Exadata Performance and AWR

Exadata Performance Diagnostics with AWR

March, 2024, Version 2.0

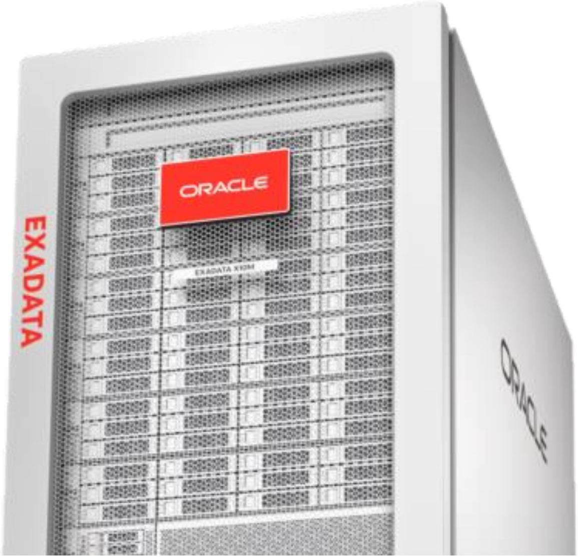
Public

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of contents



Introduction

Oracle Exadata is engineered to deliver dramatically better performance, cost effectiveness, and availability for Oracle databases. Exadata features a modern cloud-based architecture with scale-out high-performance database servers, scale-out intelligent storage servers with state-of-the-art flash drives, and an ultra-fast Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCE) internal fabric. Unique software algorithms in Exadata enable database intelligence in storage, compute, and RoCE networking to deliver higher performance and capacity at a lower cost than other platforms.

Exadata runs all types of database workloads including Online Transaction Processing (OLTP), Data Warehousing (DW), In-Memory Analytics, and consolidation of mixed workloads. Exadata can be deployed on-premises as the foundation for a private database cloud, or can be acquired using a subscription model in Oracle Cloud Infrastructure (OCI) with Exadata Database Service on Dedicated Infrastructure (ExaDB-D) or Exadata Database Service on Cloud@Customer (ExaDB-C@C), with all infrastructure managed by Oracle.

As customers around the world make Exadata the platform of choice for enterprise database deployment and consolidate an increasing number of databases onto Exadata systems, monitoring the performance of these databases from an Exadata system standpoint becomes more important than ever. This technical brief outlines how the Oracle Database Automatic Workload Repository (AWR) feature can be used in conjunction with Exadata to monitor and analyze database performance characteristics from an Exadata perspective.

The contents of this technical brief apply to all Exadata deployments – whether on-premises, ExaDB-D, or ExaDB-C@C. Specifically, with Exadata in OCI, since customers have complete administrative control over their databases, the Exadata-specific AWR capabilities apply in the same manner as when these databases are deployed on-premises.

AWR Overview

The Automatic Workload Repository (AWR), introduced in Oracle Database 10g, is the most widely used performance diagnostics tool for Oracle Database. AWR collects, processes, and maintains database performance statistics data for problem detection and self-tuning purposes. This process of data collection is repeated on a regular time interval and the results are captured in an AWR snapshot. The delta values, calculated from the data captured by the AWR snapshot, represent the changes for each statistic over the interval, and can be viewed through an AWR report for further analysis. By default, the AWR snapshots are taken at hourly intervals, and the snapshots are retained for eight days. It is recommended to increase the retention period to allow for monthly (31 days) or quarterly (90 days) comparisons, depending on your reporting and retention requirements. AWR reports can also be generated on-demand for specific time intervals.[1]

Performance and Scope

When analyzing performance issues, it is important to understand the scope of the performance problem, and to ensure that the data and the tools used for analysis matches the scope of the problem.

For example, if an issue is localized to a small set of users or SQL statements, a SQL Monitor report will have data that is relevant to the scope of the problem. A SQL Monitor report provides detailed statistics about a single execution of a SQL statement or a Database (DB) operation.

If the performance issues are instance-wide or database-wide, then an AWR report will contain data and statistics for the instance or the entire database. Active Session History (ASH), which samples active sessions, can be used for instance-wide, database-wide, and localized issues. ASH collects data across multiple dimensions that can be used to filter the data.

Maintaining Baselines

A statistical baseline is a collection of statistics usually taken over an interval when the system is performing well. The baselines can be used to diagnose performance problems by comparing statistics captured in a baseline to those captured during periods of poor performance. This enables the identification of statistics that may have increased significantly, that could be the cause of the problem.

It is recommended to collect baselines during normal processing periods, as well as critical time frames such as month-end or year-end processing. The baselines should include AWR data[2], a SQL Monitor report of a few key SQL statements, along with additional statistics from the storage servers (ExaWatcher and cell metric history)[3].

Exadata Support in AWR

Exadata support in AWR was introduced with Oracle Database 12.1.0.2.0 and Exadata System Software 12.1.2.1.0. Including Exadata statistics in the AWR report gives more visibility into the storage tier through a unified report, without having to collect additional data from the storage servers. This is of particular interest to ExaDB-D and ExaDB-C@C customers that do not have access to the storage servers.

The Exadata statistics are only available in the HTML and Active-HTML formats of the AWR Instance report, and the AWR Global report from CDB$ROOT. The Exadata statistics are not available in the text format of the report; nor are they available in the PDB-level AWR report. The Exadata sections in the report are also constantly being enhanced, as new features are included in new releases of Exadata software.[4] Exadata statistics are also available in AWR reports in Enterprise Manager. The References section in this technical brief provides a list of documents describing how to manage Exadata with Enterprise Manager.

It is also important to note that with the addition of Exadata storage level statistics in the AWR report, the performance tuning methodology does not change. Users should first look at DB time, and address performance

[1] Refer to "Gathering Database Statistics" in *Oracle Database Performance Tuning Guide* for more details on AWR.

[2] Baselines should include the actual AWR data, not just an AWR report.

[3] *Oracle Exadata System Software – Monitoring Exadata* has extensive information on AWR, ExaWatcher, and cell metric history.

[4] As the Exadata sections are constantly being enhanced, the version you see on your systems may not match the screenshots in this technical brief.

issues by analyzing the top consumers of DB time. Only when it has been determined that there may be IO issues should one start looking at the Exadata sections. The Exadata sections are meant to complement, rather than replace, existing tools and methodologies.

Challenges and AWR Exadata Solutions

A common challenge for Oracle DBAs is to better analyze and understand database performance characteristics that are directly related to the underlying infrastructure such as servers, network, and storage. Optimal database performance is dependent on an optimal configuration of the infrastructure. However, if the infrastructure is mis-configured or there are faulty components, accurately diagnosing resulting database performance issues and correlating them to the specific component is not an easy task.

The value proposition of an engineered system such as Exadata is that Oracle DBAs are now able to integrate statistics that are collected and maintained on the Exadata storage servers directly and automatically into AWR. As will be seen later in this paper, this diagnosis process is remarkably efficient compared to the time and resources that would be spent otherwise if these databases were deployed on a generic infrastructure. Oracle DBAs also benefit from the fact that Exadata specific AWR content continues to get enhanced as the core Exadata platform is enhanced with additional software and hardware capabilities.

The following sections outline specific scenarios where the Exadata-specific AWR capabilities may be leveraged.

Consolidated Environments

Exadata storage adds a new scope when analyzing performance issues. The storage subsystem may be shared by multiple databases, and as such, the statistics that come from the storage layer are for the entire system – i.e. it is not constrained to a single database or a single database instance.

In an Exadata system running several databases, it is important to identify the databases that could be consuming a significant amount of the IO bandwidth on the system, and thus affecting other databases on the system. It is strongly recommended to leverage Exadata's built-in IO Resource Management (IORM) capabilities such that IO requests within an Exadata Storage Server can be prioritized and scheduled based on configured resource plans. Please refer to "Managing IO Resource Management" in *Exadata System Software User's Guide* for more details on IORM.

The AWR report includes a **Top Databases** section[5], which shows IO requests and IO throughput. This section helps to compare each database's IO resource consumption. A subset of databases are captured in AWR snapshots based on an internal metric to identify the top N databases in each of the storage servers. The AWR report on Exadata shows the top databases by IO Requests and IO Throughput. As seen in Figure 1, the data is broken down by IOs on flash devices and IOs on hard disks. In

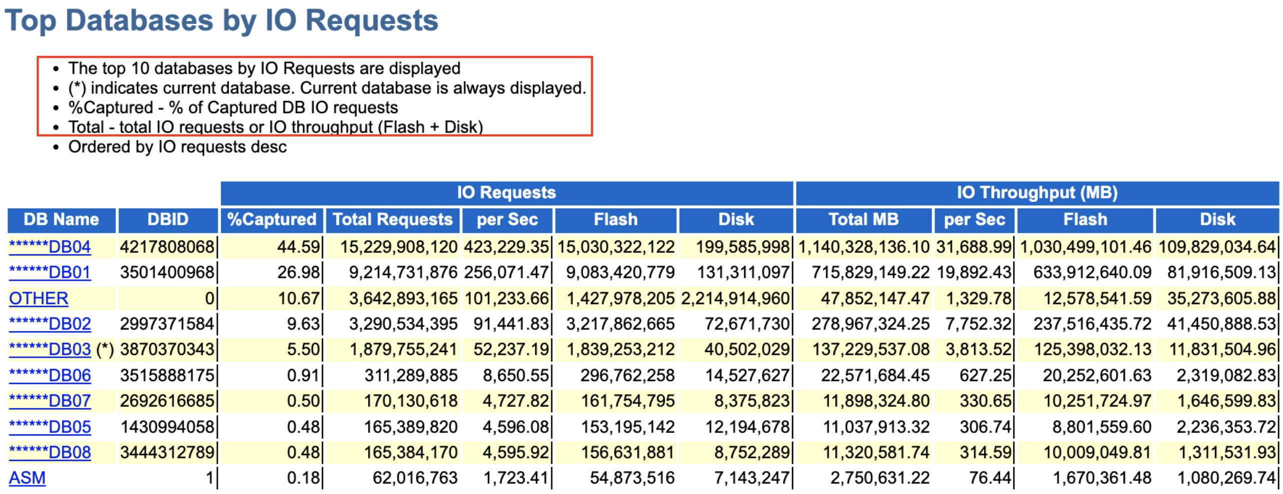
Figure 2, the requests are further broken down between small and large IOs, along with the average IO latencies, and average IORM queue times for the IOs.

Notice in Figure 1 that the report shows percent captured (%Captured) instead of percent total (%Total), as not all statistics for all databases are captured by AWR. This available data is aggregated for the entire system and per each storage cell. The current database, DB03, marked by (*), only accounts for 5 percent of the captured IOPs.

In

Figure 2, the DB03 database shows that IORM is queueing its large IOs on both flash and disk. This may indicate the use of an IORM plan to deprioritize IOs from this database.

Figure 1. Top Databases by IO Requests

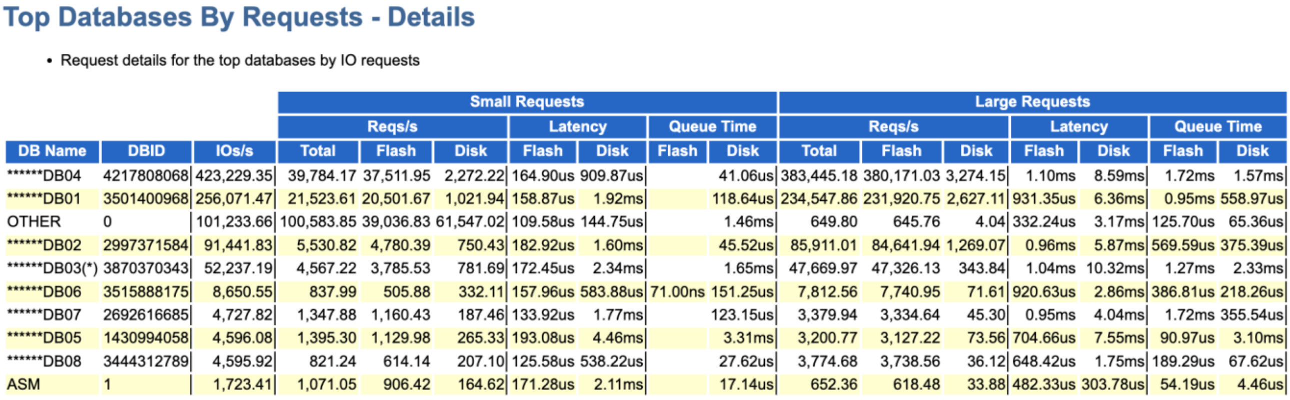


Top Databases by IO Requests

- The top 10 databases by IO Requests are displayed
- (*) indicates current database. Current database is always displayed.
- %Captured - % of Captured DB IO requests
- Total - total IO requests or IO throughput (Flash + Disk)
- Ordered by IO requests desc

		IO Requests					IO Throughput (MB)			
DB Name	DBID	%Captured	Total Requests	per Sec	Flash	Disk	Total MB	per Sec	Flash	Disk
******DB04	4217808068	44.59	15,229,908,120	423,229.35	15,030,322,122	199,585,998	1,140,328,136.10	31,688.99	1,030,499,101.46	109,829,034.64
******DB01	3501400968	26.98	9,214,731,876	256,071.47	9,083,420,779	131,311,097	715,829,149.22	19,892.43	633,912,640.09	81,916,509.13
OTHER	0	10.67	3,642,893,165	101,233.66	1,427,978,205	2,214,914,960	47,852,147.47	1,329.78	12,578,541.59	35,273,605.88
******DB02	2997371584	9.63	3,290,534,395	91,441.83	3,217,862,665	72,671,730	278,967,324.25	7,752.32	237,516,435.72	41,450,888.53
******DB03 (*)	3870370343	5.50	1,879,755,241	52,237.19	1,839,253,212	40,502,029	137,229,537.08	3,813.52	125,398,032.13	11,831,504.96
******DB06	3515888175	0.91	311,289,885	8,650.55	296,762,258	14,527,627	22,571,684.45	627.25	20,252,601.63	2,319,082.83
******DB07	2692616685	0.50	170,130,618	4,727.82	161,754,795	8,375,823	11,898,324.80	330.65	10,251,724.97	1,646,599.83
******DB05	1430994058	0.48	165,389,820	4,596.08	153,195,142	12,194,678	11,037,913.32	306.74	8,801,559.60	2,236,353.72
******DB08	3444312789	0.48	165,384,170	4,595.92	156,631,881	8,752,289	11,320,581.74	314.59	10,009,049.81	1,311,531.93
ASM	1	0.18	62,016,763	1,723.41	54,873,516	7,143,247	2,750,631.22	76.44	1,670,361.48	1,080,269.74

Figure 2. Top Databases by Requests - Details



Top Databases By Requests - Details

- Request details for the top databases by IO requests

			Small Requests							Large Requests						
			Reqs/s			Latency		Queue Time		Reqs/s			Latency		Queue Time	
DB Name	DBID	IOs/s	Total	Flash	Disk	Flash	Disk	Flash	Disk	Total	Flash	Disk	Flash	Disk	Flash	Disk
******DB04	4217808068	423,229.35	39,784.17	37,511.95	2,272.22	164.90us	909.87us		41.06us	383,445.18	380,171.03	3,274.15	1.10ms	8.59ms	1.72ms	1.57ms
******DB01	3501400968	256,071.47	21,523.61	20,501.67	1,021.94	158.87us	1.92ms		118.64us	234,547.86	231,920.75	2,627.11	931.35us	6.36ms	0.95ms	558.97us
OTHER	0	101,233.66	100,583.85	39,036.83	61,547.02	109.58us	144.75us		1.46ms	649.80	645.76	4.04	332.24us	3.17ms	125.70us	65.36us
******DB02	2997371584	91,441.83	5,530.82	4,780.39	750.43	182.92us	1.60ms		45.52us	85,911.01	84,641.94	1,269.07	0.96ms	5.87ms	569.59us	375.39us
******DB03(*)	3870370343	52,237.19	4,567.22	3,785.53	781.69	172.45us	2.34ms		1.65ms	47,669.97	47,326.13	343.84	1.04ms	10.32ms	1.27ms	2.33ms
******DB06	3515888175	8,650.55	837.99	505.88	332.11	157.96us	583.88us	71.00ns	151.25us	7,812.56	7,740.95	71.61	920.63us	2.86ms	386.81us	218.26us
******DB07	2692616685	4,727.82	1,347.88	1,160.43	187.46	133.92us	1.77ms		123.15us	3,379.94	3,334.64	45.30	0.95ms	4.04ms	1.72ms	355.54us
******DB05	1430994058	4,596.08	1,395.30	1,129.98	265.33	193.08us	4.46ms		3.31ms	3,200.77	3,127.22	73.56	704.66us	7.55ms	90.97us	3.10ms
******DB08	3444312789	4,595.92	821.24	614.14	207.10	125.58us	538.22us		27.62us	3,774.68	3,738.56	36.12	648.42us	1.75ms	189.29us	67.62us
ASM	1	1,723.41	1,071.05	906.42	164.62	171.28us	2.11ms		17.14us	652.36	618.48	33.88	482.33us	303.78us	54.19us	4.46us

Uneven Workload on Cells or Disks

Exadata is designed to evenly distribute workloads across all storage servers and disks. If a storage server or disk is performing more work compared to its peers, it has the potential to cause performance problems.

[5] If security is a concern, there is a cell attribute, dbPerfDataSuppress, that can be used to suppress databases from appearing in the v$cell_db view of other databases, and the subsequent AWR views that capture the v$cell_db data. The IOs of databases that are listed in dbPerfDataSuppress will be included in "OTHER" when the view is queried from a different database. Please refer to the Oracle Exadata System Software User's Guide on listing, altering, and describing cell attributes.

The Exadata AWR report performs outlier analysis, using several metrics to compare devices against its peers. The devices are grouped and compared by device type and size, as different device types do not have the same performance characteristics. For example, a flash device is expected to perform very differently from a hard disk. Similarly, a 1.6 TB flash device may not be able to sustain the same amount of IO as a 6.4 TB flash device.

The statistics used for outlier analysis include OS statistics, like iostat, which include IOPs, throughput, percent utilization, service time, and queue time[6]. The outlier analysis also includes cell server statistics, which break down IOPs, throughput, and latencies by the type of IO (read or write), and the size of the IO (small or large).

The Exadata AWR report identifies if a system has reached its maximum capacity. The maximum values used in the report are queried from the cell and are consistent with what is published in the Exadata data sheets. Since customer workloads will vary, the reported maximum numbers are meant to be used as guidelines rather than hard rules.

Automatic Hard Disk Scrub and Repair (scrub) is an Exadata feature that proactively inspects the sectors on the disks for physical errors and, in so doing, can detect issues that built-in hard drive features may not. Scrubbing is an automated process on Exadata that kicks in when the disks are idle (less than 25% busy) so as not to impact database performance and is set on a bi-weekly schedule by default.[7]

When Exadata scrub is running, the number of reads will typically exceed the maximum IOPs. Scrub issues sequential 16 KB reads. The Exadata software is designed to prioritize client IO over scrub IO. When client IOs are issued, the scrub IO will back off to allow the client IO to proceed, so scrub is not expected to impact client IOs.

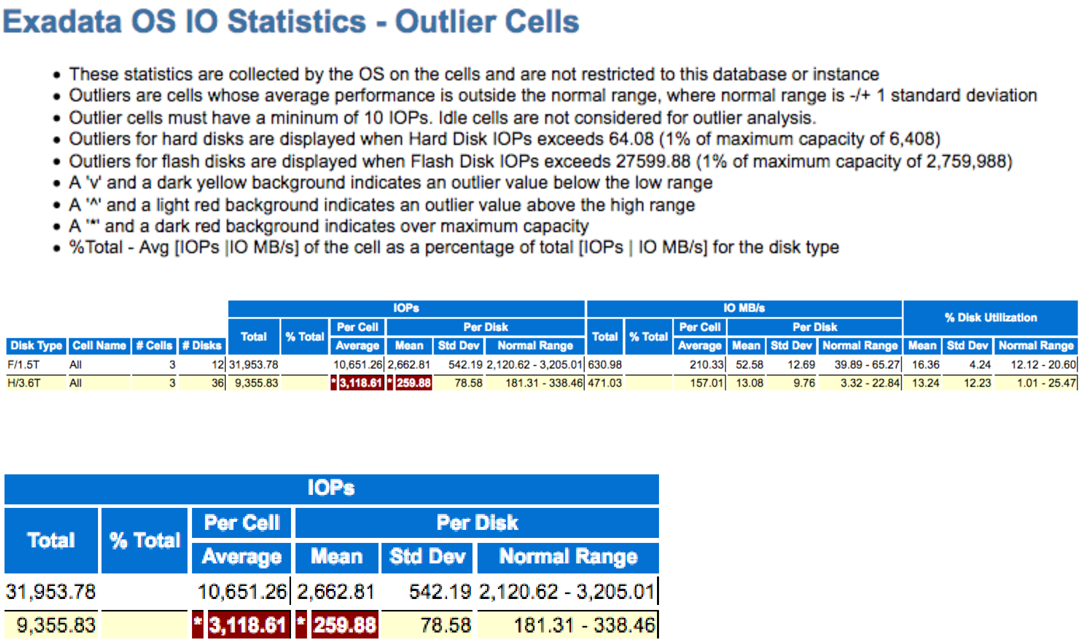
[6] Queue time reported as part of OS statistics is the device queue time, not IORM queue time.

[7] https://blogs.oracle.com/exadata/post/exadata-disk-scrubbing

Figure 3 shows an example of outlier analysis of storage cells. There are no outliers in this example, but the report has identified that the hard disks may be at maximum IOPs capacity, as indicated by the * and the dark red background. The maximum for the system is 6,408 IOPs for hard disks, and the report is currently showing 9,355.83 IOPs.

Figure 4 shows an example of outlier analysis of disks. In this example, the report has identified that the hard disks are at maximum capacity. It has also identified two disks that are performing more IOPs, compared to their peers.

