# How to Achieve **All in One** with Oracle 11g

Husnu Sensoy

Turkcell Communication Services
husnu.sensoy@gmail.com

October 12, 2009

# Content

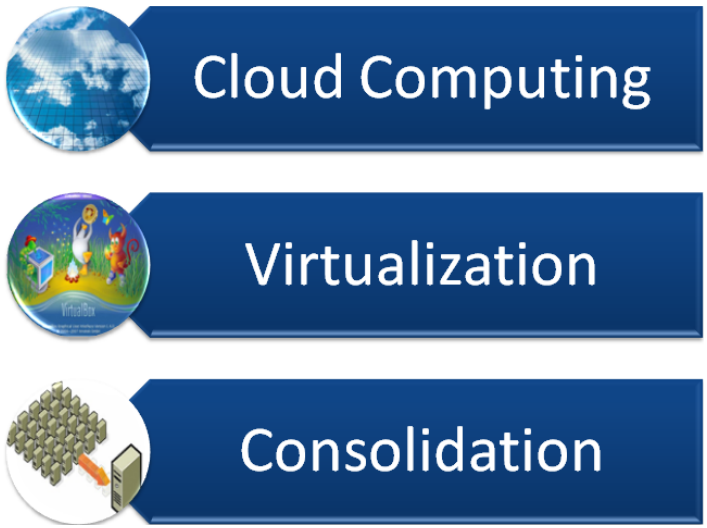


# Who am I ?

-  VLDB Expert
- Oracle ACE
- Oracle DBA of 2009
- Oracle Blogger in ▸ The great grandson of Husnu Sensoy
- Speaker in various meetings like Open World, User Groups, and Universities
- Master of Science Student at  on *I/O Scheduling on Grid Environments*

# What is it all about ?

- All those fancy names actually refer to same basic concept token as *Grid Computing* for years.
- There exists petabyte size clusters used for scientific researches for years, such as PetaShare
- Amdahl's Law was around since 1960s. What is not ready was the server, network and storage technologies.
- By the beginning of 1990s, academy has made the first move for local grids.
- Finally after 2005 (even before), commerce was also ready to GO !!!




# Why to bother ?

- Top 3 IT Trends for 2009 and 2010:
  1. Software as a Service (Cloud Computing)
  2. Virtualization
  3. Energy-Efficient Data Centers

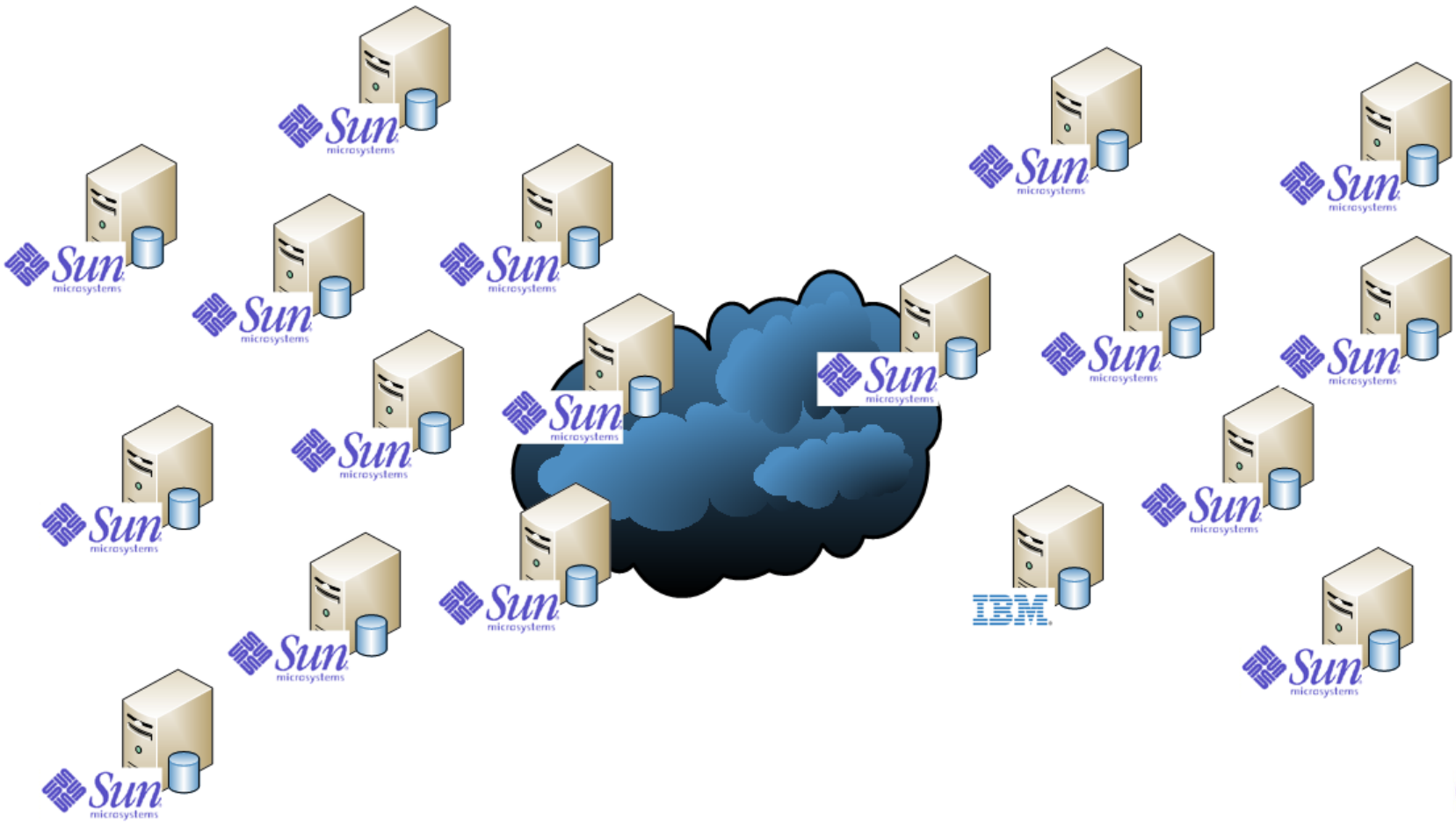
# Why to bother ?

- Top 3 IT Trends for 2009 and 2010:
  1. Software as a Service (Cloud Computing)
  2. Virtualization
  3. Energy-Efficient Data Centers
- Top 5 Priorities of CIOs
  1. Business productivity and cost reduction
  2. IT/Business alignment
  3. Agility and speed to market
  4. IT cost reduction
  5. IT reliability and efficiency

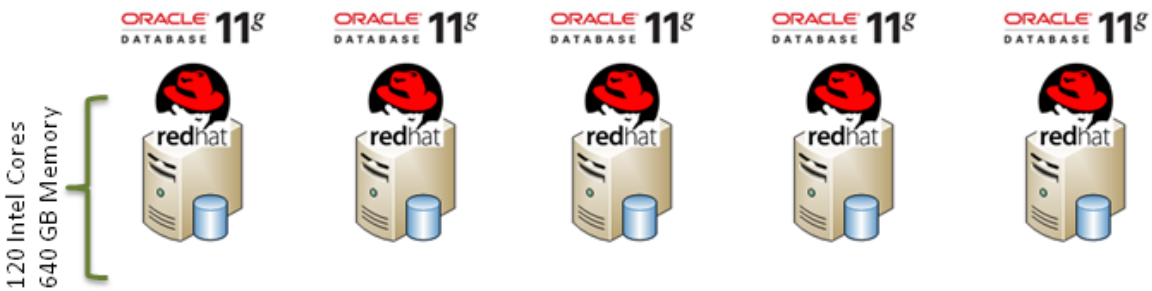
# Database Consolidation in

- A recent research by a IT consultancy company shows that Turkcell data center utilization is below 20% on the average. When we think some major systems,such as data warehouse, run over 80% utilization on the average, this statistics means some servers do nothing.
- Some DBAs in VAS database team were under extreme load. DB to DBA ratio was around 10-15 on the average.
- Yearly OPEX of SMP machines have already started to be a pain in the neck.
- Majority of databases were running on single instance architecture which always endangers our availability.

