Computing in High Energy and Nuclear Physics

Prague | Czech Republic | 21 - 27 March 2009

An Integrated Overview of Metadata in ATLAS

Elizabeth Gallas,

Solveig Albrand, Richard Hawkings, David Malon, Eric Torrence

on behalf of the ATLAS Collaboration

at the

CHEP 2009 Conference

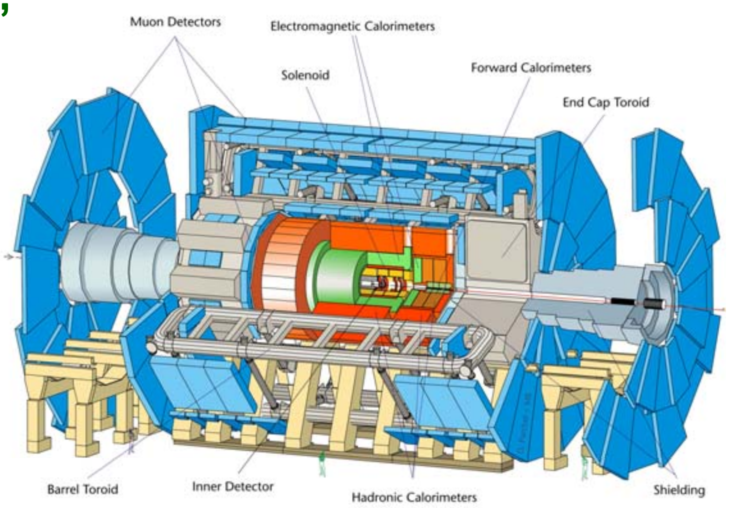
Prague, Czech Republic

March 23, 2009

UNIVERSITY OF OXFORD

Outline

- Technically: Metadata is “data about data”
- Focus today on particular Metadata
  - From data taking
  - Through large scale processing
  - To physics analysis
- Some Metadata connections in ATLAS:
  - In Events … Files … Datasets
  - To “Conditions” data (valid for a designated interval)
- Metadata and the ATLAS Computing Model
  - Role of Metadata in meeting computational challenges on the Grid
- Metadata for People:
  - How do Users find events of interest ?
- Other uses of Metadata in ATLAS
- Status and Conclusions



ATLAS: Data Path

1. Collision rate = 10^9 Hz p-p collisions (at design luminosity)
2. Trigger system selects 200 collisions (“events”)/sec for offline analysis
3. Raw event size (1.6 Mbytes/event, >1PByte/year) contains
   - A set of measured amplitudes from an event dependent subset of the detector’s 100M channels
   - A select set of derived quantities from the trigger system
4. One/More iterations (at various locations on the ATLAS grid):
   - Calibrate, align, …optimize using known processes, simulation: frequently from different sets of events recorded with different triggers
   - Reconstruct fundamental quantities representing physics objects
   - Analyze events (physics objects and global measurables)

- At each of these stages (1-4),
  - Metadata is recorded and used for a variety of purposes

