Mastering Oracle Data Pump

Dean Gagne (Oracle)

Udi Karni (Kaiser Permanente)
Program Agenda

- Deeper understanding of how Data Pump works
- What is in Data Pump’s Master Table
- How to get useful information out of the Master Table
- Hear a large health maintenance organization's performance experience with Data Pump in moving large quantities of data
- get real-life performance measurements and tips on how to make Data Pump run faster.
What is Oracle Data Pump?

- New feature starting in Oracle Database 10g Release 1
- Enables very fast bulk data and metadata movement between Oracle databases
- High-speed, parallel Export and Import utilities (expdp and impdp) as well as a Web-based Oracle Enterprise Manager interface
How Data Pump works

• **Estimate phase** – what is it used for
  – Get Table Data Objects

• **Phases of Data Pump Export**
  – Unload meta data (single worker process unloads all metadata)
  – Unload data (parallel data unload)(multiple workers or multiple pq slaves)

• **Phases of Data Pump Import**
  – Load meta data (serially)
    • Build indexes (in parallel using pq slaves)
    • Load package bodies (in parallel using multiple workers)
  – Load data (in parallel using multiple workers or multiple pq slaves)
What was my expdp command

> impdp system/manager directory=dpump_dir
dumpfile=mydmp.dmp master_only=y

SQL> Select name, value_t from SYSTEM.EXPORT_JOB_1 where process_order = -59 and name = ‘CLIENT_COMMAND’;

-------------------------------------

CLIENT_COMMAND

system/****** tables=scott.emp directory=dpump_dir
dumpfile=ss.dmp reuse_dumpfiles=y
• Indicates whether to import just the master table and then stop the job so that the contents of the master table can be examined.

• MASTER_ONLY=[YES | NO]
How many objects have been exported

> expdp system/manager full=y directory=dpump_dir
dumpfile=full.dmp metrics=yes

Processing object type DATABASE_EXPORT/SCHEMA/ROLE_GRANT

Completed 89 ROLE_GRANT objects in 1 seconds

Processing object type DATABASE_EXPORT/SCHEMA/default_role

Completed 2 DEFAULT_ROLE objects in 0 seconds

Processing object type DATABASE_EXPORT/RESOURCE_COST

Completed 1 RESOURCE_COST objects in 0 seconds

Processing object type DATABASE_EXPORT/SCHEMA/DB_LINK

Completed 15 DB_LINK objects in 0 seconds
METRICS

• Indicates whether additional information about the job should be reported to the Data Pump log file.

• METRICS=[YES | NO]

• When METRICS=YES is used, the number of objects and the elapsed time are recorded in the Data Pump log file.
What’s in my impdp job

> impdp system/manager DIRECTORY=dpump_dir1
DUMPFILE=expdat.dmp SCHEMAS=hr ABORT_STEP=-1

SQL> select object_type, object_schema, object_name from
SYSTEM.IMP_SCHEMA where process_order > 0 and duplicate = 0 and processing_status='C' and processing_state = 'R';

<table>
<thead>
<tr>
<th>OBJECT_TYPE</th>
<th>OBJ</th>
<th>OBJECT_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCEDURE</td>
<td>HR</td>
<td>ADD_JOB_HISTORY</td>
</tr>
<tr>
<td>ALTER_PROCEDURE</td>
<td>HR</td>
<td>ADD_JOB_HISTORY</td>
</tr>
<tr>
<td>INDEX</td>
<td>HR</td>
<td>REG_ID_PK</td>
</tr>
<tr>
<td>INDEX</td>
<td>HR</td>
<td>LOC_ID_PK</td>
</tr>
<tr>
<td>INDEX</td>
<td>HR</td>
<td>DEPT_ID_PK</td>
</tr>
</tbody>
</table>
ABORT_STEP

- **ABORT_STEP**\=[n | -1] Values correspond to a process order number in the master table.

- **n** - If the value is greater than 0, then the job is started and the job is aborted at the object that is stored in the master table with the corresponding process order number.

- **-1** - If the value is negative one (-1) then abort the job after setting it up, but before exporting or importing any objects.

