African School of Fundamental Physics and Applications

Modern SQL and NoSQL database technologies for the ATLAS experiment

Dario Barberis

Dario.Barberis@cern.ch

(Genoa University/INFN)

On behalf of the ATLAS Collaboration

Topics

- A bit of history
- Database usage by ATLAS in LHC Run2
- Evolution of database technologies, infrastructures and applications for Run3 and beyond

Topics

- A bit of history
- Database usage by ATLAS in LHC Run2
- Evolution of database technologies, infrastructures and applications for Run3 and beyond

Long ago ……

- When software developments started for ATLAS about 20 years ago, the generic word "database" practically referred only to relational databases
  - With only a few exceptions (this would be another talk, not for today!)
- There were very few options to store largish amounts of structured data:
  - Oracle (supported by CERN-IT, including license costs)
  - MySQL (in its early stages, not scaling yet but promising rather well)
  - Do-it yourself (files and home-made catalogues)
- So the choice was clear: fit everything into Oracle because of the CERN system-level support, and develop the ATLAS applications to make use of Oracle's tools for performance optimisation
  - Having two expert Oracle application developers for ATLAS obviously helped us a lot

Pros and Cons

- Having only one underlying technology helped to provide a robust and performant central database service, managed jointly by CERN (system level) and ATLAS (application level)
- Many time-critical applications are hosted by the Oracle infrastructure:
  - Conditions database
  - AMI (ATLAS Metadata Interface)
  - ProdSys/PanDA (distributed production system)
  - Rucio (distributed data management system)
  - AGIS (Grid information system)
  - Glance (membership, authorship, speakers etc.)
- All these applications grew in size and complexity with time and are working quite well for the Collaboration current usage
  - Oracle can be VERY FAST if database schemas and queries are well designed and optimised

Pros and Cons

- Having only one underlying technology forced some applications that have no need of relational information into fixed schemas that may be not completely optimal
  - Time-series measurements produced by DCS (Detector Control System) can be more simply represented by time-value pairs
  - DCS data have to be compressed before storing in Oracle because of their huge sizes
- Oracle schemas have to be carefully designed upfront and are then hard to extend or modify
- Data access to Oracle databases from Grid jobs was less than obvious and an interface system (Frontier) had to be adopted to allow concurrent running of over 300k jobs
- Data analytics tools started appearing on the Open Source market that can deal with huge amounts of less structured data

Pros and Cons

- Having only one underlying technology forced some applications that have no need of relational information into fixed schemas that may be not completely optimal
  - Time-series measurements produced by DCS (Detector Control System) can be more simply represented by time-value pairs
  - DCS data have to be compressed before storing in Oracle because of their huge sizes
- Oracle schemas have to be carefully designed upfront and are then hard to extend or modify
- Data access to Oracle databases from Grid jobs was less than obvious and an interface system (Frontier) had to be adopted to allow concurrent running of over 300k jobs
- Data analytics tools started appearing on the Open Source market that can deal with huge amounts of less structured data

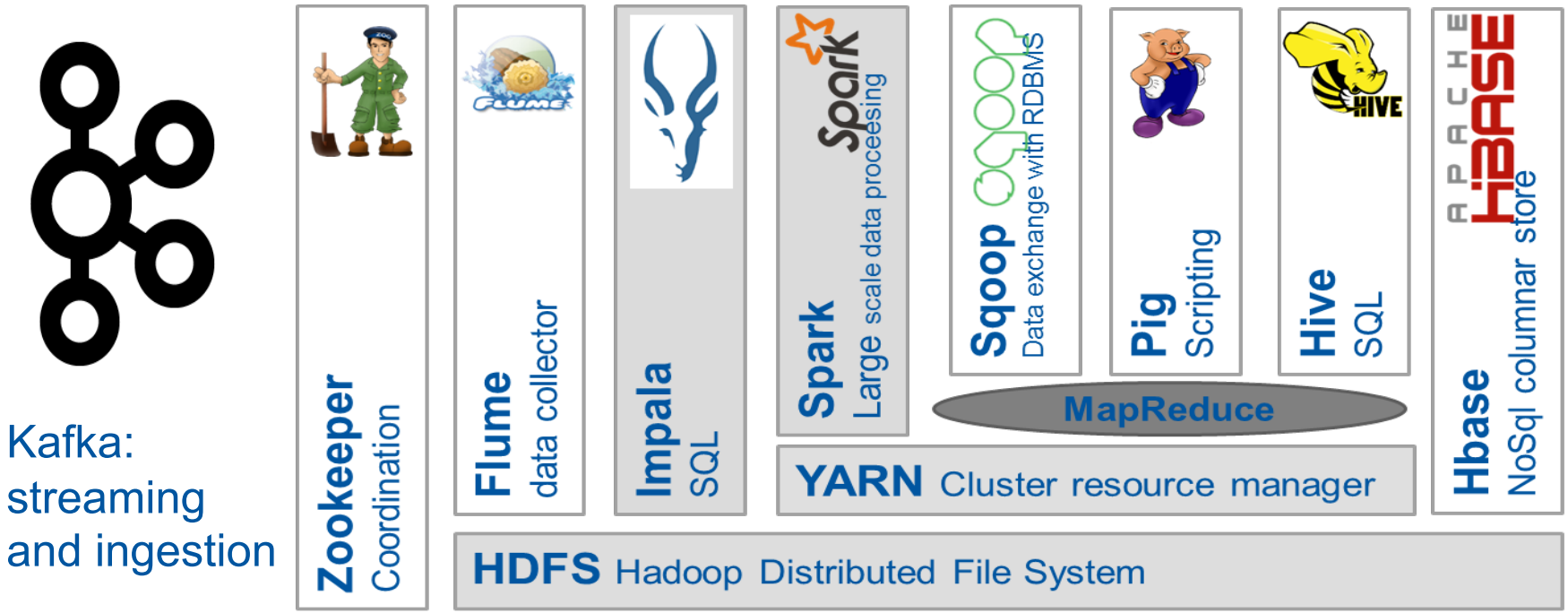
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

Before Run2

- Towards the end of Run1 in 2012 and during LS1 (2013-2014) a number of new structured data storage solutions ("NoSQL Databases") were tested as back-end support systems for new applications
  - Hadoop and the many associated tools and data formats
  - Cassandra, MongoDB, etc.
- Mostly key-value pair or column-oriented storage systems
- The WLCG Collaboration launched a few study groups on new computing technologies, one of which was the "Database Technical Evaluation Group" (DB TEG)
  - The DB TEG recommended CERN to deploy and support a Hadoop cluster for new applications, with all associated tools
  - In fact several Hadoop clusters were set up over the years to avoid destructive interference between different applications, while both system managers and application developers were learning the best practices for design and optimisation

In the meantime

- Growth of the Hadoop ecosystem as installed at CERN-IT:



- Many more tools are available
  - More possibilities
  - More diversification

Luca Canali

Topics

- A bit of history
- Database usage by ATLAS in LHC Run2
- Evolution of database technologies, infrastructures and applications for Run3 and beyond

Conditions Data

- "Conditions Data" are all non-event data that are useful to reconstruct events:
  - Detector hardware conditions:
    - Temperatures, currents, voltages, gas pressures and mixtures, etc
  - Detector read-out conditions:
    - Trigger and detector read-out configurations
  - Detector calibrations:
    - Energy calibration for calorimeters, time-over-threshold for pixels, R-T relations for drift tubes (TRT and MDT)
  - Detector alignments:
    - Relative and global alignment of sub-detectors
  - Physics calibrations:
    - Jet energy scales and resolutions, jet flavour tagging weights, trigger and reconstruction efficiencies, etc.
- All conditions data have associated intervals of validity and (for derived data) versions
- The traditional way to store and access conditions data is through a relational database
  - The COOL API is used by ATLAS for the conditions database except for the physics calibration data

Physics metadata (1)

- Metadata are "data about data"
- AMI (ATLAS Metadata Interface) provides ATLAS physicists with an interface to find the datasets they need for their analyses, including lots of accessory information
  - Number of files, file sizes, number of events
  - Data-taking periods and links to the conditions
- Information is imported from the Tier-0 system, the Grid workload management system (ProdSys/ PanDA) and the data management system (Rucio) and presented in a coherent way
- Back-end storage in Oracle
  - Master at CC-IN2P3 in Lyon, read-only copy at CERN
- Information retrieval through web i/f or using the python client
- The COMA (Conditions/Configuration Metadata in ATLAS) project began based on the following tenet:
  - to collect Run and Luminosity Block wise metadata
  - to store that data in a relational database to facilitate its use by dynamic web interfaces
- Usage of COMA information has grown into many areas.
  - It is now the master repository of ATLAS Data Period information.

