Oracle Maximum
Availability Architecture

Exadata Health and Resource Utilization Monitoring

Exadata Storage Server KPIs

ORACLE WHITE PAPER | July 2016

Contents

Introduction

Oracle Exadata provides customers with a tightly-integrated hardware and software stack. The components work together, making the solution extremely performant. One part of the stack that is often poorly understood is the I/O subsystem. Having an understanding of the components and setting up appropriate monitoring are key tasks for Exadata administrators. In this whitepaper we will discuss the components of the Exadata I/O stack, identify key performance indicators, and make use of Oracle's world class monitoring solution, Enterprise Manager, to provide a holistic approach to making sure the I/O subsystem is functioning properly.

Overview of Exadata Storage

Figure 1 shows an Exadata rack layout for a typical two-socket system (i.e. X5-2). This system would be classified as a full rack as there are eight compute nodes and fourteen storage servers. You'll also notice three InfiniBand switches as well as a Cisco Ethernet switch. Two of the InfiniBand (IB) switches, also known as leaf switches, serve as primary and backup switches for the InfiniBand fabric local to the rack. The third switch, also known as a spine switch, is used for connectivity between racks. (Note that starting with the Exadata X4 series, the spine switch is optional and no longer shipped as a standard component.) If the Exadata rack in question is an eight socket system (i.e. X5-8) the rack would be similar; however only two larger (4 RU) compute nodes would be present. In addition to full racks, half, quarter, eighth, and elastic configuration racks can also be ordered. Elastic configurations allow Oracle Exadata racks to have customer-defined combinations of database servers and Exadata Storage Servers. For example, Oracle Exadata Database Machine X5-2 Elastic Configuration can have zero to 22 database servers, zero to 19 Exadata Storage Servers, or a combination of database servers and Exadata Storage Servers.

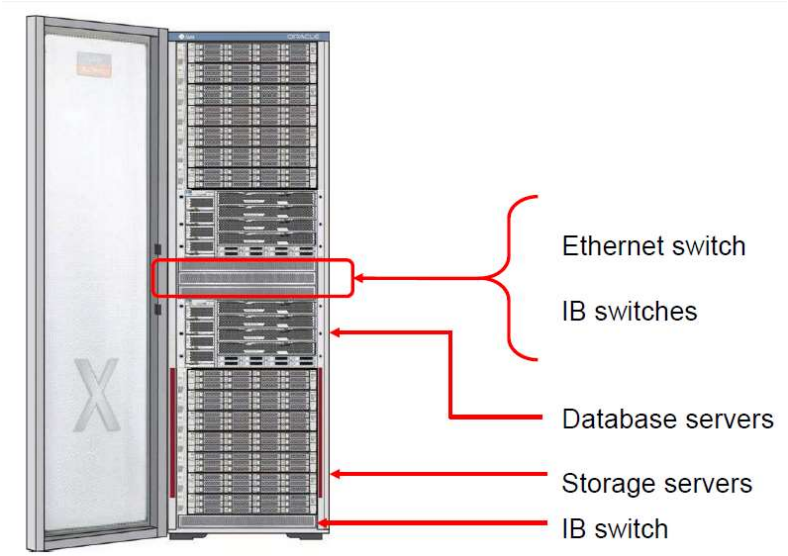


Figure 1

Storage Server Architecture

Figure 2 is an image of an Exadata Storage Server. Each Storage Server consists of 12 disks. X4 and previous Exadata generation Storage Servers consisted exclusively of physical disks. However, starting with X5 generation servers, storage cells can be ordered with either physical disks or flash disks.

Figure 2

In addition to the 12 hot-swappable SAS disks visible on the front of the Storage Server, there are also four PCI-E flash memory cards. The amount of memory on each card will vary depending on the generation of the server. The memory on the flash cards makes up the Smart Flash Cache for the Storage Server.

There are several layers of abstraction for the non-PCI storage devices in the Storage Server:

- Physical Disk – The physical disk or PCI-E flash device
- LUN – A logical abstraction of the physical device. Multiple LUNs can exist on one disk. The first two LUNs on the first two physical devices are used for storing the operating system of the Storage Server.
- Cell Disk – A layer of abstraction on top of the LUN. A cell disk by default is comprised of one LUN.
- Grid Disk – A grid disk is comprised of all or part of a cell disk. One or more grid disks can be created on a cell disk. The first grid disk that is created upon a cell disk is placed on the outer sectors of the underlying hard disk, providing it with the best performance. By default, three grid disks are created on each cell disk: DATA, RECO and DBFS with the exception of the first two cell disks which are smaller due to placement of the operating system. The first two cell disks do not by default have DBFS grid disks placed on them.
- ASM Disk group – Disk groups are created on top of the grid disks and are setup in much the same way as with a non Exadata system. ASM accesses the grid disks via the InfiniBand fabric in the rack.

The PCI-E flash cards in the Storage Server comprise the Exadata Smart Flash Cache and Smart Flash Log. Exadata Smart Flash Cache understands the different types of Database I/O operations and makes cache decisions accordingly. Exadata Smart Flash Cache can operate in two modes:

- Write Through – Provides the ability to cache read I/Os. Write operations go straight to disk.
- Write Back – In addition to caching read/IOs, provides the ability to cache write I/Os directly to PCI flash.

Exadata Smart Flash logging allows the exadata flash storage to serve as a secondary destination for redo log writes.

Key Performance Indicators

A Key Performance Indicator (KPI) is a measurement used to define and evaluate successful operation. In the context of this whitepaper we will use KPIs to evaluate whether the I/O subsystem performance is within an established specification.

