



Configuring the IBM Enterprise Storage Server™ for Oracle® OLTP Applications

April 2003

IBM Advanced Technical Support
San Jose, CA
www.ibm.com

Contents

Abstract
1.0 Introduction
2.0 ESS configuration options
2.1 Disk size and speed
2.2 Subsystem capacity
3.0 AIX and RDBMS setup/tuning
4.0 ESS Setup
4.1 ESS array formatting options
4.2 RAID 5 vs. RAID 10
4.3 Deciding on a LUN size standard
4.4 Data isolation requirements
4.5 Creating LUNs
5.0 Server setup
5.1 SDD setup
5.2 Creating volume groups
5.3 Creating logical volumes
5.3.1 AIX logical volume 4k offset
5.4 Creating data files
6.0 The S.A.M.E. methodology
7.0 Conclusions
References
Acknowledgements

Abstract

This paper illustrates the “striping and spreading” technique used to achieve highly balanced I/O activity across the entire ESS subsystem, thereby optimizing overall I/O performance. This technique requires minimal knowledge of the specific application(s) to be deployed or their underlying I/O characteristics and it reduces or eliminates the need for manual data file placement or ongoing I/O performance tuning. The technique simultaneously employs two separate but complementary I/O balancing strategies. The first strategy is using the hardware RAID-5 and/or RAID-10 “striping” technology within the ESS subsystem. The second strategy is using the AIX Logical Volume Manager (LVM) to physically partition data files across multiple ESS arrays, thereby “spreading” I/O activity across all the arrays in the ESS subsystem.

1.0 Introduction

Since the introduction of the IBM TotalStorage™ Enterprise Storage Server (code named “Shark”) (“ESS”) 3 years ago, more than 10,000 units were shipped, representing more than 22 Petabytes of usable storage capacity. The widespread acceptance of the ESS in the marketplace is due in part to its high performance, advanced functions such as FlashCopy and PPRC (Peer-to-Peer-Remote-Copy) and very attractive Total Cost of Ownership (TCO).

However, as with all I/O subsystems, good planning and data layout can make the difference between having excellent I/O throughput and application performance, and having poor I/O throughput, high I/O response times and correspondingly poor application performance. It therefore is not surprising that first-time ESS customers ask for advice on how to optimally layout their ESS subsystem.

In many cases, I/O performance problems can be traced directly to “hot” files that cause a bottleneck on some critical component (e.g., a single physical disk). This can occur even when the overall I/O subsystem is fairly lightly loaded. When bottlenecks occur, Storage or Data Base Administrators may have to identify and manually relocate the high activity data files that were contributing to the bottleneck condition. This tends to be a very resource intensive and often frustrating task. As the workload content changes in concert with the ebb and flow of normal business cycles (e.g. hour by hour through the business day or day by day through the accounting period), bottlenecks may mysteriously appear and disappear or migrate over time from one datafile or device to another.

Generally, I/O (and therefore application) performance will be best when the I/O activity is evenly spread across the entire I/O subsystem. This paper offers a straightforward technique

that can you achieve highly balanced I/O activity across the entire ESS subsystem – thereby optimizing overall I/O performance. This technique requires minimal knowledge of the specific application(s) to be deployed or their underlying I/O characteristics and reduces or eliminates the need for manual data file placement or ongoing I/O performance tuning. Conceptually, the technique simultaneously employs two separate but complementary I/O balancing strategies:

- The use of hardware RAID-5 and/or RAID-10 technology within the ESS subsystem. The fine granularity “striping” inherent to the RAID-5 and RAID-10 algorithms supports highly balanced I/O activity across all of the physical disks in the RAID array.
- The use of AIX Logical Volume Manager (LVM) to physically partition data files across multiple ESS arrays, thereby “spreading” I/O activity across all the arrays in the ESS subsystem.

This “striping and spreading” strategy has been used in a number of large customer on-line transaction processing (OLTP) application implementations with excellent results.

While this paper’s focus is on Enterprise Application Solutions (EAS) applications, this technique may be used successfully with many other types of applications. Although, no single technique works best for all possible workload situations, the one described here should perform satisfactorily for most customer workloads.

This paper focuses primarily on the AIX/pSeries server platform. However, this technique could probably be adapted to other platforms that support a functionally rich Logical Volume Manager (LVM). While Oracle RDBMS (Relational Data Base Management System) is occasionally referred to, these general techniques are also applicable to DB2 UDB, Informix or other RDBMS environments.

2.0 ESS configuration options

When configuring/ordering an ESS subsystem, there are a number of choices or options that can affect the potential performance or I/O capacity of the subsystem. It is recommended that you work with your IBM TotalStorage sales rep or IBM Business Partner to explore what specific ESS configurations may help address your performance requirements. This section provides a high level overview of some of those options, but should be considered as an introduction to the issues, rather than specific advice for a particular environment.