Physics metadata (2)

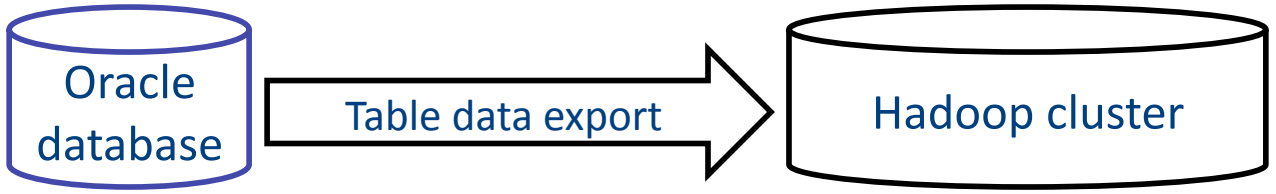
- COMA tables are populated with data from the best available sources
  - Mostly from subsystem specific databases (Conditions, Trigger, Tier-0/SFO databases) or other ATLAS Metadata Catalogs (AMI)
  - Some information in COMA is collected from non-database sources (TWiki, emails, xml files, ...) at fixed points in time.
- COMA tables are in a Relational Database (in Oracle) to which AMI has easy access
- Upload select conditions for runs of "analysis interest" (NOT all runs and not all conditions) and upload conditions related to Runs in COOL tags (with cross-checks), for example:
  - Luminosity in COMA is collected from the the conditions database using the latest/ best luminosity tag version recommended by luminosity experts for each project
  - Conditions DB metadata is collected only for instances which are active in LHC Run 1 or in preparation for Run 2
- COMA tables additionally contain a variety of Refined/Corrected/Derived information to form unique and more effective criteria
  - Information not available in other systems

Distributed computing

- The distributed production/analysis and data management systems produce and need to store a wealth of metadata about the data that are processed and stored
  - Rucio (Distributed Data Management)
    - Dataset contents catalogue: list of files, total size, ownership, provenance, lifetime, status etc.
    - File catalogue: size, checksum, number of events
    - Dataset location catalogue: list of replicas for each dataset
    - Data transfer tools: queue of transfering datasets, status etc.
    - Deletion tools: list of datasets (or replicas) to be deleted, status etc.
    - Storage resource lists, status etc.
  - ProdSys/JEDI/PanDA (Distributed Workload Management)
    - Lists of requested tasks and their input and output datasets, software versions etc.
    - Lists of jobs with status, location etc.
    - Processing resource lists, status etc.
  - Both systems use a combination of quasi-static and rapidly changing information
    - ATLAS runs over 1M jobs/day using 200k job slots and moves 600 TB/day around the world
  - Oracle supports very well both systems if the tables don't grow indefinitely
    - "Old" information is copied to an archive Oracle database and removed from the primary one
    - Accounting information is extracted from the back-up Oracle database and stored in Hadoop for further processing

