ORACLE

Oracle Database for SAP

Implementing a Data Management Infrastructure for SAP
with Oracle Database Options and Packs

Extract from Oracle for SAP Technology Update 07/2021, Page 17-34

IMPLEMENTING A DATA MANAGEMENT INFRASTRUCTURE FOR SAP WITH ORACLE DATABASE OPTIONS AND PACKS

Introduction

Database Editions

Oracle Database is available in five editions, each suitable for different development and deployment scenarios. However, only Oracle Database Enterprise Edition is certified and supported in SAP environments, as SAP applications are very demanding and cannot be run efficiently without the enterprise computing features provided by Oracle Database Enterprise Edition.

Database Options and Management Packs

In addition, Oracle offers several database options, management packs, and other products that enhance the capabilities of Oracle Database for specific purposes. They extend the power of Oracle Database Enterprise Edition to meet customer- or application-specific requirements in the areas of efficient use of disk space, performance and scalability, high availability, security and compliance, data warehousing, big data, and manageability.

Options and Packs in SAP Environments

This article is about database options and management packs for SAP customers. There are some differences between a pure Oracle Database and an Oracle Database for SAP perspective:

- Even if an option is certified, the use of some of its features may not be permitted. As this is an overview article, not all details can be discussed. If in doubt, check SAP Note 105047.
- Due to peculiarities of the SAP data model or application design, an Oracle Database option or management pack may not be optional, but required. E.g. SAP Business Warehouse (BW) on Oracle Database requires Oracle Partitioning.
- From a pure Oracle perspective, an option or pack is licensed separately. However, an Oracle Database Enterprise Edition license bought from SAP (ASFU) already includes some (but not all supported) options and packs without additional payment. For details see SAP Note 740897.

Structure and Infrastructure

As we said before, database options extend the power of Oracle Database Enterprise Edition in the areas of efficient use of disk space, performance and scalability, high availability, security and compliance, data warehousing, big data, and manageability. In this article, however, we will focus on one single aspect: *Database options help implement structure*. When the amount of data in one single database grows, when data coming from different sources or even multiple, previously independent databases are consolidated in one single data management infrastructure (as in Oracle Multitenant), the unstructured mass of data will become unmanageable at a certain point. Therefore consolidation requires differentiation. Or: *Infrastructure requires structure*.

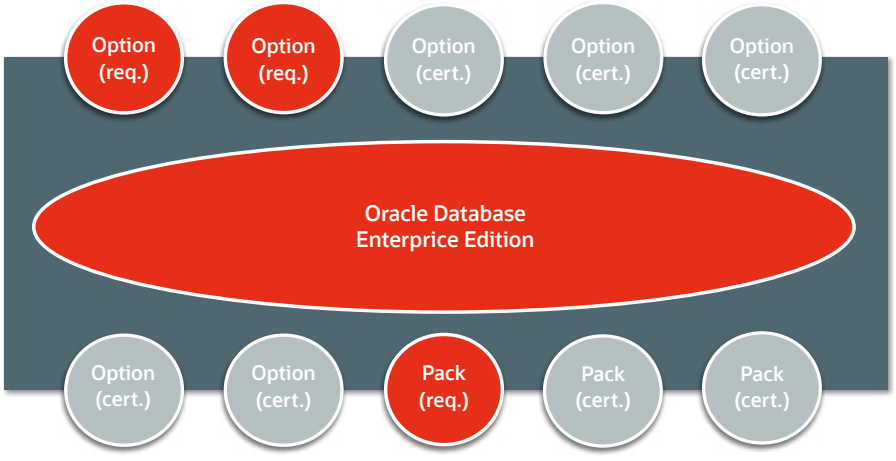

Figure 1: Structure and infrastructure

Table and Index Partitioning

Challenge: In more and more situations today the distribution of the data on disk turns out to be a problem:
(a) Single queries or complex batch jobs accessing a certain subset of the table data need too much time to complete.
(b) Data load (SAP BW) is either slow, because it must update many indexes; or indexes are dropped and rebuilt, in order to reduce load time, but this slows down user queries.
(c) Data archiving results in heavily fragmented databases.
(d) Customer wants to implement information lifecycle management Advanced Compression (Oracle Database 12c)
Value Proposition: Oracle Partitioning divides tables and indexes into smaller units (called partitions) and forces all data to be stored in the appropriate unit. Partitions can be accessed and managed individually and independently from each other. Therefore:
(a) Ideally a query now finds all relevant data in one single partition and can ignore all other partitions („partition pruning"). This can reduce the runtime considerably.

(b) If the indexes defined on a partitioned table are partitioned as well, individual index partitions can be dropped and rebuilt while all other partitions remain untouched.

(c) The data archiving strategy can be based on the partition structure, and this can avoid disk space fragmentation.

(d) Partitioning is one of the basis technologies for information lifecycle management.

Certification/Support: Oracle Partitioning is certified for all SAP NetWeaver applications.

Versions: Oracle Database 11*g* and higher

Implementation: Partitioning (range partitioning) is configured and used by default in SAP BW on Oracle. In SAP OLTP systems, it can either be implemented by customers themselves or by Oracle ACS for SAP.

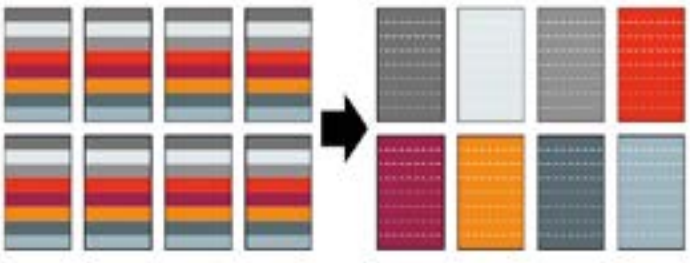

Figure 2: Table Partitioning – Physically store subsets of related data as closely together as possible

Figure 2 explains why storing related data as closely together as possible makes such a difference. It shows database blocks filled with records. The different colors represent criteria such as different months or different locations. And we assume that the applications accessing those data in most cases want to retrieve all records having the same color.

Under these circumstances, the situation shown on the left hand side is the worst that can be imagined: Each database block contains one record of each color. Or, to put it differently, all subsets of records having the same color are distributed over all blocks. This is the worst possible situation from an I/O perspective (a query looking for all records of a certain color needs to read 8 blocks), from a memory perspective (even if all users work with records of the same color, all 8 blocks need to be cached completely in database memory), therefore from a performance perspective (too much I/O means unsatisfactory performance), from a database administration perspective (subsets of records having the same color cannot be managed independently), and from an ILM perspective ("hot" and "cold" data cannot be separated).

The situation shown on the right hand side, in contrast, is optimal (again: under the circumstances described above):

All records having the same color are now stored together in one and the same database block. This is optimal from an I/O perspective (a query looking for all records of a certain color needs to read only 1 block), from a memory perspective (if all users work with records of the same color, only 1 block needs to be cached in database memory), therefore from a performance perspective (significantly less I/O means significantly better performance), from a database administration perspective (subsets of records having the same color can be managed independently), and from an ILM perspective ("hot" and "cold" data can be separated).