2.1 Disk size and speed

Storage vendors and customers are jointly beating an unmistakable path to higher capacity disks. The reasons are simple: lower cost, less floor space, and easier management.

Each generation of disks typically provides doubles the capacity of prior generations. These larger capacity disks can mean more data, more users, and more I/O operations per second and potentially greater queuing delays. Despite all of these consequences, the historical trend continues. It all works because:

- Disks keep getting faster. Higher rotation speeds and increased density usually improve performance of each generation approximately 30%.
- Disk subsystems keep getting faster. Larger caches, higher bus speeds, striping, and better attachment technology all contribute.
- Users change behavior. Less expensive storage supports more types of application data, much of which has never been “online”, and much of which has lower access requirements.

Currently, there are five available ESS disk options; 18.2, 36.4, 72.8 or 145.6 GB capacity 10k rpm drives, and 18.2 or 36.4 GB capacity 15k rpm drives.

Performance of the 10k rpm 72.8 GB drives is acceptable for the majority of customer workloads and they offer the an attractive choice for most customers, due to their ability to deliver large storage capacity in a small footprint. The 15k rpm 36.4 GB drives offer higher performance and are more suitable for applications with relatively high access density (I/Os per second per gigabyte) and/or critical I/O response time requirements.

Adoption of the 72.8 GB drives has been fairly rapid and they account for the majority of newly installed capacity, although more 36.4 GB drive subsystems (with smaller total capacity) are sold. 18.2 GB drives account for a fairly small percentage of new subsystems and capacity.

The largest capacity 145.6 GB drives might be suitable in the following circumstances:

- For less demanding database applications with a low I/O access density requirement (< .7 I/Os per second per gigabyte).
- For applications with very heavy sequential content, such as decision support applications. Sequential workloads tend to stress other components of the disk subsystem (e.g., fibre interfaces, internal data paths, etc.) in comparison to the effects of disk contention.
- Combined with RAID-10. This may provide a reasonable cost compromise in circumstances in which RAID-10 is the customer’s preferred choice.

2.2 Subsystem capacity

Disk subsystem capacities have historically been increasing, for most of the same reasons as the disks themselves. For most usage scenarios for today's applications, the "sweet spot" for the ESS Model 800 is around 6.7 TB of usable capacity (128 72.8 GB disks in a RAID-5 configuration) per subsystem. A 6.7 TB configuration should provide acceptable performance for applications with up to twice the industry average access density (about .6 I/Os per second per GB). A 6.7 TB ESS Model 800 is therefore a reasonable "starting point" for most customers looking for the best overall balance between performance, capacity and price/performance. Not surprisingly, the average shipped capacity per ESS 72.8 GB subsystem exceeds 5 TB.

Throughout this paper, we will be developing an ESS implementation scenario to demonstrate the "striping and spreading" technique. Information regarding this scenario will appear throughout the paper, in Courier font like in this paragraph.

For our example, let's assume that we have ordered a basic "starting point" subsystem - an ESS 800 with 128 10k rpm 72.8 GB disks.

3.0 AIX and RDBMS setup/tuning

There are a number of AIX tuning parameters that can affect I/O activity and performance, such as those having to do with asynchronous I/O and Journalled File System (JFS) cache. Database specific parameters, such as the size of the database buffer cache, log file cache and database block size can also affect I/O performance.

Refer to the appropriate database documentation for recommended AIX and database parameter settings. For Oracle database, a good starting point is the Administrator's Reference document. (See Chapter 2, of "Oracle8i Administrator's Reference", or Appendix A of "Oracle9i Administrator's Reference".)

4.0 ESS setup

4.1 ESS array formatting options

ESS storage is typically organized into sets of 8 disks. Disks are installed in "eight packs", which are Field Replaceable Units (FRUs) containing 8 Disk Drive Modules (DDMs). RAID arrays are also organized into sets of 8 disks, though not mapping directly to 8-packs, and may be configured in one of four possible ways (depending on the ESS model and physical location of the disk eight pack within the ESS):

- RAID-5 (all ESS models)
 - 6+P+S (7 disks including parity with 1 spare disk)

- 7+P (8 disks including parity)
- RAID-10 (ESS 800 only)
 - 3+3 (3 primary disks, 3 mirror disks and 2 spare disks)
 - 4+4 (4 primary disks and 4 mirror disks)

In the ESS Model 800, the customer has control over the array type (RAID-5, RAID-10) and array types can be intermixed. In the older E and F models, only RAID-5 is supported. An ESS requires at least two spare DDMs per internal SSA loop. The number of disks used in a particular array is dependent on the array order within a loop. For RAID-5, the first two arrays defined on a loop must be 6+P configurations. For RAID-10, the first array defined on a loop must be a 3+3 configuration. RAID-5 and RAID-10 arrays may also be intermixed on the same loop. If the first array defined on a loop is RAID-5 and the second array defined on the loop is RAID-10, there will be 3 spares on the loop.

