Faster Oracle Performance with Solid State Disks

By Woody Hutsell and Mike Ault
Texas Memory Systems

July 2008

Contents

Section 1

Executive Summary

This whitepaper discusses methods for improving Oracle database performance using solid state disks to accelerate the most resource-intensive data that slows performance across the board.

To this end, it discusses methods for identifying I/O performance bottlenecks, and it points out components that are the best candidates for migration to a solid state disk. An in-depth explanation of solid state disk technology and possible implementations are also included.

For more in-depth information, visit www.superSSD.com or contact one of the following:
-Existing customers contact support@superSSD.com.
-Potential customers contact sales@superSSD.com.

Section 2

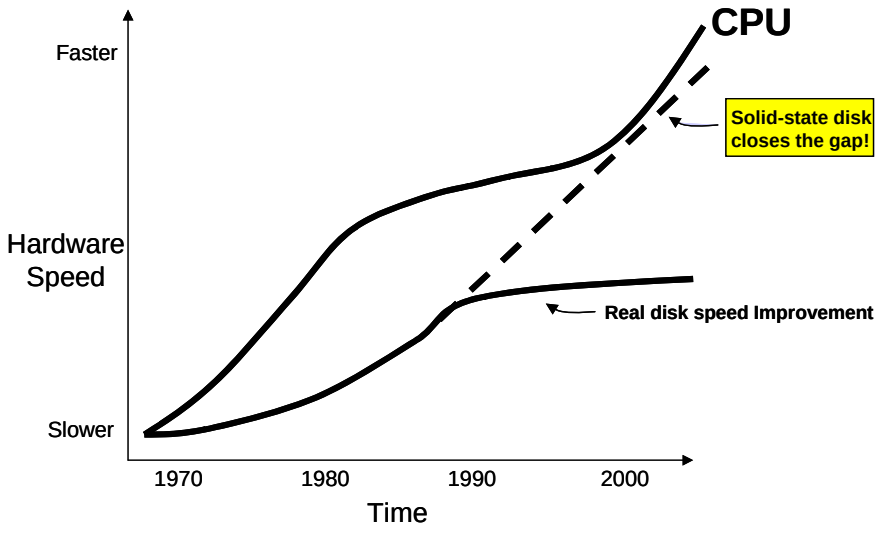
The Problem of I/O Wait Time

Often, additional processing power alone will do little or nothing to improve Oracle performance. This is because the processor, no matter how fast, finds itself constantly waiting on mechanical storage devices for its data. While every other component in the "data chain" moves in terms of computation times and the raw speed of electricity through a circuit, hard drives move mechanically, relying on physical movement around a magnetic platter to access information.

In the last twenty years, processor speeds have increased at a geometric rate. At the same time, however, conventional storage access times have only improved marginally (see Figure 1). The result is a massive performance gap, felt most painfully by database servers, which typically carry out far more I/O transactions than other systems. Super fast processors and massive amounts of bandwidth are often wasted as storage devices take several milliseconds just to access the requested data.

When servers wait on storage, users wait on servers. This is I/O wait time. Solid state disks are designed to solve the problem of I/O wait time by offering 250x faster access times (.02 milliseconds instead of 5) and 80x more I/O transactions per second (400,000 instead of 5000) than RAID.[1]

Figure 1: Comparing Processor and Storage Performance Improvements



Section 3

Traditional Approaches to Oracle Performance

Decreasing application performance under heavy user loads is not a new story for most enterprises. The last 3 years have seen dramatic changes in demands placed upon database servers. While the number of users of database system has increased so has the average amount of data stored in databases. Add to this the demand for more complex business analysis has increased the complexity of the work done by database servers. The Combination of more users, greater volume of data and more complex queries has frequently resulted in slower database response. The knee-jerk reaction to this problem is to look at two likely sources for database performance problems:

- Server and processor performance. One of the first things that most IT shops do when performance wanes is to add processors and memory to servers or add servers to server farms.
- SQL Statements. Enterprises invest millions of dollars squeezing every bit of efficiency out of their SQL statements. The software tools that assist programmers with the assessment of their SQL statements can cost tens of thousands of dollars. The personnel required for painstakingly evaluating and iterating the code costs much more. Dozens of consulting firms have appeared in the last few years that specialize in system tuning, and their number one billable service is SQL tuning.

