TechJournal

New York Oracle Users Group

Second Quarter 2011

Second Quarter General Meeting

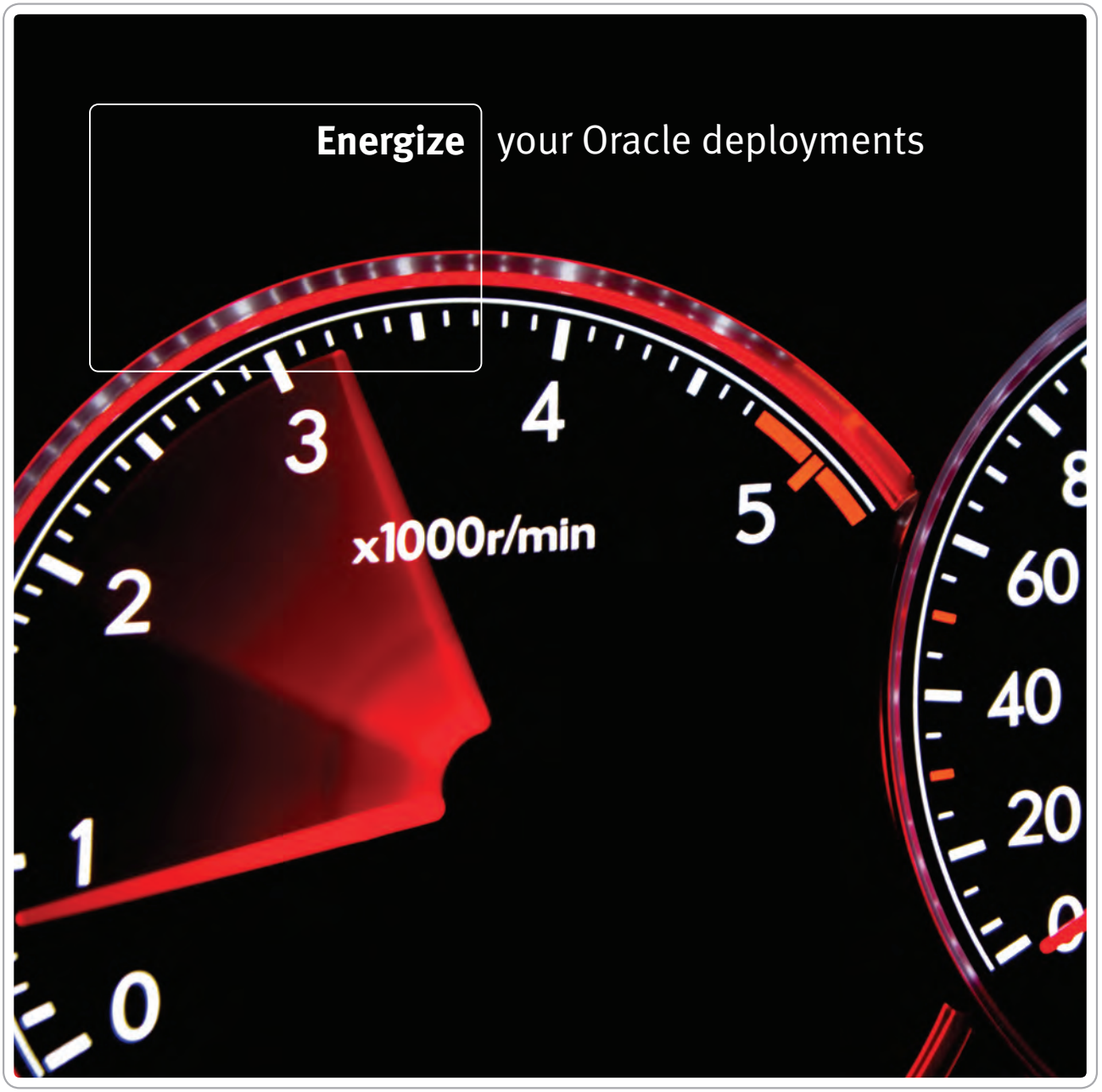
Tuesday, June 7, 2011
St. John's University – Manhattan Campus
101 Murray Street

Free for Paid 2011 Members
Don't Miss It!

In This Issue –

Presentation Papers from the December 2010 and March 2011 General Meetings
RMAN Performance Tuning Best Practices, by Timothy Chien
Just Get It Written: Deploying Applications to Weblogic Server Using JDeveloper and WLS Console, by Peter Koletzke and Duncan Mills
Oracle 11g SQL Plan Management, by Sameer Malik

EMC²
where information lives®



Reduce risk and accelerate your application deployments by drawing on the power of our Oracle® expertise.

With EMC® Proven® Solutions, your information infrastructure accelerates towards greater productivity. Learn more at www.EMC.com/oraclesolutions.

EMC²
PROVEN
SOLUTIONS

EMC², EMC, EMC Proven, and where information lives are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners. © Copyright 2010 EMC Corporation. All rights reserved. 1867

NYOUG Officers / Chairpersons

ELECTED OFFICERS - 2011

President
Michael Olin
president@nyoug.org

Vice President
Mike La Magna
vicepresident@nyoug.org

Executive Director
Caryl Lee Fisher
execdir@nyoug.org

Treasurer
Robert Edwards
treasurer@nyoug.org

Secretary
Thomas Petite
secretary@nyoug.org

CHAIRPERSONS

Chairperson / WebMaster
Thomas Petite
info@nyoug.org

Chairperson / Technical Journal Editor
Melanie Caffrey
editor@nyoug.org

Chairperson / Member Services
Robert Edwards
membership@nyoug.org

Chairperson / Speaker Coordinator
Caryl Lee Fisher
speakers@nyoug.org

Chairperson / Vendor Relations
Caryl Lee Fisher
vendorcoordinator@nyoug.org

Chairperson / DBA SIG
Simay Alpoge
dbasig@nyoug.org

Chairperson / Data Warehousing SIG
Vikas Sawhney
dwsig@nyoug.org

Chairperson / Web SIG
Coleman Leviter
websig@nyoug.org

Chairperson / Long Island SIG
Simay Alpoge
lisig@nyoug.org

Director / Strategic Planning
Carl Esposito
planning@nyoug.org

CHAIRPERSON / VENUE COORDINATOR

Michael Medved
venuecoordinator@nyoug.org

EDITORS – TECH JOURNAL

Associate Editor
Jonathan F. Miller
jonathanfmiller@earthlink.net

Contributing Editor
Arup Nanda - DBA Corner

Contributing Editor
Jeff Bernknopf - Developers Corner

ORACLE LIAISON

Kim Marie Mancusi
Kim.Marie.Mancusi@oracle.com

PRESIDENTS EMERITUS OF NYOUG

Founder / President Emeritus
Moshe Tamir

President Emeritus
Tony Ziemba

Chairman / President Emeritus
Carl Esposito
cesposi@bers.nyc.gov

President Emeritus
Dr. Paul Dorsey

Table of Contents

General Meeting – June 7, 2011 Agenda 6
Message from the President's Desk 10
RMAN Performance Tuning Best Practices 13
Avoiding Common OBIEE Mistakes: Learning from Post-Implementation Assessments 20
Unleash the Value of the Data in Your OEM Repository 35
Just Get it Written: Deploying Applications to WebLogic Server Using JDeveloper and WLS Console 48
Oracle 11g SQL Plan Management 68
APEX Collections 80
Exadata for Beginners 87
Mainframe Reliability Using Oracle RAC and Commodity Hardware 105

Legal Notice
Copyright© 2011 New York Oracle Users Group, Inc. unless otherwise indicated. All rights reserved. No part of this publication may be reprinted or reproduced without permission.

The information is provided on an "as is" basis. The authors, contributors, editors, publishers, NYOUG, Oracle Corporation shall have neither the liability nor responsibility to any person or entity with respect to any loss or damages arising from information contained in this publication or from use of programs or program segments that are included. This magazine is not a publication of Oracle Corporation nor was it produced in conjunction with Oracle Corporation.

New York Oracle Users Group, Inc.
#0208
67 Wall Street, 22nd floor
New York, NY 10005-3198
(212) 978-8890

A mandate for change is a mandate for smart.

The world is ready for change – that much is clear.

For leaders of all kinds, this moment presents a rare opportunity. Our planet is not just getting smaller and flatter. It is also becoming smarter. And that means we have the potential to change the way the world literally works.

Computational power is now being put into things we wouldn't recognize as computers – cars, appliances, cameras, roadways…even pharmaceuticals and livestock. We are interconnecting all of this through the Internet, which has come of age. And we are applying powerful new systems and sophisticated analytics to turn oceans of data into insight, knowledge and intelligence.

Consider the changes already under way.

Smart traffic systems are helping to reduce gridlock by 20%, cutting pollution and increasing ridership on public transit.

Smart food systems based on RFID technology embedded into supply chains are monitoring meat, poultry and other items from the farm to the supermarket shelf.

Smart healthcare systems are helping to lower the cost of therapy by as much as 90%.

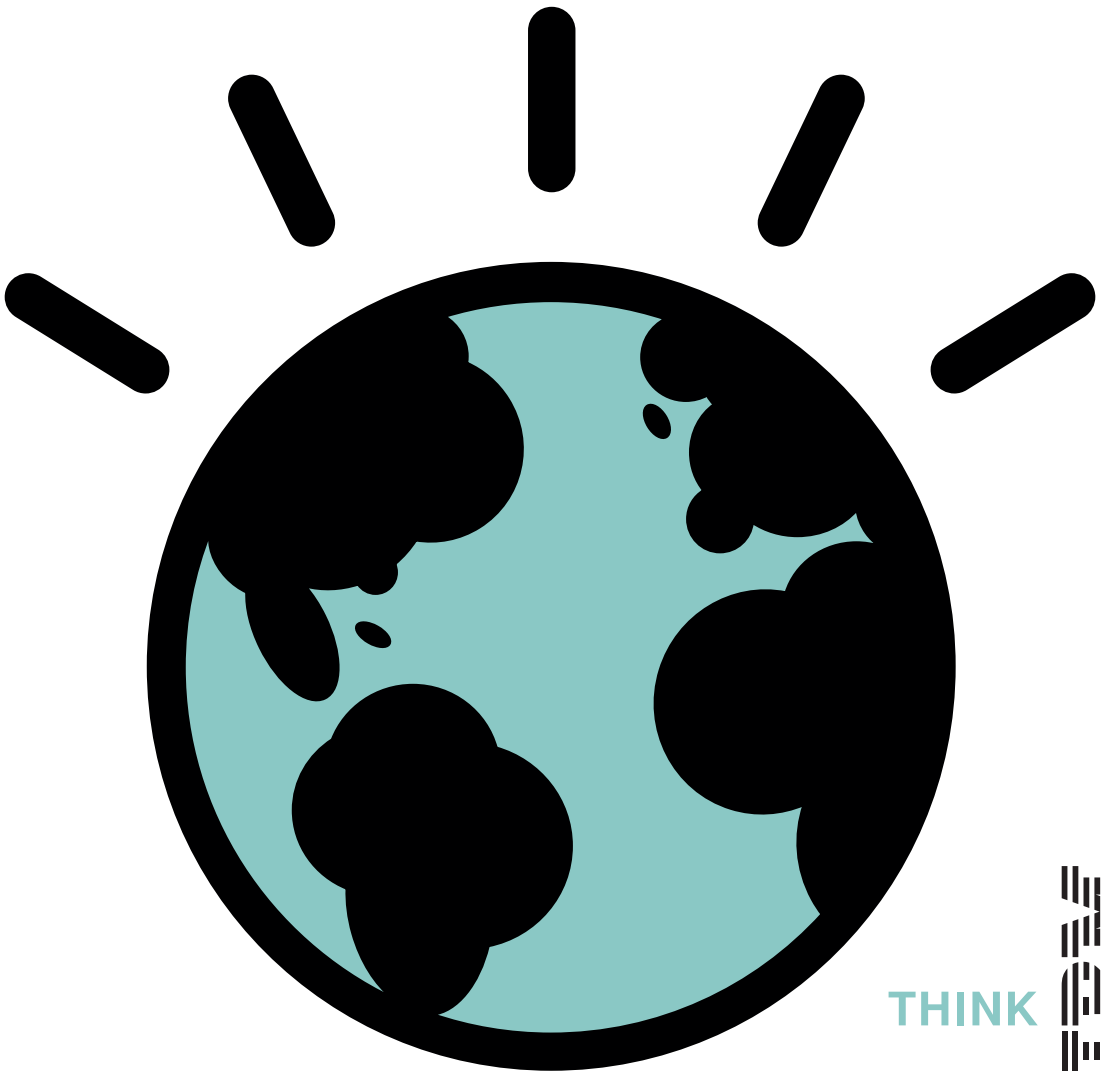
Police departments are correlating street-level information from myriad observations and devices to identify crime patterns – helping prevent crime, rather than simply punishing it.

The list is long, and the transformation is just beginning. Its benefits will be reaped not only by large enterprises, but also by mid-sized and small companies – the engines of economic growth everywhere – and by individuals and communities around the world.

Imagine how a smarter planet will transform *all* the things we seek. The ways we pursue economic growth, societal progress, environmental sustainability and cures for disease. The way we interact with each other and with the world.

The opportunity is before us, and the moment will not last forever. Will we seize it? As we look to stimulate our economies and rebuild our infrastructure, will we simply repair what's broken? Or will we prepare for a smarter future?

Join us at **ibm.com**/think



THINK

IBM, the IBM logo, ibm.com and the planet icon are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

General Meeting – June 7, 2011 Agenda

AGENDA

Time	Activity	Track/Room	Presenter
8:30-9:00	**REGISTRATION AND BREAKFAST**		
9:00-9:30	**Opening Remarks** **General Information**	(single session) Auditorium	Michael Olin NYOUG President
SESSION 1 9:30-10:30	**KEYNOTE: How Will You Build Your Next System?**	(single session) Auditorium	Dr. Paul Dorsey Dulcian, Inc. Michael Olin Systematic Solutions
10:30-10:45	**BREAK**		
SESSION 2 10:45 -11:45	**NFS Tuning for Oracle: Introducing Dtrace**	DBA Auditorium	Kyle Hailey Delphix
	Dynamic Duo: SQL and Actions	Developer Room 118	Josh Millinger Niantic Systems
SESSION 3 11:45 -12:30	**Ask the Experts Panel**	(single session) Auditorium	Michael Olin Moderator
12:30 -1:30	**LUNCH - ROOM 123**		
SESSION 4 1:30-2:30	**A Review of ADDM, ASH, and AWR in 11g**	DBA Auditorium	Earl Shaffer OracleMan Consulting
	Securing Data Today and in the Future	Developer Room 118	Ulf Mattsson Protegrity
2:30-2:45	**BREAK**		
SESSION 5 2:45-3:45	**Lessons Learned - RAC Upgrade at Verizon Wireless**	DBA Auditorium	Sridhar Doraikannu Verizon Wireless
	Looping the Loop: Different Ways of Working with Recursive Structures	Developer Room 118	Dorsey/Rosenblum Dulcian, Inc.
3:45-4:00	**BREAK**		
SESSION 6 4:00-5:00	**Web 2.0 with PL/SQL ONLY! (Product demo)**	Auditorium	Dr. Paul Dorsey Dulcian, Inc.

ABSTRACTS

9:30-10:30 AM How Will You Build Your Next System?

If you look at 10 modern systems you are likely to see 10 different architectures. Oracle is promoting the JavaEE based ADF, many users are moving to Oracle's Application Express and the rest of us are still running on Oracle Forms and wishing there was an easy alternative. Outside of Oracle everyone seems to be touting SOA and Open Source. Where does the cloud fit in? Given the fragmented world we live in with no single architecture emerging as an industry standard, how do you decide on a good development platform? Given the ever evolving environments, how do you avoid building a system that is obsolete before it is even built? This presentation will attempt to answer some of these questions.

Dr. Paul Dorsey is president of Dulcian, Inc. an Oracle consulting firm specializing in business rules and web-based application development and chief architect of Dulcian's BRIM® tool. Paul co-authored 7 Oracle Press books on JDeveloper, UML Modeling, and Oracle database tools as well as *PL/SQL For Dummies*. He is an Oracle ACE Director, SELECT Associate Editor, ODTUG Symposium chairperson, past IOUG and ODTUG volunteer of the year and an Oracle 9i Certified Master. Dr. Dorsey's submission of a Survey Generator built to collect data for The Preeclampsia Foundation was the winner of the 2007 Oracle Fusion Middleware Developer Challenge and Oracle selected him as the 2007 PL/SQL Developer of the Year. Paul can be contacted at paul_dorsey@dulcian.com

Michael Olin is president of Systematic Solutions, Inc. an independent Oracle consulting firm that he founded in 1987. For over 25 years, Michael has been designing and developing Oracle-based systems for firms in a variety of industries including: Pharmaceuticals, Health Care, Entertainment, Direct Marketing, Banking, Insurance and Financial Services. He is currently the president of the New York Oracle Users Group, having previously served as membership coordinator and vice president during his over two decades with NYOUG. Michael has presented at both local and national conferences, chaired the developer tools symposium at the East Coast Oracle conference from 1993-2000 and served as a technical reviewer for books published by Oracle Press and O'Reilly. Michael can be reached at molin@systematic-solutions.com

10:45-11:45 AM DBA TRACK: NFS Tuning for Oracle: Introducing Dtrace

NFS speeds are already on par with fiber channel and could pass up fiber channel by 2012. NFS is easier and cheaper to install, configure and maintain than fiber channel and includes robust error handling such as checksumming and retransmissions. On the other hand NFS has a reputation as being slow. This presentation explains how NFS can be optimally configured and how to analyze for NFS bottlenecks. NFS bottlenecks can be hard to identify using standard tools, but armed with the ultra powerful analysis tool Dtrace on Solaris, we can get a surprisingly rich picture NFS communication.

Kyle Hailey is the man behind the complete redesign of the Oracle Enterprise Manager 10g performance pages. His contributions shifted the screens to be graphically oriented and wait interface centric. Kyle currently works at Delphix as the performance Architect. He was also the designer of DB Optimizer, the first product in the industry to graphically describe the relationships of tables, views and subqueries of a SQL statement using Visual SQL Tuning (VST) diagrams. He has long and distinguished career with Oracle having worked at Oracle in support, porting, benchmarking, and kernel development. He has also worked at Quest, Embarcadero as well as other companies in the industry on performance tuning and optimization of Oracle. He has designed tools to improve high end performance monitoring such as direct SGA attach and interactive graphic displays of performance data. He has speaks at Hotsos, NoCOUG, RMOUG, NYCOUG, Oracle World, DB Forum as well as teaches classes around the world on Oracle performance tuning.

10:45-11:45 AM DEVELOPER TRACK: Dynamic Duo: SQL and Actions

ApEx can create highly productive applications right out of the box. However, at times a developer needs to make the application a bit more "dynamic." In this presentation, developers will learn the ins and out of using Dynamic SQL in their applications, in addition to using a new 4.0 feature called Dynamic Actions. Both of these technologies help bring applications to the next level and create a better user experience. Josh will describe the features and provide a demonstration showing examples of how to use them.

Josh Millinger was an 11 year veteran at Oracle Corporation before leaving to form Niantic Systems with Raj Mattamal. He has worked extensively with companies across verticals to help them better achieve their goals using Oracle Application Express. He presents on ApEx at various conferences nationwide.

1:30-2:30 PM DBA TRACK: A Review of ADDM, ASH, and AWR in 11g

This presentation will be a review of ADDM, ASH, and AWR in 11g. Some DBAs used it in 10g. Some did not, and have not yet used it in 11g. It has been improved in 11g, but the basic concepts remain the same. Earl will discuss ADDM - the AI-base framework of the system, its collection of info, and the application of knowledge-based systems to that info to produce important data for the DBA. The two other systems, AWR and ASH, use this framework, and their own programming, to help the DBA analyze the database with a right-now focus or a longer-term focus.

Earl Shaffer has been using Oracle products since 1988 with V5.1. Earl has been a developer, project leader, and is now a senior DBA. He has a BS and MS in Computer Science and has worked in IT for almost 30 years.

1:30-2:30 PM DEVELOPER TRACK: Securing Data Today and in the Future

With the rising cost of data security breaches and their increasing frequency, companies are starting to reevaluate how they protect their data. External and internal breaches have highlighted the need for companies to understand the flow of data within the enterprise and the need to take a more granular approach in terms of how it is secured. This session will discuss recent breaches and review different options for data protection strategies in a cloud and outsourced environment.

Ulf Mattsson created the innovative architecture of the Protegrity database security technology. Prior to starting Protegrity, he worked 20 years at IBM in software development as a consulting resource to IBM's Research organization, specializing in the areas of IT Architecture and IT Security. He is the inventor of more than 20 patents in the areas of Encryption Key Management, Policy Driven Data Encryption, Internal Threat Protection, Data Usage Control and Intrusion Prevention. He is a research member of the International Federation for Information Processing (IFIP) WG 11.3 Data and Application Security.

2:45-3:45 PM DBA TRACK: Lessons Learned – RAC Upgrade at Verizon Wireless

This presentation covers our vast experience in migrating RAC production servers from 10g to 11gr2. The main topics discussed during this session will be cluster upgrade issues and the resulting performance issues after the upgrade. Our experience of upgrading multiple RAC servers and the pitfalls and the SRs opened to solve the issues will be discussed. Sridhar will also share his experiences opening PUMA with Oracle support and how it helped.

Sridhar Doraikannu is an Oracle certified DBA with more than 20 years of IT experience including fifteen years of experience as an Oracle Database Administrator. He has a Bachelor of Science degree in Mathematics from the University of Madras, India and a Graduate Certificate in Project Management, MS and MBA degrees in Information Systems from the Stevens Institute of Technology. He has presented at Oracle OpenWorld and NYOUG.

2:45-3:45 PM DEVELOPER TRACK: Looping the Loop: Different Ways of Working with Recursive Structures

In 11gR2, Oracle introduced a new way of working with hierarchical data structures called "Recursive Subquery Factoring." This presentation compares this method to the existing CONNECT-BY implementation with regard to both performance and functionality. In addition, best practices are discussed, including whether or not hierarchical models should be implemented at all. Real-life examples are used to pinpoint key ideas and solutions.

Dr. Paul Dorsey (see keynote bio)

Michael Rosenblum is a Software Architect/Development DBA at Dulcian, Inc. where he is responsible for system tuning and application architecture. Michael supports Dulcian developers by writing complex PL/SQL routines and researching new features. He is the co-author of *PL/SQL for Dummies* (Wiley Press, 2006) and author of a number of database-related articles (IOUG Select Journal, ODTUG Tech Journal) and conference papers. Michael is a frequent presenter at various regional and national Oracle user group conferences and winner of the ODTUG Kaleidoscope 2009 Best Speaker Award. In his native Ukraine, he received the scholarship of the President of Ukraine, a Masters Degree in Information Systems, and a Diploma with Honors from the Kiev National University of Economics.

4:00-5:00 PM Vendor Demo: Web 2.0 with PL/SQL ONLY!

JavaEE is too hard to learn and too complex. Application Express is too limited unless you want to write lots of JavaScript. If your team has good PL/SQL skills, how can they build sophisticated web applications without a year or more of retraining time? The Formspider tool is easy to learn, requires only SQL and PL/SQL, and can build any sophisticated web application that you can imagine. This will be an UNSCRIPTED demo. You tell me what to build and I will build it as you watch. Come try to "stump the developer."

Dr. Paul Dorsey (see keynote bio)

Message from the President's Desk

Michael Olin

Summer, 2011

NYOUG President Speaks!

In a surprising development, attendees at the NYOUG Summer General Meeting got to hear me speak. If you're at all familiar with our meetings, and did not attend in June, you may be wondering what is so surprising about hearing from me at an NYOUG meeting. Since I became president of NYOUG, I have made opening remarks at every general meeting. I had been hosting the "Ask the Experts" session for years prior to the start of my term as president. The surprise is that after finishing my opening remarks, Dr. Paul Dorsey and I gave the keynote presentation.

To be honest, it has been quite a while since I gave a technical presentation (and I wouldn't argue with anyone who maintains that our presentation was not strictly "technical"). Paul is a familiar presence at the podium of our NYOUG meetings and he also maintains a full international schedule, giving presentations at all of the major Oracle conferences. On the other hand, I have not given a formal technical presentation in about a decade. While I am writing this column about a week before the meeting, I expect that by the time it is published, Paul and I will have delivered our keynote and given the NYOUG members in attendance something to think about.

A Couple of Old Guys Talking

Paul and I have been developing systems for a long time. My formal education in computer science came from a department that was very heavily invested in formal theory. Our faculty did not do anything as mundanely practical as designing new programming languages. They examined the theory of computation, proved running-time orders, and designed optimal algorithms for solving orthogonal range queries. In line with that focus, I wrote a RDBMS in LISP as a project for my Masters Degree. As disconnected from practical applications as my training was, I think I had it better than Paul. He was faculty. An honest-to-goodness professor, Dorsey was teaching computer science to young, impressionable undergraduates. In fact, I've worked on projects with some of his former students. While we both have spent our careers actually building systems, because of our backgrounds, we have often found ourselves discussing some less practical, more esoteric questions.

Some of these discussions, while interesting (at least to us), would easily send everyone looking to see if there was any breakfast left downstairs if we tried to present them at an NYOUG meeting. Just imagine two curmudgeons presenting a keynote on the proper treatment of NULL in set theory and whether or not it should be possible to have the empty set represented by NULL satisfy an IN predicate in SQL. The discussions that led to this keynote, however, do have very significant practical applications.

How Do You Build Systems Today?

The rate at which application architectures rise and fall has increased at an astonishing pace. Today's favored architecture may very well be a distant memory, having lost all mindshare by the time you actually get a system based on that architecture deployed. While this may not matter as much in IT shops that are so closely married to the internet that they are constantly re-architecting and rewriting their code base, much of the membership of NYOUG, and the wider community of Oracle professionals, works in a more traditional corporate environment. Budgets are planned and resources allocated months or years in advance. System development and deployment moves at a similar pace. Once a system is deployed it will likely remain in place either until the hardware or software platform has been de-supported (and the vendor will no longer accept exorbitant payments to extend the deadline). How do you select an architecture that will meet the needs of this reality? Not surprisingly, Paul and I both came to the same conclusion and that is the genesis of our keynote (hint: How often do you change your RDBMS?).

We have seen two extremes in keynote presentations at NYOUG. Rich Niemiec, who we have been fortunate enough to have at NYOUG meetings on a regular basis, usually has around 200 slides prepared for an hour-long keynote. There is no fluff, and if we ever decided to give him the entire morning, I have no doubts that he would just keep going. He always has a few extra hours of useful information that he never quite gets to cover. We have also had keynotes like that given by Wim Coekaerts (who is in charge of Linux and virtualization at Oracle). I think that Wim's keynote was not accompanied by any slides other than the Oracle logo which we projected behind him, and if you weren't in the room, there was no way to figure out what his talk was about. I'd like to think that Paul and I found a happy medium, but I think we may be much closer to Wim than Rich (or even a half-Rich). I hope it works.

Upcoming Meeting Dates

ODTUG Kaleidoscope 2011

DATES: June 26-30, 2011

LOCATION: Long Beach, CA

http://kscope11.com/registration

SAVE THE DATES:

NYOUG Fall General Meeting
DATE: Thursday September 22, 2011
LOCATION: St. John's University – 101 Murray Street

NYOUG Fall Training Session – for Members Only!
DATE: Thursday October 6, 2011
TOPIC: Performance Tuning
PRESENTER: Jonathan Lewis

Your Ad Here!

Vendors, place your advertisement in the NYOUG Tech Journal. Let our members know you want to do business with them.

Ad Options Available: Full Page – Black/White or Color
Half-Page – B/W only

Sponsorships: General Meeting – Primary and Secondary
Special Interest Group
Journal Ad only
Most sponsorship packages include color and/or black/white ads.

"SENA has raised the bar against our other vendors for quality and cost during fast paced and highly technical engagements."

Recent SENA Customer

sena

an aurionPro company

Identity • Access • Content℠ • SOA

ORACLE® PARTNERNETWORK

- *2009 ORACLE Titan Award Winner*
- *Premier Oracle Partner*

SENA Systems is an Identity, Content, SOA and Access Management-focused consulting organization which provides a full spectrum of services from strategy, architecture and custom development to implementation, project management and outsourced support. With deep industry expertise and over 10 years of experience globally, SENA provides services with pricing models for increased value and operating efficiency. We are experts in Oracle Security and Web Content Management products.

Contact SENA today to discuss how we can help with your Identity and Content Management needs.

Phone: 732-548-9408
Email: info@senasystems.com

SENA Services:

- **Identity Life Cycle Management**
- **Single Sign On and Federation**
- **Fine Grained Authorization**
- **Security Architecture and Assessment**
- **Image Processing Management**
- **Enterprise Content Management**
- **WebCenter, Portal and E2.0**
- **Offshore Center of Excellence**

sena

an aurionPro company

www.senasystems.com

East Coast:
SENA Systems
379 Thornall Street, 2nd Floor
Edison NJ 08837

West Coast:
SENA Systems
2410 Camino Ramon #339
San Ramon, CA 94583

Australia headquarters:
SENA Systems
Level 2, 215 Spring Street
Melbourne, VIC, 3000

RMAN Performance Tuning Best Practices

Timothy Chien, Oracle Corporation

Introduction

A complete high availability and disaster recovery strategy requires dependable data backup, restore, and recovery procedures. Oracle Recovery Manager (RMAN) provides a comprehensive foundation for efficiently backing up and recovering the Oracle database. It is designed to work intimately with the server, providing block-level corruption detection during backup and restore. RMAN optimizes performance and space consumption during backup with file multiplexing and backup set compression, and integrates with Oracle Secure Backup, as well as third party media management products, for tape backup.

RMAN takes care of all underlying database procedures before and after backup or restore, freeing dependency on OS and SQL*Plus scripts. It provides a common interface, via command line and Enterprise Manager, for backup tasks across different host operating systems and offers features not available through user-managed methods, such as parallelization of backup/restore data streams, backup files retention policy, and detailed history of all backups.

As database sizes continue to grow unabated, RMAN performance tuning is more critical than ever to satisfy required backup-and-recovery windows. This paper presents an inside look at RMAN backup, restore, and recovery operations, including a step-by-step tuning methodology to diagnose performance bottlenecks and implement the database and system changes to achieve the desired performance characteristics. This paper assumes the reader has an understanding of RMAN backup and recovery concepts and features. All content in this paper assumes Oracle Database 11*g* Release 2, unless otherwise noted.

RMAN Performance Tuning Primer

An RMAN backup or restore job can be divided into separate phases or components. The slowest of these phases in any RMAN job is called the bottleneck.

The goal of RMAN tuning is to identify the bottlenecks for a given job and use RMAN commands, initialization parameters, or adjustments to physical media to improve performance.

Before diving into a hands-on diagnosis of performance bottlenecks, it is helpful to understand the components involved in the overall backup data flow and a set of tuning principles based on this flow.

Backup Data Flow

The work of a backup is performed by one or more channels. A *channel* represents a stream of bytes to a storage device. A byte stream passes from the input buffers in memory through the CPU to the output buffers. From there, the backup data is written to the storage device. A diagram of the process is shown below.

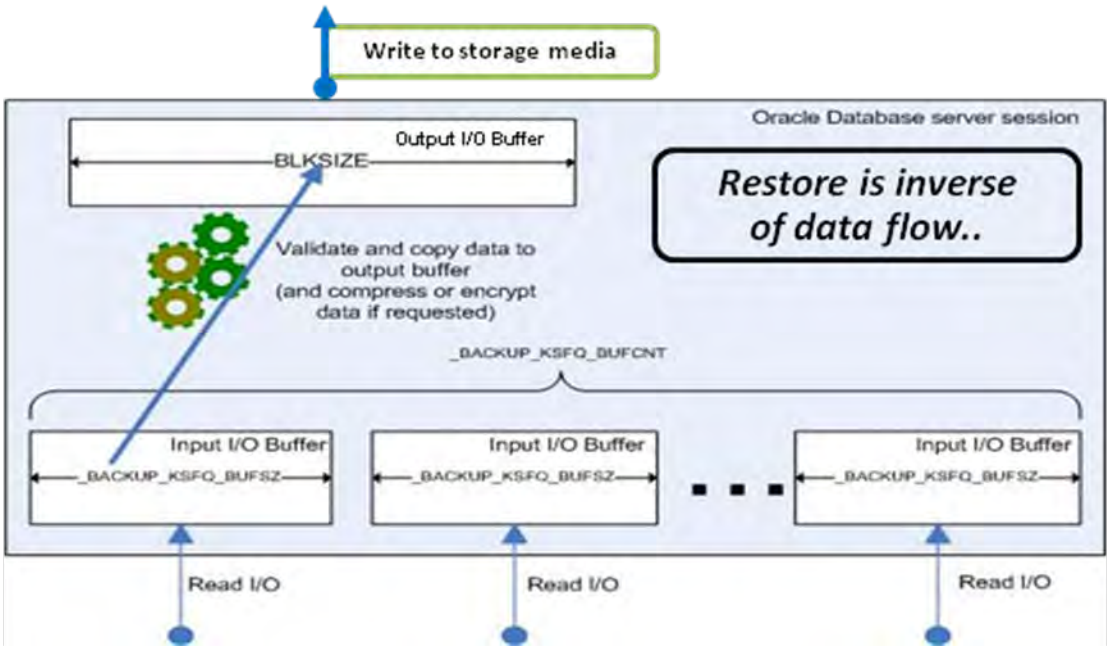


Figure 1. RMAN Backup and Restore Data Flow

The work of each channel, whether of type `DISK` or System Backup Tape (`SBT`), is subdivided into the following distinct phases:

- Read Phase
  - A channel reads blocks from files on production disk into input I/O buffers.

- Copy Phase
  - A channel copies blocks from input buffers to output buffers and performs additional processing on the blocks, including validation and optional compression and/or encryption.

- Write Phase
  - A channel writes the blocks from output buffers to storage media. In the case of SBT, the media manager handles the writes to the tape device.

Each of these phases will now be discussed with respect to a set of performance tuning principles.

Performance Tuning Principles

Read Phase

The objective of tuning the read phase is to maximize RMAN read I/O off production disk.

Principle #1: Determine the maximum input disk, output device, and network throughput

Typically, production disk will be the highest performing component in the backup data flow. Thus, the theoretical maximum backup throughput is determined by the production disk. It is helpful to determine the baseline production disk performance as a starting point in the tuning analysis.
The Oracle Orion Calibration tool can be used to simulate various database workloads (e.g. OLTP, DSS) and measure performance metrics, such as IOPS, MBPS, and I/O latency. More details on using Oracle Orion can be found in the Oracle Performance Tuning Guide

(http://download.oracle.com/docs/cd/E11882_01/server.112/e16638/iodesign.htm#CACJEEDI). If backups are taken to disk, it is also helpful to determine the baseline backup disk performance.
Secondly, the network throughput should be evaluated between all connection points. For example, between the database host and backup disk and/or media management environment. TCP/IP performance measurement tools, such as `qperf`, are helpful for this task. The media management environment (e.g. media server) network throughput to the tape system should also be evaluated.
Finally, the tape system throughput should be evaluated. In many cases, the number of available tape drives limits overall backup performance. Tools such as the `DD` operating system command can be utilized to measure tape drive throughput. Note that advertised tape throughput specifications are theoretical and do not take into account other operational factors, such as load/unload time, rewind/seek time, etc. – remember to always test with actual workloads.

Principle #2: Configure disk subsystem for optimal performance

Once the raw disk performance is determined, ensure that Oracle I/O can optimally and most efficiently utilize the underlying disk subsystem. Automatic Storage Management (ASM) is the recommended method to configure Oracle database storage. More details on deploying and configuring ASM can be found in the Automatic Storage Management Adminstrator's Guide (http://download.oracle.com/docs/cd/E11882_01/server.112/e16102/toc.htm).
To ensure that production workload I/O does not conflict with backup workload I/O, it is advised to create separate disk groups, one for data files (e.g. DATA) and one for backups (e.g. FRA), and that physical disks are also separated between the disk groups. If the physical disks cannot be separated, then it is advised to configure the outer sectors of the physical disks for the DATA disk group, which provides excellent random I/O performance for data files, and configure the inner sectors for the FRA disk group, which still provides good sequential I/O performance for writing backups.
If ASM is not deployed, then all data files should be striped across available disks with a 1 MB stripe size, in accordance with the Oracle SAME (Stripe and Mirror Everything) methodology.

Principle #3: Tune RMAN to fully utilize disk subsystem and tape

The next task is to configure RMAN to fully consume available disk and tape I/O.
Ideally, the underlying operating system already supports asynchronous I/O and RMAN automatically takes advantage of this. However, if the operating system does not, the database can simulate asynchronous I/O with `DBWR_IO_SLAVES`. By setting this parameter, 4 slave processes will be allocated per session. This is especially important for backups to disk. By default, backups to tape are a synchronous process, i.e. SBT write operation must be acknowledged before the copy-to-output buffer process continues. Therefore, it is recommended to set `BACKUP_TAPE_IO_SLAVES` so that SBT write operations can proceed asynchronously with respect to the copy-to-output buffer process. The only exception is if the media management product states otherwise.
The next task is to properly allocate channels for optimal backup parallelism. For disk backups, this means allocating as many channels as can be handled by the system – each channel will use memory, CPU, and I/O just like a standard server process, so monitor these resources carefully. For image copies, where RMAN creates a one-to-one backup copy of each datafile, one channel can only process one data file at a time, so parallelism becomes a key factor. Typically, there will be a point at which adding more channels will not result in much more overall performance gains.
For tape backups, one channel should be allocated per tape drive. This recommendation stems from the fact that allocating a higher number of channels than tape drives will result in tape-side file multiplexing where two or more backup files are interleaved on the same tape. Because of this behavior, restore time will degrade due to channel contention, i.e. one channel will attempt to read its backup file from tape while another attemps to read its backup file from the same tape. However, in many cases, increasing the number of channels increases overall backup throughput. Rather than suffer the consequence of degraded restore time, it is wiser to determine where the bottleneck actually lies. A simple test is to run `BACKUP VALIDATE` with the higher number of channels and note the time duration of this operation – this is the read phase time (A). Repeat the test with the same number of channels as tape drives and note the read phase time (B). If A is less than B, then the bottleneck most likely is in the read phase itself and must be further tuned. This will be discussed in the next section "Read Phase Tuning".

If A equals B (i.e. read phase times are about the same), then the bottleneck is more likely in the write phase, including the media management environment. Specific recommendations for this case are in "Diagnosing Performance Bottlenecks – Copy/Write Phase".

