An Oracle White Paper
Feb 2010

Effective resource utilization by In-Memory Parallel Execution in Oracle Real Application Clusters 11g Release 2

Server Hardware
Sponsored by

Introduction

NS Solutions Corporation (NSSOL) has been working to the business that utilized Oracle products as a strong partner for a long time since strategic partnership conclusion of Oracle Corporation and Nippon Steel Corporation (that is parent company of NSSOL) in 1991.

In continued strategic business development, In November 2006, Oracle Japan established a collaborative system with NSSOL and other grid-strategy partners and developed the advanced Oracle GRID Center[1] with the goal of building next-generation grid-based business solutions to optimize corporate system platforms. Lending its full support for the establishment of the Oracle GRID Center, NSSOL has been active in joint technical validation experiments with Oracle Japan based on its servers and storage products.

This white paper has been prepared with technological, hardware, and software support from Dell Japan Inc., Intel Corporation and CISCO Systems, Inc., both of which endorse the goals of the Oracle Grid Center. We are grateful for the strong support provided by the participating companies and engineers.

[1] Oracle GRID Center publishes many technical reports.
Oracle GRID Center (http://www.oracle.co.jp/solutions/grid_center/index.html)

Executive Overview

In-Memory Parallel Execution (In-Memory PQ) is introduced in Oracle Database 11g Release(Oracle Database 11g R2), which enhances its parallel execution features.

In the area of daily/monthly batch data processing, or large data warehouses (DWH) such as business intelligence, which needs a high-speed data processing with a large volume of data, conventional parallel execution of SQL gain performance by maximizing cpu resources of the servers and disks of storage systems.

In the recent business environment that develops and changes rapidly, it is a key factor to analyze large amount of data assets in the company faster and more effective to decide business improvement and business judgment timely.

With a remarkable achievement of the cpu performance and the capacity of the memory resources in a server today, even a single server has a enormous ability for data processing. On the other hand, with a increase of volume of a single data disk, it has become a trend to design storage systems with a relative small amount of disks, rather than setting a number of disks to improve read/write performance. And the marginal performance of the storage systems results in a possible bottleneck for the entitle systems to execute SQL.

In order to leverage all hardware resources effectively, Oracle Database 11g R2 has extended its features to simplify and accelerate SQL execution. The In-Memory PQ is one of the new features which loads a large amount of data into the database cache on the physical memory of the server(s) and execute data processing using multiple cpu resources. And this feature has a capacity to address the business challenge above.

The paper introduces performance improvement using In-Memory PQ in Oracle Database 11g R2, based on our test results of the technical validation experiments which Oracle Japan and NSSOL jointly worked out in Oracle GRID Center.

In-Memory Parallel Query

Traditional parallel query(PQ) execution, it adopted Direct Path Read to load data which bypassed the database buffer cache(buffer cache) and load directly from the disks. Of course reading data from memories has smaller latency than that from disks. It is, however, virtually very difficult to cache a large amount of data in the buffer cache because it is expensive to resource a sufficient number of high-volume memory modules and, in addition, the server is not capable to host such amount of physical memories. So if large amount of data is loaded via buffer cache like a normal fashion, it cannot leverage a benefit of the cache as each data is purged one after another by the newly loaded follow-on data parts. That's why conventional parallel execution of the Oracle Database adopts the way to load data from the disks directly.

However, with the latest hardware facilities, some middle-range servers can now host sufficient amount of physical memories to load large amount of data into a buffer cache and the computing environment turns out to be ready for leveraging in-memory parallel executing in Oracle Database 11g R2. For example, DELL PowerEdge R710 of Dell Corporation, which came into market in September 2009 with Intel® Xeon® Processor E5540 (2.53GHz) by Intel Corp., has memory capacity of maximum 72GB per one CPU socket.

