Diagnosing Performance Bottlenecks for AIX 3.2 and 4.1

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 describes how to check for resource bottlenecks and identify the process(es) that hog them. Resources on a system include memory, cpu, and Input/Output (I/O). This document covers bottlenecks across an entire system and is applicable to AIX versions 3.2 and 4.1. This document does not address the bottlenecks of a particular application or general network problems. The following commands are described:

- vmstat
- svmon
- ps
- tprof
- iostat
- netpmon
- filemon

Note: At AIX 4.1, PAIDE/6000 must be installed in order to use tprof, svmon, netpmon, and filemon. To check to see if this is installed, type:

```
lslpp -l perfagent.tools.
```

To order PAIDE/6000:

- In the U.S. call IBM DIRECT at 1-800-426-2255 or contact your local IBM representative.
- Other countries contact your local IBM representative.

This fax also makes reference to the vmtune and schedtune commands. These commands and their source are found in the /usr/lpp/bos/samples (3.2)

or /usr/samples/kernel (4.1) directory. They are installed with the bosadt.lib.obj (3.2) or bos.adt.samples (4.1) fileset.

Memory Bottlenecks

The following section describes memory bottleneck solutions with the following commands: vmstat, svmon, ps

- vmstat

Run the following command:

```
vmstat 1
```

Note: System may slow down when PI and PO are consistently non-zero.

pi number of pages per second that are paged in from paging space

po number of pages per second that are paged out to paging space

When processes on the system require more pages of memory than are available in RAM, working pages may be paged out to paging space and then paged in when they are needed again. Accessing a page from paging space is considerably slower than accessing a page directly from RAM. For this reason, constant paging activity can cause system performance degradation.

Note: Memory is over-committed when the FR:SR ratio is high.

fr number of pages that had to be freed to replenish the free list or to accommodate an active process

sr number of pages that had to be examined in order to Free fr number of pages

An fr:sr ratio of 1:4 means that for every one page freed, four pages had to be examined. It is difficult to determine a memory constraint based on this ratio alone and what constitutes a "high" ratio is workload/application dependent.
Note: Memory is over-committed to the point of thrashing when PO*SYS>FR

The system considers itself to be thrashing when po*SYS > fr where SYS is a system parameter; viewed with the 'schedtune' command. The default value is 6 for AIX version 3. In version 4, the default is 0 if a system has 128MB or more; otherwise, it is 6. Thrashing is the condition when the system spends more time paging rather than performing work. When this occurs, selected processes may be suspended temporarily, and the system can be noticeably slower.

- svmon

As root run the following command:

```
# svmon -Pau 10 | more
```

Sample Output:

```
Pid            Command        Inuse        Pin      Pgspace
13794             dtwm         1603          1          449

Pid:  13794
Command:  dtwm

Segid Type Description        Inuse Pin Pgspace Address Range
b23 pers /dev/hd2:24849           2   0       0 0..1
14a5 pers /dev/hd2:24842          0   0       0 0..2
6179 work lib data              131   0      98 0..891
280a work shared library text  1101   0      10 0..65535
181 work private                287   1     341 0..310:65277..65535
57d5 pers code,/dev/hd2:61722    82   0       0 0..135
```

This will list the top ten memory using processes and give a report about each one. In each process report, look where "Type" = work and "Description" = private and check how many 4K (4096 byte) pages are used under the "Pgspace" column. This is the minimum number of working pages this segment is using in all of virtual memory. A Pgspace number that grows but never decreases may indicate a memory leak. Memory leaks occur when an application fails to deallocate memory.

```
341 * 4096 =  1,396,736 or 1.4MB of virtual memory
```

- ps

Run the following command:

```
ps gv | head -n 1; ps gv | egrep -v "RSS" | sort +6b -7 -n -r
```

SIZE This is the amount of memory in KB allocated from page space for the memory segment of "Type" = work and "Description" = private for the process as would be indicated by svmon.

RSS This is the amount of memory, in KB, currently in use (in RAM) for the memory segment of "Type" = work and "Description" = private plus the memory segment(s) of "Type" = pers and "Description" = code for the process as would be indicated by svmon.

