Performance Analysis and Tuning

CS5780 System Administration

Carol Thornborrow

November 12, 2003

One of the key parts in a system administrator's job is system performance analysis. It is the responsibility of the system administrator to monitor system behavior and identify any availability shortfalls. This responsibility includes matching system resources to current application and user requirements, tuning the system to correct any problems, and anticipating changing system requirements. It may seem like a system administrator should invest in a good crystal ball but there are many tools available to analyze the performance of a system. Some of them are even free or already included in the operating system.

Before a system administrator starts analyzing system performance a good question to start with is "What is system performance?" The best answer is – it depends on whom you ask.

Merriam-Webster Dictionary defines performance as

1 a : the execution of an action
  b : something accomplished : DEED, FEAT
2 : the fulfillment of a claim, promise, or request : IMPLEMENTATION
3 a : the action of representing a character in a play
  b : a public presentation or exhibition <a benefit performance>
4 a : the ability to perform : EFFICIENCY
  b : the manner in which a mechanism performs <engine performance>
5 : the manner of reacting to stimuli : BEHAVIOR
6 : the linguistic behavior of an individual : PAROLE; also
  : the ability to speak a certain language – compare COMPETENCE

Definitions 1, 2 and 5 seem to best describe system performance where the system is asked to do something and then produces a result. Most system administrators would use definition 4 to describe the performance of a system because they are measuring the ability of the hardware and software to efficiently satisfy the demands on the system. The Unix System Administration Handbook discusses what they call "perceived performance" with essentially the same definition. "Perceived performance is determined by the efficiency with which the system's resources are allocated and shared." However if you really want to know what "perceived performance" is, ask a user. You probably won't get the same answer. Users perceive the performance of a system as the response time of the system. The response time for the user is the amount of time they wait from when they first make the request to the appearance of the result via the screen, printer, etc. Management has a different perspective on performance. They normally see system performance within the context of a tradeoff between user perceived performance and return on investment. Poor system performance reduces productivity and may even result in the loss of business. A consensus between the user community, the information management team, including the systems administrator, and management on what is an acceptable level of system performance is needed to prevent confusion. A Service Level Agreement or SLA is normally written to govern the expectations of performance that the system will satisfy. The agreement will not only cover what is acceptable performance for normal system workload but also what is acceptable system performance for periods when system workload is heavy. Short periods of slow system performance may be agreed upon as acceptable as long as they are infrequent.

We now have a pretty good idea what system performance is at least from a system administrator's perspective. The next question to ask is what are the factors that affect system performance? Since system performance is perceived as the efficient allocation of the system's resources, there is a potential for many factors to contribute to the state of the system. Among these many factors there is a general consensus in system administration that there are four main factors which have historically affected performance to a degree of notation. The first factor is the CPU time or what the CPU is doing and when. A key part is the percentage of CPU time actually spent executing instructions or waiting for data. Do the instructions come from the user or are they system instructions? The second factor is memory or the internal data storage areas in the computer. The amount of memory and how it is used can have a major impact on system performance. The third factor is hard disk I/O bandwidth or the amount of data which can be transferred to or from the disk in a certain amount of time. The final factor is the network I/O bandwidth or the amount of data which can be transferred over the network in a certain amount of time. These factors contribute more to the performance of a system over others because they are shared among many users and/or systems. This leads to a greater impact globally upon the system.

The next question is how do these factors affect system performance. If you have ever heard a traffic report then you probably have heard the term "bottleneck". A bottleneck in traffic refers to the slowing of vehicles due to congestion on the roads. This can be caused by traffic that is heavier than normal or some event such as closure of lanes due to construction. If the situation continues traffic may eventually stop altogether. System bottlenecks are very similar. System bottlenecks occur when the demand for a resource is greater than the capacity or availability. Bottlenecks can also occur because of transient problems such as a process which is using up all the CPU time or if there is a hardware failure. Each contributing factor to system performance has its own bottlenecks. In addition, each factor presents a symbiotic relationship with the other factors. Each may present symptoms of a bottleneck that actually mask bottlenecks elsewhere. For example, a memory shortage can cause a high number of I/O requests so careful analysis needs to be done to determine the exact cause of the bottleneck.