In many cases, the money spent in these two pursuits can be significant, whereas the return is often disappointing. Server performance and SQL tuning alone does not get to what is frequently the true cause of poor database performance: the gap between processor performance and storage performance. Adding servers and processors will have a minimal impact on database performance and will compound the resources wasted as even more processing power waits on the same slow storage. Tuning SQL can result in performance improvements, but even the best SQL cannot make up for poor storage I/O. In many cases, features that rely heavily on disk I/O cannot be supported by applications. In particular, programs that result in large queries and that return large data sets are often removed from applications in order to protect application performance.

When system administrators look to storage they frequently try three different approaches to resolving performance problems:

- Increase the number of disks. Adding disks to JBOD or RAID is one way to improve storage performance. By increasing the number of disks, the I/O from a database can be spread across more physical devices. As with the other approaches identified, this has a trivial impact on decreasing the bottleneck.
- Move the most frequently accessed files to their own disk. This approach will deliver the best I/O available from a single disk drive. As is frequently pointed out, the I/O capability of a single hard disk drive is very limited. At best, a single disk drive can provide 300 I/Os per second. Fast solid state disk is capable of providing 400,000 I/Os per second.
- Implement RAID. A common approach is to move from a JBOD (just a bunch of disks) implementation to RAID. RAID systems frequently offer improved performance by placing a cached controller in front of the disk drives and by striping storage across multiple disks. The move to RAID will provide additional performance, particularly in instances where a large amount of cache is used.

Section 4

Introduction to Solid State Disks

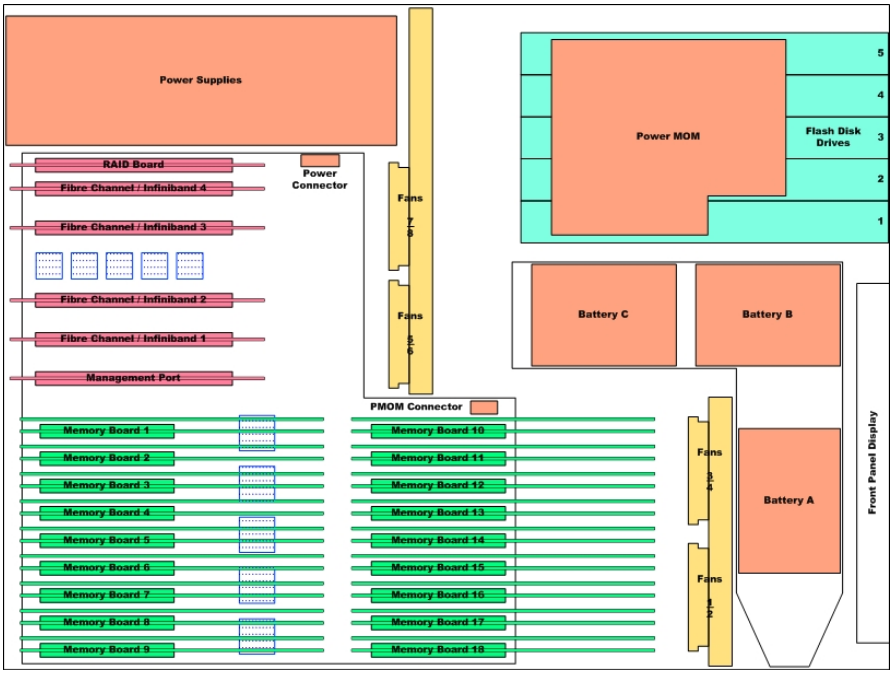
Strictly, a solid state disk (or SSD) is any storage device that does not rely on mechanical parts to input and output data. Typically, however, the term refers to storage devices that use memory (DDR or Flash) as the primary storage media. Data is stored directly on memory chips and accessed from them. This generally results in storage speeds far greater than is even theoretically possible with conventional, magnetic storage devices. To fully make use of this speed, SSDs typically connect to servers or networks through multiple high-speed channels.

DDR Solid State Disks

What separates a DDR solid state disk from conventional memory is non-volatility. A DDR SSD typically includes internal batteries and backup disks so that, in the event of power loss or shutdown, the batteries keep the unit powered long enough for data to be written onto backup disks. Because of this, SSDs offer the raw speed of system memory without the disadvantage of losing data when powered down. Because of the lack of mechanical devices in the main data chain, SSDs typically have lower maintenance costs and higher reliability (including a higher MTBF) then conventional storage.

