

# Tuning Oracle Recovery Manager

*An Oracle White Paper  
December 2001*

# Tuning Oracle Recovery Manager

Introduction .....	3
Backup Performance Basics.....	4
Optimal Disk Storage Configuration.....	4
The Processing Power of the CPU.....	5
Oracle Backup Performance Basics.....	5
RMAN Channel Parallelism.....	5
Backup Multiplexing .....	5
Asynchronous and Synchronous I/O for Disk and Tape.....	6
Disk and Tape Buffers.....	6
Media Management Software.....	7
Performance Issues for Tape Drives.....	7
Native Transfer Rate.....	8
Tape Compression .....	8
Tape Streaming.....	9
Physical Tape Block Size.....	9
Improving Backup Speed.....	10
Testing Environment.....	10
What's New in Oracle9?.....	10
RMAN Tuning Steps.....	11
Determining the Rate of Disks and Backup Media .....	11
Determining the Number of Channels to Allocate .....	11
Adjusting the Size of Backup Disk Buffers in Oracle8 <i>i</i> .....	12
Adjusting the Level of Multiplexing .....	13
Disabling Backup Disk I/O Slaves for Asynchronous I/O in Oracle8 <i>i</i> .....	15
Enabling Backup Disk I/O Slaves on a Synchronous System.....	15
Enabling Backup Tape I/O Slaves When Backing Up to Tape.....	16
Increasing BLKSIZE When Backing Up to Tape.....	17
Setting LARGE_POOL_SIZE If Oracle Fails to Allocate Shared Memory.	17
Investigating the V\$BACKUP_ASYNC_IO View as a Last Resort.....	18
Conclusion.....	19

# Tuning Oracle Recovery Manager

## INTRODUCTION

Backup and recovery is one of the most important aspects of database administration. Whether companies operate a single database or multiple databases that store hundreds of gigabytes or even terabytes of data, they share a common factor: the need for a fast and reliable backup solution. Oracle's Recovery Manager (RMAN) utility provides this solution.

When evaluating RMAN as a backup and recovery solution, many users ask the following questions:

- What is the optimal RMAN backup setting?
- How is the performance of an RMAN backup affected by the alteration of backup parameters?

Technical papers describing computer system performance can be laden with complex graphs showing how this action varies with that setting. Seldom do the graphs shed much light on *how* to make a system faster. In contrast, this technical paper is a result of extensive in-house testing of Oracle8*i* and Oracle9*i* RMAN backup performance with the aim of answering the preceding questions. As a result, this paper provides comprehensive step-by-step instructions for tuning an RMAN backup.

The first part of the document, chapter "Backup Performance Basics", explains the basic components that influence RMAN backup performance. This section also explains which components can create a bottleneck and why.

The second section "Improving Backup Speed" gives step-by-step procedures for tuning an RMAN backup. Each step is based on real-world performance test results. The performance tests were conducted on a Sun Enterprise™ 6000 Server, with 10 GB of memory and 24 UltraSPARC™ processors running on 344 MHz.

## BACKUP PERFORMANCE BASICS

Backup performance is the result of a complex interaction among many variables, many of which you can control little or not at all. However, the major factors that impact a backup are summarized in the following sections:

- Optimal Disk Storage Configuration
- The Processing Power of the CPU
- Oracle Backup Performance Basics
- Media Management Software
- Performance Issues for Tape Drives

### Optimal Disk Storage Configuration

The storage configuration subsystem is one of the most important factors affecting performance of the system. The optimal storage configuration has been discussed in a number of technical papers. The most accurate discussion of Oracle disk configuration is *Optimal Storage Configuration Made Easy* by Juan Loaiza, which recommends the following:

- Stripe all files across all disks by using a 1 MB stripe size
- Mirror data for high availability rather than using RAID-5

We measured the sequential read performance of the preceding configuration by using an internally developed utility that reads data in a similar way to an Oracle backup. We also measured a nonstriped disk configuration. As indicated in table 1, the results of these tests show that reading striped disks is significantly more efficient than reading nonstriped disks: the backup is at least 10% faster while the CPU utilization is much smaller.

Table 1: Read Rate of Striped Disks Vs. Nonstriped Disks

Test No.	Utility	Number of Disks	Striped (Yes/No)	Stripe Width (Kb)	Read Block Size (Kb)	Speed cumulative (Mb/s)	CPU utilization (%)
1.	my_dd	8	Yes	1024 Kb	1024 Kb	62.6 Mb/s	0.20%
2.	my_dd	16	Yes	1024 Kb	1025 Kb	125.3 Mb/s	0.23%
3.	my_dd	20	Yes	1024 Kb	1024 Kb	167.0 Mb/s	0.31%
4.	my_dd	8	No	N/A	1024 Kb	59.2 Mb/s	2.20%
5.	my_dd	16	No	N/A	1024 Kb	117.9 Mb/s	3.40%
6.	my_dd	20	No	N/A	1024 Kb	146.4 Mb/s	4.30%

The striped disks can achieve ~13% better read rate than non-striped disks at smaller CPU utilization.

### **The Processing Power of the CPU**

The processing power of the server can affect backup performance. During the backup, RMAN performs the following operations, which can make extensive use of the processing power:

- Validation of datafile blocks
- Copying data from memory buffers to output I/O busses

Careful analysis of the current CPU technologies shows that the processor is rarely a backup bottleneck. For example, our testing showed that a single UltraSPARC™ 344 MHz processor can validate Oracle datafile blocks at a speed of about 120 MB/s. Additionally, almost all system architectures have a special chip set that assists the CPU with data movement so that the processor power is not consumed by moving input memory buffers to output I/O busses. One example of this type of a chip set is the UltraSPARC™ U2S interface (*UltraSPARC™ Port Architecture to SBus interface*).

