Optimizing AIX 6.1 performance tuning

Tuning methodology, general enhancements, monitoring, and tuning

Skill Level: Intermediate

Ken Milberg (ken@unix-linux.org)
President and Managing Consultant, Technology Writer, and Site Expert
Future Tech

05 Feb 2008

Just when you thought you understood performance tuning on AIX® Version 5.3, here comes AIX 6.1 on its heels to throw you a curveball. In this article, get up-to-date information on the recent changes to performance monitoring and tuning in AIX 6.1, including CPU, virtual memory, and I/O (disk and network). Many of the changes are really less about kernel innovations and more about ancillary changes, such as improving default parameters to more accurately reflect real-world data processing. Other enhancements include unique tunable documentation, restricted tunables, and various other improvements to certain subsystems. You'll learn more about a performance tuning methodology, which needs to be a part of any tuning strategy.

Introduction

AIX® 6.1 has many important innovations and improvements, including:

- Virtualization—Features such as workload partitioning and Live Application Mobility
- Security—Features such as encrypted file systems and trusted AIX and role-based access control (RBAC)
- Availability—Features such as AIX concurrent updates and dynamic
tracing

- Manageability—Features such as the new Systems Director Console for AIX and Workload Partition Manager

It also has support for POWER6 performance innovations, such as advanced simultaneous multithreading (SMT), shared dedicated processors, and variable page size. It's important to fully understand which innovations and enhancements are more the reflection of the POWER6, AIX 6.1, or a combination of both. For example, purely from an operating system perspective, AIX improves on the older defaults tunables for:

- vmo
- ioo
- aio
- no
- nfso
- schedo

While there are some real performance enhancements in AIX 6.1, such as improvements in I/O pacing and AIX's implementation of AIO servers, there is nothing breathtakingly different. In fact, there were more performance changes made from AIX 5.1 to AIX 5.2 and 5.2 to 5.3 (including new monitoring tools, tuning tools, new tunables that changed how you set Virtual Memory Manager (VMM) settings, and concurrent I/O improvements) than you will see moving from AIX 5.3 to AIX 6.1. In AIX 6.1, all the tuning commands remain the same, except for the ones that have been taken away (I.E. and aioo), and there are no new monitoring tools either. Other changes reflect updates that were made to the utilities that reflect support for other workload partitioning innovations, including:

- curt
- filemon
- iostat
- netpmon
- pprof
- procmon
- proctree
- svmon
- topas
• tprof
• vmstat

Workload partitions (WPARs) allow for the usage of separate virtual partitions within one AIX image. It is more of a compliment to logical partitions (LPARs) rather than a replacement to them. WPARs actually run inside of LPARs and are very similar in concept to Solaris containers. I've built WPARs in less than 15 minutes; in fact, you'll do some of your analysis inside WPARs so that you can actually view some of the updated tools that now support WPARs. Note that WPARs are only required in AIX 6.1 and a POWER6 is not necessary. Some commands also run differently or not run at all within WPARs—a few of them will be discussed where applicable. Other AIX 6.1 performance improvements include unique tunable documentation and restricted tunables.

Tuning methodology

This section summarizes a tuning methodology that you should use to tune your systems. You don't necessarily have to use this one; there are many others, but you should be working with some kind of tuning methodology or structured program.

The five-step methodology consists of:

1. Establishing a baseline
2. Stress testing and monitoring
3. Identification of bottleneck
4. Tuning bottleneck
5. Repeat (starting with the second step)

Let's look at the steps in more depth:

1. Establishing a baseline—Before you tune or even start monitoring, you must establish a baseline. The baseline is a snapshot of what the system looks when it is performing well. This baseline should not only capture performance-type statistics, but it should also document the actual configuration of your system (amount of memory, CPU, and disk). This is important, because you need to know what a system that performs well looks like prior to troubleshooting performance that users might have complained about. Doing so helps you translate the data into a service level agreement (SLA) in which the customer signs off on.
2. Stress testing and monitoring—Here is where you monitor and stress your systems at peak workloads. You should use several monitoring tools here to help validate your findings. The monitoring section is critical, as you cannot effectively tune anything without having an accurate historical record of what has been going on in your system, particularly during periods of stress.