To understand system performance we must examine each factor and its relationship with the other factors. The most commonly perceived performance factor is CPU time. CPU time affects system performance when processes do not receive enough slices of the CPU time to complete within a reasonable amount of real time. This can be caused by several different situations. There may be too many active processes vying for CPU time causing each of those processes to wait an extended period of time for the processor. There may be a process that is CPU-bound and is hogging all the CPU time or the user has set a nice value on the process to gain execution priority. The CPU may be spending all its time running user processes and not enough system processes or vice versa. There may be a misconfigured hardware device causing the kernel to issue too many context switches or commands to change the running process. This could cause processes which would normally have higher priority to be preempted for lower priority processes. It will also cause the system to slow because it increases the number of I/O requests. Another consideration is the applications running on the system may not fit the CPU environment. If the environment is a multi-processor environment but the applications do not parellelize then some CPUs may be overloaded while others remain idle.

There are two different types of memory that affect system performance, physical memory and virtual memory. Physical memory, often referred to as main memory or RAM, holds the data for all currently running processes. Virtual memory helps to manage physical memory by mapping the relative address of a part of a process to a physical memory location. This part of the process is only brought into physical memory when it is really needed. This allows the system to handle a greater number of running processes at the same time in main memory. Since main memory has a faster access rate than disk, processes can complete faster when their data is already loaded into memory. "Memory performance begins to affect overall system performance in two instances" (System Performance Tuning, 2nd Edition, O'Reilly). The first case actually occurs in two ways, when a system accesses physical memory frequently or when the system is too slow in retrieving the data from main memory. The second case occurs when there are too many running processes and physical and virtual memory are overwhelmed. The first case is either caused by improper algorithms, which control when memory is accessed or an improper main memory speed for system demands. Too little memory or an improperly loaded system causes the second case. In either case, the system will try to manage the physical memory by either paging parts of processes out to disk or swapping whole processes out to disk. As the number of pages or swaps increase other parts of the system, such as the CPU will be waiting on memory to write and read from the disk. If the load on the system is continually high then memory may enter a state called thrashing. As the system tries to fulfill the demand for physical memory from the active processes it exceeds the configuration limit on the least amount of available memory. The page scanner or algorithm then starts looking for pages to return to disk to maintain a certain amount of free memory. As the load remains high the page scanner runs more often and actually starts taking more than its share of system resources. The page scanner eventually starts swapping out whole processes to disk in attempt to keep active processes in memory to avoid a decrease in performance. This, however further slows the system by using more available I/O resources and making other parts of the system, possibly the CPU, wait for it to complete. Users feel the effects of this memory shortage through interactive processes such as editors. If a user pauses in typing their process might be swapped out to disk causing a delay when the user starts typing again. The user will have to wait until their process is retrieved from the disk. The retrieval of the user's process will be delayed even further by the system as it tries to page or swap other processes out to disk to free memory for the user's data. This is the point where the system administrator's phone starts ringing… and ringing… and ringing.

Disk I/O handles the requests for data to be retrieved from the hard disk. There are several factors that affect the performance of the available bandwidth to the hard disk. First and foremost is the seek time of the disk. Seek time is the time it takes for the head to travel over a disk to the position of the requested data. Seek time is the slowest part of accessing the disk and overshadows any gains from faster rotational speeds of the disk or quicker buses. It isn't going to matter how fast your disk rotates or what speed you can transfer data from the disk if the heads are only reading a minimal portion in the same amount of time.

