Abstract

Performance analysts should invest their time and expertise in analyzing system performance rather than spending their time inventing or learning new tools. SAS/CPE® Software can simplify collection and management of performance data from multiple platforms, including OpenVMS, Open Systems and MVS.

Examples of analyzing CPU, memory, I/O and network utilization on each platform will be shown as well as combining data for some general management reports. Also included will be pointers to specific features of the SAS System, such as client-server support, which can make the job of performance analysis easier.

Introduction

In the multiple-platform architecture of the 90's, the job of the performance analyst has become much more complex. Analysts are now often called upon to analyze data from operating systems that are relatively new to them. If you are a performance analyst who is asked to become an instant expert on multiple platforms, you might reasonably have qualms. With the right tools for managing and analyzing performance data, however, you’ll at least be able to spend more of your time in analytical thought and less time re-inventing the wheel.

All performance statistics can be broken down into two major categories:

- those that are the same or quite similar across the platforms; and
- those that aren't!!

This paper will discuss the major performance categories generally with specific examples drawn from each of the operating systems covered (MVS, OpenVMS and UNIX). The discussion of each category will clarify the types of statistics that are comparable across systems, and those that aren’t. Each of the examples shown was, however, generated using the same software tool, SAS/CPE® Software.

One of the advantages of using SAS/CPE® Software is its data reduction features. SAS/CPE® provides utilities to reduce detail raw data into organized formats in DAY, WEEK, MONTH, and YEAR levels. In addition to saving on disk space, this also allows the analyst access to data covering a very long time period in a timely and efficient manner.

CPU

The CPU is the central resource on any platform, and is typically the first one studied because it allocates and initiates the use of the other resources. It is also the most expensive to upgrade on most platforms.

The simplest analyses of CPU utilization are not platform-specific. Concepts such as

- percent CPU idle
- peak CPU hours
- CPU consumption by process or user
- processes waiting for CPU time
- unmet CPU demand

can all be used appropriately on any system.

For example, the analyst can judge unmet CPU demand in a number of ways on many platforms. One advantage of keeping reduced data over time can be shown from this CPU utilization graph showing prime shift CPU utilization over time (figure 1).

The sharp drop happened when a new processor was installed at the site, but due to latent demand the usage climbed back up almost immediately.

![Percent CPU Busy on MVS](image)
Another measure of CPU activity that can be used on all of these systems is the number of processes that are waiting for CPU. On MVS, this is the number of address spaces that are ready to be run, but are not currently running in the system (figure 2). On other systems, the same concept is referred to as the computable queue or the run queue.

When the analyst finds a high number of processes waiting for CPU, the next diagnostic step is probably system-dependent. On MVS in the PRISM environment the analyst might find that the average CPU busy is low while the number of delayed address spaces is high during the same time period, and would want to further investigate by looking for contention from the other LPARs.

Other differences between systems occur in the less obvious, and sometimes more useful, detailed analyses. These include system-specific details or differences in:

- what the CPU's doing while it's busy
- processor-specific data on multiprocessor systems
- use of vector processing
- granularity of statistics

For example, on UNIX, the HP PCS (Performance Collection Software) collector provides CPU percent busy metrics broken down by usages: system, user, context switch, interrupt handling, memory managing, etc. (These CPU usage related metrics are stored in the table PCSGLB.) These divisions of CPU use are not available from some other collectors or platforms.

On MVS, one way to analyze who is using the CPU time in your system is by looking at workload activity. Assuming that performance groups have been set according to the major system workloads, RMF type 70 and 72 data can be used to show CPU time by workload as in figure 3.

OpenVMS divides processor utilization by mode. The relative proportion of time spent in the various modes can tell the analyst a lot: for example, if the CPU is spending more than around 8% of the time in multiprocessor synchronization, then CPU cycles may be unnecessarily spent on synchronization due to high paging or locking rates. Other OpenVMS-specific statistics include a finer granularity on computable queue statistics and on voluntary and involuntary wait states in general.

Memory

On all of these platforms, memory can be considered a single resource pool shared by all processes (and the system itself). All of the following are available on each of the systems:

- percent memory utilization
- peak memory utilization
- paging and swapping rates
- memory use or paging by process or user

An example of a daily report produced on MVS paging and swapping is seen in figure 4, but similar reports could have been generated for any of the operating systems covered.
However, the more detailed analyses often involve system-specific concepts. For example, the granularity of data on:

- demand-sensitive paging
- memory-caused wait states
- types of page faults and page caches
differs widely among the operating systems covered.

On MVS, an indication of storage usage is the high system unreferenced interval count (UIC). Its value indicates the age in seconds of the oldest frame in central storage that has gone unreferenced. Low UICs may mean that you have a storage bottleneck.

When analyzing storage usage on MVS, you need to look at virtual storage usage, central storage usage and expanded storage usage. Expanded storage is a high speed area used as intermediate storage between central storage and auxiliary storage (DASD). Even though it is fast, there is a cost in paging between central storage and expanded storage (figure 5).

Another case where OpenVMS offers detailed and system-specific data is in the analysis of process wait states. In figure 7, we see the number of processes in each memory-related wait state.