3. Identification of bottleneck—The objective of stressing and monitoring your system is to determine the bottleneck. You cannot provide the correct medicine without the proper diagnosis. If the system is in fact CPU bound, then you can run additional tools, such as trace, curt, splat, tprof, and ps to further identify the actual processes that are causing the bottleneck.

4. Tuning bottleneck—After you’ve finally identified the bottleneck, it is time to tune it. The tuning that you do is dependent on the bottleneck, for example, CPU, virtual memory, or I/O. Some subsystems lend themselves more to active tuning, such as virtual memory and using vmo, while the "cure" for CPU-type bottlenecks is usually managing your workload more efficiently or assigning (using dynamic logical partitioning (DLPAR), uncapped partitions, or Partition Load Manager) more resources to your system.

5. Repeat—Time to go through this process again, starting from the second step of the process. Only by repeating your tests and consistently monitoring your systems can you determine if your tuning has really made an impact.

AIX 6.1—memory

This section discusses the changes made in the virtual memory subsystem.

Through the years, many have complained about the default parameters of AIX. The complaints have not just reflected the reality of what most users run on top of AIX, for example, mission-critical database applications like Oracle. Because of this, systems administrators have had to change settings on many subsystems—most notably with virtual memory (in other words, minperm and maxperm). IBM® has listened and changed these parameters to reflect that reality. It should also be noted that you should not rely on these settings exclusively, and you should always check with your independent software vendor (ISV) to validate what their recommended settings are for AIX 6.1 and make the changes, accordingly.

Let’s look at a side-by-side view of vmo parameters for AIX 5.3 and AIX 6.1 in Table 1. You will see that some parameters are now in restricted mode. The most
important changes were made to address paging issues, where database servers frequently page out computational pages, even though the system has enough free memory. In prior articles on tuning (see Resources), it was recommended to change those parameters to defaults fairly close to what was indicated on the table. The changes are indicated in the AIX 5.3 tuning recommendation column. Many tunables are now also classified differently as restricted in an attempt to discourage changing certain parameters. The net is that only 29 vmo tunables can now be changed without a firm warning; 30 are now deemed restricted tunables, which IBM officially states should not be modified unless instructed by "IBM support professionals." IBM has made this change in an attempt to discourage junior administrators from changing certain parameters that were deemed critical enough to be classified as restricted.

Table 1. List of vmo parameters and tuning recommendations

<table>
<thead>
<tr>
<th>Type</th>
<th>AIX 5.3</th>
<th>AIX 5.3 tuning change recommendations</th>
<th>AIX 6.1</th>
<th>Restricted</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu_scale_memp = 8</td>
<td>8</td>
<td>8</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>data_stagger_interval = 161</td>
<td>161</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>defps = 1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>force_realias_lite = 0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>framesets = 2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>htabscale = n/a</td>
<td>n/a</td>
<td>n/a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kernel_heap_psize = 4096</td>
<td>4096</td>
<td>4096</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kernel_psize = 4096</td>
<td>4096</td>
<td>65536</td>
<td></td>
<td></td>
</tr>
<tr>
<td>large_page_heap_size = 0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lgpg_regions = 0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lgpg_size = 0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>low_ps_handling = 1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lru_file_repage = 1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lru_poll_interval = 10</td>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lrubucket = 131072</td>
<td>131072</td>
<td>131072</td>
<td></td>
<td></td>
</tr>
<tr>
<td>maxclient% = 80</td>
<td>80</td>
<td>90</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Optimizing AIX 6.1 performance tuning
© Copyright IBM Corporation 1994, 2007. All rights reserved.
maxfree = 1088 | 1088 | 1088  
maxperm = 598781 | 442683  
maxperm% = 80 | 90 | 90 | Yes  
maxpin = 635249 | 423471  
maxpin% = 80 | 80  
mbuf_heap_psize = 65536 | 64436 | Yes  
memory_affinity = 1 | 1 | Yes  
memory_frames = 786432 | 524288  
memplace_data = 2 | 2  
memplace_mapped_file = 2 | 2  
memplace_shm_anonymous = 2 | 2  
memplace_shm_named = 2 | 2  
memplace_stack = 2 | 2 | 2  
memplace_text = 2 | 2  
memplace_unmapped_file = 2 | 2  
mempools = 5 | 5  
minfree = 960 | 960 | 960  
minperm = 149694 | 149694 | 14756  
minperm% = 20 | 5 | 3  
nokilluid = 0 | 0  
npskill = 3072 | 3072 | 3072  
npsrpgmax = 24576 | 8192 | Yes  
npsrpgmin = 18432 | 6144 | Yes  
npsscrubmax = 24576 | 8192 | Yes  
npsscrubmin = 18432 | 6144 | Yes
A new `vmo` flag, `-F`, provides all parameters, including restricted ones. Most of the content here is stripped out—only a small sampling is provided that includes the restricted section (see Listing 1).

