Update on activities for the physics database services

17th March 2009

Svetozár Kapusta
Eva Dafonte Pérez
Dawid Wojcik
Luca Canali
Jacek Wojcieszuk
Maria Girone

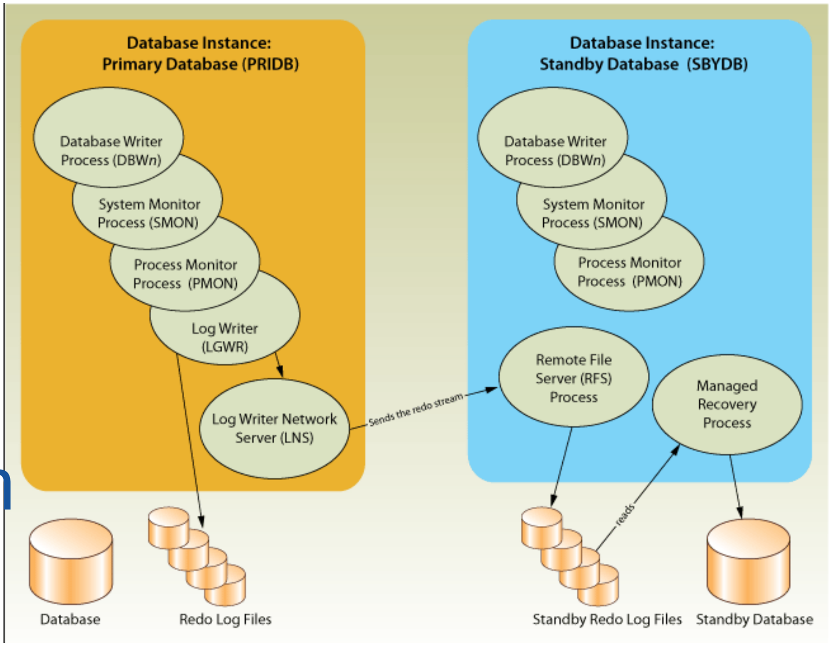
CERN openlab

Outline

- Active Data Guard
- Compression
- ACFS
- Streams

Introduction to Active Data Guard

- Data Guard belongs to MAA best practices
- Redo entries from the primary DB are applied to the standby DB continuously
- If the primary fails, the standby database can be quickly activated
- Physical standby DB can be opened for read-only access
- Reporting and backups can then run on the standby DB


Database Instance:
Primary Database (PRIDB)
Database Writer Process (DBWn)
System Monitor Process (SMON)
Process Monitor Process (PMON)
Log Writer (LGWR)
Log Writer Network Server (LNS)
Sends the redo stream
Database Instance:
Standby Database (SBYDB)
Database Writer Process (DBWn)
System Monitor Process (SMON)
Process Monitor Process (PMON)
Remote File Server (RFS) Process
Managed Recovery Process
reads
Database
Redo Log Files
Standby Redo Log Files
Standby Database

Active Data Guard Test Setup

- Installed in Computer Centre:
- HW:
  - Primary: 2 RAC nodes, 2x Xeon CPU 3.00GHz, 2MB cache, 4GB RAM
  - Standby: 2 RAC nodes, 8x Xeon CPU 2.33GHz, 6MB cache, 16GB RAM
- SW: RHEL 4, RDBMS 11.1.0.7
- ASM:
  - Primary: 2 disk arrays with 12*250GB disks each
  - Standby: 2 disk arrays with 16*370GB disks each

