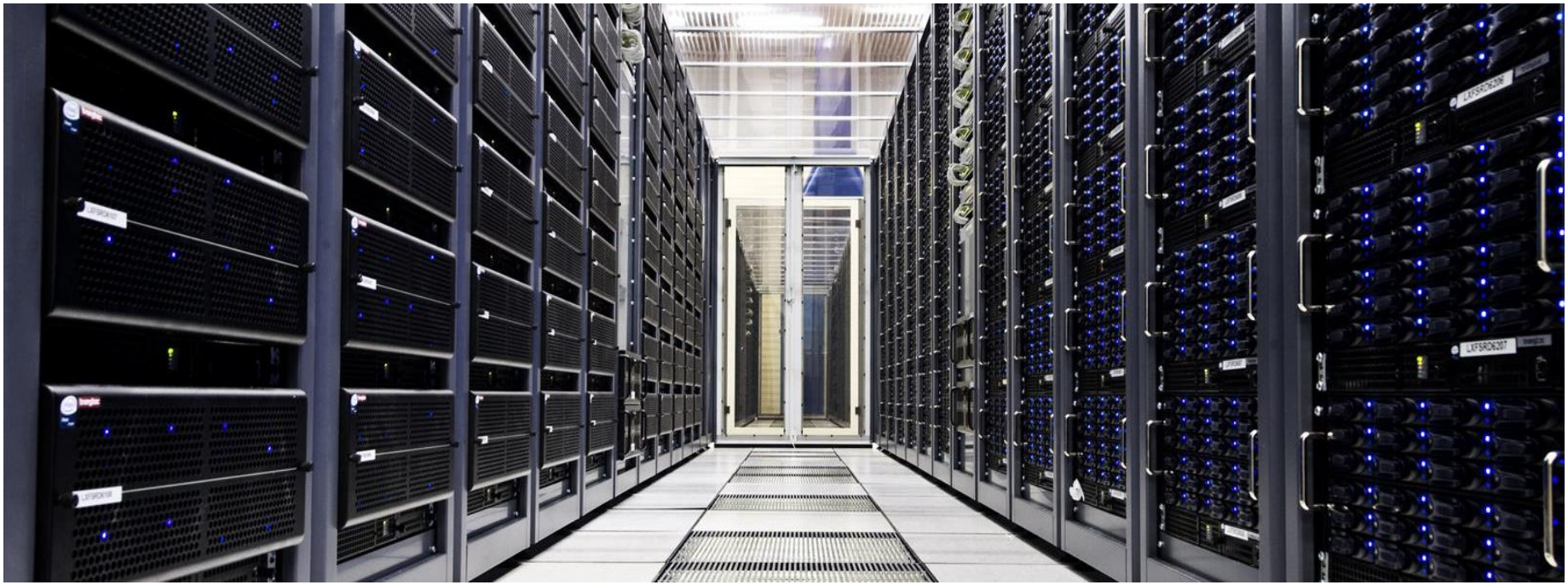
Testing Storage for Oracle RAC 11g

with NAS, ASM and DB Smart Flash Cache

Luca Canali, CERN

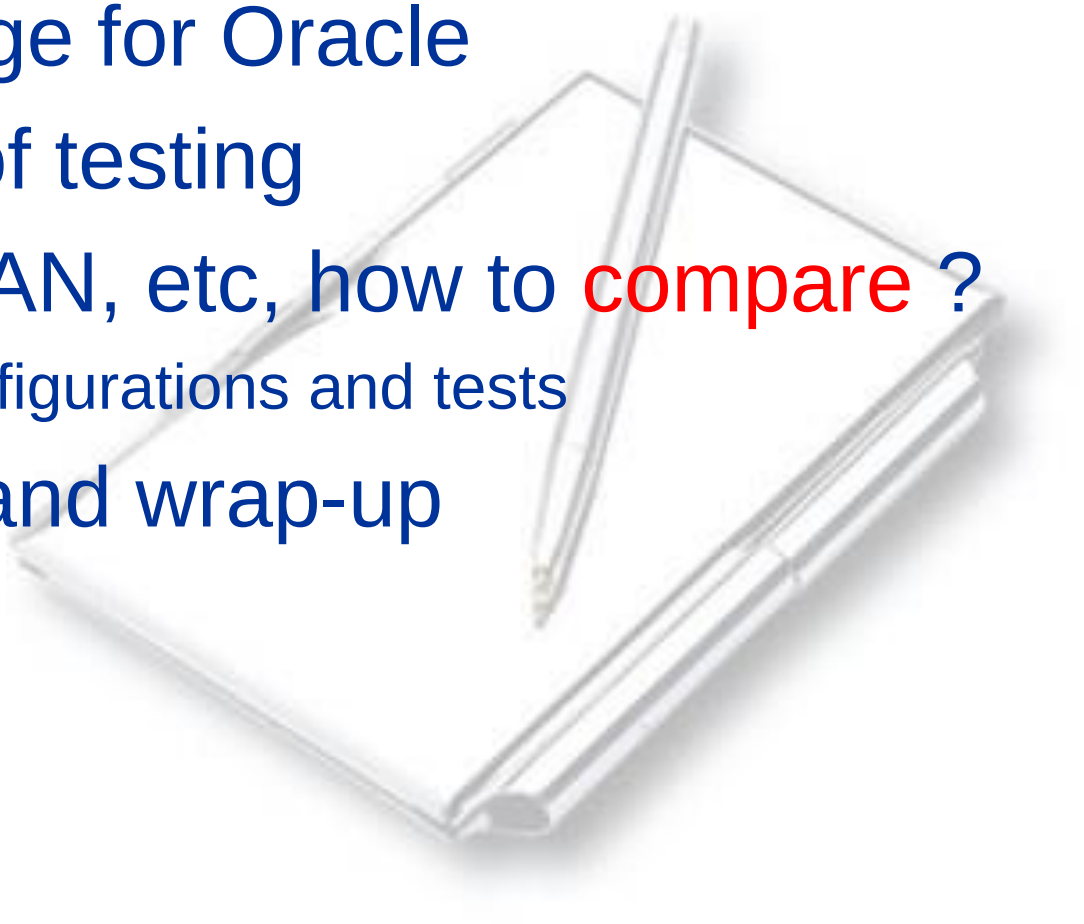
Dawid Wojcik, CERN

UKOUG Conference, Birmingham, Dec 6th,2011



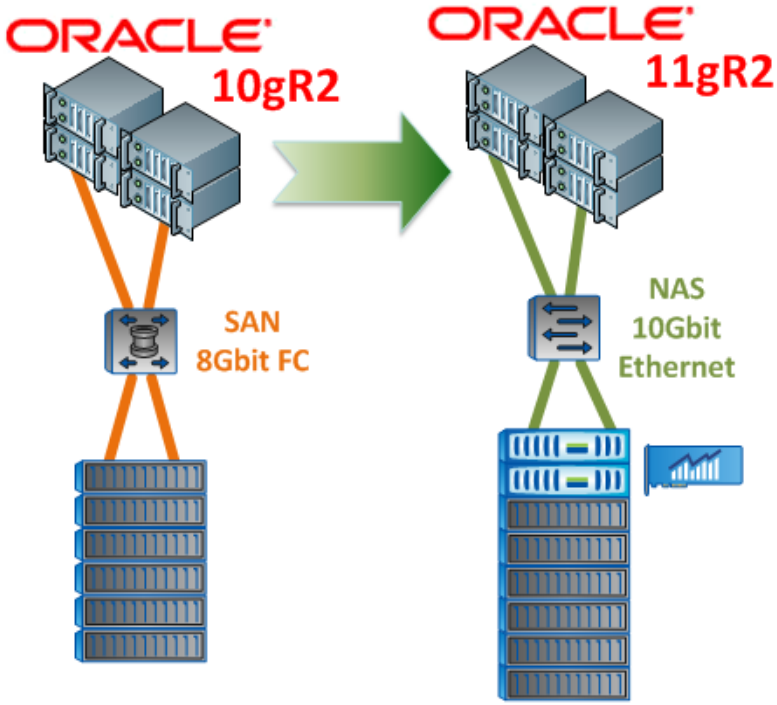
Outline

- Why testing storage for Oracle
- Define the goals of testing
- Example: NAS, SAN, etc, how to compare ?
  - Results from our configurations and tests
- Lessons learned and wrap-up



Motivations

- New HW acquisition
  - refresh cycle ~ every 3 years
  - Occasion to test and deploy new technology
- Additional input this time
  - Consolidate storage solution
  - NAS and SAN
  - Upgrade 10gR2 to 11gR2

