

# Tuning Oracle Databases with Solid State Accelerators

by Kwajah Mohiuddin  
Sigma Solutions, Inc.

Updated February 15, 2003

## Background

Disk access times are generally a key issue in most commercial data processing environments. It is disk access which is commonly the longest part of the data acquisition process. Information processing technology has continually improved by leaps and bounds in the case of CPU's, memories, and other components. The hard disk arena has improved primarily in the direction of creating high areal densities (capacity) and a lower cost per megabyte. Progress has not been very great in improving the speed of data access from the disk drive. Speed of access in a drive is mainly limited by the rotating speed of the disk or the linear speed of the access arm. A modern day CPU flies along at 1.5GHz, but it must wait for data from a disk drive which is chugging along at 15,000rpm or less. The ever-widening gap in speed differential is continuing to increase very rapidly and is causing I/O bottlenecks to become more troublesome than ever.

Disk and controller manufacturers have employed various schemes to overcome the mechanical limitations of hard disks and make data access speed seem to look better. One of the most common methods is to try and buffer data at different levels and HOPE to find the data to avoid a physical read. Disk Caching controllers are loaded with large memories to attempt buffer frequently used data. The drives themselves have intelligent CPU's and onboard memory. Operating systems have buffer cache where frequently used data can be stored. The above schemes are successful in that they do increase the probability of finding the data in any of the caches so that the mechanical part of fetching data can be avoided as much as possible. If the disk block is not found in the operating system buffer then maybe it will still be in the controller buffer and if not then hopefully in the drive buffer. This downward cascading style search might be called the HOPE algorithm. In addition to HOPE, there are other schemes like parallelizing track seeks and sector interleaving which provide some benefit in I/O intensive environments.

Effectively, if the application cannot find data in any of the buffers by using the HOPE algorithm and/or by any of the other mechanisms provided in the hardware, then the processing is slowed down by the physical disk access. Applications can very easily slip through the buffering scheme by reading and writing fresh data every time. Nothing will be found in any of the buffers and the data will have to be fetched from the drive. If application response times are critical, one method for responding to this performance drain is to speed up this process by using a solid state accelerator. Solid state disk technology is not mechanical and features a consistent access time that is at least 200 times faster than today's fastest disk drives.

A solid state accelerator uses high speed memory organized as a drive. To the server, it appears as a conventional disk drive when in reality the server is dealing with a mass of external memory. In a solid state accelerator, there are no performance-robbing mechanical limitations due to moving heads or spinning platters. Data can be written to the solid state drive at memory speeds. There is no seeking of tracks and data can be found and transferred to the server in a few microseconds, which is 200-350 times faster than the 7-10 milliseconds a regular hard drive takes to seek to a track and transfer data. Data files on a Unix file system can be created on the solid state accelerator and the operating system will not know the difference.

But data residing in memory is volatile and will disappear as soon as the power is turned off. Solid state disk manufacturers have effectively resolved this situation by adding inbuilt UPS functionality and packaging regular hard disk drives internally to the solid state drive as one all-

inclusive and compact device. In the event of power fluctuation, circuits inside the solid state accelerator sense the loss and automatically switch over to power from the internal battery-backed UPS. All data from the solid state drive is then saved to the internal disk drive. Once data is safely saved to the internal disk, the device will switch off and wait for the power to return as the data is now protected. When stable power is re-established, data is copied from the internal disk drive back to the solid state accelerator ( memory area ) and the files are made available again to be accessed at memory speeds. There is virtually no risk of ever losing data with these safe guards in place. A file created on a device like this is equally if not more safe than a file created on a regular disk drive.

The one detractor to a solid state accelerator is that it is more expensive than a conventional hard disk drive. That means there is an economic limitation on the use of solid state accelerators and the decision to employ one or more of these devices will be driven by cost versus performance benefits. Essentially the system architect or integrator will have to decide upon a scheme to provide the biggest bang for the buck.

### Using A Solid State Drive With Oracle

In a flat file based application it is easy to identify the HOT (heavily used) files and place them on a solid state accelerator to derive maximum and immediate performance benefit. When a database application is involved it is a much more complex scenario. A thorough understanding of the database and the read and write pattern of the application is required before anything should be attempted. Many times the application itself will not provide any room for improvement due to internal locking or waits for other processing to complete.

