

Oracle Database 19c: Quality of Service Management

Monitoring and Managing Oracle RAC Database Performance

ORACLE WHITE PAPER | FEBRUARY 2019



Table of Contents

Introduction	2
Datacenter Runtime Management Requirements	3
Runtime Management Best Practices – The Phases	4
Phase 1: Plan the Deployment	4
Phase 2: Runtime Measure the Deployment	7
Phase 3: Runtime Monitor the Deployment	10
Phase 4: Runtime Manage the Deployment	12
Baselining and Tracking Performance	15
Conclusion	16
Appendix	17

Introduction

The database is no longer the center of the universe. Such a statement would have been heretical just a short time ago. However, the introduction of the on-premise database cloud and DBaaS has altered the deployment strategy and database management requirements. It is no longer sufficient to plan for simply performance, scalability and high availability. These new deployment models must also consider consolidation, provisioning, patching and quality of service. Oracle Real Application Cluster databases whether in single node form as RAC One Node or multi-node RAC Cluster, provide the level of performance, availability and manageability to be the foundation of modern consolidated on-premise database clouds or Database-as-a-Service deployments.

Parallel Server	Real Applic	ation Clusters	On-Premise DB Cloud/DBaa
Performance	Performance	Performance	Performance
	 Scalability 	 Scalability 	 Scalability
		 High Availability 	High Availability
			Consolidation
			Provisioning
			 Quality of Service

Figure 1: Evolution of Oracle RAC Database

The ability to manage complex highly available database service deployments in real time is now a common requirement as enterprises adopt a database service-centric deployment model where multiple databases share common physical resources and are no longer silo'd on dedicated hardware. Where resource utilization has improved and IT spend optimized, runtime management complexity has increased. Oracle has addressed this in the Oracle 18c RAC release with Oracle Database Quality of Service Management (QoS) functionality to support all deployment types.

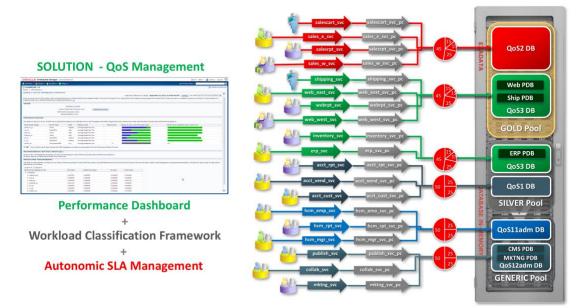


Figure 2: On-Premise Database Cloud Runtime Management

This functionality is included in the Oracle RAC and RAC One Node license and its management interface is integrated into the Enterprise Manager Cloud Control Database Plug-in. Therefore, no additional management packs are required.

Datacenter Runtime Management Requirements

The basic tenants under which Oracle Database QoS Management was developed can be distilled into three statements that must be able to be achieved in real time:

- » When resources across the datacenter are sufficient they are continuously deployed to ensure performance and availability objectives will be met.
- » When resources are insufficient to meet demand more business-critical objectives will be met at the expense of less critical ones.
- » When load conditions severely exceed capacity, resources remain available.

In order to achieve these goals, specific functionality must be built into the entire software stack to include accurate measurement of performance, resource bottleneck analysis, resource trade-off evaluation and online dynamic resource allocation.

In the end, the effectiveness of achieving the above goals is evaluated by each application's performance over time. When examining modern multi-tier applications, it should not be unexpected that most of a transaction's response time is contained in the database tier and its associated storage. This performance can be distilled at a high level into the following simple equation:

Resource Use + Resource Wait = Application Performance

It's important to realize that once an application is deployed, there is almost no ability for online management of its use of resources, as these were the responsibility of design, development, Q/A, and test teams. However,

there is the potential for the online management of the amount of time needed to wait for resources, whether these are CPU, memory, or I/O.

Fortunately, the Oracle software stack, especially the database tier, has rich resource management capabilities that have been enhanced in Oracle 18c to facilitate this when used in concert with QoS Management.

Runtime Management Best Practices – The Phases

The best practices for runtime management of an Oracle RAC-based on-premise database cloud or Databaseas-a-Service deployment may be applied in discrete phases to gain insight into the actual workloads and their use of resources as well as confidence in setting realistic service level agreements (SLAs) and the ability to manage to them. The four phases that will be discussed are as follows:

- 1. Plan the deployment
- 2. Runtime measure the deployment
- 3. Runtime monitor the deployment
- 4. Runtime manage the deployment

These phases should be implemented serially, and not combined to accelerate deployment as each captures necessary data that is used in the next phase.

Phase 1: Plan the Deployment

Planning the deployment may start at various points, but for this paper we will assume the deployment is to be a on-premise database cloud offering database services to applications each of which has an importance or criticality to the business that may vary due to calendar or events. This paper is not intended to focus on this particular task, but will introduce its elements.

Since the introduction of Oracle Database 11.2, customers have had a choice of three different cluster database deployment types – administrator-managed, policy-managed or a hybrid of the two. While it is beyond the scope of this paper to explore the pros and cons of each type, as a general rule if the databases to be deployed are 11.2 or greater, then policy-managed should be fully evaluated as it provides the most flexibility as well as deterministic high availability for on-premise database clouds. Please refer to the Appendix for additional information resources.