4.2 RAID-5 vs. RAID-10

Historically, database vendors and consultants have tended to warn against using RAID-5 technology for applications having a lot of update activity. Several factors have contributed to this position, which include:

- The classic RAID-5 “write penalty”, which refers to the fact that up to 4 internal I/Os are required for each external I/O request:
 1. read old data block
 2. read old parity block
 3. write new data block
 4. write new parity block
- Poorly-optimized implementations of RAID-5 in open systems environments. Some of these implementation lacked write cache, failed to stripe the data across disks, or were sometimes implemented in software.

However, in most cases, this “write penalty” does not affect application performance when using an ESS RAID-5 configuration. ESS normally has 100% cache write hits. This means that ESS returns a “write complete” status to the server as soon as the data block has been written to cache and to Non Volatile Storage (NVS). Therefore, as long as the backend (physical) disk configuration can support the sustained I/O rate required by the application, write I/O performance is not determined by the RAID-5 or RAID-10 choice.

Another “classic” recommendation is to never place RDBMS log files on RAID-5 devices because the RAID-5 design “cannot support high sequential write activity.” The ESS (as in all other IBM hardware RAID-5 capable storage offerings) uses a “full stripe” parity generation algorithm that eliminates the need to read existing data or parity in order to generate

the new parity information. Whereas a “classic” RAID-5 implementation may require 4 physical I/Os per logical I/O write request, the ESS “full stripe” RAID-5 write requires at most 1.17 physical I/Os per logical I/O write request (1 parity write for every 6 data writes). For high volume sequential write applications, this is an advantage for RAID-5 over RAID-10 – which may require 2 physical writes (1 to each copy) per logical write request.

As the following table shows, RAID-5 provides for a substantially higher effective capacity than RAID-10 - given an equivalent number of physical disks.

Effective Eight Pack Capacity in Gigabytes (10⁹)

	Disk Size (GB)			
	18.2	36.4	72.8	% Effectiveness
Eight-pack capacity	145.60	291.20	582.40	
RAID-5 (6+P)	105.20	210.45	420.92	72%
RAID-5 (7+P)	122.74	245.53	491.08	84%
RAID-10 (3+3)	52.50	105.12	210.39	36%
RAID-10 (4+4)	70.00	140.16	280.52	48%

Note: The ESS E and F models also support JBOD (Just a Bunch of Disks) configurations. JBOD support is not available on the ESS 800. JBOD does not provide any data protection against disk failures or support the automated I/O balancing that is inherent to RAID-5 and RAID-10 arrays. For these reasons, we DO NOT recommend the use of JBOD.

The following table summarizes the major RAID-5 vs. RAID-10 attributes.

RAID-5 vs. RAID-10 Comparison

	RAID-5	RAID-10
Sequential Read	Excellent	Excellent
Sequential Write	Excellent	Good
Random Read	Excellent	Excellent
Random Write	Fair	Excellent
\$ per MB	Excellent	Fair

For most customer workloads, RAID-5 provides comparable performance to RAID-10, with considerably lower cost per Megabyte. This makes RAID-5 the default choice for most applications. RAID-10 should be considered for workloads with a high percentage of random write activity and high I/O access densities. There is no “one size fits all” answer, but a basic rule-of-thumb would be to consider using RAID-10 if the random write content exceeds 25% and the peak sustained I/O rate is expected to exceed 50% of the array capacity. When in doubt, an IBM Storage Representative or IBM Business Partner can help model your workload and help identify a choice appropriate to your workload.

We will be using RAID-5 arrays. With a total of 16 disk eight packs (128 disks) on 8 internal SSA loops, all RAID-5 arrays will be of the 6+P type in order to satisfy the

requirement of having two spare drives per loop

4.3 Deciding on a LUN size standard

With ESS, the minimum configurable Logical Unit Number (LUN) size is 0.1 GB and the maximum is the total effective capacity of the RAID array that the LUN is defined on. In most cases, the choice of LUN size has minimal effect on performance. However, in an effort to simplify Storage Administration tasks, customers may wish to adopt a LUN size standard. This allows LUNs to be allocated and subsequently de-allocated and re-allocated in an orderly fashion, without wasting space. A consistent LUN size is also a key component of the “striping and spreading” technique.

