Consolidation and Resource Management

Consolidation

The primary drivers of consolidation centers around cost savings. Every new generation of hardware release is typically more powerful than the previous ones and this is also true for Exadata. IT shops can take advantage of this trend by consolidating the silos of database environments into a standardized powerful platform and ultimately achieve greater efficiencies by improving the total resource utilization and in effect lowering both capital and operational expenditures. On the following sections we will review the types of database consolidation and how to properly execute a consolidation exercise.

Types of Database Consolidation

Aside from resource requirements there are many factors that need to be considered when consolidating multiple databases. These factors include, but not limited to: namespaces, isolation, maintenance, upgrade, backup and recovery, cloning, service level agreements. Each approach has its own pros and cons and we will go through each of them below.

Server

This is probably the easiest route which allows you to put multiple databases into a single database server or an Oracle RAC cluster. Each application is isolated to a dedicated database which can be easily maintained and upgraded. But as more and more databases are moved to the cluster, the amount of resources that must be dedicated to each of them presents a practical limit to the consolidation density that can be achieved. This may also be the same on a resource standpoint for any virtualization consolidation solution which is probably has more overhead than a dedicated database. On these kinds of consolidation especially on developerment environments we usually see the memory capacity being reached first.

Schema

This method puts separate application schemas coming from standalone databases into a consolidated single instance or Oracle RAC database. There's a lot more planning and due diligence that has to be done when doing this kind of consolidation. The DBAs and the application team have to check if there are any conflicting schema names or hard-coded schema names on the SQLs or packages. If there are any conflicts then they have to be resolved with some application and database level changes. It would be an easy consolidation if all the application schemas fit nicely with each other, meaning there are no object name collisions due to the shared data dictionary. If this method is done correctly then there will only be one database to administer and a more efficient resource usage and density because of the shared background processes and memory (SGA/PGA) across applications.

Multi-tenancy

Oracle Multitenant is a new option in 12c with a big architecture change introducing the concept of Container (CDB) and Pluggable (PDB) databases. This enables an Oracle database (single instance or RAC) to be a Container (CDB) which has a

single set of background processes, shared memory (SGA/PGA), undo, common temp, control files, and redo log for all the Pluggable (PDB) databases which have their separate set of database files. It's like combining the full isolation of a dedicated database and efficient resource usage and density of schema consolidation. Although most of our customers I see today are still in 11gR2 the Multitenant architecture of 12c is clearly the direction of the future. With Multitenant there's only one database to administer and the consolidated application environments have their own namespace through separate PDBs which require no application changes. Any PDBs can be maintained without impacting the other PDBs residing in the same CDB and it can be easily unplugged and plugged to another CDB. There's a lot more flexibility, features, and benefits with the Multitenant architecture that can be discussed in this section of this chapter. To know more about Multitenant a good starting point is the Part VI of the Database Concepts guide http://docs.oracle.com/database/121/CNCPT/part_consol.htm#CHDGDBHJ

General Consolidation and Sizing Workflow

Capacity planning plays a very important role to ensure proper resources are available to be able to handle the expected and unexpected workloads. Exadata is not really different when it comes to that, although it has an intelligent storage every resource component (CPU, Memory, Storage Space and IO Performance) still have limited capacity. Regardless of the type of database consolidation you choose the primary principle is to ensure the application workload requirements will fit into the available capacity of the platform. Below outlines the process that a DBA must go through to ensure that the capacity can meet the resource requirements of the consolidated environment.

Gather Data and Plan

The first step on the consolidation process is getting all the resource requirements. This is done by extracting all the configuration and historical performance data across the database environments. The preferred source of performance data comes from the Automatic Workload Repository (AWR) and at least 100 to 365 days worth of data should be gathered which is a good enough representation of daily, weekly, monthly, and quarterly trends. The quantity and granularity of the data is based on AWR retention and snap interval of the databases. The table below shows the data points gathered categorized by resource components:

Source Host Details	Workload	CPU	Memory	Storage
Physical Hostname Zone/LPAR/VM Hostname Server Make & Model OS DB Name DB Version App Type/Front End Workload Type (OLTP/DW/Mix) Environment Type (prod/DR/dev/test/QA) Node count (single instance/RAC)	Top Events Load Profile Top Services Top SQLs & Modules	Cpu Make & Model CPU Speed Number of CPU cores & threads CPU Utilization Avg/Peak	Host Physical Memory SGA Size PGA Usage Size	Storage Make & Model Disk RAID Level # of I/O Channels Database Size Backup Size Peak Read + Write IOPS Peak Read IOPS Peak Write IOPS Peak Read + Write MB/s Peak Read MB/s Peak Write MB/s Peak Read + Write Ratio I/O Latency

Once all the required data are collected the DBA can now start grouping the databases by platform, workload type, and by criticality. And then the migration planning can be started with the whole team and could also include some of the important steakholders. Keep in mind that the accuracy of the whole planning, sizing, and capacity planning is dependent on the quality of the data, as much as possible guesstimates should be avoided and everything should be based on facts, numbers, and figures. This brings integrity to the end-to-end process and to the key people involved doing the consolidation and sizing. The planning team should address which of the databases will be migrated first, and which are the ones that are likely to be combined into a single database (if doing Schema Consolidation). Also, determining the databases that can live together in one cluster should be part of the discussion and if there are any databases that need to have some application changes during the migration, in this case the team should definitely allow time for PoC (Proof of Concept) or PoV (Proof of Value) and thorough testing. The other factors that must be discussed generally include, but not limited to: downtime windows, backup requirements, test/development environment refreshes and cloning, RDBMS version constraints. Simply put, the more databases that will be consolidated the more detailed due diligence and preparation have to be done.

Workload Characterization

The idea of workload characterization is to obtain a better understanding of how the resource components are being used. This is done by visualizing the collected time series data sets to get a perspective of how intense the activity of the source databases against their current environment capacity. The intensity is quantified mainly from the physical level (resource components) by assessing their utilization, rate of work, and latency which we can then gain more detailed insights by correlating it with the application level data.