**Listing 1. Using the -F flag with vmo to view all the tunable values**
Even restricted tunables can be changed. When you do make a change, you just receive a stern warning (see Listing 2).

Listing 2. Adjusting tunables warning

```
lpardml162f_pub[/] > vmo -F -a
    force_realias_lite = 0
    vmm_default_pspa = -1
##Restricted tunables
    maxperm% = 90
```

```
lpar9ml162f_pub[/] > vmo -o maxperm%=99
    Setting maxperm% to 99
    Warning: a restricted tunable has been modified

When changing the parameter after a reboot, you'll get a further rebuke and be asked to actually confirm whether or not you want do to this. You'll have to physically type in "yes" (see Listing 3).

Listing 3. Restricted tunables—answering yes to change

```
lpar9ml162f_pub[/] > vmo -o maxperm%=99
    Setting maxperm% to 99
    Warning: a restricted tunable has been modified
lpardml162f_pub[/] > vmo -p -o maxperm%=99
    Modification to restricted tunable maxperm%, confirmation required yes/no yes
    Setting maxperm% to 99 in nextboot file
    Setting maxperm% to 99
    Warning: a restricted tunable has been modified
```

The most important out-of-the-box performance values include new values for:

- minperm
- maxperm
- maxclient
- strict_maxclient

This is based on changes that first appeared during AIX 5.3, when you no longer had to turn off strict_maxclient, increase minfrree and maxfree, or reduce minperm, maxperm, and maxclient. The new recommendation (now incorporated as the default value in AIX 6.1) is to turn off the repage ratio check, lru_file_repage, which ensures that working storage is not paged and to consider only file paging. In AIX 6.1, the VMM replacement default is changed to utilize up to 90 percent of its real memory for file caching, favoring computational pages over file pages. Unless the amount of active virtual memory exceeds 97 percent of the size of real memory, minperm is reduced to three percent (see Listing 4) to ensure that computational pages will not be stolen.
Listing 4. minperm reduced to three percent

```bash
# vmo -o minperm%=97
Value of the tunable minperm% cannot be changed in a WPAR
```

Some changes will not work in WPARs. WPARs are a subset of an LPAR, but they are still part of the single operating system image.

Another important change includes VMM dynamic variable page size support (VPSS). Pages are defined as fixed length data blocks and held in virtual memory. In AIX 6.1 (on POWER6 processors only), VMM can now dynamically use the larger page size based on application memory usage, which should substantially improve performance. This feature is completely transparent to applications. AIX only uses the larger pages sized if it doesn't result in increasing process memory usage. The use of larger pages increases performance because fewer hardware address translations would need to be made. It only supports doing this on working storage memory, not persistent storage. The new parameters are `vmm_default_pspa` (it also works in conjunction with the `vmm_mpsize_support` tunable).

Listing 5. lsattr output illustrating processor architecture