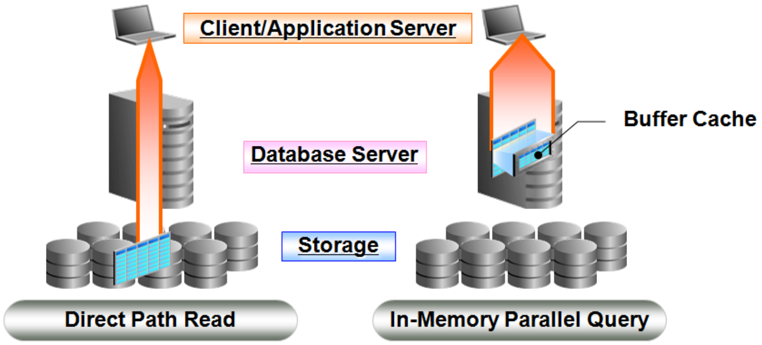


Figure 1 Direct Path Read and In-Memory PQ.

In-Memory PQ uses data of database segments (such as tables, indexes and so on) cached on the buffer cache. And it offers high-speed SQL execution environment eliminating the issue of storage performance. In case the target data is not cached, response time is less faster than direct path read method because there can be some performance degradation due to the overhead to load data into the buffer cache from disks. If PQ deals with more data than the capacity of buffer size, it automatically detect it during generating SQL execution plan and change to use direct path read instead.

In-memory PQ can be set by the initialization parameter PARALLEL_DEGREE_POLICY to "Auto" (default: MANUAL) without any change of the application. In the Oracle Real Application Clusters(Oracle RAC) environment, buffer caches of each database instances is aggregated into one virtual database buffer cache and data(database segment of the table, indexes and so on) can be distributed among the buffer, so larger amount of data can be cached than single database instance by adding nodes into Oracle RAC and expand its buffer cache.

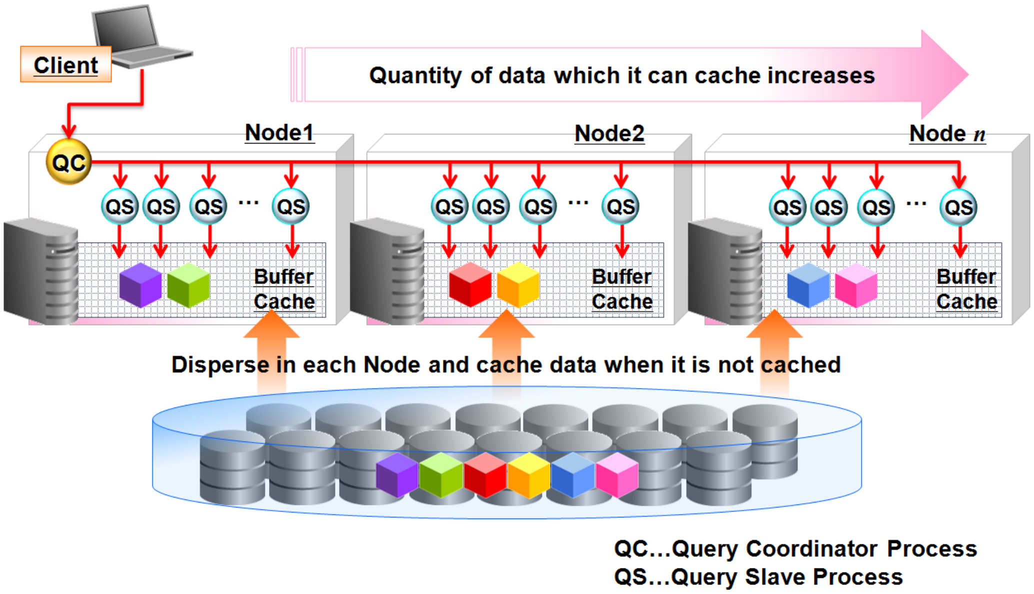

Client
Quantity of data which it can cache increases
Node1
Node2
Node *n*
QC
QS
QS
QS
…
QS
Buffer Cache
Disperse in each Node and cache data when it is not cached
QC…Query Coordinator Process
QS…Query Slave Process

Figure 2　Oracle RAC and In-Memory PQ

Oracle Real Application Clusters

Oracle Real Application Clusters is original technology by Oracle to bind multiple servers and storages so to be operable virtually as a single system, offering high-scalability, high-availability, and easy administration environment, and it keeps evolved going through releases since Oracle 9i Database R1.