A relatively simple generic example of a KPI would be CPU utilization. If a given system's performance degrades when CPU usage exceeds 95%, then 95% would be the critical threshold for the KPI. Many times it is useful to have a threshold not only for when levels are critical but before they are critical as well. In the example above 95% utilization would still be the critical threshold but a warning threshold should also be set so that administrators can be notified early enough to correct the issue before it becomes critical. For example, the warning threshold could be 90%. These are only example values to differentiate critical and warning thresholds.

In the CPU example above it is relatively easy to set the thresholds because CPU usage is well understood and in most cases the threshold would be close to the same between servers and environments. Unfortunately many other KPIs are more difficult to define, especially KPIs that relate to I/O. Not only can the thresholds be difficult to define but in some instances the KPIs themselves are hard to identify. Looking at an Exadata cell, there can be over 3,000 Storage Server metrics. Sorting through the data and identifying which ones are important can be very challenging!

When we look at a system or subsystem holistically, often one KPI isn't enough to identify issues. Take for example a vanilla Oracle Linux server. What would be necessary to identify if the server is performing within specification? In the above example, we identified CPU as a KPI. However there are other areas of server performance that would need to be taken into account such as memory, paging, disk, etc. The same holds true for the Exadata I/O subsystem. Although there are many metrics that are valid and important, no single metric can identify when the I/O system is at capacity.

For example, one indicator that is often looked at is I/Os per second (IOPS). IOPS shows the number of read and write operations to a disk. It might seem that this would be a finite number upon which it would be easy to base a threshold. However, the nature of the workload can affect the maximum number of IOPS a disk can perform. For example, a disk can perform far more small I/Os than large I/Os in a given period. So combining IOPS with other Storage Server metrics gives us a more comprehensive look at the environment.

Figure 3 shows the key metrics the MAA Team has identified during testing that will allow you to monitor if your Exadata I/O subsystem is reaching capacity.

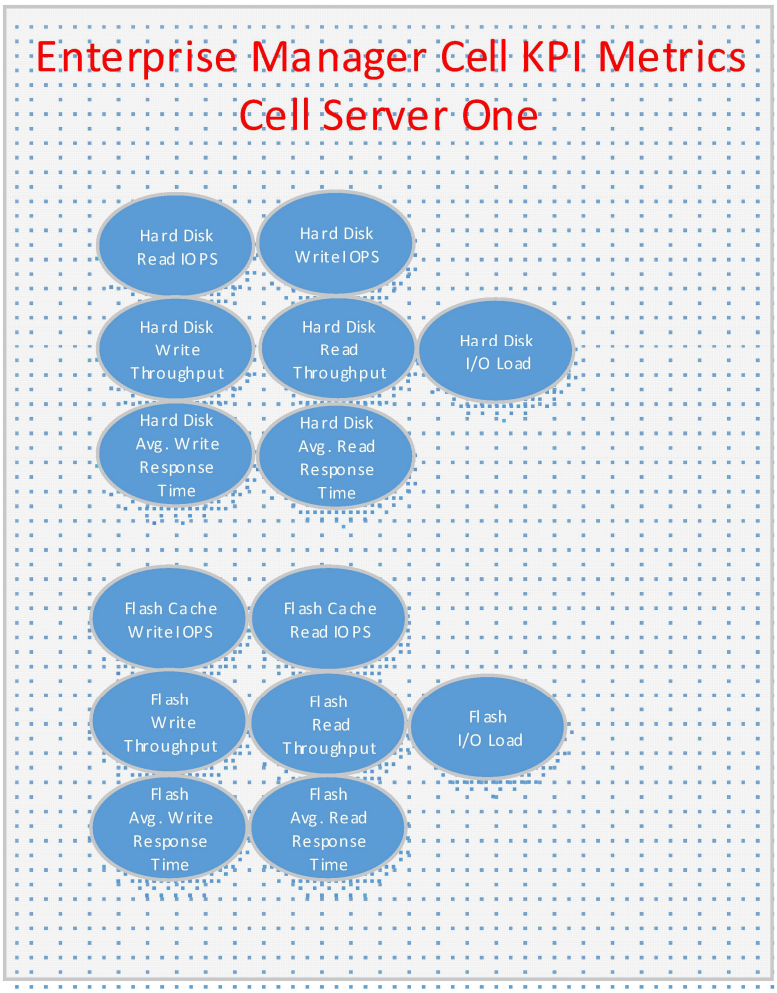


Figure 3

These Enterprise Manager metrics will be used to define the Metric Extension in the next section. As you can see, there have been seven metrics defined for Flash Cache Usage and seven metrics defined for Hard Disk Metric.

Monitoring Exadata Storage Servers with Enterprise Manager

Enterprise Manager (EM) provides extensive monitoring capabilities for Oracle Exadata. The following section will describe the basic concepts of monitoring an Exadata environment in Enterprise Manager using EM's Extensibility features to define the KPIs discussed earlier. Before digging in deeply there are a few EM terms that should be defined.

- Agent – A process that runs on a host to monitor the status, health, and performance of all managed components (also referred to as targets) on that host. In Exadata, the agents run only on the compute nodes. The Storage Server servers are monitored via the compute node agents.
- Plug-In – A Plug-in is a group of files (such as target definition files, collection scripts to collect metrics from targets, and any custom user interface (UI) components). In the following discussion, the Exadata Plug-In will be used extensively. Plug-Ins are deployed to the OMS(s) and to the agent(s) requiring them.
- Target – A component monitored by Enterprise Manager through an agent.
- Metric – Measurement used to monitor target conditions or state.
- Metric Extension – Provides the capability to extend Enterprise Manager monitoring to conditions specific to particular environments via custom scripts, SQL queries, and function calls.
- Threshold – A value defined for a metric. This value sets the level at which notifications or actions occur for the metric. There are two levels of thresholds: warning and critical.
- Service -- An entity that provides a useful function to its users. In Enterprise Manager, services can be defined and monitored. Services can be used to monitor things such as the end-to-end availability of an application (Database, Servers, Web Tier, etc.). For the purposes of this document we will use services to monitor an Exadata I/O subsystem.