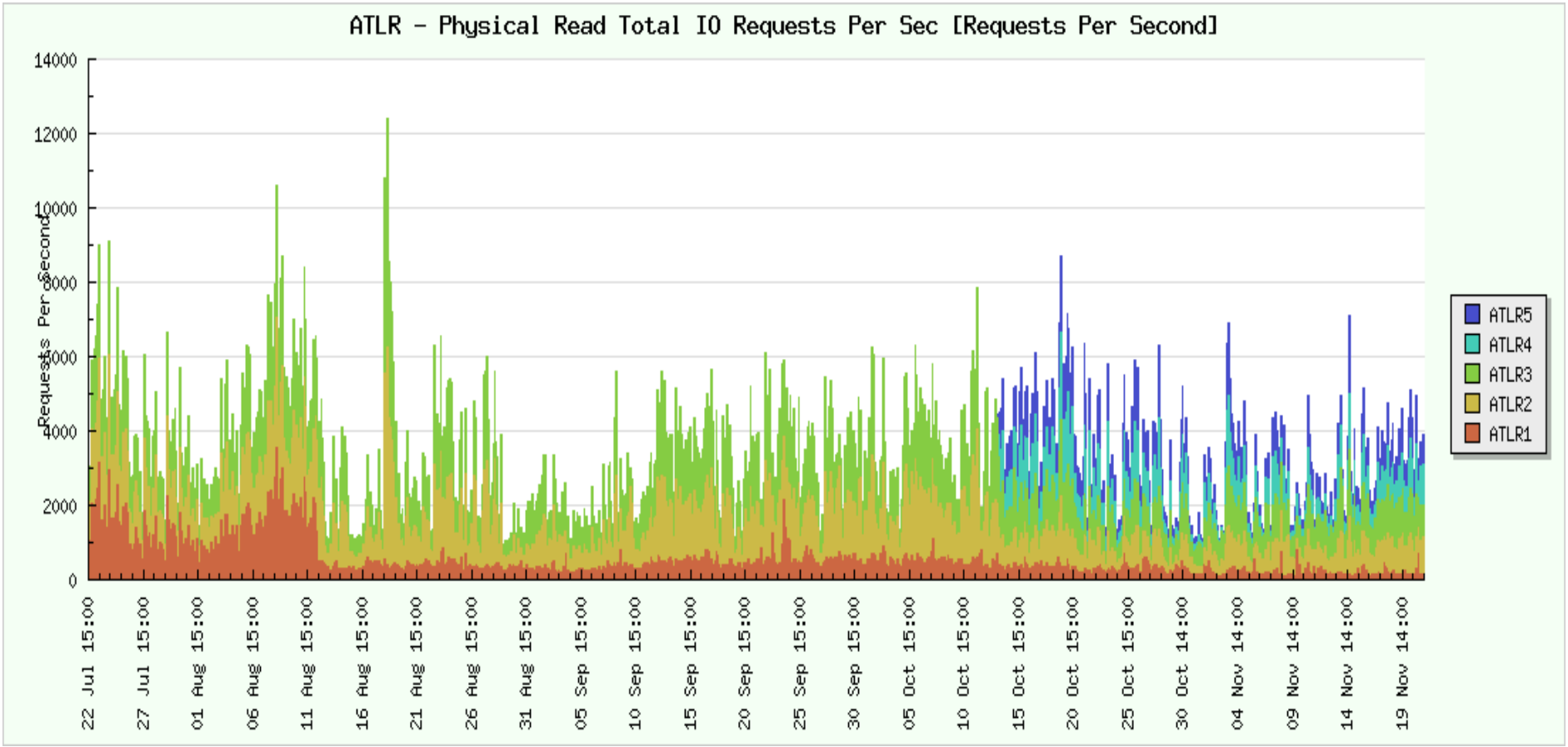


Focus on Storage

- Matching requirements from production
- What do we need (in our environment)?
- Random read IOPS most critical
  - Index-based access and nested loop joins
- Fast sequential read
  - Mostly for backup, stats gathering and some full scans
- Also critical
  - HA and management features
  - Cost and economies of scale