Individual Database Analysis

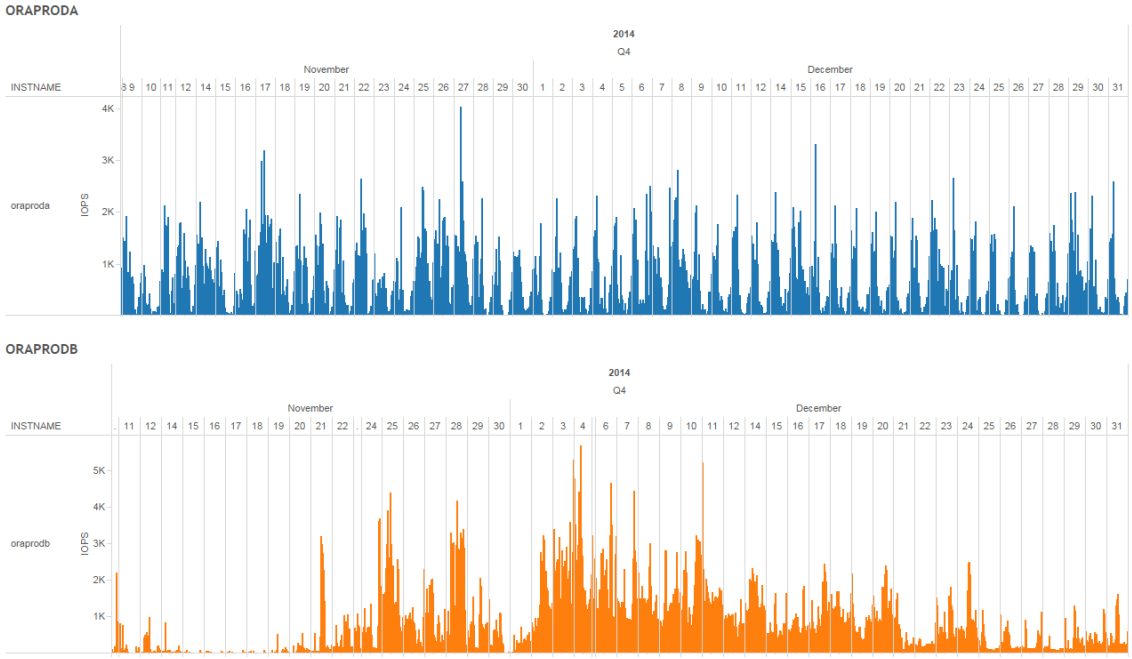
The general flow of the analysis starts with the individual databases. For every resource component, by lining them up in one graph the DBA can easily see the heavy load databases and the hosts that they are coming from. Also, if any of them is coming from a RAC environment then the visualization is pretty helpful to see the node affinity or if the workload across the cluster is skewed to just some of the nodes. Other essential thing that needs to be checked is the periods where workload spikes are happening and if there is a recurring trend and just as important is the appearance of any natural workload growth or noticeable sudden growth due to application or environment changes.

Below shows two single instance databases ORAPRODA and ORAPRODB coming from two different hosts, the workload characterization examples shown moving forward will focus more on CPU and I/O because of their dynamic and rather complex behavior. The CPU resource graph is measured as Average Active Sessions on CPU (AAS CPU) where 1 AAS CPU is equivalent to 1 CPU thread that's 100% utilized. As you can see both databases pretty much have the same CPU load at about 5 CPUs except for the spikes happening on the ORAPRODA database where it is reaching beyond its current CPU capacity.

- **Note:** Oracle instruments the CPU usage in three different ways:
  CPU - the real CPU cycles
  CPU wait - the CPU time spent on run-queue
  Scheduler - the CPU time spent above the specified CPU_COUNT

These three wait classes can all be measured in AAS (Average Active Sessions) or percentage of the total host utilization. This is discussed futher on the Database Resource Manager section.

This other graph shows the total I/O Operations Per Second (IOPS) workload of both databases. It is simply the sum of Single Block Read+Write, Redo IO, and Multiblock Read+Write across time series. Here the ORAPRODA and ORAPRODB are on the range of 4000 and 5000 individual peak IOPS respectively.

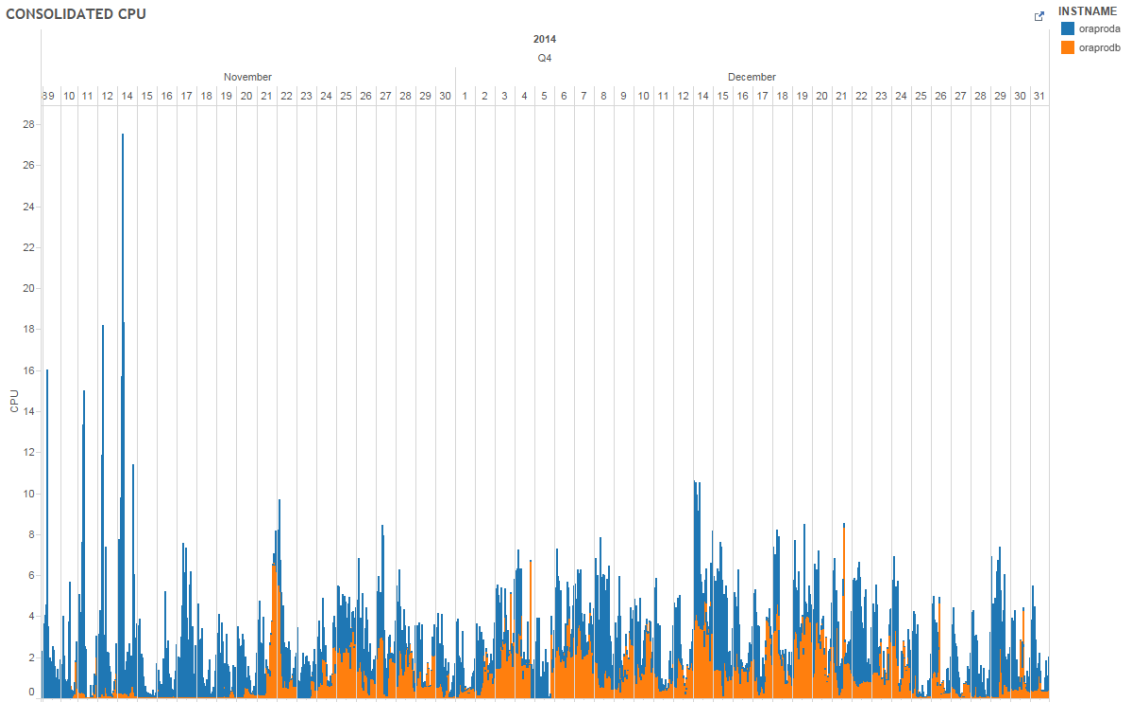


Usually the spikes with extreme values need some workload validation if it's truly a part of the application behavior or just an adhoc activity, if it's the latter then it can be considered as an outlier and can be discarded. As always the DBA can check the workload in question with the application team or with the gathered application level data.