Metric Extensions

Metric Extensions allow users to extend Enterprise Manager capabilities by defining custom metrics to cover critical information specific to the operation of their environment. There are two types of metric extensions available which are defined below:

Repository-side metric extensions: This type of metric extension allows you to use SQL scripts to extract information directly from the Enterprise Manager repository and raise alerts for the target against which the repository-side extension is run.

Metric Extensions (default): This type of metric extension allows for the creation of a script that will be deployed to the host of the monitoring agent for the chosen target. A wide variety of target types can be chosen. The Exadata target types are Compute Node, Cisco Switch, ILOM, PDU, KVM and the Storage Server.

Although EM has a wide range of metrics available for Exadata Storage Servers it is necessary to aggregate some of the information to give an idea of whether the Storage Server infrastructure is at capacity. In the below graph the KPI metrics have been reduced from the original seven out of the box metrics for Flash Cache and Hard Disk to five. Four of the five metric extensions are combinations of the out of the box metrics listed above. One of the metrics, Exadata Storage Server Hard Disk/Flash health, is created as a function of the other new metric extensions. This metric will evaluate the other KPI metrics to see how many are exceeding their thresholds. Evaluating multiple metrics together allows us to more accurately identify whether an issue is occurring.

Keep in mind that the roll up of all the data so far is still at the individual Storage Server level. At this point there is still no way to evaluate the Exadata Storage Server Grid as a whole. Figures 4 and 5 list which out of the box Enterprise Manager Metrics comprise the KPI Metrics Extension.

The warning and critical thresholds below will vary depending on many factors including rack size, Exadata version, application workload, etc. Initial values can be set using the above data as well as the product datasheets available at:
http://www.oracle.com/technetwork/database/exadata/overview/index.html.

Metric Name	Description	Warning	Critical
Total Cell Flash IOPS	Combines Aggregated total Flash read and write IOPS for a cell	112,500	125,000
Total Cell HardDisk IOPS	Combines Aggregated total HardDisk read and write IOPS for a cell	3240	3600
Total Cell Flash Throughput	Combines Aggregated total Celldisk read and write throughput for Flash for a cell	4860	5400

Total Cell HardDisk Throughput	Combines Aggregated total Celldisk read and write throughput for HardDisk for a cell	144400	16000
Total Avg Flash Response Time	Combines Aggregated average Celldisk read and write latency of Flash for a cell	8	10
Total Avg HardDisk Response Time	Combines Aggregated average Celldisk read and write latency of HardDisks for a cell	15	20
Total Avg HardDisk IO Load	Average IO Load for HardDisk for a cell	8	10
Total Avg Flash IO Load	Average IO Load for Flash for a cell	8	10
Exadata Storage Server HardDisk I/O Health	Number of identified performance metrics above that are exceeding their defined critical thresholds for Hard Disk	2	3
Exadata Storage Server FlashDisk I/O Health	Number of identified performance metrics above that are exceeding their defined critical thresholds for Flash Disk	2	3

Figure 4

Once initial warning and critical values are set, metric data should be monitored during peak usage to determine settings specific to the environment and updated with appropriate values. In environments with multiple Database machines each target should have KPIs setup using the process described above; thresholds cannot be assumed to be consistent between environments.

The composite metrics Exadata Storage Server FlashDisk I/O Health and HardDisk I/O Health depicted in Figure 5 are indicators comprised of the other KPI metrics. They summarize the KPIs that are exceeding their threshold values. Since these composite metrics are meant to summarize the overall health it is suggested that alerting for incidents be enabled for these metrics only.

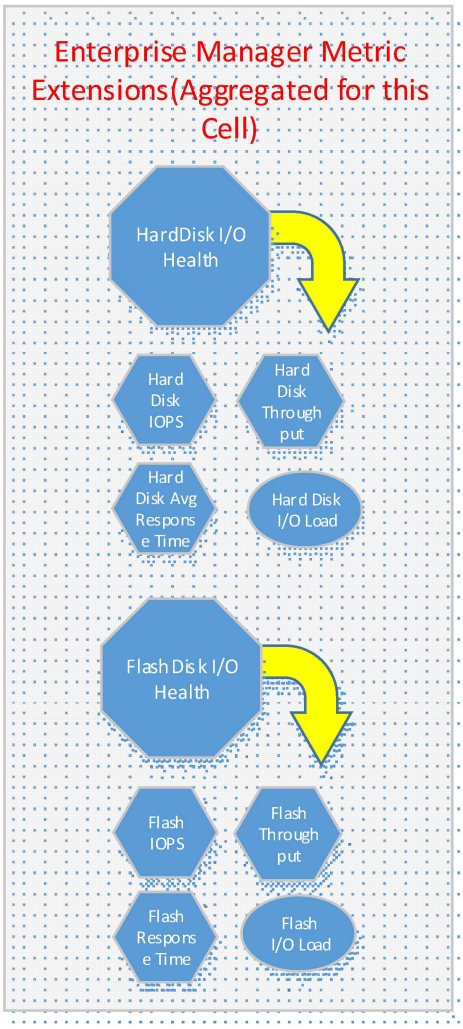

Enterprise Manager Metric Extensions(Aggregated for this Cell)
HardDisk I/O Health
Hard Disk IOPS
Hard Disk Through put
Hard Disk Avg Respons e Time
Hard Disk I/O Load
Flash Disk I/O Health
Flash IOPS
Flash Through put
Flash Respons e Time
Flash I/O Load

Figure 5

Creating KPI Metric Extensions

To create the above metric extensions follow the instructions below. Optionally the Metric Extensions can be downloaded from MOS Note 2094648.1 Note that if the Metric Extensions are downloaded and installed, it will still be necessary to publish the metric extensions and deploy them to targets.

