EMC²
where information lives®

Oracle Performance Hit
A SAN Analysis

EMC Proven Professional Knowledge Sharing 2009

Kofi Ampofo Boadi,
Sr. Solutions Architect | SAN
Kofiboadi20@yahoo.com

EMC² PROVEN PROFESSIONAL

Oracle Performance Hit | A SAN Analysis

Kofi Ampofo Boadi
Sr. Solutions Architect | SAN
Kofiboadi20@yahoo.com

Table of Contents

Introduction

The performances of applications rely heavily on the SAN. Performance issues can be centric to the Host, connectivity device, storage array, or any combination of the above. This article will elaborate the effect a Storage Area Network can have on Oracle's performance. It will emphasize the CLARiiON storage array; the Symmetrix will be touched briefly. Note that the concepts and analysis in the article can be applied to all storage arrays, irrespective of vendor types.

SAN related performance problems with Oracle can be avoided or minimized with the right disk-layout design. This is true! Specifying the right RAID type configurations for each unique Oracle component is crucial. The single reason for this article is to layout some of the troubleshooting steps and best practices that can be taken to resolve SAN related problems with Oracle. That is to say , if you follow all the best practices during the initial setup and architecture, and abruptly encounter performance issues with Oracle, your DBA will advise it's not caused by them and all fingers will point to the SAN. What do you do? How do you identify the core of the problem? How do you approach situations of this nature as a Storage Solutions Architect or engineer?

This article hopes to clarify and resolve these kinds of questions. The article is laid out strategically to gradually lead you from no-knowledge in Oracle storage design to an apt state of Oracle-SAN design and implementation. It also offers a couple of case studies that specifically highlight the core of this topic. Note that these are real-life cases and their resolution-to-fix.

We will explain the respective components of Oracle in detail and illustrate why a poor SAN configuration will have a tremendous performance hit.

Oracle Components

I/O flow can either be a Read or a write. Thus, data on drives can either be read from or written to. The behavior of IOs leads to the further classification of sequential read/write, or on the contrary a random read/write.

Sequential Reads are often found in cases of recovery. The process of recovering from a backup LUN, for instance, is very heavy sequential reads. For mpts of this nature a RAID3 or RAID5 configuration is often optimal.

Sequential writes, on the other hand, are often found in backup drives. Simply, backing up of data is sequentially written to drives. RAID5 is optimal for sequential writes especially if the IO sizes are large. RAID10 can also do the job but RAID5 is better for sequential writes.

Random Reads are exactly how they sound, data that is randomly read. There is no defined order. RAID5 is optimal; you can also use RAID30. *Random Writes* are often the nature and behavior of data tables. Updates are random. RAID10 and RAID5 are optimal depending on the application.

With the above in mind, let's take a look at the architecture of an Oracle database. An Oracle database is made up of the following components: hold on, rather than taking the conventional approach of stating the component and defining it which is *very boring,* we will approach this in an essay format:

Data tables are where the data and information is saved and accessed. They can be either partitioned or non-partitioned. The INDEX is a record of where what data is on the data tables. The Undo tablespace is used in Oracle to store the before images of the data blocks being changed so Oracle can provide 'read consistent' views of the data before a commit is performed. The 'old' data blocks are held in UNDO until a commit or Rollback is performed, and all currently running SELECTS have completed processing. For example, you want to run a SELECT that joins 3 tables, a, b, and c. You start

executing your SELECT and it reads lots of data blocks and takes a while to complete. I run an UPDATE on table b, which changes 50% of the data blocks. The changed blocks from table b get stored in UNDO so your select can continue to process them as they existed before they were updated; the same way they looked when you began your SELECT.

Now, I commit the data changes to table b. Normally, this frees up the blocks in UNDO for another update/insert process. However, your **select** is still running and takes another couple of minutes to complete. Now the blocks in UNDO are freed as long as there are no other selects processing table b, which started before I committed the changes. In this way, Oracle does not lockout SELECT while the underlying tables are being changed.

As you can see, the write and read I/Os to UNDO should be as quick as possible to avoid waits for both the data reader(s) and 'changers'.

TEMP is primarily used to provide a sort space on disk if the sort is too large to be performed in memory. It also is used in parallel processing to store data blocks processed by a slave process until the master process requests the blocks as well as several other time sensitive functions.

The REDO Logs are essentially transaction logs. Since it will take a longer time to write a full update to its data tables, Oracle writes a shorthand to the Redo logs. Thus, if Oracle needs to update its tables with "Oh My God" it will rather write "OMG" as a shorthand to Redo logs to save time. Once a commit is issued, Oracle will write a redo entry for the COMMIT to the redo buffer cache and the cache must be written to the physical redo log group to complete the commit processing. Redo data from multiple sessions are streamed together and they do not have to be in any particular session order to be placed in the redo buffer or written to the physical log. Large blocks of uncommitted redo may be written to the physical redo log, as in the case of long running inserts or updates that exceed the capacity of the redo buffer. The changed data and index blocks in the database cache may not be externalized or written to disk immediately. They are 'aged-out", and written at checkpoint processing, log switches, or several other triggering events. The main points is that once the data and 'commit' redo

entry is written to the physical redo it is considered committed; all changes are being written as they occur. The write I/O from the redo buffer to the physical redo logs must be as fast as possible, otherwise all changes being processed will wait with performance slowdowns in the entire database. This has changed slightly in 10g r2 with the introduction of Asynchronous COMMIT.

