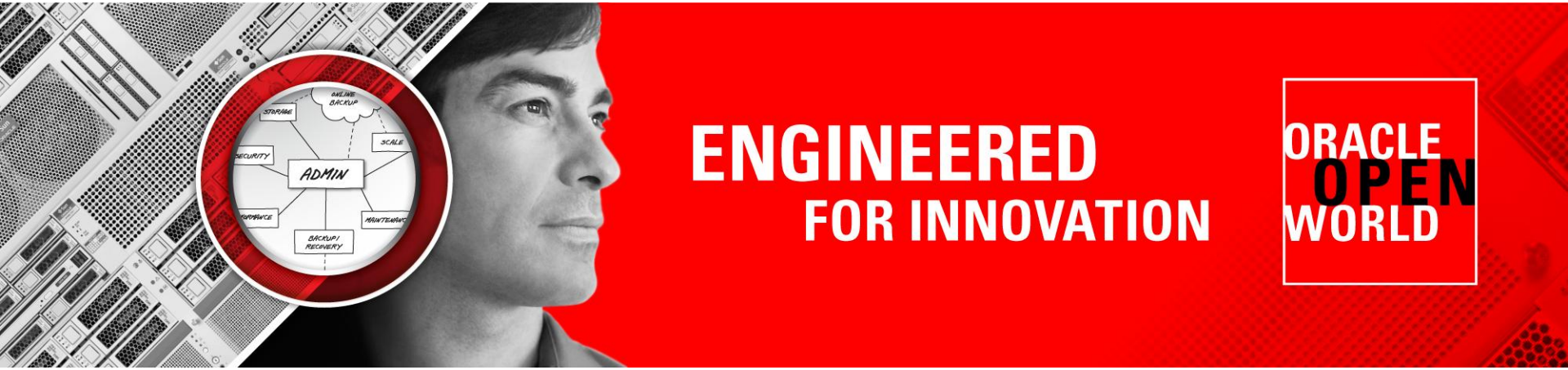

ORACLE®

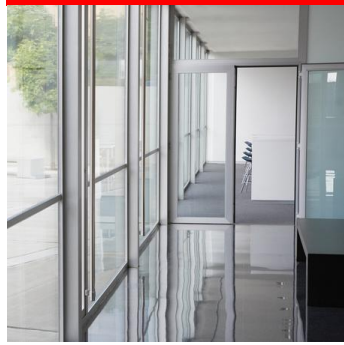


Mastering Oracle Data Pump

Dean Gagne (Oracle)
Udi Karni (Kaiser Permanente)

Presenting with

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 10*g* 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
```

MASTER_ONLY

- 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';

OBJECT_TYPE        OBJ OBJECT_NAME
------------------ --- -------------------------
PROCEDURE          HR  ADD_JOB_HISTORY
ALTER_PROCEDURE    HR  ADD_JOB_HISTORY
INDEX              HR  REG_ID_PK
INDEX              HR  LOC_ID_PK
INDEX              HR  DEPT_ID_PK
```