The next high level step is to determine the service groupings and base sizing. This will involve answering such questions as which services need to run on the same servers, which must be exclusive, or which must be dispersed as well as services that are required to be singletons.

When sizing a on-premise database cloud or DBaaS deployment, the tendency is to make use of multi-threaded CPU cores in order to increase the effective number of CPUs that each database sees in the hope that more databases can be hosted per node. The curves in Figure 3 should be observed as a warning that the level of requests per CPU is significantly reduced before response time goes to infinity and the system is in overload. It should also be noted that predicable performance is no longer achievable because the OS scheduler is now directing database workload scheduling and not the database's resource manager. This results in the CPU cost per database call rising with utilization instead of staying constant as with a single threaded core.

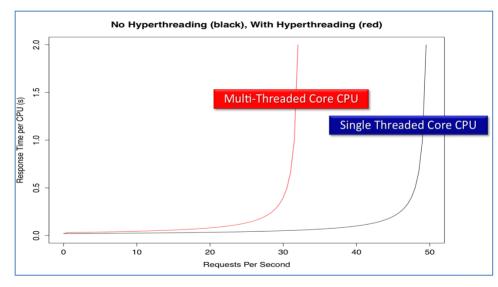


Figure 3: Consolidation Management Problem

Perhaps the most challenging step is to establish the business criticality of each service, answering such questions as:

- » Which are the services that need to be online first?
- » Which are the services that need to be the last standing?
- » Which services can I borrow resources from should a workload surge occur?
- » Which services can I shut down should a surge or failure occur?

Fortunately, these questions don't have to have static answers if a policy-managed deployment type is selected, as different business priorities can be expressed in different policies that can be switched in when appropriate. At the same time, legacy databases can coexist within their fenced servers within the Generic server pool yet still be fully supported.

Finally, services need to be group or "classified" into those that need to be tracked for performance and those that simply need to be measured. This classification may be performed by using the QoS Management Policy Editor integrated into Enterprise Manager Cloud Control to create user-defined labels or tags that group workloads for both measurement and assigning performance objectives that can be monitored or managed to as will be described in later phases.

Figure 4 shows where the QoS Management functionality can be found in Enterprise Manager Cloud Control. Note that it is accessed from the Cluster target Administration menu. This is because the scope of management is currently the entire cluster of RAC databases.

	iterprise Manager Clou	d Control 13c				Enterprise	• ©) <u>T</u> argets	1	Eavorites •	History	• 🗘 5=	tup •	२ 🌲	SYSMAN ¥
mycluster	0														node1.company.cr
Cluster 👻 Administration	• •											12	age Refresh	ed Feb 27, 2	017 4:45:51 AM GMT
A Summa Resource	Server Pools Resources Resource Types					erware Luterware 0 View Hub 0									¢
Hosts Server C Clust Policy Se			Name			Status		cidents 3	• •	Compliance Score(%)	⁹ Host				
Cluste Quality of Reconfiguration Activities Happ	Dashboard View Perform Create Policy	ance Class Qua Set	ality of Sen	vice	† †	0	0	0 0		node1.compa node2.compa	,				
Configuration Chan	nges	Edit Policy Se	ł												φ
Configuration Changes	24		View •	Target	Local target an	d Related targets		¢ Cat	egory	NI (• • •	0 🛆 🟲 0	0		
A Patch Recommend		ф	Summa	ry				Target	Sever	rity Status		calation vel	Туре		Time Since Last Update
View by O Classification Patch recomme	Target Type		No matchine Columns I	-	or problems fo	und.								Updated days	in the last 31
Patch Recommendations is unavailable. Either set the p online mode or manually u to submit a 'Refresh From	preferred My Oracle Support g pload the metadata required in My Oracle Support' job.	redentials, in	View Clus	ter Databa		:									Φ
Morecommendations to rep Job Activity Summary of jobs whose sta	ort Learn More	¢	Database	Name	Status	Incidents		Complia Score(%		Version					
			3	prod	+	0 0	0			12.2.0.1.0					

Figure 4: Creating a QoS Management Policy Set

QoS Management generates a default policy set by discovering the entire set of cluster-managed database services currently registered and creating a performance class for each one. This can be seen in Figure 5. Each Performance Class has one or more Classifiers which are the Boolean set expressions shown in the figure.

	ACLE Enterprise Manager Cloud Control 13c	
6		
Gen	eral Performance Classes Classifier Ordening Performance Policies Set Policy Review	
		Cancel Back Step 2 of 6 N
	Policy Set: Performance Classes	
forma	nce Classes are a collection of work requests for which performance objectives wish to be set.	
Add P	erformance Class Edit Performance Class Rename Performance Class Delete Performance Class	
	All Collapse All	
	Cluster / Performance Class / Classifiers	
0	∀ rwsb0508-mb1	
	∀ employee_pc	
	SERVICE_NAME INSET prod:employee	
	∀ sales_pc	
	SERVICE_NAME INSET prod:sales	
	⊘ Default_pc	
	SERVICE_NAME NOTINSET SYS\$BACKGROUND	
	∀ hompt_pc	
	SERVICE_NAME INSET prod:hcmrpt	
	V salesrpt_pc SERVICE_NAME INSET prod:salesrpt	
	Service_nvine inse i prodiseespe	
	SERVICE_NAME INSET prod:websvc	
	V webrgt_pc	
	SERVICE_NAME INSET prod:webrpt	
	V manager_pc	
	SERVICE_NAME INSET prod:manager	
Add E	erformance Class Edit Performance Class Rename Performance Class Delete Performance Class	