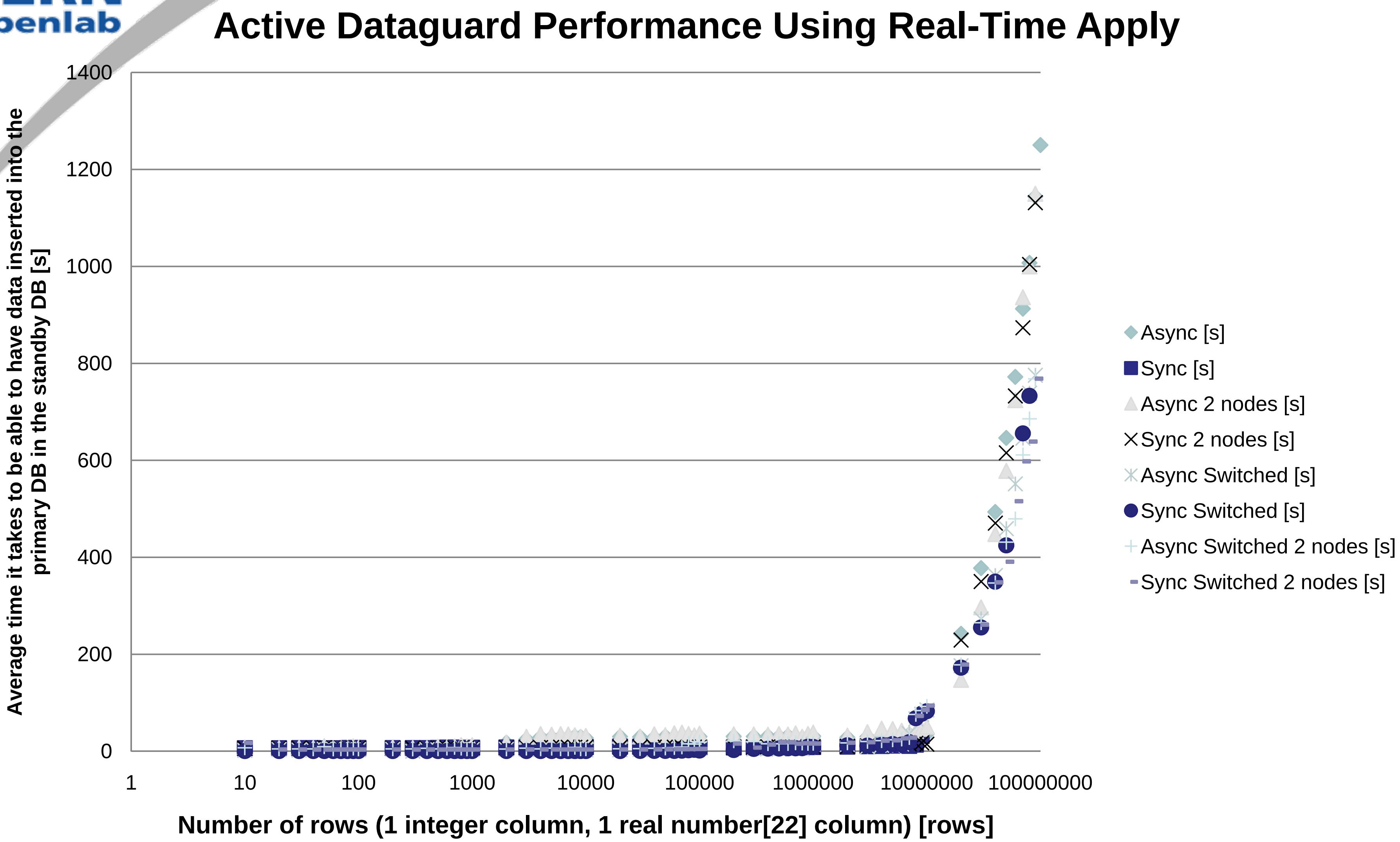
Active Data Guard Functional Verification

- Installed using RMAN's "duplicate target database for standby from active database"
  - Was successful (although not trivial)
  - Faster than conventional standby creation (didn't measure it, HITACHI measured factor 3X faster)
- Running smoothly more than 3 months even with high load
  - Only one error – ASM shared pool
- Data comparison shows error rate <16e^{-12} bit
- Verified read consistency of long transactions
- Switchover performed smoothly
- Very satisfying performance (next slide)

