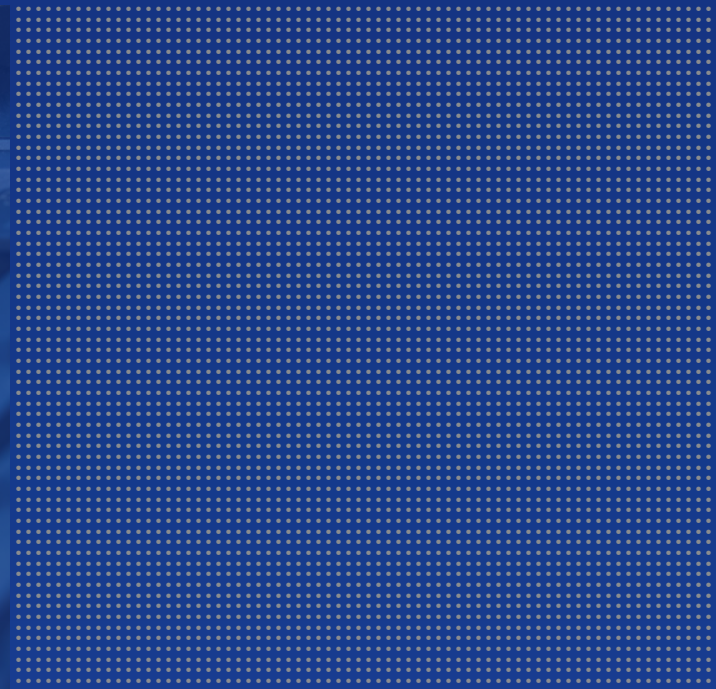


A R G E N T
ENCYCLOPEDIA

Oracle RAC Monitoring



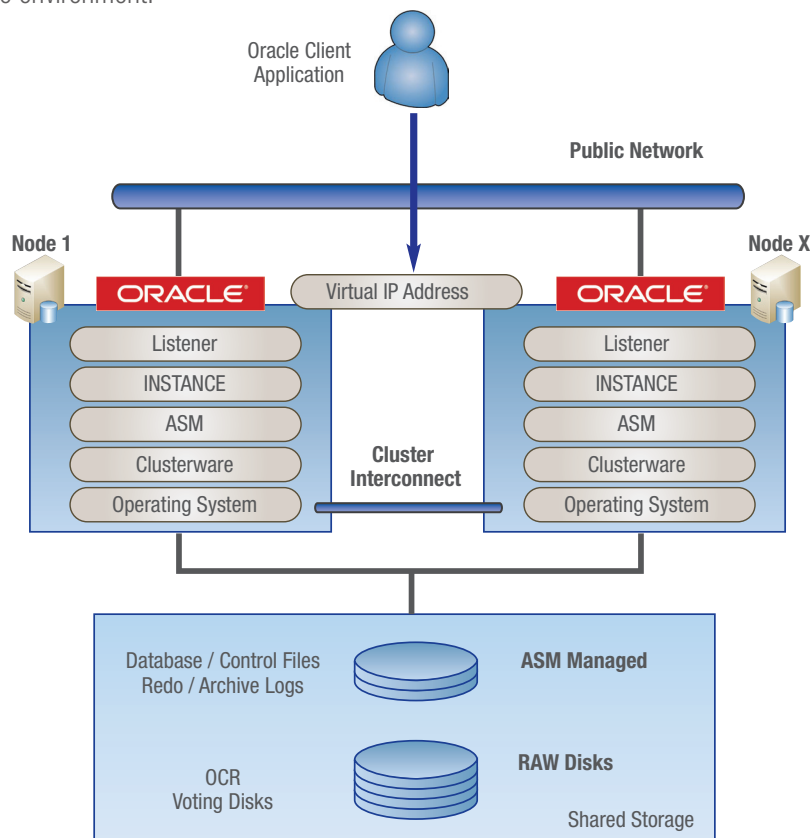
Contents

Argent Monitor For Oracle Monitoring Overview	3
Oracle RAC Monitoring	3
What Should We Monitor?	4
Operating System Monitoring	5
I/O Statistics	6
Oracle Clusterware Monitoring	7
Interacting with CRS and the OCR: srvctl	7
Checking CRS Status	7
Interconnect	8
CRS Alert Log	9
Oracle ASM Monitoring	10
Using the V\$ASM_CLIENT View	11
Using the V\$ASM_DISKGROUP View	11
Using the V\$ASM_DISK View	12
Oracle Database Monitoring	12
Database Status	12
RAC Status	12
Instance Busy Query	13
System Global Area Status Rules	13
TableSpace Query Rules	13
Backup Status	15
Cluster Latency Query	15
DB Instance Alert Log	16
Monitoring RMAN Job Progress	16
RMAN Status	17

Argent Monitor For Oracle Monitoring Overview

Oracle RAC Monitoring

There are a lot of components that make up the Oracle RAC environment from the RAW Disks all the way up to the User Application. The diagram below shows a high level view of the components that will need to be monitored to provide good visibility of the state of the Oracle environment.



