How Autonomous is the Oracle Autonomous Data Warehouse?

Christian Antognini / Dani Schnider

@chrisantognini  antognini.ch/blog
@dani_schnider  danischnider.wordpress.com
Agenda

1. Introduction
2. Set Up
3. Connectivity
4. Loading Data
5. ETL Performance
6. Query Performance
7. Monitoring Performance
8. Miscellaneous
9. Conclusion
Introduction
“There is nothing to learn, there is nothing to do”
Automated Database Administration

«...you do not need to configure or manage any **hardware**, or install any **software**. Autonomous Data Warehouse handles creating the data warehouse, **backing up** the database, **patching and upgrading** the database, and **growing or shrinking** the database.»

Automated Performance Tuning

«When you use Autonomous Data Warehouse, no tuning is necessary. You do not need to consider any details about **parallelism, partitioning, indexing, or compression**. The service automatically configures the database for high-performance queries.»

Source: Using Oracle Autonomous Data Warehouse, Chapter 1
Set Up
Creating

- Name
- Number of CPU (1..128)
- Storage (1..128 TB)
- PDB admin password
- License type
- Tags (optional)
At any moment, without downtime, it’s possible to scale up/down the number of CPU and the available storage.
### Initialization Parameters That Can Be Modified

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPROX_FOR_AGGREGATION</td>
<td>PLSCOPE_SETTINGS</td>
</tr>
<tr>
<td>APPROX_FOR_COUNT_DISTINCT</td>
<td>PLSQL_CCFLAGS</td>
</tr>
<tr>
<td>APPROX_FOR_PERCENTILE</td>
<td>PLSQL_DEBUG</td>
</tr>
<tr>
<td>AWR_PDB_AUTOFLUSH_ENABLED¹</td>
<td>PLSQL_OPTIMIZE_LEVEL</td>
</tr>
<tr>
<td>OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES²</td>
<td>PLSQL_WARNINGS</td>
</tr>
<tr>
<td>OPTIMIZER_IGNORE_HINTS</td>
<td>TIME_ZONE²</td>
</tr>
<tr>
<td>OPTIMIZER_IGNORE_PARALLEL_HINTS</td>
<td>NLS_*</td>
</tr>
</tbody>
</table>

¹System level only  ²Session level only
A plan (DWCS_PLAN) with three consumer groups is pre-configured

<table>
<thead>
<tr>
<th>Consumer Group</th>
<th>CPU</th>
<th>Session Pool</th>
<th>PX Server Limit</th>
<th>DOP Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH</td>
<td>4</td>
<td>Unlimited</td>
<td>50</td>
<td>CPU_COUNT^1</td>
</tr>
<tr>
<td>MEDIUM</td>
<td>2</td>
<td>Unlimited</td>
<td>84</td>
<td>4</td>
</tr>
<tr>
<td>LOW</td>
<td>1</td>
<td>2*CPU_COUNT^1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>OTHER_GROUPS</td>
<td>1</td>
<td>Unlimited</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

^1When several instances are used, CPU_COUNT <> “CPU Core Count”
Except for OTHER_GROUPS, two thresholds to configure when queries are cancelled can be set:

- Run time in seconds
- Amount of disk I/O in MB

Remark:
- 0 = detection disabled
Connectivity
## How to Connect to Autonomous Data Warehouse Cloud

<table>
<thead>
<tr>
<th>Tool</th>
<th>Purpose</th>
<th>Connection Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Console</td>
<td>service management</td>
<td>browser</td>
</tr>
<tr>
<td>Apache Zeppelin</td>
<td>Oracle Machine Learning</td>
<td></td>
</tr>
<tr>
<td>SQL Developer</td>
<td>development, ad-hoc queries</td>
<td>JDBC (automatic configuration)</td>
</tr>
<tr>
<td>SQL*Plus, SQLcl, Toad, ...</td>
<td>development, ad-hoc queries</td>
<td>OCI, JDBC, ODBC (manual configuration)</td>
</tr>
<tr>
<td>ETL Tools (ODI, 3rd party)</td>
<td>data integration services</td>
<td></td>
</tr>
<tr>
<td>BI Tools (OBIEE, 3rd party)</td>
<td>BI services</td>
<td></td>
</tr>
</tbody>
</table>
Service Management via Browser

- Used for:
  - Service Console
  - Start/Stop DB
  - Scale up/down
  - Restore
  - Management of credential
Oracle Machine Learning

