Argonne NATIONAL LABORATORY

Databases in LHC Experiments: Usage and Lessons Learned

Alexandre Vaniachine

SuperB Computing R&D Workshop

March 9-12, 2010

IUSS, Ferrara, Italy

U.S. DEPARTMENT OF ENERGY

Outline & Disclaimer

- Database applications and computing models of LHC experiments
  - Best practices in database development cycle: schema, contents, code deployment, etc.
- Database integration with frameworks and into an overall data acquisition, production/processing and analysis chain of the experiment(s)
  - Examples of database interfaces and tools, such as custom database interfaces and services on top of CORAL/COOL, etc.
- Distributed database applications: requirements, data complexity, update frequency, data volumes, usage, etc.
  - Choice of database technologies: RDBMS, hybrid, etc.
- Database areas in which substantial progress was made and select remaining problems
  - Scalability of database access in the distributed computing environment of LHC experiments
  - Promising areas for future R&D
- **I was responsible for Database Deployment and Operations (formerly called Distributed Database Services) in ATLAS since 2004**
  - **As a result this presentation is biased towards my personal views and experience**
    - **which includes early LHC database software development using Objectivity**

Argonne NATIONAL LABORATORY

Database Applications and Computing Models of LHC Experiments

Best practices in database development cycle: schema, contents, code deployment, etc.

U.S. DEPARTMENT OF ENERGY

What are Relational DBs Used for in each LHC Experiment: an Example of ATLAS Database Applications

- More than fifty database applications reside on the offline Oracle server
  - More than 8 TB of data (February, 2010)
    - Dominated by 4 TB of slow control data (called DCS, implemented in PVSS)
- Most of these database applications are “central” by their nature
  - like the ATLAS Authorship Database used to generate author lists for publications
- Central databases are traditional applications developed according to standard Oracle best practices with a help of our OCP database administrators
  - These database applications are designed by traditional means
- I will not cover our experience with these applications in this talk
  - I can’t claim that we advanced the existing knowledge base in these traditional areas
    - I expect that traditional applications will not justify any Computing R&D in Super-B
- Central database applications are accessed by people or by limited number of computers and do not have to be distributed world-wide
  - A subset of LHC database applications must be distributed world-wide (for scalability) since they are accessed by many computers on the Grid
    - These distributed applications have many “readers” but only few “writers” (see next slide)

Best Practices in Database Development Cycle: Schema, Contents, Code Deployment, etc

- No more database privileges than necessary
  - Few writers, many readers:
    - Owner account
    - Writer account
    - Reader-only account (plain-text password has to be distributed on the Grid)
- A side-effect of that is having a database schema for each sub-detector and database servers and/or instances for each major use case
  - online or offline, testbeam or commissioning, Monte Carlo or real data
    - To reduce duplication of data and minimize potential for accidental data overwrite
- Other standard best practices for the development cycle worked well:
  - Development server
  - Integration server
  - Production server

Computing Models of LHC Experiments

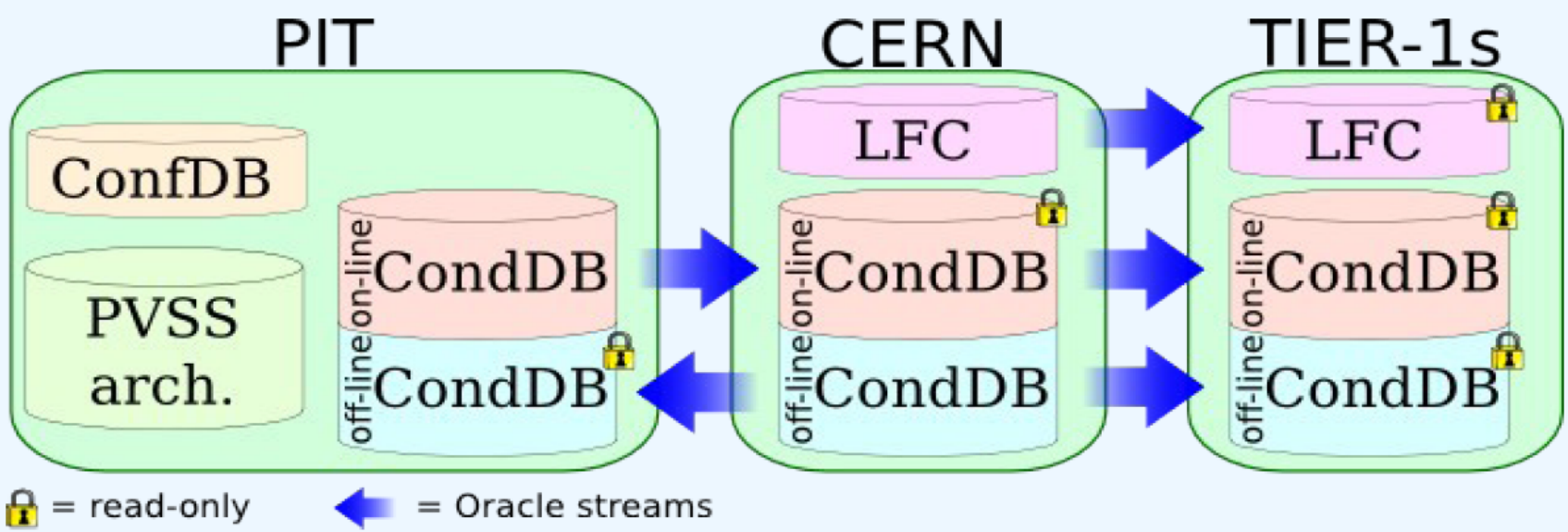
- The requirement for distributed database access on the Grid is determined by the Computing models of LHC experiments
  - LHC experiments are facing an unprecedented multi-petabyte data processing task
    - To address that challenge LHC computing adopted computing grid technologies
- LHC computing models are mostly focused on the problem of managing the petabyte-scale event data that are in a file-based data store
  - As in BABAR, event store databases were an integral part of the LHC computing models, which included Objectivity at the very beginning
  - In addition to file-based event data, LHC data processing and analysis require access to large amounts of the non-event data (detector conditions, calibrations, etc.) stored in relational databases
- Details of computing models are provided at the end of this presentation
  - In contrast to the file-based LHC event store databases, the database-resident data flow is not detailed in the "big picture" of LHC computing models
    - However, in this particular area the LHC experiments made a substantial progress (compared with other scientific disciplines that use Grids)
    - That is why I present the experience with distributed database applications introduced on the next slides

ATLAS Distributed Database Applications

- **Trigger DB**
  - Developed by ATLAS for online operations
  - A small slice for use in MC studies is distributed on the Grid
- **Geometry DB**
  - Developed by ATLAS with LCG contributions
  - First ATLAS database application deployed world-wide
  - Distributed world-wide in SQLite files (a part of Database Release)
- **Conditions DB**
  - Developed by LCG with ATLAS contributions, called COOL
  - Most challenging application – hybrid RDBMS plus files
  - Distributed world-wide via Oracle streams and POOL/ROOT files
- **Tag DB** - event-level metadata for physics (and detector commissioning)
  - Developed by LCG with ATLAS contributions
  - Distributed world-wide in files and also 4 TB are hosted at select Oracle servers
  - Given the limited data taken we do not have much experience in large scale access
    - Expected to require 40 TB per nominal LHC year of operations (TBC)

LHCb Distributed Database Applications

- Another experiment that adopted common LHC Conditions DB application

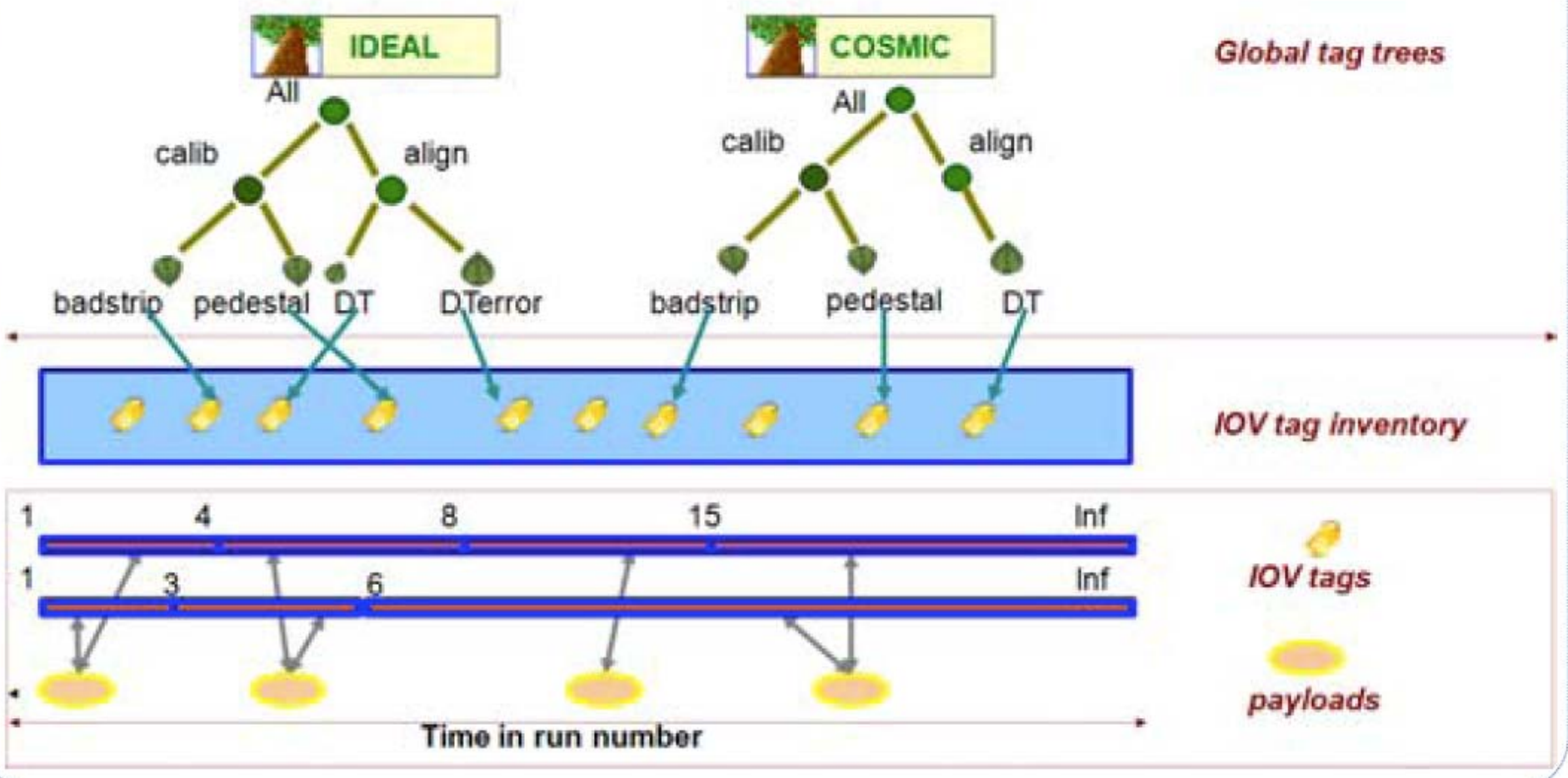


- CondDB
  - many requests per job
  - for reconstruction, analysis and HLT
- LFC
  - for reconstruction, analysis and job scheduling

Slide by M. Clemencic

CMS Distributed Database: Conditions DB

2. **Conditions DB content**: time-variant condition data (calibration and alignment) are stored in the **Condition Database** together with their sequences versions **IOV**. IOV sequences are identified by tags (IOV tag) which are organized as trees (tag tree). A tag tree can be traversed from any node (global tag).