In the instructions below, **Blue Highlighted Text** indicates a menu item on the Enterprise Manager Top Menu bar, depicted in Figure 6.

Figure 6

The first Metric Extension we will create is the Total Grid physical disk IOPS for the entire cell server

Enterprise → Monitoring → Metric Extensions

On the Metric Extension Home Page, create a new Metric Extension by selecting the following, as shown in Figure 4.2:

Actions → Create → Repository-side Metric Extension

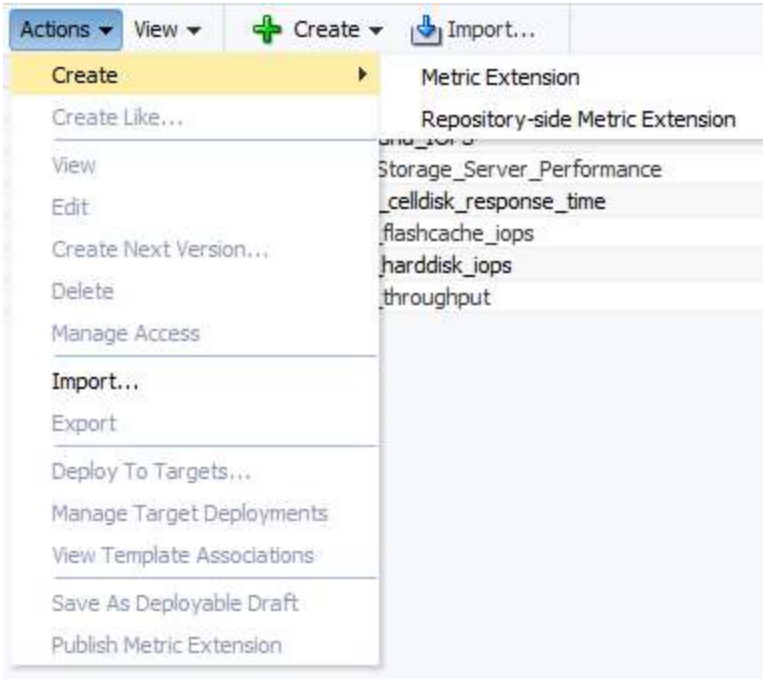


Figure 7

On the "Create New: General Properties" page, enter the following:

Target Type: "Oracle Exadata Storage Server"

Name ME$: "Total_Cell_HardDisk_IOPS"

Display Name: "Total Cell HardDisk IOPS"

Adapter: "OS Command – Multiple Columns"

Description: "Metric Extenstion to monitoring Exadata Hard Disk IOPS"

Data Collection Radio Button: Enabled

Use of Metric Data Radio Button: Alerting and Historical Trending

Frequency Drop Down: By Minutes

Repeat Every: 15 Minutes

Select the "Next" button:

Figure 8

On the "SQL Query" page, enter the following SQL:

```
SELECT target_guid,sum(value) total_iops
FROM   mgmt$metric_current
WHERE  target_type = 'oracle_exadata'
       AND metric_column in
('sum_cd_read_iops','sum_cd_write_iops')
       AND metric_name = 'Aggregated_HardNFlashDisk_Metric'
       AND key_value = 'HardDisk'
     group by target_guid
```

Select the "Next" button

On the Columns page:

Select the TOTAL_IOPS row under Columns and select the "Edit" button

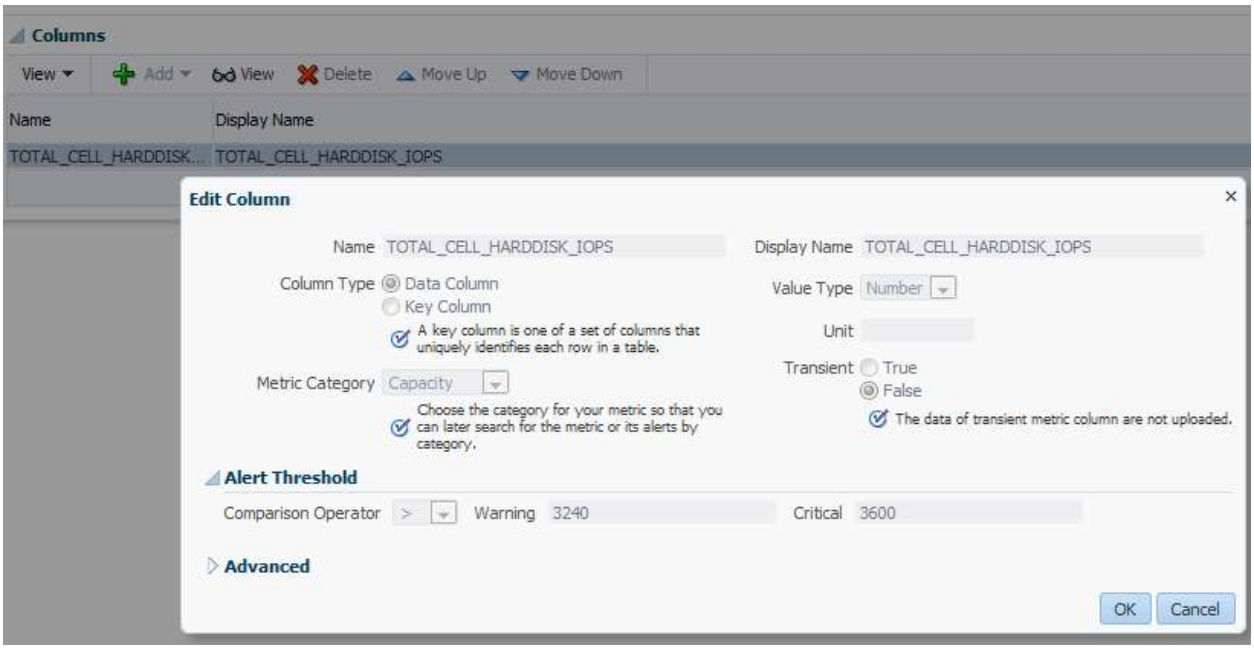
Specify the Display Name

This should be the same as the metric extension name listed above.

