AIX 6 best practice for SAS Enterprise Business Intelligence (SAS eBI) users on IBM POWER6

Joseph Pu
Alfredo Mendoza
Frank Bartucca
Harry Seifert
Frank Battaglia
IBM Corporation
ISV Business Strategy and Enablement
December 2008

Table of contents

Abstract

Monitoring each SAS component and understanding how users use the environment helps system administrators understand how to best optimize the infrastructure to deliver the best performance. In this paper, the authors focus on best practice suggestions and performance settings of the IBM AIX 6.1 operating system for a SAS Enterprise Business Intelligence (SAS eBI) deployment. They also discuss the advantages of IBM's virtualization capability. The tuning performance data in this paper was derived from the monitoring, analysis and tuning from a recent SAS 9.1.3 eBI benchmark at the SAS Enterprise Excellence Center (EEC). The benchmark was done on an IBM POWER6 4.7 GHz server running SAS 9.1.3 sp 4 and the AIX 6.1 operating system.

This test is based on SAS 9.1.3 SP4 on a 16 POWER6 processor IBM p570 system in October, 2008.

Introduction

The SAS paper available at http://support.sas.com/resources/papers/happyIT.pdf provides system management guidelines for system (hardware and operating system) resources from a SAS eBI application's perspective.

In this benchmark, the team targets a 500-user environment with varying types of workloads. The workload distribution is as follows:

- 10 SAS Enterprise Guide sessions developing reports.
- 20 advanced analytics sessions performing SAS statistical functions.
- 10 SAS Enterprise Guide sessions developing reports and SAS Data Integration Server code.
- 80 SAS Add-In for Microsoft® Office sessions.
- 80 SAS Web Report Studio ad hoc, single-user sessions.
- 70 SAS Web Report Studio static report sessions.
- 80 SAS Stored Process user sessions.
- 80 SAS OLAP cube viewer single-user sessions.
- 70 SAS Enterprise BI Server portal and dashboard viewer sessions.

A typical SAS BI solution is made up of the following components:
- SAS Metadata Server
- SAS OLAP Server
- SAS Server
- The mid-tier server

Each component has its own characteristics on how it uses the underlying resources. For example, the SAS Server is both processor and I/O intensive, whereas the OLAP Server is mostly processor intensive. So while some servers need to have I/O tuning done, some require a more dynamic approach of rebalancing processor resources.

In real-world environments, when a performance concern arises, the methodology the team put forth in this document will demonstrate the effectiveness of a heuristic approach that examines each SAS component, the resources that it runs on and the different tuning parameters that need to be modified to get the best overall performance.

Guidance on configuring the AIX6 kernel on IBM POWER6

AIX system general practices

It is always recommended to follow the best practices for environmental setup. The following list features general best practice recommendations for installing and running SAS 9 on IBM POWER™ processor-based servers with the AIX 6 operating system.

Use the latest version of AIX version with the latest maintenance level and e-fix.

Keep the Hardware Management Console (HMC) and microcode up to date

Use the "Microcode Survey and Update Tools"1 to keep the server and microcode up to date:
http://techsupport.services.ibm.com/server/mdownload/mcodetools.html

To check for various updates for AIX V6, HMC and microcode:
http://www14.software.ibm.com/webapp/set2/subscriptions/pqvcmjd

Consult your IBM Customer Engineers regarding latest microcode availability.

Use JFS2 on 64-bit AIX kernel

The IBM AIX file system, which is referred to as *Enhanced JFS* (JFS2), provides greater scalability than JFS. JFS2 is designed and optimized for the 64-bit kernel environment taking full advantage of 64-bit functionality. JFS2 is the default file system for a 64-bit kernel.

In general, SAS requires large rates of sequential disk I/O. Either AIX JFS or JFS2 can detect and exploit the read-ahead and write-behind characteristics of the application under normal file caching policy.

During the AIX installation, for example, from CD, on the Install Option panel select **3. Enable 64-bit kernel** and **4. Create JFS2 File Systems**.

Enable the POWER simultaneous multithreading hardware

Simultaneous multithreading designed into the IBM POWER6 processor is capable of exploiting instruction-level and thread-level parallelism. This hardware capability can be leveraged by the existing base of multithreaded applications. Inherent in the IBM POWER6 processor's simultaneous multithreading implementation is the capability to dynamically adjust usage of hardware resources. These features help to achieve a higher system utilization and throughput.

Note: Simultaneous multithreading is enabled as a default. Do **not** disable simultaneous multithreading.

Use a Host Ethernet Adapter

A Host Ethernet Adapter (HEA) is a physical Ethernet adapter that is integrated directly into the GX+ bus on a managed system. HEAs offer high throughput, low latency and virtualization support for Ethernet connections. HEAs are also known as *Integrated Virtual Ethernet (IVE) adapters.*

Unlike most other types of I/O devices, you can never assign the HEA itself to a logical partition (LPAR). Instead, multiple LPARs can connect directly to the HEA and use the HEA resources. This allows these LPARs to access external networks through the HEA without having to go through an Ethernet bridge on another LPAR.

It is not recommended that SASWORK or UTILLOC point to /var

File system /var is used for various purposes, such as storing temporary files, mail spool files, and all security logging information. Running out of /var space can cause SAS process from being abnormally ended.

Increase the value of *maxuproc* to prevent SAS processes from being abnormally terminated or delayed.

maxuproc is a kernel configuration option that limits the number of processes that nonroot users are allowed to have simultaneously active. The process table is sized based on this. Increase the setting of maxuproc using smit or the command: chdev –l sys0 –a maxuproc=<new value>.

It is recommended to start with the value 2000.

Increase user process resource limits for SAS users and database instance – with caution.

The default AIX user process resource limits might be too low for SAS power users or large enterprise-class deployments of SAS. When SAS processes are ending because of attempts to exceed user process resource limits, the tendency is for the system administrator to set all user process resource limits to **unlimited** (numeric value of -1) for users running SAS. The problem with this approach is that the increased support in newer versions of SAS for multithreading and greater scalability, coupled with an **unlimited** setting for user-process resource limits, allows ordinary 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.

Configuring paging space with the following suggestions:

Place Paging spaces on dedicated disk(s) to eliminate I/O contention.

Use multiple paging spaces spread over multiple disks.

Make the primary paging space hd6 a little bigger than the secondary paging spaces.

Ensure that the paging space is sufficient to support the number of concurrent SAS process as the number of SAS process can be dynamic depending upon the application workload.

