Performance Tuning — The vmstat Tool

Special Notices

Information in this document is correct to the best of our knowledge at the time of this writing. Please send feedback by fax to "AIXServ Information" at (512) 823-4009.

Please use this information with care. IBM will not be responsible for damages of any kind resulting from its use. The use of this information is the sole responsibility of the customer and depends on the customer's ability to evaluate and integrate this information into the customer's operational environment.

About this Document

This document provides an overview of the output of the vmstat command and is applicable to AIX version 3.2 - 4.2.

Introduction

Although a system may have sufficient real resources, it may perform below expectations if logical resources are not allocated properly.

Use vmstat to determine real and logical resource utilization. It samples kernel tables and counters, then normalizes the results and presents them in an appropriate format.

By default, vmstat sends its report to standard out, but it can be executed with the output redirected.

vmstat is normally invoked with an interval and a count specified. The interval is the length of time in seconds over which vmstat is to gather and report data. The count is the number of intervals to run. If no parameters are specified, vmstat will report a single record of statistics since the system was booted. There may have been inactivity or fluctuations in the workload so the results may not represent current activity. Be aware that the first record in the output presents statistics since boot (except when invoked with the -f or -s option). In many instances, this data can be ignored.

vmstat reports statistics about processes, virtual memory, paging activity, faults, CPU activity, and disk transfers. Options and parameters recognized by this tool are indicated by the usage prompt:

```
vmstat [-fs] [Drives] [Interval] [Count]
```

The figure listed below lists output for AIX 3.2.5 where the smallest work unit is called a process (PROC). In 4.1 the smallest work unit is a thread (KTHR). The "r" and "b" columns exists in both versions. The end result is that in the 4.1 output, the "r" and "b" columns represent the number of "threads" not processes placed on these queues.

```
procs   memory          page            faults          cpu
-----  --------   ----------------- -------------  -----------
r  b   avm  fre   re pi po fr sr cy  in   sy   cs   us sy id wa
0  0   6747 1253   0  0  0  0  0  0  114  10   22   0  1  26 0
1  0   6747 1253   0  0  0  0  0  0  113  118  43   17 4  79 0
0  0   6747 1253   0  0  0  0  0  0  118  99   33   8  3  89 0
```

Figure 1. Sample Output from "vmstat 1 3"

procs

The columns under the "procs" heading in the output provide information about the average number of processes on various queues.

r

The "r" column indicates the average number of processes on the run queue at one-second intervals. In 4.1 the "r" column indicates the number of kernel threads placed in run queue.

This field indicates the number of "run-able" processes. The system counts the number of ready-to-run processes once per second and adds that number to an internal counter. vmstat then subtracts the initial value of this counter from the end value and divides the result by the number of seconds in the measurement interval. This value is typically less than five with a stable workload. If this increases rapidly you should probably look for an application problem. A consistently high number may indicate that the CPU is pacing the system. If there are many processes (especially CPU-intensive ones) competing for the CPU resource, it's quite possible they will be scheduled in round-robin fashion. If each one executes for a complete or partial time slice, then the number of "run-able" processes could easily exceed 100.

b

The "b" column shows the average number of processes on the wait queue at one-second intervals. In 4.1 the "b" column indicates the number of kernel threads placed in wait queue (awaiting resource, awaiting input/output).

Processes are placed on the wait queue when scheduled for execution waiting for one of their process pages to be paged in. Once a second the system counts the processes waiting and adds that number to an internal counter. vmstat then subtracts the initial value from the end value and divides the result

by the number of seconds in the measurement interval. This value is usually near zero. Don't confuse this with "wa" — waiting on input/output (I/O).

memory

The information under the "memory" heading provides information about real and virtual memory.

avm

The avm column gives the average number of pages allocated to paging space. (In AIX, a page contains 4,096 bytes of data.)

When a process executes, space for working storage is allocated on the paging device(s) (backing store). This can be used to calculate the amount of paging space assigned to executing processes. The number in the avm field divided by 256 will yield the number of megabytes (MB) allocated to page space, systemwide. The "lsps -a" command will also provide information on individual paging space. It is recommended that enough paging space be configured on the system so that the paging space used does not approach 100 percent. When fewer than 128 unallocated pages remain on the paging device(s), the system will begin to kill processes to free some paging space.

fre

The "fre" column shows the average number of free memory frames. A frame is a 4,096 byte area of real memory.

The system maintains a buffer of memory frames, called the free list, that will be readily accessible when the Virtual Memory Manager (VMM) needs space. The nominal size of the free list varies depending on the amount of real memory installed. On systems with 64 MB of memory or more, the minimum value (MINFREE) is 120 frames. For systems with less than 64 MB, the value is two times the number of MB of real memory, less eight. For example, a system with 32 MB would have a MINFREE value of 56 free frames.