Redo logs are designed to be "dependent" on Archive logs for data consistency. Oracle requires the first Shorthand "OMG" to be written before it will write another shorthand. The redo and archive log dependency is when a physical redo log is full it must be written to the Archive log directory before it can be used again. (This is a Log Switch event) Physical redo logs can be duplexed for safety and used in a round robin manner. Once a log switch event starts, the next redo log group is written to and the DB-Archiver process copies the prior redo log to the archive directory as an archive log. After the copy operation is complete the redo log group is available for reuse. It is important that the I/) to the archive log directory and control file(s) is fast, as sessions should not wait on log switch events.

REDO, UNDO, and TEMP are all accessed directly by the Oracle kernel processing DB requests and must be on fast LUNs to perform efficiently. They are by far the most critical objects in the I/O scheme. Note that data and index files can handle slower I/O rates without affecting the overall system performance.

REDO LOG Groups

Creating Redo log groups can be important to the performance of your application. You can implement multiplexing where a couple redo logs can be paired for protection. You can also implement log switching to switch between the respective groups for better performance. We will talk more about this implementation later, and illustrate in detail each component’s behavior on the SAN and which Disk layout best fits for optimal performance.

Cache Effect

It is important to note that CLARiiONs are a low-end storage array and as such have limited performance-tuning capabilities. It has a non-scalable cache, the CX3-80 for instance has an 8G cache of which by default, and only about 3.2G is set for write cache and the difference for reads. Note that this much is global to the entire CX3-80.

Another CX performance limitation is that LUNs don't move as is in the case of a Symmetrix with *Symmetrix Optimizer*. If a heavily utilized LUN x sits at a spot on a drive close to the center and a less utilized LUN y, close to the edge, it will take x a longer time to be read from or written to, because it has a longer distance from the spindle. This results in high seek times and response times. The Symmetrix implements the DRV volume to calculate contention between LUNs and swap them accordingly. Off- course cache is scalable on the Symmetrix.

Write cache can be disabled or a write aside value set for the archive log LUN to free up cache space for the other components that need them. We will elaborate on this later in the case study.

Because of all the above, it is very crucial to layout the drives for your RAID groups intelligently, which RAID configuration to use for what component and the *combinations* i.e. if RAID5 (4+1) or (8+1) combination, since each results in a different cumulative stripe size. In above, the former will have a stripe size of 256K (4x64K) and the latter will have 512K (8x64K).

Host-Level Analysis

This usually is the servers' CPU utilization, iowaits, queue depth/length etc.
Analyze questions such as what does high (~100%) CPU utilization mean and what effect does it have on performance? What does low CPU utilization mean and what effect does it have on performance? Analysis of Oracle structure itself is important. This section is elaborated in the case study.

Switch level Analysis

I recommend that you watch your switch port speed and the HBA speed to determine if they need to be upgraded. You may need to upgrade the switch firmware in some cases. We will talk more about this in the below case study.

The Case study!

A real life Oracle performance hit on a CLARiiON array, what the cause was, and how it was rectified. This case study details most of the issues Administrators/Engineers and Architects experience and the suggested best workable resolutions. Please note that the order of the analysis is irrelevant. It is the content of the analysis that matters.

Kejetia Solutions Systems is a global *financial* solutions company that reported a major performance problem with their ETL application. This application's database is pwodw2. This is an Oracle database and runs off of a Sun Solaris server called **drfuorap2**. The server sees storage from a CX3-80. The DBAs report very high iowait times. Customers are complaining heavily about the slow response.

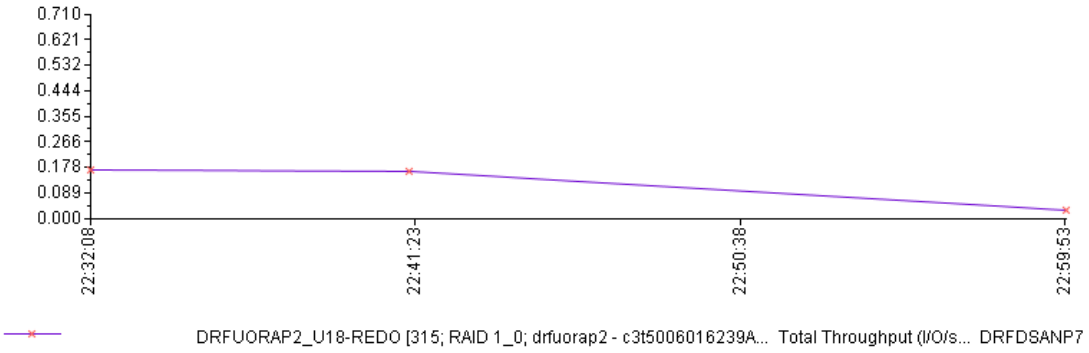
You are a new Solutions Architect at this company and have inherited the current architecture. Everyone says they haven't done anything wrong but you know certainly someone must have done something for this to go so badly. Your job is to research, find what is wrong, and fix it. Remember this storage array is maxed out in terms of available space and resources, and due to the current financial conditions management is not ready and able to acquire a new storage system. So you are being asked to make it work based on your experience, knowledge and expertise