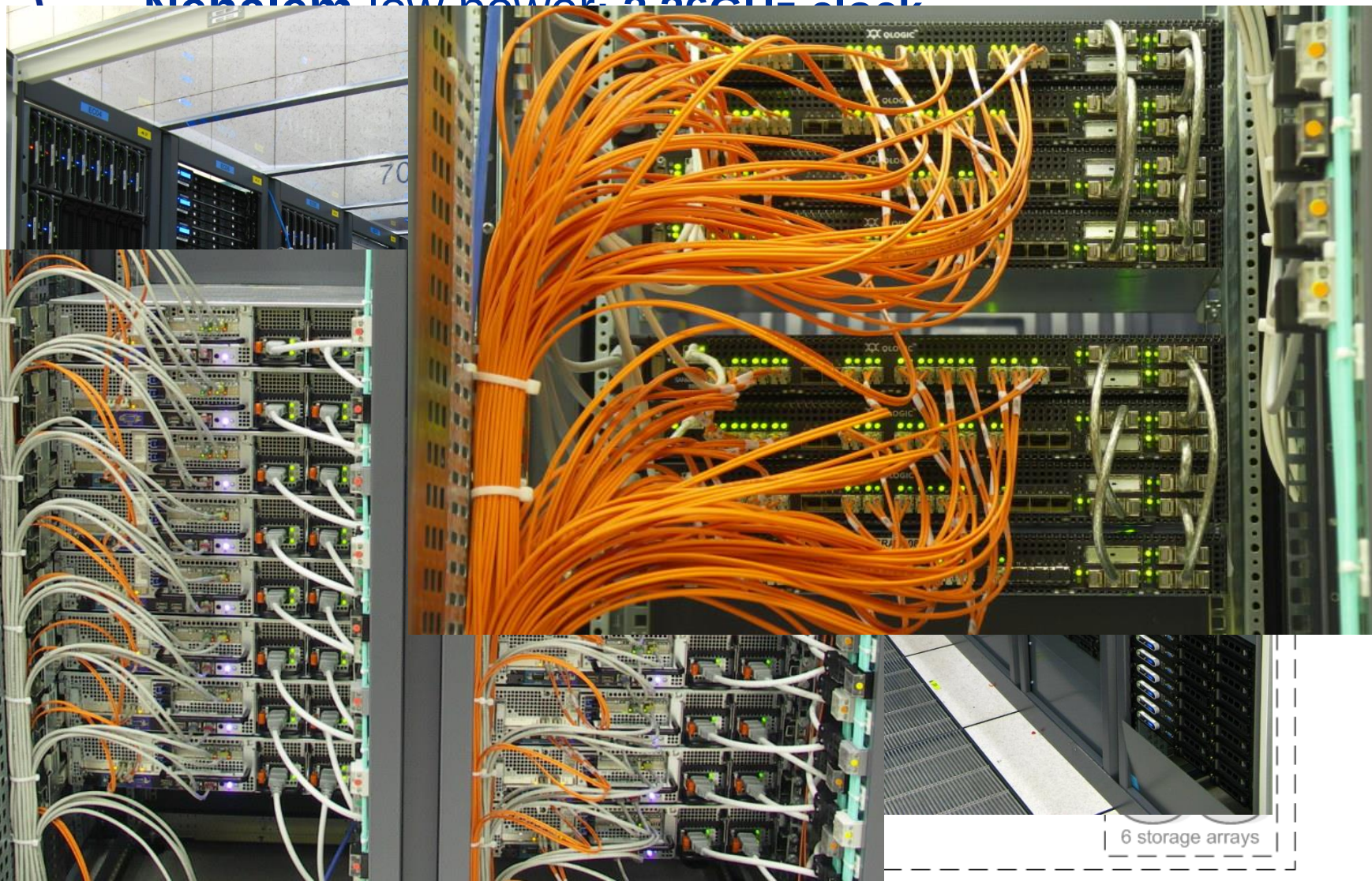
Measurements from Production

- AWR data for capacity planning and trend analysis

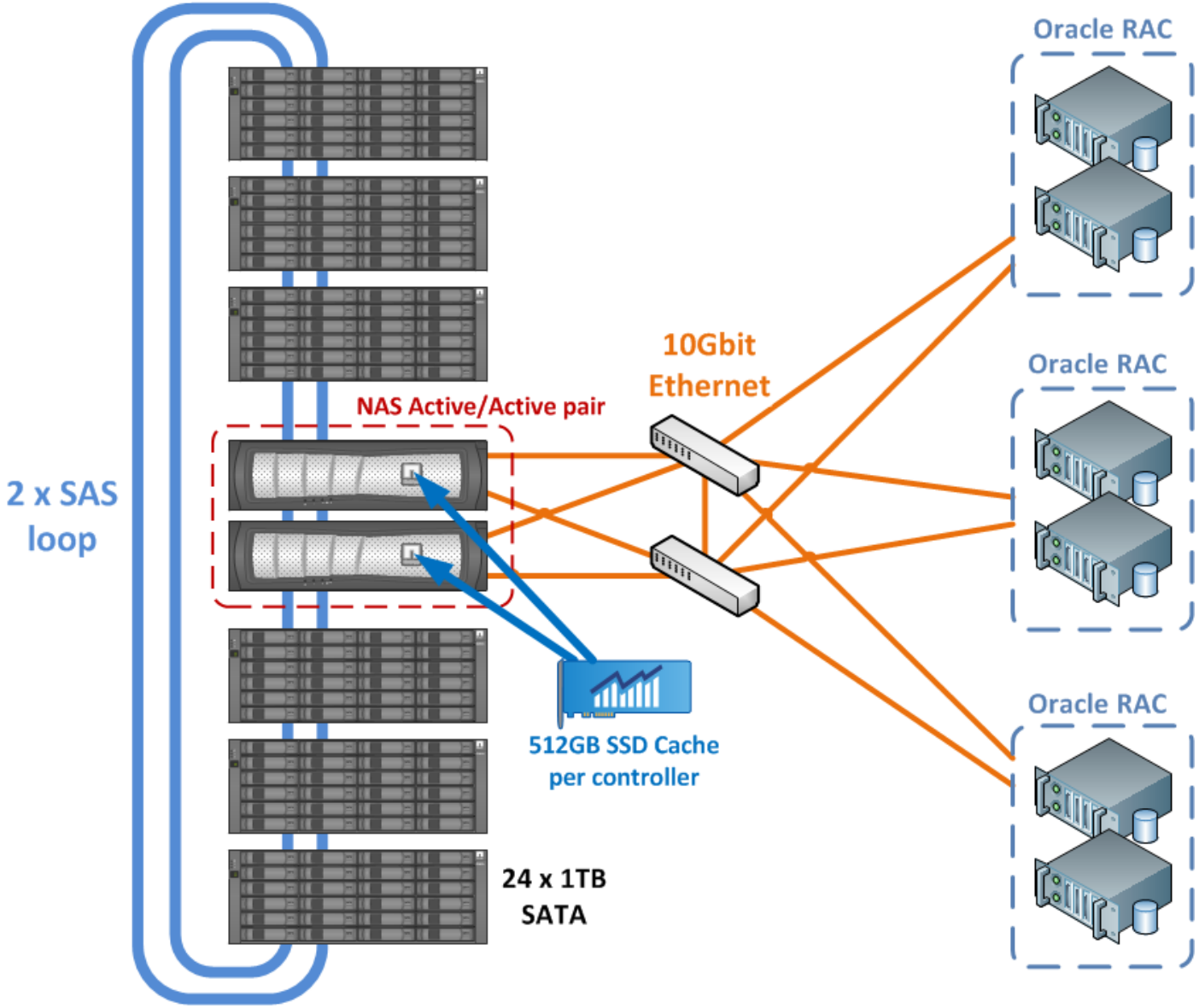


ASM Normal Redundancy

- Dual-CPU quad-core blade servers, 24GB memory, Intel Nehalem low power 2.26GHz clock



NetApp NAS


Oracle RAC
10Gbit
Ethernet
Oracle RAC
NAS Active/Active pair
2 x SAS
loop
Oracle RAC
512GB SSD Cache
per controller
24 x 1TB
SATA

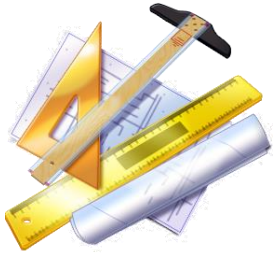
Measuring IO performance

Apples and Oranges

- Different solutions, strong and weak points
  - May be quite different
- How to compare, then?
  - Define a few configurations that we understand and make sense for us
  - Define and test the IO metrics

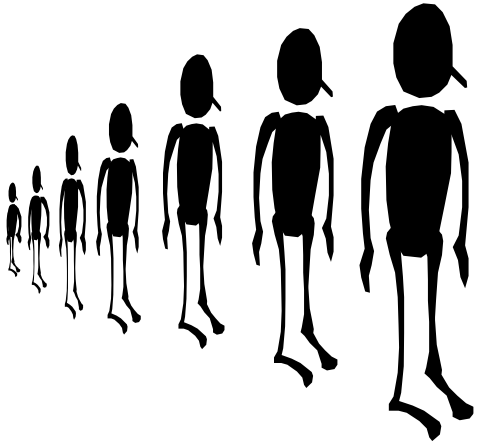
- Further tests with real application workload

Tools to Measure IO



- We want something that we can understand
  - Workload that makes sense for databases
  - Avoid caching traps (i.e. testing with little data)
- Analysis
  - Use metrics that relate to DB usage
  - Possibly allow to build a model to correlate measurements and HW architecture

Sequential Read



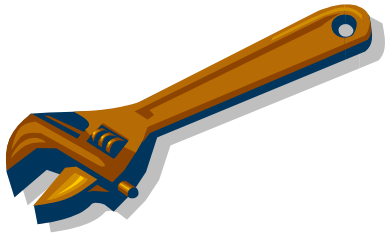
- How to test:
  - Parallel query of very large table
  - Measure throughput for example from gv$sysmetric
    - Metric: “Physical Read Total Bytes Per Sec”
    - See also SQL on slide’s comment section
- Tested SAN
  - 2 storage arrays with 12 disks each, 8+8 Gig FC
  - Throughput=1.5 GB/sec
- Tested NAS
  - 1 box with 72 disks in RAID DP, 10GigE
  - Throughput=0.7 GB/sec

Sequential Write

- How to test:
  - Multiple session running SQL for tablespace creation
  - OS tools: ORION and unix command dd
- Tested SAN
  - 2 x storage arrays with 12 disks each
  - Normal redundancy ASM (needs to write 2 copies)
  - Throughput= 700 MB/sec (limited by 8G FC)
- Tested NAS
  - 1 storage box, 72 disks
  - Throughput=300 MB/sec

Random IO and IOPS measurements

ORION



- Oracle tool, available since 10g
- Easy to use
- Used it for several years
  - good results for JBOD configs for ASM
- Output easy to understand
  - In particular for read-related metrics
- Some critique
  - Proprietary tool
  - Possible ‘cache trap’ (see also Kyle Hailey’s dtrace measurements)

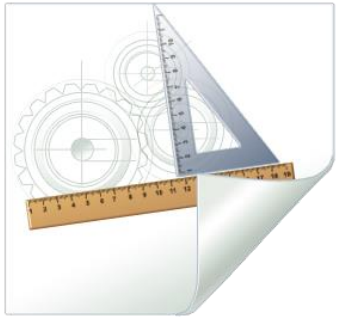
ORION - Example

- Basic IO metrics measured by ORION
  - IOPS for random I/O (8KB)
  - MBPS for sequential I/O (in chunks of 1 MB)
  - Latency associated with the IO operations
  - Since 11.2.0.2, latency histogram (helps testing SSD)

- Simple to use
  - Getting started:
    `./orion_linux_em64t -run simple -testname mytest -num_disks 24`
  - More info:
    https://twiki.cern.ch/twiki/bin/view/PDBService/OrionTests

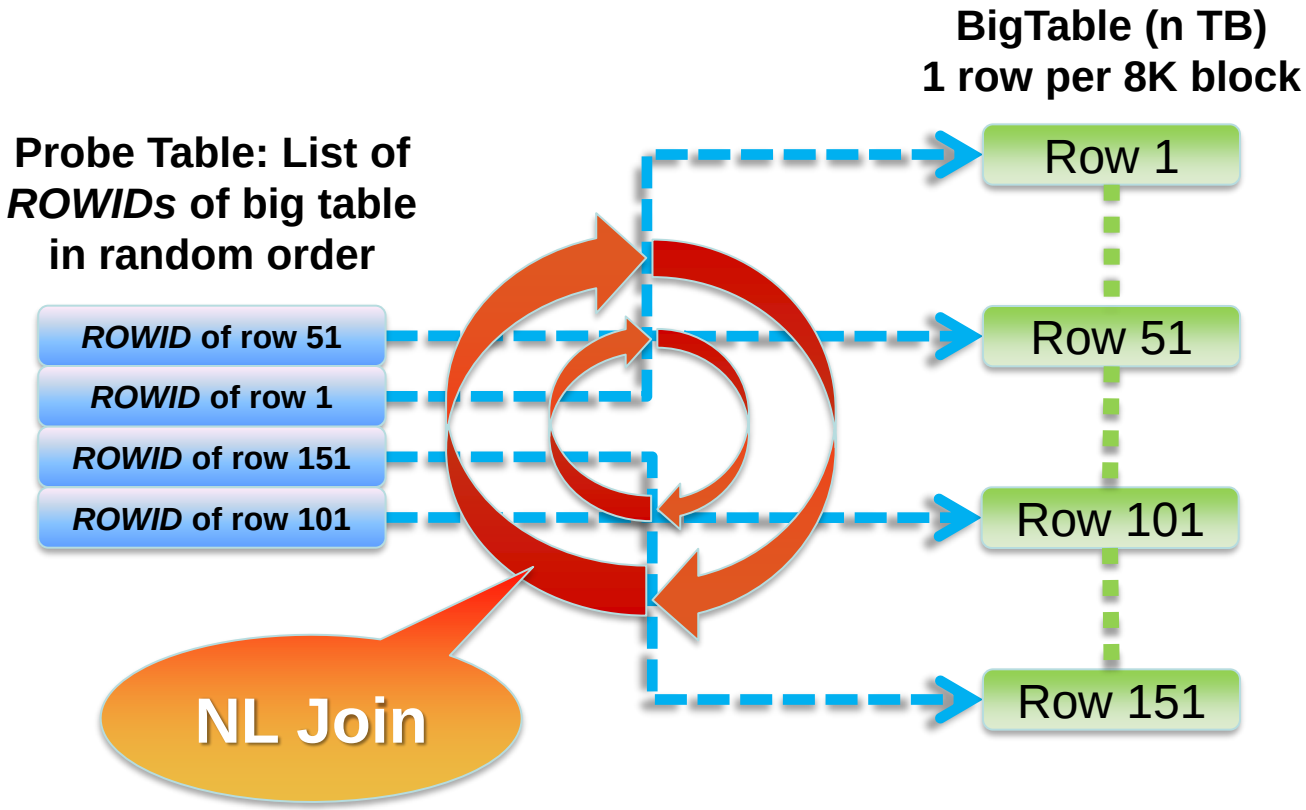
DB-Oriented Tests

- Look at your production DBs' workload
  - Read/write ratio and average IO size
  - Sequential vs. scattered access
- Create a test DB
- Define simple DB workloads
  - Test cases that match your average production workload
- Produce load on the metrics of interest
  - Random IO read
  - Sequential read
  - Sequential write



More Details on Tests

- Nested loops join with parallel query to drive load
- How to measure IOPS: from gv$sysmetric



DB Workload for Tests

- A standard way to run workload from DB would be very beneficial
- A step in this direction:
  - DBMS_RESOURCE_MANAGER.CALIBRATE_IO
  - Similar to Orion, although more difficult to interpret results
- Critique
  - From our tests IOPS seems to be overestimated
  - Sequential throughput underestimated
  - Unaware of array cache

Random reads SAN

SAN for ASM normal redundancy

- IOPS scale up with Number of disks
  - ~100 IOPS/disk for SATA,
  - ~200 IOPS/disk for SAS

- Example:
  - 24 SAS disks -> ~5000 small random read IOPS
  - Test config of 400 SATA disks: ~40K IOPS

Random Reads NAS

NAS:

- Random reads -> ~100 IOPS per disk
  - Take about 20% disks off as they are used for DP
  - Example raid group of 72 disks -> ~5000 IOPS
- Random reads served by Solid State cache
  - 512 GB PAM module
  - Up to ~33K IOPS
  - Note we find DNFS improves IOPS we can get

Solid State Disks for IOPS

IOPS-Hungry Applications

- SSD provide high IOPS and low latency
  - Great for many OLTP-like applications
- Possible usage of SSD
  - Full DB on SSD
  - Parts of the DB on SSD (e.g. critical tablespaces)
  - SSD as cache on storage controller
  - Database smart flash cache

DB Flash Cache

- Goal:
  - Cost-effective, high capacity, high read IOPS
- Problem:
  - Low-cost arrays often don't have SSD cache
- Idea:
  - Use SAN and ASM with normal redundancy with high capacity disks
  - Use DB flash cache to enhance DB buffer cache

Setup for Testing

- Supported on Solaris and OEL
  - Tip for red-hat testing: actually just need package 'enterprise-release' from OEL to replace 'redhat-release'
- HW:
  - Local SSD of 200 GB used for this test
  - Idea: low cost HW
- DB parameters
  - *.db_flash_cache_size=160g
  - inst1.db_flash_cache_file='+SSD_NODE1/flashc1.dbf'
  - inst2.db_flash_cache_file='+SSD_NODE2/flashc2.dbf'

Basic Tests

- ORION on local SSD:
  - Random small reads: 16000 IOPS
  - Latency histogram: 0.5-1ms range
  - Random small writes: 2900 IOPS
  - Sequential IO: read 240 MB/s, write 60 MB/s

- Oracle-based test
  - Measure exec time for 1M single-block cached reads
  - Time is latency bound, with SSD vs disk: 6x speedup
    - single block read from flash cache: ~0.6 ms
    - See also SQL in slide's note section

Flash Cache and SSD

- DB Flash Cache
  - (+) Can boost IOPS performance
  - (+) Can be tuned at segment level
    - SQL: storage (flash_cache keep)
  - (-) consumes CPU on DB server (DBWR)
  - (-) requires extra memory from buffer cache
  - (-) cache is local and not RAC-aware
  - (-) New feature
  - (-) runs only on some Linux distributions

Solid State Cache and NAS

- Storage controller-based solid state cache
  - (+) Easy to understand and proven
  - (+) No additional server CPU, RAM consumed
  - (+) 33K random read IOPS in tested config
  - (-) Cost can be high for large amounts of cache currently