Figure 3. Exadata OS IO Statistics - Outlier Cells



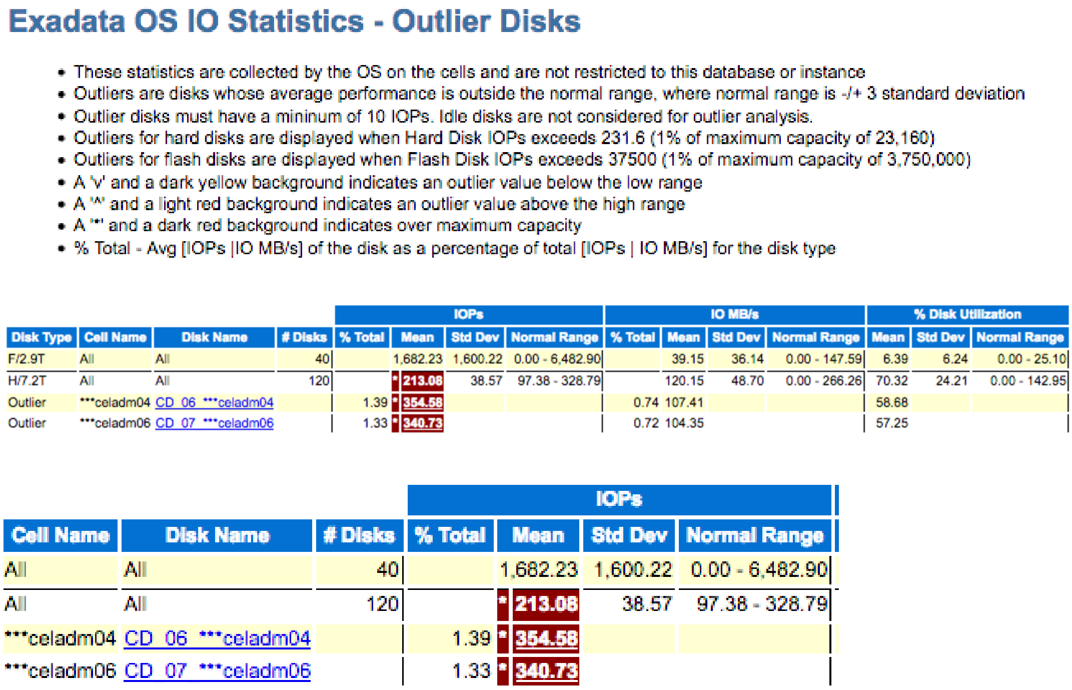
Exadata OS IO Statistics - Outlier Cells

- These statistics are collected by the OS on the cells and are not restricted to this database or instance
- Outliers are cells whose average performance is outside the normal range, where normal range is -/+ 1 standard deviation
- Outlier cells must have a mininum of 10 IOPs. Idle cells are not considered for outlier analysis.
- Outliers for hard disks are displayed when Hard Disk IOPs exceeds 64.08 (1% of maximum capacity of 6,408)
- Outliers for flash disks are displayed when Flash Disk IOPs exceeds 27599.88 (1% of maximum capacity of 2,759,988)
- A 'v' and a dark yellow background indicates an outlier value below the low range
- A '^' and a light red background indicates an outlier value above the high range
- A '*' and a dark red background indicates over maximum capacity
- %Total - Avg [IOPs |IO MB/s] of the cell as a percentage of total [IOPs | IO MB/s] for the disk type

				IOPs						IO MB/s						% Disk Utilization		
				Total	% Total	Per Cell	Per Disk			Total	% Total	Per Cell	Per Disk					
Disk Type	Cell Name	# Cells	# Disks			Average	Mean	Std Dev	Normal Range			Average	Mean	Std Dev	Normal Range	Mean	Std Dev	Normal Range
F/1.5T	All	3	12	31,953.78		10,651.26	2,662.81	542.19	2,120.62 - 3,205.01	630.98		210.33	52.58	12.69	39.89 - 65.27	16.36	4.24	12.12 - 20.60
H/3.6T	All	3	36	9,355.83		* 3,118.61	* 259.88	78.58	181.31 - 338.46	471.03		157.01	13.08	9.76	3.32 - 22.84	13.24	12.23	1.01 - 25.47

IOPs					
Total	% Total	Per Cell	Per Disk		
		Average	Mean	Std Dev	Normal Range
31,953.78		10,651.26	2,662.81	542.19	2,120.62 - 3,205.01
9,355.83		* 3,118.61	* 259.88	78.58	181.31 - 338.46

Figure 4. Exadata OS IO Statistics - Outlier Disks



Exadata OS IO Statistics - Outlier Disks

- These statistics are collected by the OS on the cells and are not restricted to this database or instance
- Outliers are disks whose average performance is outside the normal range, where normal range is -/+ 3 standard deviation
- Outlier disks must have a mininum of 10 IOPs. Idle disks are not considered for outlier analysis.
- Outliers for hard disks are displayed when Hard Disk IOPs exceeds 231.6 (1% of maximum capacity of 23,160)
- Outliers for flash disks are displayed when Flash Disk IOPs exceeds 37500 (1% of maximum capacity of 3,750,000)
- A 'v' and a dark yellow background indicates an outlier value below the low range
- A '^' and a light red background indicates an outlier value above the high range
- A '*' and a dark red background indicates over maximum capacity
- % Total - Avg [IOPs |IO MB/s] of the disk as a percentage of total [IOPs | IO MB/s] for the disk type

				IOPs				IO MB/s				% Disk Utilization		
Disk Type	Cell Name	Disk Name	# Disks	% Total	Mean	Std Dev	Normal Range	% Total	Mean	Std Dev	Normal Range	Mean	Std Dev	Normal Range
F/2.9T	All	All	40		1,682.23	1,600.22	0.00 - 6,482.90		39.15	36.14	0.00 - 147.59	6.39	6.24	0.00 - 25.10
H/7.2T	All	All	120		* 213.08	38.57	97.38 - 328.79		120.15	48.70	0.00 - 266.26	70.32	24.21	0.00 - 142.95
Outlier	***celadm04	CD_06_***celadm04		1.39	* 354.58			0.74	107.41			58.68		
Outlier	***celadm06	CD_07_***celadm06		1.33	* 340.73			0.72	104.35			57.25		

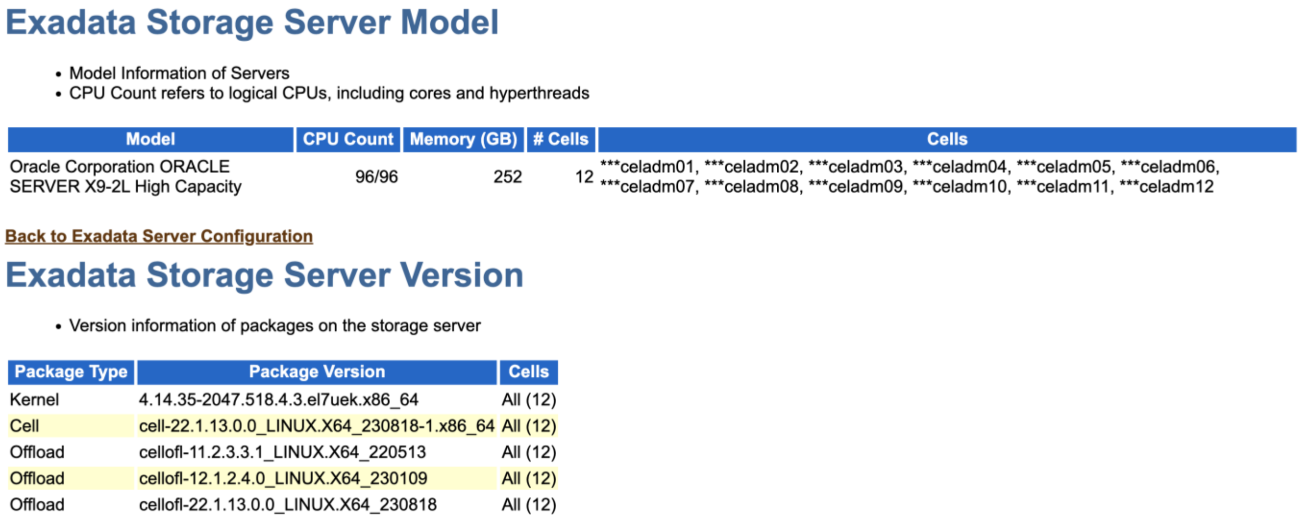
			IOPs			
Cell Name	Disk Name	# Disks	% Total	Mean	Std Dev	Normal Range
All	All	40		1,682.23	1,600.22	0.00 - 6,482.90
All	All	120		* 213.08	38.57	97.38 - 328.79
***celadm04	CD_06_***celadm04		1.39	* 354.58		
***celadm06	CD_07_***celadm06		1.33	* 340.73		

Configuration Differences

Configuration differences across the storage servers could potentially contribute to performance issues. The configuration issues could be differences in Oracle Smart Flash Cache (flash cache) or Oracle Smart Flash Log (flash log) sizes, or differences in the number of cell disks or grid disks in use.

The AWR report includes Exadata configuration information and identifies storage servers that are configured differently. Figure 5 shows an example of a system with identical storage server configurations. In the Exadata Configuration section, ‘All’ indicates an identical configuration across all storage servers. If there are differences between cell configurations, the cell names will be displayed, as seen in the **Exadata Storage Server Version** section in Figure 6.

Figure 5. System with identical storage server configurations.



Exadata Storage Server Model

- Model Information of Servers
- CPU Count refers to logical CPUs, including cores and hyperthreads

Model	CPU Count	Memory (GB)	# Cells	Cells
Oracle Corporation ORACLE SERVER X9-2L High Capacity	96/96	252	12	***celadm01, ***celadm02, ***celadm03, ***celadm04, ***celadm05, ***celadm06, ***celadm07, ***celadm08, ***celadm09, ***celadm10, ***celadm11, ***celadm12

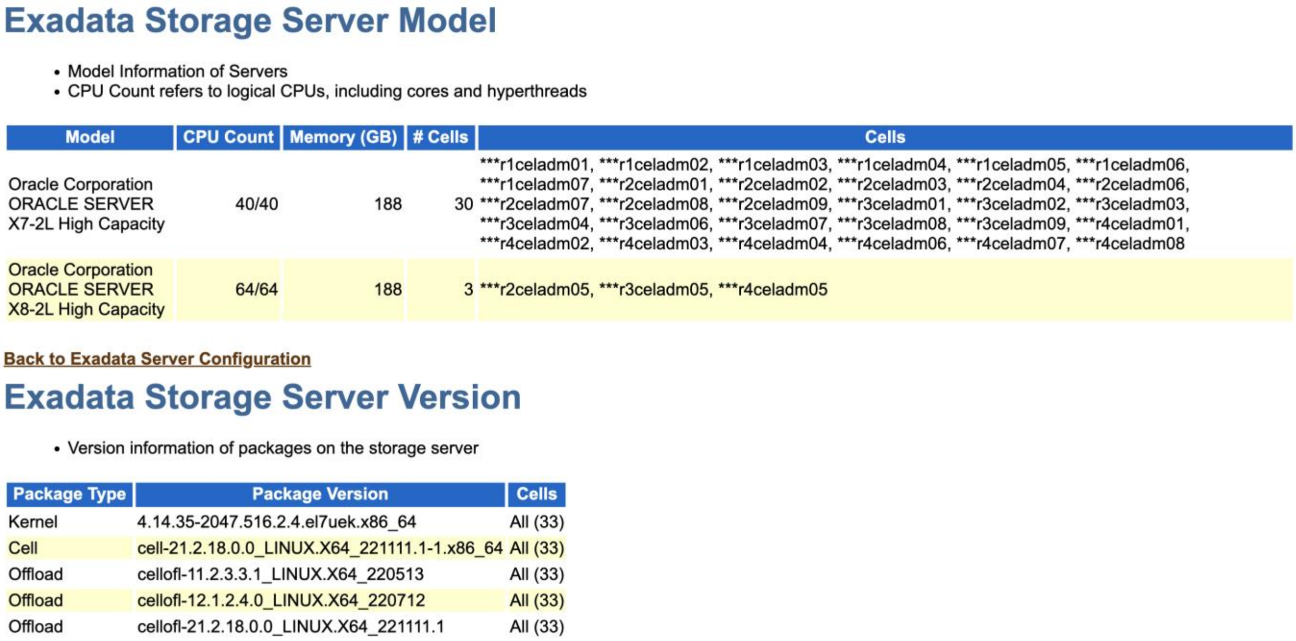
Back to Exadata Server Configuration

Exadata Storage Server Version

- Version information of packages on the storage server

Package Type	Package Version	Cells
Kernel	4.14.35-2047.518.4.3.el7uek.x86_64	All (12)
Cell	cell-22.1.13.0.0_LINUX.X64_230818-1.x86_64	All (12)
Offload	cellofl-11.2.3.3.1_LINUX.X64_220513	All (12)
Offload	cellofl-12.1.2.4.0_LINUX.X64_230109	All (12)
Offload	cellofl-22.1.13.0.0_LINUX.X64_230818	All (12)

Figure 6. System with differing storage server configurations.



Exadata Storage Server Model

- Model Information of Servers
- CPU Count refers to logical CPUs, including cores and hyperthreads

Model	CPU Count	Memory (GB)	# Cells	Cells
Oracle Corporation ORACLE SERVER X7-2L High Capacity	40/40	188	30	***r1celadm01, ***r1celadm02, ***r1celadm03, ***r1celadm04, ***r1celadm05, ***r1celadm06, ***r1celadm07, ***r2celadm01, ***r2celadm02, ***r2celadm03, ***r2celadm04, ***r2celadm06, ***r2celadm07, ***r2celadm08, ***r2celadm09, ***r3celadm01, ***r3celadm02, ***r3celadm03, ***r3celadm04, ***r3celadm06, ***r3celadm07, ***r3celadm08, ***r3celadm09, ***r4celadm01, ***r4celadm02, ***r4celadm03, ***r4celadm04, ***r4celadm06, ***r4celadm07, ***r4celadm08
Oracle Corporation ORACLE SERVER X8-2L High Capacity	64/64	188	3	***r2celadm05, ***r3celadm05, ***r4celadm05

Back to Exadata Server Configuration

Exadata Storage Server Version

- Version information of packages on the storage server

Package Type	Package Version	Cells
Kernel	4.14.35-2047.516.2.4.el7uek.x86_64	All (33)
Cell	cell-21.2.18.0.0_LINUX.X64_221111.1-1.x86_64	All (33)
Offload	cellofl-11.2.3.3.1_LINUX.X64_220513	All (33)
Offload	cellofl-12.1.2.4.0_LINUX.X64_220712	All (33)
Offload	cellofl-21.2.18.0.0_LINUX.X64_221111.1	All (33)

High Load

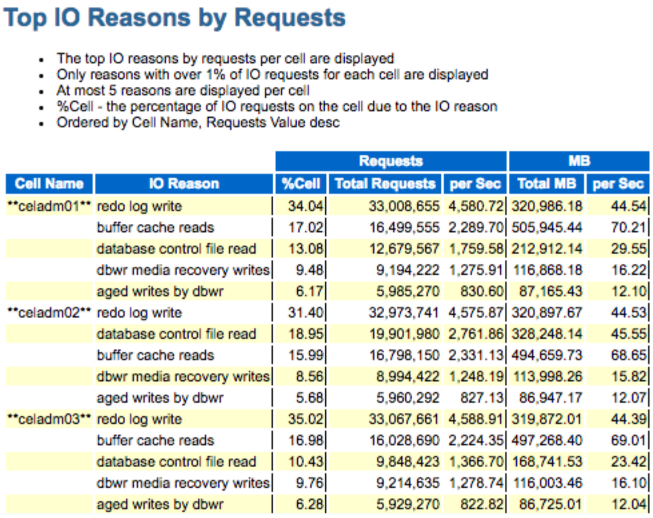
Changes in performance can be caused by increased load on the system. This can either be increased IO or CPU load on the storage servers. The increased IO load can be caused by maintenance activities, such as backups, or by changes in user IO, due to increased user workload or possible changes in execution plans.

On an Exadata system, there is additional information sent with each IO that includes the reason why the database is performing the IO. With the IO reason, we can easily determine if the additional IO load is caused by maintenance activity, or by increased database workload.

The reports also have visibility into the Exadata *Smart* features, including Smart Scan, Smart Flash Log, and Smart Flash Cache.

Figure 7 shows an example where the top IO requests are caused by typical database workload – redo log writes and buffer cache reads.

Figure 7. Top IO Reasons by Requests



Top IO Reasons by Requests

- The top IO reasons by requests per cell are displayed
- Only reasons with over 1% of IO requests for each cell are displayed
- At most 5 reasons are displayed per cell
- %Cell - the percentage of IO requests on the cell due to the IO reason
- Ordered by Cell Name, Requests Value desc

		Requests			MB	
Cell Name	IO Reason	%Cell	Total Requests	per Sec	Total MB	per Sec
celadm01	redo log write	34.04	33,008,655	4,580.72	320,986.18	44.54
	buffer cache reads	17.02	16,499,555	2,289.70	505,945.44	70.21
	database control file read	13.08	12,679,567	1,759.58	212,912.14	29.55
	dbwr media recovery writes	9.48	9,194,222	1,275.91	116,868.18	16.22
	aged writes by dbwr	6.17	5,985,270	830.60	87,165.43	12.10
celadm02	redo log write	31.40	32,973,741	4,575.87	320,897.67	44.53
	database control file read	18.95	19,901,980	2,761.86	328,248.14	45.55
	buffer cache reads	15.99	16,798,150	2,331.13	494,659.73	68.65
	dbwr media recovery writes	8.56	8,994,422	1,248.19	113,998.26	15.82
	aged writes by dbwr	5.68	5,960,292	827.13	86,947.17	12.07
celadm03	redo log write	35.02	33,067,661	4,588.91	319,872.01	44.39
	buffer cache reads	16.98	16,028,690	2,224.35	497,268.40	69.01
	database control file read	10.43	9,848,423	1,366.70	168,741.53	23.42
	dbwr media recovery writes	9.76	9,214,635	1,278.74	116,003.46	16.10
	aged writes by dbwr	6.28	5,929,270	822.82	86,725.01	12.04

DB Time and Wait Events

Storage related performance issues often manifest as increased DB time on IO related wait events. The database has various wait events that indicate the type of IO being performed. When there are storage related issues, the average wait time of these wait events, along with an increase in the percentage of DB time spent in these wait events are apparent in the AWR report.

In conjunction with the AWR Exadata sections discussed in the prior section, the database wait events can be correlated with specific statistics in the Exadata sections to determine how the IO is being processed on the storage servers.

In many cases, slow IO latencies are the result of an increased number of IOs serviced from hard disk, instead of utilizing flash cache or XRMEM Cache. The key is then to identify why the IO is not getting serviced from the Exadata caches.

Exadata Performance Summary and Scope

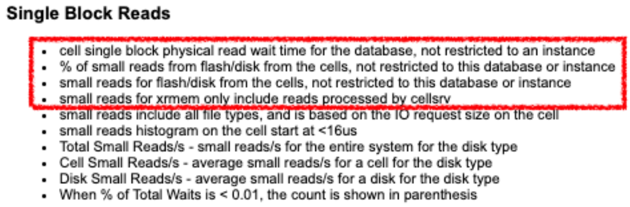
When reviewing the Exadata sections in the AWR report, be mindful of the descriptions that indicate the scope of the statistics being displayed. The Performance Summary includes both database statistics and storage statistics. This allows for easier correlation between the two. However, the database statistics shows data for the single database from which the AWR report was generated; while the storage statistics show data gathered on the storage servers which includes all databases running on those servers.

Single Block Reads

As shown in Figure 8, the wait events related to single block reads are a good indication of storage IO performance. A single block read on the database translates to a small read on the storage server. The wait event also indicates the media where the read occurred.

Single block reads are often the dominant IO wait events in an OLTP system. Long latencies for cell single block physical reads may be representative of potential storage related issues.