Also specify the warning and critical threshold values for the metric.

These thresholds will be applied to every target the metric extension is deployed to. Once deployed to a target the threshold can be adjusted on the metric and collection setting on the individual target page. This may be necessary as different Exadata environments will have different thresholds. For example IOPS thresholds for an X5 would be greater than an X2 Dbmachine.

Select the "Next" button



On the Test Screen:

Select the “Finish” button

The remaining Metric Extensions will be created the same way. The only difference will be the Name ME$, Display Name, and SQL Query. The chart below lists the remaining information. Use this information to create the remaining Metric Extensions.

*******It is critical that all the metric names be EXACTLY as listed below otherwise the composite metric will not work properly.**

Name ME$	Display Name	SQL
Total_Cell_Flash_IOPS	Total Cell Flash IOPS	SELECT c.target_guid, SUM(c.value) total_cell_flashdisk_iops FROM sysman.mgmt$metric_current c, sysman.mgmt$availability_current a WHERE c.target_type = 'oracle_exadata' AND c.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND c.key_value = 'FlashDisk' AND c.metric_column in ('sum_cd_read_iops','sum_cd_write_iops') AND c.target_guid = a.target_guid

		AND a.availability_status_code = 1 AND SYSTIMESTAMP AT TIME ZONE 'UTC'- FROM_TZ(TO_TIMESTAMP(TO_CHAR(c.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-mon-yyyy hh24:mi:ss'),c.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000' GROUP BY c.target_guid
Total_Cell_Hard Disk_IOPS	Total Cell HardDisk IOPS	SELECT c.target_guid, SUM(c.value) total_cell_harddisk_iops FROM sysman.mgmt$metric_current c, sysman.mgmt$availability_current a WHERE c.target_type = 'oracle_exadata' AND c.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND c.key_value = 'HardDisk' AND c.metric_column in ('sum_cd_read_iops','sum_cd_write_iops') AND c.target_guid = a.target_guid AND a.availability_status_code = 1 AND SYSTIMESTAMP AT TIME ZONE 'UTC'- FROM_TZ(TO_TIMESTAMP(TO_CHAR(c.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-mon-yyyy hh24:mi:ss'),c.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000' GROUP BY c.target_guid
Total_Cell_Flash _Throughput	Total Cell Flash Throughput	SELECT c.target_guid, SUM(c.value) total_cell_flashdisk_tput FROM sysman.mgmt$metric_current c, sysman.mgmt$availability_current a WHERE c.target_type = 'oracle_exadata' AND c.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND c.key_value = 'FlashDisk' AND c.metric_column in ('sum_cd_read_throughput','sum_cd_write_throughput') AND c.target_guid = a.target_guid AND a.availability_status_code = 1 AND SYSTIMESTAMP AT TIME ZONE 'UTC'- FROM_TZ(TO_TIMESTAMP(TO_CHAR(c.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-mon-

		yyyy hh24:mi:ss'),c.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000' GROUP BY c.target_guid
Total_Cell_Hard_Disk_Throughput	Total Cell Hard Disk Throughput	SELECT c.target_guid, SUM(c.value) total_cell_harddisk_tput FROM sysman.mgmt$metric_current c, sysman.mgmt$availability_current a WHERE c.target_type = 'oracle_exadata' AND c.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND c.key_value = 'HardDisk' AND c.metric_column in ('sum_cd_read_throughput','sum_cd_write_throughput') AND c.target_guid = a.target_guid AND a.availability_status_code = 1 AND SYSTIMESTAMP AT TIME ZONE 'UTC'-FROM_TZ(TO_TIMESTAMP(TO_CHAR(c.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-mon-yyyy hh24:mi:ss'),c.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000' GROUP BY c.target_guid
Total_Avg_Flash_Response_Time	Total Avg Flash Response Time	SELECT mt.target_guid,(wrsp.value*decode(iopsiv.total_iops,0,0,wiops.value/iopsiv.total_iops)) + (rrsp.value*decode(iopsiv.total_iops,0,0,riops.value/iopsiv.total_iops)) total_cell_flashdisk_rsp_time FROM sysman.mgmt$target mt, sysman.mgmt$metric_current wiops, sysman.mgmt$metric_current riops,sysman.mgmt$metric_current wrsp, sysman.mgmt$metric_current rrsp, sysman.mgmt$availability_current a, (select smmc.target_guid, sum(value) total_iops from sysman.mgmt$metric_current smmc where smmc.target_type = 'oracle_exadata' AND smmc.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND smmc.metric_column in ('sum_cd_read_iops','sum_cd_write_iops')

