Using ASM Preferred Read Groups in Oracle to Maximize Performance

Michael R. Ault, Oracle Guru, TMS Inc.

Introduction

In today's 24X7 fast paced world your Oracle database system is a lynch pin that holds the fabric of your company together. The Oracle system must perform well, be reliable and be resilient. But what do these requirements entail?

Performance – Queries, reports and screens must return in a reasonable period of time to the requestor. What is reasonable is negotiable in some cases, in others your feet are held to the fires of various service level agreements (SLA).

Reliability – A single system fault should not be able to bring the system down. In many industries loss of even a few minutes of processing impacts the bottom line.

Cost – A solution should deliver performance and reliability at a reasonable cost.

In this paper we will show a new Oracle system architecture that fulfills all of the above requirements. By leveraging the available features of Oracle itself, and new technologies available from TMS, a high-performance, reliable and cost effective system can easily be constructed.

Let's look at the different aspects of performance, reliability and cost and examine what Oracle and TMS technologies should be used to fulfill their requirements.

Performance

Most modern servers provide adequate CPU and memory resources. Rapid technological advances governed by Moore's law have allowed more memory and more powerful processors to be economically available every year. In most of the performance related issues we have come up against the issues that most affect the performance of a database are related to the IO subsystem. Figure 1 shows how CPU speeds have increased while disk access times have not kept up.

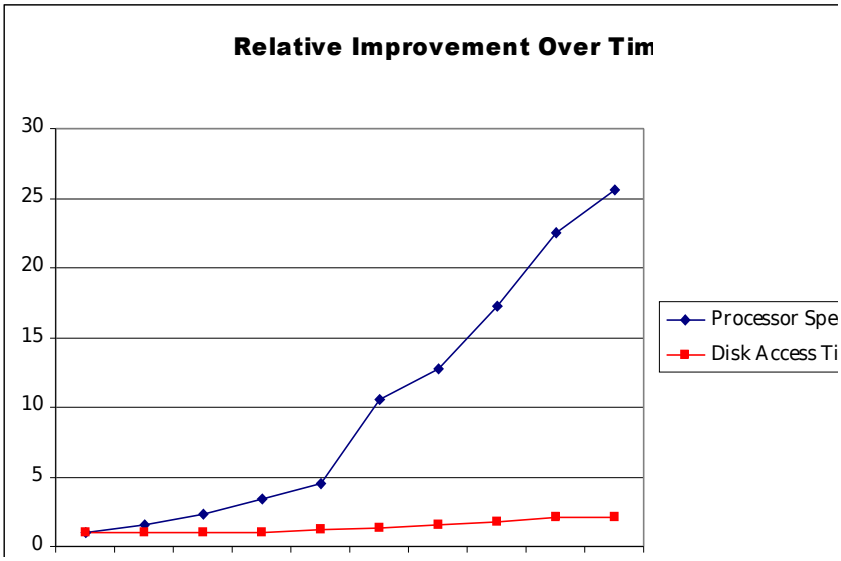

Figure 1: Relative CPU and Disk Performance

IO is the Biggest Issue

For many years the IO subsystem for any computer has been its weakest link, requiring many complex techniques to be used to squeeze the last bit of performance from disk drives. Disk drives are generally limited to 15,000 rotations per minute (RPM) based on power, motor, and physics limitations. Since the main component of a disk drive's latency is its rotational latency this limits disk drive raw response performance to between 2-5 milliseconds. Figure 2 shows the disks and head assembly from a 10K RPM, 74 GB large form factor drive.

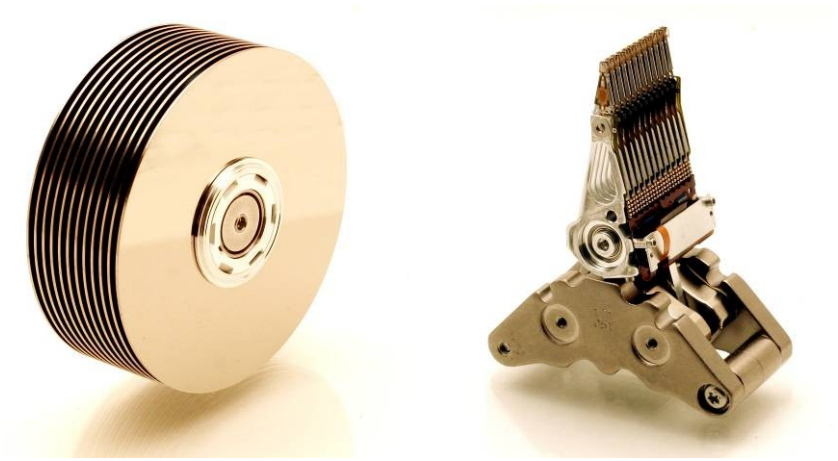

Figure 1: Disks and Head Assembly

How Do Disk Systems Compensate?