For I/O, the key performance issues on any platform are the same:

- the number of I/O operations
- I/O response time
- cache rates (attempts, hits, misses)
- capacity (space) utilization
- unmet I/O demand (I/O queue)

For example, the cache hit rate graph shown in figure 8 could have been generated for other cache types on other systems. Examining cache success/failure rates can lead to fewer I/Os by adjusting cache sizes.
On OpenVMS, for example, we can establish exactly how many of the I/Os were due to page fault reads and writes, swapping, and which were due to the various types of file I/O activity. We can also get a wealth of detail on I/O operations and buffering on a file-specific basis.

Disk space consumption can be tracked on OpenVMS by using the Disk usage utility, which is included as part of the SAS/CPE software package on OpenVMS. This enables the analyst to easily track the use of disk space by user (figure 9) and also offers other details, such as being able to list files larger than a given size, and the number of header extents, which is a measure of disk fragmentation.

On MVS, we could look at an analysis of I/O response time by DASD volumes. These response times are usually broken out by their components: response, connect, disconnect, IOS (I/O Subsystem) Queue and pending time. This level of detail is not currently available on other platforms.

Network

Compared to the other performance topics already noted, you might expect that network performance analysis is surely not linked to specific operating systems. This is mostly correct, in that both network jargon and typically used statistics relate more to the type of network than the systems present on that network. Concepts such as:

- total network utilization
- utilization by protocol and packet types
- network area and node topology
- network line errors

are not system-specific and interpreted the same across platforms (figure 10). However, the systems do vary according to how much network data is supplied by collectors available on those systems, and by the types of networks and protocols generally available.

Excellent network statistics are available on Open Systems from such collectors as HP OpenView. It includes network information such as packet error rate, packet collision rate, inbound packet count, and outbound packet count (in table HN2IFT).

On MVS, network performance information can be gathered from products such as Netspy (from Legent). Data such as network line errors can be monitored to ensure that line errors remain low. Other useful information includes outbound queue length and communications controller processor utilization.

Although the OpenVMS MONITOR and PS collectors do offer some statistics on DECnet network use, the Etherwatch utility, offered as part of the SAS/CPE product on OpenVMS, gives a lot of data on network traffic. For example, you can obtain complete node-pair link information which can show you which nodes are transmitting the most traffic (figure 11). You can also isolate DEC-specific traffic such as DECnet and LAT protocol traffic.

Figure 9  Top 10 Disk Space Users

Figure 10  Network Activity

Figure 11  Top 10 Node Pairs
Combining Data Across Platforms

Although performance analysts frequently concentrate on one specific system or group of similar systems at a time, analysts also need to be able to generate reports with data gathered from many different platforms. The features in SAS/CPE which allow easily combining tables and formulating new variable assists the analyst in making these kinds of cross-platform analyses.

For example, management may wish to identify the top disk space consumers from all multiuser platforms. In this case, the data from the SAS/CPE product and programs provided in the SAS System can be used to combine tables from multiple collectors and multiple platforms. For disk space consumption on a per-user basis, the analyst can combine these tables:

- DIUIC: Per-user disk space consumption from the DISK USAGE collector on OpenVMS
- UUDCV: Disk space usage from DCOLLECT on MVS (using generic collector)
- UUQUOT: Disk space usage from the quot -f command (using the generic collector)

For any cross-platform combination, of course, we need to make sure that the statistics are as comparable as possible. In some cases this requires that the analyst use the formula variable feature of SAS/CPE to translate the statistic given by the collector to a more generic number.

For analyzing disk space consumption, the DISK USAGE collector records space in terms of 512-byte blocks, but the DCOLLECT and quot collectors record space in terms of 1024-byte blocks; so the analyst must translate these figures to bytes or megabytes in order to meaningfully combine them.

The analyst can easily construct a new formula variable for each of the three tables of interest. For example, 'MBYTES = BLOCKS/2;' translates OpenVMS Disk usage disk-block consumption to megabytes. The SAS/CPE table combination feature can then merge data from the three tables into one. Similar techniques can be used to compare I/O operations and memory consumption across platforms. However, the analyst will need to carefully assess the meaning of these comparisons.

Client/Server

The client/server paradigm concerns the intelligent division of computing labor among computers. The mainframe is a great platform for storing and processing sheer volumes of data, such as found in CPE, but has a rather poor GUI (Graphical User Interface). PCs and workstations, on the other hand, typically have better GUIs but are rather poor platforms for processing huge volumes of data.

Since the SAS/CPE software product is part of the SAS System, users can use tools provided with SAS to divide computing labor intelligently among different computing platforms. For example, if your performance database (PDB) resides on an MVS platform and contains multiple gigabytes of data, you can generate reports for it using SAS/ASSIST Software on the PC to generate the reports and using SAS/CONNECT Software to provide access to the data.

Conclusion

Computer performance analysis is an intrinsically complex job. One way to make analysts more productive is to equip them with tools such as SAS/CPE that offer portable, useful tools for managing and analyzing data from a variety of operating systems. Other SAS features and products can be used in conjunction with SAS/CPE to expand the variety of tools offered to the performance analyst.