Overview of CS6400 Performance

S. Sridhar, CRI, Technical Support Group, San Diego, CA

ABSTRACT: *The Cray CS6400 is a scalable symmetric multiprocessing (SMP) system based on SPARC technology and runs in the Solaris 2.x environment. Presented in this paper is a high-level overview of the system's performance. The paper is intended to serve as a checklist or starting point for tuning the performance of the system. Topics covered include: factors which impact the performance, Central Processing Unit (CPU) issues, setting of swap space, Input/Output bottlenecks, and monitoring the system's performance. The material for the paper was extracted from a number of sources; in particular, from the excellent book "Sun Performance and Tuning: Sparc and Solaris", by Adrian Cockcroft, SunSoft Press, 1995, Prentice Hall, ISBN 013-149642-3. The paper concludes with a list of references that deal with the subject of system performance under Solaris 2.x.*

1 Introduction

An integral part of system administration is the task of getting good performance from the system. System performance depends on a number of factors. A thorough understanding of these factors is an essential requirement for a successful performance tuning exercise. Figure 1, a modified version of the one in Reference 1, shows a schematic of the computer system.

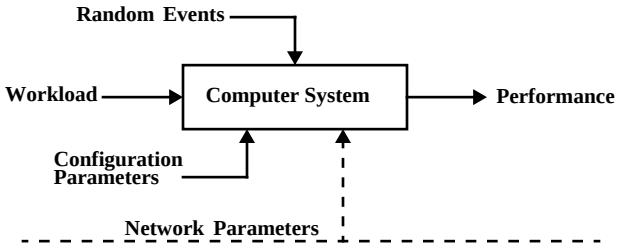

Figure 1. Schematic of Computer System

The computer's system resources covert the Workload that is put on the system to the desired output or Performance. The efficacy with which this conversion is done is influenced by Configuration or kernel parameters, network parameters and random events.

1.1 Workload

Workload characterization should be one of first steps in any performance tuning exercise. Workloads can run the gamut from "fixed" workloads where the system is dedicated to running one or two key applications to "random" workloads where there are multiple users running assorted applications. A reasonably accurate knowledge of the workload can simplify the task of defining the steps for getting good performance.

Even in the case of random workloads, monitoring the workload over a period of time can provide some insight into the nature of the workload.

1.2 Computer System

The computer system consists of the following resources: Central Processing Units (CPUs), Memory, Input/Output (I/O) devices, and the Operating System. The CPUs process instructions from the kernel and applications. Physical or main memory is the amount of semiconductor memory (RAM) in the system. I/O devices move data into and out of the computer. Though I/O devices include keyboards and printers, the primary I/O device which impacts performance is the disk drive which is used as an extension of the main memory. The performance of the system depends on how well these resources are allocated and used by the operating system.

1.3 Performance

Depending on the usage of the computer system, the measure of performance falls into one of the following categories: throughput, response time (sometimes referred to as latency), and utilization. Throughput can be defined as the amount of work that gets done in a given amount of time. Examples are

Transactions per second (TPS) and floating point operations per second (FLOPS). Response time is a measure of the time that a user has to wait for some work to complete. Examples are the time taken to see a keyed-in character echo on a terminal screen and the time taken to get an answer from a query to a database. Utilization is a measure of how much of the computer's resources are used to do a given task. Ideally, one would like the computer's resources to be fully used without adversely affecting the overall performance

1.4 Configuration Parameters

There are a number of configuration parameters, including those in the operating system's kernel, which can be set to desired values. Examples of such parameters are: kernel buffer sizes, paging parameters, shared memory segments, and database parameters. The key is to identify the parameters which need to be adjusted for the desired performance goals.

1.5 Network Parameters

Typically, systems such has the CS6400 live in a networked environment with other NFS client/servers on the network. The overall performance of the system can be very much dependent on the efficiency of the network traffic. If NFS performance is identified as a problem then the appropriate parameters that influence networking will have to be properly set.

1.6 Scope of the paper

The primary focus of this paper is on the computer system, i.e., the system resources consisting of the CPUs, memory and I/O devices. Presented in this paper are brief discussions of these resources in the context of performance tuning.

Network or NFS performance tuning, and Configuration or Kernel parameters are not discussed in the paper.

Application tuning, a key aspect of good performance, is not discussed in the paper. It is assumed that each application that runs on the system is adequately tuned in the following aspects: selection of an efficient algorithm, choice of the high level language, proper use of compiler options, and compatibility/suitability of the programming model for the system's architecture and operating system.

The CS6400 system is often used as a server for a relational database such as Oracle and Informix. Due to its complexity, database tuning calls for somewhat specialized knowledge and training. In general, unless one is prepared to invest significant effort in acquiring such expertise, it is recommended that database tuning be left to experts and consultants.