Offloading from Oracle to Hadoop

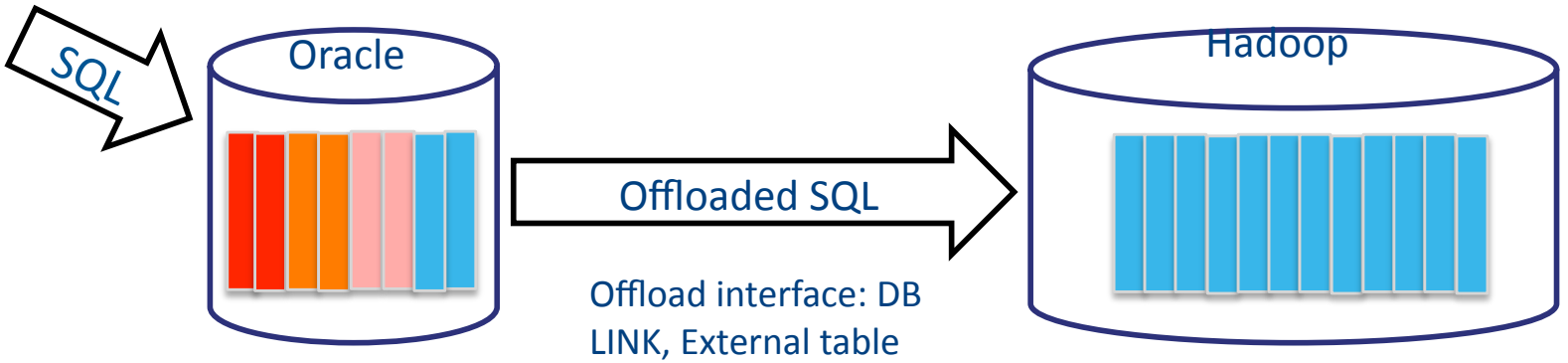
- Step1: Offload data to Hadoop



Apache Sqoop

Data formats: **Parquet**, Avro

- Step2: Offload queries to Hadoop



Offload interface: DB LINK, External table

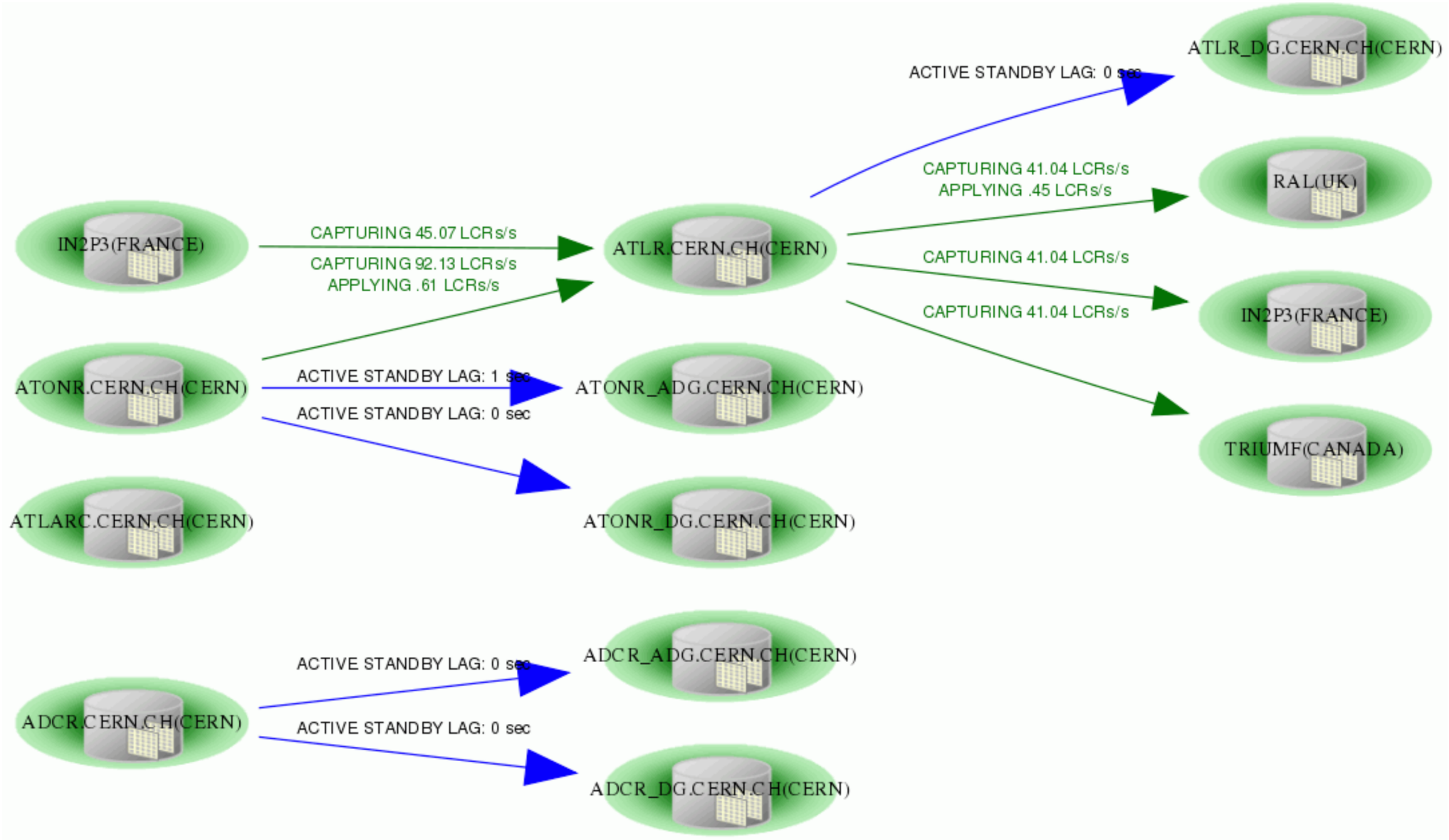
SQL engines: **Impala**, Hive

Luca Canali

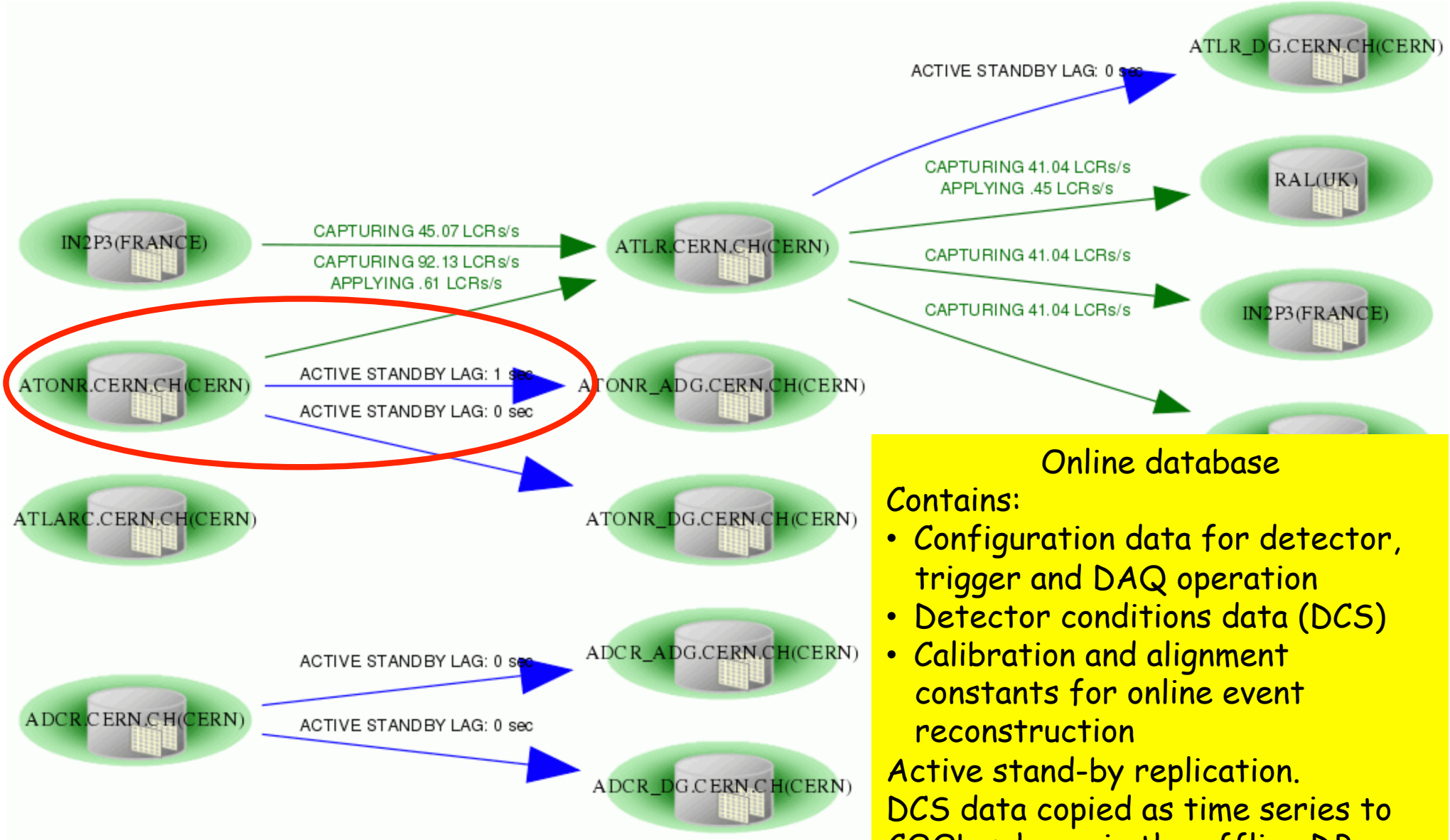
Oracle databases in ATLAS for Run2

- ATLAS relies heavily on Oracle for all major database applications
  - Licence provided by CERN
  - Oracle RACs (Real Application Clusters) provided and supported at the system level by CERN-IT
  - Lots of in-house experience on Oracle application optimisation
    - Two Oracle experts working full time for ATLAS since 2006
- Three main RACs for ATLAS (online, offline, distributed computing) plus an archive DB, all with active stand-by replicas and back-ups
- Selected users and processes have write access. All users have read access.
  - Read access normally through front-end web services (no direct access to Oracle to avoid overloading the servers):
    - Frontier for access from production and analysis jobs
    - DDM and PanDA servers for access to dataset and production/analysis task information
    - The AMI and COMA front-end servers for access to metadata

Oracle databases in ATLAS for Run2



Oracle databases in ATLAS: ATONR



Online database

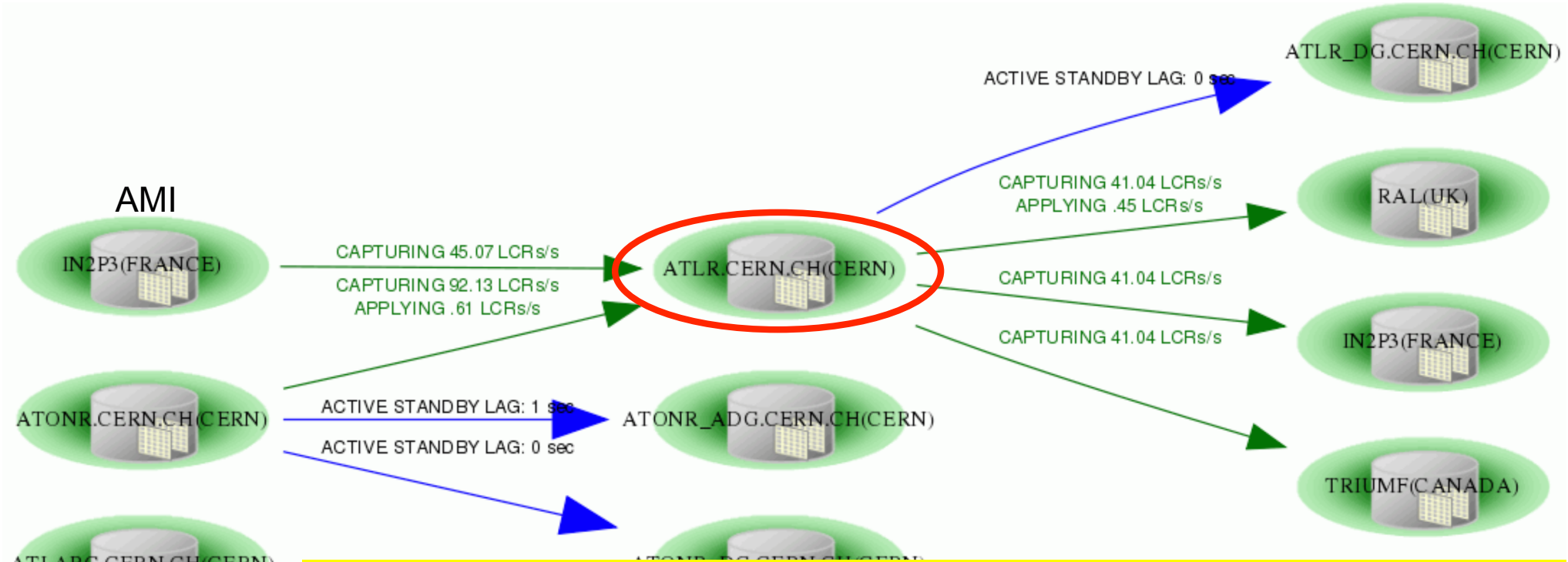
Contains:

- Configuration data for detector, trigger and DAQ operation
- Detector conditions data (DCS)
- Calibration and alignment constants for online event reconstruction

Active stand-by replication.