Figure 2: An Inside Look at the RamSan-440 Solid State Disk.

DDR-RAM memory, the primary storage media, fill the back of the unit. The front contains backup batteries, backup mechanical drives, and a front panel user interface.



Cached Flash Solid State Disks

Cached flash solid state disks seek to balance the performance offered by a large DDR cache and the fast reads, high density and lower price per capacity of flash memory. A cached flash solid state disk is as fast as a DDR solid state disk for cache hits and still 20 times faster than typical hard disk based solutions if there is a cache miss (i.e. a read from the flash memory).

Section 5

Identifying I/O Wait Time

Looking at operating system performance is the best way to identify I/O wait time. The tools to evaluate operating system performance vary by operating system. The following text gives some idea of the tools available.

Windows NT/2000/2003/XP

For Microsoft Windows operating systems the best tool for system performance analysis is Performance Monitor. Unfortunately, Performance Monitor does not provide actual I/O Wait Time statistics. It does, however, include real time processor performance levels. "Processor: % Processor Time" measures the actual work being done by the processor. If a system is hit hard by transactions and yet "% Processor Time" is well under 100% it is possible to infer sever I/O wait time. Systems that implement solid state disks will typically show high "% Processor Time" numbers.

As an example, two screen shots are included from Windows Performance Monitor. The tested system has dual Intel Xeon 2.8 GHz processors, 1 GB RAM, and is running Windows Server 2000.

Figure 3: Processor Performance When Writing to Hard Disk

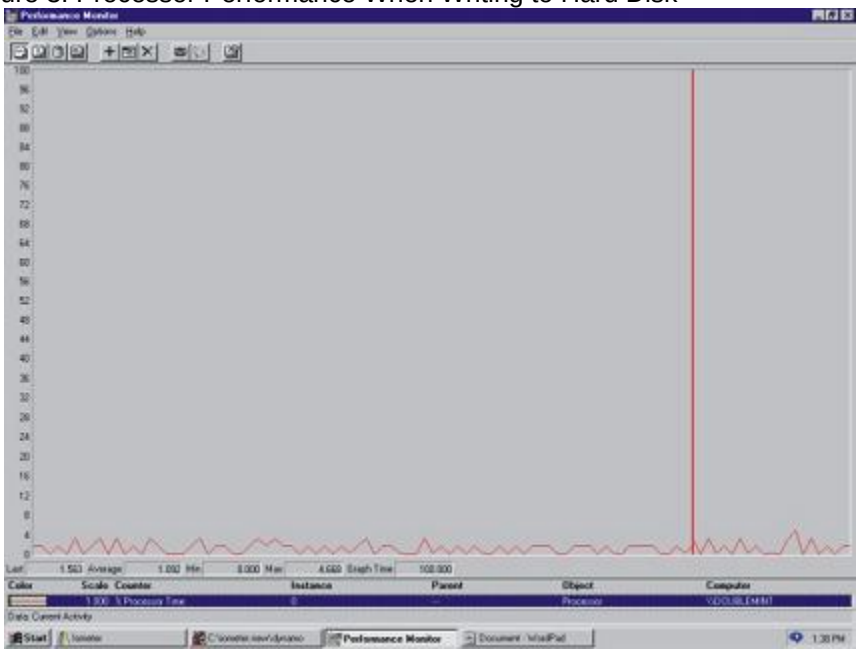


Figure 3 shows the "Processor: % Processor Time" for a Server 2000 system with Intel's IOMeter performing 100% random writes to a hard disk drive. Here you can see that the processor utilization averages around 1.8%. However, if you

were to try and run additional applications on this system, the processor utilization would only marginally increase because the processor is tied up waiting on I/O from the hard disk drive. In this example, IOMeter shows that on average there were 150 writes per second (150 IOPS) to the disk drive.

