Data Life Cycle Management for Oracle @ CERN with partitioning, compression, and archive.

Luca Canali, CERN

Orcan Conference, Stockholm, May 2010

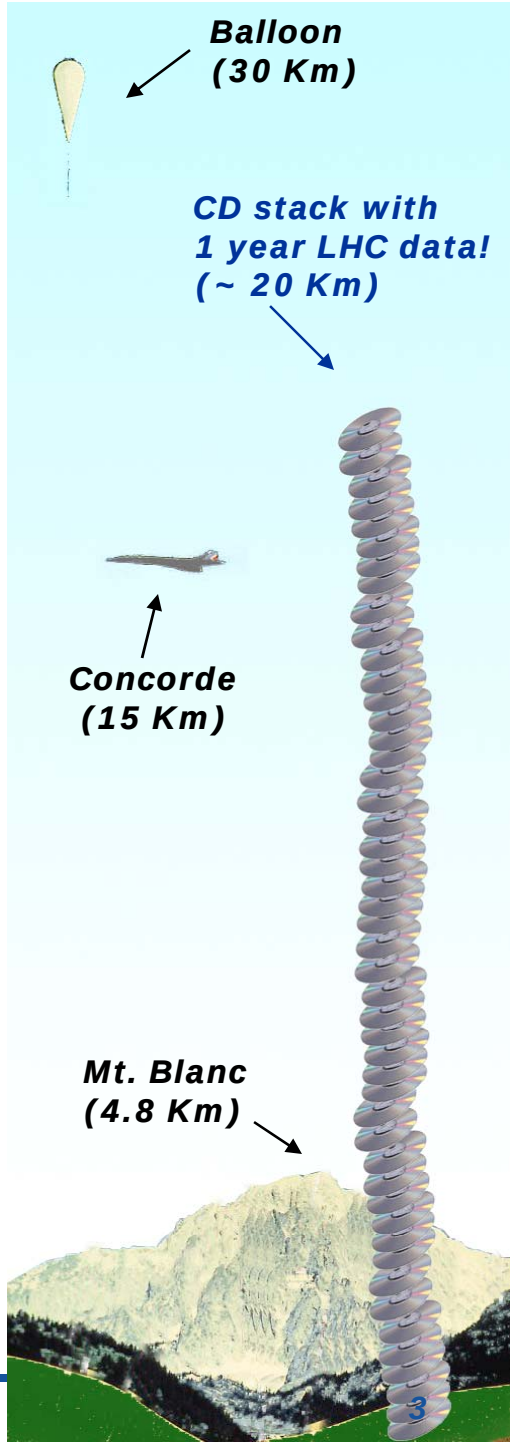
Outline

- Physics DB Services at CERN
- Motivations for Data Life Cycle Management activities
- Techniques used
- Sharing our experience with examples

LHC data

LHC data correspond to about 20 million CDs each year!

RDBMS play a key role for the analysis of LHC data



Databases and LHC

- Relational DBs play today a key role in the LHC production chains
  - online acquisition, offline production, data (re)processing, data distribution, analysis
    - SCADA, conditions, geometry, alignment, calibration, file bookkeeping, file transfers, etc..
  - Grid Infrastructure and Operation services
    - Monitoring, Dashboards, User-role management, ..
  - Data Management Services
    - File catalogues, file transfers and storage management, …
  - Metadata and transaction processing for custom tape storage system of physics data
  - Accelerator logging and monitoring systems

The RDBMS Workload

- Most applications are of OLTP type
  - Oracle used mainly for its transactional engine
  - Concurrency and multi-user environment
- Index based access paths and NL joins very important
- Sequential workload / DW type queries are not the main use case
- Another way of looking at it:
  - RDBMS stores ‘metadata’

Database SERVICES

At the heart of CERN, LHC and Experiment Operations

CERN IT Department

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

IT/DB GROUP

Experiment Offline Databases
9 Production DBs, 7 Integration DBs
5 Tests DBs, 7 Standby DBs

Streams

Experiment Online Databases
4 Production DBs

Data

CASTOR
CERN Advanced STORage manager
21 Production DBs
4 Development DBs

RAW Data

Middleware
73 Application Servers

Administrative/IT/ Engineering Databases
17 Production DBs
11 Development DBs
4 Ref/Test DBs

Accelerators ACC
12 Production DBs

LHC Operations

EDMS

AIS

Tier-1 Centres
dedicated 10 Gbit/s links
NDGF
GridKa
BNL
RAL
ASGC
SARA-NIKHEF
PIC
FNAL
CCIN2P3
INFN - CNAF
TRIUMF

LHC Experiments
LHCb
CMS
ATLAS
ALICE

PARTNERS:
CERN openlab
ORACLE

CERN

CERN Databases in Numbers

- CERN databases services – global numbers
  - Global users community of several thousand users
  - ~ 100 Oracle RAC database clusters (2 – 6 nodes)
  - Currently over 3300 disk spindles providing more than 1PB raw disk space (NAS and SAN)
- Some notable DBs at CERN
  - Experiment databases – 13 production databases
    - Currently between 1 and 9 TB in size
    - Expected growth between 1 and 19 TB / year
  - LHC accelerator logging database (ACCLOG) – ~30 TB
    - Expected growth up to 30 TB / year
  - ... Several more DBs on the range 1-2 TB

Data Lifecycle Management

- Motivated by large data volumes produced by LHC experiments
  - Large amounts of data will be collected and stored for several years
  - Different requirements on performance and SLA can often be found for ‘current’ and ‘old’ data sets
- Proactively attack ‘issues’ of databases that grow ‘too large’
  - Administration
  - Performance
  - Cost

Digression

- What is a VLDB?

"… VLDB means bigger than you are comfortable managing" (Cary Millsap)

- There is a part of it that has to do with the share DB size
  - The threshold seems to be moving with time and technology progress
  - Not too long ago 1TB Oracle DBs were classes as high end..