Oracle RAC 11g R2 is used together with Oracle Grid Infrastructure. Oracle Infrastructure refers to the combined products of Oracle Clusterware, which administers cluster configuration, and Oracle Automatic Storage Management, which manages storages. It enables to offer wide-ranged and advanced optimization of the hardware resources than the previous release.

Oracle Infrastructure offers a virtual technology called server pool, it eliminates the bindings between the RAC systems and the physical servers, and enables the resource optimization of the hardware resources across the entire server systems..

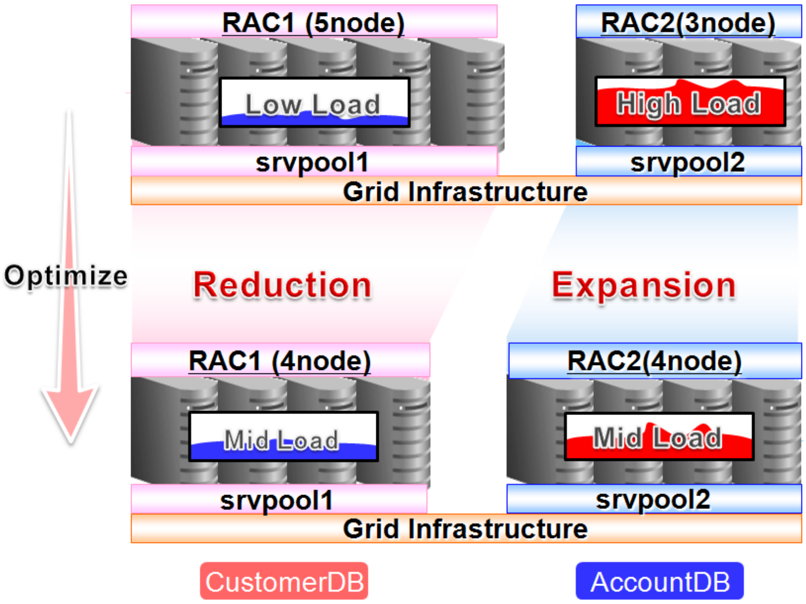


Figure 3 An example of the server resource optimization by server pool technology.

In figure 3, the number of servers for Oracle RAC ties a number of servers assigned to the server pool. The change of server pool configuration is very flexible via command line tool(srvctl commands) and web browsers(Oracle Enterprise Manager consoles), easily changing the assigned number of servers among server pools. In reducing the size of server pool, in which case a necessary numbers of instances must be stopped beforehand, changing can be done without stopping other database instances.

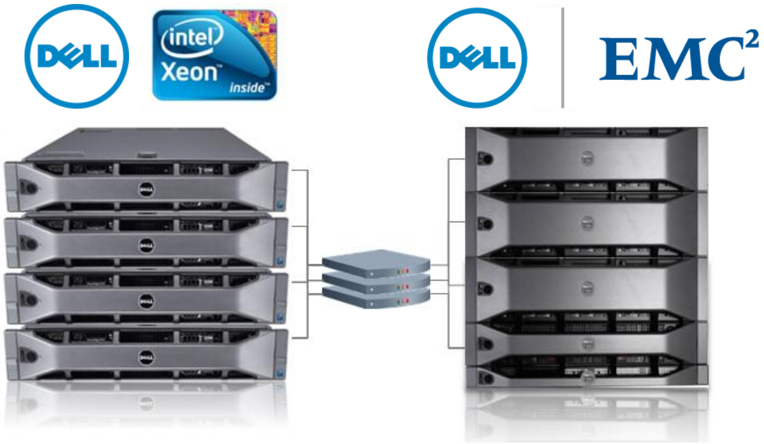
By using this server pool technology in Oracle RAC 11g R2, we can temporarily allocate necessary amount of server resources and increase the size of buffer cache in order to load all data into it and execute In-Memory PQ.