Figure 4: Processor Performance When Writing to a RamSan SSD

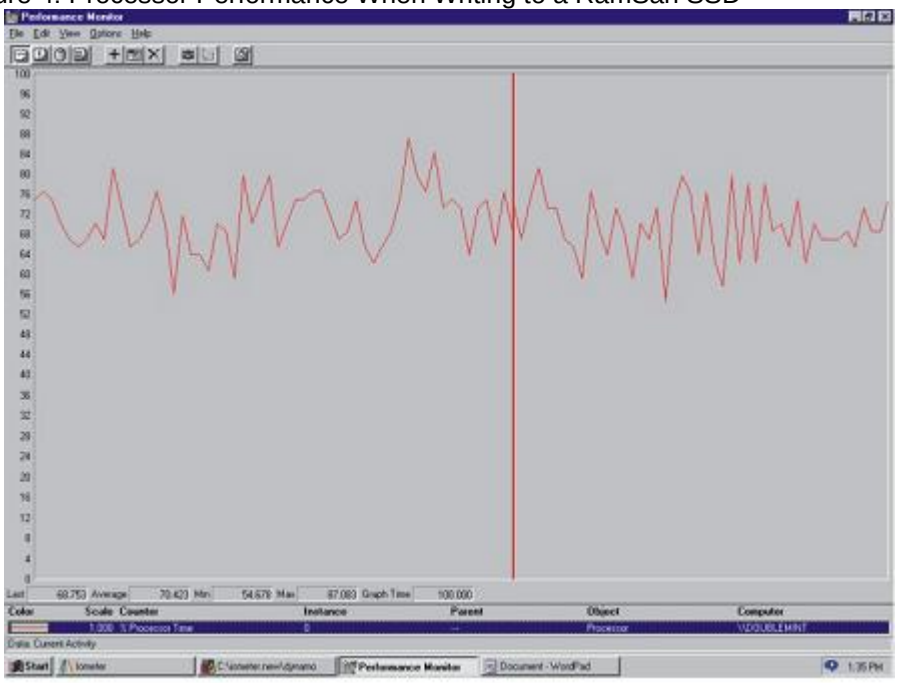


Figure 4 shows the exact same system and exact same access specifications in IOMeter running against a Texas Memory Systems RamSan. In this example, the processor averages 68% utilization. The IOMeter shows that 37,000 writes per second are going to the RamSan (37,000 IOPS). The SSD has provided a 3677% increase in server CPU utilization with total saturation of the Fibre Channel Host Bus Adaptor. And since this performance, higher than the best RAID, is only a fraction of the RamSan's capabilities, several similar servers could be hooked up and receive similar results.

In addition to processor indicators, Texas Memory Systems recommends looking at the Physical Disk: "Average Disk Queue Length" and Physical Disk: Disk Bytes per Second to detect bottlenecks in the disk subsystem. If these values are consistently high, consider moving files that are located on that disk to the solid state disk. A Disk Queue Length greater than 3 indicates a problem. Texas Memory Systems has developed an extensive whitepaper with suggestions for collecting and analyzing Windows Perfmon results. Additionally, please contact your Texas Memory Systems sales person for a free I/O analysis of your report.

UNIX

For UNIX operating systems, the following commands are useful: `top`, `iostat`, and `sar`. Depending on the command you will receive slightly different output.

The `top` command, when executed on a Solaris system, produces results that have the following format:

```
load averages:  0.09,  0.04,  0.03
16:31:09
66 processes:  65 sleeping, 1 on cpu
CPU states: 69.2% idle, 18.9% user, 11.9% kernel, 0.0% iowait, 0.0% swap
Memory: 128M real, 4976K free, 53M swap in use, 542M swap free
```

The key is that this command provides the "% iowait" for the system. It is important to note, that `top` provides a snapshot of performance.

It is also reasonable to look at the `vmstat` command. This command will tell you how frequently your system is paging to virtual memory (disk). If you have frequent paging, it makes sense to consider adding RAM to your system or using a solid state disk for paging. Paging to disk is another way that hard disk drives can introduce bottlenecks into system performance.

Oracle

Every version of Oracle since version 8.1.7.2 comes with the Statspack utility to monitor database performance. In version 10g, Oracle introduced the AWR, or Automatic Workload Repository along with ADDM, Automatic Database Diagnostic Monitor as an extra cost option to their Enterprise Manager Tool partly in response to the growing complexity and cost of managing and tuning large databases. While AWR/ADDM was a significant enhancement to Statspack, even the best database tuning can not fix slow disk subsystems. A Statspack report or AWR report should be captured during peak performance periods and can provide I/O related statistics to assist in determining which files would benefit from placement on solid state disks.