As the application is taken into production, it grows and flourishes. That can be a big burden for the DBA if he is not proactively monitoring his database. There could be several causes for this growth: disk usage might go up, network bandwidth might bottleneck, transactions may start taking too long to commit, more users and more aggressive usage. This may be good for the business but the Service Level Agreements (SLAs) still need to be met. Proactive monitoring your Oracle RAC or even a typical single node Oracle database will keep you upwind of problems.

The following sections in this document will provide some insight into what parameters need to be monitored and how to go about it.

What Should We Monitor?

The questions below can assist a DBA to optimize his routine administration tasks as well as help management make timely decisions.

1. Are we meeting our SLA (Service Level Agreements)?
Which components do we need to provide SLA reporting against?
2. Are the High Availability (HA) objectives being met?
3. What is the status of instances running?
4. What is the interconnect status?
5. CPU: Are Oracle processes getting enough resources?
6. Memory: How is the memory for System Global Area (SGA) etc?

Questions similar to these, broken up to monitor at all levels (Application, Database, OS, and HW), help a DBA to monitor the RAC environment efficiently and effectively.

Hardware level: Disks, HBAs, NICs, cabling, backup devices etc. need to function properly. All these devices need to be configured properly.

- If disks are provided via a SAN then the SAN should be monitored to provide status of the RAW disks this can be achieved via SNMP or Scripted Interface (Vendor Specific).
- HBA Vendors provide scripts to check the status of all LUN's and HBA in the system.

OS level: You need to monitor the CPU, Memory, Disk performance and Network traffic.

- CPU (%Processor time, etc.)
- I/O (queue length)
- Shared storage State
- Network (both public and private network (interconnect))
- Memory (paging, swapping, etc.)
- Operating Specific Logs

Database Level

You should monitor the cluster logs, event logs, asm logs and rdbms logs.

- Cluster - CRS alert log file
- ASM – alert log file

You should monitor status of various database components.

- Oracle Performance Queries
- RMAN Status
- ASM Status
- Instance Status

Application Level

For instance, if the application server were a Tomcat Server or JBoss, then you would be interested in all of the Catalina logs, java.util logs or log4J logging, etc.

NOTE: Argent Monitor for Oracle and Argent Guardian can make use of data from any Shell Script or Oracle Query by creating a custom rule and checking the results of the script or query. This document only describes some of the default rules and how to create others from Best Practice procedures.

Operating System Monitoring

CPU Overload Alerts – Best practice defines monitoring for high processor utilization and high run queue values.

- **LINUX_CPU_OVERLOAD** - This Argent Linux shell script checks when the total CPU utilization for all processes exceeds a pre defined Threshold.
- **LINUX_CPU_USAGE_10** - This Argent Linux shell script checks when the total CPU utilization for all processes exceeds a pre defined Threshold and reports the processes using more than the threshold.
- **LINUX_CPU_RUN_Q** - This Argent Linux shell script checks when the CPU run Q exceeds a predefined Threshold. (Not in the default rules will be added by Argent)

RAM Memory Overload Alert – Best Practice defines monitoring for RAM memory overloads the following

- **LINUX_FREE_MEMORY_64M** - This Argent Linux shell scrip checks when the amount of free memory is below 64Megabytes or another predefined Threshold.
- **LINUX_MEMORY_PAGE_IN** - This Argent Linux shell script checks when the amount of Paging-in Operations based on a predefined Threshold. (Not in the default rules)

Paged in from swap - When a page is brought back from the swap device, the process will stop execution and wait. This may affect performance.

- **LINUX_MEMORY_SCAN_RATE** - This Argent Linux shell script checks when the amount of pages scanned Operations based on a predefined Threshold. (Not in the default rules)

Scan rate (pages). Note that this number is not reported as a “rate,” but as a total number of pages scanned.

The page scanning rate is the main tipoff that a system does not have enough physical memory.

Low Free Disk Space – If the archived redo log directory become full, Oracle database will hang. All file systems should be checked for free space and be alerted if threshold is crossed.

- **LINUX_DISK_SPACE_10** - This rule uses df to check for file systems free space. The rule fails if there is less than 10% free space.

Critical Process Check – ensures that critical processes are running.

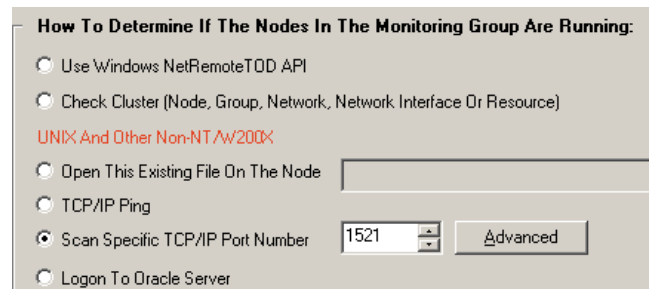
- **LINUX_ORACLE_PROCESSES** - This Argent Linux shell script checks the Oracle Database Processes using define SID are running. Default Processes configured are...

```
ora_pmon_${ORACLE_SID} ora_smon_${ORACLE_SID}
ora_lgwr_${ORACLE_SID}
ora_ckpt_${ORACLE_SID} ora_dbw0_${ORACLE_SID}
ora_reco_${ORACLE_SID} tnslsnr
```

Other oracle processes can be added to the default shell script.

Network Connectivity – Check various network components for availability.