		AND smmc.key_value = 'FlashDisk' group by smmc.target_guid) iopsiv WHERE wiops.target_type = 'oracle_exadata' AND wiops.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND wiops.key_value = 'FlashDisk' AND wiops.metric_column = 'sum_cd_write_iops' AND SYSTIMESTAMP AT TIME ZONE 'UTC'- FROM_TZ(TO_TIMESTAMP(TO_CHAR(wiops.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-mon-yyyy hh24:mi:ss'),wiops.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000' AND riops.target_type = 'oracle_exadata' AND riops.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND riops.key_value = 'FlashDisk' AND riops.metric_column = 'sum_cd_read_iops' AND SYSTIMESTAMP AT TIME ZONE 'UTC'- FROM_TZ(TO_TIMESTAMP(TO_CHAR(riops.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-mon-yyyy hh24:mi:ss'),riops.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000' AND wrsp.target_type = 'oracle_exadata' AND wrsp.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND wrsp.key_value = 'FlashDisk' AND wrsp.metric_column = 'avg_cd_write_latency' AND SYSTIMESTAMP AT TIME ZONE 'UTC'- FROM_TZ(TO_TIMESTAMP(TO_CHAR(wrsp.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-mon-yyyy hh24:mi:ss'),wrsp.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000' AND rrsp.target_type = 'oracle_exadata' AND rrsp.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND rrsp.key_value = 'FlashDisk'

		AND rrsp.metric_column = 'avg_cd_read_latency' AND SYSTIMESTAMP AT TIME ZONE 'UTC'- FROM_TZ(TO_TIMESTAMP(TO_CHAR(rrsp.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-mon-yyyy hh24:mi:ss'),rrsp.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000' AND mt.target_guid = a.target_guid AND iopsiv.target_guid = mt.target_guid AND a.availability_status_code = 1 AND wiops.target_guid = mt.target_guid AND riops.target_guid = mt.target_guid AND rrsp.target_guid = mt.target_guid AND wrsp.target_guid = mt.target_guid ORDER BY mt.target_guid
Total_Avg_Hard Disk_Response_ Time	Total Avg HardDisk Response Time	SELECT mt.target_guid, (wrsp.value*decode(iopsiv.total_iops,0,0,wiops.value/iopsiv.total_iops)) + (rrsp.value*decode(iopsiv.total_iops,0,0,riops.value/iopsiv.total_iops)) total_cell_harddisk_rsp_time FROM sysman.mgmt$target mt, sysman.mgmt$metric_current wiops, sysman.mgmt$metric_current riops,sysman.mgmt$metric_current wrsp, sysman.mgmt$metric_current rrsp, sysman.mgmt$availability_current a, (select smmc.target_guid, sum(value) total_iops from sysman.mgmt$metric_current smmc where smmc.target_type = 'oracle_exadata' AND smmc.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND smmc.metric_column in ('sum_cd_read_iops','sum_cd_write_iops') AND smmc.key_value = 'HardDisk' group by smmc.target_guid) iopsiv WHERE wiops.target_type = 'oracle_exadata'

		AND wiops.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND wiops.key_value = 'HardDisk' AND wiops.metric_column = 'sum_cd_write_iops' AND SYSTIMESTAMP AT TIME ZONE 'UTC'-FROM_TZ(TO_TIMESTAMP(TO_CHAR(wiops.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-mon-yyyy hh24:mi:ss'),wiops.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000' AND riops.target_type = 'oracle_exadata' AND riops.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND riops.key_value = 'HardDisk' AND riops.metric_column = 'sum_cd_read_iops' AND SYSTIMESTAMP AT TIME ZONE 'UTC'-FROM_TZ(TO_TIMESTAMP(TO_CHAR(riops.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-mon-yyyy hh24:mi:ss'),riops.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000' AND wrsp.target_type = 'oracle_exadata' AND wrsp.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND wrsp.key_value = 'HardDisk' AND wrsp.metric_column = 'avg_cd_write_latency' AND SYSTIMESTAMP AT TIME ZONE 'UTC'-FROM_TZ(TO_TIMESTAMP(TO_CHAR(wrsp.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-mon-yyyy hh24:mi:ss'),wrsp.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000' AND rrsp.target_type = 'oracle_exadata' AND rrsp.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND rrsp.key_value = 'HardDisk' AND rrsp.metric_column = 'avg_cd_read_latency' AND SYSTIMESTAMP AT TIME ZONE 'UTC'-FROM_TZ(TO_TIMESTAMP(TO_CHAR(rrsp.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-

		mon-yyyy hh24:mi:ss'),rrsp.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000' AND mt.target_guid = a.target_guid AND iopsiv.target_guid = mt.target_guid AND a.availability_status_code = 1 AND wiops.target_guid = mt.target_guid AND riops.target_guid = mt.target_guid AND rrsp.target_guid = mt.target_guid AND wrsp.target_guid = mt.target_guid ORDER BY mt.target_guid
Total_Cell_Flash_Load	Total Cell Flash Load	SELECT c.target_guid, c.value total_cell_flashdisk_io_load FROM sysman.mgmt$metric_current c, sysman.mgmt$availability_current a WHERE c.target_type = 'oracle_exadata' AND c.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND c.key_value = 'FlashDisk' AND c.metric_column = 'avg_cd_io_load' AND c.target_guid = a.target_guid AND a.availability_status_code = 1 AND SYSTIMESTAMP AT TIME ZONE 'UTC'-FROM_TZ(TO_TIMESTAMP(TO_CHAR(c.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-mon-yyyy hh24:mi:ss'),c.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000'
Total_Cell_HardDisk_Load	Total Cell HardDisk Load	SELECT c.target_guid, c.value total_cell_harddisk_io_load FROM sysman.mgmt$metric_current c, sysman.mgmt$availability_current a WHERE c.target_type = 'oracle_exadata' AND c.metric_name = 'Aggregated_HardNFlashDisk_Metric' AND c.key_value = 'HardDisk'