If the fre value is substantially above the MAXFREE value (which is defined as MINFREE plus eight), then it is unlikely that the system is thrashing (continuously paging in and out). However, if the system is experiencing thrashing, you can be assured that the fre value will be small. Most UNIX and AIX operating systems will use nearly all available memory for disk caching, so you need not be alarmed in the event the fre value oscillates between MINFREE and MAXFREE.

page

The information under the "page" heading includes information about page faults and paging activity.

re

The "re" column shows the number (rate) of pages reclaimed.

Reclaimed pages can satisfy an address translation fault without initiating a new I/O request (the page is still in memory). This includes pages that have been put on the free list but are accessed again before they are reassigned. It includes pages previously requested by VMM for which I/O has not yet been completed or those pre-fetched by VMM's read-ahead mechanism but hidden from the faulting segment.

pi

The "pi" column details the number (rate) of pages paged in from paging space.

Paging space is the part of virtual memory that resides on disk. It is used as an "overflow" when memory is over-committed. Paging consists of paging logical volumes dedicated to the storage of working set pages that have been stolen from real memory. When a stolen page is referenced by the process, a page fault occurs and the page must be read into memory from paging space. There is no "good" number for this due to the variety of configurations of hardware, software, and applications.

Some analysts have said that five page-ins per second should be the upper limit. This theoretical maximum should not be rigidly adhered to but used as a reference. This field is important as a key indicator of paging space activity. Look at it this way: if a page-in occurs, then there must have been a previous page-out for that page. It is also likely in a memory constrained environment that each page-in will force a different page to be stolen and, therefore, paged out.

po

The "po" column shows the number (rate) of pages paged out to paging space.

Whenever a page of working storage is stolen, it is written to paging space. If not referenced again, it will remain on the paging device until the process terminates or disclaims the space. Subsequent references to addresses contained within the faulted-out pages result in page faults, and the pages are paged in individually by the system. When a process terminates normally, any paging space allocated to that process is freed. If the system is reading in a significant number of persistent pages, you may see an increase in po without corresponding increases in pi. This does not necessarily indicate thrashing, but may warrant investigation into data access patterns of the application(s).

fr

The "fr" column details the number (rate) of pages freed.

As the VMM page-replacement code routine scans the Page Frame Table (PFT), it uses criteria to select which pages are to be stolen to replenish the free list of available memory frames. The total pages stolen — both working (computational) and file (persistent) pages — by the VMM are reported as a rate per second. Just because a page has been freed, it does not mean that any I/O has taken place. For example, if a persistent storage (file) page has not been modified, it will not be written back to the disk. If I/O is not necessary, minimal system resources are required to free a page.

sr

The "sr" column details the number (rate) of pages scanned by the page-placement algorithm.

The VMM page-replacement code scans the PFT and steals pages until the number of frames on the free list is at least the MAXFREE value. The page-replacement code may have to scan many entries in the Page Frame Table before it can steal enough to satisfy the free list requirements. With stable, unfragmented memory, the scan rate and free rate may be nearly equal. On systems with multiple processes using many different pages, the pages are more volatile and disjointed. In this scenario, the scan rate may greatly exceed the free rate.

cy

The "cy" column provides the rate of complete scans of the Page Frame Table.

cy shows how many times (per-second) the page-replacement code has scanned the Page Frame Table. Since the free list can be replenished without a complete scan of the PFT and because all of the vmstat fields are reported as integers, this field is usually zero.

faults

The information under the "faults" heading in the vmstat output provides information about process control.

in

The "in" column shows the number (rate) of device interrupts.

This column shows the number of hardware or device interrupts (per second) observed over the measurement interval. Examples of interrupts are disk request completions and the 10 millisecond clock interrupt. Since the latter occurs 100 times per second, the in field is always greater than 100.

sy

The "sy" column details the number (rate) of system calls.

Resources are available to user processes through well-defined system calls. These calls instruct the kernel to perform operations for the calling process and exchange data between the kernel and the process. Since workloads and applications vary and different calls perform different functions, it is impossible to say how many system calls per-second are too many.

cs

The "cs" column shows the number (rate) of context switches.

The physical CPU resource is subdivided into logical time slices of 10 milliseconds each. Assuming a process is scheduled for execution, it will run until its time slice expires, is preempted, or it voluntarily gives up control of the CPU. When another process is given control of the CPU, the context, or working environment, of the previous process must be saved and the context of the current process must be loaded. AIX has a very efficient context switching procedure, so each switch is inexpensive in terms of resources. Any significant increase in context switches should be cause for further investigation.

