
CERN

Eric Grancher
Katarzyna Dziedziniewicz-Wojcik

Efficient Database Cloning with Direct NFS and CloneDB

Oracle Open World 2012

Agenda

- About **CERN**
- Some theory
  - NSF vs dNFS
- CloneDB tests with RAT
- Demos
  - Dynamic cloning script
  - 10 clones
- Cloning experiences and rationale
- Summary

CERN

- European Organization for Nuclear Research
  - World's largest centre for scientific research, founded in 1954
  - Research: Seeking and finding answers to questions about the Universe
  - Technology, International collaboration, Education

Twenty Member States

Austria, Belgium, Bulgaria, Czech Republic, Denmark, Finland, France, Germany, Greece, Italy, Hungary, Netherlands, Norway, Poland, Portugal, Slovakia, Spain, Sweden, Switzerland, United Kingdom

Seven Observer States

European Commission, USA, Russian Federation, India, Japan, Turkey, UNESCO

Associate Member States

Israel, Serbia

Candidate State

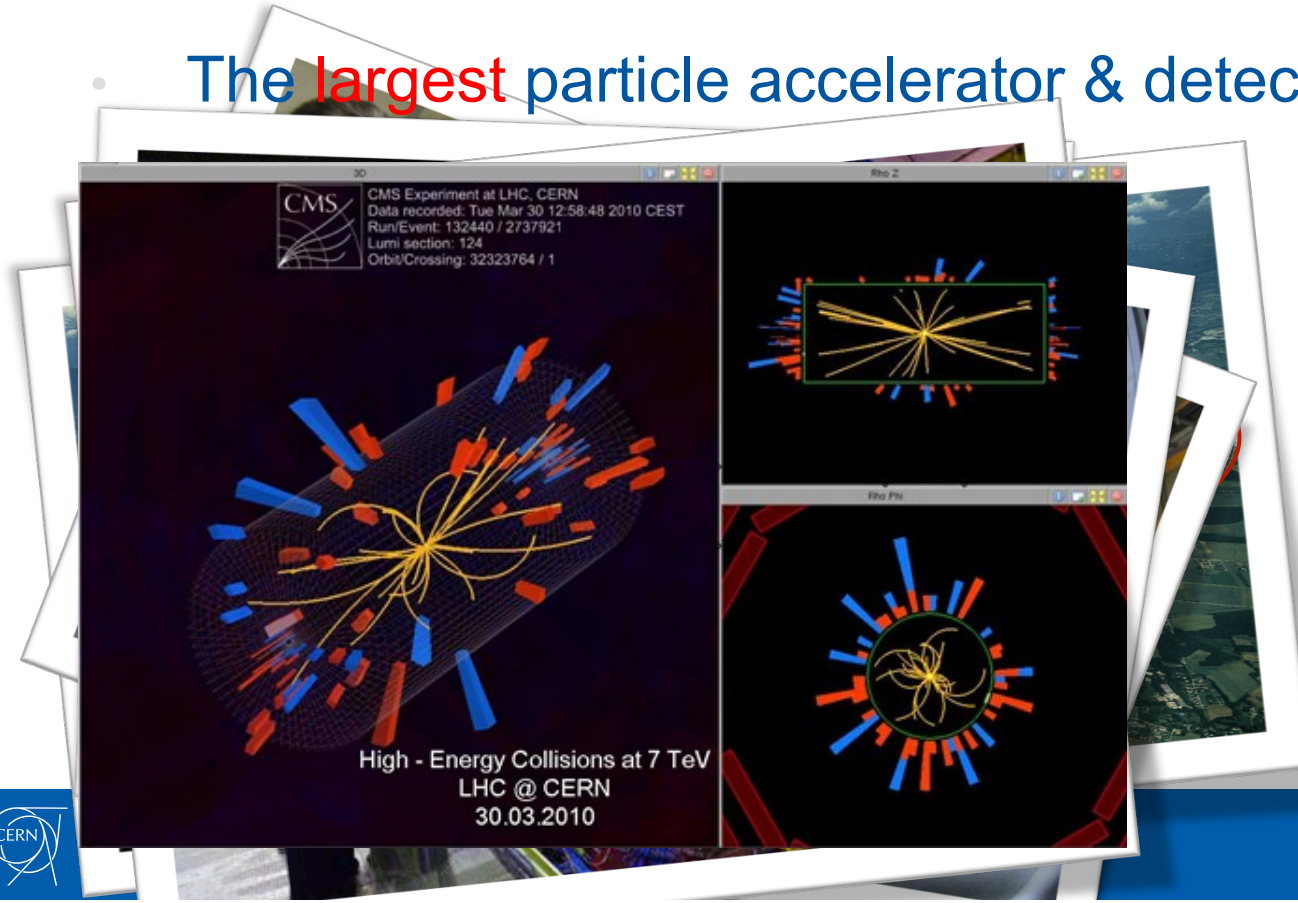
Romania

People

~2400 Staff, ~900 Students, post-docs and undergraduates, ~9000 Users, ~2000 Contractors

LHC

- The largest particle accelerator & detectors



17 miles (27km) long tunnel

Thousands of superconducting magnets

Coldest place in the Universe: *-271° C*

Ultra vacuum: 10x emptier than on the Moon

600 million collisions per second / analysis is like finding a needle in 20 million haystacks

World's largest [illegible]mputing grid - WL[illegible]

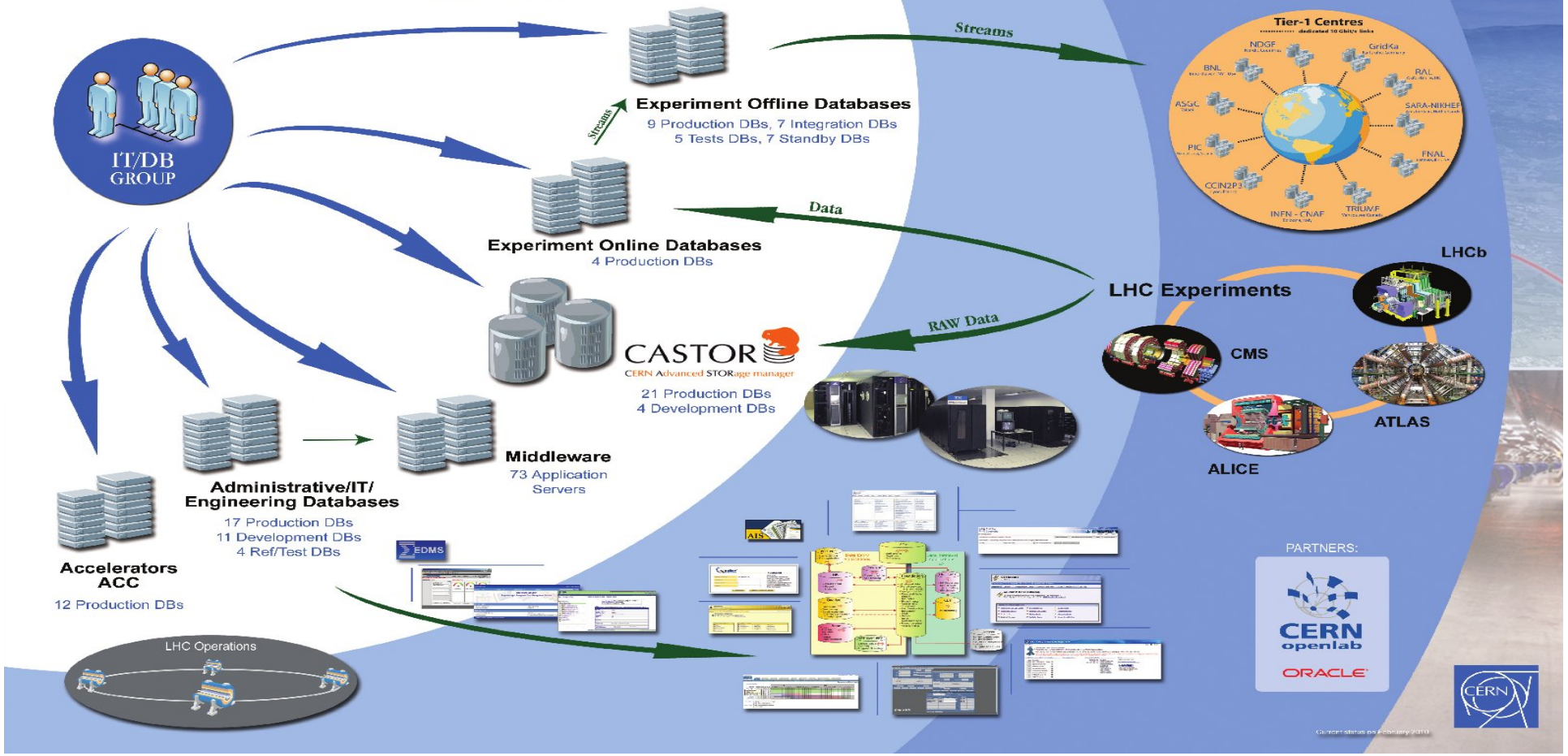
- 1PB raw data per second / >20PB of new data annually
- 68,889 physical CPUs / 305,935 logical CPUS
- 157 computer centres around the world / >8000 physicists