For an application to show benefit it must be performing disk access at a very fast pace and must be writing or reading fresh data. Oracle performs its own smart optimization by buffering all writes and reads in the System Global Area. Data is written to the disks only on a "need to" basis. Over and above this database functionality, the operating system performs its own buffering. This means there are two sets of buffers to be scanned before a physical disk access is required. This double buffering strategy works well in many applications. What it does is add a second level of caching to the HOPE algorithm. Oracle always does a synchronized write, meaning the operating system buffers are bypassed for all writes to disk. These buffers are used for the reads only if they are not marked as old or dirty. This theory is largely true except in some implementations of Oracle where the operating system buffers are bypassed completely. For example, parallel server bypasses operating system buffers to ensure read consistency in various instances.

Under Oracle, the greatest I/O activity exists with the rollback segments, redo logs and temporary segments. These files are excellent candidates for the solid state accelerator depending on the application. Oracle data files can be placed on a solid state accelerator when there are a lot of random reads and writes to specific tables.

Ensuing performance benefits can be extremely varied and generally depend on how much solid state disk is configured for the application. The following paragraphs explain using solid state accelerators with various Oracle files and the type of application which could show maximum performance benefit.

### Redo Logs

When an application commits data to the database, Oracle writes the commit information to the redo log files. An Oracle process called the log writer performs the task of writing the data to disk from the SGA. The user data could still be in Oracle's SGA but the transaction is deemed complete. The above scheme allows Oracle to support high transaction rates by saving only bare minimum data to the disk. The user data is saved to disk later on by the DBWR process which wakes up at predefined intervals.

An application with high transaction rates will write large volumes of data to the redo logs in a very short time. Redo logs, when placed on a solid state accelerator, can provide dramatic performance benefits in such a scenario. Performance gains of up to 2000% are possible though it is much more common to see improvements in the range of 20-80% by putting the redo logs on a solid state accelerator. In a transaction processing environment, data is written to both redo logs and rollback segments. Performance benefits will be greatly enhanced when both redo logs and rollback segments are placed on solid state drives. Only an application with a high volume of inserts, updates and deletes will show improvement if setup in this manner. A Decision Support environment which only reads data will not show benefit.

### **Rollback Segments**

Oracle stores previous images of data in the rollback segments. When an application makes changes to the data it is stored in the rollback segments until the user commits the transaction. All processes read the previous image of the data from the rollback segments until the transaction is committed. Oracle does its own buffering of the rollback segments in the SGA. But when large transactions occur they are written to the disk. If other processes need this previous image of the data then it will be read from the rollback segments. Rollback segments on a solid state accelerator will speed up both the reads and the writes. Only a transaction processing environment will benefit from putting rollback segments on a solid-state accelerator. A decision support system which only analyzes the data will not show benefits.

### **Temporary Segments**

Oracle uses temporary segments to store intermediate files. These files could be the result of sub queries or temporary sort files. Data is both written to and read from the temporary segments by the various Oracle processes. Small sorts are entirely performed in the SGA ( memory ) and the large sorts are performed by using disk space in the temporary segments. An application performing large sorts or making heavy use of the parallel query option can show immediate results with a solid state accelerator. A transaction processing environment with low volume sorts will not show benefits with temporary segments on a solid state accelerator. Applications which benefit from solid state accelerators are Decision Support Systems which retrieve or sort large volumes of data. Transaction processing environments with complex queries could also benefit from using a solid state accelerator.

### **Data Files**

Oracle stores data from tables and indexes in table spaces residing on Oracle's data files. All data is buffered and stored in the SGA and as indicated earlier, will go through the operating system's buffers as well. Putting data files on a solid state accelerator will provide benefit only in unique situations. The writes to the data files are done by the DBWR at predefined intervals and are therefore not an immediate priority. Excessive reads can be avoided by increasing the number of buffers and creating the heavily used tables using the CACHE option. This will improve the possibility of the table data blocks being available in the SGA on most occasions. This technique can be used for smaller tables by increasing the size of main memory and the SGA. Large tables which are read or written randomly at high frequencies will have to be placed on a solid state accelerator. Such tables can be created on separate table spaces setup on the solid state accelerator.

### **Determining The Need For a Solid State Disk**

For a solid state accelerator to be used with an Oracle application it must have a consistently high I/O rate which will saturate the SGA and also the operating system buffers. It has to beat the buffering of both Oracle and the operating system. The saturation of the SGA is indicated by the Hit Ratio. This can be determined by using a monitoring tool like the Oracle Monitor or looking

directly at the V\_\$ tables. High transfer rates to a particular data file or rollback segment could indicate possible performance benefits.