Even though testing shows that the CPU is not a bottleneck in our testing, real systems typically run jobs and applications other than backups. Thus, prudence suggests running backups at night or at other times when users are not impacting the system — just to be on the safe side.

### **Oracle Backup Performance Basics**

The following Oracle Server and RMAN parameters influence backup performance:

- RMAN Channel Parallelism
- Backup Multiplexing
- Asynchronous and Synchronous I/O for Disk and Tape
- Disk and Tape Buffers

#### **RMAN Channel Parallelism**

An RMAN channel represents one stream of data to an output device. When RMAN allocates a channel, it establishes a connection to a target database instance by starting a server session on the instance. This Oracle server session performs the backup, restore, and recovery operations. So, the RMAN channel is an operating system process or thread that reads the data from disk and sends it to the output. The backup is done in parallel on all allocated channels. Because the channels act independently, the number of active channels defines the level of parallelism.

#### **Backup Multiplexing**

One channel can simultaneously read more than one file. The level of multiplexing is the number of files read *simultaneously* on a *single* channel and then written to the same backup piece. The degree of multiplexing depends on the `FILESPERSET`

parameter of the BACKUP command as well as the MAXOPENFILES parameter of the CONFIGURE CHANNEL or ALLOCATE CHANNEL commands.

The level of multiplexing is the lesser of:

- Value of parameter FILESPERSET
- Value of parameter MAXOPENFILES
- Number of files read by a channel

The number of files in each backup set is the lesser of FILESPERSET and the number of files read by each channel. The level of multiplexing is the lesser of MAXOPENFILES and the number of files in each backup set. For example, assume that you back up two datafiles with one channel. You set FILESPERSET to 3 and set MAXOPENFILES to 8. In this case, the number of files in each backup set is 2 (the lesser of FILESPERSET and the files read by each channel), and so the level of multiplexing is 2 (the lesser of MAXOPENFILES and the number of files in each backup set).

Assume a different case in which each channel reads fifteen datafiles, FILESPERSET=10, and MAXOPENFILES=8. You can calculate the level of multiplexing as follows:

$$\min(\min(15, 10), 8) = 8$$

The default values are 16 for MAXOPENFILES and 64 for FILESPERSET.

#### **Asynchronous and Synchronous I/O for Disk and Tape**

When Oracle reads or writes data to disk, the I/O can be either synchronous or asynchronous. When I/O is synchronous, a server process can perform only one task at a time. On the other hand, when I/O is asynchronous, a server process can begin an I/O and then perform other work while waiting for the I/O to complete. It can also begin multiple I/O operations before waiting for the first to complete. The ability to do asynchronous I/O always improves performance.

You can enable native asynchronous I/O with the initialization parameter DISK\_ASYNCH\_IO. On operating systems that do not support native asynchronous I/O, Oracle can simulate it by using disk I/O slave processes that are dedicated to performing I/O on behalf of another process. You can enable the disk I/O slave processes used by the ARCH, LGWR, and backup processes by configuring the DBWR\_IO\_SLAVES initialization parameter. You can control the backup disk I/O slaves separately with the \_BACKUP\_DISK\_IO\_SLAVES initialization parameter.

Even though tapes are sequential devices, writing to and reading from a tape can also be improved by using backup tape I/O slave processes. When you enable backup tape I/O slaves, Oracle will perform some disk I/O operations while waiting for tape I/O operation to finish. You can enable these slave processes with the BACKUP\_TAPE\_IO\_SLAVES initialization parameter.

By default, both tape and disk I/O slaves are disabled.

#### **Disk and Tape Buffers**

Oracle backups use two different types of buffers for reading and writing of data: disk buffers and tape buffers. Disk buffers are used for reading and writing to disks,

whereas tape buffers are used for tape I/O operations. The size of these buffers defines the amount of data transferred in a single I/O call.

In asynchronous I/O, the number of buffers defines the number of concurrent I/O operations, that is, the number of I/O operations that are initiated before waiting for the first operation to complete. For example, if four disk buffers are used for the backup, then Oracle issues four reads before waiting for the first read to complete. So, if the data is striped on four different disks, then each read will go to a different disk. Hence Oracle will read from all four disks in parallel. Of course, if the I/O is synchronous, then the number of buffers is irrelevant because a server process can perform only one I/O operation at a time.

In Oracle8i, the default for tape buffer size is 64 KB. The size of the disk buffers is equal to 64 times the database block size. In Oracle9i, the default tape buffer size is 256 KB. The sizes of disk buffers are determined dynamically.

### **Media Management Software**

In order to save data to and retrieve data from a sequential backup media such as tape, Oracle must be integrated with media management software. A media manager is a third-party software program that writes, reads, and manages sequential media used to back up and recover data. For example, Legato NetWorker®, Veritas NetBackup™, and HP OpenView OmniBack II are popular media management software products.

The core of the RMAN and media manager software integration is the media management library, which contains an implementation of the SBT API functions that read, write, and manage data. Oracle regards a tape device as a virtual device of type SBT.

Media management software plays an important role in performance tuning. However, the goal of this paper is not to elaborate and explain how to tune media management software. For the purposes of this paper, the most important issues are as follows:

- **Software compression and encryption.** Some media management products can compress data. Real-world experience shows that software compression *does not help* improve local backup performance. It helps *only if* the data is transferred over a slow network.
- **Transfer of data inside media management software.** In some cases the media management software can be misconfigured so that the transfer of data to a backup media is very slow. For example, the data is copied by means of network protocols such as TCP/IP even though the tape is locally attached.

### **Performance Issues for Tape Drives**

The goal of this paper is not to explain tape tuning. However, some discussion of this topic is necessary because the tape drive can be an important factor in performance. The following topics are important:

- Native Transfer Rate
- Tape Compression
- Tape Streaming
- Physical Tape Block Size

#### **Native Transfer Rate**

The tape native transfer rate is the speed of writing to a tape without compression. This speed represents the upper limit of the backup rate.

