Oracle Autonomous Technologies

**Self-Driving** – Automates all database and infrastructure management, monitoring, tuning

**Self-Securing** – Protects from attacks

**Self-Repairing** – Minimizes or eliminates downtimes including planned maintenance

Reduce Costs
Risks
Focus on Innovation
Oracle Autonomous Technologies and ML

**Infrastructure Management**
Detection and recovery of failed/sick server, storage or switch/link

**Operations**
Hang Management, Anomaly Detection, Maintenance Slot Identification, Bug Identification and Prioritization

**Workload Optimizations**
Query Optimizer, Real-time statistics, Automatic Indexing
Oracle Autonomous Technologies

**Autonomous Data Warehouse (ADW)**
- Columnar Data Summaries
- Memory speeds Joins, Aggs, etc
- Real-time Statistics

**Autonomous Transaction Processing (ATP)**
- Row format Indexes
- Memory for Caching to avoid IO
- Real-time Statistics
ADW – CERN Industrial IoT (SCADA)

PSEN schema - the first step
- 750GB of SCADA data about Electrical Network
- Contains IOT partitioned tables
- One big IOT partitioned table of 620G

Improve performance of most challenging data retrieval and analytics scenario
- Trends
- Events
- Alarms
- Historical event
Create the instance

Create Autonomous Data Warehouse

Select the Cloud Service you want to start:

Featured Services  All Services

- Autonomous Data Warehouse
  Subscription ID

Administrator Credentials

Set the password for your Autonomous Data Warehouse ADRM user here.

Username: admin
Password:
Confirm Password:

License Type:
- ORGANIZATION LICENSES ORACLE BAREMETAL SOFTWARE LICENSES
- ADDITIONAL ORACLE SOFTWARE LICENSES AND LICENSE AGREEMENTS

Tags

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values that can be attached to resources.

Learn more about tagging

Create Autonomous Data Warehouse

Create Autonomous Data Warehouse
Create the instance (CLI)

Oracle Cloud Infrastructure

Autonomous Data Warehouses

Create Autonomous Data Warehouse

Displaying 2 Autonomous Data Warehouses
That is required to upload data pump export files
ADW – CERN Industrial IoT (SCADA)

Moving Data to ADWC

ADW principles: data is imported in its simplest form:

- No indexes
- No partitions
- No IOTs
- No materialized views

... 

Data pump allows the needed transformations
### ADW – CERN Industrial IoT (SCADA)

#### Moving Data to ADWC

**Export Data**

- `exclude=index, cluster, indextype, materialized_view, materialized_view_log, materialized_zonemap, db_link`
- `data_options=group_partition_table_data`
- `parallel=n`
- `schemas=schema_name`
- `dumpfile=export%u.dmp`

**Upload data to Object Storage**

- `oci os object bulk-upload --bucket-name PSEN BUCKET 1 -- src-dir /mnt/oci/ - part-size 64 --parallel-upload-count 10`

**Import Data into ADW**

- Parallel set to the number of CPUs you have
- Partitioned tables are converted to non-partitioned
- All segment attributes are ignored
- IOTs are converted to regular tables
- PK and unique indexes renamed to constraint name
- Same exclusion as during export
ADW – CERN Industrial IoT (SCADA)

Moving Data to ADWC

Import Data

Parallelism

<table>
<thead>
<tr>
<th>SID</th>
<th>EVENT</th>
<th>MODULE</th>
<th>SQL_TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>4391</td>
<td>wait for unread message on broa...</td>
<td>sys.kupcsqwe_int.get_status(2, 3); END;</td>
<td></td>
</tr>
<tr>
<td>6986</td>
<td>wait for unread message on broa...</td>
<td>sys.kupcsqwe_int.receivet(2); END;</td>
<td></td>
</tr>
<tr>
<td>7344</td>
<td>TM - contention</td>
<td>Data Pump Worker</td>
<td>INSERT /* APPEND ENABLE_PARALLEL_DML PARALLEL(&quot;EVENTHISTORY_000000008&quot;,1)*/</td>
</tr>
<tr>
<td>7679</td>
<td>TM - contention</td>
<td>Data Pump Worker</td>
<td>INSERT /* APPEND ENABLE_PARALLEL_DML PARALLEL(&quot;EVENTHISTORY_000000008&quot;,1)*/</td>
</tr>
<tr>
<td>8419</td>
<td>Data pump dump file I/O</td>
<td>Data Pump Worker</td>
<td>INSERT /* APPEND ENABLE_PARALLEL_DML PARALLEL(&quot;EVENTHISTORY_000000008&quot;,1)*/</td>
</tr>
<tr>
<td>8596</td>
<td>TM - contention</td>
<td>Data Pump Worker</td>
<td>INSERT /* APPEND ENABLE_PARALLEL_DML PARALLEL(&quot;EVENTHISTORY_000000008&quot;,1)*/</td>
</tr>
</tbody>
</table>