Figure 8. Scope of statistics



Single Block Reads

- cell single block physical read wait time for the database, not restricted to an instance
- % of small reads from flash/disk from the cells, not restricted to this database or instance
- small reads for flash/disk from the cells, not restricted to this database or instance
- small reads for xrmem only include reads processed by cellsrv
- small reads include all file types, and is based on the IO request size on the cell
- small reads histogram on the cell start at <16us
- Total Small Reads/s - small reads/s for the entire system for the disk type
- Cell Small Reads/s - average small reads/s for a cell for the disk type
- Disk Small Reads/s - average small reads/s for a disk for the disk type
- When % of Total Waits is < 0.01, the count is shown in parenthesis

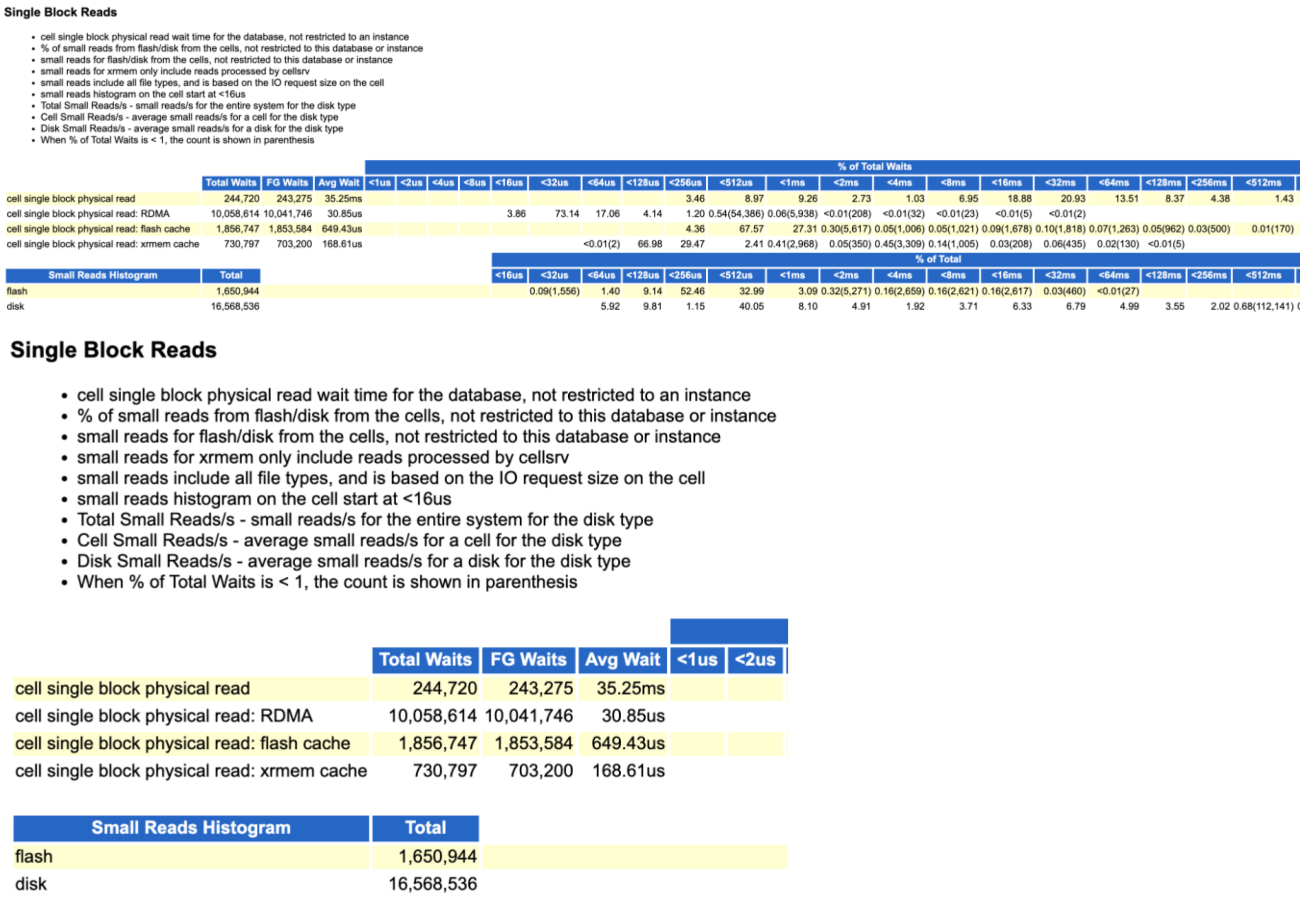
Table 1 lists the different cell single block read wait events. The average latency of each single block read will be vastly different, depending on the media where the read is coming from.

Figure 9 shows a system where the number of waits for single block read are primarily from RDMA. However, because the latency from RDMA is much lower, the larger number of waits will typically consume less overall DB time.

Table 1. Cell single block read wait events

Wait Event	Description
cell single block physical read: RDMA	Wait event for single block reads using RDMA to read from XRMEM Cache
cell single block physical read: xrmem cache	Wait event for single block reads from XRMEM Cache
cell single block physical read: flash cache	Wait event for single block reads from flash cache
cell single block physical read	Wait event for single block reads from disk or capacity optimized flash

Figure 9. Cell single block read wait events



Single Block Reads

- cell single block physical read wait time for the database, not restricted to an instance
- % of small reads from flash/disk from the cells, not restricted to this database or instance
- small reads for flash/disk from the cells, not restricted to this database or instance
- small reads for xrmem only include reads processed by cellsrv
- small reads include all file types, and is based on the IO request size on the cell
- small reads histogram on the cell start at <16us
- Total Small Reads/s - small reads/s for the entire system for the disk type
- Cell Small Reads/s - average small reads/s for a cell for the disk type
- Disk Small Reads/s - average small reads/s for a disk for the disk type
- When % of Total Waits is < 1, the count is shown in parenthesis

				% of Total Waits																			
	Total Waits	FG Waits	Avg Wait	<1us	<2us	<4us	<8us	<16us	<32us	<64us	<128us	<256us	<512us	<1ms	<2ms	<4ms	<8ms	<16ms	<32ms	<64ms	<128ms	<256ms	<512ms
cell single block physical read	244,720	243,275	35.25ms									3.46	8.97	9.26	2.73	1.03	6.95	18.88	20.93	13.51	8.37	4.38	1.43
cell single block physical read: RDMA	10,058,614	10,041,746	30.85us					3.86	73.14	17.06	4.14	1.20	0.54(54,386)	0.06(5,938)	<0.01(208)	<0.01(32)	<0.01(23)	<0.01(5)	<0.01(2)				
cell single block physical read: flash cache	1,856,747	1,853,584	649.43us									4.36	67.57	27.31	0.30(5,617)	0.05(1,006)	0.05(1,021)	0.09(1,678)	0.10(1,818)	0.07(1,263)	0.05(962)	0.03(500)	0.01(170)
cell single block physical read: xrmem cache	730,797	703,200	168.61us							<0.01(2)	66.98	29.47	2.41	0.41(2,968)	0.05(350)	0.45(3,309)	0.14(1,005)	0.03(208)	0.06(435)	0.02(130)	<0.01(5)		

		% of Total														
Small Reads Histogram	Total	<16us	<32us	<64us	<128us	<256us	<512us	<1ms	<2ms	<4ms	<8ms	<16ms	<32ms	<64ms	<128ms	<256ms	<512ms
flash	1,650,944		0.09(1,556)	1.40	9.14	52.46	32.99	3.09	0.32(5,271)	0.16(2,659)	0.16(2,621)	0.16(2,617)	0.03(460)	<0.01(27)			
disk	16,568,536			5.92	9.81	1.15	40.05	8.10	4.91	1.92	3.71	6.33	6.79	4.99	3.55	2.02	0.68(112,141)

Single Block Reads

- cell single block physical read wait time for the database, not restricted to an instance
- % of small reads from flash/disk from the cells, not restricted to this database or instance
- small reads for flash/disk from the cells, not restricted to this database or instance
- small reads for xrmem only include reads processed by cellsrv
- small reads include all file types, and is based on the IO request size on the cell
- small reads histogram on the cell start at <16us
- Total Small Reads/s - small reads/s for the entire system for the disk type
- Cell Small Reads/s - average small reads/s for a cell for the disk type
- Disk Small Reads/s - average small reads/s for a disk for the disk type
- When % of Total Waits is < 1, the count is shown in parenthesis

	Total Waits	FG Waits	Avg Wait	<1us	<2us
cell single block physical read	244,720	243,275	35.25ms		
cell single block physical read: RDMA	10,058,614	10,041,746	30.85us		
cell single block physical read: flash cache	1,856,747	1,853,584	649.43us		
cell single block physical read: xrmem cache	730,797	703,200	168.61us		

Small Reads Histogram	Total
flash	1,650,944
disk	16,568,536

Table 2 illustrates the cost of the disk reads. The total number of cell single block physical read related wait events is 12,890,878 and the total time spent on the wait events is 9,544.72 seconds. Even with only 2 percent of the cell single block physical read wait counts occurring against disk, it accounts for over 82 percent of the wait time. For this reason, if a large number of reads have to be serviced from hard disk, it often will result in visible performance issues.

Table 2. Cost of disk reads

Wait Event	Total DB time (waits * avg wait)	% of Wait Count	% of Wait Time
cell single block physical read	7,892.22s (244,720 * 32.25ms)	1.9	82.7
cell single block physical read: RDMA	310.30s (10,058,614 * 30.85us)	78.0	3.2
cell single block physical read: flash cache	1,205.83s (1,856,747 * 649.43us)	14.4	12.6
cell single block physical read: xrmem cache	136.37s (730,797 * 168.61us)	5.7	1.4

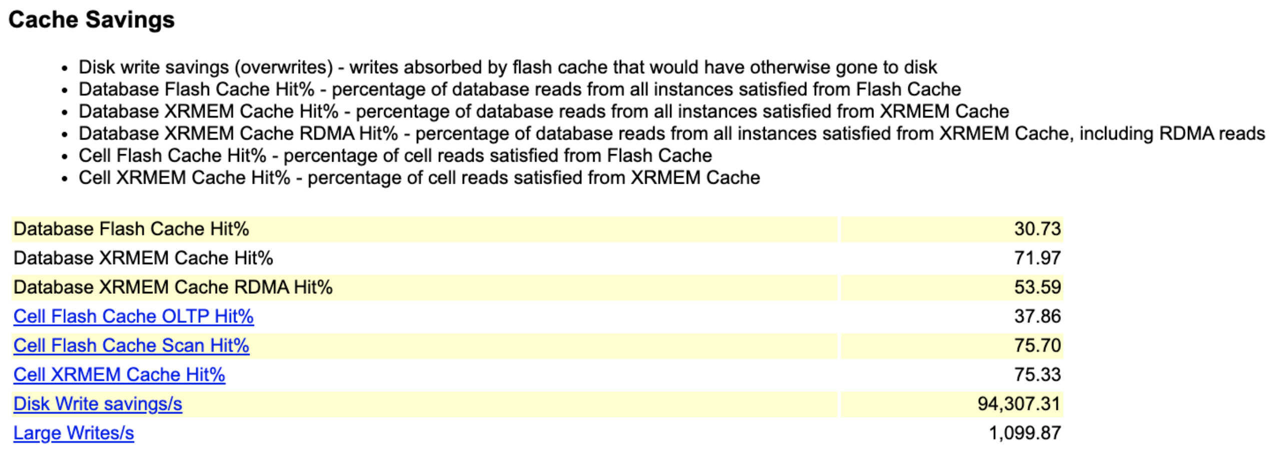
The **Exadata Performance Summary** includes a section to indicate how small reads are processed. Figure 10 indicates 30.73 percent of the Database IOs are served from flash cache, another 71.97 percent is from XRMEM cache (of which 53.59 percent is done via RDMA reads). This is a well-performing database where most of the reads are satisfied from the Exadata caches[8].

When the database performs RDMA reads with XRMEM cache, the read request does not go to the cell. Instead, the database reads directly from the XRMEM cache on the storage server, thereby achieving extremely fast read latencies. In this case, the storage servers do not account for the RDMA reads performed by the databases, and the Exadata sections do not account for these RDMA reads. Instead, the RDMA reads are taken from database statistics.

The Exadata XRMEM cache sections reflect database read requests that did not come through RDMA. In this case, the read request is sent to the storage server, the storage server processes the read request, which can then result in either a hit or a miss on the XRMEM cache.

When most of the read requests are serviced from the XRMEM cache, it is possible to get lower flash cache hit percentage as seen in Figure 10. This may not necessarily be of concern, as it could simply be due to the lower number of read requests against flash cache. This pattern typically indicates new data being accessed that needs to be read into cache.

Figure 10. Performance Summary – Cache Savings



Cache Savings

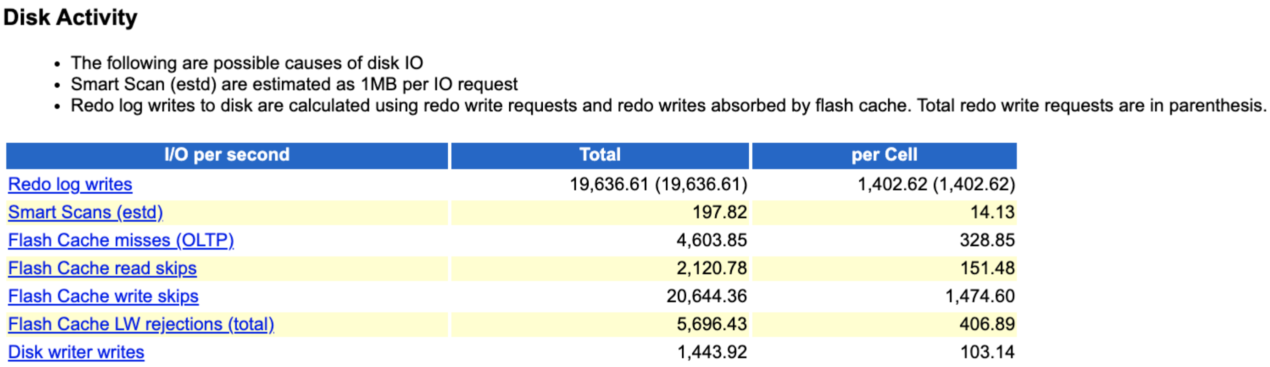
- Disk write savings (overwrites) - writes absorbed by flash cache that would have otherwise gone to disk
- Database Flash Cache Hit% - percentage of database reads from all instances satisfied from Flash Cache
- Database XRMEM Cache Hit% - percentage of database reads from all instances satisfied from XRMEM Cache
- Database XRMEM Cache RDMA Hit% - percentage of database reads from all instances satisfied from XRMEM Cache, including RDMA reads
- Cell Flash Cache Hit% - percentage of cell reads satisfied from Flash Cache
- Cell XRMEM Cache Hit% - percentage of cell reads satisfied from XRMEM Cache

Database Flash Cache Hit%	30.73
Database XRMEM Cache Hit%	71.97
Database XRMEM Cache RDMA Hit%	53.59
Cell Flash Cache OLTP Hit%	37.86
Cell Flash Cache Scan Hit%	75.70
Cell XRMEM Cache Hit%	75.33
Disk Write savings/s	94,307.31
Large Writes/s	1,099.87

[8] The total cache hit percentages can sometimes exceed 100 percent, as some types of reads are counted as cache hits (numerator), but not as physical read IO requests (denominator). Controlfile reads is an example of this.

As performance issues may stem from excessive disk IOs that do not benefit from the Exadata caches, the **Exadata Performance Summary – Disk Activity** section, as seen in Figure 11, also shows potential causes of disk IO.

Figure 11. Performance Summary – Disk Activity



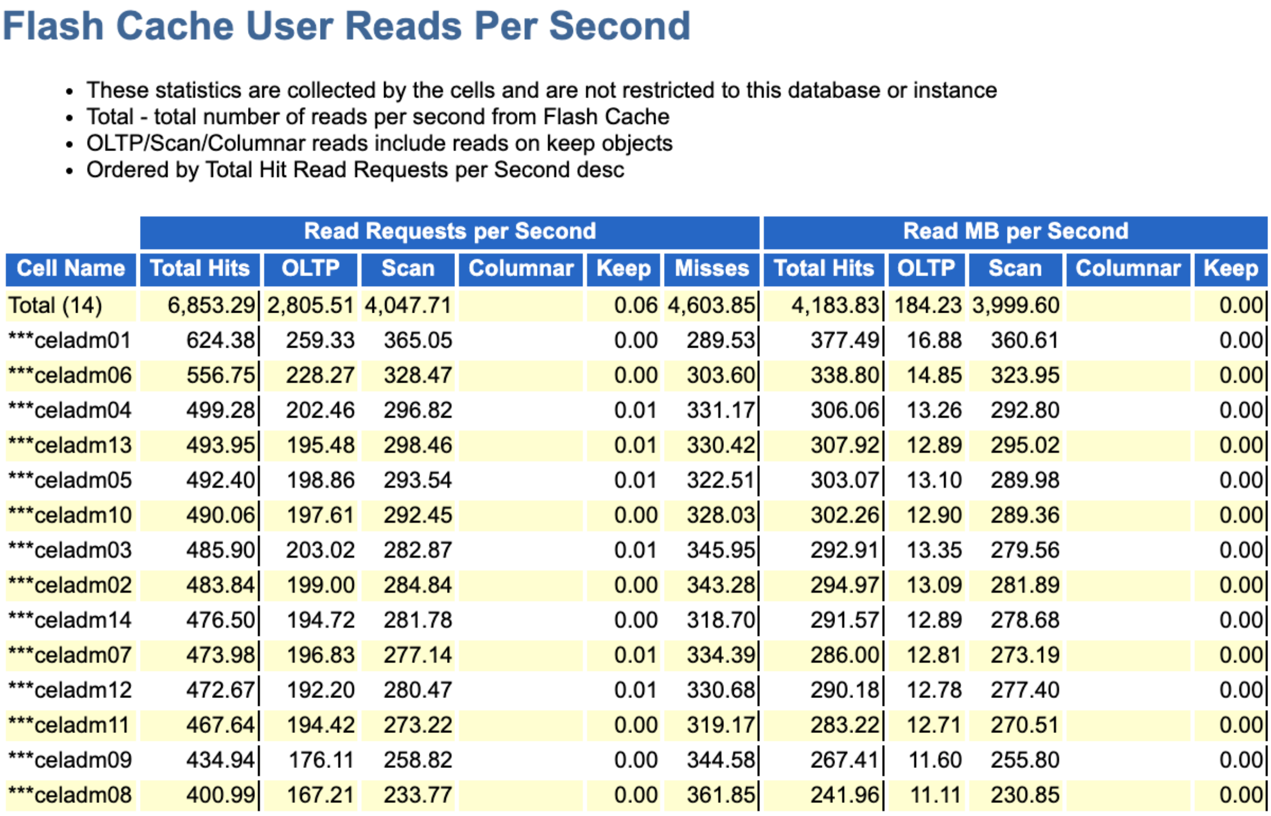
Disk Activity

- The following are possible causes of disk IO
- Smart Scan (estd) are estimated as 1MB per IO request
- Redo log writes to disk are calculated using redo write requests and redo writes absorbed by flash cache. Total redo write requests are in parenthesis.

I/O per second	Total	per Cell
Redo log writes	19,636.61 (19,636.61)	1,402.62 (1,402.62)
Smart Scans (estd)	197.82	14.13
Flash Cache misses (OLTP)	4,603.85	328.85
Flash Cache read skips	2,120.78	151.48
Flash Cache write skips	20,644.36	1,474.60
Flash Cache LW rejections (total)	5,696.43	406.89
Disk writer writes	1,443.92	103.14

As shown in Figure 12, there are only ~200 OLTP read requests per second against flash cache per cell. This includes all databases running on the Exadata system. Considering that there is a fairly low read request rate from flash cache, coupled with the high hit rates from XRMEM cache, there is an indication that the low flash cache hit rates may have minimal impact on this particular database's performance. However, if we see low cache hit rates for the database, coupled with high miss rates from flash cache, it can indicate that the reads are getting serviced from disk and that would warrant further investigation.

Figure 12. Flash Cache User Reads Per Second



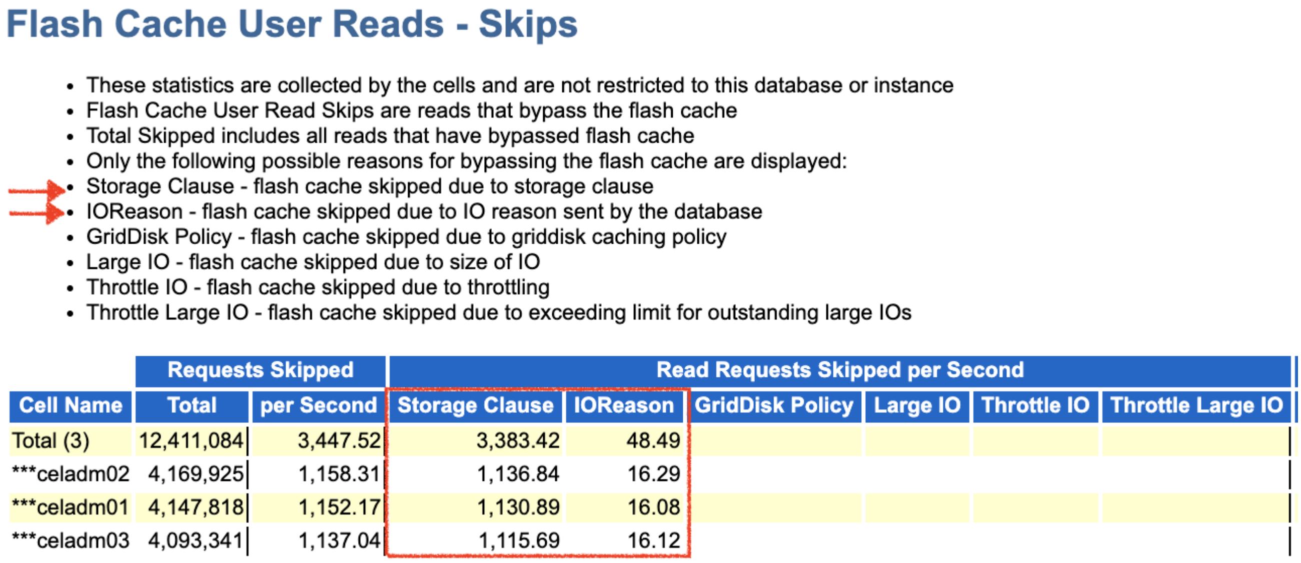
Flash Cache User Reads Per Second

- These statistics are collected by the cells and are not restricted to this database or instance
- Total - total number of reads per second from Flash Cache
- OLTP/Scan/Columnar reads include reads on keep objects
- Ordered by Total Hit Read Requests per Second desc

	Read Requests per Second						Read MB per Second				
Cell Name	Total Hits	OLTP	Scan	Columnar	Keep	Misses	Total Hits	OLTP	Scan	Columnar	Keep
Total (14)	6,853.29	2,805.51	4,047.71		0.06	4,603.85	4,183.83	184.23	3,999.60		0.00
***celadm01	624.38	259.33	365.05		0.00	289.53	377.49	16.88	360.61		0.00
***celadm06	556.75	228.27	328.47		0.00	303.60	338.80	14.85	323.95		0.00
***celadm04	499.28	202.46	296.82		0.01	331.17	306.06	13.26	292.80		0.00
***celadm13	493.95	195.48	298.46		0.01	330.42	307.92	12.89	295.02		0.00
***celadm05	492.40	198.86	293.54		0.01	322.51	303.07	13.10	289.98		0.00
***celadm10	490.06	197.61	292.45		0.00	328.03	302.26	12.90	289.36		0.00
***celadm03	485.90	203.02	282.87		0.01	345.95	292.91	13.35	279.56		0.00
***celadm02	483.84	199.00	284.84		0.00	343.28	294.97	13.09	281.89		0.00
***celadm14	476.50	194.72	281.78		0.00	318.70	291.57	12.89	278.68		0.00
***celadm07	473.98	196.83	277.14		0.01	334.39	286.00	12.81	273.19		0.00
***celadm12	472.67	192.20	280.47		0.01	330.68	290.18	12.78	277.40		0.00
***celadm11	467.64	194.42	273.22		0.00	319.17	283.22	12.71	270.51		0.00
***celadm09	434.94	176.11	258.82		0.00	344.58	267.41	11.60	255.80		0.00
***celadm08	400.99	167.21	233.77		0.00	361.85	241.96	11.11	230.85		0.00

Aside from flash cache misses, the single block reads from disk may also be caused by flash cache skips. A flash cache skip occurs when the reads bypass the Flash Cache, as they are marked as not eligible for caching in Flash Cache. Figure 13 shows the reads are bypassing flash cache due to the storage clause, which indicates that segments have specified a storage clause to include `cell_flash_cache NONE`.