- PING Virtual IP Address – A rule can be used to ensure that the VIP are up and available
- PING Public and Private network Interfaces
- TNS Listener – ensure that it is listening



- Oracle Logon – perform a logon to an Oracle Database.

I/O Statistics

Command: `iostat`

```
Linux 2.6.9-42.0.0.0.1.ELsmp (vm01.wolga.nl) 05/03/2007
avg-cpu:  %user   %nice    %sys %iowait  %idle
           1.49    0.07    5.81    2.32   90.30

Device:            tps    Blk read/s    Blk wrtn/s    Blk read    Blk wrtn
sda                2.18         2.60         76.77    2220396    65637664
sda1               9.65         2.60         76.76    2219427    65637656
sdb                0.00         0.00         0.00       2635         0
sdb1              0.00         0.00         0.00       1186         0
sdc                2.75        44.07        33.41    37681732    28563403
sdc1              2.76        44.07        33.41    37678487    28563403
sdd                1.71        10.44        33.41    8926315    28563403
sdd1              1.73        10.44        33.41    8923070    28563403
sde                1.12         1.26        22.43    1079253    19182446
sde1              1.13         1.26        22.43    1076008    19182446
sdf                4.31        502.78         4.18    429902055    3570958
sdf1              4.31        502.78         4.18    429900486    3570958
sdg                7.24       1004.57         5.15    858957930    4407357
sdg1              7.24       1004.57         5.15    858956361    4407357
sdh                1.00         1.00         0.50      858776    428077
sdh1              1.00         1.00         0.50      857207    428077
```

Shown above is an example of IOSTAT.

- Sda, sdb are the filesystem for OS installation and swap.
- Sdc, sdd and sde are the files used for OCR, Votingdisk and Spfileasm respectively.
- Sdf and Sdg are the filesystems for oradata (where all of our Oracle data files are residing) and sdh is for the flashback recovery.

You can clearly see that iowait is considerably low, which is a good thing--had it been higher you would be looking at an I/O bottleneck. On the disks (devices section) you can clearly see that our cluster is doing fine but the oradata disks are working hard.

A custom Argent Guardian Linux Rule should be created dependent on the exact information and system configuration.

Oracle Clusterware Monitoring

Cluster Ready Services, or CRS, is Oracle's own clusterware. On most platforms, Oracle supports vendor clusterware; in these cases, CRS interoperates with the vendor clusterware, providing high availability support and service and workload management. On Linux and Windows clusters, CRS serves as the sole clusterware. In all cases, CRS provides a standard cluster interface that is consistent across all platforms.

CRS consists of four processes (crsd, occsd, evmd, and evmlogger) and two disks: the Oracle Cluster Registry (OCR), and the voting disk.

CRS manages the following resources:

- The ASM instances on each node
- Databases
- The instances on each node
- Oracle Services on each node
- The cluster nodes themselves, including the following processes, or "nodeapps":
 - o VIP
 - o GSD
 - o The listener
 - o The ONS daemon

CRS stores information about these resources in the OCR. If the information in the OCR for one of these resources becomes damaged or inconsistent, then CRS is no longer able to manage that resource. Fortunately, the OCR automatically backs itself up regularly and frequently.

Interacting with CRS and the OCR: srvctl

Srvctl is the tool Oracle recommends that DBAs use to interact with CRS and the cluster registry.

To test if the nodeapps on a node are up run the following.

Command: `srvctl status nodeapps -n vm01`

Result: VIP is running on node: vm01
GSD is running on node: vm01
Listener is running on node: vm01
ONS daemon is running on node: vm01

By checking the status of individual nodes and all the necessary applications, we can see that the VIP, GSD, Listener and the ONS daemons are alive.

Checking CRS Status

This command is generally used to check the status of CRS. It shows the status of CRS on the local node – must be run on all nodes to ensure CRS is active.

Command: `crsctl check crs`

Result: Cluster Synchronization Services appears healthy
Cluster Ready Services appears healthy
Event Manager appears healthy

crs_stat: Lists the status of an application profile; used to check out resource status.

Resources can either be ONLINE or OFFLINE as shown in the STATE attribute. If the resource is online and the cluster node fails then the Clusterware tries to restart the application on another node. It goes without saying that these resources must be ONLINE unless you have a specific reason to FORCE them to stay OFFLINE. They can also be offline if the resource has a failure count higher than the failure threshold, in which case the TARGET is changed to OFFLINE.

```
crs_stat -t
```

Name	Type	Target	State	Host
ora_esxrac.db	application	ONLINE	ONLINE	vm02
ora....c1.inst	application	ONLINE	ONLINE	vm01
ora....c2.inst	application	ONLINE	ONLINE	vm02
ora....serv.cs	application	ONLINE	ONLINE	vm02
ora....ac1.srv	application	ONLINE	ONLINE	vm01
ora....ac2.srv	application	ONLINE	ONLINE	vm02
ora....SM1.asm	application	ONLINE	ONLINE	vm01
ora....01.lsnr	application	ONLINE	ONLINE	vm01
ora_vm01.gsd	application	ONLINE	ONLINE	vm01
ora_vm01.ons	application	ONLINE	ONLINE	vm01
ora_vm01.vip	application	ONLINE	ONLINE	vm01

On most It resides under the ORACLE_HOME/CRS/bin directory and there are several commands to perform various tasks, such as displaying individual resource and starting and stopping those resources.