TRS This is the amount of memory, in KB, currently in use (in RAM) for the memory segment(s) of "Type" = pers and "Description" = code for the process as would be indicated by svmon.

%MEM This is the RSS value divided by the total amount of system RAM in KB multiplied by 100.

CPU Bottlecks

The following section sescribes CPU bottleneck solutions using the following commands: vmstat, tprof, ps

- vmstat

Run the following command:

```
vmstat 1
```

Note: System may slow down when processes wait on the run queue.

id - percentage of time the cpu is idle

r - number of processes (3.2) or threads (4.1) on the run queue

If the id value is consistently 0%, this means the cpu is being used 100% of the time. Look next at the "r" column to see how many processes are placed on the run queue per second. The higher the number of processes that are forced to wait on the run queue, the more system performance will suffer.

- tprof

To find out how much CPU time a process is using run the following command as root:

```
# tprof -x sleep 30
```

This will return in 30 seconds and will create a file in the current directory called __prof.all. In 30 seconds, the cpu is checked approximately 3000 times. The "Total" column is the number of times a process was found in the cpu. If one process has 1500 in the "Total" column, this process has taken 1500/3000 or half of the cpu time. The tprof output explains exactly what processes the cpu has been executing. The "wait" process executes when no other processes require the cpu and accounts for the amount of idle time on the system.

- netpmon

To find out how much CPU time a process is using, and how much of that time is spent executing network-related code, run the followin command as root:

```
# netpmon -o /tmp/netpmon.out -O cpu -v;sleep 30;trcstop
```

This will return in 30 seconds and will create a file in the /tmp directory called netpmon.out. The "CPUTime" indicates the total amount of CPU time for the process, "%CPU" is the percent CPU usage for the process, and "Network CPU%" is the percent of total time that the process spent executing network-related code.

- ps

Run the following commands:

```
ps -ef | head -n 1
ps -ef | egrep -v "UID|0:00|\ 0\ " | sort +3b -4 -n -r
```

Look at the "C" column to see a process' penalty for recent CPU usage. The maximum value for this column is 120.

```
ps -e | head -n 1
ps -e | egrep -v "TIME|0:" | sort +2b -3 -n -r
```

Look at the "TIME" column to see a process' accumulated CPU time.

```
ps gu
ps gu | egrep -v "CPU|kproc" | sort +2b -3 -n -r
```

Look at the "%CPU" column to see a process' CPU dependency. The percent CPU is the total CPU time divided by the the total elapsed time since the process was started.

I/0 Bottlenecks

This section describes bottleneck solutions using the following commands: iostat, filemon

- iostat

Note: High iowait will cause slower performance.

Run the following command:

```
iostat 5
```

%iowait percentage of time the cpu is idle while waiting on local I/O

%idle percentage of time the cpu is idle while not waiting on local I/O

The time is attributed to iowait when no processes are ready for the cpu but at least one process is waiting on I/O. A high percentage of iowait time indicates that disk I/O is a major contributor to the delay in execution of processes. In general, if system slowness occurs and %iowait is 20% to 25% or higher, investigation of a disk bottleneck is in order.

%tm_act percentage of time the disk is busy

Note: High TM_ACT percentage can indicate a disk bottleneck.

When %tm_act or time active for a disk is high, noticeable performance degradation can occur. On some systems, a %tm_act of 35% or higher for one disk can cause noticeably slow performance. Look for busy vs. idle drives. Moving data from more busy to less busy drives may help alleviate a disk bottleneck. Check for paging activity by following the instructions in MEMORY BOTTLENECKS. Paging to and from disk will contribute to the I/O load.

- filemon

to find out what files logical volumes, and disks are most active run the following command as root:

```
# filemon -u -O all -o /tmp/fmon.out; sleep 30;trcstop
```

In 30 seconds, a report will be created in /tmp/fmon.out. Look for most active segments, logical volumes, and physical volumes in this report. Look for reads and writes to paging space to determine if the disk activity is true application I/O or if it is due to paging activity. Look for files and logical volumes that are particularly active. If these are on a busy physical volume, moving some data to a less busy disk can improve performance.