Read Phase Tuning

Assuming the production disk has been optimally configured and its performance has been baselined, the other major factors in tuning the read phase are file multiplexing and the resulting memory buffer usage during the backup.
RMAN multiplexing is defined as the maximum number of files read by one channel, at any time, during backup.
Multiplexing is typically defined by **`MAXOPENFILES`** (default: 8), a parameter of **`ALLOCATE CHANNEL`** or **`CONFIGURE CHANNEL`**, as shown below.

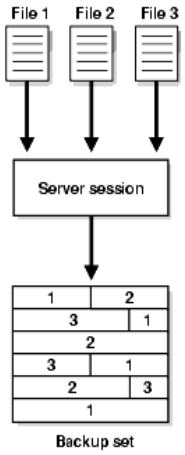


Figure 2. RMAN Backup with `MAXOPENFILES=3`

More importantly, the level of multiplexing dictates the number and size of input memory buffers utilized during the backup, as shown below:

- **MULTIPLEXING ≤ 4**
  - Each buffer = 1MB, total buffer size for channel is up to 16MB
- **4 < MULTIPLEXING ≤ 8**
  - Each buffer = 512KB, total buffer size for channel is up to 16MB. Number of buffers per file will depend on number of files.
- **MULTIPLEXING > 8**
  - Each buffer = 128KB, 4 buffers per file, so each file will have 512KB buffer

The actual number and size of buffers used during a backup can be found in `V$BACKUP_ASYNC_IO` (assuming asynchronous I/O or if not, that disk slaves are enabled). All buffers are allocated from the PGA, unless disk or tape I/O slaves are enabled. If slaves are enabled, then buffers are allocated from the SGA or `LARGE_POOL` (if set).
For ASM and striped storage, it is recommended to set `MAXOPENFILES=1`, to get the most number and largest sized buffers during the backup, i.e. 16 buffers with each sized 1 MB. Since the data files are striped across all disks (i.e. data blocks are spread evenly across disks), issuing as many input buffers as possible will allow the overall disk I/O utilization to be maximized.
For non-striped storage, it is recommended to use the default setting (`MAXOPENFILES=8`, i.e. 4 buffers per file with each sized 512 KB). Since each file resides entirely on one disk, this will allow a smaller number of buffers to work on each disk, to more effectively spread out I/O usage.

Once the appropriate number of channels and `MAXOPENFILES` has been set, re-run `BACKUP VALIDATE` and monitor the `EFFECTIVE_BYTES_PER_SECOND (EBPS)` column. If `EPBS` is still less than expected baseline disk performance and there is still available CPU and memory, then the number and size of buffers can be manually increased.
With Oracle Database 11*g* Release 11.1.0.7 or lower versions, increase **`_BACKUP_KSFQ_BUFCNT`** (default 16) = # of input disks. This will set the number of input buffers per file allocated. Carefully monitor memory usage as different settings are tried, e.g. setting this parameter to 500 for 500 input disks may exceed tolerable memory consumption.
Set **`_BACKUP_KSFQ_BUFSZ`** (default 1048576 or 1 MB) = stripe size (in bytes). This parameter should only be adjusted if the ASM allocation unit or stripe size is not 1 MB.
With Oracle Database 11*g* Release 2, the parameters are renamed **`_BACKUP_FILE_BUFCNT`** and **`_BACKUP_FILE_BUFSZ`.** No changes are needed for ASM starting with 11.2, as all disk buffers are automatically sized for optimal performance.
Note that restore performance issues may also be alleviated, as these settings will also increase the output buffers used during restore.
Refer to **Support Note 1072545.1** for more details.

Copy Phase

During the copy phase, the input buffer blocks are read and validated for physical corruptions (default) and logical corruptions (optional). The blocks are then compressed and/or encrypted, if requested, and copied to the output buffers, as shown below:

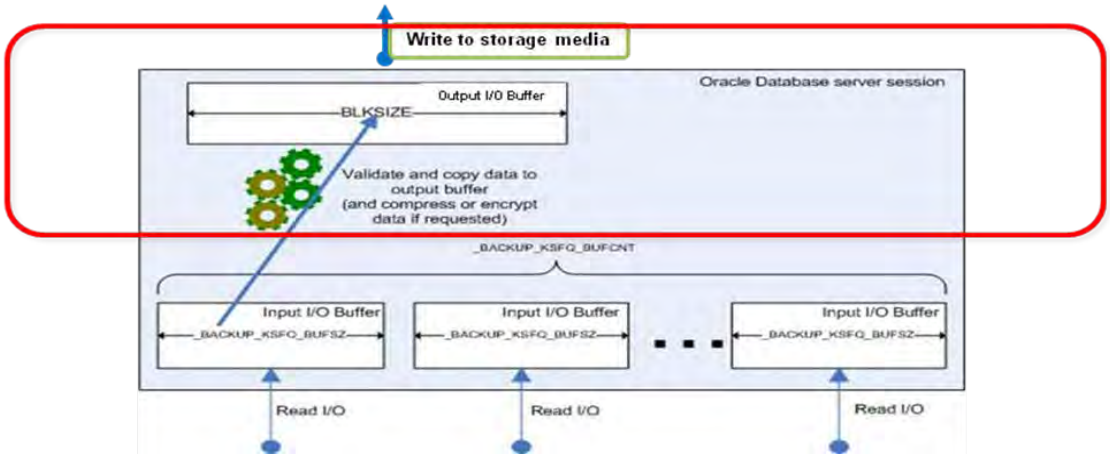


Figure 3. RMAN Backup Data Flow – Copy and Write Phases Highlighted

Generally, the in-memory copy utilizes minimal resources. More often, the copy phase becomes a bottleneck due to high CPU utilization when compression and/or encryption are requested. The goal of tuning the copy phase is to minimize the resource impact due to compression and encryption.

Principle #4: Tune RMAN backup compression and encryption to minimize resource contention

The basic tuning options are to either (a) increase CPU resources, (b) use a faster compression (`LOW/MEDIUM`) or encryption setting (`AES128`), or (c) allocate more channels for number of CPUs/cores, since one channel only works on one CPU.
If Transparent Data Encryption (TDE) is utilized in the database and encrypted backups are specified, CPU utilization can be affected, depending on whether backup compression is used and whether TDE column or tablespace encryption is used, as shown below.

- TDE column encryption
  - If encrypted backups are specified, data is double encrypted (i.e. encrypted columns treated as if they were not

encrypted) – thus, CPU utilization is higher.

- TDE tablespace encryption
  - If compressed & encrypted backups are specified, TDE-encrypted tablespaces are decrypted, compressed, then re-encrypted – thus, CPU utilization is higher.
  - If encrypted backups are specified, TDE-encrypted blocks pass through the backup unchanged – thus, no change in CPU utilization.

Write Phase

The goal of tuning the write phase is to ensure that the output buffers are written out as quickly as possible to the appropriate backup device. Disk backups are written out by the RMAN channels themselves while tape backups are written out by the media management processes.

Principle #5: Tune RMAN output buffers to increase write performance

By default, 4 output buffers are allocated per channel, whether `DISK` or `SBT`. The default buffer size is 1 MB (for disk) and 256 KB (for SBT). For SBT, the buffer size should be set to the media management client buffer size via the `BLKSIZE` channel parameter.

In **`V$BACKUP_ASYNC_IO`**, where type=`'output'`, monitor the `EFFECTIVE_BYTES_PER_SECOND (EBPS)` to see if that value matches the backup disk speed. If not and assuming the read phase has been properly tuned (as discussed previously), the output buffer count & size can be manually adjusted per Support Note 1072545.1 - RMAN Performance Tuning Using Buffer Memory Parameters. Note that with Oracle Database 11g Release 2 & ASM as backup disk, output buffers are automatically sized for optimal performance.

Diagnosing Performance Bottlenecks

With an understanding of the RMAN backup data flow and its components, in addition to key tuning principles, we can now apply this knowledge to a step-by-step, top-down performance bottleneck diagnosis and tuning methodology.

They first step is to query **`EFFECTIVE_BYTES_PER_SECOND`** column (`EBPS`) for the '**`AGGREGATE`**'row in **`V$BACKUP_ASYNC_IO`**. If `EBPS` does not match the expected backup disk or tape throughput, then determine the read phase time by running the same backup command with the `VALIDATE` option.

Read Phase Bottlenecks

If the observed read phase time is approximately the same as the overall backup time, then the read phase is most likely the bottleneck and thus, this phase should be tuned appropriately, as previously discussed in the "Read Phase" section. In addition to the guidance given for RMAN multiplexing and buffer usage parameters, there may be a possibility that a few files have slower access patterns than the rest. This can be determined by finding the data file with an above average **`(LONG_WAITS/IO_COUNT)`** ratio from **`V$BACKUP_ASYNC_IO`.** If production disk is ASM, consider adding disk spindles or re-balancing disks. If non-striped disk, move the 'slow' file to a new disk and re-run `BACKUP VALIDATE`.

Copy/Write Phase Bottlenecks

If the observed read phase time is much less than the overall backup time, then look to the copy or write phase as the most likely bottleneck.

If compression and/or encryption is used, refer to the previous section "Principle #4: Tune RMAN compression and encryption to minimize resource contention".

For disk backup, look to increasing the number of output buffers as prescribed in **Support Note 1072545.1**

For tape backup, check the media management (MML) settings, TCP/IP buffer size, media management client/server buffer size, client/socket timeout, media server hardware, connectivity from media server to tape (e.g. SAN). Tape-level compression should also be evaluated – note that RMAN backup compression and tape compression should not be used together as resulting savings will be minimal for the resource impact during processing.

Recovery Performance Tuning

While backup and restore operations are where the majority of tuning analysis is spent, recovery performance should also be evaluated. The following guidelines can help increase recovery performance:

- Minimize archive log application by using incremental backups
- Use block media recovery for isolated block corruptions
- Keep adequate number of archived logs on disk
- Tune database for I/O, DBWR performance, CPU utilization
- Refer to MAA Media Recovery Best Practices paper
    - Active Data Guard 11g Best Practices (http://www.oracle.com/technetwork/database/features/availability/maa-wp-11gr1-activedataguard-1-128199.pdf - includes best practices for Redo Apply)

Conclusion

The key question to answer when starting an RMAN performance tuning exercise is: "Where is my bottleneck?" Then, endeavor to eliminate it (or shift it to a different part of the system) to achieve the desired performance.

Each backup phase can be analyzed to determine its maximum throughput/ceiling. The most critical phase is the read phase, since this is the point at which the data is read from the production disk – the read phase should consume most or all available disk I/O at the expense of additional memory usage if needed, so that RMAN buffers are continuously being filled and there is no wait in reading data from disk. RMAN multiplexing and memory buffer parameters are used to tune this phase.

Once the read phase has been tuned, the copy phase should only be of concern if RMAN backup compression and/or encryption are used. This is due to the higher-than-normal CPU consumption for these operations. Thus, the available CPU becomes the bottleneck for the backup processing. Adding CPUs and allocating more channels, and/or using higher performant compression settings are recommended for this case. If backup data only needs to be encrypted, consider using TDE tablespace encryption, which will pass through RMAN unchanged, and thus consume no additional CPU.

Finally, the write phase should be investigated, if overall backup performance still does not improve after tuning read and copy phases. For disk backups, ensure that the backup storage throughput is as expected without RMAN involved. The output buffer count and size can be increased, if needed.

For tape backups, check with media management vendor for tuning recommendations. Typically, this involves ensuring that buffer sizes, network connectivity, media server hardware, etc. are all performing as expected to support the incoming RMAN backup data stream. The DBA should have some knowledge of the media management environment – at least enough to be dangerous.

Remember – at the end of the day, what ultimately matters is not the individual backup environment components, but the end-to-end backup performance that can be achieved!

Avoiding Common OBIEE Mistakes: Learning from Post-Implementation Assessments

Brian Halloran, Archetype Consulting

Introduction

First impressions are everything. When hearing about Business Intelligence for the first time, people awe at the simple concept of transforming raw data into better business decisions. Often, the effort to incorporate Business Intelligence and the maintenance responsibility that comes along with it goes overlooked. Successful BI is delivered through strict, best practice adherence to the Methodologies, Processes, and Architectures & Technologies that enable it.

Specific to OBIEE there are four main transformation components that exist: Data Storage, Metadata Definitions, Dashboards & Reports and User Analysis. This article follows each of those transformation components with a scenario of how things can go wrong. Further, this article will explain a solution and how the problem could've been avoided entirely if best practices were followed from the beginning.

*In every example in this article the name of the company has been altered to protect its privacy.

Data Storage – Wayne Enterprises Case Study

An Overview

Wayne Enterprises is a well-respected manufacturing company that produces consumer appliances. The diagram below details a consumer's appliance registration process.

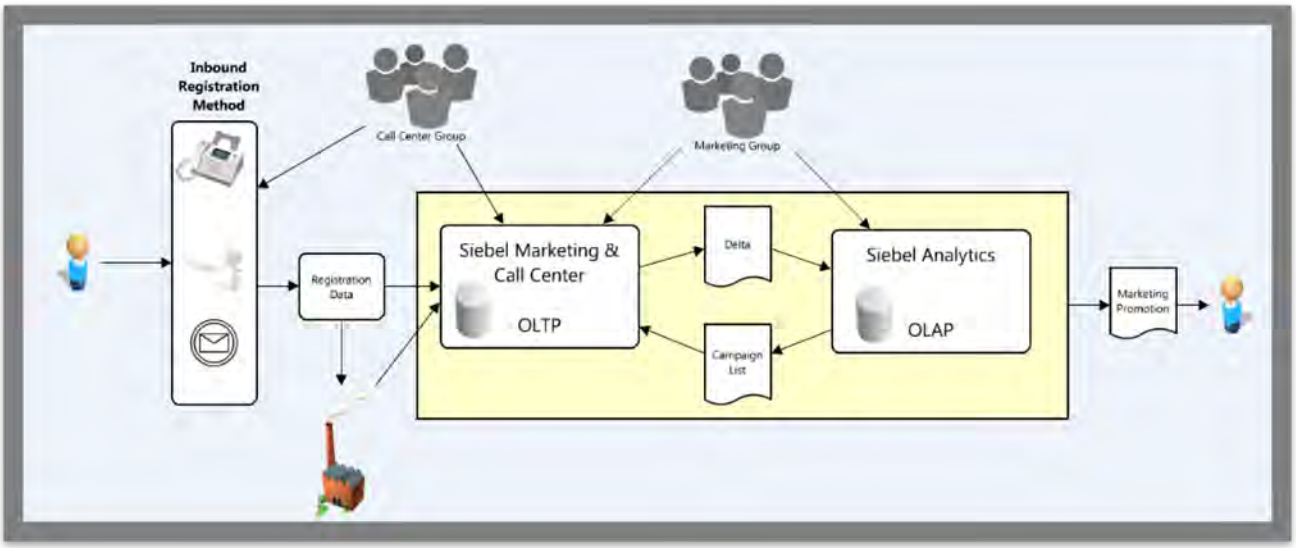


Figure 1. Wayne Enterprises Consumer Appliance Registration Process

1. After purchasing an appliance the consumer can have it registered through a call center, website or direct mail.
2. Once Wayne Enterprises has received the appliance registration it sends the consumer information off to a 3rd party vendor for cleansing, merging and matching.
3. Wayne Enterprises receives the consumer's registration information with 3rd party IDs for the consumer, home address and company (if available) and loads it into a Siebel application for both the Call Center and the Marketing teams.

4. The registration information, along with any delta data is refreshed into the analytics warehouse for reporting and marketing segmentation
5. Once loaded, the marketing group can now apply segmentation logic off the registration data to create marketing list that outline consumers to be sent a registration notification.
6. Consumer attributes are added to the marketing lists and exported back to the Siebel transactional application where it is loaded and made visible for the Call Center and Marketing groups.
7. The last step is to export the final marketing list of consumers to receive a registration package out of the transactional Siebel System.
8. Steps 5-7 of the process above are repeated for every marketing registration campaign that Wayne Enterprises has.

The Problem
Wayne Enterprises' marketing department cannot execute their 20 daily email campaigns within 24 hours.

The Cause
Frequently queried 3rd party data was being stored in a remote area of the marketing database causing performance issues. The performance issues were slowing down the daily execution of the marketing campaigns.

The Catch
The third party data was correctly being stored in a proven Oracle "out of the box" data schema.

Oracle Marketing Terms
In order to fully explain the case below a few marketing terms will be defined to help understand the issue.

Target Level
A Subject area with a defined Qualified List Item (QLI) is a Target Level. Target levels are used to create marketing segment lists.

Qualified List Item (QLI)
The distinct value that a marketing list is grouped by is referred to as a Qualified List Item or (QLI). This value is set at the Target Level. Examples include; Consumer ID, Response ID, Email Address etc.

Marketing Cache Table
The Marketing Cache table or MC_Target_Level table is where marketing lists are stored after they are created by marketing segments. The Marketing Cache table stores the QLI value for the segment's target level and a system GUID value uniquely describing the segment execution.

QLI	GUID
10	TTAM0I4NAT1Q94IN8100M0

Figure 2. Marketing Cache Table

Campaign Load Format
The fields that makeup the file that is loaded back into the OLTP system is called the Campaign Load Format. This file is created using the MC_TARGET_LEVEL table and the tables of the fields listed in the Campaign Load Format. Fields in the campaign load format are normally related attributes of the QLI.

Uncovering the Issue
To understand the performance issue of the marketing campaigns, the campaign execution process was traced and SQL was captured from steps 5-7 above in the registration process. Below is a deeper look at the marketing execution process.

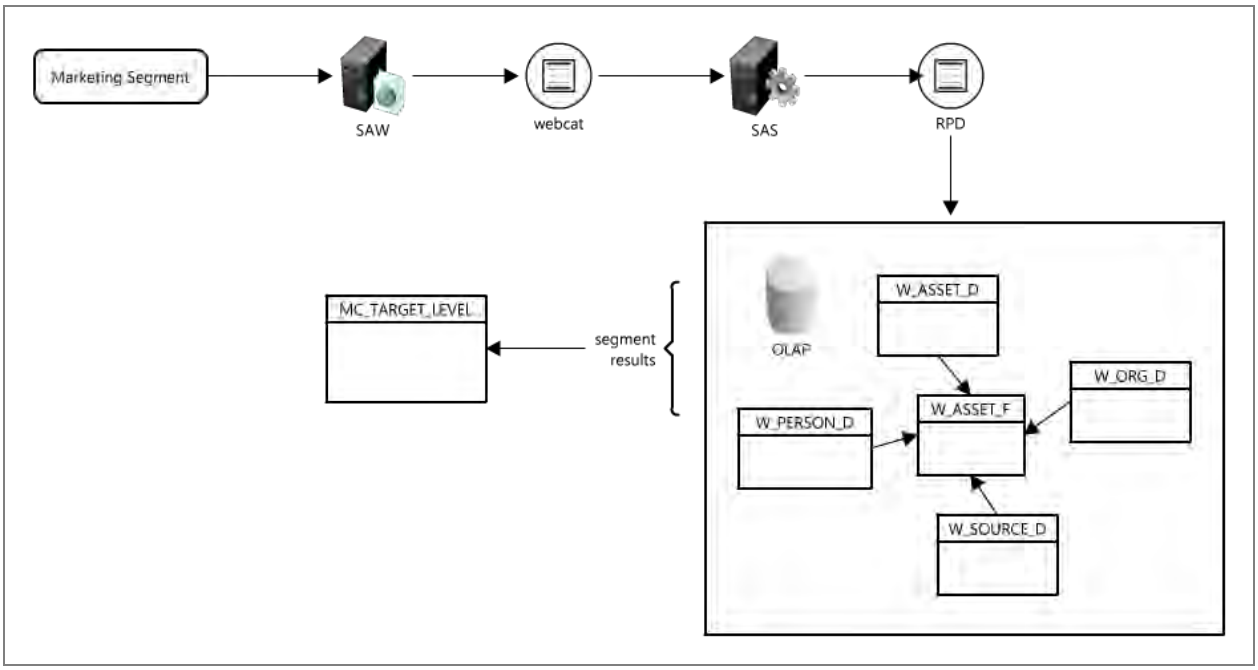


Figure 3. Marketing Process – Segment Execution

1. Marketing Campaign Segment is scheduled in Siebel Marketing (OLTP) and a query is executed against OLAP database (Siebel Analytics)
2. Query results are stored in MC_Target_Level table

After step 2 the SQL was captured from the segment query and reviewed. It was confirmed that the performance issue had not occurred at this point in the process.

3. Attributes are added to the segment results in MC_Target_Level defined in the campaign load so that it can be loaded into OLTP (Call Center, Siebel Marketing)

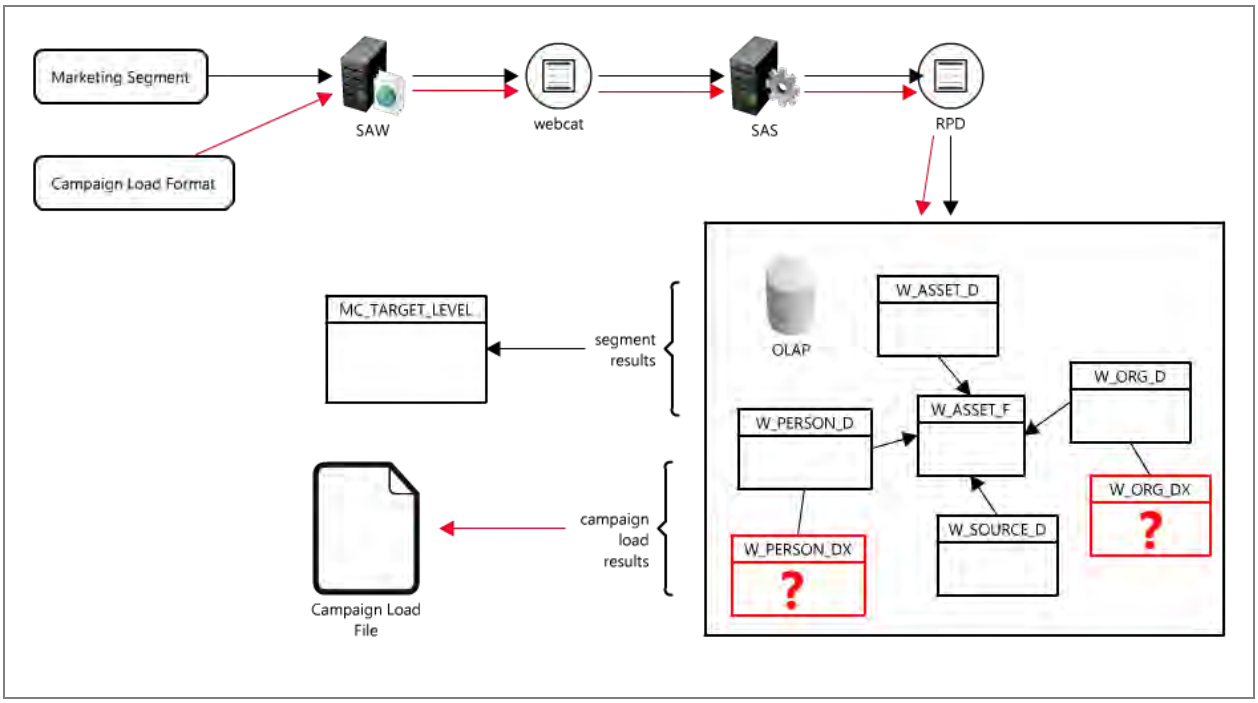


Figure 4. Marketing Process – Campaign Load Execution

While Step 3 was occurring, the SQL running to add the attributes to the QLI was captured and examined. The results found that there were two dimension extension tables with outer joins that were causing a serious performance problem. Items in red in the SQL below highlight the areas with performance issues.

```
(select distinct T259560."ROW_WID" as c1,
T259555."LAST_NAME" as c2,
T259555."FST_NAME" as c3,
T259555."PRSP_CON_UID" as c4,
T259551."INTEGRATION_ID" as c5,
T259555."INTEGRATION_ID" as c6,
T259574."ADDRESS_KEY" as c7,
T259574."PERSISTENT_ID" as c8,
T259555."ROW_WID" as c9,
concat(coalesce(cast(T259555."INTEGRATION_ID" as  VARCHAR ( 20 ) ) , ''), coalesce('~'
, '')) as c10,
T259552."ADDRESS_KEY" as c11,
T259551."ROW_WID" as c12,
T259551."ST_ADDRESS" as c13,
T259552."ST_ADDRESS2" as c14,
T259551."CITY" as c15,
T259551."STATE" as c16,
T259551."ZIPCODE" as c17,
T259551."COUNTRY" as c18,
T259557."INTEGRATION_ID" as c20,
T259558."PROVIDED_MODEL_NUM" as c21,
T259557."SERIAL_NUM" as c22,
T259557."ROW_WID" as c23,
concat(coalesce(cast(T259557."INTEGRATION_ID" as  VARCHAR ( 20 ) ) , ''), coalesce('0'
, '')) as c24
from
"SIEBEL"."W_PERSON_D" T259555,
"SIEBEL"."W_PERSON_DX" T259574,
"SIEBEL"."W_ORG_DX" T259552,
```

```
"SIEBEL"."W_ORG_D" T259551,
"SIEBEL"."W_ASSET_D" T259557,
"SIEBEL"."W_ASSET_F" T259560
where  (
and  "SIEBEL"."W_PERSON_D"."ROW_WID" += “SIEBEL"."W_PERSON_DX"."ROW_WID"
and "SIEBEL"."W_ASSET_D"."ROW_WID"      = “SIEBEL"."W_ASSET_F"."ASSET_WID"
and "SIEBEL"."W_ORG_D"."ROW_WID" = "SIEBEL"."W_ASSET_F"."ACCNT_WID"
and "SIEBEL"."W_ORG_D"."ROW_WID" += "SIEBEL"."W_ORG_DX"."ROW_WID"
and "SIEBEL"."W_PERSON_D"."ROW_WID" = "SIEBEL"."W_ASSET_F"."CONTACT_WID"
and "SIEBEL"."W_PERSON_D". FST_NAME is not null
and "SIEBEL"."W_PERSON_D"."PRSP_CON_UID" is not null
and "SIEBEL"."W_PERSON_D"."LAST_NAME" is not null
```

Figure 5. SQL Captured from Campaign Load Execution

Technical Solution

Move the Third Party Data

All third party data in the campaign load format was moved from the dimension extension tables to the dimensions themselves.

The Result

Wayne Enterprise’s marketing programs were completing in 1/3 of the original execution time allowing for all daily campaigns to be completed.

The Origin of Wayne Enterprise’s Problems

Both the Business and IT failed to communicate with each other resulting in the following

- Business failed to communicate workload of marketing campaigns
- Business felt that IT’s first priority was not to enable business goals
- IT followed best practices by limiting any customization to out of the box data mappings
- IT failed to understand how 3rd party was going to be used

Preventative Measures – An Organizational Solution

SME’s from IT and the business created a weekly meeting to more closely understand each other’s issues and discover areas for improvement.

Metadata Definitions – A Vandelay Industries Case Study

An Overview

Vandelay Industries owns a pharmaceuticals division that utilizes a Siebel Marketing and OBIEE system to run its CRM campaigns. Similar to Wayne Enterprise’s appliance registration process, the following diagram outlines the Vandelay Industries’ consumer fulfillment process.

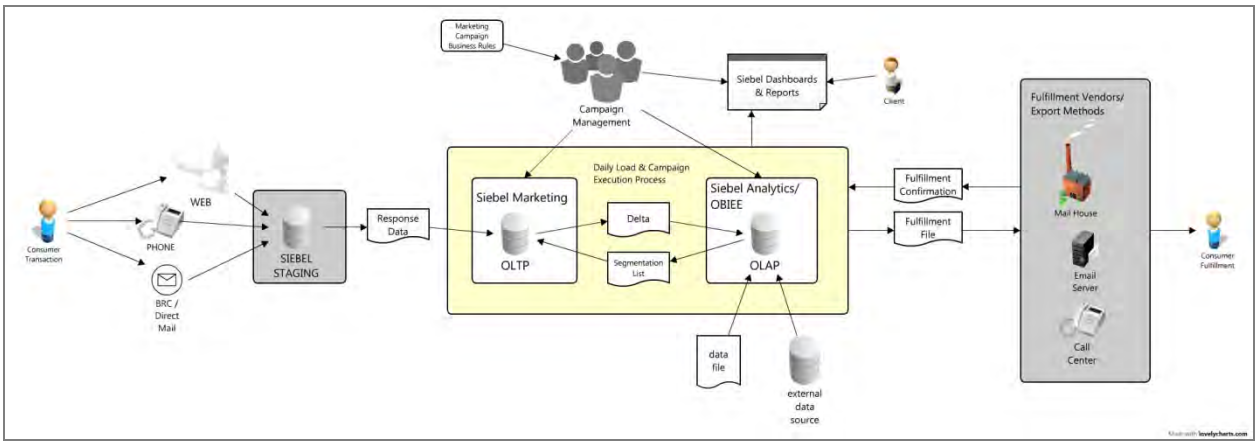


Figure 6. Vandelay Industries Consumer Fulfillment Process

1. Consumers may request more information about a particular brand at Wayne Enterprises by filling out a survey through a call center, website or Business Reply Card (BRC).
2. Wayne Enterprises' IT receives the survey answers and internally performs cleansing, merging and matching of the data.
3. Consumer response data is then loaded into the Siebel Marketing transactional database before being refreshed over to the OBIEE environment for reporting and segmentation.
4. Once loaded, the marketing group can now apply segmentation logic against the consumer request data to create marketing list that outline consumers to be an informational package.
5. Consumer attributes are added to the marketing lists and exported back to the Siebel transactional application where it is loaded and made visible for the Marketing group.
6. A final version of the marketing lists of consumers set to an informational package is exported out of the transactional Siebel System and sent to a fulfillment vendor.

The Problem
Dashboard reports show consumers requesting information on Brand X but receiving information on Brand Y.

The Cause
The dashboard reports were comparing consumer requests (at the request level) to consumer fulfillment (at the consumer level).

The Catch
The issue occurred when segment results of two target levels with different granularity are stored in the same history table (W_CAMP_HIST_F).

A Deeper Look into the Problem
Two different marketing strategies from two separate target levels make up the issue in the campaign history table. Each approach is described below.

Inbound Request Marketing

Scenario Example
Consumer Requests Information about Product A

Segmentation Package	Consumer Eligibility	Target Level	Qualified List Item (QLI)	Campaign Load Format
Informational Package A1	• **valid email address** • **Over age 18** • **expressed interested in product A** • **Request within 5 days**	Inbound Request Target Level	Request ID	• **First Name** • **Last Name** • **Email Address** • **Request ID** • **Package ID**

Campaign Execution Process – Inbound Request Target Level

The process described in Figure 7 differs from Figure 6 by showing how the segment list gets stored back into the OLAP database in campaign history (W_CAMP_HIST_F). It also shows where the Inbound Request Target Level segment and campaign load queries run in the database.

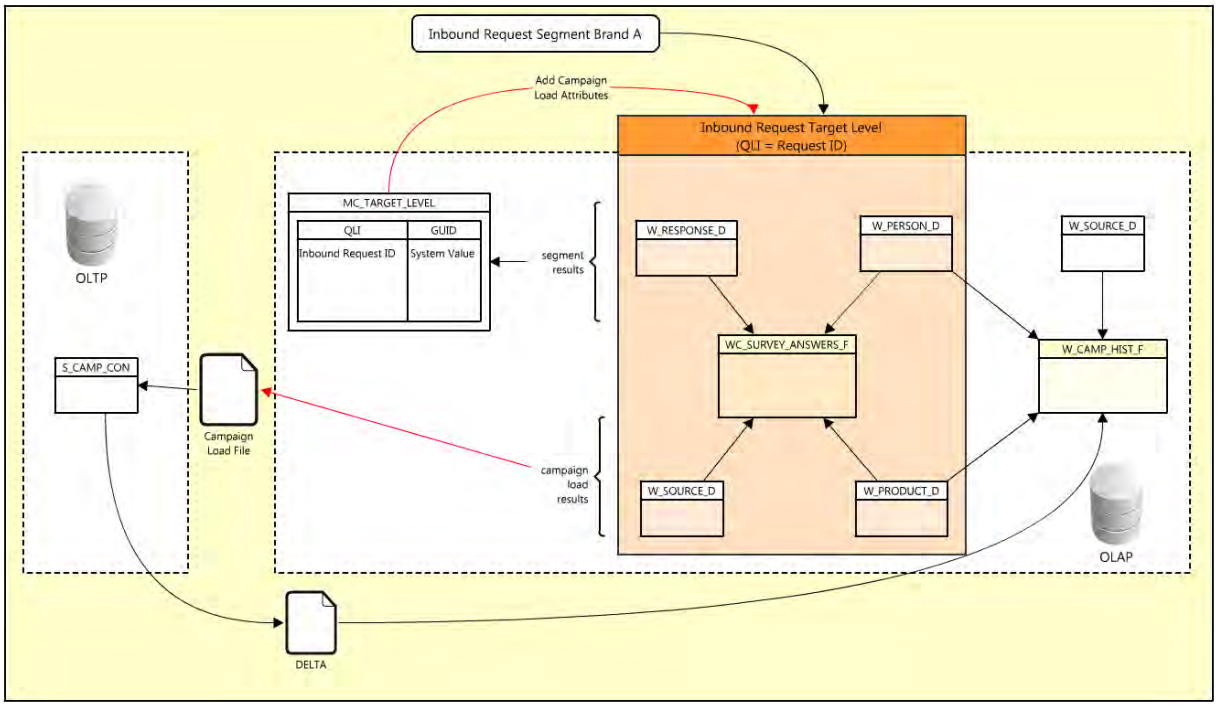


Figure 7. Vandelay Industries Campaign Execution Process – Inbound Request Target Level

Inbound Marketing Target Level – Writing to Campaign History

Below, Figure 8 shows the marketing list record of the scenario example in campaign history (W_CAMP_HIST_F).

INBOUND REQUEST SEGMENT – STORED CAMPAIGN HISTORY					
W_CAMP_HIST_F					
REQUEST DATE	DROP DATE	CONSUMER ID	REQUEST ID	BRAND	PACKAGE
1/2/2011	1/3/2011	10	1	A	A1

Figure 8. Inbound Marketing Record in Campaign History

Campaign History Marketing

Scenario Example

Send Consumer a Second Package after First is received

Segmentation Package	Consumer Eligibility	Target Level	Qualified List Item (QLI)	Campaign Load Format
Informational Package A2	• **They have received information package A1** • **Have not opted out**	Campaign History Target Level	Consumer ID	• **First Name** • **Last Name** • **Email Address** • **Request ID** • **Package ID**

Campaign Execution Process – Campaign History Target Level

The process described in Figure 9 shows how a segment and campaign load process query the database when they are made from the Campaign History Target Level. It also shows how the campaign load file gets loaded back into campaign history.

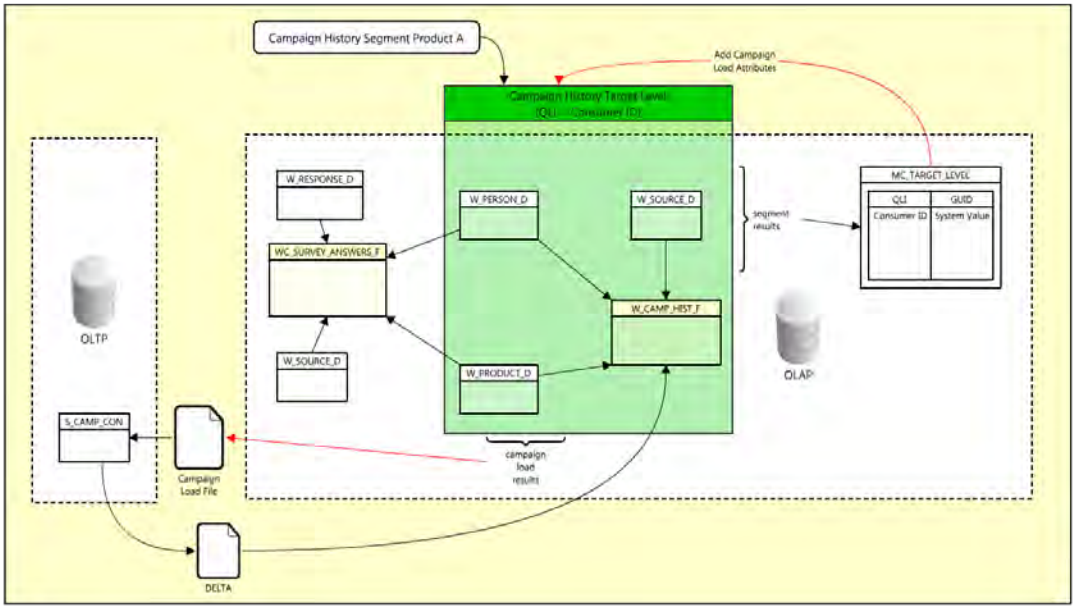


Figure 9. Campaign History Segment - Execution Flow

Campaign History Target Level – Writing to Campaign History

Below, Figure 10 shows a Campaign History marketing list record in campaign history (W_CAMP_HIST_F) ,

CAMPAIGN HISTORY SEGMENT - HISTORY					
W_CAMP_HIST_F					
REQUEST DATE	DROP DATE	CONSUMER ID	REQUEST ID	BRAND	PACKAGE
1/2/2011	1/3/2011	10	1	A	A1
1/2/2011	2/21/2011	10	1	A	A2

Figure 10. Campaign History Target Level List Record in Campaign History

When the Problem Occurs

In the example above the records stored in W_CAMP_HIST_F were accurate. This, however, is not guaranteed to happen if the consumer has responded more than once. Figure 11 shows another take on the campaign history segment scenario.