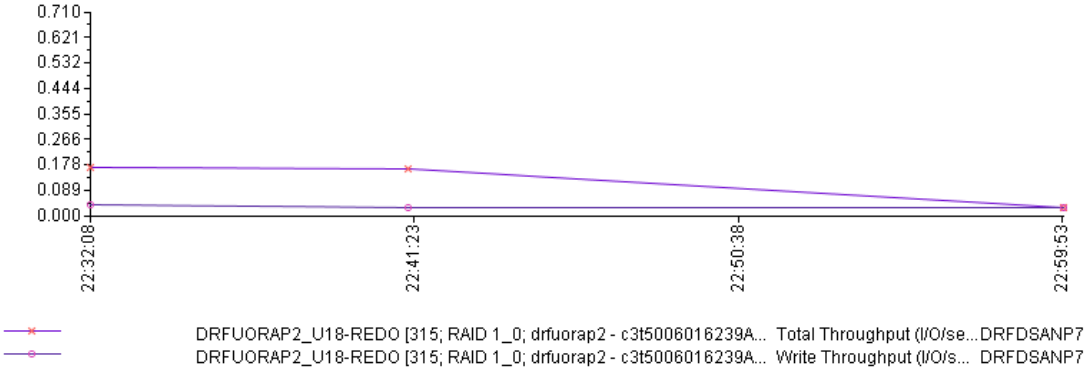
Analysis/Resolution

First, let's start by monitoring the performance of this Oracle server: drfuorap2.
See the performance of CLARiiON cache below, Server CPU and LUNs/mpts during the time when ETL kicks in. It's obvious that the ETL application burst for its first hours and then falls after it is in progress.

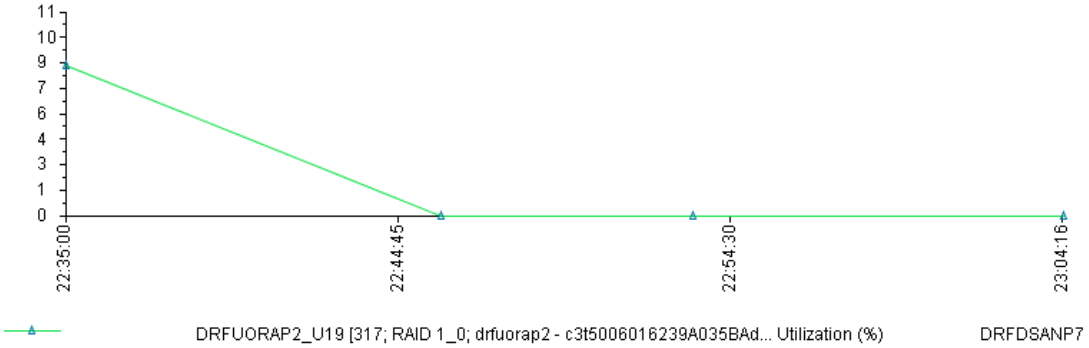
Cache performance peeked quiet frequently at 99%, 96% etc. and stayed at this percentage for a significant period of time. The high percent of dirty pages in cache were frequent and constant; cache coalescing and flush to disk for this application was not smooth.

Please note the graphs below and on the next page.





Total Throughput against Write throughput



Server Performance Statistics.

load averages: 1.30, 1.09, 1.08 ***drfuorap2*** **22:50:43**
212 processes: 209 sleeping, 1 stopped, 2 on cpu
CPU states: 75.1% idle, 10.8% user, 4.2% kernel, **99% iowait**, 0.0% swap
Memory: 24.0G real, 11.4G free, 10.9G swap in use, 34.4G swap free

PID USERNAME THR PR NCE SIZE RES STATE TIME FLTS CPU COMMAND
15704 oracle 1 60 0 7.9G 5.4G sleep 12:57 29 4.47% oracle
15706 oracle 1 30 0 7.9G 5.4G cpu17 13:52 2 4.21% oracle
15311 oracle 258 59 0 7.9G 5.4G sleep 21:32 0 0.95% oracle
15313 oracle 258 59 0 7.9G 5.4G sleep 21:26 0 0.92% oracle
15317 oracle 258 59 0 7.9G 5.4G sleep 10:44 0 0.45% oracle
15315 oracle 258 59 0 7.9G 5.4G sleep 10:24 0 0.41% oracle
15319 oracle 23 59 0 8.0G 5.4G sleep 6:50 79 0.17% oracle
15353 oracle 16 59 0 7.9G 5.4G sleep 0:33 0 0.07% oracle
18424 root 1 59 0 3160K 2104K cpu03 0:01 0 0.03% top
12668 oracle 1 59 0 7.9G 5.4G sleep 2:17 0 0.03% oracle
17954 root 1 59 0 3264K 2392K sleep 0:02 0 0.02% top
15461 oracle 1 59 0 7.9G 5.4G sleep 0:01 0 0.02% oracle
15382 oracle 1 59 0 7.9G 5.4G sleep 8:24 0 0.01% oracle
15453 oracle 1 59 0 7.9G 5.4G sleep 0:01 0 0.01% oracle
15468 oracle 1 59 0 7.9G 5.4G sleep 0:01 0 0.01% oracle