		AND c.metric_column = 'avg_cd_io_load' AND c.target_guid = a.target_guid AND a.availability_status_code = 1 AND SYSTIMESTAMP AT TIME ZONE 'UTC'-FROM_TZ(TO_TIMESTAMP(TO_CHAR(c.collection_timestamp,'dd-mon-yyyy hh24:mi:ss'),'dd-mon-yyyy hh24:mi:ss'),c.timezone_region) AT TIME ZONE 'UTC' < '0 01:00:00.000'
Exadata_Storage_Server_Flash Disk_IO_Health	Exadata Storage Server FlashDisk IO Health	select mt.target_guid,NVL(mmciv.exceptions,0)exceptions from mgmt$target MT left outer join (select mmc.target_guid, count(*) exceptions from SYSMAN.mgmt$metric_current mmc, mgmt$target_metric_settings tms where 1=1 and mmc.target_guid=tms.target_guid and mmc.metric_guid=tms.metric_guid and mmc.key_value=tms.key_value and tms.metric_name in ('ME$Total_Cell_Flash_Load','ME$Total_Cell_Flash_IOPS','ME$Total_Cell_Flash_Throughput','ME$Total_Avg_Flash_Response_Time') and to_number(mmc.value) > to_number(tms.critical_threshold) group by mmc.target_guid) MMCIV on (mt.target_guid=mmciv.target_guid) order by mt.target_name desc
Exadata_Storage_Server_HardDisk_I/O_Health	Exadata Storage Server HardDisk I/O Health	select mt.target_guid,NVL(mmciv.exceptions,0)exceptions from mgmt$target MT left outer join (select mmc.target_guid, count(*) exceptions from SYSMAN.mgmt$metric_current mmc, mgmt$target_metric_settings tms where 1=1 and mmc.target_guid=tms.target_guid and mmc.metric_guid=tms.metric_guid and mmc.key_value=tms.key_value and tms.metric_name in ('ME$Total_Cell_HardDisk_Load','ME$Total_Cell_HardDisk_IOPS','ME$Total_Cell_Hard_Disk_Throughput','ME$Total_Avg_HardDisk_Response_Time')

		/*and tms.metric_column in ('TOTAL_CELL_HARDDISK_IO_LOAD','TOTAL_IOPS', 'TOTAL_CELL_THROUGHPUT','AVG_CELLDISK_RESPONSE_TIME')*/ and to_number(mmc.value) > to_number(tms.critical_threshold) group by mmc.target_guid) MMCIV on (mt.target_guid=mmciv.target_guid) order by mt.target_name desc

Figure 9

Now save each Metrics Extension as deployable drafts. They can now be deployed to targets as shown in Figure 10. It is required that all metric extensions be deployed to all Storage Server targets in monitored dbmachine targets. Detailed instruction of Metric Extension Lifecycle can be found in the Enterprise Manager Administrator Guide.

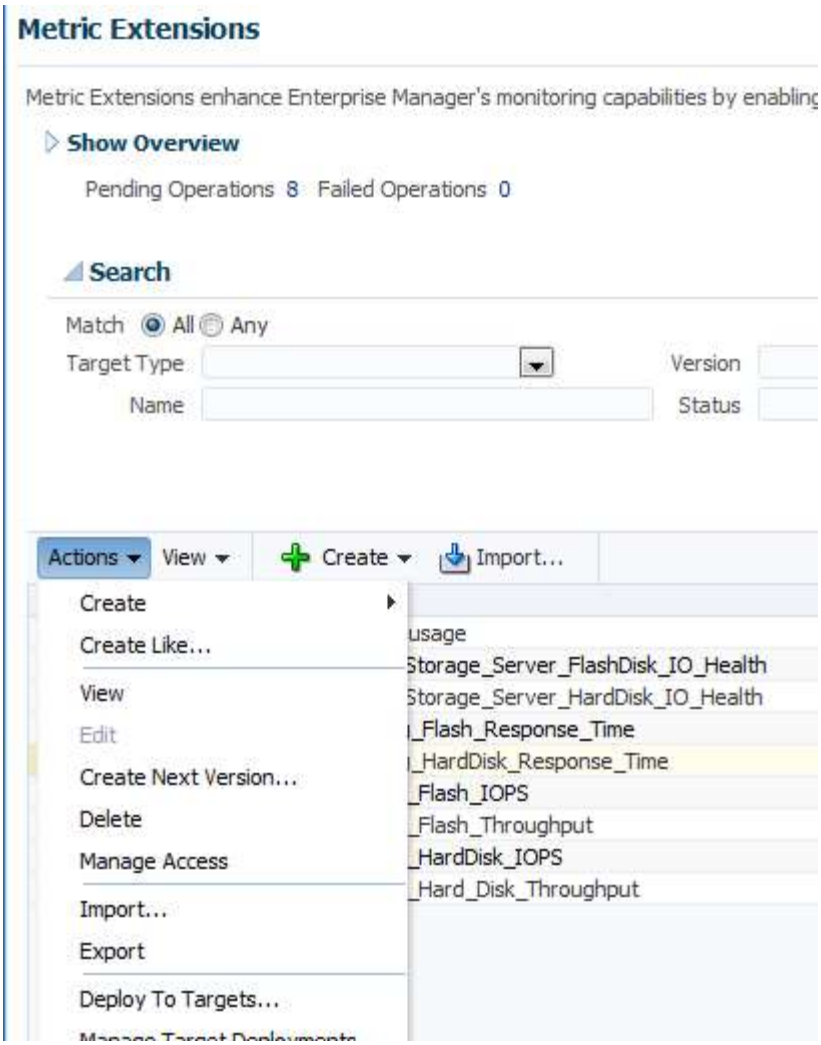


Figure 10

By default, after deployment the collection threshold for the Metric Extensions will be fifteen minutes. If more granular data is required this can be adjusted to a lower value in the target's metric collection setting page. Be aware that lowering the threshold below the default values can place additional load on the agent deployed on the Exadata environment as well as Enterprise Manager Repository. Always test these changes in a non-production environment.