DCS data copied as time series to COOL schema in the offline DB.

Oracle databases in ATLAS: ATLR



Offline conditions database

Contains:

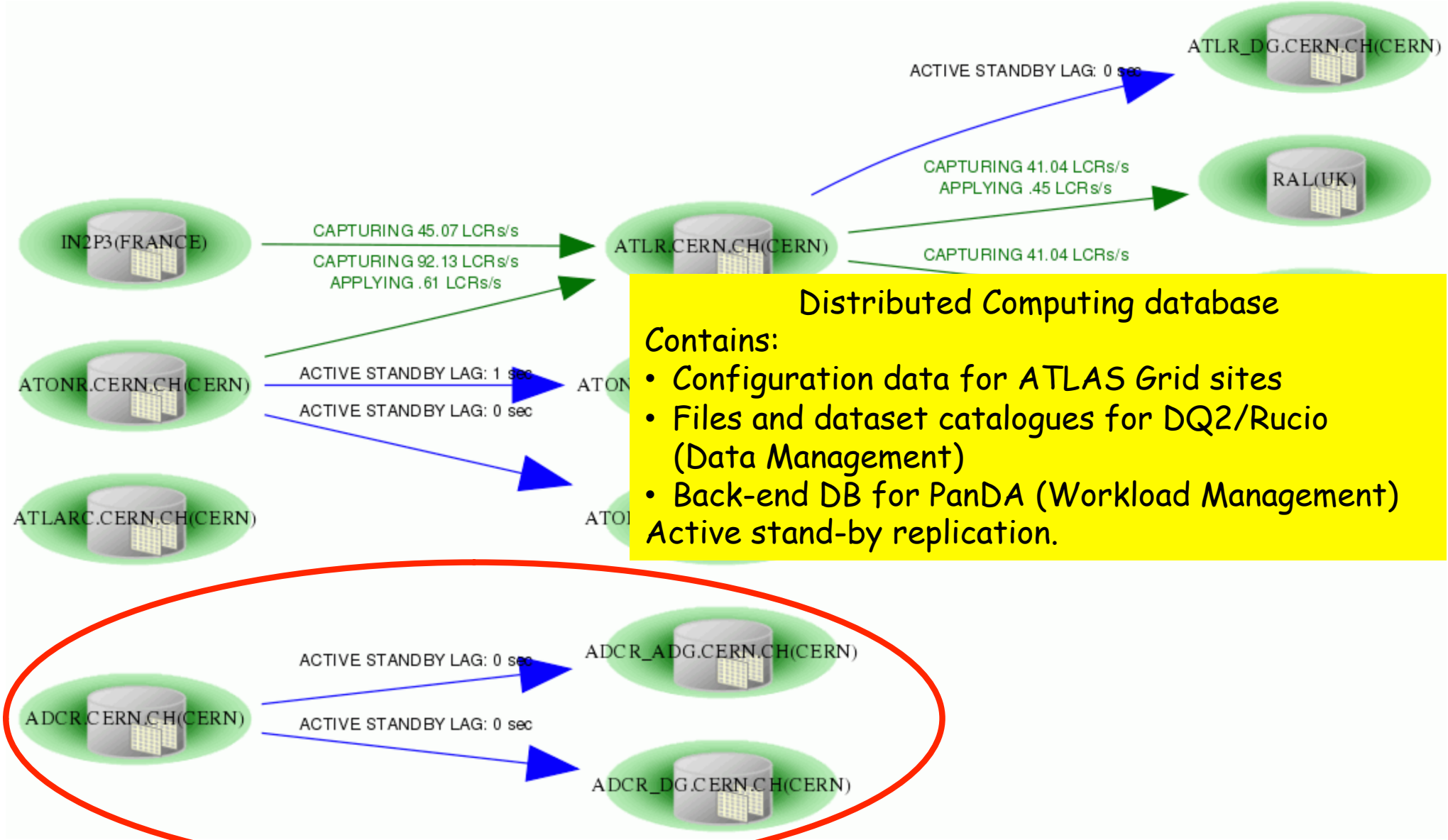
- Detector geometry and trigger DB
- Detector conditions data (DCS)
- Calibration and alignment constants for offline event reconstruction
- COMA metadata database

Active stand-by replication.

COOL data replicated to IN2P3-CC, RAL and TRIUMF for Frontier access.

Gets data from ATONR and AMI master DB in Lyon.

Oracle databases in ATLAS: ADCR



Distributed Computing database

Contains:

- Configuration data for ATLAS Grid sites
- Files and dataset catalogues for DQ2/Rucio (Data Management)
- Back-end DB for PanDA (Workload Management)

Active stand-by replication.

Non-relational storage

- The main distributed computing applications (Rucio and ProdSys/PanDA) have a very high transaction rate
  - The Oracle database is very efficient in dealing with this large information flow
  - Applications such as monitoring and accounting, that only read from the database, are instead better suited for different storage systems, with needed data extracted from Oracle and formatted appropriately for the expected queries
- Tasks to extract the relevant information from Oracle and store it in Hadoop run continuously and provide input to several other tools:
  - Dataset popularity
  - Task monitoring
  - Data management accounting
  - Etc. etc.

ElasticSearch

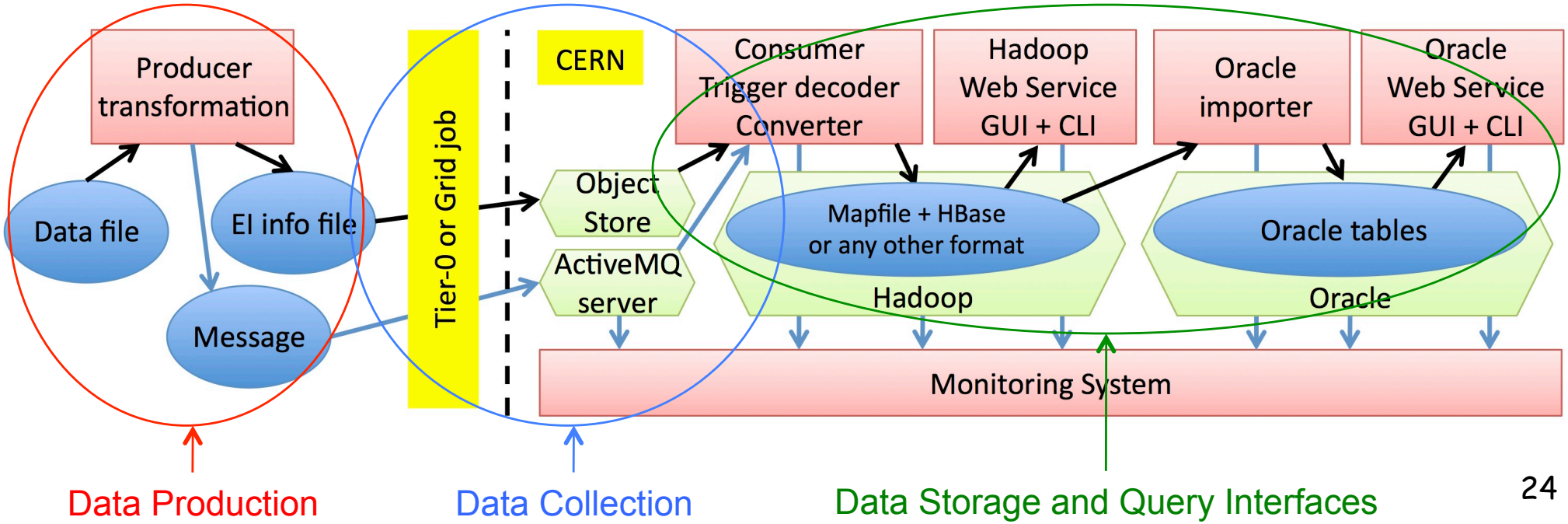
- ElasticSearch became popular in the last couple of years as a "quick" way to search information
  - Now used by several distributed computing analytics applications
- The ElasticSearch storage needs filling with data extracted from logfiles or databases, then interactive tools can be used to generate plots that are displayed with Kibana
  - Very useful for monitoring and to find out what is going on in case of unexpected failures, correlating information from different sources
    - Example: if a Frontier server becomes unresponsive, we can look up which jobs or tasks caused that, where they ran (or are running) and correlate with the PanDA status of that site
- The ElasticSearch performance gets degraded if the amount of accumulated data gets large and the hardware is not sufficient for the data size and the tasks to be performed
  - Careful provisioning is needed (like for any other computing system!)