2 CPU: Processes

The workload is broken down into processes which run on the CPUs. A process is simply an instance of a program in execution, and can consist of multiple lightweight threads (LWPs), and multiple application threads. The kernel schedules the various LWPs and threads to run on the CPUs.

2.1 Process Status

A frequently used command for monitoring the execution status of active processes in the system is the `/usr/bin/ps` command. The information from the command includes: current status of the process, user ID, priority, memory used and CPU time used. Some of the problems that can be identified from the output of the `/usr/bin/ps` command are:

- Identical jobs owned by the same user requiring identical resources.
- A process which has accumulated a large amount of time, possibly due to a programming error such as an infinite loop.
- A process running with a priority that is too high. For example, an incorrect assignment of a "Real time" priority to a job can prevent other "System" and "Timesharing" from running.
- A runaway process that progressively uses more and more CPU time.

2.2 Processes Management

The following commands can be used to manage the processes running in the system (refer to the individual man pages for details):

- `/bin/priocntl` for assigning priority to processes.
- `/bin/kill` for removing a process that is causing problems.
- `/bin/vmstat` for a convenient summary of performance.
- `/bin/mpstat` for a breakdown of the CPU loading for each processor in a multiprocessor system.
- `/usr/sbin/sar` for monitoring system activity.

2.3 CPU Guidelines

A key piece of information in understanding the CPU loading is the System Load Average which can be defined as the average number of processes in the kernel's run queue during an interval. In Solaris 2.x, the `/bin/vmstat` command reports the number of jobs that are waiting to run. Table 1, excerpted from Reference 1, gives some guidelines based on the system load average. In the table:

r ... from the output of `/bin/vmstat -30`

N ... number of CPUs in the system

Table 1. Guidelines for CPUs

r/N	Status	Action
0	CPU idle	None
0-3	OK	None
3-5	CPU busy	May need to add CPUS
>= 5	CPU busy	Add CPUs

3 Memory

System performance suffers drastically when programs consistently require more physical memory than is available. When this happens the operating system has to resort to "paging" and "swapping."

3.1 Paging and Swapping

Paging involves moving memory pages that have not been recently referenced to a free list of available pages. This applies to user processes only and not to the kernel which, for the most part, is always memory resident. Swapping is the moving of sleeping or stopped LWPs to/from disks. It occurs if demand for memory is very high. If there are no sleeping or stopped LWPs, a runnable process will be swapped out.

In Solaris 2.x, paging and swapping are controlled by complex algorithms which form the heart of memory management in the system. The key kernel parameters which control the paging and swapping algorithms are: lotsfree, desfree, minfree, fastscan, slowscan, and maxpgio.

3.2 Swap Space

Swap space is simply the amount of disk space allocated for swapping activities. Following is a clarification of swap space allocation in SunOS 4.x and SunOS 5.x (Solaris 2.x). In SunOS 4.x, the amount of swap space and virtual memory are one and the same. This called for a swap space that was at least as much as the amount of real memory; typically, swap space was set to twice the real memory.

In SunOS 5.x, virtual memory is defined to be the real memory plus the swap space. Thus, using the old twice-real-memory rule can result in wasted disk space. That raises the question of how much swap space is needed in Solaris 2.x? The answer is - it depends on the workload. The ideal situation is to have zero swap space, i.e., have enough memory for the workload. Since that may not be practical, a good insight into the workload is essential for making an intelligent allocation of the swap space. If the system is dedicated to running third party applications or databases, the vendor should be able specify the swap space requirements.

Once the swap requirements have been determined, it is suggested that the following guidelines be used for swap allocation:

- Spread the swap devices over many disks/controllers.
- Avoid having 2 or more swap devices on a single controller.
- Never place 2 swap areas on the same disk.
- Keep the first swap device listed from the `swap -l` command large enough to hold a "vmcore" crash dump.
- - If swap performance is a problem, avoid using the Operating System disk for a swap device.

3.3 Useful Commands

The following commands are useful for monitoring and controlling memory and swap activities in the system:

- `/bin/sbin/sar` for monitoring of system activity.
- `/bin/sbin/swap` for monitoring and controlling swap.
- `/bin/vmstat` for monitoring paging and swapping.
- `/bin/mkfile` for increasing swap space.

3.4 Swap Space Guidelines

Table 2, excerpted from Reference 1, gives broad guidelines on the sufficiency of swap space, based on the "swap" output of the `/bin/vmstat -30` command.

Table 2. Guidelines for swap space

swap	Status	Action
>= 100 MB	Swap Waste	Reduce swap
10-100 MB	OK	None
4- 10 MB	Swap low	May need more swap?
1- 4 MB	Swap low	Increase swap

4 Input/Output (Disk I/O)

More often than not, performance problems are traceable poor disk I/O. The system will usually have a disk bottleneck. The key parameter to monitor is the "service time". This is the time between a user process requesting an I/O operation and the operation completing; for example, time between the initiation of a read request and the read completing. If many processes are accessing the same disk, service times of up to 1 second can occur.

