



SAS on IBM AIX 5L: Tuning guidelines

Last updated: June 23, 2009

This paper provides a starting point for performance optimization from a system-wide perspective to create an enhanced environment for SAS® 9 on IBM® POWER™ processor-based servers that run IBM AIX® 5.

- **Technology levels:** Use the latest version of IBM AIX 5L™ Version 5.3 with the latest maintenance level and e-fix. See the following Web site:
ibm.com/support/fixcentral
 - At publication of this paper, TL10 with e-fix is considered to be the latest technology level.
- **IBM Hardware Management Console (HMC):** Keep the HMC and microcode up-to-date.
 - Sign up for daily notification of updates for AIX 5L, HMC and microcode:
www14.software.ibm.com/webapp/set2/subscriptions/pqvcmj
- **VMM:** Tune VMM by using the AIX **vmo** command to enable Enhanced Java File system (JFS2) to perform efficient caching. It is suggested that you start with the following parameter values:
maxclient%=80
maxperm%=80
minperm%=5
lru_file_repage=0
strict_maxperm=0
minfree=128 * #logical CPUs
maxfree=minfree +
j2_maxPageReadAhead * #logical CPUs
nokilluid=10
- **I/O:** Tune io at the filesystem layer by using the AIX **ioo** command to enable JFS2 to perform efficient caching. It is suggested that you start with the following parameter values:
j2_dynamicBufferPreallocation=256
j2_maxPageReadAhead=128
j2_minPageReadAhead=8
- **Network:** Tune network parameters by using the **no** command. If SAS applications (such as SPDS, SPDE and/or SAS/CONNECT) are being used, set the following parameter value:
tcp_nodelayack=1
- **Maximum user process:** If you anticipate that the maximum number of processes for a single user will exceed 2000, increase the value of maxuproc to prevent abnormal delay or shutdown of SAS processes.
 - Increase the maxuproc setting by using the AIX **smit** or **chdev** command. for example:
chdev -l sys0 -a maxuproc=<new value>
- **User limits:** Increase the AIX user-process resource limits for SAS users and database instances, as appropriate (for example, **unlimited** or some tuned value for all resources (See **Note**).
 - In the /etc/security/limits file, set **-1** for all resources.
Note: The default user-process resource limits might be too low for SAS power users or large enterprise-class SAS deployments. When SAS processes end because of attempts to exceed these resource limits, the system administrator typically sets all user-process resource limits to **unlimited** (a numeric value of **-1**) for users who run SAS. The problem with this approach is that the increased multithreading and scalability support in newer versions of SAS, coupled with an **unlimited** setting for user-process resource limits, allows other users to potentially exhaust system resources such as processor, memory, I/O and paging space. Before increasing user-process resource limits, such as memory, to very high values, carefully consider the potential consequences.
- **Paging space:** Configure the paging space to include at least with the following suggestions:
 - Place paging spaces on dedicated disks to eliminate I/O contention.
 - Use multiple paging spaces that are spread across multiple disks.
 - Make the primary paging space hd6 a little bigger than the secondary paging spaces.
 - Ensure sufficient paging space to support all concurrent SAS processes (because the number of SAS processes can be dynamic, depending on application workload).
- **Volume groups (VGs):** Use the AIX Scalable or Big volume group.
 - The scalable VG implementation provides configuration flexibility with respect to the number of physical volumes (PVs) and logical volumes (LVs) that a given instance of the new VG type can accommodate. The

- configuration options allow any scalable VG to contain 32, 64, 128, 256, 512, 768 or 1024 disks and 256, 512, 1024, 2048 or 4096 LVs. You do not need to configure the maximum values of 1024 PVs and 4096 LVs when creating the VG to account for potential future growth. You can increase the initial settings later, as required.
- **Disk layout:** Minimize disk contention between SAS temporary space and data spaces.
 - Avoid disk contention by placing SAS temporary-space file systems and SAS data file systems on physically separate disks.
 - Use multiple storage-server controllers to further separate and isolate the I/O traffic between SAS temporary and data spaces.
 - Use multiple mount points for SAS file systems. Place the system operating system, SAS, user, SAS temporary space and SAS data file systems on separate physical disks.
 - Consider creating multiple SAS WORK areas that can be used by groups of SAS users.
 - Create separate JFS2 log files on separate physical disks for each SAS file system.
 - Spread the I/O workload across many physical disk spindles rather than across fewer, larger-capacity disks. Determine the sizing, based on the quantity of disks rather than disk capacity. Do not wrap luns around the same spindle sets.
 - Do not share disk spindles with a RDBMS.
 - **Host bus adapters (HBAs):** Use an adequate number of HBAs from storage to the host server to provide the required application bandwidth.
 - Consider high-performance storage channels, such as fibre-channel technology instead of slower mediums.
 - If possible, use dynamic multipathing to spread the I/O load across multiple adapters.
 - **Release behind mechanism for JFS2:** This feature allows the file system to release the file pages from file system buffer cache as soon as an application has read or written the file pages. This feature helps when the SAS application performs a great deal of sequential reads or writes and most often, once accessed, these file pages are not accessed again in the near future. This feature can be configured on a file system basis. When using the **mount** command, enable release-behind by specifying one of the three flags below:
 - release-behind sequential read flag (*-rbr*),
 - release-behind sequential write flag (*-rbw*),
 - release-behind sequential read and write flag (*-rbrw*).
 - **RAID:** Implement storage system RAID striping across multiple physical disks.
 - Use RAID10 or RAID5, depending on the level of redundancy and total capacity instead of the usable capacity that is needed for each file-system type.
 - Use LVM striping instead of concatenation.
 - **LVM striping:** When choosing the disk stripe or segment size, or array stripe size, note that AIX file systems are aligned on a 16 KB boundary.
 - A strip is the size of data to be written to each physical disk in the array. A stripe is the size of the full write across all the physical disks in the array. For example: stripe size x number of disks = stripe size
 - The AIX LVM stripe size that you can select from the **smit lv create** panel is actually the single strip size (not stripe). In other words, it is the size of data to be written to each of the array disks; it is not the full stripe size across all the physical disks. Consider using an LVM stripe size of 64K or 128K.
 - Synchronize SAS BUFSIZE with the storage-system stripe size and the AIX LVM stripe size (if using LVM striping) and VMM read-ahead increments.
 - Synchronizing I/O sizes streamlines I/O processing and reduces the number of I/O requests to the storage subsystem.
 - **AIX 5L thread tuning:** To improve SAS application performance, set the following parameter:


```
AIXTHREAD_SCOPE=S
```
 - **Resources:** For detailed tuning information, refer to the following publications:
 - *Best Practices for Configuring your IO Subsystem for SAS9 Applications*
<http://support.sas.com/rnd/papers/sgf07/sgf2007-iosubsystem.pdf>
 - *Tuning Guide for SAS 9 on AIX 5L*
ibm.com/support/techdocs/atmastr.nsf/WebIndex/PRS2304