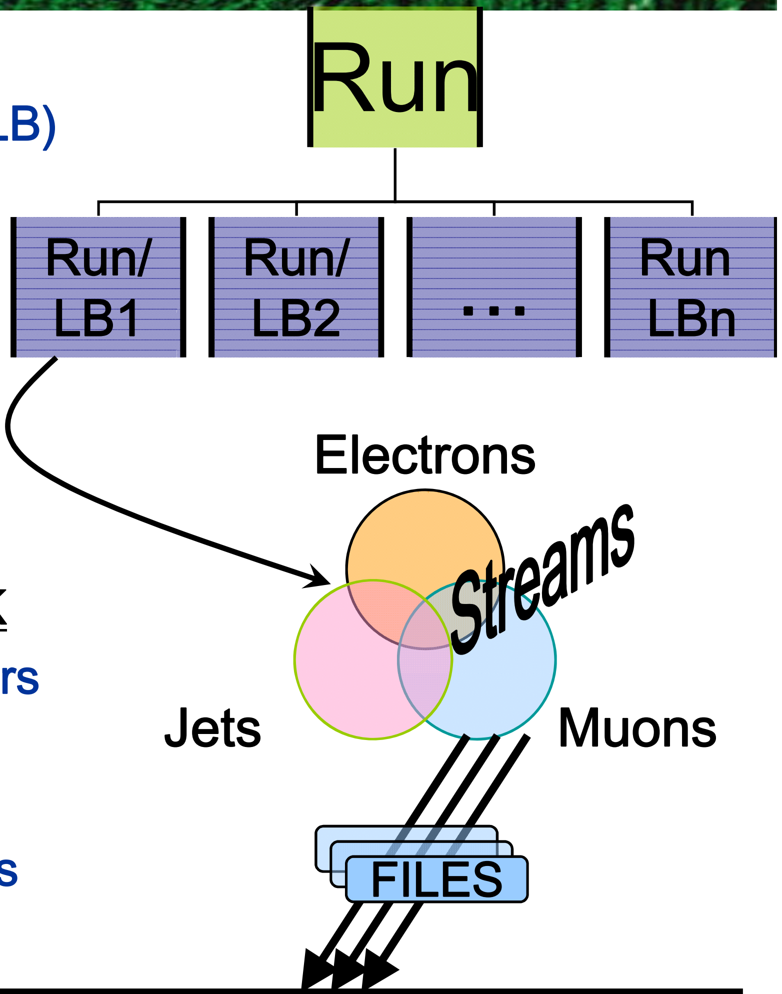
The Luminosity Task Force and the Metadata Task Force: Inventoried the Metadata required for analysis connecting stages 1 THROUGH 4

From Online Runs → Offline Datasets

- Events recorded in Runs (~hours)

  Divided into one/more “Luminosity Blocks” (LB)

  ~minutes, with known start/end time
- Events written to one/more Streams
  - Based on trigger decision(s)
- Streams are written to files
  - On LB boundaries
- Files are processed into Datasets
- Processing of Datasets is defined by a TASK
  - Task definitions include Metadata pointers
    - To versions of released software
    - And other configuration information…
  - Produce new Datasets in various formats defined in the ATLAS computing model



Dataset with name format:

Project.runNumber.streamName.prodStep.dataType.Version

AMI: Metadata about Files and more

"MIF (Metadata in Filenames) is evil" – Shaun Roe

- Fragile: changing standards, spelling/case variations, moving files
- Closed: Addition of metadata is not possible
- Limiting: It probably doesn't contain everything you want.

- But we all do it at many levels … we must recognize the areas where Naming Rules are important with a view to the long term
  - Datasets follow strict nomenclature rules (ATL-COM-GEN-2007-003)
    - Components of Dataset name structure are Metadata pointers

These rules <u>and more</u> are included in <u>AMI – The ATLAS Metadata Interface</u>

- See presentations at this and past CHEP conferences
- Infrastructure and Interfaces to searching for available ATLAS Datasets
  - Including whether they are complete (valid) or failed (and reasons)
- Configuration of tasks producing those Datasets
- Provenance of Datasets: trace parentage back to the data's origin
- Links to:
  - Datasets in the DDM (Data Distribution Management)
  - Links to Reports on Run and Trigger information stored elsewhere

A lesser evil: Metadata in events… files… datasets…

- The event records themselves know just enough to connect to Metadata at a higher level (their Run, LB, Event #, ~timestamp…) but they have a limited world view.
  - They know which trigger bits were satisfied,
    - But they don't know the requirements behind those bits or their names.
  - They may not contain muon tracks, but they don't know why
    - Were there no muons ? Or was the muon system disabled ?
- When Events are put into Files, "File Level Metadata" is added including
  - The Run/LB ranges from which the sample was derived
  - Some configuration information
    - Geometry
    - Trigger
    - …

Obviously, file volume concerns keep the level of "In File Metadata" in check. The balance between Content / Volume / Backward compatibility is evolving.