Now, multiply the number of records and blocks shown on figure 2. Then a partition is the subset of all blocks containing records of the same color.

Advanced Compression (Oracle Database 11*g* and higher)

Challenge: In more and more cases today the size and the expected future growth of the database becomes a problem. Aspects of this problem include: Storage cost, performance guarantees (SLAs), cloning and backup of database files within a reasonable time.

Value Proposition: Oracle Advanced Compression uses a different format for storing table data. Together with other compression technologies, which come with Oracle Database Enterprise Edition (e.g. Index Key Compression), it helps reduce the database size by 50% or more. This is *the essential* benefit in the sense that this is the effect Advanced Compression is designed for. The benefit from a smaller source database footprint is that the creation of backups and other copies will require less time.

As an *additional* benefit customers using Advanced Compression may see a performance improvement. Additional (as opposed to essential) here means: It may, but it is not guaranteed to happen.

Certification/Support: Oracle Advanced Compression is certified for all SAP NetWeaver applications. Implementations are supported by SAP.

Implementation: Oracle Advanced Compression can be implemented easily in SAP environments, as SAP provides the tool BRSPACE, which is aware of all SAP-specific requirements. For details check SAP Note 1431296.

On the left-hand side, figure 3 shows a typical Oracle Database which forms an integral part of an SAP (in this case: SAP ERP) system. Approximately one third of the allocated disk space is used for indexes (red), and two thirds contain table data (blue). Table data in turn can be divided into structured data (organized in columns) and unstructured data (PDF or image files, table data „clustered“ by SAP).

Oracle Database 11g and higher can compress all three types of data:

- *Index Key Compression* is available for indexes. *Index-Organized Tables (IOTs)* can be compressed as well. Both features are included in Oracle Database Enterprise Edition, and do not require Advanced Compression.
- *OLTP Compression*, a major feature of Advanced Compression, can be used to compress structured table data. It is not restricted to OLTP systems, but can be implemented in SAP BW systems as well.
- *SecureFiles Compression* (an Advanced Compression feature, too) can be used to compress unstructured table data.

If all features are implemented and all appropriate database objects are compressed, customers have seen 55% disk space savings on average. (This assumes a completely reorganized database. If a fragmented database has not been reorganized, the effects of reorganization and compression are combined, in which case customers have seen up to 80% disk space savings).

The maximum disk space savings which can be achieved depend on the characteristics of the data, and the data characteristics depend on the SAP applications used. Usually SAP BW data can be compressed more efficiently than SAP ERP data, and SAP CRM data permit even higher disk space savings.

Oracle Database Advanced Compression offers more than OLTP and SecureFiles compression. Backup files created by RMAN and export files created by Data Pump can be compressed substantially, even if tables and indexes in the production database are already compressed. Additionally, redo log data can be compressed as well before being shipped from the production to a standby database (see the Data Guard section in this article).

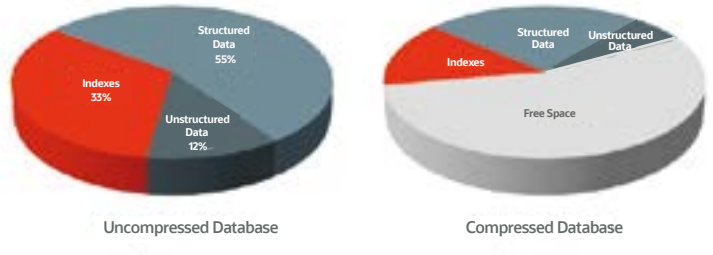

Figure 3: Oracle Database 11g Index Key Compression and Advanced Compression (OLTP Compression, SecureFiles Compression)

Advanced Compression (Oracle Database 12*c* and higher)

Challenge: (a) Data compression in Oracle Database 11*g* has several limitations. In particular, tables having more than 255 columns cannot be compressed.

(b) Data load is slowed down, if target tables are compressed.

(c) No support for automated Information Lifecycle Management.

Value Proposition: Oracle Database 12*c* Advanced Compression lifts the 255-columns limitation and allows more tables to be compressed. Completely new features (Heat Map, Automatic Data Optimization) enable customers to implement deferred data compression and sophisticated Information Lifecycle Management (ILM) strategies.

Certification/Support: The Oracle Advanced Compression features introduced with Oracle Database 12*c* are certified for all SAP NetWeaver applications.

Implementation: See SAP Note 2258061 – Enhancements for ADO/ILM for table conversions or system copy.

Related Feature: Oracle Database 12*c* Hybrid Columnar Compression (which is not included in Advanced Compression, but is a feature available on Oracle's Engineered Systems) provides stronger compression algorithms particularly suited for "cold" (i.e. historical) data. HCC (with row-level locking) can be used in SAP environments.

Limitations removed

In Oracle Database 11*g*, the index and table compression features have several limitations. That is why Oracle Database 12*c* Advanced Compression offers a new, more efficient index compression algorithm (Advanced Index Compression) and increases the maximum number of columns for tables to be compressed. For details see the article "Oracle Database 12*c* for SAP – Roadmap and Base Certification Features."

Heat Map and Automatic Data Optimization

In addition to those improvements, Oracle Database 12*c* Advanced Compression comes with two completely new features. *Heat Map* automatically tracks modification and query timestamps at the row and segment levels, providing detailed insights into how data is being accessed. *Automatic Data Optimization (ADO)* automatically moves and compresses data according to user-defined policies based on the information collected by Heat Map.

Deferred Compression

Based on the information provided in the *Advanced Compression (Oracle Database 11g)* section, it might seem that compression simply reduces the disk space needed, and has nothing to do with database structure. This is an illusion. Even in Oracle Database 11g, we need to distinguish between tables that benefit and tables that do not benefit from compression (if this were not the case, compression could be made the default), i.e. between tables that should and tables that should not be compressed.

Nevertheless, this is still a very basic and inflexible distinction. Take, e.g., an SAP BW table that is used for data load. On the one hand, such a table *could* be compressed, because for most of the time it is accessed in read-only mode. On the other hand, it *should* not be compressed, because this would slow down the load operations considerably. In Oracle Database 11*g* the recommendation is: Do not compress such a table.

Heat Map and Automatic Data Optimization allow you to introduce a new differentiating parameter: If a table or partition should be compressed, *when* would you like it to be compressed? In Oracle Database 11*g* compression happens immediately or not at all. In Oracle Database 12*c* and higher you can specify that data should be loaded today and (automatically) compressed tomorrow.

Information Lifecycle Management

Even more parameters can be introduced thanks to the new features in Oracle Database 12*c* Advanced Compression. One of them is location. When you have "hot" (current) and "cold" (historical) data in your database, and you have two different types of storage, too, then you can ask the question: *Where?* Where would you like which data to be stored?

Using Partitioning in addition to Advanced Compression, you can – or rather: you can have the system – move data from one tablespace (= storage tier) to another, when they "cool down", thus freeing up space on a more expensive storage tier for more important ("hot") data. This is called *(automatic) storage tiering*

Hybrid Columnar Compression (HCC)

Finally, if you run Oracle Database 12*c* or higher on Oracle Exadata, you can introduce the question: *How?* How, i.e. using which algorithm, do you want data to be compressed? Where would you like which data to be stored?