The Most Active Segments report lists the most active files by file system and inode. The mount point of the file system and inode of the file can be used with the ncheck command to identify unknown files:

```
# ncheck -i <inode> <mount point>
```

This report is useful in determining if the activity is to a filesystem (segtype = persistent), the JFS log (segtype = log), or to paging space (segtype = working).

By examining the "reads" and "read sequences" counts, you can determine if the access is sequential or random. As the read sequences count approaches the reads count, file access is more random. The same applies to the "writes" and "write sequences."

SMP PERFORMANCE TUNING

Performance Tools

- SMP only

  ```
  :cpu_state -l
  ```

  Shows what state each processor is currently in (enabled, disabled, or unavailable).

- AIX tools that have been adapted in order to display more meaninful information on SMP systems

  ```
  ps -m -o THREAD
  ```

  The BND column will indicate the processor number to which a process/thread is bound to (if it is bound).

  ```
  pstat -A
  ```

  The CPUID column will indicate the processor number to which a process/thread is bound to.

  ```
  sar -P ALL
  ```

  Load on all the processors.

  ```
  vmstat
  ```

Now shows "kthr"(kernel threads) instead of "procs."

`netpmon -t`

Prints CPU reports on a per-thread basis.

- Other AIX tools that did not change

  filemon iostat svmon tprof

Tunning Methodology

- Check availablity of processors

  `cpu_state -l`

- Check balance between processors

  `sar -P ALL`

- Identify bound processes/threads

  ```
  ps -m -o THREAD
  pstat -A
  ```

- Unbind any bound processes/threads that can and should be unbound
- CONTINUE AS WITH UNIPROCESSOR SYSTEM

MISC INFO

KBUFFERS vs. VMM

The Block I/O Buffer Cache (KBUFFERS) is only used when directly accessing a block device such as /dev/hdisk0. Normal access through the Journaled File System (JFS) is managed by the Virtual Memory Manager (VMM) and therefore does not use the traditional method for caching the data blocks. Any I/O operations to either raw logical volumes or physical volumes does not use the Block I/O Buffer Cache.

I/O Pacing

Users of AIX occasionally encounter long interactive-application response times when other applications in the system are doing large writes to disk. Because most writes are asynchronous, FIFO I/O queues of several megabytes can build up, which can take several seconds to complete. The performance of an interactive process is severely impacted if every disk read spends several seconds working its way through the queue. I/O pacing limits the number of I/O requests that can be outstanding against a file. When a process tries to write to a file whose queue is at the high-water mark (which should be a multiple of 4 plus 1), it is suspended until enough I/Os have completed to bring the queue for that file to the low-water mark. The delta between the high and low water marks should be kept small.

I/O pacing can be configure on the system via SMIT. Enter the following at the command line as root:

```
# smitty chgsys
```

Async I/O

Async I/O is performed in the background and does not block the user process. This improves performance because I/O operations and application processing can run concurrently. However, applications must be specifically written to take advantage of aysnc I/O which is managed by the aio daemons running on the system.

Async I/O can be configured on the system via SMIT. Enter the following at the command line as root:

```
# smitty aio
```

FURTHER INFORMATION

Consult Line Performance Analysis - The AIX Support Family offers a system analysis with tuning recommendations. For more information (U.S.) call 1-800-CALL-AIX , other countries call your local support structure.

Performance Tuning Guide (SC23-2365) - This IBM publication covers performance monitoring and tuning of AIX systems. Order through your local IBM representative or (U.S.) by calling IBM Publications at 1-800-879-2755.

For detailed system usage on a per process basis, a free utlity called UTLD can be obtained by anonymous ftp from ftp.software.ibm.com in the /aix/tools/perftools/utld directory. For more information see the README file /usr/lpp/utld after installation of the utld.obj fileset.

08/20/96, 4FAX# 2445

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 (bottleneck.lxp, 4FAX# 2445)