Protect system processes from being terminated if AIX runs out of paging space

If AIX runs out of paging space, it starts ending processes. Some processes are system processes and they are critical to the operation of AIX. Some services become unavailable or AIX will hang or even crash if these system processes are ended. These system processes are owned by system-defined users with UIDs of low numeric value. Here is an example of some of these system defined users and their IDs

Name	UID
root	0
daemon	1
bin	2

```
sys              3
adm              4
uucp             5
```

Determine the numeric value of the highest UID that you want to protect during exhausted page space conditions. Do this by running the following command to sort the /etc/passwd file by ascending UID.

```
sort -t : -n +2 /etc/passwd | more
```

Here are the first 13 lines of output for a typical system:

```
root:!:0:0::/:/usr/bin/ksh
daemon:!:1:1::/etc:
bin:!:2:2::/bin:
sys:!:3:3::/usr/sys:
adm:!:4:4::/var/adm:
uucp:!:5:5::/usr/lib/uucp:
invscout:!:6:12::/var/adm/invscout:/usr/bin/ksh
nuucp:*:7:5:uucp login user:/var/spool/uucppublic:/usr/sbin/uucp/uucico
pconsole:*:8:0::/var/adm/pconsole:/usr/bin/ksh
lpd:!:9:4294967294::/:
mqm:*:10:15::/var/mqm:/usr/bin/ksh
lp:*:11:11::/var/spool/lp:/bin/false
guest:!:100:100::/home/guest:
```

The user "guest" usually has the lowest number UID for regular users.

To protect UIDs 11 and lower set nokilluid=12 so that UIDs lower than 12 will be protected. The AIX default setting for nokilluid is 0, which means that no UIDs are protected.

```
# vmo -o nokilluid=12
```

To avoid exhausted page space conditions in the first place make sure that enough page space is allocated to support the highest anticipated use of page space by the workload.

Use the appropriate number of host bus adapters from the storage to the host server to provide the required front end application bandwidth

Many SAS I/O workload patterns can be throughput intensive. However, this is not always the case for all SAS applications or necessarily true during the entire SAS application's execution.

High-performance storage channels should be considered such as Fibre Channel technology over slower mediums.

Use dynamic multipathing if possible to spread the I/O load over multiple adapters. Otherwise, be careful when locating SAS data libraries on mount points.

Spread the I/O workload across many physical disk spindles rather than fewer larger capacity disks

Provide better I/O performance by sizing for quantity of disks instead of capacity of disks.

Implement storage system RAID striping across multiple physical disks.

Note: In general testing, it has been observed that there is a slight performance advantage to using RAID 10 over RAID 5 for SAS temp space file systems. This is not necessarily the case for other SAS file systems. Use RAID 10 or RAID 5 depending on the level of redundancy and total capacity compared to usable capacity that is required for each type of file system.

Use LVM stripping instead of concatenation.

Minimize disk contention between SAS temporary space and data spaces

Avoid disk contention by placing SAS temp 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 temp and data spaces. This also provides a more robust disk back end to handle I/Os.

Use multiple mount points for SAS file systems. Place system O/S, SAS, user, SAS temp, and SAS data file systems on separate physical disk.

If multiple users will share the SAS temp space (SASWORK) and increase disk or file system contention consider separating each user into separate SAS temp space file systems with physically separate disk.

Create separate JFS2 log files on separate physical disks for each SAS file system.

Isolate SAS I/O from non-SAS workloads

In general, SAS applications can be highly sequential large I/O workloads. Disk contention between SAS applications and other non-SAS small I/O random IOPS applications will increase service times of all applications and decrease I/O performance.

Use the AIX Scalable volume group or Big volume group with mklv –T 0 option to avoid the logical volume control block reserve of the first 4K of space.

With the LVCB present, the first data block will start with a 4K offset. When LVCBs exist on an lv, they can cause I/Os to span multiple physical volumes due to this offset.

Be mindful that AIX file systems are aligned on a 16K boundary when choosing the disk stripe or segment size or array stripe size.

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.
Example: strip size x number of disks = stripe size.

Note: The AIX LVM stripe size that can be selected from the smit lv create panel is actually the single strip size (not stripe) or size of data to be written to each of the array disks and not the full stripe size across all the physical disks.

Sync up SAS BUFSIZE with the storage system stripe size and the AIX LVM stripe size (if using LVM striping), and VMM read-ahead increments.

Syncing of I/O sizes results in more-efficient I/Os while reducing the total number of I/O requests to the storage subsystem.

Note: LVM striping might or might not provide better performance depending on the SAS application or the storage subsystem configuration. Testing your specific application is recommended.

AIX kernel components tuning

CPU threads

The AIX environment variable "AIXTHREAD_SCOPE=S" has shown to improve the performance of the SAS application.

AIXTHREAD_SCOPE={P|S} is an AIX environment variable. It is set to "P" for process-wide scope by default.

AIXTHREAD_SCOPE=S changes the scheduling policy from process-based scheduling (m:n) to system based scheduling (1:1). In 1:1 model, each user thread is mapped one-to-one to an AIX kernel thread, and each user thread runs on one virtual processor. In an SMP environment, system scope generally performs better than process scope. With process scope, the pthreads library manages internal thread scheduling using virtual processors which has higher overhead.

To set this environment variable, use **export AIXTHREAD_SCOPE=S**.

VMM

The AIX virtual memory manager (VMM) is used to control memory resources. Virtual memory provides two major functions. First, it provides a security infrastructure for memory. Second, it allows the active amount of memory to be greater than the real memory in the system. This is accomplished by keeping the active portions of storage in real memory and spilling the less active portions to disk.

Virtual memory is managed on the basis of pages and segments. Pages are the granules of storage allocation. Groups of pages are managed within segments. Segments are divided into three types, based on the location of the storage that is used to back the page: *persistent* (JFS-backed), *working* (temporary or nonpersistent storage) and *client* (JFS2, NFS, and others). Virtual-memory segments are classified as containing either computational (nonpersistent) or file (persistent). The delineation of storage types between computational and file is used to balance the types of pages stolen by page-replacement algorithm.

Before AIX 6, there are a couple of VMM kernel parameters that need to be changed from their default value in order to support enterprise system loads. In AIX6, a new set of defaults are set to provide better out-of-box VMM performance. The following table shows the parameters and their new values. The benefit of the new values makes it less likely to have an application's computational pages (contain application code) to be paged out. As the result of these default value changes, AIX 6 users should use the default. These values can be changed using the **vmo** command.

vmo name	AIX 6 default values
minperm%	3
maxperm%	90
maxclient%	90
Lru_file_repage	0
page_steal_methos	1

Large sequential writes can cause a system to become unresponsive for interactive applications.

The VMM file I/O pacing is the mechanism to improve this situation. Before AIX 6, the default for this feature is **off**. AIX 6 sets the defaults to **on**. The table shows the parameters and their values.

sys0 name	AIX 6 default values
minpout	1
maxpout	1

These two parameters are associated with device "sys0". Change them by using the **chdev** command.

I/O

Use the appropriate number of host bus adapters from the storage to the host server to provide the required front end application bandwidth.

Many SAS I/O workload patterns can be throughput intensive. High performance storage channels devices should be considered such as Fibre Channel technology over slower mediums.

AIX 6 (MPIO) provides multipathing to spread the I/O load over multiple adapters. Based on disk I/O needs, the SAS Server tier was configured with multiple (3) fibre channels to provide better I/O throughput.

Spread the I/O workload across many physical disk spindles rather than fewer larger capacity disks.

This provides better I/O performance by sizing for quantity of disks instead of large capacity of disks.

If using a storage subsystem's RAID (5 or 10), using LVM stripping at the OS level is redundant. Instead, in the AIX LVM creation process, select **max_range** for LPARs. This spreads out the construction of LVM across all physical disks within the volume group.

