CMS Calibration-Alignment Databases for The Magnet Test and Beyond

Vincenzo Innocente
On Behalf of CMS Collaboration

3D Workshop
October 2005

Acknowledgements

- Many people have contributed much to the DB planning and development including: Oliver Buchmuller, Michael Case, Frank Glege, Yuyi Guo, Vincenzo Innocente, Saima Iqbal, Chris Jones, Lee Lueking, Gennadiy Lukhanin, Zhen Xie
- Plus many detector people: Ricky Egeland, Shahram Rahatlou, Fedor Ratnikiov, Igor Vorobiev, and many more.
- And the POOL team

Contents

- Overview of non-event data
  - Not covered: “data booking system”
- EventSetup in EDM Framework
- Non-event data objects
- Software stack and deployment
- Milestones and schedule
- Conclusion
  - More details on online-offline and T0->Tn tomorrow

Overview

- Areas included are 1) Geometry and Alignment, 2) Calibration, 3) DB infrastructure.
- The short-term objective is providing a functioning system by October 2005
- This will be used by sub-detector groups for initial reconstruction and analysis testing, and for the Cosmic Challenge (aka Magnet test) in 2006.
- The longer term plans include stress testing and providing additional functionality for offline subsystems beyond the alignment and calibration data.

Definitions

- IOV: Interval of Validity – The range of time for which a set of non-event data is valid.
- Non-event data: Data who's IOV spans more than one event.

Kinds of Non-event Data (1)

- **Monitoring (single version)**
  - Examples
    - Sensor channel values (HV, LV, Temp, Pressure,…)
    - Thresholds, noise, other characteristics of detectors.
  - Characteristics
    - Sensor locations have geometry scheme that is different from detector signal channels and change value independently of each other in time.
    - Data is collected in real time and has a single "*version*".
      - It may be looked at as an "event" but not a "physics event"
      - Potentially large volume, but often WORN
    - IOV is the time region for which a value does not change.
      - (Offline) Data volume and rates depend on detector stability

Kinds of Non-event Data (2)

- **Managed Interval of Validity (multi-version)**
  - Examples
    - Calibration
    - Alignment
    - Channel mappings
    - Dead channels, Hot channels
    - Data quality
  - Characteristics
    - Data loaded into database in “*sets*” for a detector subsystem.
    - An “*algorithm*” is used to create the data. More than one algorithm might be stored, and each might have multiple “*versions*”.
    - IOV is the time region over which a set is valid.

IOV Defined "Offline"

In summary, this document defines the general concept of the offline IOV service that will be utilized not only for pure offline operation but also for operation of the High Level Trigger. The assignment of the IOV is a pure offline task and supposed to be carried out by the algorithm (or person) in charge of providing a certain conditions data set for the offline (and HLT) operation.

Some more technical aspects

The behavior of the IOV service is defined by its interface. The general requirements on the IOV service are the following:

I. given a time, it returns an object that is guaranteed to be valid TILL a given time. It also returns the corresponding TILL time.

II. one or more pairs of SINCE time and data object can be registered to the IOV service.

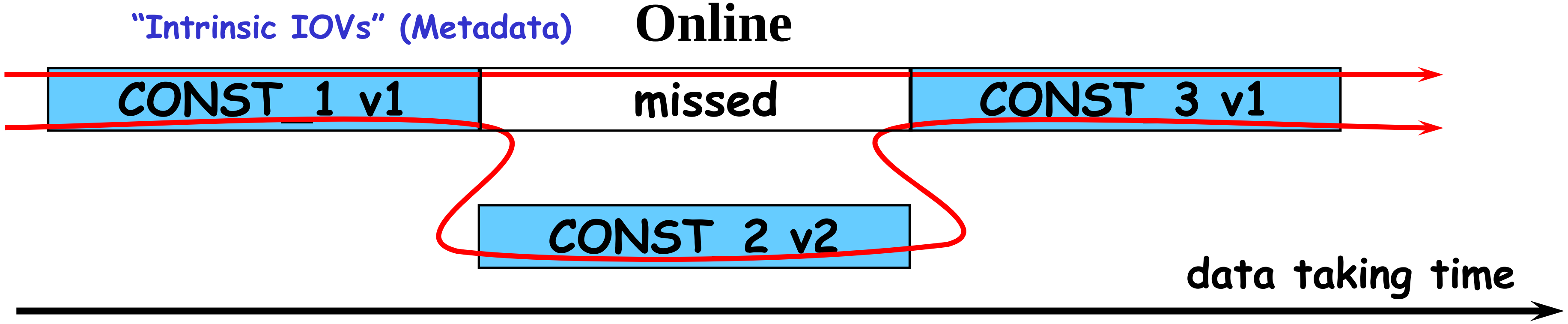
III. metadata, such as tags, can be attached to the IOV service to allow user cataloging, selection of data , and provenance tracing.

IV. the time semantics in the IOV service should be the same as that defined by the framework so that the conditions data are synchronized with the event data.

V. The interface might expose both SINCE and TILL time. But for a given functionality (member function), it requires/returns only one of them.

