09/24/96,

Network Tuning

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

(for complete updated document, order publication SC23-2365)

AIX 3.2 Network Tuning Guide
Revision: 1.0
April 10, 1992

IBM AIX System Performance
11400 Burnet Road
Austin, TX 78758
Tom Kochie

Revision History
Revision 1.0

Tuning the memory buffer (mbuf) pools

1.1 Why tune the mbuf pools

The network subsystem uses a memory management facility that revolves around a data structure called an mbuf. An mbuf is simply a structure that is used to store data, for the most part used to store data for incoming and outbound network traffic. Having the mbuf free pools configured properly to the right size can have a very positive effect on network performance. If configured improperly network performance can suffer as well as overall system performance. AIX offers the convenience of run time mbuf pool configuration. With this convenience comes the complexity of knowing when the pools need adjusting and how much they should be adjusted.

1.2 Overview of the mbuf management facility

To understand how the various mbuf tuning parameters effect the mbuf pools and how they are organized, a brief description of the mbuf management facility and the underlying mbuf kernel data structures is needed.

The mbuf management facility is comprised of many services to allow for the manipulation, allocation, and deallocation of mbufs. The following is a high level discussion intended to give the reader enough information to allow them to understand how tuning the mbuf pools effects system and network performance.

Within the mbuf management facility there exists 2 free pools of buffers, 1 pool of small buffers (256 bytes each) and 1 pool of large buffers (4096 bytes each). The pools are created from system memory by making an allocation request to the Virtual Memory Manager (VMM). The pools consist of pinned pieces of virtual memory; this means that they must always reside in physical memory and are never paged out. The underlying result of this is that the VMM's resources have been decreased by the amount that the mbuf pools have been increased. The VMM is responsible for managing the memory for the whole system; if it's resources are improperly balanced or abused performance for the whole system can be effected.

The initial size of the mbuf free pools is system-dependant. There is a minimum number of small and large free buffers allocated for each system, but then as system configurations vary the free pools will be increased by some amount. One factor affecting how much they are increased is the number of communications adapters that exist in the system. The default pool sizes are initially configured to handle small to medium size network loads (network traffic 100-500 packets/second). The pool sizes dynamically increase and decrease as network loads increase and decrease. To optimize network performance, the administrator should balance mbuf free pool sizes with network loads (packets/second). If the network load is particularly oriented towards UDP traffic (e.g. NFS server) the small mbuf pool should be 2 times the packet/second rate. This is due to UDP traffic consuming an extra small mbuf.

To provide an efficient mbuf allocation service, an attempt is made to maintain a minimum number of buffers in the pools at all times. The following network options (which can be manipulated using the no command) are used to define these minimum values:

- lowmbuf
- lowclust

The lowmbuf option controls the minimum number of buffers for the small pool. The lowclust option controls the minimum number of buffers for the large pool. (A large mbuf is more commonly called a cluster.) When the number of buffers in the free pools drop below the lowmbuf or lowclust thresholds the pools are expanded by some amount. The expansion of the mbuf free pools is not done immediately but is scheduled to be done by a kernel process with the process name of "netm". When netm is dispatched the pools will be expanded to meet

the minimum requirements of lowclust and lowmbuf. Having a kernel process do this work may seem inefficient but is required due to design features within the VMM.

An additional function that netm provides is to not allow the pool of large mbufs to grow beyond a maximum limit. The following network option is used to define this maximum value:

- mb_cl_hiwat

The mb_cl_hiwat option controls the maximum number of buffers the large mbuf free pool will be allowed to expand to. When the number of large mbufs on the free pool exceeds mb_cl_hiwat, netm will be scheduled to release some of the large mbufs back to the VMM.

The last network option that is used by the mbuf management facility is:

- thewall

The thewall option controls the maximum number of bytes (in K bytes) that the mbuf management facility can allocate from the VMM. This option is used to prevent unbalanced VMM resources which result in poor system performance.

1.3 When to tune the mbuf pools

When and how much to tune the mbuf pools is directly related to the network load a given machine is being subjected to. A server machine that is supporting many clients is a good candidate for having the mbuf pools tuned to optimize network performance. It is important for the system administrator to understand the networking load for a given system. By using the netstat -I <interface name> <interval> you can get a rough idea of the network load in packets/second. In addition to understanding the network load there are some mbuf statistics available to help determine a system's mbuf needs and whether the mbuf pools should be reconfigured.

There are 2 mbuf statistics that suggest the mbuf pools may need additional tuning. These may be viewed using the netstat command with the -m option:

- netstat -m shows requests for memory denied > 0
- netstat -m shows Kbytes allocated to network approaching "thewall" network options value

The netstat -m "requests for memory denied" counter is maintained by the mbuf management facility and is incremented each time a request for an mbuf allocation is not able to be satisfied. Normally the "requests for memory denied" value will be zero. If a system experiences a high burst of network traffic the default configured mbuf pools will not be sufficient to meet the demand of the incoming burst, causing the error counter to be incremented once for each mbuf allocation request that fails. Usually this is in the thousands due to the large number of packets arriving all at once. The request for memory denied statistic

will correspond with dropped packets on the network. Dropped network packets means re-transmissions, resulting in degraded network performance. If the "requests for memory denied" value is greater than zero it may be appropriate to increase some of the mbuf management default configuration values.