Disk I/O and seek times are affected by the placement of the data on the disk or disks. When data cannot be accessed in sequential blocks the disk is fragmented. Fragmentation occurs when files that once occupied a certain amount of space are rewritten and no longer take up as much space. The disk reclaims these small amounts of space for use by other files. The space may not be large enough to hold the entire file data so parts of the file end up scattered over the

disk. The access time it takes to retrieve this file from the disk is slowed by the extra number of seeks and added latency. Slow disk I/O could also mean that large files, which are accessed often, are on a filesystem with a smaller blocksize or on a location of the disk that increases seek times.

The load also affects disk I/O or the number of I/O requests on the disk. If a disk is receiving a disproportionate number of I/O requests it might be caused by the type of filesystems stored on the disk. Certain filesystems are accessed more than others. The root filesystem for example receives a large amount of I/O requests. Placing the root filesystem on a disk with another filesystem, which also receives many I/O requests, could cause the disk I/O to slow considerably. The disk drive will be busy for a high percentage of its time while other drives are idle.

Hardware is another factor that can slow the performance of disk I/O. Disk I/O performance is only as fast as it slowest hardware component. If the speed on the disk is faster than the disk controller then the disk throughput will be decreased.

Depending on the size of the organization, there may be a network administrator who is responsible for the performance of the network or the system administrator may be responsible. Either way the system administrator needs to know what common bottlenecks occur on a network so they know when to involve the network administrator in a problem. Network bottlenecks occur when hardware is faulty or misconfigured, servers are overloaded, or if the network bandwidth is not enough to handle the workload. All of these bottlenecks generally produce the same effect on the network – slow response times.

Faulty or misconfigured hardware generates problems in a network such as the inability to connect to or access applications on the network. Misconfigured firewalls or faulty switches may keep authorized users from accessing the network.

Overloaded servers can cause slow network response to requests. The server can be overloaded due to several familiar reasons. The server interface may not be able to handle the traffic. The memory on the server may not be enough for the workload of the network. The disk I/O bandwidth may not be enough to handle I/O requests.

Slow response times and system timeouts are the symptoms of insufficient network bandwidth. Users may be "kicked" off the network while trying to complete processes.

So now the system administrator knows some common problems they will generally handle in system performance. When a problem arises it is the system administrator's responsibility to diagnose a cause and implement or suggest a solution. As a doctor can monitor a person's heart rate and blood pressure for irregularities, a system administrator can monitor factors in the system to create a baseline for comparison when poor performance occurs. This baseline will paint a picture of the system under acceptable performance conditions. When a system administrator is well informed about the performance of their system over time then they are better prepared to isolate system bottlenecks. Monitoring the system will aid in identifying potential system bottlenecks before they become a problem. If a system administrator notices a process which consumes most of a certain resource, say the CPU, they may be able to reschedule the job to run during off peak hours. Persistent monitoring of the system will also help in projecting a system's future needs. Indicators such as an increase in the percentage of time that disk I/O is busy can help predict replacement needs.

So how does a system administrator monitor a system to compare metrics of good performance to degraded performance. There are all kinds of tools available to the system administrator to monitor a system and the operating system even provides some of them. Common utilities such as vmstat or sar, iostat, ps, and uptime might report slightly different output and takes different options dependent upon the UNIX variant such as BSD or System V. The system administrator will need to check the man pages for specifics if they plan on using some of these tools. Performance tracking and reporting programs can be purchased from vendors and usually include some kind of graphical interface. Unless the service level agreement specifies what and how often monitoring will take place this is left up to the system administrator. How critical a resource is to the business may determine what type and how often the resource is monitored. If the server the system administrator is monitoring runs the online payment software for a credit card company then it may need more than just the performance monitoring utilities included in the operating system. Some of the factors a system administrator will need to address is how quickly is a solution needed and what is the budget. The system administrator will also need to keep in mind that any performance-monitoring tool will use resources on the system. Tools that use windows based or graphical interfaces will normally use more resources than text based tools. If the tool to be used generates a lot of data then the cost of storing the data needs to be considered.