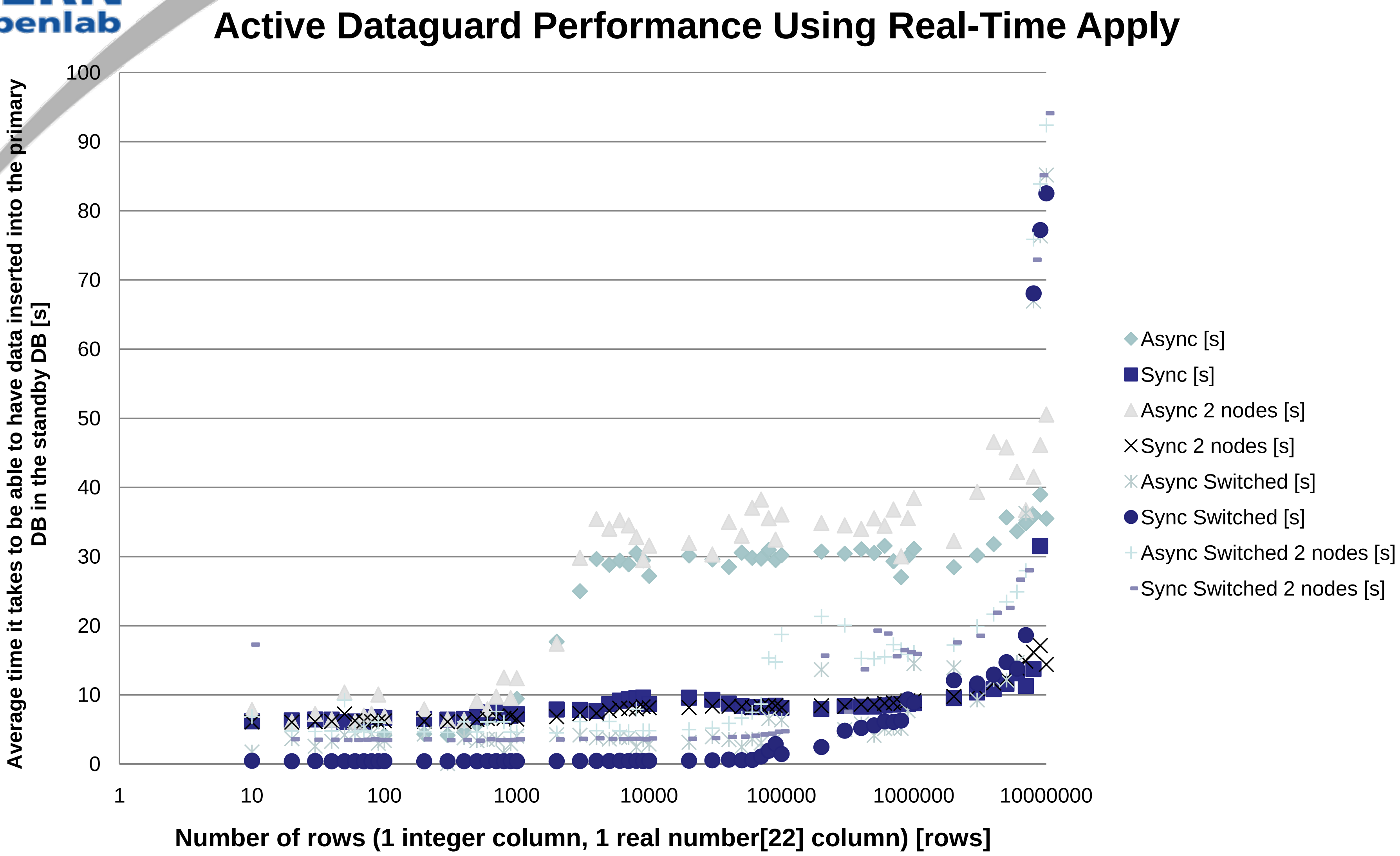
Active Data Guard Standby Performance

- Measuring algorithm:
- Loop X times over:
- Insert N data rows into the primary DB
- Measure the time it takes that any row appears in the standby DB
- Repeat the above for:
  - Varying the number of inserted rows in one transaction from 10 to 10e+7
  - 1 and 2 nodes of physical standby RAC active
  - Synchronous and asynchronous REDO transport
- Repeat all the above after a switchover

Active Data Guard Standby Performance



Active Data Guard Standby Performance



Active Data Guard Test Results

- 1 node standby RAC is slightly more performing then a 2 node one
- Synchronous REDO transport of course outperforms the async one in these tests
- Truncate table with a subsequent query on the table on standby gives ORA-08103 (Service Request assigned to development)
- Confirmed that the standby DB could be used for read-only at all times
- Verified the long term stability
- Performed a quick and smooth switchover