The only way to boost the IO capability of disk based systems is to increase the number of active disks. Each disk can sustain a maximum of 200 random IO per second (IOPS.) So if your system needs to perform 10,000 IOPS to maintain performance you will need at least 50 disk drives just to maintain a 2-5 millisecond response time. If you have a large number of simultaneous users or do many full table scans or multi-block accesses then this number of needed drives increases dramatically. In the TPC-H data warehouse benchmark for example, the number of drives needed to get adequate performance may result in the purchase of 30-40 times the needed storage capacity simply to get the needed IOPS to support the needed IO capacity.

In many cases we see that 10-20 percent of the database is accounting for 90 percent of the IOPS. These IOPS intensive data tables and indexes "poison" the IO subsystem resulting in slow performance for the entire application.

Solid State Disks

With the performance gap between processors and hard drive-based storage systems widening, solid state storage is entering the limelight. Because solid state systems rely on memory chips for data storage, they offer unprecedented access times and tremendously high IOPS rates, which narrow the gap between the processor speeds and storage speeds.

Solid state disks are a proven technology – they have been designed and manufactured for over 31 years. 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 results in storage speeds far greater than is even theoretically possible with conventional, magnetic storage devices. To fully make use of this speed, SSDs connect to servers or networks through multiple high-speed channels. A 5 terabyte SSD is shown in Figure 3.

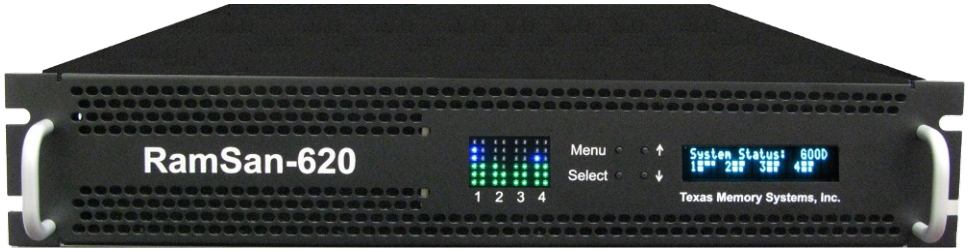

Figure 2: The 5 Terabyte SSD Solid State Disk

Reliability

Reliability means that no single component failure will result in loss of the system. Usually we meet reliability by means of redundant components: multiple CPUS, multiple power supplies, multiple fans, multiple HBAs, and redundant storage. With Oracle, Oracle Real Application Clusters (RAC) can be deployed with two or more servers to ensure a database doesn't go down in the event of a total server failure. Of course HVAC and power must also be redundant, but those are beyond the scope of this paper.

A common approach to ensuring reliability in the storage has been deploying SAN systems with redundant storage "heads". Each head has connections to the servers and to the storage, and the ability to create a RAID layout across the disks. This approach is effective with lots of storage that is used to support many applications, but it is less effective when you are focusing on a single important application such as a large Oracle database. This is because the storage heads tend to be very expensive and add latency to every IO access - a bad proposition when the storage is SSD.

Oracle Automatic Storage Management

Oracle introduced Automatic Storage Management (ASM) in 10g as a compelling option to allow Oracle to provide storage redundancy and eliminate the expense of redundant storage heads. ASM provides striping and mirroring capabilities across multiple storage devices. This allows any storage component to fail without data loss and full redundancy can quickly be restored through automatic rebalancing. A new feature was added in Oracle 11g ASM that enables extremely cost effective SSD usage. The Preferred Mirror Read option enables a mirror to be created between storage devices with different performance characteristics. Writes are sent to both storage devices to ensure redundancy is maintained, but reads are only serviced by the storage that is specified as preferred. This enables SSDs to be mirrored with traditional storage when the read IO performance is the primary concern.

Cost

Solid State disks have an interested combination of characteristics: they are expensive for capacity but they are extremely inexpensive for performance. To leverage SSDs effectively the capacity used must be limited to just what is necessary. In the architecture laid out in this document SSDs are used in combination with disks in order to control the cost. This is done without sacrificing performance or reliability, all data is mirrored to a separate physical enclosure, and every time Oracle must wait on an IO request it will be delivered at SSD speed.