Figure 13. Flash Cache User Reads - Skips



Flash Cache User Reads - Skips

- These statistics are collected by the cells and are not restricted to this database or instance
- Flash Cache User Read Skips are reads that bypass the flash cache
- Total Skipped includes all reads that have bypassed flash cache
- Only the following possible reasons for bypassing the flash cache are displayed:
- Storage Clause - flash cache skipped due to storage clause
- IOReason - flash cache skipped due to IO reason sent by the database
- GridDisk Policy - flash cache skipped due to griddisk caching policy
- Large IO - flash cache skipped due to size of IO
- Throttle IO - flash cache skipped due to throttling
- Throttle Large IO - flash cache skipped due to exceeding limit for outstanding large IOs

	Requests Skipped		Read Requests Skipped per Second					
Cell Name	Total	per Second	Storage Clause	IOReason	GridDisk Policy	Large IO	Throttle IO	Throttle Large IO
Total (3)	12,411,084	3,447.52	3,383.42	48.49				
***celadm02	4,169,925	1,158.31	1,136.84	16.29				
***celadm01	4,147,818	1,152.17	1,130.89	16.08				
***celadm03	4,093,341	1,137.04	1,115.69	16.12				

In addition to reviewing the Flash Cache sections, one should also check for:

- Imbalance across cells/disks
- Top Databases
- Small Read Histogram – if the histograms on the cells do not show issues, but the histograms on the database indicate long latencies, this indicates it is not the IO causing the larger latencies, but potentially something on the network or IORM queueing.

Smart Scans

The wait events for smart scans can vary from system to system, and even from query to query. The wait events include all the processing offloaded on the storage, in addition to the IO time. The processing cost is dependent on the type of operations being offloaded, as some operations are more CPU intensive than others.

In most cases, users will see a *cell smart table scan* wait event. If passthru is occurring, the wait event will indicate the passthru reason.

Table 3. Smart Scan Wait Events

Wait Event	Description
cell smart table scan	Wait even when the session is waiting for smart scans to complete
cell smart table scan: db timezone upgrade	Wait event when the cells are unable to offload because a database timezone upgrade is in progress
cell smart table scan: disabled by user	Wait event when the cells are unable to offload due to a user setting
cell smart table scan: passthru	Wait event when the cells are unable to offload the smart scan

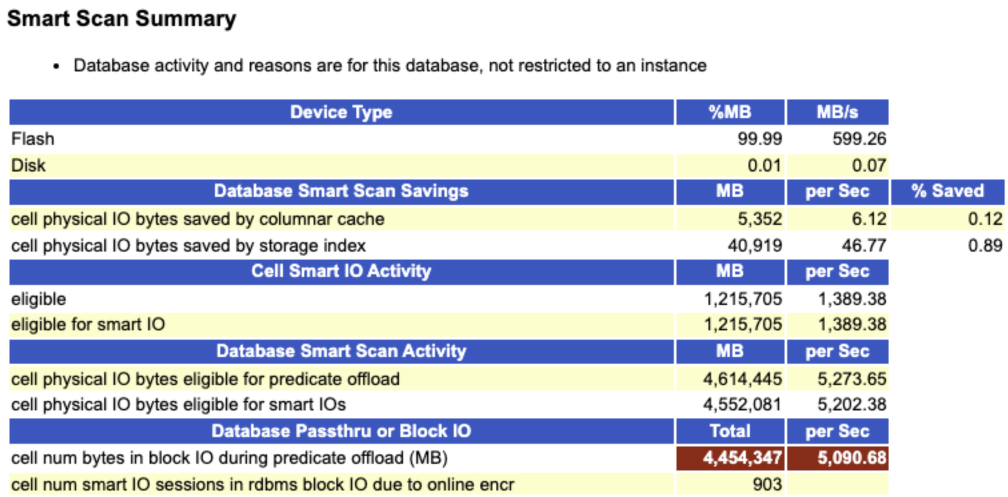
Increased cell smart table scan latencies can occur for a variety of reasons. Some of the common causes are passthru, increased disk IO, lack of Storage Index savings, or lack of Columnar Cache. In some cases, a database may fall back to executing in block IO mode. [9]

For smart scans, it is often useful to look at specific queries that are affected by the poor performance. SQL Monitor is another tool that is extremely useful for diagnosing smart scan issues. There are also database statistics that can be used to understand the smart scan performance and correlate it with the storage server statistics.

The Performance Summary section includes a **Smart Scan Summary** where one can correlate database and storage side statistics. In Figure 14, we see the database is issuing about 5.2 GB/s eligible for smart scan (*cell physical IO bytes eligible for smart IOs*), of which most of it is falling back in block IO mode (*cell num bytes in block IO during predicate offload*), due to an ongoing online encryption.

There are a number of database side statistics that indicate why smart scans may not be offloaded. The statistics are described in the *Exadata Storage Software User's Guide – Monitoring Exadata*.

Figure 14. Performance Summary: Smart Scan Summary



Smart Scan Summary

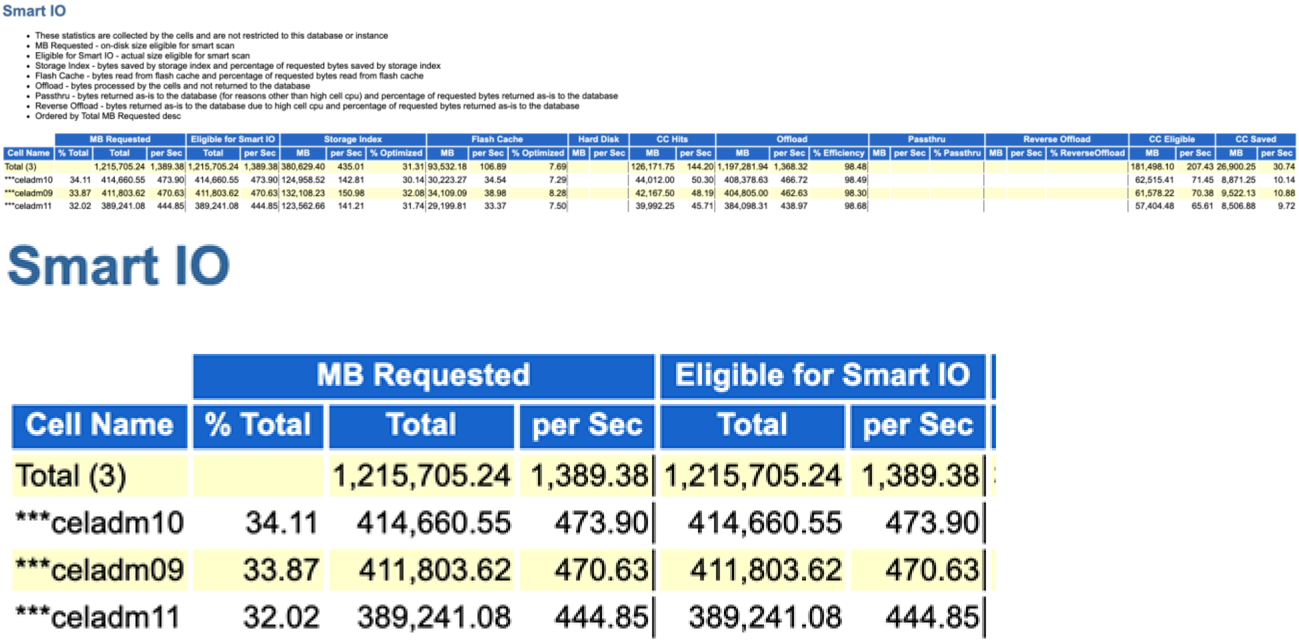
- Database activity and reasons are for this database, not restricted to an instance

Device Type	%MB	MB/s	
Flash	99.99	599.26	
Disk	0.01	0.07	
Database Smart Scan Savings	**MB**	**per Sec**	**% Saved**
cell physical IO bytes saved by columnar cache	5,352	6.12	0.12
cell physical IO bytes saved by storage index	40,919	46.77	0.89
Cell Smart IO Activity	**MB**	**per Sec**	
eligible	1,215,705	1,389.38	
eligible for smart IO	1,215,705	1,389.38	
Database Smart Scan Activity	**MB**	**per Sec**	
cell physical IO bytes eligible for predicate offload	4,614,445	5,273.65	
cell physical IO bytes eligible for smart IOs	4,552,081	5,202.38	
Database Passthru or Block IO	**Total**	**per Sec**	
cell num bytes in block IO during predicate offload (MB)	4,454,347	5,090.68	
cell num smart IO sessions in rdbms block IO due to online encr	903		

[9] There are database statistics that indicate when passthru or block IO mode occur. These are documented in the Exadata Storage Software User's Guide – Monitoring Exadata.

The Exadata **Smart IO** section, shown in Figure 15, indicates the amount of IOs the storage servers are processing that are eligible for smart scans. It also indicates storage index savings bytes read from flash, bytes read from disk, along with columnar cache usage. It also indicates if passthru or reverse offload are occurring.

Figure 15. Exadata Smart IO



Smart IO

- These statistics are collected by the cells and are not restricted to this database or instance
- MB Requested - on-disk size eligible for smart scan
- Eligible for Smart IO - actual size eligible for smart scan
- Storage Index - bytes saved by storage index and percentage of requested bytes saved by storage index
- Flash Cache - bytes read from flash cache and percentage of requested bytes read from flash cache
- Offload - bytes processed by the cells and not returned to the database
- Passthru - bytes returned as-is to the database (for reasons other than high cell cpu) and percentage of requested bytes returned as-is to the database
- Reverse Offload - bytes returned as-is to the database due to high cell cpu and percentage of requested bytes returned as-is to the database
- Ordered by Total MB Requested desc

	MB Requested			Eligible for Smart IO		Storage Index			Flash Cache			Hard Disk	
Cell Name	% Total	Total	per Sec	Total	per Sec	MB	per Sec	% Optimized	MB	per Sec	% Optimized	MB	per Sec
Total (3)		1,215,705.24	1,389.38	1,215,705.24	1,389.38	380,629.40	435.01	31.31	93,532.18	106.89	7.69		
***celadm10	34.11	414,660.55	473.90	414,660.55	473.90	124,958.52	142.81	30.14	30,223.27	34.54	7.29		
***celadm09	33.87	411,803.62	470.63	411,803.62	470.63	132,108.23	150.98	32.08	34,109.09	38.98	8.28		
***celadm11	32.02	389,241.08	444.85	389,241.08	444.85	123,562.66	141.21	31.74	29,199.81	33.37	7.50		

	CC Hits		Offload			Passthru			Reverse Offload			CC Eligible		CC Saved	
Cell Name	MB	per Sec	MB	per Sec	% Efficiency	MB	per Sec	% Passthru	MB	per Sec	% ReverseOffload	MB	per Sec	MB	per Sec
Total (3)	126,171.75	144.20	1,197,281.94	1,368.32	98.48							181,498.10	207.43	26,900.25	30.74
***celadm10	44,012.00	50.30	408,378.63	466.72	98.49							62,515.41	71.45	8,871.25	10.14
***celadm09	42,167.50	48.19	404,805.00	462.63	98.30							61,578.22	70.38	9,522.13	10.88
***celadm11	39,992.25	45.71	384,098.31	438.97	98.68							57,404.48	65.61	8,506.88	9.72

Smart IO

In addition to the Exadata **Smart IO** section, the **Flash Cache User Reads** section also shows the amount of IOs that are being done for scans and for columnar cache, as shown earlier in Figure 12.

In addition to reviewing the Smart IO and the various Flash Cache and Columnar Cache sections, one should also check:

- **Imbalance across cells/disks**: smart scans are expected to hit all cells/disks evenly. If a single cell/disk is performing slowly, it will impact the scan latencies. Scans will often be large reads on the storage servers.
- **Top Databases**: IORM queue times for large IOs can also impact smart scan latencies.

Temp Spills

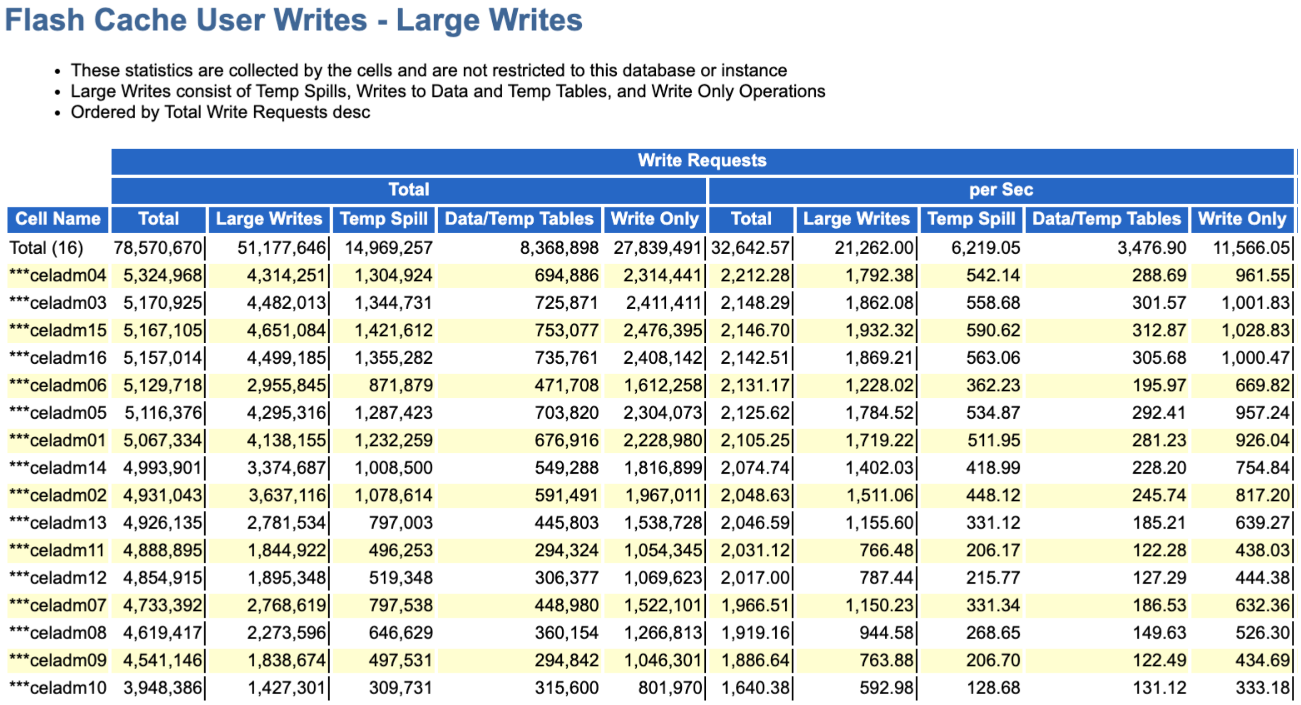
When the database performs temp IO, the temp IO is expected to be absorbed into flash cache. Other large writes are eligible to be absorbed in flash cache but may be rejected for a variety of reasons. When the latencies for the database temp IO related wait events increase, it is often associated with temp not getting absorbed into flash cache.

Table 4. Temp related wait events

Wait Event	Description
direct path write temp	Wait event when the session is writing temp
direct path read temp	Wait event when the session is reading temp

The **Flash Cache User Writes - Large Writes** section, as shown in Figure 16, shows the amount of Large Writes and the type of Large Writes that the storage servers are processing.

Figure 16. Flash Cache User Writes - Large Writes



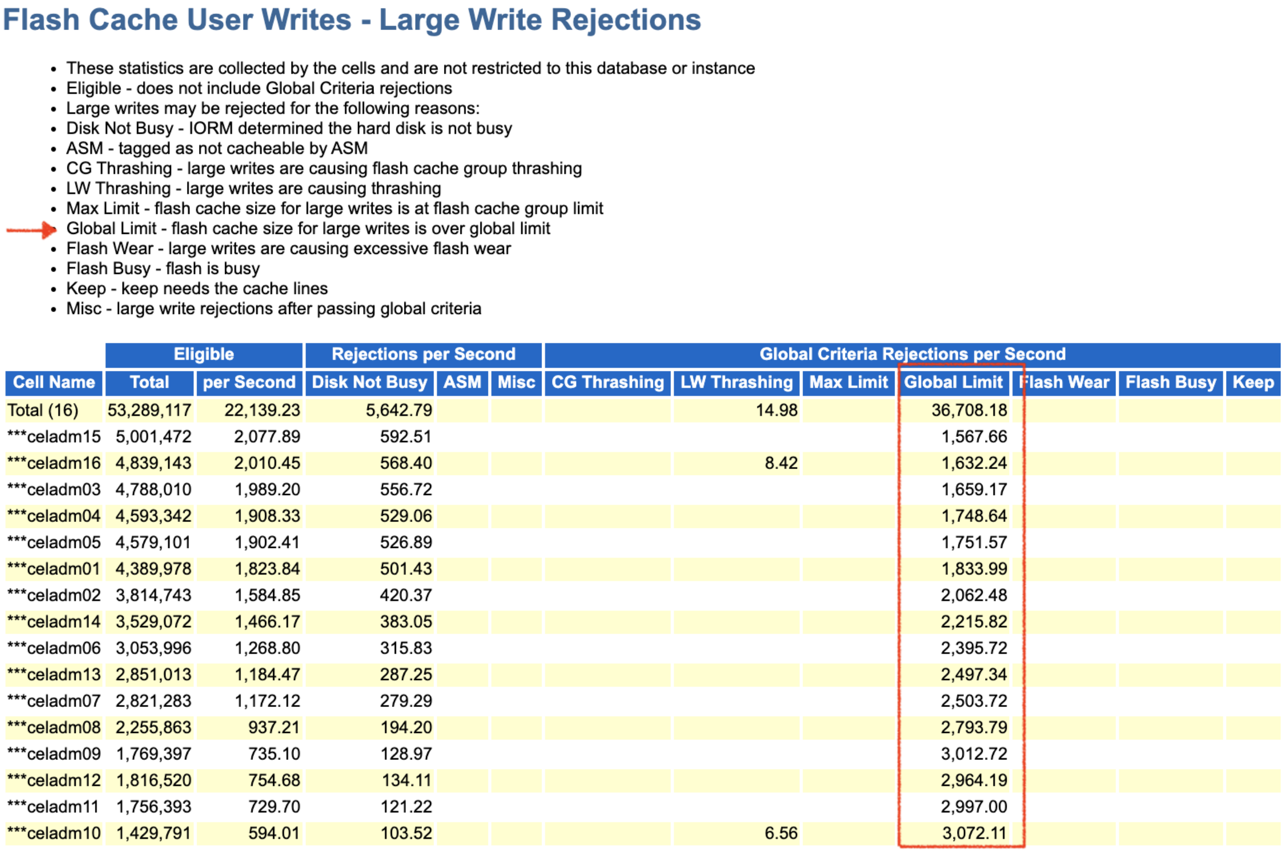
Flash Cache User Writes - Large Writes

- These statistics are collected by the cells and are not restricted to this database or instance
- Large Writes consist of Temp Spills, Writes to Data and Temp Tables, and Write Only Operations
- Ordered by Total Write Requests desc

	Write Requests									
	Total					per Sec				
Cell Name	Total	Large Writes	Temp Spill	Data/Temp Tables	Write Only	Total	Large Writes	Temp Spill	Data/Temp Tables	Write Only
Total (16)	78,570,670	51,177,646	14,969,257	8,368,898	27,839,491	32,642.57	21,262.00	6,219.05	3,476.90	11,566.05
***celadm04	5,324,968	4,314,251	1,304,924	694,886	2,314,441	2,212.28	1,792.38	542.14	288.69	961.55
***celadm03	5,170,925	4,482,013	1,344,731	725,871	2,411,411	2,148.29	1,862.08	558.68	301.57	1,001.83
***celadm15	5,167,105	4,651,084	1,421,612	753,077	2,476,395	2,146.70	1,932.32	590.62	312.87	1,028.83
***celadm16	5,157,014	4,499,185	1,355,282	735,761	2,408,142	2,142.51	1,869.21	563.06	305.68	1,000.47
***celadm06	5,129,718	2,955,845	871,879	471,708	1,612,258	2,131.17	1,228.02	362.23	195.97	669.82
***celadm05	5,116,376	4,295,316	1,287,423	703,820	2,304,073	2,125.62	1,784.52	534.87	292.41	957.24
***celadm01	5,067,334	4,138,155	1,232,259	676,916	2,228,980	2,105.25	1,719.22	511.95	281.23	926.04
***celadm14	4,993,901	3,374,687	1,008,500	549,288	1,816,899	2,074.74	1,402.03	418.99	228.20	754.84
***celadm02	4,931,043	3,637,116	1,078,614	591,491	1,967,011	2,048.63	1,511.06	448.12	245.74	817.20
***celadm13	4,926,135	2,781,534	797,003	445,803	1,538,728	2,046.59	1,155.60	331.12	185.21	639.27
***celadm11	4,888,895	1,844,922	496,253	294,324	1,054,345	2,031.12	766.48	206.17	122.28	438.03
***celadm12	4,854,915	1,895,348	519,348	306,377	1,069,623	2,017.00	787.44	215.77	127.29	444.38
***celadm07	4,733,392	2,768,619	797,538	448,980	1,522,101	1,966.51	1,150.23	331.34	186.53	632.36
***celadm08	4,619,417	2,273,596	646,629	360,154	1,266,813	1,919.16	944.58	268.65	149.63	526.30
***celadm09	4,541,146	1,838,674	497,531	294,842	1,046,301	1,886.64	763.88	206.70	122.49	434.69
***celadm10	3,948,386	1,427,301	309,731	315,600	801,970	1,640.38	592.98	128.68	131.12	333.18

The **Flash Cache User Writes – Large Writes Rejections** indicates the reasons why we may not be absorbing the Large Writes (or temp spills) into flash cache. In Figure 17, the majority of the large writes are getting rejected due to Global Limit. This means that the Large Writes have exceeded the maximum amount of Flash Cache space available for Large Writes.