The first NoSQL tool: EventIndex

- A system designed to be a complete catalogue of ATLAS events, real and simulated data
- Main use cases:
  - Event picking (give me this event in that format and processing version)
  - Count and select events based on Trigger decisions
  - Production completeness and consistency checks (data corruption, missing and/or duplicated events)
  - Trigger chain overlap counting
  - Derivation overlap counting
- Contents:
  - Event identifiers (run and event numbers, trigger stream, luminosity block, BCID)
  - Trigger decisions
  - References (GUID plus internal pointer) to the events at each processing stage in all permanent files generated by central productions (for event picking)
    - [RAW], [ESD], AOD, DAOD for real events
    - EVNT, [RDO], [ESD], AOD, DAOD for simulated events

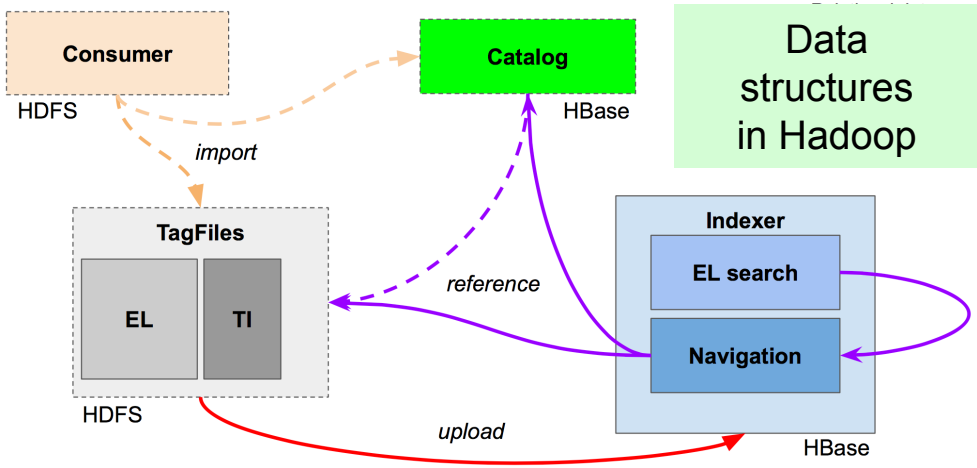
EventIndex Architecture

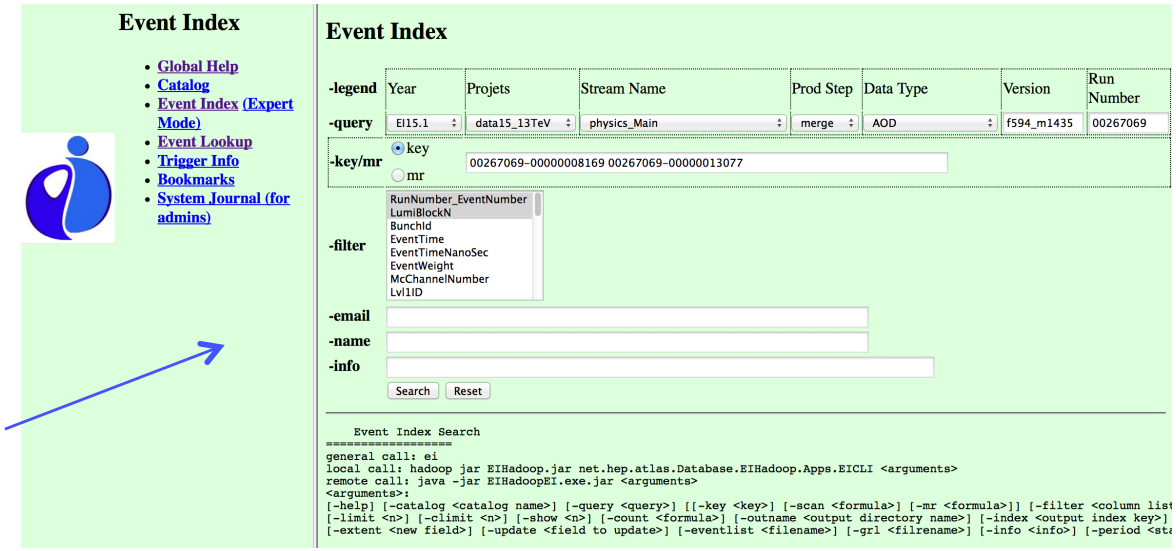
- Partitioned architecture, following the data flow
- Data Production: extract event metadata from files produced at Tier-0 or on the Grid
- Data Collection: transfer EventIndex information from jobs to the central servers at CERN
- Data Storage: provide permanent storage for EventIndex data and fast access for the most common queries + finite-time response for complex queries
  - Full info in Hadoop; reduced info (only real data, no trigger) in Oracle for faster queries
- Monitoring: keep track of the health of servers and the data flow