Argonne NATIONAL LABORATORY

Database integration with frameworks and into an overall data acquisition, data processing and analysis chains of the experiments

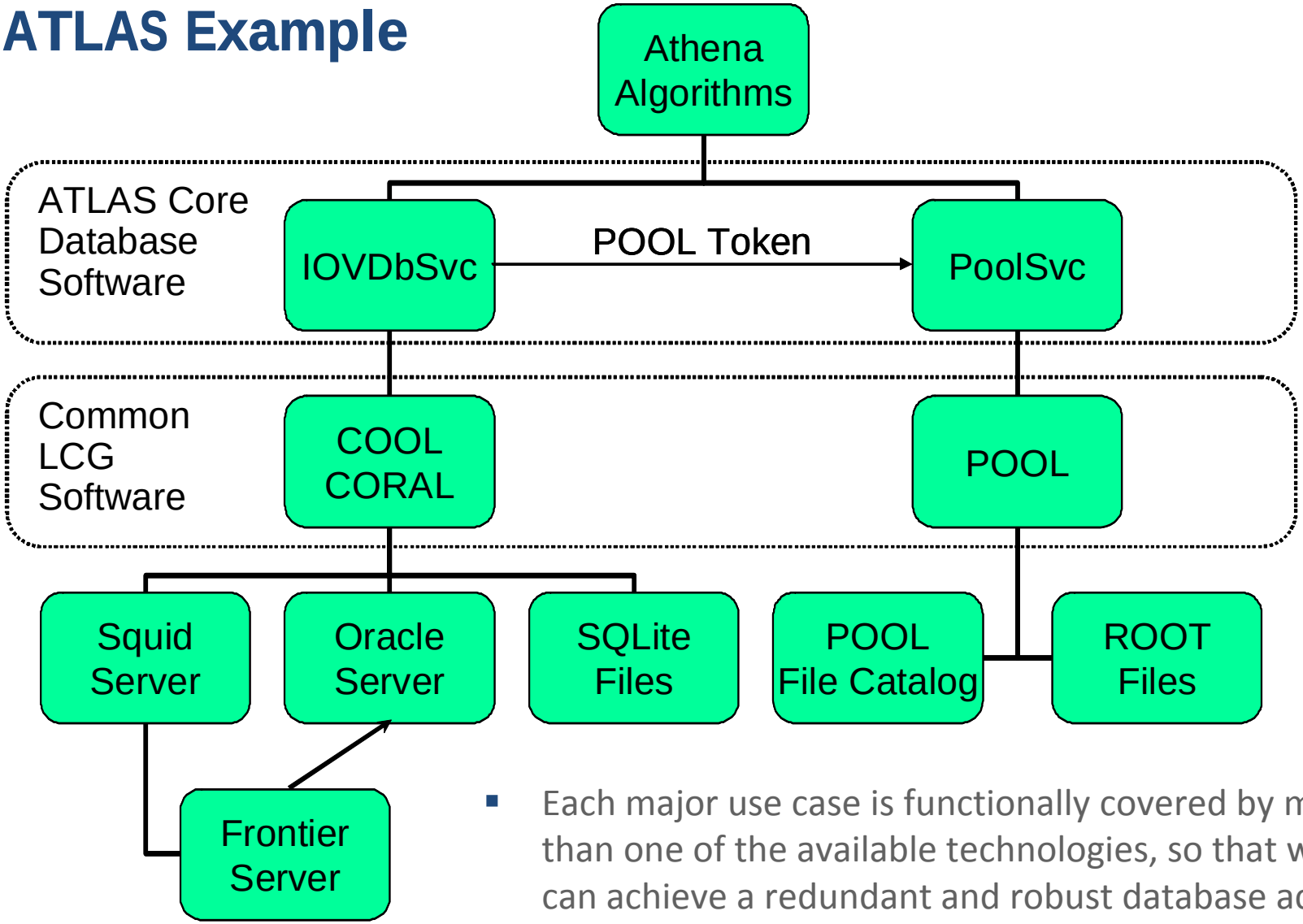
Examples of database interfaces and tools, such as custom database interfaces and services on top of CORAL/COOL, etc.

U.S. DEPARTMENT OF ENERGY

Why Interfaces are Important?

- A key feature of the common framework for LHCb and ATLAS experiments---Gaudi/Athena---is on-demand data access
  - This makes Oracle easy to use
    - In contrast, the delivery of the required Conditions DB data in files is challenging, but can be implemented for a well-organized workflow, such as reprocessing
- In the on-demand data access environment having a redundant infrastructure for database access turns out to be advantageous
  - This is achieved through common LHC interfaces for persistent data access
    - which assure independence on available technologies (Oracle, SQLite, Frontier,…)
  - No changes in the application code are needed to switch technologies

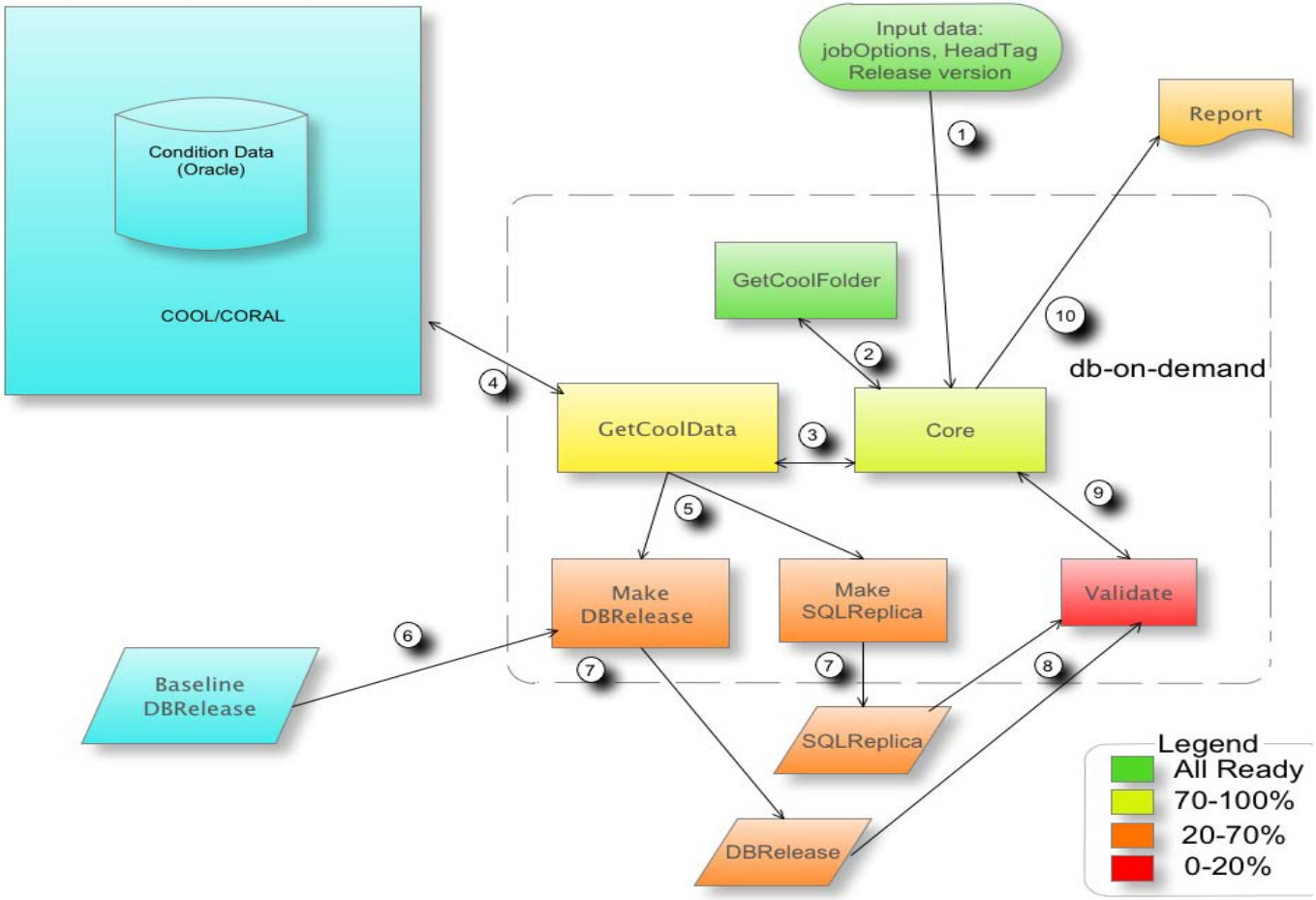
Common LHC Interfaces for Database Access: ATLAS Example



- Each major use case is functionally covered by more than one of the available technologies, so that we can achieve a redundant and robust database access system, ready for the LHC long run

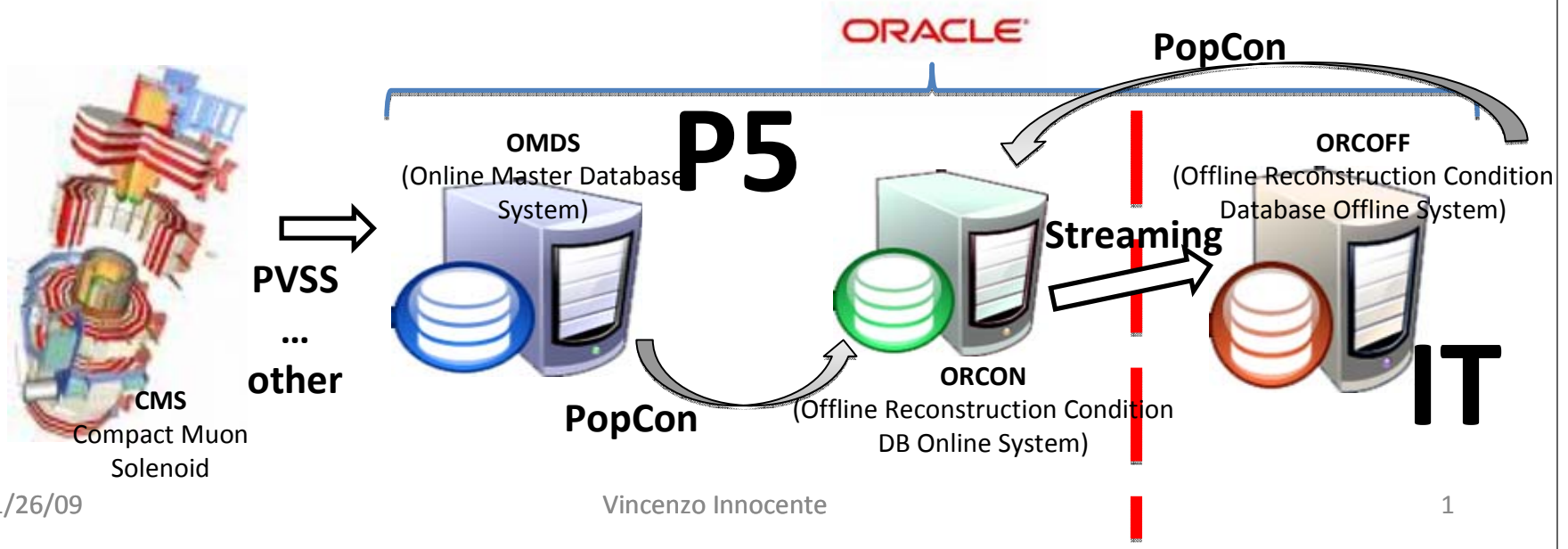
Why Tools are Important?