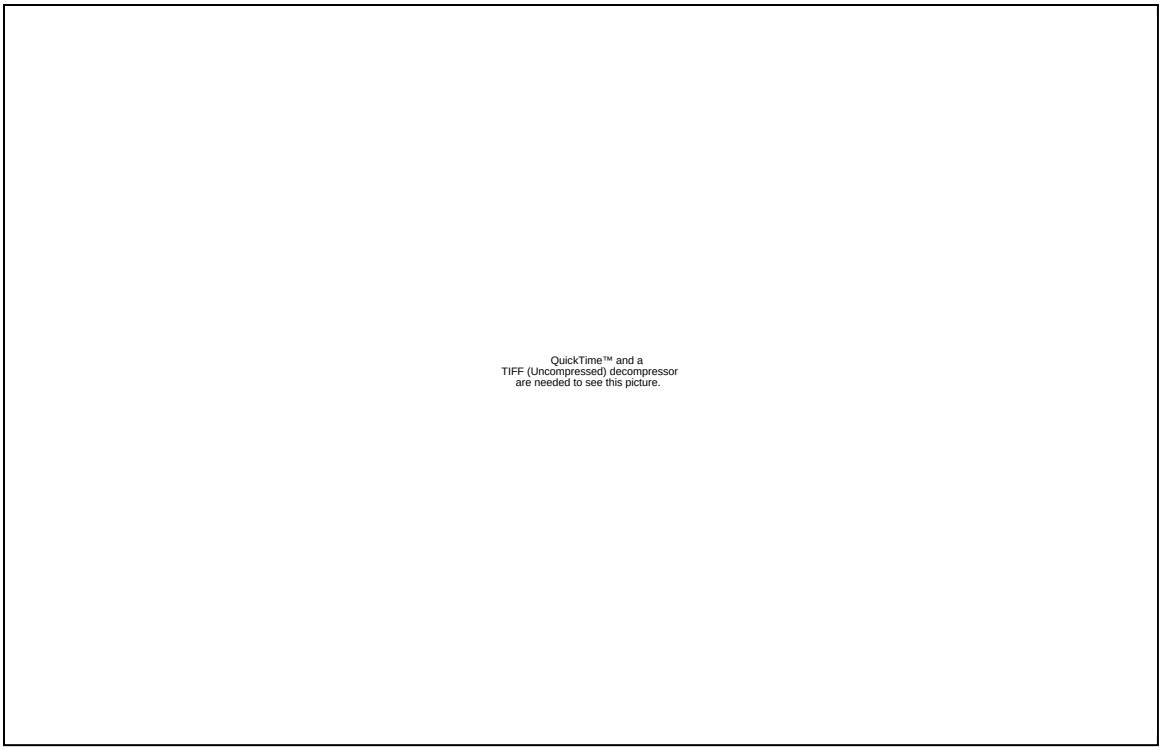
load averages: 1.06, 1.02, 1.06 ***drfuorap2*** **22:48:32**
212 processes: 209 sleeping, 1 stopped, 2 on cpu
CPU states: 89.2% idle, 4.2% user, 2.5% kernel, **90% iowait**, 0.0% swap
Memory: 24.0G real, 11.4G free, 10.9G swap in use, 34.4G swap free

PID USERNAME THR PR NCE SIZE RES STATE TIME FLTS CPU COMMAND
15704 oracle 1 21 0 7.9G 5.4G cpu22 11:42 0 3.60% oracle
15706 oracle 1 60 0 7.9G 5.4G sleep 13:00 0 2.03% oracle
15311 oracle 258 59 0 7.9G 5.4G sleep 21:20 0 0.51% oracle
15313 oracle 258 59 0 7.9G 5.4G sleep 21:15 0 0.49% oracle
15315 oracle 258 59 0 7.9G 5.4G sleep 10:18 0 0.26% oracle
15317 oracle 258 59 0 7.9G 5.4G sleep 10:39 0 0.24% oracle
15319 oracle 23 59 0 8.0G 5.4G sleep 6:47 22 0.10% oracle
15353 oracle 16 59 0 7.9G 5.4G sleep 0:32 0 0.03% oracle
18424 root 1 59 0 3160K 2104K cpu02 0:00 0 0.03% top
17954 root 1 59 0 3264K 2392K sleep 0:02 0 0.02% top
15382 oracle 1 59 0 7.9G 5.4G sleep 8:24 0 0.02% oracle
12668 oracle 1 59 0 7.9G 5.4G sleep 2:16 0 0.02% oracle

Screen Shots of the Cache Percentage values as the ETL Run.

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

Screenshots not appearing on this or previous page



The Cache contention confirms why ETL performs badly whenever we execute changes that are “write” intensive during or near its start time. ETL is a Kill, I will say! From what the above shows, it heavily utilizes almost all of Write cache. Any change around or near its start time will cause drastic competition for cache.

Note that anytime there is a heavy write intense change like Mirroring at host level, etc during and around its time of execution, drfuorap2 does worse than regular and the above explains it.

My advice is to move towards a high-end storage array based on above and the cumulative IOPS the CX is making coupled with how-many servers are heavily utilizing the CX,. This way we can accommodate the performance needs and SLAs expected. But before that let’s take a holistic look at the problem and a proposed cost-efficient solution.

The approach is to simplify drfuorap2's storage space architecture with the hope of reducing the need to stripe many mpts within Oracle while maintaining the most performance. Based on several helpful conversations with Oracle DBAs coupled with lots of Research within Oracle and the SAN determined the cumulative capacity for each Oracle component, and architected the best solution based on *available resource* for this server. I've listed all the issues identified and incorporated their resolutions in the final Re-architecture.