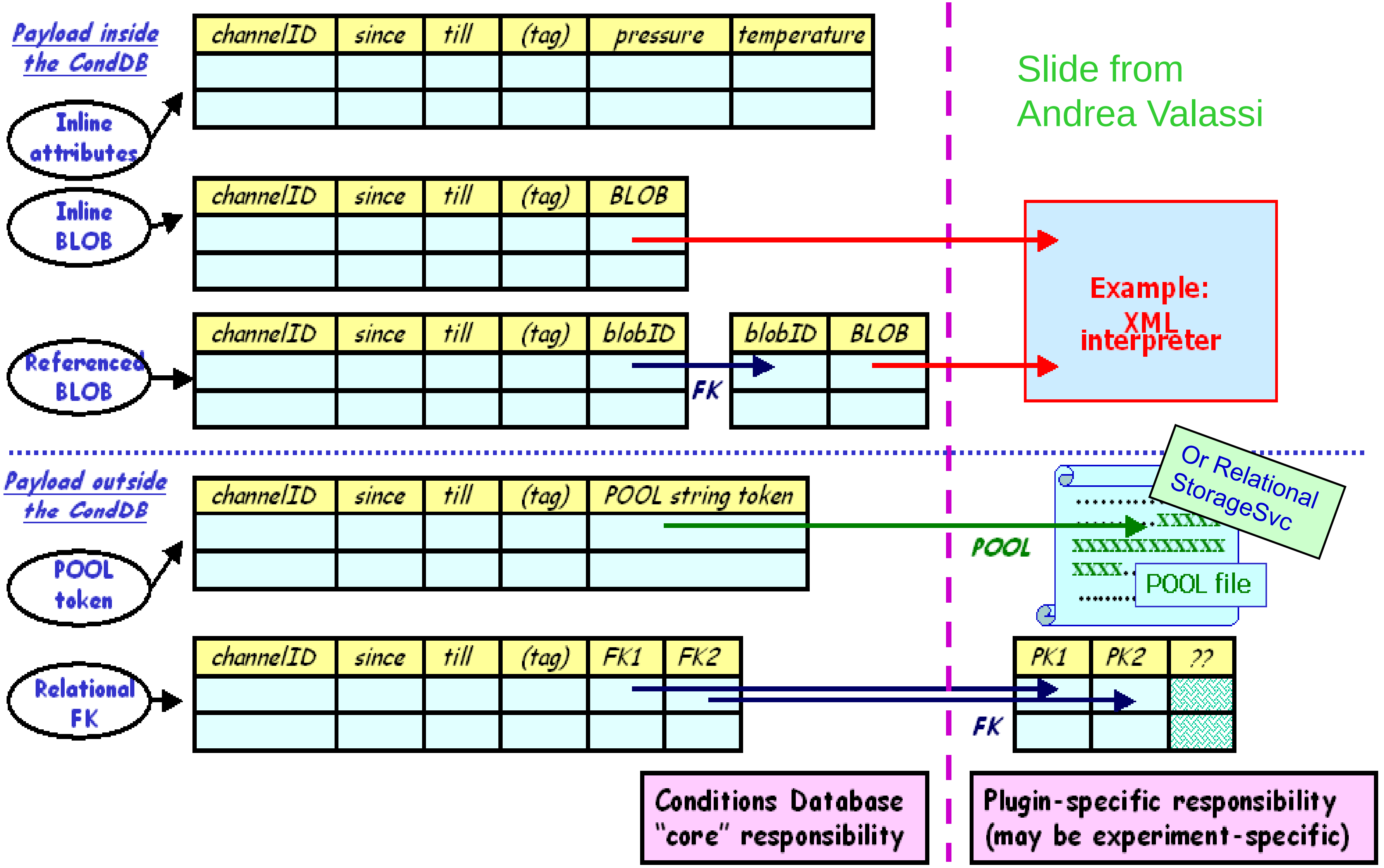
Tasks find a broader view of the "Conditions" behind these Metadata pointers using information in the ATLAS Conditions Database…

ATLAS “Conditions” Data

Subsystems need to store information which is needed in offline analysis which is not “event-wise”: the information represents conditions of the system for an interval ranging from very short to infinity.