About twenty different tape technologies currently exist on the market. The native backup speed of these technologies ranges from 400 KB/s (DDS-1 tapes) to 15 MB/s (new LTO technology).

For example:

- The transfer rate of the DLT4000™ is 1.5 MB/s. It is more than doubled by the DLT7000™, which has a rate of 5 MB/sec. The DLT8000™ delivers a 6 MB/s native transfer rate.
- Sony AIT-2 has about a 6 MB/s native transfer rate.
- Ultrium LTO tape technology offers a 15 MB/s native transfer rate.

The upper limit of your backup performance should be the aggregate transfer rate of all of your tape drives. If your backup is already performing at that rate, and is not using an excessive amount of CPU, then RMAN performance tuning will not help.

#### **Tape Compression**

The level of **tape compression** is very important for backup performance. If the tape has good compression, then the sustained backup rate is faster. For example, if the compression ratio is 2:1 and native transfer rate of the tape drive is 6 MB/s, then the resulting backup speed is 12 MB/s. Different tape technologies use different compression algorithms. All claim a compression ratio of about 2:1.

The following examples indicate the compression ratios of common tape drives:

- Exabyte Mammoth uses the IDRC (IBM's Improved Data Recording Capability) algorithm with an average compression ratio of 1.8:1.
- DLT technology uses DLZ (Digital Lempel Ziv) algorithm with a compression ratio of 1.9:1.
- Sony AIT uses ALDC (Advanced Lossless Data Compression) with a compression ratio of 2.4:1.
- LTO standard uses OPD (Linear Tape-Open Data Compression) with a compression ratio of 2.3:1.

Note that the above results are just examples. The actual ratios fluctuate depending on data patterns.

### Tape Streaming

One of the most interesting issues for backup performance is **tape streaming**. Almost all tape drives currently on the market are fixed-speed, streaming tape drives. In other words, these drives can only write data at one speed. As a result, when they run out of data to write to tape, they must slow down and stop. For example, when the drive's buffer empties, the tape is moving so quickly that it actually overshoots and must rewind past the point where it stopped writing.

Different tape drives vary in how easily they can stop and restart. Some tape drives can stop in a second, while others require multiple seconds to stop and restart. For example, DLT technology moves tape at a very fast rate of 150 inches of tape per second, so they have difficulty stopping and starting the motors quickly. On the other hand, some tape drive technologies move the tape at a slower rate of one inch per second, so they can stop easily.

Testing showed that a problem with tape streaming occurs *very rarely*. New tape drives have large internal buffers and they continually monitor the data flow and automatically adjust both the reconnect and motion thresholds to match variations in the data transfer rate.

### Physical Tape Block Size

The **physical tape block size** can affect backup performance. The block size is the amount of data written by media management software to a tape in one write operation. The common rule is that a larger tape block size leads to a faster backup. Note that physical tape block size is not controlled by RMAN or the Oracle server, but by media management software.

Larger physical tape block size leads to a faster backup.  
The physical tape block size is controlled by media management software.

## **IMPROVING BACKUP SPEED**

This section describes the short procedure for tuning RMAN backup performance. The procedure is based on our RMAN performance tests. It is generic and should not depend on media managers or tape drives.

### **Testing Environment**

The tests were conducted on a Sun Enterprise™ 6000 Server, with 24 UltraSPARC™ processors running on 344 MHz. The system contained 10 GB of memory. For disk storage, we used Seagate 9.1GB ST19171FC Barracuda® 9FC Fibre Channel hard-disk drives, which were part of the Sun StorEdge™ A1000 disk array without Sun StorEdge™ FastWrite Cache. The disks had a minimum transfer speed of 1024 KB data buffers of about 10 MB/s. The disks were connected through a Fibre Channel interface with a maximum transfer rate of 106.3 MB/s. The actual maximum read speed of the disks was about 9 MB/s.

We tested backups that used channels of type DISK and sbt. For the sbt tests, we implemented a special sbt library to simulate a typical media manager backup to a single LTO tape drive. The library did everything that a typical media manager does: coping blocks to its internal memory, adding header, and so on. The output of our sbt interface was a disk volume, which was tuned to achieve the constant throughput of 22 MB/s — very close to actual LTO tape drive throughput. For DISK tests, we backed up to the same disk as used for output of our sbt library.

### **What's New in Oracle9i?**

The main Oracle9i performance improvement is in the algorithm used to read disks. Now, Oracle automatically sets buffer sizes and the number of disk buffers depending on the level of multiplexing. For example, consider the following disk configuration scenarios:

- If you stripe datafiles across a large number of disks, then set the level of multiplexing to 1. Because the datafiles are already striped, RMAN does not need to do additional multiplexing. When multiplexing is 1 and the datafiles are located on multiple disks, RMAN issues more concurrent reads in order to read the disks in parallel as much as possible. For example, suppose that the datafiles are striped across 16 disks. If the level multiplexing is 1 (for example, FILESPERSET=1), then RMAN allocates 16 disk buffers, each of size 1 MB. Hence, each channel has 16 concurrent reads, which means all 16 disks are read in parallel.
- If you do not stripe your disks, then set the level of multiplexing between 6 and 10 because RMAN must perform the multiplexing itself. In this case, RMAN allocates just 4 disk buffers because many concurrent reads are unnecessary if the file resides on just one disk.

Another Oracle9i performance improvement is that the Oracle server is able to make more intelligent decisions about when backup disk I/O slaves are needed. Finally, in Oracle9i the backup tape buffer size is increased from 64 KB to 256 KB.

