Transportable Tablespaces for Scalable Re-Instantiation

Eva Dafonte Pérez

Physics Databases Services Update

Dawid Wojcik

Transportable Tablespaces for Scalable Re-Instantiation

Eva Dafonte Pérez

Outline

- Overview
- Transportable Tablespaces
- Pulling Tablespaces
- Pros and Cons
- Streams Re-synchronization procedure using Transportable Tablespaces
  - Idea
  - Example and Steps
- Summary

Overview

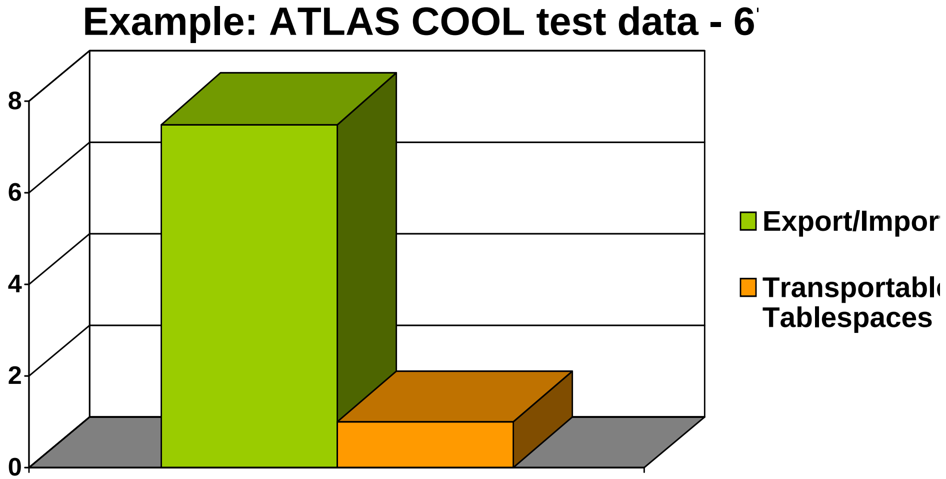
- Target site out of the Streams recovery window
- Complete transfer of data (schemas and tables) using Oracle Data Pump utility to destination database may take days
  - Example ATLAS Conditions data

➔Find **fastest** way to move data

- Options:
  - Transportable Tablespaces
  - Pulling Tablespaces (new in Oracle 10g)

Transportable Tablespaces

- Move a set of tablespaces from one Oracle database to another
- Export metadata of tablespace instead of data in tablespace
  - Datafiles containing all data copied to destination
  - Plug the tablespace by importing metadata
- Moving data using transportable tablespaces is much faster than export/import



Transportable Tablespaces

- Restrictions:
  - Database block size and character set must be the same at source and target
  - The Oracle release and the compatibility must be the same or higher at target
  - Tablespace must be self contained
  - User/s must exist at target database
- Cross-Platform since Oracle 10g
  - Oracle provides a byte-order-conversion solution that uses Oracle Recovery Manager (RMAN)

Pulling Tablespaces

- Single packaged procedure
  - DBMS_STREAMS_TABLESPACE_ADM
    - PULL_SIMPLE_TABLESPACE
    - PULL_TABLESPACES
  - Uses Data Pump to transport tablespaces
  - Transfers the data files to the target system's format
    - Using DBMS_FILE_TRANSFER
  - Performs any required endian conversion automatically
  - Creates a log file
- All tasks are encapsulated

Pros and Cons

- Using Transportable Tablespaces:
  - Enables a full set of tablespaces to be moved
  - Fastest approach
  - Tablespaces need to be set to read-only while the files are copied

- Using Pulling Tablespaces:
  - Combines all steps in the transportable tablespaces approach into a single step
  - Less flexibility - multiple files transferred sequentially
  - Tablespaces need to be set to read-only