Minimize disk contention between SAS temporary space and data spaces

Avoid disk contention by placing SAS temp 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 temp and data spaces. This also provides a more robust disk back end to handle I/Os.

Use multiple mount points for SAS file systems. Place system OS, SAS, user, SAS temp, and SAS data file systems on separate physical disk.

If multiple users will share the SAS temp space (SASWORK) and increase disk or file system contention consider separating each user into separate SAS temp space file systems with physically separate disk.

Create separate JFS2 log files on separate physical disks for each SAS file system.

Isolate SAS I/O from non-SAS workloads

In general, SAS applications can be highly sequential large I/O workloads. Disk contention between SAS applications and other non-SAS small I/O random IOPS applications increase service times of all applications and decrease I/O performance.

Use the AIX Scalable volume group or Big volume group with mklv –T 0 option to avoid the logical volume control block reserve of the first 4K of space.

With the LVCB present the first data block will start with a 4K offset. When LVCBs exist on an lv, they can cause I/Os to span multiple physical volumes due to this offset.

Disk (LUN) on storage subsystem

In this section, the focus is on tuning the middle layers consisting of hdisk and adapter device drivers. The goal is to improve simultaneous I/O capability and realize efficient queue handling. See the following table for some of the parameters that can affect disk and adapter performance. In general, SAS applications benefit from careful consideration and tuning of these parameters.

Parameter	Description
Max_xfer_size	FC adapter maximum I/O that will be issued.
Max_transfer	Disk maximum I/O that will be issued.
queue_depth	• Disk maximum number of simultaneous I/Os. • The default is 20 but can be set as high as 256 for IBM Enterprise Storage Server (ESS), IBM System Storage DS6000™ and DS8000™.
Num_cmd_elems	FC adapter maximum number of simultaneous I/Os. The default is 200 per adapter but can be set up to 2048.
qdepth_enable	• Subsystem Device Driver (SDD) data path optimizer (dpo) device queuing parameter. • The default is yes. A setting of no disables SDD queuing. • Use this with ESS, DS6000 and DS8000 storage.
lg_term_dma	Long term DMA - Memory area the FC adapter uses to store I/O commands and data.
LTG	• AIX volume group Logical Track Group parameter. • LTG specifies the largest I/O the LVM will issue to the device driver. • The LTG dynamically matches the disk maximum transfer parameter.

Both the disk and adapter have maximum transfer parameters that can be adjusted to handle larger I/O, reduce I/O splitting and coalesce I/O as it moves up and down the stack. In addition, both have I/O queues that can be adjusted to accept additional I/Os.

It is important to understand the I/O characteristics of the application in order to properly tune within the I/O stack layers. If the SAS application is predominantly large I/Os, then the application performance can benefit from adjusting maximum transfer sizes, long term DMA, and the LTG. The recommended starting values for a large I/O highly sequential workload are lg_term_dma=0x800000, and max_xfer_size=0x200000.

It is recommended to increase the num_cmd_elems for the FC adapter from the default (can start at 400). Some of these parameters require a system reboot to take effect. For additional guidelines, see the tuning guide links found in this documents additional information section.

Use the following commands to display and modify disk and adapter parameters and settings.

Disk – max_transfer, queue_depth

- 'lquerypv –M hdisk#' displays maximum I/O size a disk supports.
- 'lsattr –El hdisk#' displays current disk values.
- 'lsattr -Rl max_transfer hdisk#' displays allowable values.
- 'chdev –l hdisk# -a max_transfer=value –P' modify current disk values
- Note: The device should be in an offline/disabled state before changing any parameters. Then cfgmgr will need to be issued.

Adapter – max_xfer_size, lg_term_DMA, num_cmd_elems

- 'lsattr –El fcs#' displays current value.
- 'chdev –l fcs# -a max_xfer_size=value -P' modify current value.
- Note: The device should be in an offline/disabled state before changing any parameters. Then cfgmgr will need to be issued.

File system (JFS2) tuning

The AIX file system (FS) is called *Journaled File System (JFS)* or *enhanced Journaled File System (JFS2)*. An FS presents a logical view of files and directories linked together to form a hierarchical tree structure.

In general, SAS applications have a great deal of large sequential read and write disk I/O. If the workload has many large I/Os to a file system (e.g. large sequential I/O to JFS2), the I/Os might be in a bottleneck at the file system level while waiting for a construct called *bufstructs.* The bufstructs for JFS2 is dynamic and the number of *bufstructs* per file system can be increased. The file system must be remounted for the new value to take effect.

The I/O characteristics of SAS usually create the situation where VMM read-ahead, and write-behind algorithm can be used to improve the performance of sequential file access. The parameters listed in table 2 can be tuned using the **ioo** command.

See the following table for the most frequently used AIX FS tuning parameters.

Parameters	Description
j2_dynamicBufferPreallocation	This tunable (16 by default) specifies the number of 16k chunks to preallocate when the file system is running low of bufstructs. The parameter should be tuned in conjunction with the following *j2_nBufferPerPagerDevice*. This should be tuned first. Tune it by increasing the value by 16 each time. Symptom: Vmstat –v

|	
	… external pager file system I/Os blocked with no fsbuf (increasing value)
j2_nBufferPerPagerDevice	When vmstat -v shows a shortage of file system buffstructs for a JFS2, the *j2_dynamicBufferPreallocation* tunable should be tuned first before making any change to *j2_nBufferPerPagerDevice* parameter. In Enhanced JFS, the number of bufstructs is specified with the *j2_nBufferPerPagerDevice* parameter. The default number of bufstructs for a JFS2 is currently 512. The number of bufstructs per JFS2 (*j2_nBufferPerPagerDevice*) can be increased using the **ioo** command. The value takes effect only when a file system is mounted. Tune it by increasing the value 512 each time. Symptom: Vmstat –v … external pager file system I/Os blocked with no fsbuf (increasing value)
j2_maxPageReadAhead	This tunable (128 by default) specifies the upper limit for AIX JFS2 prefetching. It affects efficiently when doing large I/O. Sequential Read-Ahead can be tuned separately for Enhanced JFS. JFS2 Page Read-Ahead can be tuned with *j2_minPageReadAhead* and *j2_maxPageReadAhead*. Tune it by increasing the value by 128 each time. However, for most file sizes there is no advantage to increasing *j2_minPageReadAhead*.
j2_nPagesPerWriteBehindCluster	For Enhanced JFS, you can use the **ioo -o j2_nPagesPerWriteBehindCluster** command to specify the number of pages to be scheduled at one time. The default number of pages for an Enhanced JFS cluster is 32, which implies a default size of 128 KB for Enhanced JFS. You can use the **ioo -o j2_nBufferPerPagerDevice** command to specify the number of file system bufstructs. The default value is 512. For the value to take effect, the file system must be remounted. Tune it by increasing the value by 32 each time.