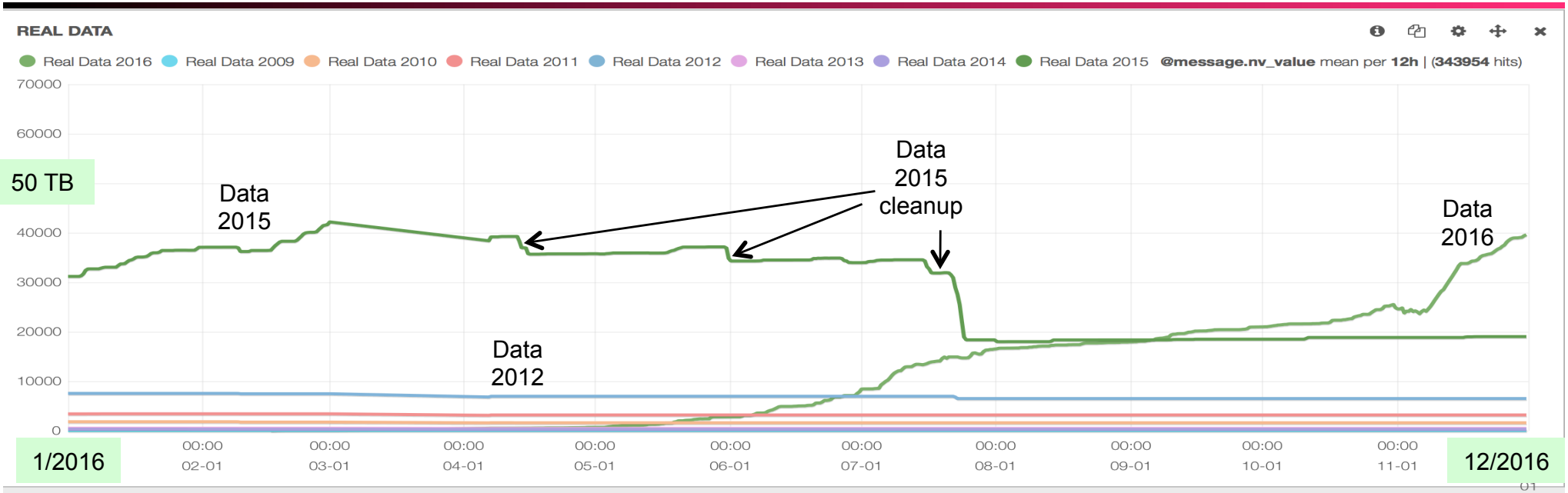
Hadoop storage and queries

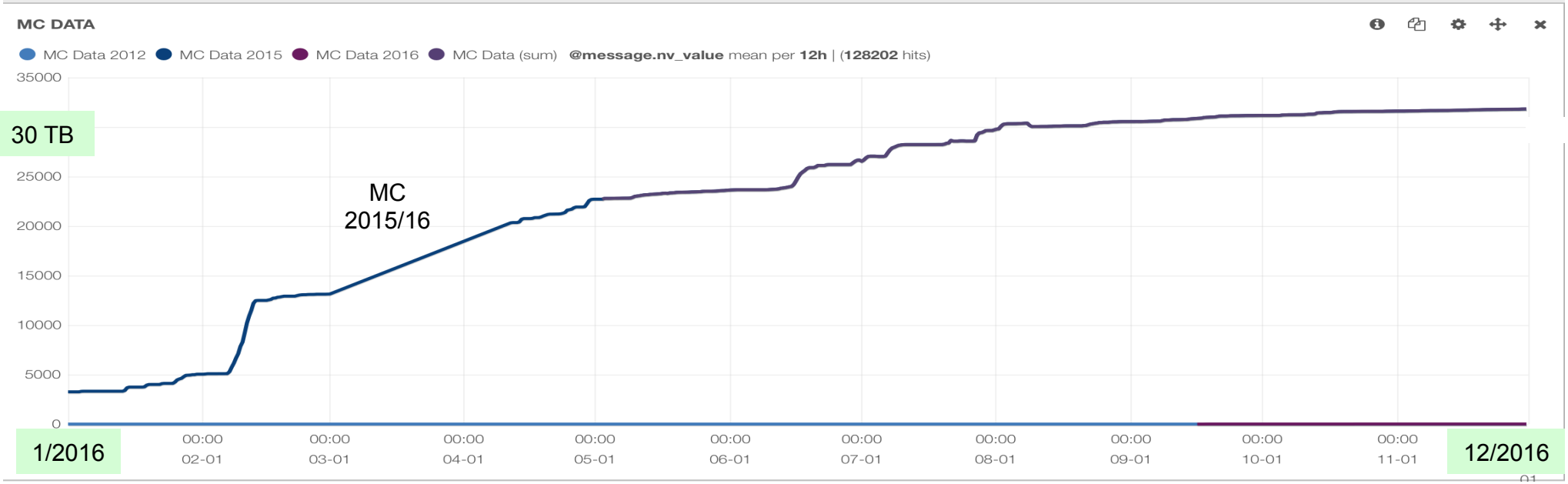
- Hadoop is the baseline storage technology
  - It can store large numbers (10s of billions) of simply-structured records and search/retrieve them in reasonable times
- Hadoop "MapFiles" (indexed sequential files) are used as data format
  - One MapFile per dataset
  - Internal catalogue in HBase (the Hadoop database) keeps track of what is where and dataset-level metadata (status flags)
  - Event Lookup index in HBase
- Data volumes:
  - Real 2009-2016: 89 TB
  - Simul mc15-mc16: 33 TB
  - Other (incl. backup): 126 TB
- CLI, RESTful API and GUI interfaces available for data inspection, search and retrieval