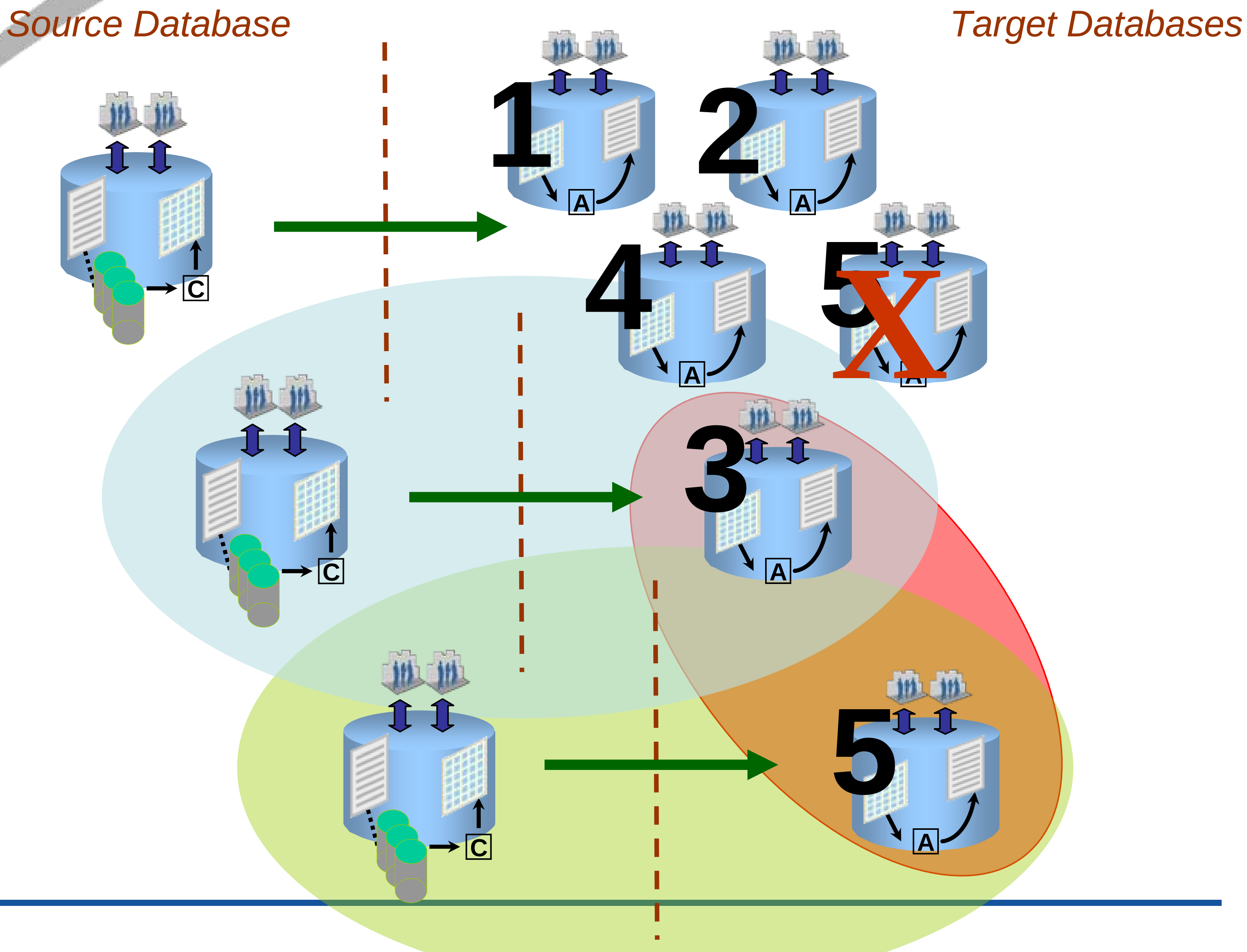
Streams Re-Synchronization

- Idea:
  - Use transportable tablespaces to move data faster
  - Complete re-instantiation using Streams

- Transport tablespaces from Tier0 is **NOT** possible
  - Cannot be read-only
  - Contains more data than Tier1's tablespaces

- BUT, we can transport tablespaces from Tier1
  - Can be read-only
  - Contains exactly the data that must be transferred

Streams Re-Synchronization
Example and Steps



Streams Re-Synchronization

Example and Steps

- Example:
  - Original streaming to sites 1, 2 , 4 and 5
  - Separate streaming for site 3
  - Site 3 is out of sync
  - Would like to use site 5 to synchronize site 3
  - Have to move tablespaces “TS1” and “TS2”
- Steps:
  - Check tablespaces set is self-contained

```
EXEC SYS.DBMS_TTS.TRANSPORT_SET_CHECK ('TS1','TS2');
SELECT * FROM TRANSPORT_SET_VIOLATIONS;
```

  - Split streaming for site 5

Streams Re-Synchronization
Example and Steps

- Steps:
  - Check sites 3 and 5 are able to connect to each other
  - Create database links between databases
  - Create directories pointing to datafiles and grant access to streams administrator on both sites
  - Stop replication to site 5
  - Ensure tablespaces are read-only

```
SELECT SELECT STATUS FROM DBA_TABLESPACES
WHERE TABLESPACE_NAME IN ('TS1','TS2');

ALTER TABLESPACE TS1 READ ONLY;

ALTER TABLESPACE TS2 READ ONLY;
```