Typically, once a disk bottleneck is identified and fixed, the bottleneck may show up at another disk. Thus the process of identifying and fixing the problem may have to be done several times.

4.1 Disk I/O Factors

Following are some of the major factors that influence disk I/O performance.

- SCSI Controller: narrow or fast/wide.
- Number of disks on one controller: The following general guidelines are suggested:
  - SCSI (10 MB/s) ... 3 or 4 active disks
  - SCSI (20 MB/s) ... 6 to 8 active disks
- Sequential or Random access.
- Raw disks or File systems: I/O performance can be significantly impacted by whether data is stored on raw disk slices or in a file system. In case of file systems, the type of file system (ufs, nfs or cachefs) influences the I/O performance.
- Disk Management: Increasingly, the only practical way to manage the large number of disks found in servers such as the CS6400 is to place the disks under the control of a disk management software such as OnlineDisksuite and Sparc Storage Array (SSA) Volume Manager. To ensure good disk

I/O performance, it is important the follow the recommendations that come with these software packages.

- Disk data layout: Typically, large disk configurations involve striping, mirroring, and hot sparing to achieve performance and high availability. Good performance is attainable only by using lots of disks and controllers, with a minimum number of disks per controller. Additionally, if high availability is also a primary requirement, the only option is to add more disks and resort to mirroring.

4.2 Disk load monitoring and balancing

As stated earlier, the key parameter for monitoring the disk I/O performance is the service time, svc_t, which is an output of the `/bin/iostat [-x 30]` command. As a general guideline, a disk bottleneck is indicated when:

- svc_t of more than 50 ms.
- a disk that is consistently more than 30% busy when averaged over a 30 second interval.

4.3 Disk I/O Guidelines

Table 3, excerpted from Reference 1, gives broad guidelines for disk load balancing, based the values of svc_t from the `/bin/iostat -x 30` command. Note that Table 3 is applicable to a single disk.

Table 3. Guidelines for disk I/O [Applies to a Single Disk]

%b	svc_t	Other disks	Disk Status	Load Rebalance
< 5	-	OK	OK	No
< 5	-	Busy	Idle	Yes
>= 5	< 30	-	OK	No
>= 20	30-50	-	Busy	May be
>= 20	>= 50	-	Busy	Yes

5 Performance Monitoring

The Solaris 2.x operating system is extensively instrumented. While the system is running, a number of counters in the operating system are incremented to keep track of the various system activities.

5.1 Activities tracked by the OS.

Following are some of the key system activities that are tracked:

- CPU utilization
- Buffer Usage
- Input/Output (I/O) activity
- System call activity
- Context switching
- File access
- Queue activity
- Interprocessor Communications
- Paging
- Free memory and swap space
- Kernel Memory Allocation (KMA)

5.2 Monitoring Commands

Following a list of the useful commands that can be used for the monitoring of system activities and performance. It is also possible to set up automatic data collection from these commands

- `/usr/sbin/sar` for overall system activity.
- `/bin/vmstat` for virtual memory statistics, CPU load, paging etc.
- `/bin/iostat` for disk I/O statistics.
- `/usr/openwin/bin/perfmeter` and `/opt/local/bin/proctool` for graphical displays of system activity. These are, essentially, GUI interfaces to the other commands.
- `/usr/sbin/sar` for information comparable to that from /bin/vmstat.
- `/usr/lib/sa/{sadc,sa1,sa2}` for automatic data collection and report generation.

6 Summary

Good performance tuning is a non-trivial exercise and has to be done carefully and methodically. The following points must be kept in mind when attempting such an exercise.

- Look at the full spectrum of factors that impact performance. This runs the gamut from selection of an efficient algorithm for the application, to the vehicle used for outputting the final results.
- Collect and analyze all system activity information, including those pertaining to CPUs, memory management and disk I/O.
- Past experience may not be applicable because of the rapid advances in hardware, and the changes and improvements built into newer releases of the operating system.

7 References

1. *Sun Performance and Tuning: Sparc and Solaris*, by Adrian Cockcroft, SunSoft Press, 1995, Prentice Hall, ISBN 013-149642-3.
2. *System Performance Tuning*, by Mike Loukides, O'Reilly & Associates, ISBN 0-93-717560-9.
3. *The Magic Garden Explained*, by Berny Goodheart and James Cox, Prentice Hall, 1994, ISBN 013-098138-9
4. *The Art of Computer Systems Performance Analysis*, by Raj Jain, Wiley, ISBN 047-150336-3.
5. CS6400 and Solaris Answerbooks
   5a. CRAY SUPERSERVER 6400 Answerbook
   5b. Solaris 2.x Reference Manual Answerbook
   5c. Solaris 2.x System Administrator Answerbook
   5d. Solaris 2.x on Sun Hardware Answerbook