Figure 3 shows an example of user scenario: There are two server pools defined in Oracle Grid Infrastructure. Srvpool1 is assigned to CustomerDB which serves online transaction and srvpool2 for AccountDB. At night, when AccountDB starts to batch processing but there's not sufficient server resources to execute In-Memory PQ, some nodes of the srvpool1 is relocated to srvpool2 to be ready for In-Memory PQ temporarily until it completes.

Verification Environment

Described below is the verification environment established at the Oracle GRID Center for the evaluations discussed in this document.

Installed Oracle Database 11g R2 on PowerEdge R710 which has Intel® Xeon® processor E5540 (2.53GHz) and created a database on CX4-240 connected with 4 Fibre Channel (FC) of 4Gbps.



Database Server: DELL Power R710 x 4

CPU	Intel® Xeon® processor E5540 (2.53GHz) [cores：4] x 2
Memory	36GB
OS	Oracle Enterprise Linux 5 Update 3 x86-64
Oracle Database	Oracle Database 11g Enterprise Edition Release 11.2.0.1

Storage: EMC CX4-240

Controller	2Controller (Intel® Xeon® processor x 1 [2core] per Controller)
Physical Memory	8GB (4GB per Controller)
Max. Cache Memory	2.5GB (1.26GB per Controller)

In-Memory PQ execution by utilized server pools

Assuming the following system environment in Figure 5 that two nodes are assigned to srvpool1 for online transaction in the daytime. It does not satisfy the buffer cache size to use In-Memory PQ with a large amount of data when executing batch processing at night. We verify the scenario that by adding one node into a srvpool1 temporarily and increase the pool size to 3 nodes as long as it completes batch processing and executing batch processing with In-Memory PQ.

At first we create the integrated database environment where two Oracle RAC database (each two nodes) are built onto the 4-node Grid Infrastructure environment.

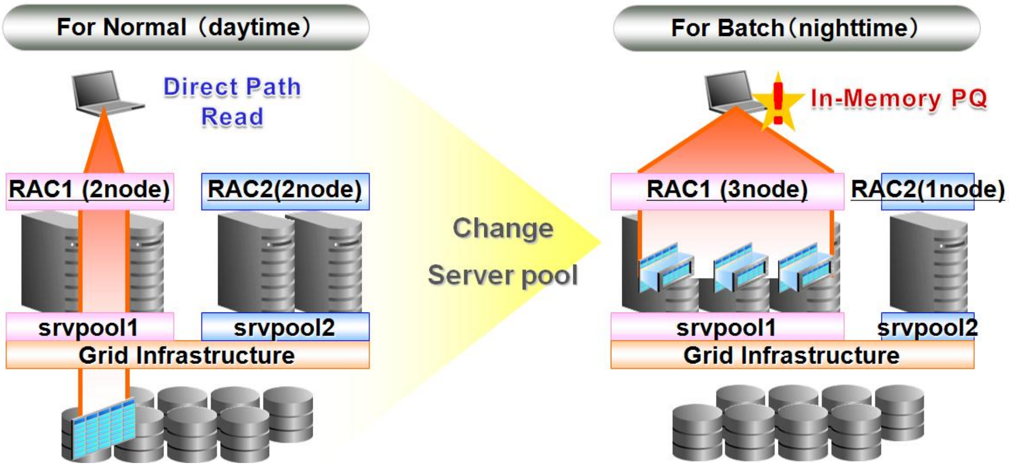


Figure 4 Executing In-Memory PQ by changing server pool configuration

To change configuration of each server pool, you can use srvctl command or Oracle Enterprise Manager. The followings is an example of srvctl command.