- This type of information is stored in the ATLAS Conditions Database
  - Based on LCG Conditions Database infrastructure
    - Using ‘COOL’ (Conditions database Of Objects for LHC) API
  - Allowed interval (the IOV or ‘Interval of Validity’) are based on
    - Timestamps or
    - Run / Luminosity Blocks
  - A wide variety of storage options are available within the Conditions infrastructure
    - Optimized to the type of information being stored
    - Alternatively, the information in the Conditions DB may be a pointer to an external structure (POOL file) or to another table.

COOL conditions DB table examples



Slide from Andrea Valassi

Payload inside the CondDB

Inline attributes

channelID	since	till	(tag)	pressure	temperature

Inline BLOB

channelID	since	till	(tag)	BLOB

Referenced BLOB

channelID	since	till	(tag)	blobID

FK

blobID	BLOB

Example: XML interpreter

Payload outside the CondDB

POOL token

channelID	since	till	(tag)	POOL string token

POOL

Or Relational StorageSvc

POOL file

Relational FK

channelID	since	till	(tag)	FK1	FK2

FK

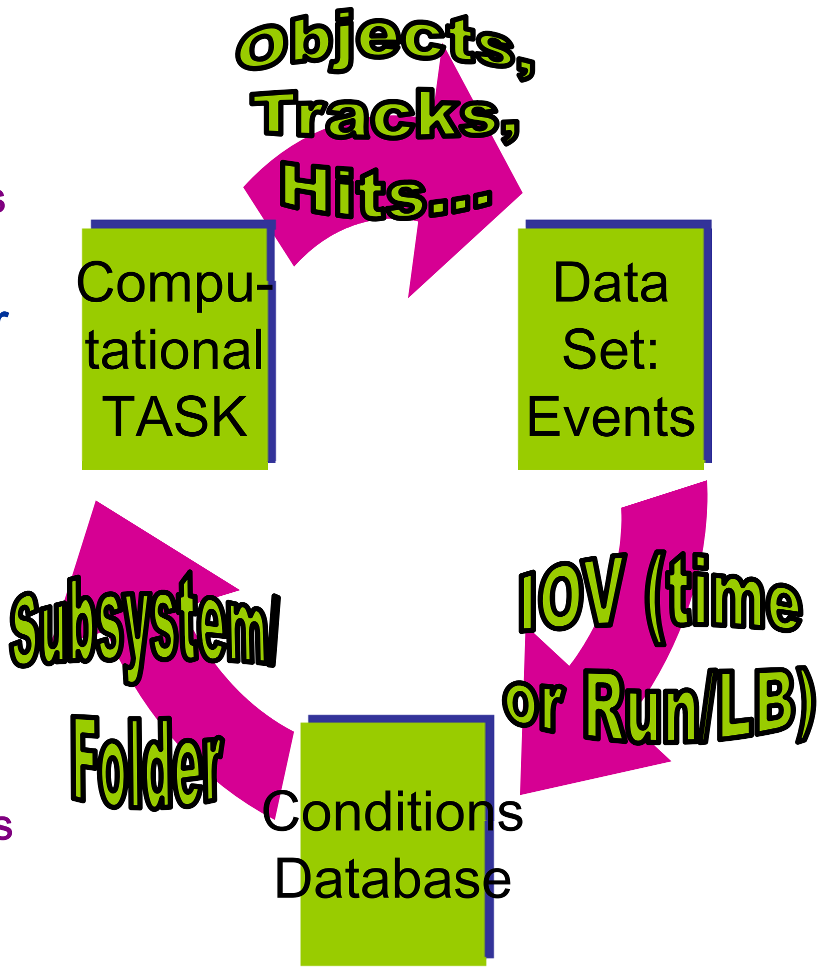
PK1	PK2	??
		▨
		▨

Conditions Database "core" responsibility

Plugin-specific responsibility (may be experiment-specific)