Oracle uses 1-10 database)or dirty buffer) writer processes to write changed data and rollback/undo data to disk as well as log writer processes to write redo log data and archive log processes to write archived logs to secondary storage locations. However, each user process in Oracle does its own read from the disks, in large Oracle systems this can mean hundreds if not thousands of concurrent disk read requests. This requirement for large amounts of concurrent read access to the storage system is usually poorly understood by system administrators and is a major cause of contention issues in improperly configured disk RAID setups.

With regard to Oracle IO issues, the latest version or Oracle has added to the demands placed upon storage. In the good old days Oracle had a database, some redo logs, archive logs and backups. Now along with these Oracle provides flashback functionality which while greatly enhancing the ability of the database DBA or developer to react to changes, it is one more area where data is being stored. The change from rollback segment to the new undo tablespace was a

great benefit, but there are undo tablespaces in production systems that exceed 800GB, and rapid IO is critical to healthy system performance.

As mentioned at the beginning of this paper one of the ways that system performance improvement was approached was to add more systems. Oracle new clustered database, RAC is one approach to add more systems to meet the demands of some of these massive database systems. One key point to this architecture is that all of these individual computers may be adding more computer power, but they all access the same shared disk systems.

The "Top 5 Timed Events" is the first place to look to begin understanding if the database is I/O bound. Figure 5 shows the top 5 events from a sample Statspack report.