```bash
lpar9ml162f_pub[/usr/samples/tcpip] > lsattr -E1 proc0
Frequency 1654344000 Processor Speed False
smt_enabled true Processor SMT enabled False
smt_threads 2 Processor SMT threads False
state enable Processor state False
type PowerPC POWER5 Processor type False
```

Let's view the setting for tunable settings for VPSS.

Listing 6. VPSS automatically disabled on a POWER5 system

```bash
lpar9ml162f_pub[/usr/samples/tcpip] > vmo -a | grep pspa
  vmm_default_pspa = -1
```

AIX 6.1—CPU

This section highlights changes made in the CPU subsystem.

While `schedo` can also be used to tune certain memory facilities, such as paging load control, it is mostly there for CPU scheduler tuning. In AIX 6.1, only 27 of the 42 tunables are restricted, leaving 15 tunables that can be modified without explicit warnings. Though some defaults were changed, there were no substantial changes that were made with respect to CPU monitoring and tuning on AIX 6.1 (see Listing 7).
AIX 6.1—Disk I/O

This section gives an overview of the changes that were made in the disk I/O subsystem.

In AIX 6.1, of the 48 tunables you can control with `ioo`, 27 of them are now restricted, leaving 21 tunables that can be modified without explicit warnings. The most important changes impact I/O pacing and AIO dynamic tunables (see Listing 8).

Listing 8. Restricted ioo tunables

```bash
# ioo -F -a
##Restricted tunables
  aio_fastpath = 1
  aio_fsfastpath = 1
  aio_kprocprio = 39
  aio_multitidsusp = 1
  aio_sample_rate = 5
  aio_samples_per_cycle = 6
  j2_maxUsableMaxTransfer = 512
  j2_nBufferPerPagerDevice = 512
  j2_nonFatalCrashesSystem= 0
  j2_syncModifiedMapped = 1
```
JFS2

Changes were made to the Enhanced Journaling File System (JFS2) that now allow you to mount a JFS2 f/s without logging. While this can substantially improve performance, it is not recommended you do this. If you need to recover your data, you have to use the dreaded fsck, which has been pretty much banished from memory since journaling file systems. Some useful circumstances can include restoring data from backups and saving time during an activity where you might have a very small window and availability is not a concern.

iSCSI

The target software driver can now be used over a gigabyte Ethernet adapter, which should clearly improve performance in this type of environment. The target driver exports local disks or logical volumes to iSCSI initiators that connect to AIX using the iSCSI protocol. The proliferation of iSCSI is a viable alternative over Fiber-based storage, making this an important enhancement.

I/O pacing

Disk I/O pacing is a mechanism that allows you to limit the number of pending I/O requests to a file. What this does is prevents disk I/O-intensive processes (usually in the form of large sequential writes) from exhausting the CPU. AIX 6.1 enables I/O pacing by default. In AIX 5.3, you needed to explicitly enable this feature. The way it does this is by setting the sys0 settings of minpout and maxput parameters to 4096 and 8193, respectively:

```
lsattr -El sys0
```

Figure 1. lsattr output illustrating new I/O pacing settings
Asynchronous I/O (AIO)

AIO is an AIX software subsystem that allows processes to issue I/O operations without waiting for I/O to finish. Because both I/O operations and applications processing run concurrently, they essentially run in the background and improve performance. This is particularly important in a database environment. There are two types of AIX subsystems: Legacy AIO and POSIX AIO. The differences between them involve different parameter passing at the application layer. In other words, the developers pick the implementation that the application uses. Regardless of which subsystem is chosen, both run concurrently on AIX. In AIX 5L™, if applications used AIO, the subsystem would need to be explicitly activated in the \textit{autoconfig} parameter. The system also requires a reboot because the kernel extensions need to be loaded. In fact, any release prior to AIX 5.3 TL5 required reboots if there were any changes to the either of the following tunables:

- \textit{maxreqs}
- \textit{maxservers}
- \textit{minservers}

In AIX 5.3, you also had the \texttt{aioo} command, which allowed you to make these changes dynamically without a reboot (decreasing required reboots). It should be noted that the command did not change the Object Data Manager (ODM) attributes, meaning they would not be persistent across a reboot.

In AIX 6.1, the tunables \texttt{fastpath} and \texttt{fsfastpath} are now restricted tunables and set to 1 by default. It has the following effect on the tunables (see Listing 9):

- \texttt{fastpath}: AIO requests that raw logical volumes get passed directly to the disk layer.
- \texttt{fsfastpath}: AIO requests that files opened with concurrent I/O on JFS2 get passed directly to LVM or disk.

\textbf{Listing 9. Restricted tunables}

\begin{verbatim}
log_p_g_dealloc true Log predictive memory page deallocation events True
max_capacity 0.25 Maximum potential processor capacity False
max_logname 9 Maximum log name length at boot time True
maxbuf 20 Maximum number of pages in block I/O BUFFER CACHE True
maxmbuf 0 Maximum Kbytes of real memory allowed for MBUFS True
maxnout 8193 HIGH water mark for pending write I/Os per file True
maxproc 128 Maximum number of PROCESSES allowed per user True
min_capacity 0.25 Minimum potential processor capacity False
minnout 4096 LOW water mark for pending write I/Os per file True
node_name IBM9117-570 Machine name False
xargs 256 ARG/ENV list size in 4K byte blocks True
xfs acl compat secure NFS4 ACL Compatibility Mode True
\end{verbatim}
Further, AIO subsystems are now loaded by default and not activated. They are automatically started at the time when the application initiates the AIO I/O requests. There is no more aioo command (what a short life span) and these tunables are now used only with ioo (see Listing 10).

Listing 10. AIX old method (AIX 5.3)

```bash
lpar9ml24f_pub[] > # aioo -a
    minservers = 1
    maxservers = 1
    maxreqs= 4096
    fsfastpath = 0
```

Listing 11 provides the new method for AIX 6.1.

Listing 11. AIO new method (AIX 6.1)

```bash
lpar9ml162f_pub[] > ioo -a | grep active
    aio_active = 0
    posix_aio_active =
```

It's also worth noting that there are no more AIO devices in the ODM.

As you can see, two new parameters have also been added to ioo: aio_active and posix_aix_active. These can only be changed by AIX, and they are set to 1 only when AIO kernel extensors are used and pinned. If you like grep, you won't find any more AIO servers. You'll now see aioLpools and aioPpools; these are the kernel processes that manage the AIO subsystems for Legacy and POSIX. As a result of this change, there is less pinned memory and fewer processes that are running on the system; both have positive affects on overall systems performance (see Listing 12).

Listing 12. New AIO kernel processes

```bash
lpar9ml162f_pub[] > pstat -a | grep aio
  39 a  2704e  1 2704e  0 0  1 aioLpool
  40 a  28050  1 28050  0 0  1 aioPpool
lpar9ml162f_pub[] >
```

The minserver and maxserver parameters, as they relate to AIO servers, are now tuned per each CPU tunable. Changing these values will not result in changes to the number of available servers on the system; the amount is dependent on the
number of concurrent I/O requests (see Listing 13).

### Listing 13. Changes in default parameters for minservers and maxservers

```shell
lpar9ml162f_pub[/] > ioo -a | grep minservers
  aio_minservers = 3
  posix_aio_minservers = 3
lpar9ml162f_pub[/] > ioo -a | grep maxservers
  aio_maxservers = 30
  posix_aio_maxservers = 30
```

It's also worth noting that there are no more AIO devices in the ODM.

### AIX 6.1—Network and Network File System (NFS) I/O

This section outlines changes that were made in the Network I/O subsystem, including NFS.

Of the 133 not tunables, only five have been classified as restricted (see Listing 14).

### Listing 14. Restricted tunables