Connecting Metadata: Conditions | Datasets

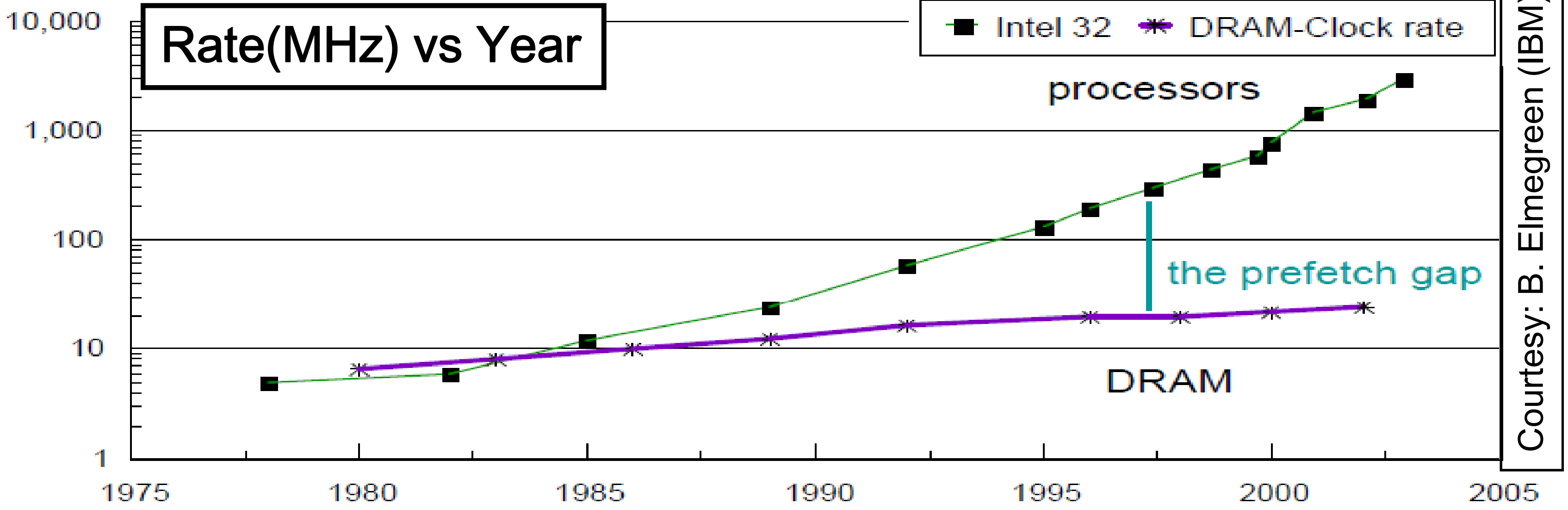
- Metadata of:
  - Dataset index: Run/LB or timestamp
    - Since Datasets are always based on files which respect Run/LB boundaries
  - Conditions Database indexes: IOV, Conditions DB Tag, sub-system, folder
    - Where the IOV can be based on timestamps or Run/LB



- Computing Model: The data processing/analysis chain is based on
  - Task definition (a program)
    - Creates new entities for an event based on the existing entities in events in the input dataset. Creates output results.
  - an ATLAS Dataset and
  - its associated “Conditions” data

Getting Conditions data to CPU intensive tasks

- Storage capacity and processing speed per unit cost have increased exponentially while DRAM access speed has hardly improved

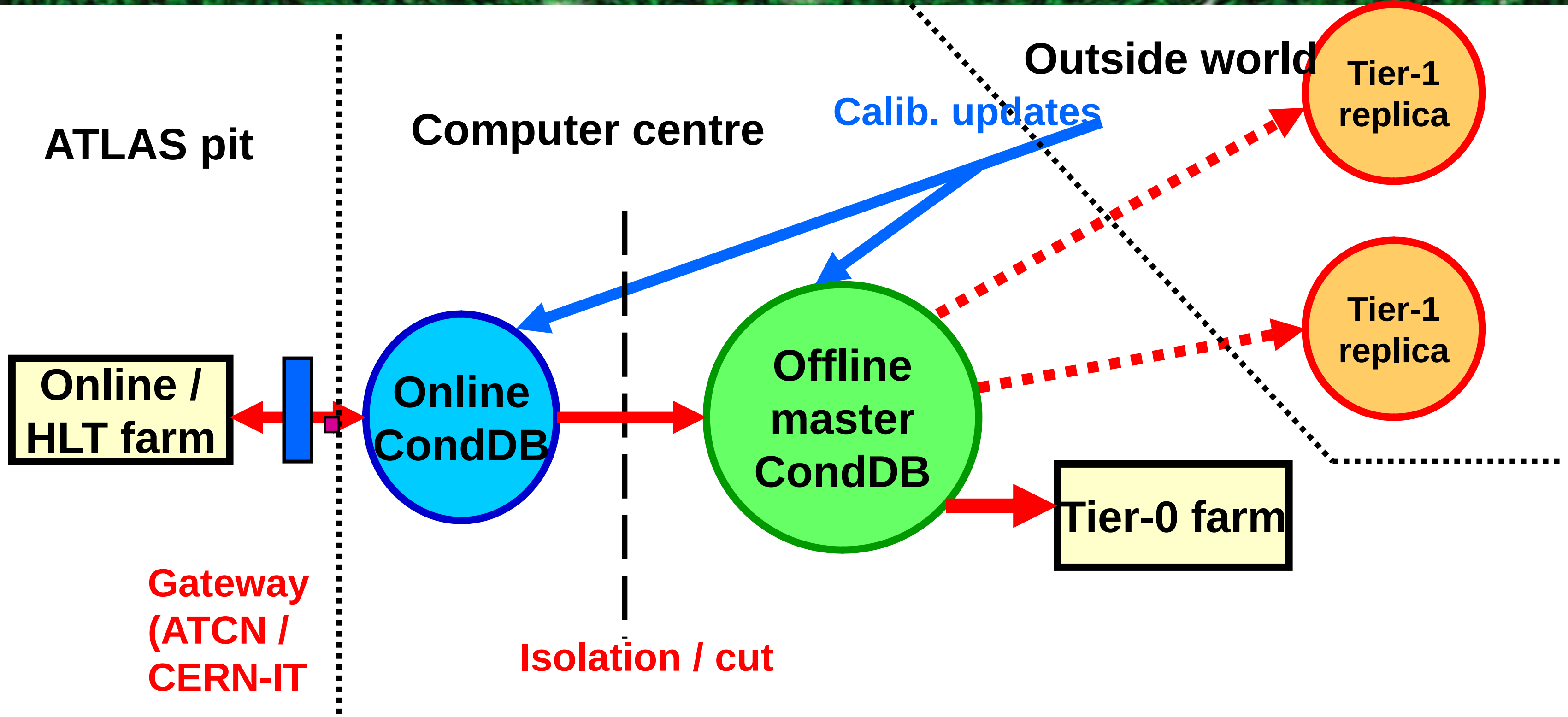


Courtesy: B. Elmegreen (IBM)

- “The prefetch gap” == performance gap between CPU speed and disk access latency: continues to widen

ATLAS uses Metadata to find the data needed by a task to help bridge the gap to facilitate computing intensive tasks (calibration, alignment, processing, reprocessing, analysis)

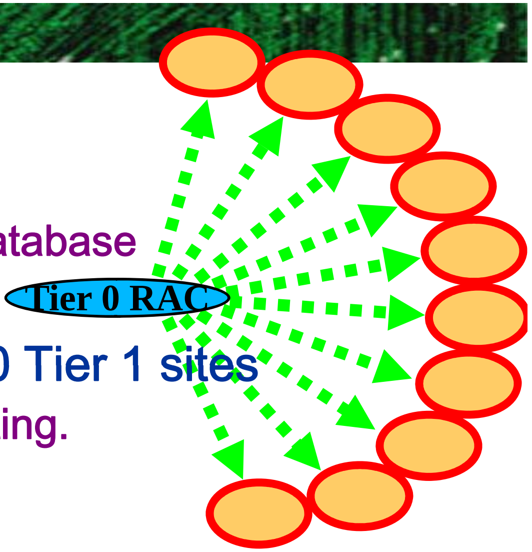
Conditions Database replication: Tier-0 and Tier-1



- Data replicated: using Oracle Streams
  - Can include non-COOL data (subdetector CORAL tables)
- Oracle Streams replication to Tier-1
  - Only that data needed for offline reconstruction or analysis

How does metadata help ?

- In the ATLAS computing model
  - All conditions data needed in offline analysis
    - must be in or referenced in the ATLAS Conditions Database
  - Master copy stored at CERN Tier 0 in Oracle
  - Conditions needed on the Grid is replicated to all 10 Tier 1 sites
    - See R.Walker's talk on details, usage, scalability testing.



2 Cases:

- Pre-knowledge of the input data needed by a job (production tasks)
  - Use Metadata to create local instance of data on/near where the job will be run
- Ad-hoc queries (generally user tasks), where the task is less orchestrated:
  - Tier 1 RACs on multicore servers make ad-hoc queries more performant
    - These are replicas of the Tier 0 database
  - Metadata, as well as the data it points to may be in the database or the Conditions DB reference is the Metadata is used to find the data
  - Metadata is used to retrieve the required data from the closest location

How do ATLAS physicists find events of interest ?

Physicists have broad interests/responsibilities in ATLAS. They need to find and analyze events offline. How do they find events for their purpose ?

Examples:

- Sub-detector experts looking for cosmic ray data
  - And they'd like events with their subdetector engaged …
- Physicist in Group X wants to find events in Runs designated by X to be 'good'

One Solution: RunQuery tool (described by Shaun Roe – this conference)

- Web based system for querying the Conditions Database
- Allows the user to find the Runs of interest satisfying various Conditions (Detectors configured, Detector status flags…)

But what if the user wants to choose data based on event-wise criteria ?

Example:

- Physicist wants to select events with offline electrons with p_T > 30 GeV…

➔ This is the basis for the ATLAS TAGs Application (next slide)

Metadata for Users: ATLAS TAGs

- PURPOSE: Facilitates event selection for analysis
- Available in File and Database formats (Storage: kB/event,>1TB/year)
  - Technical challenges in Poster on ATLAS TAGs distribution/management
- 'TAG Database' Application includes
  - Event-level Metadata produced routinely in data processing campaigns
    - About 200 indexed variables for each event: Identification keys, global event quantities, Trigger decisions, number of reconstructed objects (with their pT, eta, phi for highest-pT objects), Detector status,quality, physics, and performance words….
  - 'Run Metadata' at Temporal, Fill, Run, Lumi-block levels
    - Has potential to add new/improved information (after TAG production)
      - Data Quality assessments
      - Efficiency calculations
      - Luminosity corrections
  - References to Files for back-navigation
  - A variety of supporting tools and infrastructure
- Various components of the ATLAS TAG application are described in other CHEP presentations.

Other uses of Metadata in ATLAS

- I have alluded to, but had insufficient time today, to describe Metadata usage to:
  - Get statistics without reading large amounts of data
  - Determine the status of various processes
  - Looking for event/file losses
  - Checking data integrity
  - Job management
  - File and Dataset management
  - Get Dataset provenance (history of processing)

Status

- Many pitfalls exist in the dependence on Metadata
  - Metadata sources
    - may not be currently or consistently filled (cross check inputs!)
    - may be a ‘mashup’ of information from various systems
    - may change with/without notice
  - The scope of Metadata usage must be well defined
    - As well as the Business logic of each application using it
- Work is ongoing
  - To deliver integrated metadata services in support of physics analysis,
  - On infrastructure to connect seamlessly the Events to the Condition intervals
  - To incorporate Good Run List definition and management, and associated tools
  - To address identification and consequences of processing failures

Conclusions

- This talk reflects aspects of metadata handling/usage in ATLAS with a focus on the metadata along the event analysis chain
  - Beam conditions and data taking → Data processing (and re-processing) → Data selection (and rejection) → Data analysis to formulate measurements
    - In a way that is reproducible and is extendable across larger statistical samples to formulate more significant conclusions
- Metadata usage and performance
  - Is currently being exercised with commissioning and simulated data
  - Ultimately awaits the trial of reality
    - Actual pp collisions in 2009 / 2010
    - The larger volumes of data way over yonder
- Successful Metadata usage 'takes a collaboration' !!!
  - Each of our collaborators contributes to it
  - At every level
    - From the Metadata itself
    - To the data behind the Metadata