Since the usable capacity of any ESS rank is some multiple (6, 7, 3 or 4) of the disk size, using the physical disk size (roughly) as the standard LUN size allows for efficient allocation of the available ESS disk capacity, even when multiple array configurations are used. This would equate to a LUN size of 17.5 GB (for 18.1 GB drives), 35.0 GB (for 36.2 GB drives), 70.1 GB (for 72.8 GB drives) or 140.2 GB (for 145.6 GB drives).

When there is a mix of physical disk sizes in your environment, consider basing LUN size on the size of the smallest disk. Or, for environments where ESS storage is shared across a large number of relatively small servers and smaller allocation units are desirable, consider some fraction of the disk size – such as 8.7, 17.5, or 35.0 GB (when using 72.8 GB drives). It is also possible to define more than one LUN size standard for an enterprise (e.g. 35.0 GB for large environments and 4.3 GB for small environments). Having multiple standard LUN sizes somewhat increases the complexity of the storage management task, but may provide for somewhat more efficient storage allocation if properly managed.

As we will further discuss in the “Create Volume Groups” section, the AIX Logical Volume Manager (LVM) has limits on Physical Partition size and the number of Physical Partitions per Physical Volume. Therefore, from an AIX LVM point of view, the most “optimum” LUN sizes are determined by the formula $2n \times 1016 \times 1 \text{ MB}$ (where $n = 0$ through 7). Since ESS LUN sizes need to be in .1 GB increments, this would equate to possible LUN sizes of 0.9, 1.9, 3.9, 17.9, 15.8, 31.7, 63.5, or 127 GB. These are somewhat different from the LUN sizes suggested earlier. The AIX “optimum” LUN sizes may result in slightly better performance (through smaller Physical Partition sizes) and allow for Volume Groups with slightly larger capacity. However, since the ESS Physical Disk sizes are not evenly divisible by the optimum AIX LUN sizes, their use will typically reduce the effective ESS storage capacity by 10-15%. (For example, for an 6+P array of 36.2 GB drives, it is possible to get six 35.0 GB LUNs for a total capacity of 210

GB, or six 31.7 GB LUNS for a total capacity of 190.2 GB – a 9.4% difference.)

We will use physical drive size as the “default” LUN size standard - 70 GB usable after sector formatting overhead. This allows for 6 LUNs per RAID-5 array. Since our ESS 800 subsystem has 16 arrays, we will a total of 96 LUNs. 96 LUNs of 70 GB each yields 6.72 TB of usable capacity.

4.4 Data isolation requirements

In the past Oracle has recommended that several categories of RDBMS files be isolated from each other (e.g. separate redo logs from data, indexes from tables, isolate rollback segments etc.). However, isolating each class of data file on its own set of arrays can result in sub-optimization of the overall ESS subsystem performance. In general, the best overall performance can be achieved if storage is managed with the minimal number of physical data groupings or pools.

One non-performance related consideration is whether or not to isolate recovery related data (e.g. active redo logs, archive logs, database backups etc.) from the primary table data. Both RAID-5 and RAID-10 architecture helps protect against single disk failures within an array. Although exceedingly unlikely, double disk failures do occasionally occur. In this event, some or all of the data on that array could be lost. If protection against double disk failures is critical, all recovery related data for a given database instance should be placed on separate ESS arrays (on separate ESS subsystems if practical) than the data being protected/backed up. Since indexes can be rebuilt from the table data, the only reason for separating index data from recovery related data may be for recovery performance.

In order to provide protection against double disk failures, we will isolate all recovery related data on a separate set of ESS arrays than where the table data is stored.

4.5 Creating LUNs

Use the ESS Specialist to create LUNs of the “standard” size determined above. LUNs are typically assigned to the host (or hosts in a RAC or OPS environment) that will use them at creation time. Subject to any data isolation requirements (see above section), LUN assignments should be done in such a way that at any given time, the assigned LUNs are evenly distributed across all of the disk groups (ranks) in the ESS – Assuming a 16 rank (128 disk) ESS subsystem, after 16 LUNs have been assigned, there should be one assigned LUN per array, after 32 LUNs assigned, there should be two assigned LUNs per array etc. Within the “evenly distributed” constraint defined above, LUNs should be assigned in a quasi-random fashion. A strict round-robin assignment can

potentially result in undesirable “convoy” affects for certain types of sequential workloads.

If multiple servers are attached (either direct attach or via a SAN) to the same ESS, the LUNs assigned to any particular server should be even distributed across the ESS arrays.

We will initially allocate LUNs for two separate Oracle instances, DB01 on Server A and DB02 on Server B. DB01 requires 1,700 GB for table/index data and 100 GB for recovery related data and DB02 requires 1,350 GB for table/index data and 50 GB for recovery related data. Therefore, we will need 27 LUNs ($1,700/70 = 25 + 100/70 = 2$) for DB01 and 21 LUNs ($1,350/70 = 20 + 50/70 = 1$) for DB02. The table below is a logical view of the ESS layout, showing the initial LUN assignment within the ESS. Remember, in our ESS configuration, there are 16 arrays, with six (6) 70 GB LUNs per array. The array designations (down the left side) are in Device Adapter Pair / Cluster / Loop notation. For example, 11B = Device Adapter Pair 1, Cluster 1, Loop B. Volume (LUN). The relative Volume (LUN) number within a given array is shown along the top.