New network device

In an LPAR environment, it is common to create a special LPAR, called Virtual I/O Server (VIOS) partition, to provide the shared Ethernet connectivity. The newly available Host Ethernet Adapter (HEA) is also called Integrated Virtual Adapter (IVE) on IBM POWER6 systems, which allows multiple LPARs to share the same physical adapter. Thus network traffic does not have to go through the VIOS layer and becomes more efficient.

For POWER5 systems:
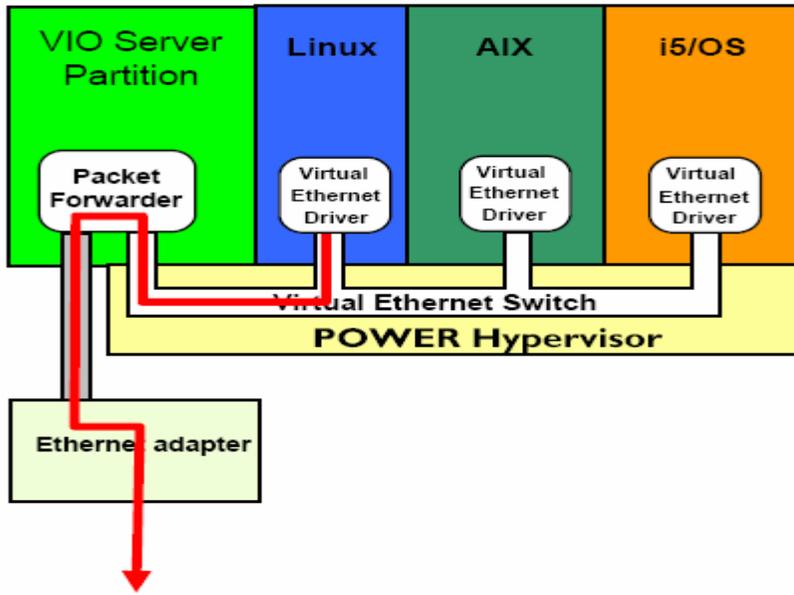
For POWER6 systems:
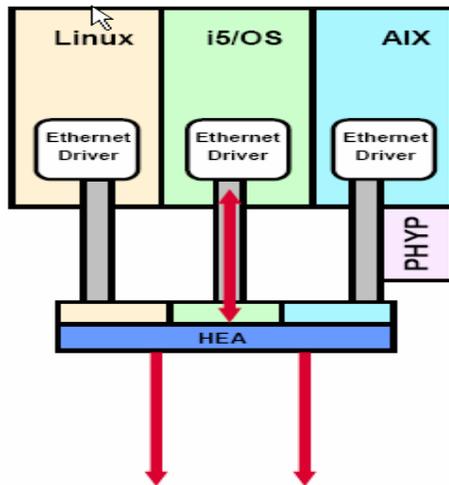
HEA device can be added to a LPAR using either HMC or Integrated Virtualization Manager (IVM). A case of using HMC is demonstrated in the following screen capture.

The traditional AIX performance-tuning methodology

At every stage in the performance tuning process, (illustrated in the following figure as a progression of steps from lower left to upper right) the usage of various hardware components (processor, memory, disk I/O, and network I/O) within the system are examined. Of course, the emphasis on one component or another depends on your environment. This figure also shows the performance objectives for each system component.
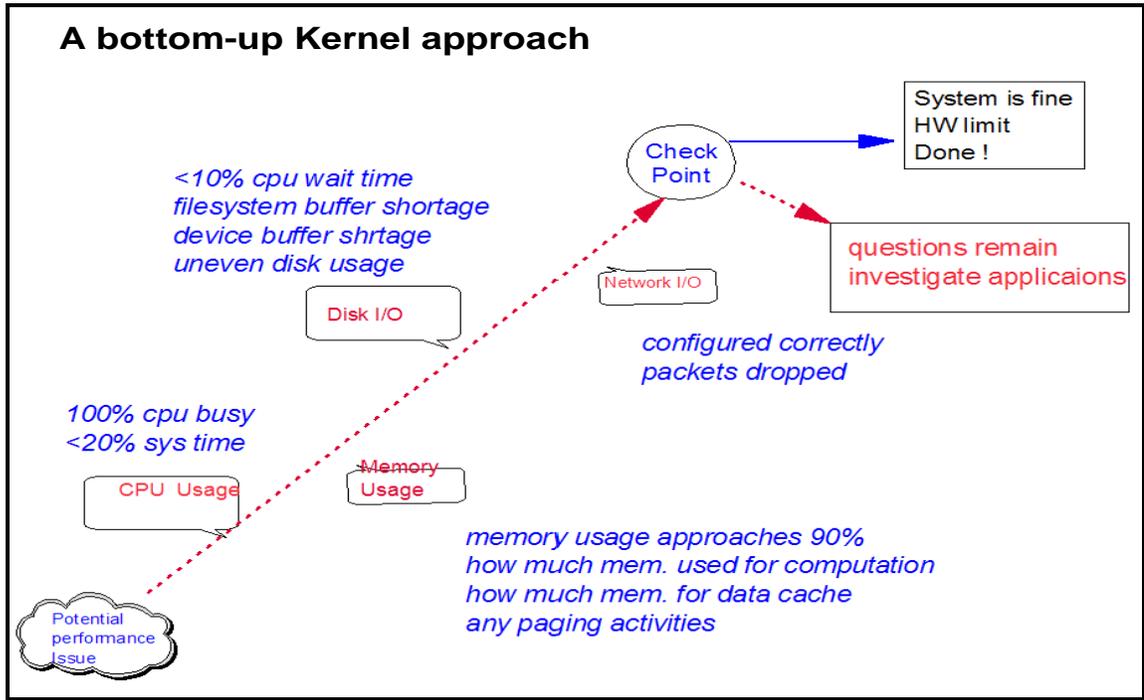
This flowchart was based on a single OS instance. The same procedure can be adopted in a LPAR environment. The first objective is to achieve a reasonable processor and then memory resource balance across all LPARs. Then focus will be directed to individual LPARs that might need fine-tuning within the OS instance.

The top-down approach is proved to be very effective in managing system resources (mainly CPU and memory) across many LPARs on the same physical system.

Common native AIX system monitoring commands

system	CPU	Memory	I/O	Network	LPAR
ps	curt	svmon	iostat	netpmon	lparstat
sar	trace	vmstat	filemon	netstat	mpstat
tprof	truss				
	vmstat				

ps:
   The **ps** command displays information about active processes to standard output.

sar:
   The **sar** command gathers statistical data about the system.

tprof:
   The **tprof** command is a versatile profiler that provides a detailed profile of processor usage by every process ID and name.

curt:
   The CPU Utilization Reporting Tool (**curt**) command converts an AIX trace file into a number of statistics related to CPU utilization and either process, thread or pthread activity.

trace:
   The **trace** daemon configures a trace session and starts the collection of system events. The data collected by the trace function is recorded in the trace log. A report from the trace log can be generated with the **trcrpt** command.

truss:
   Traces a process's system calls, dynamically loaded user level function calls, received signals and incurred machine faults.

vmstat:
   The **vmstat** command reports statistics about kernel threads, virtual memory, disks, traps and processor activity.

svmon:
   Captures and analyzes a snapshot of virtual memory.

iostat:
   The iostat command is the fastest way to get a first impression, whether or not the system has a disk I/O-bound performance problem.

filemon:
   Monitors the performance of the file system, and reports the I/O activity on behalf of logical files, virtual memory segments, logical volumes, and physical volumes.

netpmon:
   The netpmon command uses the trace facility to obtain a detailed picture of network activity during a time interval.

netstat:
   The netstat command is used to show network status.

lparstat:
  The **lparstat** command provides a report of LPAR related information and utilization statistics.

mpstat:
  Collects and displays performance statistics for all logical CPUs in the system.