Please jump to page 15 if you're only interested in the proposed resolution, otherwise keep reading to find out what the issue is with –drfuorap2 and why the final architecture stated below is the best resolution.

Striping is good and increases performance, but excessive striping will degrade performance, especially if you have little visibility of how "the mpts" you are striping are laid-out on the SAN. I prefer and encourage minimal Oracle-level striping.

The research included a collection of data from the DBAs on this database and compared it with our storage records (SAN Layout Spreadsheet). Here I saw lots of inconsistency.

Storage requests made for defined components (like DATA or INDX) and architected as such on the *SAN* were in fact been used for entirely different purposes. For example, data components do more Reads and were architected with a RAID5 configuration. I realized that these LUNs are being used as INDX, which requires a different RAID configuration (RAID10). Please see ***figure 1.1.*** All the RAID5s were supposed to have a DATA component but below shows INDX and tables both reside on them. This, I later found, was a DBA's idea of balancing IO load across the different mpts, an interesting idea but with devastating results.

There are, as seen in the below spread sheet, a good number of the mpts that are non-partitioned and have more than one component on them. /U04 - /U14, Ten mpts have both Data components and INDX. This is not good. Oracle Data and INDX have a relationship. These two components are accessed simultaneous. It is true that

sometimes Oracle will complete a request without touching the INDX because it finds it in its buffer; but in the cases where it does not it will need to access from disk. Having both components on the same disk will drastically impact performance. It will not be able to access *two* on the **same disk,** simultaneously. One will have to wait for the other. It is always a good practice to keep these components on different spindles. There are many others besides /U04 - /U14 like /U32 that have three components on that mpt. ***See figure 1.1***

I also pulled up a list of all the mpts on this server. There are just too many mpts - almost 50. This limits available resources for other server's components. Remember that we try to desist from having more than one Oracle component on the same spindle. Having that many mpts leaves the SAN very little options for other servers' components.

An Exact Reflection of Mpts vs Usage on DRFUORAP2 as of Dec 23, 2008						
Mounted on	**Database**	**Usage**			**Size**	**RAID**
/r01	Pwofdw2	partitioned Tables	**Figure 1.0**		30	**RAID10**
/r02	Pwofdw2	partitioned Tables			30	
/r03	Pwofdw2	partitioned Tables			30	
/r04	Pwofdw2	partitioned Tables			30	
/r05	Pwofdw2	partitioned Tables			30	
/r06	Pwofdw2	partitioned Tables			30	
/r07	Pwofdw2	partitioned Tables			30	
/r08	Pwofdw2	partitioned Tables			30	
/r09	Pwofdw2	partitioned Tables			30	
/r10	Pwofdw2	partitioned Tables			30	
/r11	Pwofdw2	partitioned Tables			30	
/r12	Pwofdw2	partitioned Tables			30	
/u01	Pwofdw2	Oracle Home (binaries and log files)			25	**RAID5**
/u02	Pwofdw2	Oracle System Tablespaces, control file			10	**RAID5**
/u03	Pwofdw2	Oracle System Tablespaces			50	**RAID10**
/u04	Pwofdw2	Non Partitioned Tables and Indexes	**Figure 1.1**		110	**RAID5**
/u05	Pwofdw2	Non Partitioned Tables and Indexes, control file			75	**RAID5**
/u06	Pwofdw2	Non Partitioned Tables and Indexes, control file			100	**RAID5**
/u07	Pwofdw2	Non Partitioned Tables and Indexes			80	**RAID10**
/u08	Pwofdw2	Non Partitioned Tables and Indexes			110	**RAID5**
/u09	Pwofdw2	Non Partitioned Tables and Indexes			75	**RAID5**
/u10	Pwofdw2	Non Partitioned Tables and Indexes			110	**RAID5**
/u11	Pwofdw2	Non Partitioned Tables and Indexes			130	**RAID10**
/u12	Pwofdw2	Non Partitioned Tables and Indexes			110	**RAID5**

/u13	Pwofdw2	Non Partitioned Tables and Indexes			105	**RAID10**
/u14	Pwofdw2	Non Partitioned Tables and Indexes			110	**RAID5**
/u15	Pwofdw2	Oracle Archive Log Directory			150	**RAID10**
/u16	Pwofdw2	TEMP Tablespace 1x2			60	**RAID5**
/u17	Pwofdw2	UNDO Tablespace 1x2			100	**RAID10**
/u18	Pwofdw2	Unused	**Figure 1.2**		50	**RAID10**
/u19	Pwofdw2	Unused			50	**RAID10**
/u20	Pwofdw2	Oracle Backups			1650	**RAID5**
/u21	Pwofdw2	TEMP Tablespace 2x2			60	**RAID10**
/u22	Pwofdw2	UNDO Tablespace 2x2			50	**RAID10**
/u30	Pwofdw1	Oracle System Tablespaces, control file			5	**RAID5**
/u31	Pwofdw1	UNDO Tablespace			20	**RAID10**
/u32	Pwofdw1	Non Partitioned Tables and Indexes, TEMP Tablespace			15	**RAID5**
/u33	Pwofdw1	Unused			5	**RAID5**
/u34	Pwofdw1	Non Partitioned Tables and Indexes			5	**RAID5**
/u35	Pwofdw1	Non Partitioned Tables and Indexes			100	**RAID5**
/u36	Pwofdw1	Non Partitioned Tables and Indexes			100	**RAID5**
/u37	Pwofdw1	Oracle Backups			250	**RAID5**
/u51	Pwofdw2	Oracle Redo Log groups	**Figure 1.3**		10	**RAID5**
/u52	Pwofdw2	Oracle Redo Log groups			10	
/u53	Pwofdw2	Oracle Redo Log groups			10	
/u54	Pwofdw2	Oracle Redo Log groups			10	
/u55	Pwofdw2	Indexes for Part Tables			10	**RAID5**
/u56	Pwofdw2	Indexes for Part Tables			10	**RAID5**