Note that in order to isolate base data from recovery related data, Volume 003 on Device Adapter Pair 3, Cluster 1, Loop B cannot be used for the DB02-Data20 LUN since the DB02-Recov1 LUN is already allocated on that Device Adapter Pair/Cluster/Loop.

	001	002	003	004	005	006
42B	DB01-Data12	DB01-Data20	DB02-Data10			
42A	DB01-Data3	DB01-Data23	DB02-Data14			
32B	DB01-Data14	DB01-Data18	DB02-Data6			
32A	DB01-Recov2	DB02-Data4	DB02-Data17			
22B	DB01-Data9	DB01-Data25	DB02-Data19			
22A	DB01-Data4	DB01-Data15	DB02-Data7			
12B	DB01-Data1	DB02-Data1	DB02-Data15			
12A	DB01-Data6	DB01-Data21	DB02-Data12			
41B	DB01-Data13	DB02-Data3	DB02-Data18			
41A	DB01-Data2	DB01-Data19	DB02-Data9			
31B	DB01-Data5	DB02-Recov1				
31A	DB01-Data7	DB01-Data24	DB02-Data11			
21B	DB01-Data11	DB01-Data16	DB02-Data5	DB02-Data20		
21A	DB01-Recov1	DB02-Data2	DB02-Data16			
11B	DB01-Data8	DB01-Data22	DB02-Data13			
11A	DB01-Data10	DB01-Data17	DB02-Data8			

	Assigned to Server A (DB01)
	Assigned to Server B (DB02)
	Not Assigned

5.0 Server setup

The “striping” affects of RAID-5 and RAID-10 provide effective I/O balancing within individual arrays in the ESS and can potentially offer better performance with less manual intervention than alternative non-striped I/O subsystems. However, poor data placement can still result in I/O activity that is not well balanced across all of the arrays in the ESS. If the I/O demand against a single array exceeds the I/O capacity of that array, poor overall performance can result, even when the other arrays in the ESS are underutilized.

The objective of the “spreading” strategy described in this section is to evenly balance I/O activity across all of the arrays in the ESS subsystem, without requiring detailed knowledge of application workload or specific data file i/o characteristics, and with minimal ongoing performance “tuning” requirements.

5.1 SDD setup

The Subsystem Device Driver (SDD) can be used to automatically load balance I/O activity across the available host I/O adapters and/or provide alternate paths if one (or more) of the available adapters fails. In most cases, the “spreading” strategy to be described here should provide for fairly well balanced activity across the host adapters with or without the use of SDD. However, SDD is still recommended for High Availability (HA) environments in order to avoid outages due to host adapter failures.

When multiple paths are defined for a single ESS LUN, each path will show up as a single AIX hdisk. The smitty “Define and Configure All Data Path Devices” function can be used to

create vpath definitions. “vpath” is a logical grouping of hdisks, where each vpath represents all of the possible paths (hdisks) to a particular ESS LUN.

In our example, we will be using SDD, so vpath names will be assigned to each grouping of hdisks corresponding to a particular ESS LUN. When vpaths are assigned a group of newly assigned ESS LUNs, they are normally created in ESS Device Adapter Pair / Cluster / Loop / Volume order (as shown in the table on the next page). However, you should be sure to verify the vpath name assignments are as you expect.