Figure 5: Sample Oracle Statspack

```
Top 5 Timed Events
~~~~~~~~~~~~~~~~~~                                                     % Total
Event                                               Waits    Time (s) Ela Time
--------------------------------------------- ------------ ----------- --------
db file sequential read                       181,554,617      97,213    89.13
CPU time                                                       10,151     9.31
control file parallel write                        58,062         924      .85
db file scattered read                            287,924         535      .49
latch free                                          7,176         103      .09
          -------------------------------------------------------------
```

The "Top 5 Timed Events" provide a snapshot of the database activity during the time period that a statspack report covers. If these top events show that a majority of the database time is spent handling disk I/O, then solid state disks could provide a dramatic performance improvement. Figure 6 below provides a partial list of common events that indicate solid state disks should be investigated, and the database components that will benefit from solid state disks.

Figure 6: Statspack Events

Event	Description
db file sequential read	The sequential read event is caused by reads of single blocks by the Oracle Database of a table or index. This is generally caused by an index read. The amount of time spent waiting for this event can be greatly reduced by moving the indexes to solid state disks.
db file scattered read	The scattered read event is caused by reads of multiple blocks by the Oracle Database of a table or index. This is generally caused by a full table scan of the data tables. The amount of time spent waiting for this event can be greatly reduced by moving some of the data files to solid state disks.

CPU time	This is the amount of time that the Oracle database spent processing SQL statements, parsing statements, or managing the buffer case. Tuning the SQL statements and procedures, or increasing the server's CPU resources generally best reduce this event. It is an event that is generally not helped by solid state disks.
log file parallel write	This event is caused by waiting for the writes of the redo records to the redo log files. This event can be greatly alleviated by using solid state disks for all copies of the redo logs.
log file sync	This event is caused by waiting for the LGWR to post after a session performs a commit. This can be tuned by reducing the number of commits. Placing the redo logs onto solid state disks can also alleviate this wait.
log file single write	This event is caused by waiting for the writes of the redo records to the redo log files. This event can be greatly alleviated by using solid state disks for some or all copies of the redo logs.
free buffer wait	This wait occurs when a session needs a free buffer and cannot find one. A slow DBWR process that cannot quickly flush dirty blocks from the buffer cache can cause this. Moving the files that are receiving the majority of the writes to solid state disks can help to alleviate the wait event. If poor I/O does not cause this wait write capacity, you can tune your instance by increasing the buffer cache.
control file parallel write	This wait is caused by waiting on writes to the control files. Moving the control files onto solid state disks can help alleviate this wait.
buffer busy waits	The primary cause of these waits is contention for a block that is being used in a non-sharable way (so that a read/write cannot be performed until the process that is using it is complete). Increasing the speed of the disk system by using solid state disks can alleviate this.
direct path read	This wait event is caused by reads that skip the database buffer. If there are lots of sorts and hashes taking place, then this can be caused by slow access to the TEMP space. Moving the TEMP space to solid state disks can reduce this event.
direct path write	This wait event is caused by writes that skip the database buffer. If there are lots of sorts and hashes taking place, then this can be caused by slow access to the TEMP space. Moving the TEMP space to solid state disks can help reduce this event.

Section 6

Components to Move to a Solid State Disk

Once you determine that your system is experiencing I/O subsystem problems the next step is to determine which components of your Oracle database are experiencing the highest I/O and in turn causing I/O wait time. The following database components should be looked at:

Entire Database. There are some databases that should have all of their files moved to solid state disk. These databases tend to have at least one of the following characteristics:

- High concurrent access. Databases that are being hit by a large number of concurrent users should consider storing all of their data on solid state disk since, as we know from the previous section, each user process in Oracle does its own disk reads. This will make sure that storage is not a bottleneck for the application and maximize the utilization of servers and networks. I/O wait time will be minimized and servers and bandwidth will be fully utilized.
- Frequent random accesses to all tables. For some databases, it is impossible to identify a subset of files that are frequently accessed. Many times these databases are effectively large indices.
- Small to medium size databases. Given the fixed costs associated with buying RAID systems, it is often economical to buy a solid state disk to store small to medium sized databases. A RamSan-400, for example, can provide 128GB of database storage for the price of some enterprise RAID systems.