One of the first performance monitoring tools that may be useful is uptime. Uptime reports the current time, the cumulative time since the last reboot, the number of current users and the average number of processes in the run queue in the last five, ten and fifteen minutes.

1:49pm up 153 day(s), 2:25, 8 users, load average: 0.00, 0.01, 0.02

Tracking the load average as part of the system baseline is a good reference for comparison. If the system load average is still within the system's historically normal limits when performance is degraded then the problem may very well be outside of the system.

The overall utilization of the system can be reported by using vmstat. This utility is useful in diagnosing problems such as CPU bottlenecks or misconfigured hardware.

vmstat 5 5

procs			memory		page						disk				faults			cpu		
r	b	w	swap	free	re	mf	pi	po	fr	de	sr	dd	dd	f0	--	in	sy	cs	us	sy	id
0	0	0	512392	38672	7	39	2	0	0	0	0	1	0	0	0	310	345	81	4	5	91
0	0	0	463072	35584	0	2	0	0	0	0	0	0	0	0	0	306	20	70	0	0	100
0	0	0	463672	35904	50	106	0	0	0	0	0	0	1	0	0	318	123	70	3	2	95
0	0	0	463416	35776	4	27	0	0	0	0	0	1	1	0	0	314	75	79	0	1	98
0	0	0	463432	35784	2	16	0	0	0	0	0	0	0	0	0	314	66	81	0	0	99

Some useful metrics here are the number of runnable processes waiting for the CPU designated by the r field under procs. If there are a high number of processes in this column then there may be a process that is hogging all the CPU time.

Another metric under procs to watch is the w field or the number of processes that are runnable but have been swapped out to disk for some reason. A non-zero value in this column could indicate a memory shortage under current system load. Two other indicators of memory problems are the free column under memory and the de column under page. The free column reports amount of kilobytes of memory on the system's free list. Some computation may need to be done but another general rule of thumb is if this number is consistently below three percent of total memory then the system is not maintaining enough free memory. A system administrator might see high numbers of pages in (pi) and out (po) or even a swap (w) as the system tries to

free memory for processes. The de column is the number of kilobytes of "predicted short-term memory shortfall" and is another indicator of serious memory problems. Values above 100 in this column indicate memory shortages.

The value in the cs column under faults reports the number of context switches. Context switches are initiated by the kernel to change which process is currently running on the CPU. A high value in this column could mean that a hardware device is incorrectly configured.

The us, sy and id columns under cpu are the percentages of time that the CPU spends in the user, system and idle states. The balance between these metrics is a good indicator of what the CPU is doing. If the CPU idle time is consistently less than 20 percent then the system is starting to be bound by the CPU. The system administrator will also be able to tell if the CPU is spending too much time responding to system calls by the percentage of time in the sy column. A general rule of thumb is the CPU should spend 50% of its time or less in the system space.

Other useful utilities are ps or top if you need to track down any active processes that are using up too much CPU or memory. Iostat is another useful utility and is normally used to monitor disk I/O. A metric in iostat to watch is the time the drive is active (%tm_act). A general rule of thumb is if this metric exceeds 35% then the system may be I/O bound. A network reporting utility is netstat.

A utility that comes standard on most Unix systems is sar. Sar allows the system administrator to automate the reporting of system activity through the system crontab file. A system administrator can use this to research past performance issues. Sar can also be run interactively to report on current system performance.

There are many third party programs available to monitor performance. If the resource is critical to the business this may be a consideration for the system administrator to obtain real-time data. Some vendors such as Hewlett-Packard offer a variety of monitoring tools such as GlancePlus, which is an online monitoring system or Sarcheck. Sarcheck not only analyzes the output from the system utilities such as sar and ps it then goes on to recommend changes to the system to correct potential system bottlenecks.