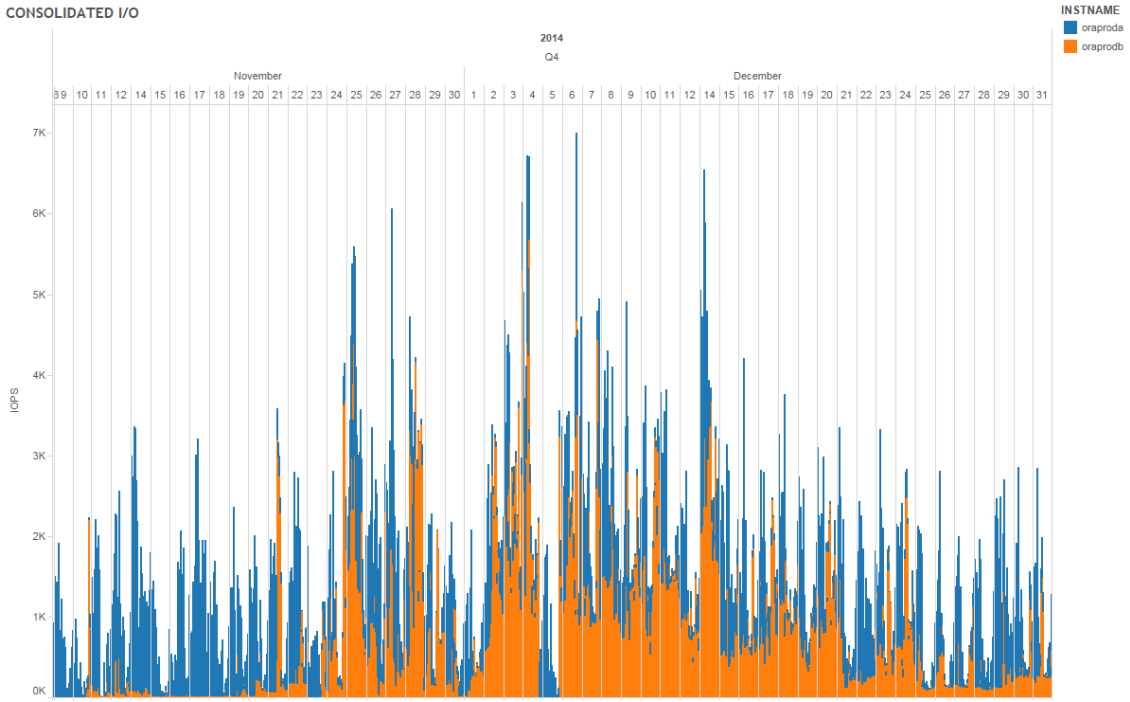
Combined Workload Analysis

The next step is combining the individual workloads to just one stacked graph lined up in the same time dimension. This is crucial because by having a merged view we will be able to tell if the combined workload numbers will introduce higher peaks and in effect increase the total resource requirements. The other benefit to this is the DBA can have an idea of what will the end workload would look like even before migrating the databases. Also by seeing the workload distribution the potential resource hoggers which could cause some response time fluctuations can earlier be identified.

Below graph shows the consolidated CPU load numbers of both databases and as you can see the overall range now changed from 5 (for the individual CPUs) to 10 CPUs (when combined) while the extreme spike on ORAPRODA is at 27 CPUs.



The consolidated IOPS graph shows the range is now at 7000 IOPS.



When consolidating databases the end result of workload characterization is to come up with the final resource requirement for each of the resource component. The requirements must be geared towards right-sizing the target environment and being too conservative may risk it to be overprovisioned and overpriced. The opposite would under-size the target environment and the overall performance and stability would suffer. Below are the two methods to get the final numbers:

Sum of all individual peaks

This can also be called the "worst case scenario" where all of the peak period of the databases line up altogether at the same time. On the Individual Database Analysis example if we consider all the data samples the CPU peaks of ORAPRODA and ORAPRODB are at 27 and 8 CPUs respectively so the final CPU resource requirement will be **35 CPUs**. On the I/O side, we have 4000 and 5000 individual peak IOPS respectively so that will be **9000 IOPS** resource requirement.

Consolidated workload peak

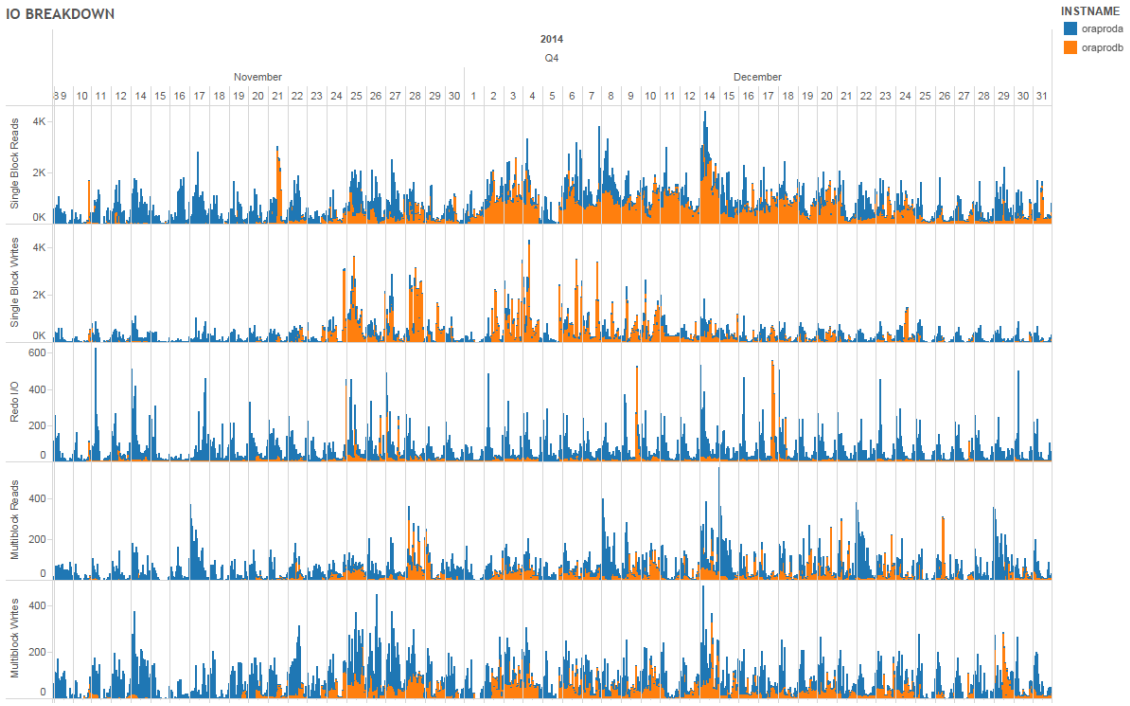
This method is mainly doing a combined workload analysis and then getting the peak. With this method all workloads line up in the same time dimension so the values here is definitely much lower to the sum of all individual peaks. This can also be called the "best case scenario" but then sometimes a lot of unforeseen things can change including the workload windows when all of the databases are migrated so there might be some risk of under-sizing the environment. On the Combined Workload Analysis example above, the CPU requirement of both ORAPRODA and ORAPRODB would be **27 CPUs**. On the I/O side, it's **7000 IOPS**.