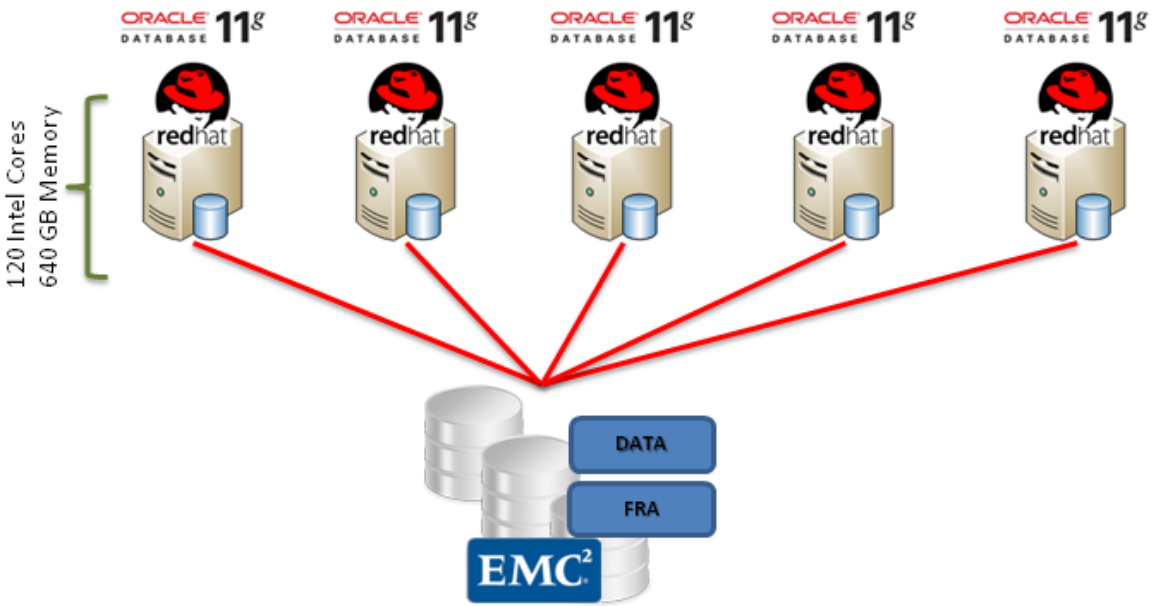
# The Picture Before Allinone in



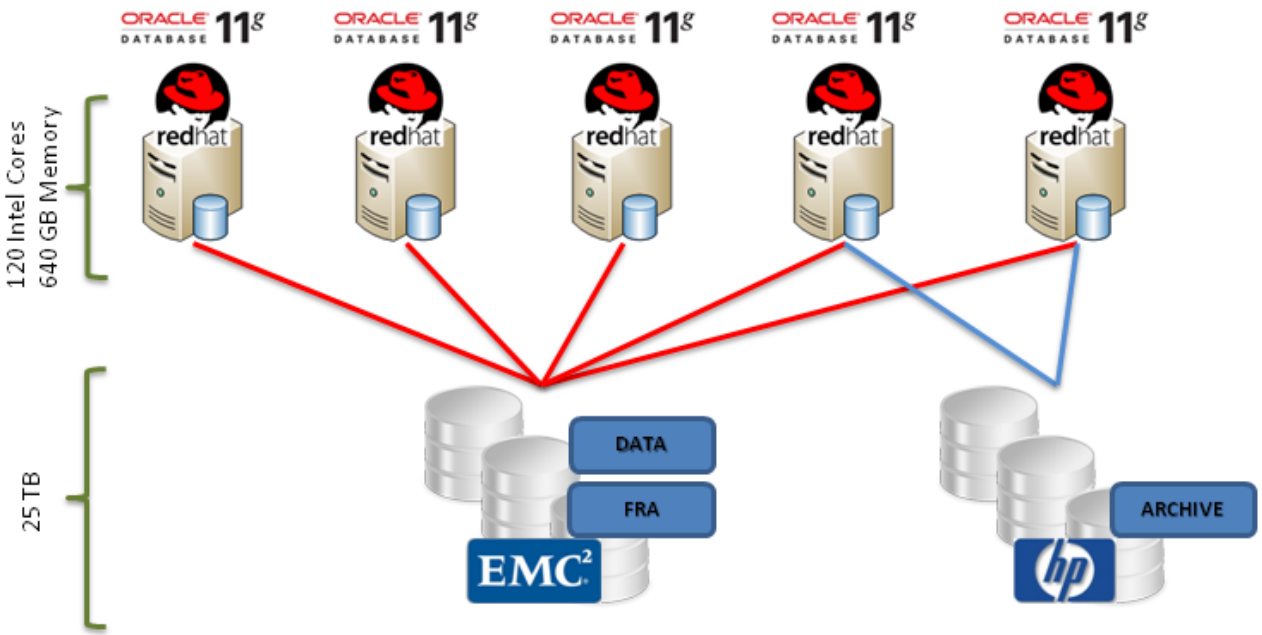
# The Picture After Allinone in



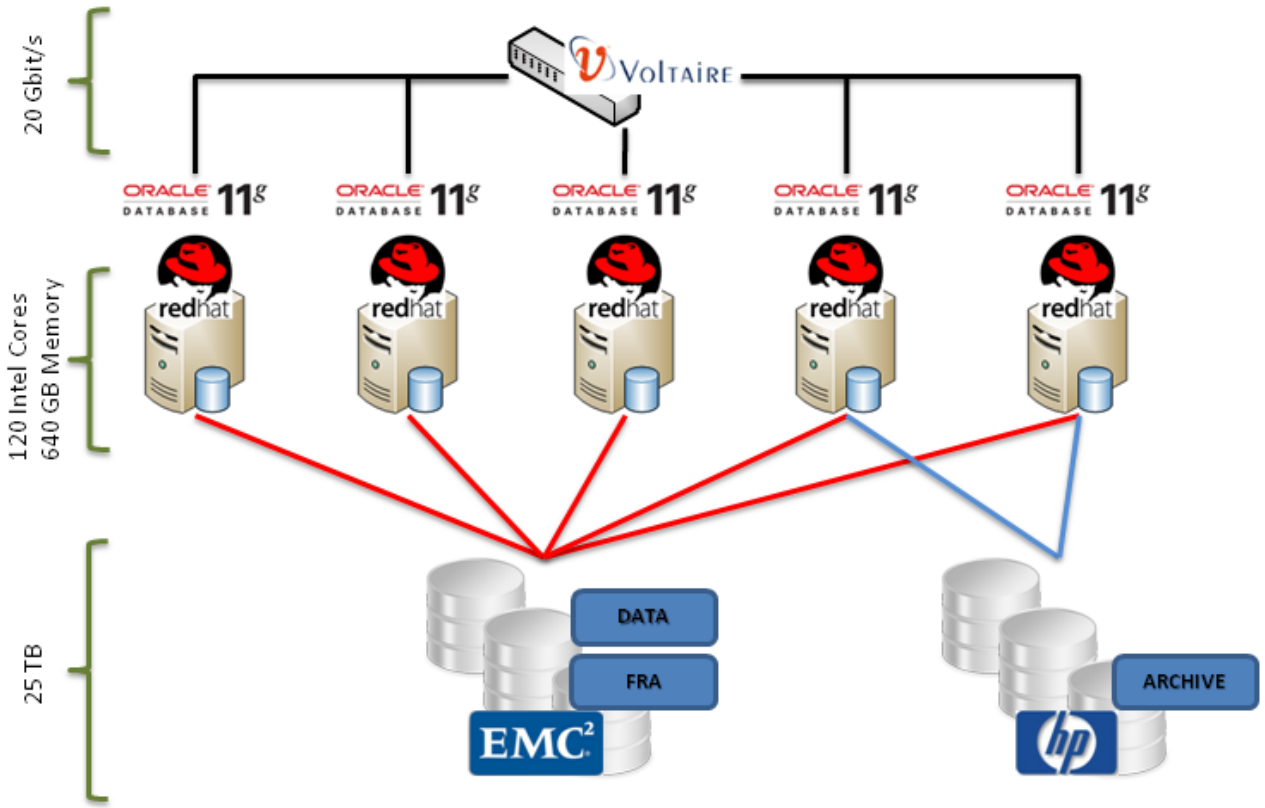
# The Picture After Allinone in



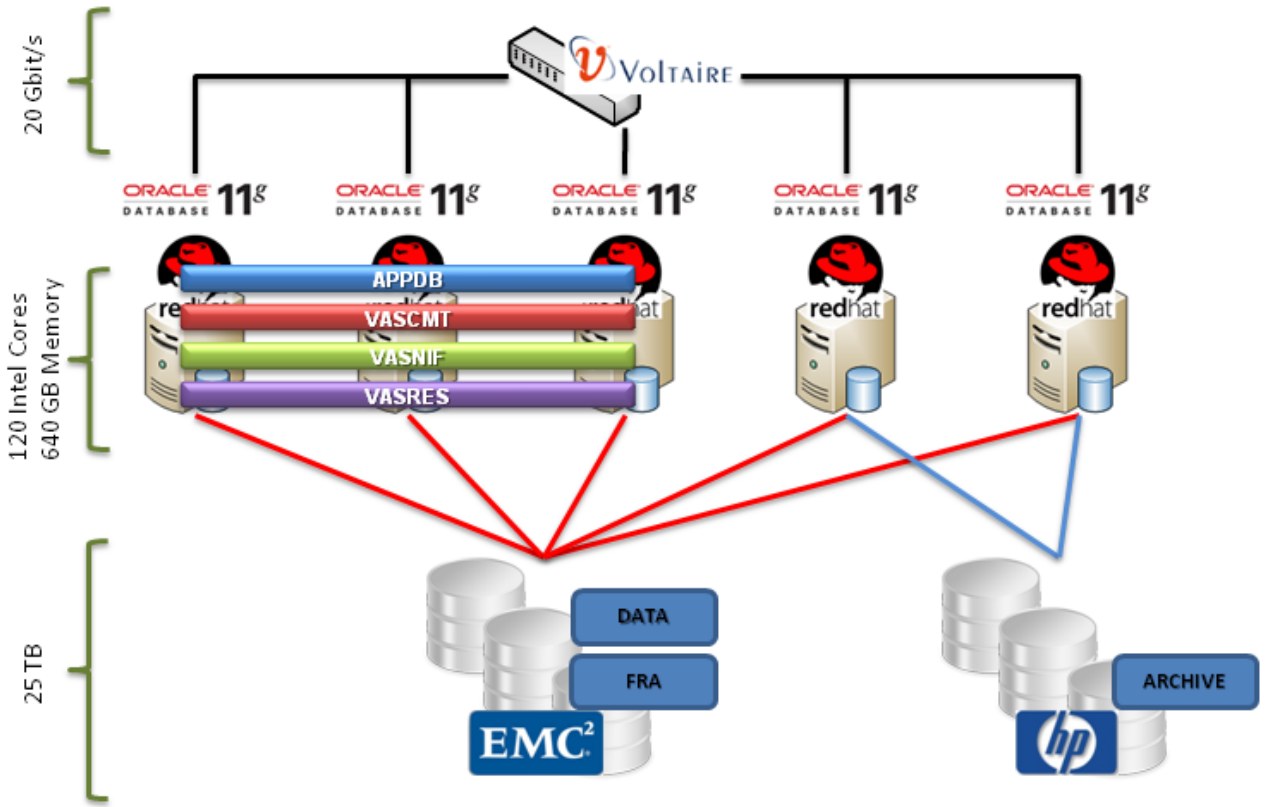
# The Picture After Allinone in



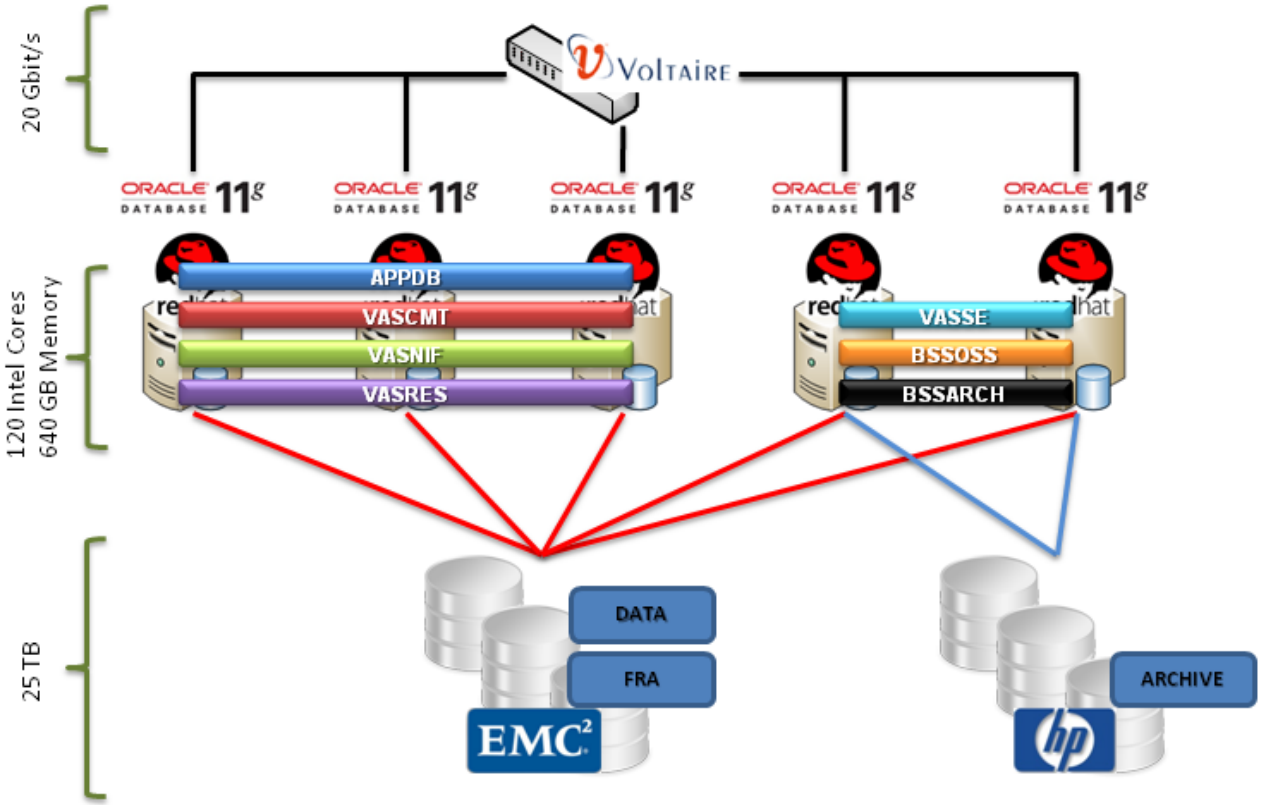
# The Picture After Allinone in



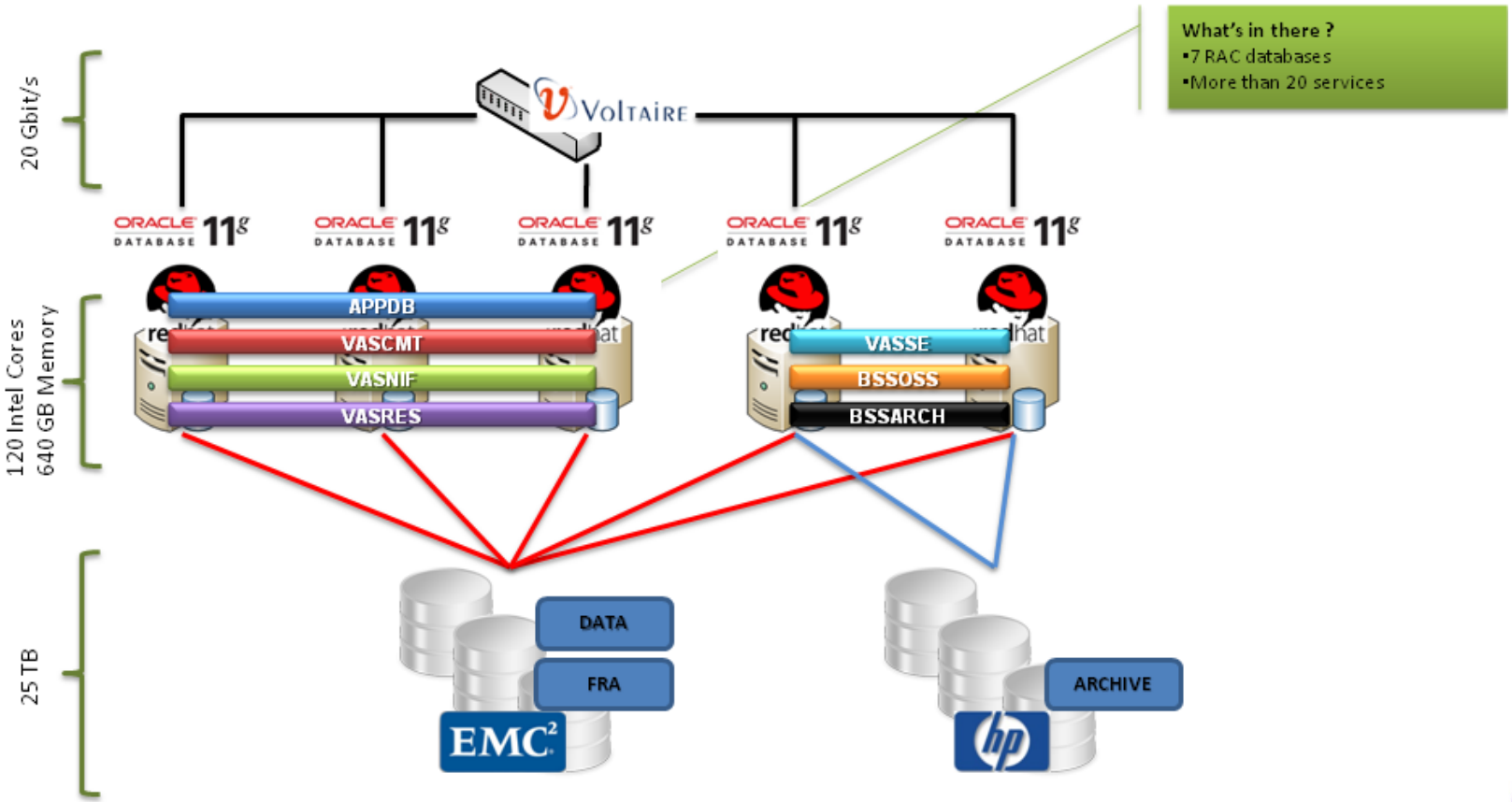
# The Picture After Allinone in



# The Picture After Allinone in



# The Picture After Allinone in



# Candidates to Consolidate

Nobody can give you a silver bullet about this because it is all about your company's need(s). You should first define your goal(s). Why do you consolidate ?

- just to see whether we can
- to increase average server processing,I/O,memory,network,etc. utilization
- to move from SMP machines to commodity hardware
- to renew technology
- to consolidate our data centers
- to reduce our license costs

# Candidates to Consolidate

Nobody can give you a silver bullet about this because it is all about your company's need(s). You should first define your goal(s). Why do you consolidate ?

- just to see whether we can
- to increase average server processing,I/O,memory,network,etc. utilization
- to move from SMP machines to commodity hardware
- to renew technology
- to consolidate our data centers
- to reduce our license costs

**Real Life**

Here are our criterias to choose candidates

- Server processor utilization
- Data Center location
- Application and vendor limitations (version dependencies)
- Dependencies with other systems
- Business streams
- Risk level
- I/O characteristics

# Candidates to Consolidate

After defining your reason(s), if your reason(s) requires the collection of system metrics to define right candidates, you need a repository that contains this data. Never skip this step because this is very crucial for