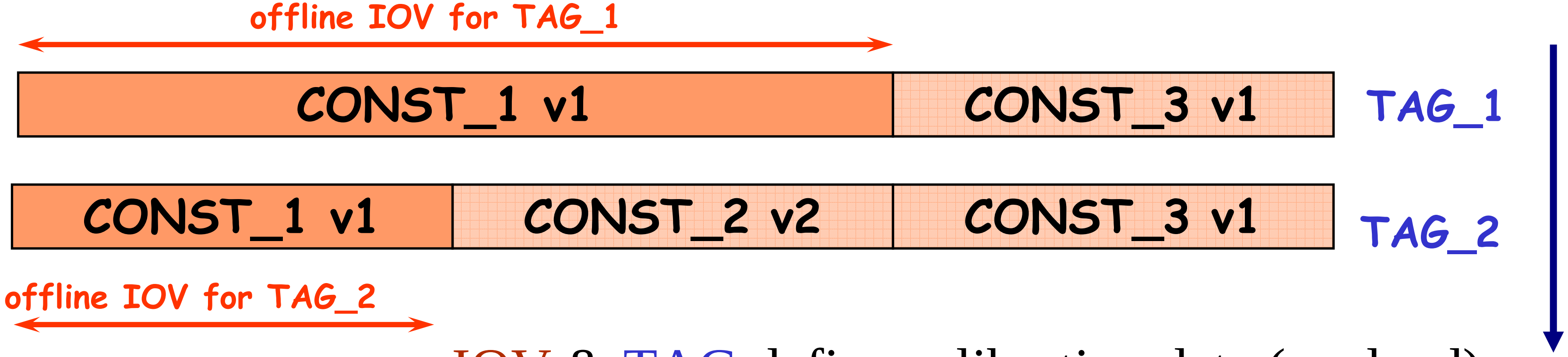
Offline Interval Of Validity (IOV)

Online



"Offline IOV is defined at the offline level"

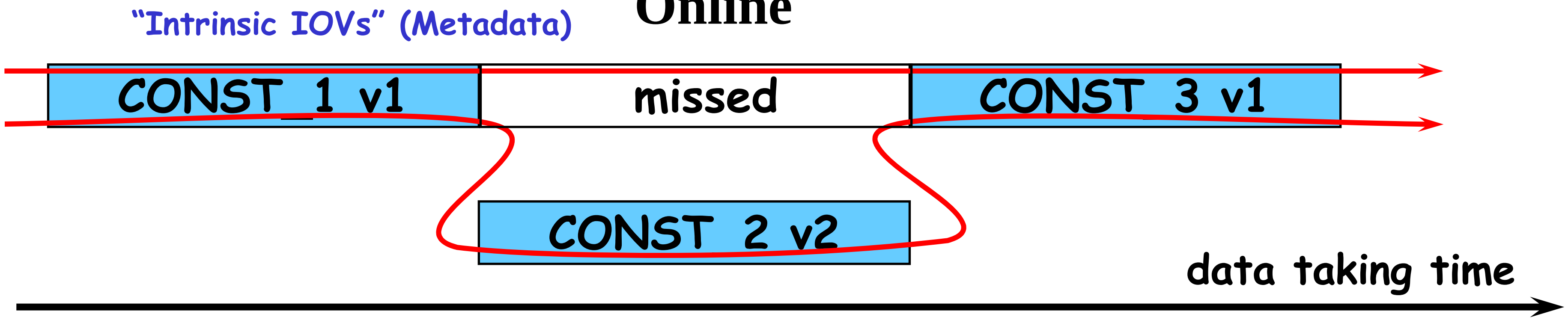
Offline



IOV & TAG define calibration data (payload)

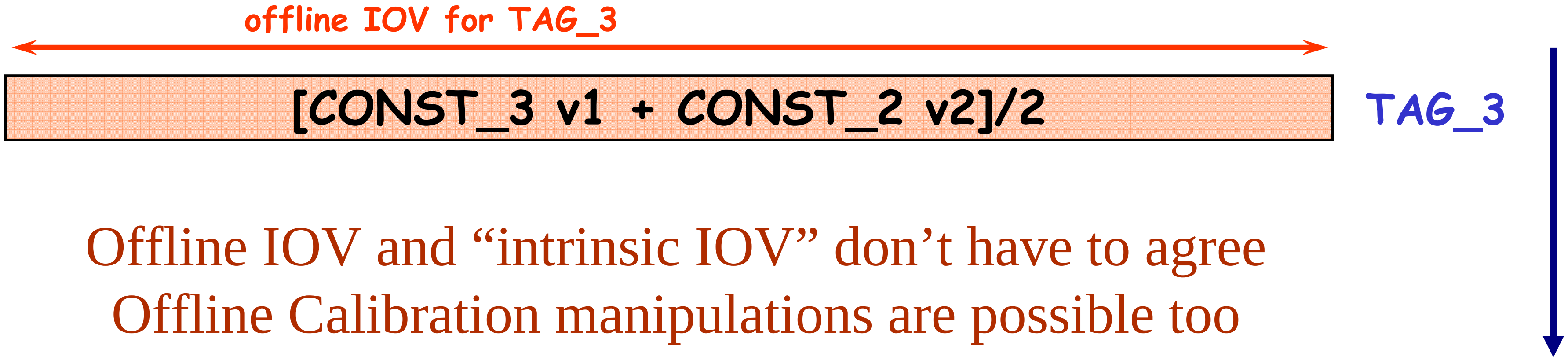
Offline Interval Of Validity (IOV)

Online



"Offline IOV is defined at the offline level"