	Oracle Database 11g	Oracle Database 12c
Partitioning	Certified	Certified
Basic Table & Index Compression	Certified	Certified
Hybrid Columnar Compression	Not Certified	Certified (Prerequisite: ADO)
Tool Support for Identifying Relevant Objects	None	Heat Map (requires ACO)
Tool Support for Storage/Compression Tiering	None	Automatic Data Optimization (= ADO; requires ACO)

Figure 4: Oracle Database 12c Advanced Compression – Support for Information Lifecycle Management (ILM)

In addition to OLTP and SecureFiles Compression, these Engineered Systems support Hybrid Columnar Compression. As the name implies, this technology utilizes a combination of both row and columnar methods for storing data. This hybrid approach achieves the compression benefits of columnar storage, while avoiding the performance shortfalls of a pure columnar format. The compression ratios that can be achieved by using HCC are much higher than those seen with "normal" compression. Therefore HCC is particularly suited for "cold" data.

Due to the missing row-level locking feature, it had not been possible to certify Oracle Database 11*g* Hybrid Columnar Compression for use in SAP environments. In Oracle Database 12*c* and higher, however, this feature is available on Oracle Exadata and Oracle SuperCluster. On this machine it is now possible to implement *(automatic) compression tiering*. This means that, while "hot" data remain uncompressed, "warm" data may be compressed using the standard compression algorithms (Advanced Compression) and "cold" data using Hybrid Columnar Compression.

Oracle Database In-Memory

Challenge: In more and more systems, meeting analytics performance requirements turns out to be a challenge. This is true for long-running queries in BW. However, it can also happen in OLTP systems, e.g. if a very flexible implementation of operational planning/reporting allows users to create many, slightly different query variants.

Value Proposition: Oracle Database 12*c* In-Memory allows administrators to dedicate a certain amount of database server memory to the Column Store – a memory structure that stores data in column format instead of in row format. Setup of the Column Store is fast and easy. Having data available in column format can improve query performance substantially.

Certification/Support: Oracle Database In-Memory is certified for all SAP NetWeaver applications.

Versions: Oracle Database 12*c* and higher

Implementation: For an overview and pointers to more detailed documents see SAP Note 2178980.

Memory: The New Dual-Format Architecture

Oracle Database has traditionally stored data in a row format. This format is ideal for *online transaction (OLTP)* systems, as it allows quick access to all columns in a record. *A column format* database stores each of the attributes about a transaction or record in a separate column structure. This format is ideal for *analytics*, as it allows for faster data retrieval when only a few columns are selected but the query accesses a large portion of the data set.

But what happens, when your system is characterized by a mixed workload? Up until now you have been forced to pick just one format and suffer the trade-off of either sub-optimal OLTP or sub-optimal analytics performance. The only way to optimize for both OLTP and analytics had been to copy data from OLTP systems to analytic systems using complex ETL processes that add a great deal of expense and latency.

Oracle Database In-Memory optimizes both analytics and mixed workload OLTP, delivering outstanding performance for transactions while simultaneously supporting real-time-analytics, business intelligence, and reports. This breakthrough

capability is enabled by the dual format architecture of Oracle Database In-Memory. This architecture eliminates the trade-off by representing tables simultaneously using traditional row format and a new in-memory column format. The Oracle SQL Optimizer automatically routes analytic queries to the column format and OLTP transactions to the row format, transparently delivering best-of-both-worlds performance. Oracle Database automatically maintains full transactional-consistency between the row and the column formats, just as it maintains consistency between tables and indexes today.

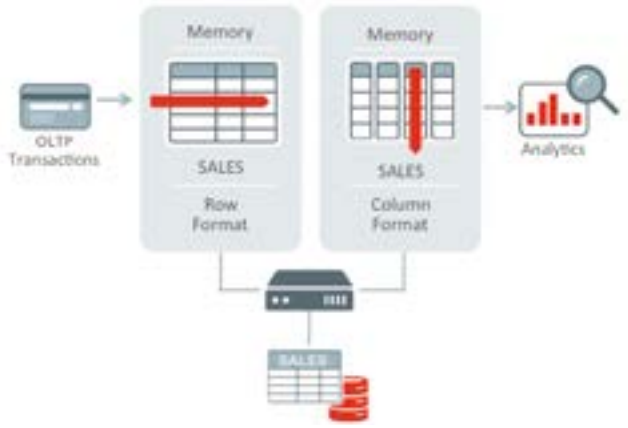

Figure 5: Oracle Database In-Memory – Dual memory format, single disk format

Disk: Nothing Has Changed

The new column format is a pure in-memory format. Tables are stored on disk using Oracle's existing row-based (or – on Engineered Systems – hybrid columnar) formats. Since there is no persistent columnar storage format, there are no additional storage costs or storage synchronization issues. Nor is there a need to modify the database. Oracle Database In-Memory can be implemented without a database migration or a table reorganization.

As a result, the new Oracle Database In-Memory feature is fully compatible with existing standard or optional database features such as table and index compression, table encryption and table partitioning. It is also compatible with the scale-out architecture provided by Real Application Clusters (RAC) and with all existing high availability technologies (such as Data Guard). These features work exactly the same way with and without Oracle Database In-Memory.

Easy to Implement and Manage

In addition to being compatible at the database feature and application level, Oracle Database In-Memory is easy to implement and manage. Enabling Oracle Database In-Memory is easy as setting the size of the in-memory column store and identifying tables or partitions to bring into memory. Background processes populate data from storage into in-memory columns while the database remains fully active and accessible.

Fine-Grained Control

An easy start based on intelligent defaults for typical situations – this is what Oracle customers expect. In addition, however, Oracle customers expect mechanisms, which allow for fine-grained control and tuning. Oracle Database In-Memory provides such mechanisms. Examples are:

- Tables can contain "cold" data, which are neither up-dated anymore nor accessed by queries. If those tables are very large, it would be a waste of memory to keep them completely in the in-memory column store. Therefore administrators may want to restrict the population process to the data really needed by DSS queries. Table partitioning allows them to make this happen. If the table is partitioned in a useful way (e.g. by month), this internal structure can be used to define a *horizontal* subset of the table data to be kept in the in-memory column store.

- One or several table column(s) may contain data that are not relevant for DSS queries. Again the database administrator may wish to restrict the data to be kept in the in-memory column store, but in this case the goal is to define a vertical subset of the table data, i.e. to exclude one or more columns from the population process. And again it is possible to make this happen, because Oracle Database In-Memory allows administrators to specify different in-memory characteristics for different table columns.

- The Oracle Database has been optimized and tuned for decades to scale-up on SMP servers. Large SMP servers are well suited for in-memory workloads, too, because all memory is accessible to all processors over an extremely high speed Back-Plane. In addition to being able to scale up, Oracle Database In-Memory can also scale out to very high memory and CPU capacities by using all of the memory and processors in a cluster of servers (RAC). In such environments, all objects populated into memory will by default be *distributed* across all in-memory column stores in the cluster. On Oracle Engineered Systems, objects can also be *duplicated*. This means that an object (or part of an object, e.g. a partition) populated into the in-memory column store will have a mirrored copy placed on one of the other nodes in the RAC cluster. Duplicating data provides in-memory fault tolerance as it ensures data is still accessible via the in-memory column store even if a node goes down or is taken down for maintenance.