Other Performance Limiters in Oracle

Generally speaking Oracle is read latency *sensitive* and write latency *insensitive* when it comes to data. If you delay data reads Oracle performance will suffer. In the case of data writes Oracle uses a lazy-write methodology known as "delayed block clean-out." Delayed block clean-out simply means than Oracle only writes out a data block back to the IO subsystem when the block is needed by another process for a different bit of data. Thus, a block may reside in the cache for several seconds before it is actually written back to disk after an Oracle process is finished with it.

However, there are three types of writes that need to be accomplished quickly, these are:

- Redo block writes
- Undo block writes
- Temporary block writes

Each of the above writes can result in a transaction waiting for the write to complete before it continues. High latency writes on redo, undo and temporary blocks cannot be tolerated in a busy system. In cases where there as a large number of commits per second, redo writes can be a great concern. In situations where there may be a large number of rollbacks, undo blocks become a read and write concern and any temporary IO that is excessive is always a concern for any database.

Redo and Undo are usually sequential writes and whenever possible should be isolated from other forms of IO. Temporary tablespace IO, even in system with zero disk sorts, can be a major contributor to the IO picture. Remember that temporary IO is used not just for sorts, but also for hash joins, bitmap operations and temporary table operations that exceed PGA limits. We have seen temporary IO be the major source of IO in a system even when there were zero sorts going to disk.

Optimized Architecture for ASM PRG

Given these concerns: performance, reliability and resiliency, how can we create a single system that will fulfill all of the requirements in each area and still perform? The only way is to combine the best features of Oracle in an architecture where disk and SSD are used in harmony.

Components of Architecture

What do we need to use with this ultimate system?

Hardware

1. SSD (prefer highest speed and capacity available.
2. Disks – Enterprise 10K RPM SFF SAS drives
3. Servers – Lots of memory to hold the most active dataset and dirty data buffers.

Oracle 11g Features:

1. ASM – Automatic Storage Management
2. RAC – Real Application Clusters

Server Redundancy

At the server level, the system will have 2 or more multi-CPU servers configured with plenty of memory to serve hold the frequently reused database segments and dirty data buffers. The servers utilize low latency, high capacity interconnections and Oracle RAC is utilized between the servers to handle server failure. At a minimum 2 servers are required, however, for maximum redundancy with no loss of capacity in the event of a single server failure N+1 servers are required where N is the number of servers needed for performance and scalability.

IO Subsystem Tiers

In addition to Oracle features, we need to utilize a multi-tier IO subsystem. Figure 4 shows the storage pyramid indicating the various tiers.