## RMAN Tuning Steps

To tune an RMAN backup, follow the steps described in these sections:

- Determining the Rate of Disks and Backup Media
- Determining the Number of Channels to Allocate
- Adjusting the Size of Backup Disk Buffers in Oracle8i
- Adjusting the Level of Multiplexing
- Disabling Backup Disk I/O Slaves for Asynchronous I/O in Oracle8i
- Enabling Backup Disk I/O Slaves on a Synchronous System
- Enabling Backup Tape I/O Slaves When Backing Up to Tape
- Increasing BLKSIZE When Backing Up to Tape
- Setting LARGE\_POOL\_SIZE If Oracle Fails to Allocate Shared Memory
- Investigating the V\$BACKUP\_ASYNC\_IO View as a Last Resort

### Determining the Rate of Disks and Backup Media

Before adjusting RMAN and Oracle parameters, the first task is to determine the speed of the input disks and the output backup media. Obviously, the backup rate cannot be faster than the speed of the output device or the input disks.

If you back up to tape, then the easiest way to determine the speed of a tape is to measure how fast the media management software backs up an operating system file to a tape. This test will also give you an idea of the media manager throughput. If the output device is a disk, then the easiest way to measure the speed is to use a utility such as the UNIX dd command.

If the datafiles are on striped disks, then determine the stripe size as well as how many disks are striped. As mentioned previously, Oracle works best when the stripe size is 1 MB and many disks are striped. To measure the maximum input speed, you can use the UNIX dd command or some other backup utility.

### Determining the Number of Channels to Allocate

The most important rule in determining how many channels to allocate is not to allocate too many. It is a common misconception that allocating multiple channels improves performance. In fact, the number of channels *should not exceed* the number of tape drives (if backing up to tape) or the number of output disks subsystems (if backing up to disk).

If you back up to tape, then one RMAN channel per tape device should give the best performance. However, in some very rare cases, you can improve performance by allocating two or three channels per tape device. In this case, the media manager will also multiplex the streams from different channels, which can improve performance especially if the speed of the underlying disks varies.

If you are backing up to tape, then one RMAN channel per output device gives the best performance.

If you are backing up to disk, then one RMAN channel per output disk subsystem should give the best performance.

For Oracle8*i*, set the disk buffer size to 1 MB with initialization parameter DB\_FILE\_DIRECT\_IO\_COUNT.

Oracle9*i* automatically sets the size of the disk buffers.

If you back up to disk, then allocate one channel per each disk subsystem. For example, if the output is spread across three different physical disks, then allocate three channels. Rarely do you need to allocate more channels than output disks.

#### Adjusting the Size of Backup Disk Buffers in Oracle8*i*

The size of disk buffers defines the amount of data transferred in a single I/O call. Our tests on Oracle8*i* showed that the read buffer size should be set to at least 1 MB by using the initialization parameter DB\_FILE\_DIRECT\_IO\_COUNT. The parameter specifies the size in number of blocks and default value is 64.

If the input disks are striped with a 1 MB stripe size (as suggested in section “Optimal Disk Storage Configuration”), then it is especially important to set the read buffer size to 1 MB. Table 2 shows that the backup speed of 1 MB striped disks with a 1 MB read buffer is about 20 MB/s, while the speed with 128 KB disk buffers is only about 10 MB/s. Hence, setting DB\_FILE\_DIRECT\_IO\_COUNT increases the speed by 100%.

Because the Oracle8*i* backup algorithm is more efficient with larger buffer sizes, the read buffer size should be set to 1 MB even if the input disks are not striped or if the stripe size is smaller than 1 MB. Table 3 shows that disk backup performance increases from 12.05 MB/s to 21 MB/s just by increasing read buffer size from 128 KB to 1 MB. However, the read buffer size does not have such a large influence if the backup device is SBT.

Note that Oracle9*i* automatically sets the size of the disk buffers depending on the level of multiplexing. Also, due to the improvement in the Oracle9*i* backup algorithm, the performance of nonstriped disks is not dependent on read buffer sizes.

**Table 2: Oracle8*i* Backup on Nonstriped Disks with Different Disk Buffer Sizes**

Test No.	RMAN version	Output device	Number of Striped Disks	RMAN Multiplexing	Disk Buffer Size (Kb)	Async IO (Yes/No)	I/O Slaves (Yes/No)	Speed Cumulative (MB/s)	CPU utilization (%)
1.	RMAN 8 <i>i</i>	DISK	None	4	128 Kb	Yes	No	12.05 MB/s	5.19%
2.	RMAN 8 <i>i</i>	DISK	None	4	512 Kb	Yes	No	17.09 MB/s	4.75%
3.	RMAN 8 <i>i</i>	DISK	None	4	1024 Kb	Yes	No	21.48 MB/s	4.50%
4.	RMAN 8 <i>i</i>	SBT	None	4	128 Kb	Yes	No	16.67 MB/s	5.00%
5.	RMAN 8 <i>i</i>	SBT	None	4	512 Kb	Yes	No	17.24 MB/s	4.65%
6.	RMAN 8 <i>i</i>	SBT	None	4	1024 Kb	Yes	No	17.56 MB/s	4.21%

For disk backups, the backup rate depends on the disk buffer size – even though the disks are not striped.

If the backup goes to tape, and if the disks are not striped, then the backup rate does not depend on the disk buffer size.

**Table 3: Oracle8*i* Backup of Striped Disks with Different Disk Buffer Sizes**

Test No.	RMAN version	Output device	Number of Striped Disks	RMAN Multiplexing	Disk Buffer Size (Kb)	Async IO (Yes/No)	I/O Slaves (Yes/No)	Speed Cumulative (MB/s)	CPU utilization (%)
1.	RMAN 8 <i>i</i>	DISK	4	1	128 Kb	Yes	Nb	9.80 MB/s	3.05%
2.	RMAN 8 <i>i</i>	DISK	4	1	512 Kb	Yes	Nb	14.70 MB/s	3.44%
3.	RMAN 8 <i>i</i>	DISK	4	1	1024 Kb	Yes	Nb	20.00 MB/s	3.08%
4.	RMAN 8 <i>i</i>	SBT	4	1	128 Kb	Yes	Nb	9.71 MB/s	3.25%
5.	RMAN 8 <i>i</i>	SBT	4	1	512 Kb	Yes	Nb	14.29 MB/s	3.56%
6.	RMAN 8 <i>i</i>	SBT	4	1	1024 Kb	Yes	Nb	19.23 MB/s	3.95%

The fastest backup occurs when the read block size is one Mb.