	001	002	003	004	005	006
42B	SvrA vpath15 DB01-Data12	SvrA vpath26 DB01-Data20	SvrB vpath20 DB02-Data10			
42A	SvrA vpath14 DB01-Data3	SvrA vpath25 DB01-Data23	SvrB vpath19 DB02-Data14			
32B	SvrA vpath11 DB01-Data14	SvrA vpath23 DB01-Data18	SvrB vpath16 DB02-Data6			
32A	SvrA vpath10 DB01-Recov2	SvrB vpath3 DB02-Data4	SvrB vpath15 DB02-Data17			
22B	SvrA vpath7 DB01-Data9	SvrA vpath21 DB01-Data25	SvrB vpath13 DB02-Data19			
22A	SvrA vpath6 DB01-Data4	SvrA vpath20 DB01-Data15	SvrB vpath12 DB02-Data7			
12B	SvrA vpath3 DB01-Data1	SvrB vpath DB02-Data1	SvrB vpath8 DB02-Data15			
12A	SvrA vpath2 DB01-Data6	SvrA vpath18 DB01-Data21	SvrB vpath7 DB02-Data12			
41B	SvrA vpath13 DB01-Data13	SvrB vpath4 DB02-Data3	SvrB vpath18 DB02-Data18			
41A	SvrA vpath12 DB01-Data2	SvrA vpath24 DB01-Data19	SvrB vpath17 DB02-Data9			
31B	SvrA vpath9 DB01-Data5	SvrB vpath2 DB02-Recov1				
31A	SvrA vpat8 DB01-Data7	SvrA vpath22 DB01-Data24	SvrB vpath14 DB02-Data11			
21B	SvrA vpath5 DB01-Data11	SvrA vpath19 DB01-Data16	SvrB vpath10 DB02-Data5	SvrB vpat11 DB02-Data20		
21A	SvrA vpath4 DB01-Recov1	SvrB vpath1 DB02-Data2	SvrB vpath9 DB02-Data16			
11B	SvrA vpath1 DB01-Data8	SvrA vpath17 DB01-Data22	SvrB vpath6 DB02-Data13			
11A	SvrA vpath0 DB01-Data10	SvrA vpath16 DB01-Data17	SvrB vpath5 DB02-Data8			

	Assigned to Server A (DB01)
	Assigned to Server B (DB02)
	Not Assigned

5.2 Creating volume groups

Once vpath names (or hdisk names if SDD is not being used) have been assigned, one or more Volume Groups can be created using those vpaths (or hdisks). This can be done using smitty or the “mkvg” or “mkvg4vp” AIX command.

If the recovery related data is to be isolated from the regular database data, one (or more) Volume Group(s) should be

created for the recovery related data and one (or more) Volume Group(s) should be created for the database data. Generally, using a small number of Volume Groups will provide more opportunity for overall ESS subsystem performance than using a large number of Volume Groups. However, AIX Logical Volume Manager (LVM) restrictions on the number of physical volumes in a volume group, Physical Partition size, etc. may more or less require the use of multiple Volume Groups for large database environments. If multiple Volume Groups are required, try to place several vpaths (or hdisks) in each Volume Group (preferably one vpath/hdisk from every ESS array or rank).

By default, Volume Groups can accommodate up to 255 Logical Volumes and 32 Physical Volumes. If the “-B” flag is used on the `mkvg` or `mkvg4vp` command, the resulting Volume Group will support up to 512 Logical Volumes and 128 Physical Volumes.

When creating the Logical Volumes (see next section), we will be “spreading” each Logical Volume across as many ESS LUNs (vpaths or hdisks) as possible. In order to do that optimally, we want to set the Physical Partition (PP) size for the Volume Group as small as possible (using the `mkvg` or `mkvg4vp` “-s” option), given the size and number of Physical Volumes (vpaths or hdisks) in the Volume Group. Spreading, works very well for most OLTP workloads. However, for single stream sequential I/O (e.g. a single connected user doing a tablespace scan) performance will be limited to the capacity of a single ESS array. If higher single stream sequential throughput is required, this can be accomplished through the use of LVM small grain striping (with a 128k strip size) rather than Physical Partition spreading. However, caution should be used when doing LVM striping over ESS RAID-5 or RAID-10 striping, because the LVM striping can potentially defeat the ESS sequential detect/prefetch cache management algorithms.

By default, there is a limitation of 1016 Physical Partitions per Physical Volume. Therefore, in order to fully utilize the space on a given Physical Volume, the Physical Partition size must be \geq Physical Volume size / 1016. However, if a Volume Group is going to contain less than the maximum number of Physical Volumes, it may be possible to adjust the “factor value” (`mkvg` or `mkvg4vp` “-t” option) to allow for more than 1016 Physical Partitions per Physical Volume. The following table shows the optimal “-s” and “-t” settings for some possible Physical Volume (LUN) sizes and number of volumes in the volume group. It is always a good idea to leave room in every volume group for at least one additional Physical Volume to be added at a later time.

Physical partition size (GB) given for LUN size							
# PVs	Factor	4.3	8.7	17.5	35.0	70.0	140.0
1-2	64	1	1	1	1	2	4
3-4	32	1	1	1	2	4	8
5-8	16	1	1	2	4	8	16
9-16	8	1	2	4	8	16	32
17-32	4	2	4	8	16	32	64
33-64	2	4	8	16	32	64	128
65-128	1	8	16	32	64	128	256

For DB01, we will create one Volume Group for the recovery related data and one Volume Group for the remaining database data. The recovery related Volume Group will contain 2 Physical Volumes. Allowing for future growth, we will use a "factor" of 32 and a Physical Partition size of 4 MB.

```
mkvg4vp -B -t 32 -s 4 -y DB01_RECOV_VG1
vpath4 vpath10
```