Conclusions and Future Work on ADG

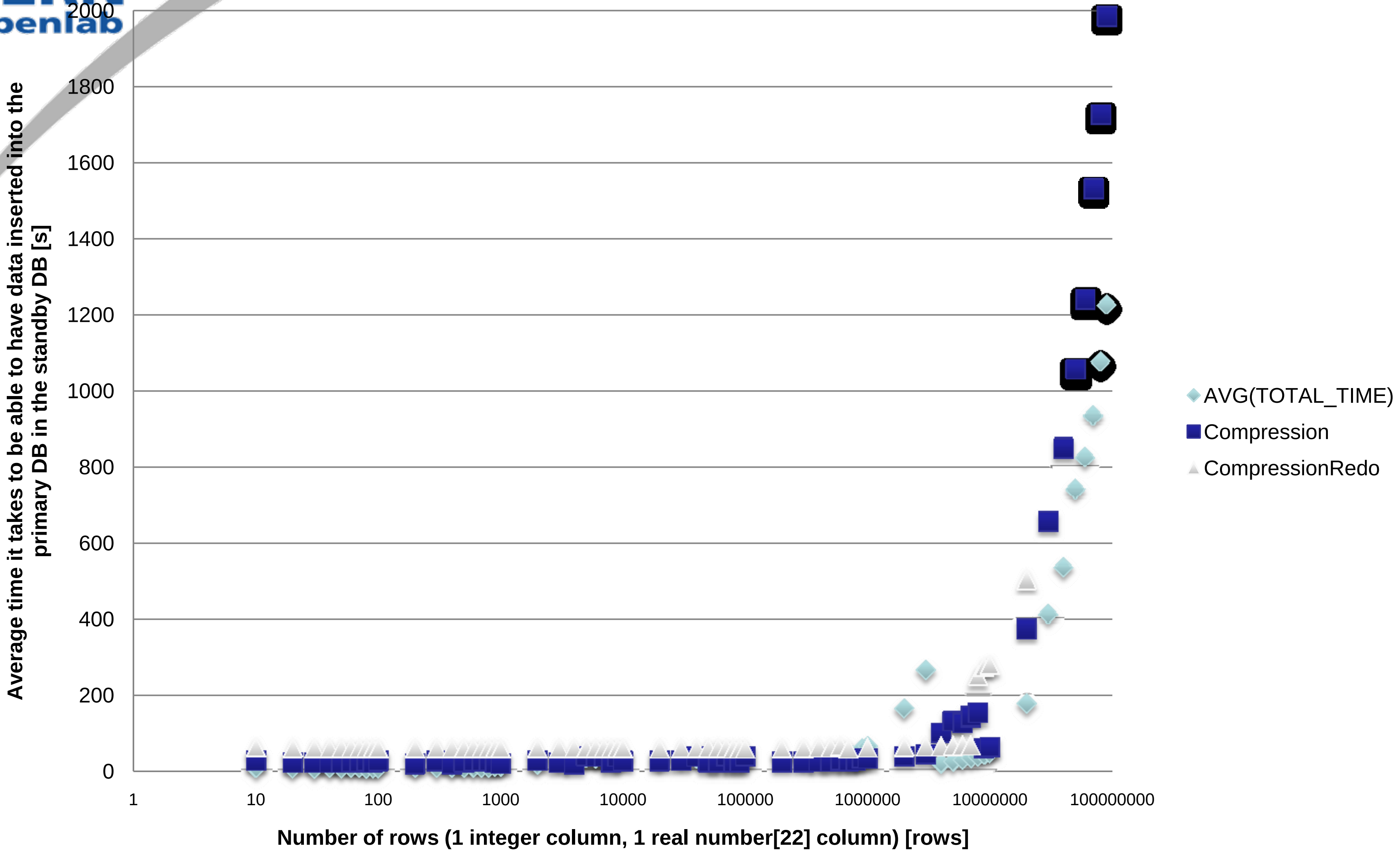
- Active Data Guard is a very promising technology
- LHC Experiments looking forward to use it

- Repeat tests with 11gR2, also using Data Guard Broker and Fail Over tests

Compression Tests

- Repeated the same tests for:
  - No table compression
  - 11gR1 compression (compress for all operations)
  - 11gR1 compression + REDO log compression
- 11gR1 compression factor for PVSS test data measured to be ~2.3

Preliminary Compression Test Results



Conclusions and Future Work on Compression Tests

- Use of compression for REDO not recommended on LANs.
- Eager to test 11gR2 columnar compression on large PVSS datasets (up to 40X compression ratio)

```
sys@BETA11G> select table_name, blocks, num_rows from dba_tables where owner like '%TEST%' order by blocks desc;

TABLE_NAME                         BLOCKS     NUM_ROWS
------------------------------ -------------- --------------
EVENTHISTORY_00000004_NOCOMPR       6220        643187
EVENTHISTORY_00000004_11GR1         1757        643187
EVENTHISTORY_00000004_10GR2         1578        643187
EVENTHISTORY_00000004_11GR2          154        643187
```

- Will be highly useful since the LHC Experiments would like to have preferably most or even all data online/read-only