Both of the methods should be used and compared and together with the insights from the Individual and Combined Workload Analysis the DBA will have a better idea of what numbers to use. If either of the numbers is way too high then we can smoothen the sample data by removing anything above the 99th percentile which will bring it down to a more reasonable number range. The point here is, end numbers need to conservative with just enough allowance but not too conservative (overprovisioned). For now keep the above numbers in mind and we will use this on the Resource Requirements section.

Quantifying the Breakdown of Workload Data

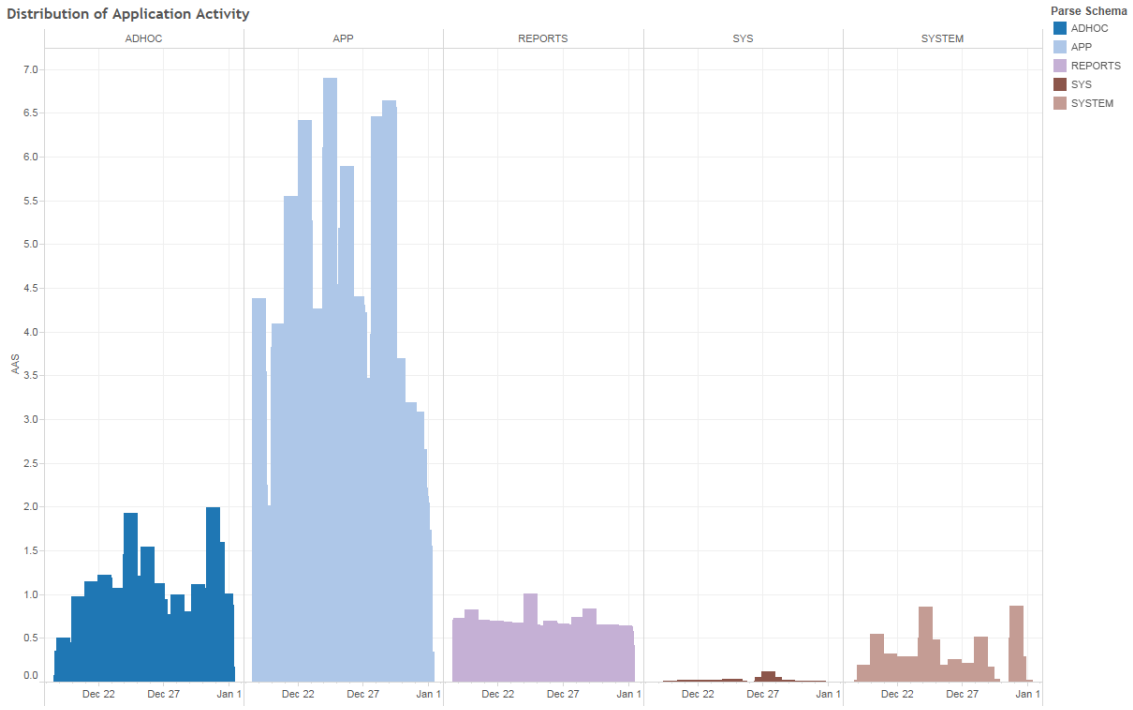
On more critical exercise is to understand the breakdown of IO usage and application activity. This is used to determine if the ratio of IO activity is heavier on writes than reads or IOPS vs bandwidth which will also help the DBA to plan for some Exadata features like the write-back flash cache (if it's heavy on writes) or the proper setting for I/O Resource Management. The distribution of the application activity on the other hand is very useful on the Database Resource Management planning on coming up with the optimal settings for the resource shares or percentage allocation.

Below is the breakdown of IO by Single Block Read+Write, Redo IO, and Multiblock Read+Write across time series. The combined IO profile of both databases is equally heavier on Single Block Read and Write IOPS than the bandwidth (Multiblock IOs).



On the application activity which graphed by Average Active Sessions and by the way this can be any of the delta columns (disk reads, executions, etc.) on DBA_HIST_SQLSTAT, the APP schema is driving most of the workload while there are some reports

going on for ADHOC and REPORTS schemas with a little bit of activity on the side for SYSTEM. This graph is useful for Resource Management planning and depending on the criticality on performance or SLAs this distribution can be used to get the proper settings on throttling the activity of non-priority users or application schemas.



Resource Requirements

The end result of the workload characterization is ultimately the resource requirement numbers. The table below shows the final numbers for the two databases that will be migrated and consolidated to Exadata.

Host Speed (SPECint_rate 2006/core)	Total Host CPUs	CPU Utilization	DB name	PGA Max (G)	SGA Max (G)	Database Storage (G)	Total R+W IOPS
32	24	100.00%	ORAPRODA	20	40	5445	4000
26	16	50.00%	ORAPRODB	4	10	6144	5000

Although we focused mainly on CPU and I/O on the workload characterization examples all the relavant columns are explained below:

CPU

There are three columns related to the CPU. These are SPECint_rate2006/core, Total Host CPUs, and CPU Utilization. The idea here is to get the amount of current CPUs used and the CPU speed of the host. In this way when the DBA models the source requirements to the target Exadata capacity the speed differences of the CPUs will be accounted. Simply put, the 2 CPUs requirement on a slower CPU will drop down to 1 CPU if the target environment has twice faster CPUs. The servers for the ORAPRODA and ORAPRODB are HP DL360e Gen8 and ML370 G6 respectively and the target environment is the Exadata X5-2 with Intel Xeon E5-2699 v3 @ 2.30GHz rated at 39 SPECint_rate2006/core. So the speed differences for ORAPRODA and ORAPRODB are 32/39=.82 and 26/39=.67 respectively. Which simple means 1 CPU usage of ORAPRODA is equivalent to .82 CPU of Exadata X5-2. If the speed differences is not accounted then the target environment could be overprovisioned.

We can interpret the numbers as follows:

- 100% of the total 24 CPUs at speed of 32 SPECint_rate2006/core for ORAPRODA which is equivalent to 24 AAS CPU
- 50% of the total 16 CPUs at speed of 26 SPECint_rate2006/core for ORAPRODB which is equivalent to 8 AAS CPU

Notice that we ended up with 24 AAS CPU instead of 27 on ORAPRODA. It's because we have validated the workload and that peak is an adhoc activity, but then we still want to be conservative that's why we put it at 100% utilization. This brings the total CPU requirement to 24+8=**32 AAS CPUs** which is somewhere in the middle of our "Worst Case" (35) and "Best Case" (27) numbers. Again, conservative but not too conservative (overprovisioning).