- Large read-intensive databases. Given the fixed costs associated with architecting a RAID system for performance (buying a large cache, buying a lot of spindles for striping), it is economical and much faster to buy a RamSan-500 cached flash solution in order to accelerate large read-intensive databases. A single RamSan-500 can scale to 2TB of capacity.
- Database performance is key to company profitability. There is some subset of databases that help companies make more money, lose less money, or improve customer satisfaction if they process faster. Solid state disks can help make these companies more profitable.

Redo Logs. Redo logs are one of the most important factors in the write performance for Oracle databases. Whenever a database write occurs, Oracle creates a redo entry. Redo logs are used in sequence with the best practice configuration using mirrored redo log groups, a minimum of 2 groups is required. Each redo entry is written to the two or more mirrored redo logs. Oracle strongly encourages the use of mirrored redo logs so that a backup redo log is available in the event of a failure. The operation is considered committed

once the write to the redo logs is complete. Redo logs are used with linear output, if desired, the administrator can also configure redo logs to automatically archive. Archiving makes a copy of a filled log to another location before it can be reused. Archiving can be another source of waits in a slow disk based system.

The redo logs are a source of constant I/O during database operation. It is important that the redo logs are stored on the fastest possible disk. Writing a redo log to a solid state disk is a natural way to improve overall database performance. For additional reliability, it is useful to use host-based mirroring to mirror solid state disk drives.

Indices. An index is a data structure that speeds up access to database records. An index is usually created for each table in a database. These indices are updated whenever records are added and when the identifying data for a record is modified. When a read occurs an index is consulted so that Oracle can quickly get to the correct record. Furthermore, many concurrent users may read any index simultaneously. The activity to the disk drive is characterized by frequent, small, and random transactions. Under these conditions, disk drives are unable to keep up with demand and I/O wait time results.

By storing indices on a solid state disk, performance of the entire application can be increased. For on-line transaction processing (OLTP) systems with a high number of concurrent users this can result in faster database access. Because indices can be recreated from the existing data, they have historically been a common Oracle component to be moved to solid state disk.

Temporary Tablespace. Temporary segments are used to support temporary data during certain Oracle operations. The temporary tablespace segments support complex sort, hash, global temporary table and bitmap index operations. Because temporary tablespaces support many kinds of operations they can quickly become fragmented. In internal tests at Texas Memory Systems, we have found that Oracle database performance degrades quickly as data becomes fragmented.

When complex operations occur they will complete more quickly if the temporary tablespace is moved to solid state disk. Because the I/O to the temporary tablespaces can be frequent, disk drives cannot easily handle them.

Rollback Data. In databases with a high number of concurrent users, the rollback segments (undo tablespace in newer versions) can be a cause of contention. Undo data is created any time an Oracle transaction changes a record. In other words, if a delete command is issued, all of the original data is stored in the undo tablespace until the operation commits. If the transaction is rolled-back, then the data is moved from the undo tablespace back to the table(s) it was removed from.

Because the undo tablespace is hit with every change, operation, it is useful to have the undo tablespace stored on solid state disk. This will provide fast writes

when the update transaction is created and will make undo tablespace available more quickly for the next operation.

Frequently Accessed Tables. It is estimated that only 5%-10% of data stored in OLTP systems is frequently accessed. These tables typically account for a large percentage of all database activity and thus I/O to storage. When a large number of users hit a table, they are likely going after different records and different attributes. As a result, the activity on that table is random. Disk drives are notoriously bad at servicing random requests for data. In fact, the peak performance of a disk drive drops as much as 95% when servicing random transactions. When a table experiences frequent access, transaction queues develop where other transactions are literally waiting on the disk to service the next request. These queues are another sign that the system is experiencing I/O wait time.

It makes sense to move the frequently accessed tables to solid state disk. SSD performance is not impacted if performance is random. Additionally, solid state disks by definition have faster access times than disk drives. Therefore, application performance can be improved up to 25x if frequently accessed tables are moved to SSD.

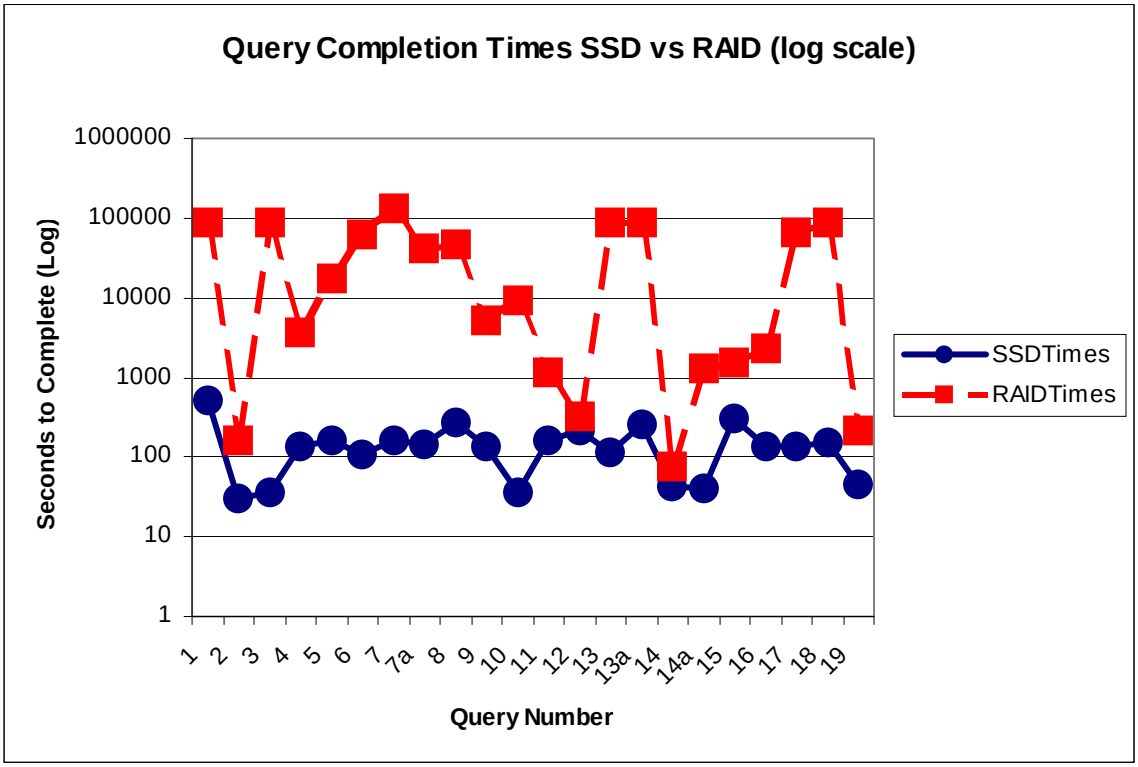
Section 7

Burleson Consulting Oracle Benchmarks