Interconnect

To verify the accessibility of the cluster nodes, specified by node_list, from the local node or from any other cluster node, specified by src-node, use the component verification command nodereach as follows:

Command:

```
cluvfy comp nodereach -n node_list [ -srcnode node ] [-verbose]
```

When you issue the nodecon command as shown in the previous example, it instructs the CVU to perform the following tasks:

- o Discover all the network interfaces that are available on the cluster nodes.
- o Review the corresponding IP addresses and subnets for the interfaces.
- o Obtain the list of interfaces that are suitable for use as VIPs and the list of interfaces to private interconnect.
- o Verify the connectivity among all the nodes through those interfaces.

When you run the nodecon command in verbose mode, it identifies the mappings between the interfaces, IP addresses, and subnets. To verify the connectivity among the nodes through specific network interfaces, use the comp nodecon command with the -i option and specify the interfaces to be checked with the interface_list argument:

Command:

```
cluvfy comp nodecon -n node_list -i interface_list [-verbose]
```

For example, you can verify the connectivity among the nodes docrac1, docrac2, and docrac3, through the specific network interface eth0 by running the following command:

Command:

```
cluvfy comp nodecon -n docrac1, docrac2, docrac3 -i eth0 -verbose
```

All of these commands can be wrapped into a Shell Script or Oracle Query and used by the Argent Monitor for Oracle to generate alerts based on custom defined thresholds.

CRS Alert Log

Oracle Clusterware posts alert messages when important events occur. For example, you might see alert messages from the Cluster Ready Services (CRS) daemon process when it starts, if it aborts, if the failover process fails, or if automatic restart of a CRS resource failed.

Argent Data Consolidator can scan this log file and post an alert if an error is detected. For example, if a voting disk is not available, a CRS-1604 error is raised, and a critical alert is posted on the Argent Console and email sent.

The location of the Oracle Clusterware log file is CRS_home/log/hostname/alerthostname.log, where CRS_home is the directory in which Oracle Clusterware was installed and hostname is the host name of the local node.

Oracle ASM Monitoring

Check the status of the ASM on both nodes.

Command: `srvctl status asm -n vm01`

Result: ASM instance +ASM1 is running on node vm01.

ASM Views - The ASM configuration can be viewed using the views, which often contain different information depending on whether they are queried from the ASM instance, or a dependant database instance.

View	ASM Instance	DB Instance
V\$ASM_ALIAS	Displays a row for each alias present in every disk group mounted by the ASM instance.	Returns no rows
V\$ASM_CLIENT	Displays a row for each database instance using a disk group managed by the ASM instance.	Displays a row for the ASM instance if the database has open ASM files.
V\$ASM_DISK	Displays a row for each disk discovered by the ASM instance, including disks which are not part of any disk group.	Displays a row for each disk in disk groups in use by the database instance.
V\$ASM_DISKGROUP	Displays a row for each disk group discovered by the ASM instance.	Displays a row for each disk group mounted by the local ASM instance.
V\$ASM_FILE	Displays a row for each file for each disk group mounted by the ASM instance.	Displays no rows.
V\$ASM_OPERATION	Displays a row for each file for each long running operation executing in the ASM instance.	Displays no rows.
V\$ASM_TEMPLATE	Displays a row for each template present in each disk group mounted by the ASM instance.	Displays a row for each template present in each disk group mounted by the ASM instance with which the database instance communicates.

Using the V\$ASM_CLIENT View

An example query result list below:

Row #1

Column #1 (GROUP_NUMBER) = 1
 Column #2 (INSTANCE_NAME) = +ASM1
 Column #3 (DB_NAME) = TEST1
 Column #4 (STATUS) = CONNECTED
 Column #5 (SOFTWARE_VERSION) = 11.1.0.7.0
 Column #6 (COMPATIBLE_VERSION) = 11.0.0.0.0

An Argent Rule could look like this:

Run Following SQL Query Or Stored Procedure

☒ SQL Statement ☐ Stored Procedure

Select * FROM V\$ASM_CLIENT

SQL Timeout: 30 Seconds (Value 0 for default timeout)

Check Option: Check ALL Rows

Comparison Column: 0 (Used When Firing Events Separately)

Rule Is NOT Broken If The Following Conditions Are True:

String(\$4) Containing "CONNECTED"

The rule above would alert if Column 4 did not have a STATUS = CONNECTED

Using the V\$ASM_DISKGROUP View

An example query result list below:

Row #1

Column #1 (GROUP_NUMBER) = 1
 Column #2 (NAME) = DATA
 Column #3 (SECTOR_SIZE) = 512
 Column #4 (BLOCK_SIZE) = 4096
 Column #5 (ALLOCATION_UNIT_SIZE) = 1048576
 Column #6 (STATE) = CONNECTED
 Column #7 (TYPE) = EXTERN
 Column #8 (TOTAL_MB) = 98603
 Column #9 (FREE_MB) = 16152
 Column #10 (REQUIRED_MIRROR_FREE_MB) = 0
 Column #11 (USABLE_FILE_MB) = 16152
 Column #12 (OFFLINE_DISKS) = 0
 Column #13 (UNBALANCED) = N
 Column #14 (COMPATIBILITY) = 10.1.0.0.0
 Column #15 (DATABASE_COMPATIBILITY) = 10.1.0.0.0