■ **Note:** The webinar "Where did my CPU go?" discusses in more detail the different CPU events and CPU speed comparison using SPECint_rate2006 and Oracle Logical IO benchmark tools like cputoolkit and SLOB http://enkitec.tv/2015/03/20/where-did-my-cpu-go-enkitec-redgate-webinar-by-karl-arao

---

Memory

Memory requirement applies to the total SGA plus PGA. This is based on the most recent sum of SGA components from DBA_HIST_SGA or GV$SGA plus the "total PGA allocated" of DBA_HIST_PGASTAT which is the actual PGA used and not the value from PGA_AGGREGATE_TARGET parameter. For the PGA memory size the DBA has to get the max value across time series to be conservative.

Storage Space

The storage space is pretty straightforward. This is based on the most recent sum of PERMANENT, UNDO, and TEMPORARY tablespaces across time series.

I/O Performance

The IOPS is based on the individual peaks of both databases which brings the total to **9000 IOPS** resource requirement which is a good enough conservative number and a little bit higher than the Consolidated workload peak. Although this high level number is like an abstraction of the IO breakdown we've seen, as long as the target Exadata hardware can meet this IOPS number then the bandwidth requirement is also covered.

■ **Note:** The resource requirements collection scripts can be downloaded at https://karlarao.wordpress.com/scripts-resources under "run_awr-quickextract"

With all the final numbers in place we can now have an idea of what hardware configuration will fit our requirements. Our two databases can definitely fit in an Exadata X5-2 Eighth Rack with enough headroom to grow. But then, if there's one more database like ORAPRODA to be migrated then the storage space capacity would not be enough even at ASM Normal Redundancy and should instead be sized as a Quarter Rack. Also, there are a few more essential things to consider and this will be discussed in the next section.

Modeling of the Capacity

This is where we model the capacity of the target Exadata environment by making use of the resource requirements that came out of the workload characterization. Here we make sure that the resource utilization across the nodes is balanced and the main resource components (CPU, Memory, Storage Space and IO Performance) are well within their ideal usage. Also, there are other essential things that can affect the overall capacity of the environment which needs to be accounted like the failure of a node or the ASM redundancy level. All of these factors are critical for capacity planning in a consolidated environment and will be discussed in this section.

Create a Provisioning Plan

Instance Mapping

The first step is to decide where the instances will run when moved to the target environment. This is also called the Instance Mapping or the Node Layout. We will model our example databases ORAPRODA and ORAPRODB across the two nodes. On a RAC environment by default the workload will be load balanced between the available RAC instances, so if a database running

on a two node RAC has 72 CPUs requirement given that the services are properly configured the load is then equally distributed and each node gets 50/50 percent share of the CPU requirement. If that's an X5 compute node (72 CPUs capacity) then that makes it 50% utilization across each node. What we don't want to happen is an unbalanced utilization where the load of the 1st node is 100% and the 2nd is idle. And regardless of the node layout, we are still making use of 72 CPUs cluster-wide but the users are taking a performance hit on the overprovisioned 1st node. Below is an example of the Instance Mapping:

Host Speed (SPECint_rate 2006/core)	Total Host CPUs	CPU Utilization	DB name	PGA Max (G)	SGA Max (G)	Database Storage (G)	Total R+W IOPS	**node 1**	**node 2**
								2 instance 55% cpu used 29% mem used 55G HPages	1 instance 15% cpu used 5% mem used 11G HPages
32	24	100.00%	ORAPRODA	20	40	5445	4000	**P**	F
26	16	50.00%	ORAPRODB	4	10	6144	5000	**A**	**P**

There are three values for the Instance Mapping:

- P = preferred node (green)
  - Oracle RAC preferred instance for a service or **primary node**
  - The instance accepts client connections and consumes CPU and Memory resources
- A = available node (blue)
  - If the preferred instance fails this is the **secondary node**
  - The client connections are just pre-connected, sessions will failover only when the preferred node fails/shutdown
  - The instance is up and running and consumes only Memory resources
- F = failover node (red)
  - Also a **secondary node**, it is configured as a preferred instance but intentionally disabled
  - This instance does not accept client connections
  - The instance is down and does not consume resources

Let us summarize:

- The ORAPRODA and ORAPRODB are both running as two node RAC databases. For this configuration we have an Exadata X5-2 Eighth Rack which is 36 CPUs and 256GB Memory on each node.
- Here we have ORAPRODA and ORAPRODB running as Preferred on node 1 and node 2 respectively. In this case, if node 2 encounters a power failure then ORAPRODB can readily use the node 1 as sessions are pre-connected. But if node 1 fails, then ORAPRODB is not affected and ORAPRODA service on node 2 has to be enabled and started first before it can be used.
- For the Memory resource, as the database gets spread across more nodes (either Preferred or Available) the more physical memory it will consume unlike the CPU resource where it is being split across nodes. The reasoning behind this is when a node fails the survining node should be able to take over the Memory resource needs of the failed node. But then this can be adjusted or resized accordingly.
- The resource utilization numbers on node 1 are computed as follows:
  - CPU
    - ORAPRODA = **19.69** Exadata X5-2 CPUs
    - (24 CPUs x 100% utilization) x (32 / 39 SPECint_rate2006/core)
    - Exadata X5-2 node 1 CPU utilization = **55%**
    - 19.69 / 36 = 54.69% or 55%
  - Memory
    - ORAPRODA (Preferred) = 20 PGA + 40 SGA = **60GB**
    - ORAPRODB (Available) = 4 PGA + 10 SGA = **14GB**
    - Exadata X5-2 node 1 Memory utilization = 74GB / 256GB = 28.9% or **29%**
- On the Instance Mapping, on top of the P,A,F are the following:
  - The number of instances running on that node.
  - The node level CPU utilization
  - The node level Memory utilization
  - The Huge Pages settings with 10% allowance computed from the total SGA size per node. The proper settings is important and be aware not to overallocate on the Huge Page configuration. Also remember to resize as you add/remove instances.

Moving on, the ideal Instance Mapping for ORAPRODA and ORAPRODB are as follows:

Host Speed (SPECint_rate 2006/core)	Total Host CPUs	CPU Utilization	DB name	PGA Max (G)	SGA Max (G)	Database Storage (G)	Total R+W IOPS	node 1	node 2
								2 instance 35% cpu used 29% mem used 55G HPages	2 instance 35% cpu used 29% mem used 55G HPages
32	24	100.00%	ORAPRODA	20	40	5445	4000	P	P
26	16	50.00%	ORAPRODB	4	10	6144	5000	P	P

Note: The "Managing Workloads Using Dynamic Database Services" section of the "Database 2 Day + Real Application Clusters Guide" is a good starting point on how to create services using OEM12c or SRVCTL https://docs.oracle.com/database/121/TDPRC/configwlm.htm#TDPRC303

Node Failure Scenarios

The Instance Mapping should also depend on the failure scenarios where let's say if one node goes down the end resource utilization of the remaining nodes should still be on an acceptable range (70-75% below). Below scenario is a complete failure of node 2 and you can see that the sessions failed over to the remaining Preferred node which caused an increase in resource utilization.