**NOTE:** Job is restartable after abort_step is used.
Did my job run parallel

• 2 Types of parallel
  1. multiple workers
  2. Parallel execution slaves

> expdp system/manager parallel=5
directory=dpump_dir
dumpfile=scott.dmp keep_master=y
Data unload in parallel

SQL> select m.object_schema, m.object_name, 
       (select count(*) from system.export_table t 
        where t.process_order = m.process_order and 
        t.duplicate!=0) pq_count 
    from system.sys_export_table_01 m 
    where m.process_order > 0 and m.object_type= 
      'TABLE_DATA'

    SCHEMA   NAME  PQ_COUNT
    ------   ----  -------
    SCOTT    EMP   2
How many workers started

select count(*) from
hr.sys_import_table_01
where process_order = -42;

COUNT(*)
--------
 1
KEEP_MASTER

- Indicates whether the master table should be deleted or retained at the end of a Data Pump job that completes successfully. The master table is automatically retained for jobs that do not complete successfully.

- KEEP_MASTER=[YES | NO]
ACCESS_METHOD

• Instructs Export to use a particular method to unload data.

• ACCESS_METHOD=[AUTOMATIC | DIRECT_PATH | EXTERNAL_TABLE]

• Provided so that you can try an alternative method if the default method does not work for some reason. Oracle recommends that you use the default option (AUTOMATIC).
Choosing the wrong access_method

> impdp system/manager tables=scott.foo_long directory=dpump_dir dumpfile=s.dmp access_method=external_table

Processing object type TABLE_EXPORT/TABLE/TABLE_DATA

ORA-31696: unable to export/import TABLE_DATA: "SCOTT"."FOO_LONG" using client specified EXTERNAL_TABLE method
Documented Parameters

Diagnostic Parameters

• Access_method
• Keep_master
• Metrics
• Abort_step
• Master_only
What is the Master Table

• Oracle table that is used to store information about the Data Pump job
  – Export/import parameters
  – Current status
  – Object information
• Can access the master table in SQLPLUS
• Not deleted if job is stopped
• Deleted after job is complete or killed
Master Table Contents

Some Interesting columns

• Process_order (+/- numbers)
• Object_type
• Object_schema
• Object_name
• Processing_state
• Processing_status
Interesting Process Orders

• Positive process orders describe objects that have been exported.

• Negative process orders describe the Data Pump job
  – -1/-2 Job state row – contains job status
  – -5/-6 completion rows – status for each object type
  – -41/-42 – worker status rows
  – -51/-52 – data filter rows
  – -53/-54 – metadata filter rows
  – -57/-58 – metadata transform rows
  – -59/-60 – job parameter rows
What object types are left

SQL> select unique object_type_seqno, object_type
from system.sys_import_full_01
where process_order > 0 AND processing_state = 'R'
   and processing_status = 'C';

OBJECT_PATH_SEQNO   OBJECT_TYPE
-----------------       ------------------
   103         PROCEDURE
     119     ALTER_PROCEDURE
   137         VIEW
What’s left for the current object

SQL> select object_schema, object_name
    from system.sys_import_full_01
    where process_order > 0 and processing_state = 'R'
        and processing_status = 'C' and
        object_path_seqno = 103;

<table>
<thead>
<tr>
<th>OBJECT_SCHEMA</th>
<th>OBJECT_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR</td>
<td>ADD_JOB_HISTORY</td>
</tr>
<tr>
<td>HR</td>
<td>SECURE_DML</td>
</tr>
</tbody>
</table>
Pharmacy Analytical Services

- Business unit in National Pharmacy Programs and Services, Kaiser Permanente.
- Kaiser Permanente is a not-for-profit health plan in 8 regions across the United States.
- Provides affordable, high-quality health care services to improve the health of our members and the communities we serve.
- 8.7 million members.
- 15,000 physicians and 165,000 employees.
- 35 hospitals and 454 medical offices.
- A large integrated electronic health record.
Pharmacy Analytical Services

- Chronic disease management for 1.1 million members.
- Member outreach with letters, telephony, secure messaging.
- Proactive management of member encounters.
- Measurement of clinical strategic goals.
- Forecasting, purchasing, inventory, distribution, prescribing, dispensing and monitoring of drug therapy.
- Intranet portal with over 100 applications and 30,000 users.
- Oracle databases since 1998 with 8.0.4, 8.1.7.3, 10.2.0.3 and now 11.2.0.2.
Oracle at Pharmacy Analytical Services

- 10 DBAs, 80 large databases (0.5T – 240T).
- 100% of large databases on Oracle.
- 95% data warehouse, datamart and business intelligence.