### Adjusting the Level of Multiplexing

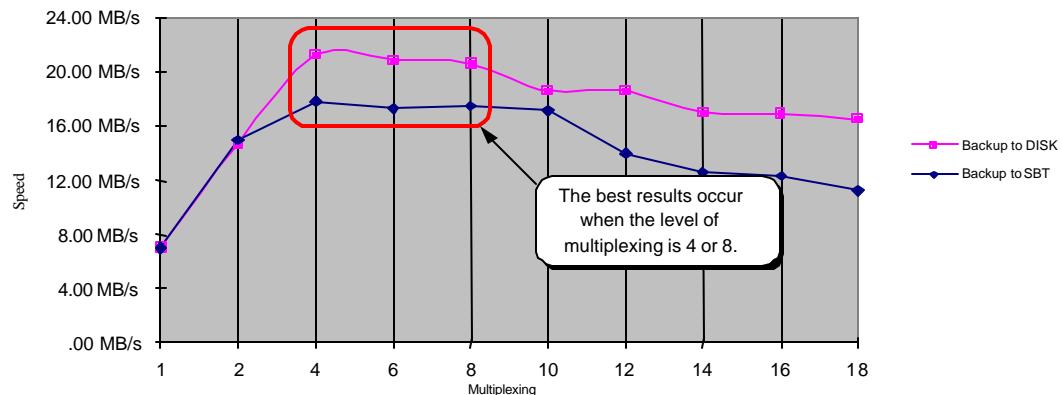
If the disks are not striped, then the multiplexing level should not exceed 8; if the disks are striped, then the multiplexing level should be 1 or 2.

The level of multiplexing is the number of files read simultaneously on a single channel. Our tests indicate that adjusting the level of multiplexing to achieve the best performance follows a simple rule: if the disks are not striped, then the multiplexing level should not exceed 8; if the disks are striped, then the multiplexing level should be 1 or 2.

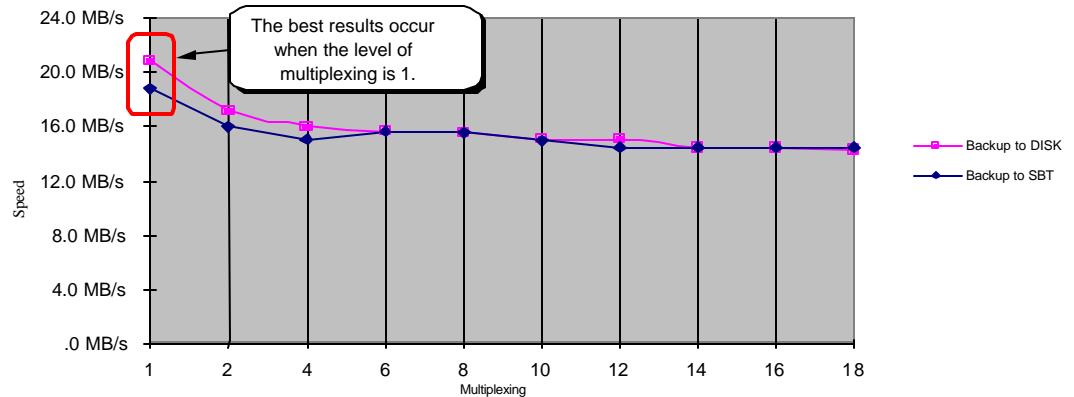
The preceding rule is proved with the results of tests displayed in the following figures:

- Figure 4 shows that the best Oracle 8*i* backup performance for *nonstriped* disks is achieved with a multiplexing level of four. We achieved a backup speed of about 18 MB/s for *sbt* and 21 MB/s for *DISK*.
- Figure 5 show that if the disks are *striped* then the best performance is achieved when Oracle8*i* does not multiplex the input disks. In this case, we achieved a speed of slightly less than 19 MB/s for *sbt* and 21 MB/s for *DISK*.
- Figure 6 and figure 7 show that Oracle9*i* exhibits similar behavior to Oracle8*i*. Figure 7 shows that Oracle 9*i* is slightly faster in a backup of striped disks: Oracle9*i* achieves 22 MB/s. The speed improvement can be explained by fact that Oracle9*i* allocates more disk buffers than Oracle8*i*, so Oracle9*i* issues more concurrent reads, which in turn causes better utilization of striped disks. On the other hand, Figure 7 also shows that an Oracle9*i* backup of striped disks is slower with a high level of multiplexing. This result occurs because Oracle9*i* automatically adjust the sizes of read buffers according to the level to multiplexing. If the multiplexing level is 18, then the disk buffer size is set to 128 KB, which is not optimal for reading disks with a stripe size of 1 MB.