An Argent Rule could look like this:

Run Following SQL Query Or Stored Procedure

☒ SQL Statement ☐ Stored Procedure

Select * FROM V\$ASM_DISKGROUP_STAT

SQL Timeout: 30 Seconds (Value 0 for default timeout)

Check Option: Check ALL Rows

Comparison Column: 0 (Used When Firing Events Separately)

Rule Is NOT Broken If The Following Conditions Are True:

String(\$6) Containing "CONNECTED"
 OR
 String(\$9) > 5000
 OR
 String(\$12) = 0

The rule above would alert if Column 6 does not have a STATUS = CONNECTED or

If Column 9 does not have FREE_MB > 5000 or

If Column 12 does not have OFFLINE_DISKS = 0

Using the V\$ASM_DISK View

An example query result list below:

Row #1

Column #1 (GROUP_NUMBER) = 1
 Column #2 (DISK_NUMBER) = 0
 Column #3 (COMPOUND_INDEX) = 16777216
 Column #4 (INCARNATION) = 3915949486
 Column #5 (MOUNT_STATUS) = OPENED
 Column #6 (HEADER_STATUS) = UNKNOWN
 Column #7 (MODE_STATUS) = INVALID
 Column #8 (STATE) = NORMAL
 Column #9 (REDUNDANCY) = UNKNOWN
 Column #10 (LIBRARY) = ASM Library - Generic Linux,
 version 2.0.4 (KABI_V2)
 Column #11 (TOTAL_MB) = 19531
 Column #12 (FREE_MB) = 0
 Column #13 (NAME) = DISK01
 Column #14 (FAILGROUP) =
 Column #15 (LABEL) = DISK01
 Column #16 (PATH) = ORCL:DISK01
 Column #17 (UDID) =
 Column #18 (PRODUCT) =
 Column #19 (CREATE_DATE) = 99/99/99 99:99:99
 Column #20 (MOUNT_DATE) = 99/99/99 99:99:99
 Column #21 (REPAIR_TIMER) = 0
 Column #22 (READS) = 6699512
 Column #23 (WRITES) = 3278078
 Column #24 (READ_ERRS) = 0
 Column #25 (WRITE_ERRS) = 0
 Column #26 (READ_TIME) = 67732.483243
 Column #27 (WRITE_TIME) = 38791.776854
 Column #28 (BYTES_READ) = 288144087552
 Column #29 (BYTES_WRITTEN) = 57990982144

An Argent Rule could look for the following states.

If Column 8 does not have a STATE = NORMAL or

If Column 24 does not have READ_ERRS = 0 or

If Column 25 does not have WRITE_ERRS = 0

Oracle Database Monitoring

Database Status

Command: `srvctl status database -d esrac`

Result: Instance esrac1 is running on node vm01
 Instance esrac2 is running on node vm02

RAC Status

Query the gv\$instance view to get the instances, hostname, and status of the node apps.

Command: `select instance_name, host_name, archiver, thread#, status from gv$instance`

Result:

INSTANCE_NAME	HOST_NAME	ARCHIVE	THREAD#	STATUS
-----	-----	-----	-----	-----
esrac1	vm01.test	STARTED	1	OPEN
esrac2	vm02.test	STARTED	2	OPEN

Run Following SQL Query Or Stored Procedure

☒ SQL Statement
 ☐ Stored Procedure

select instance_name, host_name, archiver, thread#, status from gv\$instance

SQL Timeout: Seconds (Value 0 for default timeout)

Check Option:

Comparison Column: (Used When Firing Events Separately)

Rule Is NOT Broken If The Following Conditions Are True:

String(\$5) Containing "OPEN"

Argent Oracle Query Rule could look like this.

Instance Busy Query

EXCESSIVE_CPU_9 - Excessive CPU Usage

```

SELECT COUNT(1) "Excessive CPU time SQL count"
FROM v$sqlarea
WHERE (cpu_time/1000) > 90

```

System Global Area Status Rules

QRY_SGA_DB_BUFFER - This rule checks the database buffer size in SGA

select value from v\$sga where name = 'Database Buffers'

QRY_SGA_REDO_BUFFER - This rule checks the redo buffer size in SGA

select value from v\$sga where name = 'Redo Buffers'

QRY_SGA_VARIABLE_SIZE - This rule checks the variable size in SGA

select value from v\$sga where name = 'Variable Size'

TableSpace Query Rules

TEMP_TABLESPACE - This rule checks if the temporary tablespace runs low.

```

SELECT tablespace_name, SUM(bytes_used), SUM(bytes_free) FROM V$temp_space_header
GROUP BY tablespace_name