Figure 5: Overview of Performance Classes and Classifiers

In some cases, there may be significantly different types of workloads using the same service. Under this condition additional performance classes may be created that can differentiate the workloads using the database

session parameters if Module, Action UserName and Program. An example of differentiating browsing users of the sales service from those who are purchasing is seen in Figure 6 where salescart_pc Performance Class is being created specifying a different database user. Session Module, Action and Program can be populated in the same way.

ORAC	LE' Enterprise Manager Cloud Cont	rol 13c			SYSMAN V
Add Performa The Performance C Performance Class Classifiers	ance Class Jass Name is used to identify a collection of work reque	sts that flow through the system and against which pe	rformance objectives wish to be set.		
Delete Classifier					
Select All Select Select Service	t None	Module	Action	UserName	Program
prod:sal	es O	In Set	In Set Not In Set	SOE3 In Set	In Set Not In Set

Figure 6: Creating a Performance Class

Should a group of services have similar resource use and performance objectives where it is desired to manage them together, this can be done by adding additional classifiers to a single performance class.

Phase 2: Runtime Measure the Deployment

Once the planning phase is completed, the measurement phase may begin. This is not the same type of measurement that occurs in single application Q/A or testing but in either the production environment or a test one where all databases and these services are running as they would in production. To set up the ability to perform these actual runtime measurements, a measure-only Performance Policy is created in the same QoS Management Policy Editor. This is shown in Figure 7. What distinguishes this policy from others is that no performance objectives are specified and the Measure Only box for each performance class is checked.

DRACLE' Er	nterprise Manager Cloud Control 13	2					SYSMAN V
General Performance C	Classes Classifier Ordering Performance	e Policies Set Policy Review					
lit Policy							
	olicy Name Measure Policy						
	This is the measure policy.						
Policy I	Description						
		1					
erformance Definition f	or Policy: Measure Policy						
pecify the business ranking an	nd performance objectives for the Performance Cl	sses below. By checking "Measure Only	" a Performance Class will be monitored	but not managed by QoS M	nagement.	^	
Performance Class	Rank	Objective Type		Objective Valu	(sec) Current Value (sec)	Measure Or	nly
employee_pc	Medium 🗧	Average Response Time		0.	U0000 Unknown	2	
sales_pc	Medium ᅌ	Average Response Time		0.	Unknown		
Default_pc	Medium 🗢	Average Response Time		0.	Unknown	•	
alescart_pc	Medium ᅌ	Average Response Time		0.	Unknown		
ncmrpt_pc	Medium ᅌ	Average Response Time		0.	Unknown	•	
alesrpt_pc	Medium 🗢	Average Response Time		0.	U0000 Unknown	•	
vebsvc_pc	Medium 🗢	Average Response Time		0.	Unknown	•	
webrpt_pc	Medium 🗢	Average Response Time		0.	Unknown	•	
manager_pc	Medium 🗢	Average Response Time		0.	Unknown	2	
uthorized Actions							
elect types of resource allocati	ion actions that may be automatically implemente	d by QoS Management.				V	
Promote or Demote a Perfor	rmance Class Consumer Group. Move a CPU b	tween databases within a server pool.	Move a server between server pools				
Server Pool Directive O	Override						
ver Pool Directive Overrides of	change the server pool availability properties when	the associated policy is in effect.					
			Server Po	ol Size			
erver Pool	Min Current Value	Override	Max Current Value	Override	Current	Importance Current Value	Override
eneric	0	0	-1	-1	2	0	0 Override
ee	0	0	-1	0	0	0	0
	-		-		2	÷	

Figure 7: Creating a QoS Management - Measure-Only Policy

Once the Policy Editor wizard is completed and the policy set submitted to the QoS Management server with this measure-only policy activated, the QoS Management Dashboard is displayed as seen in Figure 8. Note that all of the performance classes are listed and the actual server pools where work is occurring are specified.