The operating system must be monitored for I/O activity. Utilities like sar and iostat can be used to analyze the operating system buffer activity. If the operating system buffers are large enough then there will be more logical than physical accesses. Now if the database ( Oracle's monitor ) shows a high number of physical reads but the operating system does not, then the chances of getting any benefit from a solid state accelerator would be minimal. Only an application which causes physical reads both at the database level and at the operating system level will benefit from the solid state accelerators performance attributes. The exception to this is the Parallel Server which always bypasses the operating system buffers.

An application with intermittently high I/O rates for short periods of time will not generally show much performance improvement on a solid state accelerator because the database could work on flushing the buffers during the lean periods. I/O transfer rate rather than how much I/O is performed determines the need for a solid-state accelerator. The reads and writes must also slip through the buffering schemes before any benefits can be seen.

### Raw Devices

A solid state accelerator can be configured as a raw device under Unix. Data written or read from any disk configured as a raw device always bypasses the operating system buffers. This will eliminate the overhead of buffer management. Raw devices are good for bulk volume writes. A process shows good I/O bandwidth when large volumes of data are written at one time, because it is now bypassing the buffering scheme. A process which writes data at random in small bits and pieces will run slow because it has lost the benefit of using the buffers and now has to wait for the disk to respond. Using a solid state accelerator as a raw device under Oracle will increase the throughput because the data will move directly from Oracle's SGA to the solid state accelerator.

### RAM Disks

RAM disks have been available with every operating system for a long time. A RAM disk is a part of main memory that can be designated as a drive and the system will use that area to store files or any other user data. The RAM disk will cost the same as main memory and can be good for storing small volumes of data or doing some high speed sorts in a flat file based environment. However the main disadvantage of using RAM disk is that the CPU now performs the I/O rather than delegating work to the I/O controller. A process performing I/O with RAM disks will run very fast but it will start loading the CPU with read and write requests. This can start to slow down the other processes. The whole system could appear sluggish when large data volumes are written to the RAM disk, because the workload on the CPU has increased.

Placing Oracle's redo logs, rollbacks or temporary segments on RAM disks could potentially slow the entire system down in a heavy transaction processing environment. Also, the effect of RAM disks on Symmetric Multi Processing Machines with multiple CPU's must be evaluated. A SMP environment with multiple CPU's and RAM disks could potentially saturate the system bus and cause a rapid slow down of the whole system in some hardware architectures. Herein lies another advantage of the solid state accelerator – the ability to share contents across multiple servers in clustered scenarios which is not possible with internal server memories.

Also recall that anything written to RAM disks is volatile and does not come with the automatic battery backup capabilities of the solid state accelerator. If power is lost or the machine is rebooted, data will be lost unless there are other planned alternatives.

From a hardware point of view, a RAM disk will use up more slots in the backplane of the computer because more memory boards will have to be installed, thereby limiting the number of slots available for other purposes.

### Some Performance Figures

As mentioned earlier performance figures for an application are very subjective. The following table provides some statistics on the disk parameters and performance figures. The numbers are from a Unix server running Oracle database and Imperial MegaRam-5000 solid state accelerator.

Activity / Specification	Solid-State Drive	Mechanical Hard Drive
Latency Time	0	2 milliseconds
Seek Time	0	5 milliseconds
Total Access Time	0.035 milliseconds	7 milliseconds
ReDoLogs (4x100 MB file)		
Insert	<b>08:05,6</b> (43% incr)	<b>14:06,2</b>
Update	<b>01:42,5</b> (70% incr)	<b>05:37,0</b>
Delete	<b>01:24,0</b> (73% incr)	<b>05:16,5</b>
Cost per Megabyte	\$2.00 - 7.50	\$.01-.25

### Conclusion

Solid state accelerators provide outstanding performance improvement in Oracle environments depending on the robustness of the application. The DBA and the Oracle consultant can utilize the much faster access times of solid state accelerators by placing the "HOT " files on these drives. It does not replace disk drives or cached arrays but is used in key areas to resolve I/O bottlenecks. Applications that are not I/O intensive will not benefit from this technology.

### About the Author

Kwajah Mohiuddin has over 20 years of experience in the industry. He has worked extensively in various environments as project leader, Unix administrator and DBA. His special interests are Parallel Processing, Object Technology and Distributed databases.

Copyright 2003, Imperial Technology, Inc. All company and product names used herein may be trademarks, or registered trademarks, of their respective companies.

**Further Information**

**310-544-9439 voice**

**310-544-9309 fax**

[info@evenenterprises.com](mailto:info@evenenterprises.com)

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