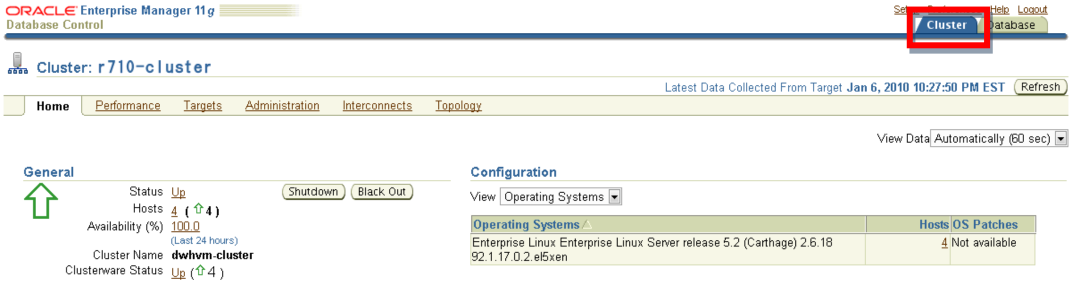
```
$ ### Reducing the nodes in srvpool2 (2 → 1)
$ srvctl stop instance -d dbname -n nodename
$ srvctl modify srvpool -g srvpool2 -u 1
$ srvctl status srvpool -g srvpool2
Server Pool Name: srvpool2
Active Servers: 1

$ ### Expanding the nodes in srvpool1 (2 → 3)
$ srvctl modify srvpool -g srvpool1 -u 3
$ srvctl status srvpool -g srvpool1
```

By publishing srvctl stop instance command, database is shut down with immediate mode by dafault. If you want to guarantee completion of transaction, you can use transactional mode (In this case you add "-o t" option).

When you use Oracle Enterprise Manager to change server pool configuration, do the follwoing:

- After login to Oracle Enterprise Manager, press the "cluster" tab on the right above.



Cluster: r710-cluster

Latest Data Collected From Target **Jan 6, 2010 10:27:50 PM EST**

View Data Automatically (60 sec)

General

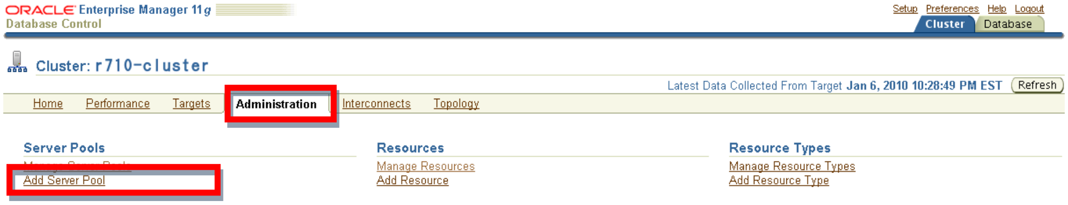
Status Up
Hosts 4 (⇧ 4)
Availability (%) 100.0
(Last 24 hours)
Cluster Name **dwhvm-cluster**
Clusterware Status Up (⇧ 4)

Configuration

View Operating Systems

Operating Systems △	Hosts	OS Patches
Enterprise Linux Enterprise Linux Server release 5.2 (Carthage) 2.6.18 92.1.17.0.2.el5xen	4	Not available

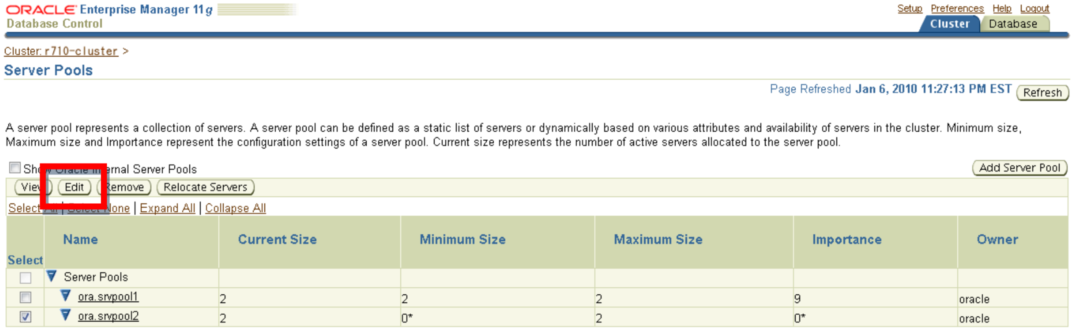
- Choose "Manage Server Pools" in "Administration" tab.