- Because of the large volume of Conditions DB for real data, the full database copy can not be used on the Grid
  - Tools are required to produce the “slice” of the required data for the Grid jobs



CMS Tool for Conditions DB Population

- PopCon (Populator of Condition Objects tool):
  - is an application package fully integrated in the overall CMS framework intended to transfer, store, and retrieve condition data in the Offline Databases;
  - Assigns metadata information (tag and IOV).
- CMS relies on three ORACLE databases for the condition data.



Argonne NATIONAL LABORATORY

Distributed database applications: requirements, data complexity, update frequency, data volumes, usage, etc.

Choice of database technologies: RDBMS, hybrid, etc.

U.S. DEPARTMENT OF ENERGY

Detector Description DB ("Geometry" DB)

- Requirements
  - Separated from Conditions DB to keep static information, such as nominal geometry
    - But the time-dependent alignment corrections are stored in the Conditions DB
- Data complexity - moderate
  - Recent statistics (DB Release 8.7.2)
    - 434 data structures described in 872 tables in 1 schema
    - Rows: 555,162
    - SQLite data volume: 33 MB
- Update frequency: "Static" i.e. upon request
- Interface: CORAL HVS
- Usage: each data reconstruction job makes ~3K queries to this database
- Replicated on the Grid via SQLite in validated DB Releases
  - *Reside in Offline Oracle, where it is not for production access*
  - *For Tier-0 operations the SQLite snapshot is made nightly*
  - 67 validated SQLite snapshots produced during 2009, 40 in 2008

ATLAS Conditions DB for Monte Carlo Simulations

- Requirements
  - Historically separated from the Conditions DB for real data
    - remains separate to prevent accidental overwrite of the Conditions DB for real data
- Data complexity: high
  - Statistics for recent DB Release 8.7.2
    - 2,893 tables in 4 schemas
    - Rows: 842,079
    - Data volume: 376 MB in SQLite
      - plus 247 MB in 1049 POOL/ROOT files grouped in 25 datasets
- Update frequency: “Static” i.e. upon request
  - 67 validated releases (SQLite snapshots) produced during 2009
- Usage: via CORAL COOL interface
- Replicated on the Grid via SQLite in DB Release tarballs

ATLAS Conditions DB for Real Data

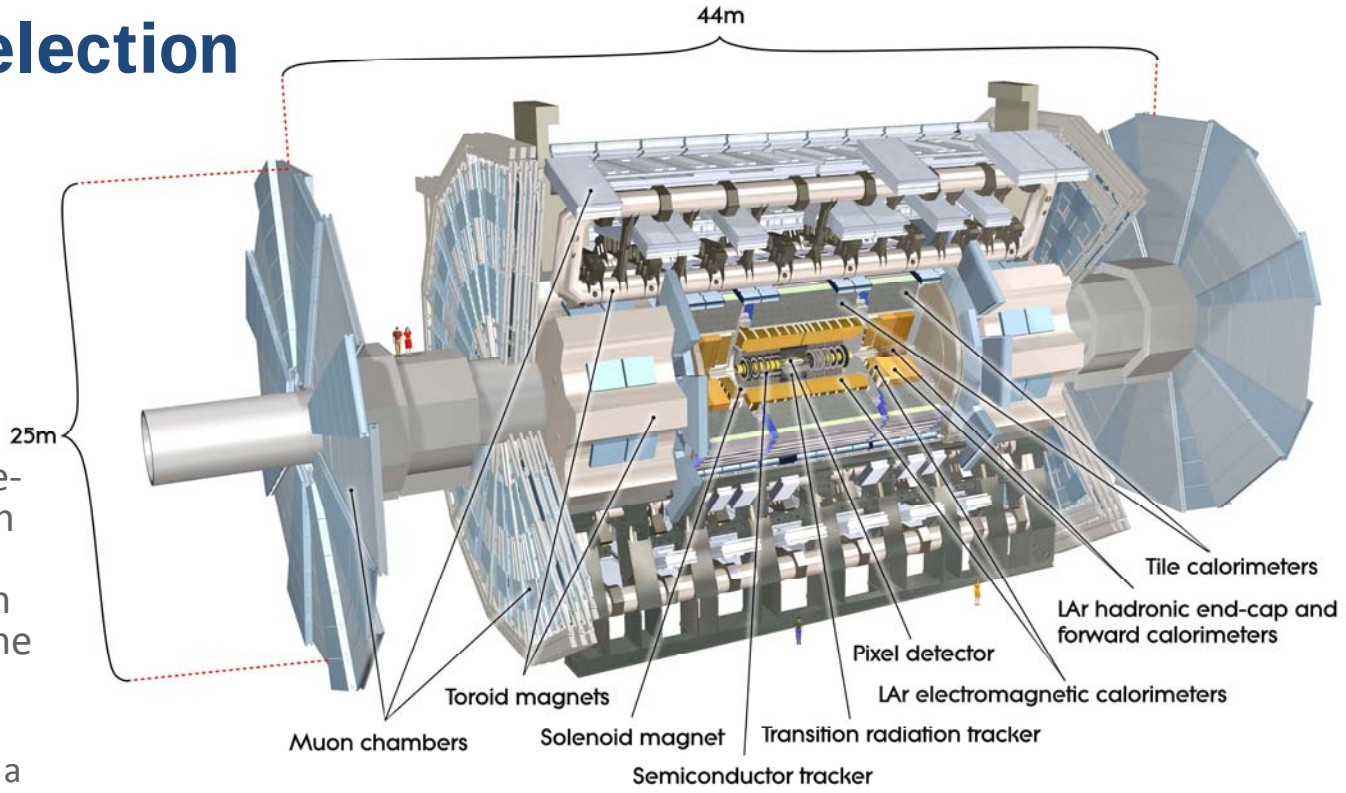
- Requirements
  - Historically separated from Conditions DB for real data (see previous slide)
- Data complexity: very high
  - Statistics for February 2, 2010
    - 7,954 tables in 29 active schemas (out of 45 schemas total)
      - Rough schema count – 15 subsystems each with two schemas (onl+ofl) plus one (old) combined schema (to be decommissioned)
    - Rows: 761,845,364
    - *Data volume: 0.5 TB in Oracle*
      - *plus 0.2 TB in POOL/ROOT files* grouped *in 48 datasets*
- Update frequency: continuous
- Usage: via CORAL COOL interface
- Replicated on the Grid via SQLite in DB Release tarballs
- To process a 2 GB file with 1K raw events a typical reconstruction job makes ~11K queries to read more than 70 MB of database-resident data
  - Some jobs read tens of MB extra
  - Plus about the same volume of data is read from external POOL files

CMS Conditions DB Partitioning by Usage

- Till now the database has been “partitioned” into accounts following development and deployment criteria
  - Keep separated areas of independent development
    - By sub-detectors, by software release
- Move to “partitioning” by use-cases
  - Keep separated independent use-workflow
    - MonteCarlo: copy all relevant data into a dedicated account
      - Even a single sqlite file!
    - Re-processing at remote Tiers: make a read-only snapshot of the whole condition DB and use that (through FronTier)
    - Prompt processing (reconstruction, calibration, analysis at T0 and CAF) keeps using the master DB (updated in real time)

Technology Selection

- Driven by the complexity of the subdetectors requirements, ATLAS Conditions DB technology is hybrid:
  - It has both database-resident information and external data in separate files, which are referenced by the database-resident data
    - These files are in a common LHC format called POOL



- ATLAS database-resident information exists in its entirety in Oracle but can be distributed in smaller "slices" of data using SQLite
- Oracle was chosen as a database technology for the online DB
- Benefits of uniform technology:
  - avoids translating from one technology to another
  - support for Oracle from CERN IT and WLCG 3D Services

CMS Conditions DB Implementation

- All relations in conditions DB are purely logical and application specific
  - No RDBMS consistency enforced
  - Full flexibility in copying (deep and shallow) at all level of the structure
  - Simple policy: NO DELETE, NO UPDATE
    - Only the current IOV-Sequence can be extended
  - Payloads implemented as POOL/ORA objects
  - Management through application specific tools

Access: CondDB

- Applications
  - Reconstruction and analysis jobs
  - Direct connection from the CEs to Oracle
  - Via COOL/CORAL libraries
- Pattern
  - Limited amount of data transfer (~40MB) in the first few minutes
- SQLite used in special cases (MC)

Slide by M. Clemencic

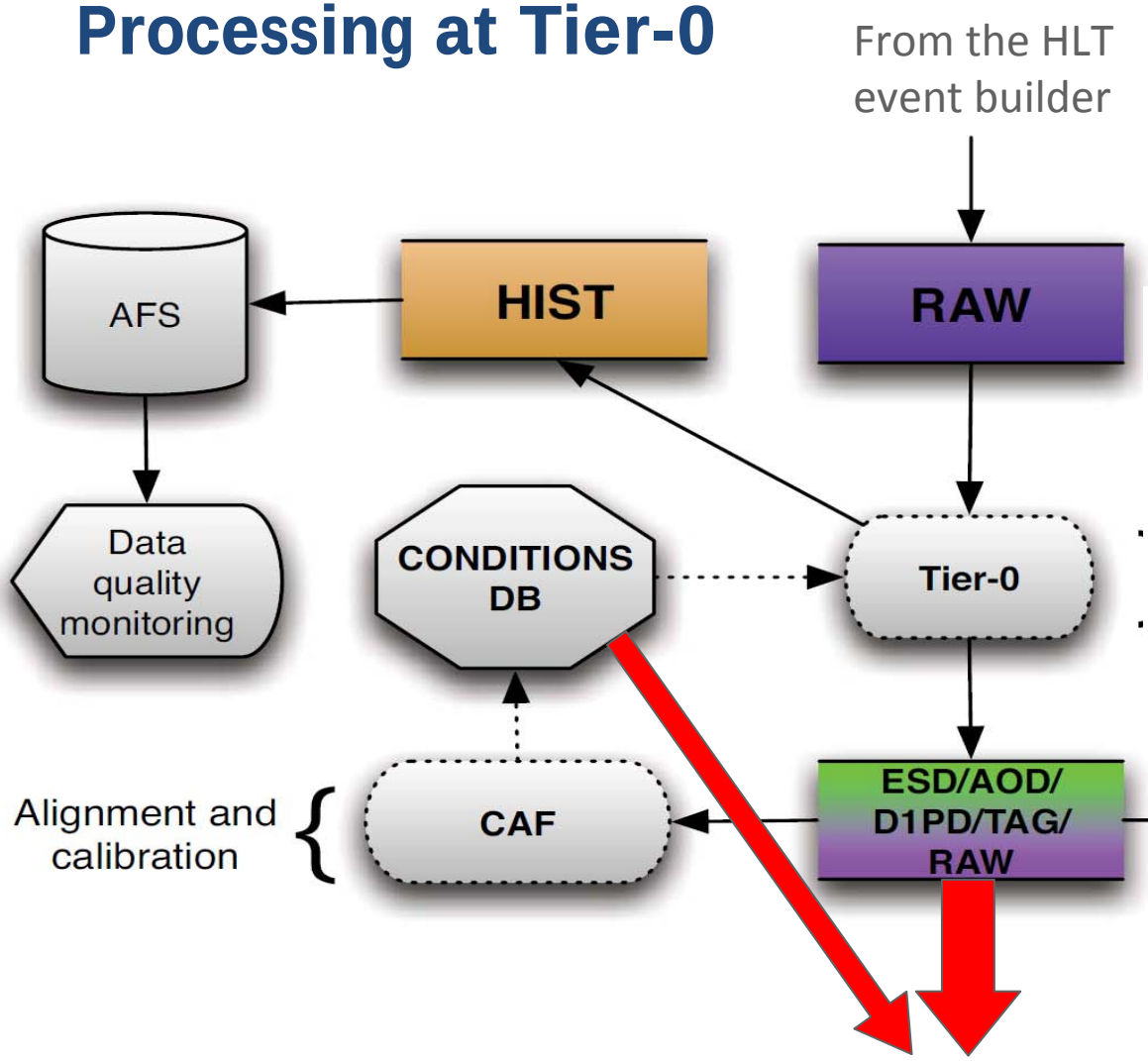
Argonne NATIONAL LABORATORY

Scalability of database access in the distributed computing environment of LHC experiments