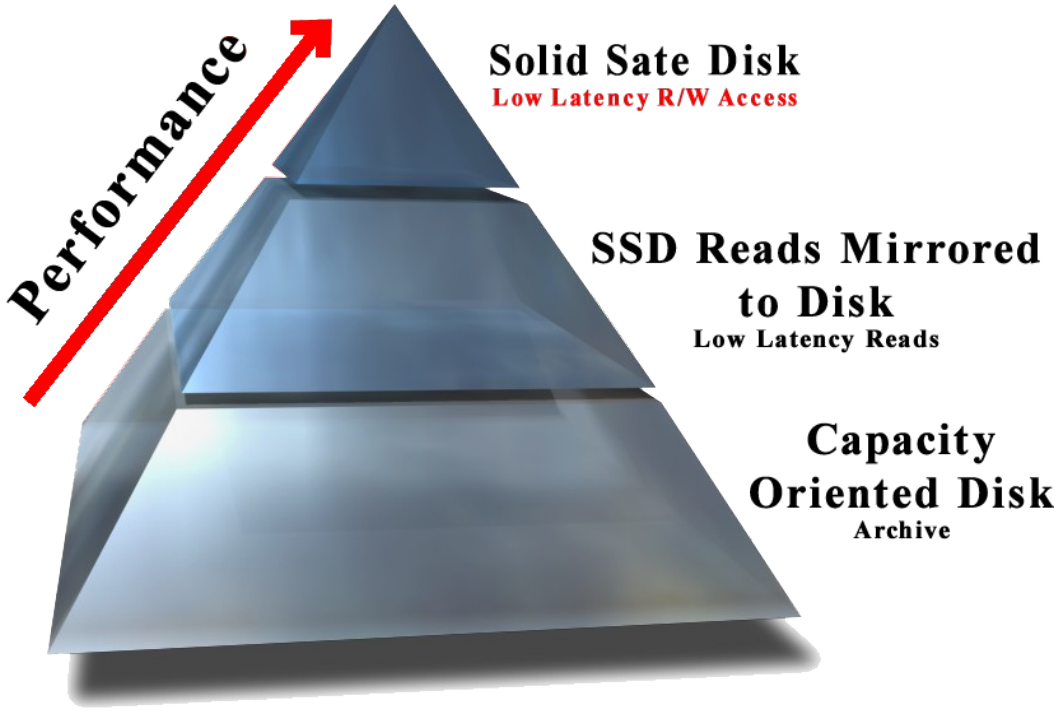

Figure 3: Tiered Storage Pyramid

In order to meet our performance criteria, our write sensitive areas: redo, undo and temp, need to be on low-latency high bandwidth storage. The best response time for writes is provided by battery backed RAM. This is provided by the SSD, the best configuration uses RAM buffers with super capacitor backing to provide 80 microsecond write times. The bulk Flash storage enables the draining of the RAM buffers at up to 250,000 writes/s and ensures that the read performance of Tier 0 is excellent. To ensure there is no single point of failure, this Tier is mirrored across two separate SSD chassis. The capacity of this tier is limited as the circular logs in Oracle require performance but not a tremendous amount of capacity.

The next tier, tier 1, is for the database tables and indexes. This tier leverages the low latency read performance of the SSD (0.25 ms), and the cost effectiveness of disks for capacity and limited IO/s requirements. In this tier the SSD's are placed into an ASM *preferred read mirror* with traditional disks. In a preferred read mirror, the non-preferred member of the mirror (the disks) receives only writes. This tier only contains write-insensitive files so the latency of the writes to this tier does not matter. However, this does not mean that the write *throughput* of this tier isn't important. This tier must be capable of emptying the Oracle's cache at the same rate that it is being filled by the application. This means that the write IOPS and Bandwidth of this tier must be respectable. This requires the number disks in this tier to be sized for aggregate write IOPS, not just to meet the capacity required for a mirror of the SSDs.

The last tier, Tier 2, consists of storage for backup and is purely disk based. This tier is made from the extra capacity left over from the disks used in tier 1. The extra capacity is aggregated and mirrored with ASM. Since the performance of the disks used by Tier 1 when the database is under heavy load, this tier is used to store the nightly backup of the database. Disks are partitioned such that the high performance areas are used by the PRG mirror.

Let's See It

Figure 5 shows a diagram of this architecture.