Notebook style application for advanced SQL users

- Apache Zeppelin
- Interactive data analysis
- Graphical reports

Typical Users

- Data scientists
- Developers
- Business users
Connection with SQL Developer

Download client credential file
- Via Service Console

Create new connection
- Type “Cloud PDB”
- Import credential file

Enter keystore password (only for versions < 18.2)

Select service level (LOW, MEDIUM, HIGH)
Manual Connection Configuration

Connections using Oracle Net Services (SQL*Net)

- For SQL*Plus, SQLcl, Toad, ...
- ETL and BI tools, 3rd party tools

Oracle Call Interface (OCI)
JDBC OCI
JDBC Thin Client
ODBC

Encrypted SSL connection
TCP/IP
Loading Data
Loading Data

Object storage service
- Oracle Object Storage
- Amazon Simple Storage Service (S3)
- Microsoft Azure Blob Storage

Cloud

Client computer
- Oracle SQL*Loader
- 3rd party application
- Command-line interface

HTTPS / REST

Oracle SQL*Loader

SQL*Net

HTTPS / REST

3rd party application

Microsoft Azure Blob Storage

Cloud

Oracle Object Storage

Amazon Simple Storage Service (S3)
ADW supports the following object storage services

- Oracle’s Object Storage
- Amazon Simple Storage Service (S3)
- Microsoft’s Azure Blob Storage

Those services store data as objects within buckets (containers)

Objects are identified with an URI

The access to the objects stored within buckets is protected

- Credentials managed through DBMS_CLOUD are used
Command Line Interface (CLI)

Oracle provides a CLI to work with Oracle Cloud Infrastructure (OCI) objects and services.

The CLI is built on Python and runs on Windows, Linux and Mac.

The CLI makes calls to the OCI API.

Among other things, the CLI can be used to manage buckets and objects stored in the supported OSS.
It provides features to

- manage credentials to access an OSS
- manage external tables that can be used to query data stored in an OSS
- handle objects stored in an OSS
- handle files stored on ADW in DATA_PUMP_DIR

Data Pump 12.2 supports importing (but not exporting) a file stored in an OSS

```
ALTER DATABASE PROPERTY SET default_credential = 'ADMIN.CHRIS'
```

```
impdp dumpfile=default_credentials:https://.../test.dmp
```
Example – Loading Data into a 1TB TPC-DS Schema (1)

Step 1: Upload the text files from the file system of a server hosted in the cloud (same region) to the Oracle’s object storage

```
oci os object bulk-upload --bucket-name tpcds
  --src-dir /data/tpcds
  --include "*.dat"
```

- By default, such a statement uses 10 parallel threads
- The load took 6 hours

Step 2: Load the data into the database through DBMS_CLOUD
- The number of allocated CPU cores determines how fast data is loaded
- With 32 cores it took 67 minutes
Example – Loading Data into a 1TB TPC-DS Schema (2)

Example of scalability: load 22GB into the CATALOG_RETURNS table

ETL Performance
Prepared for High ETL Performance

Good ETL performance is possible
- Setup is optimized for DWH and ETL
- But details must be considered

- Parallel DML Execution
- Direct-Path Load Operations
- Online Statistics Gathering
- No Indexes
- Hybrid Columnar Compression
Parallel DML Execution and Direct-Path Operations

**Parallel DML** (PDML) is enabled by default
- Only if CPU core count > 1 and consumer group is *MEDIUM* or *HIGH*
- Hint `/*+ parallel */` can be added (usually not needed)
  - Set `optimizer_ignore_parallel_hints = FALSE` (default: *TRUE*)

**Direct-Path INSERT** is used
- For parallel DML
- If hint `/*+ append */` is added
Parallel DML Execution and Consumer Groups

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>TQ</th>
<th>IN-OUT</th>
<th>PQ Distrib</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>INSERT STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>LOAD AS SELECT</td>
<td>CUSTOMERS_1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PX COORDINATOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PX SEND QC (RANDOM)</td>
<td>:TQ10000</td>
<td>Q1,00</td>
<td>P-&gt;S</td>
<td>QC (RAND)</td>
</tr>
<tr>
<td>4</td>
<td>OPTIMIZER STATISTICS GATHERING</td>
<td></td>
<td>Q1,00</td>
<td>PCWC</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>PX BLOCK ITERATOR</td>
<td></td>
<td>Q1,00</td>
<td>PCWC</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>TABLE ACCESS STORAGE FULL</td>
<td>CUSTOMERS</td>
<td>Q1,00</td>
<td>PCWP</td>
<td></td>
</tr>
</tbody>
</table>