Real Application Clusters (RAC)

Challenge: When the workload on a database server increases (due to new application versions, additional applications, or more users), the traditional solution is to replace the existing server with a bigger one (scale-up). Big servers, however, are very expensive.

In order to guarantee high availability of the database server, traditionally a failover cluster is implemented. Such a solution, however, has at least two disadvantages:

(a) A failover cluster relies on the concept that, at any given moment, only one database instance running on one machine can be active. The other machine (most probably an expensive server, too) is always idle.

(b) When a problem on the primary machine is detected, an Oracle Database Server instance needs to be started on the secondary machine. In this particular situation, startup can take up to 30 minutes – which means: up to 30 minutes unplanned downtime.

Value Proposition: Real Application Clusters (RAC) allows multiple instances to be up and running and to access the same database at the same time. As these instances can and in most cases do run on different machines, customers have the option to implement a scale-out approach: 4, 6, or 8 small servers can handle the same workload as one big server. However, they are much cheaper. And they can be added as needed.

In this architecture, all Oracle instances are up and running at the same time. Therefore no restart is required. If one of the RAC servers fails, the other instances can take over. A reconnect of the affected users is a matter of seconds, not of minutes.
To summarize all this in a few words: The value proposition of Oracle Real Application Clusters combines workload distribution, scalability, high availability, better manageability, and cost savings.

Certification/Support: Oracle Real Application Clusters is certified for all SAP NetWeaver applications.

Versions: Oracle Database 11*g* and higher

Implementation: Customers can use any general-purpose machine (Unix, Linux, or Windows) certified by SAP to build a RAC system. In addition, Oracle offers Engineered Systems (Exadata, SuperCluster), which make the implementation considerably easier. Oracle Grid Infrastructure provides a set of base technologies that simplify implementation and help save money.

Figure 6 illustrates the RAC benefits which have been mentioned before:
Scalability: Using RAC, the scale-out approach, which is always supported on the SAP Application Server level, can be implemented on the database level too. In this example, 5 SAP Application Server instances, running on 5 different machines, are connected to 4 Oracle Database Server instances, running on 4 different machines.

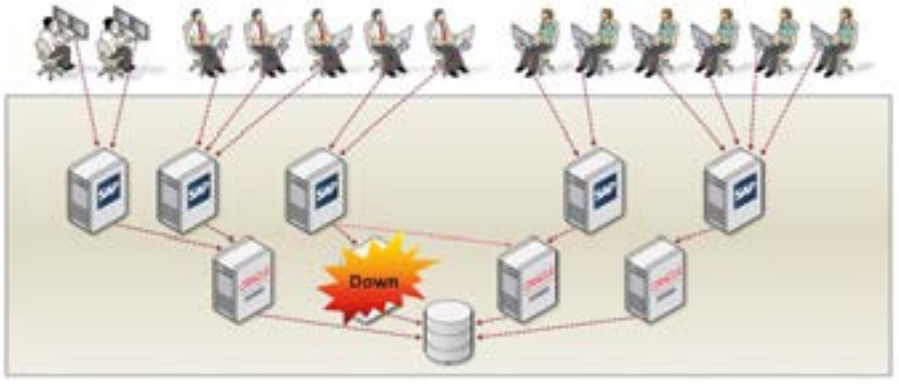

Figure 6: Real Application Clusters (RAC) for scale-out and immediate (instance) failover

High Availability: If one of the Oracle instances goes down, the affected SAP instance(s) is/are automatically reconnected to one of the available Oracle instances. After this operation users can continue their work. The failover occurs within seconds.

Oracle Grid Infrastructure provides the base technologies that are required to enable RAC. It can be divided into two main components:

- In order to let multiple Oracle instances access the database files at the same time, a cluster file system is needed. Therefore Oracle provides *Oracle Automatic Storage Management (ASM)*. Unlike other, third-party cluster file systems, it is optimized for Oracle Database files, and it is available for free.
- *Oracle Clusterware* is the cross-platform cluster software required to run the RAC option for Oracle Database. It enables the nodes to communicate with each other, allowing them to form a cluster of nodes which behaves as a single logical server. Similar to Oracle ASM, which eliminates the need for a third-party cluster file system, Oracle Clusterware eliminates the need for third-party cluster management software.

Oracle Clusterware can provide high availability and resource management for SAP resources just as it does for Oracle resources. Therefore Oracle/SAP Development has created an Oracle Clusterware tool, *SAP Control (SAPCTL),* to enable customers to easily manage SAP high availability resources.

Data Guard and Active Data Guard

Challenge: RAC provides high availability by multiplying the number of Oracle instances. Such high availability, however, is restricted to the instance level. Even in a RAC-based system, the database remains a single point of failure. This means that DBA errors, data corruption, server or data center failures can make the whole system unavailable.

Value Proposition: Data Guard removes this single point of failure. The technology allows customers to set up a standby (shadow) database as a copy of the primary (production) database and then keep the two databases synchronized. Please note that *Data Guard* is included in Oracle Database Enterprise Edition. It is not an option.

However, *Active Data Guard* is an option. In Oracle Database 11*g* it offers additional features such as Automatic Block Repair and Fast Incremental Backup.

Active Data Guard Far Sync, the main new feature with Oracle Database 12*c*, allows customers to combine high performance (a characteristic of asynchronous log shipping) and zero data loss (a characteristic of synchronous log shipping).

Certification/Support: Oracle *Data Guard* is certified for all SAP NetWeaver applications. However, only physical standby databases are supported, logical standby databases are not.

Oracle Active Data Guard is certified for all SAP NetWeaver applications. However, Real-Time Query is not possible in SAP environments, because even report generation is not a read-only operation.

Versions: Oracle Database 11*g* and higher

Implementation: Standard Oracle setup procedures apply. In the white paper "Oracle Standby Database" SAP describes BR*Tools support.

Data Guard can provide both zero data loss protection and near-immediate restoration of service should a production database become unrecoverable for any reason. This is accomplished using the combination of Data Guard synchronous redo transport and a replication-aware apply process at the standby database. The impact that any synchronous replication method can have on database performance, however, often makes it impractical to implement zero data loss protection when large distances separate the primary and replica database(s). Rather than impact database performance, many enterprises will compromise on data protection by implementing asynchronous replication and accept that an unrecoverable outage will result in varying degrees of data loss.

Active Data Guard Far Sync, a new capability with Oracle Database 12c, eliminates the need to compromise by extending zero data loss protection to a replica database located at any distance from the primary database. Far Sync provides zero data loss protection for a production database by maintaining a synchronized standby database located at any distance from the primary location, and can do so without performance impact and with minimal cost or complexity. A new type of Data Guard destination called a far sync instance receives changes synchronously from a primary database and forwards them asynchronously to a remote standby. Production can be quickly failed over, manually or automatically, to the remote standby database with zero data loss.

A far sync instance is a light-weight entity that manages only a control file and log files. It requires a fraction of the CPU, memory, and I/O of a standby database. It does not have user data files, nor does it run recovery. Its only purpose is to transparently offload a primary database of serving remote destinations. A far sync instance can save network bandwidth by performing transport compression using Oracle Advanced Compression.