Administration of VLDB

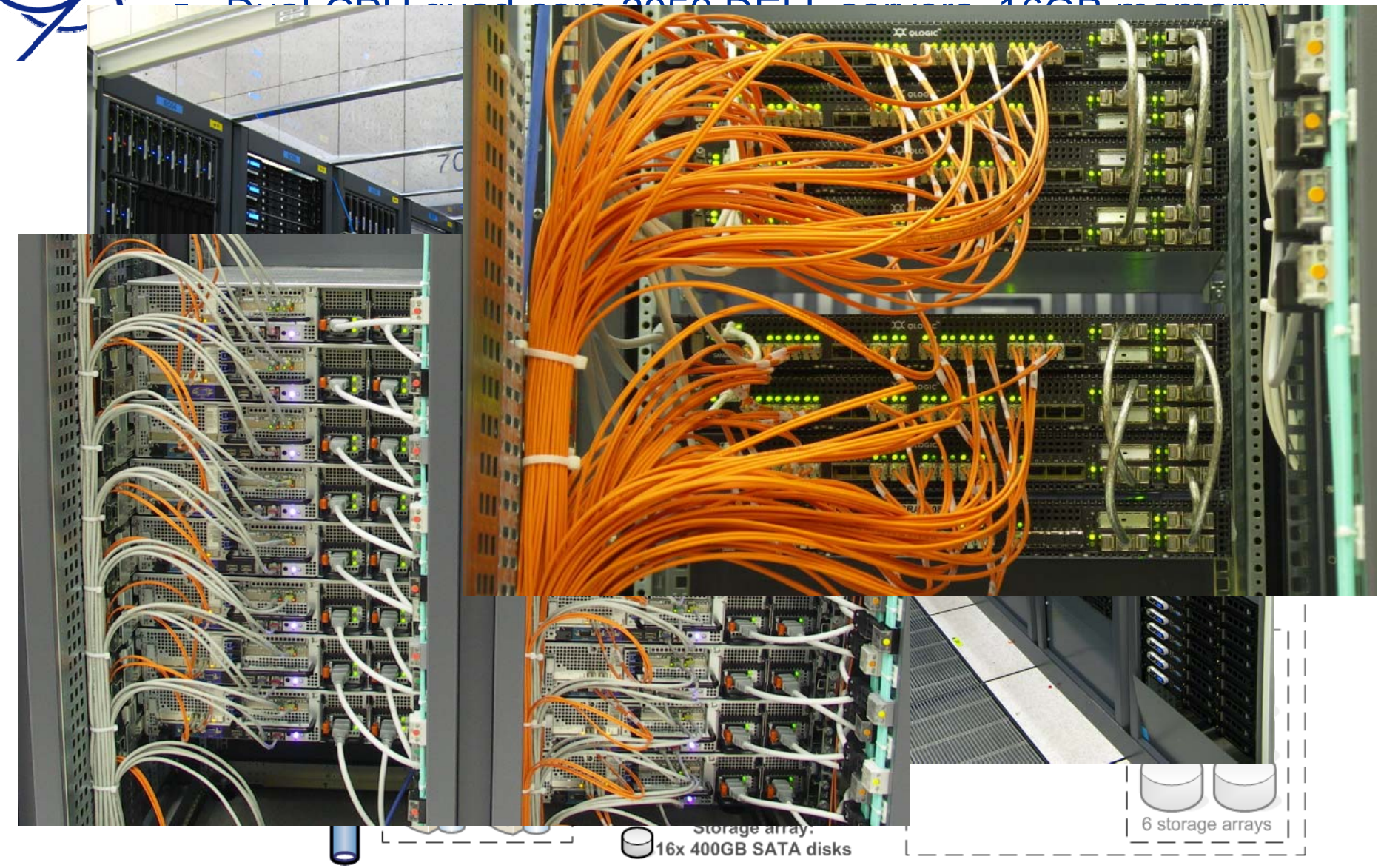
- VLDB and consolidation advantages:
  - Data consolidation, application consolidation
  - Some data sets are very large by nature
- VLDB and consolidation disadvantages
  - DB-wide operations can become slow (backup, stats gathering, full scan of largest tables)
  - Dependencies between applications
- Task
  - Identify how to get advantages of consolidation coexist with the idea of having a ‘lean’ DBs for manageability and performance

Attack problem from multiple sides

- No out of the box solutions available
- Attack the problem where possible
  - Applications
  - Oracle and DB features
  - HW architecture
- Application layer:
  - focus on discussing with developers
  - build life cycle concepts in the applications
- Oracle layer
  - Leverage partitioning and compression
  - Movement of data to an external ‘archival DB’