Offline



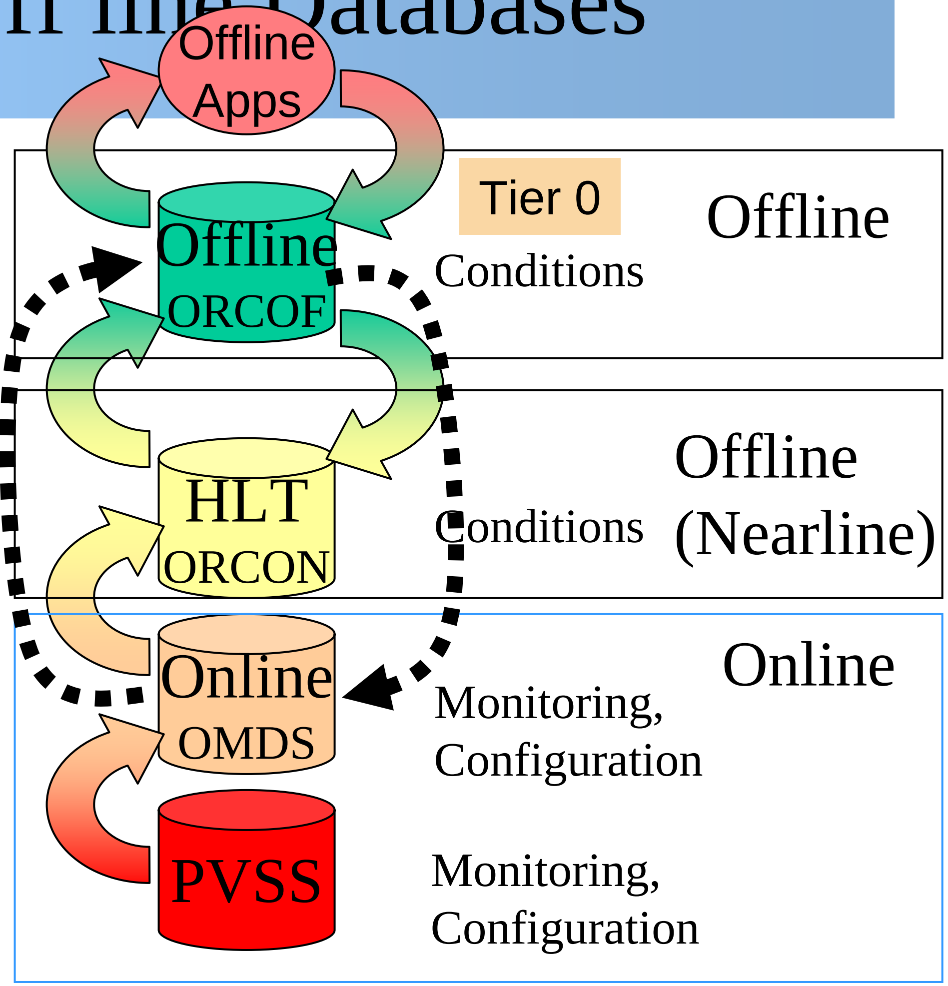
Offline IOV and "intrinsic IOV" don't have to agree

Offline Calibration manipulations are possible too

Etc …

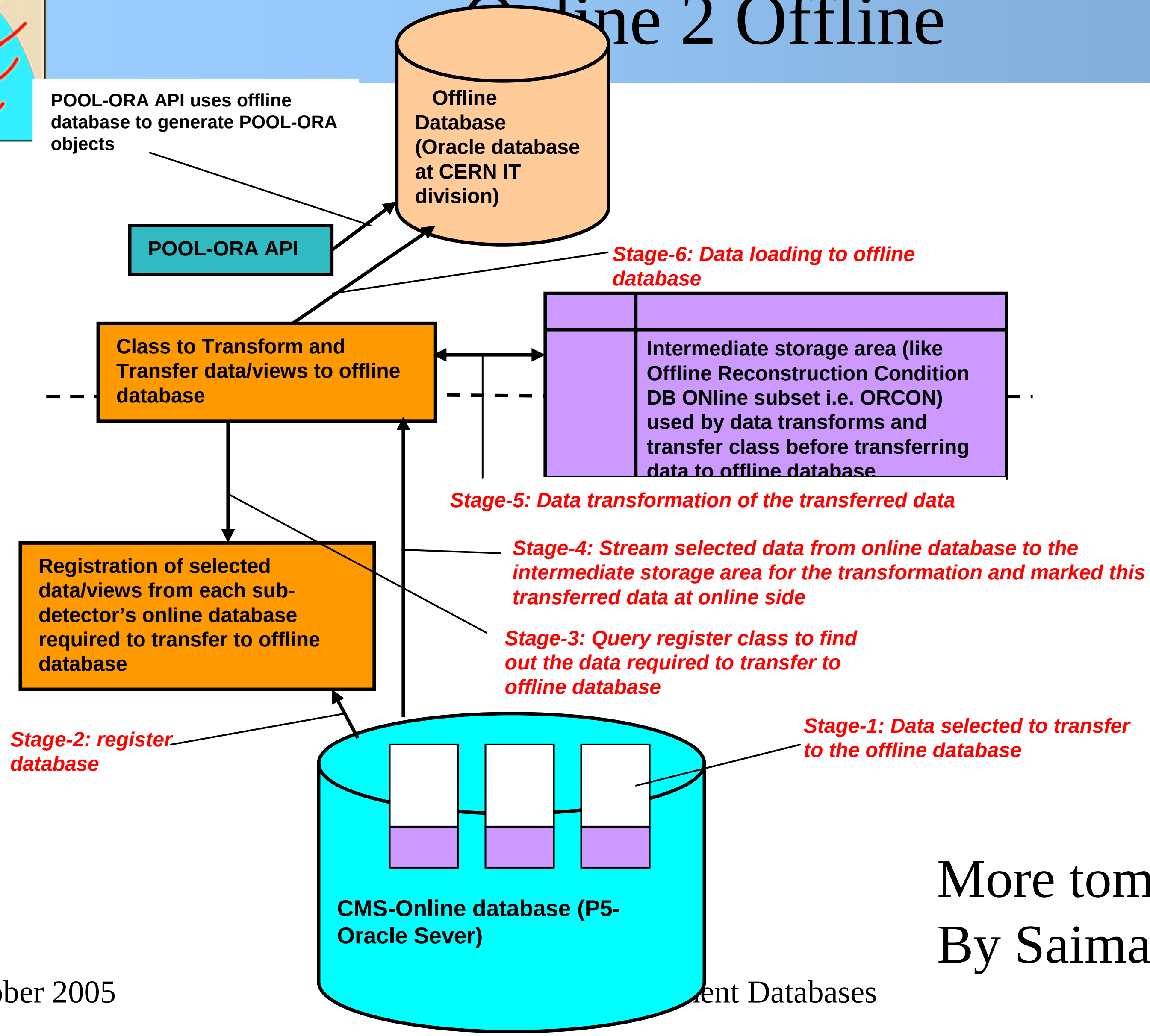
On and Off line Databases

- The offline (ORCOF) DB will run on a production server in CERN/IT. Integration testing (test server) will begin soon.
- The other servers will be located at P5 (current thinking).
- Assumed flow of data is shown by colored arrows. Additional flows are possible (dashed arrows).
- Currently developing mechanism(s) for managing these flows in a fail-safe way.