Oracle versions
- 10.2.0.3 20%
- 11.1.0.7 4%
- 11.2 76%

Host platforms
- Windows 2003 Server 20%
- Oracle Enterprise Linux 80%
Oracle at Pharmacy Analytical Services

- Commodity server approach
- High performance, low cost
- Multiple environments
- Scale horizontally
- Replace server and storage every 3–4 years
- R&D to identify next “building block”
- Current “building block”
  - HP Proliant DL580G5 16 cores or 24 cores w/256 GB RAM
  - 60 terabytes of direct attached storage
  - 1 gigabit network connection
  - Oracle Enterprise Linux 5.5 and Oracle Database 11.2.0.2
Overview of Oracle Environments

Data Staging

Data Warehouse

On-line Database

Production

Training

Test

Development

On-line Database

- EHR Integration
- Telephony
- Business Intelligence
- Pharmacy Integration
## Business Challenge – Oracle Database Server Migration in 24 Hr

<table>
<thead>
<tr>
<th>Existing 11 DB servers</th>
<th>New 13 DB servers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware</strong></td>
<td><strong>Hardware</strong></td>
</tr>
<tr>
<td>◦ HP x64 16 cores</td>
<td>◦ HP x64 24 cores</td>
</tr>
<tr>
<td>◦ HP Itanium 8 cores</td>
<td>◦ HP x64 16 cores</td>
</tr>
<tr>
<td>◦ HP AMD x64 32 cores</td>
<td></td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td><strong>Storage</strong></td>
</tr>
<tr>
<td>◦ 10 – 30 terabytes</td>
<td>◦ 60 terabytes</td>
</tr>
<tr>
<td><strong>OS</strong></td>
<td><strong>OS</strong></td>
</tr>
<tr>
<td>◦ Windows Server</td>
<td>◦ OEL 5.5</td>
</tr>
<tr>
<td><strong>Oracle</strong></td>
<td><strong>Oracle</strong></td>
</tr>
<tr>
<td>◦ 10.2, 11.1</td>
<td>◦ 11.2.0.2</td>
</tr>
</tbody>
</table>
## Solution Alternatives

<table>
<thead>
<tr>
<th>Solution</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Estimated Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel database environment</td>
<td>• Current database processes not affected</td>
<td>• Prolonged data loads</td>
<td>4–6 weeks based on prior experience</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Maintenance of scripts in two environments</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Extensive regression testing</td>
<td></td>
</tr>
<tr>
<td>Database backup and restore</td>
<td>• Traditional methodology</td>
<td>• Write back up file to file server.</td>
<td>110 hours = 20 terabytes @ 1 gigabit per second x 2</td>
</tr>
<tr>
<td></td>
<td>• Disaster recovery process</td>
<td>• Read back up file from file server.</td>
<td></td>
</tr>
<tr>
<td>Database link</td>
<td>• Know process</td>
<td>• Single threaded</td>
<td>55 hours plus = 20 terabytes @ 1 gigabit per second</td>
</tr>
<tr>
<td></td>
<td>• Multiple simultaneous links</td>
<td>• Separate migration for metadata</td>
<td></td>
</tr>
<tr>
<td>Data Pump in Network Link Mode plus 10 gigabit</td>
<td>• Move objects and metadata</td>
<td>• Unknown technology</td>
<td>14 hours = 20 terabytes @ 4 gigabits per second</td>
</tr>
<tr>
<td></td>
<td>• Direct source to target move</td>
<td>• Unproven technology</td>
<td></td>
</tr>
</tbody>
</table>
The best solution is ...