After identifying what is the probable bottleneck of the system, what can the system administrator do to tune the system to prevent or correct bottlenecks. First and foremost the system administrator needs to know what they are doing. Without an understanding of what affect a change will have on the system the change may do more harm than good. The system administrator may need to consult with the hardware or software vendor before implementing any changes. The vendors may even have a solution to the problem available.

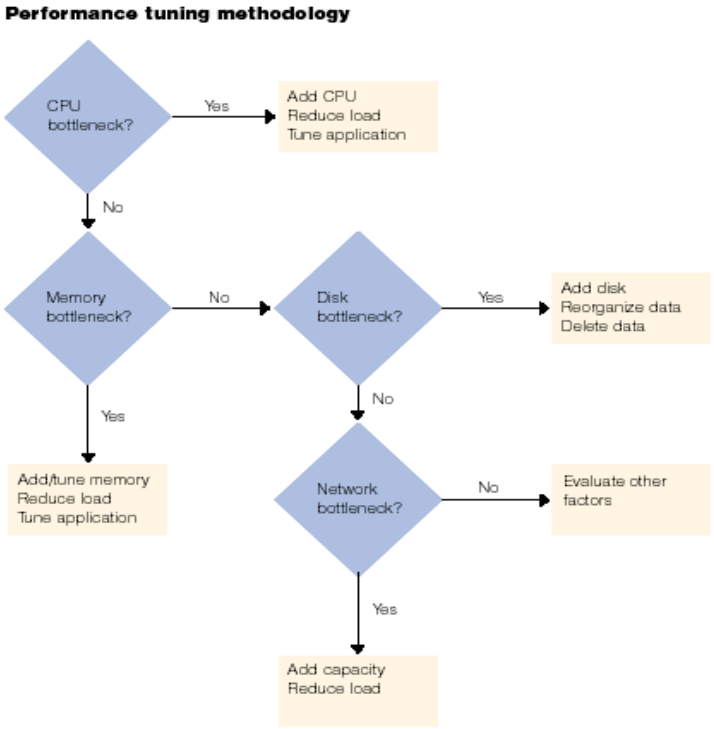
Tuning also takes time and patience. Changing one part of the system may not increase performance to an acceptable level or actually make the performance worse. Further changes may need to be made. System administrators will need to track these changes and be able to recover the system to its original state if necessary.

If the system administrator does make changes to tune one part of the system they need to be aware of the consequences on other parts of the system. If the system administrator uses filesystem striping and high-speed disk controllers to improve the I/O performance, the CPU and memory may become overloaded by data volume that the I/O system is now able to transfer. By solving the bottleneck in one area the system administrator may actually create another one in another area.

The system administrator also needs to consider the cost of the recommended change. Replacing a CPU to increase processor speed may cause a whole machine to be replaced and

possibly even an operating system upgrade. Tuning applications may not really be cheaper to gain memory if it consumes too much of the system administrator's and/or programmers time when purchasing more memory is relatively cheap. Sometimes the cost of in-depth analysis and tuning is more expensive than purchasing hardware.

IBM has developed a general methodology to assist in guiding performance tuning decisions.



If the system administrator determines that the CPU needs replaced by a faster CPU there are some considerations to take into account. The entire machine may need to be replaced if a new CPU would overload other parts of the system such as the system bus. The CPU upgrade may also require a newer and possibly slower version of software. Another consideration is adding another CPU to spread the workload out. This is possible if the applications can multithread or parellelize but be careful of the overhead caused by synchronization between the CPUs. The additional CPU may actually slow the system more than improve performance.

The load on the CPU may be able to be reduced by moving some applications or users to another system if the CPU on that system has a lighter load. Testing will need to be done to assure the system administrator is not moving the CPU bottleneck to another system.

