

CMS offline conditions database software

Z. Xie (Princeton University)
G. Govi(Northeastern University)
V. Innocente(CERN PH-SFT)

For the CMS collaboration

Outline

- Introduction
- Data Model
- Software
 - Decomposition
 - From usage-centered design view point
- Deployment Model
- Status
- Conclusions

Introduction

- Objectives and scope:
 - Manage non-event data whose time validity span more than one event
 - Manage the versioning of the validity
 - Data needed by offline acquired by online operations and/or calculated by offline algorithm
 - e.g. Calibration, alignment, channel map, dead/hot channels, data quality
- Technology choices:
 - Several relational databases behind technology neutral software layer
 - Relational tables managed as C++ objects

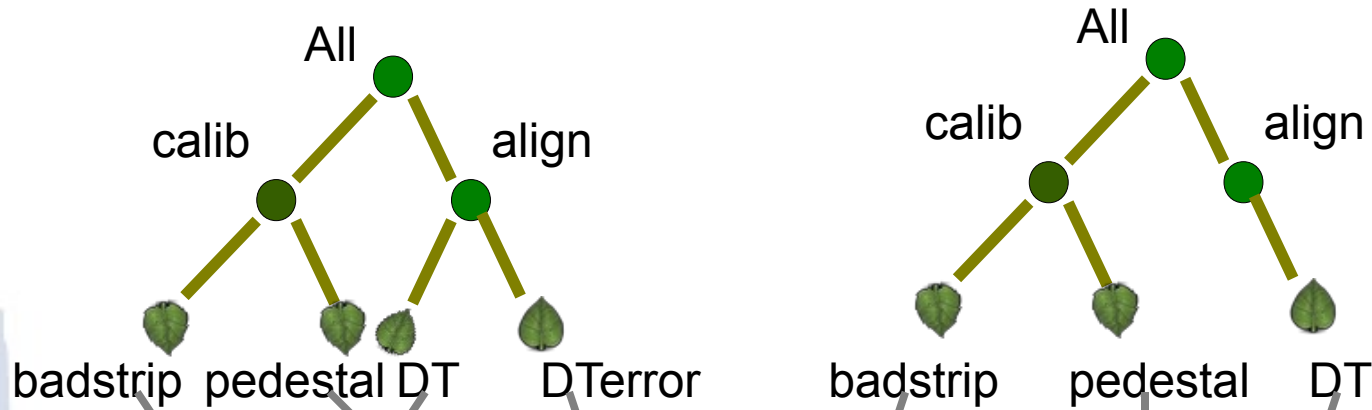
Data model characteristics

- Conditions data are C++/POOL objects(Payload) indexed by Interval-Of-Validity(IOV) in time(runnumber,universal timestamp,luminosity section id).
- IOV sequence is consecutive. No holes, no overlap. IOV sequences are C++/POOL objects as well.
- Versions of IOV are identified by tags which are the lowest level of access to the data
- A set of tags can be organized in a tree structure where each node is a data access point (global tag)
- Tag trees are implemented with the light-weight relational “Nested Set” model of hierarchy: one table per tree, fast traversal with no multiple self-joins, no recursions, no limit on the depth

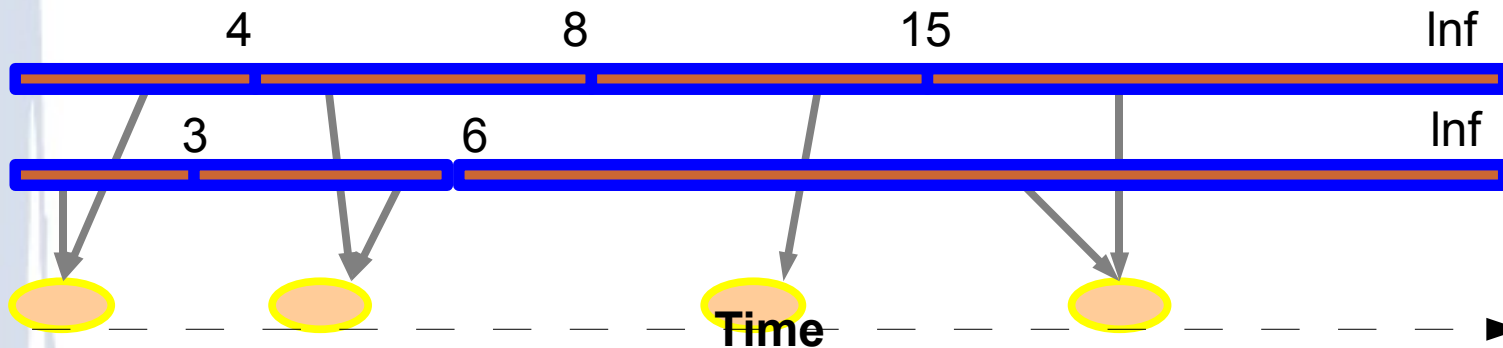
Data model - overview



Global tag trees



Tag inventory




IOV tags


payloads

21-27 March 2009

CHEP'09 Prague, Czech Republic

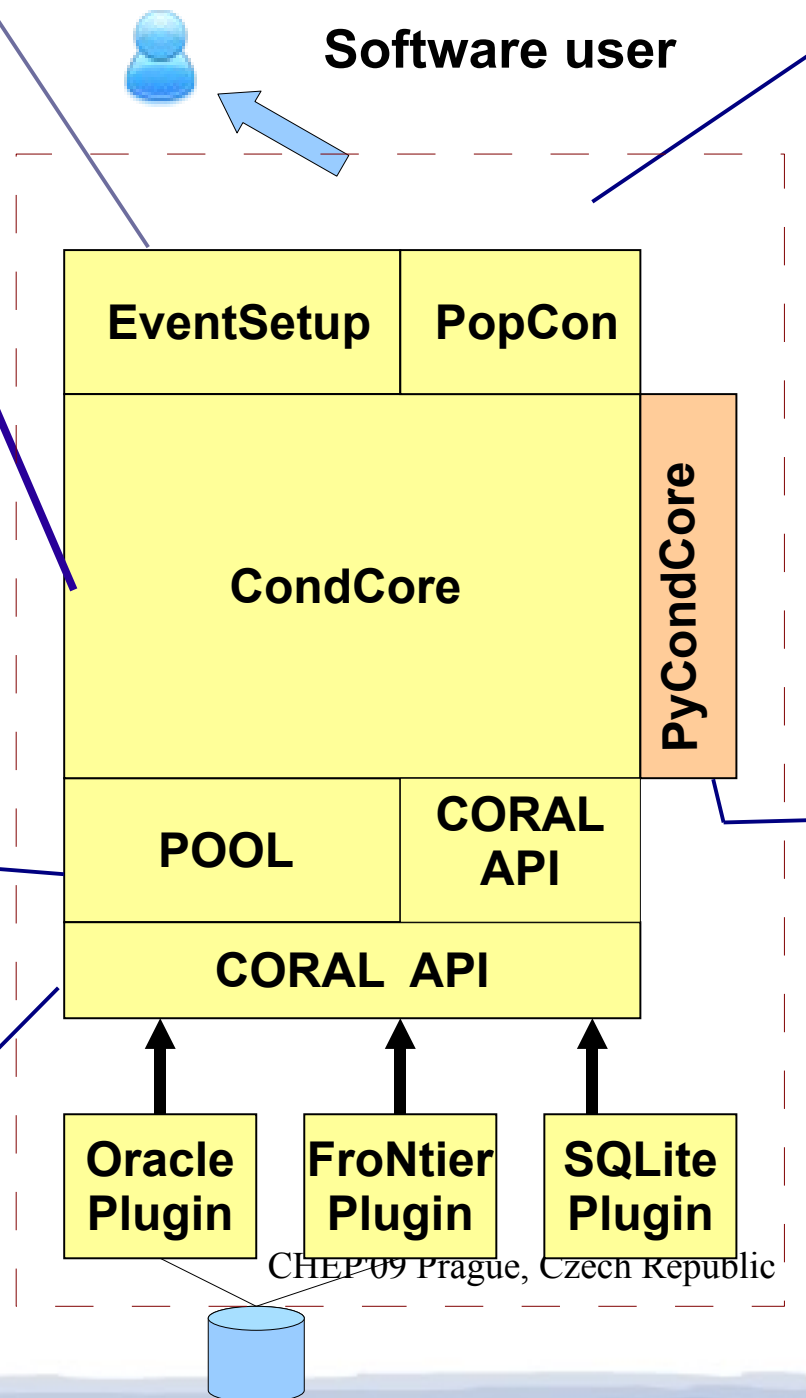
Software components

General non-event data framework provides detector conditions at an instant in time

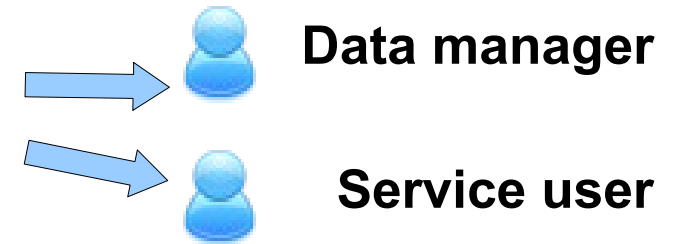
Core component isolates persistency layer; Manages connection, transaction, IOV and tags

Maps objects to tables; Object I/O

Abstract API for DB connection, query; dispatches technology-specific calls to plugins



Mini data output and transfer framework. Transfer data from external source to offline format and database; Integrated transfer logging And IOV consistency check



Python-C, Python layer exposes functionalities in Python as building block for utilities and services

CHEP'09 Prague, Czech Republic

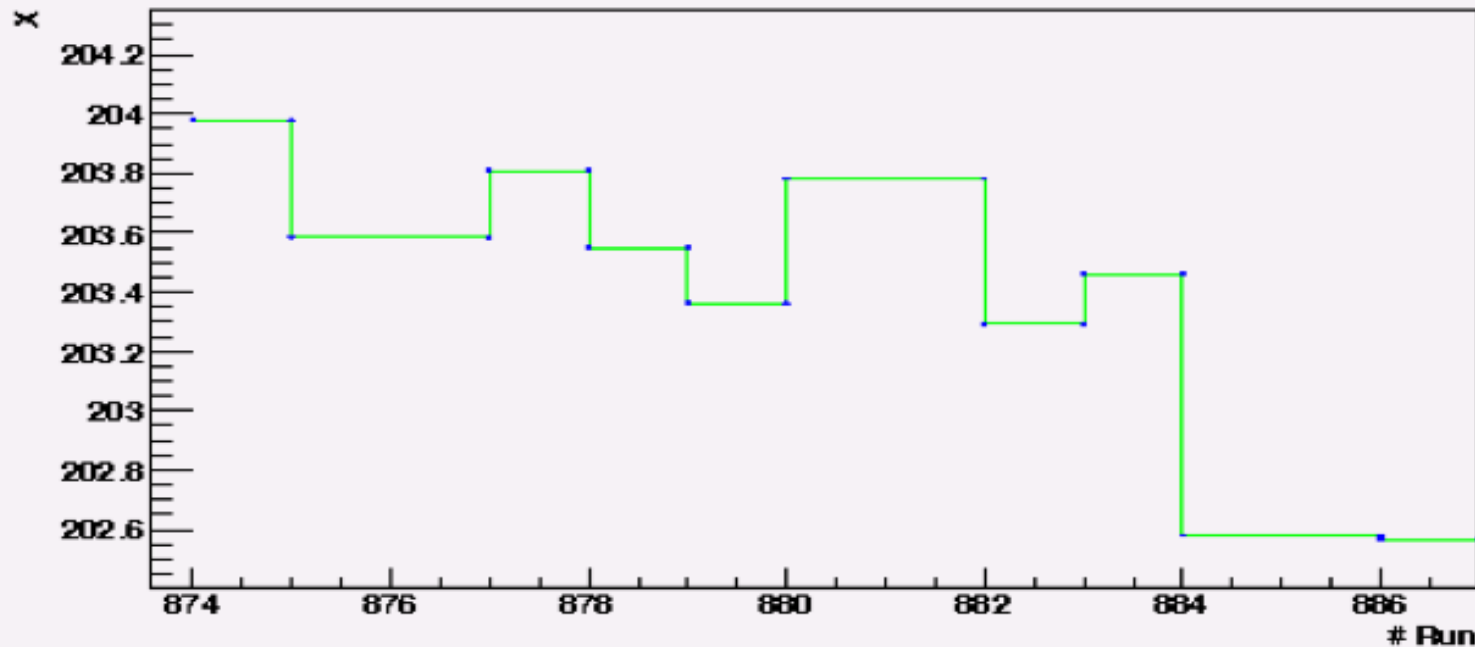
Usage-centered design: focus differently for different user roles

- **Core software users**: No desire for knowledge beyond event processing framework
 - No conditions nor database API. Generic EventSetup delivers C++ object. Data dictionary mechanism identical to event data.
- **Data management users**: Don't care about the payload content; simple tag management and data transfer operations
 - Tag management layer uses light-weight relational model and is built on Python; Utilities of data copy cross databases
- **Data miners**: Browse tags, payload and summary data searching for anomalies and patterns. Like graphic and offline capability, not needy for speed
 - Web service for tag and summary data browsing and plotting; Web service can be deployed standalone on laptop; Utilities to plot data from ROOT

Browsing data in ROOT

```
TCanvas * cc = new TCanvas ("X vs Run","X vs Run",10,10,700,400);  
TGraph *Graph = new TGraph(counter,Run,X);  
Graph->Draw("APL");
```

Graph EcalPedestals X vs Run Bar



Courtesy A.Massironi

Web service screenshot

IOV tag management:

Service

cms_orcoff_int2r

Detector / Task

ECAL

List all tags

Available tags:

CSA06calibs0.00
CSA06calibs0.04
EcalADCToGeVConstant_trivial
EcalGainRatios_trivial
EcalIntercalibConstants_trivial
EcalPedestals_from_online
EcalPedestals_test
EcalPedestals_trivial
EcalTBWeights_trivial
EcalWeightXtalGroups_trivial

Display IOV for selected tags

Available IOVS:

Container name: EcalPedestals

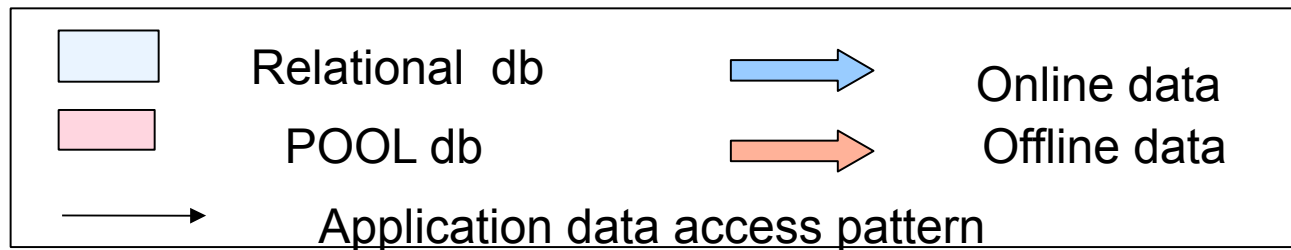
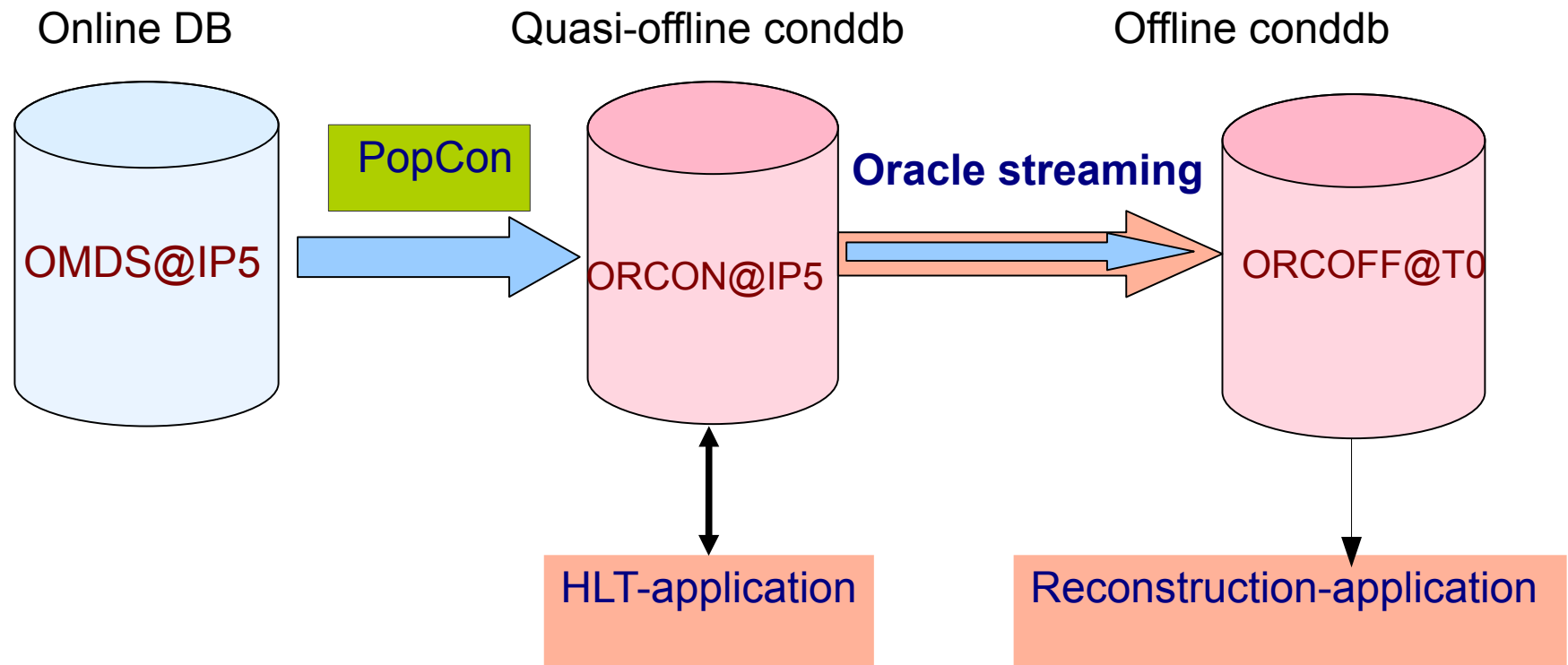
Since	Till
1	489
490	490
491	491
492	492
493	493

Courtesy K.Dziedziniewicz

Deployment model

- Central oracle databases deployed at CERN
 - one master instance connected to the restrictive online network; one instance at Meyrin site connected to the public network; Oracle streaming makes data in the online master instantly available to CMS public
- Frontier cache connected to both central oracle instances ensure read-only access in CMS world-wide computing centers
- Data are written to the online Oracle master through PopCon mini-framework or imported from sqlite files in a controlled manner
- Development, Integration and Production Oracle services are provided for each detector. Additionally, SQLite files are used for testing and validation.

Deployment model graphical view



Status

- Core components development started in 2006
 - Complete functionalities
 - First tested in Computing-Software-Analysis challenge 2006 and used in all the global run, cosmic run and MC data productions afterwards. (more deployment experience see poster session)
- Service components development started in 2007
 - In production as standard cms web service since 2008 (more see poster session)
 - More functionalities are being added

Conclusions

- Usage-centered design provides different functionalities and services for different user roles.
- Object-relational data model for data combined with light-weight relational model for tag metadata
- Centrally managed Oracle database instances with world-wide read-only access via Frontier. SQLite database is also widely used for testing and data exchange purpose.
- Software system and computing infrastructure has been tested and used in all productions since 2006