mycluster	0						ande1.c	ompany.co
Cluster 👻 Adminis	stration 👻							
	> QoS Management I							
uality of Service	Management Da	isnboard						
					rom Target September 18, 2016 3:23:06 A			
			d by workloads (performance locate resources to address I		this cluster. The status of how these performance	e classes are doing	against their objectives for a particula	ar policy is
ieneral					~ ~ ~ ~			
eneral		os Status	Easternet					
				Active Policy				
				verve Policy				
	Performance	class QoS L rendations N						
	Pacco IIII	endations in	None					
Performance Overvi		endations in	vone					
	ew			rics. OoS Managem	ent identifies a Target Performance Class and g	ovides recommenda	ations to help meet its Performance O	Diective.
his table provides an ov	ew			rics, QoS Managem Measure Only	ent identifies a Target Performance Class and pr Resource Use vs Wait Time (Last 5 sec)		ations to help meet its Performance C Satisfaction Metric (Last 5 min)	Objective.
his table provides an ov Performance Classes	ew verview of Performance	Class Metric	s.Based on the collected met					Objective.
his table provides an ov Performance Classes employee_pc	ew verview of Performance Server Pools	Class Metric	s.Based on the collected met Objective Type	Measure Only				Objective.
his table provides an ow Performance Classes employee_pc hcmrpt_pc	ew verview of Performance Server Pools prodpool	Class Metrics Rank Medium	s.Based on the collected met Objective Type Average Response Time	Measure Only				Objective.
his table provides an ov Performance Classes amployee_pc hcmrpt_pc manager_pc	ew verview of Performance of Server Pools prodpool prodpool	Class Metrici Rank Medium Medium	s.Based on the collected met Objective Type Average Response Time Average Response Time	Measure Only				Objective.
his table provides an ow Performance Classes employee_pc hcmrpt_pc manager_pc sales_pc	ew of Performance of Server Pools prodpool prodpool prodpool	Class Metrice Rank Medium Medium Medium	s.Based on the collected met Objective Type Average Response Time Average Response Time Average Response Time	Measure Only				Objective.
his table provides an ov Performance Classes amployee_pc nomrpt_pc manager_pc sales_pc sales_pc	ew verview of Performance (Server Pools prodpool prodpool prodpool prodpool	Class Metrice Rank Medium Medium Medium Medium	s.Based on the collected met Objective Type Average Response Time Average Response Time Average Response Time	Measure Only				Dibjective.
his table provides an ov Performance Classes amployee_pc cmrpt_pc ananager_pc aales_pc aales_pc aalescart_pc aalescart_pc	ew verview of Performance of prodpool prodpool prodpool prodpool prodpool	Class Metrics Rank Medium Medium Medium Medium	s.Based on the collected met Objective Type Average Response Time Average Response Time Average Response Time Average Response Time	Measure Only				Ibjective.
his table provides an ow Performance Classes amployee_pc cmrpt_pc manager_pc aalesc_pc aalescart_pc aalescart_pc aalescart_pc aalescart_pc	ew verview of Performance of Server Pools prodpool prodpool prodpool prodpool grodpool Generic	Class Metrics Rank Medium Medium Medium Medium Medium	s.Based on the collected met Objective Type Average Response Time Average Response Time Average Response Time Average Response Time Average Response Time Average Response Time	Measure Only				Ibjective.
his table provides an ov Aerformance Classes memployee_pc hemrpt_pc alles_pc allescart_pc allescart_pc allescart_pc allestpt_pc allestpt_pc	ew verview of Performance of Server Pools Server Pools prodpool prodpool prodpool prodpool grodpool Generic prodpool	Class Metrici Rank Medium Medium Medium Medium Medium Medium Medium	s.Based on the collected met Objective Type Average Response Time Average Response Time Average Response Time Average Response Time Average Response Time Average Response Time Average Response Time	Measure Only				Dispective.
	ew verview of Performance of Server Pools prodpool prodpool prodpool prodpool prodpool Generic generic	Class Metrico Rank Medium Medium Medium Medium Medium Medium Medium Medium	s.Based on the collected met Objective Type Average Response Time Average Response Time	Measure Only				Dipjective.

Figure 8: QoS Management Runtime Measure Dashboard

Examining the displayed bar graph for each performance class in Figure 9, two important metrics are displayed. The blue bar shows the actual portion of the response time that represents the use of system resources such as CPU memory and I/O. When hovered over the value displayed in seconds represents the absolute best performance that can be achieved with the deployed resource capability. The gray bar shows the actual portion of the response time that represents the wait for system resources. This time is a function of how busy the system resources are and may be altered via runtime resource management controls. When added together the two represent the actual performance which would be the minimum recommended performance objective set for this performance class given the other workloads.

Resource Use vs Wait Time (Last 5 sec)	Performance Satisfaction Metric (Last 5 min)
	n/a
	n/a
	n/a
Wait Time 0.023781 secs	n/a
	n/a
Use Time 0.065678 secs	n/a
	n/a
	n/a
	n/a
No demand	

Figure 9: Runtime Measurement Detail

Moving down the QoS Management Dashboard is a table that breaks down the resource wait time for each performance class into four categories as shown in Figure 10 – CPU, Global Cache, IO, and Other.

Performance Class/Server Pool	CPU (sec)	Global Cache (sec)	IO (sec)	Other (sec)
salescart_pc	0.091057	0.000000	0.000000	0.000911
▷ manager_pc	0.019457	0.000000	0.000000	0.000242
> websvc_pc	0.019811	0.000000	0.000000	0.000397
> employee_pc	0.018620	0.000000	0.000000	0.000162
hcmrpt_pc	0.016843	0.000000	0.000000	0.000152
▷ sales_pc	0.032019	0.000000	0.000000	0.000333
▷ salesrpt_pc	0.028000	0.000000	0.000000	0.000145
> webrpt_pc	0.017762	0.000000	0.000000	0.000077
Default_pc	0.000000	0.000000	0.000000	0.000000

Figure 10: Resource Wait Detail by Performance Class

These metrics are very useful in understanding whether there are runtime issues beyond simple resource availability with a workload. For example, if Global Cache wait time was the largest and thus the bottleneck, it is most likely that the workload doesn't scale well across more than one instance and its service should be a singleton or the data should be partitioned. If Other wait is the bottleneck, this means that there are SQL issues in the database that should be investigated via an AWR report.