Streams Re-Synchronization
Example and Steps

- Steps:
  - Transfer the data files of each tablespace to the remote system
  - Import tablespaces metadata in the target

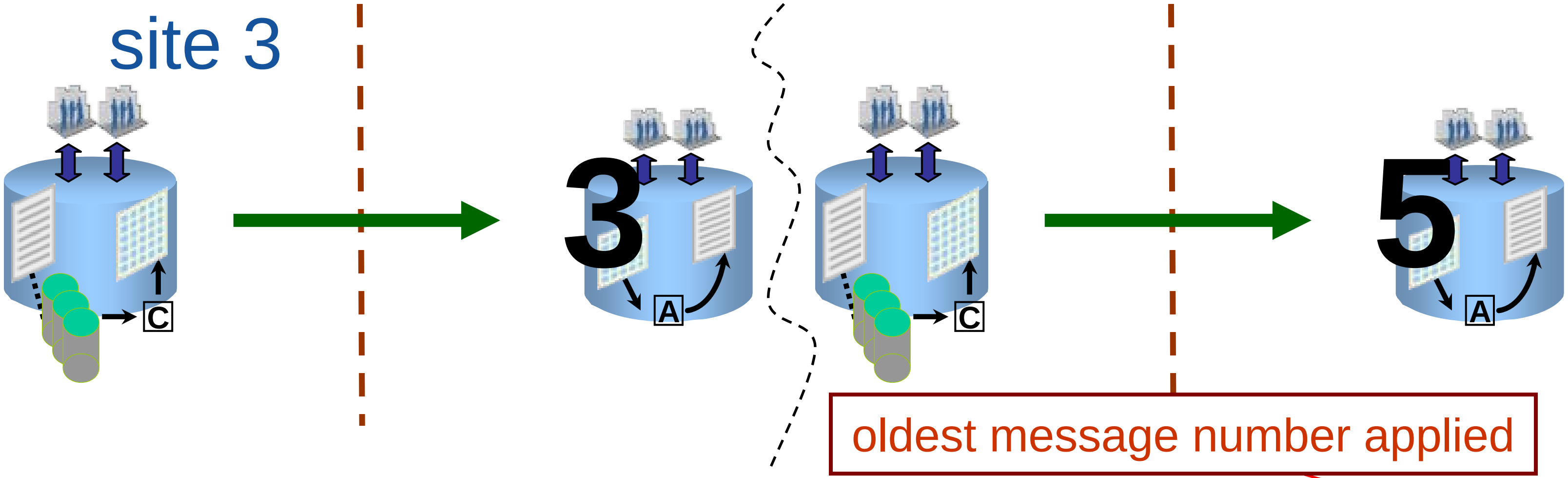
```
impdp user/pwd
TRANSPORT_DATAFILES="'/oradata/ts1_1.dbf','/oradata/ts2_1.dbf'"
NETWORK_LINK='srcdb'
TRANSPORT_TABLESPACES=(TS1,TS2)
NOLOGFILE=Y
```

  - Make tablespaces read-write

```
ALTER TABLESPACE TS1 READ WRITE;
ALTER TABLESPACE TS2 READ WRITE;
```

Streams Re-Synchronization
Example and Steps

- We have a consistent copy of the data between sites 3 and 5
- And we need to re-configure replication to site 3