Single-instance AIX monitoring

topas

The **topas** command reports selected statistics about the activity on the local system. The command uses the curses library to display its output in a format suitable for viewing on an 80x25 character-based display or in a window of at least the same size on a graphical display.

The variable part of the topas display can have one, two, three, four, or five subsections. If more than one subsection displays, they are always shown in the following order:

1. Processor
2. Network interfaces
3. Physical disks
4. File system
5. Workload Manager classes
6. Workload Partitions (WPARs)
7. Processes

When the **topas** command is started, it displays all subsections for which hot entities are monitored.

The **topas** command comes with the default AIX installation. So there is no additional installation required.

```
                                   aixterm
Topas Monitor for host:     ibmaix6          EVENTS/QUEUES      FILE/TTY
Mon Oct  6 10:04:58 2008     Interval:  2     Cswitch    3511   Readch    949.0K
                                              Syscall    5409   Writech  1428.0K
Kernel     2.3   |#                      |    Reads       941   Rawin          0
User      14.3   |#####                  |    Writes      314   Ttyout       527
Wait       0.0   |                       |    Forks         0   Igets          0
Idle      83.4   |###################### |    Execs         0   Namei        491
                                              Runqueue    2.5   Dirblk         0
Network  KBPS     I-Pack  O-Pack   KB-In   KB-Out  Waitqueue  0.0
Total   2093.1    671.0   394.0   676.8  1416.3                MEMORY
                                              PAGING             Real,MB    24576
Disk    Busy%     KBPS     TPS KB-Read KB-Writ  Faults      63   % Comp      32.2
Total     5.0     50.0    12.5     0.0    50.0  Steals        0   % Noncomp  15.5
                                              PgspIn        0   % Client   15.5
FileSystem        KBPS     TPS KB-Read KB-Writ  PgspOut       0
Total            555.1    804.0   507.6    47.5  PageIn        0   PAGING SPACE
                                              PageOut       0   Size,MB      512
Name            PID  CPU%  PgSp Owner         Sios          0   % Used       5.5
java         209378   8.2 283.1 root                           % Free      95.5
java         184800   3.4 280.0 root          NFS (calls/sec)
java         213468   2.8 273.9 root          SerV2         0   WPAR Activ     0
java         114736   1.3 275.1 root          CliV2         0   WPAR Total     0
httpd        303394   1.1   8.7 nobody        SerV3         0   Press: "h"-help
java         143686   0.5 164.6 root          CliV3         0          "q"-quit
httpd        295194   0.3   8.0 nobody
```

nmon

This cost-free tool gives you a huge amount of information all on one screen. You can get a wealth of performance statistics. Why use five or six tools when one free tool can give you everything you need?

The nmon tool is designed for AIX and Linux® performance specialists to use for monitoring and analyzing performance data, including:

- CPU utilization
- Memory use
- Kernel statistics and run queue information
- Disks I/O rates, transfers, and read/write ratios
- Free space on file systems
- Disk adapters
- Network I/O rates, transfers, and read/write ratios
- Paging space and paging rates
- CPU and AIX specification
- Top processors
- IBM HTTP Web cache
- User-defined disk groups
- Machine details and resources
- Asynchronous I/O — AIX only
- Workload Manager (WLM) — AIX only
- IBM ESS disks — AIX only
- Network File System (NFS)
- Dynamic LPAR (DLPAR) changes — only IBM pSeries® p5 and IBM OpenPower® for either AIX or Linux

```
                              X Desktop
 ┌nmon12e────────j=JFS──────────Host=ibmaix5──────────Refresh=2 secs───10:57.16─┐
 │ CPU-Utilisation-Small-View                  EntitledCPU=  3.00 UsedCPU=  0.037│
 │Logical  CPUs        0──────────25──────────50──────────75──────────100        │
 │CPU User%  Sys% Wait% Idle%│                                                    │
 │  0  0.0   2.5   0.0  97.5│s       >                                           │
 │  1  0.0   0.0   0.0 100.0│>                                                   │
 │  2  0.0   0.0   0.0 100.0│>                                                   │
 │  3  0.0   0.0   0.0 100.0│>                                                   │
 │  4  0.0   0.0   0.0 100.0│>                                                   │
 │  5  0.0   0.0   0.0 100.0│>                                                   │
 │  6  0.0   0.0   0.0 100.0│>                                                   │
 │  7  0.0   0.0   0.0 100.0│>                                                   │
 │  8  0.0   0.0   0.0 100.0│>                                                   │
 │  9  0.0   0.0   0.0 100.0│>                                                   │
 │ 10  0.0   0.0   0.0 100.0│>                                                   │
 │ 11  0.0   0.0   0.0 100.0│>                                                   │
 │EntitleCapacity/VirtualCPU +───────────│───────────│───────────│───────────+  │
 │ EC  0.2   0.7   0.0   0.3│──────────────────────────────────────────────────│
 │ VP  0.1   0.4   0.0   0.1│──────────────────────────────────────────────────│
 │EC=  1.2%  VP=   0.6%     +──No Cap───│──Folded=3──│───────────100% VP=6 CPU+  │
 │ Shared-CPU-Logical-Partition                                                  │
 │Partition:Number=5 "ibmaix5wrkspace"                                           │
 │Flags:LPARed DRable SMT-bound Shared UnCapped PoolAuth Migratable Not-Donating.│
 │Summary: Entitled= 3.00  Used     0.04 ( 1.2%)  0.2% of CPUs in System         │
 │         PoolCPUs=15    Unused   14.9271       0.2% of CPUs in Pool            │
 │CPU-Stats────────  Capacity──────────────  ID-Memory──────────────────        │
 │max Phys in sys  16  Cap. Processor Min  0.10  LPAR ID Group:Pool   32773:0    │
 │Phys CPU in sys  16  Cap. Processor Max  7.00  Memory(MB) Min:Max  1024:49152  │
 │Virtual Online    6  Cap. Increment     0.01  Memory(MB) Online    36864       │
 │Logical Online   12  Cap. Unallocated   0.00  Memory Region LMB    256MB min   │
 │Physical pool    15  Cap. Entitled      3.00  Time────────────────Seconds      │
 │SMT threads/CPU   2   -MinReqVirtualCPU 0.10  Time Dispatch Wheel     0.0100    │
 │CPU─────────Min-Max  Weight───────────────────  MaxDispatch Latency    0.0100  │
 │Virtual       1    7  Weight Variable      128  Time Pool Idle      14.92030    │
 │Logical       1   14  Weight Unallocated     0  Time Total Dispatch  0.0365     │
 │────────────────────────────────────────────────────────────────────────────  │
 │Event= 0 ───     ───     SerialNo Old=───   Current=C08B42 When=───            │
 │────────────────────────────────────────────────────────────────────────────  │
 │Shared_Pools   MaxPoolCapacity=15.00 MyPoolMax =15.00 SharedCPU-Total=15.00    │
 │SharedCPU=15   EntPoolCapacity=11.00 MyPoolBusy= 0.10 SharedCPU-Busy = 0.10    │
 │ Fibre-Channel-Adapter                                                         │
 │        Adapter   Receive  Transmit     -Requests-        --Size KB--          │
 │Number Name       KB/s      KB/s        In     Out        In      Out          │
 │    1      fcs0    0.0      0.0         0.0    0.0        0.0      0.0          │
 │    2      fcs1    0.0      0.0         0.0    0.0        0.0      0.0          │
 │    3      fcs2    0.0      0.0         0.0    0.0        0.0      0.0          │
 │    4      fcs3    0.0      0.0         0.0    0.0        0.0      0.0          │
 │    5      fcs4    0.0      0.0         0.0    0.0        0.0      0.0          │
 │    6      fcs5    0.0      0.0         0.0    0.0        0.0      0.0          │
 │        Totals     0.0      0.0 MB/s    0.0    0.0                             │
 └────────────────────────────────────────────────────────────────────────────┘
```