Database SERVICES

At the heart of CERN, LHC and Experiment Operations

CERN IT Department

http://cern.ch/it-dep/db/



CERN databases in numbers

- CERN databases services
  - ~130 databases, most of them database clusters
    Currently over 3000 disk spindles providing more than ~3PB raw disk space (NAS and SAN)
  - MySQL OnDemand Service
- Some notable databases at CERN
  - Experiments' databases – 14 production databases
    - Currently between 2 and 17 TB in size
    - Expected growth between 1 and 10 TB / year
  - LHC accelerator logging database (ACCLOG)
    - 145 TB, expected growth up to 70TB / year
  - ... Several more DBs in the 1-2 TB range

(kernel) Network File System

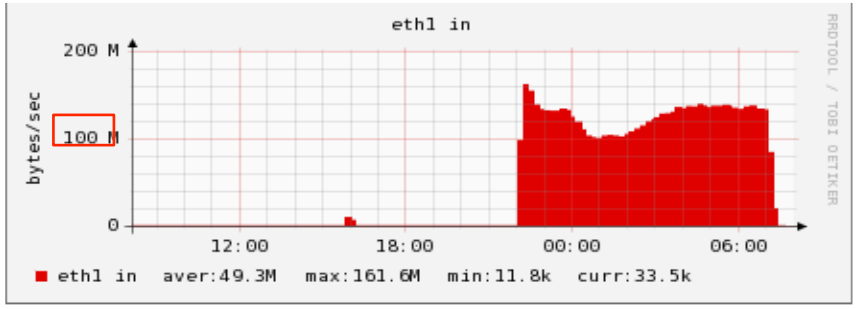
- Distributed file system protocol developed in 1984
- Based on TCP/IP
- Advantages
  - Centrally managed
  - Access granularity
  - Uses Ethernet (fast and less costly than Fiber Channel)
- Disadvantages
  - Security
  - Less performance than …

Direct NFS

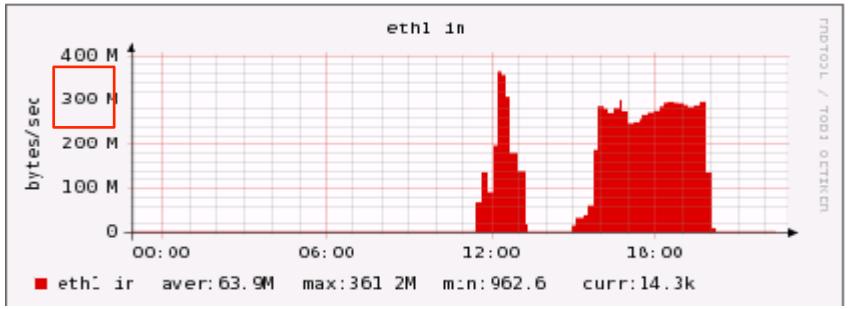
- NFS integrated into Oracle RDBMS
- Eases administration – uniform across platforms
- I/O performance optimized for DB
- Transparent port scaling

NFS vs dNFS

- Performance: dNFS vs Kernel NFS
  - RMAN on-disk backups (kernel NFS ID 1117597.1)
  - single channel backup validate