Host Speed (SPECint_rate 2006/core)	Total Host CPUs	CPU Utilization	DB name	PGA Max (G)	SGA Max (G)	Database Storage (G)	Total R+W IOPS	node 1	node 2
								2 instance 70% cpu used 29% mem used 55G HPages	0 instance 0% cpu used 0% mem used 0G HPages
32	24	100.00%	ORAPRODA	20	40	5445	4000	P	X
26	16	50.00%	ORAPRODB	4	10	6144	5000	P	

Again the CPU and Memory has to be adequately sized to be able to handle the node failure scenarios. This process is essential to the availability planning of the whole cluster in terms of patching or maintenance. For large database consolidations this is also the part where we do trial and error until we get the sweet spot of the Instance Mapping where we have already failed over each of the nodes and the end utilization of the remaining nodes are still on an acceptable range. For Exadata half and full rack, you should plan for failure of two nodes for maximum uptime across the cluster.

Review of the utilization

As we change the Instance Mapping we can quickly check back to the node level utilization and see the resource component effects of the change. The imbalance will appear obvious on the node level but we should still keep an eye on the cluster-wide resource utilization. Below we added one more database to migrate which are very identical to the ORAPRODA database in terms of the source machine and resource requirements. Take note that our hardware is still Exadata X5-2 Eighth Rack. The "red highlight" shows any resource component that reaches above 75% utilization which is the limit to our acceptable utilization range. Here, we are on the critical level for the Storage Space. But then if you look closely, the CPU is also at risk and if a node failure happens then the remaining node will be at 124% utilization.

Overall Utilization:

Total CPUs & pct% USED	Total Mem GB & pct% USED	Total DATA Storage GB & pct% USED	Total RECO Storage GB & pct% USED	DATA+RECO Storage GB & pct% USED
44.72 CPUs 62.1%	268 GB 52.3%	17034 GB 50.96%	17034 GB 50.96%	34068 GB 101.93%

Recommended Hardware:

Equivalent compute nodes	Total Workload IOPS	ASM Normal Redundancy HW IOPS	ASM High Redundancy HW IOPS
1.2 nodes	13000	23400	31850

Node Level Utilization:

Host Speed (SPECint_rate 2006/core)	Total Host CPUs	CPU Utilization	DB name	PGA Max (G)	SGA Max (G)	Database Storage (G)	Total R+W IOPS	node 1	node 2
								3 instance 62% cpu used 52% mem used 99G HPages	3 instance 62% cpu used 52% mem used 99G HPages
32	24	100.00%	ORAPRODA	20	40	5445	4000	P	P
26	16	50.00%	ORAPRODB	4	10	6144	5000	P	P
32	24	100.00%	ORAPRODC	20	40	5445	4000	P	P

The goal is to keep the utilization way below 75% such that there's more storage space to grow and we can still be at a good CPU utilization when one node fails. This means we need to add more Storage and Compute nodes or ultimately we can choose to upgrade to the next Exadata Configuration which brings us to next topic of evaluating the ideal hardware.

Note: The Exadata Provisioning Worksheet used to Model the Capacity can be downloaded at https://karlarao.wordpress.com/scripts-resources under "Consolidation" section. The tool works around the basic Capacity Planning formula: Utilization = Requirements / Capacity

---

Evaluate the ideal hardware

All the data points gathered translates to resource requirements and then to the amount or size of hardware it needs to run smoothly. This section is essential for validating if the hardware that we currently have or planning on buying is enough to run all the databases that will be migrated or consolidated. In our case, we need to make sure that the ideal hardware for the databases ORAPRODA to C is well within the ideal cluster-wide utilization across resource components. From the previous section, we found out that we are Storage Space and CPU constrained.

We can start by comparing the Exadata X5-2 datasheet against our resource component numbers. Taken from the previous section, our resource requirements are as follows:

- CPU = **44.72 CPUs** (threads or logical CPUs)
  - Sum of all ORAPRODA to C CPU requirements
- Memory = **268GB**
  - Sum of all ORAPRODA to C Memory requirements
- Storage Space = **34,068 GB**
  - This Storage Space requirement accounted for the recovery area (RECO) which we usually size the same as the database space (DATA) which means we want to have a space for at least 1 full backup of the database. Of course all of the backups will be RMAN based. This just gives a conservative sizing for the recovery area. An alternative would be to put the backups outside of Exadata (ZFS appliance or NFS), but for our case we will put it in RECO.
- Storage IOPS = **23,400 IOPS**
  - This is being sized for a Normal Redundancy disk group. The IOPS requirement accounts for the read and write ratio which is 50:50 as what we have seen from the distribution of I/O workload.
  - Normal Redundancy Writes penalty is 2 while High Redunancy is 3. Meaning 2 IOs are being done on the Storage Cell side for one write I/O call on the database side.
  - Also we are putting a 30% allowance for the possible Write-back Flash Cache destage and flash cache metadata writes I/O overhead (categorized as OTHER_DATABASE on Exadata Storage Cell metrics).
  - The computation is below
    - (Total Workload IOPS x Read Ratio) + (Total Workload IOPS x Write Ratio x Redundancy x 30% writes allowance)
    - (13000 x 0.5) + (13000 x 0.5 x 2 x 1.3) = 23,400 IOPS

Below is the most useful section of the Exadata X5-2 data sheet which shows the key performance and capacity metrics. The encircled ones (from top to bottom) are the Flash, Disk, and Storage Space IO Performance and Space capacity of a High Capacity Quarter Rack configuration which is just the right size for our resource requirements. Also it is worth mentioning that we do not factor in the features Offloading and Table Compression on the sizing just to be conservative. Although these Exadata features gives extra capacity on Disk Space or CPU when realized in reality all workloads are different and unless proven through a PoC (Proof of Concept) or PoV (Proof of Value) and thorough testing then we can take these features into account.