Oracle Data Pump in Network Link Mode with 10 gigabit
# Migration Progress Monitoring

<table>
<thead>
<tr>
<th>% Disk Write Time</th>
<th>0.003</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Idle Time</td>
<td>99.752</td>
<td>71.517</td>
<td>73.387</td>
<td>87.377</td>
<td></td>
</tr>
<tr>
<td>Avg. Disk Bytes/Read</td>
<td>987136.000</td>
<td>1048576.000</td>
<td>973577.714</td>
<td>1039213.714</td>
<td>9</td>
</tr>
<tr>
<td>Avg. Disk Bytes/Transfer</td>
<td>960866.779</td>
<td>1048576.000</td>
<td>973577.714</td>
<td>1039213.714</td>
<td>9</td>
</tr>
<tr>
<td>Avg. Disk Bytes/Write</td>
<td>884300.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Avg. Disk Queue Length</td>
<td>5.906</td>
<td>0.094</td>
<td>0.367</td>
<td>0.127</td>
<td></td>
</tr>
<tr>
<td>Avg. Disk Read Queue Length</td>
<td>5.927</td>
<td>0.094</td>
<td>0.267</td>
<td>0.127</td>
<td></td>
</tr>
<tr>
<td>Avg. Disk sec/Read</td>
<td>0.000</td>
<td>0.019</td>
<td>0.019</td>
<td>0.019</td>
<td>0.019</td>
</tr>
<tr>
<td>Avg. Disk sec/Transfer</td>
<td>0.000</td>
<td>0.019</td>
<td>0.019</td>
<td>0.019</td>
<td>0.019</td>
</tr>
<tr>
<td>Avg. Disk sec/Write</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Avg. Disk Write Queue Length</td>
<td>0.001</td>
<td>0.002</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Current Disk Queue Length</td>
<td>0.001</td>
<td>0.003</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Disk Bytes/sec</td>
<td>5242809079</td>
<td>1362592782</td>
<td>7223683771</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>Disk Read Bytes/sec</td>
<td>5242809079</td>
<td>1362592782</td>
<td>7223683771</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>Disk Write Bytes/sec</td>
<td>57065315</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Split IO/Sec</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Processes</th>
<th>CPU Usage</th>
<th>CPU Usage History</th>
</tr>
</thead>
<tbody>
<tr>
<td>1170</td>
<td>51.6%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>197624.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Physical Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>6710532</td>
</tr>
<tr>
<td>Available</td>
</tr>
<tr>
<td>50443600</td>
</tr>
<tr>
<td>Processes 73</td>
</tr>
<tr>
<td>System Cache</td>
</tr>
</tbody>
</table>

---

**Note:** The image includes various monitoring statistics and graphs related to disk usage, CPU usage, and network I/O. The highlighted areas indicate specific metrics or sections of interest, such as CPU usage and network traffic.
# Migration Recap – 1.5 TB/hr

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>HP DL580G5 16 core</td>
</tr>
<tr>
<td>Operating system</td>
<td>Windows 2003 Server</td>
</tr>
<tr>
<td>Oracle database</td>
<td>v11.1</td>
</tr>
<tr>
<td>Data read</td>
<td>286 MB/sec</td>
</tr>
<tr>
<td>CPU sending</td>
<td>82%</td>
</tr>
<tr>
<td>Network receiving</td>
<td>187 MB/sec</td>
</tr>
<tr>
<td>Data written</td>
<td>278 MB/sec</td>
</tr>
<tr>
<td>CPU receiving</td>
<td>51%</td>
</tr>
</tbody>
</table>
Progress at 4 Hours