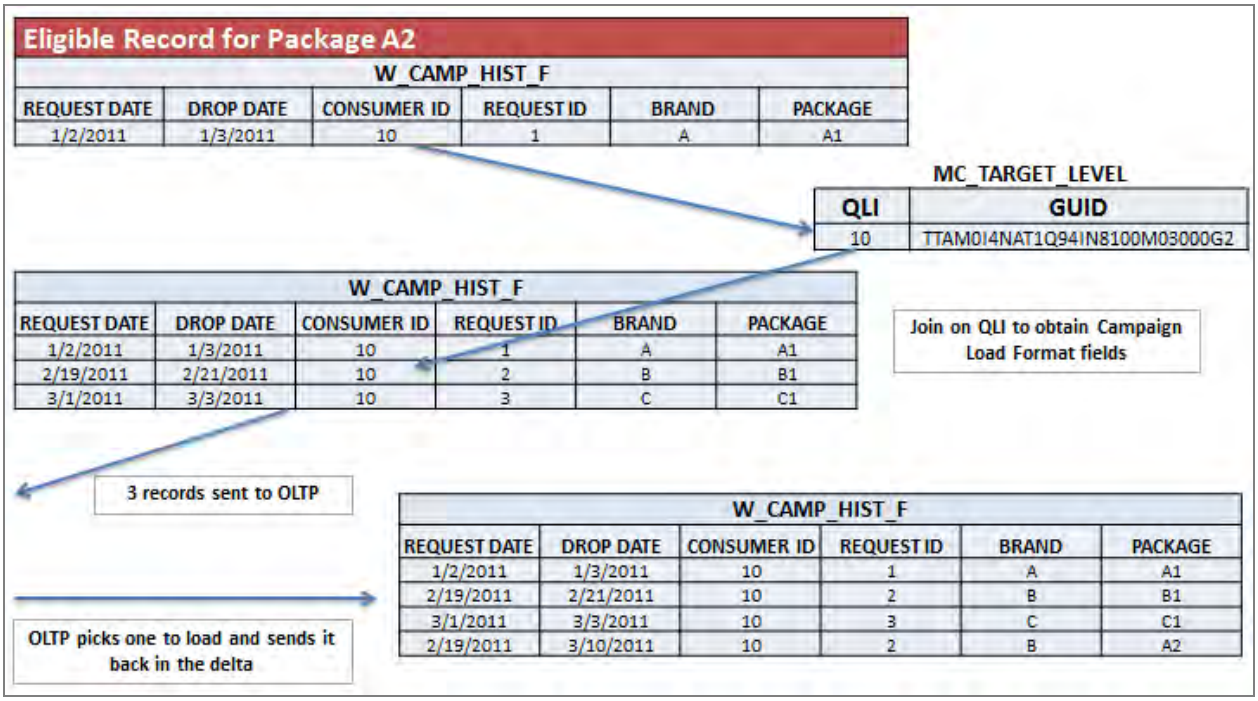


Figure 11. Campaign History Campaign Load Format Process

When the QLI (Consumer ID) joins to W_CAMP_HIST to add Response ID, it will retrieve all responses for that Consumer. The example in Figure 11, 3 responses are sent along to the transactional marketing application to be loaded. The load process only selects one of the three to load and sends it back in the daily delta refresh to be loaded into W_CAMP_HIST_F. More often than not it is incorrect.

The Solution

The QLI in Campaign History Target Level needs to be changed so that it is on the same level as the Inbound Request Target Level. Changing the QLI from Consumer ID to Request ID will most accurately populate the campaign load files with the correct data.

The Origin of Vandelay Industries Problems

Both Business and IT failed to communicate with each other resulting in the following

- Business failed to express importance of request to package tracking functionality during implementation
- IT failed to recognize the inconsistencies within its own data and understand basic business need

Organizational Solution

The Business groups started hiring more technical SME's for internal communications with IT.

Dashboards and Reports

Everyone knows that reports are often misunderstood. This can happen for a number of reasons but in many cases this can be prevented if the purpose for the report or what the report is supposed to represent is understood. Below are three reporting examples where this applies.

Quality KPI's

Consider the following reports. In Figure 12, it appears that Brand B is performing the best due to the number of inbound calls it is receiving in the call center.

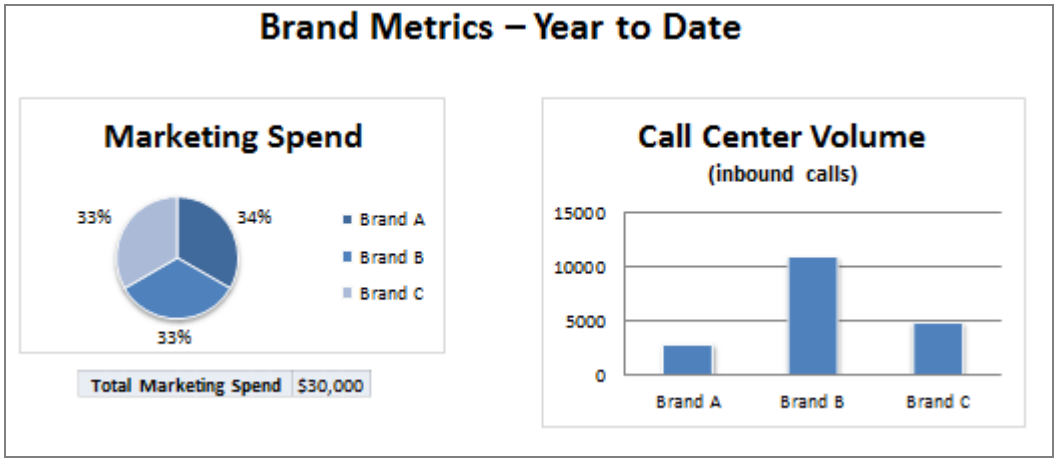


Figure 12.

In this example, the goal of the Call Center is to collect actionable consumers. If a report, like Figure 13, measuring unique individuals with valid addresses (Qualified Leads) was added, the dashboard would tell a different story.

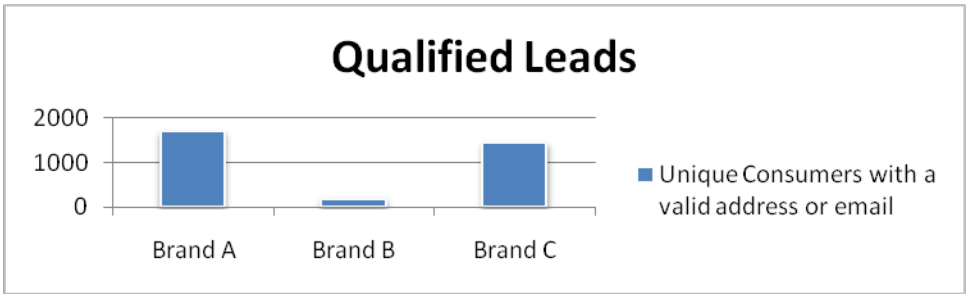


Figure 13.

In Figure 13, Brand B has the lowest number of Qualified Leads. This means that the brand with the highest call volume has the worst lead capture volume. This is a huge red flag. It might mean that there is an issue with the automated voice messaging system that is causing callers to hang up. The opinion of this brand is less favorable now because of a more accurate KPI.

Show Volume, Not Just Percentage, When Showing Comparisons

Without a count of actuals, Brand A appears to be performing better than Brand B.

Figure 14.

When actual counts are applied like in Figure 15, it is much more obvious that Brand B has far more Qualified Leads than Brand A.

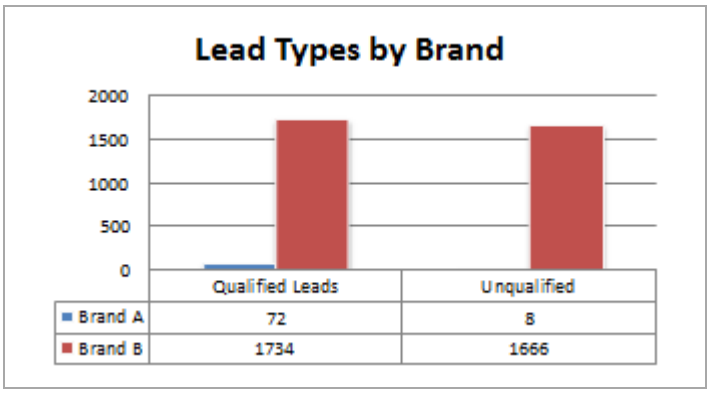


Figure 15.

Rounding to the Millions – Consider the Consequences

Example A below shows an expected result for rounding.

Example A.

	Rounded	Actual
Premiums earned	405	405,321,632.21
Net investment income	40	39,989,123.12
Net realized investment gains/(losses)	-	21,231.00
Fee and other revenues	3	3,018,897.43
Total Revenues	448	448,350,883.76

Example B below shows an unexpected consequence because of rounding.

Example B.

	Rounded	Actual
Premiums earned	-	412,987.00
Net investment income	-	448,345.00
Net realized investment gains/(losses)	-	401,654.00
Fee and other revenues	-	422,987.00
Total Revenues	2	1,685,973.00

Consider scenarios in your own system that are similar to Example B before deciding to round to such a large value.

User Analysis

Do you know your End Users? Ask yourself the questions found in Figure 16.

- Do you have smart/proactive end users?
  Yes
- Do they work in a fast, results driven climate?
  Yes
- Do you know what they are responsible for?
  I have an idea
- During their busy peaks do you see a big increase in requests?
  A little

Figure 16.

Did you answer anything like this? If so, it might be time to conduct an internal assessment to uncover some areas for improvement and automation opportunities.

Discover Offline User Activities through Workshops

Figure 17 shows a process map of a Financial Reporting group. Some of the discoveries from the workshop included

- Hand off files from other groups
- Large number of Manual Calculations
- Reporting directly off Excel Files (5 Layers Deep)
- Excel files with multiple data sources

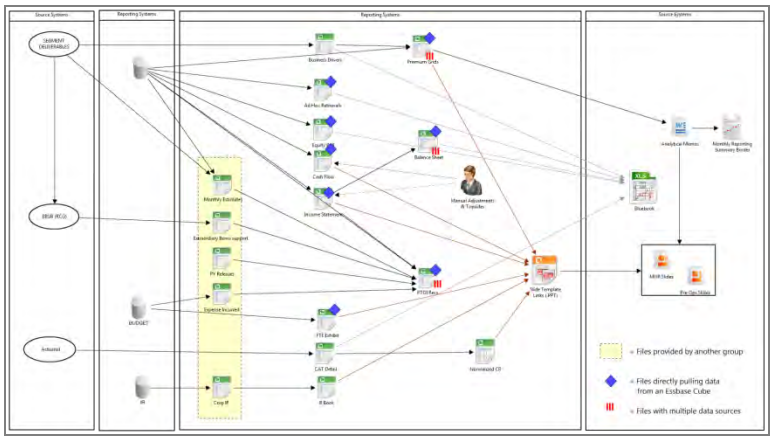


Figure 17.

Solutions to the Madness

Technical

- Push all manual activities to the data source & reporting tool
- Review reporting deliverables and make them available in the reports

Preventative Methods

- Determine an agreed upon review period and conduct workshops to expose activity opportunity areas
- Prioritize biggest impact areas first

Figure 18.

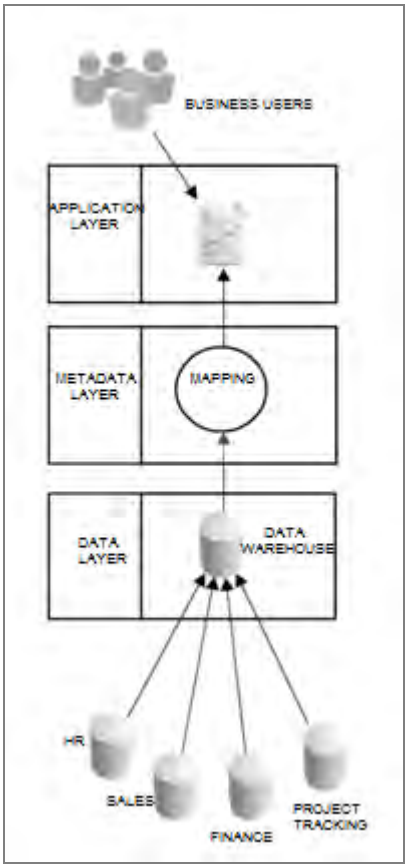


Figure 18 to the right shows business users pulling data from an application where there is one version of the truth. The calculations of occur in the database and no handoffs are occurring.

Lessons Learned

Most Business Intelligence issues start because of a lack of understanding between Business and IT. Figure 19 shows ways to remove the communication barrier between Business and IT and minimize risk in any Business Intelligence project.

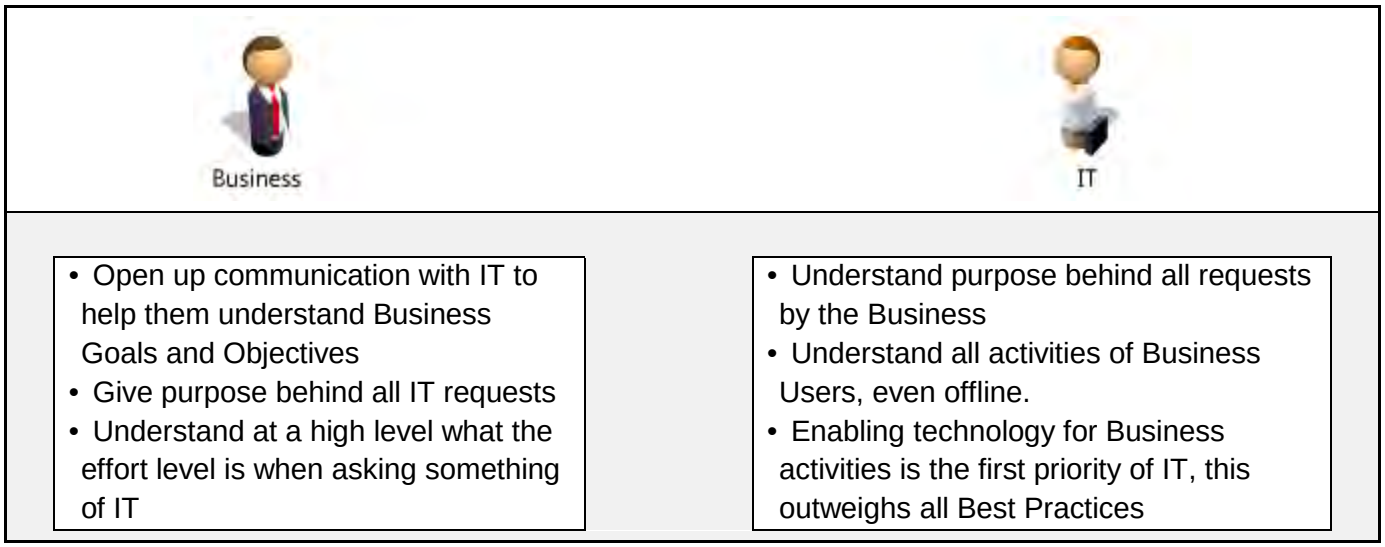


Figure 19.

SIGS, SIGS and more SIGS!

The following Special Interest Groups (SIG) hold meetings throughout the year for the benefit of NYOUG members:

DBA SIG – Database Administration
Data Warehouse SIG – Business Intelligence
Web SIG – Web / XML / Java / Weblogic / APEX / Fusion
Long Island SIG – Nassau/Suffolk area - All topics

Accelerate Oracle 25X

SAN, NAS and Direct Attach

RamSan-630 SSD

+ 10 TB SLC Flash
+ Reads/Writes 1,000,000 IOPS
+ Throughput up to 10 GB/s
+ 500 Watts
+ InfiniBand

NEW

Workstations and Servers

RamSan-20 SSD

+ 450 GB SLC Flash
+ Reads/Writes 120,000/50,000 IOPS
+ Throughput up to 700 MB/s
+ 15 Watts
+ PCIe

Reliable, affordable, energy efficient solid state disks from Texas Memory Systems, makers of the World's Fastest Storage®

When you accelerate your Oracle applications with a RamSan Solid State Disk you can cost-effectively handle extreme transaction volumes and an increasing number of simultaneous users.

You'll need fewer servers, licenses and hard disk drives so you can save resources, power, and space in your data center. And, you'll save a lot of money!

For over 30 years RamSan SSDs are the first choice for speed and reliability. They are used by 9 major financial exchanges, 7 of the 11 world's largest telecoms and by businesses, government and military organizations in over 35 countries.

FREE EBOOK

You can learn more, request a self-service price quote and get a free copy* of Oracle Performance Tuning with Solid State Disks by visiting TxMemSys.com/nyoug

TEXAS MEMORY SYSTEMS
RAMSAN: THE WORLD'S FASTEST STORAGE®

World's Fastest Storage and RamSan are trademarks of Texas Memory Systems, Inc.
All other trademarks belong to their respective owners. | © 2011 Ref. 377-347

Unleash the Value of the Data in Your OEM Repository

Iordan K. Iotzov, News America Marketing

Introduction

Access to comprehensive and reliable data is essential for any methodical analysis or forecasting. Oracle Enterprise Manager (OEM) repository contains a wealth of raw and partially processed database-related information that can be put into use for various projects. Looking for historical patterns is vital in troubleshooting performance problems. OEM repository data, in conjunction with the data in Oracle Automatic Workload Repository (AWR)[1], dynamic performance views, and trace files can provide a broad view of the state of a system. Resource management and forecasting are other areas where OEM repository data can be of enormous help. Since the repository resides in an Oracle database, we not only get access to the data, but we can also utilize the computing power of the Oracle server, specifically its analytical and statistical built-in functions and packages.

Most of the useful data in the OEM repository is collected to support functionalities related to OEM metrics. The data is initially gathered by OEM agents that are deployed on each monitored machine (Figure 1).

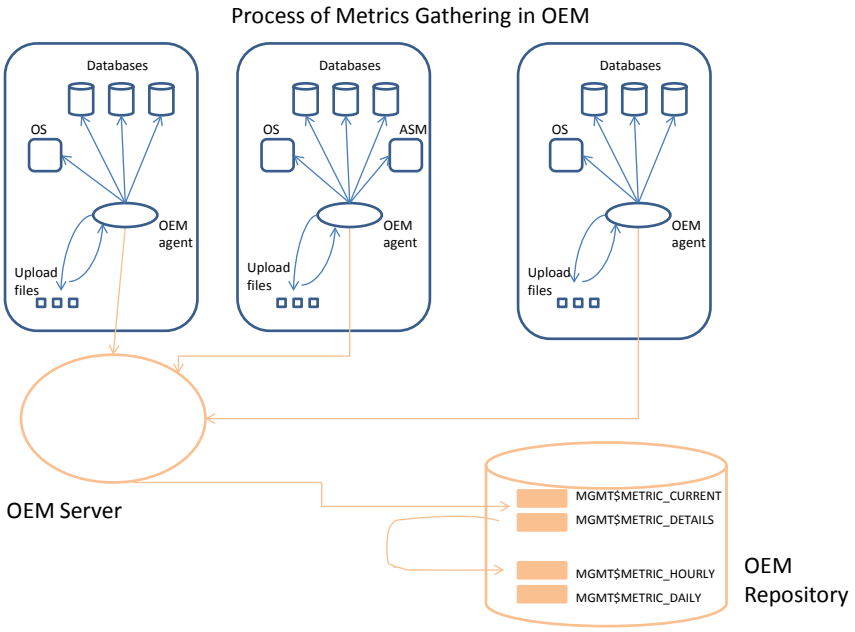


Figure 1. Overview of Metric-gathering in OEM

Each metric has its own collection frequency that in most cases can be easily changed via the "Metric and Policy" OEM tab. For metrics that do not have thresholds, such as the Tablespace Allocation metric, changing the frequency can be accomplished with the command line utility emcli. We can also control the retention period for many metric-related repository tables in OEM – detailed information can be found in *Oracle® Enterprise Manager Advanced Configuration*

[1] Oracle Automatic Workload Repository (AWR) is a separately licensed feature

10g Release 5 (10.2.0.5) and in *Metalink Note* ***[ID 430830.1] (Grid Control Repository: How to Modify the Retention and Purging Policies for Metric Data?)]***
After being transmitted from an OEM agent, the metric data goes into MGMT$METRIC_CURRENT and MGMT$METRIC_DETAILS views. Those views hold the most recent entry and the most recent 25 hours of raw data respectively. As most of the time we are looking for cumulative information, we are usually dealing with views that are aggregated from raw data at a regular interval. MGMT$METRIC_HOURLY view contains count, average, minimum, maximum, and standard deviation aggregated every hour for the last 30 days. MGMT$METRIC_DAILY stores the daily aggregations for one year. Detailed documentation of OEM repository views can be found in *Oracle® Enterprise Manager Extensibility 10g Release 2 (10.2) for Windows or UNIX* and in *Metalink Note* ***831243.1 (Examples: Creating Custom Reports)*.**
OEM User Defined Metrics (UDM) are collected and stored the same way as all other metrics. The UDMs that are generated by a shell script have a metric_name of "UDM", while those that come from a SQL statement are under "SQLUDM" metric_name. Even though Oracle comes with several hundred predefined metrics, UDMs are very useful because they allow us to aim at a specific target parameter. Often the overall performance characteristics, such as CPU utilization and I/O wait, are not as important as the performance of the most critical business transaction. UDMs are ideal for this type of customizations.

Useful Applications

Troubleshooting performance problems often necessitate evaluating different hypothesis about the root cause of an issue. The data in OEM can be used to accept or reject a wide range of theories. Moreover, by utilizing Oracle DB links, the information in OEM can be easily combined with AWR data for an even more comprehensive view of the performance of the enterprise.
OEM provides access to the data it collects in a variety of ways. OEM metrics have good GUI screens and some ability to customize the output.

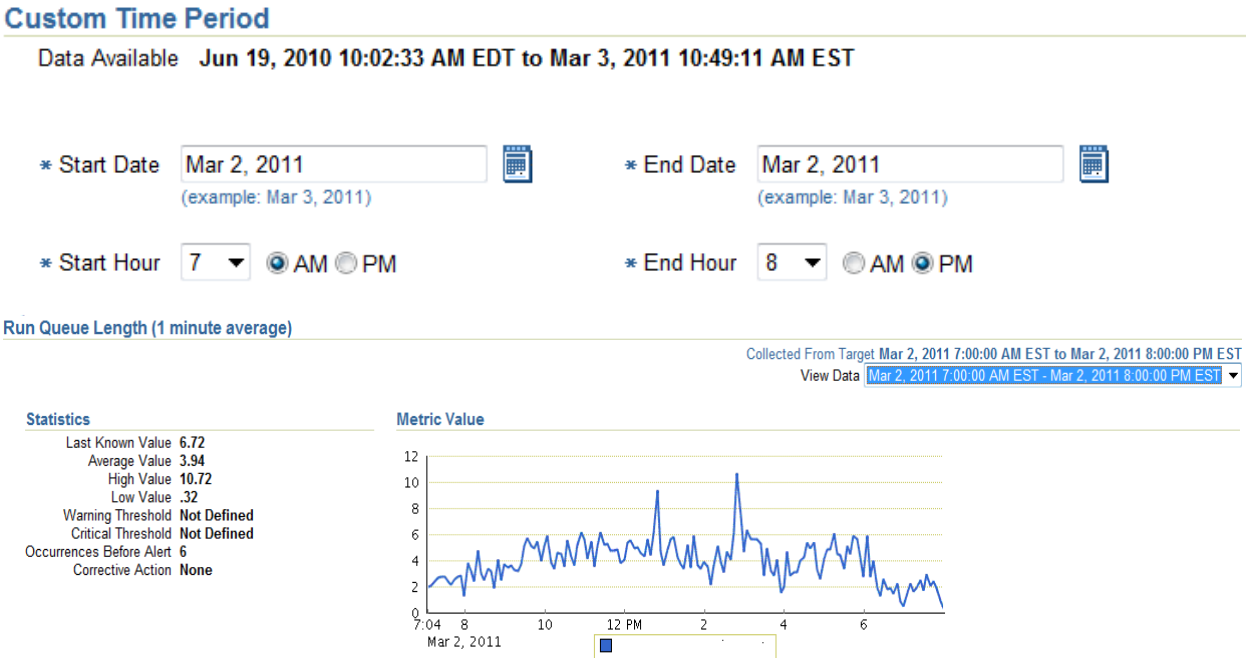


Figure 2. Customization of Metrics Display

The flexibility of those screens is rather limited though (Figure 2). A quite typical request for run queue length during business hours (7 am – 8 pm), averaged over 30 days, cannot be satisfied with the existing built-in functionality. That problem can be easily solved by working directly with the data in the repository:

```
select
      avg(average)
from
       mgmt$metric_hourly
where
      rollup_timestamp  > sysdate – 30
and   target_name             = 'dbtest01'
and   metric_name             = 'Load'
and   column_labal            = 'Run Queue Length (1 minute average)'
and   to_char(rollup_timestamp,'DAY') not in ('SATURDAY','SUNDAY')
and   rollup_timestamp  between trunc(rollup_timestamp,'DD') + 7/24
                  and trunc(rollup_timestamp,'DD') + 20/24
```

2.970532852564102564102564102564102564 11

The OEM repository stores data for all databases in an enterprise, making it the ideal place to implement procedures that monitor and enforce enterprise-wide policies. We can monitor Force Logging status of all production databases by utilizing UDMs and OEM repository tables. Each production DB would pass along its Force Logging status to the repository with an UDM (Figure 3).

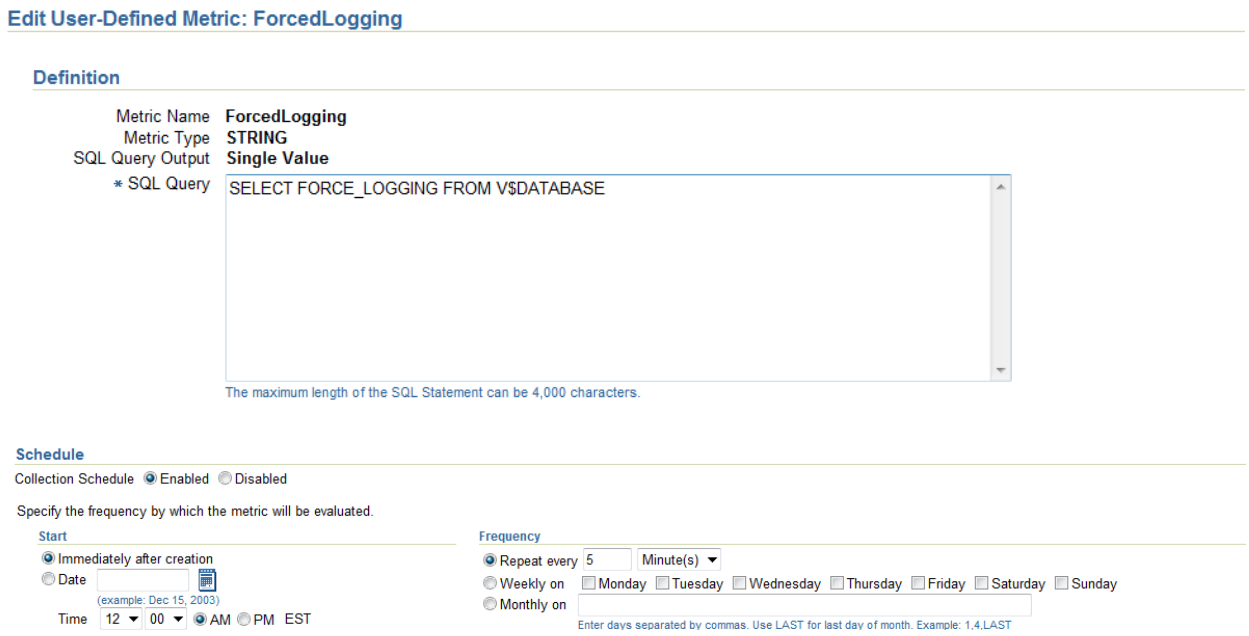


Figure 3. UDM for Collecting "Force Logging" Mode Information on Production DB

The actual monitoring is implemented as a different SQL UDM against the OEM repository:

```
select
      count( member_target_guid )
from
      mgmt$group_derived_memberships o ,
      mgmt$target t
where
      o.composite_target_name       = 'PROD'
and   o. member_target_guid   = t.target_guid
and    ( t.target_type         ='rac_database'
                or
```

```
          (t.target_type              ='oracle_database'
                and t.type_qualifier3    != 'RACINST‘
          )
      )
and   not exists (
            select
                  *
            from
                  mgmt$metric_current i
            where
                  i.target_guid           = o.member_target_guid
            and   metric_name             = 'SQLUDM'
            and   column_label            = 'ForcedLogging'
            and   Metric_Column           = 'StrValue'
            and   collection_timestamp    > sysdate - 20/1440
            and   value             = 'YES‘
             )
```

With this setup, every production database must have Force Logging status ON and a local UDM in place in order not to trigger the OEM repository UDM. This framework allows great flexibility as well. If we would like to exempt a production database from the Force Logging policy, we can create an exempt UDM on that production DB (Figure 4).

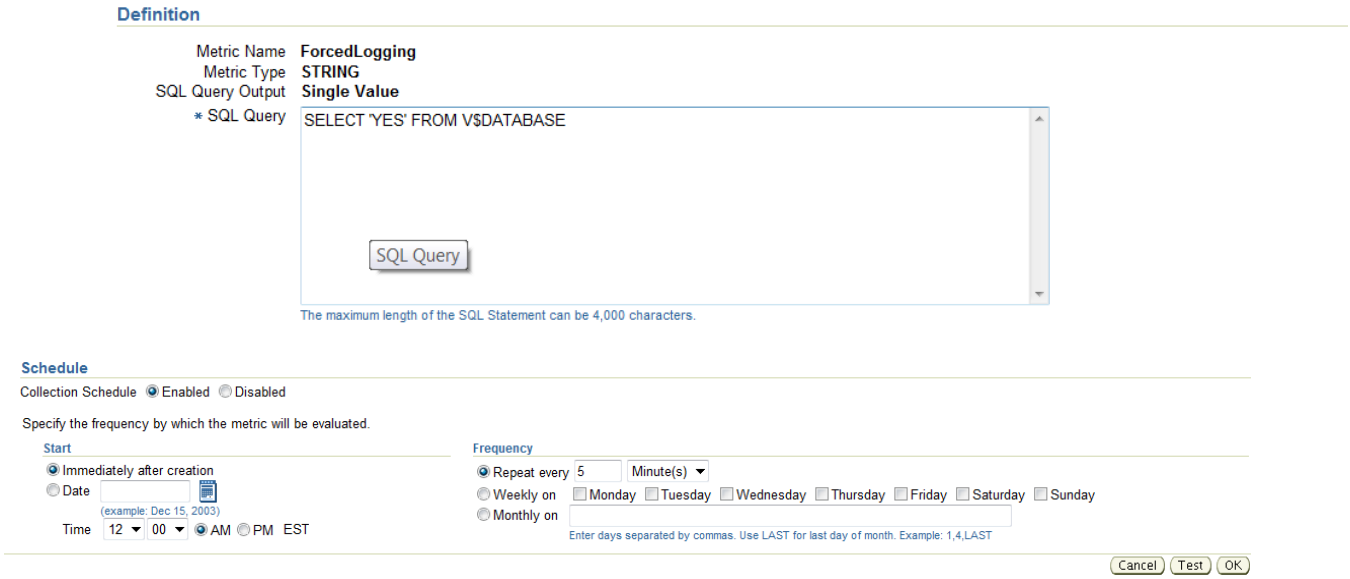


Figure 4. UDM for Exempting a DB from the "Force Logging" Policy

The flexibility can go even further - we can allow only certain tablespaces to be exempted from the Force Logging policy, while making sure that all other tablespaces have the setting (Figure 5).

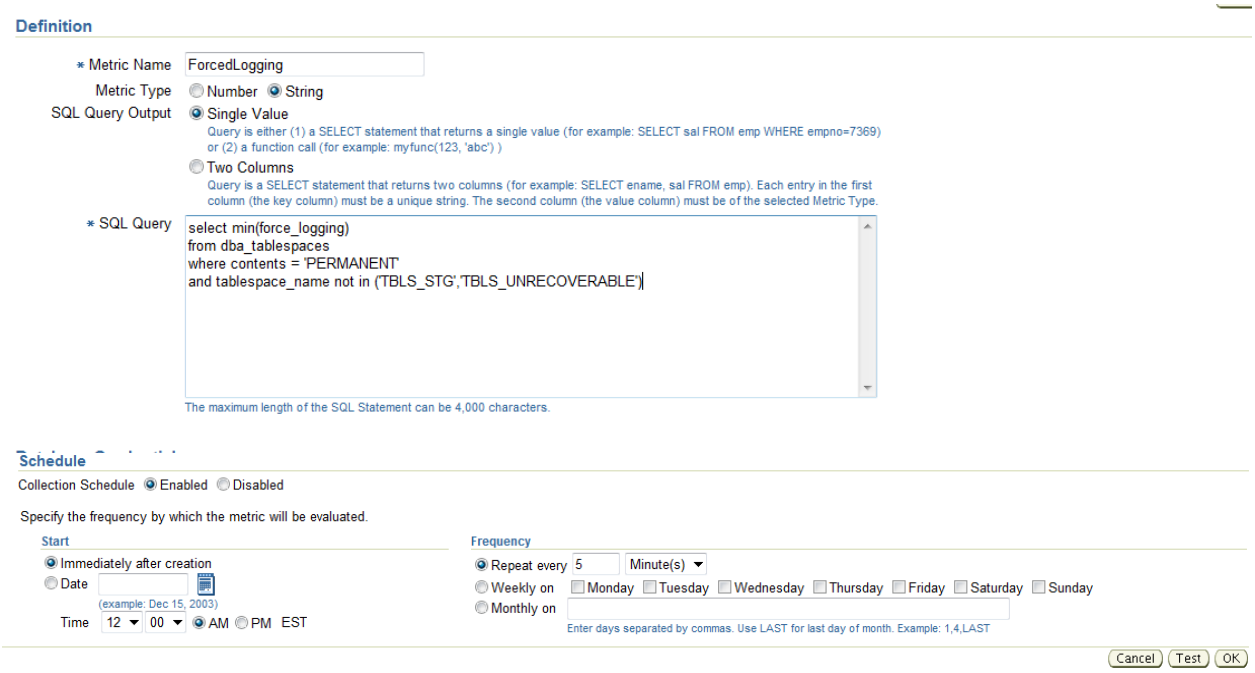


Figure 5. UDM for exempting tablespaces from the "Force Logging" policy

One of the advantages of working directly with the OEM repository is that we can easily connect it with other sources of valuable database performance data that are also stored in Oracle. AWR/ASH is one such suitable candidate – we can effectively run any query against any table/view in any of the two repositories. That option tremendously increases our ability to prove or revoke theories about performance problems.
Let's investigate the variation in a single block read time on a database that sits on ASM. A single query based on ASH and OEM repository can enable us to check various relationships:

```
select
      corr(db.time_waited , oem.value) , count(*)
from
      (select
            sample_time ,
             time_waited
      from
            dba_hist_active_sess_history@prod_db
      where
            event        = 'db file sequential read'
      and   session_state      = 'WAITING'
      and   sample_time        > sysdate - 1 ) db ,
      ( select
            collection_timestamp ,
             value
      from
            mgmt$metric_details
      where
            target_name        = '+ASM_PROD'
        and        metric_name   = 'Single_Instance_DiskGroup_Performance'
        and        metric_column       = '<ASM Metric>' ) oem
  where
       oem.collection_timestamp      between db.sample_time - 15/1440
                  and db.sample_time
```

The result of this analysis (Table 1) contains many helpful leads. It shows that the volume of IO writes, rather than the volume of IO reads, is what influences the speed of a single read. We cannot expect high absolute correlation in this case because the frequencies of gathering in ASH and OEM repository ('Single_Instance_DiskGroup_Performance') respectively are very different.

ASM Metric	Correlation
writes_ps	0.04
reads_ps	-0.04
write_throughput	0.04
read_throughput	0.01

Table 1. Correlation between Select ASM Metrics and Time of Single Block Read (DB)

We can utilize the OEM repository to find out what caused a significant latency on the apply side of a Streams configuration. Some believe that the load on the IO system on the apply side causes the latency. Using CORR SQL function and the data in OEM, we can put that hypothesis to a test:

```
select
      corr(streams_latency.value , IO_load.value)
from
      ( select *
      from
            mgmt$metric_details
      where
            target_name = 'STRDEST'
      and   metric_name = 'streams_latency_throughput'
      and   column_label = 'Latency' ) streams_latency ,
      (select *
      from
            mgmt$metric_details
      where
            target_name = ‘STRDEST'
      and   metric_name = 'instance_throughput'
      and   column_label = 'I/O Megabytes (per second)') IO_load
where
      streams_latency.collection_timestamp between IO_load.collection_timestamp
                                    and IO_load.collection_timestamp  + 5/(60*24)
and   streams_latency.collection_timestamp  > sysdate – 1
```

```
-0.0010434366348630079753544589947484440986 31
```

It seems that this is not a viable lead. Is it possible that the latency is linked to the amount of redo generated on the capture database? A single query can check that idea as well:

```
select corr(streams_latency.value , redo_source_1.value + redo_source_2.value)
from
      ( select *
        from
            mgmt$metric_details
```

```
        where
            target_name = 'STRDEST'
        and       metric_name = 'streams_latency_throughput'
        and column_label = 'Latency' ) streams_latency ,
      (select *
      from
            mgmt$metric_details
      where
            target_name = 'STRSRC_STRSRC 1'
      and   metric_name = 'instance_throughput'
      and   column_label = 'Redo Generated (per second)' ) redo_source_1 ,
       (select *
      from
            mgmt$metric_details
      where
            target_name = 'STRSRC_STRSRC 2'
      and   metric_name = 'instance_throughput'
      and   column_label = 'Redo Generated (per second)' ) redo_source_2
where
      streams_latency.collection_timestamp       between
redo_source_1.collection_timestamp                          and
redo_source_1.collection_timestamp  + 5/(60*24)
and   streams_latency.collection_timestamp       between
redo_source_2.collection_timestamp
                  and redo_source_2.collection_timestamp  + 5/(60*24)
and   streams_latency.collection_timestamp > sysdate – 1
```

0.764129749129402128979924364231927912862 9

Here, the verdict is clear – the redo size on the capture database correlates heavily with the Streams latency on the target database and is most likely the root cause of this problem.
OEM reports are another way to tap into the value of the data stored in the OEM repository. There are a few built-in reports, but most of them accept few parameters. Those restrictions can be overcome with using custom OEM reports. In fact, this has been my preferred way of delivery for most of the complex projects that I have worked on.
I typically use custom OEM reports with a SQL select statement, an option that provides simplicity and versatility for the report users. That choice presents a little bit of a challenge because it limits the ability to use intermediary tables in the underlying logic. Moreover, the programming choices are affected because some built-in PL/SQL functions that require a table column as an argument, such as DBMS_STAT_FUNCS, cannot be used. One way to get over those issues is to use pipelined functions and autonomous transactions.