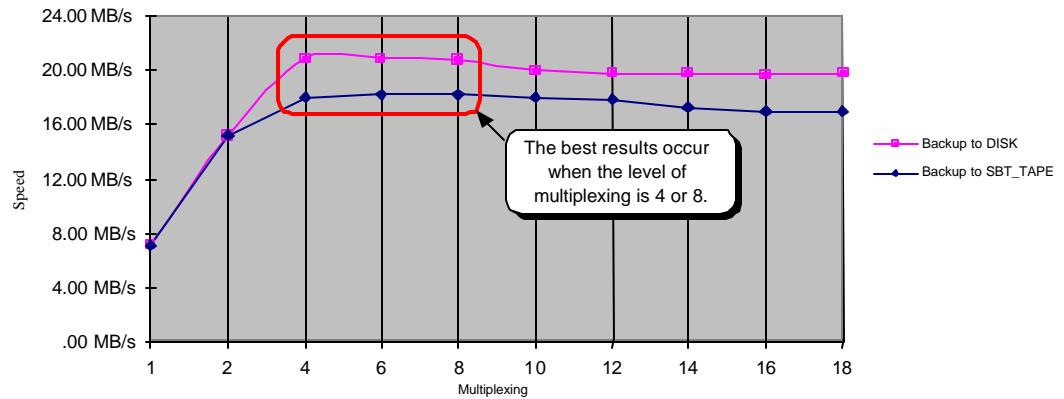
**Figure 4: Oracle8*i* Backup of Nonstriped Disks with Asynchronous I/O**



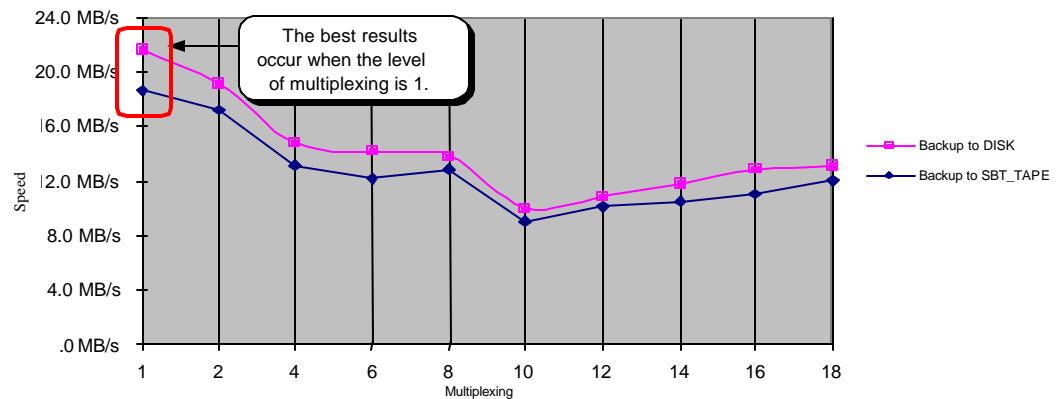
**Figure 5: Oracle8i Backup of Striped Disks with Asynchronous I/O**



**Figure 6: Oracle9i Backup of Nonstriped Disks with Asynchronous I/O**



**Figure 7: Oracle9i Backup of Striped Disks with Asynchronous I/O**



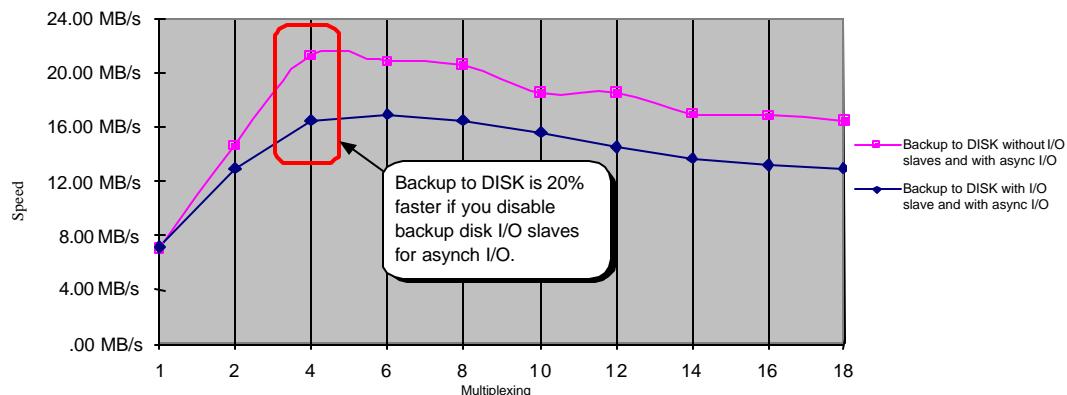
For Oracle8i, you should disable backup disk I/O slaves for asynchronous I/O. For Oracle9i, backup disk I/O slaves is automatically disabled for asynchronous I/O.

#### Disabling Backup Disk I/O Slaves for Asynchronous I/O in Oracle8i

Performance testing and investigation of our backup algorithm revealed that a combination of asynchronous I/O and backup disk I/O slaves can hurt performance. As indicated in figure 8, the Oracle8i tests clearly indicate that the backup is slower when you enable backup disk I/O slaves with asynchronous read. For example, if the multiplexing level is 4, then the backup is up to 20% slower with backup disk I/O slaves enabled.

For Oracle8i, the backup disk I/O slaves can be disabled by setting DBWR\_IO\_SLAVES initialization parameter to 0 or by setting \_BACKUP\_DISK\_IO\_SLAVES to 0. Note that setting DBWR\_IO\_SLAVES to 0 will also disable disk I/O slave processes used by the database writer process, ARCH process, and LGWR process. Setting \_BACKUP\_DISK\_IO\_SLAVES to 0 disables only backup disk I/O slaves. In contrast, Oracle9i automatically disables backup disk I/O slaves when they are not needed.

**Figure 8: Oracle8i Asynchronous I/O Backup of Nonstriped Disks With and Without Backup Disk I/O Slaves**



#### Enabling Backup Disk I/O Slaves on a Synchronous System

If the system does not support native asynchronous I/O, simulate it with backup disk I/O slaves.

If your system does not support native asynchronous I/O, or if it is disabled for some reason, then you can simulate asynchronous I/O by enabling backup disk I/O slave processes. You can enable backup disk I/O slaves with the initialization parameters DBWR\_IO\_SLAVES or \_BACKUP\_DISK\_IO\_SLAVES. Note that Oracle uses SGA memory for tape buffers when backup tape I/O slaves are enabled, so setting LARGE\_POOL\_SIZE is recommended.