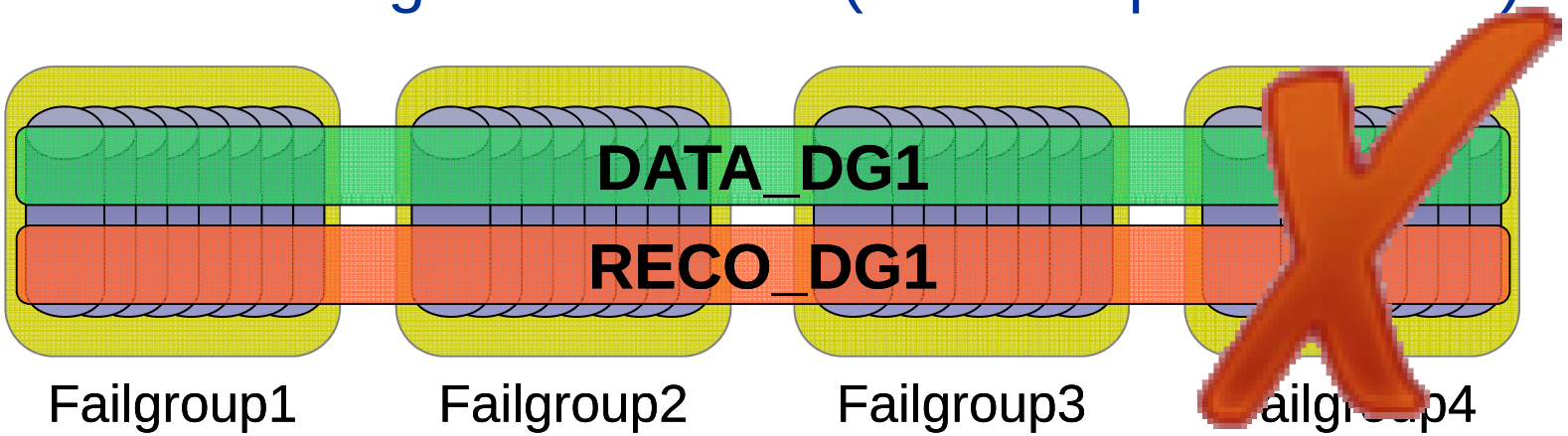
Commodity HW

- Dual CPU quad core 2950 DELL servers, 16GB memory



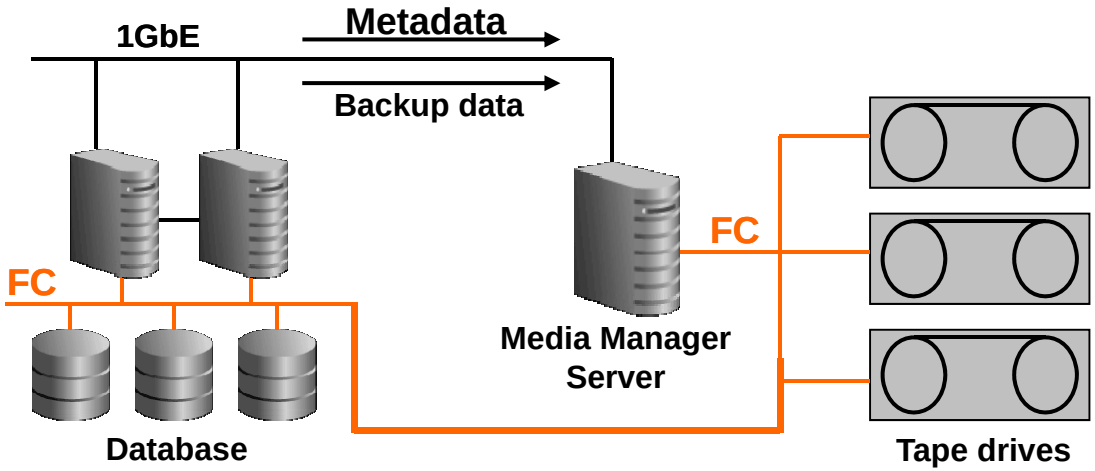
High capacity storage, resiliency and low cost

- Low cost HA storage with ASM
- Latest HW acquisition:
  - 492 disks of 2TB each -> almost 1 PB of raw storage
  - SATA disk for price/perf and high capacity
- ASM can take care mirroring
- Destroking can be used (external part for data)



Backup challenges

- Backup/recovery over LAN becoming problem with databases exceeding tens of TB
  - Days required to complete backup or recovery
  - Some storage managers support so-called LAN-free backup
    - Backup data flows to tape drives directly over SAN
    - Media management server used only to register backups
    - Very good performance observed during tests (FC saturation, e.g. 400MB/s)
  - Alternative – using 10Gb Ethernet



Application Layer

- Data Life Cycle policies cannot be easily implemented from the DBA side only
- We make sure to discuss with application developers and application owners
  - To reduced amount of data produced
  - To allow for DB structure that can more easily allow archiving
  - Define data availability agreements for online data and archive
  - Joint sessions to identify how to leverage Oracle features

Use Case: Transactional application with historical data

- Data has an active part (high DML activity)
- Older data is made read-only (or read-mostly)
- As data ages, becomes less and less used

Active Dataset

- Many Physics applications are structured as write-once read-many
  - At a given time typically only a subset of data is actively used
  - Natural optimization: having large amounts of data that are set read only
  - Can be used to simplify administration
  - Replication and backup can profit too
- Problem
  - Not all app are ready for this type of optimization

Time-Organized data

- Several key database tables are naturally time organized
  - this leads to range-based partitioning
  - Other solution is 'manual split' i.e. multiple similar tables in different schemas
- Advantages
  - Partitions can be treated as separate tables for bulk operations
  - Full scan operation, if they happen, do not span all tables

Techniques: Oracle Partitioning

- Range partitioning on timestamp attributes
- Note: unique indexes and local partitioning
  - Partitioning key must be part of index
- Partitions for ‘future time ranges’
  - Currently pre-allocated
  - 11g interval partitions will come handy
- 11g reference partitioning
  - Not used yet although interesting, will be tested

Partitioning Issues

- Index strategy
  - Indexes need to be local partitioned in the ideal case to fully make use of ‘partition isolation’
  - Not always possible, depends on application
  - Sometimes global partitioning better for performance
- Data movement issues
  - Using ‘Transportable tablespace’ for single partitions is not straightforward
- Query tuning
  - App owners and DBAs need to make sure there are no ‘stray queries’ that run over multiple partitions by mistake

‘Manual’ Partitioning

- Range partitioning obtained by creating multiple schemas and sets of tables
  - Flexible, does not require partitioning option
    - And is not subject to partitioning limitations
  - More work goes into the application layer
    - Application needs to keep track of ‘catalog’ of partitions
- CERN Production examples
  - PVSS (commercial SCADA system)
  - COMPASS (custom development at CERN)

Details of PVSS

- Main ‘event tables’ are monitored to stay within a configurable maximum size
  - A new table is created after the size threshold is reached
  - PVSS metadata keep track of current and historical tables
  - Additional partitioning by list on sys_id for insert performance
  - Historical data can be post-processed with compression
- Current size (Atlas offline): 4.6 TB

Details of COMPASS

- Each week of data is a separate table
  - In a separate schema too
    - DST and RAW also separated
  - IOT table used
  - Up to 4.6 billion rows per table
  - Key compression used for IOT
- Current size: ~10 TB

Details for PANDA Archive migration to Oracle

- 'Jobsarchived' table consolidates many smaller tables previously in MySQL
  - historical data coming from production
  - Range partitioning by time
  - One partition per month
  - One tablespace per year
- Performance
  - Application modified to add time range in all queries -> to use partition pruning
  - All indexes are local
  - Compromise change-> unique index on panda_id changed to be non-unique

Details of MDT DCS

- DATA tables
  - Contain live and historical data
  - Range partitioned
  - 1 partition per month
  - Additional table compression for historical data
- Indexes
  - primary key is local partitioned (prefixed with time column)
  - Other indexes are local too
  - Key compression on indexes
- Current size: ~300GB