Conclusions

- Many interesting lessons learned by testing
- New technology of high impact
  - Solid State Disks for OLTP applications
  - 10 GigE
- 11g new features of interest
  - Direct NFS and db flash cache
- Storage for Oracle (RAC)
  - Complex ecosystem, evolving fast

Acknowledgments

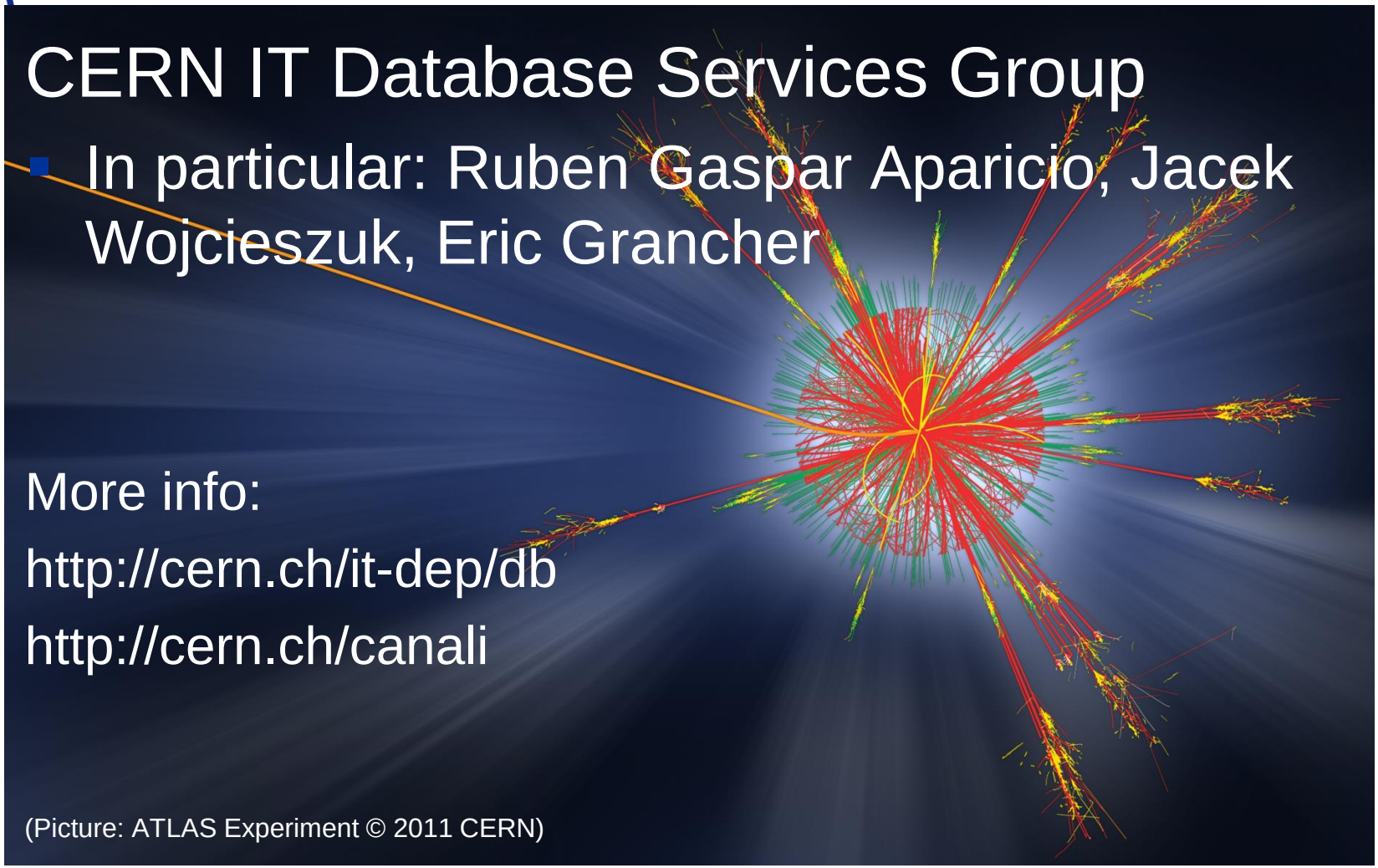
CERN IT Database Services Group

- In particular: Ruben Gaspar Aparicio, Jacek Wojcieszuk, Eric Grancher

More info:

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

http://cern.ch/canali



(Picture: ATLAS Experiment © 2011 CERN)