Figure 17. Flash Cache User Writes - Large Write Rejections



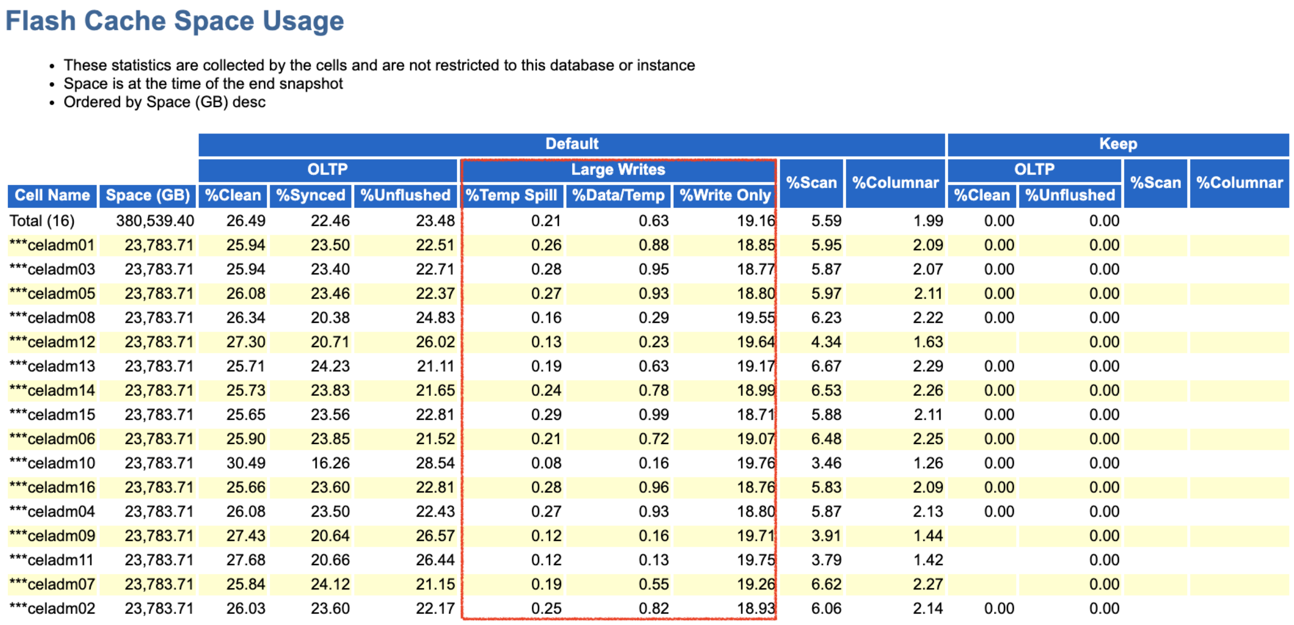
Flash Cache User Writes - Large Write Rejections

- These statistics are collected by the cells and are not restricted to this database or instance
- Eligible - does not include Global Criteria rejections
- Large writes may be rejected for the following reasons:
- Disk Not Busy - IORM determined the hard disk is not busy
- ASM - tagged as not cacheable by ASM
- CG Thrashing - large writes are causing flash cache group thrashing
- LW Thrashing - large writes are causing thrashing
- Max Limit - flash cache size for large writes is at flash cache group limit
- Global Limit - flash cache size for large writes is over global limit
- Flash Wear - large writes are causing excessive flash wear
- Flash Busy - flash is busy
- Keep - keep needs the cache lines
- Misc - large write rejections after passing global criteria

	Eligible		Rejections per Second			Global Criteria Rejections per Second						
Cell Name	Total	per Second	Disk Not Busy	ASM	Misc	CG Thrashing	LW Thrashing	Max Limit	Global Limit	Flash Wear	Flash Busy	Keep
Total (16)	53,289,117	22,139.23	5,642.79				14.98		36,708.18			
***celadm15	5,001,472	2,077.89	592.51						1,567.66			
***celadm16	4,839,143	2,010.45	568.40				8.42		1,632.24			
***celadm03	4,788,010	1,989.20	556.72						1,659.17			
***celadm04	4,593,342	1,908.33	529.06						1,748.64			
***celadm05	4,579,101	1,902.41	526.89						1,751.57			
***celadm01	4,389,978	1,823.84	501.43						1,833.99			
***celadm02	3,814,743	1,584.85	420.37						2,062.48			
***celadm14	3,529,072	1,466.17	383.05						2,215.82			
***celadm06	3,053,996	1,268.80	315.83						2,395.72			
***celadm13	2,851,013	1,184.47	287.25						2,497.34			
***celadm07	2,821,283	1,172.12	279.29						2,503.72			
***celadm08	2,255,863	937.21	194.20						2,793.79			
***celadm09	1,769,397	735.10	128.97						3,012.72			
***celadm12	1,816,520	754.68	134.11						2,964.19			
***celadm11	1,756,393	729.70	121.22						2,997.00			
***celadm10	1,429,791	594.01	103.52				6.56		3,072.11			

The **Flash Cache Space Usage** in Figure 18 shows about 20% of the flash cache is used for Large Writes, which is the maximum amount that can be used for Large Writes in flash cache. Oftentimes, this can be addressed by tuning the queries to reduce demand for temp space. In some cases, the increase in temp demand is attributed to either application upgrades, which result in a new set of queries, or database upgrades that result in optimizer execution plan changes. If the workload requires more flash cache space for temp IO, you can also review the use of the main_workload_type database parameter as described in the *System Software User's Guide for Exadata Database Machine – Optimizing Exadata Smart Flash Cache For Analytical Workloads*.

Figure 18. Flash Cache Space Usage



Flash Cache Space Usage

- These statistics are collected by the cells and are not restricted to this database or instance
- Space is at the time of the end snapshot
- Ordered by Space (GB) desc

		Default								Keep			
		OLTP			Large Writes			%Scan	%Columnar	OLTP		%Scan	%Columnar
Cell Name	Space (GB)	%Clean	%Synced	%Unflushed	%Temp Spill	%Data/Temp	%Write Only			%Clean	%Unflushed		
Total (16)	380,539.40	26.49	22.46	23.48	0.21	0.63	19.16	5.59	1.99	0.00	0.00		
***celadm01	23,783.71	25.94	23.50	22.51	0.26	0.88	18.85	5.95	2.09	0.00	0.00		
***celadm03	23,783.71	25.94	23.40	22.71	0.28	0.95	18.77	5.87	2.07	0.00	0.00		
***celadm05	23,783.71	26.08	23.46	22.37	0.27	0.93	18.80	5.97	2.11	0.00	0.00		
***celadm08	23,783.71	26.34	20.38	24.83	0.16	0.29	19.55	6.23	2.22	0.00	0.00		
***celadm12	23,783.71	27.30	20.71	26.02	0.13	0.23	19.64	4.34	1.63		0.00		
***celadm13	23,783.71	25.71	24.23	21.11	0.19	0.63	19.17	6.67	2.29	0.00	0.00		
***celadm14	23,783.71	25.73	23.83	21.65	0.24	0.78	18.99	6.53	2.26	0.00	0.00		
***celadm15	23,783.71	25.65	23.56	22.81	0.29	0.99	18.71	5.88	2.11	0.00	0.00		
***celadm06	23,783.71	25.90	23.85	21.52	0.21	0.72	19.07	6.48	2.25	0.00	0.00		
***celadm10	23,783.71	30.49	16.26	28.54	0.08	0.16	19.76	3.46	1.26	0.00	0.00		
***celadm16	23,783.71	25.66	23.60	22.81	0.28	0.96	18.76	5.83	2.09	0.00	0.00		
***celadm04	23,783.71	26.08	23.50	22.43	0.27	0.93	18.80	5.87	2.13	0.00	0.00		
***celadm09	23,783.71	27.43	20.64	26.57	0.12	0.16	19.71	3.91	1.44		0.00		
***celadm11	23,783.71	27.68	20.66	26.44	0.12	0.13	19.75	3.79	1.42		0.00		
***celadm07	23,783.71	25.84	24.12	21.15	0.19	0.55	19.26	6.62	2.27		0.00		
***celadm02	23,783.71	26.03	23.60	22.17	0.25	0.82	18.93	6.06	2.14	0.00	0.00		

In addition to reviewing the Flash Cache sections, one should also check:

- **IO latencies**: temp spills will often be large writes on the storage servers.
- **Top Databases**: IORM queue times for large IOs can also impact smart scan latencies. Increased IORM queue times on hard disks are expected when the hard disks are busy.

Example Scenario: Analyzing Exadata-specific AWR Data

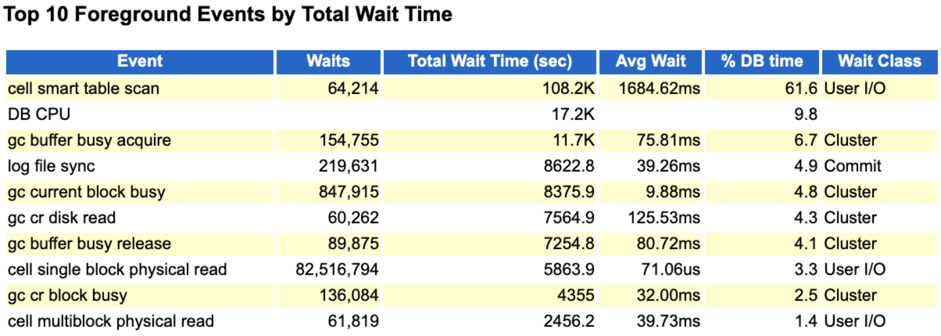
To help familiarize yourself with the Exadata sections, we will walk through an example that reflects a real customer use case. In this example, the customer has migrated to a new Exadata system and is experiencing performance regressions.

Reviewing the Database Statistics

A good place to start is to check if the performance issues are potentially storage related by reviewing the **Top Timed Events** from the AWR report. Figure 19 and Figure 20 show the **Top Timed Events** from a single instance, but in this case all of the instances look similar. The wait events show that almost 62 percent of DB time was spent on *cell smart table scan*, with an average wait time of just over 1.6 seconds.

In Figure 19, the database release in this example does not have the separation that indicates where the read occurred for the wait event that was earlier described in Table 3. However, a latency of 71.06 microseconds for *cell single block physical read* implies most of the single block reads are performed via RDMA.

Figure 19. Top 10 Foreground Events by Total Wait Time

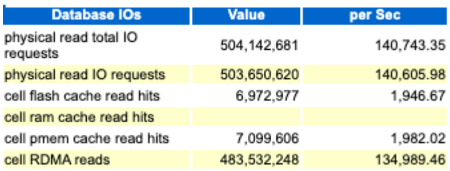


Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Avg Wait	% DB time	Wait Class
cell smart table scan	64,214	108.2K	1684.62ms	61.6	User I/O
DB CPU		17.2K		9.8	
gc buffer busy acquire	154,755	11.7K	75.81ms	6.7	Cluster
log file sync	219,631	8622.8	39.26ms	4.9	Commit
gc current block busy	847,915	8375.9	9.88ms	4.8	Cluster
gc cr disk read	60,262	7564.9	125.53ms	4.3	Cluster
gc buffer busy release	89,875	7254.8	80.72ms	4.1	Cluster
cell single block physical read	82,516,794	5863.9	71.06us	3.3	User I/O
gc cr block busy	136,084	4355	32.00ms	2.5	Cluster
cell multiblock physical read	61,819	2456.2	39.73ms	1.4	User I/O

From the **Exadata Performance Summary** section shown in Figure 20, we can see that a lot of IOs are getting serviced by RDMA reads. However, RDMA reads are normally for OLTP or block IO format and will not benefit smart scans.

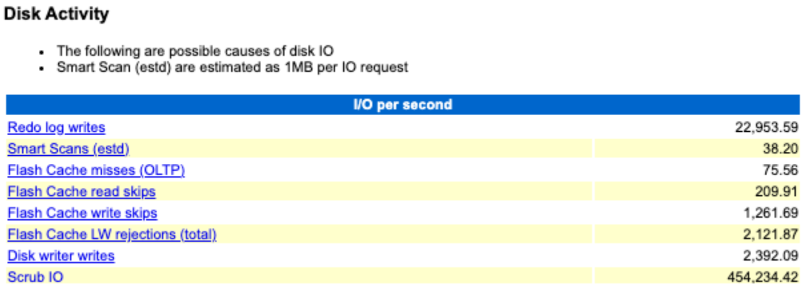
Figure 20. Exadata Performance Summary



Database IOs	Value	per Sec
physical read total IO requests	504,142,681	140,743.35
physical read IO requests	503,650,620	140,605.98
cell flash cache read hits	6,972,977	1,946.67
cell ram cache read hits		
cell pmem cache read hits	7,099,606	1,982.02
cell RDMA reads	483,532,248	134,989.46

If we look further at the sources of disk IO in Figure 21, we can see that there are high numbers for *Flash Cache read skips* and *Flash Cache write skips*. Skips indicate IOs that are not eligible for flash cache. We also see scrub is running (*Scrub IO*), but as mentioned previously, the system is designed to prioritize client IO over scrub IO.

Figure 21. Disk Activity



Disk Activity

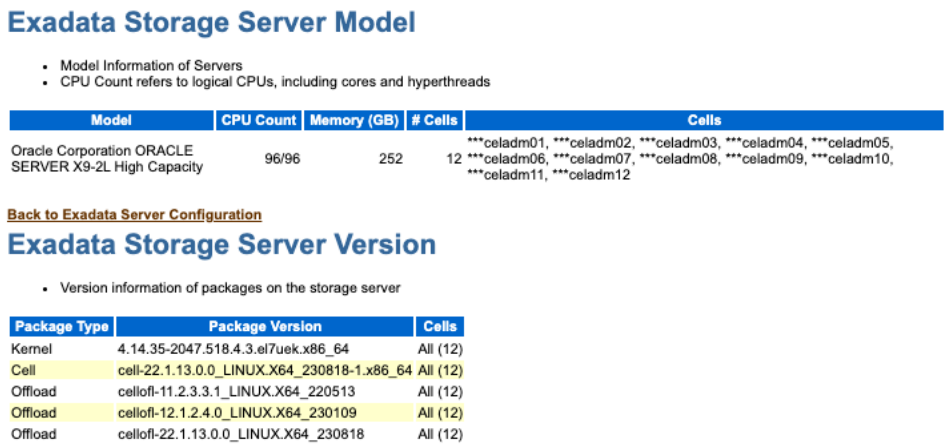
- The following are possible causes of disk IO
- Smart Scan (estd) are estimated as 1MB per IO request

I/O per second	
Redo log writes	22,953.59
Smart Scans (estd)	38.20
Flash Cache misses (OLTP)	75.56
Flash Cache read skips	209.91
Flash Cache write skips	1,261.69
Flash Cache LW rejections (total)	2,121.87
Disk writer writes	2,392.09
Scrub IO	454,234.42

Exadata Configuration

A review of the Exadata Configuration section from Figure 22 shows that this system is an X9M-2 with 12 storage servers.

Figure 22. Exadata Configuration: Exadata Storage Server Model



Exadata Storage Server Model

- Model Information of Servers
- CPU Count refers to logical CPUs, including cores and hyperthreads

Model	CPU Count	Memory (GB)	# Cells	Cells
Oracle Corporation ORACLE SERVER X9-2L High Capacity	96/96	252	12	***celadm01, ***celadm02, ***celadm03, ***celadm04, ***celadm05, ***celadm06, ***celadm07, ***celadm08, ***celadm09, ***celadm10, ***celadm11, ***celadm12

Back to Exadata Server Configuration

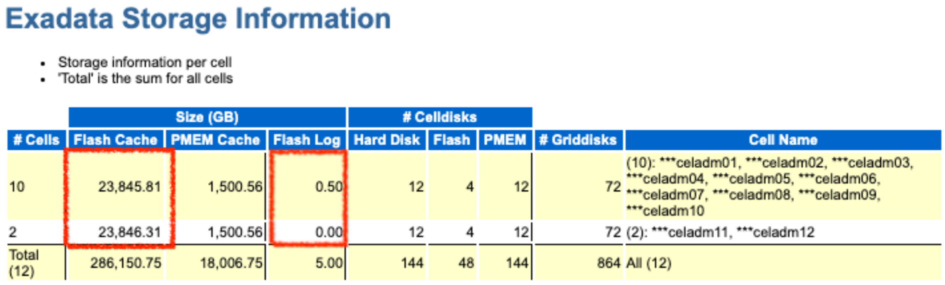
Exadata Storage Server Version

- Version information of packages on the storage server

Package Type	Package Version	Cells
Kernel	4.14.35-2047.518.4.3.el7uek.x86_64	All (12)
Cell	cell-22.1.13.0.0_LINUX.X64_230818-1.x86_64	All (12)
Offload	cellofl-11.2.3.3.1_LINUX.X64_220513	All (12)
Offload	cellofl-12.1.2.4.0_LINUX.X64_230109	All (12)
Offload	cellofl-22.1.13.0.0_LINUX.X64_230818	All (12)

Figure 23 shows that the first indication of a potential issue is that celadm11 and celadm12 have a slightly differently configuration from the other storage cells – no flash log, and a slightly larger flash cache.

Figure 23. Exadata Configuration: Exadata Storage Information



Exadata Storage Information

- Storage information per cell
- 'Total' is the sum for all cells

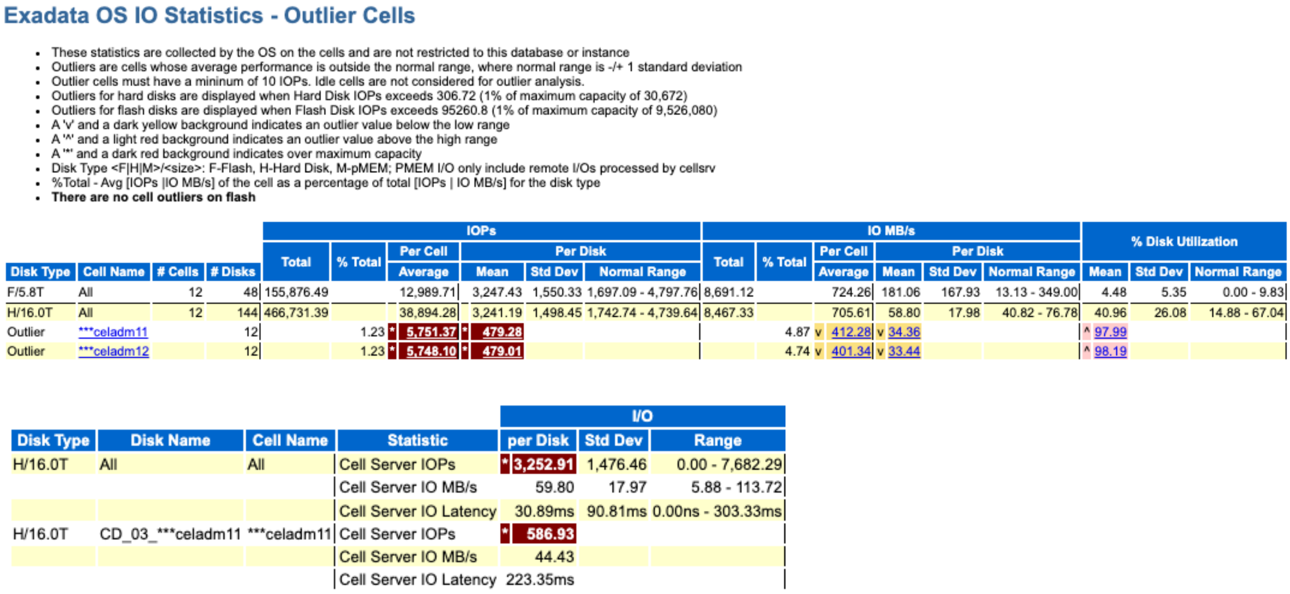
	Size (GB)			# Celldisks				
# Cells	Flash Cache	PMEM Cache	Flash Log	Hard Disk	Flash	PMEM	# Griddisks	Cell Name
10	23,845.81	1,500.56	0.50	12	4	12	72	(10): ***celadm01, ***celadm02, ***celadm03, ***celadm04, ***celadm05, ***celadm06, ***celadm07, ***celadm08, ***celadm09, ***celadm10
2	23,846.31	1,500.56	0.00	12	4	12	72	(2): ***celadm11, ***celadm12
Total (12)	286,150.75	18,006.75	5.00	144	48	144	864	All (12)

IO Distribution

The outlier sections report the IOs per device type, as different types of devices are expected to have different performance characteristics. The format used to identify the device type is *<F|H>/<size>*, where F is for Flash devices, and H for hard disks. In the case of Extreme Flash storage servers, capacity-optimized and performance-optimized flash devices will both be reported as *F/<size>*, with the *<size>* indicating the type of flash device (e.g. X10M EF servers will report capacity-optimized flash devices as *F/14.0T* and performance-optimized flash devices as *F/5.8T*).