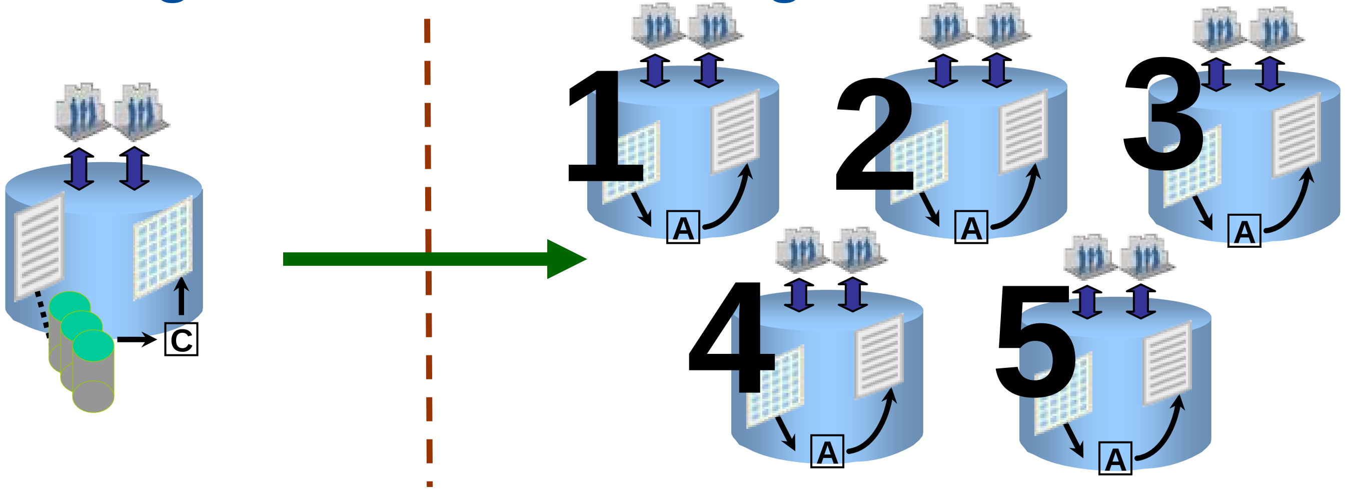


oldest message number applied

➔ first_scn = previous dictionary build
➔ start_scn =

Streams Re-Synchronization
Example and Steps

- Enable replication to site 5
- Enable replication to site 3
- Last transaction might be reapplied!!!!
  - Just ignore the error
- Wait
- Merge all the streaming



Summary

- Transportable tablespaces is a fastest way to move data between databases
- Can be used between Tier1 sites when complete synchronization is needed
- Successfully tested during CNAF re-synchronization for ATLAS conditions data

Physics Databases Services Update

Dawid Wojcik

Outline

- Service Expansion and Upgrade
- New Monitoring Features

Service Expansion and Upgrade

- Moved most production services from 32-bit Linux to 64-bit Linux
  - 60 new mid-range servers (dual quadcores) in total (production and integration), 34 new SAN arrays
  - Migration using **Oracle DataGuard** (direct cross platform – supported by Oracle)
    - minimum downtime required (independent of database size)
    - easy to rollback if something goes wrong
    - recompilation of all PL/SQL required at the end of the process (few minutes)
    - old databases left in Data Guard configuration as standbys
  - Installation & DataGuard procedures reviewed and updated