A challenging area in which substantial progresses was made

U.S. DEPARTMENT OF ENERGY

ATLAS Offline Data Processing at Tier-0



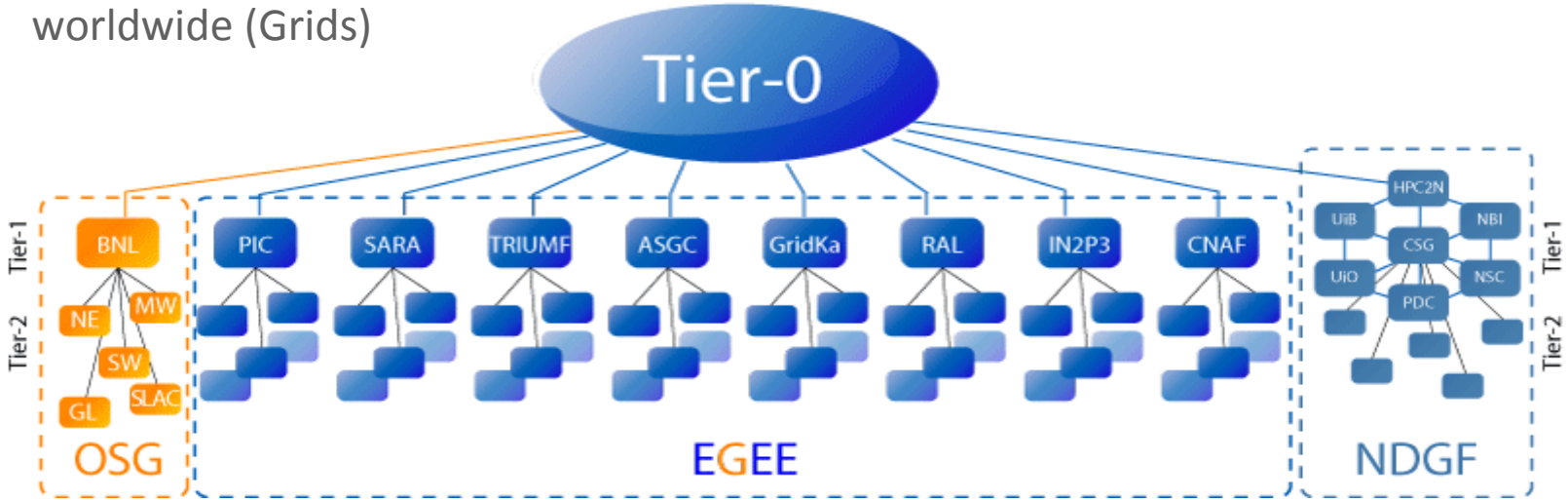
- Conditions DB is critical for data reconstruction at CERN using alignment and calibration constants produced within 24 hours: the "first pass" processing
- Two solutions assure scalability of access to Conditions DB database:
  - Replicated AFS volume for POOL files
  - Throttling of job submission at Tier-0

- To distributed computing facilities (the Grid) – see next slide

Diagram by J. Catmore

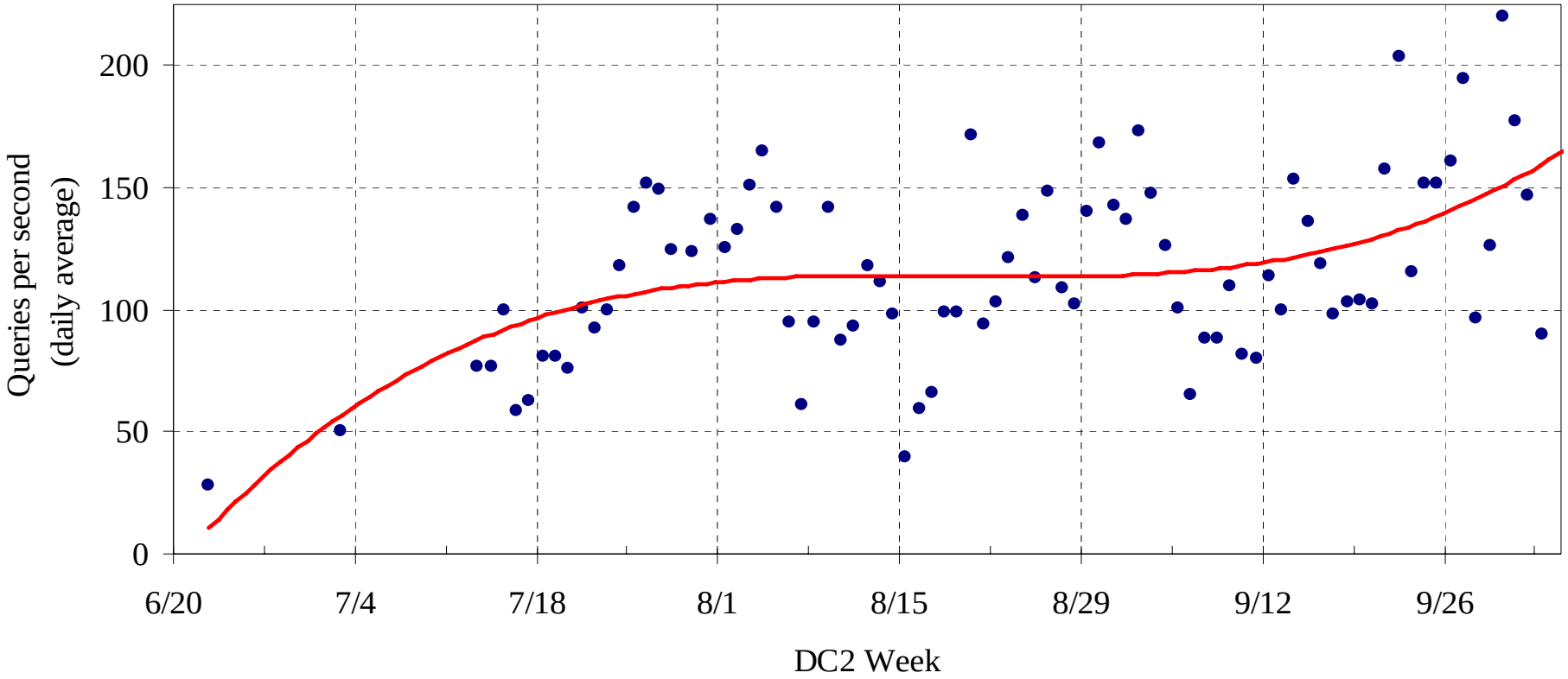
Typical LHC Distributed Computing Infrastructure

- For data processing with improved alignment and calibration constants (reprocessing) LHC uses shared computing resources, which are distributed worldwide (Grids)



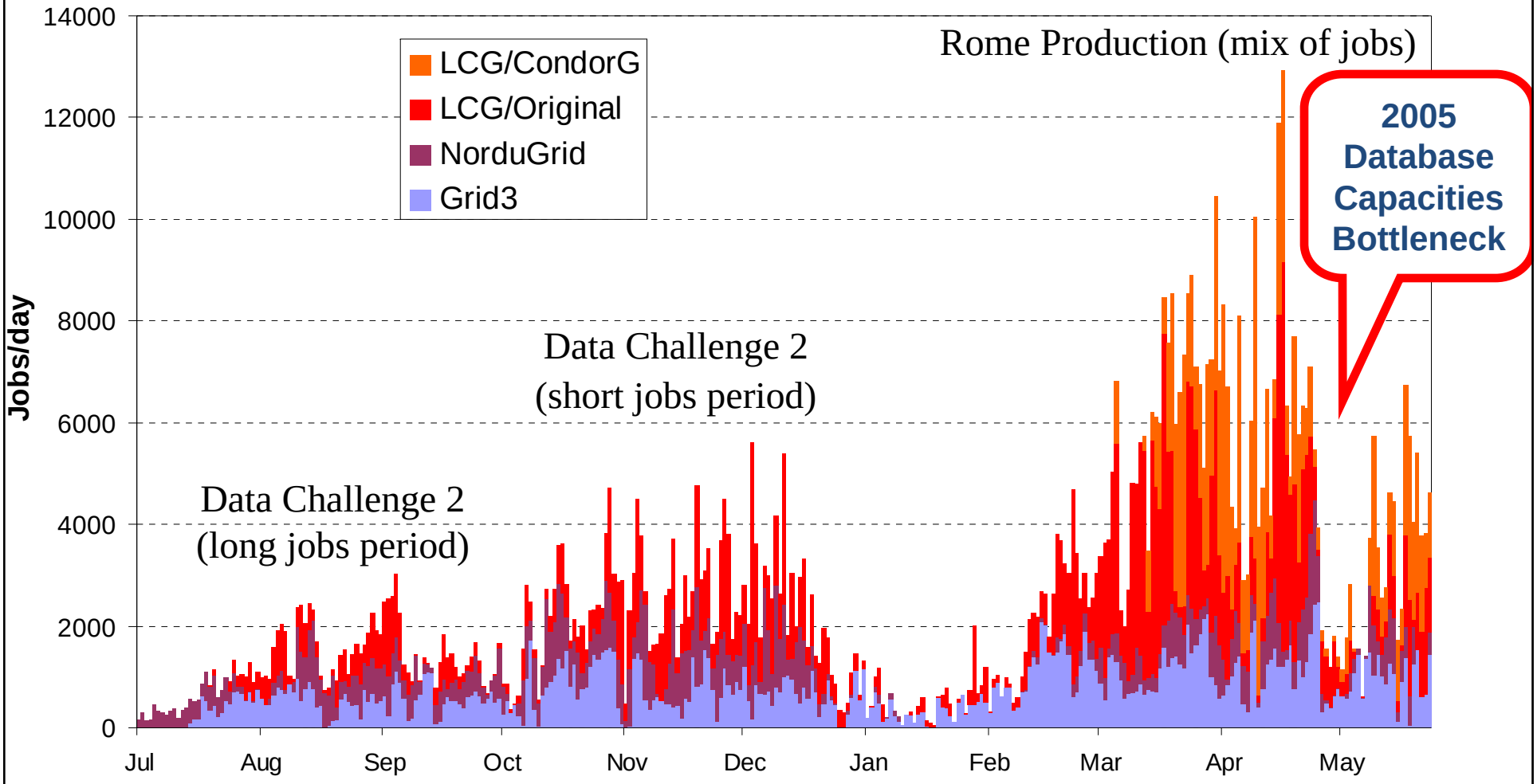
- ATLAS uses three Grids (each with a different interface) split in ten "clouds" organized as large computing centers with tape data storage (Tier-1 sites) each associated with 5-6 smaller computing centers (Tier-2 sites); plus more than a hundred of Tier-3 sites – this is a physicist's own computing facility at the university or the department
  - ATLAS distributed computing power is many times higher than that at Tier-0
- None of Tier-0 solutions for scalable database access is available on the Grid
  - Experience with database access on the Grid provided many "lessons learned"