# Candidates to Consolidate

After defining your reason(s), if your reason(s) requires the collection of system metrics to define right candidates, you need a repository that contains this data. Never skip this step because this is very crucial for

1. Candidate determination

# Candidates to Consolidate

After defining your reason(s), if your reason(s) requires the collection of system metrics to define right candidates, you need a repository that contains this data. Never skip this step because this is very crucial for

1. Candidate determination
2. Initial capacity planning

# Candidates to Consolidate

After defining your reason(s), if your reason(s) requires the collection of system metrics to define right candidates, you need a repository that contains this data. Never skip this step because this is very crucial for

1. Candidate determination
2. Initial capacity planning

**Real Life**

Turkcell has 3 data centers and 100s of database hosts. It takes 3 weeks for us to complete all grid agent installations on missing servers. So keep in mind that for large data centers this task may not be trivial.
We have used Oracle Grid Control repository for CPU and memory metrics and EMC ECC to gather host based disk I/O metrics.

# Candidates to Consolidate

After defining your reason(s), if your reason(s) requires the collection of system metrics to define right candidates, you need a repository that contains this data. Never skip this step because this is very crucial for

1. Candidate determination
2. Initial capacity planning

**Real Life**

Turkcell has 3 data centers and 100s of database hosts. It takes 3 weeks for us to complete all grid agent installations on missing servers. So keep in mind that for large data centers this task may not be trivial.
We have used Oracle Grid Control repository for CPU and memory metrics and EMC ECC to gather host based disk I/O metrics.

You can now choose a superset of candidate list satisfying your requirements. Next step will be to fine-tune the list by performing one-to-one meetings with the DBAs of those systems to ensure nothing is wrong for the consolidation of candidate database.