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;
OBJECT_SCHEMA                  OBJECT_NAME
------------------------------ ------------------------------
HR                             ADD_JOB_HISTORY
HR                             SECURE_DML
```

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 communites 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.

KAISER PERMANENTE®

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.

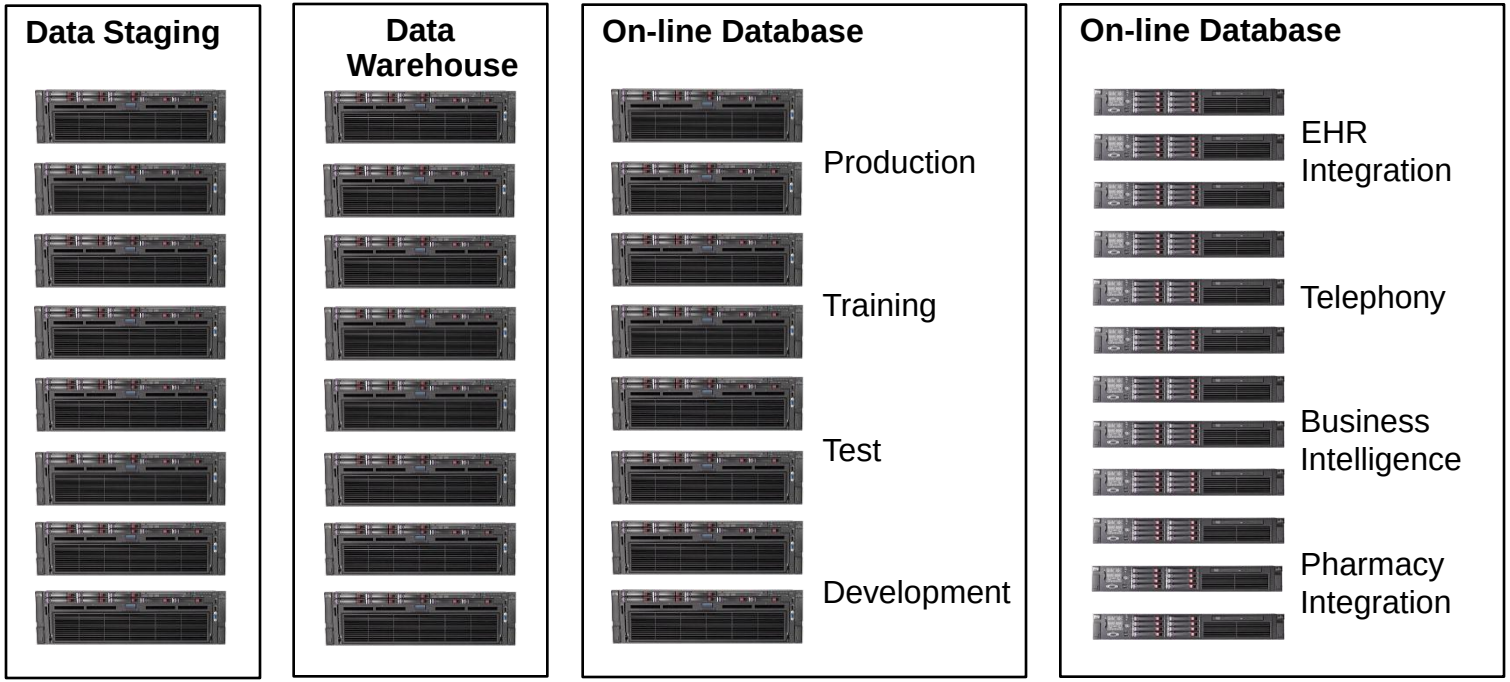
Oracle at Pharmacy Analytical Services

- 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

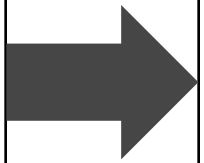
Overview of Oracle Environments



Business Challenge – Oracle Database Server Migration in 24 Hr

Existing 11 DB servers

- Hardware
  - HP x64 16 cores
  - HP Itanium 8 cores
  - HP AMD x64 32 cores
- Storage
  - 10 – 30 terabytes
- OS
  - Windows Server
- Oracle
  - 10.2, 11.1



New 13 DB servers

- Hardware
  - HP x64 24 cores
  - HP x64 16 cores
- Storage
  - 60 terabytes
- OS
  - OEL 5.5
- Oracle
  - 11.2.0.2

KAISER PERMANENTE®

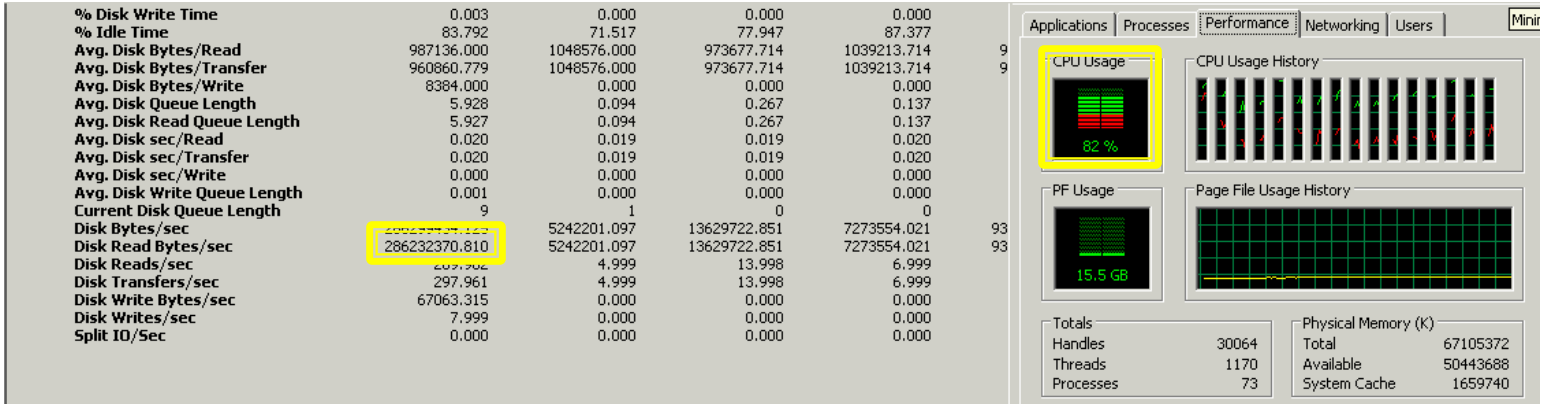
Solution Alternatives

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

The best solution is …

Oracle Data Pump in Network Link Mode with 10 gigabit

Migration Progress Monitoring