Take for example an existing asynchronous Data Guard configuration with a primary in Boston and a standby in San Francisco. Upgrade to zero data loss simply by using Active Data Guard to deploy a far sync instance within synchronous replication distance of Boston (less than 150 miles). There is no disruption to the existing environment nor is there any requirement for proprietary storage, specialized networking, more database licenses, or complex management.

Figure 7: Active Data Guard Far Sync – High performance, zero data loss across large distance WAN

Oracle Multitenant

Challenge: Many SAP landscapes consist of a few large and a considerable number of small or very small systems. However, the existence of many small SAP systems based on as many independent database servers has several disadvantages:

- Many small systems (even virtualized ones) use too many hardware resources (memory, CPU)
- Too much time is spent for the administration of so many small database systems

Value Proposition: Oracle Database Multitenant introduces a new architecture that enables customers to easily consolidate multiple databases, without changing their applications. This new architecture delivers all the benefits of managing many databases as one, yet retains the isolation and resource prioritization of separate databases. Oracle Multitenant reduces resource consumption by separating "container" and "pluggable" databases. It simplifies administration by moving standard operations to the "container database" level.

Certification/Support: Oracle Multitenant can be used in SAP environments.

Version: Oracle Database 12*c* and higher

Implementation: For more information please see SAP Notes 2336881, 2335850, and 2333995.

Consolidation Approaches

Large enterprises may use hundreds or thousands of databases. Often these databases run on different platforms on multiple physical servers. A database may use only a fraction of the server hardware capacity. This is an expensive approach which fails to maximize the usage of both the hardware and human resources.

A typical response to the management problem is to place *multiple databases on each server* (either as direct installs or using virtual machines). The problem is that the multiple database instances do not share background processes, system and process memory, or Oracle metadata. Another response is to *logically separate the data into schemas* (schema consolidation). The problem is that these virtual entities are difficult to manage, secure, and transport.

Oracle Multitenant Architecture

Oracle Database Multitenant is based on an approach called *database consolidation*. It delivers a new architecture that allows one single Container Database (CDB) to hold many Pluggable Databases (PDBs). See figure 8.

An existing database can simply be "plugged into" a CDB. At any time, then, it can be unplugged and plugged into another CDB. Unplug/plug is even supported across Oracle Database software versions.

From the point of view of the client application connecting to the database server via Oracle Net, the PDB is the database. A PDB is fully compatible with a non-CDB – a rule also known as the PDB/non-CDB compatibility guarantee.

Resource Utilization and Resource Management

The many PDBs in a single CDB *share its memory and background processes*. This enables consolidation of many more databases compared to the old architecture, offering similar benefits to schema-based consolidation but with none of the major application changes required by that approach.

Horizontal partitioning of the Oracle data dictionary (a conceptual partitioning, not a physical table partitioning) removes the need to store and manage system-wide metadata in every single database. The "lower" half (implemented in the CDB) holds the *system-wide metadata* – and nothing else, while the "upper" halves (implemented in the PDBs) hold *application-specific metadata* – and nothing else.

Creating pluggable databases, moving pluggable databases between containers, and cloning pluggable databases are done with new SQL commands and take just seconds. When the underlying file system supports *thin provisioning*, many terabytes can be cloned almost instantaneously.

Sharing of background processes, memory structures, system-wide metadata, and database files results in considerably *decreased resource consumption*. In addition, Oracle Database Resource Manager is extended with specific functionality to *control the competition for resources* between PDBs within a CDB.

Manage Many Databases as One

By consolidating existing databases as pluggable databases, administrators can manage many databases as one. Benefits include:

- The investment of time and effort to patch one CDB results in patching all of its many PDBs. To upgrade all PDBs hosted in a CDB, simply upgrade the CDB, and all hosted PDBs are upgraded "in-place".
- Instead of executing separate database backups, administrators only require to back up their database at the CDB level. In other words, all PDBs consolidated into a container will be backed up as one, and administrators retain the flexibility to perform recovery operations at individual PDB level, if required.
- Administrators maintaining standby systems in another data center (using Data Guard or Active Data Guard) will only need to set up a standby configuration at the CDB level, to replicate all PDBs consolidated in that container.

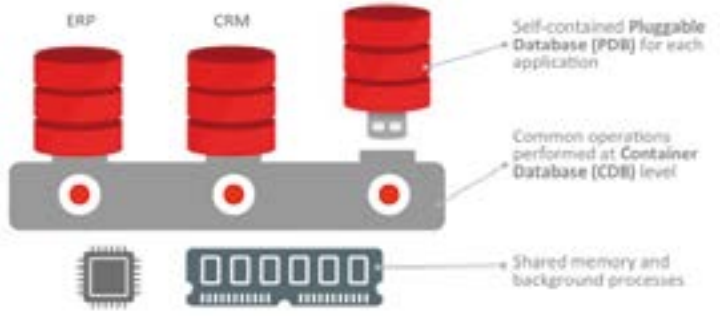

Figure 8: Oracle Multitenant – New architecture for consolidating databases and simplifying operations

Oracle Advanced Security

Challenge: In order to read or update data in an Oracle Database that is the data store of an SAP application, the obvious and only choice for legitimate users is this particular application. Attackers, however, who want to bypass SAP's user management and access control, could use either a network sniffing tool to capture data in transit or some kind of file editor to read data at rest, i.e. in a database file copy.

Value Proposition: Oracle Advanced Security is a bundle of features that allow administrators to encrypt data and make it harder for attackers to understand what they see. Oracle Network Encryption can be used to protect data in transit, while Oracle Transparent Data Encryption and Backup Set Encryption protect data in the production database files as well as their backup copies.

Certification/Support: Oracle Advanced Security is certified for all SAP NetWeaver applications. Implementations are supported by SAP.

Versions: Oracle Database 11*g* and higher

Implementation: Advanced Security features are activated either via the Oracle Net configuration or using SAP's BR*Tools. Details can be found in SAP Notes 973450 (all versions), 974876 (11g), 2591575 (12c), and 1324684 (all versions)

Protecting Data in Transit: Oracle Network Encryption

In SAP environments, users do not directly connect to the Oracle Database server. They connect to an SAP application server instance, and the SAP application server instance in turn connects to the Oracle Database server. So, in this case, the application server instance is the Oracle client, and Oracle's network encryption encrypts all data traveling between application server and database server.

Oracle's network encryption requires Oracle software, which is not installed on end-user devices. Therefore other technologies or products must be used to protect the communication between SAP users and SAP application servers.

However, people do not only attempt to read data in transit, they can also try to intercept and modify them. Therefore, in addition to network encryption, Oracle also supports crypto-checksumming to ensure data integrity. Both encryption and crypto-check- summing are completely transparent to the application, and in both areas the system administrator can chose between several algorithms.[1]

[1] Please note: As of June 2013, network encryption and crypto-checksumming are no longer part of Oracle Advanced Security. They are included without additional cost in all licensed editions of all supported releases of the Oracle database.