.. imported "LKUSER"."KPHC_KEPT_APPT_OLD" 368,451,699 rows
.. imported "LKUSER"."KPHC_KEPT_APPT" 376,117,378 rows
.. imported "LKUSER"."OUTPAT_ENCOUNTERS_OLD" 488,651,654 rows
.. imported "LKUSER"."SUPERDAILY_CS_MONTH" 159,244,195 rows
.. imported "LKUSER"."OUTPAT_ENCOUNTERS" 575,578,889 rows
.. imported "LKUSER"."KRMS_CMT" 1,192,264,533 rows (billion)
.. imported "LKUSER"."KRMS_CMT_OLD" 1,192,264,533 rows (billion)
.. imported "LKUSER"."CS_LAB_HIST_OLD" 1,216,560,021 rows (billion)
.. imported "LKUSER"."LAB_HIST_CN" 1,031,083,214 rows (billion)
.. imported "LKUSER"."CS_LAB_HIST" 1,225,545,758 rows (billion)
.. imported "LKUSER"."LAB_HIST_CN_OLD" 1,023,007,500 rows (billion)
.. imported "LKUSER"."KPHC_OUTPAT_DX_OLD" 674,288,611 rows
.. imported "LKUSER"."OUTPAT_DX_OLD_O" 1,159,060,030 rows (billion)
.. imported "LKUSER"."MEMHIST" 1,009,593,929 rows (billion)
.. imported "BTUSER"."PT_ECS_FULL_INIT_WK_CN" 646,657,576 rows
.. imported "BTUSER"."KPHC_ROC_COV_MEMBERSHIP_OLD" 893,791,915 rows
.. imported "LKUSER"."KPHC_OUTPAT_DX" 689,554,580 rows
.. imported "LKUSER"."ABSTRACT_ENCOUNTERS_DX" 138,161,570 rows
.. imported "BTUSER"."VT_10A_PIMS_EXTR_99Q11_TEST" 499,758,332 rows
Network Data Pump Workers

Job: SYS_IMPORT_SCHEMA_01
  Operation: IMPORT
  Mode: SCHEMA
  State: EXECUTING
  Bytes Processed: 0
  Current Parallelism: 16
  Job Error Count: 0

Worker 1 Status:
  Process Name: DW00
  State: EXECUTING
  Object Schema: BTUSER
  Object Name: KC_MRR_10B_NW_ABC1_88Q44
  Object Type: SCHEMA_EXPORT/TABLE/TABLE_DATA
  Completed Objects: 15
  Total Objects: 23,926
  Worker Parallelism: 1

Worker 2 Status:
  Process Name: DW01
  State: EXECUTING
  Object Schema: BTUSER
  Object Name: KC_DIR_10A_PARTD1_99
  Object Type: SCHEMA_EXPORT/TABLE/TABLE_DATA
  Completed Objects: 9
  Total Objects: 23,926
  Worker Parallelism: 1

Worker 3 Status:
  Process Name: DW02
  State: EXECUTING
  Object Schema: LKUSER
  Object Name: JUTILSUMM_HIST
  Object Type: SCHEMA_EXPORT/TABLE/TABLE_DATA
  Completed Objects: 1
  Total Objects: 23,926
  Worker Parallelism: 1
Clean up After Transfer is Complete

- Check log for successful transfer of metadata, objects and packages.
- Update tnsnames.ora.
- Create database links.
- Re-activate users and passwords.
- Regression test processes and check results.
- Roll back to old server, if needed.
Data Pump on 10 gigabit & OEL 5.5

Gigabits per Second

16 Core Server | 24 Core Server | 32 Core Server | 64 Core Server
Data Pump on 1 gigabit vs 10 gigabit
Minutes to move 1,159 GB

<table>
<thead>
<tr>
<th>Cores</th>
<th>1 Gb</th>
<th>10 Gb</th>
<th>10 Gb</th>
<th>10 Gb</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>120</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Our Next Database Migration
64 cores @ 7 Gb/sec (3 TB/hr)
HP DL980 w/64 cores
Network Data Pump Learnings

- Parallel data pump workers
- Compression of network transfer
- Already compressed table
- Append hint
- Partition syntax and locking
- Statistics
- Platform agnostic
- Windows limit on 10 gigabit
Another Use for Data Pump – Backup

- Transform parameter
“Don’t forget the milk!”

- Network link data pump works

- Important patches to consider
  - Append (9721663)
  - Partition wise syntax (11677757)
  - Partition truncate (8692663 for v11.1)

- Oracle transfer on 10 gigabit network
  - Windows 2003 @ 3 Gb/sec
  - Windows 2008 @ 5 Gb/sec
  - OEL 5.5 @ 7 Gb/sec
Demogrounds

• Come see us in the Demogrounds at Moscone South 7460
• Demos of what’s in a Master Table
Hardware and Software
Engineered to Work Together