```

QRY_TBLSPC_USED_LT_90 - This rule checks whenever any tablespaces contain less space than specified in this case 90%. With AUTO_EXTEND ON the tablespace will automatically extend but at least an alert will acknowledge this to the DBA.

```

SELECT a.tablespace_name, a.bytes bytes_used, b.bytes bytes_free, b.largest,
       ROUND (((a.bytes-b.bytes)/a.bytes)*100,0) percent_used FROM (
  SELECT tablespace_name,
         SUM(bytes) bytes
  FROM sys.dba_data_files
  GROUP BY tablespace_name
) a, (
  SELECT tablespace_name,
         SUM(bytes) bytes,
         MAX(bytes) largest
  FROM sys.dba_free_space
  GROUP BY tablespace_name
) b
WHERE a.tablespace_name=b.tablespace_name
ORDER BY percent_used DESC

```

FREE_EXTENT – This rule checks if the free extent amount is too low.

```
select TABLESPACE_NAME,sum(BLOCKS) free_blocks
from dba_free_space group by TABLESPACE_NAME
```

Backup Status

QRY_CRASHED_BACKUP - This alert will report on any component of dbms_scheduler that has been disabled, or failed to execute at its scheduled time. This rule should be executed in the time when the backup is supposed to have completed. If any tablespace is left in BACKUP mode, the backup was crashed and manual recovery is required.

```
select a.tablespace_name,a.file_name,b.status from dba_data_
files a, v$backup b where a.file_id=b.file# and b.status='ACTIVE'
```

BUFFER_CACHE_HIT_RATIO_9 - Determines if a buffer cache hit ratio is too low. It is very useful for locating times when decision support type queries are being run, since a large number of large-table full table scans will make the data buffer hit ratio drop.

```
SELECT DECODE(NAME, 'DEFAULT', DECODE(block_size, 2
048, 'CACHE_2K', 4096, 'CACHE_4K', 8192, 'CACHE_8K',
16384, 'CACHE_16K', 'CACHE_32K'), NAME) "NAME",
PHYSICAL_READS, DB_BLOCK_GETS, CONSISTENT_
GETS, 1 -
((DECODE(PHYSICAL_READS, 0, 1, PHYSICAL_READS))/
DECODE((DB_BLOCK_GETS + CONSISTENT_GETS),0,
DECODE(PHYSICAL_READS, 0, 1,
PHYSICAL_READS),(DB_BLOCK_GETS + CONSISTENT_
GETS))) "Hit_Ratio"
FROM V$BUFFER_POOL_STATISTICS
```

QRY_SORT_TO_DISK_RATIO - Determines if the sort to disk ratio is too low. This report is very useful for monitoring the amount of activity against the TEMP tablespace, and it is also useful for ensuring that sort_area_size is set to an optimal level. As a general rule, increasing sort_area_size will reduce the number of disk sorts, but huge sorts will always need to be performed on disk in the TEMP tablespace.

```
SELECT A.VALUE "DISK SORTS", B.VALUE "MEMORY
SORTS",
```

```
ROUND(A.VALUE/(A.VALUE+B.VALUE),2) "DISK SORT
RATIO"
```

```
FROM v$sysstat A, v$sysstat B WHERE A.NAME = '
sorts (disk)' and B.NAME = 'sorts (memory)'
```

QRY_LIBRARY_CACHE_HIT_RATIO - Determines if the library hit ratio is too low. Library cache hit ratio means the proportion of requests for an object which were satisfied by finding that object already in memory (Library cache). When the library cache miss ratio is greater than .02, you may want to increase shared_pool_size.

The ratio is only meaningful after the instance has been up and running for a "long time" under normal database activity. The long time is a relative term and is the time for which the instance should be running to reach normal database activity. The library cache memory area contains shared SQL and PL/SQL statements. When a user submits a SQL and PL/SQL statement to the database, Oracle parses it into executable form before executing the code. Oracle then temporarily stores the parsed SQL statement in the library cache memory buffers, so when a different user submits the same query, Oracle can simply use the parsed SQL statement already in memory, instead of having to recompile it all over again. When buffering parsed SQL, Oracle uses the LRU algorithm to make room for newer statements by aging out the older, least recently used statements.

You can monitor statistics reflecting library cache activity by examining the dynamic performance view V\$LIBRARYCACHE. These statistics reflect all library cache activity since the most recent instance startup. Each row in this view contains statistics for one type of item kept in the library cache. The item described by each row is identified by the value of the NAMESPACE column. Rows with the following NAMESPACE values reflect library cache activity for SQL statements and PL/SQL blocks:

```
SELECT ROUND(SUM(PINHITS)/SUM(PINS)*100,2) as "Li-
brary Cache Hit Ratio"
FROM v$librarycache
```

PINS: Requests for objects of this namespace.

PINHITS: The number of times a PIN all of the meta data pieces of the library object were found in memory
(PINS - RELOADS).

RELOADS: Number of times objects stored in the library cache had to be reloaded into memory because part of the object had been flushed from the cache. If there are a significant number of reloads, then reusable information is being flushed from the library cache. This requires a reload/rebuild of the object before it can again be accessed.

QRY_DICTIONARY_CACHE_HIT_RATIO - Determines if the dictionary hit ratio is too low. The dictionary cache consists of memory buffers that cache information from the data dictionary. Oracle uses data dictionary information behind the scenes to manage database operations. If oracle can read data dictionary information in memory from the dictionary cache, it can avoid having to read the dictionary tables from the disk.