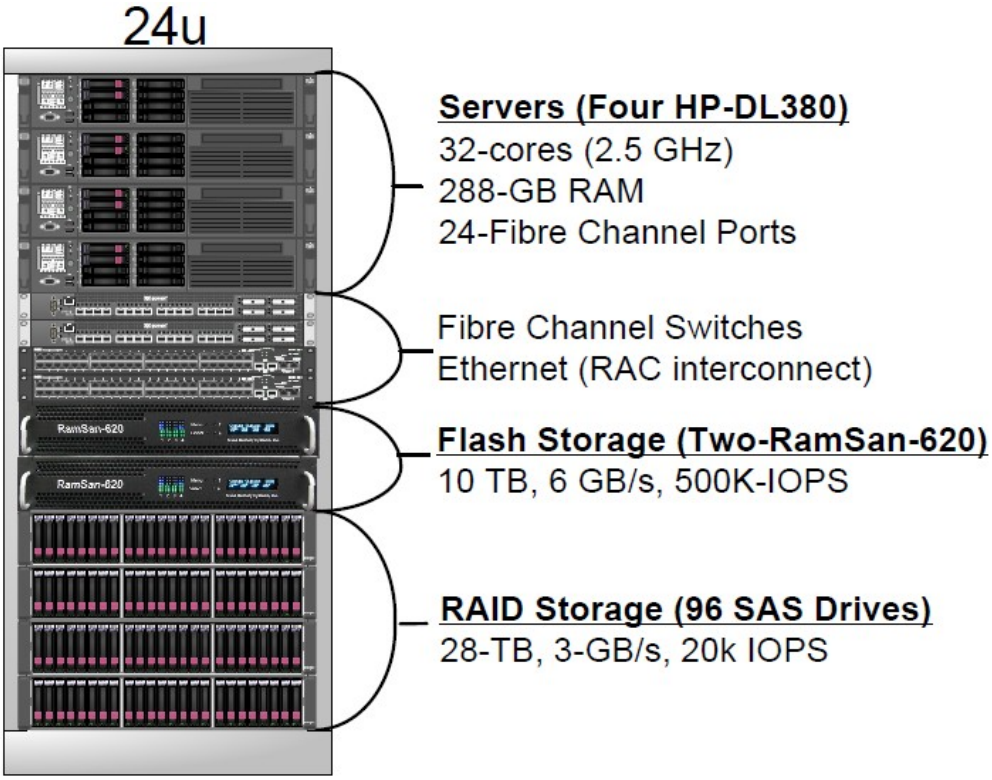

Figure 4: Architecture

Architecture - In Depth

As you can see in Figure 5 the architecture is configured with 4 servers. This provides the database with 32 CPU cores for processing and 288 GB of memory for the frequently reused data and dirty data buffer waiting to be written out. With this much memory available to the database **every** blocking IO to disk needs to be handled at SSD speed.

In our SAN setup we see that two 5-terabyte SSDs and 96-300 GB disks are provided. Each SSD provides up to 5 TB of capacity (10 TB total), 250,000 IOPS (500,000 IOPS total) at a latency of 0.08 ms for writes and 0.25 milliseconds for reads. Each SSD's 5 TB capacity is divided into a 500 GB section and a 4.5 TB section used for Tier 0 and Tier 1. In Tier 0 the two 500 GB sections are mirrored with ASM and used for redo, undo, and Temp. The remaining spaces is aggregated with ASM and are mirrored to disk through Oracle 11g ASM's preferred mirror read technology to provide 9 TB of fully redundant capacity. Thus there is no disk IO time penalty for reads and writes are buffered by the larger RAM buffer cache and written out to the RamSan/Disk mirror by the lazy writer process. The disks are 96 enterprise SAS drives capable of providing 20,000 IOPS at 5-10 ms latency. Each disk is divided into a small 93 GB partition and a large 206 GB partition. The 96-93 GB partitions are grouped with ASM and used for the disk half of the tier 1 mirror (9 TB). The larger 206 GB partitions are grouped then mirrored providing 9.9 TB to hold a full backup of the database.

This Optimized ASM-PRM architecture allows **every** blocking IO to be served from low latency SSD while disks provide redundancy and backup to keep costs reasonable.

Performance

During operation the database now serves reads from high speed, low latency SSD via the preferred read mirror. Writes to logs and other write sensitive Oracle structures are to the dedicated SSD mirror while database writes (normally not a perfor5mance sensitive item) write to both SSD and disk. In some

architectures where reads and writes can be buffered in caches at the HDD array level, turning on write caching can reduce or eliminate writes as a source of wait activity. Note that turning on write caching is only suggested when the primary mirror is on persistent SSD technology.

Some Actual numbers

Using tables from a TPC-H setup and a 3 terabyte TPC-H configuration the performance differences between using disk and SSD can be easily shown. Figure 6 shows the tables used for the test.

- Three Tables
  - Part (600m rows)
  - Supplier(30m rows)
  - PartSupp (2.4b rows)
- Three Indexes
  - (partkey, suppkey, partkey+suppkey)