REDO & REDO LOG's ARCHIVE

We currently have four REDO LOG groups with four mpts; /U51, /U52, /U53, /U54. The architecture within oracle is currently as below;

Group1	Group2	Group3	Group4
A C	B D	**A C**	**B D**

Where A, B, C, D are the mount points respectively. AC is one Duplex and BD is another. This means that even though we have 4 Redo log groups, we really have two. There is a Log Switch within Oracle that switches one Redo log group to the other when the previous one becomes full. With our duplex architecture, let's say Group2 becomes full at one time where Group1 hasn't yet completed writing its content to the Archive log LUN. The system will have to wait. It will not switch to Group3 or Group 4 because the mount point there is the same as in Group1 and 2.

My proposal is to either create 4 mount points, say EF and GH for group3 and group4 respectively, thus if Oracle DBA requires a duplex. Or better yet, split the Duplex and have four independent mpts, one for each group. This way in the above scenario there would be no wait. The Log switch in Oracle will have EF and GH available to switch to. The Redo log LUNs are also too small. Currently they are only 10G RAID5. ***See figure 1.3*** They need to be RAID10's because they do lots of Random writes. These are transaction logs, updates to the data tables and very random.

The spreadsheet actually has /u18 and u19 purposed for Redo. They are currently unused. Its content was moved onto the above four mpts. /u18 and /u19 are 50G each and RAID10. This is good. From drfuorap2's history, these mpts performed very badly when they were used as Redo, that's why they were moved to the above four RAID5 mpts.

From my analysis, they performed badly because of below not their configuration. The configuration is perfect.

/u18 and /U19 are in RAID groups 32 and 51 respectively. Remember a RAID group is a collection of physical drives, so logically it's one spindle. Take a look at the content of the RAID groups below.

Logical Drive	Raid Group	RAID	Size (GB)
DRFLORAP8_U09-INDX(ACCTSVC)	32	R10 (5+5) 146/15K	30
DRFLORAP3_U19x	32	R10 (5+5) 146/15K	100
DRFLORAP3_U20x	32	R10 (5+5) 146/15K	100
DRFLORAP3_U21x	32	R10 (5+5) 146/15K	100
DRFLORAP8_U35-UNDO(GAP)	32	R10 (5+5) 146/15K	25
DRFLORAP8_U59-TMPD(ODS)	32	R10 (5+5) 146/15K	35
DRFUORAP2_U19-REDO	32	R10 (5+5) 146/15K	50
DRFUORAP5_U19-INDX	32	R10 (5+5)	20

		146/15K R10 (5+5)	
DRFUORAP5_U20-INDX	32	146/15K R10 (5+5)	20
DRFUORAP5_U67-REDO	32	146/15K R10 (5+5)	10
DRFUORAP2_R06	32	146/15K R10 (5+5)	30
DRFVCTXP03	32	146/15K R10 (5+5)	45
DRFVHISP1	32	146/15K R10 (5+5)	15
DRFUORAP2_R04	32	146/15K R10 (5+5)	30
DRFVRAPPP01	32	146/15K R10 (5+5)	47
FREE	32	146/15K	11
DRFVJMASQLP010-DATA	51	R10 (5+5) 146/15K	160
DRFLORAP8_U26-INDX(GAP)	51	R10 (5+5) 146/15K	20
DRFLORAP8_U40-DATA(ODS)	51	R10 (5+5) 146/15K	20
DRFLORAP8_U41-DATA(ODS)	51	R10 (5+5) 146/15K	20
DRFLORAP8_U42-DATA(ODS)	51	R10 (5+5) 146/15K	20
DRFLORAP8_U43-DATA(ODS)	51	R10 (5+5) 146/15K	20
DRFUORAP2_U18-REDO	51	R10 (5+5) 146/15K	50
JMSCVMWARE-VSWAP-324	51	R10 (5+5) 146/15K	105
DRFVJMSCSAVP01	51	R10 (5+5) 146/15K	50
DRFSIISFNTP01-NEW	51	R10 (5+5) 146/15K	50
FREE	51	R10 (5+5) 146/15K	153

DRFLORAP8 and DRFUORAP5 have INDX, REDO, DATA, UNDO, about all of Oracles' components on the same spindle. Please note that since these components belong to different servers, this should not have been a problem. But in this case, it is because the above named server had applications that run at the same time DRFUORAP2 runs. This in essence means logically all the above mentioned components were accessed on the

same spindle at the same time while ETL was running ... leading to the very poor performance of the two mpts witnessed (u18 and u19).

Please note that the mpts /u51-/u54 seemed to do better briefly because they didn't have the above-mentioned contention in they RAID group.