Once this phase is run during all the different workload periods, the metrics will provide a baseline set of minimum performance objective values that may be used in the next phase. It will also provide data that will help determine if base sizing and resources are sufficient to meet the business objectives and whether multiple policies may be useful in meeting these.

Phase 3: Runtime Monitor the Deployment

The third phase is to actually monitor the deployment using performance objectives derived from the previous phase. This requires a different QoS Management performance policy which can be added to the policy set using the Policy Editor in EM Cloud Control. Figure 11 shows an example of such a policy. Note that this monitor policy is quite similar to the previous measure-only policy. The difference is that now actual performance objective values are entered.

JIRACLE I	Interprise Manager Cloud Control	13c					SYSMAN V
· · · · · ·		• • • • • •					
General Performance	Classes Classifier Ordering Perform	ance Policies Set Policy Review					
lit Policy							
	Policy Name Monitor Policy						
	This is the monitor policy.						
Policy	y Description	4					
erformance Definition	for Policy: Monitor Policy						
ecify the business ranking a	and performance objectives for the Performance	Classes below. By checking "Measure Onl	y" a Performance Class will be monitored	but not managed by Qos	5 Management.		
erformance Class	Rank	Objective Type		Objective \	laue (sed) Current Value ((sec) Measure On	ly
mployee_pc	Medium 📀	Average Response Time			0.08000 0.04367	2	
ales_pc	Medium 📀	Average Response Time			0.10000 0.04221	2	
Default_pc	Lowest 🗢	Average Response Time			0.00000 Unknown	2	
alescart_pc	Highest 😂	Average Response Time			0.04000 0.05006	Ø	
cmrpt_pc	Medium 😳	Average Response Time			1.00000 0.04040	•	
alesrpt_pc	Medium 😂	Average Response Time			2.00000 0.04116		
websvc_pc	High 🖸	Average Response Time			0.70000 0.04112		
webrpt_pc	Medium 😌	Average Response Time			3.00000 0.03430		
manager_pc	High 😌	Average Response Time			0.07000 0.04430		
uthorized Actions					\vee		
	ation actions that may be automatically implem						
Server Pool Directive	formance Class Consumer Group. Move a CP	U between databases within a server pool.	Move a server between server pools.				
	change the server pool availability properties v	hen the associated policy is in effect.					
			Server Po	ol Size			
	Min		Max			Importance	
rver Pool	Current Value	Override	Current Value	Override	Current	Current Value	Overr
neric	0	0	-1	-1	2	0	
	0	0	-1	0	0	0	

Figure 11: QoS Management - Monitor Policy

Once this policy is submitted and activated, the QoS Management Dashboard changes as displayed in Figure 12. Additional colored bars appear and the Performance Satisfaction Metric column also becomes relevant.

	Enterprise Ma	anager (Cloud Control 13c			•	H .	œ۰	œ۰	~	÷.	SYSMA	N ¥	
mycluster	0											node1.co	ompany.	.co
Cluster 🔻 Adminis	istration 🔻													
Cluster: mycluster Quality of Service	> QoS Management Da		1											
			Lat	est Data Collected F	From Target Septer	ber 18, 201	6 3:50:55 A	M GMT	Refresh Vi	ew Data	Real Tim	e: Manual Ref	fresh	\$
			d by workloads (performance llocate resources to address				e performanc	e classes an	e doing agai	nst their o	bjectives f	'or a particula	r policy i	is
General														
	c	oS Status	Enabled											
	Current Ac	tive Policy	fonitor Policy Change	Active Policy										
	Performance	Class QoS	Details											
	Recomm	endations 1	None											
Performance Overvi	lew													
Performance Overvi														
		Class Metric	s.Based on the collected met	rics, QoS Managem	ent identifies a Targel	t Performance	Class and p	rovides reco	mmendation	s to help n	meet its Pe	erformance Ot	bjective.	
'his table provides an ov		Class Metric Rank	s.Based on the collected met Objective Type	rrics, QoS Managemi Measure Only	ent identifies a Targel Resource Use vs V				mmendation rmance Sati				bjective.	
'his table provides an ov Performance Classes	verview of Performance												bjective.	
This table provides an ov Performance Classes salescart_pc	verview of Performance Server Pools	Rank	Objective Type	Measure Only									bjective.	
his table provides an ov Performance Classes salescart_pc manager_pc	verview of Performance Server Pools prodpool	Rank Highest	Objective Type Average Response Time	Measure Only									bjective.	
	Server Pools prodpool prodpool	Rank Highest High	Objective Type Average Response Time Average Response Time	Measure Only									bjective.	
his table provides an ov Performance Classes salescart_pc manager_pc sales_pc websvc_pc	Server Pools prodpool prodpool prodpool	Rank Highest High High	Objective Type Average Response Time Average Response Time Average Response Time	Measure Only									bjective.	
This table provides an ov Performance Classes salescart_pc manager_pc sales_pc websvc_pc employee_pc	Server Pools Server Pools prodpool prodpool prodpool prodpool	Rank Highest High High High	Objective Type Average Response Time Average Response Time Average Response Time Average Response Time	Measure Only									bjective.	
This table provides an ov Performance Classes salescart_pc manager_pc sales_pc websvc_pc employee_pc hcmrpt_pc	Server Pools Server Pools prodpool prodpool prodpool prodpool prodpool	Rank Highest High High High Medium	Objective Type Average Response Time Average Response Time Average Response Time Average Response Time Average Response Time	Measure Only									bjective.	
"his table provides an ov Performance Classes salescart_pc manager_pc sales_pc webbyc_pc employee_pc hcmrpt_pc salesrpt_pc	verview of Performance of Performance of Performance of Performance of Prodpool prodpool prodpool prodpool prodpool	Rank Highest High High High Medium Medium	Objective Type Average Response Time Average Response Time Average Response Time Average Response Time Average Response Time Average Response Time	Measure Only									bjective.	
his table provides an ov Performance Classes salescart_pc sales_pc websvc_pc employee_pc hompt_pc salesrpt_pc webrpt_pc	Server Pools prodpool prodpool prodpool prodpool prodpool prodpool prodpool	Rank Highest High High High Medium Medium	Objective Type Average Response Time Average Response Time Average Response Time Average Response Time Average Response Time Average Response Time	Measure Only									bjective.	
This table provides an ov Performance Classes salescart_pc manager_pc sales_pc	Server Pools Server Pools prodpool prodpool prodpool prodpool prodpool prodpool prodpool	Rank Highest High High High Medium Medium Medium	Objective Type Average Response Time Average Response Time Average Response Time Average Response Time Average Response Time Average Response Time Average Response Time	Measure Only									bjective.	