The dictionary cache hit ratio is a measure the proportion of requests for information from the data dictionary, the collection of database tables and views containing reference information about the database, its structures, and its users. On instance startup, the data dictionary cache contains no data, so any SQL statement issued is likely to result in cache misses. As more data is read into the cache, the likelihood of cache misses should decrease. Eventually the database should reach a "steady state" in which the most frequently used dictionary data is in the cache.

```
SELECT ROUND((SUM(gets - getmisses)) /  
SUM(gets)*100,2) "Dictionary Cache Hit Ratio"  
FROM v$rowcache
```

GETS: Total number of requests.

GETMISSES: Number of times oracle had to retrieve information from the data dictionary tables on disk because it had been aged out of the data dictionary cache.

Cluster Latency Query

Another useful SELECT for determining latency measures the overall latency, including that for queue, build, flush and send time. These statistics are also found in the gv\$sysstat view. The SELECT is shown below:

```
SELECT  
    a.inst_id "Instance",  
    (a.value+b.value+c.value)/d.  
    value "LMS Service Time"  
FROM  
    GV$SYSSTAT A,  
    GV$SYSSTAT B,  
    GV$SYSSTAT C,  
    GV$SYSSTAT D  
WHERE  
    A.name = 'global cache cr  
block build time' AND  
    B.name = 'global cache cr  
block flush time' AND  
    C.name = 'global cache cr  
block send time' AND  
    D.name = 'global cache cr  
blocks served' AND  
    B.inst_id=A.inst_id AND  
    C.inst_id=A.inst_id AND  
    D.inst_id=A.inst_id  
ORDER BY  
    a.inst_id;
```

Instance	LMS Service Time
1	1.07933923
2	636687318

The following code can be used to examine the individual components of the service time to determine the source of the problem:

```
SELECT
    A.inst_id "Instance",
    (A.value/D.value) "Consistent Read
    Build",
    (B.value/D.value) "Log Flush Wait",
    (C.value/D.value) "Send Time"
FROM
    GV$SYSSTAT A,
    GV$SYSSTAT B,
    GV$SYSSTAT C,
    GV$SYSSTAT D
WHERE
    A.name = 'global cache cr block
    build time' AND
    B.name = 'global cache cr block flush
    time' AND
    C.name = 'global cache cr block send
    time' AND
    D.name = 'global cache cr blocks
    served' AND
    B.inst_id=a.inst_id AND
    C.inst_id=a.inst_id AND
    D.inst_id=a.inst_id
ORDER BY
    A.inst_id;
```

Instance	Consistent Read Build	Log Flush Wait	Send Time
1	.00737234	1.05059755	.02203942
2	.04645529	.51214820	.07844674

If problems are detected, operating system specific commands should be used to pinpoint the node having difficulties. Check the following for issues with the cluster interconnects.

* A large number of processes in the run queue waiting for CPU or scheduling delays.

* Platform specific operating system parameter settings that affect IPC buffering or process scheduling.

* Slow, busy or faulty interconnects. In these cases, look for dropped packets, retransmits, or cyclic redundancy check (CRC) errors. Ensure that the network is private and that inter-instance traffic is not routed through a public network.

DB Instance Alert Log

Oracle Instances posts alert messages when important events occur. Argent Data Consolidator can scan this log file and post an alert if an error is detected. For example, any lines with (ORA-) indicates an error, and a critical alert is posted on the Argent Console and email sent.

The default location of the Oracle Instance log file is ORACLE_HOME/rdbms/trace if it is not specified in the BACKGROUND_DUMP_DEST parameter.

Monitoring RMAN Job Progress

There is an easy way to monitor your RMAN backup scripts in a unix environment. Log the rman output to a file and then grep it for errors.

```
rman target / CMDFILE /home/oracle/
scripts/rman.sql LOG $logfile
errors=`cat $logfile|grep -i error|wc -l`
if test $errors -gt 0
then
`cat $logfile`
!
fi
```

Another method would be to examine the return code from the rman client. RMAN returns a value of 0 to the shell if the backup was successful. Otherwise it returns a nonzero value.

```
rman target / CMDFILE /home/oracle/
scripts/rman.sql LOG $logfile
status=$?

if [ $status -gt 0 ] ; then
STATUS="BACKUP FAILED"
`cat $logfile`
!
else
STATUS="SuccessfulBackup"
`cat $logfile`
!
fi
```


RMAN Status

Displays the finished and on-going RMAN jobs. For on-going jobs, this view displays progress and status. The jobs which are in progress are stored only in memory while the finished jobs are stored in the controlfile.