% Disk Write Time	0.003	0.000	0.000	0.000	
% Idle Time	83.792	71.517	77.947	87.377	
Avg. Disk Bytes/Read	987136.000	1048576.000	973677.714	1039213.714	9
Avg. Disk Bytes/Transfer	960860.779	1048576.000	973677.714	1039213.714	9
Avg. Disk Bytes/Write	8384.000	0.000	0.000	0.000	
Avg. Disk Queue Length	5.928	0.094	0.267	0.137	
Avg. Disk Read Queue Length	5.927	0.094	0.267	0.137	
Avg. Disk sec/Read	0.020	0.019	0.019	0.020	
Avg. Disk sec/Transfer	0.020	0.019	0.019	0.020	
Avg. Disk sec/Write	0.000	0.000	0.000	0.000	
Avg. Disk Write Queue Length	0.001	0.000	0.000	0.000	
Current Disk Queue Length	9	1	0	0	
Disk Bytes/sec	[illegible]	5242201.097	13629722.851	7273554.021	93
Disk Read Bytes/sec	286232370.810	5242201.097	13629722.851	7273554.021	93
Disk Reads/sec	[illegible]	4.999	13.998	6.999	
Disk Transfers/sec	297.961	4.999	13.998	6.999	
Disk Write Bytes/sec	67063.315	0.000	0.000	0.000	
Disk Writes/sec	7.999	0.000	0.000	0.000	
Split IO/Sec	0.000	0.000	0.000	0.000	

Applications | Processes | Performance | Networking | Users

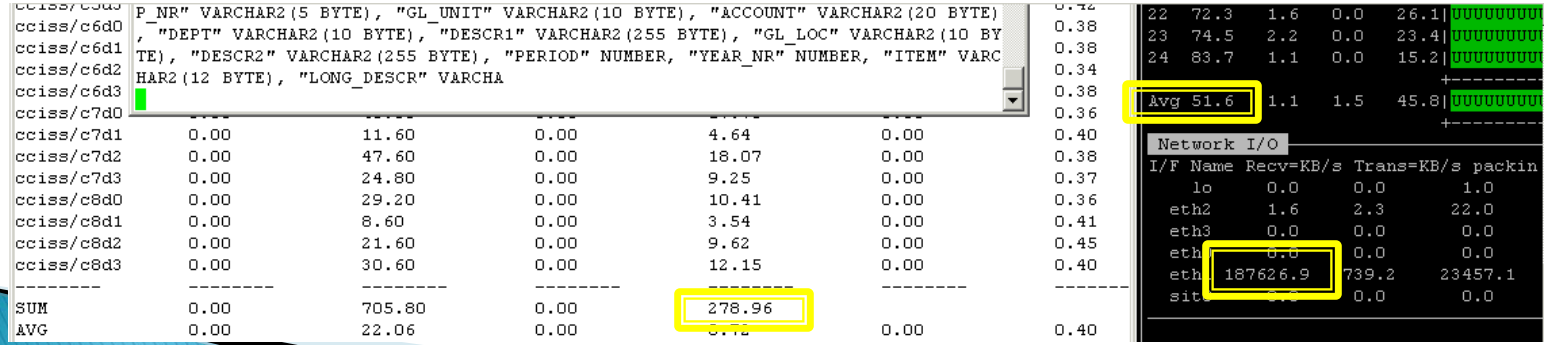
CPU Usage: 82 %

CPU Usage History

PF Usage: 15.5 GB

Page File Usage History

Totals		Physical Memory (K)	
Handles	30064	Total	67105372
Threads	1170	Available	50443688
Processes	73	System Cache	1659740



