show the real cause of the problem. You can change the characteristics of the identified resource (increasing its performance or adding more of it), and see the effect on the calculated performance of the system. If the performance improves significantly, then you have identified the root cause of the problem. It is possible that there might be more than one overloaded resource contributing to the poor performance of the system, and each resource needs to be identified in turn.

Monitor Ongoing Performance

Once the initial system activity measurements are taken, you need to continually monitor performance to ensure that performance remains within targets.

The simplest form of monitoring involves running regular reports and comparing the results with the baselines established earlier. Start by identifying any performance issues that are outside of the targets. This narrows down the number of systems for further investigation. For the systems that exceed the targets, you can also report on the level of deviation from the expected values.

Additionally, you can take the KPIs and the normal values for the system and use these as threshold values for your monitoring software. An alarm is raised when a value exceeds the threshold. You can set ranges of values for allowable deviations and implement a hierarchy of alarm levels. A simple example uses colors where green indicates a system within threshold values, yellow for a warning (for example, a system five percent over the threshold value), and red for critical situations. More sophisticated examples have more levels between good and critical.

Configure the alarm thresholds according to the most important KPIs, otherwise there will be more alarms than necessary.

Control Resources

With a clear understanding of the resources normally consumed by the application, you can choose to impose limits on the amount of resources the application is allowed to consume. This is a more proactive form of control. This is most useful on systems that concurrently run more than one application, and can stop one application from grabbing all resources on a system to the detriment of the other applications.

Resource controls can be implemented in hardware or software. With hardware controls, the resources are physically separated to control resource allocation. This is a form of a hard limit. Examples of hardware resource controls include separate systems, domains within a Sun server (logical systems within a single physical chassis), and processor sets.

With software controls, either the operating system or middleware restricts the amount of resources used. With some software controls it is possible to dynamically change resource allocation and to allow active applications to use resources that are unused by the other applications. This is a form of soft limit. An example of this is the Solaris™ Resource Manager software, which allocates shares of a resource to groups of processes.

Track Changes for Capacity Planning

With stable, well-performing systems under constant monitoring, you will build up a large collection of data of measured system activity. After several months, you will be able to perform trend analysis to see if the system workload is static or growing. The data might reveal linear, non-linear, or periodic peaks and troughs.

If there is a significant increase in system activity, examine the data to determine which workload is contributing to the increase. If growth continues, use the model created earlier to predict if the thresholds will be exceeded, and when.

It is important to know that performance degradation does not occur in a linear fashion. When a resource becomes saturated and a queue forms, response time increases substantially, often doubling very quickly. This is not obvious from a straight line extrapolation of system workload. But the modeling software will correctly report the nonlinear increase.

Once you identify an increase in workload, you can use the model to estimate future performance by applying simulated changes to the workloads, and by adding workloads. The model calculates the resulting performance benefits or hindrances. If the results do not meet the targets specified in the SLA and KPIs, investigate changing the resources in the model.

First, identify the saturated resources. Using the model, experiment by replacing the saturated resources with additional resources (more disks, processors, and so on), or by upgrading the resources for faster versions. This allows you to calculate in advance how much computer hardware is required for a certain workload. The result is superior to waiting until the system becomes saturated and performance is unacceptable.

Investigate Consolidation of Resources

Use the measured system activity data and models to calculate the effects of server consolidation.

In simple terms, you can combine the resource profiles of each separate application, and end up with a final system configuration. This is accomplished by combining the resource profiles and then combining the system behavior models. Then model the behavior of the new combinations with the desired workload. Any contention between the workloads and resources are revealed. You can change the model to increase the critical resources, and recalculate the performance.

Summary

Adopting and executing a thorough performance management strategy puts you in the position of being proactive and in control of your applications, not vice versa. No longer will you be reacting to performance problems when they have reached a level of disaster. Instead you will be predicting problems in advance, and taking remedial action before they impact the business applications and services. You'll make the most of the compute power in your environment, and save revenue at the same time.

About the Author

John Brady is a Principal Solution Technology Consultant in the Performance Optimisation Speciality group in Sun, UK. Since starting as an application developer many years ago, John has devoted his career to working with relational database systems running UNIX. During the past 10 years, he has worked with some of the largest and most powerful UNIX systems, especially multiprocessor (SMP) systems, for different system manufacturers. During this time, he worked in both technical pre-sales and consultancy delivery roles, always focussing on performance of large databases. As a result, he has gained invaluable experience in designing, building and deploying large complex server-based solutions.

References

The following URLs provide additional information about topics discussed in this article.

BMC Software, Incorporated: http://www.bmc.com

TeamQuest Corporation: http://www.teamquest.com

Wustenhoff, Edward "Service Level Agreement in the Data Center," *Sun BluePrints Online*, April 2002.
To access this article online go to:
http://www.sun.com/solutions/blueprints/browsesubject.html

Ordering Sun Documents

The SunDocs[SM] program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The docs.sun.com web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is http://docs.sun.com/

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: http://www.sun.com/blueprints/online.html