EXADATA DATABASE MACHINE X5-2 KEY CAPACITY AND PERFORMANCE METRICS

Metric	Full Rack		Half Rack		Quarter Rack		Eighth Rack	
	HC[1]	EF[2]	HC	EF	HC	EF	HC	EF
Flash Metrics								
Maximum SQL Flash Bandwidth [3]	140 GB/s	263 GB/s	70 GB/s	131 GB/s	30 GB/s	56 GB/s	15 GB/s	28 GB/s
Maximum SQL Flash Read IOPS[4]	4,144,000	4,144,000	2,072,000	2,072,000	1,036,000	1,036,000	518,000	518,000
Maximum SQL Flash Write IOPS [5]	2,688,000	4,144,000	1,344,000	2,072,000	576,000	1,036,000	288,000	518,000
Data Capacity (raw) [6]	89.6 TB	179.2 TB	44.8 TB	89.6 TB	19.2 TB	38.4 TB	9.6 TB	19.2 TB
Effective Flash Cache Capacity [8]	Up to 672TB		Up to 336 TB		Up to 144 TB		Up to 72 TB	
Disk Metrics								
Maximum SQL Disk Bandwidth [3]	20 GB/s		10 GB/s		5 GB/s		2 GB/s	
Maximum SQL Disk IOPS [4]	33,000		16,000		7,000		3,500	
Data Capacity (raw) [5]	672 TB	179 TB	336 TB	90 TB	144 TB	38 TB	72 TB	19 TB
Combined Metrics								
Data Capacity (usable) [7]	300 TB	80 TB	150 TB	40 TB	63 TB	17 TB	30 TB	8 TB
Maximum Data Load Rate [9]	21.5 TB/hour	21.5 TB/hour	10.5 TB/hour	10.5 TB/hour	5.0 TB/hour	5.3 TB/hour	2.5 TB/hour	3.0 TB/hour

Actual system performance varies by application.

[1]HC = High Capacity [2]EF = Extreme Flash

[3] Bandwidth is peak physical scan bandwidth achieved running SQL, assuming no database compression. Effective user data bandwidth is higher when database compression is used.

[4] Based on 8K IO requests running SQL. Note that the IO size greatly affects Flash IOPS. Others quote IOPS based on smaller IOs that are not relevant for databases.

[5] Based on 8K IO requests running SQL. Flash write I/Os measured at the storage servers after ASM mirroring, which usually issues multiple storage IOs to maintain redundancy.

[6] Raw capacity is measured in standard disk drive terminology with 1 GB = 1 billion bytes. Usable capacity is measured using normal powers of 2 space terminology with 1 TB = 1024 * 1024 * 1024 * 1024 bytes.

[7] Actual space available for a database after mirroring (ASM normal redundancy) while also providing adequate space (one disk on Quarter and Half Racks and two disks on a Full Rack) to reestablish the mirroring protection after a disk failure in the normal redundancy case.

[8] Effective Flash Capacity is larger than the physical flash capacity and takes into account the high flash hit ratios due to Exadata's intelligent flash caching algorithms, and the size of the underlying disk storage. It is the size of the data files that can often be stored in Exadata and be accessed at the speed of flash memory.

[9] Load rates are typically limited by database server CPU, not IO. Rates vary based on load method, indexes, data types, compression, and partitioning.

http://www.oracle.com/technetwork/database/exadata/exadata-x5-2-ds-2406241.pdf

Pluging in the Exadata X5-2 High Capacity Quarter Rack configuration numbers brings us back to the healthy cluster-wide level utilization.

Overall Utilization:

Total CPUs & pct% USED	Total Mem GB & pct% USED	Total DATA Storage GB & pct% USED	Total RECO Storage GB & pct% USED	DATA+RECO Storage GB & pct% USED
44.72 CPUs 31.1%	**268 GB** 52.3%	**17034 GB** 25.44%	**17034 GB** 25.44%	**34068 GB** 50.89%

Recommended Hardware:

Equivalent compute nodes	Total Workload IOPS	ASM Normal Redundancy HW IOPS	ASM High Redundancy HW IOPS
0.6 nodes	**13000**	**23400**	**31850**

Node Level Utilization:

Host Speed (SPECint_rate 2006/core)	Total Host CPUs	CPU Utilization	DB name	PGA Max (G)	SGA Max (G)	Database Storage (G)	Total R+W IOPS	node 1	node 2
								3 instance 31% cpu used 52% mem used 99G HPages	3 instance 31% cpu used 52% mem used 99G HPages
32	24	100.00%	ORAPRODA	20	40	5445	4000	P	P
26	16	50.00%	ORAPRODB	4	10	6144	5000	P	P
32	24	100.00%	ORAPRODC	20	40	5445	4000	P	P

The Storage Space and CPU issues are resolved and the cluster can now afford to loose one node and still be able to perform. But how long is this hardware configuration going to last? That brings us to next topic of Headroom Projection.

Headrom Projection

Headroom is simply the environment's allowance for growth before experiencing any performance degradataion or outage and this is usually communicated by deriving the *Headroom Time* which is the number months to consume all of that allowance. In this section we will focus on the Headroom of the CPU. This is where we factor in the cluster level CPU utilization and consolidated yearly growth rate of the databases to define the capacity headroom of the target hardware or environment. Any optimization projects on the application side can be optionally factored in because it affects the overall workload by gaining more headroom or capacity (of course when successfully done). Most IT shops usually go with 2 to 3 years (at max 5 years) of headroom projection but this should be aligned with the environment's growth rate, budget, and hardware refresh cycle so the final hardware capacity is adjusted accordingly.

The formula for the Headroom and Headroom Time Calculation are as follows:
Headroom = (Ideal Usage% x Max Capacity%) - Current Usage% - (12 months growth rate% - 12 months optimization%)
Headroom Time = (Headroom% / (12 months growth rate% - 12 months optimization%))

The equation states that the *Headroom* of an environment is equal to the *Ideal Usage* from the *Maximum Capacity* minus the *Current Usage* minus the *12 months growth rate* minus the optimization *projects that will be done over 12 months*. This can be better explained by making use of our previous example where we upgraded from an Exadata X5-2 Eighth Rack to a Quarter Rack when we ran out of CPU resource because of the new ORAPRODC database.

The variables are as follows:

- Max capacity = The maximum resources you have for the resource component
- Ideal usage = The ideal utilization, for CPU we usually go with 75%. This is simply the amount of resources that is planned for usage. We don't go with 100% because as the workload move towards the max capacity unpredictability happens due to diminishing returns on performance.
- Current usage = The current utilization of the resource component
- Growth = The sum of rate of growth for 12 months, this is a variable of business growth or just natural growth of the workload. In our case the projected workload growth will make the CPU +30% more then that's 30/12=2.5 rate per month
- Optimization = The tuning exercises. If we tune the SQLs or optimize the code then we gain headroom and this is being deducted to the workload growth (above). So let's say in 12 months we will have a tuning exercise that will decrease the workload by 15% so that's 15/12=1.25 decrease per month. In effect, the new growth rate is 2.5 – 1.25 = 1.25% growth rate per month.

Let's quantify this based on our previous Eighth to Quarter Rack example.