Protecting Data at Rest: Oracle Transparent Data Encryption

Oracle Transparent Data Encryption (TDE) is applied to data in the files which make up the production database (as opposed to backup files, which will be discussed in the next paragraph). As the name indicates, TDE is transparent to the application; no application changes arer required. Starting with SAP NetWeaver version 7.20, you can use BRSPACE to set the encryption attribute on a tablespace level. BRSPACE can also be used to manage the wallet, which stores the encryption keys.

Transparent Data Encryption comes in two flavors. The first one (available since Oracle Database 10*g*) is called Column Encryption, because you select just a few of the many SAP tables, or even individual columns of these tables that contain sensitive data and encrypt them. Everything else remains unencrypted. The second one (available since Oracle Database 11*g*) is called Tablespace Encryption. It allows you to encrypt complete tablespaces, which may contain hundreds, thousands, or tens of thousands of tables.

Protecting Data at Rest: Oracle Backup Encryption

If you decide to use column encryption, consider to use encryption with your database backup. It is generally much easier to steal backups of the database files than directly from the production database itself. Therefore, the third set of features in Oracle Advanced Security is related to backup encryption.

If you simply backup your database files, only those columns that are encrypted in the production database files are encrypted in the backup copies. However, combining Oracle Recovery Manager (Oracle RMAN) and Oracle Advanced Security, whole backup sets (that is, all data) can be encrypted.

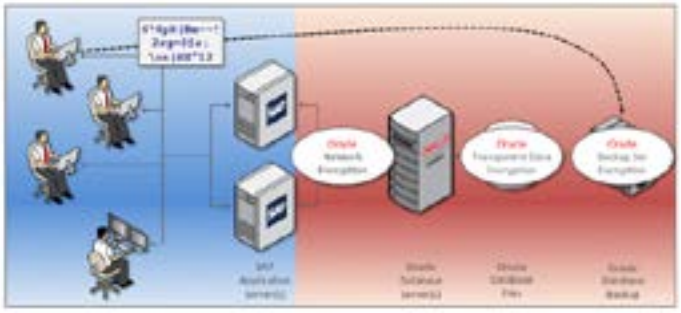

Figure 9: Oracle Advanced Security – Encrypt data in transit or at rest

Database Vault

Challenge: Data encryption does not help, if attackers do not use third-party tools, but Oracle tools to bypass the SAP applications. This is particularly dangerous in the case of privileged database users (database administrators). And it is particularly relevant, if database administration is outsourced or data are stored in the cloud.

Value Proposition: Oracle Database Vault replaces the traditional database privilege management strategy with a new, more flexible and more powerful one. It goes far beyond traditional user–privilege or user–role correlations.

Oracle Database Vault allows companies to implement and enforce concepts such as the segregation of duties or the four eyes principle.

Certification/Support: Oracle Database Vault is certified for all SAP NetWeaver applications. Implementations are supported by SAP.

Versions: Oracle Database 11*g* and higher

Implementation: For detailed instructions see SAP Notes 1597194 (11*g*) and/or 2218115 (12*c* and higher).

Privileged database users – like database administrators – can use DBA tools and directly connect to the database, thus bypassing SAP's security checks (see figure 10). Against this kind of threat, encryption does not help. If someone is able to connect successfully using a sufficiently privileged account, and if he or she then sends a query, the Oracle Database will generously deliver the result set to this user. If the requested data is encrypted, Oracle will decrypt it. From an Oracle Database perspective the request sent by this user seems to be a perfectly valid request.

This can happen, because traditionally, if you were explicitly granted a sufficient number of system privileges, you implicitly received object privileges for all tables as well. For decades people found this acceptable. Recently, however, companies began to ask: Is it really necessary and is it really desirable that a database administrator, who is supposed to manage the database structure, is by default able to read (and even change) all data in the database?

Oracle Database Vault

A solution to this problem requires a new privilege management strategy in the database. This strategy should continue to provide system privileges and object privileges, but it should get rid of implicitly granted object privileges.

This is exactly what Oracle Database Vault does. It replaces the traditional, somewhat clumsy privilege management strategy with a new, more flexible one. It eliminates all implicit grants and instead provides a means to explicitly define access rights as well as the circumstances under which they are effective. This goes far beyond traditional user–privilege or user–role correlations.

Oracle Database Vault allows companies to implement and enforce concepts such as the segregation of duties or the four eyes principle.

Oracle Database Vault for SAP

Oracle Database Vault, as sold by Oracle, is just a tool box. It is true that it comes with predefined realms and roles, but those are realms for system tables and very general (if fundamental) roles. These predefined components allow Oracle Database Vault to be functional and allow you to use it, but they do not protect your application-specific data. That is because Oracle does not know anything about your applications and your data. Oracle can only give you a toolbox, and it is up to you to determine your security requirements and translate these requirements into an access control policy.

But there is an important difference. As long as customers use home grown applications, it is simply not plausible for Oracle to do more. However, if thousands or tens of thousands of companies use a certain standard application and the security requirements of all these companies are (at least to a certain point) identical, because they are results of the application design, then it makes much more sense for Oracle to analyze the requirements and implement a basic security policy.

Oracle has actually done this, saving customers the time that is needed to implement the boring basics of an application-specific security policy, and protecting these companies from forgetting to implement some of the basics. As of today, Oracle provides a whole family of predefined application-specific Oracle Database Vault policies, and Oracle Database Vault for SAP is a member of this family.

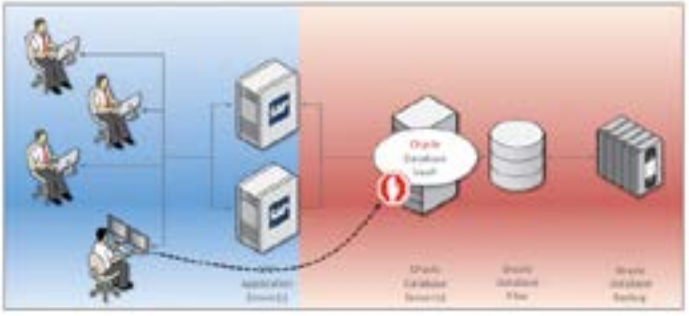

Figure 10: Oracle Database Vault – Privileged user access control and analysis

Real Application Testing (RAT)

Challenge: Database software patches or upgrades, modified database server configurations, and the implementation of new database features/options can influence performance, availability and security of the database server and the whole system. In particular if the implementation must take into account customer- or application-specific characteristics, administrators will want to know in advance how the new feature or configuration works in the production system.

Value Proposition: The main problem with many test systems is that the workload applied to them is smaller than or different from the workload of the production system, and that a new feature or configuration works well in the test system, but not in the production system. Therefore Oracle Real Application Testing allows customers to capture production database workload and replay it on a test system. Combining these two steps, allows for the true effects of the changes to be understood using a real-life workload before they are implemented in production systems.

Certification/Support: Oracle Real Application Testing is certified for all SAP NetWeaver applications. Implementations are supported by SAP.

Versions: Oracle Database 11*g* and higher.

Implementation: As this is a database-only feature, no SAP tool support is required. For details see SAP Note 1426980.