Importing Issues

- imported "PSIEN, "EVENTHISTORY_000000008, "EVH_000000008, 2017041900" 6021 MB 8986484 rows
- imported "PSIEN, "EVENTHISTORY_000000008, "EVH_000000008, 2015082700" 578 MB 91131163 rows
- imported "PSIEN, "EVENTHISTORY_000000008, "EVH_000000008, 2015071300" 574 MB 9125033 rows
- imported "PSIEN, "EVENTHISTORY_000000008, "EVH_00000008, 2016052300" 593 MB 9428793 rows

ORA-00924: One or more workers have prematurely exited
ORA-08007: too many papers: "DW00" prematurely terminated
ORA-01871: Worker process DW00 had an unhandled exception.
ORA-00600: Internal error code arguments [40302, [40015D23E2, 5, 1, 1, 1, 1], 1]
ORA-08512: at "SYS.DBMS_STATS", line 1778
ORA-08512: at "SYS.DBMS_STATS", line 16778
ORA-08512: at "SYS.DBMS_STATS", line 32268
ORA-08512: at wal 1
ORA-08512: at "SYS.DBMS_SQL", line 1721
ORA-08512: at "SYS.KUPDBDATA", line 1149
ORA-08512: at "SYS.KUPDBDATA", line 12392
ORA-08512: at "SYS.KUPDBDATA", line 5434
ORA-08512: at "SYS.KUPDBDATA", line 4719
ORA-08512: at "SYS.KUPDBDATA", line 6417
ORA-08512: at "SYS.KUPDBSPACE", line 22345
ORA-08512: at "SYS.KUPDBSPACE", line 6028
ORA-08512: at "SYS.KUPDBSPACE", line 33365
ORA-08512: at "SYS.KUPDBSPACE", line 2397
ORA-08512: at line 2
ADW – CERN Industrial IoT (SCADA)

Moving Data to ADWC

- Import Data
- Where are my logs?
  - Alert logs
  - Trace files
  - Data pump logs

SQL Code:

```
SELECT PAYLOAD FROM VDIAG_TRACE_FILE_CONTENTS WHERE
   TRACE_FILENAME = 'ehlpad08_ora_115533.trc' ORDER BY line_number FETCH FIRST 15 ROWS ONLY;
```

```
DECLARE
   l_file UTL_FILE.file_type;
   l_location VARCHAR2(100) := 'DATA_PUMP_DIR';
   l_filename VARCHAR2(100) := 'import_PSEN_03002018.log';
   l_text VARCHAR2(32767);
BEGIN
   -- Open file.
   l_file := UTL_FILE.fopen(l_location, l_filename, 'r', 32767);
   BEGIN
      LOOP
         UTL_FILE.get_line(l_file, l_text, 32767);
         DBMS_OUTPUT.put_line(l_text);
      END LOOP;
   EXCEPTION
      WHEN NO_DATA_FOUND THEN
         NULL;
   END;
   -- Close the file.
   UTL_FILE.fclose(l_file);
END;
```
**ADW Insights**

**Comparison ADW and on-premise:**
- Different Schemas
  - IoT Tables – Compressed Tables + PK Index
  - Executions plans changed a lot
- Very specific use case
  - But that is real-life

**Missing Statistics (after data pump import):**
- We gather them again
- Due to no use recommended parameters

**High compression:**
- Hybrid Columnar Compression (HCC) reduces tables size by a factor of 10
- Full scans are smaller!
**ADW Insights**

**Indexes:**
- Default scenario is not use indexes on ADW
- HCC reduces full scans workloads
- Exadata Smart Scan
- Storage Indexes

**Seamless provisioning and fully elastic**
- CPU and storage can be adjusted online, at anytime in few seconds!
- So you can start with few resources and grow only if needed

**Optimization:**
- Automatic and transparent access to Oracle optimization features
Autonomous Next Steps

Deeper analysis on performance and scale data volumes
- Indexes
- HCC
- Exadata Smart Scan
- Storage Indexes

Oracle Autonomous Transaction Processing

Oracle Autonomous Analytics Cloud Services:
The goal of the project is to deploy Physics Data processing and Machine learning on cloud resources, notably using CERN cloud and Oracle Cloud Infrastructure (OCI).

First Use cases

- Data reduction from CMS Big Data project performing event selection and dimuon invariant mass calculations on 1 PB of data read from EOS.

Scale rapidly from prototype

Share a cluster regardless of where it is deployed and of the chosen software stack

Fully featured environment, allows users to focus on analysis rather than data engineering and work collaboratively
QUESTIONS?

Manuel.Martin.Marquez@cern.ch