ACFS (ASM Clustered File System) Tests

- Tests conducted using
  - local disks (RAID 1 - 500GB SATA)
  - SAN storage (3 storages over 4GBit FC - dual channel with multipathing - 16 SATA 400GB disks).
- 2 node 11gR2 beta RAC, 8x 16GB RAM
- Test scenarios were the following:
  - - creating 11GB zeroed file
  - - deleting the file
  - - repeating last two tests from 2 nodes of the cluster in parallel
  - - extracting 2,5GB tar file (Oracle Home) into the same location
  - - deleting extracted directory tree
  - - creating 50 000 empty files (zero length)
  - - deleting previously created files

ACFS Test Results And Future Work

				ADVM – NORMAL Redundancy		
Action Performed	**UNIT**	**EXT3 local**	**EXT3 on ADVM**	**EXT2 on ADVM**	**ACFS**	**ACFS (2 parallel threads)**
Creating 11GB empty file (bs=1M)	**MB/s**	38	35	182	230	160
Creating 11GB empty file (bs=128k)	**MB/s**		37	233	225	150
Creating 11GB empty file (bs=32k)	**MB/s**		39	241	210	150
Extracting 2,5GB tar file (Oracle Home)	**s**	180	180	94	100	117
Deleting home directory tree	**s**	2	3	1	11	12
Touching 50k files	**s**	63	63	90	75	
Deleting them	**s**	1	1	1	18	
Size on disk (du -ks)	**kB**	1412	1380	988	3276	

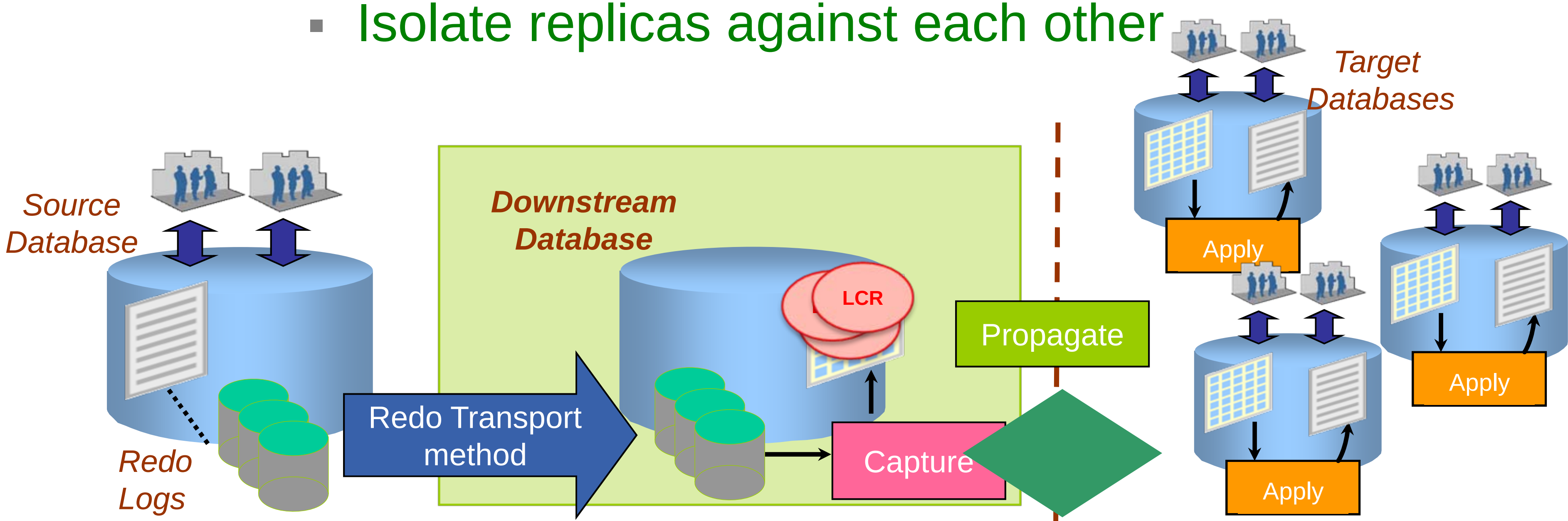
- ACFS much faster than ext3 while comparable or less CPU usage
- ACFS is slower during file deletion for both empty and bigger files
- Will test RMAN backup, large file (~TB) creation, random and sequential read

Update on Streams Activities

Automatic Split and Merge

- Motivation:
  - When one of the target databases is down:
    - LCRs are not removed from the queue
    - Capture process might be paused by flow control

    ➔ impact on replication performance
- Objective:
  - Isolate replicas against each other