- automatic DOP: Computed Degree of Parallelism is 8

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>INSERT STATEMENT</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>LOAD TABLE CONVENTIONAL</td>
<td>CUSTOMERS_1</td>
</tr>
<tr>
<td>2</td>
<td>TABLE ACCESS STORAGE FULL</td>
<td>CUSTOMERS</td>
</tr>
</tbody>
</table>

- automatic DOP: Computed Degree of Parallelism is 1 because of no expensive parallel operation
- PDML disabled because object is not decorated with parallel clause
- Direct Load disabled because no append hint given and not executing in parallel
Parallel DML / Direct-Path and Constraints

Restrictions must be considered:

- If FK constraints are defined, PDML / direct-path is disabled
- Conventional load is used

Recommendation:
- Define reliable constraints

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>INSERT STATEMENT</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>LOAD TABLE CONVENTIONAL</td>
<td>CUSTOMERS_1</td>
</tr>
<tr>
<td>2</td>
<td>PX COORDINATOR</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PX SEND QC (RANDOM)</td>
<td>:TQ10000</td>
</tr>
<tr>
<td>4</td>
<td>PX BLOCK ITERATOR</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>TABLE ACCESS STORAGE FULL</td>
<td>CUSTOMERS</td>
</tr>
</tbody>
</table>

Note

- automatic DOP: Computed Degree of Parallelism is 8
- PDML disabled because parent referential constraints are present

```
ALTER TABLE customers_1
ADD FOREIGN KEY (country_id) REFERENCES countries
RELY DISABLE NOVALIDATE
```
Online Statistics Gathering for Direct-Path Loads

Statistics are gathered automatically

Unlike 12c, this works also

■ for non-empty tables
■ for histograms

Two new undocumented parameters

■ _optimizer_gather_stats_on_load_all (default: TRUE)
■ _optimizer_gather_stats_on_load_hist (default: TRUE)
Statistics Gathering for Conventional Loads

Attention: statistics are not gathered automatically

Call DBMS_STATS with default values

Automatic Statistics Gathering job is enabled, but maintenance windows are disabled
No Indexes

Indexes are not allowed in ADW

```
CREATE INDEX s_order_item_delivery_date_idx
ON s_order_item (delivery_date);
ORA-01031: insufficient privileges
```

- Good! Avoids many ETL performance issues
- Indexes are not needed for (most) analytical queries

Exception: indexes are created for primary key and unique constraints
Index Maintenance (PK and Unique Constraints)

Index maintenance after direct-path load

- Very time-consuming, cannot be parallelized
- Workaround: create PK with RELY DISABLE NOVALIDATE
Hybrid Columnar Compression