AIX monitoring across multiple LPARs

Advantage of LPAR

The Dynamic LPAR (DLPAR) technology has provided flexibility for IBM customers to better use the system resources to meet their business needs.

It is also a key component to support a business' server consolidation strategy.

It delivers virtualization on POWER processor-based systems, designed to drive maximum energy efficiency, increase resource utilization, and lower operating costs, helping companies respond immediately to new business demands.

SAS solution consists of many components. This architecture is ideal for deploying in a LPAR environment. Each LPAR itself provides the software isolation but also be able to share the system resources (processor, memory, HEA) in a virtual and dynamic fashion.

lparmon is an IBM Power Systems monitoring tool the specifically designed for the multiple LPAR environment. Traditional instance based tools, such as *sar* or *vmstat* limit the monitoring scope within the single OS instance. In a typical multi-tiered SAS deployment, each SAS component is in a different tier and has different run time resource usage profile. An effective performance monitoring approach requires a tool that can offer a comprehensive view across all LPARs at the same time. Thus, it should provide a clear view to see the effects of any single OS tuning on the whole deployment environment.

This is the advantage of having a single tool to view all systems in a glance.
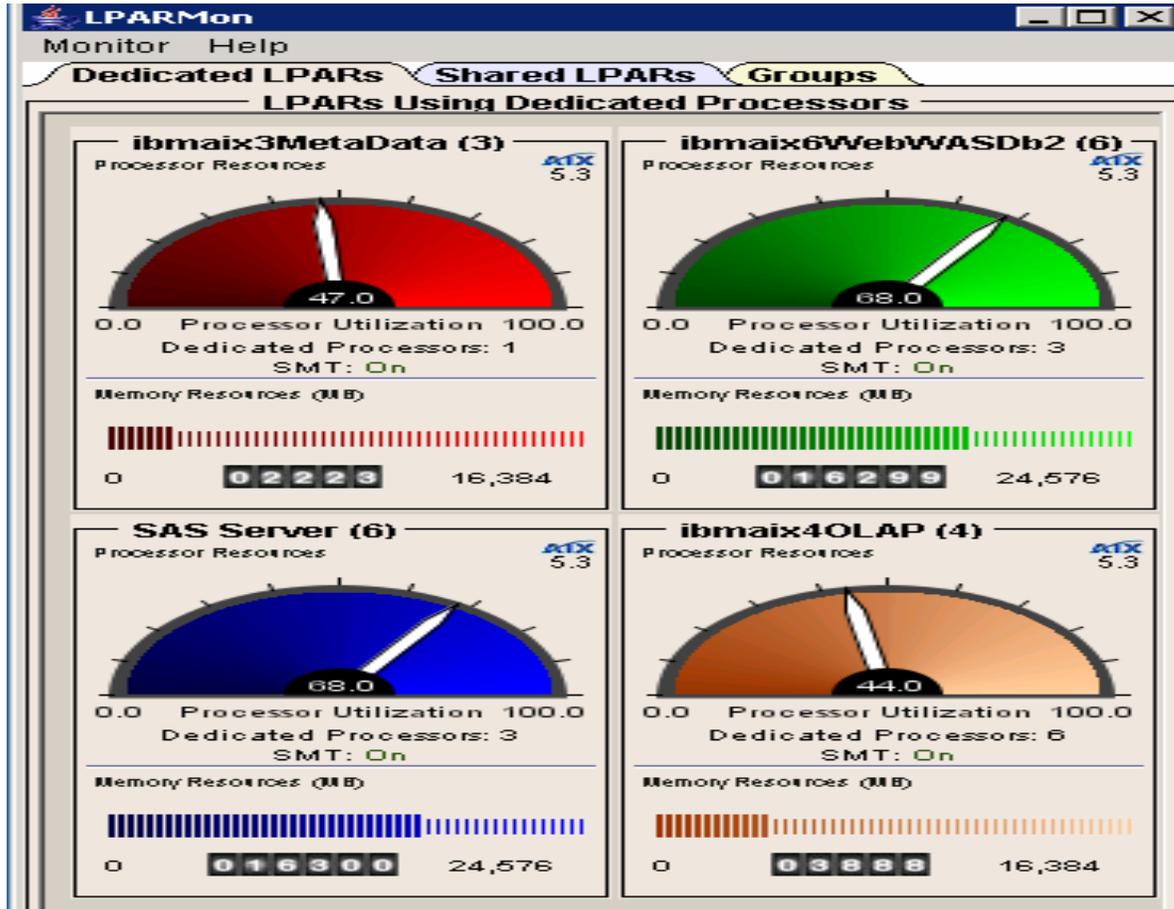
In this lparmon snapshot, it contains system information about 4 LPARs for this multi-tiered SAS eBI deployment.
    "ibmaix3" runs as a SAS Metadata server.
    "ibmaix4" runs as a SAS OLAP server.
    "SAS Server" runs SAS eBI server
    "ibmaix6" runs IBM HTTP, IBM WebSphere® and IBM DB2® servers.

Each contains the LPAR processor and memory resource information. It also uses easy to understand GUI (dial) to represent the real-time processor resource usages. For example, the SAS Server is using 68 percent of the processor capacity and 16.3 GB out of a total of 24 GB memory allocated to the LPAR.

The following two graphs show the effect of monitoring multiple LPARs using the topas tool.
Because topas is single AIX instance based, to monitor four LPARs, you need to run topas on each AIX instance and put them together to monitor the real-time condition. This is a rather busy graph to work with.

Another way to see all four LPARs runtime processor and memory usage is to run topas with the option to provide a simplified cross LPARs view. You only need to run one instance of topas from one of the LPARs.