A review of the IOs on the storage servers in Figure 24 indicate outliers on celadm11 and celadm12. Although we're seeing a large number of IOs from the other cells (due to scrub), we see the pattern is quite different on celadm11 and celadm12. These two cells are showing ~479 IOPs with almost 100% utilization.

Figure 24. OS Statistics - Outlier Cells



Exadata OS IO Statistics - Outlier Cells

- These statistics are collected by the OS on the cells and are not restricted to this database or instance
- Outliers are cells whose average performance is outside the normal range, where normal range is -/+ 1 standard deviation
- Outlier cells must have a mininum of 10 IOPs. Idle cells are not considered for outlier analysis.
- Outliers for hard disks are displayed when Hard Disk IOPs exceeds 306.72 (1% of maximum capacity of 30,672)
- Outliers for flash disks are displayed when Flash Disk IOPs exceeds 95260.8 (1% of maximum capacity of 9,526,080)
- A 'v' and a dark yellow background indicates an outlier value below the low range
- A '^' and a light red background indicates an outlier value above the high range
- A '*' and a dark red background indicates over maximum capacity
- Disk Type <F|H|M>/<size>: F-Flash, H-Hard Disk, M-pMEM; PMEM I/O only include remote I/Os processed by cellsrv
- %Total - Avg [IOPs |IO MB/s] of the cell as a percentage of total [IOPs | IO MB/s] for the disk type
- **There are no cell outliers on flash**

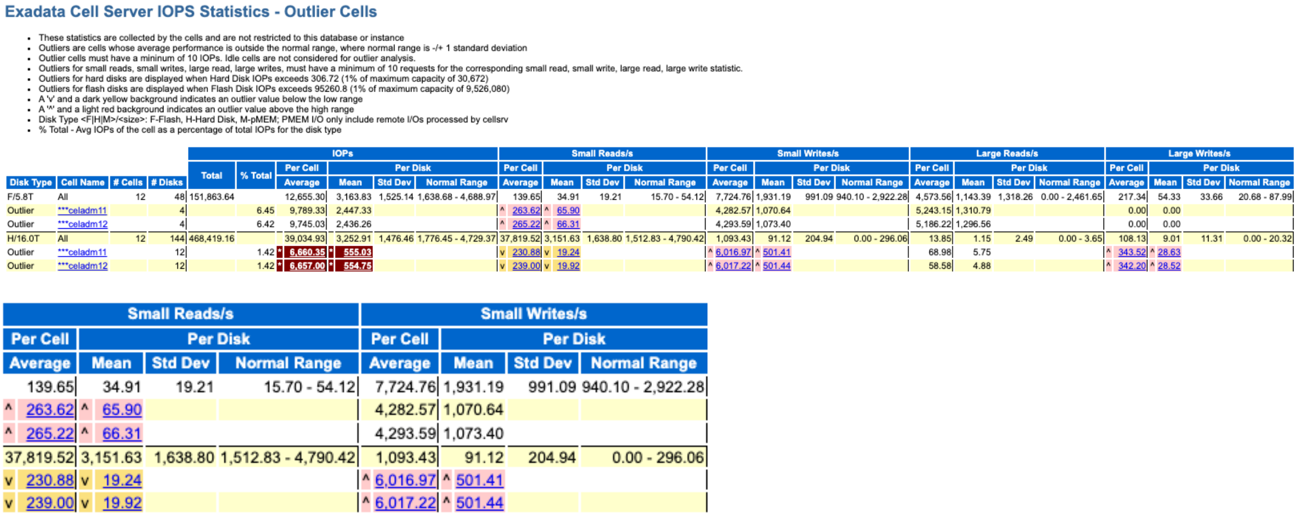
				IOPs						IO MB/s						% Disk Utilization		
				Total	% Total	Per Cell	Per Disk			Total	% Total	Per Cell	Per Disk					
Disk Type	Cell Name	# Cells	# Disks			Average	Mean	Std Dev	Normal Range			Average	Mean	Std Dev	Normal Range	Mean	Std Dev	Normal Range
F/5.8T	All	12	48	155,876.49		12,989.71	3,247.43	1,550.33	1,697.09 - 4,797.76	8,691.12		724.26	181.06	167.93	13.13 - 349.00	4.48	5.35	0.00 - 9.83
H/16.0T	All	12	144	466,731.39		38,894.28	3,241.19	1,498.45	1,742.74 - 4,739.64	8,467.33		705.61	58.80	17.98	40.82 - 76.78	40.96	26.08	14.88 - 67.04
Outlier	***celadm11		12		1.23	* 5,751.37	* 479.28				4.87	v 412.28	v 34.36			^ 97.99		
Outlier	***celadm12		12		1.23	* 5,748.10	* 479.01				4.74	v 401.34	v 33.44			^ 98.19		

				I/O		
Disk Type	Disk Name	Cell Name	Statistic	per Disk	Std Dev	Range
H/16.0T	All	All	Cell Server IOPs	* 3,252.91	1,476.46	0.00 - 7,682.29
			Cell Server IO MB/s	59.80	17.97	5.88 - 113.72
			Cell Server IO Latency	30.89ms	90.81ms	0.00ns - 303.33ms
H/16.0T	CD_03_***celadm11	***celadm11	Cell Server IOPs	* 586.93		
			Cell Server IO MB/s	44.43		
			Cell Server IO Latency	223.35ms		

Looking further at the Cell Server Statistics in Figure 25, we also see different IO types on celadm11 and celadm12 – the IOs are mostly small writes with some large writes. The small reads on the other storage cells are scrub-related.

Although scans would consist of large reads on the storage servers, the different IO profile and, specifically, the large number of writes with the high disk utilization on celadm11 and celadm12 are of concern.

Figure 25. Cell Server Statistics - Outlier Cells



Exadata Cell Server IOPS Statistics - Outlier Cells

- These statistics are collected by the cells and are not restricted to this database or instance
- Outliers are cells whose average performance is outside the normal range, where normal range is -/+ 1 standard deviation
- Outlier cells must have a mininum of 10 IOPs. Idle cells are not considered for outlier analysis.
- Outliers for small reads, small writes, large read, large writes, must have a minimum of 10 requests for the corresponding small read, small write, large read, large write statistic.
- Outliers for hard disks are displayed when Hard Disk IOPs exceeds 306.72 (1% of maximum capacity of 30,672)
- Outliers for flash disks are displayed when Flash Disk IOPs exceeds 95260.8 (1% of maximum capacity of 9,526,080)
- A 'v' and a dark yellow background indicates an outlier value below the low range
- A '^' and a light red background indicates an outlier value above the high range
- Disk Type <F|H|M>/<size>: F-Flash, H-Hard Disk, M-pMEM; PMEM I/O only include remote I/Os processed by cellsrv
- % Total - Avg IOPs of the cell as a percentage of total IOPs for the disk type

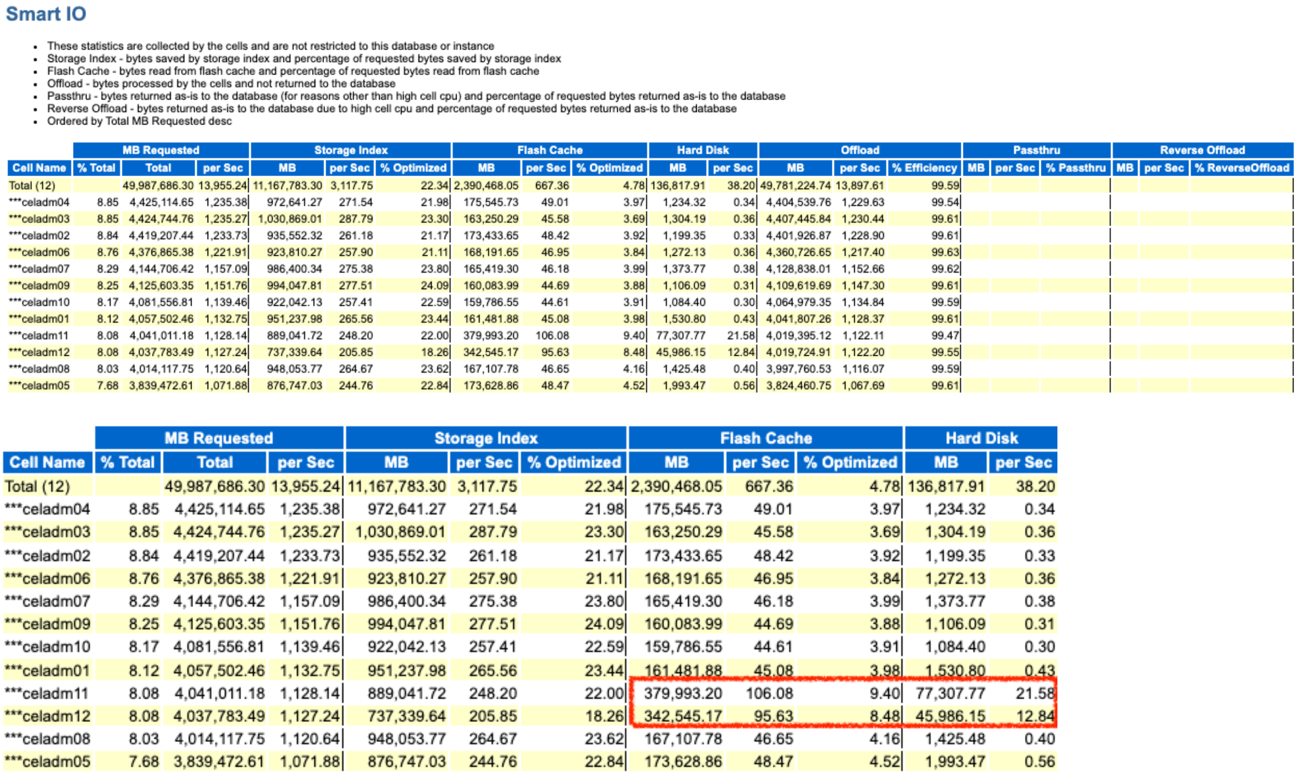
				IOPs						Small Reads/s				Small Writes/s				Large Reads/s				Large Writes/s			
				Total	% Total	Per Cell	Per Disk			Per Cell	Per Disk			Per Cell	Per Disk			Per Cell	Per Disk			Per Cell	Per Disk		
Disk Type	Cell Name	# Cells	# Disks			Average	Mean	Std Dev	Normal Range	Average	Mean	Std Dev	Normal Range	Average	Mean	Std Dev	Normal Range	Average	Mean	Std Dev	Normal Range	Average	Mean	Std Dev	Normal Range
F/5.8T	All	12	48	151,863.64		12,655.30	3,163.83	1,525.14	1,638.68 - 4,688.97	139.65	34.91	19.21	15.70 - 54.12	7,724.76	1,931.19	991.09	940.10 - 2,922.28	4,573.56	1,143.39	1,318.26	0.00 - 2,461.65	217.34	54.33	33.66	20.68 - 87.99
Outlier	***celadm11		4		6.45	9,789.33	2,447.33			^ 263.62	^ 65.90			4,282.57	1,070.64			5,243.15	1,310.79			0.00	0.00		
Outlier	***celadm12		4		6.42	9,745.03	2,436.26			^ 265.22	^ 66.31			4,293.59	1,073.40			5,186.22	1,296.56			0.00	0.00		
H/16.0T	All	12	144	468,419.16		39,034.93	3,252.91	1,476.46	1,776.45 - 4,729.37	37,819.52	3,151.63	1,638.80	1,512.83 - 4,790.42	1,093.43	91.12	204.94	0.00 - 296.06	13.85	1.15	2.49	0.00 - 3.65	108.13	9.01	11.31	0.00 - 20.32
Outlier	***celadm11		12		1.42	* 6,660.35	* 555.03			v 230.88	v 19.24			^ 6,016.97	^ 501.41			68.98	5.75			^ 343.52	^ 28.63		
Outlier	***celadm12		12		1.42	* 6,657.00	* 554.75			v 239.00	v 19.92			^ 6,017.22	^ 501.44			58.58	4.88			^ 342.20	^ 28.52		

Small Reads/s				Small Writes/s			
Per Cell	Per Disk			Per Cell	Per Disk		
Average	Mean	Std Dev	Normal Range	Average	Mean	Std Dev	Normal Range
139.65	34.91	19.21	15.70 - 54.12	7,724.76	1,931.19	991.09	940.10 - 2,922.28
^ 263.62	^ 65.90			4,282.57	1,070.64		
^ 265.22	^ 66.31			4,293.59	1,073.40		
37,819.52	3,151.63	1,638.80	1,512.83 - 4,790.42	1,093.43	91.12	204.94	0.00 - 296.06
v 230.88	v 19.24			^ 6,016.97	^ 501.41		
v 239.00	v 19.92			^ 6,017.22	^ 501.44		

Smart Scans

The **Smart IO** section also shows an overall picture of smart IO activity on the system and gives an idea of how well smart scans are performing[10]. In Figure 26, we see celadm11 and celadm12 behaving differently from the other storage cells. Although the disk IOs aren't that high, they are significantly higher than the other cells, where the smart scans are coming almost entirely from flash cache.

Figure 26. Smart IO from the AWR report that shows smart scan information



Smart IO

- These statistics are collected by the cells and are not restricted to this database or instance
- Storage Index - bytes saved by storage index and percentage of requested bytes saved by storage index
- Flash Cache - bytes read from flash cache and percentage of requested bytes read from flash cache
- Offload - bytes processed by the cells and not returned to the database
- Passthru - bytes returned as-is to the database (for reasons other than high cell cpu) and percentage of requested bytes returned as-is to the database
- Reverse Offload - bytes returned as-is to the database due to high cell cpu and percentage of requested bytes returned as-is to the database
- Ordered by Total MB Requested desc

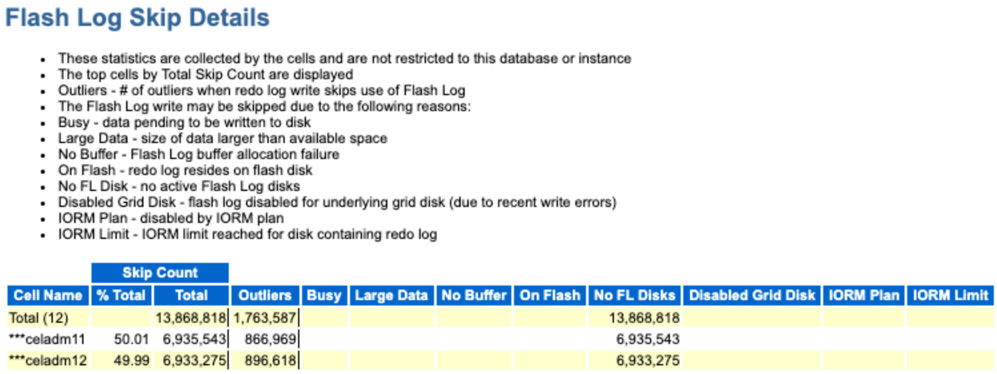
	MB Requested			Storage Index			Flash Cache			Hard Disk		Offload			Passthru			Reverse Offload		
Cell Name	% Total	Total	per Sec	MB	per Sec	% Optimized	MB	per Sec	% Optimized	MB	per Sec	MB	per Sec	% Efficiency	MB	per Sec	% Passthru	MB	per Sec	% ReverseOffload
Total (12)		49,987,686.30	13,955.24	11,167,783.30	3,117.75	22.34	2,390,468.05	667.36	4.78	136,817.91	38.20	49,781,224.74	13,897.61	99.59						
***celadm04	8.85	4,425,114.65	1,235.38	972,641.27	271.54	21.98	175,545.73	49.01	3.97	1,234.32	0.34	4,404,539.76	1,229.63	99.54						
***celadm03	8.85	4,424,744.76	1,235.27	1,030,869.01	287.79	23.30	163,250.29	45.58	3.69	1,304.19	0.36	4,407,445.84	1,230.44	99.61						
***celadm02	8.84	4,419,207.44	1,233.73	935,552.32	261.18	21.17	173,433.65	48.42	3.92	1,199.35	0.33	4,401,926.87	1,228.90	99.61						
***celadm06	8.76	4,376,865.38	1,221.91	923,810.27	257.90	21.11	168,191.65	46.95	3.84	1,272.13	0.36	4,360,726.65	1,217.40	99.63						
***celadm07	8.29	4,144,706.42	1,157.09	986,400.34	275.38	23.80	165,419.30	46.18	3.99	1,373.77	0.38	4,128,838.01	1,152.66	99.62						
***celadm09	8.25	4,125,603.35	1,151.76	994,047.81	277.51	24.09	160,083.99	44.69	3.88	1,106.09	0.31	4,109,619.69	1,147.30	99.61						
***celadm10	8.17	4,081,556.81	1,139.46	922,042.13	257.41	22.59	159,786.55	44.61	3.91	1,084.40	0.30	4,064,979.35	1,134.84	99.59						
***celadm01	8.12	4,057,502.46	1,132.75	951,237.98	265.56	23.44	161,481.88	45.08	3.98	1,530.80	0.43	4,041,807.26	1,128.37	99.61						
***celadm11	8.08	4,041,011.18	1,128.14	889,041.72	248.20	22.00	379,993.20	106.08	9.40	77,307.77	21.58	4,019,395.12	1,122.11	99.47						
***celadm12	8.08	4,037,783.49	1,127.24	737,339.64	205.85	18.26	342,545.17	95.63	8.48	45,986.15	12.84	4,019,724.91	1,122.20	99.55						
***celadm08	8.03	4,014,117.75	1,120.64	948,053.77	264.67	23.62	167,107.78	46.65	4.16	1,425.48	0.40	3,997,760.53	1,116.07	99.59						
***celadm05	7.68	3,839,472.61	1,071.88	876,747.03	244.76	22.84	173,628.86	48.47	4.52	1,993.47	0.56	3,824,460.75	1,067.69	99.61						

	MB Requested			Storage Index			Flash Cache			Hard Disk	
Cell Name	% Total	Total	per Sec	MB	per Sec	% Optimized	MB	per Sec	% Optimized	MB	per Sec
Total (12)		49,987,686.30	13,955.24	11,167,783.30	3,117.75	22.34	2,390,468.05	667.36	4.78	136,817.91	38.20
***celadm04	8.85	4,425,114.65	1,235.38	972,641.27	271.54	21.98	175,545.73	49.01	3.97	1,234.32	0.34
***celadm03	8.85	4,424,744.76	1,235.27	1,030,869.01	287.79	23.30	163,250.29	45.58	3.69	1,304.19	0.36
***celadm02	8.84	4,419,207.44	1,233.73	935,552.32	261.18	21.17	173,433.65	48.42	3.92	1,199.35	0.33
***celadm06	8.76	4,376,865.38	1,221.91	923,810.27	257.90	21.11	168,191.65	46.95	3.84	1,272.13	0.36
***celadm07	8.29	4,144,706.42	1,157.09	986,400.34	275.38	23.80	165,419.30	46.18	3.99	1,373.77	0.38
***celadm09	8.25	4,125,603.35	1,151.76	994,047.81	277.51	24.09	160,083.99	44.69	3.88	1,106.09	0.31
***celadm10	8.17	4,081,556.81	1,139.46	922,042.13	257.41	22.59	159,786.55	44.61	3.91	1,084.40	0.30
***celadm01	8.12	4,057,502.46	1,132.75	951,237.98	265.56	23.44	161,481.88	45.08	3.98	1,530.80	0.43
***celadm11	8.08	4,041,011.18	1,128.14	889,041.72	248.20	22.00	379,993.20	106.08	9.40	77,307.77	21.58
***celadm12	8.08	4,037,783.49	1,127.24	737,339.64	205.85	18.26	342,545.17	95.63	8.48	45,986.15	12.84
***celadm08	8.03	4,014,117.75	1,120.64	948,053.77	264.67	23.62	167,107.78	46.65	4.16	1,425.48	0.40
***celadm05	7.68	3,839,472.61	1,071.88	876,747.03	244.76	22.84	173,628.86	48.47	4.52	1,993.47	0.56

Smart Flash Log

From the Exadata Configuration section, we saw that both flash cache and flash log were configured differently on celadm11 and celadm12 than on the other storage cells. Although redo log writes were not an issue, we still see a difference in flash log behavior on these two cells. In Figure 27, we see skips on the two cells because flash log has not been configured. This is consistent with the information that was observed in the **Exadata Configuration - Storage Information** section.

Figure 27. Flash Log Skip Details shows celadm11 and celadm12 do not have flash logs



Flash Log Skip Details

- These statistics are collected by the cells and are not restricted to this database or instance
- The top cells by Total Skip Count are displayed
- Outliers - # of outliers when redo log write skips use of Flash Log
- The Flash Log write may be skipped due to the following reasons:
- Busy - data pending to be written to disk
- Large Data - size of data larger than available space
- No Buffer - Flash Log buffer allocation failure
- On Flash - redo log resides on flash disk
- No FL Disk - no active Flash Log disks
- Disabled Grid Disk - flash log disabled for underlying grid disk (due to recent write errors)
- IORM Plan - disabled by IORM plan
- IORM Limit - IORM limit reached for disk containing redo log

	Skip Count										
Cell Name	% Total	Total	Outliers	Busy	Large Data	No Buffer	On Flash	No FL Disks	Disabled Grid Disk	IORM Plan	IORM Limit
Total (12)		13,868,818	1,763,587					13,868,818			
***celadm11	50.01	6,935,543	866,969					6,935,543			
***celadm12	49.99	6,933,275	896,618					6,933,275			

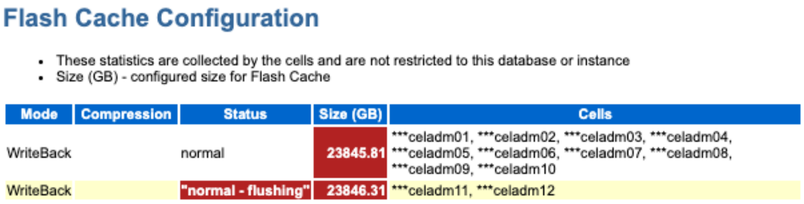
[10] When looking at specific smart scan issues that only affect a small set of SQL statements, SQL Monitor is a good diagnostic tool.