~140MB/s



~300MB/s

CloneDB

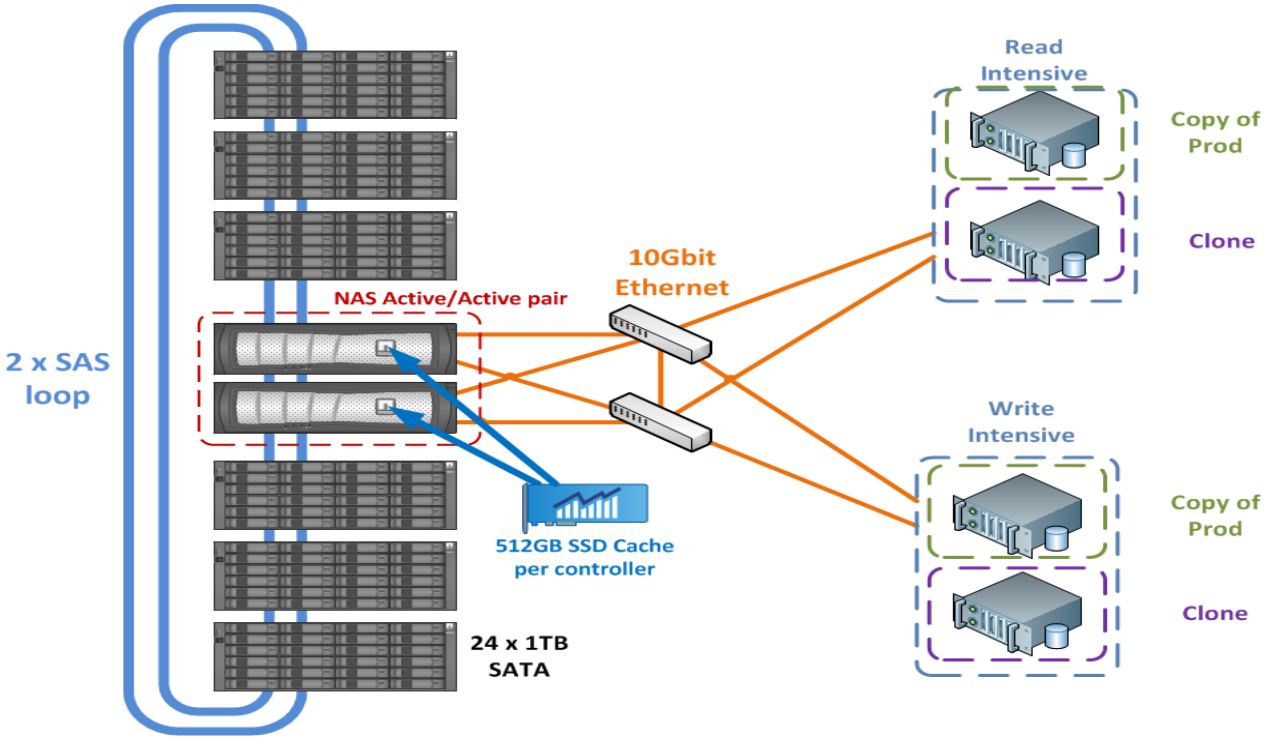
- Thin provisioned database
- Based on existing backup copy and sparse files
  - Copy-on-Write technology
    - Sparse files contain changed blocks only
    - Saves space
- One backup sufficient to provide multiple clones
  - Saves space and time
- In 12c it will be possible to clone a pluggable database

RAT and CloneDB - Rationale

- CloneDB was not designed for performance testing, but… we were curious anyway
- RAT can also be used for:
  - For platform change (hardware / OS change)
  - For database parameters change (sga_max_size, compatible, etc.)
  - For application schema change (new/removed index, etc.)
  - For execution plan changes (outline/SQL Plan Management…)
- CloneDB offers a possibility to build test environments easily

… so why not use them together

Test environment



Test environment

- Nodes
  - Physical machines
  - Dual Quadcore Intel Xeon 2.27 HGz
  - 48 GB RAM
  - RHEL 5.8
- Storage
  - NetApp FAS3240
- Databases
  - Both storing metadata for mass storage system CASTOR
  - Namespace database – read intensive load
  - Stager database – write intensive load

What was done…

- 2 Production DBs with different type of workload
- Physical Standby DB recovery stopped
  - Backed up using RMAN from standby
    - Possible with most storage snapshot technologies
  - 8 hour RAT capture started on production

How to…

- Set environment:

```
$ export ORACLE_SID=DOLLYNS
$ export MASTER_COPY_DIR=/ORA/dbs05/BACKUP/backupNS
$ export CLONE_FILE_CREATE_DEST=/ORA/dbs03/DOLLY/dollyNS
$ export CLONEDB_NAME=DOLLYNS
```