Advanced Forecasting Example

To illustrate the potential of directly using OEM repository data, we'll go though the development of enterprise-wide disk space forecasting utility for Oracle databases. The model behind the utility is linear regression. It assumes that tablespace size (tbls) is a linear function of time (t):

```
 tbls(t) = tbls0 + incr*t
```

Parameter tbls0 is called intercept; incr is named slope. The first task is to find an estimate of tbls0 and incr based on the historical information about tablespace size that is kept in the OEM repository. Oracle built-in linear regression functions (prefixed REGR) are a convenient way not only to get the intercept and the slope, but also to get many other useful linear regression parameters.

```
select
      regr_intercept (average , rollup_timestamp -  sysdate  ) ,
      regr_slope (average , rollup_timestamp -  sysdate  ),
```

```
      regr_r2 (average , rollup_timestamp -  sysdate  ),
      regr_count (average ,rollup_timestamp -  sysdate  ),
      regr_avgx (average ,rollup_timestamp -  sysdate  ),
      regr_sxx (average ,rollup_timestamp -  sysdate  )  ,
      regr_syy (average ,rollup_timestamp -  sysdate  ),
      regr_sxy (average ,rollup_timestamp -  sysdate  )
 into
      …..
 from
      raw_data
```

where raw_data table gets populated with the following query

```
insert into raw_data
select
      m.rollup_timestamp ,
      m.average
from
      mgmt$metric_daily m ,
      mgmt$target_type t
where
      (t.target_type    ='rac_database'
          or (t.target_type       ='oracle_database'
                and t.type_qualifiers3 != 'RACINST'))
and   m.target_guid           =p_target_guid    (Database)
and   m.target_guid     = t.target_guid
and   m.metric_guid            =t.metric_guid
and   t.metric_name           ='tbspAllocation‘
and   t.metric_column          = 'spaceUsed'
and   m.rollup_timestamp       >= sysdate - p_period_hist
and   m.rollup_timestamp      <= sysdate
and   m.key_value             = p_tablespace_name;
```

Some tablespaces grow in a straight and orderly fashion, while others have significant variations and hard to identify trends. Newly created tablespaces have little historical data to base our predictions on. Those concerns bring us to the next challenge – how to find a range for the forecasted tablespace size? The formula can be found in statistics textbooks; below is its implementation in PL/SQL:

```
l_ci:=sqrt ( 1/ l_cnt
                     +
                        power((p_period_forecast - l_avgx),2)
                        /
                        l_sxx
                    );
 l_sigma:= sqrt(
      abs(
            (l_syy
            -
            (power(l_sxy,2)/l_sxx)
            )
              )
              /(l_cnt-2)
      )  ;
 p_out_estimate_95_range:=(l_ci
                                *
                                f_forecast_t_dist(l_cnt-2,0.95)
```

```
                                        *
                                        l_sigma) ;
```

The p_period_forcast is the time we are making the forecast for, in number of days. The f_forecast_t_dist is a custom function that accepts degrees of freedom and confidence level and returns the t-distribution statistic. The rest of the parameters are from the linear regression query in the previous paragraph. These estimates should be used for forecasting values that are "near" the historical data. For instance, forecasting 3 years ahead based on 3 months of history would probably not produce reliable results.

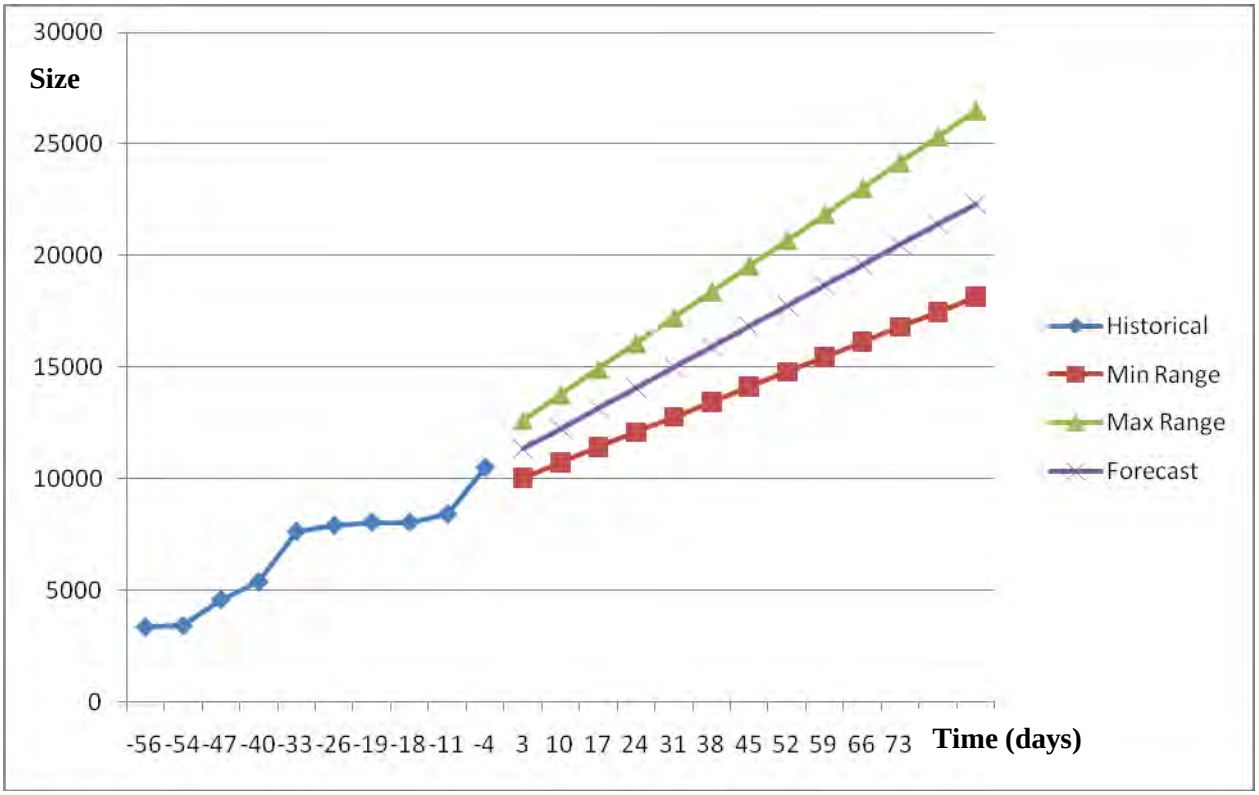


Figure 6. Linear Regression with 95% Prediction Interval

Figure 6 provides a graphical illustration of this method. We can see how the forecasted values follow the historic trend. We can also observe that the further in the future we forecast, the wider the range becomes.

An important part of forecasting is to find out if your data is suitable for the model -in our case linear regression. Normal distribution, lack of autocorrelation of residual errors and lack of heteroscedasticity are the prerequisites for linear regression. The better that our data set conforms to those prerequisites, the more reliable the forecasted range would be. The forecasting utility is implemented with PL/SQL functions in a package that resides in the OEM repository.

```
PACKAGE FORECAST AS
FUNCTION EST_PROD_DBS ( p_lookback NUMBER , p_lookforward NUMBER …)  RETURN
forecast_compact_type PIPELINED ;
FUNCTION EST_DB ( p_target_guid VARCHAR2 , p_lookback NUMBER  ……)  RETURN
forecast_compact_type PIPELINED ;
FUNCTION EST_TABLESPACE ( p_target_guid VARCHAR2 , p_tablespace_name VARCHAR2 ……)
RETURN NUMBER ;
END FORECAST;
```

EST_DB and EST_PROD_DBS functions generate forecast reports for a database and all production databases respectively. Those functions call EST_TABLESPACE, which performs most of the statistics computations. Figure 7 shows how the custom OEM report for this project is defined.

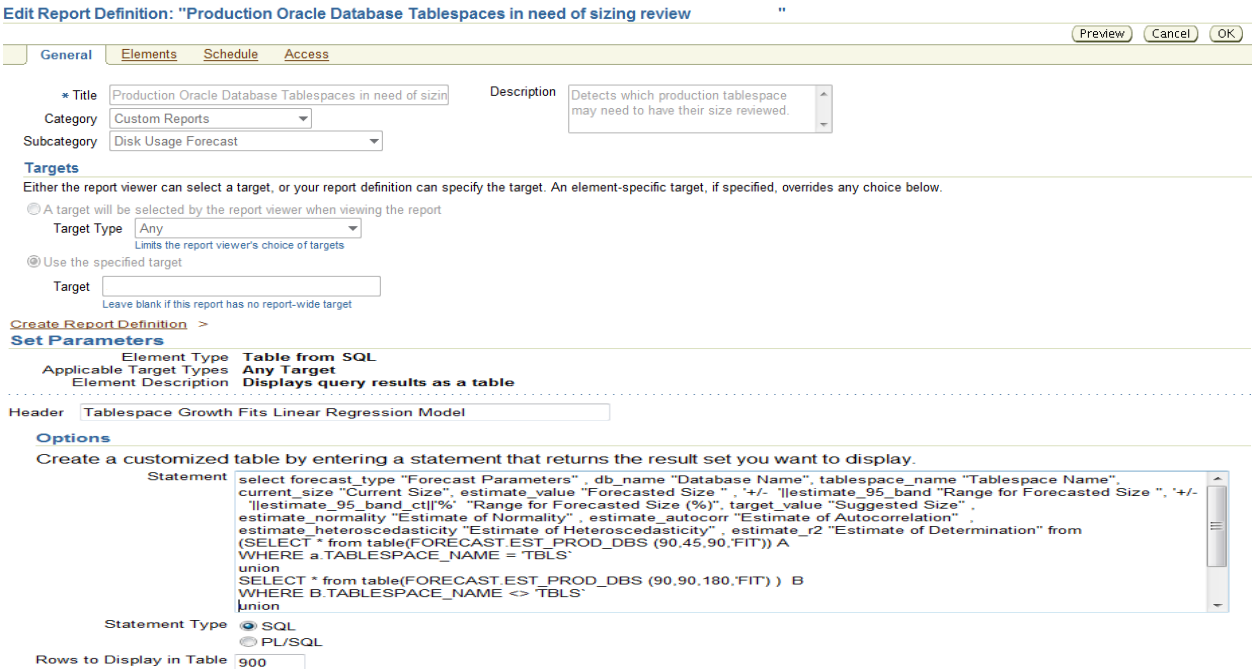


Figure 7. Custom OEM Forecast Report Definition

The final report (Figure 8) provides lots of valuable information to users who are in charge of disk allocation decisions. The first column [Forecast Parameters] shows the parameters used for the specific forecast. In this case we look for the most recent 90 days worth of data, and based on that we generate a forecast for 90 days in the future [Forecasted Size] as well as 180 days in the future [Suggested Size]. In addition to forecasting the tablespace sizes, we compute the range for those estimates, as an absolute value [Range for Est. Forecast] and as a percentage [Range for Est. Forecast (%)], as well as number of other regression variables. If the forecasted size is larger than the current size, then we show the tablespace in the report. If the linear regression variables are reasonable, then the tablespace goes to the “Fits Linear Regression” portion of the report. Otherwise, it goes to the “does not Fit Linear Regression” portion. Flexibility is a key to this technique – everything from forecast and look back periods to the definition of “reasonable” regression variables can be changed with little effort.

Tablespace Growth Fits Linear Regression Model

Forecast Parameters	Database Name	Tablespace Name	Current Size	Forecasted Size	Range for Forecasted Size	Range for Forecasted Size (%)	Suggested Size	Estimate of Normality	Estimate of Autocorrelation	Estimate of Heteroscedasticity	Estimate of Determination
90 days look back - 90 days forecast - 180 days sizing goal	DRRP	DRRP_DATA	83,100	98,199	+/- 26654	+/- 27%	128,354	0	0.65	0.27	0.54
90 days look back - 90 days forecast - 180 days sizing goal	ZFFO	LFD_DATA	16,000	17,645	+/- 3182	+/- 18%	25,518	0.83	0.56	0.05	0.87
90 days look back - 90 days forecast - 180 days sizing goal	ZFFO	LFD_INDX	3,100	3,589	+/- 783	+/- 21%	5,282	0.87	0.41	0.01	0.83
90 days look back - 90 days forecast - 180 days sizing goal	ZFFO	POSI_DATA	344,177	610,715	+/- 77604	+/- 12%	903,460	0.21	0.54	0.23	0.94
90 days look back - 90 days forecast - 180 days sizing goal	ZFFO	DMWW_DATA	19,000	20,224	+/- 3147	+/- 15%	25,342	0.25	0.71	0.28	0.74
90 days look back - 90 days forecast - 180 days sizing goal	ZFFO	SYSTEM	20,520	34,678	+/- 14119	+/- 40%	54,144	0.23	0.74	0.08	0.67
90 days look back - 90 days forecast - 180 days sizing goal	WPPT	TXCD	150	153	+/- 33	+/- 21%	197	0.55	0.54	0.08	0.64

Tablespace Growth Does not Fit Linear Regression Model

Forecast Parameters	Database Name	Tablespace Name	Current Size
90 days look back - 45 days forecast - 180 sizing goal	LSPPR	TEMP	1,000
90 days look back - 45 days forecast - 180 sizing goal	LSPPR	UNDOTBS1	516
90 days look back - 45 days forecast - 180 sizing goal	JBOB	TEMP	58
90 days look back - 45 days forecast - 180 sizing goal	FQQ	TEMP	14,300

Figure 8. Sample Output of Forecasting Report

Conclusion

Troubleshooting performance problems and analyzing resource usage could be a challenging job. We need all the reliable information we can get to be effective in those endeavors. OEM repository data is centrally stored, readily available, and easy to use, making it a valuable data source for a variety of purposes.

Even though OEM comes with considerable functionality to report on the vast amounts of data it collects, we can benefit greatly by working directly with the underlying tables and views. We can utilize the OEM repository not only to support operational DBA activities, but also to build advanced resource management utilities.

704

Number of Oracle Business Intelligence Partners

1

Number with Oracle BI Pillar and Sun Partner of the Year status

CORPORATE TECHNOLOGIES

Oracle BI Pillar Partner, Northeast
Sun 2008 Partner of the Year

Find out about our free on-site OBIEE workshop!

CORPORATE TECHNOLOGIES

ORACLE CERTIFIED PARTNER

Just Get it Written: Deploying Applications to WebLogic Server Using JDeveloper and WLS Console

Peter Koletzke, Quovera and Duncan Mills, Oracle Corporation

> *Don't get it right, just get it written.*
>
> —James Thurber *(1893–1961)*,
> *The Sheep in Wolf's Clothing*

> *However beautiful the strategy,*
> *you should occasionally look at the results.*
>
> —Winston Churchill (1874–1965)

After you create and debug the code for your Java Platform, Enterprise Edition (Java EE) web application, you will probably need to install it on a test server. The term *deployment* refers to the process of copying and installing the necessary application, library, and configuration files into a specific server runtime environment.

Although your organization may require that a group other than developers is in charge of deployments, the developer is ultimately responsible for ensuring that the deployment package is complete and bug free. Therefore, once you have completed development and unit testing for the application on a local (desktop) server, you (or someone else) will deploy the application to an environment that matches a production situation for further testing; this deployment will test the ability of your application to run on an external server. It will also test the completeness of the deployment; for example, it will help you understand if all files necessary to run the application are available and if the application's library versions work in the runtime environment.

Deployment is not a one time task in an application's life cycle. After the test deployment just mentioned, your organization may require other levels of testing (for user acceptance tests or volume tests), and these tests will require additional preproduction deployments. After all testing is complete, the application will be deployed into the production environment. But even this is not the end. When an application is enhanced or its bugs are fixed, the deployment process will need to be repeated to make the new application code available.

One objective of this white paper is to introduce deployment principles, so that you can be effective in creating (and possibly deploying) deployment files. It assumes you now have, or will soon have, a working familiarity with the Java EE runtime environment and application files. Another objective of this white paper is to show the techniques used to deploy an application to Oracle WebLogic Server (WLS) using JDeveloper or the WLS console. Since this objective is best met with an actual demonstration, we fill the last part of this white paper with a hands-on practice where we show—and, hopefully have you try out—the steps required for these two deployment methods.

Deployment Principles

Deploying an application to a Java EE server requires these two stages:

1. **Prepare the deployment file** This stage consists of collecting application files and supporting libraries into a single Java archive (JAR) file sometimes called "the deployment file." So that it can be properly handled by a Java EE-compliant server, such as WLS, the archive file must contain a standard structure with appropriate configuration files.
2. **Copy the deployment file to the application server** In this stage, you use deployment tools (JDeveloper or the

WLS console) to upload the archive file to the server and then allow the server to install the application in the proper directory. The application will be available to users at the end of this process.

Java Archive Files

Java EE defines standards for deployment file packages and locations as well as for standard configuration files expected by an application server that is Java EE–compliant. Using Java EE configuration and archive files, any Java EE–compliant server can find the proper files and process them in a standard way.

> **Note**
>
> This white paper uses as an example an Application Development Framework (ADF) application created in JDeveloper 11g as a subject for deployment. Although ADF applications are deployable to any Java EE-compliant server, the server must support JavaServer Faces (JSF) 1.2. WLS 10.3 and beyond fulfills this requirement.

The first step in deploying an application is gathering together potentially thousands of application and library files that are located in various directories. Java EE defines standard *archive files,* single files that contain more than one file and directory. Tools such as JDeveloper help you assemble the files into these archives. Each file contains one or more files in one or more directories. A single archive file can contain the equivalent of one or more file system directories (with nested subdirectories) and files. The JVM can access files inside an archive file in the same way as if they were located in a real file system directory. Any archive file can be viewed with an archive viewer (such as the Archive Viewer in JDeveloper) and manipulated with an archive file program (such as WinZip or 7-Zip). The Java archive files used for Java EE deployment are web application archive and enterprise application archive files. In addition, another archive—metadata application archive—is required for Metadata Services (MDS) customizations in Fusion applications.

Web Application Archive (WAR)

Also called "web archive," a *web application archive,* or *WAR,* file is an archive file that contains all files required for the application's runtime. The WAR file contains all JSF files in your application with the directories set up in the ViewController project. The WAR file also contains a number of files and directories inside a WEB-INF directory. These files are a combination of standard Java EE XML descriptor files (such as web.xml) and possibly library JAR files that support the application. The deployment process expands the WAR file into its component files and directories. A copy of the WAR file is kept in the project root directory. WAR files are named with a "war" extension.

> **Tip**
>
> ADF applications depend on many individual JAR files which are installed as shared resources on the target application server. These libraries of shared JAR files are referenced from a common location and are not packaged within the WAR file. A list of these shared JAR files and the libraries that they are included in is available in the *Oracle Fusion Middleware Administrator's Guide 11g Release 1* (www.oracle.com/pls/as111120/portal.portal_db).

Enterprise Application Archive (EAR)

Also called "enterprise archive," an *enterprise application archive,* or *EAR,* file is an archive file used for standard Java EE deployments. It provides a single archive that contains all other archive and other files needed for an entire enterprise (many applications). The EAR file can contain one or more WAR files, JAR files, and EJB JAR files, as well as several deployment descriptor files. EAR files are named with an "ear" extension.

Metadata Application Archive (MAR)

The MAR file is an optional format that is not part of the Java EE standard and is specific to ADF based applications that contain metadata customizations. The MAR is essentially the same as an EAR file except that it contains all of the seeded MDS customizations for an application as well. The WebLogic container has special logic to correctly install applications in a MAR file and will both install the Java EE portion of the application and publish the MDS metadata to the metadata repository. In addition to WAR, EAR, and MAR files, a deployment will likely include normal JAR files, which contain supporting library code (and use a .jar file extension).

> **Note**
>
> Although some Java EE applications can be deployed as a simple WAR files without being embedded in an EAR, ADF applications will generally deploy the EAR or MAR files.

Hands-On Practice: Deploy a Java EE Web Application

This hands-on practice provides the steps for deploying an existing Java EE application written with Oracle ADF technologies. The practice steps refer to an application —the second version of The Ultimate Human Resources Application (TUHRA or TUHRA2)—developed for the authors' book *Oracle JDeveloper 11*g *Handbook*. Although the practice refers to specific features and files in this application, you can use an application of your own and adjust appropriately.

Should you want to follow the steps exactly as written, you can start with the same sample application referred to in this practice. Start by pointing your web browser to tuhra2.samplecode.oracle.com. (The samplecode website requires a free Oracle Technology Network login, which you may already have.) Then click the Downloads tab and download the file for Chapter 21, Phase IV. Use this sample application as a starting point for this hands-on practice.

> **Note**
>
> This practice was developed with JDeveloper, version 11.1.1.1.0 (build 5407). If you use a later release, you may need to adjust steps to changes introduced with that later release. However, the principles and steps are the same.

For the purposes of this hands-on practice you will use the existing standalone WebLogic software that is part of the JDeveloper installation. Although the process followed in this hands-on practice uses an existing WebLogic does not employ a full application server, it very closely emulates the process used for deployment to a full application server environment. As mentioned, although in many organizations it is not the responsibility of the developer to finally deploy applications into production, testing the deployment is nevertheless an essential step in the testing of the application before you hand it over to those who will perform the deployment.

This practice follows these phases:

I. Set up a standalone WebLogic server for testing

- Create and run a WebLogic domain
- Run the Oracle WebLogic Server Administration Console

II. Configure application-specific settings

- Define database connection information
- Set up users and groups

III. Deploy the application

- Deploy the application from JDeveloper
- Deploy and install the application from an EAR file

I. Set Up a Standalone WebLogic Server for Testing

Within your JDeveloper 11*g* installation, you already have all of the pieces required to set up a standalone WebLogic server for testing. When you run the TUHRA application from within JDeveloper, the IDE takes various shortcuts to speed up the runtime and so does not exactly emulate the final deployment environment with the application server. Therefore, this additional step to deploy to a standalone server is a valuable one because, if you've tested an EAR file by deploying it to a standalone server, the hand-off process for the final production environment will be smoother.

Note

You can read more information about the WebLogic Server in the online manual "Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server 11g Release 1 (10.3.1)," currently available at download.oracle.com/docs/cd/ E12839_01/web.1111/e13702/toc.htm.

Tip

Links to specific on-line documents may change from time to time, however, you can always start from otn.oracle.com and click **Documentation | Fusion Middleware** on the shortcuts menu bar to get to the very latest documentation on WebLogic.

Create and Run a WebLogic Domain

When you set up a server environment in WebLogic, you need to create a *WebLogic domain*, one or more servers, all configured with a certain set of shared libraries. As shown in Figure 1, on the left, a WebLogic Server (WLS) software installation on a single physical server machine may support multiple WLS domains configured for ADF, or, on the right, a single ADF WLS domain may be distributed across many physical servers, each having a WLS installation.
As a further twist, the architecture supports two kinds of servers: *administration servers,* used to control and monitor other servers but not intended to run applications, and *managed servers,* used to run applications. In a production domain there would generally be one administration server and many managed servers, to which applications would be deployed. For local testing purposes, it's fine to just create a single administration (admin) server and use that server for deployment.
The instructions in this section are based on the use of the command line rather than the Windows Start menu, although instead you can use the Windows Start menu options (in the **Start** | **Programs** | **Oracle Fusion Middleware 11.1.1.x.x** | **WebLogic Server 11gR1** | **Tools** program group). In all cases the scripts mentioned will have two versions, a ".cmd" version for Windows and a ".sh" version for Unix/Linux. Use the version that applies to your platform and the appropriate path separator in your commands (";" for Windows and ":" for Unix/Linux).

1. If JDeveloper is open, be sure to stop all server instances (default and application) using the Terminate (red box) button in the toolbar. (Stopping the default server will automatically stop any running application instances.)
2. Open a command shell (command-line) window and change (CD) to the directory into which you installed Oracle Fusion Middleware. In these examples, we use "FMW_HOME" to represent the directory into which you installed JDeveloper (for example, C:\Oracle\Middleware).
   Additional Information: In Windows, you can open a command-line window by selecting Start | Run, entering "CMD," and clicking OK.
3. Change to the <FMW_HOME>\jdeveloper\common\bin directory and run the config.cmd script. The Oracle WebLogic Configuration Wizard will start. (You can alternatively run this utility using the Configuration Wizard selection in the Start menu program group mentioned earlier.) The dialog in Figure 2 will appear.

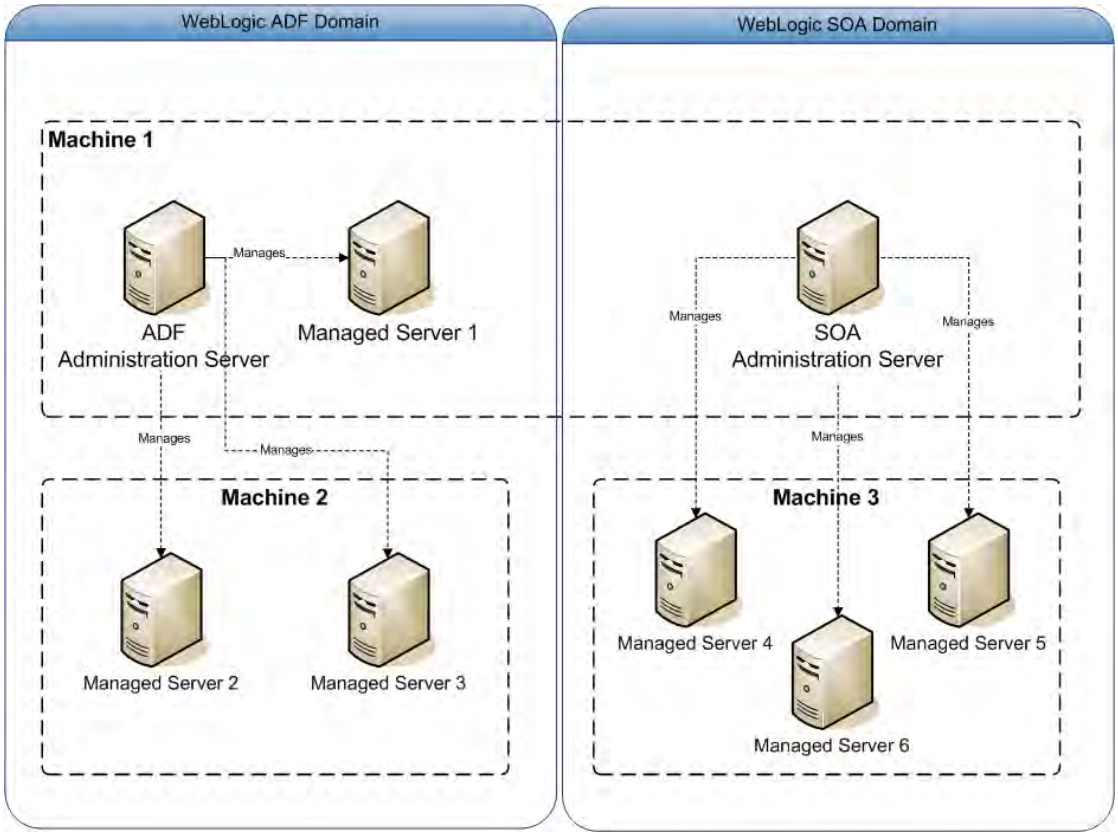


Figure 1. Domains in a WebLogic Server

4. If no existing standalone domain exists, leave the *Create a new WebLogic domain* option selected and click Next.
5. The Select Domain Source page will appear with a list of possible products (libraries) to configure into the domain. Select the *Oracle JRF* checkbox and click Next.
   Additional Information: The Oracle JRF (Java Runtime Foundation) contains all of the shared JAR files used by ADF BC and ADFm; it also includes the ADF Faces components.
6. On the Specify Domain Name and Location page, set *Domain name* as "adf_test" and leave the *Domain location* as the default (for example, "<FMW_HOME>\user_projects\domains"). Click Next.

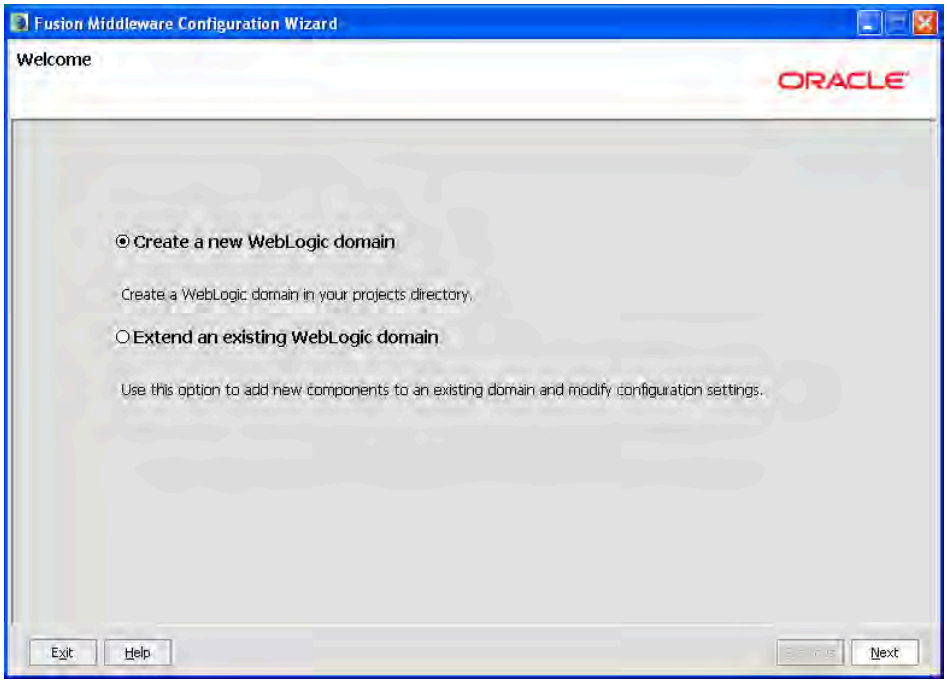


Figure 2. Welcome Page of the Fusion Middleware Configuration Wizard

7. The Configure Administrator User and Password page will appear where you define the account used to manage the server. By convention the default user name used for the administrator is "weblogic," and that default will work. Set User password and Confirm user password to "weblogic1" (the password must contain a number). Click Next.
8. The Configure Server Start Mode and JDK page will appear. Ensure that the WebLogic Domain Startup Mode is set to "Development Mode," and accept the default for JDK Selection, which should be the same JDK version with which you are running JDeveloper (for example, "1.6.0_11"). Click Next.
9. The Select Optional Configuration page will appear. You will not need any of these options, so just click Next.
10. The Configuration Summary page will appear as in Figure 3. Review the settings. You can return to earlier pages using the Previous buttons.
11. Once you have reviewed and confirmed the settings, click Create. The Creating Domain progress page will appear.
12. When the creation process completes, select the Start Admin Server checkbox in the lower-right corner of the dialog and click Done. The Configuration Wizard will exit, and another command-line window will open and display messages about the server startup. Wait for the message that ends with "<Server started in RUNNING mode>."

Caution

For now, leave the server command-line window. Closing this window will stop the server.

13. The original command-line window from which you started the Configuration Wizard will return to a command-line prompt. Close this window (but not the server window).

Tip

If you accidentally close the server window, or forgot to start the server at the end of the configuration wizard, jump ahead to the sidebar "Starting the WebLogic Domain Server from the Command Line" for instructions on starting the server.

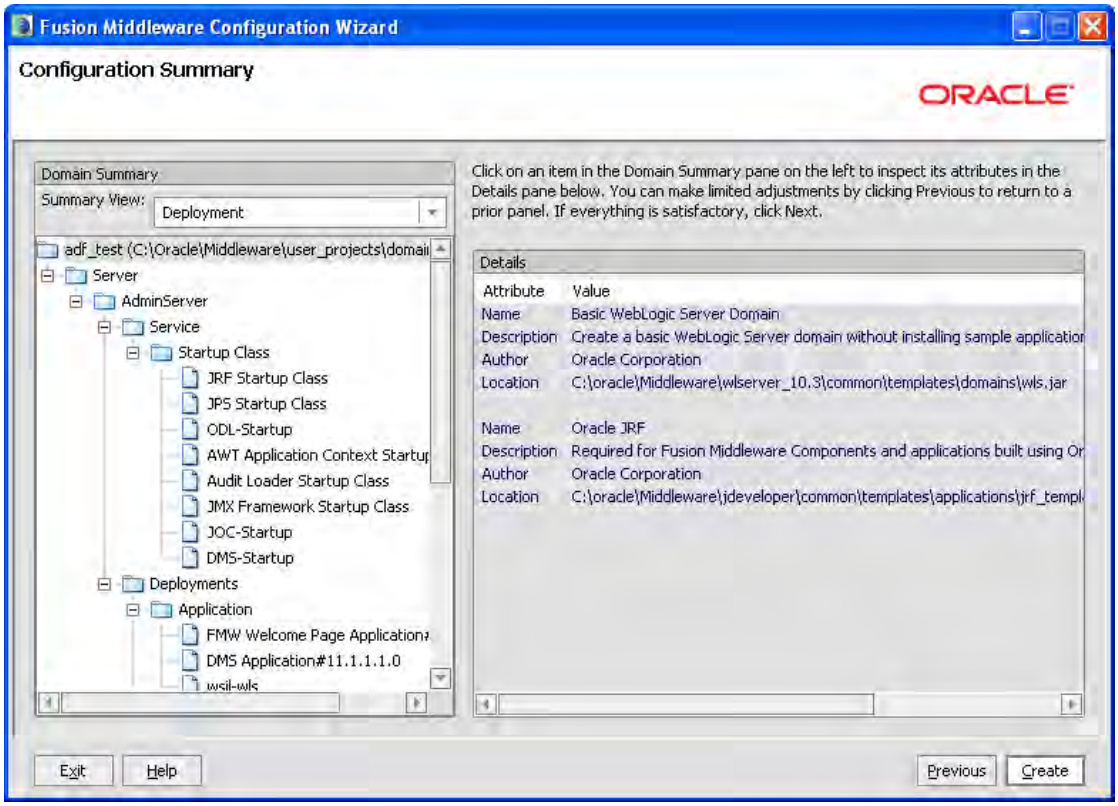


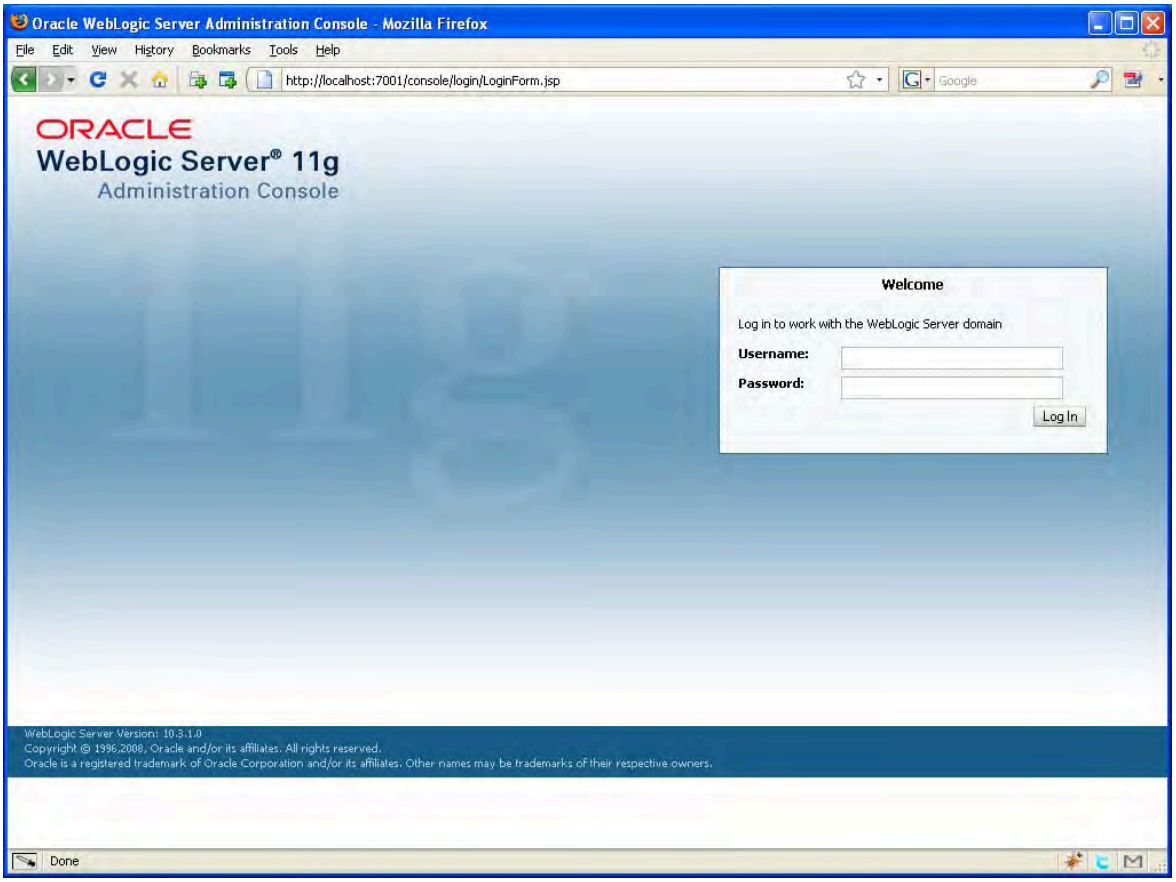
Figure 3. Configuration Summary Page of the Configuration Wizard

Run the Oracle WebLogic Server Administration Console

Next you need to configure some key attributes to allow the TUHRA2 application to successfully deploy and run—specifically, connection information for the database and a list of valid users for the security was added to the application . (Refer to the authors' white paper "WebLogic Server Application Security— Implementing the Superstition in JDeveloper" for more information about application security.) This configuration is performed using the Oracle WebLogic Server Application Console ("the console").

> **Tip**
>
> A single WebLogic admin server can be used for multiple concurrently running server instances. If you want to set up multiple servers, you would fill in some of the Configuration Wizard screens to assign a unique port number to each server on a physical machine. The default port is 7001.

1. Open a web browser and navigate to the URL: http://localhost:7001/console. This will launch the WebLogic console application, which is installed into every administration server. You will see a message saying "Deploying application for /console", then the following login screen will display.



2. Log in to the console using the user name and password that you specified earlier (weblogic/weblogic1). The administration console page will then display as shown in Figure 4. Notice the domain name in the top right-hand corner ("Connected to") area.

What Did You Just Do?

The configuration wizard that you ran in this phase creates a set of configuration files and scripts, which will start a WebLogic Server instance. This server is configured to use the default port: 7001, which is different from the port that is used when running within JDeveloper (port 7101), so you can safely use both the internal server and this new external testing server at the same time. The domain that you have created is set up with all of the libraries required to run an ADF application.