Oracle Real Application Testing enables you to perform real-world testing. By capturing production database workloads and assessing the impact of system changes before production deployment, it minimizes the risk of instabilities associated with changes. Oracle Real Application Testing comprises two components: *Database Replay* and *SQL Performance Analyzer.*

Database Replay

Load testing today is generally done using tools that allow testing teams to generate synthetic workloads based on what they expect users to do on a system. These workloads can then be replayed by application virtual users, which simulate the end users by submitting requests to the application. Although widely used, this approach has a number of shortcomings when it comes to testing database level changes:

- Creating the synthetic workload can take a considerable time and requires programming expertise.
- User behavior is not well understood, so many possible workflows are often missed in the synthetic tests.
- Production scale database concurrency is near impossible to simulate with these tools.
- A full application stack is required for testing as these tools simulate end users.

The Database Replay feature included in Oracle Real Application Testing provides DBAs and system administrators with the ability to faithfully, accurately and realistically rerun actual production workloads, including online user and batch workloads, in test environments. By capturing the full database workload from production systems, including all concurrency, dependencies and timing, Database Replay enables you to realistically test system changes by essentially recreating production workloads on the test system. This is something that a set of scripts can never duplicate. With Database Replay, DBAs and system administrators can test:

- Database upgrades, patches, parameter and schema changes, etc.
- Configuration changes such as conversion from a single instance to RAC, ASM, etc.
- Hardware and operating system migrations.

SQL Performance Analyzer

Database Replay delivers half of what Oracle calls Real Application Testing; the other half is provided by another tool, SQL Performance Analyzer. The main difference between these two tools is the scope involved: Whereas Database Replay applies to the capture and replay of all activities in a database, SQL Performance Analyzer allows you to capture specific SQL statements and replay them. The latter offers a significant advantage for SQL tuning, because you can tweak the SQL statement issued by an application and assess its impact.

SQL Performance Analyzer (SPA) can predict and prevent SQL execution performance problems caused by environment changes. It provides a granular view of the impact of environment changes on SQL execution plans and statistics by running the SQL statements serially before and after the changes.

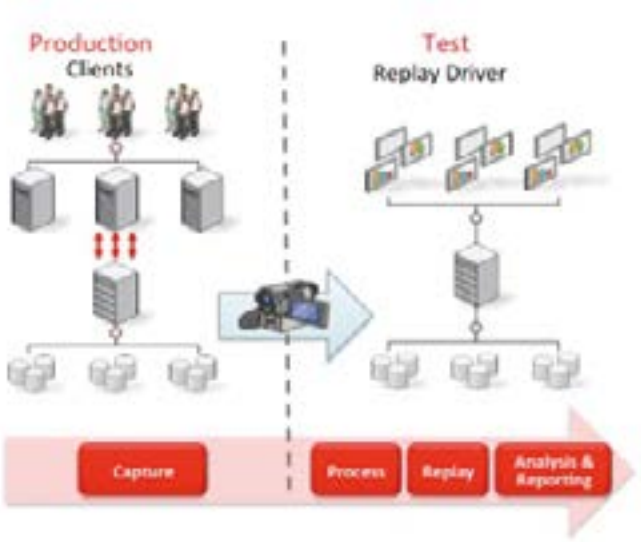

Figure 11: Real Application Testing (RAT) – Capture and replay real database workload

Enterprise Manager Packs

Challenge: Monitoring and managing a complete IT infrastructure often can be challenging. Administrators may end up with a considerable number of management tools designed for this or that particular purpose.

Value Proposition: Oracle, being a vendor of the whole IT stack from hardware via databases to applications, realized that there is a need for an integrated enterprise management tool. However, in order to avoid unnecessary complexity, Oracle Enterprise Manager Grid/Cloud Control is divided into a base product and several packs dedicated to special features. With or without packs – Oracle Enterprise Manager always provides a complete view of the IT landscape.

Certification/Support: Generally speaking, SAP on Oracle customers have a choice. On the one hand there is the BR*Tools family of management tools provided by SAP. These tools include SAP-specific knowledge and are therefore easy to use. On the other hand there are the powerful Oracle Enterprise Manager and its packs. However, in some cases Oracle Enterprise Manager or one of its packs are required – either because there is no BR*Tools support for an Oracle Database feature (example: Database Vault) or because SAP and Oracle functionality are integrated so completely that the former is not possible without the latter (example: SAP DBA Cockpit and Enterprise Manager Diagnostics Pack).

Versions: Oracle Database 11*g* and higher.

Implementation: For general information about using Oracle Enterprise Manager in SAP environments see SAP Notes 355770 and 1028068. Special use cases are discussed in additional notes and white papers.

SAP DBA Cockpit and Enterprise Manager Diagnostics Pack

The Oracle Diagnostic Pack provides automatic performance diagnostic and advanced system monitoring functionality. The Diagnostic Pack includes the following features:

- **Automatic Workload Repository (AWR):** AWR is a built-in repository within every Oracle Database that contains operational statistics about that particular database and other configuration and usage information. At regular intervals, the Oracle Database takes a snapshot of all its performance statistics and workload information and stores it in AWR. AWR forms the foundation for most of the self-management functionality of Oracle Database. It is the source of information that gives the Oracle Database a historical perspective on how it is being used and enables it to make decisions, which are accurate and specifically tailored for the system's environment. Most of the self-managing features of the Oracle Database rely heavily on the information captured in AWR. The data in AWR is also useful for diagnosing all types of performance issues while minimizing administrative overhead.
- **Automatic Database Diagnostic Monitor (ADDM):** ADDM builds upon the data captured in AWR. ADDM makes it possible for the Oracle Database to diagnose its own performance and determine how any identified problems could be resolved. ADDM runs automatically after each AWR statistics capture and makes the performance diagnostic data available immediately. ADDM examines data captured in AWR and performs analysis to determine the major issues on a proactive basis, recommends solutions and quantifies expected benefits.
- **Active Session History (ASH):** All active database sessions are automatically sampled once every second and stored in the ASH. The data is captured in a rolling buffer in database memory. The ASH data shows where the database is currently spending its time and highlights any performance bottlenecks. As ASH captures the session state with many performance attributes, the in-memory ASH data can be very effectively used to understand the database workload profile and pro-actively diagnose any transient performance issue, such as a CPU spike or an I/O storm, that occurs for a very short duration.

Oracle Diagnostics Pack diagnostics and analysis features such as ADDR, ASH, and AWR are tightly integrated with SAP's DBA Cockpit. Therefore DBA Cockpit has a mandatory requirement for Oracle Diagnostics Pack.

Database Lifecycle Management Pack

The Provisioning and Patch Automation Pack automates the deployment of software, applications, and patches. It makes critical data center operations easy, efficient and scalable, resulting in lower operational risk and cost of ownership. The ability to provision the entire software stack that includes the operating system, middleware and database, supplemented by comprehensive reporting tools, make the Provisioning and Patch Automation Pack an extremely significant entity in overall system management space.

The Provisioning and Patch Automation Pack includes an end-to-end *patching solution* that works seamlessly across a wide range of products and customer environments. The patching application automates the deployment of Oracle patches for the database and the underlying operating system. The application takes care of pre- and post-patching steps such as shutdown and startup of services, and dictionary changes, if required. It also leverages the flexible Deployment Procedure framework, that lets users add custom steps for specific actions within the patching process.