All tables are automatically HCC compressed, with row-level locking enabled

```
SELECT table_name, compression, compress_for
FROM user_tables ORDER BY table_name

<table>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>COMPRESS</th>
<th>COMPRESS_FOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>H_BEER</td>
<td>ENABLED</td>
<td>QUERY HIGH ROW LEVEL LOCKING</td>
</tr>
<tr>
<td>H_BOTTLING</td>
<td>ENABLED</td>
<td>QUERY HIGH ROW LEVEL LOCKING</td>
</tr>
<tr>
<td>H_BREW</td>
<td>ENABLED</td>
<td>QUERY HIGH ROW LEVEL LOCKING</td>
</tr>
<tr>
<td>H_CUSTOMER</td>
<td>ENABLED</td>
<td>QUERY HIGH ROW LEVEL LOCKING</td>
</tr>
<tr>
<td>H_MALT</td>
<td>ENABLED</td>
<td>QUERY HIGH ROW LEVEL LOCKING</td>
</tr>
<tr>
<td>H_ORDER</td>
<td>ENABLED</td>
<td>QUERY HIGH ROW LEVEL LOCKING</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Example: Loading Data Vault Schema

Data Vault Model of Craft Beer Brewery

- 25 target tables (Hubs, Links, Satellites)
- 20 beers, 272K customers
- 16M orders, 173M order items
- “Daily loads” for 8 months
  - Random data generator
  - Data Vault load patterns
  - Delta detection and versioning
  - 1.4M order items / day
Example: Loading Data Vault Schema
Query Performance
Result Cache

It is enabled by default on ADW

- RESULT_CACHE_MODE = FORCE

For queries being re-executed, it can lead to a tremendous performance improvement

To avoid caching, use the NO_RESULT_CACHE hint

- Even if OPTIMIZER_IGNORE_HINTS = TRUE (default on ADW)
Improve Query Performance

What cannot be done
- Partition tables
- Create materialized views
- Use In-Memory Column Store
- Create indexes (except for PK/UK)

What can be done
- Scale up the number of CPU cores
- Use different service
- Use constraints to enable query transformations (e.g. join elimination)
Since no index is automatically created, the star transformation is not available instead, the query optimizer can use the vector transformation.

- Introduced in 12.1.0.2 for In-Memory Aggregation
- In many situations, faster than star transformation

Note
- vector transformation used for this statement

- Works even if tables are not populated in IMCS
- In-Memory is enabled (but not used) in ADW
  - INMEMORY_SIZE = 1073741824
Example – TPC-DS Queries

Query 38 – Execution Time in Seconds

- 2 CPU Cores: HIGH 174, MEDIUM 175, LOW 305
- 4 CPU Cores: HIGH 90, MEDIUM 91, LOW 305
- 8 CPU Cores: HIGH 55, MEDIUM 90, LOW 311
- 16 CPU Cores: HIGH 31, MEDIUM 90, LOW 303

Service
- HIGH
- MEDIUM
- LOW

21.09.2018
How Autonomous is the Oracle Autonomous Data Warehouse?
Example – Sample Star Schema Benchmark (SSB)

<table>
<thead>
<tr>
<th>SSB.DWDATE</th>
<th>2556</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_DATEKEY</td>
<td>DATE</td>
</tr>
<tr>
<td>D_DATE</td>
<td>CHAR (18 BYTE)</td>
</tr>
<tr>
<td>D_DAYOFWEEK</td>
<td>CHAR (9 BYTE)</td>
</tr>
<tr>
<td>D_MONTH</td>
<td>CHAR (9 BYTE)</td>
</tr>
<tr>
<td>D_YEAR</td>
<td>NUMBER</td>
</tr>
<tr>
<td>D_YEARMONTH</td>
<td>NUMBER</td>
</tr>
<tr>
<td>D_YEARMONTH</td>
<td>CHAR (7 BYTE)</td>
</tr>
<tr>
<td>D_DAYNUMINWEEK</td>
<td>NUMBER</td>
</tr>
<tr>
<td>D_DAYNUMINMONTH</td>
<td>NUMBER</td>
</tr>
<tr>
<td>D_DAYNUMINYEAR</td>
<td>NUMBER</td>
</tr>
<tr>
<td>D_MONTHNUMINYEAR</td>
<td>NUMBER</td>
</tr>
<tr>
<td>D_WEEKNUMINYEAR</td>
<td>NUMBER</td>
</tr>
<tr>
<td>D_SELLINGSEASON</td>
<td>CHAR (12 BYTE)</td>
</tr>
<tr>
<td>D_LASTDAYINWEEKFL</td>
<td>CHAR (1 BYTE)</td>
</tr>
<tr>
<td>D_LASTDAYINMONTHFL</td>
<td>CHAR (1 BYTE)</td>
</tr>
<tr>
<td>D_HOLIDAYFL</td>
<td>CHAR (1 BYTE)</td>
</tr>
<tr>
<td>D_WEEKDAYFL</td>
<td>CHAR (1 BYTE)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSB.LINEORDER</th>
<th>6 billions</th>
</tr>
</thead>
<tbody>
<tr>
<td>LO_ORDERKEY</td>
<td>NUMBER</td>
</tr>
<tr>
<td>LO_ORDERDATE</td>
<td>DATE</td>
</tr>
<tr>
<td>LO_ORDERPRIORITY</td>
<td>CHAR (15 BYTE)</td>
</tr>
<tr>
<td>LO_SHIPMODE</td>
<td>CHAR (10 BYTE)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSB.PART</th>
<th>2 millions</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_PARTKEY</td>
<td>NUMBER</td>
</tr>
<tr>
<td>P_NAME</td>
<td>VARCHAR2 (22 BYTE)</td>
</tr>
<tr>
<td>P.SHIPCR</td>
<td>CHAR (9 BYTE)</td>
</tr>
<tr>
<td>P.CATEGORY</td>
<td>CHAR (7 BYTE)</td>
</tr>
<tr>
<td>P.ERAND1</td>
<td>CHAR (9 BYTE)</td>
</tr>
<tr>
<td>P_COLOR</td>
<td>VARCHAR2 (11 BYTE)</td>
</tr>
<tr>
<td>P.TYPE</td>
<td>VARCHAR2 (25 BYTE)</td>
</tr>
<tr>
<td>P_SIZE</td>
<td>NUMBER</td>
</tr>
<tr>
<td>P_CONTAINER</td>
<td>CHAR (10 BYTE)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSB.SUPPLIER</th>
<th>30 millions</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_SUPKEY</td>
<td>NUMBER</td>
</tr>
<tr>
<td>S_NAME</td>
<td>CHAR (25 BYTE)</td>
</tr>
<tr>
<td>S_ADDRESS</td>
<td>VARCHAR2 (25 BYTE)</td>
</tr>
<tr>
<td>S_PHONE</td>
<td>CHAR (15 BYTE)</td>
</tr>
<tr>
<td>S_PHONE</td>
<td>CHAR (15 BYTE)</td>
</tr>
</tbody>
</table>

21.09.2018 How Autonomous is the Oracle Autonomous Data Warehouse?
Example – Sample Star Schema Benchmark (SSB)

Query 1 (1 Dimension, 1 Fact Table, 1 Aggregated Row)

- 1st Exec
  - 4 CPU Cores: 189
  - 8 CPU Cores: 42
  - 16 CPU Cores: 15

- 2nd Exec
Example – Sample Star Schema Benchmark (SSB)

Query 14 (4 Dimensions, 1 Fact Table, 800 Rows)

<table>
<thead>
<tr>
<th></th>
<th>1st Execution</th>
<th>2nd Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 CPU Cores</td>
<td>413</td>
<td>362</td>
</tr>
<tr>
<td>8 CPU Cores</td>
<td>132</td>
<td>120</td>
</tr>
<tr>
<td>16 CPU Cores</td>
<td>42</td>
<td>44</td>
</tr>
</tbody>
</table>

413 362

4 CPU Cores
8 CPU Cores
16 CPU Cores
Automated Tuning?

Industry-Leading Performance

- Integrated machine-learning algorithms drive automatic caching, adaptive indexing, advanced compression, and optimized cloud data-loading to deliver unrivaled performance.
- Automatic adaptive performance tuning delivers faster analytics.

source: https://www.oracle.com/database/data-warehouse.html

Exadata storage indexes?

Result cache? (in addition to disk I/O caching)

Not observed 😞
Monitoring Performance
Monitoring Performance and SQL Statements

Service Console with monitoring capabilities:

- Activity and utilization
  - Storage and CPU usage
  - Real-time or time period
- Running SQL statements
  - Time & wait statistics, I/O statistics
  - Runtime execution plan
  - Parallel processes
  - Downloadable real-time SQL Monitoring report

Default 8 days, can be changed with AWR settings.
Monitoring Performance with AWR and ASH Reports

AWR reports can be generated (only on PDB level)
- With DBMS_WORKLOAD_REPOSITORY
- With SQL Developer reports