```
P_NR" VARCHAR2(5 BYTE), "GL_UNIT" VARCHAR2(10 BYTE), "ACCOUNT" VARCHAR2(20 BYTE)
, "DEPT" VARCHAR2(10 BYTE), "DESCR1" VARCHAR2(255 BYTE), "GL_LOC" VARCHAR2(10 BY
TE), "DESCR2" VARCHAR2(255 BYTE), "PERIOD" NUMBER, "YEAR_NR" NUMBER, "ITEM" VARC
HAR2(12 BYTE), "LONG_DESCR" VARCHA
```

cciss/c6d0						0.38
cciss/c6d1						0.38
cciss/c6d2						0.34
cciss/c6d3						0.38
cciss/c7d0						0.36
cciss/c7d1	0.00	11.60	0.00	4.64	0.00	0.40
cciss/c7d2	0.00	47.60	0.00	18.07	0.00	0.38
cciss/c7d3	0.00	24.80	0.00	9.25	0.00	0.37
cciss/c8d0	0.00	29.20	0.00	10.41	0.00	0.36
cciss/c8d1	0.00	8.60	0.00	3.54	0.00	0.41
cciss/c8d2	0.00	21.60	0.00	9.62	0.00	0.45
cciss/c8d3	0.00	30.60	0.00	12.15	0.00	0.40
SUM	0.00	705.80	0.00	278.96		
AVG	0.00	22.06	0.00	[illegible]	0.00	0.40