The other Volume Group for DB01 will contain 25 Volumes. Allowing for future growth, we will use a "factor" of 4 and a Physical Partition size of 32 MB. Note that if we expected to have a large number of small heavily accessed tables, we may want to create multiple Volume Groups so that the Physical Partition size could be reduced to allow for more of the small tables to be spread across multiple volumes in the volume group.

```
mkvg4vp -B -t 4 -s 32 -y DB01_DATA_VG1 vpath3
vpath12 vpath14 vpath6 vpath9 vpath2 vpath8
vpath1 vpath7 vpath0 vpath5 vpath15 vpath13
vpath11 vpath20 vpath19 vpath16 vpath23
vpath24 vpath26 vpath18 vpath17 vpath25
vpath22 vpath21
```

We won't show the details here, but we would create similar Volume Groups for the DB02 database.

5.3 Creating logical volumes

Now that the Volume Group(s) have been defined, the Logical Volumes can be created. We will "spread" the data for each Logical Volume across the maximum number of the Physical Volumes within the Volume Group by using the mkiv "-e x" option from the command line or through smitty. Spreading greatly reduced the risk of I/O hotspots caused by one or more heavily accessed datafiles.

When Raw Devices are going to be used for Oracle Database files, be sure to adhere to existing Oracle limitations on the maximum data file (raw device size). Also note that currently an AIX Logical Volumes may not have more than 32,512 Physical Partitions.

For DB01, let's initially create a total of 6 Logical Volumes, 3 for data and 3 for index. The Logical Volumes for data will each be 4 GB in size (128 32 MB partitions) and the Logical Volumes for indexes will each be 1 GB in size (32 32 MB partitions). Note that Logical Volumes for both index and table (row) data can be contained within the same Volume Group.

```
mklv -e x -y DB01_DATA_01 DB01_DATA_VG1 128
mklv -e x -y DB01_DATA_02 DB01_DATA_VG1 128
mklv -e x -y DB01_DATA_03 DB01_DATA_VG1 128
mklv -e x -y DB01_INDX_01 DB01_DATA_VG1 32
mklv -e x -y DB01_INDX_02 DB01_DATA_VG1 32
mklv -e x -y DB01_INDX_03 DB01_DATA_VG1 32
```

For simplicity, we have not shown the creation of Logical Volumes for such things as Oracle executable libraries, rollback segments, redo logs etc. These would be required for a complete Oracle implementation.

5.3.1 AIX logical volume 4K offset

By default, the first 4k of each AIX Logical Volume is reserved for the Logical Volume Control Block (LVCB). This means that the first Oracle data block begins at a 4k offset into the Logical Volume. When fine granularity striping is used (either within AIX LVM or within ESS RAID-5 or RAID-10 arrays), this can result in a slight I/O performance degradation when an Oracle DB Block Size is greater than 4k is used. (An 8k DB Block Size is typical for OLTP applications and a 16k DB Block size is typical for Data Warehouse applications.) This is because every few DB blocks are physically split across device boundaries – with the first part of the DB block residing on one physical disk and the remainder of the DB block residing on another physical disk. This can result in two physical I/Os being required to read or write a single DB block. The larger the DB Block size used, the higher the percentage of split blocks and the greater the potential for I/O performance degradation.

When running Oracle9i Release 2 (or later), it is possible to eliminate the 4k offset. This is recommended for new Oracle implementations or for existing applications with extremely high I/O performance requirements. Currently, the capability to do this is delivered in two parts:

1. IBM AIX e-fix (APAR IY36656 for AIX 5.1 or APAR IY38578 for AIX 4.3) and
2. Oracle patch (bug 2620053).

The functionality will be included in future release levels of AIX and in Oracle 9.2.0.3 or later. Once the prerequisite software has been installed, do the following to take advantage of the zero offset feature:

- Create a “big” Volume Group using the `mkvg -B` flag.
- Create one or more Logical Volumes in that Volume group using the `mklv -T O` flag. The “-T O” option indicates to Oracle that it can safely use a 0 offset for this Logical Volume.

In order to eliminate the 4k offset for an existing Oracle database, new Logical Volumes must be created and the existing data must be migrated to the new Logical Volumes using normal migration procedures.

5.4 Creating data files

If you are will be using Raw Devices, this step is omitted. However, if you are using the Journalled File System (JFS) to manage your data files, you would create one or more data files (via the Oracle CREATE/ALTER TABLESPACE “ADD DATAFILE” clause) per AIX Logical Volume. Large, heavily accessed data files are sometimes subject to inode contention. We generally recommend that individual RDBMS related data files to be no more than 2 GB in size in order to keep inode contention at a minimum.