Figure 13 provides a close look at the right-most two columns. The Resource Use vs Wait Time column now has additional bars. Since there is now a performance objective specified in the policy that is represented by the right end of the entire bar. As the response time is the sum of the blue and gray bars, the green bar represents the additional time until the performance objective is met. This is known as the headroom and can be viewed as the amount of shareable resources that can be contributed without exceeding the performance objective. This will be clear in the next phase.





If a performance objective is specified too optimistically and is exceeded the gray bar becomes red to indicate that the wait for resources has caused the response tome of the associated performance class to exceed its performance objective. To provide an indicator of how far it has exceeded, the performance objective is displayed as a blue line with the red bar.

The Performance Satisfaction Metric (PSM) is a unique QoS Management metric as it quantifies in a normalized way how the performance class is doing against its objective. Whether a performance objective is 5ms or 5s, it

reports a value between -100% and +100% to indicate the degree response time is meeting or violating its objective. The PSM column is a binary indication as to how the performance classes have been doing against their objectives for the last 5 minutes thereby providing trending information. In this example, the continuous red indicates that the performance objective should be re-evaluated if the load is in expected normal range.

It is obviously not convenient or efficient to require constant monitoring of the QoS Management Dashboard; therefore, support is provided in the EMCC notification system for reporting negative PSMs that persist for userspecified times. Both warning and critical levels can be alerted based upon specified durations for each performance class as shown in Figure 14.

$\mathbb {V}$ QoS Management - Performance Satisfaction Metric			
\triangledown Negative PSM Duration (seconds)			
salescart_pc	>	120	300
manager_pc	>	240	480
websvc_pc	>	240	480
All others	>	300	600

Figure 14: EM Negative PSM Duration Notification Setup

This phase will likely need to be executed in an iterative fashion to establish realistic performance objectives. During this process, it may be determined that a single policy is not sufficient to capture the workload phases such as daytime, nighttime, weekends, end of quarter, etc. In that case create multiple policies which can be either switched manually or via a scheduler such as EMCC or CRON and the included qosctl command line utility.

Once this phase has been completed, the decision can be made to move to the final phase. If, for example there are sufficient resources under all demand phases, it may not be necessary to transition to the runtime management phase. However, if this on-premise database cloud is exposed to open workloads such as the Internet, the ability to respond with just-in-time intelligent resource allocation may be critical to maintain business continuity.

Phase 4: Runtime Manage the Deployment

This final phase brings resource agility into runtime management of a on-premise database cloud. Many resource management systems are, in the end, simply issue-response sets of thresholds and rules. While they may work for simple systems, they are inadequate for the complexity of an enterprise database cloud as resources cannot be provisioned on the fly. Instead resource trade-offs and agility within the existing deployment must be able to be evaluated, taking into account business priorities. This is where the expert system in the QoS Management server comes into play.

As with the other phases, this one requires a different policy. Figure 15 is an example of one such policy. It is differentiated from the previous ones in several important ways. First, it takes into account the rank of each performance class which is settable to one of 5 levels. This rank expresses how critical to the business it is for a performance class to continue to meet its objective. Starting with release 12.1.0.2, this ranking also governs the order in which the performance classes' hosting databases are started and allotted real-time LMS processes.