Details of LCG SAME and GRIDVIEW

- Critical tables are partitioned
  - Range partitioned using timestamp
  - 1 partition per month
  - Contain live and historical data
  - All indexes are local
  - LCG_SAME makes use of partitioned LOBs
- Current size
  - 1.7 TB for LCG_SAME and 1.2 TB for GRIDVIEW

Techniques: Schema Reorganization

- When downtime of part of the application can be afforded
  - Alter table move (or CTAS)
  - Alter index rebuild
- More sophisticated
  - DBMS_REDEFINITION
  - Allows to reorganization of tables online (add partitioning for example)
  - Users experience, works well but it has let us down a couple of times in presence of high transaction rates
    - hard to debug and test ahead

Use Case: Archive DB

- Move data from production to a separate archive DB
  - Cost reduction: archive DB is sized for capacity instead of IOPS
  - Maintenance: reduces impact of production DB growth
  - Operations: archive DB is less critical for HA than production

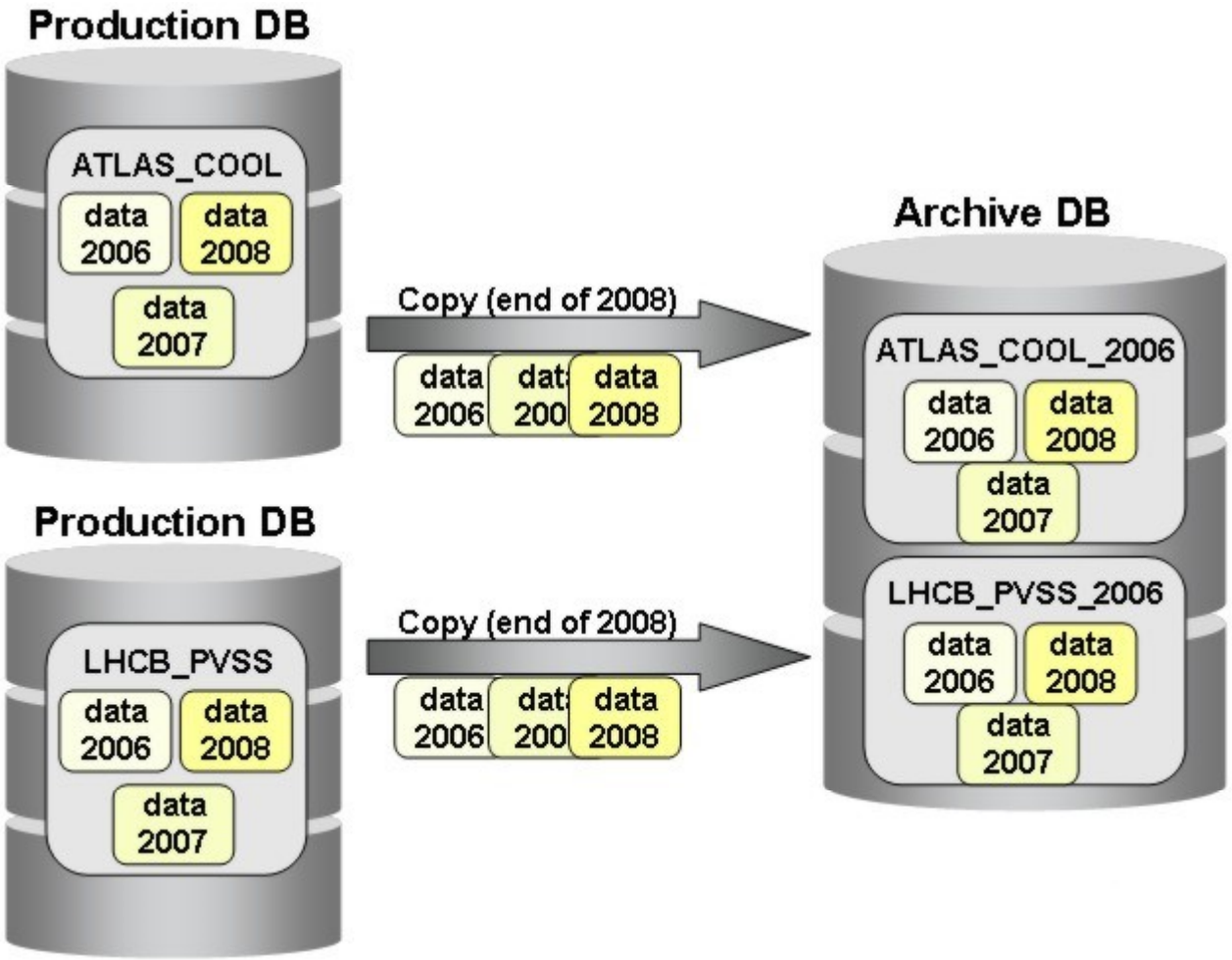
Archive DB service

- Proposal of archive DB
  - First presented Q4 2008, production Q4 2009
  - It's an additional DB service to archive pieces of applications
  - Data in archive DB mainly for read-only workload (with a lower performance)
- How to move data to archive and possible back in case of restore?
  - Most details depend on application owner
  - It's complicated by referential constraints
  - Area of custom work, still in progress

Archive DB in Practice

- Detach ‘old’ partitions form prod and load them on the archive DB
  - Can use partition exchange to table
  - Also transportable tablespace is a tool that can help
  - Archive DB post-move jobs can implement compression for archive (exadata 11gR2)
  - Post-move jobs may be implemented to drop indexes
- Difficult point:
  - One needs to move a consistent set of data
  - Applications need to be developed to support this move
    - Access to data of archive need to be validated with application owners/developers
    - New releases of software need to be able to read archived data

Example of Data Movement



Techniques: Data Movement

- impdp/expdp
  - Very useful although performance issues found
  - Impdp over DB link in particular
- Partitioning and data movement
  - Exchange partition with table
- Transportable tablespaces
  - Very fast Oracle-oracle data movement
  - Requires TBS to be set read only
    - Can be a problem in production
  - Workarounds:
    - Use of standby or a restored backup to move data from

Compression

- The ‘high level’ view:
  - Databases are growing fast, beyond TB scale
  - CPUs are becoming faster
  - There is a opportunity to reduce storage cost by using compression techniques
  - Gaining in performance while doing that too
  - Oracle provides compression in the RDBMS

Making it Work in Real World

- Evaluate gains case by case
  - Not all applications can profit
  - Not all data models can allow for it
  - Compression can give significant gains for some applications
  - In some other cases applications can be modified to take advantage of compression
- Comment:
  - Our experience of deploying partitioning goes on the same track
  - Implementation involves developers and DBAs

Evaluating Compression Benefits

- Compressing segments in Oracle
  - Save disk space
    - Can save cost in HW
    - Beware that capacity in often not as important as number of disks, which determine max IOPS
  - Compressed segments need less blocks so
    - Less physical IO required for full scan
    - Less logical IO / space occupied in buffer cache
    - Beware compressed segments will make you consume more CPU

Compression and Expectations

- A 10TB DB can be shrunk to 1TB of storage with a 10x compression?
  - Not really unless one can get rid of indexes (!)
  - Data warehouse-like with only FULL SCAN operations
  - Data very rarely read (data on demand, almost taken offline)
- Licensing costs
  - Advanced compression option required for anything but basic compression
  - Exadata storage required for hybrid columnar compression

Archive DB and Compression

- Non-active data can be compressed
  - To save storage space
  - In some cases speed up queries for full scans
  - Compression can be applied as post-processing
    - On read-mostly data partitions
      - (Ex: Atlas' PVSS, Atlas MDT DCS, LCG's SAME)
    - With alter table move or online redefinition
- Active data
  - Non compressed or compressed for OLTP (11g)

Data Archive and Indexes

- Indexes do not compress well
  - Drop indexes in archive when possible
  - Risk archive compression factor dominated by index segments
- Important details when using partitioning
  - Local partitioned indexes preferred
    - for ease of maintenance and performance
    - Note limitation: columns in unique indexes need be superset of partitioning key
    - May require some index change for the archive
      - Disable PKs in the archive table (Ex: Atlas PANDA)
      - Or change PK to add partitioning key

Oracle Segment Compression - What is Available

- Heap table compression:
  - Basic (from 9i)
  - For OLTP (from 11gR1)
  - 11gR2 hybrid columnar (11gR2 exadata)
- Other compression technologies
  - Index compression
    - Key factoring
    - Applies also to IOTs
  - Secure files (LOB) compression
    - 11g compression and deduplication
  - Compressed external tables (11gR2)
  - Details not covered here

Table Compression In Practice

- Measured compression factors for tables:
  - About 3x for BASIC and OLTP
    - In prod at CERN, example: PVSS, LCG SAME, Atlas TAGs, Atlas MDT DCS
  - 10-20x for hybrid columnar (archive)
  - more details in the following
- Compression can be of help for the use cases described above
  - Worth investigating more the technology
  - Compression for archive is very promising

Compression for Archive – Some Tests and Results

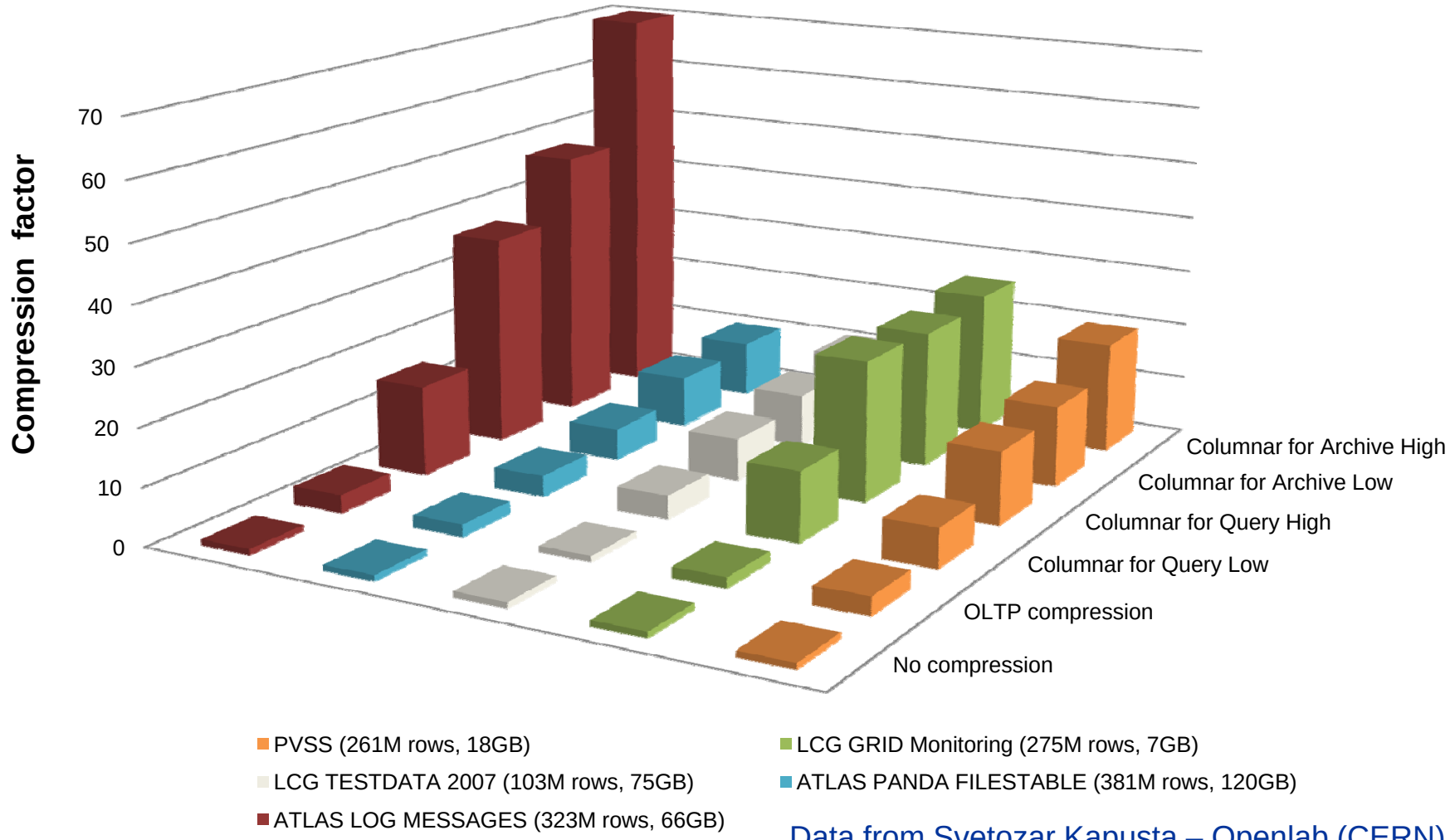
- Tests of hybrid columnar compression
- Exadata V1, ½ rack
- Oracle 11gR2
- Courtesy of Oracle
  - Remote access to a test machine in Reading (Q3 2009)

Advanced Compression Tests

- Representative subsets of data from production exported to Exadata V1 Machine:
  - Applications: PVSS (slow control system for the detector and accelerator)
  - GRID monitoring applications
  - File transfer applications (PANDA)
  - Log application for ATLAS
  - Exadata machine accessed remotely to Reading, UK for a 2-week test
- Tests focused on :
  - OLTP and Hybrid columnar compression factors
  - Query speedup

Hybrid Columnar Compression on Oracle 11gR2 and Exadata

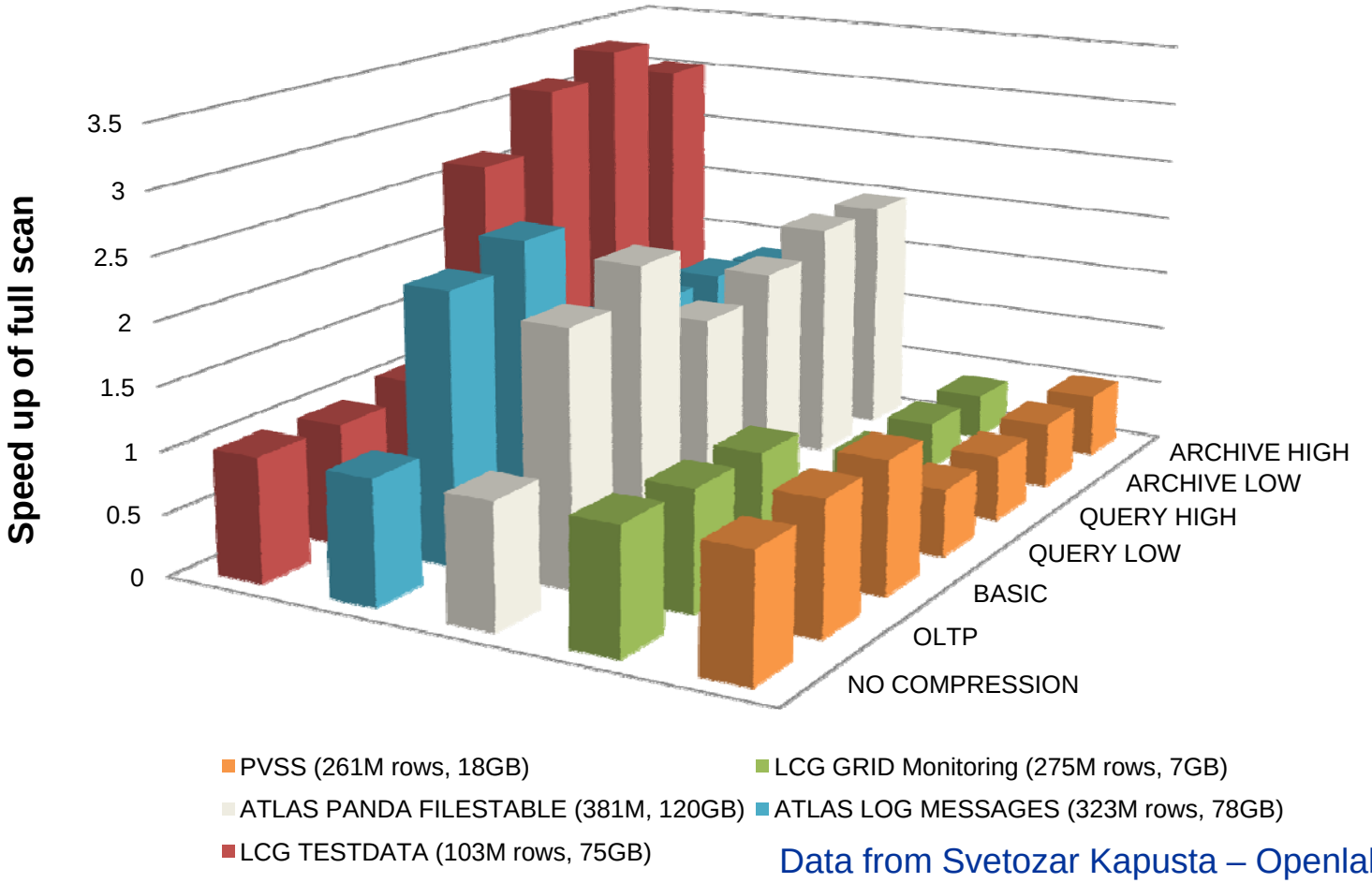
Measured Compression factor for selected Physics Apps.



Data from Svetozar Kapusta – Openlab (CERN).