In this phase, you also ran the console, the tool you will use for the rest of the configuration tasks. Before you run the console, you need to start the server. The steps in this phase ran the server from the Configuration Wizard. The next time you need to start the server, you will be using the command line batch file or shell script (or shortcut) as described in the sidebar "Starting the WebLogic Domain Server from the Command Line."

What Could You Do Next?

You can also access the administration console for the default server in JDeveloper. Once JDeveloper starts the default server (after you run an application, for example), use the default server URL (http://localhost:7101/console) to start the console. The user name and default password are the same as those used in this practice (weblogic / weblogic1).

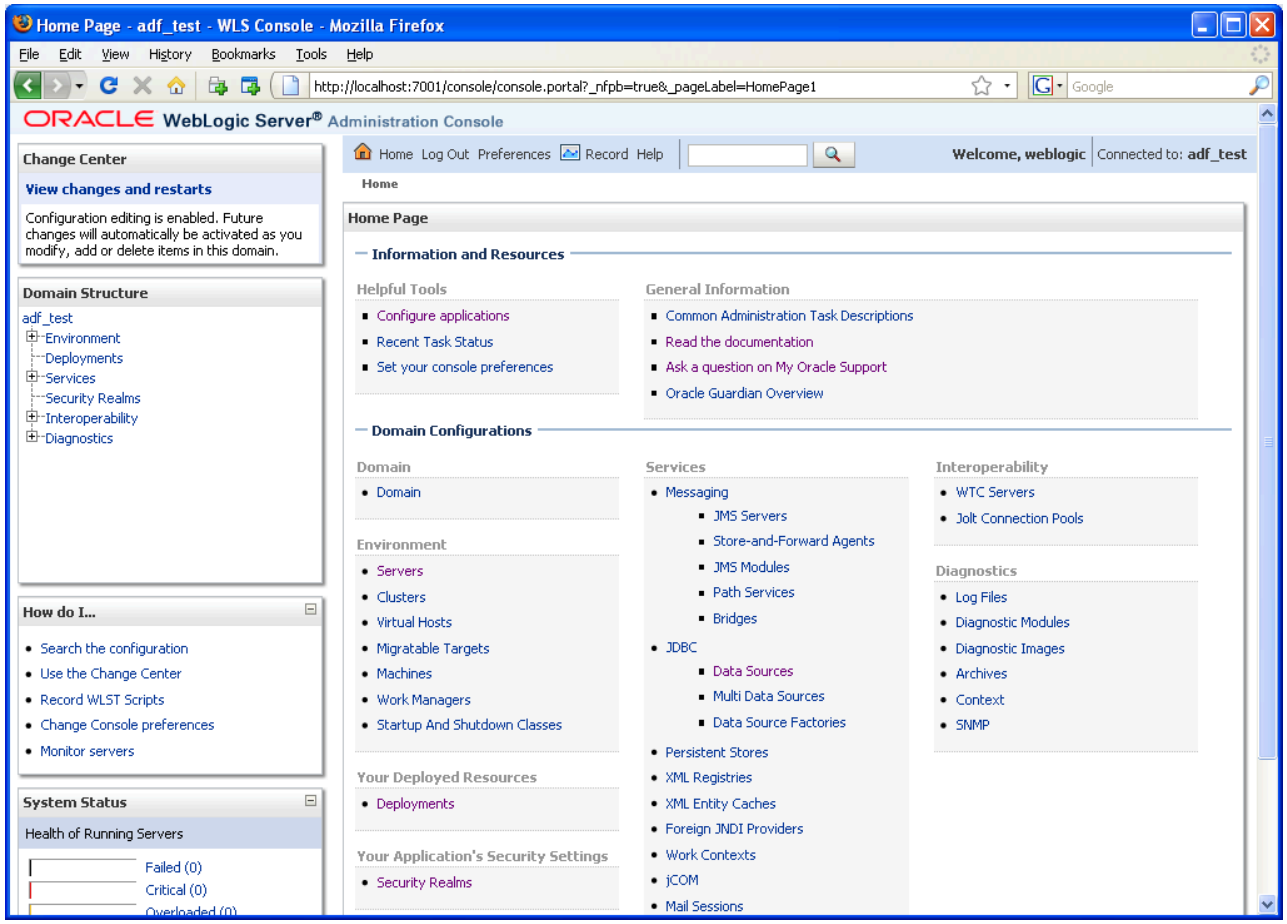


Figure 4. Oracle WebLogic Server Administration Console for the adf_test Domain

Starting the WebLogic Domain Server from the Command Line

The last step of the Oracle WebLogic Configuration Wizard allowed you to start the server (domain). If you are not running the wizard, you can manually start up the server using the following steps. If the server is running and you want to try these steps, press ctrl-c, then enter "Y" at the prompt to stop the server, and then close the command-line window.

1. From a new command-line window, navigate to the FMW_HOME directory and then navigate to the user_projects\domains\adf_test. Notice that "adf_test" corresponds to the name and location of the domain that you created earlier.
2. At the command line, run the script startWebLogic.cmd (or .sh) script to start the server instance. As before, wait for the "Server started in RUNNING mode." message. Leave this window open. Reconnect to the console as described earlier.
3. To stop the server you can run the corresponding stopWeblogic.cmd (or .sh) script in the adf_test\bin directory rather than pressing ctrl-c in the server window.

Tip

As with any frequently used program, you will probably want to create a shortcut on your desktop for startWebLogic.cmd (Windows) or startWebLogic.sh (Unix or Linux).

II. Configure Application-Specific Settings

The steps in the preceding phase centered around tasks you need to complete regardless of the application. This phase focuses on tasks that are more specific to an application, for example, database connection information and users and groups.

Define Database Connection Information

Running the TUHRA2 application from within JDeveloper requires a database connection called HR_DB. The sample application altered its ADF BC configuration "TUHRAServiceLocal" to make the application use the HR_DB connection information in a JDBC data source rather than as a hard-coded user name and password. This *JDBC data source* mechanism allows you to abstract the connection information and hold it separately from the application definition. *Data sources* are a named connection resource that the application requests from the application server for rather than holding the connection information directly. This feature allows the application to remain unchanged as it is moved from server to server, for example, test to production. In this section you will define this data source in the adf_test WebLogic domain.

1. Return to the WLS console (http://localhost:7001/console). In the Domain Structure (left-hand side) region expand the Services\JDBC node and click the Data Sources node. The screen will change to look like Figure 5.
   Additional Information: The Multi Data Sources node in the services section is used when configuring a data source to work with an Oracle Real Application Clusters (RAC) database. The "Data Source Factories" page is used only for backward compatibility.
2. Click New (above the Data Sources table) to launch the Create a New JDBC Data Sources wizard, which you will use to create a data source. On the first page of the wizard, set the values as follows:

Field	Value	Purpose
Name	TuhraDS	A name that identifies this resource in the console.
JNDI Name	jdbc/HR_DBDS	The name that is used to look up the connection. This value matches the value defined in the ADF BC configuration without the java:comp/env prefix.
Database Type	Oracle	This is the default; it assumes you're accessing an Oracle database.
Database Driver	Oracle's Driver (Thin) for Service Connections	The correct driver for a local XE or local Oracle 11*g* database installation. Do not select the "Thin XA" driver.

3. Click Next. On the next page leave all options as their defaults and click Next.
4. Fill in the fields on this page as in the following table to define the core connection information. The following values are based on a default Oracle XE database installed on the local machine; adapt these if you are using a different database type.

Field	Value	Purpose
Database Name	XE	The SID or service name of the database ("XE" for Oracle XE and "ORCL" for an 11*g* install, by default).
Host Name	localhost	This setting will loop back to your local machine; alternatively use the database machine's host name, if any.
Port	1521	The default port for an Oracle database.
Database User Name	hr	The schema owner (HR).
Password	hr	Use the password for hr on your system.
Confirm Password	hr	The same as the password.

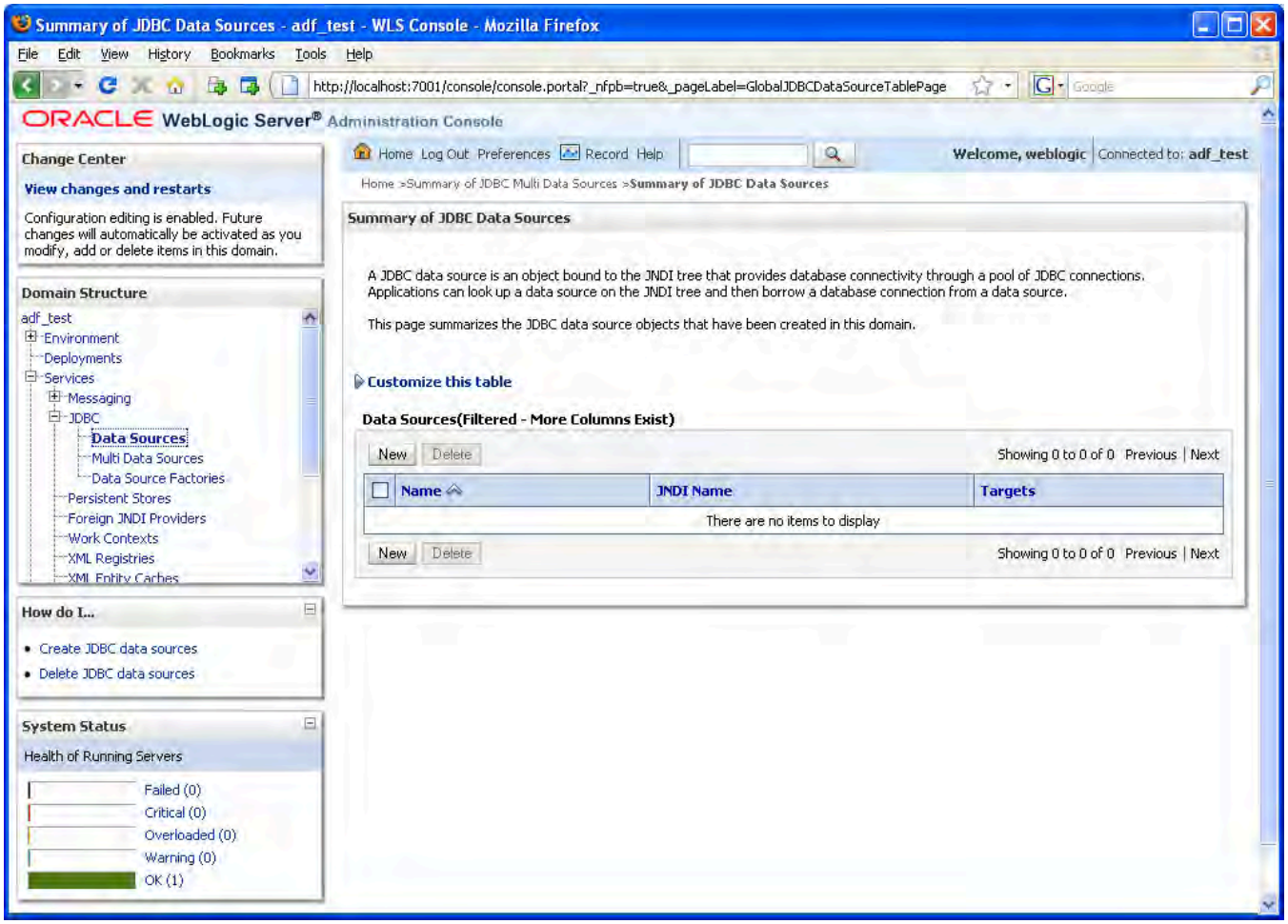


Figure 5. Summary of JDBC Data Sources Page of the Console

5. Click Next. The screen will change slightly adding fields at the bottom, one of which, *Test Table Name*, allows you to enter a sample SQL statement (one is entered by default) for testing purposes.
6. Be sure your database is started and click the Test Configuration button to check that the database is accessible. A green "Connection test succeeded" message will appear near the top of the screen if the connection is correctly defined and the database is reachable.
   Additional Information: If the connection fails, cross-check the settings on all pages with the settings in the preceding steps and with the HR_DB database connection information in JDeveloper.
7. Click Next to display the Select Targets page. At this point, you need to activate the data source for the administration server. Select the *AdminServer* checkbox.
   Additional Information: If you miss this step or need to deactivate the data source later, you can click the data source link after displaying the list of data sources, select the Targets tab, select (or deselect) the target server, and click Save.
8. Click Finish. The wizard will set up the data source and return you to the Data Sources list, which now contains TuhraDS.

Set Up Users and Groups

The next configuration task is to configure the WebLogic server's LDAP service for the users and roles that TUHRA2 needs. When you run the application from within the IDE, JDeveloper takes care of this for you, but running on a standalone server requires some setup steps.

1. In the WebLogic console's Domain Structure navigator area, click Security Realms to display a single predefined realm called "myrealm." Click myrealm.

> **Note**
>
> Remember that a realm in security is just a namespace. It does not matter that the realm you are using here is different from the default (jazn.com).

2. Click the Users and Groups tab in the "Settings for myrealm" page and click New to define each of the application users (CDAVIES, NKOCHHAR, and TFOX, all with passwords of "welcome2tuhra"). The screen should appear as in Figure 6.
   Additional Information: Feel free to add descriptions (as shown in Figure 6) and create extra users as well, as long as you assign them to the correct groups, as you will do for the main users in the next steps.
3. Click the Groups subtab.
4. Click New on the Groups table to add a group called "cs_admin" that will represent the application's admin group. Provide a suitable description. Click OK.
   Additional Information: Naming the server group differently from the application's logical roles helps illustrate the kind of mapping that will often have to take place when using existing credential stores such as Oracle Internet Directory (OID). Each of the server's LDAP credential store groups will correspond to one of the local server groups you set up when testing security from within JDeveloper.
5. Repeat the preceding step to define groups for cs_manager and cs_user. Provide a suitable description for each group.
6. Click the Users subtab.
7. Click CDAVIES to display the "Settings for CDAVIES" page. Click the Groups subtab. Shuttle "cs_user" from Available to Chosen. The screen will contain the areas shown in Figure 7.
8. Click Save. Click the "Users and Groups" link in the breadcrumb area (above the title "Settings for myrealm").
9. Repeat the preceding two steps to assign NKOCHHAR to the cs_manager group and TFOX to the cs_admin group.
10. Click Home (under the WLS Server banner at the top of the page) to display the main console page.

What Did You Just Do?

In this phase you have carried out some essential configuration work. First, you ran the Oracle WebLogic Server Application Console, where you can perform configuration and administrative tasks for the server instance. You defined a JDBC data source for the application to use for its database connections. Then you created a set of groups and test users in the internal LDAP server provided by Oracle WebLogic so that security in the deployed application can be tested. You also assigned users to groups.

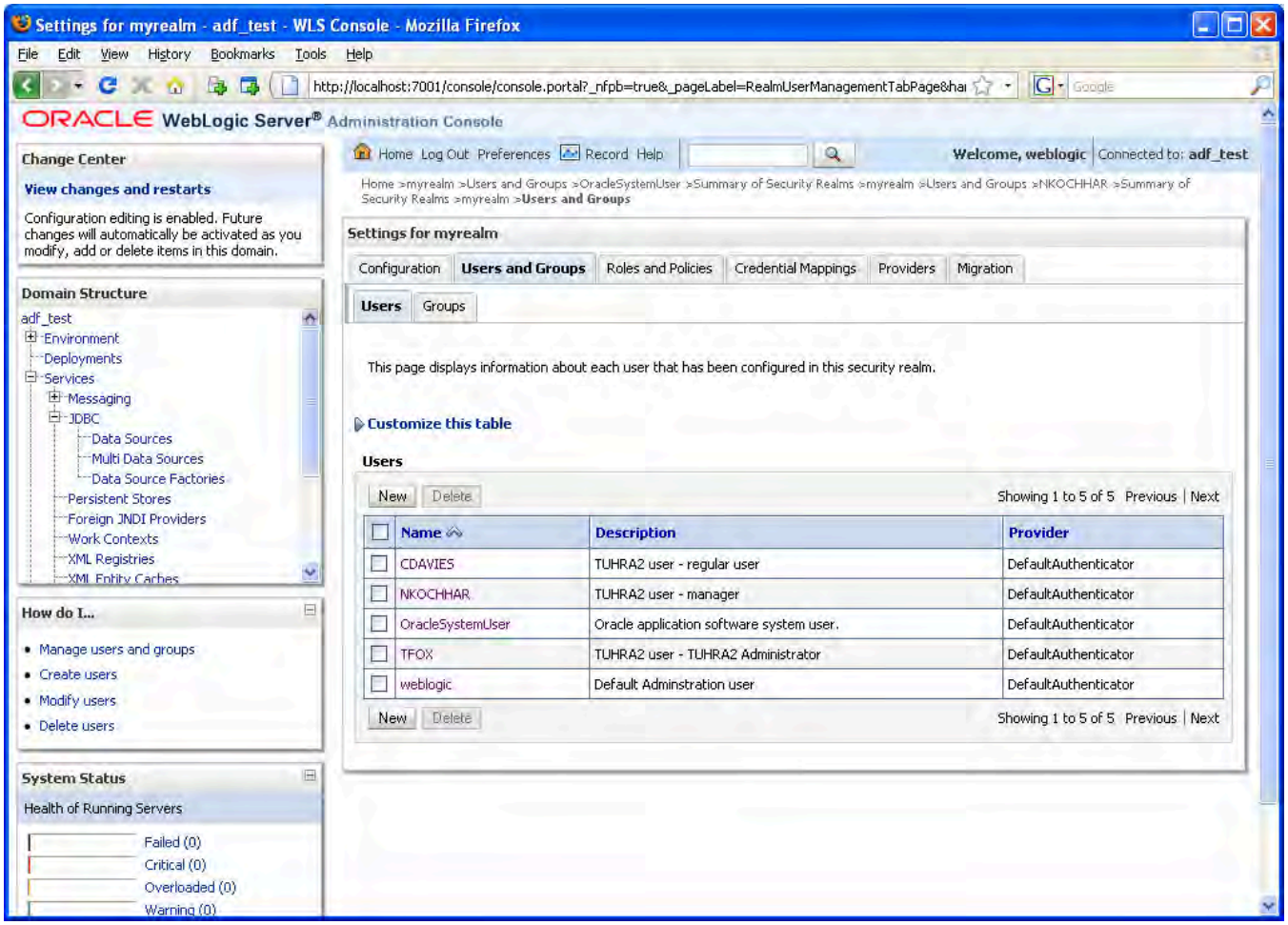


Figure 6. Users and Groups Tab of the Security Realm Page

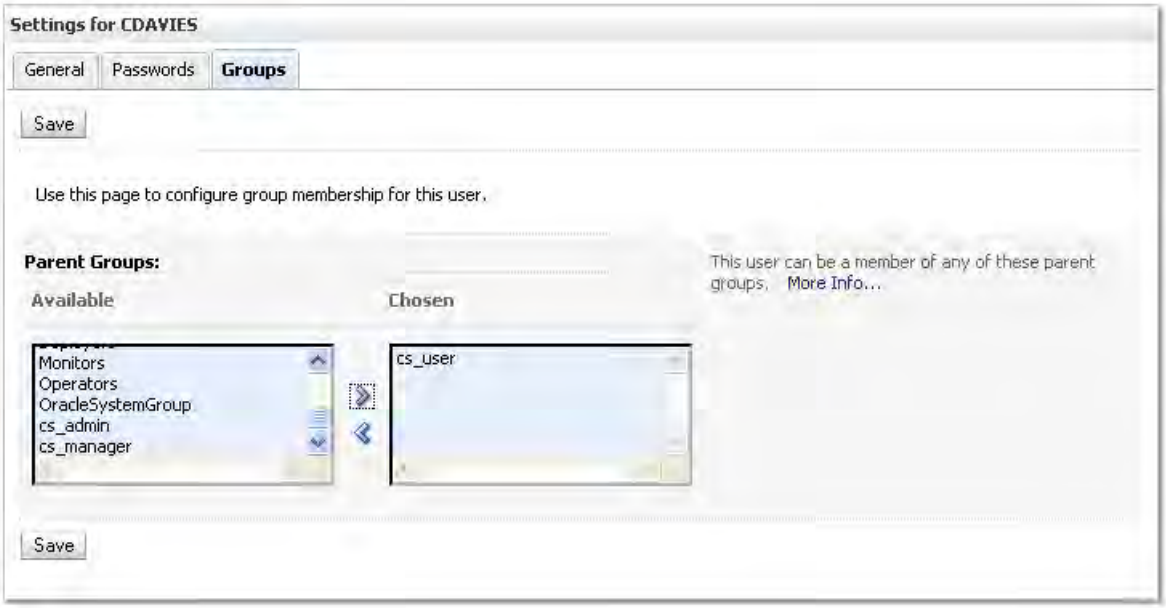


Figure 7. Assigning Groups to a User

III. Deploy the Application

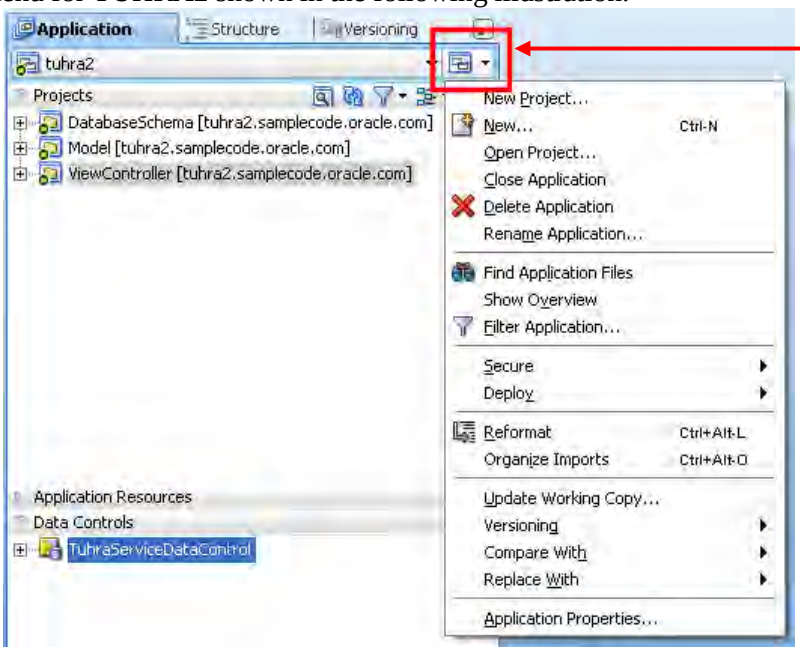
Now that the server is set up, you can deploy the TUHRA2 application to it. This section will actually explain two alternative forms of deployment: deployment directly from JDeveloper, which you would use to test the deployment on a local WebLogic Server instance, and deployment using an EAR file, which is the technique that you or an administrator would usually use to roll out an application into a full application server environment.

> **Note**
>
> If you have been using an earlier version of ADF security, you may be familiar with the concept of having to manually migrate security permissions from the application jazn-data.xml file into the application server. One of the big improvements you will see in this version of the framework is that this permission deployment is now handled by the direct deployment from JDeveloper.

Deploy the Application from JDeveloper

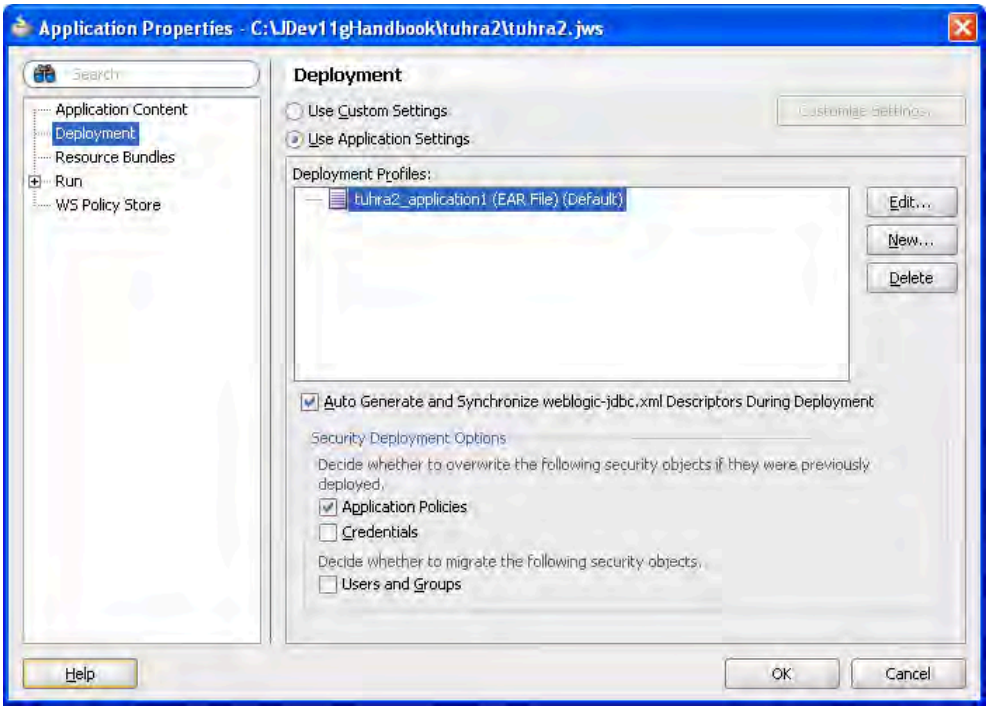
JDeveloper makes the process of deploying to a local or a remote WebLogic server, a relatively easy, point-and-click task.

1. In JDeveloper, with the TUHRA2 application open in the Application Navigator, display the project properties for the ViewController project (by double clicking the ViewController node). On the Java EE Application page, ensure that Java EE Web Application Name and Java EE Web Context Root are both set to "tuhra2."
2. Click OK.
3. Click the application menu for TUHRA2 shown in the following illustration:



Application menu

4. Select Application Properties (also available from **Application | Application Properties**), and in the Deployment page ensure that the *Application Policies* checkbox is selected and the *Auto Generate and Synchronize weblogic-jdbc.xml Descriptors During Deployment*, *Credentials*, and *Users* and *Groups* checkbox are unselected as shown here:

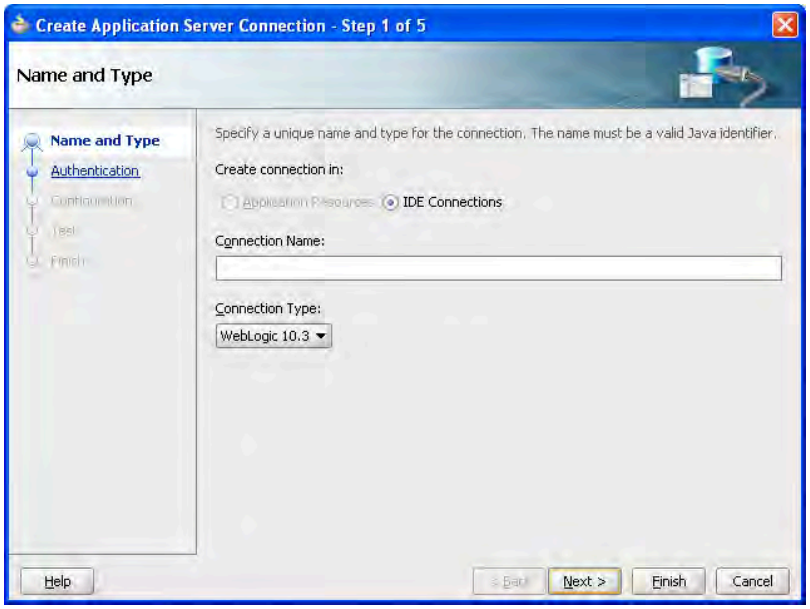


Additional Information: You have already explicitly defined the database connection, credentials, users, and groups on the server side and do not need to move those items.

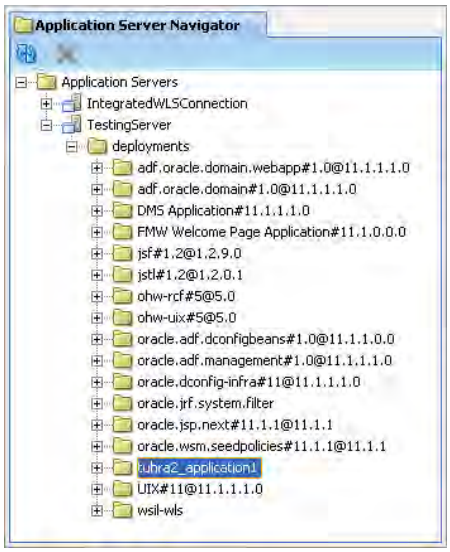
Caution

If you fail to deselect the *Auto Generate and Synchronize weblogic-jdbc.xml Descriptors During Deployment* option the final deployment from an EAR file in Phase III will fail. This is because this option embeds a separate data source definition for the application with the EAR file but cannot embed the password for it (the password has to be encrypted after installation onto the target WebLogic server). It is possible to complete this data source definition from the WebLogic console after installation; however, because you have already defined a global data source it is much simpler to avoid the whole problem by not generating the extra application level data source. Note that you will need to re-select the Auto Generate and Synchronize weblogic-jdbc.xml Descriptors During Deployment option if you want to run the application again in the embedded server.

5. Click OK to dismiss the dialog.
6. Be sure the WebLogic server is running from the preceding phase. If it is not, return to the sidebar "Starting the WebLogic Domain Server from the Command Line" in Phase I and start the server.
7. From the Application menu, select **Deploy** | **tuhra2_application1** | **to** | **New Connection**. The Create Application Server Connection dialog will appear as shown next:



8. Set the *Connection Name* as "TestingServer." Leave the *Connection Type* as "WebLogic 10.3." Click Next.
9. On the Authentication page, provide the user name and password, which you used to log in to the WebLogic console for the adf_test domain (weblogic/weblogic1). Click Next.
10. .On the Configuration page, leave all of the defaults except for setting the WLS Domain value to the name of your domain, "adf_test." Click Next.
11. Click Test Connection and wait for the test to complete. All of the tests should succeed. Click Finish.
    Additional Information: In the future, you will be able to select this connection from the Deploy menu option without having to step through this connection wizard.
12. The deployment will proceed, and you will see messages appear in the Log window.
    Additional Information: If more than one server is defined for the domain, an additional popup dialog allows you to select the target server.
13. When the process is complete, the Log window will display "Deployment finished."
14. Open the Application Server Navigator (View | Application Server Navigator). Expand the Application Servers\TestingServer\ deployments node to view all deployed applications as shown on the right with the TUHRA2 deployment highlighted.
    Additional Information: The right-click menu on a deployed application node allows you to undeploy the application.
15. Test the deployed application by opening a browser window and entering the URL http://localhost:7001/tuhra2/faces/employeeSearch into the browser location field.
    Additional Information: The Employee Search page will appear. Alternatively, you could test the login.html page to confirm that the login works in this server arrangement.
16. Test various application functions and close the browser when you are finished testing.



Exploring the EAR File

As mentioned earlier in this white paper, an EAR file is the standard deployment unit for Java EE applications; it contains all the bits of the application that have to be deployed except for the connection information and security setup, which are held in the server.

The EAR file created by this process was created, copied to the server, and deployed automatically by the JDeveloper deployment process. However, you can open and examine the contents of this file using the following steps:

1. In JDeveloper, look in the Deployment log window for a message about the EAR file that was created ("Wrote Enterprise Application Module to…"). Notice the file system location for the EAR file. Select File | Open and navigate to that deploy directory under TUHRA_HOME.
   Additional Information: The deploy directory is parallel to the Model and ViewController directories. You will see two nodes for the one TUHRA2_application1.ear file. One node (displayed as a file icon) allows you to open the EAR as a file; the other node (displayed as a directory icon with a "?") allows you to look inside the EAR file and select one or more files.

> **Tip**
>
> You can copy the file system location from the Log window and paste it into the Open dialog instead of navigating through the file system.

2. Select the version of the EAR file that displays as a file (page with lines) icon and click Open. The list of EAR file contents should open in the Archive Viewer. The list of files in the EAR file is mostly configuration files and the WAR file, tuhra2_ViewController_webapp1.war.
3. Double click the WAR file to examine its contents. You should see familiar files as you browse this list, for example, Model project files that are required for the application, library JAR files, and ViewController files (JSF files and images).
4. You can open any of these files in an editor or viewer window by double clicking the file name. Close the Archive Viewer when you are finished browsing.

Deploy and Install the Application from an EAR file

Although the approach of deploying the application directly from JDeveloper will work successfully for both local and remote application servers, many organizations would prefer that their developers not directly update production servers. In these cases the developer will need to create an EAR file of the application to hand over to the administrator for deployment.

Most Java EE application servers (with exceptions such as Apache Tomcat, which is not a complete Java Enterprise Edition application server) understand how to automatically install an EAR file. The process of deployment directly from JDeveloper that we explained in the previous section is really just automating the process of packaging the EAR file, copying it to the target server machine, and triggering the installation. However, since most administrators will use manual deployment steps, we will run through that technique in this section. Although you've already created the EAR file as part of the JDeveloper deployment steps before, the following creates it again to demonstrate the steps. If you are in charge of deployment to the production application server, you can use these steps to practice the production deployment steps on a standalone local server.

1. Starting from the JDeveloper Application Navigator open the application menu, and select **Deploy | tuhra2_Application1 | to EAR File**. By default this will package the application into an EAR and copy it to the deploy subdirectory underneath the application directory root. You will see "Deployment finished" in the Log window when the process completes.

Additional Information: You would then pass this EAR file to the application server administrator, who would deploy the application to the server, but the following steps demonstrate the process of deployment using the WebLogic console in your standalone WebLogic server.

2. Next, log in to the WebLogic console (http://localhost:7001/console) using the user name/password you set up (weblogic/weblogic1). From the Domain Structure tree on the left-hand side of the screen, click Deployments.
3. Browse the list of the deployed applications until you find tuhra2_application1 that you deployed from JDeveloper as shown here:

4. Select the checkbox next to this deployment and click Delete to remove it.

> **Note**
>
> This step just demonstrates how to remove a deployment, but it is not really necessary. You could choose instead to update the deployment.

5. Click the Install button in the Deployments table. This will launch the Install Application Assistant, the first screen of which will allow you to navigate to and select the EAR file for the application (tuhra2_application1.ear) from your local disk.
6. Find the file (as mentioned before, the name is provided at the end of the Log window messages in a line starting with "Wrote Enterprise Application Module to"). Select the radio button next to the file name and click Next.
7. On this page, leave the default option of "Install this deployment as an application" selected and click Next.
8. On this page, leave all of the options as default and click Finish. The "Summary of Deployments" page will display.
9. Scroll to the page containing tuhra2_application1 to confirm that the State is "Active" and Health is "OK."
10. Open a new browser session and connect to the application as before (http://localhost:7001/tuhra2/faces/employeeSearch). Test the application.
11. Close the browser and stop the server using the stopWeblogic script or by pressing ctrl-c and closing the server's command-line window.

What Did You Just Do?

In the final phase of this practice, we showed how to deploy the application you developed into a standalone instance of WebLogic Server. In the process you used both automated deployment from JDeveloper as well as the more manual

process using the console to deploy an EAR file. As mentioned, you would only need to deploy an application once, but this phase showed both methods to give you experience with the processes.
When deploying directly from JDeveloper, you also migrated the permissions used by ADF to secure bindings and task flows. The manual migration of these permissions is also discussed in some detail in the JDeveloper online help if you need to understand the steps in more detail.

Summary

This white paper has introduced the principles of deploying a Java EE web application to Oracle WebLogic Server. It described how application files are assembled into the Java archive files used for deployment—WAR and EAR (and occasionally, MAR). It also described the main second stage of deployment—copying the archive file to the server—and how the server expands the file into its components files and directory structures. The white paper then concluded with a demonstration of these principles in the form of a hands-on practice that you can follow using JDeveloper and WLS.

This is not the end.
It is not even the beginning of the end.
But it is, perhaps,
the end of the beginning.

—Winston Churchill (1874–*1965*),
Nov. 10, 1942, Mansion House

About the Authors

Note: Part of this white paper is based on material written by the authors for the book *Oracle JDeveloper 11*g *Handbook*, Oracle Press, McGraw-Hill/Osborne (2009).

Peter Koletzke is technical director and principal instructor for the Enterprise e-Commerce Solutions practice at Quovera, in Mountain View, California, and has over 26 years of industry experience. Peter has presented at various Oracle users group conferences more than 270 times and has won awards such as Pinnacle Publishing's Technical Achievement, Oracle Development Tools Users Group (ODTUG) Editor's Choice (twice), ODTUG Best Speaker, ECO/SEOUC Oracle Designer Award, ODTUG Volunteer of the Year, and NYOUG Editor's Choice (twice). He is an Oracle Certified Master, Oracle ACE Director, and coauthor of eight Oracle Press Books about Oracle development tools the latest of which are: *Oracle JDeveloper 11*g *Handbook* (with Duncan Mills and Avrom Roy-Faderman) and *Oracle JDeveloper 10*g *for Forms & PL/SQL Developers* (with Duncan).

Duncan Mills is senior director of product management for Oracle's Application Development Tools, including Oracle JDeveloper, Oracle Enterprise Pack for Eclipse, Forms, and the ADF Framework. He has been working with Oracle in a variety of application development and DBA roles since 1988. For the past 15 years he has been working at Oracle in both customer support and product development, spending the last 9 years in product management for the development tools platform. Duncan is a frequent presenter at industry events and has many publications to his credit including coauthorship of the Oracle Press books: *Oracle JDeveloper 11*g *Handbook* and *Oracle JDeveloper 10*g *for Forms and PL/SQL Developers*.

ROLTA

TUSC

Innovative Technology for Insightful Impact

- ✓ Finance & Performance Management
- ✓ Business & Operational Intelligence
- ✓ Strategic & Tactical IT Solutions

innovation insights impact

ORACLE CERTIFIED ADVANTAGE PARTNER

GSA Contract Holder Contract GS-35F-0149U

www.rolta.com – www.tusc.com – 1.800.755.TUSC

Oracle 11g SQL Plan Management

Sameer Malik, Tata Consultancy Services

All DBAs encounter several performance issues whenever there is a database/server upgrade or change in any optimizer parameter settings. The optimizer can come up with several new execution plans and can cause severe performance issues as the new plans are not verified against regression. This presentation addresses the challenge of eliminating/minimizing these issues using Oracle 11g's new feature - SQL Plan Management (SPM). It uses preventive mechanism to automatically control SQL plan evolution with the help of what are called SQL plan baselines. The SPM's job is to capture and evaluate the execution plans over time and build SQL plan baselines containing only efficient execution plans. When a new plan is found for a SQL statement it will not be used until it has been verified by the database to have comparable or better performance than the current plan.