- Prepare pfile: initDOLLYNS.ora
  - parameter cloneDB must be set to TRUE
- Prepare DB creation scripts – Clone.pl from ..

```
$ perl Clone.pl initDOLLYNS.ora createDBout.sql
renameOut.sql
```

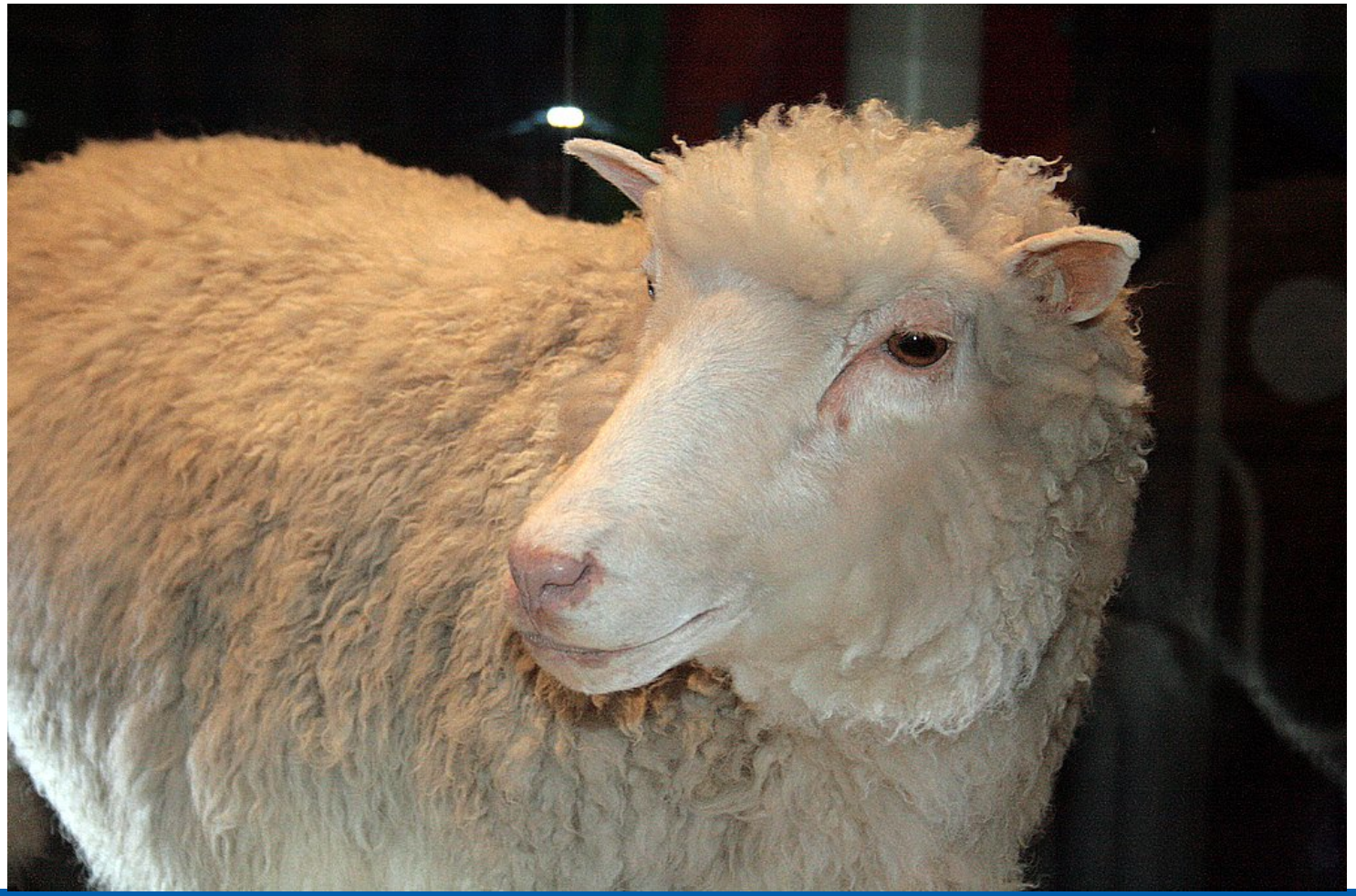
- Enable dNFS for `CLONE_FILE_CREATE_DEST`