Example: Tuning processor and memory based on the lparmon tool and using the HMC

LPAR "ibmaix4" has two dedicated processors and 4 GB memory. Processing power is fully consumed. It uses close to 1.9 GB of memory out of total of 4 GB.
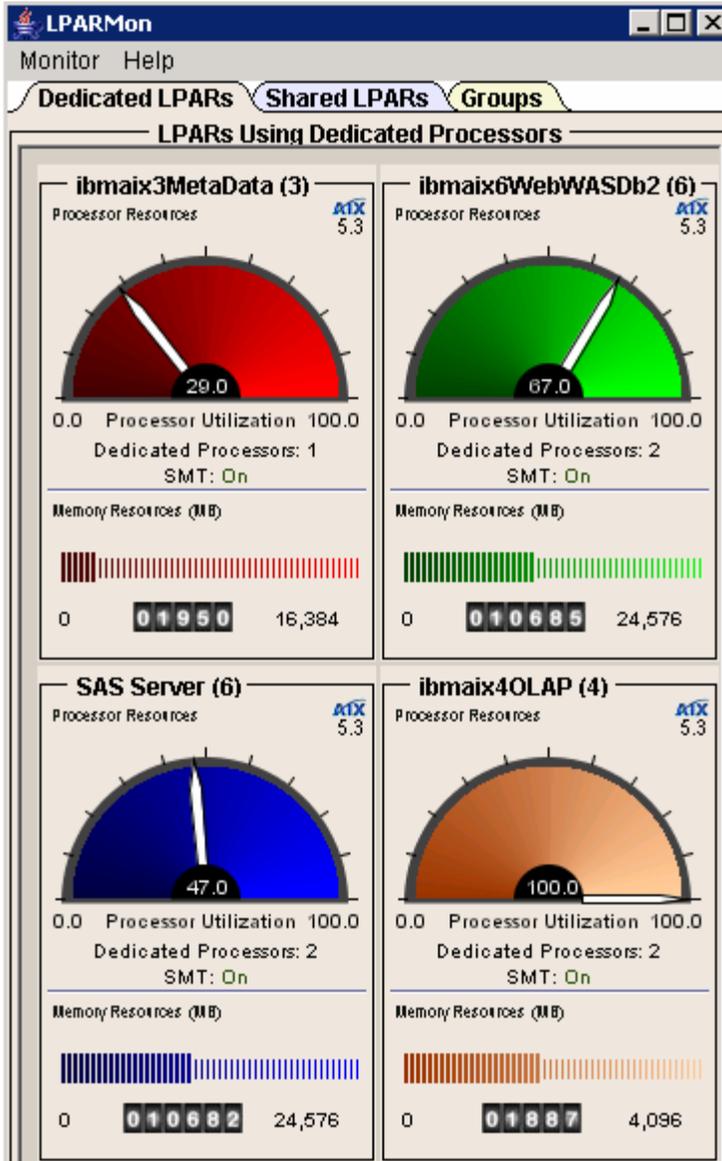
Assume the user wants to handle more load though out this four-LPAR deployment. In order to do that, the LPAR ibmaix4OLAP processors and memory need to be increased.

This can be accomplished at real time using the HMC, whose operations are presented though a GUI.

Using the HMC to manage processor resource

Select LPAR **ibmaix4OLAP -> Dynamic Logical Partition -> Processor -> Add or Remove**. Then follow the interface to add three more processors (screen capture omitted).

Using the HMC to manage memory resource

To add more memory in real time using the HMC to add 12 GB more memory to LPAR ibmaix4OLAP, perform the same basic steps.

LPAR ibmaix4 has five processors and 16 GB memory now and can handle more workload

After you have added more processors and memory resource to LPAR ibmaix4OALP, the run-time processor usage is not 100 percent anymore (78 percent). And there is plenty of memory available.
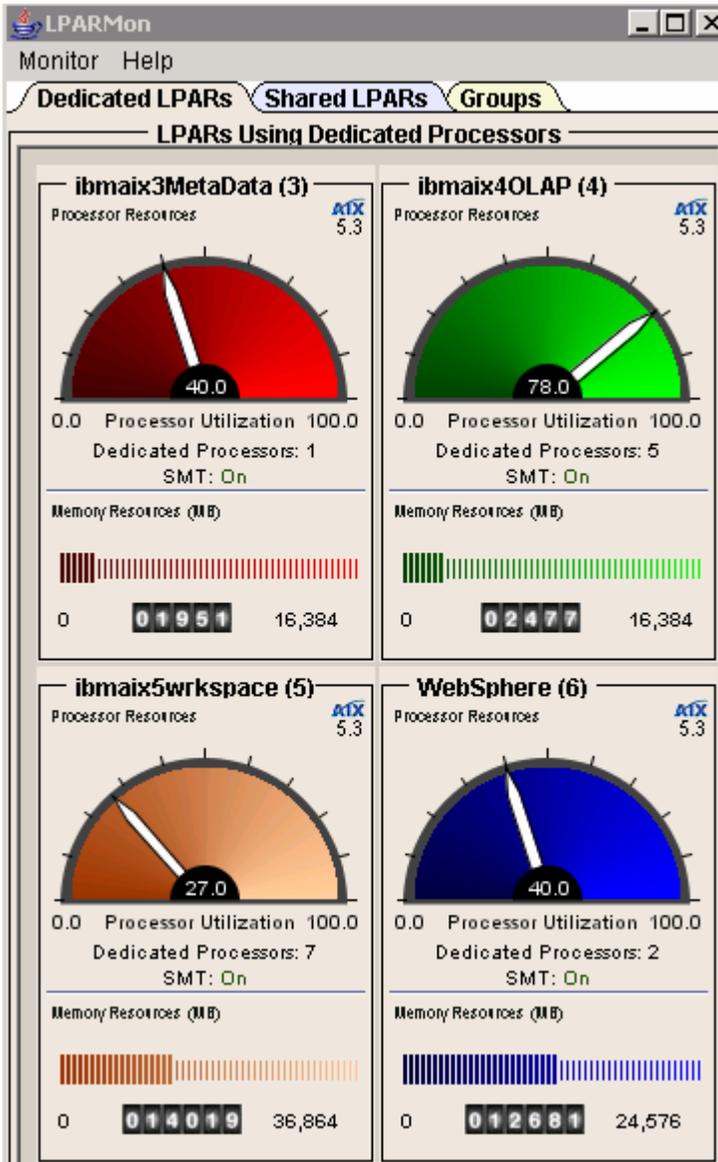
Example: Tuning AIX MPIO managed disks for load balancing across all fibre channels using the nmon tool

With AIX Multiple Path I/O (MPIO), a logical disk (hdisk) can have multiple connections, or paths to the physical device.

This means there should be at least two fibre-channel paths between the host and the storage subsystem to have a balanced fibre-channel load.