Lessons Learned in 2004: Load Fluctuations during Database Access from the Grid

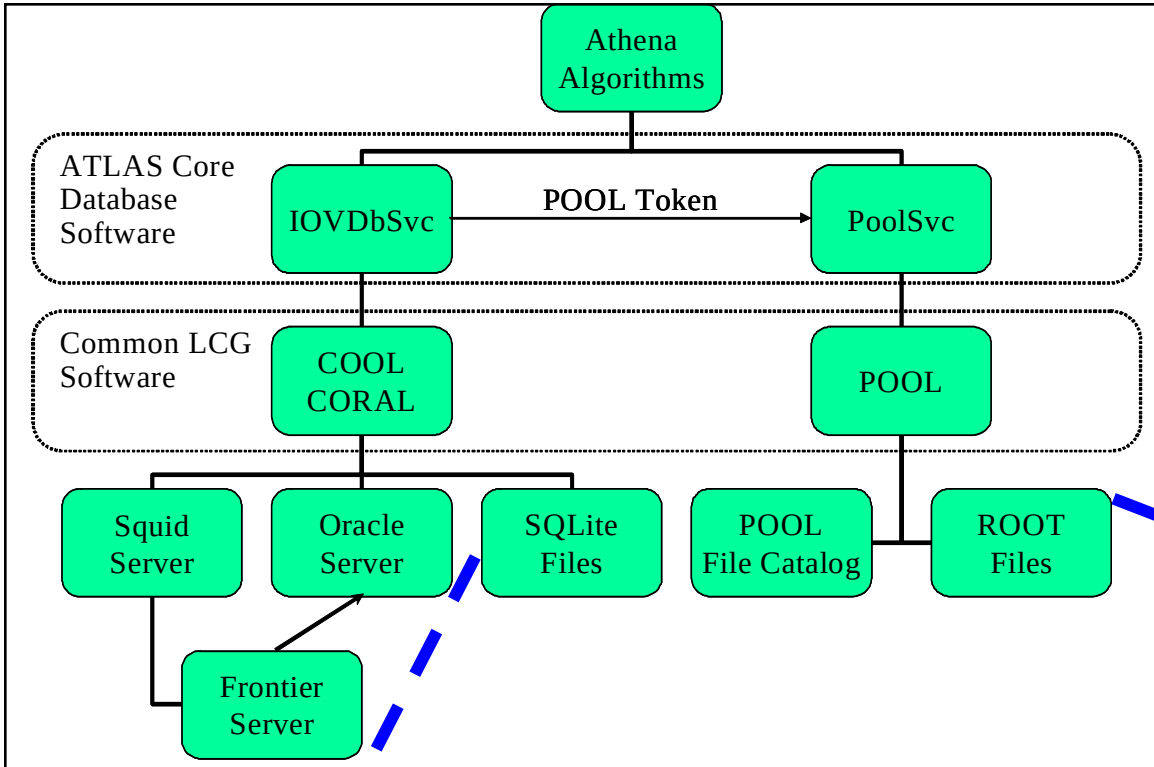


- The chaotic nature of Grid computing increases fluctuations in database load
  - Daily fluctuations in the load are fourteen times higher than purely statistical
- To avoid bottlenecks in production, database servers capacities should be adequate for peak demand (see next slide)

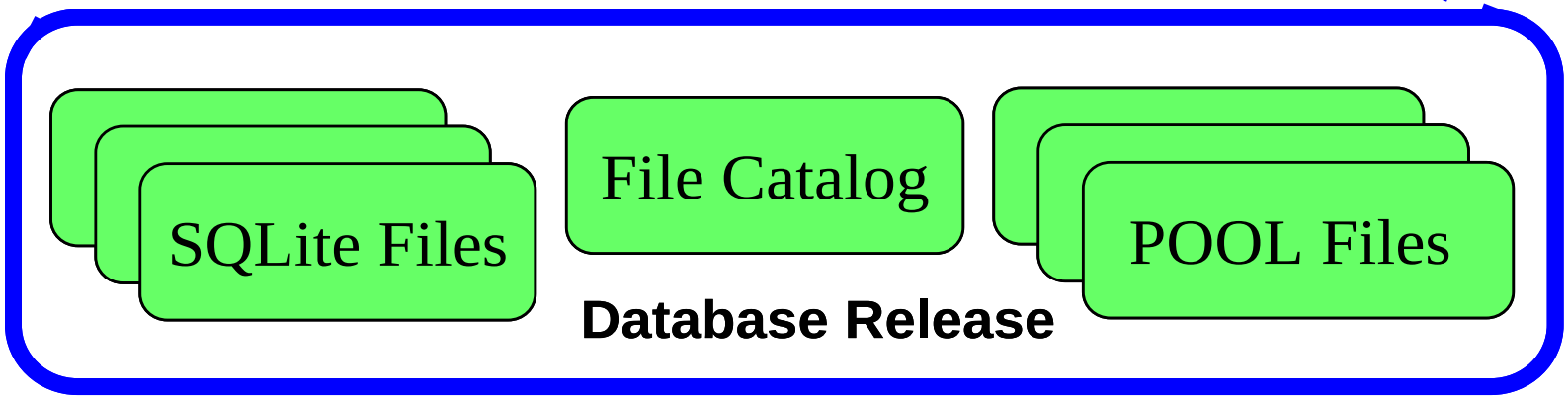
Lessons Learned in 2005: Production Bottleneck Caused by Limited Database Servers Capacities



Database Release



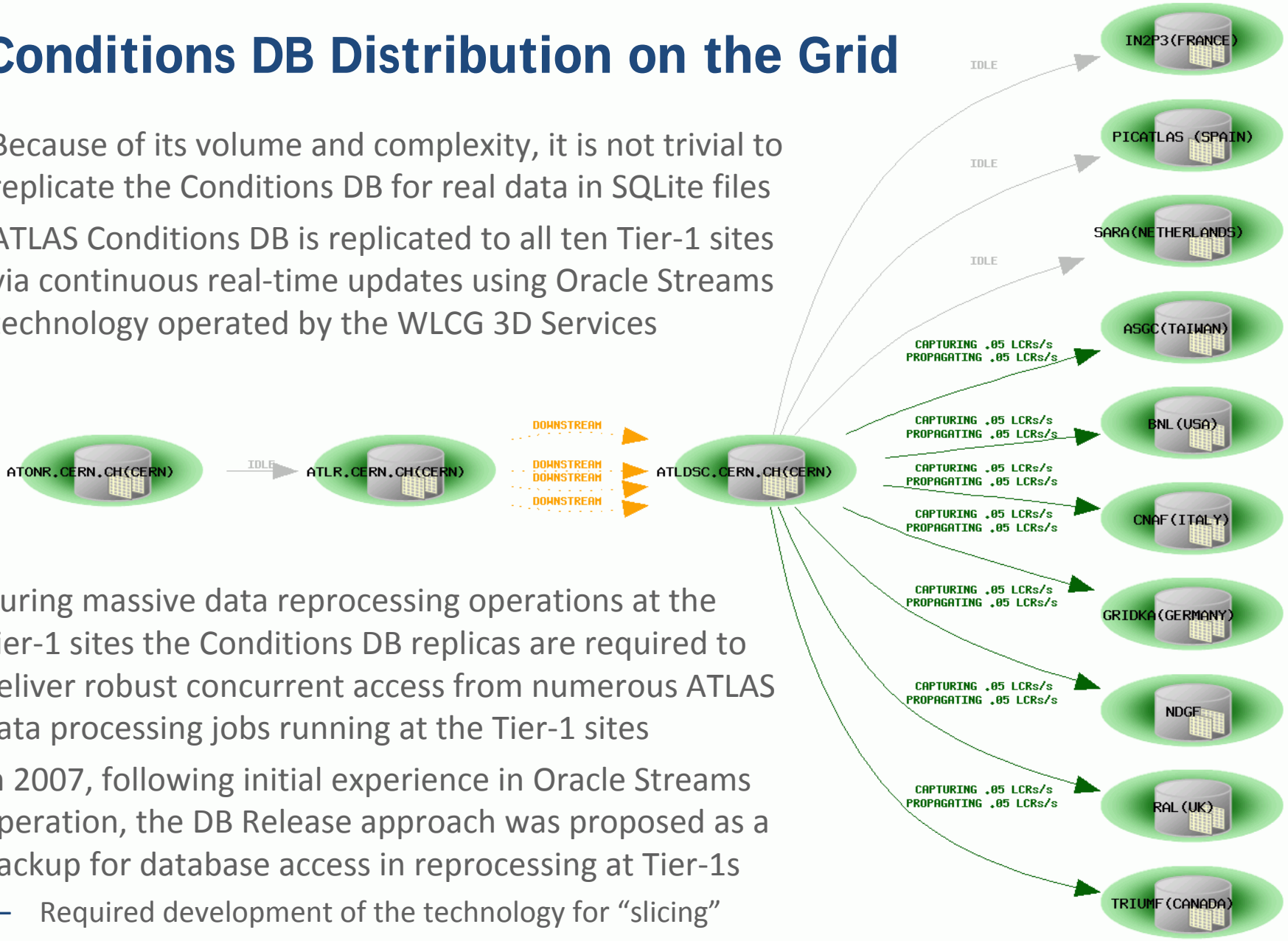
- In 2005, to overcome scalability limitations in database access on the Grid, ATLAS introduced the Database Release concept



- Takes full advantage of the technology independence in persistent data access
  - Full SQLite replicas assure scalable database access for Monte Carlo production on the Grid