0		0								
General Performance Classes		e Policies Set Policy Review	ı							
it Policy										
Policy N	Manage Policy This is the manage policy.									
Policy Descri										
Policy Descri	puon									
	P	11								
erformance Definition for Po										
ecify the business ranking and pert erformance Class	formance objectives for the Performance Cl	Isses below. By checking "Measure On Objective Type	y" a Performance Class will be monitored	Performance Class will be monitored but not managed by QoS Manageme Objective Value (sec)						
mployee_pc	Medium	Average Response Time			000 0.05550	Mensure Only				
ales_pc	Medium	Average Response Time			000 0.08510					
	T									
efault_pc	lowest	Average Response Time								
alescart_pc	Highest 🧧	Average Response Time		0.04	000 0.09134					
cmrpt_pc	Medium	Average Response Time		1.00	000 0.05606					
alesrpt_pc	Medium 💽	Average Response Time		2.00	000 0.08335					
ebsvc_pc	High	Average Response Time		0.70	000 0.05647					
ebrpt_pc	Medium	Average Response Time		3.00	000 0.05554					
banager_pc	Hgh 👌	Average Response Time		0.07	000 0.05756					
uthorized Actions										
	tions that may be automatically implemente	d by QoS Management.								
Promote or Demote a Performance	e Class Consumer Group. Move a CPU b	etween databases within a server pool	Move a server between server pools	>						
Server Pool Directive Overr	ide									
ver Pool Directive Overrides change	e the server pool availability properties whe	the associated policy is in effect.								
		Server Pool Size								
erver Pool	Min Current Value	Override	Max Current Value	Override	Current	Importance Current Value Override				
eneric	Current Value	Overnde	-1	-1	2	0 0				
	Ű	•	-	-	-	-				

Figure 15: QoS Management – Manage Policy

Second, all performance classes that are to be managed have their Measure Only checkmarks removed. Any that remain checked will be considered donors should resources be required by those classes that are managed.

Third, a list of resource management type actions is offered to authorize QoS Management to autonomously that that action should it be required. This option is not recommended to be enabled until the accuracy and effectiveness of the recommendations and actions have been in production for some time.

Finally, there is the Server Pool Override Directive section which is used when multiple policies require different known base resource allocations. An example would be adding a server to a pool responsible for end of quarter reporting.

Once this type of policy is active, instead of simply being alerted to a performance class problem and viewing the extent of the issue as in the monitor phase, the QoS Management Dashboard displays a recommended action which will have a positive effect on relieving the bottleneck by trading off resources between workloads. Figure 16 shows an example of one such recommendation. In this case the performance class, salescart_pc, is experiencing a resource bottleneck in trying to get access to the CPU. The QoS Management performance model evaluation, determined that moving 15 CPU shares from the prod HCMPDB database to the SALESPDB database will provide more CPU time to the more critical workload and reduce its bottleneck thereby improving response time.

Recommendations (less than a	minute ago)			
Quality of Service Management periodi	cally provides recommendations to help a Per	formance Class meet its Performance Objective.		
Performance Class to help	salescart_pc	(approximately 12 minutes ago)		
Resource Type to help	node1:prod.cpu	(approximately 12 minutes ago)		
Recommended Action	Move 15 CPU shares from PDB prod.HCMPDB to PDB prod.SALESPDB.			
Recommendation Details	Implement			

Figure 16: QoS Management Recommendation

If greater insight is desired before clicking the Implement button, full details on the recommendation are available as shown in Figure 17. Since this is a trade-off evaluation a full view of the projected positive and negative impacts to each performance class is presented along with the improvement to the target class.

Action	Move 15 CPU shares from PDB prod.HCMPDB to PDB prod.SALESPDB.									
Estimated Time	approximately 3 minutes									
Rationale	All potential CPU share changes for each donor PDB have been analyzed.									
Evaluation	The beneficiary's PSM value is expected to change by 25.34 percentage points. The sum of all PSM values is expected to change by 6.679 percentage points. This action is a candidate for recommendation.									
Projected Results		Performance Satisfaction Metric (Last 5 min)		Average Response Time						
	Performance Class	Projected (%)	Projected Change (%)	Objective Value (sec)	Current Value (sec)	Projected Value (sec)				
	Default_pc	100	0.0	0.00000	Unknown	Unknown				
	webrpt_pc	99	-0.0	3.00000	0.03856	0.03969				
	hcmrpt_pc	95	-0.9	1.00000	0.04274	0.05147				
	sales_pc	62	12.9	0.10000	0.05138	0.03850				
	salesrpt_pc	98	0.5	2.00000	0.04182	0.03142				
	employee_pc	29	-13.7	0.08000	0.04621	0.05714				
	websvc_pc	93	-0.2	0.70000	0.04716	0.04876				
	manager_pc	12	-17.3	0.07000	0.04971	0.06179				
	salescart_pc	6	25.3	0.04000	0.04931	0.03742				



QoS Management acts as a governor on existing Oracle resource management functionality. It can adjust CPU shares within a single database to manage schema consolidated services, the number of CPU shares allocated to each PDB in a CDB multitenant database as in the example, the movement of CPUs between databases permitting management of multiple databases sharing the same servers and finally in policy-managed deployments, the ability to move servers between server pools allows for cluster consolidation.

When altering resource allocations, it's critically important to track the performance over time. Either singularly or overall, various graphical metric views can be access from the QoS Management Dashboard. Figure 18 is an overall view of the demand that the cluster is seeing and the obvious surge to the salescart_pc that caused it to violate its objective.