```shell
#no -F -a
##Restricted tunables
  extendednetstats = 0
  inet_stack_size = 16
  net_malloc_police = 16384
  pseintrstack = 24576
  use_isno = 1
```

A new network caching daemon has also been introduced to improve performance when resolving using DNS.

This daemon can be started from the System Resource Controller (SRC).

It's main configuration file is in `/etc/netcd.conf`, and you can copy the one in `/usr/samples/tcpip` to `/etc` and use that as a template. The command used to manage the daemon is `netcdctrl`. With this command, you can dump the cache contents to a file, display statistics of cache usage, flush the cache table, and change the logging level of the daemon. Listing 15 displays the stats.

### Listing 15. Displaying caching stats with netcdctrl

```shell
lpar9ml162f_pub[/usr/samples/tcpip] > netcdctrl -t dns -e hosts -s /tmp/netcd.stats
lpar9ml162f_pub[/usr/samples/tcpip] > more /tmp/netcd.stats
CACHE dns, hosts, name
  Hash index : 0, Max number of entries : 0, Current number of entries : 0
```
Regarding the /etc/netsvc.conf file, nothing has changed; this file is still necessary in determining the order of resolving.

NFS

Of the 24 NFS tunables, 21 have been classified as restricted (see Listing 16).

Listing 16. Restricted tunables

```bash
#nfso -a
##Restricted tunables
  lockd_debug_level = 0
  nfs_allow_all_signals = 0
  nfs_auto_rbr_trigger = 0
  nfs_dynamic_retrans = 1
  nfs_gather_threshold = 4096
  nfs_iopace_pages = 0
  nfs_max_threads = 3891
  nfs_repeat_messages = 0
  nfs_socketsize = 600000
  nfs_tcp_duplicate_cache_size = 5000
  nfs_tcp_socketsize = 600000
  nfs_udp_duplicate_cache_size = 5000
  nfs_v2_pdts = 1
  nfs_v3_pdts = 1
  nfs_v4_pdts = 1
  nfs_v2_vm_bufs = 10000
  nfs_v3_vm_bufs = 10000
  nfs_v4_vm_bufs = 10000
  statd_debug_level = 0
  statd_max_threads = 50
  udpchecksum = 1