EventIndex Hadoop data volume

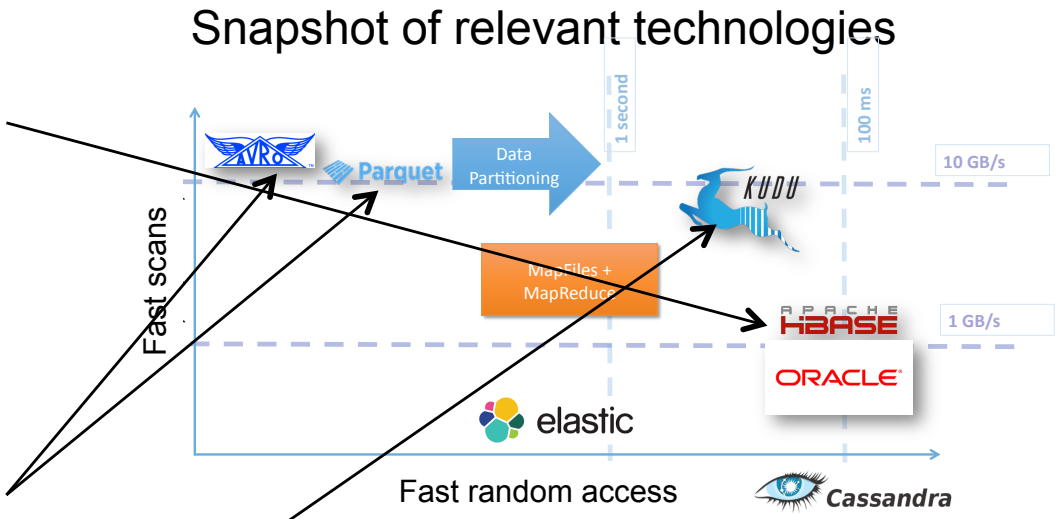


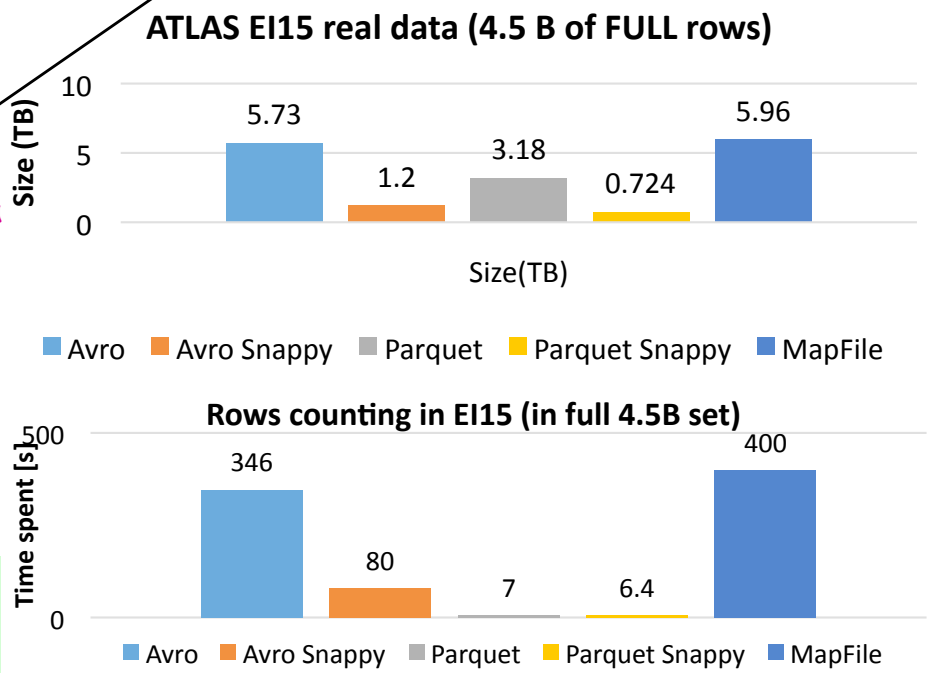


Hadoop storage developments

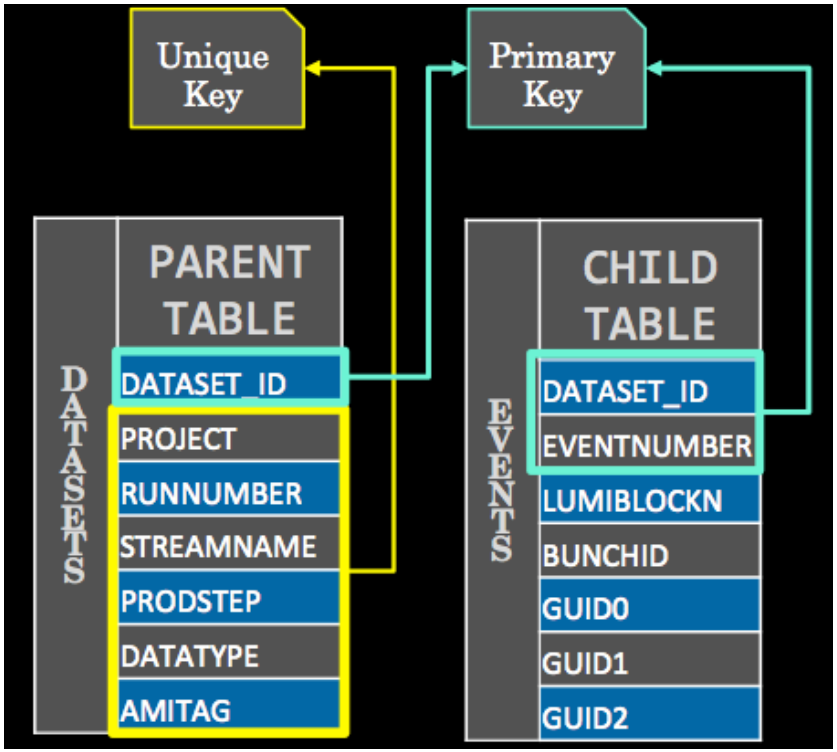
- Active R&D program to explore different data store formats in Hadoop
- "Pure HBase" (database organised in columns of key-value pairs):
  - Was one of the original options in 2013
  - Did not work in 2015 because of poor lxhadoop performance
  - More promising now — good performance for event picking
- Avro and Parquet data formats being explored
  - Tested on 2015 real data
  - Looked promising (for different reasons)
- Kudu is a new technology in the Hadoop ecosystem
  - New column oriented storage layer from Cloudera
  - Complements HDFS and Hbase
  - More flexible to address a wider variety of use cases
  - "fast analytics on fast data"
  - Currently version 1.3 available with security for the first time
  - Tests continuing this year

Zbigniew Baranowsky





EventIndex Oracle storage



- Simple schema with dataset and event tables
  - Exploiting the relational features of Oracle
- Filled with all real data, only event identification and pointers to event locations
  - Optimised for event picking
  - Very good performance also for event counting by attributes (lumi block and bunch ID)

- Connection to the RunQuery and AMI databases to check dataset processing completeness and detect duplicates
- Easy calculation of dataset overlaps
- GUI derived from COMA database browser to search and retrieve info (next slide)

Currently:

- 106 billion event records
- stored in table of 2.2 TB
- plus 2 TB index

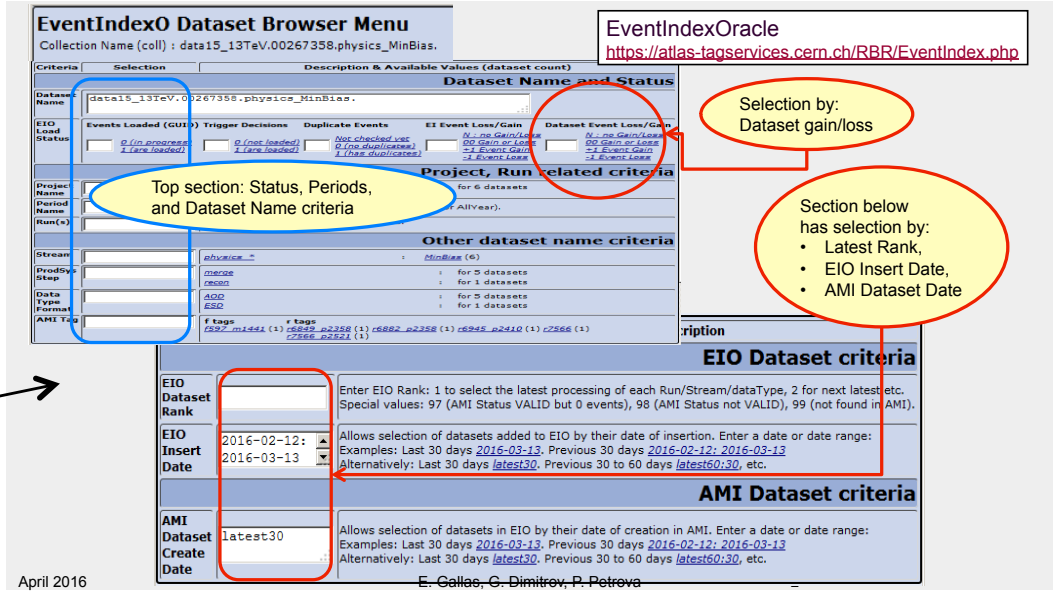
EventIndexOracle data browser

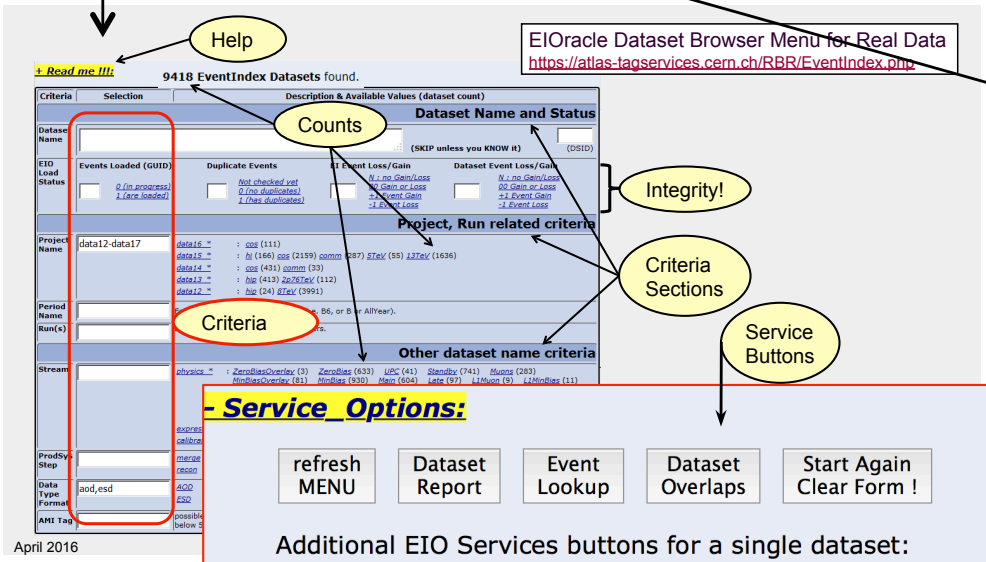
Entry point for dataset search and event lookup

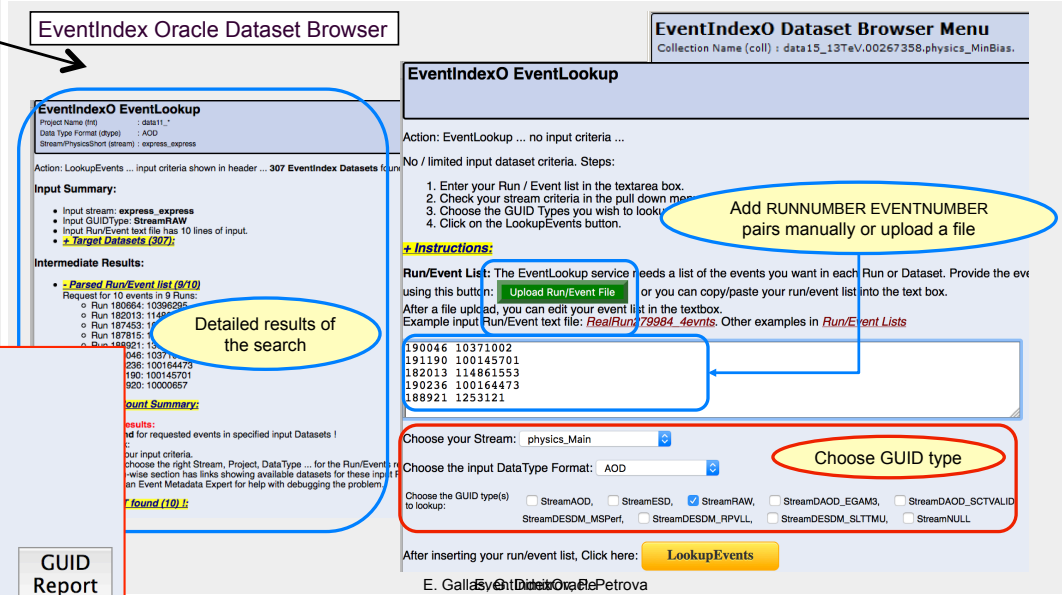
Real data only

Extensive use of the relational DB capabilities to implement the search and retrieve GUIs