Services

Many times, looking at one Enterprise Manager Target isn't enough to gauge the health of an Environment. An Enterprise Manager Service allows for multiple targets to be viewed holistically to determine the environment's health. Monitoring Exadata Storage Server capacity necessitates that we evaluate all the Storage Servers as a complete Cell Grid in addition to monitoring individual Storage Servers. Services provide us with this ability.

In addition to monitoring multiple targets, to complete the holistic approach to monitoring the Storage Server Grid, many times one metric exceeding its threshold is not significant enough to conclude that one or more Storage Servers are having issues or are at capacity. When we look at multiple metrics and how they interact with each other, a more accurate picture starts to develop that allows for an accurate diagnosis and reduces the number of false alerts raised.

The Service will be created to monitor the overall status of the Exadata Storage Server Grid

Targets → Services

Now create start the Create Service workflow:

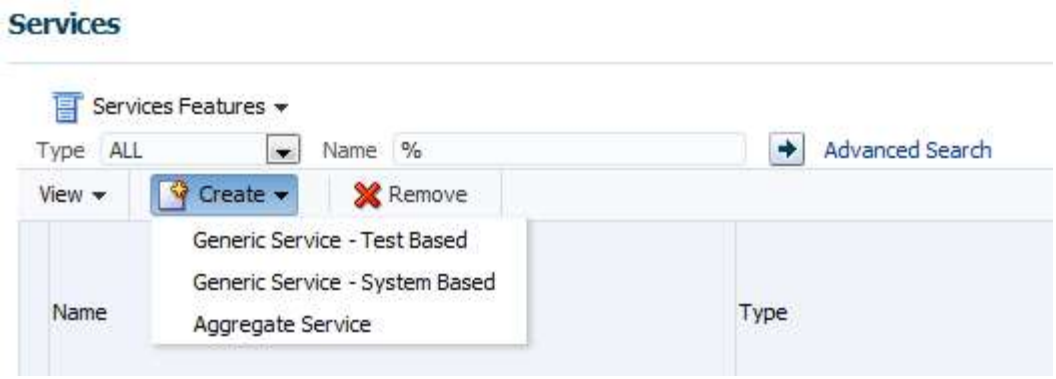


Figure 11

Create→ Generic Service - System Based

On the "Create Generic Service: General" page enter the following:

Name: such as "Exadata Storage Grid <DBRACK>"

Time Zone

Select the "Next" button.

On the "Create Generic Service: System" page:

Select the magnifying glass icon by the System label (see Figure 12).

In the Search window:

Select Target Type of “Oracle Exadata Storage Server Grid”

Select the “Search” button.

Select the Exadata Grid for which the service is being created.

Click the “Select” button.

Select the “Submit” button at the top of the “Create Generic Service: System” page.

Figure 12

On the “Services” page:

Click on the Service that was just created.

On the “Grid Service” page:

Select Generic Service→Administration→Performance Metrics

On the “Performance Metrics” page:

Select the “Go” button to add a new metric

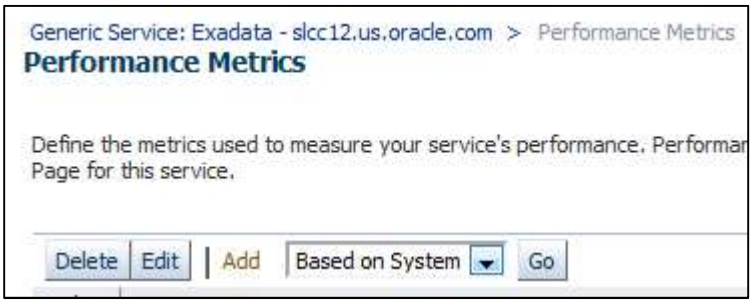


Figure 13

On the "Add Performance Metric based on System" page enter the following:

Target Type: "Oracle Exadata Storage Server (Indirect Member)"

Metric: "Exadata Hard Disk Total IOPS"

Select the "Aggregate the metrics across multiple components" radio button (make sure all the correct Cell Servers are checked)

Aggregate Function: Sum

Click the "Continue" button

Create the remaining Service Performance Metrics using the same approach as above. The data needed for the workflow is provided in the below table.

Metric	Aggregate Function	Warning Threshold	Critical
Total Cell FlashDisk IOPS	Sum	None	None
Total Cell HardDisk IOPS	Sum	None	None
Total Cell FlashDisk Throughput	Sum	None	None
Total Cell HardDisk Throughput	Sum	None	None
Total Avg FlashDisk Response Time	Average	None	None
Total Avg HardDisk Response Time	Average	None	None
Total Avg HardDisk IO Load	Average	None	None
Total Avg FlashDisk IO Load	Average	None	None

Flash KPIs Exceeding Thresholds	Average	1	2
HardDisk KPIs Exceeding Thresholds	Average	1	2

Figure 14

Now that the Service is created, incidents will be created whenever warning or critical thresholds are crossed. These incidents can be viewed in Incident Manager or the Service's "Performance/Incidents" page. If external communication (email, snmp, ticketing, etc.) is required ensure that the appropriate notifications are setup. Information on setting up notifications can be found at: http://docs.oracle.com/cd/E24628_01/doc.121/e24473/notification.htm#EMADM9066.

In addition to displaying incidents, the Service's "Performance/Incident" page provides usage information on metrics defined for the Service (see Figure 15). This allows for a quick one-stop view for evaluating the Service's performance.

Figure 15

Conclusion

With the creation of these metrics and metrics extensions, we've created an overall picture in Enterprise Manager enabling administrators to effectively evaluate the state of the Exadata Cell Grid. By setting up appropriate alerts and thresholds, administrators will also be proactively notified of potential issues before they impact business service level agreements.

ORACLE

Exadata Health and Resource Usage Monitoring

August 2015
Author: Mike Chafin
Contributors: Curtis Dinkel, Werner De Gruyter, Jim Viscusi
Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.
Worldwide Inquiries: