Handling of time-critical Condition Data in the CMS experiment
Experience of the first year of data taking

CHEP 2012
New York, 22 May 2012

Giacomo Govi (Fermilab & CERN)
On behalf of CMS
Outline

• Conditions in CMS
• Strategy and Choices
• Data and storage models
• Core system
• Operation
• Outlook
Condition Data in CMS

In HEP “Conditions” are usually non-event data varying with time

In CMS a specific subset is critical for the dataflow
• DAQ and early processing stages:
  – Status and Configuration for Detectors, Triggers
  – Run Information
• Reprocessing
  – Detector Calibration
  – Beam and Luminosity information

These data need to be properly re-synchronized during the processing

‘Condition Database’ is the dedicated infrastructure
  – Software packages in CMSSW framework
  – Applications providing services for both online and offline environments
Working context

• Data providers/consumers for Condition Database include the entire experiment infrastructure
  – Detectors, Trigger systems, DAQ and Monitoring systems

• Potentially little control on volumes expected, technologies, standards, rules, access patterns
  – Access for reading/updating the database cannot be completely automatized
  – Many institutions of the collaboration are involved
  – Heterogeneous know-how and practices

• Need to narrow to access to the database to well controlled use cases
  – Critical operation have to robust and protected by mistakes
Data model

- **Payload**: data structure designed according to the detector/task needs.
  - Typically: header + param container(s)

- **IOV** (Interval Of Validity): array of intervals (time or run number) with their associated Payload references

- **Tag**: label identifying/categorizing a specific IOV.
- **Global Tag**: A consistent set of Tags required for a given workflow
Applications: strategy

• **Enforce the DB access via a common software**
  • Public interface with a single, concrete implementation

• **Support of a well defined set of use cases**
  • Allow to control the volumes and access patterns
  • Queries are predefined and can be tuned a priori
  • No support for arbitrary query on the IOV or Payloads

• **Focus on data integrity**
  • IOV updated appending new intervals
  • IOV never deleted
  • Limited manipulation of tags
The storage is based on ORACLE

Data grouped by source (*Detector or Task*)
- Individual schema (ORACLE *account*)

Object-Relational approach
- reduces the ‘relational’ complexity of the schema
- object instances are mapped to records in their tables
- blob streaming adopted
  - for large arrays (>200 elements)
  - for multi-parameters or complex structure
  - for files
Queries

An useful data set is identified by few queries all involving 1 table, 1PK
  – Resolve the IOV OID from the Tag
  – Load the IOV identified by the previous ID
  – Load the Payload corresponding to the target time window

Two more queries are required to resolve the mapping Object/Relational at run time

• Queries are simple and well established
• Cursors contain most of the time one row only!
Synchronization: time critical

- A set of conditions is critical for operation of the data production
  - L1 Trigger parameters
  - Detectors parameters for HLT
  - Run summary data
  - Beam spot data
- Need to centralize the control of the applications providing the synchronization
  - Using a common application (“PopCon”)
  - Deploying the jobs in reserved nodes in the Online network

Giacomo Govi
Synchronization: offline calibration

- Detector and Hardware calibration require offline processing
  - May be completed in several iterations
  - Frequently consists in a manual operation
- The final set has to be available in the whole dataflow
  - Need to be stored in ORCON (online private network)
- A “DropBox” service allows the automatic exportation
  - Pulls the files handling the firewall issues
  - Synchronize multiple data set fragments
  - Update existing Tag/ create new Tag

<table>
<thead>
<tr>
<th>offline</th>
<th>Firewall</th>
<th>scp</th>
<th>online</th>
<th>ORCON</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>10m</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- check metadata
- check payload
- get Run Number
- export
Reading/Distribution

• Load from the reco jobs at Tier0/1s is potentially high
  • >200 condition objects to read with 3-5 tables => ~800 queries
• Data is read-only for a time scale of 10 mins
• FronTier caches minimize the direct access to ORACLE servers
  – At the price of the latency implied by the refreshing policy
  – 2 Frontier services (P5 and Tier0/1)  more details:[220,D.Dykstra]
• Snapshot from Oracle DB ensure reproducibility
  – Data set exported in a dedicated server
• SQLite files for simple shipping data through the network
  – Used by the Offline DropBox to export calibration data into ORCON
Operation

• The Condition DB has been working stably during the data taking
  – No major failure causing interruptions
  – The Monitoring system plays an essential role  \textit{see: [202, Di Guida]}

• Fixes for mistakes or invalid operations
  – Mostly on the actions performed ‘by hand’
  – On single Tags
  – On the export instructions for the DropBox (further consistency check added)

• Dedicated exportation for specific needs
  – Account migration, data set cleanup

• Changes in the Detector’s Data Models required the most of the operation activity
  – They are reflected in the DB schema
  – Schema evolution is supported for basic changes
  – New classes describing conditions treated with new DB accounts
Outlook

• Most of the work is currently spent in operation
  – Follow-up of data taking and processing needs
  – Migration of existing data sets to a new CMS proprietary format

• Only little development are still ongoing

• Focus of the current phase is consolidation of the (still) critical areas
  – Bookkeeping system for the DropBox
  – Security for DB access (authentication and authorization)
  – Improvements for Monitoring System
  – Handling of schema evolution for Blob-based storage

• No major changes are foreseen in the system for 2 years
  – Some investigation for considering alternatives to RDBMS as storage could start in the future
Summary

• The Condition DB plays a key role in the CMS operation.

• The application is part of the CMS Software framework. The storage is based on the Oracle RDBMS technology.

• Focus of the choices is the control of a potentially large set of access patterns into a single software supporting predefined use-cases.

• The strategy chosen for the implementation allowed to operate with high stability and reliability during the data taking.

• No major change are expected in the system in the near future.