The netstat -m "Kbytes allocated to the network" statistic is maintained by the mbuf management facility and represents the current amount of system memory that has been allocated to the mbuf pools. A network options parameter (thewall) is used to prevent the mbuf management facility from consuming too much of a system's physical memory. The default value for thewall parameter limits the mbuf management facility to 2048K bytes (2 megabytes) of system memory. If the Kbytes currently allocated to the network approaches "thewall" value it may be appropriate to increase some of the mbuf management default configuration values.

The netm kernel process runs at a very favored priority (fixed 37). Because of this, excessive netm dispatching can cause not only poor network performance but poor system performance as well due to competition with other system and user processes. Minimizing netm dispatching can be done by properly configuring the mbuf free pools to match system and networking needs. Improperly configured free pools can result in netm "thrashing" due to conflicting network traffic needs and improperly tuned thresholds.

There are cases where the above indicators reflect additional tuning may be necessary but may be a result of some other system problem that should be corrected first. The following are conditions where the mbuf pools should NOT be tuned:

- mbuf memory leak
- queued data not being read from socket or other internal queueing structure

An mbuf memory leak is a condition where some kernel or kernel extension code path forgot to release the mbuf resource and destroyed the pointer to this memory location thereby losing the address to the mbuf forever. If this occurs frequently enough eventually all the mbuf resources will be used up. This condition is caused by a software defect within the kernel or kernel extension. An administrator can monitor the netstat -m mbuf statistics and look for a gradual increase in usage that never decreases. Alternately the administrator can look for high mbuf usage on a relatively idle system. If either of these conditions exist the mbuf memory leak should be isolated and fixed.

It is also possible due to an application defect to have excessive amounts of mbufs queued at the socket layer. Normally an application program would read data from the socket thereby causing the mbufs to be released back to the mbuf management facility. Again an administrator can monitor the netstat -m mbuf statistics and look for high mbuf usage while there is no expected network traffic. The administrator can also view the current list of running processes (ps -ef) and scan for those that use the network subsystem with large

amounts of cpu time being used. If this behavior is observed the suspected application defect should be isolated and fixed.

1.4 How to tune the mbuf pools

With the understanding of how the mbuf pools are organized and managed, tuning the mbuf pools is simple for AIX and may be done at run-time (unlike other unix systems where the kernel must be recompiled and the system rebooted).

The network options command may be used by root users to modify these mbuf pool parameters that effect the size of the mbuf pools and how they are managed:

lowmbuf - Specifies the minimum number of small mbufs to maintain in the small mbuf free pool. If the number of free buffers in the pool drops below this value, the small mbuf free pool is expanded so that at least lowmbuf free buffers are available.

lowclust - Specifies the minimum number of large mbufs to maintain in the large mbuf free pool. If the number of free buffers in the pool drops below this value, the large mbuf free pool is expanded so that at least lowclust free buffers are available.

mb_cl_hiwat - Specifies the maximum number of large mbufs to maintain in the large mbuf free pool. If the number of free buffers in the pool increases above this value, the pool is decreased so that at most mb_cl_hiwat free buffers are available. The number of free buffers beyond mb_cl_hiwat are released back to the system memory allocator (VMM).

thewall - Specifies the maximum amount of memory in K bytes that can be allocated to the small and large mbuf free pools. The default value of 2048 will allow up to 2 megabytes of memory to be allocated.

The following items should be considered when tuning the mbuf pools:

- After expanding the pools use the vmstat command to insure paging rates are not excessive. Consider checking the paging rates prior to expanding the pools and comparing after the expansion. If you are not able to expand the pools to the necessary levels without adversly effecting the paging rates additional memory may be required.
- When adjusting lowclust, lowmbuf should be adjusted by at least the amount that lowclust is. For every large mbuf there will exist a small mbuf that points to it.
- mb_cl_hiwat should remain at least 2 times greater than lowclust at all times. This will prevent the netm thrashing discussed earlier.

- When adjusting lowclust and lowmbuf, "thewall" may need to be increased to prevent pool expansions from exceeding thewall limit.

tcp_keepidle and tcp_keepintvl definitions:

In 3.2 and 3.1.x (with apar fix IX21955 applied), the new kernel has two additional tuning parameters regarding socket connection maintenance. With these two parameters set, if a socket connection is inactive for the number of 1/2 seconds in tcp_keepidle, the host will start sending query packets (probes) repeating them every tcp_keepintvl * 1/2 sec. until either the remote socket responds or eight probes are sent. If a response is not received after 8 queries, then the socket is closed. Previously these parameters were not adjustable and defaulted to defaults of 2 hours for keepidle and 75 seconds for keepintvl. They are now adjustable through the no command but are initially set to the above defaults.

You can use netstat -m following any changes to verify the size of the free pool of large mbufs (also known as mapped pages). To verify the size of the free pool of small mbufs you will have to examine a kernel data structure mbstat (see /usr/include/sys/mbuf.h) using the crash command. The kernel address of mbstat can be displayed while in crash using od mbstat. You will then need to od <kernel address> 30, to dump the mbstat structure. The first word in the structure is the size of the small mbuf free pool.

Note: For additional network tuning and performance information, see the IBM AIX Version 3.2 for the RISC System/6000 Performance Monitoring and Tuning Guide, SC23-2365.

09/24/96,

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:

09/24/96,

END OF DOCUMENT (network.tuning.tcp)