Another possibility is to examine and tune the applications run on the system in an effort to distribute the workload. Applications which can be moved to off peak hours should be run by cron jobs. Other applications, which may not need to run as quickly, will benefit performance by lowering their priority or running in batch-queues. The applications themselves may be able to be tuned to run more efficiently. If a system administrator notices a high number of context switches when a particular application is running then the application may be generating many system calls. If the application is "in house" a review of the code might produce a more efficient program. Other applications may benefit from being moved to a different filesystem. An

application that generates many I/O requests may be more efficient if it is moved from an NFS filesystem.

If the system seems to be short on memory then one possible solution is to add more since it is relatively inexpensive. Increasing the amount of memory may cause the system to be I/O bound as it tries to synchronize large amounts of memory at designated intervals. System administrators may want to check their flushing daemons to see if there is a way to spread this workload out. Solaris and Linux accomplish this using fsflush and bdflush along with autoup. Increasing autoup divides the memory in smaller portions to help alleviate I/O demands.

If a disk I/O bottleneck is being caused by an unbalanced load on the disk drives then it is prudent to balance the load by reorganizing the data, eliminating fragmentation or by filesytem striping if available. All these methods aim to reduce the seek times and access times on the disk.

A system administrator can gain performance by examining the locations of swap space. Swap space placed on slow disks or controllers or on a disk with heavy I/O activity from other sources is only slowing the performance of the system. Swap space should also not be placed on NFS disks as this slows performance by requiring the I/O to go through filesystem commands instead of writing directly to the disk.

Another way to reorganize the disk is to separate busy filesystems from other busy filesystems thus reducing the number of I/O requests on the disk. The root system receives many requests and performs best when placed on the fastest disk with the fastest controller.

Large sequential files should be placed on filesystems with large block sizes to reduce the seek times. Files which are commonly accessed should not be stored where a large amount of directories will need to opened to access them. Each directory that needs to be opened increases seek time as the head needs to move to another part of the disk.

Fragmentation can be managed by running fsck to consolidate parts of files spread over the disk. A filesystem is normally unmounted for fsck to run so plan this operation during scheduled downtimes or during system reboots.

If filesystem striping is available then filesystems may be organized to take advantage of the parallel access created from storing them on multiple disks and multiple controllers.

Disks and controllers may be upgraded or added to reduce the workload on present disks or to increase disk speeds.

Tuning is not normally recommended in the kernel. Adjusting kernel variables may actually slow the performance of a system under certain conditions as most kernels are already primed for reasonable performance under most conditions. If the system administrator finds himself or herself needing to adjust the kernel then backing up the current kernel is highly recommended so it may be restored if the changes are detrimental. A backup of the filesystem is also recommended in case of corruption.

Everything discussed so far has been referenced in the system framework that the system administrator can control. However, today's systems are much more complex and the system administrator may find problems arise on their system due to factors out of their control. Many local systems are connected to other networks. The local system's performance is affected by this network and any other systems included on the network. If the system services user locations in New York, Florida and California but the California site is the only one experiencing a problem then it may be an ISP network problem, which services the connection to California, rather than a problem with the local system. The system

administrator is the one who is still going to get the call. The system administrator must then contact their service vendor and report the problem.

A good system administrator will set up guidelines to keep on top of and ahead of most system performance issues. Some general guidelines are

a. Match the system to the purpose, number of users, application types, and amount of estimated disk space
b. Keep the system current with any vendor patches
c. Know what "normal" system performance so "abnormal" system performance stands out
d. Monitor system performance so you can prevent problems
e. Upgrade the system when necessary

These guidelines will help keep a system administrator out of hot water, most of the time.

References

System Performance Tuning, 2nd Edition, Musumeci, Louikides, O'Reilly

Unix System Administration Handbook, Third Edition, Nemeth, Snyder, Seebass, Hein, Pearson Education

Essential System Administration, Frisch, O'Reilly

System Performance Concepts, InSPEC Administration Manual, Elegant Communication, Inc

Addressing UNIX and NT server performance, IBM