# Initial Hardware Capacity Planning

After defining your final candidates, next step will be to do a initial capacity planing.

- How many cores, HBAs,etc ?
- What type of interconnect and what capacity ?

Remember that consolidation environment is built to be flexible and horizontal scalable. So you will do only an initial capacity planning. In the next phases of environment you will be adding more building blocks (identical nodes) to your grid.

# General Rule of Thumb

- Use additive linear models in capacity planning because it has been proven statistically that linear models are the most successful ones although they are very simple.

# General Rule of Thumb

- Use additive linear models in capacity planning because it has been proven statistically that linear models are the most successful ones although they are very simple.
- There is no commercial hardware with 17 cores in it. So round up your requirements to the first available commercial hardware.

# General Rule of Thumb

- Use additive linear models in capacity planning because it has been proven statistically that linear models are the most successful ones although they are very simple.
- There is no commercial hardware with 17 cores in it. So round up your requirements to the first available commercial hardware.
- Some components are correlated with each other in commercial hardware, such as CPU cores and memory. So usually it will be sufficient to compute one of them.

# General Rule of Thumb

- Use additive linear models in capacity planning because it has been proven statistically that linear models are the most successful ones although they are very simple.
- There is no commercial hardware with 17 cores in it. So round up your requirements to the first available commercial hardware.
- Some components are correlated with each other in commercial hardware, such as CPU cores and memory. So usually it will be sufficient to compute one of them.
- Always put a 10%-20% error rate for any computation you made.