Column	Datatype	Description
SID	NUMBER	Session ID of the session which is running this RMAN operation
RECID	NUMBER	Record ID of the row in the controlfile
STAMP	NUMBER	Timestamp of the row (RECID + STAMP is unique)
PARENT_RECID	VARCHAR2(40)	Record ID of the parent row of this row (corresponding V\$RMAN_STATUS row with ROW_LEVEL = ROW_LEVEL - 1)
PARENT_STAMP	VARCHAR2(40)	Timestamp of the parent row of this row (corresponding V\$RMAN_STATUS row with ROW_LEVEL = ROW_LEVEL - 1)
SESSION_RECID	NUMBER	Record ID of the session (corresponding V\$RMAN_STATUS row with ROW_LEVEL = 0)
SESSION_STAMP	NUMBER	Timestamp of the session (corresponding V\$RMAN_STATUS row with ROW_LEVEL = 0)
ROW_LEVEL	NUMBER	Level of the row. The session has level 0.
ROW_TYPE	VARCHAR2(19)	Type of the row: SESSION COMMAND RECURSIVE OPERATION
COMMAND_ID	VARCHAR2(33)	Command ID set by the RMAN SET COMMAND ID command. If not set, then RMAN will create a unique number.
OPERATION	VARCHAR2(33)	Name of the command in the execution explained by this row
STATUS	VARCHAR2(23)	Status of the operation: RUNNING RUNNING WITH WARNINGS RUNNING WITH ERRORS COMPLETED COMPLETED WITH WARNINGS COMPLETED WITH ERRORS FAILED
MBYTES_PROCESSED	NUMBER	Percentage of the job completed; null if not applicable for the operation
START_TIME	DATE	Start time of the job
END_TIME	DATE	End time of the job

RMAN Status

Monitor the progress of backups and restores by querying the view `V$SESSION_LONGOPS`. RMAN uses two types of rows in `V$SESSION_LONGOPS`: detail and aggregate rows. Detail rows describe the files being processed by one job step, while aggregate rows describe the files processed by all job steps in an RMAN command. A job step is the creation or restore of one backup set or datafile copy. Detail rows are updated with every buffer that is read or written during the backup step, so their granularity of update is small. Aggregate rows are updated when each job step completes, so their granularity of update is large.

The table describes column in `V$SESSION_LONGOPS` that are most relevant for RMAN. Typically, you will view the detail rows rather than the aggregate rows to determine the progress of each backup set.

Column	Description for Detail Rows
SID	The server session ID corresponding to an RMAN channel.
SERIAL#	The server session serial number. This value changes each time a server session is reused.
OPNAME	A text description of the row. Examples of details rows include RMAN: datafile copy, RMAN: full datafile backup, and RMAN: full datafile restore. Note: RMAN: aggregate input and RMAN: aggregate output are the only aggregate rows.
CONTEXT	For backup output rows, this value is 2. For all other rows except proxy copy (which does not update this column), the value is 1.
SOFAR	The meaning of this column depends on the type of operation described by this row: <ul style="list-style-type: none"> • For image copies, the number of blocks that have been read. • For backup input rows, the number of blocks that have been read from the files being backed up. • For backup output rows, the number of blocks that have been written to the backup. • For restores, the number of blocks that have been processed to the files that are being restored in this one job step. • For proxy copies, the number of files that have been copied.
TOTALWORK	The meaning of this column depends on the type of operation described by this row: <ul style="list-style-type: none"> • For image copies, the total number of blocks in the file. • For backup input rows, the total number of blocks to be read from all files processed • For backup output rows, the value is 0 because RMAN does not know how many blocks that it will write into any backup piece. • For restores, the total number of blocks in all files restored in this job step. • For proxy copies, the total number of files to be copied in this job step.

Each server session performing a backup or restore reports its progress compared to the total amount of work required for a job step. For example, if you perform a database restore that uses two channels, and each channel has two backup sets to restore (a total of four sets), then each server session reports its progress through a single backup set. When that set is completely restored, RMAN begins reporting progress on the next set to restore.

To monitor job progress - create a Oracle Query Rule containing the following SQL statement:

```
SELECT SID, SERIAL#, CONTEXT, SOFAR, TOTALWORK,
ROUND(SOFAR/TOTALWORK*100,2) "%_COMPLETE"
FROM V$SESSION_LONGOPS
WHERE OPNAME LIKE 'RMAN%'
AND OPNAME NOT LIKE '%aggregate%'
AND TOTALWORK != 0
AND SOFAR <> TOTALWORK
```

While the job is running, run an Oracle Query Rule while connected to the target database, and execute the longops script to check the progress of the RMAN job. If you repeat the query while the restore progresses, then you see output such as the following:

```
SQL> @longops
```

SID	SERIAL#	CONTEXT	SOFAR	TOTALWORK	%_COMPLETE
8	19	1	10377	36617	28.34

```
SQL> @longops
```

SID	SERIAL#	CONTEXT	SOFAR	TOTALWORK	%_COMPLETE
8	19	1	21513	36617	58.75

```
SQL> @longops
```

```
no rows selected
```

If you run the script at intervals of two minutes or more and the %_COMPLETE column does not increase, then RMAN is encountering a problem. If you frequently monitor the execution of long-running tasks, you could create an Oracle Query Rule.

Note: ArgSoft Intellectual Property Holdings Limited has created this White Paper for informational purposes only. ArgSoft Intellectual Property Holdings Limited makes no warranties, express or implied, in this document. The information contained in this document is subject to change without notice. ArgSoft Intellectual Property Holdings Limited shall not be liable for any technical or editorial errors, or omissions contained in this document, nor for incidental, indirect or consequential damages resulting from the furnishing, performance, or use of the material contained in this document, or the document itself. All views expressed are opinions of ArgSoft Intellectual Property Holdings Limited. All trademarks are the property of their respective owners.