Conditions=calibration,alignment, slow controls data

Online 2 Offline

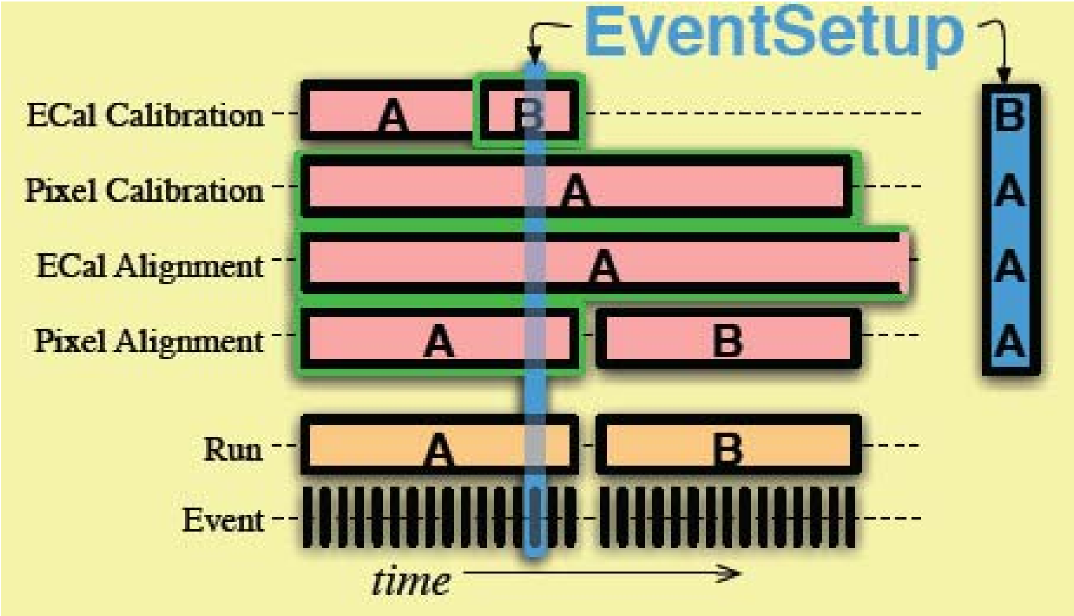


More tomorrow in talk
By Saima A.

EDM Framework EventSetup

- Provides access to “non-event” data needed to properly process an event.
  - Conditions (configuration, calibration, etc.)
  - Geometry and alignment
- Guarantees that the data returned is appropriate for the event being processed.
- Manages the “non-event” data C++ object’s lifetime.

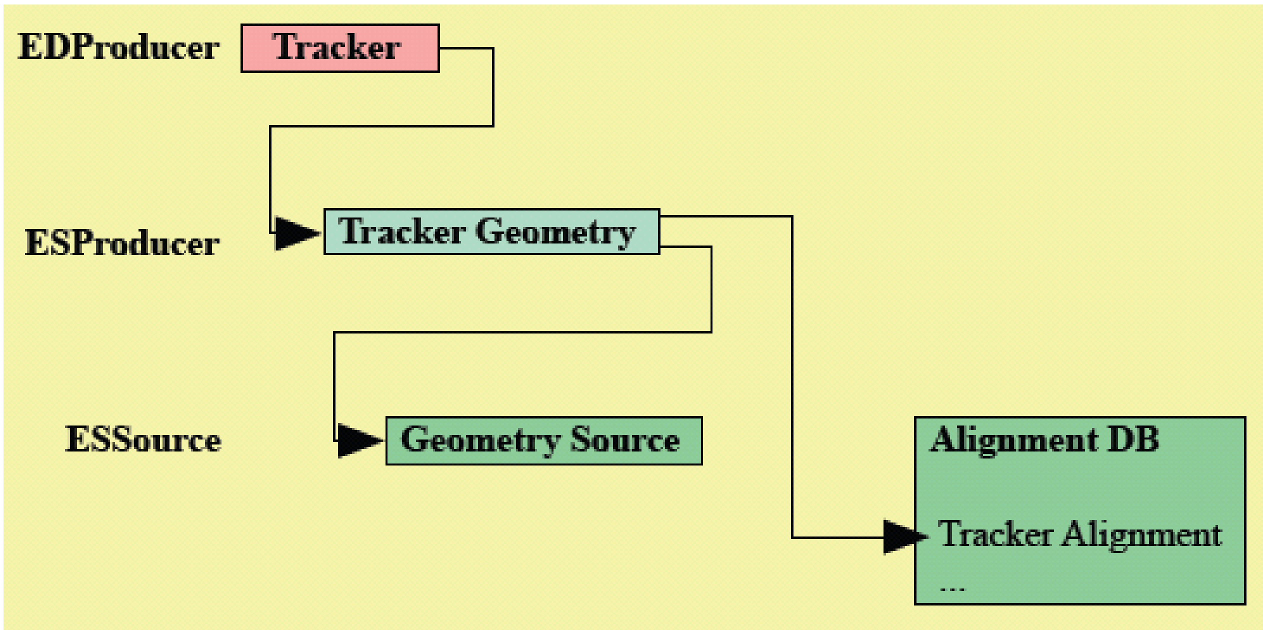
Model



- Provides a unified access mechanism for non-Event data
- **EventSetup** "snapshot" of detector at an instant in time
- **Record**: holds data with same interval of validity
- Not a new idea: has been used by CLEO experiment since 1998

Data Retrieval

- To a user, EventSetup appears to have all its data loaded
- To avoid unnecessary computation, data is retrieved on the first request



POOL-ORA for Offline

- POOL-ORA (Object Relational Access) was chosen to be the offline model for non-event data access.
- It is an LCG-AA product and will be used by the other LHC experiments also.
- POOL-ORA provides an object (as in C++) to relational (as in DB) mapping. The schema in the DB is “generated” based on the “shape” of the object.
- Various tools are being provided that will facilitate storing and managing POOL objects.
- There is an IOV management model that is in proto-type now and interfaced with the EventSetup.
  - Released together with the new framework software
- More work required to armonize EventSetup and persistent organization of IOVs (naming, versions etc)

Data Objects

- Data Objects must be described for each kind of non-event data.
- The description of the data object consists of its C++ class in a header file.
- This information is used by the POOL API to generate the schema, and subsequently store the data.
- Try to define few common classes
  - Geometry: CMS geometry compact view (CMS Note 2004-011)
  - Alignment: Vector of position and Euler angles
  - Calibration: Vector of value and error.
  - Channel ID: Vector of channelID

Data Objects

Geometry Service:

⇒ Use CompactView

Alignment Condition Service:

⇒ Alignment object:

ID	Δx	Δy	Δz	$\Delta\alpha$	$\Delta\beta$	$\Delta\gamma$
0	xx	yy	zz	xy	xz	yz
1	xx	yy	zz	xy	xz	yz
….						
N	xx	yy	zz	xy	xz	yz

Calibration Service:

⇒ Calibration object:

ID	Value	Error
0	val1	err1
1	val2	err2
….		
N	valn	errn

Channel ID Service:

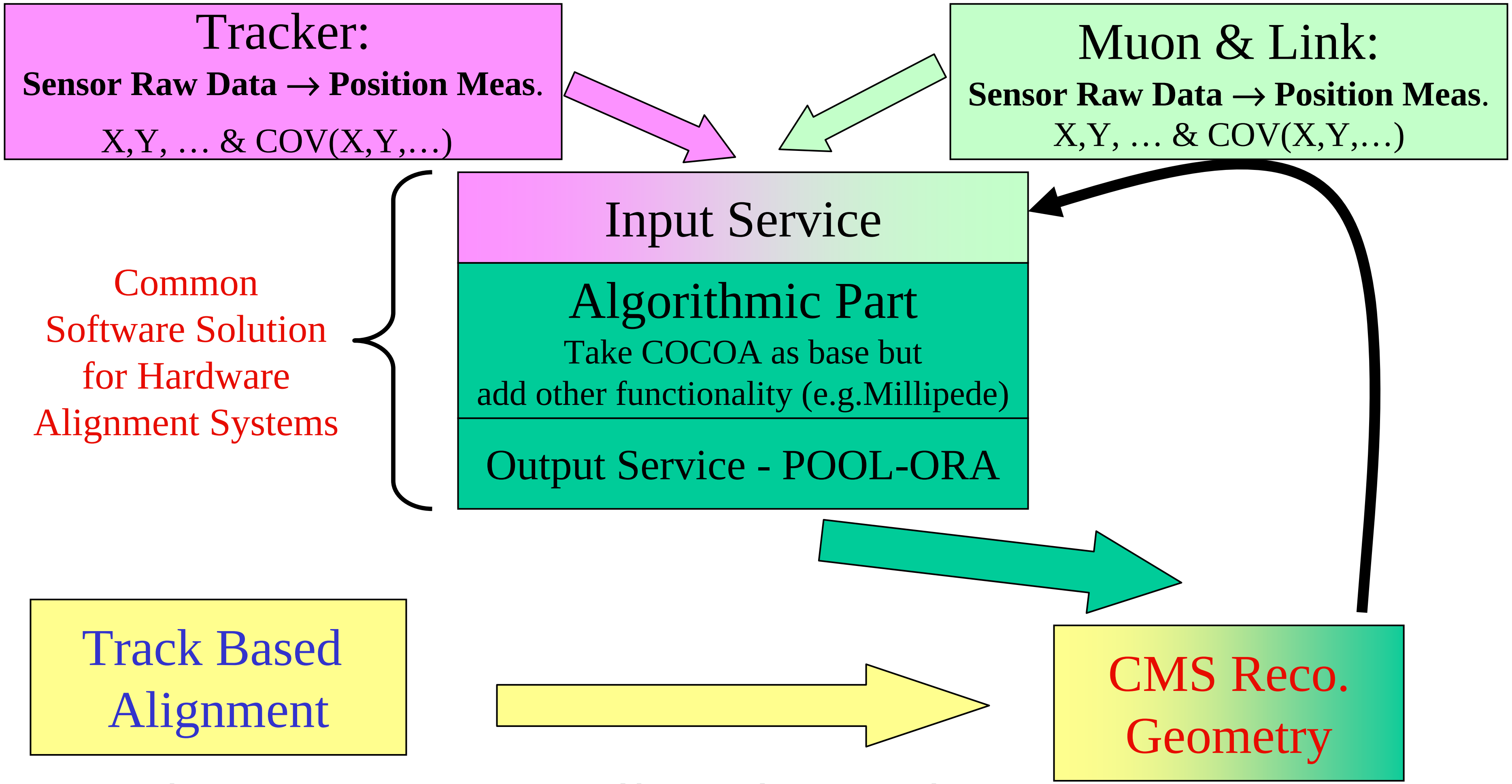
⇒ Channel ID object:

ID	ChannelID
0	ChanID1
1	ChanID2
…	
N	ChanIDn

Offline Calibration

- Ultimate resolution will be achieved only after accurate offline calibration
  - Essentially an analysis task
  - Baseline is to use the Event-Processing framework
  - Output back in ORCOF after validation of results

Example: Alignment



A First Implementation Plan

"online"

Construction Info
e.g. survey measurements & errors

Position Measurements
from Sensor Raw Data

"O2O"

Provide survey result via the O2O in form of an POOL-ORA alignment object

Provide sensor positions using event data format to ensure the full functionality of the framework - already the case for the tracker

Ideal Geometry via ESSource

POOL-ORA alignment object containing survey results

"Event Data" to be used "in CMSSW"

Pool-ORA ali. Object containing latest ali.

Geometry:

X_{ideal}
$+ \Delta X_{survey}$
$+ \Delta X_{ali.}$

Input as standard Framework Module
=
EventSetup +Event

$X_{HA\ system}$ & Position Data

A First Implementation Plan

Algorithmic Part and Output Service

Input Service

HA Geometry **&** **Position Data**

Algorithmic Part

Use COCOA as base but add other functionality like Millepede

Same as standard reconstruction algorithm

Output Service

POOL-ORA Ali. For reconstruction

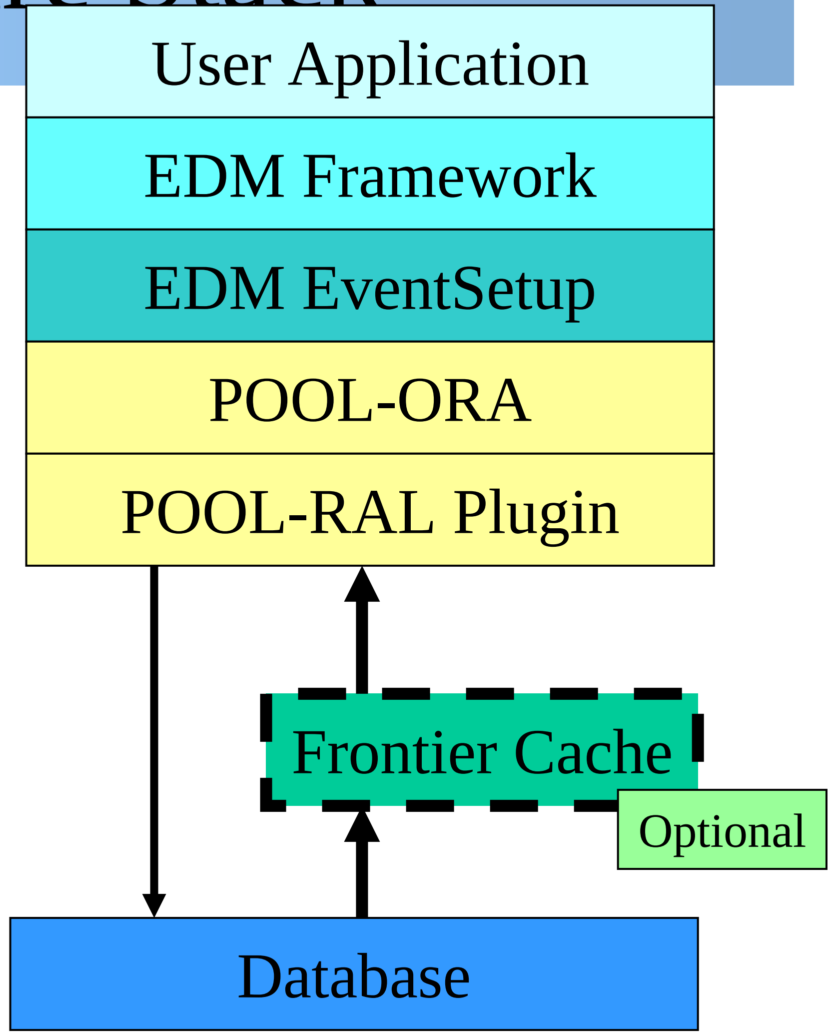
POOL-ORA Ali. For internal use

Track Based Alignment Pool-ORA ali

CMS Reco. Geometry

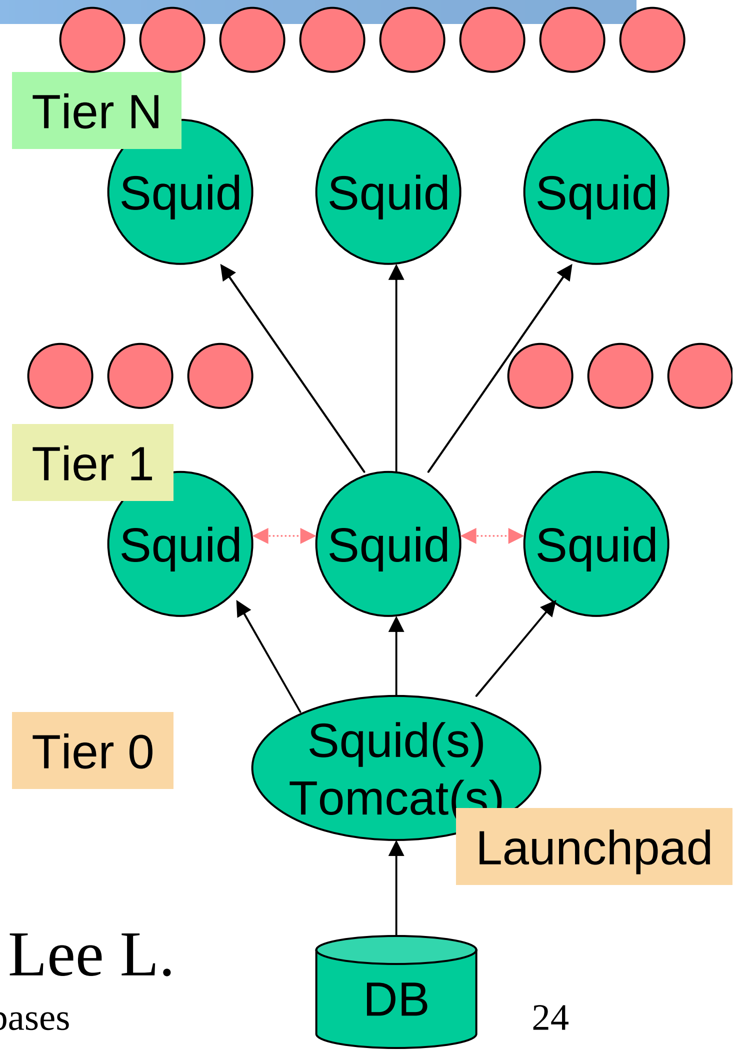
Software Stack

- EDM EventSetup assures the correct non-event data is accessed and available for the user application.
- POOL-ORA (Object Relational Access) is used to map C++ objects to Relational schema.
- A POOL-RAL/FroNTier-Oracle plug-in is used to to enable a middle-tier proxy/caching service for read-only access.
- The central DB is ORACLE, but other technologies can be used for testing (MySQL, SQLite, …)



N-Tier Deployment

- Redundant Tomcat + Squid servers are deployed at Tier 0 (launchpad).
- Squids are deployed at Tier 1, and Tier N sites. Will be an "edge service" for Grid computing centers. Configuration includes:
  - Access Control List (ACL)
  - Cache management (Memory and Disk)
  - Inter Cache sharing (if desired).
- Applications needing access to conditions data are configured with a list of servers, and proxies. For Example:
  - FRONTIER_SERVER1=http://host1.domain.net:8000/Frontier export
  - FRONTIER_SERVER2=http://host2.domain.net:8000/Frontier export
  - FRONTIER_PROXY1=http://proxy.domain.net:10000



More tomorrow in talk by Lee L.

October Milestones (1)

- Our goal is to have initial solutions for Calibration and Alignment storage and access, for most sub-detectors, in place by October 2005.
- Sub-detectors must define the calibration and alignment data they need, and map it into and out of the POOL repository.
- Access from the EDM framework will be through EDM EventSetup plug-ins provided for/by each sub-detector.
- All offline sub-detector applications will use POOL for storing and retrieving conditions data

DONE!

October Milestones (2)