Introduction

The performance of any applications using Oracle database depends on how well the Oracle Optimizer performs the query execution. But once the cost optimizer provides an efficient execution plan for a SQL statement, it cannot be assumed that the optimizer will always use that execution plan. There may be any number of changes in the database ranging from changes such as the addition or deletion of an index, changes in the composition of data that affects factors such as cardinality and selectivity, or the most obvious changes such as database or server upgrade.

The execution plan can change for better or for worse performance. Optimizer not being able to guarantee a plan will always change for the better - has led some customers to freeze their execution plans (using Stored Outlines) or lock their optimizer statistics. However, doing so prevents such environments from ever taking advantage of new optimizer functionality or access paths, which would improve the SQL statements performance. Being able to preserve the current execution plan amidst environment changes and allowing changes only for the better would be the ultimate solution.

Oracle database 11g solves this problem with introduction of brand new feature "SQL Plan Management" (SPM), which provides a framework for completely transparent controlled execution plan evolution. With SPM the optimizer automatically manages execution plans and ensures only known or verified plans are used. When a new plan is found for a SQL statement it will not be used until it has been verified by the database to have comparable or better performance than the current plan.

SQL Plan Management

The SQL Plan Management has three main phases:

SQL Plan Baseline Capture

The SQL Plan baselines can be captured by either having the database automatically capture the plan or by manually loading them using various techniques described later in the paper. The baselines contain 2 sets of plans - the one accepted being stored in SQL Plan baseline, and the non-accepted plan being stored in the SQL Plan history.

SQL Plan Baselines

A SQL plan baselines are set of all accepted plans in the plan history that the database maintains for each repeatable SQL statement that the database executes. The database defines a plan as acceptable when it verifies that the plan doesn't lead to performance regression when compared to the other plan in the plan history. The very first execution plan the database generates for a SQL statement is always considered acceptable by the optimizer and becomes the original SQL plan baseline as well as the plan history for the statement. Later execution plans will not be included in the SQL baseline unless the database verifies that they don't lead to performance regression.

Plan History

Database maintains plan history for repeatable SQL statements and not for adhoc SQL statements. The Plan history contains all the plans generated for the specific SQL statements over time, and includes all information used by the optimizer when figuring out an optimal execution plan, including information regarding the SQL text, bind variables and the environment in which the SQL statements is being executed.

SQL Plan Baseline Selection

Once the SQL plans are collected either from the AWR (automatic workload repository) or from the cursor cache, the next steps in SPM is to use those SQL Plan baselines. The use of SQL Plan baselines is controlled by initialization parameter "optimizer_use_sql_plan_baselines" which has the default value set to true, which means the use of SQL Plan baselines are enabled by default.

SQL Plan Baseline Evolution

Evolving SQL Plan baseline is the critical phase when the database changes a non-accepted plan in the plan history to an accepted plan and makes it as part of SQL Plan baselines. Only the plans which are captured automatically on the fly are subject for the Plan evolution, the manually loaded plans are always added as an accepted plan.

Scenarios for SQL Plan Management

The SPM can be used to preserve the SQL performance when the following system changes are encountered:

1. Database upgrades
2. New optimizer versions
3. Changes in optimizer parameters
4. Changes in system settings.
5. Changes in schema and metadata definitions.
6. Deployment of new application modules.

Capturing SQL Plan Baselines

Automatic Plan Capture

Setting the new initialization parameter "optimizer_capture_sql_plan_baselines" to true lets the database automatically create and maintain a SQL plan history for each repeatable SQL statements. By default the parameter is set to false. This is a dynamic parameter and hence can be changed on the fly.
The very first SQL plan that the database generates for any SQL statements is automatically integrated into the corresponding SQL plan baseline. The automatic SQL plan capturing mechanism can be used to retain good execution plan for use after database upgrade as shown below:

1. Upgrade the Oracle database to Oracle 11g.
2. Set initialization parameter optimizer_capture_sql_plan_baselines=TRUE.
3. Set initialization parameter “optimizer_features_enable”=10.2. (Or whatever release you have upgraded from, as long as it's at least set to 10.0.0 which is the minimum level you can set the compatible parameter to, if you set compatible parameter to less than 11.1 than the features that requires 11.1 compatibility generate an error if you try to use them)
4. The SQL plan management feature will collect the pre-Oracle database 11g execution plans and store them as SQL plan baselines in the upgraded database.
5. Once the database goes through a complete workload and captures all possible SQLs and hence SQL plan baselines, and then set the optimizer_features_enable to 11.x.x.x.

Manual Plan Loading

The manual plan loading can be used instead of and/or together with automatic plan capture. There is however one significant difference between the manual plan loading and automatic plan loading - when you load plan manually the database does not verify them for performance regression and it immediately adds the plans you added manually as accepted plans to the SQL Plan baseline. Manual loading of execution plans is especially useful when a database is being upgraded from a previous version to Oracle Database 11g or when a new application is being deployed. Manual bulk loading can be done in conjunction with or instead of automatic plan capture.
Execution plans that are bulk loaded are automatically accepted to create new SQL plan baselines or to add to an existing one. The SQL Management Base can be bulk loaded using four different techniques.

Bulk Load Using SQL Tuning Set (STS) and AWR Snapshots

SQL Tuning set (STS) and AWR Snapshot Load

Bulk loading execution plans from a STS is an excellent way to guarantee no plan changes as part of a database upgrade. Given below are the steps to load plan manually using STS and AWR snapshots:

1. In an Oracle Database 10gR2 create an STS that includes the execution plan for each of the SQL statements.

Creating SQL Tuning sets (STS)

An SQL tuning set is created using the CREATE_SQLSET procedure:

```
BEGIN
DBMS_SQLTUNE.create_sqlset (sqlset_name => 'SPM_TEST_1',
description => 'A test SQL tuning set.');
END;
/
```

Statements are added to the set using the LOAD_SQLSET procedure which accepts a REF CURSOR of statements retrieved using one of the following pipelined functions:
SELECT_WORKLOAD_REPOSITORY - Retrieves statements from the AWR.
SELECT_CURSOR_CACHE - Retrieves statements from the cursor cache.
SELECT_SQLSET - Retrieves statements from another SQL tuning set.
SELECT_SQL_TRACE Function – using the SQL trace files.

Loading SQL Tuning sets

STS Load from the AWR for Snapshot Range

```
DECLARE
l_cursor  DBMS_SQLTUNE.sqlset_cursor;
BEGIN
OPEN l_cursor FOR
SELECT VALUE(p)
FROM   TABLE (DBMS_SQLTUNE.select_workload_repository (
998,  -- begin_snap
999,  -- end_snap
NULL, -- basic_filter (The SQL predicate to filter the SQL from the workload repository
defined on attributes of the SQLSET_ROW)
NULL, -- object_filter (The objects to selected  from the SWRF)
NULL, -- ranking_measure1 (An order-by clause on the selected SQL)
NULL, -- ranking_measure2
```

```
NULL, -- ranking_measure3
NULL, -- result percentage ((A filter which picks the top N% as per ranking measure
given.it applies only if one ranking measure is given)
10,   -- result_limit (The top L(imit) SQL from the (filtered) source ranked by the
ranking measure)
NULL) --attribute_list (List of SQL statement attributes to return in the result. The
possible values are: Basic,Typical,ALL) ) p;
DBMS_SQLTUNE.load_sqlset (sqlset_name     => 'SPM_TEST_1',
populate_cursor => l_cursor);
END;
```

STS Load from AWR for Predefined Baseline

```
DECLARE
l_cursor  DBMS_SQLTUNE.sqlset_cursor;
BEGIN
OPEN l_cursor FOR
SELECT VALUE(p)
FROM   TABLE (DBMS_SQLTUNE.select_workload_repository (
'peek_baseline',
----   ) p;
DBMS_SQLTUNE.load_sqlset ( sqlset_name => 'SPM_TEST_1', populate_cursor => l_cursor);
END;
/
```

STS Load from the Cursor Cache

```
DECLARE
l_cursor  DBMS_SQLTUNE.sqlset_cursor;
BEGIN
OPEN l_cursor FOR
SELECT VALUE(p)
FROM   TABLE (DBMS_SQLTUNE.select_cursor_cache (
NULL, -- basic_filter
NULL, -- object_filter
NULL, -- ranking_measure1
NULL, -- ranking_measure2
NULL, -- ranking_measure3
NULL, -- result_percentage
1)    -- result_limit ) p;
DBMS_SQLTUNE.load_sqlset (sqlset_name => 'SPM_TEST_1', populate_cursor => _cursor);
END;
DBMS_SQLTUNE.SELECT_CURSOR_CACHE ('parsing_schema_name <> ''SYS''')) P;
```

STS Load from the CAPTURE_CURSOR_CACHE_SQLSET

Over a specified time interval this procedure incrementally captures a workload from the cursor cache into a SQL tuning set. The procedure captures a workload from the cursor cache into a SQL tuning set, polling the cache multiple times over a time period and updating the workload data stored there. It can execute over as long a period as required to capture an entire system workload.

```
DBMS_SQLTUNE.CAPTURE_CURSOR_CACHE_SQLSET(
sqlset_name =>'SPM_TEST_1', -- The SQL tuning set name
time_limit => '1800',       -- The total amount of time, in seconds, to execute
repeat_interval=>'300',     -- The amount of time, in seconds, to pause between sampling
capture_option =>'MERGE'    --either insert new statements; update existing ones, or
```

```
both. 'INSERT', 'UPDATE', or 'MERGE'
sqlset_owner =>'SYS');
END;
/
```

STS Load Using Existing STS

```
DECLARE
l_cursor  DBMS_SQLTUNE.sqlset_cursor;
BEGIN
DBMS_SQLTUNE.create_sqlset (
sqlset_name  => 'SPM_TEST_2',
description  => 'Another test SQL tuning set.');
OPEN l_cursor FOR
SELECT VALUE(p)
FROM   TABLE (DBMS_SQLTUNE.select_sqlset (
'SPM_TEST_1', -- sqlset_name   )     p;
DBMS_SQLTUNE.load_sqlset (sqlset_name => 'SPM_TEST_2',  populate_cursor => l_cursor);
END;
/
```

2. Load the STS into a staging table and export the staging table into a flat file.

Load the STS into a Staging Table and Export the Staging Table

```
BEGIN
DBMS_SQLTUNE.create_stgtab_sqlset(table_name => 'SQLSET_TAB',
schema_name => 'SPMUSER', tablespace_name => 'USERS');
END;
/
```

Next, use the PACK_STGTAB_SQLSET procedure to export SQL tuning set into the staging table.

```
BEGIN
DBMS_SQLTUNE.pack_stgtab_sqlset (sqlset_name => 'SPM_TEST_1',
sqlset_owner => 'SYS', staging_table_name   => 'SQLSET_TAB',
staging_schema_owner => 'SPMUSER');
END;
/
```

3. Import the staging table from a flat file into an Oracle Database 11g and unload the STS.

Import the Staging Table and Unload the STS

Once the SQL tuning set is packed into the staging table, the table can be transferred to the test system using Datapump, Export/Import or via a database link. Once on the test system, the SQL tuning set can be imported using the UNPACK_STGTAB_SQLSET procedure.

```
BEGIN
DBMS_SQLTUNE.unpack_stgtab_sqlset(sqlset_name => '%', sqlset_owner => 'SYS',
replace => TRUE, staging_table_name => 'SQLSET_TAB',  staging_schema_owner => ‘SPMUSER');
END;
/
```

4. DBMS_SPM.LOAD_PLANS_FROM_SQLSET to load the execution plans into the SQL Management Base.

Load the Plan from STS to SQL Plan Baseline
To load plans from a SQL Tuning Set, use the LOAD_PLANS_FROM_SQLSET function of the DBMS_SPM package:

```
DECLARE
my_plans pls_integer;
BEGIN
my_plans := DBMS_SPM.LOAD_PLANS_FROM_SQLSET(sqlset_name =>SPM_TEST_1');
END;
/
SQL> variable pls number;
SQL> exec :pls := dbms_spm.load_plans_from_sqlset(sqlset_name => 'SPM_TEST_1', -
basic_filter => 'sql_text like ''select%p.prod_name%'''-- 'sql_text like ''select
/*LOAD_STS*/%''' or basic_filter => 'sql_id="b62q7nc33gzwx"'
fixed=>NULL -- Default 'NO' means the loaded plans will be used as 'non-fixed' plans.
enable=>NULL --Default 'YES' means the loaded plans are enabled for use by optimizer
commit_rows=> NULL --Number of SQL plans to load before doing a periodic commit. This
helps to shorten the undo log.);
```

From Stored Outlines

When either access to SQL Tuning Sets is not available or upgrading from a version earlier than Oracle Database 10gR2, the existing execution plan can be captured using Stored Outlines. Stored Outlines can be loaded into the SQL Management Base as SQL plan baselines using the PL/SQL procedure DBMS_SPM.MIGRATE_STORED_OUTLINE or through Oracle Enterprise Manager (EM). Next time these statements are executed, the SQL plan baselines will be used. There are two ways to capture Stored Outlines - by manually creating one for each SQL statement using the CREATE OUTLINE command or letting Oracle automatically create a Stored Outline for each SQL statement that is executed. Given below are the steps needed to let Oracle automatically create the Stored Outlines for you.

1. Begin by starting a new session and switch on the automatic capture of a Stored Outline for each SQL statement that gets parsed from now on until you explicitly turn it off.
2. Execute the workload either by running the application or manually issuing SQL statements. NOTE: if you manually issue the SQL statements ensure you use the exact SQL text used by the application, if it uses bind variables you will have to use them too.
3. After the execution of critical SQL statements, turn off the automatic capture.
4. The actual Stored Outlines are stored in the OUTLN schema. One can either export the schema and import it into the 11g database or upgrade the existing database to 11g.
5. Use EM or DBMS_SPM.MIGRATE_STORED_OUTLINE to load the Stored Outlines into the SQL Management Base.

After migrating the stored outlines to SQL plan baselines, one needs to ensure that when executing a SQL statement, the database creates plan baselines but does not create stored outlines. The database only uses stored outlines when the equivalent SQL plan baselines do not exist.
For example, the following SQL statements instruct the database to create SQL plan baselines instead of stored outlines when a SQL statement is executed. The example also instructs the database to apply a stored outline in category all_rows or DEFAULT only if it exists and has not been migrated to a SQL plan baseline. In other cases, the database applies SQL plan baselines instead.

```
SQL> ALTER SYSTEM SET CREATE_STORED_OUTLINE = false;
SQL> ALTER SESSION SET USE_STORED_OUTLINES = all_rows;
```

From the Cursor Cache

Starting in Oracle Database 11g it is possible to load plans for statements directly from the cursor cache into the SQL Management Base. By applying a filter - on the module name, the schema, or the SQL_ID - you can identify the SQL statement or set of SQL statement you wish to capture.

Loading plans directly from the cursor cache can be extremely useful if application SQL has been tuned by hand using hints. Since it is unlikely the application SQL can be changed to include the hint, by capturing the tuned execution plan as a SQL plan baseline you can ensure that the application SQL will use that plan in the future. By using the simple steps you can use SPM to capture the hinted execution plan and associate it with the non-hinted SQL statement.

You begin by capturing a SQL plan baseline for the non-hinted SQL statement.

1. In a SQL*Plus session run the non-hinted SQL statement so we can begin the SQL plan baseline capture.

```
SQL> SELECT prod_name, SUM(amount_sold) FROM Sales s, Products p WHERE
s.prod_id=p.prod_id AND prod_category = :ctgy GROUP BY prod_name;
```

2. Then find the SQL_ID for the statement in the V$SQL view.

```
SQL> SELECT sql_id, sql_fulltext FROM V$SQL WHERE sql_text LIKE '%SELECT prod_name,
SUM(%';
SQL_ID SQL_FULLTEXT
------------- ----------------------------------------
chj6q8z7ykbyy SELECT PROD_NAME, SUM(AMOUNT_SOLD)
```

3. Using the SQL_ID create a SQL plan baseline for the statement.

```
SQL> variable cnt number;
SQL> EXECUTE: cnt :=DBMS_SPM.LOAD_PLAN_FROM_CURSOR_CACHE(sql_id=>'chj6q8z7ykbyy');
```

4. The plan that was captured is the sub-optimal plan and it will need to be disabled. The SQL_HANDLE & PLAN_NAME are required to disable the plan.

```
SQL> SELECT sql_handle, sql_text, plan_name, enabled FROM dba_sql_plan_baselines;
SQL_HANDLE SQL_TEXT PLAN_NAME ENABLE
------------------------ ---------------------- ------------------------ ------
SYS_SQL_bf5c9b08f72bde3e SELECTPROD_NAME  SQL_PLAN_byr4v13vkrrjy42949306 Y
```

5. Using DBMS_SPM.ALTER_SQL_PLAN_BASELINE disable the bad plan.

```
SQL> variable cnt number;
SQL> exec :cnt :=DBMS_SPM.ALTER_SQL_PLAN_BASELINE(SQL_HANDLE =>
'SYS_SQL_bf5c9b08f72bde3e',PLAN_NAME=> 'SQL_PLAN_byr4v13vkrrjy42949306',
ATTRIBUTE_NAME => 'enabled', ATTRIBUTE_VALUE => 'NO');
SQL> SELECT sql_handle, sql_text, plan_name, enabled FROM dba_sql_plan_baselines;
SQL_HANDLE SQL_TEXT PLAN_NAME ENABLE
------------------------ ---------------------- ------------------------ ------
SYS_SQL_bf5c9b08f72bde3e SELECTPROD_NAME, SUM SQL_PLAN_byr4v13vkrrjy42949306 N
```

6. Now you need to modify the SQL statement using the necessary hints & execute the modified statement.

```
SQL> SELECT /*+ INDEX(p) */ prod_name, SUM(amount_sold) FROM Sales s, Products p WHERE
```

```
s.prod_id=p.prod_id AND prod_category = :ctgy GROUP BY prod_name;
```

7. Find the SQL_ID and PLAN_HASH_VALUE for the hinted SQL statement in the V$SQL view.

```
SQL> SELECT sql_id, plan_hash_value, fulltext FROM V$SQL WHERE sql_text LIKE '%SELECT /*+
INDEX(p) */ prod_na%';
SQL_ID PLAN_HASH_VALUE SQL_FULLTEXT
------------- --------------- ---------------------------
djkqjd0kvgmb5 3074207202 SELECT /*+ INDEX(p) */
```

8. Using the SQL_ID and PLAN_HASH_VALUE for the modified plan, create a new accepted plan for original SQL statement by associating the modified plan to the original statement's SQL_HANDLE.

```
exec cnt:=dbms_spm.load_plans_from_cursor_cache(sql_id=>'djkqjd0kvgmb5',plan_hash_value
=> 3074207202,sql_handle => 'SYS_SQL_bf5c9b08f72bde3e');
```

Bulk Load Using SQL Plan Baselines from a Staging Table

The deployment of a new application module means the introduction of completely new SQL statements into the database. With Oracle Database 11g, any 3rd party software vendor can ship their application software along with the appropriate SQL plan baselines for new SQL being introduced. This guarantees that all SQL statements that are part of the SQL Plan baseline will initially run with the plans that are known to give good performance under a standard test configuration. Alternatively, if an application is developed or tested in-house, the correct plans can be exported from the test system and imported into production using the following steps:

1. On the original system, create a staging table using the DBMS_SPM.CREATE_STGTAB_BASELINE procedure.
2. Pack the SQL plan baselines you want to export from the SQL management base into the staging table using the DBMS_SPM.PACK_STGTAB_BASELINE function.
3. Export the staging table into a flat file using the export command or Oracle Data Pump.
4. Transfer this flat file to the target system.
5. Import the staging table from the flat file using the import command or Oracle Data Pump.
6. Unpack the SQL plan baselines from the staging table into the SQL management base on the target system using the DBMS_SPM.UNPACK_STGTAB_BASELINE function.

Transferring SQL Plan Baselines

The DBMS_SPM package provides functionality for transferring SQL plan baselines between databases. First, a staging table must be created in the source database using the CREATE_STGTAB_BASELINE procedure.

```
BEGIN
DBMS_SPM.CREATE_STGTAB_BASELINE (table_name => 'spm_stageing_tab',   table_owner=> TEST',
tablespace_name => 'USERS');
END;
/
```

The PACK_STGTAB_BASELINE function exports the SQL plan baselines to the staging table. There are several parameters allowing you to limit amount and type of data you export. The following example exports all SQL plan baselines.

```
SET SERVEROUTPUT ON
DECLARE
l_plans_packed  PLS_INTEGER;
BEGIN
```

```
l_plans_packed := DBMS_SPM.pack_stgtab_baseline( table_name => 'spm_stageing_tab',
table_owner     => 'TEST');
DBMS_OUTPUT.put_line('Plans Packed: ' || l_plans_packed);
END;
/
```

The staging table is then transferred to the destination database using data pump or the original export/import utilities. Once in the destination database, the SQL plan baselines are imported into the dictionary using the UNPACK_STGTAB_BASELINE function. Once again, there are several parameters allowing you to limit amount and type of data you import. The following example imports all SQL plan baselines owned by the user "TEST".

```
DECLARE
l_plans_unpacked  PLS_INTEGER;
BEGIN
l_plans_unpacked := DBMS_SPM.unpack_stgtab_baseline( table_name  => spm_stageing_tab',
table_owner     => 'TEST',  creator         => 'TEST');
DBMS_OUTPUT.put_line('Plans Unpacked: ' || l_plans_unpacked);
END;
/
```

Selecting SQL Plan Baselines

To enable the database to use the SQL plan baselines captured by automatic capture or manual capture or both, set the initialization parameter "optimizer_use_sql_plan_baselines" to true. This parameter is set to true by default. When this parameter set to true either by default or manually will enable the use of SQL plan baselines stored by database in the SQL Management Base (SMB).

When a SQL statement is hard parsed, the cost based optimizer produces several execution plans and selects the one with the lowest cost. If a SQL plan baseline is present, the optimizer compares the plan it just produced with the plans in the SQL plan baseline (refer Figure 1). If a matching plan is found that is flagged as accepted the plan is used. If the SQL plan baseline doesn't contain an accepted plan matching the one it just created, the optimizer evaluates the accepted plans in the SQL plan baseline and uses the one with the lowest cost. If the execution plan originally produced by the optimizer has a lower cost than those present in the SQL plan baseline, it is added to the baseline as a not-accepted plan, so it is not used until it is verified not to cause a reduction in performance. If a system change affects all existing accepted plans, so they are considered non-reproducible, the optimizer will use the plan with the lowest cost.

Note:-Oracle only stores the outline for each plan in the SMB, therefore the optimizer will reproduce the actual execution plan from the stored outline of the selected plan and execute the statement using that plan.

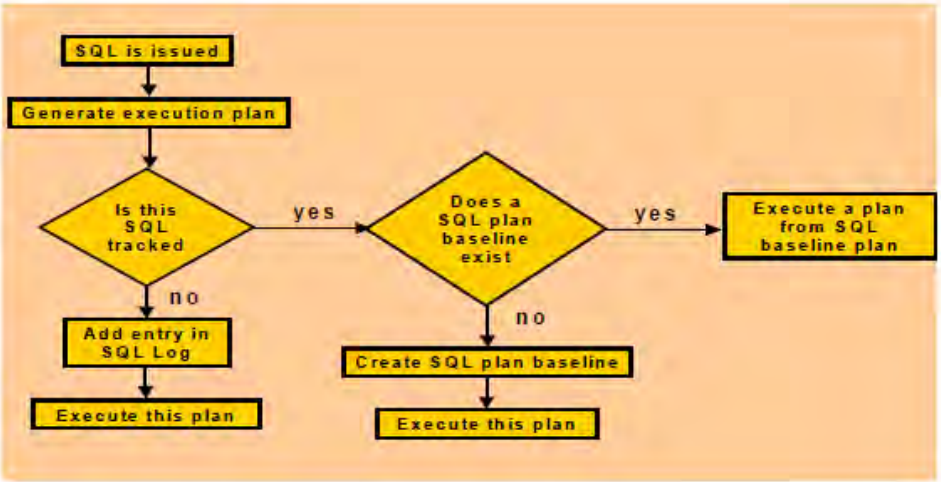


Figure 1. SQL Plan Management Flowchart

Evolving SQL Plan Baselines

The database routinely evaluates new plan performance with a view to integrating plans with superior performance into SQL plan baseline for the corresponding SQL statement. Evolving SQL plan baseline changes a non-accepted plan in the plan history to an accepted plan and makes it a part of SQL plan baseline as an accepted plan. In order to deem a plan in the history as an accepted plan, its performance must be better than all of the already accepted plans in the baseline.
The plans which are captured automatically with automatic plan capture are the only ones which formally goes through evolving process. On the other hand, if you are using the manual plan capture through an STS or a cursor cache, the moment you load a new plan into a SQL plan baseline they are considered as an accepted plan and thus don't go through the evolution process.

EVOLVE_SQL_PLAN_BASELINE Function

The EVOLVE_SQL_PLAN_BASELINE function determines whether a new plan added to the plan history performs better than the plan from the corresponding SQL plan baseline. If it performs better then it adds the new plan to the SQL Plan baseline as an accepted plan.

```
DECLARE
report clob;
BEGIN
report := DBMS_SPM.EVOLVE_SQL_PLAN_BASELINE(
sql_handle => 'SYS_SQL_593bc74fca8e6738');
DBMS_OUTPUT.PUT_LINE (report);
END;
/
DBMS_SPM.EVOLVE_SQL_PLAN_BASELINE (
sql_handle   IN VARCHAR2 := NULL,  plan_name    IN VARCHAR2 := NULL,
time_limit  IN INTEGER  := DBMS_SPM.AUTO_LIMIT,  verify    IN VARCHAR2 := 'YES',
commit       IN VARCHAR2 := 'YES')
RETURN CLOB;
DBMS_SPM.EVOLVE_SQL_PLAN_BASELINE (
plan_list  IN DBMS_SPM.NAME_LIST, time_limit  IN INTEGER := DBMS_SPM.AUTO_LIMIT, verify
IN VARCHAR2 := 'YES',
commit  IN VARCHAR2 := YES')
RETURN CLOB;
```

parameters for EVOLVE_SQL_PLAN_BASELINE:-
sql_handle:-SQL statement identifier. Unless plan_name is specified, NULL means to consider all statements with non-accepted plans in their SQL plan baselines.
plan_name :-Plan identifier. Default NULL means to consider all non- accepted plans in the plan baseline of either the identified SQL statement or all sql statements if sql_handle is NULL.
plan_list :-A list of plan names. Each plan in the list can belong to same or different SQL statement.
time_limit :-Time limit in number of minutes. This applies only if verify = 'YES'. The time limit is global and it is used as follows: The time limit for first non- accepted plan verification is set equal to the input value; the time limit for second non-accepted plan verification is set equal to (input value - time spent in first plan verification); and so on.Possible values are shown below
DBMS_SPM.AUTO_LIMIT (Default) lets the system choose an appropriate time limit based on the number of plan verifications required to be done.
DBMS_SPM.NO_LIMIT means there is no time limit.
N (A positive integer value represents a user specified time limit).
Verify:- Specifies whether to execute the plans and compare the performance before changing non-accepted plans into accepted plans. A performance verification involves executing a non-accepted plan and a plan chosen from corresponding

SQL plan baseline and comparing their performance statistics. If non-accepted plan shows performance improvement, it is changed to an accepted plan. Possible values are shown below
'YES' (Default) will verify that a non-accepted plan gives better performance before changing it to an accepted plan.
'NO' means not to execute plans but only change non-accepted plans into accepted plans.
-commit :-Specifies whether to update the ACCEPTED status of non-accepted plans from 'NO' to 'YES'.
'YES' (Default) - perform updates of qualifying non-accepted plans and generate a report that shows the updates and the result of performance verification
'NO' - generate a report without any updates. Note that commit = 'NO' together with verify = 'NO' represents a no-op.

Evolving SQL Plan with SQL Tuning Advisor

You can also evolve SQL plan baselines by running the SQL Tuning Advisor. This applies to both automatic and manual execution of SQL Tuning Advisor.SQL Tuning Advisor recommends accepting a SQL profile only if the Plan with SQL profile is better than the Original plan. Once you accept the recommended SQL profile, the advisor adds the plan to the SQL plan baseline for that SQL statement.

```
SQL> var tname varchar2(30);
SQL> exec :tname := dbms_sqltune.create_tuning_task(sql_id => 'abcdefghijkllss');
SQL> exec dbms_sqltune.execute_tuning_task(task_name => :tname);
SQL> select dbms_sqltune.report_tuning_task(:tname, 'TEXT', 'BASIC') FROM dual;
SQL> exec dbms_sqltune.accept_sql_profile(task_name => :tname);
```

Fixed SQL Plan Baselines

If a SQL plan baselines contains one or more enabled plans for which the fixed attribute value is set to yes, the baseline is considered fixed. When the plan for a SQL gets fixed then the optimizer gives preference to the fixed plans over the nonfixed plans even if some of the nonfixed plans are cheaper with a lower cost of execution. The database doesn't evolve a fixed SQL plan baseline because the optimizer will not add any new execution plans to a fixed SQL plan baseline. You may however evolve even a fixed SQL plan baseline by manually loading a new plan either from the cursor cache or a STS.

```
DBMS_SPM.ALTER_SQL_PLAN_BASELINE (
sql_handle        IN VARCHAR2 := NULL,  plan_name   IN VARCHAR2 := NULL,
attribute_name    IN VARCHAR2,  attribute_value   IN VARCHAR2)
RETURN PLS_INTEGER;
attribute_name :- Name of plan attribute to set.
attribute_value :-Value of plan attribute to use (see table below)
Enabled ('YES' means the plan is available for use.It may or may not be used depending on
accepted status.possible values 'YES' or 'NO')
fixed ('YES' means the SQL plan baseline is not evolved over time.possible values 'YES'
or 'NO')
autopurge ('YES' means the plan is purged if it is not used for a time period. 'NO' means
it is never purged.possible values 'YES' or 'NO')
plan_name (Name of the plan.possible values String of up to 30-characters)
```

Managing SQL Plan Baselines

You can view the SQL plans stored in the SQL plan baseline for a specific sql/sqls using following method.

1. DBA_SQL_PLAN_BASELINES data dictionary
2. DBMS_XPLAN.DISPLAY_SQL_PLAN_BASELINE function

```
DBMS_XPLAN.DISPLAY_SQL_PLAN_BASELINE(DSPB.sql_handle, NULL,'TYPICAL +NOTE');
```

SQL Management Base (SMB)

The database stores all SPM related information such as statement logs,plan history and SQL profiles and SQL plan baselines in a new data dictionary component called the SQL Management Base (SMB). The database stores the SMB in the SYSAUX tablespace.

Configuring the SQL Management Base (SMB)

Space usage is controlled by altering two name-value attributes using the CONFIGURE procedure of the DBMS_SPM package.
-space_budget_percent (default 10) : Maximum size as a percentage of SYSAUX space. Allowable values 1-50.
-plan_retention_weeks (default 53) : Number of weeks unused plans are retained before being purged. Allowable values 5-523 weeks.
The current settings are visible using the DBA_SQL_MANAGEMENT_CONFIG view.

```
SELECT parameter_name, parameter_value  FROM   dba_sql_management_config;
Default settings can be changed by DBMS_SPM.configure.
BEGIN
DBMS_SPM.configure('space_budget_percent', 11);
DBMS_SPM.configure('plan_retention_weeks', 54);
END;
```

Purging Policies

A weekly purging task that is the part of the automated tasks that run during the maintenance windows takes care of removing older unused baselines, to conserve space in the SMB. By default the database purges all SQL plan baselines which are not being used for 53 weeks, this setting can be change by adjusting the plan_retention_weeks attribute of the DBMS_SPM.configure procedure.

```
DBMS_SPM.configure('plan_retention_weeks', 10);
```

In addition manual purging can be done as well to purge the sql plan baseline for a specific sql statment by using PURGE_SQL_PLAN_BASELINE procedure.

```
SQL>exec DBMS_SPM.drop_sql_plan_baseline (sql_handle => 'SYS_SQL_756hhhddd90440b9);
```

References

Oracle® Database PL/SQL Packages and Types Reference 11*g* Release 1 (11.1)
http://download.oracle.com/docs/cd/B28359_01/server.111/b28274/optplanmgmt.htm
http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28419/d_spm.htm#CACJJFDB

Oracle corporation white paper (SQL Plan Management in Oracle Database 11g)

Alapati, S. 2008. *OCP Oracle Database 11g New Features for Administrators Exam Guide.* Tata McGraw-Hill.

SQL Plan Management in Oracle Database 11g Release 1
http://www.oracle-base.com/articles/11g/SqlPlanManagement_11gR1.php

Czuprynski, J. 2008. Oracle Database 11g: SQL Plan Management, Part 1 & 2
http://www.databasejournal.com/features/oracle/article.php/3723676/Oracle-Database-11g-SQL-Plan-Management-Part-1.htm
http://www.databasejournal.com/features/oracle/article.php/3730391/Oracle-Database-11g-SQL-Plan-Management-Part-2.htm

APEX Collections

Josh Millinger, Niantic Systems

Oracle Application Express has countless productivity features that allow developers to quickly create highly effective web-based applications. These features include declarative validations, wizard based form and report creation, and even declarative Javascript creation. However, there is a feature that is commonly not used by developers, mostly due to unawareness. This feature is known as Collections.

What are Collections?

An Application Express Collection is a set of rows that are maintained by the ApEx engine that are session specific. To further define it, Collections provide a developer the necessary structures to maintain scalar values that are automatically associated with and only with a specific user's session.

An immediate thought by any Oracle developer is that this sounds quite similar to Oracle Global Temporary Tables (GTTs). In fact, they would be right. Both of these constructs allow for "private" data for a particular user's session. However, there is a very good reason why ApEx applications do not use GTTs regularly. The reason involves how GTTs operate vs. how ApEx operates.

Global Temporary Tables are defined to last for either the length of a transaction or the length of a session. No data can exist past the end point of the longer of these two, a session termination. This model works well and makes good sense for a database transaction that involves a constant connection to the database. A user can issue insert after update after delete, etc without committing or ending a session. The data remains available.

Now, let's think about how ApEx works. Every page view is a new session to the database. This means if ApEx used GTTs there would be no data available after the first page was viewed. GTTs would not work if data needed to be maintained across page views. Therefore, a different paradigm was needed: Collections.

ApEx Collections allow session specific "private data" to be stored for the length of the ApEx session. The data is stored in the metadata dictionary in the APEX_0XXXXX schema and is maintained within two tables:

- WWV_FLOW_COLLECTIONS$
- WWV_FLOW_COLLECTION_MEMBERS$

A user accesses the data by querying a public view named APEX_COLLECTIONS which is defined as:

```
  COLLECTION_NAME   VARCHAR2(4000 BYTE),
  SEQ_ID            NUMBER,   -- unique ordering of rows per session collection
  C001              VARCHAR2(4000 BYTE),  -- c001 … c050 are called attributes of the
member
  C002              VARCHAR2(4000 BYTE),
  C003              VARCHAR2(4000 BYTE),
  C004              VARCHAR2(4000 BYTE),
  C005              VARCHAR2(4000 BYTE),
  C006              VARCHAR2(4000 BYTE),
  C007              VARCHAR2(4000 BYTE),
  C008              VARCHAR2(4000 BYTE),
  C009              VARCHAR2(4000 BYTE),
  C010              VARCHAR2(4000 BYTE),
  C011              VARCHAR2(4000 BYTE),
  C012              VARCHAR2(4000 BYTE),
  C013              VARCHAR2(4000 BYTE),
  C014              VARCHAR2(4000 BYTE),
  C015              VARCHAR2(4000 BYTE),
  C016              VARCHAR2(4000 BYTE),
```

```
  C017              VARCHAR2(4000 BYTE),
  C018              VARCHAR2(4000 BYTE),
  C019              VARCHAR2(4000 BYTE),
  C020              VARCHAR2(4000 BYTE),
  C021              VARCHAR2(4000 BYTE),
  C022              VARCHAR2(4000 BYTE),
  C023              VARCHAR2(4000 BYTE),
  C024              VARCHAR2(4000 BYTE),
  C025              VARCHAR2(4000 BYTE),
  C026              VARCHAR2(4000 BYTE),
  C027              VARCHAR2(4000 BYTE),
  C028              VARCHAR2(4000 BYTE),
  C029              VARCHAR2(4000 BYTE),
  C030              VARCHAR2(4000 BYTE),
  C031              VARCHAR2(4000 BYTE),
  C032              VARCHAR2(4000 BYTE),
  C033              VARCHAR2(4000 BYTE),
  C034              VARCHAR2(4000 BYTE),
  C035              VARCHAR2(4000 BYTE),
  C036              VARCHAR2(4000 BYTE),
  C037              VARCHAR2(4000 BYTE),
  C038              VARCHAR2(4000 BYTE),
  C039              VARCHAR2(4000 BYTE),
  C040              VARCHAR2(4000 BYTE),
  C041              VARCHAR2(4000 BYTE),
  C042              VARCHAR2(4000 BYTE),
  C043              VARCHAR2(4000 BYTE),
  C044              VARCHAR2(4000 BYTE),
  C045              VARCHAR2(4000 BYTE),
  C046              VARCHAR2(4000 BYTE),
  C047              VARCHAR2(4000 BYTE),
  C048              VARCHAR2(4000 BYTE),
  C049              VARCHAR2(4000 BYTE),
  C050              VARCHAR2(4000 BYTE),
  N001              NUMBER,  -- new column in 4.0
  N002              NUMBER, -- new column in 4.0
  N003              NUMBER, -- new column in 4.0
  N004              NUMBER, -- new column in 4.0
  N005              NUMBER, -- new column in 4.0
  D001              DATE, -- new column in 4.0
  D002              DATE, -- new column in 4.0
  D003              DATE, -- new column in 4.0
  D004              DATE, -- new column in 4.0
  D005              DATE, -- new column in 4.0
  CLOB001           CLOB DEFAULT empty_clob(),
  BLOB001           BLOB DEFAULT empty_blob(),-- new column in 4.0
  MD5_ORIGINAL       VARCHAR2(4000 BYTE),
  SECURITY_GROUP_ID  NUMBER NOT NULL,
  XMLTYPE001        SYS.XMLTYPE  -- new column in 4.0
)
```

As users execute the application, rows get inserted, deleted, and updated in the metadata tables. Each user might be using a Collection Named: P5_CART, but each would only see their rows. Their unique session id would help differentiate which user sees which rows.