Mike Ault of Burleson Consulting completed a series of tests involving Oracle 9i running on two Linux servers each with two 1.88 GHz processors. A set of tests compared a SATA Disk Array versus a RamSan-320 solid state disk. The testing was done using Oracle version 9.2.0.4 following TPC Benchmark™ H.

The test results were staggering and are displayed in Figure 7 below (note the results are presented in a logarithmic scale).

Figure 7: Query Run Times SSD vs. RAID



These results demonstrate the possible impact of a solid state disk running Oracle. The total length of time it took the SSD to finish running all 7 tests was approximately 3 days. The total length of time it took the SATA disk array to run the tests was 58 days. Additionally, any query that took longer than 30 hours to run was terminated. There were 6 queries terminated on the SATA array and no runs terminated on SSD. Mike Ault summarized the benchmark data as follows*:

- "In the SSD verses ATA benchmark the gains for insert and update processing as shown in the database loading and index build scenarios was a respectable 30%. This 30% was due to the CPU overhead involved in the insert and update activities. If the Oracle level processing for insert and update activities could be optimized for SSD, significant performance gains might be realized during these activities.
- "The most significant performance gain comes in the use of SSD in query based transaction loads. The performance gains for using SSD can be quite spectacular; factors of 176 times better performance of standard disk technologies are documented in the report.
- "Even when only data files can be placed on SSD assets, the performance gains are phenomenal as also shown in the benchmarks. "

*The complete details of this test are outlined in the white paper: Comparison of Solid State disks to ATA Disks with Oracle9i, by Mike Ault with Burleson Consulting. This article and additional information about Oracle with solid state disks can be found at: http://www.superssd.com/oracleperformance.htm.

Section 8

Statspack Analyzer

Oracle applications are at the core of the modern enterprise. When database performance suffers, the enterprise suffers; the database administrator is on the front-line solving the problem. StatspackAnalyzer.com provides the DBA with an easy-to-access, easy-to-use tool that provides actionable performance tuning advice in seconds.

StatspackAnalyzer.com is a free tool for Oracle professionals. After registration for the site, Oracle DBAs can upload a text report from their Oracle Statspack or Automated Workload Repository (AWR) reports using a simple web-based interface. Every Oracle instance from 8.1.7.2 on up has the ability to produce Statspack reports.

Oracle Statspack reports are loaded with large quantities of data. StatspackAnalyzer.com sifts through the data, applies rules created by a panel of Oracle experts, and outputs easy to implement recommendations for improving Oracle performance. StatspackAnalyzer.com performs the following actions in order for you to get the most out of your statspack report:

- Organizes and cross-correlates data
- Provides in-depth I/O review and suggestions
- Analyzes Oracle parameters to ensure proper load alignment
- Detects missing “Best Practices” settings
- Evaluates solid state disk fit for your Oracle application

SSD and Your StatspackAnalyzer.com Report

If your Oracle database has an I/O bottleneck, StatspackAnalyzer will point out whether the use of solid-state disk technology, such as the RamSan product line from Texas Memory Systems, is a valid hardware option for improving performance.

SSD, non-volatile storage systems that store data in DDR memory, are a cost effective solution to a number of common enterprise database performance problems and I/O bottlenecks. Enterprises deploying SSD solutions have seen numerous benefits by making better use of their existing servers, adding concurrent users, shortening batch cycles, reducing response time delays, and improving user satisfaction. These systems have become increasingly sophisticated, higher performing, and lower cost.

"Burleson Consulting, in conjunction with Texas Memory Systems, has taken on the challenge of building an expert system for intelligent analysis of Statspack reports. The Web site (www.StatspackAnalyzer.com), is the result of hundreds of hours of work by a consortium of Oracle performance tuning experts, all working to create valid decision rules that might be applied to any time-series Oracle report.

"As the StatspackAnalyzer effort gains momentum, the team of experts will continuously refine and expand the decision rules, working towards a software solution that will mimic the analysis of a human expert. "Obviously, StatspackAnalyzer will never be able to model the intuition of an expert because it's impossible to quantify those 'I have a feeling" hunches that distinguish the real experts, but this tool shows great promise as a research tool.'"

From Don Burleson, Oracle Expert

Section 9

For More Information

TMS specialists are available to provide feedback on what the RamSan solid state disk product line can do in any particular application or environment. Call the main office in Houston, Texas at 713-266-3200 or do one of the following:

-For more in-depth information, visit www.superSSD.com
-Existing customers contact support@superSSD.com.
-Potential customers contact sales@superSSD.com.

Also consider these other articles and white papers, available online or through your TMS representative:

-What the Tera-RamSan can do for Business
-Improving Application Performance with Solid State Disk
-Understanding IOPS
-The Storage Performance Dilemma
-And many more…ask your TMS rep!