1par9ml162f_pub[/]
```

The only noteworthy change here is that RFC1323 (on the TC/IP stack) is now enabled by default, which allows TCP connections to use the TCP scaling window for any NFS connections. Further, the default number of biod daemons has now increased to 32 for each NFS V3 mount point.

Summary

AIX 6.1 provides many innovations, such as the use of WPARs and live application availability. It also provides for many changes with respect to performance tuning. Some of the more important changes include improving default parameters to more accurately reflect real-world data processing, unique tunable documentation, the incorporation of restricted tunables, and enhancements to I/O pacing and AIO servers. AIX 6.1 is simply wonderful, but also remember that in the world of IBM, AIX
6.1 is the new 6.0. If you recall, AIX 5.2 was a lot better and more stable than AIX 5.1. Consider everything before doing an operating system upgrade, but certainly don't be afraid of moving over if your application fully supports it and you feel the innovations are important enough in your environment to make the move.
Resources

Learn

- Use RSS feed to request notification for the upcoming articles in the following series:
  - Optimizing AIX 5L™ performance: Tuning disk performance (Ken Milberg, developerWorks, November 2007)
  - Optimizing AIX 5L™ performance: Tuning disk performance (Ken Milberg, developerWorks, July 2007)
  - Optimizing AIX 5L performance: Tuning your memory settings (Ken Milberg, developerWorks, June 2007)
  - Optimizing AIX 5L performance: Monitoring your CPU (Ken Milberg, developerWorks, April 2007)

(Find out more about RSS feeds of developerWorks content.)

- "Use Direct I/O to improve performance of your AIX applications" (Shiv Dutta, developerWorks, November 2002): This article discusses the benefits of direct I/O and tells how to implement it.

- Improving database performance with AIX concurrent I/O: Read this white paper for more information on how to improve database performance.

- AIX memory affinity support: Learn more about affinity support from the IBM System p™ and AIX InfoCenter.

- IBM Redbooks: Database Performance Tuning on AIX is designed to help system designers, system administrators, and database administrators design, size, implement, maintain, monitor, and tune a Relational Database Management System (RDMBS) for optimal performance on AIX.

- Test 234: AIX 5L Performance and Systems Tuning: This exam is part of the IBM certification program.

- "Processor Affinity on AIX" (developerWorks, November 2006): Using process affinity settings to bind or unbind threads can help you find the root cause of troublesome hang or deadlock problems. Read this article to learn how to use processor affinity to restrict a process and run it only on a specified central processing unit (CPU).

- "CPU Monitoring and Tuning" (March, 2002): Learn how standard AIX tools can help you determine CPU bottlenecks.

- IBM Redbooks: For a comprehensive guide about the performance monitoring and tuning tools that are provided with AIX 5L Version 5.3, read AIX 5L Practical Performance Tools and Tuning Guide.
• "AIX 5L Version 5.3: What's in it for you?" (Shiv Dutta, developerWorks, June 2005): Learn what features you can benefit from in AIX 5L Version 5.3.

• IBM Redbooks: *AIX 5.3 Performance Management Guide* provides application programmers, customer engineers, system engineers, system administrators, experienced end users, and system programmers with complete information about how to perform such tasks as assessing and tuning the performance of processors, file systems, memory disk I/O, NFS, Java™, and communications.

• Operating System and Device Management from IBM provides users and system administrators with complete information that can affect your selection of options when performing such tasks as backing up and restoring the system, managing physical and logical storage, and sizing appropriate paging space.

• IBM Redbooks: The AIX 5L Differences Guide Version 5.3 Edition focuses on the differences introduced in AIX 5L Version 5.3 when compared to AIX 5L Version 5.2.

• Check out other articles and tutorials written by Ken Milberg:
  - Across developerWorks and IBM

• AIX and UNIX: The AIX and UNIX developerWorks zone provides a wealth of information relating to all aspects of AIX systems administration and expanding your UNIX skills.

• New to AIX and UNIX?: Visit the New to AIX and UNIX page to learn more about AIX and UNIX.

• AIX Wiki: Discover a collaborative environment for technical information related to AIX.

• Search the AIX and UNIX library by topic:
  - System administration
  - Application development
  - Performance
  - Porting
  - Security
  - Tips
  - Tools and utilities
  - Java technology
  - Linux®
  - Open source
• **Safari bookstore**: Visit this e-reference library to find specific technical resources.

• **developerWorks technical events and webcasts**: Stay current with developerWorks technical events and webcasts.

• **Podcasts**: Tune in and catch up with IBM technical experts.

**Get products and technologies**

• **IBM trial software**: Build your next development project with software for download directly from developerWorks.

**Discuss**

• Participate in the **developerWorks blogs** and get involved in the developerWorks community.

• Participate in the AIX and UNIX forums:
  - **AIX**—technical forum
  - **AIX for Developers Forum**
  - **Cluster Systems Management**
  - **IBM Support Assistant**
  - **Performance Tools**—technical
  - **Virtualization**—technical
  - **More AIX and UNIX forums**

**About the author**

Ken Milberg
Ken Milberg, Project Management Professional and IBM Certified Advanced Technical Expert, is a technical editor and writer for *IBM Systems Magazine, Power Systems* edition. He's the president and managing consultant of Unix-Linux Solutions. Ken holds a bachelor's degree in computer and information science and a master's degree in technology management from the University of Maryland. He is the founder and group leader of the NY Metro POWER-AIX/Linux Users Group. Through the years, he has worked for both large and small organizations and has held diverse positions from CIO to senior AIX engineer. He is a PMI certified Project Management Professional, an IBM Certified Advanced Technical Expert (IBM System p5 2006), and is also IBM SCon certified. You can contact him at ken@unix-linux.org.
Trademarks

IBM, AIX, AIX 5L, and System p are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.