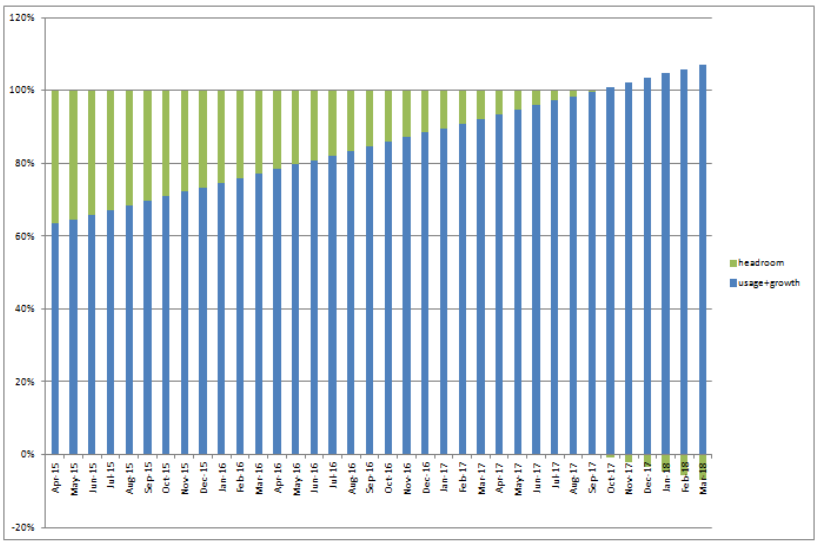
Exadata X5-2 Eighth Rack:

Headroom Calculation:

	ideal usage	max capacity	current usage	12months growth	12months optimization
CPU	75%	100%	62.1%	30%	15%

Headroom Time Calculation:

-2% headroom
-1.7 more months after 1st year to reach ideal usage 75%
OR
10.3 months to reach ideal usage 75% by Feb-16



When we added the ORAPRODC database the cluster-wide CPU resource usage went up to 62.1% and this left us with negative headroom. Let's go through the calculations below:

- Let's say the current month is April 2015
- With the Exadata X5-2 Eighth Rack the maximum capacity is 72 CPUs across two nodes (threads)
- 75% is the ideal cluster-wide CPU utilization which is 54 CPUs (72 x .75)
- 62.1% is the current cluster-wide usage which is 44.72 CPUs (72 x .621)
- The headroom without accounting for the growth and optimization is 12.9%
    - 9.28 CPUs (54 – 44.72)
- Accounting for growth (30%) and optimizations (15%) the end number is 15% growth rate per 12 months
    - Which is 15/12=1.25% rate per month
- The headroom after accounting the growth and optimization is -2.1% (12.9 – 15)
- Take note that if the result is a negative number we don't have enough resources to last for the rest of the year. If it's positive, then we have room to grow for the next year.

Now, the question is what is the **-2.1% headroom**? That is simply the amount of resources we have left, to calculate:

- (Headroom / (Growth – Optimization)) which is -2.1 / 15 = -1.7 years
- Then just convert it to months by doing -1.7 x 12 = 10.3 months
- Which means we have 10.3 more months to reach the 75% ideal cluster-wide CPU utilization

From evaluating the headroom, here we found out that the Exadata X5-2 Eighth Rack is CPU constrained and also have 10.3 months to grow. This triggered us to re-evaluate the capacity model and chose the Exadata X5-2 Quarter Rack. The other option to adding capacity to gain more headroom is to further tune the application. It is the best route and this is the recommended way to remediate capacity. If we can have an optimization rate of 25% then that leaves the end growth rate to just 5% but then this requires change, time, and right skills/expertise. If optimizaition is done successfully it will truly make the platform or environment last longer at less cost.

Below shows the new headroom when we moved to Exadata X5-2 Quarter Rack:

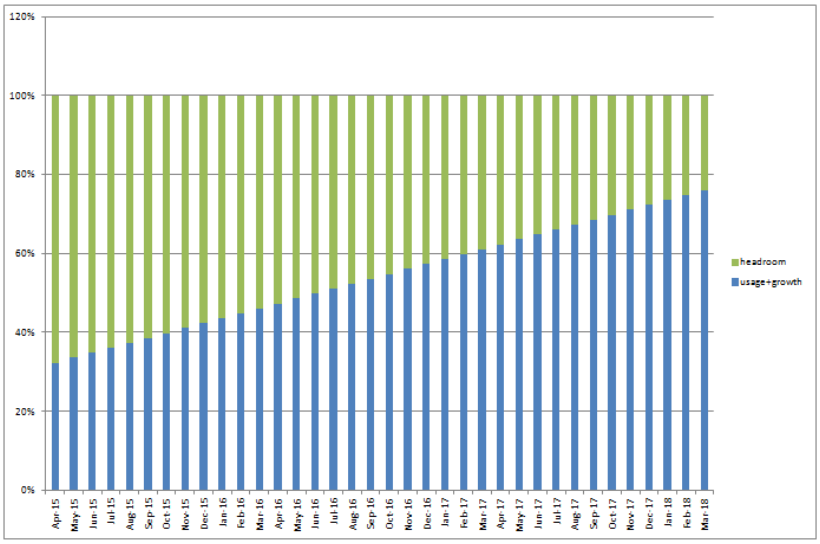
Exadata X5-2 Quarter Rack:

Headroom Calculation:

	ideal usage	max capacity	current usage	12months growth	12months optimization
CPU	75%	100%	31.1%	30%	15%

Headroom Time Calculation:

29% headroom
23.2 more months after 1st year to reach ideal usage 75%
OR
35.2 months to reach ideal usage 75% by Mar-18



By adding CPU capacity the headroom number became positive and resulted to more room to grow for next year. Applying the same headroom calculations (using 144 CPUs) we now have 35.2 more months to reach the 75% ideal cluster-wide CPU utilization.

Note: The idea of Headroom Equation came from the book "The Art of Scalability" by Marty Abbott and Michael Fisher of AFK Partners

---

To summarize, a database consolidation project can be quantified even before it takes place by gathering the performance and configuration data, characterizing the consolidated workload, and modeling the capacity once all the resource requirements are in place. I especially like the modeling exercise because it sets some parameters to the steakholders or customers and it alters their thinking when they become aware that they are bounded with limited amount of resources and we do when we model is set scenarios backed by real numbers coming from the environments. This brings integrity to the whole consolidation and sizing process and the key people involved in the planning and doing the sizing.

Using Resource Manager for Consolidation

On the previous sections we went through the entire database consolidation process and how to properly execute it. There's another piece that's very critical to Consolidation and that is called Resource Management. Most of the production databases we've seen have a more critical workload contending with an adhoc user or a set of resource intensive jobs. It can also be another database from the same cluster. And with the resource contention, comes degraded performance or response time fluctuations. With Resource Management we can limit the resource usage of the non-critical users or databases and ultimately provide resource guarantees which results to predictable response times and overall happier users. The rest of this chapter is devoted to demonstrating all the Resource Management features available for Exadata to ensure the critical resource components continue to provide maximum performance.