```sql
SELECT output FROM TABLE(
    DBMS_WORKLOAD_REPOSITORY.AWR_REPORT_HTML
    (3951758934, 7, 3524, 3535)
)
```

ASH reports can be generated based on
- V$ACTIVE_SESSION_HISTORY / DBA_HIST_ACTIVE_SESS_HISTORY
- SQL Developer reports
Miscellaneous
Backup & Recovery

Automatic (incremental) backups take place daily
- The start time cannot be set
- The retention period for automatic backups is 60 days

Manual backups can be initiated through the console
- They are stored in the object store
- The DEFAULT_CREDENTIAL and DEFAULT_BUCKETS database properties must be set

Recovery at any point-in-time can be initiated through the console
Patching

Oracle is patching the service on a regular basis

- No announcement
- No downtime

There is no way to “schedule” when the installation of the patches takes place
Oracle Support

As soon as a problem cannot be solve because of missing privileges, an SR has to be opened.

Support has (almost) no visibility:
- E.g. no access to the alert.log of the PDB
- E.g. no OS access

Support relies on Operations to fix things.

The issues we experienced were fixed in a time frame going from one days to one week.
Conclusion
Conclusion

- Appropriate setup for DWH
- Better than many manual configured DWHs
- Easy (limited) administration, ready to use
- No partitioning, no In-Memory (yet)
- Support in case of a service request can take too long
- Sometimes “shaky” (no control about patches / changes)
- Not as simple as it looks at first sight
- Knowhow about physical DB design still important
Further Information in Blog Posts

- DBMS_CLOUD Package – A Reference Guide

- Which Privileges Are Required to Use the ADWC Service Console?
  https://antognini.ch/2018/07/which-privileges-are-required-to-use-the-adwc-service-console/

- Observations About the Scalability of Data Loads in ADWC

- External Tables in Autonomous Data Warehouse Cloud

- Gathering Statistics in the Autonomous Data Warehouse Cloud

- 10 Tips to Improve ETL Performance – Revised for ADWC

- Star Schema Optimization in Autonomous Data Warehouse Cloud
How Autonomous is the Oracle Autonomous Data Warehouse?