My resolution is to create Redo LUNs, RAID10 configuration at the appropriate size advised by DBA in a RAID group that doesn't *have sharing servers* that run applications at the same time DRFUORAP2 does.

Currently, the Redo log's Archive LUN is at RAID10 configuration. This really needs to be changed to a RAID5 configuration. Archive LUN is making lots of writes that is true, but the catch here is that the writes are sequential (not random). They are also very large. RAID5 is in fact better than RAID10 on large sequential writes. Cache should be disabled on this LUN or a write–aside set.

The PLAN

Approach A

From multiple data collected from Oracle DBAs, below are the total storage requirements including growth.

Component	CurrentSize	RoundOff Cumulative
Users	67MB	Will keep Same
UNDO	55GB	
SYSTEM	5GB	
INDX	130.9GB	140GB
DATA	539GB	540GB
TMP	120GB	Will keep Same
REDO	40GB	100GB

Stage 1

IBM DS4500 has 1771GB un-configured space. The plan is to:

- Configure this space to suit the space requirement above. By creating new array groups/RAID groups; RAID10 and 5 accordingly.
- Provision this storage to DRFUORAP2
- Create new mpts/fs as in below
- Oracle DBA will then move all the **data-only** components onto the dedicated mpts for data with RMAN.
- Do the same for **INDX**, **REDO**, **UNDOs** and **TEMPs** on their dedicated mpts
- Move all the Redos from the current RAID5 10GB to the provided RAID10 50GB

Mpts	Sizes	Components	RAID Configuration
/u60	**10GB**	**SYSTEM**	**RAID5**
/u61	**55GB**	**UNDO**	**RAID5**
/u62	**120GB**	**TEMP**	**RAID5**
/u63	135GB	DATA	**RAID5**
/u64	135GB	DATA	**RAID5**
/u65	135GB	DATA	**RAID5**
/u66	135GB	DATA	**RAID5**
/u67	70GB	INDX	**RAID10**
/u68	70GB	INDX	**RAID10**
/u69	50GB	REDO	**RAID10**
/u70	50GB	REDO	**RAID10**
/u71	150GB	ACHIVE	**RAID5**

Stage 2

- Recover all mpts space for server on the CX3-80 and re-organize space to accommodate the below layout.
- Create new mpts on the server for CX3-80 storage from the recovered space.
- Migrate back onto to CX3-80. See below mpts for CX3-80

Layout at the End of Stage Two			
Mpts	Size	Component	RAID Configuration
/u11	10GB	SYSTEM	RAID5
/u12	**55GB**	**UNDO**	**RAID5**
/u13	**120GB**	**TEMP**	**RAID5**
/u14	135GB	DATA	RAID5
/u15	135GB	DATA	RAID5
/u16	135GB	DATA	RAID5
/u17	135GB	DATA	RAID5
/u18	70GB	INDX	RAID10
/u19	70GB	INDX	RAID10
/u20	50GB	REDO	RAID10
/u21	50GB	REDO	RAID10
/u22	150GB	ACHIVE	**RAID5**

Please note the mpt containing Oracle Backup for this server will remain the same on its current mpts. Also note that cache contention is still prevalent on CX3-80. We have the option to not do Stage 2 if Management is okay with the performance on IBM DS4500. This suggested re-architecture will in a few words ***simplify*** and ***nullify*** the issue on drfuorap2. It will most importantly keep defined components on specific spindles to relieve the system from lots on contention.

Approach B

Management reviewed the above approach and advised they are concerned about moving the storage onto IBM SAN because of known low performance issues. The IBM DS4500 cannot do RAID10 but only RAID5, RAID3 and RAID1.

From the above concern, the best resolution is detailed below. This approach will require down time. Management has given the approval and a down window of Saturday 12:00pm –Sun 12:00pm.

This approach will first ensure a full backup of all the mount points. Due to the critical nature of this process, management is concerned about the risk of having one backup

LUN, hence I will be creating another, a secondary backup on a different array IBM DS4500. Oracle admin will write two backups to the now two backup LUNs. Once this is complete, management is now secured with a redundant backup solution.

At the beginning of the project, due to space constraints, I will first off unmask all the storage for drfuorap2 on the CX3 80. I will dissolve the space and rebind new LUNs to accommodate the sizes and performance requirements explained earlier. This is a data loss process, hence the reason for the backup.

Once this is complete, I will wipe away all the mpts on the server and clean fstab. Then, I will recreate new mpt for the new storage and update fstab.

At this point Oracle DBA will, with the use of RMAN, recover each Oracle component onto the defined mpt purposed for it.

Execution

Task Description	Est Duration	Est Start Time	Comments
OUTAGE Sat 2/7			
Create Secondary backup LUN 600G on IBM DS4500 and zone to drfuorap2	3hr	Completed	should be completed anytime b4 commence of project
Oracle DBA needs to write a backup to the new redundant backup LUN			Should be completed b4 start of project| *mpt name will be provided*
Unmask all the storage for drfuorap2	45min	12:00	
Dissolve all the mpts on drfuorap2 and clean fstab	45min	12:50	
Unbind all the LUNs for drfuorap2 on the cx3 80	30min	1:30	
Architect new storage for drfuorap2 for performance based on available consolidated space	6hrs	2:00	