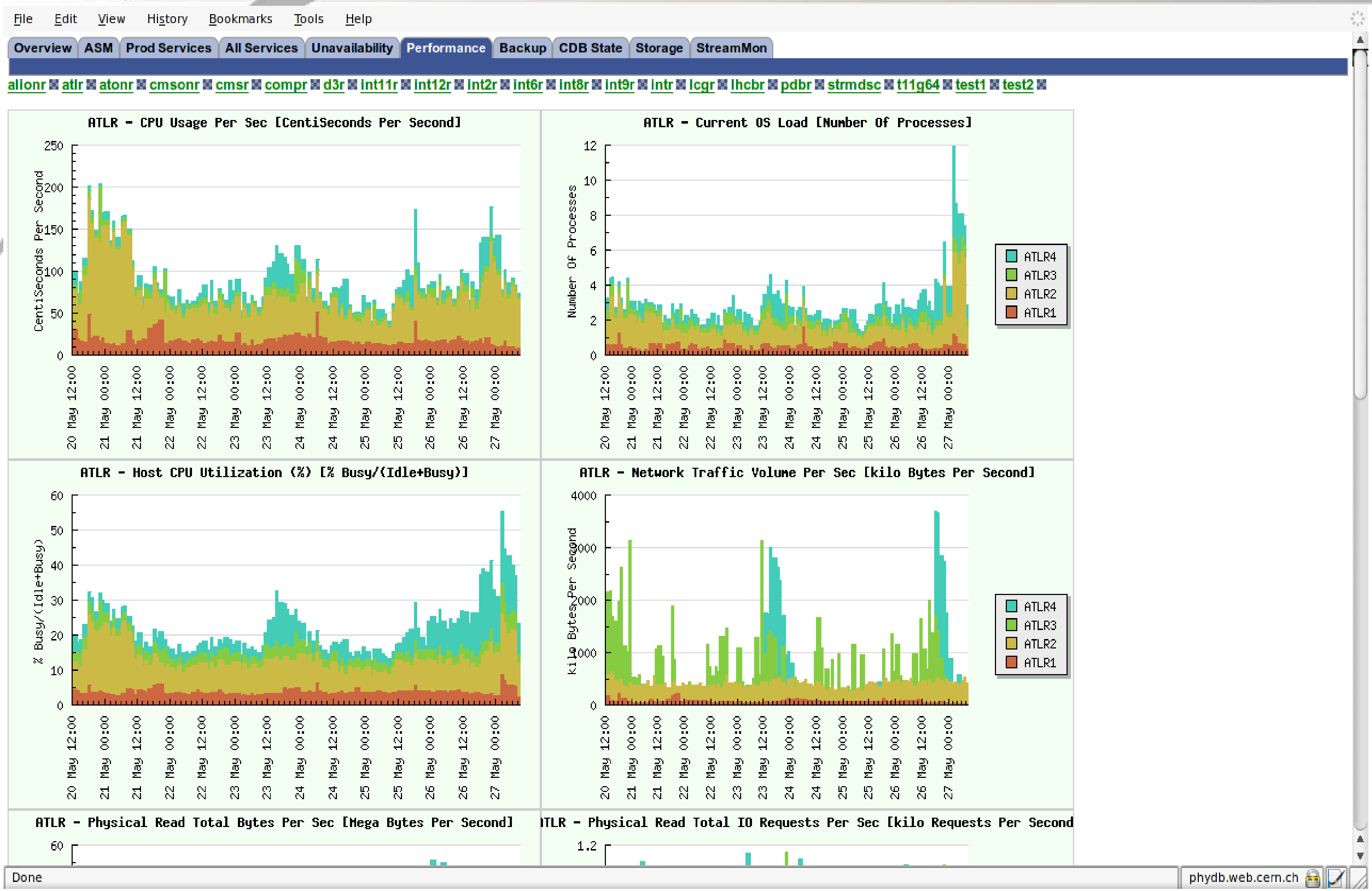
Service Expansion and Upgrade

- Upgrade of production databases to Oracle 10.2.0.4
  - 10.2.0.4 evaluated by experiments on 64-bit Linux in test and integration environments over last month
  - No major problems observed
    - agent compatibility issues discovered
  - Upgrades of production databases currently individually scheduled with experiments

Monitoring Update

- Oracle Enterprise Manager for 3D
  - Securing agents in all T1s
  - Revoked Super Administrators privileges from T1 DBAs
    - Automating grouping and privilege assignment implemented
- Streams monitoring secured with password
- RAC monitoring
  - Many improvements (additional components monitored – storage, huge memory usage)
  - Improved performance plotting mechanisms (used in experiment reports)