# Start with capacity planning for CPU

- If your team is not very experienced with Oracle RAC, choose larger machines if possible. For example, instead of 6x8 core Intel machines use 2x24 core Intel machines. Fewer nodes mean simpler grid architecture.
- Use physical cores for Intel, AMD, and old Sun SPARC processors and physical thread numbers for new SPARC cores.Be careful about IBM processors. Their core architectures are design to be faster than other commercial cores. So assume each IBM core as 2 times faster than an Intel/AMD/SPARC core/thread (not valid for Nehalem)

# Start with capacity planning for CPU

- If your team is not very experienced with Oracle RAC, choose larger machines if possible. For example, instead of 6x8 core Intel machines use 2x24 core Intel machines. Fewer nodes mean simpler grid architecture.
- Use physical cores for Intel, AMD, and old Sun SPARC processors and physical thread numbers for new SPARC cores.Be careful about IBM processors. Their core architectures are design to be faster than other commercial cores. So assume each IBM core as 2 times faster than an Intel/AMD/SPARC core/thread (not valid for Nehalem)
- Build a time-based (more granular is better) table of candidate servers' maximum (worst case analysis) CPU utilizations covering last 1-3 months:

| TStamp | S1 (16 UltraSparc) | S2 (8 IBM) | S3 (32 Intel) | Total Intel Need |
|---|---|---|---|---|
| 00:00 | 100% | 40% | 20% | 16x1+8x2x0.4+32x0.2=28.8 |
| 00:05 | 80% | 30% | 100% | 49.6 |
| 00:10 | 10% | 10% | 40% | 16 |
| 00:15 | 40% | 50% | 50% | 30.4 |

# Start with capacity planning for CPU

- If your team is not very experienced with Oracle RAC, choose larger machines if possible. For example, instead of 6x8 core Intel machines use 2x24 core Intel machines. Fewer nodes mean simpler grid architecture.
- Use physical cores for Intel, AMD, and old Sun SPARC processors and physical thread numbers for new SPARC cores.Be careful about IBM processors. Their core architectures are design to be faster than other commercial cores. So assume each IBM core as 2 times faster than an Intel/AMD/SPARC core/thread (not valid for Nehalem)
- Build a time-based (more granular is better) table of candidate servers' maximum (worst case analysis) CPU utilizations covering last 1-3 months:

| TStamp | S1 (16 UltraSparc) | S2 (8 IBM) | S3 (32 Intel) | Total Intel Need |
|---|---|---|---|---|
| 00:00 | 100% | 40% | 20% | 16x1+8x2x0.4+32x0.2=28.8 |
| 00:05 | 80% | 30% | 100% | 49.6 |
| 00:10 | 10% | 10% | 40% | 16 |
| 00:15 | 40% | 50% | 50% | 30.4 |

- In this toy example the CPU need is 55 Intel cores (with 10% error margin). So if your choices are 8,16,24,32 core machines you will start with a 7,4,3, or 2 node RAC implementation.

# Verify Memory Requirement

- Usually memory is one of the abundant components of commodity machines.

# Verify Memory Requirement

- Usually memory is one of the abundant components of commodity machines.

**Real Life**

HP DL 580 nodes we are using is boxed 24 Intel cores with 128 GB of memory which is a pretty large value.

# Verify Memory Requirement

- Usually memory is one of the abundant components of commodity machines.

**Real Life**

HP DL 580 nodes we are using is boxed 24 Intel cores with 128 GB of memory which is a pretty large value.

- One thing to ensure for your cluster is that sum of candidate database hosts' memory (plus 10%) should be less than equal to your grid total memory size.
- One Oracle remark is that before 11g Release 2 PGA memory on RAC is not a virtual memory like SGA that can be shared by cross processes.

# Disk Storage Capacity Planing

Storage requirement has two components

1. Storage area size including RAID,spare,etc
2. Throughput

# Disk Storage Capacity Planing

Storage requirement has two components

1. Storage area size including RAID,spare,etc
2. Throughput

Throughput capacity planing

- Do not mix sequential large I/O load databases (DSS) with the small low latency random I/O demanding ones (OLTP).
- Ensure that all databases dominated by small random I/O you are consolidating have an average I/O response time below 10ms on the average and 20ms at max.
- Keep in mind that no hardware will deliver its spec value in real life. Practically you will get an effective value which is 70%-80% of spec value (For a 4 Gbit HBA your practical throughput average will be 280-320 MB/s at peak).
- If you are consolidating OLTP environments 3 HBAs per node will be sufficient (2 HBA for SAN storage and 1 for backup SAN)

# Disk Storage Capacity Planing

Storage requirement has two components

1. Storage area size including RAID,spare,etc
2. Throughput

Throughput capacity planing

- Do not mix sequential large I/O load databases (DSS) with the small low latency random I/O demanding ones (OLTP).
- Ensure that all databases dominated by small random I/O you are consolidating have an average I/O response time below 10ms on the average and 20ms at max.
- Keep in mind that no hardware will deliver its spec value in real life. Practically you will get an effective value which is 70%-80% of spec value (For a 4 Gbit HBA your practical throughput average will be 280-320 MB/s at peak).
- If you are consolidating OLTP environments 3 HBAs per node will be sufficient (2 HBA for SAN storage and 1 for backup SAN)

Storage size capacity planing

- If you like to use Incrementally Updated Backup strategy, which I strongly recommend, remember to double (indeed slightly more) your storage requirement.

# Confirm Your Capacity Plan(Verification)

Any capacity plan may fail due to various reasons. Such as, assumptions, hardware spec errors,configuration errors,etc. You should ensure that your capacity planing is correct. This is made up of two steps :

# Confirm Your Capacity Plan(Verification)

Any capacity plan may fail due to various reasons. Such as, assumptions, hardware spec errors,configuration errors,etc. You should ensure that your capacity planing is correct. This is made up of two steps :

Verification is the question of "Do I have the right capacity ?"

Benchmark/Stress testing tools are your best friends, such as ORION for disk I/O, uperf for network, or rds-stress for RDSoIB.

# Confirm Your Capacity Plan(Verification)

Any capacity plan may fail due to various reasons. Such as, assumptions, hardware spec errors,configuration errors,etc. You should ensure that your capacity planing is correct. This is made up of two steps :

Verification is the question of "Do I have the right capacity ?"

Benchmark/Stress testing tools are your best friends, such as ORION for disk I/O, uperf for network, or rds-stress for RDSoIB.

**Real Life**

In our grid, we require 8000 IOPS with a latency of 10ms on the average. In order to deliver this value, our storage administrator has changed the SAN configuration 6 times. Without verification process there is no way to catch it.

# Confirm Your Capacity Plan(Validation)

Validation is the question of "Is this capacity sufficient for my grid ?"

# Confirm Your Capacity Plan(Validation)

Validation is the question of "Is this capacity sufficient for my grid ?"

Before 11g Release 1, capacity planing validation was done by using hammerora like tools. But now Oracle gives us very strong tool called Database Replay

# Confirm Your Capacity Plan(Validation)

Validation is the question of "Is this capacity sufficient for my grid ?"

Before 11g Release 1, capacity planing validation was done by using hammerora like tools. But now Oracle gives us very strong tool called Database Replay

## Real Life

For validation, here is the steps we followed in our project:

1. Capture the load in candidate databases (18 10g databases) between time T1-T2.
2. Perform a point in time recovery of those databases on grid (T1)
3. Simultaneously start all 18 replays over cluster.
4. Check for AWR reports of all 18 databases for any problem.

# Differentiation of Linux Users

In a consolidated database environment separation of roles gain more importance.

- Do you want a DBA of a DB on cluster to shut down ASM instance on one of the nodes causing all other DB instances to shut down?
- Do you want a DBA to shut down the listener of a DB he/she is not responsible with on the cluster.
- Do you want an interim patch/patchset to be applied to affect all databases at a time ?

Not to face with outcomes of those scenarios, separating required OS users,groups and ORACLE_HOME directories are essential

# Differentiation of Linux Users

In a consolidated database environment separation of roles gain more importance.