By default, I/O traffic is not balanced across multiple fibre channels. Start nmon and then enter ^ to bring up the fibre-channel monitor.

```
9.3.245.18 - PuTTY
-nmon12e———U=Top-with-WLM———Host=lsvlab018———Refresh=2 secs——04
 Fibre-Channel-Adapter
        Adapter   Receive Transmit      -Requests-      --Size KB--
Number Name         KB/s     KB/s       In     Out       In      Out
     1      fcs0     0.0      0.0        0.0     0.0      0.0       0.
     2      fcs1     0.0 132326.9        0.0  1033.8      0.0      128
         Totals     0.0    129.2 MB/s    0.0  1033.8
```

To balance the I/O across the fibre channels, you need to configure each hard disk (hdisk) that the fibre channels serve. In this example, you know the hdisks that is served by fibre channels fcs0 and fcs1 belong to the volume group of testvg.

To change the hdisk device value, the volume group `testvg` needs to be `varyoffvg` to free the hdisks.

```
9.3.245.18 - PuTTY
root@lsvlab018 /tmp: > lspv
hdisk0         00cbe24deed99729            rootvg          active
hdisk1         00cbe24def031084            rootvg_aix2_3
hdisk2         00cbe24def038ad4            rootvg_Lop_6
hdisk3         00cbe24ddfe15005            sybasevg        active
hdisk4         00cbe24ddf281504            testvg          active
hdisk5         00cbe24ddf2815f9            testvg          active
hdisk6         00cbe24ddf2816e4            testvg          active
hdisk7         00cbe24d66a997d2            testvg          active
root@lsvlab018 /tmp: >
```

Use the smitty tool to change each hdisk's device values.

Set Algorithm to `round_robin` and Reservation Policy to `no_reserve`, respectively.

After the setting, the nmon tool shows that the disk I/O traffic is balanced across these two fibre channels.

Initially sizing the hardware capacity needed for multi-tiered eBI deployment

CPU

In this SAS eBI testing on a 16-core IBM POWER6 system, a heavy load (call it X) was conducted on a multi-tiered SAS deployment. This deployment used four LPARs. The rootvg of each LPAR was served by a VIOS partition. The load X consumed about total of 15 core processing power. The following table provides the processing power distribution among these 4 LPARs. This can serve as a processing sizing scale for different load situations:

LPAR	Cores
SAS Metadata Server	1
SAS OLAP Server	6
SAS Server	6
Xythos/WebSphere/DB2	2
VIOS	1

Memory

Different SAS components require different memory. In general, it is advised to have a minimum of 4 GB per core. The upside could be 8 GB per core. However, quite a memory usage variation is observed among different SAS components in the multi-tiered deployment.

SAS OLAP Server is very processing intensive but has a very low memory requirement. It uses six cores but needs a total of 8 GB memory only.

The **SAS Server** tier generates the most disk I/O. About half of the I/O volume is read access. In AIX, the read content is stored in the memory cache. Thus, this component was allocated a relatively large amount of memory in order to support a good cache hit ration and hence a good throughput. A Direct I/O (DIO) test on this tier shows the increased physical read I/O. This is likely due to cache misses for the data that usually resides in memory cache. In a DIO configuration, data is not stored in memory cache. An AIX file system cache enhances eBI results.

The **xythos/WebSphere/DB2** tier also has a good amount of memory. The WebSphere instance is configured in a cluster of four Java™ instances. Each Java instance consumes close to 2 GB for its work space.

Thus the memory allocations look like the following table:

LPAR	Memory
SAS Metadata Server	8
SAS OLAP Server	8
SAS Server	36
Xythos/WebSphere/DB2	24
VIOS	2

Summary

The recommendations covered in this paper have been proven effective in this eBI test. IBM PowerVM virtualization technology is effective in multi-tiered SAS deployment. This takes the advantage of the Hypervisor feature's ability to automatically move processing capacity in real time across different SAS components. General best AIX practices were applied. Also include were tuning disk and adapter parameters. In addition to performance tuning, regularly monitoring of the environment and collecting data for analysis is recommended as well. These are the suggested techniques to enhance SAS performance and create a winning environment for SAS workload.

About the authors

Joseph Pu's focus is in the area of AIX performance, tuning, and sizing. He has extensive experience in software development, from graphics to software simulation. He started his AIX development experience more than 20 years ago. Joe was graduated from the University of Texas at Austin with a degree in Computer Science.

Alfredo (Freddie) Mendoza is an IT specialist in the Systems and Technology Group ISV Organization at IBM. He has more than 19 years of experience in application architecture and systems management. His current work involves helping solution providers enable their applications onto IBM cross-platform systems. Mr. Mendoza has authored or co-authored several patent invention publications on thread management and software metering algorithms used in advanced Web services environments. He is the author of *Guide to Utility Computing Strategies and Technologies* and co-author of *The Linux to UNIX Porting Guide.*

Frank Bartucca is a certified IT specialist with IBM and has 25 years of experience with UNIX systems. He has been providing technical consulting to the SAS Institute on IBM System p servers since 2001.

Harry Seifert, an IBM certified IT specialist and Open Group Master certified IT specialist, has been architecting and deploying enterprise-class distributed computing solutions across the globe for nearly 20 years. Harry's technical training and experience is focused almost exclusively on business process enablement and performance optimization for clients worldwide. His current role as a Technical Solution Architect includes the development and enablement of reference architectures for SAS solutions that are optimized with IBM hardware and middleware.

Frank Battaglia is an IBM certified IT specialist who has worked with many IBM customers in the areas of end-to-end infrastructure capacity and performance planning. He is an expert in the areas of infrastructure performance monitoring, analytical and simulation capacity planning methods and tools. Frank currently works as a Technology Advocate and Enablement Manager, with IBM Systems and Technology Group, assisting key IBM Business Partners to optimized and exploit IBM's latest technologies.

Your comments and questions are valued and encouraged. You can contact the authors at: joepu@us.ibm.com

Resources

- IBM Systems on PartnerWorld®
  ibm.com/partnerworld/systems

- IBM Publications Center
  http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US

- AIX 5L V5.3 Performance Management Guide and AIX 5L V5.3 Performance Tools Guide and Reference
  http://publib.boulder.ibm.com/infocenter/pseries/index.jsp

- AIX 5L Performance Tools Handbook
  http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg246039.pdf

- IBM AIX Version 6.1 Differences Guide
  http://www.redbooks.ibm.com/abstracts/sg247559.html?Open

- Information about using IBM SSA Storage for SAS:
  http://www.sas.com/partners/directory/ibm/storagessa.pdf

- I/O subsystem Tuning Recommendations:
  DS4000 Best Practices and Performance Tuning Guide, SG24-6363-02
  http://www.redbooks.ibm.com/abstracts/sg246363.html?Open

- IBM TotalStorage DS6000 Series: Performance Monitoring and Tuning, SG24-7145-00
  http://www.redbooks.ibm.com/abstracts/sg247145.html?Open

- IBM TotalStorage DS8000 Series: Performance Monitoring and Tuning, SG24-7146-00
  http://www.redbooks.ibm.com/abstracts/sg247146.html?Open

- nmon for AIX & Linux Performance Monitoring
  http://www.ibm.com/developerworks/wikis/display/WikiPtype/nmon

- Graphical LPAR Monitor for System p5 Servers

  http://www.alphaworks.ibm.com/tech/lparmon

Trademarks and special notices

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.