Bind new LUNs with new architect and mask to drfuorap2	45min-1hr	8:00PM	
Create new mpts on server	45mins	8:45	
Update fstab and create Filesystems	45mins	9:30PM	
PWOFDW1 AND PWOFDW2 Storage Migration			
Create RMAN DB Restore Scripts for both databases 'renaming' the data files to be restored on the newly created mount points.	8 hrs		To be completed Friday 1/16/08
New mount points are created and available			
Create new OFA directories on new storage MPs	15 min		
OUTAGE Sunday 2/8			
Receive Notification from Application Team that all Sunday morning processing is complete		9:30	
Pause Grid Control Jobs	10 min	9:30	Both databases
Create redo log groups on new storage	30 min	10:00	Both databases
Stop Oracle Listener	5 min	10:30	Databases Unavailable
Shutdown databases and restart in 'MOUNT' mode.	5 min	10:30	Required for RMAN Cold Backup
Take Cold Database Backup of PWOFDW1	30 min	10:30	Change backup destination
Take Cold Database Backup of PWOFDW2	3 hrs 30 min	10:30	
Remove existing PWOFDW1 data files and directories	10 min	11:00	Clean up of 'old' mount points.
Edit PWOFDW1 SPFILE and change Control File placements (3)	10 min	11:10	

Restore PWOFDW1 database to new Mount points	1 hr	11:20	start NOMOUNT and Restore controlfile , alter Mount and restore Data fles, Open resetlogs
Remove existing PWOFDW2 data files and directories	10 min	14:00	Clean up of 'old' mount points.
Edit PWOFDW2 SPFILE and change Control File placements (3)	10 min	14:10	
Restore PWOFDW2 database to new Mount points	4 hrs 30 min (est)	14:30	start NOMOUNT and Restore controlfile , alter Mount and restore Data files, Open resetlogs
Start Oracle Listener	5 min	19:00	Databases available
Take HOT Database Backup of both Databases	3 hrs 30 min	19:00	all prior hot backups unusable
Resume Grid Control Jobs	5 min	19:00	

Once this is completed, data will be verified and project completes. The secondary backup will be dissolved per management approval.

NEW MPT's			
Mpts	**Size**	**Component**	**RAID Configuration**
/u02 (Pwofdw1)	10GB	SYSTEM, control files	RAID5
/u03 (Pwofdw2)	20GB	SYSTEM, control files	RAID5
/u04	**100GB**	**UNDO**	**RAID5**
/u05	**100GB**	**TEMP**	**RAID5**
/u06	500GB	DATA	RAID5
/u07	500GB	DATA	RAID5
/u08	150GB	INDX	RAID10
/u09	150GB	INDX	RAID10
/u10	50GB	REDO	RAID10
/u11	50GB	REDO	RAID10
/u12	150GB	ACHIVE_LOG	**RAID5**
/u60	1TB	Sec-Bkup	RAID5

Please note that there are two databases on this server; Pwofdw1 and Pwofdw2. Both of their respective components will be installed on the above mpts. Above shows that we created two separate mpts /u02 and /u03 for each of their system and control files. Note that the below mpt would not be removed for their respective reasons.

/u01 (this contains the Oracle Home- Binaries)
/u20 (This is the backup for pwofdw2) 1.5TB
/u37 (This is the backup for pwofdw1) 250Gb

These mpts are currently in the oradg disk group in Veritas. I have added the secondary backup /u60 to the same disk group. I am will remove the current Pwofdw2_dg disk group, will leave the oradg diskgroup and create a new disk group called Pwofdw_dg This will contain both databases. Note that before the content of both diskgroups: oradg and Pwofdw2_dg were mixed with both databases. Not good.

Conclusion

The goal is to have all of each Oracle component recovered onto the defined mpts. This architect best addresses all the issues identified on this server with very limited available resources. The Data files are purposefully set to RAID5 (8+1) combination thus: 512K stripe size each, hence a 1MB stripe dept cumulative for both mpts. *(Unit element size: 64K)*

The Archive log is also set at RAID5 (4+1) combination to yield a 256K stripe and a write–aside cache is set at 511 on this LUN. All of the above are the best practices for an Oracle implementation.

Performance analysis can be approached in several ways. The key is to understand what you are analyzing and with the help of performance tools, logically see the problem. Note that all the different approaches lead to the same result.

In this particular case study, after all of the above was done, the server's performance improved slightly. The ultimate resolution in this particular case was to upgrade to a high-end storage array. ***Sometimes that's just it!***

Biography

In the past eight years I have worked extensively in the IT industry. I started off as a regular IT Analyst and eventually delved into Storage Administration. In my time in Storage, I have gained extensive experience in Implementation, operational- administration and architecture of SAN and solutions design. I have done several SAN consultancies for EMC under the EMC's Residency program. I was part of EMC's SMS team, where I was a Sr. Storage Consultant resident at GE Corporate. I later took a Lead Senior Storage Admin position with Cox Communications as part of their Messaging Storage team. I was also part of EMC's Federal Residency program | Professional Services where I took a consulting roles at the NASA Michoud Assembly Facility for the NFC | USDA as a Sr. Solutions Architect. I am currently working for JM Family, Inc as their Sr. Solutions Architect | SAN.

I have extensive work experience with EMC, 3PAR and IBM SAN Management & Architecture. I hold a Masters degree in Information Systems (Kennesaw State University) and a Bachelors degree in Mathematics & Computer Science (Concordia College). I am a Certified EMC SAN Specialist.