Cluster: r710-cluster

Latest Data Collected From Target **Jan 6, 2010 10:28:49 PM EST**

- Select the server pool to shrink and press "edit" button.



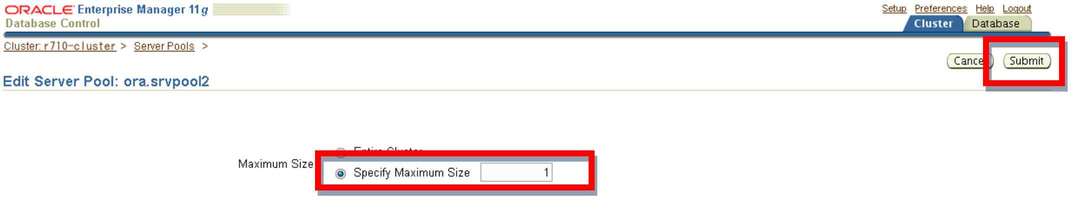
Server Pools

Page Refreshed **Jan 6, 2010 11:27:13 PM EST**

A server pool represents a collection of servers. A server pool can be defined as a static list of servers or dynamically based on various attributes and availability of servers in the cluster. Minimum size, Maximum size and Importance represent the configuration settings of a server pool. Current size represents the number of active servers allocated to the server pool.

Select	Name	Current Size	Minimum Size	Maximum Size	Importance	Owner
□	Server Pools					
□	ora.srvpool1	2	2	2	9	oracle
☑	ora.srvpool2	2	0*	2	0*	oracle

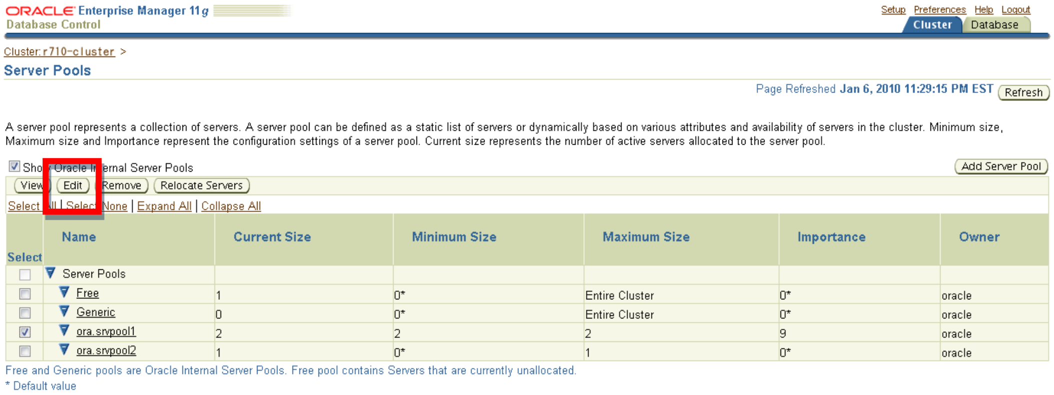
- Select "Specify Maximum Size" and set the number of size after changing(in this case set '1'). Then press "Submit" button.


ORACLE Enterprise Manager 11g
Database Control
Setup Preferences Help Logout
Cluster
Database
Cluster: r710-cluster > Server Pools >
Cancel
Submit

Edit Server Pool: ora.srvpool2

Maximum Size ○ Entire Cluster
● Specify Maximum Size 1

- Select the server pool to expand and press "edit" button


ORACLE Enterprise Manager 11g
Database Control
Setup Preferences Help Logout
Cluster
Database
Cluster: r710-cluster >
Refresh
Add Server Pool
View
Edit
Remove
Relocate Servers
Select All | Select None | Expand All | Collapse All