Testing shows that synchronous I/O backups without backup disk I/O slaves can achieve a maximum speed of 7 MB/s. However, with backup disk I/O slaves started, the backup speed is *tripled* to 20 MB/s. As a result, the speed of synchronous backup is basically limited to the speed of a single disk – even though files are striped over many disks.

### Enabling Backup Tape I/O Slaves When Backing Up to Tape

To improve the performance of tape backups, enable the BACKUP\_TAPE\_IO\_SLAVES initialization parameter.

When RMAN writes data to a tape, the I/O to the tape is synchronous *by default*. The synchronous I/O is bad for performance because a server process can perform only one task at a time. You can simulate asynchronous I/O to tape by enabling tape I/O slave processes with the initialization parameter BACKUP\_TAPE\_IO\_SLAVES. This parameter *is disabled by default*. Note that Oracle uses SGA memory for tape buffers when backup tape I/O slaves are enabled, so setting LARGE\_POOL\_SIZE is recommended.

As shown in table 9 and table 10, both Oracle8*i* and Oracle9*i* backups to sbt are approximately 20% faster with backup tape I/O slaves enabled. For example, Oracle9*i* achieved a speed of about 16 MB/s without tape I/O slaves, while enabling tape I/O slaves boosted performance to just under 19 MB/s. Also, the speed of an Oracle8*i* backup improved from 14.85 MB/s to 18.83 MB/s, a performance improvement of more than 20%.

It is important to note that sbt implementation of some media management vendors works in that way that when Oracle call the sbt function to write data, the calls juts copies the data in internal buffers and returns immediately. In these cases the performance improvements will be much smaller.

**Table 9: Oracle8*i* Backup of Striped Disks With and Without Backup Tape Slaves**

Test No.	RMAN version	Output device	Number of Striped Disks	RMAN Multiplexing	Backup Tape Slaves (Yes/No)	Async IO (Yes/No)	Backup Disk Slaves (Yes/No)	Speed Cumulative (MB/s)	CPU utilization (%)
1.	RMAN 8 <i>i</i>	SBT	8	1	Nb	Yes	No	14.85 MB/s	2.21%
2.	RMAN 8 <i>i</i>	SBT	8	1	Yes	Yes	No	18.83 MB/s	2.31%
3.	RMAN 8 <i>i</i>	SBT	8	2	Nb	Yes	No	12.35 MB/s	2.30%
4.	RMAN 8 <i>i</i>	SBT	8	2	Yes	Yes	No	15.05 MB/s	2.41%

To achieve optimal backup speed to tape, enable backup tape I/O slaves.

**Table 10: Oracle9*i* Backup of Striped Disks With and Without Backup Tape Slaves**

Test No.	RMAN version	Output device	Number of Striped Disks	RMAN Multiplexing	Backup Tape Slaves (Yes/No)	Async IO (Yes/No)	Backup Disk Slaves (Yes/No)	Speed Cumulative (MB/s)	CPU utilization (%)
1.	RMAN 9 <i>i</i>	SBT	8	1	No	Yes	No	16.45 MB/s	2.25%
2.	RMAN 9 <i>i</i>	SBT	8	1	Yes	Yes	No	18.97 MB/s	2.41%
3.	RMAN 9 <i>i</i>	SBT	8	2	No	Yes	No	15.35 MB/s	2.12%
4.	RMAN 9 <i>i</i>	SBT	8	2	Yes	Yes	No	16.15 MB/s	2.75%

To achieve optimal backup speed to tape, enable backup tape I/O slaves.

### Increasing BLKSIZE When Backing Up to Tape

The RMAN channel parameter `BLKSIZE` defines the size of the tape buffers. Our testing shows that the size of tape buffers should be at least 256 KB. In some cases, even larger values improve performance.

**For tape backups, the size of the tape buffers must be at least 256 KB.**

You can change the size of each tape buffer with the `PARMS` parameter of the `ALLOCATE CHANNEL` or `CONFIGURE CHANNEL` command. If the `BLKSIZE` parameter for `PARMS` is supported on your platform, then you can set it to the desired size of each buffer. For example, to set tape buffer size to 512 KB, configure an SBT channel as follows:

```
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS="BLKSIZE=524288";
```

In Oracle8*i*, the default `BLKSIZE` is 64 KB, whereas in Oracle9*i* the default is increased to 256 KB.

**Table 11: Oracle 8*i* Backup of Striped Disks to Tape with Different Tape Buffer Sizes**

Test No.	RMAN version	Output device	Number of Striped Disks	RMAN Multi plexing	Tape Buffer Size (Kb)	Async I/O (Yes/No)	I/O Slaves (Yes/No)	Speed Cumulative (MB/s)	CPU utilization (%)
1.	RMAN 8 <i>i</i>	SBT	8	1	64 Kb	Yes	No	15.15 MB/s	5.21%
2.	RMAN 8 <i>i</i>	SBT	8	1	256 Kb	Yes	No	18.83 MB/s	2.31%
3.	RMAN 8 <i>i</i>	SBT	8	2	64 Kb	Yes	No	13.35 MB/s	8.30%
4.	RMAN 8 <i>i</i>	SBT	8	2	256 Kb	Yes	No	15.05 MB/s	2.41%

To achieve optimal backup speed to tape, tape buffer size must be at least 256 KB.