Smart Flash Cache

From the **Exadata Configuration - Storage Information** section, we know flash cache was slightly larger on the two cells that are being investigated. The Flash Cache Configuration section shown in Figure 28 gives us additional insight, and in this case is highlighting the most likely cause of the issues.

The **Flash Cache Configuration** shows the two cells of interest have a status of *normal - flushing*. Storage cells in *normal - flushing* status indicates that the data in flash cache is currently being flushed to hard disk, and client IOs will not be able to use flash cache. Instead, those client IOs will be redirected to hard disk.

Figure 28. Flash Cache Configuration shows differences between the cells



Flash Cache Configuration

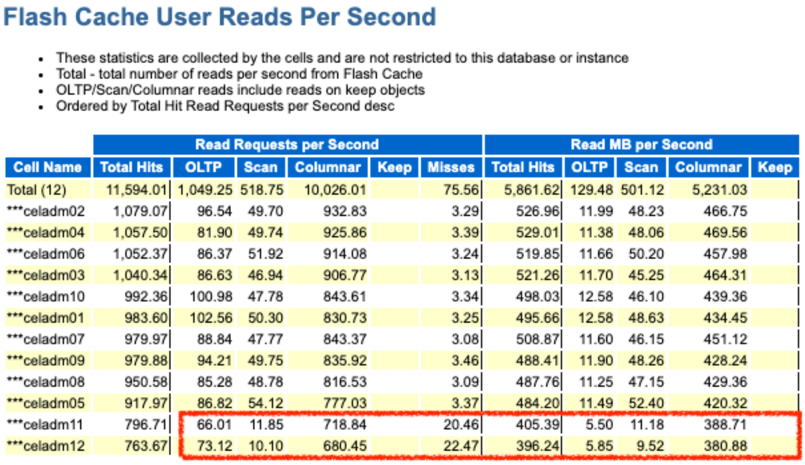
- These statistics are collected by the cells and are not restricted to this database or instance
- Size (GB) - configured size for Flash Cache

Mode	Compression	Status	Size (GB)	Cells
WriteBack		normal	23845.81	***celadm01, ***celadm02, ***celadm03, ***celadm04, ***celadm05, ***celadm06, ***celadm07, ***celadm08, ***celadm09, ***celadm10
WriteBack		"normal - flushing"	23846.31	***celadm11, ***celadm12

Although it already looks like this is the cause of the issues, for due diligence, we review the other sections to ensure that there are no other issues on the other cells.

The **Flash Cache User Reads Per Second** in Figure 29 shows the same two cells with a higher miss rate and a lower request rate than the other cells.

Figure 29. Flash Cache User Reads Per Second



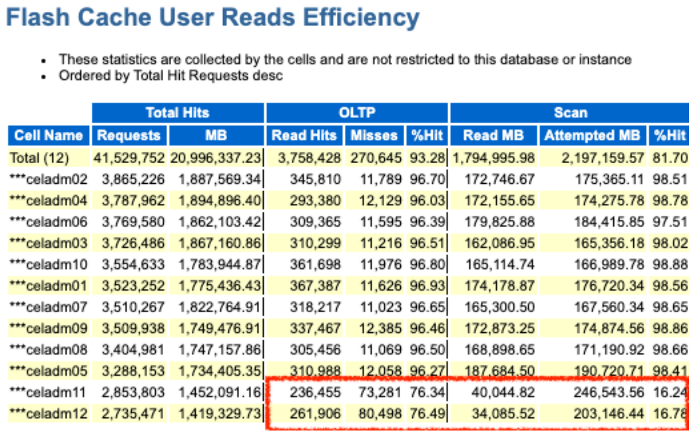
Flash Cache User Reads Per Second

- These statistics are collected by the cells and are not restricted to this database or instance
- Total - total number of reads per second from Flash Cache
- OLTP/Scan/Columnar reads include reads on keep objects
- Ordered by Total Hit Read Requests per Second desc

	Read Requests per Second						Read MB per Second				
Cell Name	Total Hits	OLTP	Scan	Columnar	Keep	Misses	Total Hits	OLTP	Scan	Columnar	Keep
Total (12)	11,594.01	1,049.25	518.75	10,026.01		75.56	5,861.62	129.48	501.12	5,231.03	
***celadm02	1,079.07	96.54	49.70	932.83		3.29	526.96	11.99	48.23	466.75	
***celadm04	1,057.50	81.90	49.74	925.86		3.39	529.01	11.38	48.06	469.56	
***celadm06	1,052.37	86.37	51.92	914.08		3.24	519.85	11.66	50.20	457.98	
***celadm03	1,040.34	86.63	46.94	906.77		3.13	521.26	11.70	45.25	464.31	
***celadm10	992.36	100.98	47.78	843.61		3.34	498.03	12.58	46.10	439.36	
***celadm01	983.60	102.56	50.30	830.73		3.25	495.66	12.58	48.63	434.45	
***celadm07	979.97	88.84	47.77	843.37		3.08	508.87	11.60	46.15	451.12	
***celadm09	979.88	94.21	49.75	835.92		3.46	488.41	11.90	48.26	428.24	
***celadm08	950.58	85.28	48.78	816.53		3.09	487.76	11.25	47.15	429.36	
***celadm05	917.97	86.82	54.12	777.03		3.37	484.20	11.49	52.40	420.32	
***celadm11	796.71	66.01	11.85	718.84		20.46	405.39	5.50	11.18	388.71	
***celadm12	763.67	73.12	10.10	680.45		22.47	396.24	5.85	9.52	380.88	

In addition, the **Flash Cache User Reads Efficiency** in Figure 30 shows significantly lower hit percentage rates (%Hit) for both OLTP and Scans on the same two cells.

Figure 30. Flash Cache User Reads Efficiency



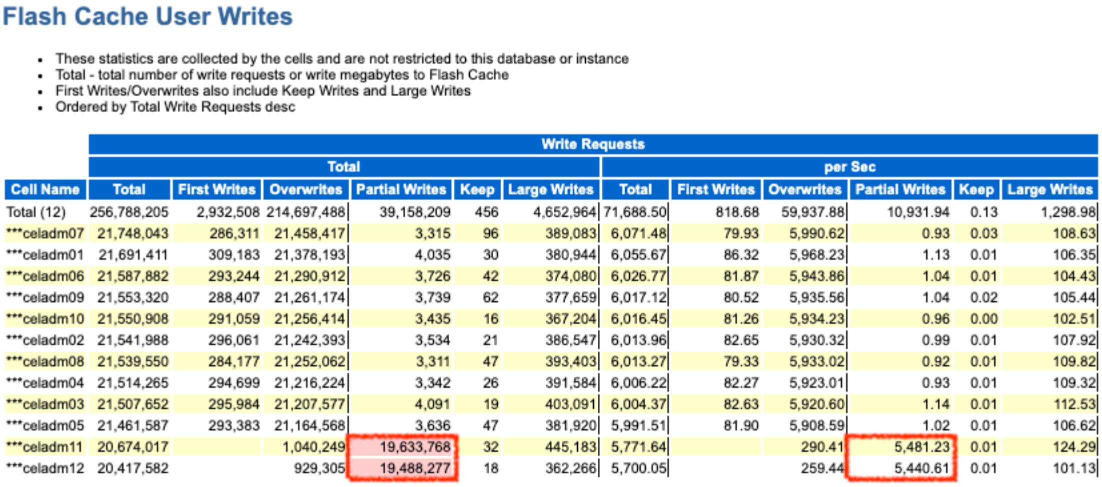
Flash Cache User Reads Efficiency

- These statistics are collected by the cells and are not restricted to this database or instance
- Ordered by Total Hit Requests desc

	Total Hits		OLTP			Scan		
Cell Name	Requests	MB	Read Hits	Misses	%Hit	Read MB	Attempted MB	%Hit
Total (12)	41,529,752	20,996,337.23	3,758,428	270,645	93.28	1,794,995.98	2,197,159.57	81.70
***celadm02	3,865,226	1,887,569.34	345,810	11,789	96.70	172,746.67	175,365.11	98.51
***celadm04	3,787,962	1,894,896.40	293,380	12,129	96.03	172,155.65	174,275.78	98.78
***celadm06	3,769,580	1,862,103.42	309,365	11,595	96.39	179,825.88	184,415.85	97.51
***celadm03	3,726,486	1,867,160.86	310,299	11,216	96.51	162,086.95	165,356.18	98.02
***celadm10	3,554,633	1,783,944.87	361,698	11,976	96.80	165,114.74	166,989.78	98.88
***celadm01	3,523,252	1,775,436.43	367,387	11,626	96.93	174,178.87	176,720.34	98.56
***celadm07	3,510,267	1,822,764.91	318,217	11,023	96.65	165,300.50	167,560.34	98.65
***celadm09	3,509,938	1,749,476.91	337,467	12,385	96.46	172,873.25	174,874.56	98.86
***celadm08	3,404,981	1,747,157.86	305,456	11,069	96.50	168,898.65	171,190.92	98.66
***celadm05	3,288,153	1,734,405.35	310,988	12,058	96.27	187,684.50	190,720.71	98.41
***celadm11	2,853,803	1,452,091.16	236,455	73,281	76.34	40,044.82	246,543.56	16.24
***celadm12	2,735,471	1,419,329.73	261,906	80,498	76.49	34,085.52	203,146.44	16.78

Similarly, reviewing the **Flash Cache User Writes** section in Figure 31, celadm11 and celadm12 have over 19 million partial writes for the report interval, and 5000 partial writes per second. Partial writes are not common, and occur when the write goes to both flash cache and hard disk. The partial writes on these two cells are, again, a result of the *normal - flushing* status for flash cache.

Figure 31. Flash Cache User Writes



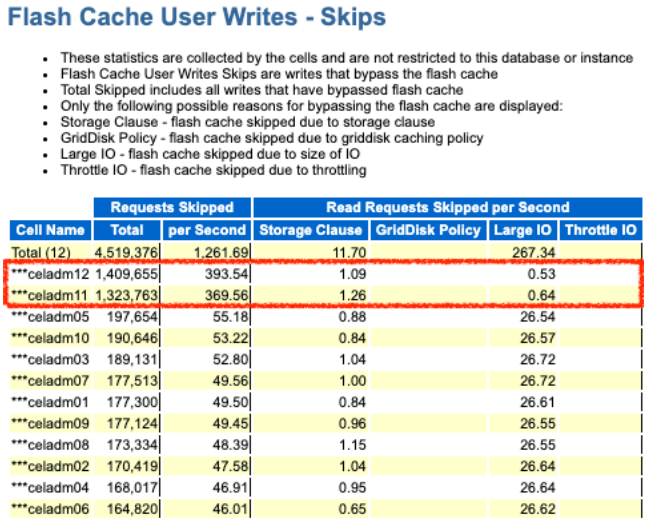
Flash Cache User Writes

- These statistics are collected by the cells and are not restricted to this database or instance
- Total - total number of write requests or write megabytes to Flash Cache
- First Writes/Overwrites also include Keep Writes and Large Writes
- Ordered by Total Write Requests desc

	Write Requests											
	Total						per Sec					
Cell Name	Total	First Writes	Overwrites	Partial Writes	Keep	Large Writes	Total	First Writes	Overwrites	Partial Writes	Keep	Large Writes
Total (12)	256,788,205	2,932,508	214,697,488	39,158,209	456	4,652,964	71,688.50	818.68	59,937.88	10,931.94	0.13	1,298.98
***celadm07	21,748,043	286,311	21,458,417	3,315	96	389,083	6,071.48	79.93	5,990.62	0.93	0.03	108.63
***celadm01	21,691,411	309,183	21,378,193	4,035	30	380,944	6,055.67	86.32	5,968.23	1.13	0.01	106.35
***celadm06	21,587,882	293,244	21,290,912	3,726	42	374,080	6,026.77	81.87	5,943.86	1.04	0.01	104.43
***celadm09	21,553,320	288,407	21,261,174	3,739	62	377,659	6,017.12	80.52	5,935.56	1.04	0.02	105.44
***celadm10	21,550,908	291,059	21,256,414	3,435	16	367,204	6,016.45	81.26	5,934.23	0.96	0.00	102.51
***celadm02	21,541,988	296,061	21,242,393	3,534	21	386,547	6,013.96	82.65	5,930.32	0.99	0.01	107.92
***celadm08	21,539,550	284,177	21,252,062	3,311	47	393,403	6,013.27	79.33	5,933.02	0.92	0.01	109.82
***celadm04	21,514,265	294,699	21,216,224	3,342	26	391,584	6,006.22	82.27	5,923.01	0.93	0.01	109.32
***celadm03	21,507,652	295,984	21,207,577	4,091	19	403,091	6,004.37	82.63	5,920.60	1.14	0.01	112.53
***celadm05	21,461,587	293,383	21,164,568	3,636	47	381,920	5,991.51	81.90	5,908.59	1.02	0.01	106.62
***celadm11	20,674,017		1,040,249	19,633,768	32	445,183	5,771.64		290.41	5,481.23	0.01	124.29
***celadm12	20,417,582		929,305	19,488,277	18	362,266	5,700.05		259.44	5,440.61	0.01	101.13

Similarly, as seen in Figure 32, the **Flash Cache User Writes - Skips** shows writes bypassing the flash cache on celadm11 and celadm13. In this particular AWR report, we see that there are writes bypassing the flash cache, but we don't see the reasons why this is happening. Additional information that will make the reasons more apparent is available starting Oracle Database 19.19. In this case we can assume these writes are likely due to the *normal - flushing* status of flash cache.

Figure 32. Flash Cache User Writes - Skips



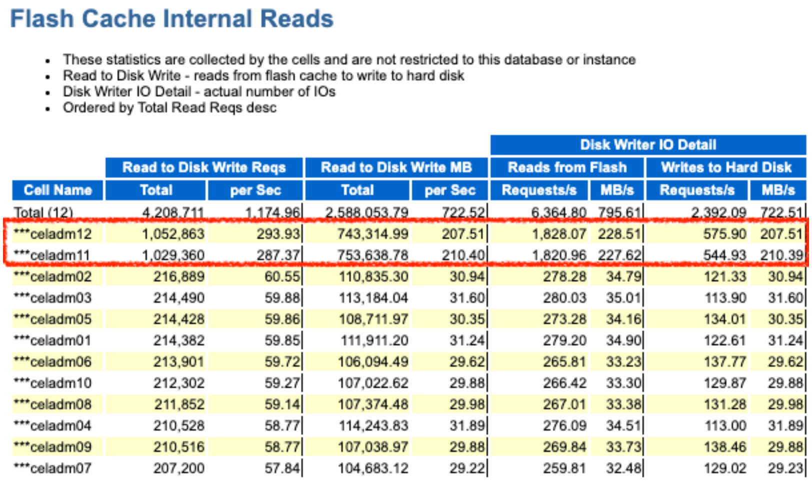
Flash Cache User Writes - Skips

- These statistics are collected by the cells and are not restricted to this database or instance
- Flash Cache User Writes Skips are writes that bypass the flash cache
- Total Skipped includes all writes that have bypassed flash cache
- Only the following possible reasons for bypassing the flash cache are displayed:
- Storage Clause - flash cache skipped due to storage clause
- GridDisk Policy - flash cache skipped due to griddisk caching policy
- Large IO - flash cache skipped due to size of IO
- Throttle IO - flash cache skipped due to throttling

	Requests Skipped		Read Requests Skipped per Second			
Cell Name	Total	per Second	Storage Clause	GridDisk Policy	Large IO	Throttle IO
Total (12)	4,519,376	1,261.69	11.70		267.34	
***celadm12	1,409,655	393.54	1.09		0.53	
***celadm11	1,323,763	369.56	1.26		0.64	
***celadm05	197,654	55.18	0.88		26.54	
***celadm10	190,646	53.22	0.84		26.57	
***celadm03	189,131	52.80	1.04		26.72	
***celadm07	177,513	49.56	1.00		26.72	
***celadm01	177,300	49.50	0.84		26.61	
***celadm09	177,124	49.45	0.96		26.55	
***celadm08	173,334	48.39	1.15		26.55	
***celadm02	170,419	47.58	1.04		26.64	
***celadm04	168,017	46.91	0.95		26.64	
***celadm06	164,820	46.01	0.65		26.62	

The **Flash Cache Internal Reads** section, as shown in Figure 33, shows disk writer activity. Disk Writer is responsible for syncing dirty data from flash cache to hard disk. Beginning Oracle Database 19.19, the AWR reports also include the type of disk writer writes, which will show if the writes are flush related. In this case, we see more writes on celadm11 and celadm12, which we can assume is a result of the flush.

Figure 33. Flash Cache Internal Reads



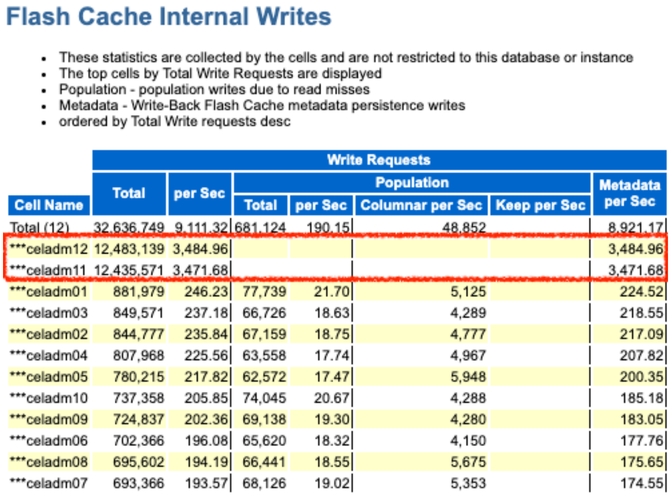
Flash Cache Internal Reads

- These statistics are collected by the cells and are not restricted to this database or instance
- Read to Disk Write - reads from flash cache to write to hard disk
- Disk Writer IO Detail - actual number of IOs
- Ordered by Total Read Reqs desc

					Disk Writer IO Detail			
	Read to Disk Write Reqs		Read to Disk Write MB		Reads from Flash		Writes to Hard Disk	
Cell Name	Total	per Sec	Total	per Sec	Requests/s	MB/s	Requests/s	MB/s
Total (12)	4,208,711	1,174.96	2,588,053.79	722.52	6,364.80	795.61	2,392.09	722.51
***celadm12	1,052,863	293.93	743,314.99	207.51	1,828.07	228.51	575.90	207.51
***celadm11	1,029,360	287.37	753,638.78	210.40	1,820.96	227.62	544.93	210.39
***celadm02	216,889	60.55	110,835.30	30.94	278.28	34.79	121.33	30.94
***celadm03	214,490	59.88	113,184.04	31.60	280.03	35.01	113.90	31.60
***celadm05	214,428	59.86	108,711.97	30.35	273.28	34.16	134.01	30.35
***celadm01	214,382	59.85	111,911.20	31.24	279.20	34.90	122.61	31.24
***celadm06	213,901	59.72	106,094.49	29.62	265.81	33.23	137.77	29.62
***celadm10	212,302	59.27	107,022.62	29.88	266.42	33.30	129.87	29.88
***celadm08	211,852	59.14	107,374.48	29.98	267.01	33.38	131.28	29.98
***celadm04	210,528	58.77	114,243.83	31.89	276.09	34.51	113.00	31.89
***celadm09	210,516	58.77	107,038.97	29.88	269.84	33.73	138.46	29.88
***celadm07	207,200	57.84	104,683.12	29.22	259.81	32.48	129.02	29.23

Finally, the **Flash Cache Internal Writes** section, as shown in Figure 34, indicates writes are not going to flash cache on celadm11 and celadm12. Typically, populations of the flash cache occur as a result of misses on the flash cache. In this case, despite the higher number of misses, there is no population write, which is consistent with the flush operation.

Figure 34. Flash Cache Internal Writes



Flash Cache Internal Writes

- These statistics are collected by the cells and are not restricted to this database or instance
- The top cells by Total Write Requests are displayed
- Population - population writes due to read misses
- Metadata - Write-Back Flash Cache metadata persistence writes
- ordered by Total Write requests desc

	Write Requests						
	Total	per Sec	Population				Metadata per Sec
Cell Name			Total	per Sec	Columnar per Sec	Keep per Sec	
Total (12)	32,636,749	9,111.32	681,124	190.15	48,852		8,921.17
***celadm12	12,483,139	3,484.96					3,484.96
***celadm11	12,435,571	3,471.68					3,471.68
***celadm01	881,979	246.23	77,739	21.70	5,125		224.52
***celadm03	849,571	237.18	66,726	18.63	4,289		218.55
***celadm02	844,777	235.84	67,159	18.75	4,777		217.09
***celadm04	807,968	225.56	63,558	17.74	4,967		207.82
***celadm05	780,215	217.82	62,572	17.47	5,948		200.35
***celadm10	737,358	205.85	74,045	20.67	4,288		185.18
***celadm09	724,837	202.36	69,138	19.30	4,280		183.05
***celadm06	702,366	196.08	65,620	18.32	4,150		177.76
***celadm08	695,602	194.19	66,441	18.55	5,675		175.65
***celadm07	693,366	193.57	68,126	19.02	5,353		174.55

The Flash Cache sections all indicate activity (or in some cases, lack of IOs) in flash cache on celadm11 and celadm12, both of which show a status of *normal - flushing*.

At this point, we can safely say that the state of the flash cache on those two cells are primarily responsible for the issues observed. But we will continue with the remaining sections to validate our hypothesis.

IO Reasons

The IO Reasons section tells us why the IOs are issued on the storage servers. The IOs in IO Reasons include both reads and writes, as well as hard disk and flash devices.