Server Pools

Page Refreshed Jan 6, 2010 11:29:15 PM EST

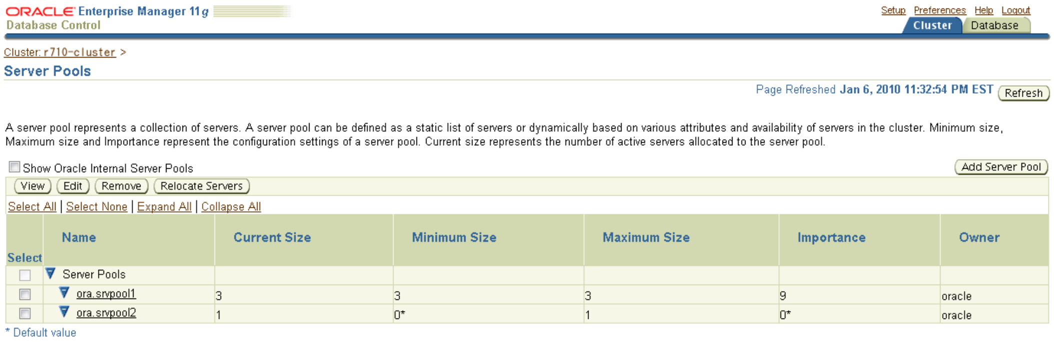
A server pool represents a collection of servers. A server pool can be defined as a static list of servers or dynamically based on various attributes and availability of servers in the cluster. Minimum size, Maximum size and Importance represent the configuration settings of a server pool. Current size represents the number of active servers allocated to the server pool.

☑ Show Oracle Internal Server Pools

Select	Name	Current Size	Minimum Size	Maximum Size	Importance	Owner
□	Server Pools					
□	Free	1	0*	Entire Cluster	0*	oracle
□	Generic	0	0*	Entire Cluster	0*	oracle
☑	ora.srvpool1	2	2	2	9	oracle
□	ora.srvpool2	1	0*	1	0*	oracle

Free and Generic pools are Oracle Internal Server Pools. Free pool contains Servers that are currently unallocated.
* Default value

- Select "Specify Maximum Size" and set the number of size after changing(in this case set '3'). Then press "Submit" button.


ORACLE Enterprise Manager 11g
Database Control
Setup Preferences Help Logout
Cluster
Database
Cluster: r710-cluster >
Refresh
Add Server Pool
View
Edit
Remove
Relocate Servers
Select All | Select None | Expand All | Collapse All

Server Pools

Page Refreshed Jan 6, 2010 11:32:54 PM EST

A server pool represents a collection of servers. A server pool can be defined as a static list of servers or dynamically based on various attributes and availability of servers in the cluster. Minimum size, Maximum size and Importance represent the configuration settings of a server pool. Current size represents the number of active servers allocated to the server pool.

□ Show Oracle Internal Server Pools

Select	Name	Current Size	Minimum Size	Maximum Size	Importance	Owner
□	Server Pools					
□	ora.srvpool1	3	3	3	9	oracle
□	ora.srvpool2	1	0*	1	0*	oracle

* Default value

Figure 6 shows the result of relative response time between executing Table Full Scan queries using Direct Read PQ with 2 nodes(left column) and In-Memory PQ with 3 nodes(right column).