**Table 12: Oracle 8*i* Backup of Nonstriped Disks to Tape with Different Tape Buffer Sizes**

Test No.	RMAN version	Output device	Number of Striped Disks	RMAN Multi plexing	Tape Buffer Size (Kb)	Async IO (Yes/No)	I/O Slaves (Yes/No)	Speed Cumulative (MB/s)	CPU utilization (%)
1.	RMAN 8 <i>i</i>	SBT	None	4	64 Kb	Yes	No	14.60 Mb/s	5.21%
2.	RMAN 8 <i>i</i>	SBT	None	4	256 Kb	Yes	No	17.56 Mb/s	4.21%
3.	RMAN 8 <i>i</i>	SBT	None	6	64 Kb	Yes	No	14.85 Mb/s	8.30%
4.	RMAN 8 <i>i</i>	SBT	None	6	256 Kb	Yes	No	17.05 Mb/s	4.30%
5.	RMAN 8 <i>i</i>	SBT	None	8	64 Kb	Yes	No	14.98 Mb/s	7.65%
6.	RMAN 8 <i>i</i>	SBT	None	8	256 Kb	Yes	No	17.43 Mb/s	4.30%

To achieve optimal backup speed to tape, tape buffer size must be at least 256 KB.

### Setting LARGE\_POOL\_SIZE If Oracle Fails to Allocate Shared Memory

If disk or tape I/O slaves are enabled, then Oracle uses the SGA to allocate memory for disk and tape buffers. If Oracle reports an error in the `alert.log` stating that it does not have enough memory and that it will not start I/O slaves, then set or increase the initialization parameter `LARGE_POOL_SIZE`. The message in the `alert.log` looks something like the following:

```
ksfqxcre: failure to allocate shared memory means sync I/O will be
used whenever async I/O to file not supported natively
```

For Oracle8*i*, the formula for setting LARGE\_POOL\_SIZE (for backup) is as follows:

```
LARGE_POOL_SIZE = (
    4 *
    <number of allocated channels> *
    DB_BLOCK_SIZE *
    DB_DIRECT_IO_COUNT *
    <level of multiplexing>
) +
(
    4 *
    <number of allocated channels> *
    <size of tape buffer>
)
```

In Oracle9*i*, the total size of disk buffers is limited to 16 MB per channel. So, the Oracle9*i* formula for LARGE\_POOL\_SIZE for backup is as follows:

```
LARGE_POOL_SIZE = <number of allocated channels> *
(
    16 MB +
    <size of tape buffer>
)
```

The size of a single tape buffer is defined by the RMAN channel parameter BLKSIZE.

#### **Investigating the V\$BACKUP\_ASYNC\_IO View as a Last Resort**

If you have completed all of the tuning steps and find that the backup is still slow, then as a last resort you can investigate the V\$BACKUP\_ASYNC\_IO view to locate the cause of the slowdown. Follow these steps:

1. If the V\$BACKUP\_ASYNC\_IO view *is empty* and the V\$BACKUP\_SYNC\_IO view *is not empty*, then the backup is not being performed in asynchronous mode. So enable backup tape I/O slaves, and if the system does not support native asynchronous I/O, then also enable backup disk I/O slaves.
2. Query V\$BACKUP\_ASYNC\_IO to determine the effective reading speed during the backup. This value is reported in the column EFFECTIVE\_BYTES\_PER\_SECOND in the row that has the TYPE column equal to AGREGGATE. This row represents the backup speed. Sometimes backups are slow not because reading or writing is slow, but because of other issues: connecting to the media manager, retrying while starting media manager processes, committing the backup of the tape, and so on. In such cases, troubleshoot the problem in the media management software.

3. Investigate the `IO_COUNT`, `READY`, `SHORT_WAITS`, and `LONG_WAITS` columns:
  - The column `IO_COUNTS` represents the total number of the I/O calls.
  - The `READY` column is the number of asynchronous I/O reads or writes for which a buffer was immediately ready for use.
  - The column `SHORT_WAITS` represents the number of times that a buffer was not immediately available, but a buffer became available after doing a nonblocking poll for I/O completion.
  - `LONG_WAITS` is the number of times that a buffer was not immediately available, and only became available after a blocking wait was issued. In other words, the number of I/O calls where Oracle waited for the disk or tape.

**The bottleneck is in the datafile that has the largest ratio of `LONG_WAITS/IO_COUNTS`.**

As you can see from the preceding explanation, the value `LONG_WAITS` is increased by a value of 1 when a device or disk is too slow. So, the bottleneck is in the datafile that has the largest ratio for `LONG_WAITS/IO_COUNTS`.

## CONCLUSION

Tuning an RMAN backup does not have to complex and labor intensive. By following the simple procedures outlined in this document, you can always achieve good backup performance. You can achieve a fast Oracle backup by obeying the following rules:

- Use a disk configuration with all files striped across all disks using one MB stripe size.
- Use asynchronous I/O. If the system does not support native asynchronous I/O, simulate it with backup disk and tape I/O slaves.
- Do not allocate too many channels. In general, you should allocate one channel for each output tape device or disk subsystem.
- Do not set a high level of multiplexing. If the underlying disks are not striped, then the optimal multiplexing level is between 4 and 8. If underlying disks are striped, then the optimal multiplexing level is 1 or 2.
- The sizes of the disk and tape buffers should be relatively large. For Oracle8*i*, set the disk buffer size to 1 MB by changing the initialization parameter `DB_FILE_DIRECT_IO_COUNT`. For Oracle9*i*, this change is not needed.

These suggestions are based on careful analysis of current Oracle algorithms combined with results of real-world tests.



Tuning Oracle Recovery Manager

July 2001

Author: Senad Dizdar

Contributing Authors: Lance Ashdown

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

**Worldwide Inquiries:**

Phone: +1.650.506.7000

Fax: +1.650.506.7200

[www.oracle.com](http://www.oracle.com)

Oracle Corporation provides the software  
that powers the internet.

Oracle is a registered trademark of Oracle Corporation. Various  
product and service names referenced herein may be trademarks  
of Oracle Corporation. All other product and service names  
mentioned may be trademarks of their respective owners.

Copyright © 2001 Oracle Corporation  
All rights reserved.