For example:

At any particular time there are several instances of a collection

User: Abby Session: 1234 Collection: P5_CART

Seq Id	C001	C002	C003	C004
1	A	B	C	d
2	J	Q	X	Y
3	Z	Z	Z	B

User: Barry Session: 789866 Collection: P5_CART

Seq Id	C001	C002	C003	C004
1	D	23	X	Z
2	d	dsf	Y	Z
3	34	2	Z	Z

User: James Session: 23455 Collection: P5_CART

Seq Id	C001	C002	C003	C004
1			JM	DY
2	B	W	D	234
3	C	D	asdf	3d
5	D	ZD	2dsdf	fdf

As you can see, these three users all have data in their cart, but the data is maintained separately by also storing their session id with the data, thereby effectively creating different data sets.
Some of the benefits Collections provide are:

- Persistence across page views
- Maintained at session level so they are secure
- No extra DDL objects needed in schema
- Maintained by the Apex Engine

Some of the cons of Collections are:

- Attributes are not indexed
- Attributes may need to be changed to correct data type during queries (somewhat alleviated in Apex 4.0)

Some of the common use cases that are used with Collections are:

- Shopping carts
- Maintaining checkbox values across pages
- Avoiding the Dynamic SQL limitation with Interactive Reports,
- Performance gains when multiple regions query the same large data set on the same page render.

How to Use a Collection

An ApEx collection is coded to be created by the developer and is commonly named something that is informative to what the collection is being used for. For instance, if the collection was to track shopping cart items being selected on

page 30, then the collection might be called P30_CART. This name is used by queries and API to reference the session data.
ApEx provides an API, *APEX_COLLECTIONS*, to manipulate Collections.
To create a Collection names P30_CART a developer would have a process call:

```
APEX_COLLECTION.CREATE_COLLECTION('P30_CART')
```

To delete Collection P30_CART:

```
APEX_COLLECTION.DELETE_COLLECTION('P30_CART')
```

If the use case was to create a collection P40_ROWS from a query:

```
declare
  l_sql varchar2(1000);
begin
  -- define query for collection creation
  l_sql := ' select object_name, object_type from obj';

                  -- Create collection P30_ROWS using the query l_sql created above
  APEX_COLLECTION.CREATE_COLLECTION_FROM_QUERY_B('P30_ROWS', l_sql);
end;
```

To reference ApEx Collections from an ApEx or PL/SQL code a developer queries the APEX_COLLECTIONS view. For instance, if we the application placed product id's into a user's shopping cart on Page 30 and the user was on the cart verification page (31), we could query the view where we stored the product id by issuing a query like:

```
select product_name,
            product_description
  from products p,
           apex_collections a
              where a.collection_name = 'P30_CART'
    and a.c001 = p.product_id
```

Another common use case is to test if a collection exists and if so reset it with the truncate command.

```
Begin
   apex_collection.create_or_truncate_collection('P10_OBJECTS');
End;
```

Interactive Report Issue

Interactive Reports will only work against static queries. They do NOT work when a query is generated dynamically by piecing together parameter values or other logic. Collections are used to bridge the gap between what Interactive Reports have to offer and their limitations on how they can be created. The bridge is created by using a collection to gather up the data with a dynamic query and then querying the collection in the definition of the IR.
In a Before Regions Page Process, a developer can create the collection using the API's and then the definition of the IR would just query the collection.

```
select
c001 product_id,
c002 product_name,
c003 product_description
from apex_collections a
```

```
where collection_name = 'P50_IR_ROWS'
```

It is common to take the column names in the collection view and alias them with more useful column names that will make sense to end users.

New in Application Express 4.0

Application Express 4.0 introduced a couple of new features to Collections:

- Five new Number columns N001 .. N005
- Five new Date columns D001 .. D005
- One new BLOB column: BLOB001
- One new XMLTYPE column: XMLTYPE001
- Additional API calls:
  - CREATE_COLLECTION_FROM_QUERY2
  - CREATE_COLLECTION_FROM_QUERYB2
- New support for bind variables in queries

Getting Help

To see more information on how Collections work and detail about the APEX_COLLECTION API, please use the Help button in the Apex development environment. Further navigate to the API_REFERENCE section and find APEX_COLLECTION. Once there, all the functions and procedures will be defined with example uses.

- ADD_MEMBER Function
- ADD_MEMBERS Procedure
- COLLECTION_EXISTS Function
- COLLECTION_HAS_CHANGED Function
- COLLECTION_MEMBER_COUNT Function
- CREATE_COLLECTION Procedure
- CREATE_OR_TRUNCATE_COLLECTION Procedure
- CREATE_COLLECTION_FROM_QUERY Procedure
- CREATE_COLLECTION_FROM_QUERY2 Procedure
- CREATE_COLLECTION_FROM_QUERY_B Procedure
- CREATE_COLLECTION_FROM_QUERYB2 Procedure
- DELETE_ALL_COLLECTIONS Procedure
- DELETE_ALL_COLLECTIONS_SESSION Procedure
- DELETE_COLLECTION Procedure
- DELETE_MEMBER Procedure
- DELETE_MEMBERS Procedure
- GET_MEMBER_MD5 Function
- MERGE_MEMBERS Procedure
- MOVE_MEMBER_DOWN Procedure
- MOVE_MEMBER_UP Procedure
- RESEQUENCE_COLLECTION Procedure

RESET_COLLECTION_CHANGED Procedure
RESET_COLLECTION_CHANGED_ALL Procedure
SORT_MEMBERS Procedure
TRUNCATE_COLLECTION Procedure
UPDATE_MEMBER Procedure
UPDATE_MEMBERS Procedure
UPDATE_MEMBER_ATTRIBUTE Procedure Signature 1
UPDATE_MEMBER_ATTRIBUTE Procedure Signature 2
UPDATE_MEMBER_ATTRIBUTE Procedure Signature 3
UPDATE_MEMBER_ATTRIBUTE Procedure Signature 4
UPDATE_MEMBER_ATTRIBUTE Procedure Signature 5
UPDATE_MEMBER_ATTRIBUTE Procedure Signature 6
APEX_WEB_SERVICE
Index

Conclusion

While Application Express provides so many helpful features, probably the most underused one is Collections. They provide all the necessary pieces to allow for developers maintain data for a user specific session across page views, which is not possible using Global Tempoary Tables. Collections are helpful when building applications such as shopping carts or when it is necessary to have dynamically created data available for many regions on a page in a quick manner. The simple view and API calls make it a snap to integrate this feature into any application. Once Collections become part of the developers toolbox, they quickly become a favorite go-tool weapon of choice.

Josh Millinger
Niantic Systems, LLC
202.642.6845
jmillinger@nianticsystems.com

Are your business objectives and technology initiatives **out of sync?**

Zanett helps our clients align business objectives with technology to create Real Enterprise Value™

- ▶ **Enhance organizational effectiveness**
- ▶ **Respond quickly to changing business dynamics**
- ▶ **Deliver timely management information**

Business Consulting
- Human Capital Management
- Supply Chain Management
- Finance & Compliance
- Information Technology
- Strategic Planning
- Governance & Risk Management

Application Consulting
- Software Selection
- Optimization Roadmaps
- Custom Development
- Application Upgrades
- Implementation & Deployment

Infrastructure Technology
- Fusion Middleware
- Services Oriented Architecture (SOA)
- Business Process Execution Language
- Database Consulting

Managed Services
- Hosted Administration
- Remote Administration
- Application Management
- Application Upgrades
- Business Process Outsourcing

ORACLE Platinum Partner

zanett

www.zanett.com | 1.866.611.3282

Proudly Serving the Northeast for Over Ten Years

Exadata for Beginners

Rich Niemiec, Rolta TUSC

With the acquisition of Sun, a paradigm shift is occurring in the database market, the hardware world, and the IT industry as a whole. This shift is not unknown to Wall Street; it is called: Industry Consolidation. It happened after the industrial revolution – check your local antique store for the consolidated companies of that era. While this shift is just beginning, the early tremors are showing that the major shift is coming and will be big. Oracle is widening its influence into the world of hardware and will challenge the Fortune 500 as one of the top performers. With Java, their influence could quickly move into the consumer world. Exadata is the beginning of their move into the hardware world. Their slogan "Software and Hardware Engineered Together" is certainly accurate for how well Exadata is performing, but also shows a barrier to entry for others that would try to compete with them. Exadata is hardware that combines the power of the database, while leveraging features at the hardware level that other hardware providers will not be able to replicate easily (or at all). Make no mistake, this solution is "Smokin' Fast!" I've now worked on a quarter, half, and full rack and every one of them performed with client-gasping performance. Every client I've talked to using Exadata is already talking about adding more to other areas of their business.

Some of the topics that I will cover here include:

- Terminology & the Basics
- Flash Cache
- Storage Index
- Smart Scans
- Hybrid Columnar Compression (HCC)
- Enterprise Manager & Grid Control
- Enterprise Manager Exadata Simulation
- I/O Resource Manager
- Security
- Best Practices
- Summary
- References
- Biography

Terminology and the Basics

When I think of Exadata (V2-2), I think of it as a "prebuilt 8-Node RAC cluster with Super SAN." It has all the CPU power you need (64 cores), mega DRAM Server Memory (576G), a super-mega Flash Memory (5.3T), a super fast interconnect (40Gb/s), and either 100T of SAS disk (28T useable) or 336T of SATA storage. The database could store even more because of compression. My take on it is: If you need it & can afford it – You want it!!

As I said, you can load a full rack with 337T of SATA storage. SATA is big & slow, I think of it like a 33 1/3 LP vinyl album (7200 RPM). SATA stands for Serial Advanced Technology Attachment (or basically – something plugged into your computer). You can also have 100T of SAS storage instead. SAS is smaller/faster; I think of it like the old 45's records (15K RPM). SAS stands for Serial Attached SCSI (Small Computer System Interface)… or again, basically speaking – something plugged into your computer.

How FAST is it? In general, it's 5-100x faster than most data warehousing competition and 20x faster than most OLTP competition. It also includes hot swappable redundant power, each

Database Server has a dual port InfiniBand (40Gb/s card), each Database Servers have Disk Controller HBA (Host Bus Adapter) with 512M battery backed up cache and 4 x 1GbE interfaces & ILOM (Integrated Lights Out Management – Remote power on…etc.).

What's Making it FAST? Fast hardware, many CPUs, Flash Cache, lot's of DRAM (Parallel Query in DRAM in 11.2), Compression (save 10x-70x), Partition Pruning (save 10x-100x), Storage Indexes (save 5x-10x), Smart Scan (save 4x-

10x). You can turn a 1T search into a 500M search or even 50M when these are all leveraged. If anything, I think Oracle might be understating the true power of this hardware (Oracle calls it a database "machine").

Oracle Now Has a Newer Version of Exadata As Well

The original version was on HP hardware (V1) and quickly became an ALL Oracle/Sun solution after the Sun acquisition. The first version of V2 was V2-2. The next version (announced at OpenWorld in the fall of 2010 was V2-8). Lastly, a modified version (best targeted at cloud solutions) is called Exalogic and is heavy DRAM, heavy CPU. Each of these is listed below.

An Exadata (V2-2) Machine which is only manufactured by Oracle (Sun) includes:

- 8 compute servers (x4170's)
  - 8 servers x 2 CPU sockets x 4 cores = 64 cores
- 8 compute servers x 72G DRAM = 576G DRAM
- 3 InfiniBand Switches x 36 ports = 108 ports
- 14 Storage Servers with 112 CPU cores & Flash Cache
  - 96G x 4 banks = 394G DRAM per storage server
  - 14 storage servers x 394G = 5.376T Flash Cache
  - 12 disks per storage server x 14 servers = 168 disks
  - 168 disks x 600G SAS = 101T SAS or…
  - 168 disks x 2T SATA = 336T SATA

Pictured below (thanks Oracle – I added some details) gives you a feel for this hardware.

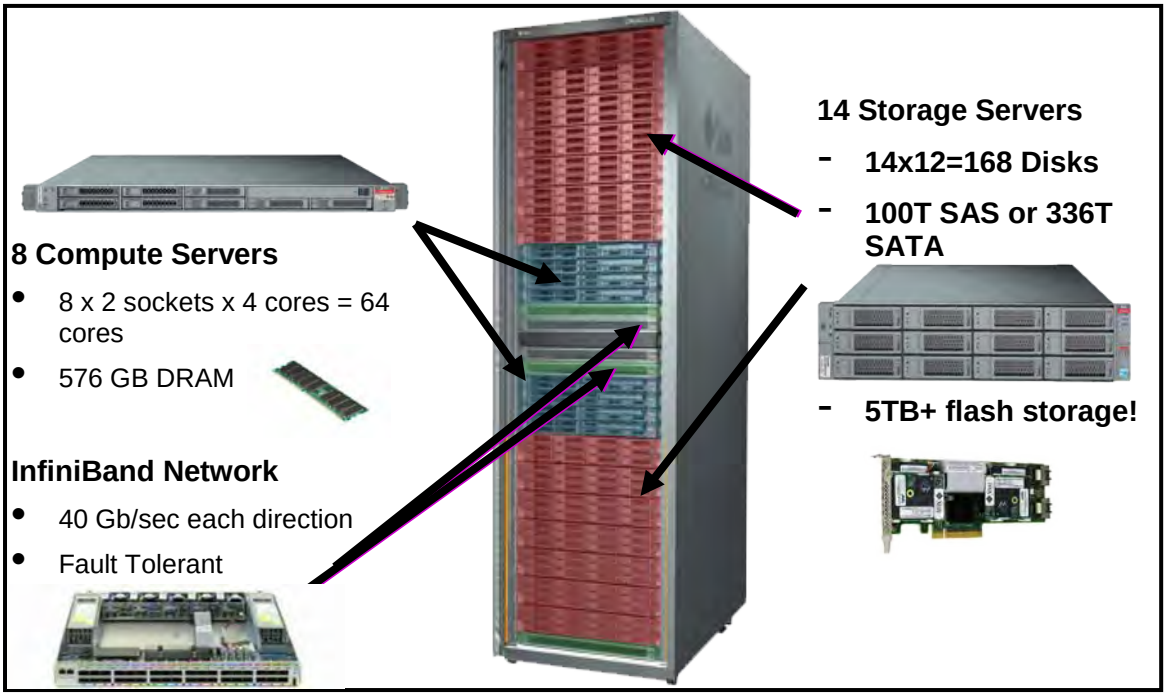


Diagram 1. Exadata V2-2

An Exadata (V2-8) Machine which is only manufactured by Oracle (Sun) includes:

- 2 compute servers (7560 CPU at 2.26 GHz & 5T SAS)
  - 2 servers (x4800's) x 8 CPU sockets x 8 cores = 128 cores
- 2 compute servers x 1T DRAM = 2T DRAM
- 3 InfiniBand Switches x 36 ports = 108 ports

- 14 Storage Servers with 112 CPU cores & Flash Cache
  - 96G x 4 banks = 394G DRAM per storage server
  - 14 storage servers x 394G = 5.376T Flash Cache
  - 12 disks per storage server x 14 servers = 168 disks
  - 168 disks x 600G SAS = 101T SAS or…
  - 168 disks x 2T SATA = 336T SAS

An Exalogic (V2-2) machine includes (serves 1M HTTP requests per second):
- 360 CPUs, 2.8T DRAM, 980G FlashFire SSD, 40T SAS – Will help Fusion Apps Smoke!
- 1M HTTP/sec – could fit Facebook on 2 of these even thought there are 500M people on Facebook

How They Got These NUMBERS?
- *8 compute servers (x4170's)*
  - 8 servers x 2 CPU sockets x 4 cores = 64 cores (E5540 2.53 GHz)
- *8 servers x 72G DRAM = 576G DRAM (400G useable)*
- *14 Storage Servers total 336G DRAM = 912G Total DRAM*
- *3 InfiniBand Switches x 36 ports = 108 ports*
- *14 Storage Servers (100-336T) with Flash Cache (5T+)*
  - 96G x 4 banks = 394G flash cache per storage server
  - 14 storage servers x 394G = 5.376T Flash Cache
  - 12 disks per storage server x 14 servers = 168 disks
  - 168 disks x 600G SAS = 101T SAS
  - 168 disks x 2T SATA = 336T SATA
  - Additional total storage of 4.672T on Database Servers (146G drives)
- *14 storage servers x 2 quad core E5540 = 112 additional cores*

Compute Servers – Like an 8 Node RAC!
- *8 compute servers (x4170's)*
  - 8 servers x 2 CPU sockets x 4 cores = 64 cores
- *8 compute servers x 72G (18x4G) DRAM =576G DRAM*
- *4 x 146G drives x 8 = 4.67T (in addition to storage servers)*

Storage Servers – Full Rack:
- *14 Storage Servers (x4275's) with Flash Cache*
  - 96G x 4 cards = 394G per storage server of flash cache
  - 24Gx14= 336G of DRAM (in addition to database servers)
  - 14 storage servers x 394G = 5.376T Flash Cache
  - 12 disks per storage server x 14 servers = 168 disks
  - 168 disks x 600G SAS = 101T SAS
  - 168 disks x 2T SATA = 336T SAS

Where Did All My Disk Space Go?
- 100T SAS = 28T usable (your mileage will vary)
- 336T SATA = 100T usable (your mileage will vary)

Apply Some Compression and Get It Back:
- 28T usable x 10 = 280T SAS
- 100T usable x 10 = 1P SATA

InfiniBand - 40G/s Each Way

- 3 InfiniBand Switches x 36 ports = 108 ports

Full Rack or Start with ½ or ¼ (Check with Oracle for Latest Configurations):

	Full	Half	Quarter
Compute Servers/cores 8/64	4/32	2/16	
Storage Servers/disks*	14/168	7/84	3/36
Storage SAS /IOPs	100T	50T	21.6T
Storage SATA	336T	168T	72T
Flash IOPs (max)	1,000,000	500,000	225,000
InfiniBand Switches	3	2**	2
Data Load Rates	5T/hr	2.5T/hr	1T/hr

* 600G SAS or 2 T SATA;
** Soon to be 3

Smart Scans

Smart scans are done internally by Oracle and 10x savings is common. Oracle filters things based on WHERE clause (predicates) & filters on row / column / join condition. There is also Incremental Backup filtering. It works with Uncommitted data, Locked rows, Chained rows, Compressed Data, & Encrypted Data (11.2). You can SEE & monitor the benefits with Grid Control (OEM). Smart Scans also leverage Bloom Filters used for Join Filtering:

- A way to quickly search for matches (simplistic meaning)
- Saves space & is transparent to the user
- Makes things faster – hardware level filtering
- Tests if elements to search for are in a set
- Google BigTable uses Bloom filters to reduce disk lookups
- Join Filtering is a perfect application for Bloom Filters!!

Without Smart Scan (Push Whole Table via Network):

- 5T Table Scan
- Network bandwidth (40Gb/s) slows things
- 40Gb/s = 5GB/s; with 14 storage cells = 0.357GB/s each

16 minutes, 40 seconds (5T/5GB/s)

With Smart Scan (Limit First at Hardware Level):

- 5T Table Scan
- Limit result BEFORE it hits the network
- Effectively scan 21GB/s (1.5G/storage cell * 14 cells)

3 minutes, 58 seconds (5T/21G/s)

Flash Cache

The Flash Cache is solid state disks (information stored on chips), which is around 20x-50x faster than disk (depending on the disks). Keep in mind – flash cache wears out faster than disks (see specs for more information). Oracle does give you uncounted "extra" space to make up for this.
The Flash Cache caches HOT Data (data frequently used). It does this as the LAST step (so it returns data to the user FIRST and then caches it for next time based on the settings you give it). It uses PCIe based Flash cards (PCI = Peripheral Component Interconnect express). It knows which objects NOT to cache such as full table scans, but YOU can

specify exactly WHAT you want to cache using the STORAGE clause or at the Table/Partition level with CREATE/ALTER:

- STORAGE (CELL_FLASH_CACHE KEEP)
- Table/Partition level with CREATE or ALTER

It also has a write through cache which is used to accelerate reads. With this feature, data that is written to disk is also written to cache (again – stored in the cache AFTER the user gets it written to disk) for future reads.

It Will Cache:

- Hot Data/Index Blocks
- Control File reads/writes
- File header reads/writes

It Does NOT Cache

- Mirror copies / Backups / Data Pump
- Tablespace Formatting
- Table Scans (or at least this is rare – for example: small tables are ok)

There is a Flash Cache LRU (least recently used). The settings for caching are listed below:

- CELL_FLASH_CACHE storage clause
  - DEFAULT (normal – large I/O's not cached)
  - KEEP (use flash cache more aggressive / May not occupy > 80%)
  - NONE (flash cache not used)
- CACHE (NOCACHE) Hint
  - I/O cached/not-cached in the flash cache
  - SELECT /*+ CACHE */ …
- EVICT Hint – Data removed from the flash cache
- ASM rebalance data is evicted from cache when done
- Large I/O (Full Table Scans) on objects with CELL_FLASH_CACHE set to DEFAULT are not cached

Using the KEEP Cache:

```
ALTER TABLE CUSTOMER
STORAGE (CELL_FLASH_CACHE KEEP);
```

Table Altered.

Query to See What Cache a Table Is Currently Set to:

```
SELECT      TABLE_NAME, TABLESPACE_NAME,
            CELL_FLASH_CACHE
FROM        USER_TABLES
WHERE       TABLE_NAME = 'CUSTOMER';

TABLE_NAME        TABLESPACE_NAME         CELL_FL
------------      -------------------     -------
CUSTOMER          R_TEST                  KEEP
```

The way the cache works is that a database request comes to CELLSRV (Cell storage server). The CELLSRV (first time) gets data from disk and the data is cached based on settings, hints … etc. The data that goes to WRITE may also be cached after being written if it is deemed that it may be needed again. The CELLSRV (next time) checks the Memory Hash Table that lists what is cached. If cached – goes to flash cache; If not cached …may cache based on settings…etc. You can also query the hardware cells directly as listed below:

```
CELLCLI> list flashcache detail (allows you to monitor)
CELLCLI> list flachcachecontent where ObjectNumber=62340 detail
```

SQL*Plus Query (Is It Working?):

```
SELECT      NAME, VALUE
FROM        V$SYSSTAT
WHERE       NAME IN (
                  'physical read total IO requests',
                  'physical read requests optimized');

Name                                          Value
--------------------------------------------  --------
physical read total IO requests               36240
physical read requests optimized              23954
```

*(this second line (*8192) is flash cache used – It's working!)*

It IS working … 4G Query:

```
SELECT      NAME, VALUE, VALUE*8192 VALUE2
FROM        V$SYSSTAT
WHERE       NAME IN (
                  'physical read total IO requests',
                  'physical read requests optimized');


NAME                                VALUE            VALUE2
---------------------------------   --------         --------
physical read total IO requests  10,862,844     88,988,418,048
physical read requests optimized  2,805,003     22,978,584,576
run2.....
physical read total IO requests  11,320,185     92,734,955,520
physical read requests optimized  3,203,224     26,240,811,008
run4 .....
physical read total IO requests  11,993,845     98,253,578,240
physical read requests optimized  3,793,000     31,072,256,000
```

It IS working … V$SQL:

```
select sql_text, optimized_phy_read_requests,
       physical_read_requests,
       io_cell_offload_eligible_bytes
from        v$sql
where       sql_text like '%FIND YOUR SQL%'

SQL_TEXT     OPTIMIZED_PHY_READ_REQUESTS PHYSICAL_READ_REQUESTS
------------ --------------------------- ----------------------
IO_CELL_OFFLOAD_ELIGIBLE_BYTES
------------------------------
SELECT....   567790                      688309
4.2501E+10
Run 2.....
SELECT…      762747                      906729
4.9069E+10
run 4 ....
SELECT...    1352166                     1566537
6.8772E+10
```

<u>Note:</u> Exadata PCIe card Smart Flash Cache (Exadata hardware PCIe Card Cache stored) is NOT the same as 11gR2 Database Flash Cache (file stored) used with Oracle Enterprise Linux.

Storage Indexes

A storage index utilizes min/max values to help queries run faster. A performance gain of 10x is common with storage indexes. They primarily maintain summary information about the data *(like Meta Data in a way).* The CELL LEVEL (storage) is where the memory structure resides. It groups data into Min/Max for various columns & eliminates I/Os where there is no match. It is 100% transparent to the user. It is performed at the hardware level and it typically has one index for every 1M of disk. It is NOT like a B-Tree Index…more like partition elimination where you skip data NOT meeting conditions of the query. It is 100% done by Oracle based on query patterns and data use – NO COMMANDS ARE NEEDED!!

Is It Working for Me?

```
SELECT      NAME, VALUE
FROM        V$SYSSTAT
WHERE NAME LIKE ('%storage%');

NAME                                            VALUE
----------------------------------------------  -------
cell physical IO bytes saved by storage index   25604736
```

(actual savings from Exadata built storage index – It's working!)

Check BOTH servers … (Only Savings on One of Them During This Test):

```
SELECT      NAME, VALUE
FROM        GV$SYSSTAT
WHERE       NAME LIKE ('%storage%');

NAME                                                VALUE
---------------------------------------------- -----------
cell physical IO bytes saved by storage index 19693854720
cell physical IO bytes saved by storage index           0
```

(actual savings from Exadata built storage index – It's working!)

Hybrid Columnar Compression (11.2)

Hybrid Columnar Compression (HCC) or also known as Exadata Hybrid Columnar Compression (EHCC) is data that is organized by a hybrid of column/row and compressed vs. data stored by basic row format. A performance gain any where from 10x to 30x is common. The tables are organized in Compression Units (CU) which contain around 1000 rows (more or less depending on the amount of data in the rows). CU's span many blocks. It is very good for data that is bulk loaded, but is not built for OLTP or single block read operations. It is primarily built for Data Warehouses and Queried Data and is NOT built for frequently updated data.

In the old OLTP compression algorithm, you could get 2-3x compression or so. Yet, 10x compression in a typical data warehouse is common with HCC. In limited tests, we got anywhere from 4-11x. There is also archive compression (used for cold data) that gives you anywhere from 15x to 70 (we got around 32x in limited tests). The nice thing about hybrid columnar compression is that operations are faster since the query runs without decompression. The compressed version is processed in FLASH CACHE which results in lower I/O, the compressed version is also sent over InfiniBand, it is cloned compressed, and it is even backed up compressed! As a result, it scans MUCH less (compressed) data!

It is worth noting that you can still use standard table compression for OLTP and a single block lookup is still generally FASTER than other columnar storage. The updated rows migrate to normal / lower level compression. Some things that are fully supported:

- B-Tree Indexes
- Bitmap Indexes

- Text Index
- Materialized Views
- Partitioning
- Parallel Query
- Data Guard Physical Standby
- Logical Standby and Streams (FUTURE release)
- Smart Scans of HCC tables!

There are also many other types of Oracle Compression not to forget that also give Oracle an advantage over competitors. These include:

- Data Pump Compression
  - Compression = {ALL | DATA_ONLY | NONE}
- RMAN Backup Compression
  - Compression Level LOW/HIGH (New in 11.2)
- Secure File Compression (large objects / images)
  - LOW/MEDIUM/HIGH (2-3x compression)
  - Deduplication & Encryption
- Normal OLTP Table Compression (since 9.2)
  - 11g now supports INSERT/UPDATE
  - FASTER Algorithm
- Data Guard Redo Transport Compression

Enterprise Manager and Grid Control for Exadata

Oracle Enterprise Manager (OEM) & Grid Control has a specific plug in for Exadata. This allows you to easily manage and view many of the advantages that Oracle is providing with Exadata. It is also a good way to ensure things are working properly and efficiently. This section shows a few of the screen shots that you might find useful. A description of each is listed under each figure.

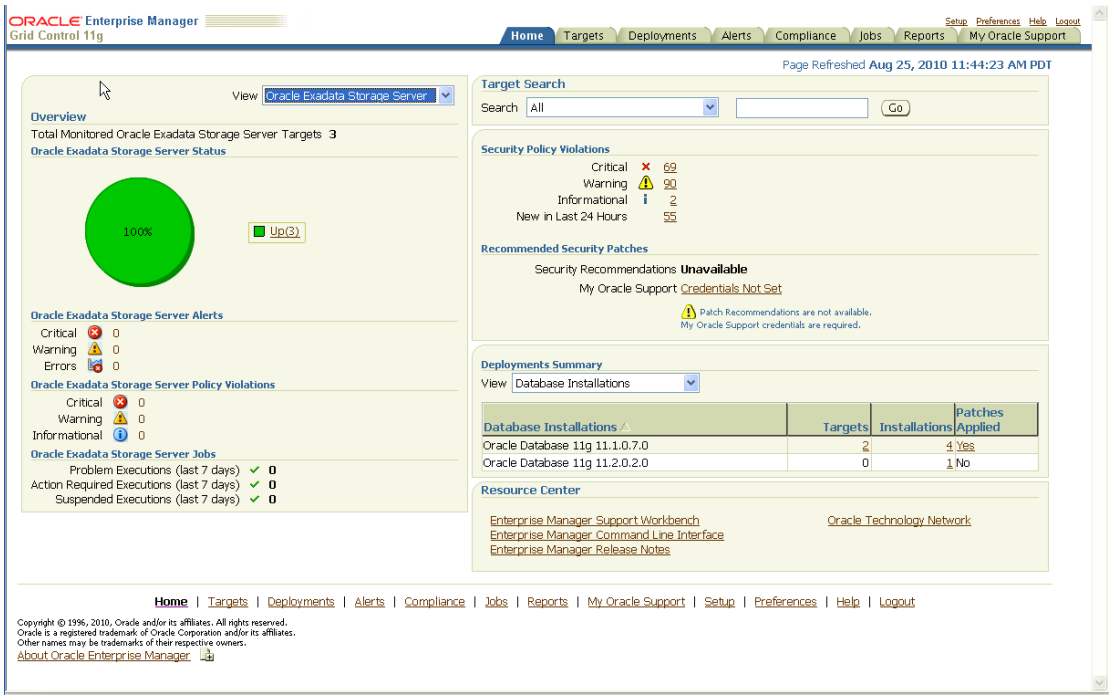


Figure 1. Viewing the Exadata Storage Server Information

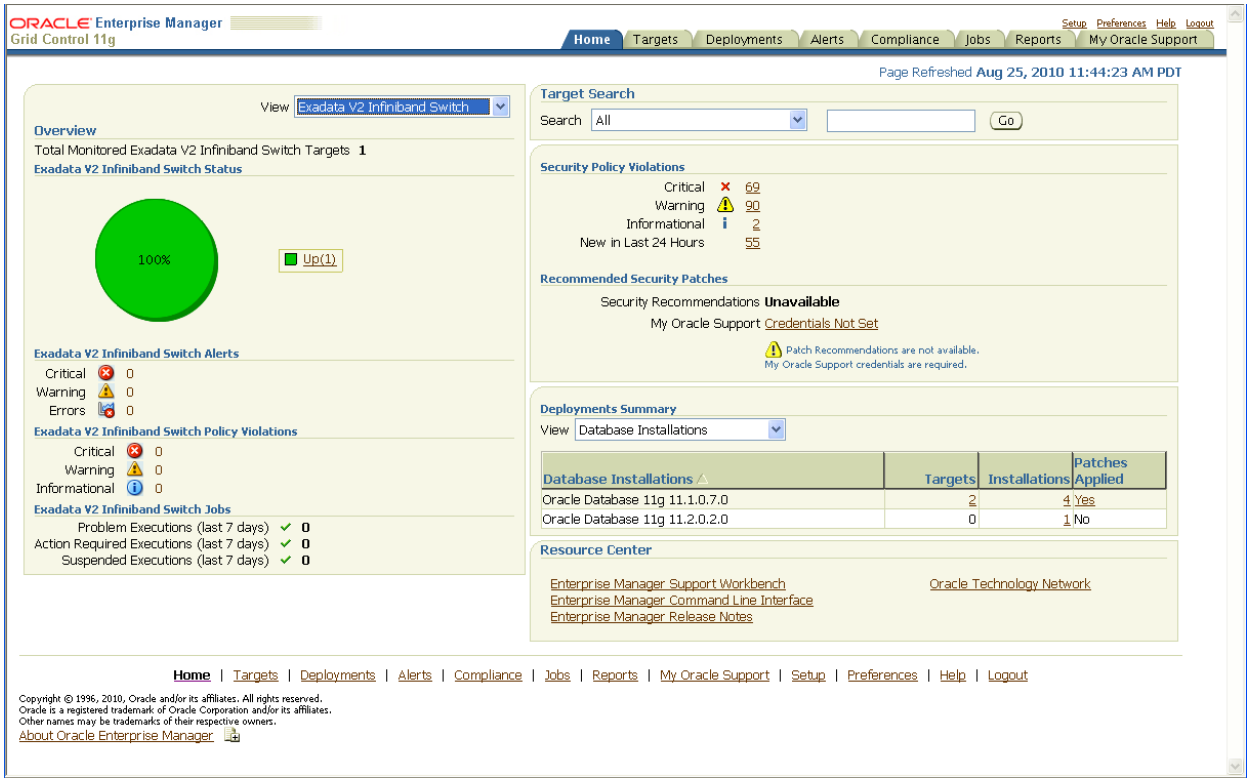


Figure 2. Viewing the InfiniBand Information

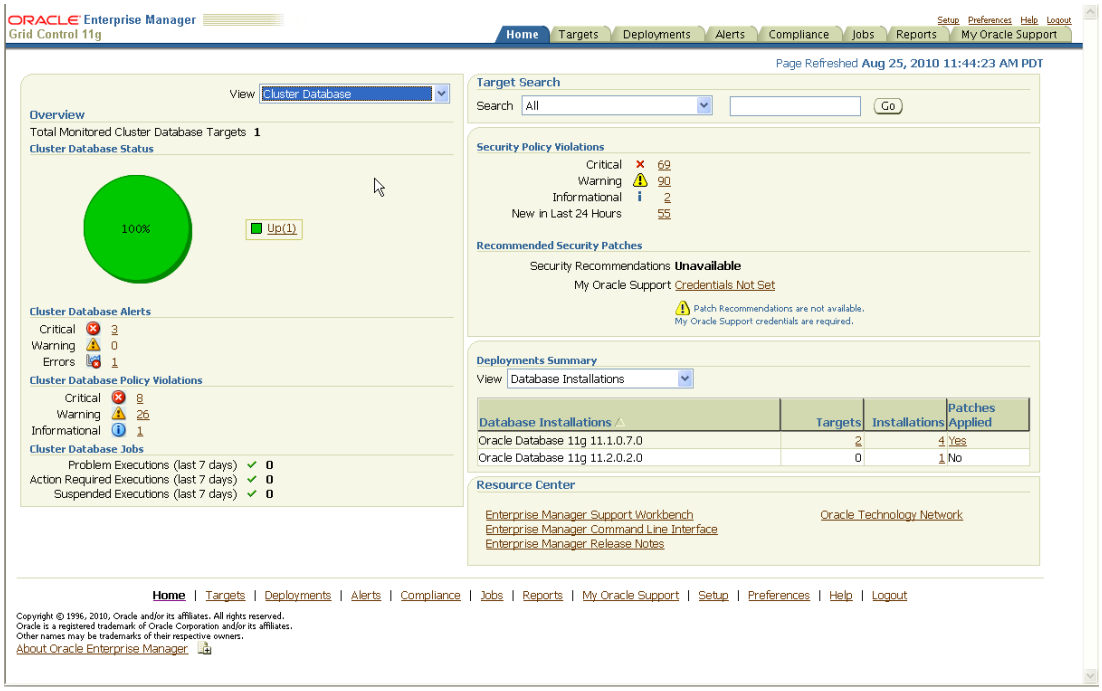


Figure 3. Viewing the Cluster Database

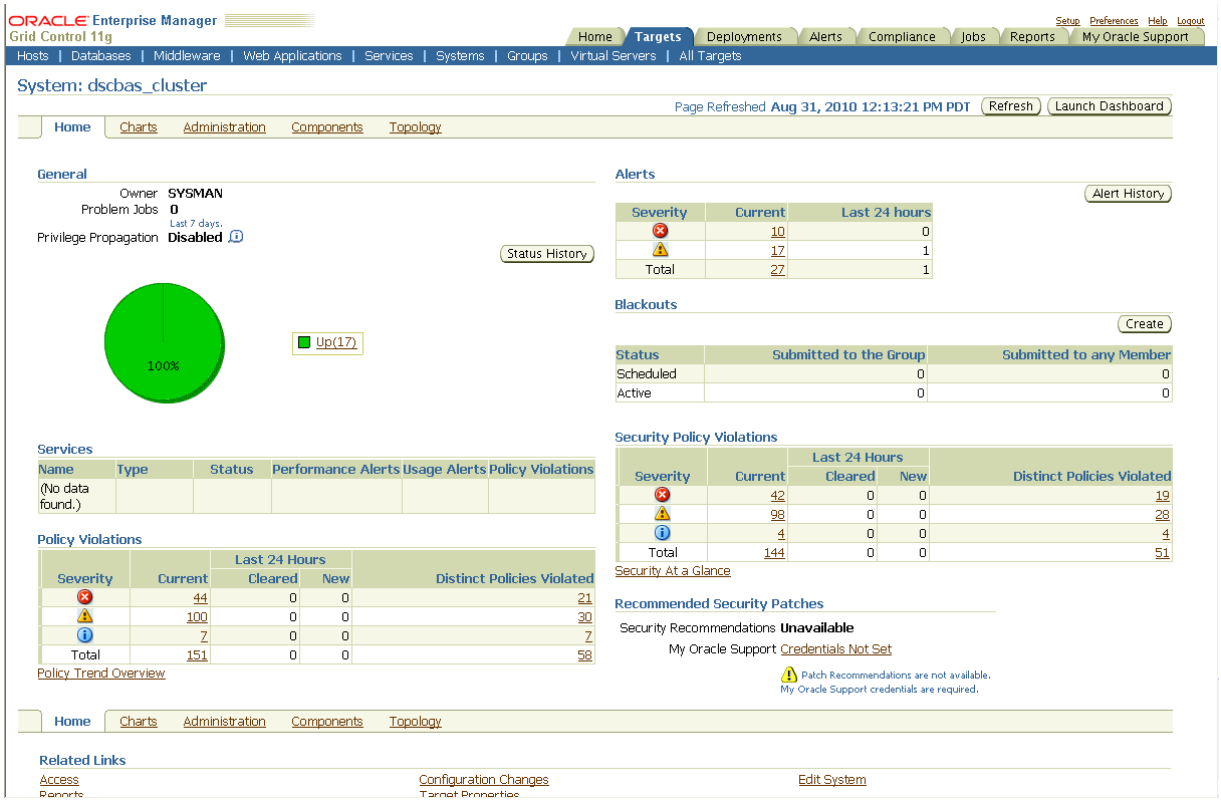


Figure 4. Viewing the Cluster Home with Alerts … etc.