- Do you want a DBA of a DB on cluster to shut down ASM instance on one of the nodes causing all other DB instances to shut down?
- Do you want a DBA to shut down the listener of a DB he/she is not responsible with on the cluster.
- Do you want an interim patch/patchset to be applied to affect all databases at a time ?

Not to face with outcomes of those scenarios, separating required OS users,groups and ORACLE_HOME directories are essential

**Real Life**

Here is the list of Oracle related users of our cluster

| User | Primary Group | Other Groups | ORACLE_HOME |
|---|---|---|---|
| crs | oinstall | dba,asmadmin | /u01/app/crs/product/11.1.0 |
| oracle | oinstall | dba,oper,dbasm | /u01/app/grid/product/10.2.0/agent10g |
| asm | oinstall | dba,asmadmin,oper | /u01/app/asm/product/11.1.0 |
| vasres | oinstall | dba,oper,dbasm | /u02/app/oracle/product/11.1.0 |
| vascmt | oinstall | dba,oper,dbasm | /u03/app/oracle/product/11.1.0 |
| appdb | oinstall | dba,oper,dbasm | /u05/app/oracle/product/11.1.0 |

# Verify your grid's HA with Crash Tests

Any hang resolution/bonding/multipathing mechanism relies on some timeouts. Default timeout values may not be compliant with Oracle CRS timeout values sometimes.

# Verify your grid's HA with Crash Tests

Any hang resolution/bonding/multipathing mechanism relies on some timeouts. Default timeout values may not be compliant with Oracle CRS timeout values sometimes.

**Real Life**

Default timeout value of Voltaire IB switches we use is too high. With default values if you lose the interconnect connection going over the active switch, CRS will reboot all but one nodes before switch fail-over is completed

You should check the expected behaviour of your grid in case that following events occur before going to production

- Node/Instance/ASM Instance/Listener Failure
- CRS process failure
- Public network/NIC failure
- Interconnect network/NIC/switch failure
- Voting Disk Access Loss
- and many others

# Verify your grid's HA with Crash Tests

Any hang resolution/bonding/multipathing mechanism relies on some timeouts. Default timeout values may not be compliant with Oracle CRS timeout values sometimes.

**Real Life**

Default timeout value of Voltaire IB switches we use is too high. With default values if you lose the interconnect connection going over the active switch, CRS will reboot all but one nodes before switch fail-over is completed

You should check the expected behaviour of your grid in case that following events occur before going to production

- Node/Instance/ASM Instance/Listener Failure
- CRS process failure
- Public network/NIC failure
- Interconnect network/NIC/switch failure
- Voting Disk Access Loss
- and many others

**Real Life**

I have spent 6 hours with our system architect in our system room to complete all tests comprehensively.

# MEMORY_TARGET vs SGA_TARGET

- If the memory of a single node is pretty high in your grid then default memory page size (4K in 2.6) cause millions of memory pages to be managed by OS kernel(TLB management) which in return increases kernel CPU utilization. This problem can be bypassed by the use of huge pages in Linux.

# MEMORY_TARGET vs SGA_TARGET

- If the memory of a single node is pretty high in your grid then default memory page size (4K in 2.6) cause millions of memory pages to be managed by OS kernel(TLB management) which in return increases kernel CPU utilization. This problem can be bypassed by the use of huge pages in Linux.
- By 11g Release 1, Oracle has leveraged its memory management one step further with new MEMORY_TARGET parameter which allows you to let Oracle do whole memory management automatically.

# MEMORY_TARGET vs SGA_TARGET

- If the memory of a single node is pretty high in your grid then default memory page size (4K in 2.6) cause millions of memory pages to be managed by OS kernel(TLB management) which in return increases kernel CPU utilization. This problem can be bypassed by the use of huge pages in Linux.
- By 11g Release 1, Oracle has leveraged its memory management one step further with new MEMORY_TARGET parameter which allows you to let Oracle do whole memory management automatically.
- However this new architecture does not support huge pages (Metalink Note 749851.1). So it is still better to use good old SGA_TARGET on Linux platforms with lots of memory.

# MEMORY_TARGET vs SGA_TARGET

- If the memory of a single node is pretty high in your grid then default memory page size (4K in 2.6) cause millions of memory pages to be managed by OS kernel(TLB management) which in return increases kernel CPU utilization. This problem can be bypassed by the use of huge pages in Linux.
- By 11g Release 1, Oracle has leveraged its memory management one step further with new MEMORY_TARGET parameter which allows you to let Oracle do whole memory management automatically.
- However this new architecture does not support huge pages (Metalink Note 749851.1). So it is still better to use good old SGA_TARGET on Linux platforms with lots of memory.
- You can enable Huge Pages by setting *vm.nr_hugepages*. This parameter defines the number of total huge pages available.

# MEMORY_TARGET vs SGA_TARGET

- If the memory of a single node is pretty high in your grid then default memory page size (4K in 2.6) cause millions of memory pages to be managed by OS kernel(TLB management) which in return increases kernel CPU utilization. This problem can be bypassed by the use of huge pages in Linux.
- By 11g Release 1, Oracle has leveraged its memory management one step further with new MEMORY_TARGET parameter which allows you to let Oracle do whole memory management automatically.
- However this new architecture does not support huge pages (Metalink Note 749851.1). So it is still better to use good old SGA_TARGET on Linux platforms with lots of memory.
- You can enable Huge Pages by setting *vm.nr_hugepages*. This parameter defines the number of total huge pages available.

**Real Life**

We have set 20000 (40GB) huge pages per node on our grid. This decreases the number of allocated memory pages from 32 M to 64K. Monitor your huge page status by

```
#  cat /proc/meminfo | grep Huge
HugePages_Total: 20000
HugePages_Free:  10131
HugePages_Rsvd:   2538
Hugepagesize:     2048 kB
```

# How much to overlap ?

Oracle supports multiple instances to overlap over a single node, just like in our architecture. In theory there is no limit for the number of instances to be overlapped over a node as long as there are sufficient resources.
But indeed there are practical inhibitors

# How much to overlap ?

Oracle supports multiple instances to overlap over a single node, just like in our architecture. In theory there is no limit for the number of instances to be overlapped over a node as long as there are sufficient resources.
But indeed there are practical inhibitors

**Real-time Oracle Processes** There are critical Oracle processes like LMSn which run in real-time priority. Those processes use the benefit of pre-emptive capabilities of OS process scheduler to use CPU whenever they like. However if the number of cores are less than the total number of real-time processes then you will be damaging the real-time behaviour of those processes.

# How much to overlap ?

Oracle supports multiple instances to overlap over a single node, just like in our architecture. In theory there is no limit for the number of instances to be overlapped over a node as long as there are sufficient resources.
But indeed there are practical inhibitors

**Real-time Oracle Processes** There are critical Oracle processes like LMSn which run in real-time priority. Those processes use the benefit of pre-emptive capabilities of OS process scheduler to use CPU whenever they like. However if the number of cores are less than the total number of real-time processes then you will be damaging the real-time behaviour of those processes.

**Management Difficulties** Oracle does not provide a mechanism to consolidate AWR,ASH,etc reports of separate instances on the same node. In case of a problem you should manually understand which DB is the source of problem by looking each and every report manually.

# Install ORACLE_HOME to all nodes

- For various benefits like rolling patching, use non-shared Oracle Home architecture in your grid.
- Due to some space considerations, you might wish to install RDBMS ORACLE_HOMEs just to the nodes on which member instances will run.However installing all RDBMS homes on all nodes have various benefits.

# Install ORACLE_HOME to all nodes

- For various benefits like rolling patching, use non-shared Oracle Home architecture in your grid.
- Due to some space considerations, you might wish to install RDBMS ORACLE_HOMEs just to the nodes on which member instances will run.However installing all RDBMS homes on all nodes have various benefits.

**Advantages**

1. It will give you the flexibility to move any database to any node you like on the grid immediately (you will not be waiting for the storage to be provisioned for new ORACLE_HOME instance).
2. Remember that, simple architectures are the best ones. By installing everything to everywhere you will obtain a homogeneous view in your cluster which certainly ease grid management.