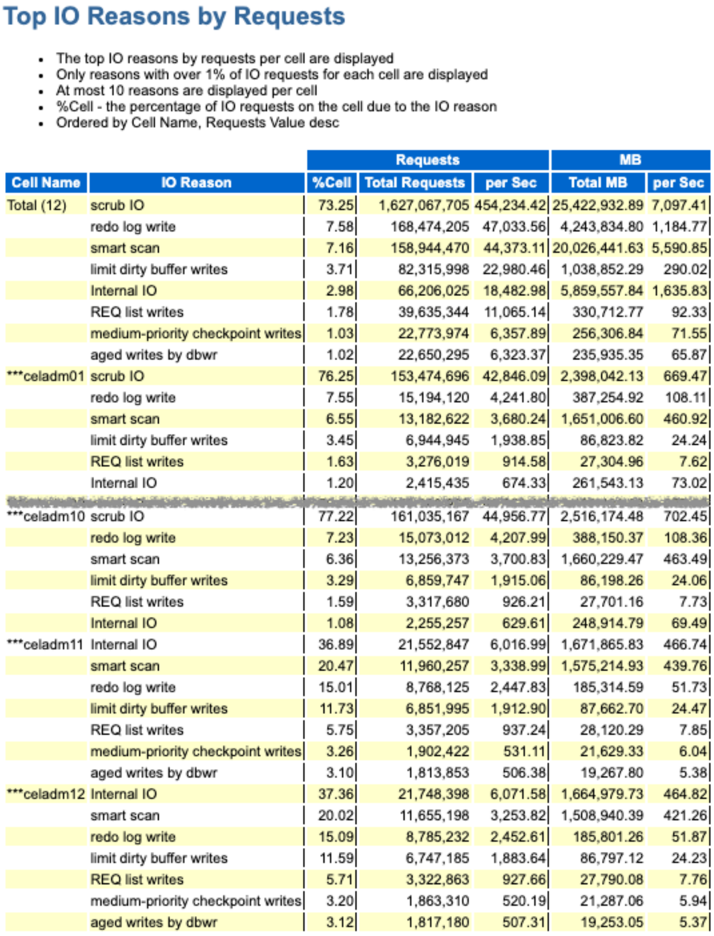
In Figure 35 and Figure 36, we see the IO reasons on celadm11 and celadm12 are different from the other storage cells.

The other cells show:

- Scrub IO – this results in small reads from hard disk, and typically does not impact client.
- Redo log writes – writes to the redo logs. With Smart Flash Log and Smart Flash Log Write-Back,[11] the redo log writes are expected to go to flash log and flash cache.
- Smart scan – smart scan activity.
- Limited dirty buffer writes, aged writes by DBWR, and medium-priority checkpoint writes – writes from database writers.

Storage cells celadm11 and celadm12 show 36-37 percent of the IOs are due to Internal IO, which is much higher than that of the other cells. Also, note that scrub is not running on celadm11 and celadm12 – the two cells where we are seeing the issues.

Figure 35. IO Reasons by Requests



Top IO Reasons by Requests

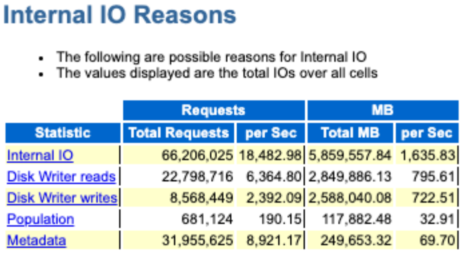
- The top IO reasons by requests per cell are displayed
- Only reasons with over 1% of IO requests for each cell are displayed
- At most 10 reasons are displayed per cell
- %Cell - the percentage of IO requests on the cell due to the IO reason
- Ordered by Cell Name, Requests Value desc

		Requests			MB	
Cell Name	IO Reason	%Cell	Total Requests	per Sec	Total MB	per Sec
Total (12)	scrub IO	73.25	1,627,067,705	454,234.42	25,422,932.89	7,097.41
	redo log write	7.58	168,474,205	47,033.56	4,243,834.80	1,184.77
	smart scan	7.16	158,944,470	44,373.11	20,026,441.63	5,590.85
	limit dirty buffer writes	3.71	82,315,998	22,980.46	1,038,852.29	290.02
	Internal IO	2.98	66,206,025	18,482.98	5,859,557.84	1,635.83
	REQ list writes	1.78	39,635,344	11,065.14	330,712.77	92.33
	medium-priority checkpoint writes	1.03	22,773,974	6,357.89	256,306.84	71.55
	aged writes by dbwr	1.02	22,650,295	6,323.37	235,935.35	65.87
***celadm01	scrub IO	76.25	153,474,696	42,846.09	2,398,042.13	669.47
	redo log write	7.55	15,194,120	4,241.80	387,254.92	108.11
	smart scan	6.55	13,182,622	3,680.24	1,651,006.60	460.92
	limit dirty buffer writes	3.45	6,944,945	1,938.85	86,823.82	24.24
	REQ list writes	1.63	3,276,019	914.58	27,304.96	7.62
	Internal IO	1.20	2,415,435	674.33	261,543.13	73.02
***celadm10	scrub IO	77.22	161,035,167	44,956.77	2,516,174.48	702.45
	redo log write	7.23	15,073,012	4,207.99	388,150.37	108.36
	smart scan	6.36	13,256,373	3,700.83	1,660,229.47	463.49
	limit dirty buffer writes	3.29	6,859,747	1,915.06	86,198.26	24.06
	REQ list writes	1.59	3,317,680	926.21	27,701.16	7.73
	Internal IO	1.08	2,255,257	629.61	248,914.79	69.49
***celadm11	Internal IO	36.89	21,552,847	6,016.99	1,671,865.83	466.74
	smart scan	20.47	11,960,257	3,338.99	1,575,214.93	439.76
	redo log write	15.01	8,768,125	2,447.83	185,314.59	51.73
	limit dirty buffer writes	11.73	6,851,995	1,912.90	87,662.70	24.47
	REQ list writes	5.75	3,357,205	937.24	28,120.29	7.85
	medium-priority checkpoint writes	3.26	1,902,422	531.11	21,629.33	6.04
	aged writes by dbwr	3.10	1,813,853	506.38	19,267.80	5.38
***celadm12	Internal IO	37.36	21,748,398	6,071.58	1,664,979.73	464.82
	smart scan	20.02	11,655,198	3,253.82	1,508,940.39	421.26
	redo log write	15.09	8,785,232	2,452.61	185,801.26	51.87
	limit dirty buffer writes	11.59	6,747,185	1,883.64	86,797.12	24.23
	REQ list writes	5.71	3,322,863	927.66	27,790.08	7.76
	medium-priority checkpoint writes	3.20	1,863,310	520.19	21,287.06	5.94
	aged writes by dbwr	3.12	1,817,180	507.31	19,253.05	5.37

[11] Smart Flash Log Write-Back was introduced in Oracle Exadata System Software 20.1.0

Figure 36 shows the potential causes of the Internal IOs. This is aggregated across all cells. The links will directly show the relevant sections of the AWR report which will have the statistics for the individual cells.

Figure 36. Internal IO Reasons.



Internal IO Reasons

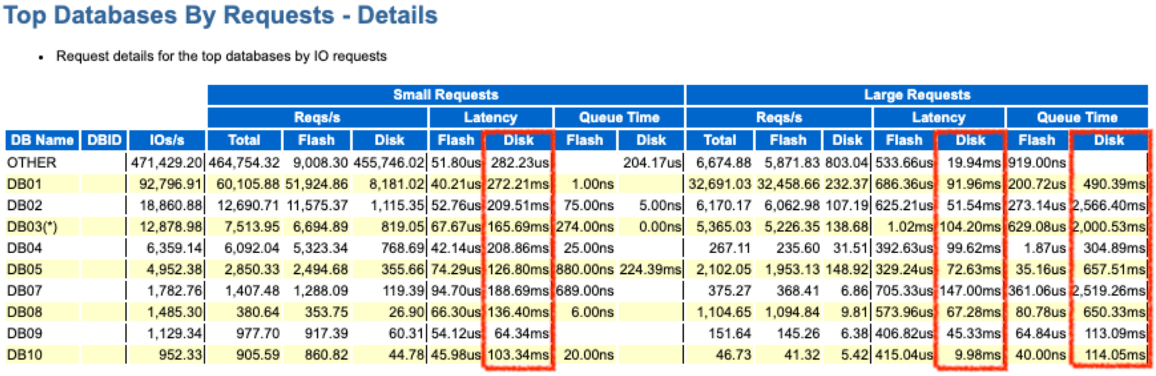
- The following are possible reasons for Internal IO
- The values displayed are the total IOs over all cells

	Requests		MB	
Statistic	Total Requests	per Sec	Total MB	per Sec
Internal IO	66,206,025	18,482.98	5,859,557.84	1,635.83
Disk Writer reads	22,798,716	6,364.80	2,849,886.13	795.61
Disk Writer writes	8,568,449	2,392.09	2,588,040.08	722.51
Population	681,124	190.15	117,882.48	32.91
Metadata	31,955,625	8,921.17	249,653.32	69.70

Top Databases

Reviewing the Top Databases in Figure 37 shows increased disk IO latency for large IOs on all of the databases. The increased disk IO latency also results in increased IORM queueing for large IOs. These large latencies directly affect the *cell smart table scan* wait event seen on the database.

Figure 37. Top Databases By Requests - Details



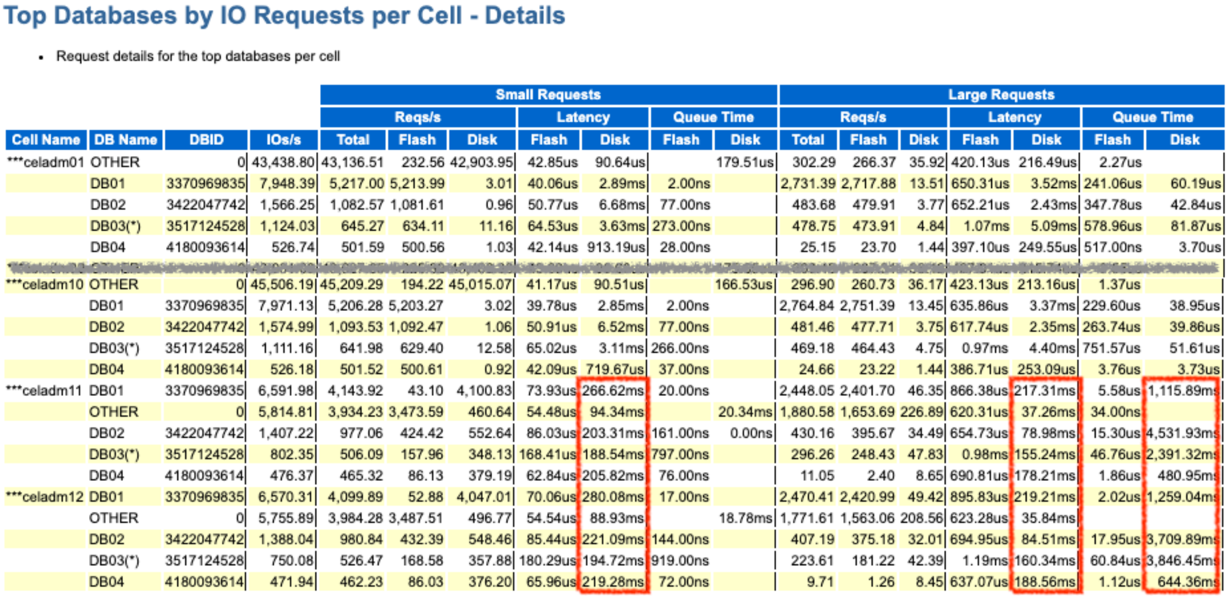
Top Databases By Requests - Details

- Request details for the top databases by IO requests

			Small Requests							Large Requests						
			Reqs/s			Latency		Queue Time		Reqs/s			Latency		Queue Time	
DB Name	DBID	IOs/s	Total	Flash	Disk	Flash	Disk	Flash	Disk	Total	Flash	Disk	Flash	Disk	Flash	Disk
OTHER		471,429.20	464,754.32	9,008.30	455,746.02	51.80us	282.23us		204.17us	6,674.88	5,871.83	803.04	533.66us	19.94ms	919.00ns	
DB01		92,796.91	60,105.88	51,924.86	8,181.02	40.21us	272.21ms	1.00ns		32,691.03	32,458.66	232.37	686.36us	91.96ms	200.72us	490.39ms
DB02		18,860.88	12,690.71	11,575.37	1,115.35	52.76us	209.51ms	75.00ns	5.00ns	6,170.17	6,062.98	107.19	625.21us	51.54ms	273.14us	2,566.40ms
DB03(*)		12,878.98	7,513.95	6,694.89	819.05	67.67us	165.69ms	274.00ns	0.00ns	5,365.03	5,226.35	138.68	1.02ms	104.20ms	629.08us	2,000.53ms
DB04		6,359.14	6,092.04	5,323.34	768.69	42.14us	208.86ms	25.00ns		267.11	235.60	31.51	392.63us	99.62ms	1.87us	304.89ms
DB05		4,952.38	2,850.33	2,494.68	355.66	74.29us	126.80ms	880.00ns	224.39ms	2,102.05	1,953.13	148.92	329.24us	72.63ms	35.16us	657.51ms
DB07		1,782.76	1,407.48	1,288.09	119.39	94.70us	188.69ms	689.00ns		375.27	368.41	6.86	705.33us	147.00ms	361.06us	2,519.26ms
DB08		1,485.30	380.64	353.75	26.90	66.30us	136.40ms	6.00ns		1,104.65	1,094.84	9.81	573.96us	67.28ms	80.78us	650.33ms
DB09		1,129.34	977.70	917.39	60.31	54.12us	64.34ms			151.64	145.26	6.38	406.82us	45.33ms	64.84us	113.09ms
DB10		952.33	905.59	860.82	44.78	45.98us	103.34ms	20.00ns		46.73	41.32	5.42	415.04us	9.98ms	40.00ns	114.05ms

Looking at the storage cells in more detail in Figure 38, we see the high latencies and the high IORM queue times only occur on the two cells, celadm11 and celadm12 – the same two cells we have been reviewing all along.

Figure 38. Top Databases by IO Requests per Cell - Details



Top Databases by IO Requests per Cell - Details

- Request details for the top databases per cell

				Small Requests							Large Requests						
				Reqs/s			Latency		Queue Time		Reqs/s			Latency		Queue Time	
Cell Name	DB Name	DBID	IOs/s	Total	Flash	Disk	Flash	Disk	Flash	Disk	Total	Flash	Disk	Flash	Disk	Flash	Disk
***celadm01	OTHER	0	43,438.80	43,136.51	232.56	42,903.95	42.85us	90.64us		179.51us	302.29	266.37	35.92	420.13us	216.49us	2.27us	
	DB01	3370969835	7,948.39	5,217.00	5,213.99	3.01	40.06us	2.89ms	2.00ns		2,731.39	2,717.88	13.51	650.31us	3.52ms	241.06us	60.19us
	DB02	3422047742	1,566.25	1,082.57	1,081.61	0.96	50.77us	6.68ms	77.00ns		483.68	479.91	3.77	652.21us	2.43ms	347.78us	42.84us
	DB03(*)	3517124528	1,124.03	645.27	634.11	11.16	64.53us	3.63ms	273.00ns		478.75	473.91	4.84	1.07ms	5.09ms	578.96us	81.87us
	DB04	4180093614	526.74	501.59	500.56	1.03	42.14us	913.19us	28.00ns		25.15	23.70	1.44	397.10us	249.55us	517.00ns	3.70us
***celadm10	OTHER	0	45,506.19	45,209.29	194.22	45,015.07	41.17us	90.51us		166.53us	296.90	260.73	36.17	423.13us	213.16us	1.37us	
	DB01	3370969835	7,971.13	5,206.28	5,203.27	3.02	39.78us	2.85ms	2.00ns		2,764.84	2,751.39	13.45	635.86us	3.37ms	229.60us	38.95us
	DB02	3422047742	1,574.99	1,093.53	1,092.47	1.06	50.91us	6.52ms	77.00ns		481.46	477.71	3.75	617.74us	2.35ms	263.74us	39.86us
	DB03(*)	3517124528	1,111.16	641.98	629.40	12.58	65.02us	3.11ms	266.00ns		469.18	464.43	4.75	0.97ms	4.40ms	751.57us	51.61us
	DB04	4180093614	526.18	501.52	500.61	0.92	42.09us	719.67us	37.00ns		24.66	23.22	1.44	386.71us	253.09us	3.76us	3.73us
***celadm11	DB01	3370969835	6,591.98	4,143.92	43.10	4,100.83	73.93us	266.62ms	20.00ns		2,448.05	2,401.70	46.35	866.38us	217.31ms	5.58us	1,115.89ms
	OTHER	0	5,814.81	3,934.23	3,473.59	460.64	54.48us	94.34ms		20.34ms	1,880.58	1,653.69	226.89	620.31us	37.26ms	34.00ns	
	DB02	3422047742	1,407.22	977.06	424.42	552.64	86.03us	203.31ms	161.00ns	0.00ns	430.16	395.67	34.49	654.73us	78.98ms	15.30us	4,531.93ms
	DB03(*)	3517124528	802.35	506.09	157.96	348.13	168.41us	188.54ms	797.00ns		296.26	248.43	47.83	0.98ms	155.24ms	46.76us	2,391.32ms
	DB04	4180093614	476.37	465.32	86.13	379.19	62.84us	205.82ms	76.00ns		11.05	2.40	8.65	690.81us	178.21ms	1.86us	480.95ms
***celadm12	DB01	3370969835	6,570.31	4,099.89	52.88	4,047.01	70.06us	280.08ms	17.00ns		2,470.41	2,420.99	49.42	895.83us	219.21ms	2.02us	1,259.04ms
	OTHER	0	5,755.89	3,984.28	3,487.51	496.77	54.54us	88.93ms		18.78ms	1,771.61	1,563.06	208.56	623.28us	35.84ms		
	DB02	3422047742	1,388.04	980.84	432.39	548.46	85.44us	221.09ms	144.00ns		407.19	375.18	32.01	694.95us	84.51ms	17.95us	3,709.89ms
	DB03(*)	3517124528	750.08	526.47	168.58	357.88	180.29us	194.72ms	919.00ns		223.61	181.22	42.39	1.19ms	160.34ms	60.84us	3,846.45ms
	DB04	4180093614	471.94	462.23	86.03	376.20	65.96us	219.28ms	72.00ns		9.71	1.26	8.45	637.07us	188.56ms	1.12us	644.36ms

Analysis Summary

Here is a summary of what we know from our analysis of this example:

- Database is experiencing poor IO performance, primarily observed on the *cell smart table scan* wait event.
- Flash Cache Configuration indicates two cells have a status of *normal - flushing*. This means that flash cache is flushing all its data to hard disk, and IOs on these cells may not be able to use flash cache. The difference in IO pattern on these two cells are evident in all the other Flash Cache sections.
- IO Outliers show IO outliers on the same two cells. The IO pattern on these two cells indicates increased write activity. The other cells show increased small read activity due to scrub.
- Smart IO again indicates a difference in how the IOs are being serviced from these two cells.
- IO reasons show a different IO pattern on these two cells, consistent with what was observed in the other Flash Cache sections.
- Top Databases show an increase large IO latency, which results in increased IORM queue time on these two cells. These latencies directly impact the *cell smart table scan* wait event.

Reviewing the data indicates the main issue is that a flush was issued for flash cache on the two cells. Flush should not be executed on system with active database workloads, as that option stops data from being cached in the flash cache. In this case, a maintenance activity was performed to try and mitigate the lack of flash log on the two cells, but it was inadvertently done during a peak period, which resulted in the performance issues observed. To alleviate the issue, the flush operation had to be cancelled using `ALTER FLASHCACHE CANCEL FLUSH`.

Exadata Performance Data

In addition to the AWR report, there is also a wealth of performance data available on Exadata, which includes cell metrics and ExaWatcher.

Table 5 summarizes the available data and relative characteristics of each performance data category:

Performance Data

Table 5. Available performance data on Exadata.

AWR	Cell Metrics	ExaWatcher
Characteristics		
• Widely available • Usually sufficient • Integrated with existing database tools • Provides system-level view (all cells) and per-cell view • Averaged over report interval (default: 1 hour)	• Per-cell collection • Includes cumulative and per-second rates (calculated every minute) • Retention: 7 days • For more granular data and longer retention, one can use Real Time Insight[12]	• Per-cell collection • Every 5 seconds • Retention: 7 days • Charting available with GetExaWatcherResults.sh
Available Data		
• Configuration Information • OS Statistics (iostat, etc) • Cell Server Statistics • Exadata Smart Features • IO Reasons • Top Databases	• Exadata Smart Features, such as Smart Flash Cache, Smart Flash Log, IORM, Smart Scan	• OS Statistics • Cellsrvstat (Exadata smart features)

Conclusion

Automatic Workload Repository is the most widely used performance diagnostics tool for Oracle Database. AWR data in Oracle databases running on Exadata includes additional Exadata statistics. The integration of Exadata statistics in the AWR report enables significantly better and easier analysis of database performance issues than what would be possible if the databases were deployed on a generic infrastructure.

For more information, refer to *Oracle Exadata System Software User's Guide – Monitoring Exadata*

[12] See *Oracle Exadata System Software – Using Real-Time Insight*

Reference

1. Oracle Exadata System Software Users's Guide – Monitoring Exadata
2. Exadata Health and Resource Usage Monitoring
3. Exadata Health and Resource Utilization Monitoring - Exadata Database Machine KPIs
4. Exadata Health and Resource Utilization Monitoring - Adaptive Thresholds
5. Exadata Health and Resource Utilization Monitoring - System Baselining for Faster Problem Resolution
6. Oracle Enterprise Manager for Exadata Cloud - Implementation, Management, and Monitoring Best Practices
7. Enterprise Manager Oracle Exadata Database Machine Getting Started Guide

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle