# Data Lifecycle Review and Outlook

Luca Canali, CERN

Distributed Database Operations Workshop

April 20, 21 Barcelona

CERN IT Department
CH-1211 Genève 23
Switzerland
**www.cern.ch/it**

# Outline



- Motivations and main problems addressed by datalife cycle management
- Implementation ideas and techniques for the physics DBs
- Sharing results from production
- Ideas for future work


# Data Lifecycle Management



- Motivated by large data volumes produced by LHC experiments
  - Large amounts of data will be collected and stored for several years
  - Discussion started in November 2008
- Main issues of databases that grow ‘too large’
  - Administration
  - Performance
  - Cost

CERN IT Department
CH-1211 Genève 23
Switzerland
**www.cern.ch/it**

# Administration of VLDB

- Administration and performance of VLDBs can suffer from
  - Full table/partition/index scan
  - Data reorganization and/or bulk delete
  - DB-wide operations (restore, stats gathering)
- VLDB advantages:
  - Data consolidation, application consolidation
- Task
  - Identify how to get advantages of consolidation coexist and a ‘lean’ DBs for manageability and performance

CERN IT Department
CH-1211 Genève 23
Switzerland
**www.cern.ch/it**

# Active Datasets



- Many applications are structure 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

# 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
  - Some times a local index is not optimal for performance
- Data movement issues
  - ‘Transportable tablespace’ of partitioned tables 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

CERN IT Department
CH-1211 Genève 23
Switzerland
**www.cern.ch/it**

# Partitioning, Production Examples



- Production examples
  - PANDA Archive and MDT_DCS for Atlas
  - LCG SAME and GRIDVIEW
- Other applications that are using partitioning
  - Atlas TAGS
  - ATLAS dashboard_dm
- Candidates for using partitioning
  - COOL (on the to-do list)
  - CMS conditions, dashboard

CERN IT Department
CH-1211 Genève 23
Switzerland
**www.cern.ch/it**

# Other implementations



- Range partitioning obtained by creating new schemas or set of tables
  - More flexible but puts more work on app
  - App needs to keep track of ‘catalog’
- Production examples
  - PVSS
  - COMPASS

# Details for PANDA archive



- 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

CERN IT Department
CH-1211 Genève 23
Switzerland
**www.cern.ch/it**

# Details of MDT CONF

- DATA tables
  - Contain live and historical data
  - Range partitioned
  - 1 partition per month
  - Additional table compression for historical data
- Indexes
  - primary key in local (prefixed with time column)
  - Other indexes are local too
  - Key compression on indexes
- Current size: 153GB

# 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
  - 721 GB for LCG_SAME and 365 GB for GRIDVIEW

# 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): 2.2TB

# 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 billion rows per table
  - Key compression used for IOT
- Current size: 7 TB

CERN IT Department
CH-1211 Genève 23
Switzerland
**www.cern.ch/it**

# Compression



- Compression of read only data for archiving
  - Step1, table and index compacting and/or restructuring
  - Step 2, use of oracle compression for table data of direct operations in 10g
- Our experience
  - Positive experience, or example PVSS production compression in 2008
- Drawback, index ‘key compression’ capabilities in Oracle are limited
  - Key compression, not universally applicable
  - Although when it works is beneficial for online applications too

CERN IT Department
CH-1211 Genève 23
Switzerland
**www.cern.ch/it**

# Compression, outlook

- Compression for archiving
  - Very high compression factors
  - Very promising to keep online archive data
- Our experience
  - PVSS, 40x compression for table data
  - High CPU usage
  - Indexes are not compressed (often indexes can take space with same order of magnitude as table data)
  - Work in progress
- Other promising usages of compression
  - Compression for online table data

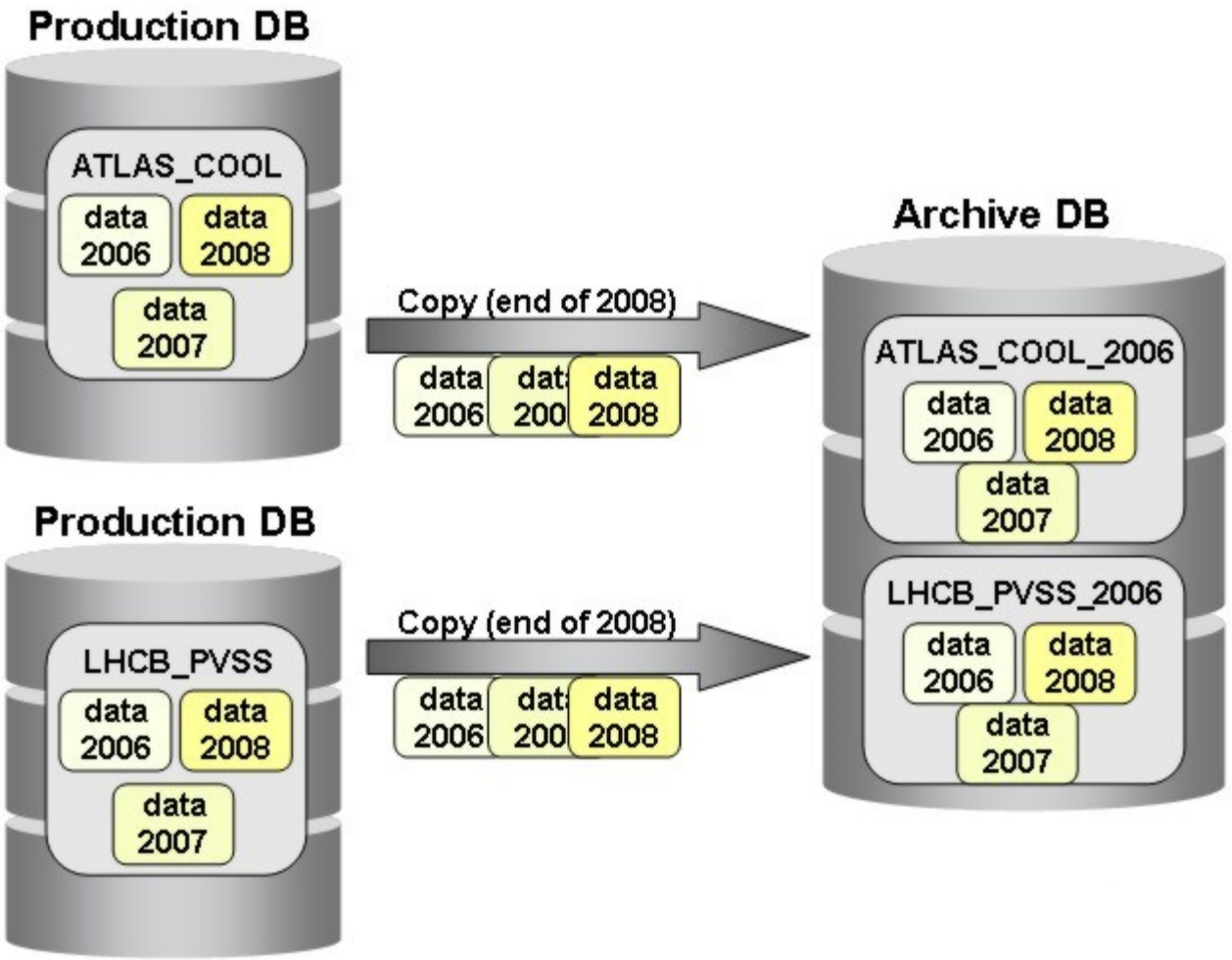
CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

# Archive DB service

- Proposal of archive DB
  - Presented last November
  - It's an additional DB service to archive pieces of applications
  - Data will be available in the archive DB exceptionally 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 investigation and future work

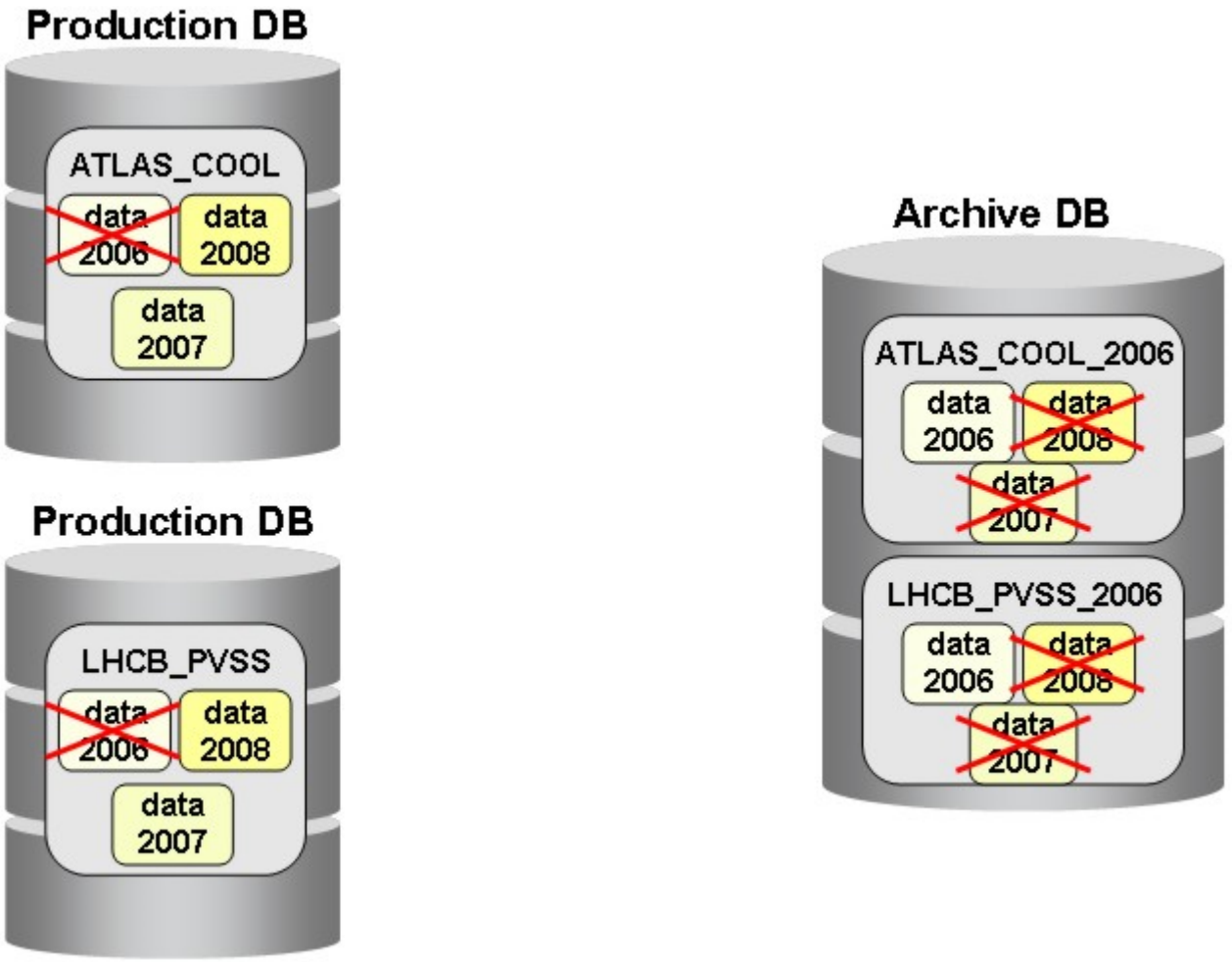
CERN IT Department
CH-1211 Genève 23
Switzerland
**www.cern.ch/it**

# Example of data movements




CERN IT Department
CH-1211 Genève 23
Switzerland
**www.cern.ch/it**

# Example of data purging after archive




CERN IT Department
CH-1211 Genève 23
Switzerland
**www.cern.ch/it**

# Summary



- Data lifecycle management is needed
  - Physics database schemas are growing fast
  - Implementing partitioning and other data handling mechanism require collaboration of application developers
  - Several successful example have been shown
  - Partitioning and compression are important
  - This is still an ongoing effort
  - Tests with the archive DB service will start soon