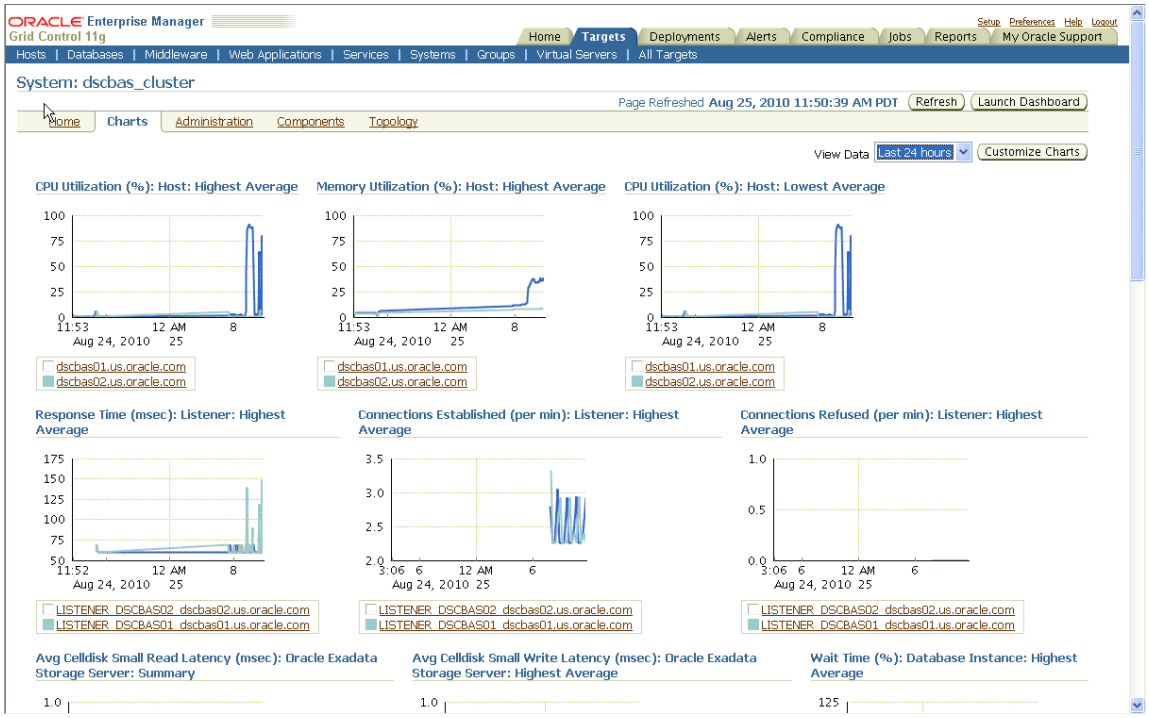


Figure 5. Viewing the Cluster Charts – Note: This is Monitoring Two Database Servers

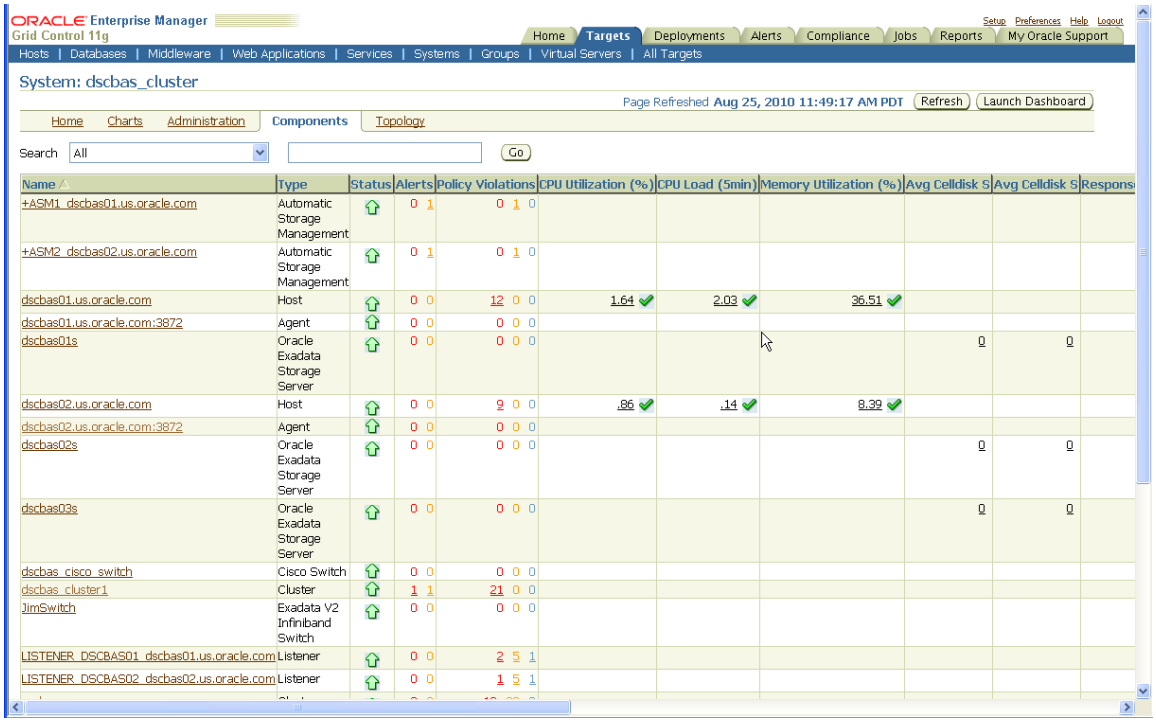


Figure 6. Viewing the Cluster Components

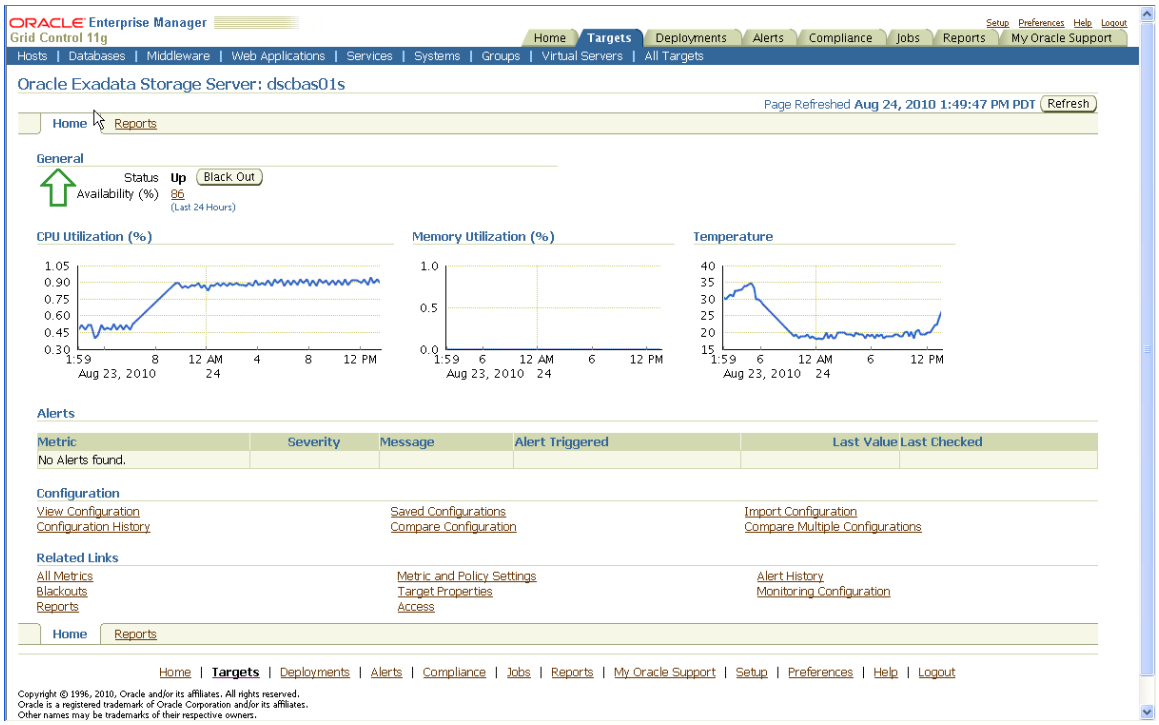


Figure 7. Viewing the Storage Server 1

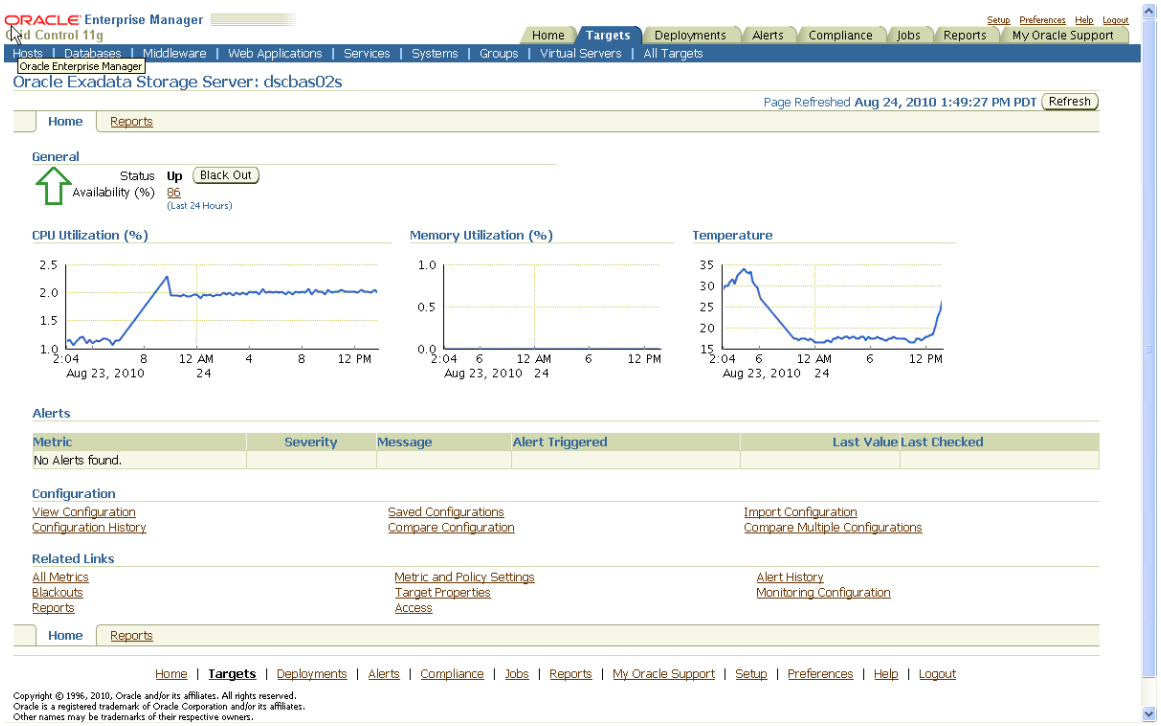


Figure 8. Viewing the Storage Server 2

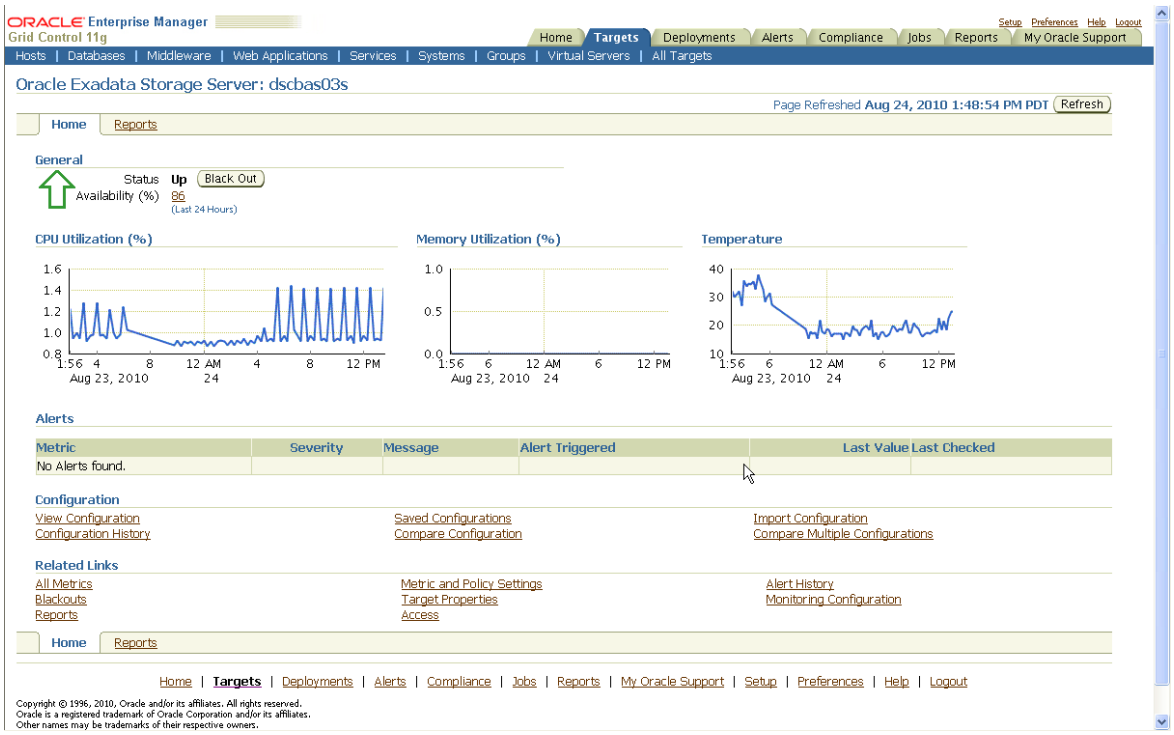


Figure 9. Viewing the Storage Server 3

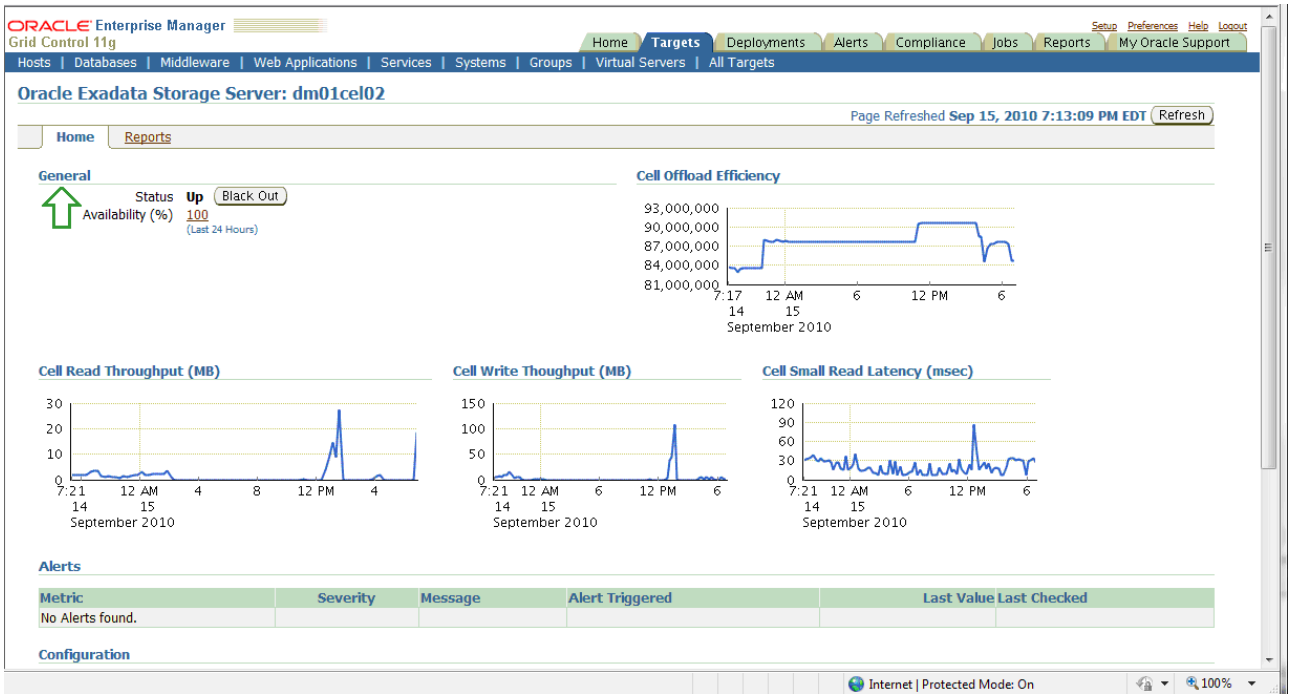
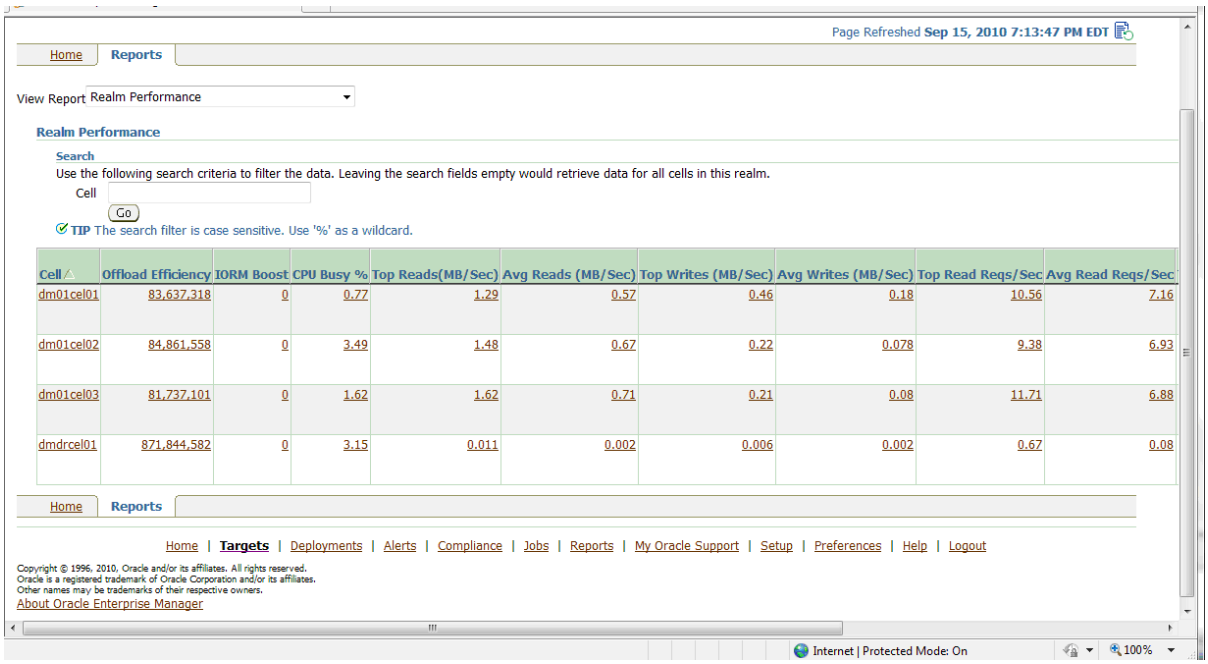


Figure 10. Viewing Cell Performance Information



Cell	Offload Efficiency	IORM Boost	CPU Busy %	Top Reads(MB/Sec)	Avg Reads (MB/Sec)	Top Writes (MB/Sec)	Avg Writes (MB/Sec)	Top Read Reqs/Sec	Avg Read Reqs/Sec
dm01cel01	83,637,318	0	0.77	1.29	0.57	0.46	0.18	10.56	7.16
dm01cel02	84,861,558	0	3.49	1.48	0.67	0.22	0.078	9.38	6.93
dm01cel03	81,737,101	0	1.62	1.62	0.71	0.21	0.08	11.71	6.88
dmdrcel01	871,844,582	0	3.15	0.011	0.002	0.006	0.002	0.67	0.08

Figure 11. Viewing Cell Performance Information

Figure 12. Viewing Cell Performance Information

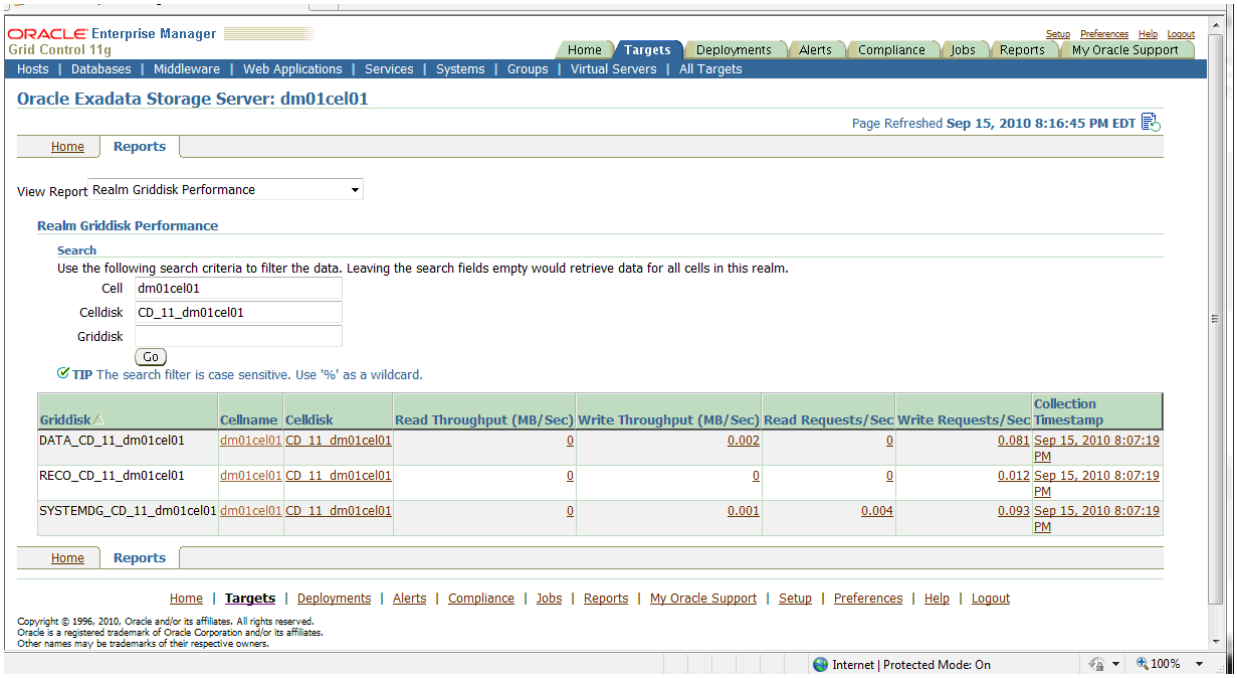


Figure 13. Viewing Cell Performance Information

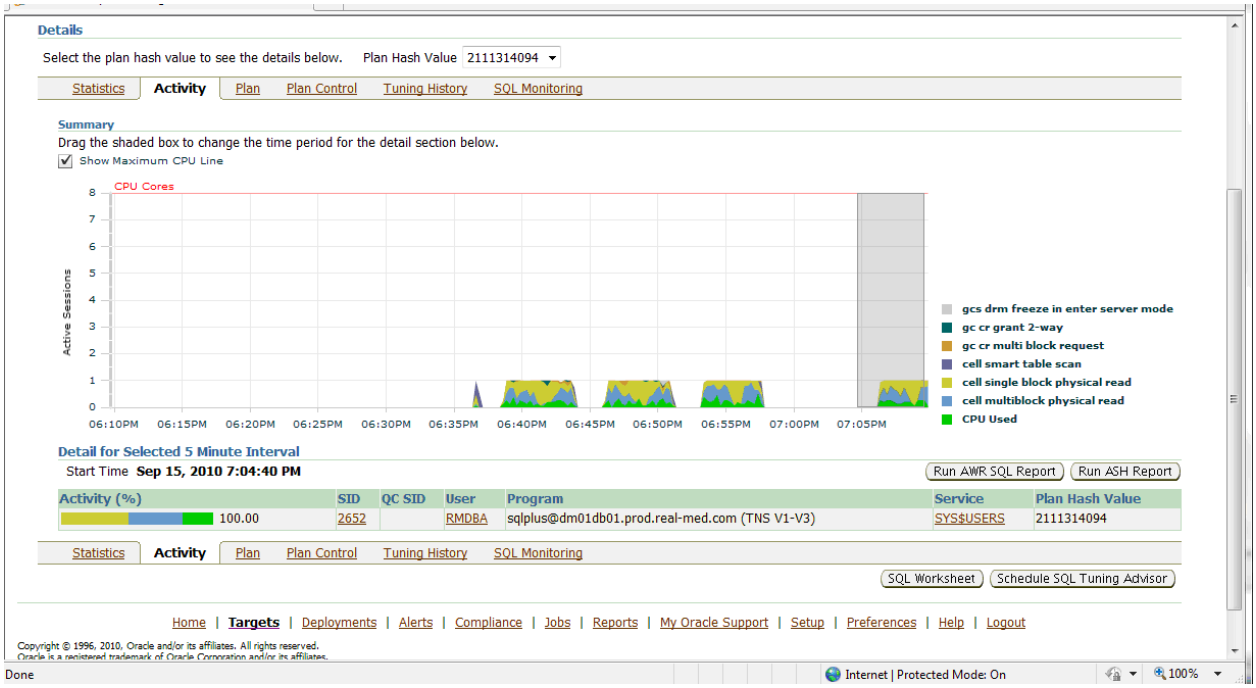


Figure 14. Viewing Smart Cache Performance Gains

You can also simulate your current system before Exadata and after Exadata (simulated). I did this below for a system that was basic and should NOT show any gain. It showed no gain for this simple system (I wasn't able to try a complex one), but I wanted to show how the screen looks that compares it (below).

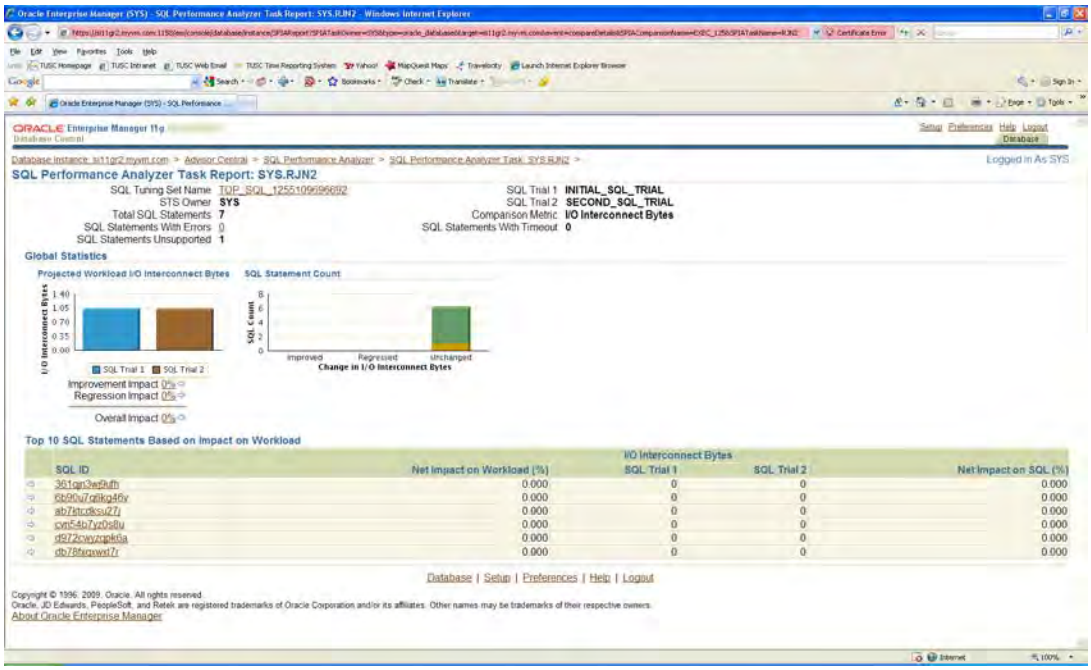


Figure 15. Exadata Simulation

You can see from all of the examples in this section that Oracle continues to leverage the software tools that they've built over the years along with their hardware. They are both leveraging the strength of the Oracle software within the hardware acquisition they've made with Sun.

Resource Management (IORM)

Oracle's latest IORM (I/O Resource Management) tool can be used with Exadata to manage multiple workloads and set resources as you deem necessary for each. While I don't cover this in detail here, please consider the following ways that you set up three instances with different resources. Instance A gets 50%, Instance B gets 30% and instance C gets 20%. We can further breakdown the 50% that Instance A gets into the various user and task related percentages.

Set I/O Resources for Different Instance

- Instance A = 50%
- Instance B = 30%
- Instance C = 20%

Further Set I/O Based on Users and Tasks

- Instance A Interactive = 50%
- Instance A Reporting = 20%
- Instance A Batch = 15%
- Instance A ETL = 15%

You can also continue to use DBRM (Database Resource Manager) as in the past. DBRM has been enhanced for Exadata. It allows management of inter and intra Database I/O. With Inter-DB – We manage via IORM & Exadata storage software. For Intra-DB – We manage via Consumer Groups. You can set CPU, Undo, DOP (Degree of Parallelism), Active Sessions, & much more. Please see the docs for more details. This section is only an overview so that you can see what's possible.

Use ALL Oracle Security Advantages with Exadata

Oracle is known for their incredible security. Their first customer was the CIA ,so they've always been focused on security (over 30+ years). Don't forget to investigate these security options:

- Audit Vault
- Total Recall / Flashback
- Database Vault
- Label Security
- Advanced Security
- Secure encrypted backup- Please Use! (also available: incremental backup with Change Tracking File – much faster)
- Data Masking
- Data Guard
- Failure Groups – (automatic-for storage cell failure)

Best Practices (Subject to Change – Refer to Oracle Docs)

There are many Best Practices that will help you get the most out of Exadata. I'll start with a list of MUST Have's and DON'T Do's! Note that these can change at any time (check Oracle's docs for the latest).

- Must have Bundle Patch 5 (See note: 888828.1 for latest)
- Must have ASM to use Exadata
- Must have the correct data center COOLING! (VERY important – research this)
- 3 tiles with holes for full rack (400 CFM/tile) – don't melt it! (current reqs.)
- Must have the correct power needs
- Must use Oracle Linux 5.3 (x86_64) & Oracle DB 11.2 (currently)
- Must use RMAN for backups

- Consider StorageTek SL500 Tape backup (many positive reviews but pricey)
- Use an ASM allocation unit (AU) size of 4M (currently)
- Don't add any foreign hardware or – No Support!
- Don't change BIOS/Firmware or – No Support!

Next, I'd like to list some of the Best Practices that Oracle has stated. Note that these can change at any time, so please check your docs for the latest. Best practices include:

- "Create ALL" celldisk and griddisks
- Use DCLI to run on ALL Storage Servers at once (helpful & saves time)
- Use IORM for Resource Management
- Decide Fast Recovery Area (FRA) & MAA Needs before you install
- Database 11.2.0.1+ (11.2.1.3.1) and ASM 11.2.0.1+ (minimums currently)
- COMPATIBLE 11.2.0.1+ (minimum currently)
- Logfile size at 32G (Whoa!)
- LMT (Locally Managed Tablespaces) with at 4M uniform extents
- Move Data with Data Pump (usually – but there are many other options)

Summary: Exadata = Paradigm Shift!

We've covered a lot of topics in this paper including the Terminology & the Basics about Exadata, Flash Cache, Storage Index, Smart Scans, Hybrid Columnar Compression (HCC), Enterprise Manager & Grid Control, Enterprise Manager Exadata Simulation, I/O Resource Manager, Security, and Best Practices. What makes it fast is: Fast Hardware, Many CPUs, Fast Flash Cache, Lot's of DRAM on Database Servers and Storage, Compression (save 10x-70x), Partition Pruning (save 10-100x), Storage Indexes (save 5-10x), Smart Scan (save 4-10x), and other features not covered (see Oracle Docs for more information. Exadata is the best way to turn a 1T search into a 500M search or even 50M search. I believe that Exadata is The Real Deal and will drive accelerated hardware innovation by all major hardware vendors. It's Smokin' Fast!!

References

Exadata V2 – Sun Oracle Database Machine, Oracle
Oracle Exadata Implementation Workshop, Oracle Corporation, McLean, Virginia
Oracle Learning Library – multiple sessions/topics (This is VERY good!!)
Oracle 11g R1/R2 Best Features, Rich Niemiec
Advanced Compression with Oracle Database 11g R2 Oracle Corporation, Steven Lu
Oracle10g Performance Tuning Tips and Techniques; Niemiec, 2007
Oracle11g Performance Tuning Tips and Techniques; Niemiec, 2011 (out soon)
Oracle Enterprise Manager Deployment and High Availability Best Practices, Jim Viscusi (Oracle Corporation), Jim Bulloch (Oracle Corporation), Steve Colebrook-Taylor (Barclays Global Investors)
www.tusc.com, www.oracle.com, www.ioug.org

Author Biography

Rich Niemiec is the former CEO and co-founder of TUSC and currently is an advisor to Rolta International Board. Rich served previously as Rolta's President of International EICT (Enterprise Information & Communications Technology) and President of Rolta TUSC – A Rolta Company. Rolta TUSC is a Chicago-based systems integrator of Oracle-based business solutions since 1988; Rolta TUSC was the Oracle Partner of the Year in 2002, 2004, 2007, 2008 & 2010. Rolta is an international market leader in IT-based geospatial solutions, and caters to industries as diverse as infrastructure, telecom, electric, airports, defense, homeland security, urban development, town planning and environmental protection. Rich is the past President of the International Oracle Users Group (IOUG) and the current President of the Midwest Oracle Users Group (MOUG). Rich is one of six originally honored worldwide Oracle Certified Masters. In 2007, he authored the Oracle bestseller "Oracle10g Performance Tuning Tips & Techniques," an update of his previous 2 Oracle

best sellers on Oracle8i and Oracle9i Performance Tuning. Rich was inducted into the Entrepreneurship Hall of Fame in 1998.

Rolta TUSC is an expert level consultancy that helps companies optimize their investment in Oracle technology. We provide integrated functional and technical solutions since 1988 in the areas of Oracle's E-Business Suite, Business Intelligence/Data Warehousing, Custom Development, Managed Services/Remote DBA, Database Services, Training & Mentoring and Oracle Licensing. Rolta also has software that works with the Oracle database in the areas of GIS, SOA Services and Business Intelligence. Please report errors in this article to rich@tusc.com. Neither TUSC nor the author warrants that this document is error-free. Rich Niemiec / Rolta TUSC © 2011. This document cannot be reproduced without expressed written consent from an officer of TUSC except Collaborate 2011 may make copies and make this paper available as needed for the conference and proceedings.

All companies and product names are trademarks or registered trademarks of the respective owners.

Mainframe Reliability Using Oracle RAC and Commodity Hardware

Steele Arbeeny, PhD and Director of Strategic Business Technologies GL Associates

INTRODUCTION TO RAC ARCHITECTURE

Even in a down economy, organizations are taking the opportunity to upgrade and enhance key enterprise systems to increase productivity and reduce costs. A database upgrade can often be part of this process. Consequentially, organizations can face the possibility of spending millions of dollars for hardware upgrades, if they want to make their existing iSeries platforms capable of handling new ERP system releases. That would be a significant new investment for a legacy database. However, there is also another cost-effective alternative which utilizes RAC (Real Application Clusters) and commodity hardware.

COSTS OF RAC VS. PROPRIETARY MIDRANGE AND MAINFRAME SYSTEMS

There is a paradigm, where a large organization is asked to invest a million dollars for a legacy system upgrade. What is not properly understood is if the company has other options. One such alternative, which also happens to be a more cost-effective path, is the combination of commodity hardware, Oracle Enterprise Linux, and Oracle Database, using Real Application Clusters (RAC). At first companies can be very skeptical, because it is hard to imagine how a million dollar iSeries upgrade could be replaced by a hardware and software combination at only 10 – 15% of the original estimated cost. However, the RAC technology and infrastructure can be incredibly robust, when applied correctly in the appropriate environments.

KEY COMPONENTS OF AN EFFECTIVE MIGRATION

Effective system design is critical to the success of any IT project. In this sample case study, the benchmarked organization is very large with thousands of users, so the recommended computers are not home PCs. These are serious 64-bit machines with 24 processor cores and 256GB of RAM per node. Coupled with a high performance SAN and cluster interconnect, the configuration represents the pinnacle of what is currently capable with this type of equipment.

REAL WORLD SOLUTIONS AND REDUCED TCO (TOTAL COST OF OWNERSHIP)

The original cluster was designed for this organization to run the databases of two separate instances of their ERP system. As time progressed, though, the company realized its RAC could handle many more databases and still deliver superior performance to the legacy system. Even now, the RAC is continually being optimized by adding databases to this super-cluster, reclaiming unused hardware, and reducing power consumption -- all while reducing support costs and rack space in its data center. Even from a servicing perspective, unexpected benefits are being realized from this type of configuration. The organization can upgrade one node of the database while servicing requests on another. This type of rolling upgrade allows for hardware replacements or upgrades while the cluster still services user requests.

BENEFITS OF AN OPTIMAL INTEGRATION

With the migration to a clustered database on commodity hardware, this organization has reaped huge benefits including savings in hardware, software, support, and power savings. Additionally the company has seen increased performance and decreased down-time. Literally, when using commodity hardware compared to proprietary midrange and mainframe systems, hardware and upgrade costs can be reduced by orders of magnitude. Oracle's Real Application Clusters (RAC) is a key enabling technology that can deliver carrier-grade reliability at a fraction of the cost.

MAKE EVERY PERFORMANCE A HIT WITH COMPUWARE

Get clear insight into end-user experience and make IT a strategic revenue driver with Compuware Business Service Delivery (BSD) solutions.

compuware.com

Compuware®
We make **IT** rock around the world

NYOUG 2011 Sponsors

The New York Oracle Users Group wishes to thank the following companies for their generous support.

EMC (www.emc.com)
Oracle (www.oracle.com)

#1 Middleware

- ✓ #1 in Application Servers
- ✓ #1 in Application Infrastructure Suites
- ✓ #1 in Enterprise Performance Management

ORACLE®

oracle.com/goto/middleware
or call 1.800.ORACLE.1

Copyright © 2009, Oracle. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.