Waking Dolly up

- Correct output files
  - In DB creation script check proper backup file naming and DB creation parameters (pe. MAXDATAFILES)
  - In the rename script:
    - check proper naming of both backup and sparse files
    - Replace ‘DROP TEMPORARY TABLESPACE’ clause with ‘ADD TEMPORARY FILE’
- Create clone

```
$ sqlplus / as sysdba
$ @createDBout.sql
$ @renameOut.sql
```

  - In case temporary tablespace operation fails with ORA-25153
  - `ALTER SYSTEM SET "_system_trig_enabled" = FALSE SCOPE=memory;` and repeat operation



Dolly pic by Toni Barros

Short RAT intro - capture

- Capturing workload

```
SQL> create directory capture_dir as '/tmp/Captured/';
SQL> grant write on directory capture_dir to public;
SQL> exec
dbms_workload_capture.start_capture(name=>'capture1', dir=>
'REPLAY_DIR');
SQL> exec dbms_workload_capture.finish_capture();
```

Short RAT intro - replay

- Replaying workload

```
SQL> create directory replay_dir as '/tmp/Captured/';
SQL> grant read on directory replay_dir to public;
SQL> exec dbms_workload_replay.process_capture('REPLAY_DIR');
SQL> exec DBMS_WORKLOAD_REPLAY.initialize_replay('replayName',
'REPLAY_DIR');
SQL> exec DBMS_WORKLOAD_REPLAY.prepare_play;
```

- Start replay clients

```
$ORACLE_HOME/bin/wrc system/manager1 replaydir=/tmp/Capture
```

- Start replay

```
SQL> exec DBMS_WORKLOAD_REPLAY.start_replay;
```

Test results

- Replay effects with respect to load type

DB Type	Filesize before rep (GB)	Filesize after rep (GB)	Sparsefile size after RAT(GB)	Replay time
Stager	119.5	120	-	6h 1m
Stager Clone	-	-	18	6h 8m

Load profile comparison

- As expected load looks similar but…

	Primary	Clone DB
Redo size/transaction	1700	2200
Logical reads/transaction	800	900
Block changes/transaction	12	14
Physical reads	13	10
Physical writes	0.5	1

Primary vs Clone

- Different wait events for Clone
  - SQL execution times differ because of IO
  - Read IO request rate much higher for Clone
  - Up to 7 times for Direct read

DB	Event	Waits	Time(s)	Wait(s)	% DB time
Clone	Disk file operations I/O	3,553,493	24,360	7	41.2
Clone	db file sequential read	398,760	2,426	6	4.1
Primary	db file sequential	411,305	6,850	17	16.6
Primary	db file scattered read	401,405	4,234	11	10.3
Primary	direct path read	124,566	2,091	17	5.1

Primary vs Clone

- Top SQL comparison

DB	SQL ID	Time per Exec(s)	%CPU	%IO
Clone	ck0sr7gtnadg3	2.04	15.3	88.4
Primary	ck0sr7gtnadg3	0.55	45.5	35.5
Clone	7r91kghk5cv5g	24.44	9.4	94.9
Primary	7r91kghk5cv5g	4.51	35.9	40.4

Test conclusions

- There seems to be different IO pattern for clones
- Tests for given workload should be repeated
- Performance tests make sense when two clones are used
- Oracle doesn’t recommend it!

Steps, MOS note 1210656.1

1. Have/create a reference copy (storage snapshot, RMAN image copy, etc.)
2. Create control file
   - alter database backup controlfile to trace;
   - Modify datafile location to reference copy
3. Change location
   - dbms_dnfs.clonedb_renamefile(backup_file_name, new_data_filename)
4. Recover database and alter database open resetlogs

Example 1 (1/2)

- drop tablespace data01 including contents and datafiles;
- create tablespace data01;
- create table grancher.t (a1 number, a2 varchar2(100));
- insert into grancher.t select rownum,rpad('-', 10,rownum) from dba_objects where rownum<10;
- commit;

Example 1 (2/2)

- Look at the DATA01 datafile
- Clone
- Look at the cloned DATA01 datafile
- Update the cloned database
- Look at the cloned DATA01 datafile

CloneDB memory

- You should take into account a 2MB shared pool

```
SQL> select name,bytes from v$sgastat
where pool ='shared pool' and name not
in (select name from v$sgastat@db112ol6
where pool ='shared pool');
NAME                                 BYTES
---------------------------   ----------
ksfdss bitmap array                2114048
```