# Install ORACLE_HOME to all nodes

- For various benefits like rolling patching, use non-shared Oracle Home architecture in your grid.
- Due to some space considerations, you might wish to install RDBMS ORACLE_HOMEs just to the nodes on which member instances will run.However installing all RDBMS homes on all nodes have various benefits.

## Advantages

1. It will give you the flexibility to move any database to any node you like on the grid immediately (you will not be waiting for the storage to be provisioned for new ORACLE_HOME instance).
2. Remember that, simple architectures are the best ones. By installing everything to everywhere you will obtain a homogeneous view in your cluster which certainly ease grid management.

## Disadvantages

1. You will need an extra storage ( around 2.5-3GB) for each new home you install.
2. Installation/Upgrade tasks will take slightly more time because you have more homes to be install/patched.

# OS Level Monitoring

- OS level system statistics gathering is extremely important in a grid system because of overlapping database instances.
- Another importance of OS level statistics are that they are the guidelines of your future capacity planning.
- OSWatcher 3.0 is one the best options you can take (Last version also gathers ▸ IB and ▸ RDS statistics).

# Red Hat Linux vs Oracle Enterprise Linux

- Obviously in terms of software you get, Oracle Enterprise Linux and Red Hat Linux are almost the same.
- What makes difference is the support you get (Red Hat Support or Unbreakable Oracle Support).
- Both support teams are good only if you have a direct access. But keep in mind that in many countries,as in Turkey, you get Red Hat Linux support over hardware vendor which might be very ineffective.

# Red Hat Linux vs Oracle Enterprise Linux

- Obviously in terms of software you get, Oracle Enterprise Linux and Red Hat Linux are almost the same.
- What makes difference is the support you get (Red Hat Support or Unbreakable Oracle Support).
- Both support teams are good only if you have a direct access. But keep in mind that in many countries,as in Turkey, you get Red Hat Linux support over hardware vendor which might be very ineffective.

**Real Life**

Due to a bug in OFED stack (IB protocol stack) our system were constantly crashing until we reached correct people in Red Hat Support over HP.

# Red Hat Linux vs Oracle Enterprise Linux

- Obviously in terms of software you get, Oracle Enterprise Linux and Red Hat Linux are almost the same.
- What makes difference is the support you get (Red Hat Support or Unbreakable Oracle Support).
- Both support teams are good only if you have a direct access. But keep in mind that in many countries,as in Turkey, you get Red Hat Linux support over hardware vendor which might be very ineffective.

**Real Life**

Due to a bug in OFED stack (IB protocol stack) our system were constantly crashing until we reached correct people in Red Hat Support over HP.

- Remember that working with minimum number of vendors will really increase your solution speed in case of a problem.

# ASM Configuration

- Without any doubt ASM is your practically only option for such a cluster because of its tight integration with CRS, and many management benefits.

# ASM Configuration

- Without any doubt ASM is your practically only option for such a cluster because of its tight integration with CRS, and many management benefits.
- Create your ASM diskgroups explicitly with compatibility level 11.1 to get the benefits like variable extent sizing. If you don't specify anything, ASM will create with compatibility 10.2 by default in 11g Release 1.

# ASM Configuration

- Without any doubt ASM is your practically only option for such a cluster because of its tight integration with CRS, and many management benefits.
- Create your ASM diskgroups explicitly with compatibility level 11.1 to get the benefits like variable extent sizing. If you don't specify anything, ASM will create with compatibility 10.2 by default in 11g Release 1.
- Set PROCESSES parameter for ASM instances above 100 because you will have more than one database served by them.

# ASM Configuration

- Without any doubt ASM is your practically only option for such a cluster because of its tight integration with CRS, and many management benefits.
- Create your ASM diskgroups explicitly with compatibility level 11.1 to get the benefits like variable extent sizing. If you don't specify anything, ASM will create with compatibility 10.2 by default in 11g Release 1.
- Set PROCESSES parameter for ASM instances above 100 because you will have more than one database served by them.
- Yet another important issue is the provisioning of diskgroups to grid nodes. If your configuration consists of more than one storage tier, you might wish to isolate some diskgroups from some nodes. You can do this by simply setting ASM_DISKGROUPS parameter at instance level.

# ASM Configuration

- Without any doubt ASM is your practically only option for such a cluster because of its tight integration with CRS, and many management benefits.
- Create your ASM diskgroups explicitly with compatibility level 11.1 to get the benefits like variable extent sizing. If you don't specify anything, ASM will create with compatibility 10.2 by default in 11g Release 1.
- Set PROCESSES parameter for ASM instances above 100 because you will have more than one database served by them.
- Yet another important issue is the provisioning of diskgroups to grid nodes. If your configuration consists of more than one storage tier, you might wish to isolate some diskgroups from some nodes. You can do this by simply setting ASM_DISKGROUPS parameter at instance level.

**Real Life**

Here is the ASM instance configuration in our grid configuration

| | +ASM1 | +ASM2 | +ASM3 | +ASM4 | +ASM5 |
|---|---|---|---|---|---|
| processes | 200 | 200 | 200 | 200 | 200 |
| asm_diskgroups | DATA,FRA | DATA,FRA | DATA,FRA | ARCHIVE,FRA | ARCHIVE,FRA |

# Backup & Recovery Strategy

Backup & Recovery best practices of such a grid requires a dedicated presentation.

**Real Life**

Just to stress how important it is, as the DB architect of the system, B&R guide was the first document I have written throughout the project.

# Backup & Recovery Strategy

Backup & Recovery best practices of such a grid requires a dedicated presentation.

**Real Life**

Just to stress how important it is, as the DB architect of the system, B&R guide was the first document I have written throughout the project.

For more details let's meet tommorow (on 13th of October) in Moscone South Room 104 at 14:30 for

- Backup & Recovery Best Practices in 11g

# Merging Databases Into Databases

As we have already discussed, the number of separate databases can not be defined by an arbitrary number. This means you should decrease the number of databases as much as you can.

# Merging Databases Into Databases

As we have already discussed, the number of separate databases can not be defined by an arbitrary number. This means you should decrease the number of databases as much as you can.

## Advantages

1. You will significantly increase the number of databases that can be consolidated.
2. Decreasing the number of individual databases will reduce your administrative costs (less backup,performance tuning,etc)
3. Starting merged databases as services in their new environment will allow you to manage your grid easily .

# Merging Databases Into Databases

As we have already discussed, the number of separate databases can not be defined by an arbitrary number. This means you should decrease the number of databases as much as you can.

## Advantages

1. You will significantly increase the number of databases that can be consolidated.
2. Decreasing the number of individual databases will reduce your administrative costs (less backup,performance tuning,etc)
3. Starting merged databases as services in their new environment will allow you to manage your grid easily .

## Disadvantages

1. Merge operation is usually not a trivial one. You should deal with parameter file merge, conflict management,etc.
2. You should scan all your application domain and may need to synchronize some applications with the new architecture,for example schema name changes.

# How to Migrate ?

Oracle gives many tools to be used for migration of a database, such as

- Transportable Tablespaces
- Physical/Logical Standby
- Even DB Link

When we look the consolidation migration nature for a few TB (upto 5-6 TB) databases use Data Pump (in parallel) definitely

# How to Migrate ?

Oracle gives many tools to be used for migration of a database, such as

- Transportable Tablespaces
- Physical/Logical Standby
- Even DB Link

When we look the consolidation migration nature for a few TB (upto 5-6 TB) databases use Data Pump (in parallel) definitely

**Advantages**

1. Platform difference management is transparent in data pump.
2. Version difference management is easy in data pump.
3. Data pump can run in parallel.
4. For schema conflict management data pump is the best option.
5. Learning curve for data pump is not though.