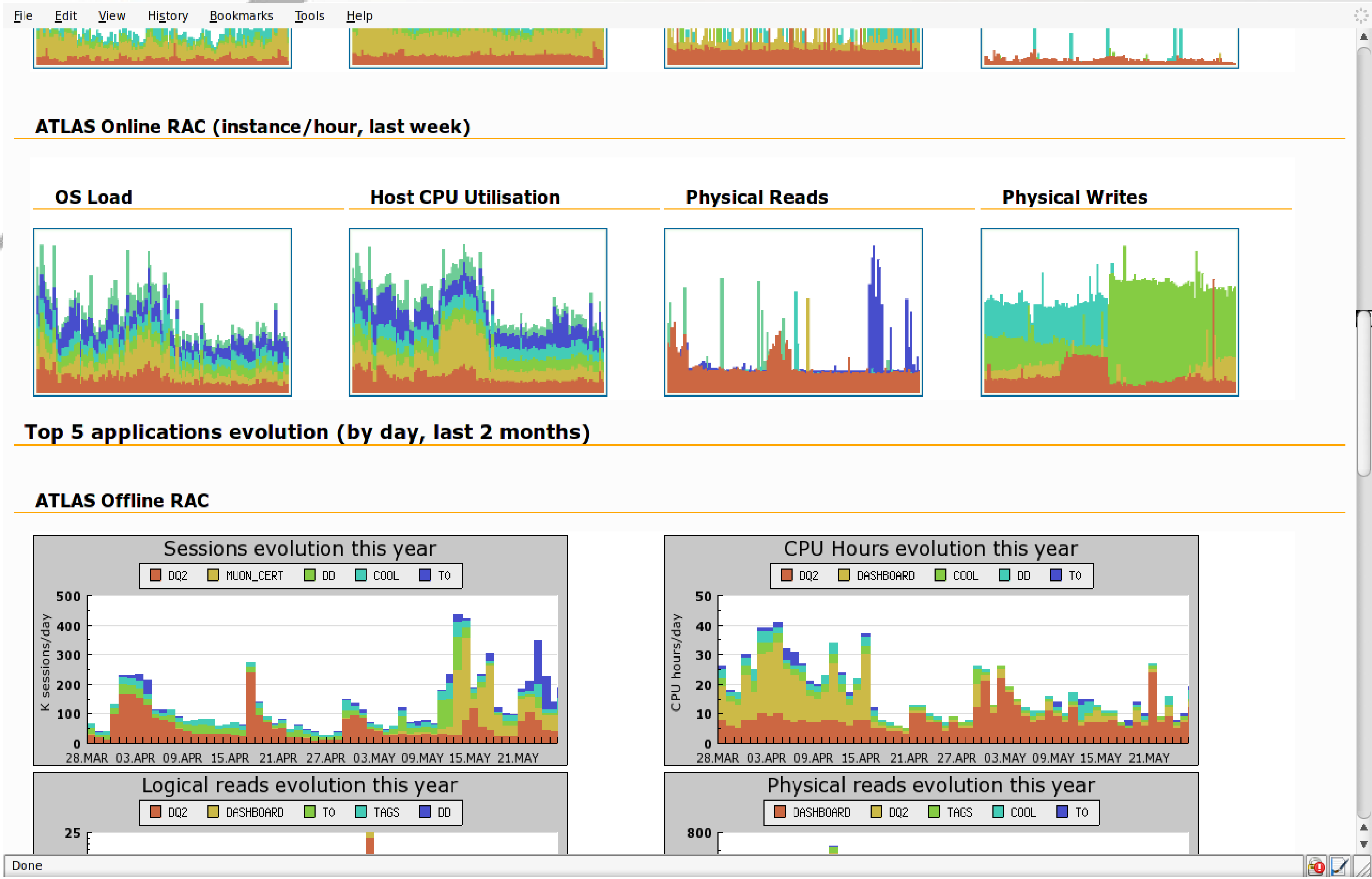
Monitoring Update


File Edit View History Bookmarks Tools Help
Overview | ASM | Prod Services | All Services | Unavailability | Performance | Backup | CDB State | Storage | StreamMon
allonr ☒ atlr ☒ atonr ☒ cmsonr ☒ cmsr ☒ compr ☒ d3r ☒ int11r ☒ int12r ☒ int2r ☒ int6r ☒ int8r ☒ int9r ☒ intr ☒ lcgr ☒ lhcbr ☒ pdbr ☒ strmdsc ☒ t11g64 ☒ test1 ☒ test2 ☒
ATLR - CPU Usage Per Sec [CentiSeconds Per Second]
CentiSeconds Per Second
250
200
150
100
50
0
ATLR - Current OS Load [Number Of Processes]
Number Of Processes
12
10
8
6
4
2
0
🟦 ATLR4
🟩 ATLR3
🟨 ATLR2
🟥 ATLR1
ATLR - Host CPU Utilization (%) [% Busy/(Idle+Busy)]
% Busy/(Idle+Busy)
60
50
40
30
20
10
0
ATLR - Network Traffic Volume Per Sec [kilo Bytes Per Second]
kilo Bytes Per Second
4000
3000
2000
1000
0
🟦 ATLR4
🟩 ATLR3
🟨 ATLR2
🟥 ATLR1
20 May 12:00
21 May 00:00
21 May 12:00
22 May 00:00
22 May 12:00
23 May 00:00
23 May 12:00
24 May 00:00
24 May 12:00
25 May 00:00
25 May 12:00
26 May 00:00
26 May 12:00
27 May 00:00
ATLR - Physical Read Total Bytes Per Sec [Mega Bytes Per Second]
60
TLR - Physical Read Total IO Requests Per Sec [kilo Requests Per Second
1.2
Done
phydb.web.cern.ch

Monitoring Update

- New experiment reports
  - Resource usage of experiments grouped per application (Oracle services)
  - Experiment dashboards prepared reusing existing monitoring infrastructure (SLS, Streams Monitoring, RAC Mon and experiment reports) and additional development

Monitoring Update



ATLAS Online RAC (instance/hour, last week)

OS Load

Host CPU Utilisation

Physical Reads

Physical Writes

Top 5 applications evolution (by day, last 2 months)

ATLAS Offline RAC

Questions ?