- A plugin at the POOL-RAL layer enables a FroNTier caching layer between the POOL client and DB server. This work is done in collaboration with LCG3D.
- Continued Database to EDM (and user application) testing (A.K.A. E2E testing) to exercise the EDM and database.
- Establish Online to Offline transfer tools, configuration, and procedures. Expand on these as need is understood.
- Use IT/DB development server initially. Schedule test system for dedicated testing. Develop plan for production needs.

In progress (more tomorrow)

Beyond the Oct. Milestone

- As additional EDM plug-ins are developed integration will begin.
- Build a simple Framework application that retrieves non-event data for many sub-detectors.
- Load a significant amount of test data in to the database and exercise its access.
  - Simulate a few month's worth of data for all detectors.
  - Simulate the expected load for world wide access to this data for various patterns of reconstruction and analysis.
  - Simulate the HLT data and access needs
- Additional applications are anticipated next year for Trigger and Luminosity bookkeeping.

Fall 05 Testing Plan

1. Load estimated data for ~1 year's running (~500GB)
2. Exercise various loading and access patterns to simulate stable operations and worst case scenarios. Limited distributed deployment (eg CERN, FNAL and HLT Farm @ CERN).
3. Deploy Squid servers to Tier 1 and Tier 2 centers. Repeat testing under various loads and access patterns.

Non EDM Dependent Activities

Example of online to offline transfers (simple)

Example of online to offline transfer (complex)

Define objects for Geometry and Alignment (agree on DDCompactView for E2E Test)

Implement first Full version of POOL-ORA model for performance test on LCG/3D Infrastructure

First implementation and testing of FroNTier plugin for POOL-RAL

Additional monitoring features for client and FroNTier cache.

A larger collection of offline conditions is defined then implemented.

Develop various POOL Mapping and transfer tools

Large Scale Testing: multiple objects, multiple jobs, writing, reading,

EDM Dependent Activities

First E2E test results for Alignment and Geometry Service

Various End to End Test (E2E Test) activities to further scrutinize and improve the User to DB chain

Include Calibration Service in E2E

| May | Jun | Jul | Aug | Sep |

Assumed Framework development

EDM EventSetUp (Context)

First EDM “main” for E2E Partial functionality

Reconstruction software w/ EDM pieces used for tests. ORCA dependencies are replaced with EDM as they are made available.

CMS Calibration, Alignment, and Geometry Service Schedule

FroNTier plugin for POOL-RAL

Online to offline transfer testing P5 to IT/DB

Additional monitoring features for client and FroNTier cache.

Preparation for Cosmic Challenge
All detectors using calibration service

Large Scale Testing: multiple objects, multiple jobs, writing, reading, WAN access

Test evalu-ation

Integration and testing

Cosmic Challenge

CMS offline conditions DB for cosmic challenge on Production ORACLE Server in Tier0

offline conditions for most sub-detectors implemented.