Bitmap file

- Can be found in ${ORACLE_HOME}/dbs/${ORACLE_SID}_bitmap.dbf
- $ ls -lk ${ORACLE_HOME}/dbs/${ORACLE_SID}_bitmap.dbf
- -rw-r----- 1 oracle ci 2056 Sep 7 03:40 /ORA/dbs01/oracle/home/app/oracle/product/11.2.0/dbhome_1/dbs/C112OL6_bitmap.dbf

Modified Clone.pl script

- Connects to the reference database
  - Retrieves the list of datafiles
  - Retrieves the character set
- Reads spfile for the parameters, adds clonedb=true, filters more (and case)
- Adds the tempfiles (parameter from source DB)
- Does not write on the backup copy dir (RO snapshot)

Example 2

- 10TB database
- Automation for the cloning
- Launch creation of 10 clones
- Check space in the clone directory

Integration “OnDemand”

- It can be made so that the developers get a clone “whenever needed”
- Example 3
  - giveclone
  - sqlplus …

Space usage

- CloneDB: Oracle blocks
- NAS clone (depends): example 4kB NetApp
- In all cases, only changes

Storage cloning versus CloneDB

- + less overhead
- + same spfile / redolog

Storage level cloning

- + no mount operation (root)
- + no clone license required
- + no storage admin operation required

CloneDB

Why clone?

- Space is precious (cost, power, space, management)
- Time is precious (developer, DBA, system team)

Usage

- Validate backup
- Test operation (change character set, change compatible, upgrade, platform or version change, etc.)
- Test change impact on application (LIO)
- Test application change
  - Integration with development environment

Deployment / recommendation

- With NAS snapshot, no need for a full copy
- Not on the production storage
  - At least use another host or VM (avoid path issue)
- On a snapshot (or copy of a snapshot)
  - Avoids a copy of the datafiles
- Can be used with DataGuard

Conclusion

- CloneDB is a new promising feature, performance comparison done with Real Application Testing (single instance)
- Space usage is ~similar with storage clone
- CloneDB + storage snapshot make a very good combination
- Can be integrated without root/storage admin access, “on demand” clones
- Modified Clone.pl available (dynamic)

www.cern.ch

NFS vs dNFS (by R. Gaspar)

CERN IT Department

- Inner table (3TB) where a row = a block (8k). Outer table (2% of Inner table) each row cotains rowid of inner table
- v$sysstat ‘physical reads’
  - Starts with db file sequential read but after a little while changes to db file parallel read

```
select /*+ leading(p) USE_NL(t) parallel(p 100)*/ sum(1) from testtable_3t t, probetest3t_2pct p where t.rowid=p.id;

Plan hash value: 377594698

-----------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                          | Name              | Rows  | Bytes | Cost (%CPU)| Time     |    TQ  |IN-OUT| PQ Distrib |
-----------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |                   |     1 |    22 | 80200   (1)| 00:16:03 |        |      |            |
|   1 |  SORT AGGREGATE                    |                   |     1 |    22 |            |          |        |      |            |
|   2 |   PX COORDINATOR                   |                   |       |       |            |          |        |      |            |
|   3 |    PX SEND QC (RANDOM)             | :TQ10000          |     1 |    22 |            |          | Q1,00  | P->S | QC (RAND)  |
|   4 |     SORT AGGREGATE                 |                   |     1 |    22 |            |          | Q1,00  | PCWP |            |
|   5 |      NESTED LOOPS                  |                   |  7200K|   151M| 80200   (1)| 00:16:03 | Q1,00  | PCWP |            |
|   6 |       PX BLOCK ITERATOR            |                   |       |       |            |          | Q1,00  | PCWC |            |
|   7 |        TABLE ACCESS FULL           | PROBETEST3T_2PCT  |  7200K|    68M|   178   (0)| 00:00:03 | Q1,00  | PCWP |            |
|   8 |       TABLE ACCESS BY USER ROWID   | TESTTABLE_3T      |     1 |    12 |     1   (0)| 00:00:01 | Q1,00  | PCWP |            |
-----------------------------------------------------------------------------------------------------------------------------
```

Random Read IOPS*	No PAM	PAM + Kernel NFS (RHE5)	PAM + dNFS
First run	2903	795	3827
Second run	2900	16397	37811

~160 data disks

~365 data disks

*fas3240, 32 disks SATA 2TB, Data Ontap 8.0.1