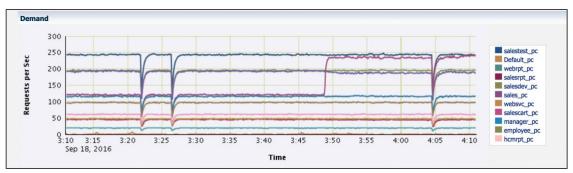


Figure 18: Overall Demand per Performance Class

Figure 19 shows a companion graph of the PSM values during the same period and how the demand surge impacted multiple performance classes. It also shows how through multiple recommendations and reallocations, performance was ultimately restored.

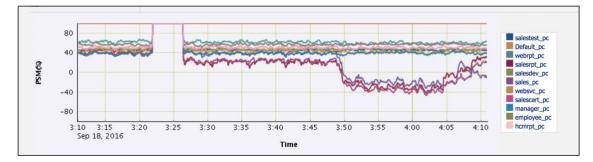


Figure 19: Overall PSM Status per Performance Class

Currently this management phase is targeted at managing workloads whose demand is independent of the response time. These open workloads are the dominant type on the Internet and are particularly difficult to provision for. By clustering resources and making them agile, workload surges can be efficiently accommodated thereby minimizing idle capacity.

Baselining and Tracking Performance

While EMCC provides performance graphs for the most current hour, it is valuable to be able to track performance over days or weeks, especially when determining a baseline set of performance objectives or whether more than one policy is required. Beginning in Oracle 19c, historical data is stored in the Grid Infrastructure Management Repository (GIMR) that resides as part of the grid infrastructure. Reports can be generated in interactive HTML format using the **qosctl -gethistory** command. An example output of the historical performance overview is shown in Figure 20.



Figure 20: Historical Performance Report - Overview

This report can be interacted from a time axis as well as Performance Class dimension. In addition to Performance Satisfaction Metric, Demand and Average Response Time graphs, the Resource Use Time and Resource Wait Time can be explored to provide increased insight into the nature of any performance bottlenecks. This data is also presented for each discrete data point as seen in Figure 21 using your mouse, as well as available for machine processing in JSON format in its data.js file located in the report output directory.

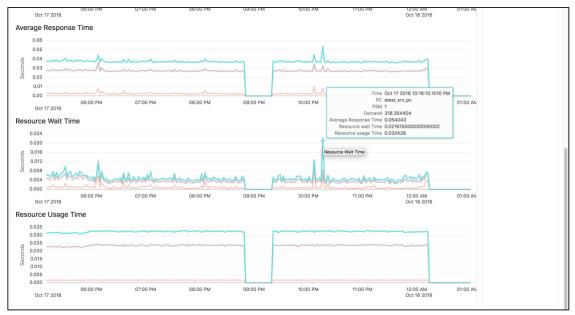


Figure 21: Historical Performance Report - Detail

Conclusion

The desire for database consolidation and a provision-on-demand DBaaS to meet the growing demand without the costs of growing datacenters brings with it new infrastructure functionality and a management paradigm that is both flexible in its resource allocation and deterministic in its operational and failure behavior. There is a learning curve with this type of change, but the ability to implement these types of deployments in managed best practice phases mitigates risk while delivering greater resource utilization and subsequent higher return on investment.

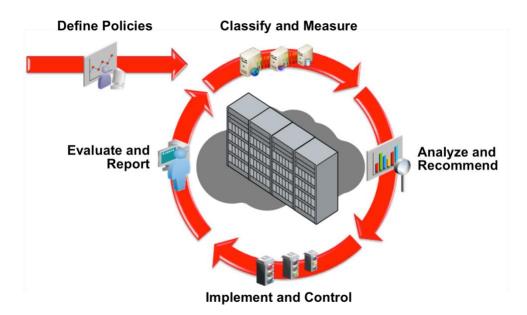


Figure 20: QoS Management in Action

Through its continuous measuring, monitoring and managing of the Oracle RAC on-premise database cloud or DBaaS deployments as illustrated in Figure 20, Oracle Database Quality of Service Management delivers the following key elements for runtime management:

- » Cluster-wide real-time dashboard view of all database workloads
- » Continuous workload health view
- » Real-time resource bottleneck identification
- » Workload-specific notifications of performance issues
- » Intelligent and targeted bottleneck resolution recommendations
- » Action audit trail and performance history

Appendix

Further information on Policy-Managed RAC databases, Clusterware policies and server pools as well as QoS Management is available from the following links:

Oracle Autonomous Health Framework 19c Documentation

Oracle QoS Management on OTN

Oracle QoS Management FAQ

Oracle Database 12c: Why and How You Should Be Using Policy-Managed Oracle RAC Databases



CONNECT WITH US

B blogs.oracle.com/oracle

facebook.com/oracle

twitter.com/oracle

oracle.com

Oracle Corporation, World Headquarters 500 Oracle Parkway Redwood Shores, CA 94065, USA Worldwide Inquiries Phone: +1.650.506.7000 Fax: +1.650.506.7200

Integrated Cloud Applications & Platform Services

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. This document is provided *for* information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0116

Oracle Database 18c: Quality of Service Management - Monitoring and Managing Oracle RAC Database Performance January 2018 Author: Mark Scardina, Oracle Corporation