Additional deployment to Tier 1 and Tier 2 centers

Oct | Nov | Dec | Jan | Feb | Mar

CMS Framework development

Reconstruction software w/ EDM pieces used for tests.
ORCA dependencies are replaced with EDM as they are made available.

Conclusions

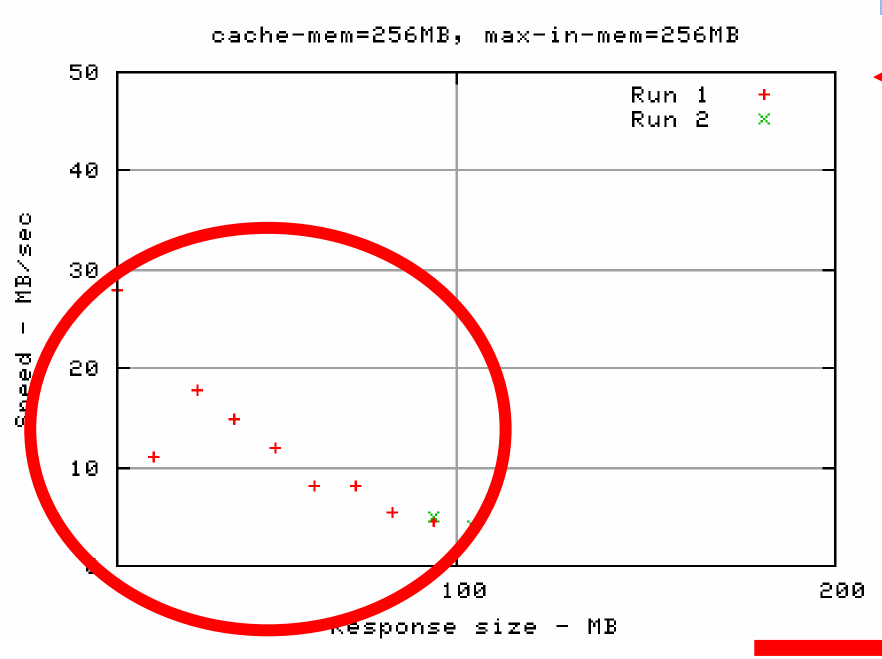
- Plan to have offline non-event data system ready by October (DONE!).
- POOL is used as the model for all offline conditions data, with ORACLE as the central repository.
- E2E testing will continue, and online to offline testing will soon begin.
- Integration and stress testing will begin in the fall as the software nears completion.
- Difficult to estimate realistic data volume and rates due to uncertainties in detector stability in LHC environment
- Collaboration among Software, Computing, and Detector groups, and with LCG/3D, is important to assure the system will meet our needs for the magnet test, and beyond.

The End

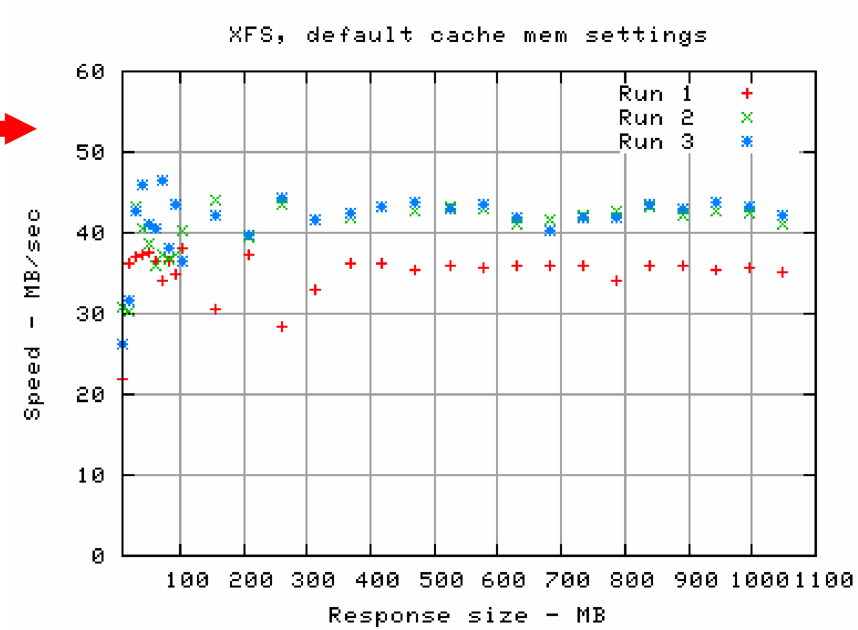
CMS HLT: Challenging Environment

- The High Level Trigger Farm is is a very interesting environment.
  - 1000 nodes, running ~4000 processes
  - Object sizes range up to several hundred MB.
  - Near real-time demands for new object caching.
- It has not been established yet that the Frontier approach will be used, however it is attractive.
- Concerns:
  - Will performance be sufficient for large data objects?
  - Is reliability sufficient under the heavy load?
  - What are the hardware and configuration needs?

Initial Squid Tests



- Attempting to use a large Squid memory cache fails miserably.
- cache_mem 256MB maximum_object_size_in_memory 256MB
- Obviously, memory cache is not designed to work with big objects.



- Performance much better when NOT using Squid memory cache.
- In this test cache_dir was created on XFS disk partition.
- Results are 7 to 10 MB/sec better, compared to Ext2, with large RAM and good disk hardware XFS can perform even better.

Needed Monitoring

- Client-Level: Client library (for POOL) to send access info through UDP to central info collector. Need to connect this to MonALISA.
- Squid-Level: SNMP interface is provided with SQUID. MRTG works well, also works with MonALISA.
- FroNtier Server Level: Log files or additional interface could be developed. In production, this tends to not be a hot spot, but could be during development and ramp-up.
- Database Level: OEM (Oracle Enterprise Manager) may be sufficient. Probably would like some additional stats to monitor trends.