# How to Migrate ?

Oracle gives many tools to be used for migration of a database, such as

- Transportable Tablespaces
- Physical/Logical Standby
- Even DB Link

When we look the consolidation migration nature for a few TB (upto 5-6 TB) databases use Data Pump (in parallel) definitely

**Advantages**

1. Platform difference management is transparent in data pump.
2. Version difference management is easy in data pump.
3. Data pump can run in parallel.
4. For schema conflict management data pump is the best option.
5. Learning curve for data pump is not though.

**Disadvantages**

1. You will need an auxiliary storage to keep dmp files
2. Migration with data pump should be OFFLINE

# Merge & Conflict Resolution

It is almost inevitable that some global database object will be conflicting during the database merge operation. Most common ones are

- Schemas and their items
- Public Synonyms
- Public DB Links
- Tablespaces

# Merge & Conflict Resolution

It is almost inevitable that some global database object will be conflicting during the database merge operation. Most common ones are

- Schemas and their items
- Public Synonyms
- Public DB Links
- Tablespaces

Fortunately, Data Pump REMAP feature will help you to handle schema conflicts by automatically renaming transferred schema

```
# expdp system SCHEMAS=hr
DIRECTORY=dpump_dir1
DUMPFILE=hr.dmp
```

```
# impdp system DIRECTORY=dpump_dir1
DUMPFILE=hr.dmp
REMAP_SCHEMA=hr:scott
```

# Merge & Conflict Resolution

It is almost inevitable that some global database object will be conflicting during the database merge operation. Most common ones are

- Schemas and their items
- Public Synonyms
- Public DB Links
- Tablespaces

Fortunately, Data Pump REMAP feature will help you to handle schema conflicts by automatically renaming transferred schema

```
# expdp system SCHEMAS=hr
DIRECTORY=dpump_dir1
DUMPFILE=hr.dmp
```

```
# impdp system DIRECTORY=dpump_dir1
DUMPFILE=hr.dmp
REMAP_SCHEMA=hr:scott
```

In 11g Release 1, Data Pump can map the followings which will help a lot during migration

REMAP_DATA is very useful in regenerating conflicting primary keys of tables during migration.

# Merge & Conflict Resolution

It is almost inevitable that some global database object will be conflicting during the database merge operation. Most common ones are

- Schemas and their items
- Public Synonyms
- Public DB Links
- Tablespaces

Fortunately, Data Pump REMAP feature will help you to handle schema conflicts by automatically renaming transferred schema

```
# expdp system SCHEMAS=hr
DIRECTORY=dpump_dir1
DUMPFILE=hr.dmp
```

```
# impdp system DIRECTORY=dpump_dir1
DUMPFILE=hr.dmp
REMAP_SCHEMA=hr:scott
```

In 11g Release 1, Data Pump can map the followings which will help a lot during migration

REMAP_DATA is very useful in regenerating conflicting primary keys of tables during migration.

REMAP_TABLE is very useful to recover from conflicting tables of the same schema

# Merge & Conflict Resolution

It is almost inevitable that some global database object will be conflicting during the database merge operation. Most common ones are

- Schemas and their items
- Public Synonyms
- Public DB Links
- Tablespaces

Fortunately, Data Pump REMAP feature will help you to handle schema conflicts by automatically renaming transferred schema

```
# expdp system SCHEMAS=hr
DIRECTORY=dpump_dir1
DUMPFILE=hr.dmp
```

```
# impdp system DIRECTORY=dpump_dir1
DUMPFILE=hr.dmp
REMAP_SCHEMA=hr:scott
```

In 11g Release 1, Data Pump can map the followings which will help a lot during migration

REMAP_DATA is very useful in regenerating conflicting primary keys of tables during migration.

REMAP_TABLE is very useful to recover from conflicting tables of the same schema

REMAP_TABLESPACE is very useful to rename conflicting tablespace names of different databases.

# Merge & Parameter File Aggregation

Once you start to merge multiple databases into one database, yet another task is to appropriately set additive server parameters of new RDBMS, such as memory and process related parameters. While setting those parameters use the following basic strategy

# Merge & Parameter File Aggregation

Once you start to merge multiple databases into one database, yet another task is to appropriately set additive server parameters of new RDBMS, such as memory and process related parameters. While setting those parameters use the following basic strategy

1. Sum parameter value of all migrated databases

# Merge & Parameter File Aggregation

Once you start to merge multiple databases into one database, yet another task is to appropriately set additive server parameters of new RDBMS, such as memory and process related parameters. While setting those parameters use the following basic strategy

1. Sum parameter value of all migrated databases
2. Divide the sum to the minimum number of preferred nodes. This will let your system to run smoothly even that some nodes are not available.

# Merge & Parameter File Aggregation

Once you start to merge multiple databases into one database, yet another task is to appropriately set additive server parameters of new RDBMS, such as memory and process related parameters. While setting those parameters use the following basic strategy

1. Sum parameter value of all migrated databases
2. Divide the sum to the minimum number of preferred nodes. This will let your system to run smoothly even that some nodes are not available.

**Real Life**

In our project, sum of SESSIONS parameter of 5 databases to be consolidated in a 3-node RAC was 7000. Instead of setting *.SESSIONS parameter (per instance) to 2300 (7000/3) we have chosen 3500 (7000/2) to ensure that we will not face any *ORA-00018: maximum number of sessions exceeded* error when we shutdown one RAC instance.

# Merge & Parameter File Aggregation

Once you start to merge multiple databases into one database, yet another task is to appropriately set additive server parameters of new RDBMS, such as memory and process related parameters. While setting those parameters use the following basic strategy

1. Sum parameter value of all migrated databases
2. Divide the sum to the minimum number of preferred nodes. This will let your system to run smoothly even that some nodes are not available.

**Real Life**

In our project, sum of SESSIONS parameter of 5 databases to be consolidated in a 3-node RAC was 7000. Instead of setting *.SESSIONS parameter (per instance) to 2300 (7000/3) we have chosen 3500 (7000/2) to ensure that we will not face any *ORA-00018: maximum number of sessions exceeded* error when we shutdown one RAC instance.

Setting some of those parameters require a modification at OS site also.

**Real Life**

There are 4 database instances (plus ASM instance) on a single node with 1700,2850,2200,3175 (200) PROCESSES parameters. In order to Linux to create and manage that many processes we have modified *kernel.sem* to an appropriate value:

```
# sysctl -w kernel.sem="5010 641280 5010 128"
```

# Merge & Grouping Strategy

## Real Life

We have merged 16 databases to 4 databases in our consolidation projects. And start each migrated database as preferred RAC service on three nodes. In grouping different databases the grouping strategy was based on business streams. Databases of similar VAS consolidated within same database.

# Other Migration Best Practices

- Create all databases by following above best practices without starting any migration. This is crucial because most probably you will not be able to concentrate on fine details during migration chaos.
- Choose an incremental model for migration because
  - This will protect you in case that you hit a serious hardware/software problem.
  - This will help you to gradually check your initial capacity plan.

# Other Migration Best Practices

- Create all databases by following above best practices without starting any migration. This is crucial because most probably you will not be able to concentrate on fine details during migration chaos.
- Choose an incremental model for migration because
  - This will protect you in case that you hit a serious hardware/software problem.
  - This will help you to gradually check your initial capacity plan.

**Real Life**

We have faced with the OFED bug after the migration of 5th database. We rolled the last database migrated back and suspend the migration schedule until the problem has been addressed.

# Other Migration Best Practices

- Create all databases by following above best practices without starting any migration. This is crucial because most probably you will not be able to concentrate on fine details during migration chaos.
- Choose an incremental model for migration because
  - This will protect you in case that you hit a serious hardware/software problem.
  - This will help you to gradually check your initial capacity plan.

**Real Life**

We have faced with the OFED bug after the migration of 5th database. We rolled the last database migrated back and suspend the migration schedule until the problem has been addressed.

- Prioritize your migration. Start with non-critical candidates and end with critical ones.

# Motto for Rookie Consolidators

You never know how long the battle is going to be,
You have to be ever ready

▸ Steven Seagal


Q&A