Conditions DB Distribution on the Grid

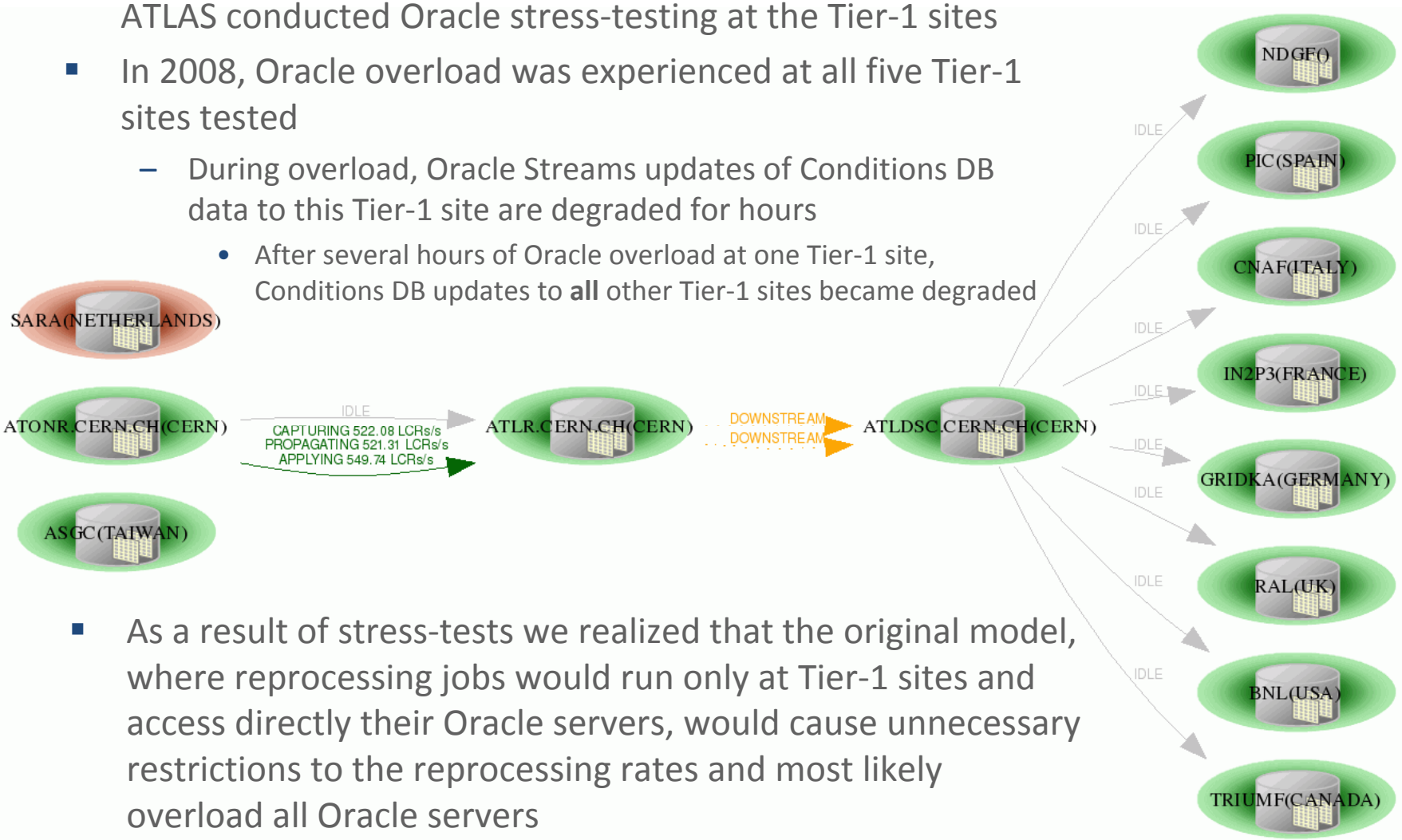
- Because of its volume and complexity, it is not trivial to replicate the Conditions DB for real data in SQLite files
- ATLAS Conditions DB is replicated to all ten Tier-1 sites via continuous real-time updates using Oracle Streams technology operated by the WLCG 3D Services



- During massive data reprocessing operations at the Tier-1 sites the Conditions DB replicas are required to deliver robust concurrent access from numerous ATLAS data processing jobs running at the Tier-1 sites
- In 2007, following initial experience in Oracle Streams operation, the DB Release approach was proposed as a backup for database access in reprocessing at Tier-1s
  - Required development of the technology for “slicing”

Problem: Escalation of Oracle Incidents

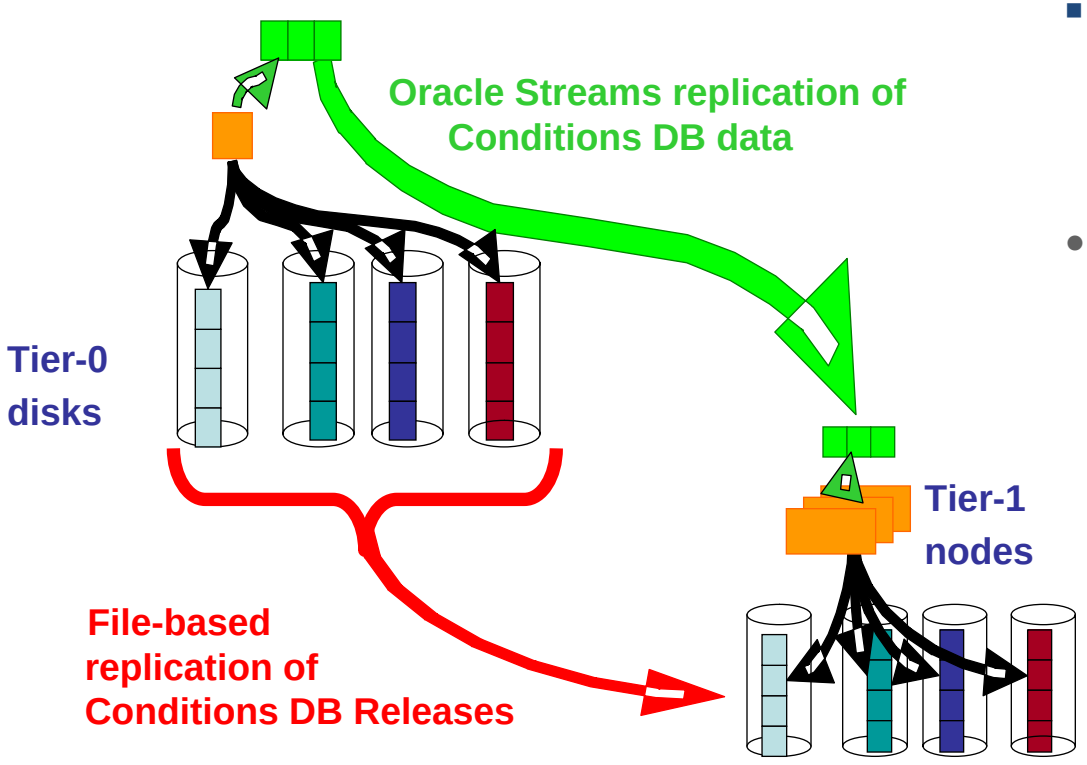
- To assure scalable database access during reprocessing ATLAS conducted Oracle stress-testing at the Tier-1 sites
- In 2008, Oracle overload was experienced at all five Tier-1 sites tested
  - During overload, Oracle Streams updates of Conditions DB data to this Tier-1 site are degraded for hours
    - After several hours of Oracle overload at one Tier-1 site, Conditions DB updates to **all** other Tier-1 sites became degraded



- As a result of stress-tests we realized that the original model, where reprocessing jobs would run only at Tier-1 sites and access directly their Oracle servers, would cause unnecessary restrictions to the reprocessing rates and most likely overload all Oracle servers

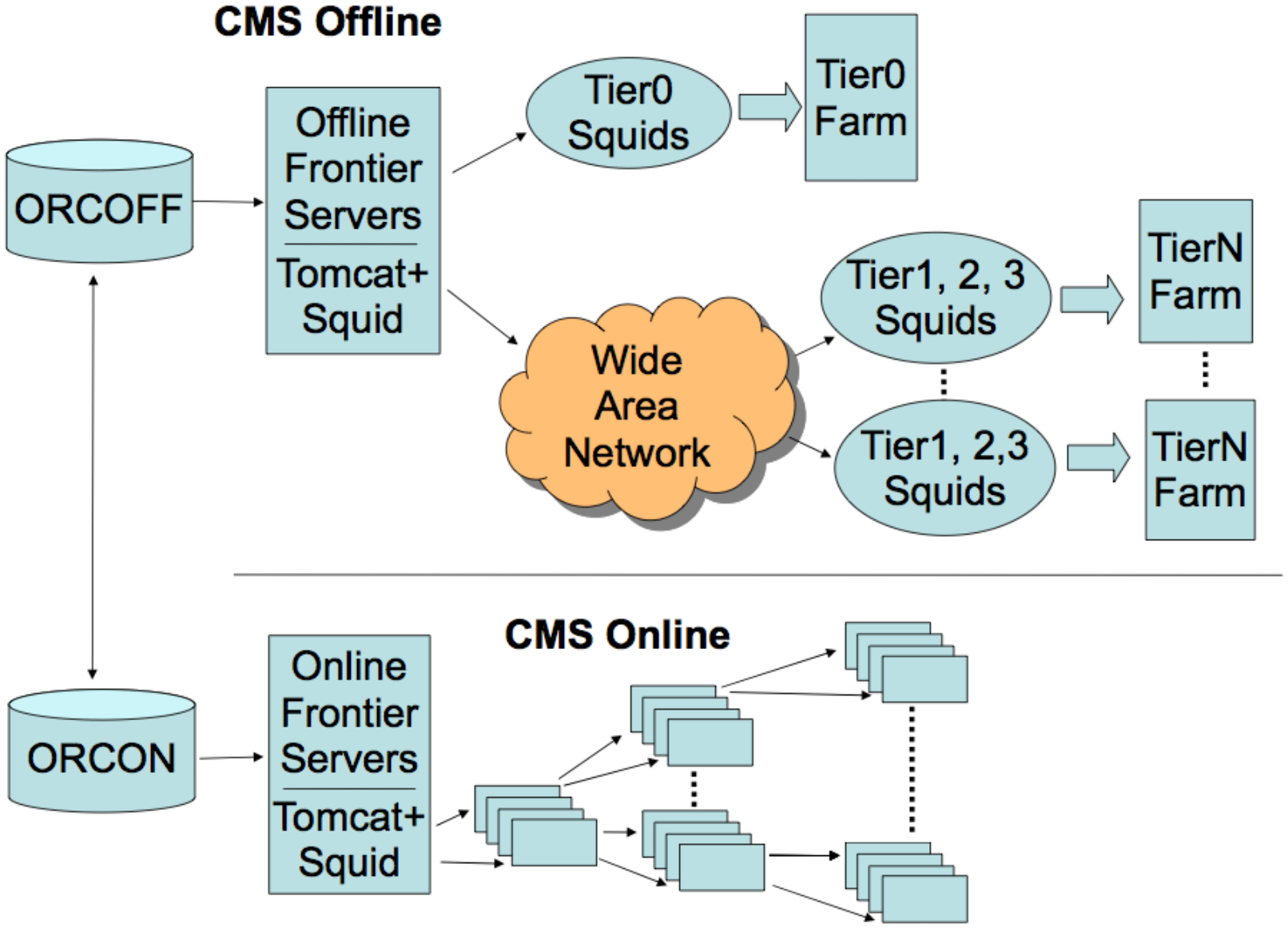
ATLAS Strategic Decisions for Reprocessing

- Read most of database-resident data from SQLite
- Optimize SQLite access and reduce volume of SQLite replicas
- Maintain access to Oracle
  - to assure a working backup technology, when required