Full Scan Speedup – a Basic Test

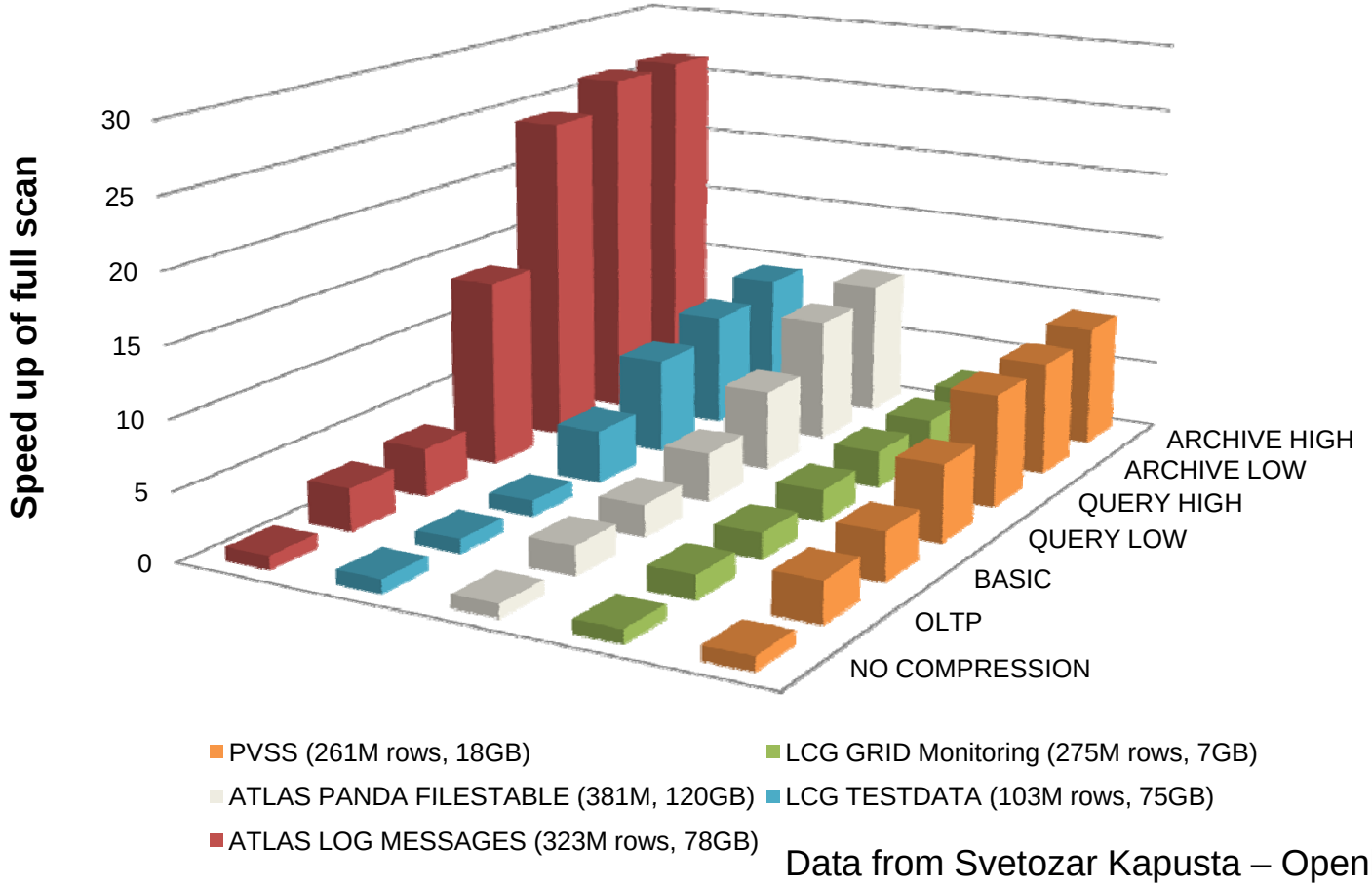
Full table scan speedup of compressed tables for count(*) operation (speed=1 for 'no compression')



Data from Svetozar Kapusta – Openlab (CERN).

IO Reduction for Full Scan Operations on Compressed Tables

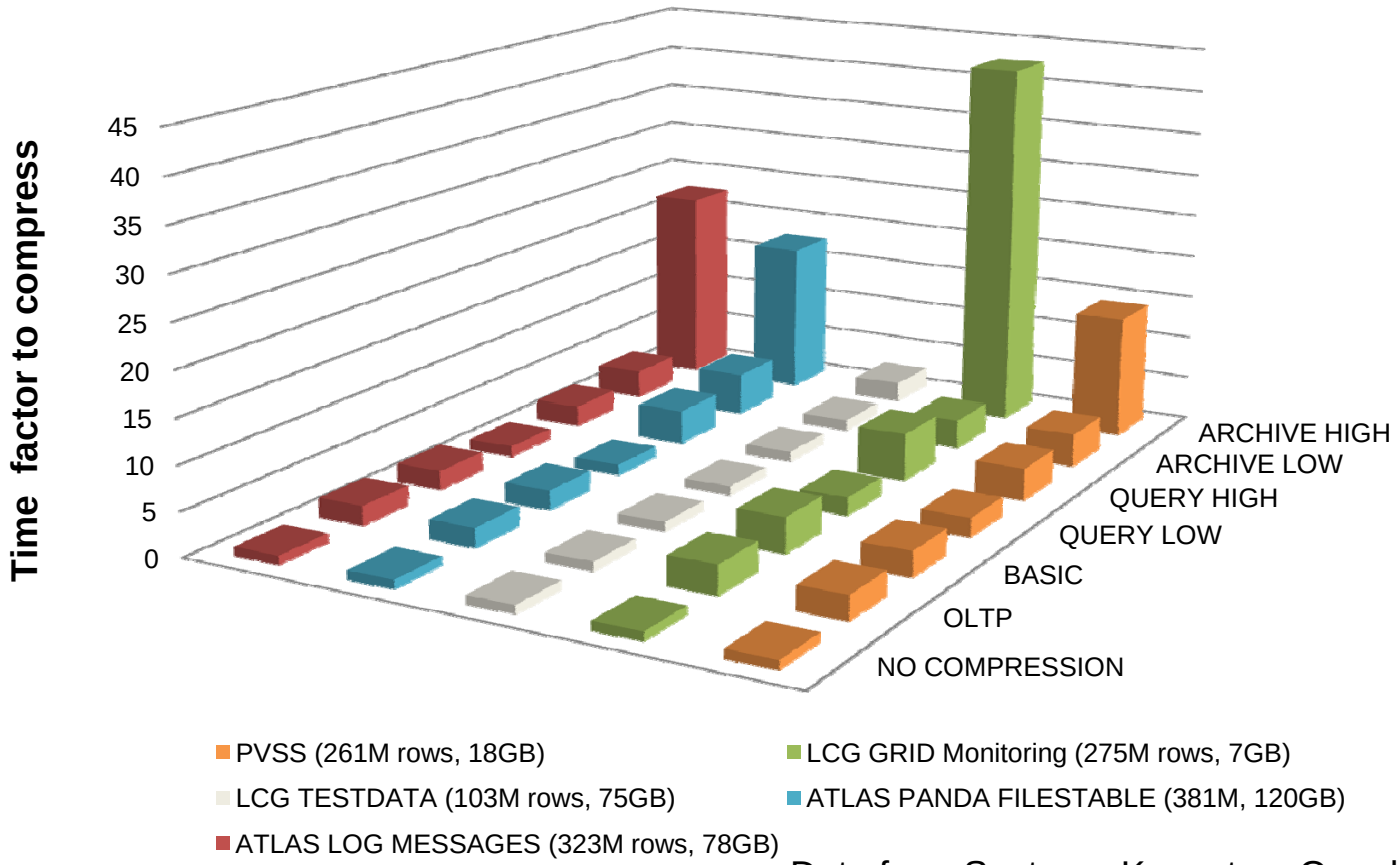
Hypothetical full scan speed up for count(*) operations
Obtained by disabling cell offloading in exadata.