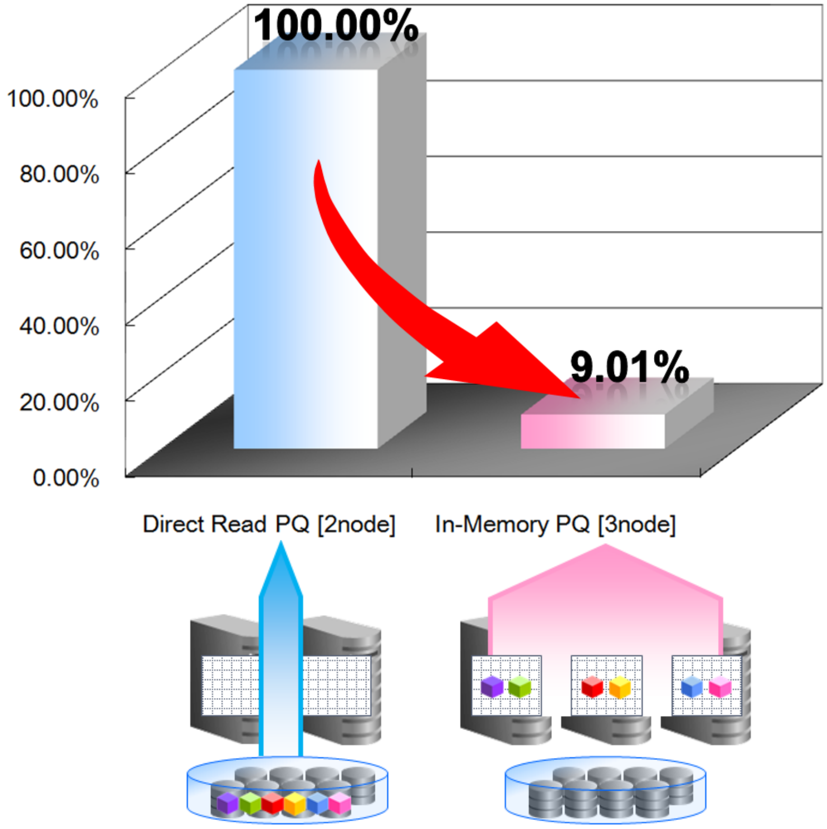


Figure 5 Effect of the In-Memory PQ.

We confirmed that query performance reaches 10 times faster than conventional Direct Read PQ by increasing the number of nodes in server pool "srvpool1" and enables In-Memory PQ. In addition, we also clear up that increasing the total amount of buffer cache of the target Oracle RAC database and every database segments for the query is cached across the buffer cache of the each database instance, In-Memory PQ is executed.

As for In-Memory PQ, it is supposed that every target data is cached on the buffer cache in advance, the result above is obtained after configuring server pool (2nodes to 3 nodes) and loading all data into buffer cache. Executing query before the target data is not loaded into the buffer cache, the response time become slower than Direct Path Read as there is an overhead for caching.

Summary

Throughout the verification test, we demonstrated the effective resource utilization by In-Memory Parallel Query on Oracle Real Application Clusters environment. Combination of server pool technologies and aggregation of multiple database systems by Oracle RAC helps easily construct high volume RAC systems (8nodes, 16nodes and more).

Server pool technology helps expand/shrink the scale of systems among databases easily, enables resource allocation in accordance with the traffic and resource requirement of each database. So it is possible to build systems with fewer server resources than taking a conventional way in which the number of each database nodes was calculated based on the individual peak load.

And with server pool technology, the size of the database can expand temporarily and increase the size of buffer cache of Oracle RAC database. It leads to be able to configure In-Memory PQ for the large amount of data segments, for which more resource allocation is usually necessary. Using In-Memory PQ instead of Direct Path Read, we can maximize the cpu utilization and minimize disk I/O of the storage which is often the cause of performance bottleneck of Parallel Query with a large amount of data.

As In-Memory PQ deals with the cached data on the buffer cache, it is suitable to process data on the same segment repeatedly(e.g. creating multiple materialized-view from one fact table), promising overwhelming performance. In this verification, we were able to prove the effectiveness of In-Memory Parallel Query on Oracle Real Application Clusters.

NS Solutions

Contributing Authors; Satoru Yagi
Ryoichi Tanaka

NS Solutions Corporation
2-20-15, Shinkawa, Chuo-ku, Tokyo

Dell Japan Inc.
Solid Square East Tower 20F
580 Horikawa-cho, Saiwai-ku, Kawasaki,
212-8589 Japan

Intel Corporation
Kokusai Bldg., 5F
3-1-1 Marunouchi, Chiyoda-Ku

Effective resource utilization using by In-Memory
Parallel Query on Oracle Real Application Clusters
Feb 2010

Author: Tsukasa Shibata
Contributing Authors: Nobuaki Tanigawa
Akira Kusakabe

Oracle Corporation Japan
Oracle Aoyama Center
2-5-8, Kita Aoyama, Minato-ku,
107-0061 Tokyo, Japan

Oracle is committed to developing practices and products that help protect the environment

0109