The Provisioning and Patch Automation Pack also comes with out-of-the-box Deployment Procedures to *provision* the Oracle Database (both single instance database and RAC), Oracle Clusterware and Oracle Automatic Storage Management from "gold images" following the best practices for maximum availability. The gold images are tested and approved software images and can be patched to any level before deployment.

For SAP environments, the MOPatch utility is integrated with the deployment procedures of Oracle Enterprise Manager to automate the orchestration of patching for Oracle Databases. See white paper "Patching of Oracle Databases in SAP Environments using Oracle Enterprise Manager".

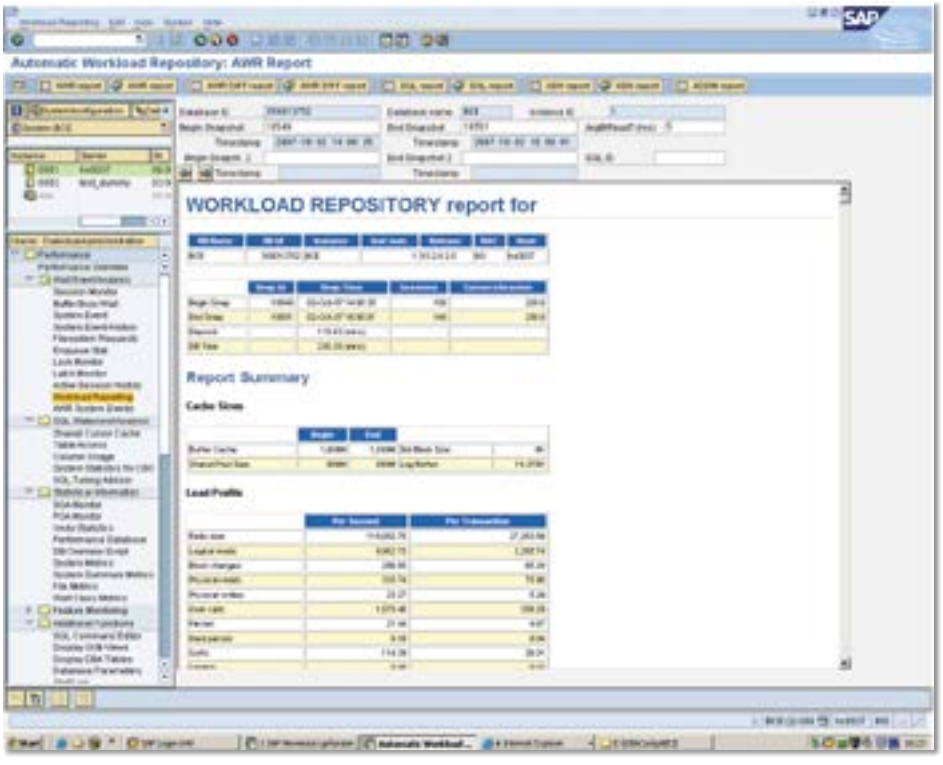

Figure 12: Oracle Enterprise Manager Diagnostics Pack and SAP DBA Cockpit

Summary

In a certain sense, the Oracle Database has always been a data management infrastructure, and therefore there has always been a need to provide support for two opposite goals: consolidation, and separation or differentiation. Oracle has always been a multi-user, multi-schema, and possibly multi-application database. Therefore – apart from the very basic split of the available data into different tables – there was a need to separate application-specific schemas. For performance reasons, there was a need to separate the files used for user/application data from those used for redo log or undo information. Being a complete data management system, the Oracle Database has nevertheless always provided adapters, gateways, and other means that allow interactive users and batch jobs to join data stored in the Oracle Database with data coming from other, external data sources.

However, as a result of the growing amount of data, increasing workload, and shrinking maintenance windows that many customers experience, Oracle needed to provide additional, more sophisticated solutions. Starting with Oracle Database 11*g*, this has happened in all recent Database versions (12c, 18c, 19c).

Defining Subsets of Data

Released for the first time several versions and many years ago, *Table and Index Partitioning* is still a good example of data differentiation and a powerful technology. Therefore it has been discussed at the very beginning of this article. It is used to define subsets of table data which can be accessed and managed independently.

Subsets and Access Patterns

When determining a specific partitioning strategy, it is important to not only look at the data but to consider the access patterns as well. Partitioning a table by month may turn out to be useless or even negatively impact performance, if most queries access the data by location.

However, distinguishing subsets of data by access pattern in new, previously impossible ways is the main goal of many Oracle Database features and options.

Heat Map, a feature included in Advanced Compression, automatically tracks modification and query timestamps, thus providing detailed insights into how data is being accessed. *Automatic Data Optimization* automatically moves and-compresses data according to user-defined policies based on the information collected by Heat Map. This means that storage and compression tiering can be implemented, i.e. the selection of different storage media and different compression algorithms for different subsets of data which are defined by different access characteristics.

Oracle Database In-Memory allows administrators to distinguish between different types of transactions as well as different data representations in memory and to avoid the trade-off which results from the requirement to pick one single data format for all types of transactions and applications. Based on the In-Memory option, Oracle Database can provide data used by OLTP transactions in row format, and at the same time data used by analytics in column format.

Workload Distribution

Other options focus on workload distribution. A prominent example is *Real Application Clusters (RAC)*, which allows customers to split the system workload and let many servers, running at least as many Oracle instances, handle the workload. It is up to the customer to decide whether all instances should handle the same type (or mixture) of workload(s) or different instances should be responsible for different types of workload (e.g. interactive transactions vs. batch jobs).

In addition to enabling workload distribution, RAC increases the system availability. This is even more true for *Data Guard*. Nevertheless, Data Guard does not require the standby database to be idle. It can be used for operations such as backup or – under certain conditions – reporting. In other words: Data Guard helps implement workload distribution as well.

Data Access Policies

In a somewhat different sense, Oracle's security options are related to access patterns as well. In particular *Database Vault* lets security administrators define and enforce access policies which prevent privileged users from reading or manipulating data they are not supposed to access. Phrases such as "segregation of duties" make it very clear that even here we are talking about the balance between separation and combination.

Database Administration

Oracle Database Multitenant moves the data management infrastructure principle to a completely new level: We do not simply talk about an infrastructure for many tables or many users. We talk about an infrastructure for many databases. Here consolidation clearly requires separation.

To put it simply, Multitenant is about the distinction between those administration tasks that should be executed for every single database individually (because there are differences) and those that can be executed once for all or at least for a certain group of databases (because there are no differences). E.g. to apply a certain patch to 25 database systems means to repeat the very same process 25 times. Therefore Oracle Multitenant moves this operation to the Container Database layer, where it can be done once for all Pluggable Databases.

On the other hand, a point-in-time recovery is an operation to be performed on the PDB level, as it should not affect other PDBs.

In addition, the Provisioning and *Patch Automation Pack* allows administrators to automate those operations which are similar in all cases. Examples are the deployment of Oracle patches or the provisioning of new systems based on "gold images".

Monitoring and Testing

Will your systems benefit from these features? How can you be sure? And how should the features be implemented in order to have a positive impact on your systems? *Real Application Testing* will help you to answer these and similar questions.