Data from Svetozar Kapusta – Openlab (CERN).

Time to Create Compressed Tables

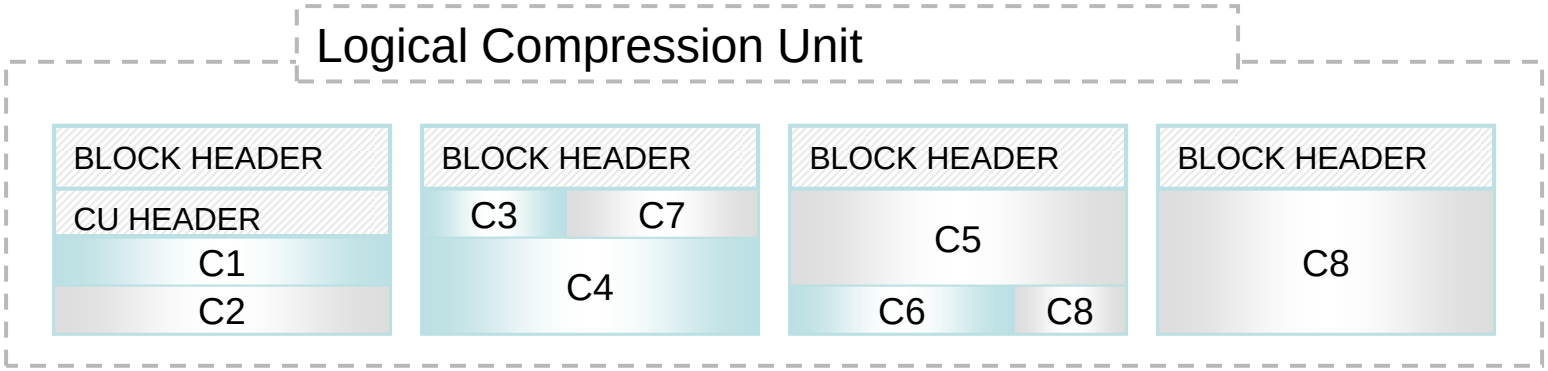
Table creation time for various compression types of various physics applications. T = 1 for 'no compression'.



Data from Svetozar Kapusta – Openlab (CERN).

A Closer Look at Hybrid Columnar Compression

- Data is stored in compression units (CUs), a collection of blocks (around 32K)
- Each compression unit stores data internally 'by column':
  - This enhances compression



Picture and info from: B. Hodak, Oracle (OOW09 presentation on OTN).

Compression Factors

- An artificial tests to put compression algorithms into work
- Two different types of test table
  - Constant: each row contains a random string
  - Random: each row contains a repetition of a given string
  - 100M rows of about 200 bytes each
  - Details of the test in the notes for this slide

Compression Factors

Table Type	Compression Type	Blocks Used	Comp Factor
Constant Table	no compression	2637824	1
Constant Table	comp basic	172032	15.3
Constant Table	comp for oltp	172032	15.3
Constant Table	comp for archive high	3200	824.3
Constant Table	comp for query high	3200	824.3
---	---	---	---
Random Table	no compression	2711552	1
Random Table	comp basic	2708352	1.0
Random Table	comp for oltp	2708352	1.0
Random Table	comp for archive high	1277952	2.1
Random Table	comp for query high	1449984	1.9

Hybrid Columnar and gzip

- Compression for archive reaches high compression
  - How does it compare with gzip?
  - A simple test to give a 'rough idea'
  - Test: used a table populated with dba_objects
    - Results ~20x compression in both cases

Method	Uncompressed	Compressed	Ratio
gzip -9	13763946 bytes	622559 bytes	22
compress for archive high	896 blocks	48 blocks	19

Compression and DML

What happens when running row updates on compressed tables? What about locks?

- BASIC and OLTP:
  - the updated row stays in the compressed block
  - 'usual' Oracle's row-level locks
- Hybrid columnar:
  - Updated row is moved, as in a delete + insert
    - How to see that? With dbms_rowid package
  - New row is OLTP compressed
  - Lock affects the entire CU that contains the row

Compression and Single-row Index Range Scan access

Table Compr Type	Consistent gets (select *)	Consistent gets (select owner)
no compression	4	4
basic compression	4	4
comp for oltp	4	4
comp for query high	9	5
comp for query low	9	5
comp for archive low	10	5
comp for archive high	24	5

- Test SQL: select from a copy of dba_objects with a index on object_id
- Predicate: ‘where object_id=100
- Note: ‘owner’ is the first column of the test table

Conclusions

- Data Life Cycle Management experience at CERN
  - Proactively address issues of growing DBs
  - manageability
  - performance
  - cost
- Involvement of application owners is fundamental
- Techniques within Oracle that can help
  - Partitioning
  - Archival DB service
  - Compression

Acknowledgments

- CERN-IT DB group and in particular:
  - Jacek Wojcieszuk, Dawid Wojcik, Maria Girone
- Oracle, for the opportunity of testing Oracle Exadata
  - Monica Marinucci, Bill Hodak, Kevin Jernigan

- More info:

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

  http://cern.ch/canali