---

cpu

The information under the "cpu" heading in the vmstat output provides a breakdown of CPU usage.

us

The "us" column shows the percent of CPU time spent in user mode.

Processes execute in user mode or system (kernel) mode. When in user mode, a process executes within its code and does not require kernel resources to perform computations, manage memory, set variables, etc.

sy

The "sy" column details the percent of CPU time spent in system mode.

If a process needs kernel resources, it must execute a call and go into system mode to make that resource available. I/O to a drive, for example, requires a call to open the device, seek, and read/write data. This field shows percent of time the CPU was in system mode. Optimum use would have the CPU working 100 percent of the time. This holds true in the case of a single-user system with no need to share the CPU. Generally, if us+sy time is below 90 percent, a single-user system is not considered CPU constrained. However, if us+sy time on a multi-user system exceeds 80 percent, the processes may spend time waiting in the run queue. Response time and throughput might suffer.

id

The "id" column shows time (percent) the CPU is idle with no pending disk I/O.

If there are no processes available for execution (the run queue is empty), the system dispatches a process called wait. The ps report (with the -k or g option) identifies this as kproc with a process ID (PID) of 514. In 4.1 the PID of the wait process was changed from 514 to 516. Don't worry if your ps report shows a high aggregate time for this process. It means you have had significant periods of time when no other processes could run. If there are no I/Os pending to a local disk, all time charged to "wait" is classified as idle time.

wa

The "wa" column details CPU idle time (percent) with pending local disk I/O.

If there is at least one outstanding I/O to a local disk when "wait" is running, the time is classified as "waiting on I/O." A wa value over 40 percent could indicate that the disk subsystem may not be balanced properly, or it may be the result of a disk-intensive workload. If there is only one process available for execution — often the case on a technical workstation — there may be no way to avoid waiting on I/O.

Summary Statistics

vmstat with the -s option reports absolute counts of various events since the system was booted. There are 23 separate events reported in the "vmstat -s" output; here are four that have proven most helpful. The 19 remaining fields contain a variety of activities from address translation faults to lock misses to system calls. The information in those 19 fields is also valuable, although less frequently used.

page ins

The "page ins" field shows the number systemwide page-ins.

When a page is read from disk to memory, this count is incremented. It is a count of VMM-initiated read operations and, with the page outs field, represents the real I/O (disk reads/writes) initiated by the VMM.

page outs

The "page outs" field shows the number of systemwide page-outs.

When a page is written to the disk, this is count incremented. It is a total count of VMM-initiated write operations and, with the page ins field, represents the total amount of real I/O initiated by the VMM.

paging space page ins

The "paging space page ins" field is the count of **only** pages read from paging space.

paging space page outs

The "paging space page outs" field is the count of **only** pages written to paging space.

Using the Summary Statistics

The four fields above can be used to indicate how much of the system's I/O is for persistent storage. If the value for paging space page ins is subtracted from the (systemwide) value for page ins, the result will be the number of pages that were read from persistent storage (files). Likewise, if the value for paging space page outs is subtracted from the (systemwide) value for page outs, the result will be the number of persistent pages (files) that were written to disk.

Remember, these are counts since system initialization. If you need counts for a given time interval, execute "vmstat -s" at the time you want to start monitoring and again at the end of the interval. The deltas between like fields of successive reports will be the count for the interval. It is easier to redirect the output of the reports to a file and then perform the math. The remaining fields produced by the s, f, and Drives options of vmstat are fully documented in InfoExplorer and the AIX Tuning Guide, AIX Version 3.2 for RISC System/6000 — Performance Monitoring/Tuning Guide, publication number SC23-2365.

Reader's Comments

Please fax this form to (512) 823-4009, attention "AIXServ Information". You may also e-mail comments to: elizabet@austin.ibm.com. These comments should include the same customer information requested below.

Use this form to tell us what you think about this document. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

If you need technical assistance, contact your local branch office, point of sale, or 1-800-CALL-AIX (for information about support offerings). These services may be billable. Faxes on a variety of subjects may be ordered free of charge from 1-800-IBM-4FAX. Outside the U.S. call 415-855-4329 using a fax machine phone.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Note: If you have a problem report or item number, supplying that number may help us determine why a procedure did or did not work in your specific situation.

Problem Report or Item #: Branch Office or Customer #:

Be sure to print your name and fax number below if you would like a reply:

Name: Fax Number:

END OF DOCUMENT (vmstat.perf.tune.lxp, 4FAX# 6220)