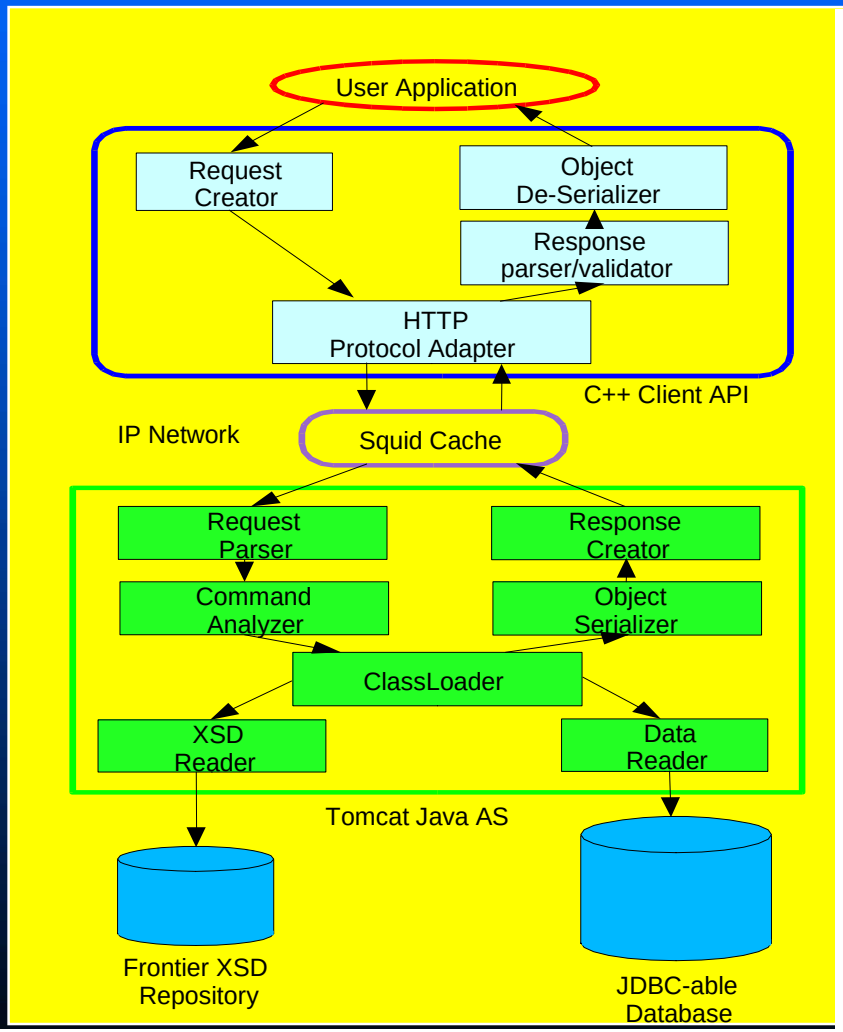
- As a result of these decisions we overcome the Conditions DB scalability challenges in ATLAS reprocessing
- ATLAS DB Release technology fully satisfies the Computing Model requirements of data reprocessing and Monte Carlo production
  - Fast: < 10s per job
  - Robust: failure rate < 10^{-6} per job
  - Scalable: served 1B queries in one of reprocessing campaigns
  - Frozen to guarantee reproducibility

CMS: Not Accessing WLCG 3D Databases at the Tier-1 Sites



Frontier & POOL (Simplified)

Sergey Kosyakov



- FroNTier is a web data caching system for scalable database access
- In collaboration with CMS and WLCG 3D, ATLAS started an evaluation of the Frontier technology in 2006
  - We found good performance gains for cached data
- Since initial Frontier/squid implementation did not maintain cache consistency by itself, it was found that considerable efforts must be spent to assure that applications obtain stable results in the case of ongoing changes to the Conditions DB
- Leveraging recent CMS & CERN IT developments addressing the cache consistency problem, ATLAS completed successful Frontier development and testing in 2008

Slide by L. Lueking, 3D Workshop, December 14, 2004

ATLAS Frontier/Squid Deployment

- In 2009 ATLAS deployed Frontier for database access in user analysis, with five Frontier servers in production deployed at BNL, FZK, TRIUMF, RAL, PIC Tier-1 sites
  - Most ATLAS data analysis jobs use Frontier/squid infrastructure

Map by D. Smith

Conclusions

- LHC experiments developed and deployed distributed database infrastructure ready for the LHC long run
- In ATLAS each major use case is functionally covered by more than one of the available technologies to assure a redundant and robust database access
- In CMS a novel web data caching system---Frontier/Squid---assures scalability of Conditions DB access on the Grid
  - Frontier/Squid technologies are adopted by ATLAS as a part of end-to-end solution for Conditions DB access in User Analysis
- I think that distributed database is the area where the LHC experiments made a substantial progress
  - compared with other scientific disciplines that use Grids
    - The remaining problems provide promising areas for future R&D

Argonne NATIONAL LABORATORY

Promising Areas for Future R&D

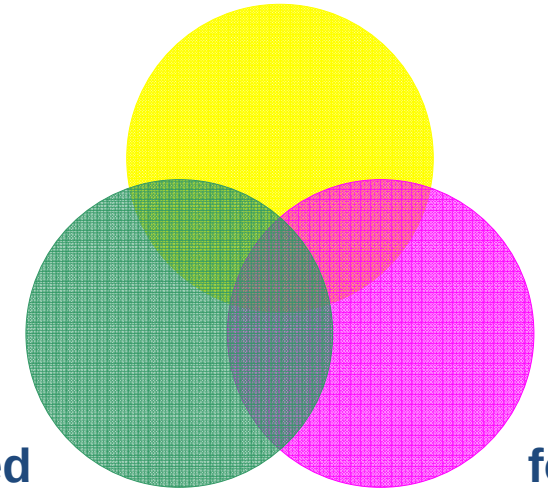
U.S. DEPARTMENT OF ENERGY

Is Single Database Access Technology Possible?

- LHC experiments were unable to converge on a single technology that can cover all use cases satisfactorily

- Example from ATLAS:

Oracle for Central Operations



Oracle technologies for:
- Online data taking
- Central databases:
  - COOL for Tier-0
  - Files Catalog
  - Jobs Database
  - …

SQLite for Distributed Production Support

- End-to-end database access solution for data reprocessing and MC production on the Grid

Frontier for Distributed Analysis Support

- End-to-end solution for database access in User Analysis on the Grid

Is Single Data Access Interface Feasible?

- LHC architecture for persistent data access is independent of the implementation technology
- As a result, without any changes in the software code, data processing applications can access database-resident data through various technologies (Oracle, SQLite, FroNTier/Squid…) by using the common LCG-AA software packages COOL/CORAL and POOL/ROOT
  - Can COOL/CORAL and POOL/ROOT be integrated behind a single interface?
    - A single interface should ease the management of hybrid data in files and in databases
    - Should add an extra operational flexibility – ease migration of data from POOL/ROOT files to RDBMS and back when necessary

Built-in Database Consistency and Data Locking

- Neither common LHC COOL nor CMS Conditions DB implementation assures data consistency using traditional RDBMS methods
- Experience shows that relying on policies and tools for data locking and consistency checking presents challenges in operations
  - Due to benign human errors
- Would not it be nice to have the consistency of the database application, such as Conditions DB, guaranteed by the database application itself?
  - Using traditional RDBMS methods such as constraints, triggers, etc.
- Can operational complexities, such as database partitioning by use-workflow, be avoided because data are protected by design?
  - RDBMS transactions can lock data down to the row-level

Can Capacities for Database Access be Provisioned?

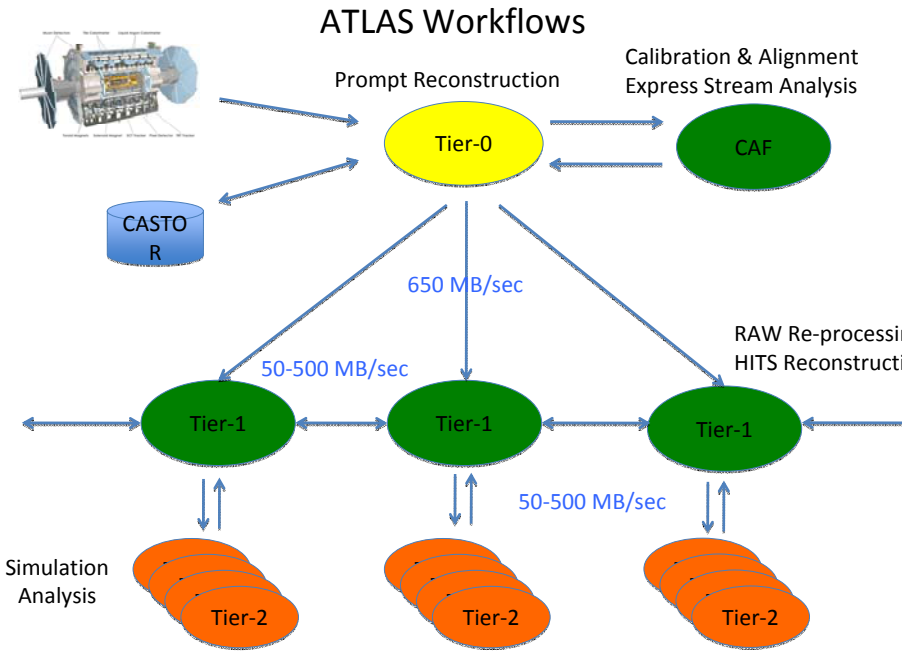
- Cloud Computing technologies may present same scalability problems in database access as the Grid Computing
- Today, Cloud Computing technologies provision CPU, memory and disk
  - A promising R&D area is a technology for complementary provisioning of the on-demand database capacities accessed by the applications running on the Could
    - A successful technology has a potential as another high impact spin-off of HEP Computing

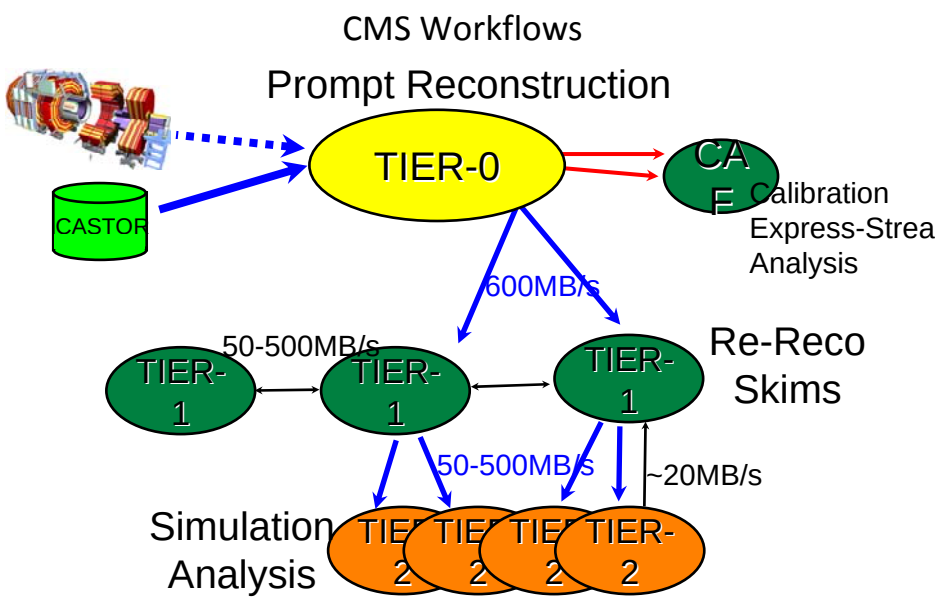
Emerging Database Technologies for Event Store

- SuperB Event Store ("Micro" data size: ~2 kB/event) vs. ATLAS TAG database (~2 kB/event)
  - Only much larger number of events in SuperB
- ATLAS is preparing an LOI for Computing R&D for upgrades
  - The current TAG database implementation (Oracle) may not scale to future challenges
    - Emerging database technologies such as SciDB are considered as a candidate for the upgrade
- Ideally, a balanced database architecture fits nicely within future memory/storage hierarchy avoiding bottlenecks in aggregating results of parallel data access in analysis:
  - on-core memory caches are progressively shared among many-core CPUs:
    - Level-1 cache: ~0.2 MB
    - Level-2 cache: ~1 MB
    - Level-3 cache: ~10 MB
- This is followed by the hierarchy of NUMA main memories:
  - Local memory: ~2 GB
  - Remote memory: ~32 GB
- Then, the hierarchy of disks for "data localization" on computing farm nodes
  - SSD: ~256 GB
  - Local disks: ~1 TB
  - Remote I/O disk storage: 1-10 PB
- Tape: 10-100 PB
- The balanced architecture for SuperB data analysis may provide high-throughput because
  - Event store data are read in parallel from the local disks via high-throughput sequential I/O
  - In contrast, data that require random disk access (software and Conditions DB) are placed on SSDs
- Would such data access pattern be supported by the SciDB architecture?