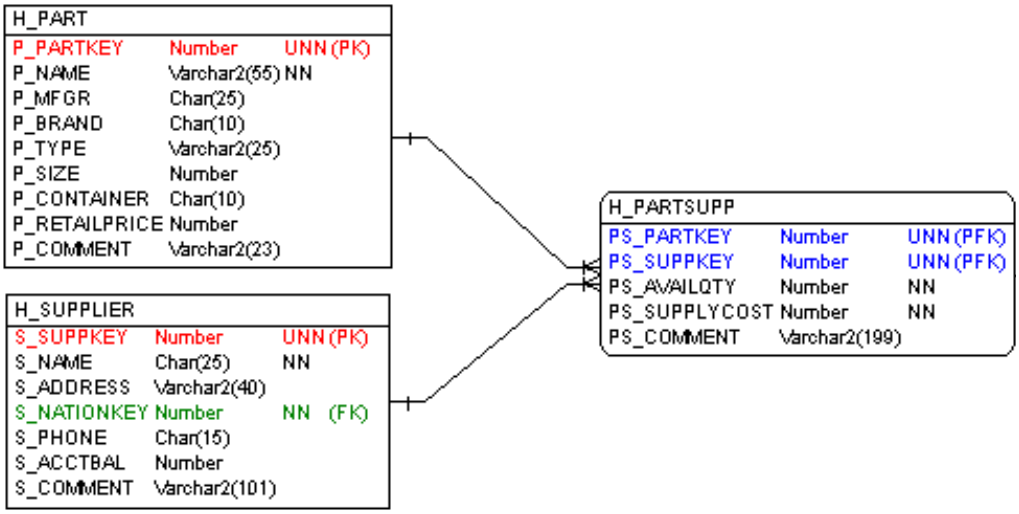

Figure 6: Schema of Test Tables

Using an indexed, optimized query (shown in Figure 7) we generate 7 IOPS per query execution (on the average.) Now, 7 IOPS wouldn't really test the performance of any system, but, we utilize PL/SQL to generate this query with random key values in multiple sessions so that we actually execute the query 2,000,000 times for a total of 14,000,000 IOPS.

The query finds the total amount owed to all suppliers for a particular part.

```
select sum(s_acctbal) into sum_s_acctbal
    from
 supplier
 where
 s_suppkey
 in (
     select
     ps_suppkey
     from partsupp
     where ps_partkey =  (x)
 );
```

Figure 7: Test Query

- From each server (4 total), 50 simulated users run a stored procedure 10 times that submits this query 1000 times

- 4*50*10*1000 = 2,000,000 Queries
- Test with disks or SSD set to preferred

```
SQL> alter system set ASM_PREFERRED_READ_FAILURE_GROUPS = 'HYBRID.SSD';

System altered.

SQL> alter system set ASM_PREFERRED_READ_FAILURE_GROUPS = 'HYBRID.DISK';

System altered.
```

When the test is run against the architecture with PRG set to HYBRID.DISK we see the following results:

- ~4000 IOPS per RAC node
  - 16,000 IOPS total
- 12.25 minutes to complete with 4 nodes running (2m queries).

```
[oracle@opera1 ~]$ time ./spawn_50.sh

real    12m15.434s
user    0m5.464s
sys     0m4.031s
```

When the test is run against the architecture with PRG set to HYBRID.SSD we see the following results:

- 40,000 IOPS per RAC node
  - 160,000 total in this test
- 1.3 minutes to complete with 4 nodes running (2m queries).

```
[oracle@opera1 ~]$ time ./spawn_50.sh

real    1m19.838s
user    0m4.439s
sys     0m3.215s
```

Figure 8 shows the top five wait events from AWR for both the HDD and SDD PRG runs.

HDD Results

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
db file sequential read	257,293	3,293	13	82.54	User I/O
db file parallel read	30,915	567	18	14.22	User I/O
DB CPU		75		1.88	
gc cr grant 2-way	199,215	36	0	0.91	Cluster
reliable message	346	10	28	0.24	Other

SSD Results

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
gc cr grant 2-way	1,703,359	1,344	1	35.93	Cluster
db file sequential read	2,250,261	1,253	1	33.51	User I/O
DB CPU		637		17.02	
gc cr multi block request	367,691	356	1	9.52	Cluster
db file parallel read	276,130	111	0	2.96	User I/O

Figure 8: AWR Top Five Waits

Figure 9 shows the difference in total IOPS for the same time duration for both HDD and SSD during the test runs.

HDD Test Run

Tablespace IO Stats

- ordered by IOs (Reads + Writes) des

Tablespace	Reads	Av Reads/s
TS_S	131,487	1,677
TS_I_LORDERKEY	124,720	1,590
TS_PS	58,061	740
SYSAUX	3,761	48
UNDOTBS3	178	2
DISKS_TEMP	38	0
SYSTEM	68	1

SSD Test Run

Tablespace IO Stats

- ordered by IOs (Reads + Writes) desc

Tablespace	Reads	Av Reads/s
TS_S	1,161,958	15,562
TS_I_LORDERKEY	1,117,768	14,970
TS_PS	520,385	6,969
SYSAUX	2,448	33
UNDOTBS3	713	10
SYSTEM	296	4
DISKS_TEMP	41	1
UNDOTBS1	3	0

Figure 9: AWR Tablespace IO Statistics

Summary

Let's summarize the results of incorporating the ASM PRG architecture:

- ALL blocking IO is handled by the SSD
    - *>10* times *faster performance* than HDDs!
- Disks provide redundancy in order to keep costs reasonable.
- No sacrificing redundancy
- Allows reuse of legacy hardware

In this paper we have shown the needed components of a performance oriented, reliable and resilient ASM-PRG Oracle system. We delineated the limitations of 100% disk based architectures and showed how addition of solid state technology can dramatically improve a system. Finally we showed the physical layout for the architecture.