1) Search dataset
2) Refine search
3) Get results







Topics

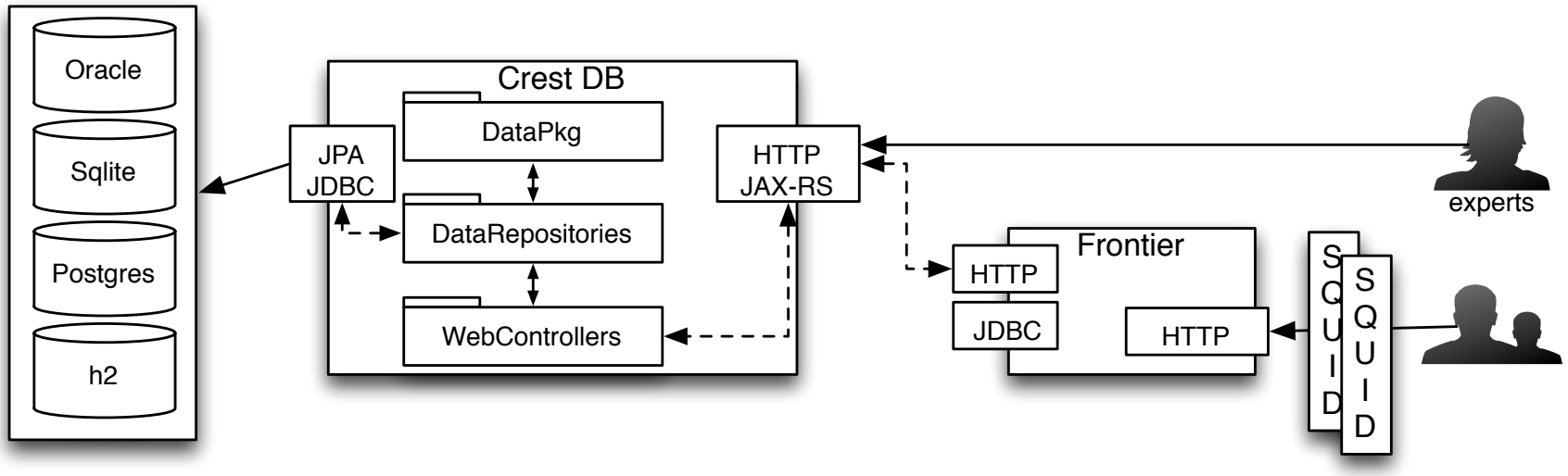
- A bit of history
- Database usage by ATLAS in LHC Run2
- Evolution of database technologies, infrastructures and applications for Run3 and beyond

Evolution of Databases for Run3

- Oracle is OK for the time being but we were warned by CERN that the licence conditions may change at the end of the current agreement
  - Diversification may be needed
- Some types of data and metadata fit naturally into the relational DB model, other data much less
  - Large amounts of useful but static data on DDM datasets (accounting), completed PanDA production and analysis tasks, event metadata
- As long as access to the data is done through an interface server, the user won't actually see the underlying storage technology
- Keeping only the "live" data in Oracle means that at some point in the future we could change technology for the SQL DB without too much trouble (only in case of need of course)
  - See examples in the following slides

A Conditions Data Service for Run3

- CREST: a new architecture for Conditions Data services for HEP experiments
  - Developed by ATLAS, CMS, NA62, Belle-II etc.
- Based on the relational schema simplification introduced by CMS for Run2
  - Data identified by type, interval of validity, version
  - Payload data in BLOBs
  - Only data used for processing in this schema (no dump of raw info)
- Partitioning of functions:
  - Relational DB only for payload data identification
  - Payload can be anywhere, including files in CVMFS
  - Web server (with cache) for interactions with the relational database
    - Data input, search and retrieval
  - Frontier servers (with cache) for Grid access
  - Local Squids for quick access to cached information

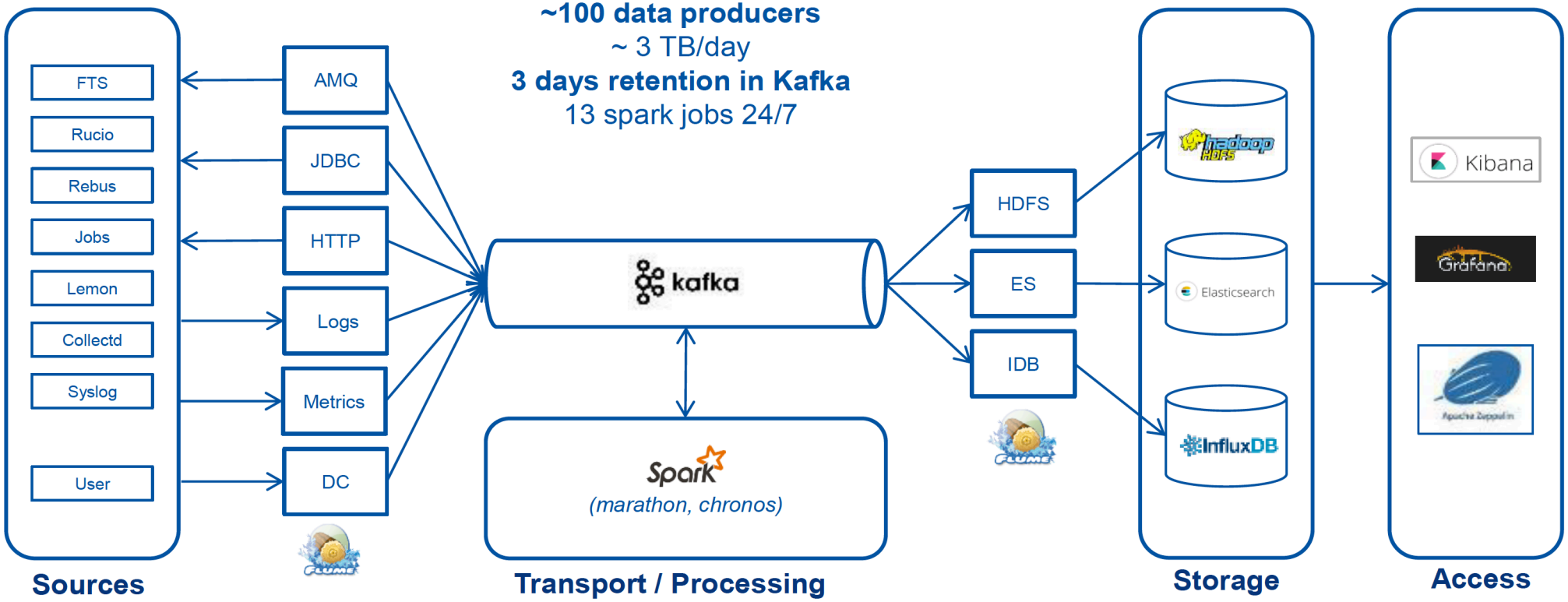


Time series databases

- Time series are (for example) streams of DCS data, where for each data type a raw data record consists of a time stamp and one or a few values
- This information is currently stored in Oracle using COOL, after averaging over (short) time periods, or storing new values only when sufficiently different from previous ones
  - Data sizes are anyway enormous compared to other data types
  - Direct use of this information in reconstruction jobs is not a good idea
- It is much better to store this information in a system that is designed for time series and has useful tools
  - Averaging over predefined time intervals
  - Threshold detection
  - Integrated display
- CERN-IT decided to provide InfluxDB and seem happy with it
- ATLAS started evaluations in the online and offline context, including displaying the time series with grafana

New CERN-IT Monitoring infrastructure

- Used initially for internal monitoring of the CERN Computer Centre
- Now being extended to WLCG and experiment distributed computing applications



Dario Barberis: ATLAS Databases

Luca Canali

Closing Remarks

- ATLAS is always following technology developments in the database and structure data storage fields
- The lifetime of ATLAS computing tools and infrastructure is much longer than the active lifetime of many open source products
  - This is a very strong constraint on product selection
- Collaboration with CERN-IT is essential for providing performant and robust services to the Collaboration
  - The tool that is invisible to most users is the one that works without problems all the time!
- In any case we need to continue the R&D programs to make the best use of new upcoming computing technologies
  - Without neglecting ongoing operations of course

Thank you!