Oracle reads the header block of all database data files at database startup to determine whether or not database recovery is required. Therefore in general, the larger the number of data files, the longer the startup time. And, since all open data files need to be closed during a normal database shutdown, shutdown is also somewhat proportional to the number of data files. For High Availability (HA) environments that employ single instance HACMP failover (as opposed to doing an Oracle9i Real Application Clusters (RAC) coordinated failover), reducing the number of data files may reduce time required for failover.

Oracle provides I/O performance information at a data file (or raw device) level. Therefore, having a large number of data files may provide more accurate information regarding the I/O characteristics of particular database objects (e.g. tables, indexes). For some applications, this may be an argument for having more rather than fewer data files.

In our example, we will be using raw devices, so individual data files will not be created.

6.0 The S.A.M.E. Methodology

S.A.M.E. (Stripe and Mirror Everything) is an Oracle-recommended methodology based on making extensive use of striping across large sets of disks. The key objectives of S.A.M.E. are to provide protection against single disk failures, mirroring being one way to accomplish that and to reduce I/O hotspots by balancing I/O activity across multiple physical disks.

The “striping and spreading” technique outlined in this paper is entirely consistent with S.A.M.E.:

- ESS RAID-10 and RAID-5 arrays both provide protection against single disk failures.
- ESS RAID-10 and RAID-5 both balance I/O activity across all the physical disks in a RAID-10 or RAID-5 array.

However, ESS and the “striping and spreading” technique offers a number of additional benefits:

- The choice of RAID-10 (mirroring) or RAID-5. RAID-5 provides comparable performance to RAID-10 for most customer workloads with better price/performance.
- Good, automatic I/O balancing across all arrays within the ESS subsystem as well as within a single ESS RAID-5 or RAID-10 array.

7.0 Conclusions

I/O subsystem performance can have a significant effect on overall application performance. Manual I/O “hotspot” management can be a very resource intensive activity.

The “striping and spreading” technique described in this paper provides for balanced I/O activity across the ESS subsystem without requiring detailed knowledge of the application data and I/O characteristics and without significant ongoing “hotspot” management.

This “striping and spreading” strategy has been used in a number of large customer application implementations with excellent results.

References

IBM TotalStorage Enterprise Storage Server Introduction and Planning Guide, IBM Publication Number GC26-7444-00

IBM TotalStorage Enterprise Storage Server Model 800 Performance whitepaper by Bruce McNutt, July 2002

AIX 5L Version 5.1 Commands Reference, IBM Publication SBOF-1877

Oracle8i Administrator's Reference – Release 3 (8.1.7) for AIX-Based Systems – Oracle Part Number A85348-01

Oracle9i Administrator's Reference – Release 2 (9.2.0.1.0) for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel and Sun Solaris – Oracle Part Number A97297-01

Acknowledgements

Author:

- Dale Martin, Senior Certified I/T Specialist
Enterprise Application Solutions, IBM Advanced Technical Support – Americas

Contributors:

- John Aschoff, Database and Performance, IBM Storage Systems Group
- Dan Braden, pSeries Support, IBM Advanced Technical Support - Americas
- Martin Carangelo, Enterprise Application Solutions, IBM Advanced Technical Support – Americas
- Ralf Schmidt-Dannert, Enterprise Application Solutions, IBM Advanced Technical Support - Americas
- Jaymin Yon, Enterprise Application Solutions, IBM Advanced Technical Support - Americas

© Copyright IBM Corporation 2002
IBM Storage Systems Group
5600 Cottle Road
San Jose, CA 95136
Produced in the United States

January 2003

All Rights Reserved

No part of this document may be reproduced or transmitted in any form without written permission from IBM Corporation.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This information could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or programs(s) at any time without notice.

The performance data contained herein was obtained in a controlled, isolated environment. Actual results that may be obtained in other operating environments may vary significantly. While IBM has reviewed each item for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere.

Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead. It is the user's responsibility to evaluate and verify the operation of any non-IBM product, program or service.

The information provided in this document is distributed "AS IS" without any warranty, either express or implied. IBM EXPRESSLY DISCLAIMS any warranties of merchantability, fitness for a particular purpose OR INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. IBM is not responsible for the performance or interoperability of any non-IBM products discussed herein.

Information concerning non-IBM products was obtained from the suppliers of those products, their published

announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: AIX, IBM, AS/400, DFSMSdftp, DFSMSdss, DFSMSHsm, DFSMSrmm, DFSORT, Enterprise Storage Server, ESCON, FICON, FlashCopy, iSeries, Magstar, MVS/ESA, Netfinity, OS/390, OS/400, pSeries, RS/6000, S/390, SANergy, Tivoli, TotalStorage, VM/ESA, VSE/ESA, xSeries, z/OS, z/VM and zSeries

Other company, product or service names may be trademarks or service marks of others.