```
22  72.3    1.6    0.0    26.1|uuuuuuuuu
23  74.5    2.2    0.0    23.4|uuuuuuuuu
24  83.7    1.1    0.0    15.2|uuuuuuuuu
                              +---------
Avg 51.6    1.1    1.5    45.8|uuuuuuuuu
                              +---------
```

Network I/O

I/F Name	Recv=KB/s	Trans=KB/s	packin
lo	0.0	0.0	1.0
eth2	1.6	2.3	22.0
eth3	0.0	0.0	0.0
eth	[illegible]	0.0	0.0
eth	187626.9	739.2	23457.1
sit	[illegible]	0.0	0.0

KAISER PERMANENTE®

Migration Recap – 1.5 TB/hr

	Source	Target
Server	HP DL580G5 16 core	HP DL580G5 24 core
Operating system	Windows 2003 Server	OEL 5.5
Oracle database	v11.1	v11.2
Data read	286 MB/sec	
CPU sending	82%	
Network receiving		187 MB/sec
Data written		278 MB/sec
CPU receiving		51%

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"               1,159,060,603 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
```

KAISER PERMANENTE®

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

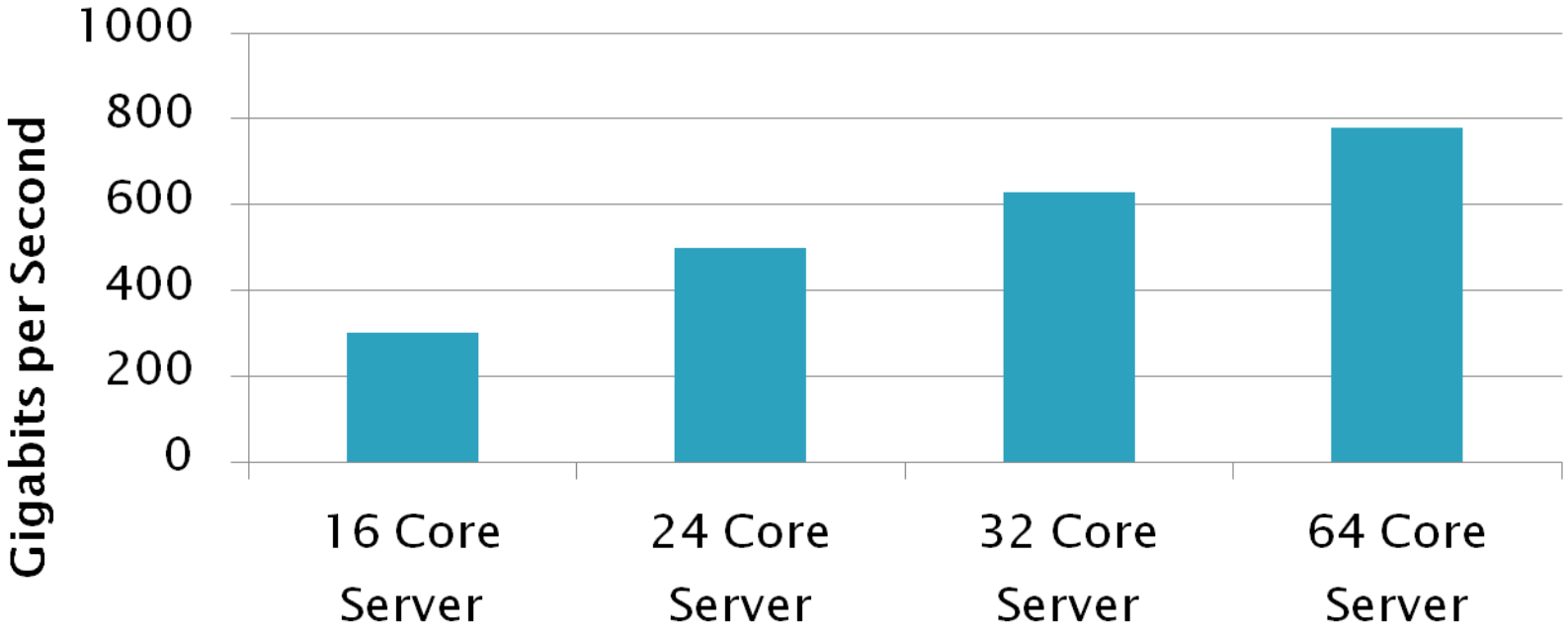
Worker 3 Status:
  Process Name: DW02
  State: EXECUTING
  Object Schema: LKUSER
  Object Name: JUTILSUMM_HIST
  Object Type: SCHEMA_EXPORT/TABLE/TABLE_DATA
  Completed Objects: 1
  Total Objects: 23,926
  Worker Parallelism: 1

KAISER PERMANENTE®

Clean up After Transfer is Complete

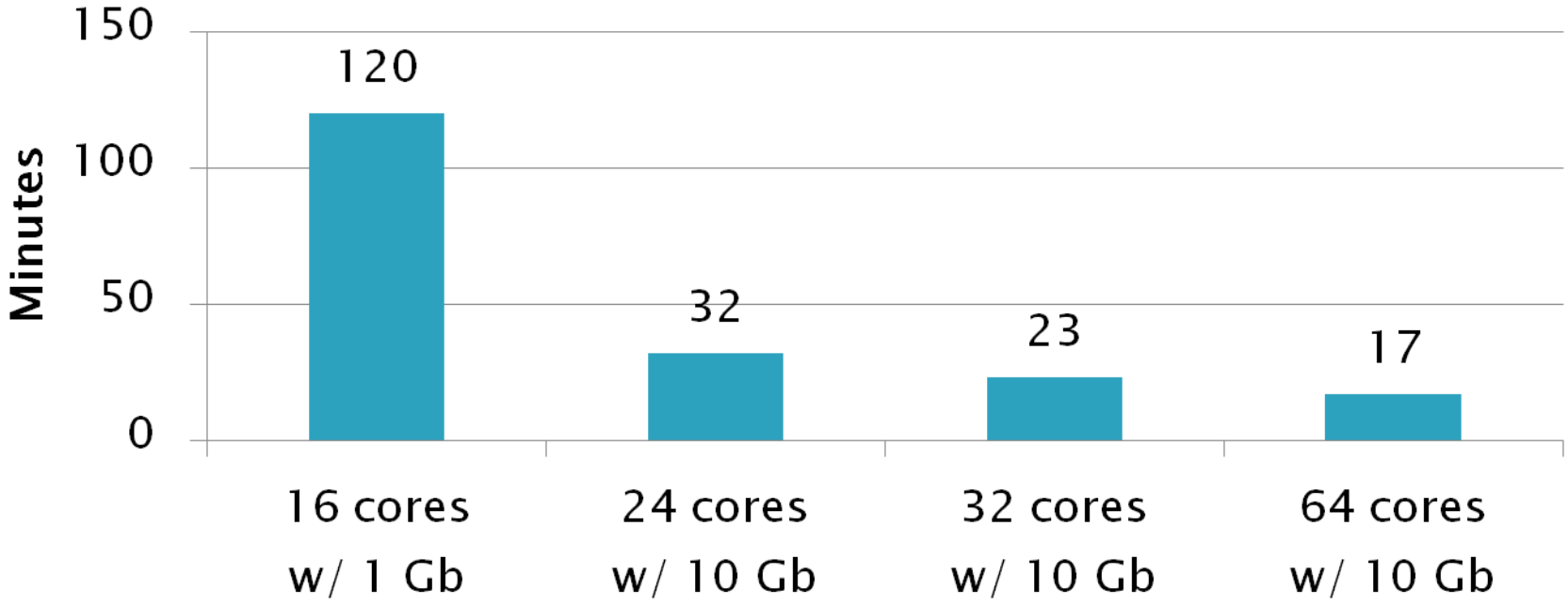
- Check log for successful transfer of metadata, objects and packages.
- Update tnsnames.ora.
- Create database links.
- Re-activate users and passwords.
- Regression test processes and check results.
- ~~Roll back to old server, if needed~~.