Automatic Split and Merge

- Split one or more target databases from the main Streams setup
  - drop propagation to unavailable target
    - ORA-600 [KWQBMCRCPTS101] error when dropping propagation fixed by Oracle
      - Patches: 7263055 and 7480651
    - spilled messages are removed from the source queue
  - scheduled downtime
    - new streams setup (queue, capture and propagation) is created in parallel to the main setup
  - unscheduled downtime
    - execute resynchronize once the site is up again

Automatic Split and Merge

```
SQL> exec split ('STREAMS_PROP_STREVA_STRMTEST’,
                 'STREAMS_CAP_TEMP’, 'STRM_QUEUE_TEMP', 'STREAMS_PROP_TEMP');

Original Capture: STRMADMIN_CAPTURE_STREVA
Original Queue: STREAMS_QUEUE_STREVA_CA Primary Inst: 1 Secondary Inst: 2
Source database name: D3R.CERN.CH
Capture Rule Set name: RULESET$_18
Propagation Rule Set name:
Destination queue name: STREAMS_QUEUE_STREVA_AP
Destination db link: STRMTEST.CERN.CH
Destination is down - execute resynchronize_site manually later
exec
   resynchronize_site('STRMTEST.CERN.CH','STREAMS_CAP_TEMP','STRM_QUEUE_TEMP
   ',1,2,'STREAMS_PROP_TEMP','STREAMS_QUEUE_STREVA_AP','RULESET$_18','');
Stopping original capture....
Original capture process STRMADMIN_CAPTURE_STREVA successfully stopped
Dropping original propagation....
Original propagation job STREAMS_PROP_STREVA_STRMTEST successfully dropped
Starting original capture....
Original capture process STRMADMIN_CAPTURE_STREVA successfully started

PL/SQL procedure successfully completed
```

Automatic Split and Merge

```
SQL> exec resynchronize_site('STRMTEST.CERN.CH',
         'STREAMS_CAP_TEMP', 'STRM_QUEUE_TEMP',1,2, 'STREAMS_PROP_TEMP',
         'STREAMS_QUEUE_STREVA_AP',

Start scn: 6049612857832
First scn: 6049546987123 Log name:
+D3R_RECODG1/d3r/archivelog/2009_03_09/threa
Creating clone queue....
Queue STRM_QUEUE_TEMP(TB_STRM_QUEUE_TEMP) has been successfully created
Creating clone propagation....
Propagation job STREAMS_PROP_TEMP to destination STRMTEST.CERN.CH has been
successfully created
Creating clone capture....
Capture process STREAMS_CAP_TEMP has been successfully created
Capture process STREAMS_CAP_TEMP is NOT started
ALTER DATABASE REGISTER OR REPLACE LOGICAL LOGFILE
'+D3R_RECODG1/d3r/archivelog/2009_03_09/thread_2_seq_6811.305.681065869'
FOR 'STREAMS_CAP_TEMP’

PL/SQL procedure successfully completed.
```

last applied message number
@target database

identify the appropriate streams
dictionary for the given start scn

register the archived log file which
contains the dictionary with the
clone capture process

Automatic Split and Merge

- Merge both setups in one
  - capture process might start in a old archived log file

```
SQL> exec merge('STRMADMIN_CAPTURE_STREVA','STREAMS_CAP_TEMP',
                'STREAMS_PROP_STREVA_STRMTEST','STREAMS_PROP_TEMP');

Stopping original capture....
Original capture process STRMADMIN_CAPTURE_STREVA successfully stopped
Stopping clone capture....
Clone capture process STREAMS_CAP_TEMP successfully stopped
Stopping clone propagation....
Clone propagation job STREAMS_PROP_TEMP successfully stopped
Propagation job STREAMS_PROP_STREVA_STRMTEST to destination STRMTEST.CERN.CH
has been successfully added
Starting original capture....
Original capture process STRMADMIN_CAPTURE_STREVA successfully started
Merge procedure has finished successfully. Please clean temporary processes
and queues!

PL/SQL procedure successfully completed.
```

using the minimum required checkpoint scn between the 2 capture processes

Automatic Split and Merge

- Benefits
  - the fix for the propagation problem simplifies the work
    - before it was needed to re-create all the streams components
  - “manual” intervention is avoided
    - easy to make mistakes
  - the procedures have been extended to all the database administrators in the section

Thank you to Eva, Dawid, Luca, Jacek and Maria!

Thank you for your attention!

Any questions or comments?