Data Pump on 10 gigabit & OEL 5.5



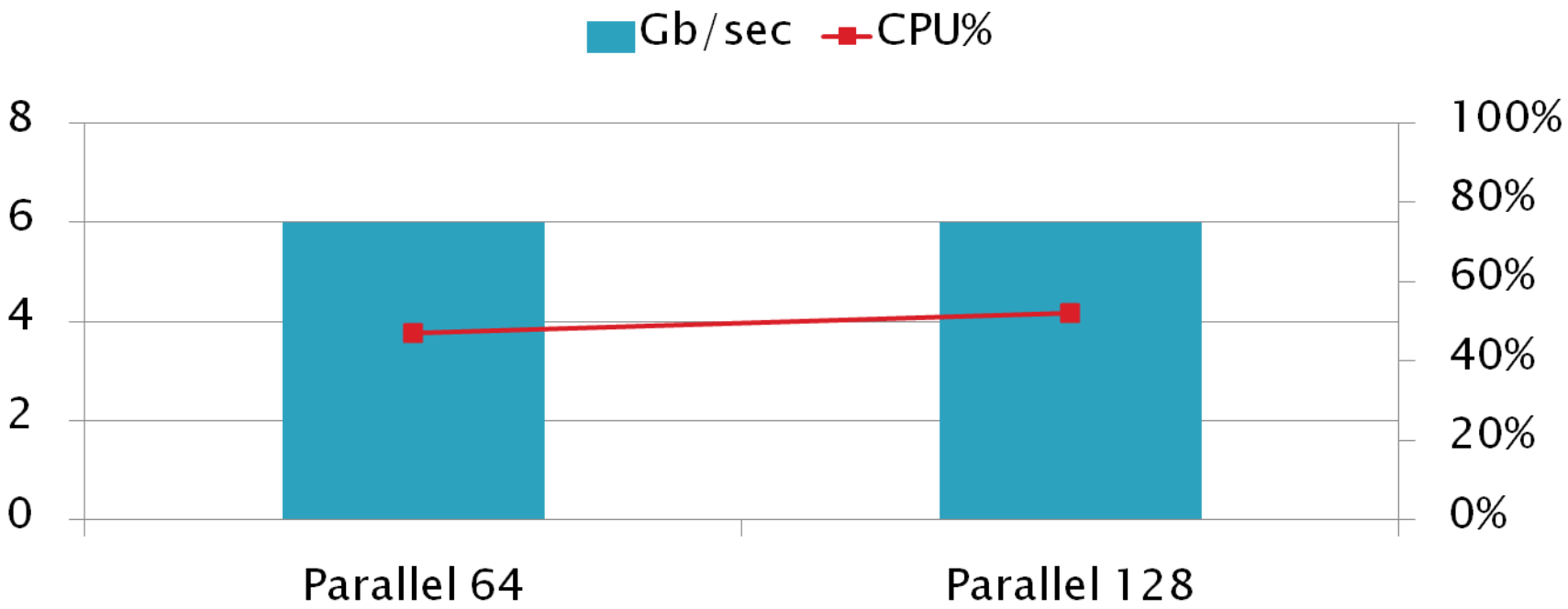
KAISER PERMANENTE®

Data Pump on 1 gigabit vs 10 gigabit
Minutes to move 1,159 GB

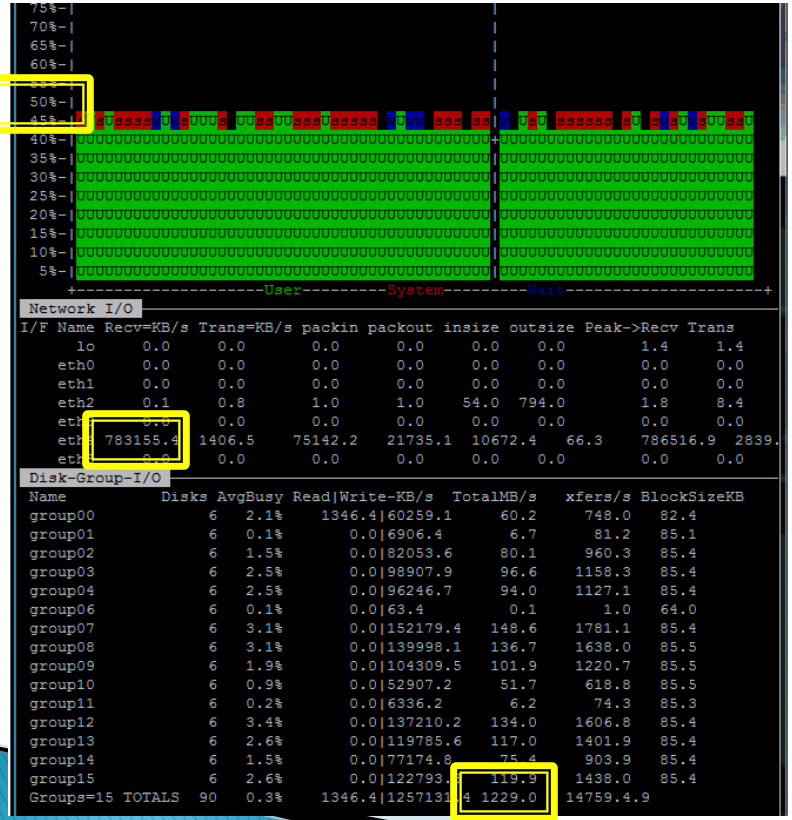


KAISER PERMANENTE®

Our Next Database Migration
64 cores @ 7 Gb/sec (3 TB/hr)



HP DL980 w/64 cores



. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2010_01" 2135188 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2003_12" 2023239 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2005_03" 2203362 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2007_10" 2102147 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2008_02" 2131969 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2002_05" 2082077 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2006_01" 2084158 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2003_05" 2073318 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CS_2010_01" 2087725 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2005_08" 2017751 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2006_12" 2009055 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2009_12" 2212587 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2004_01" 2014382 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2007_12" 1987293 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CS_2009_01" 2104171 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CS_2009_03" 2104520 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2009_05" 2068737 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2010_02" 2024050 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2009_07" 2117748 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2006_03" 2205938 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2008_07" 2063063 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2009_01" 2172426 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2008_04" 2166387 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CS_2009_12" 2109116 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2007_03" 2222745 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2007_04" 2024668 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2008_03" 2137890 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CS_2009_10" 2122992 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2004_03" 2131659 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CS_2010_04" 2108474 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2005_12" 2108890 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2008_12" 2143973 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2007_06" 2023911 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CS_2009_07" 2044299 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2007_05" 2143160 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2002_03" 2034949 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2009_09" 2046738 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2003_01" 2077223 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2008_01" 2246326 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2010_04" 2184369 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2006_06" 2044064 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2003_03" 2069259 rows
. . imported "DBAUSER"."UDI_JUTILSUMM_HIST":"JUTILSUMM_CN_2007_08" 2047402 rows

KAISER PERMANENTE®

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 on10 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

Q&A

Hardware and Software

ORACLE

Engineered to Work Together


ORACLE®