Comparison of LHC Computing Models

Similarities & Differences: CMS vs. ATLAS

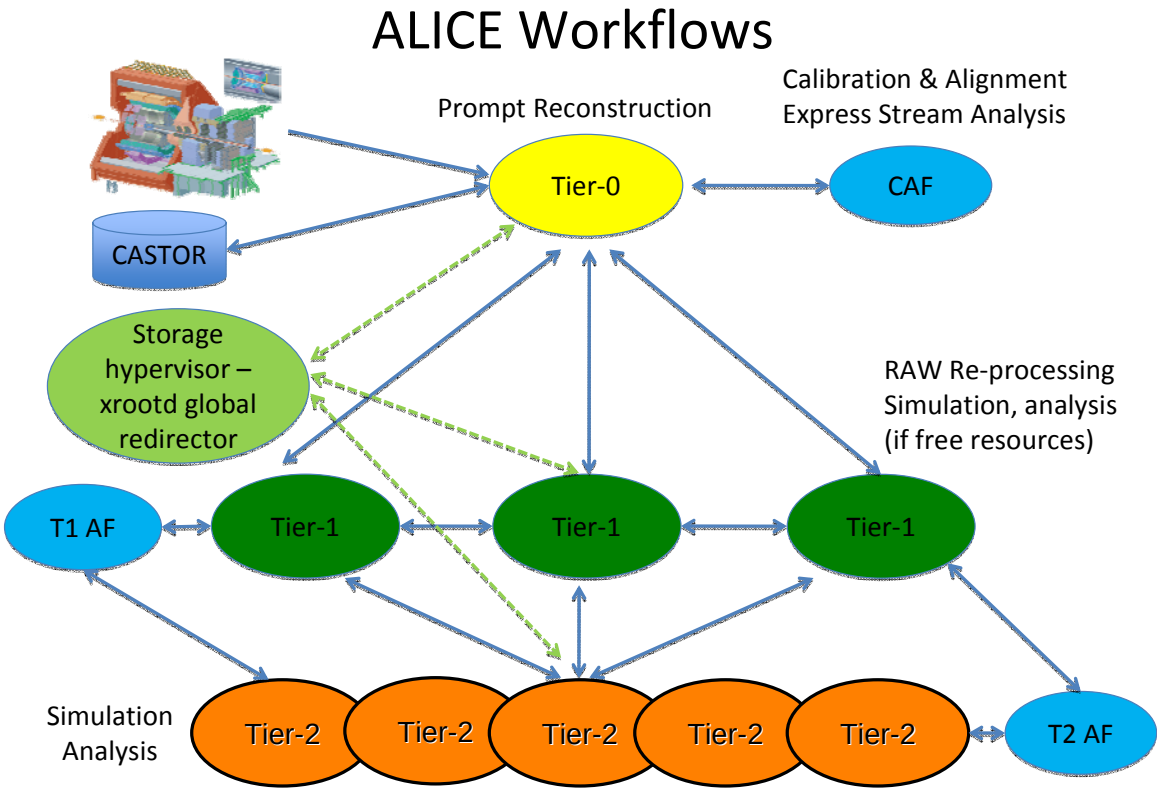




- Tier-0 and CAF very much the same functionality
- Rates are quite similar
- Functionality of Tier-1's much the same: re-reconstruction
- Functionality of Tier-2's much the same: Simulation and analysis

Slide by K. Bos

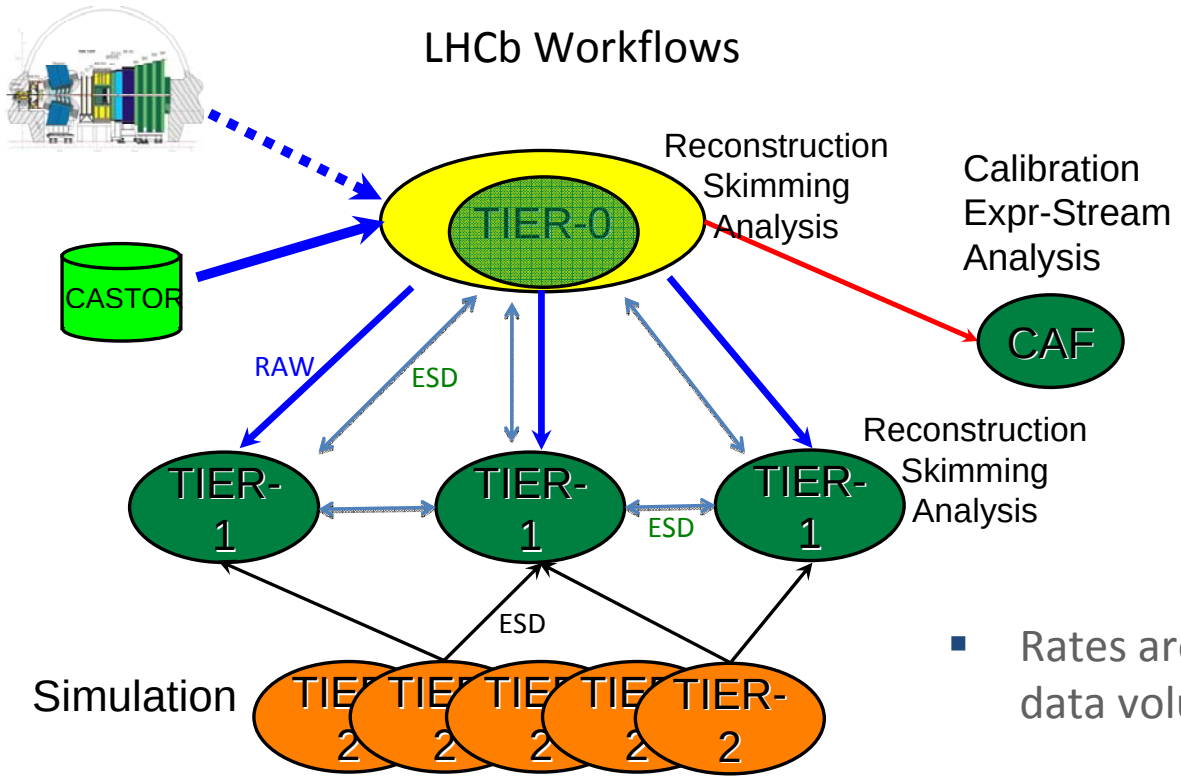
Similarities & Differences CMS vs. ATLAS vs. ALICE



- Tier-0 and CAF very much the same functionality
- Functionality of Tier-1's much the same: re-reconstruction
- If resources available, T1s can do MC and analysis (ALICE job queue prioritization)
- Functionality of Tier-2's much the same: Simulation and analysis

Slide by K. Bos

Similarities & Differences CMS & ATLAS vs LHCb



- Rates are much higher but data volume much smaller

- CAF very much the same functionality
- Different functionality of Tier-1: reconstruction, skimming and analysis
- The Tier-0 acts as another Tier-1: reconstruction, skimming and analysis
- The Tier-2's do only simulation (+digitization +reconstruction) production

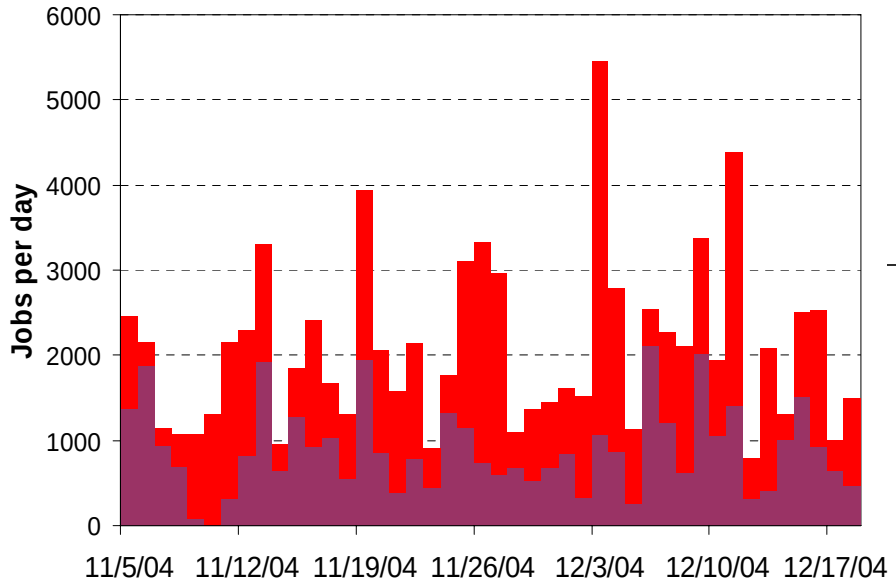
Slide by K. Bos

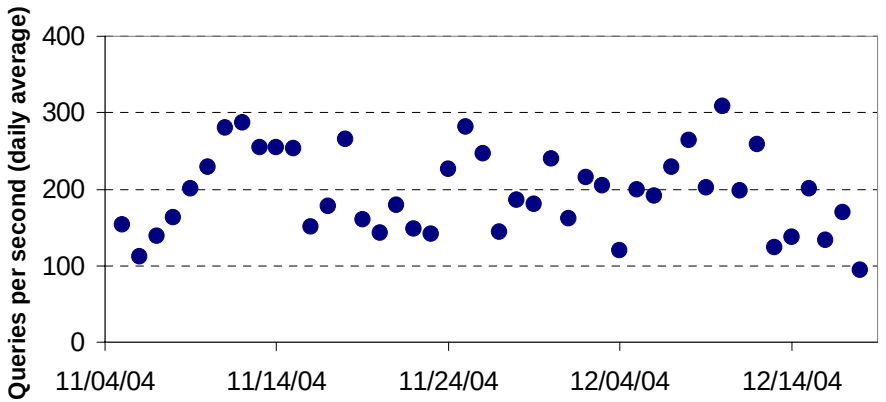
Supporting Materials

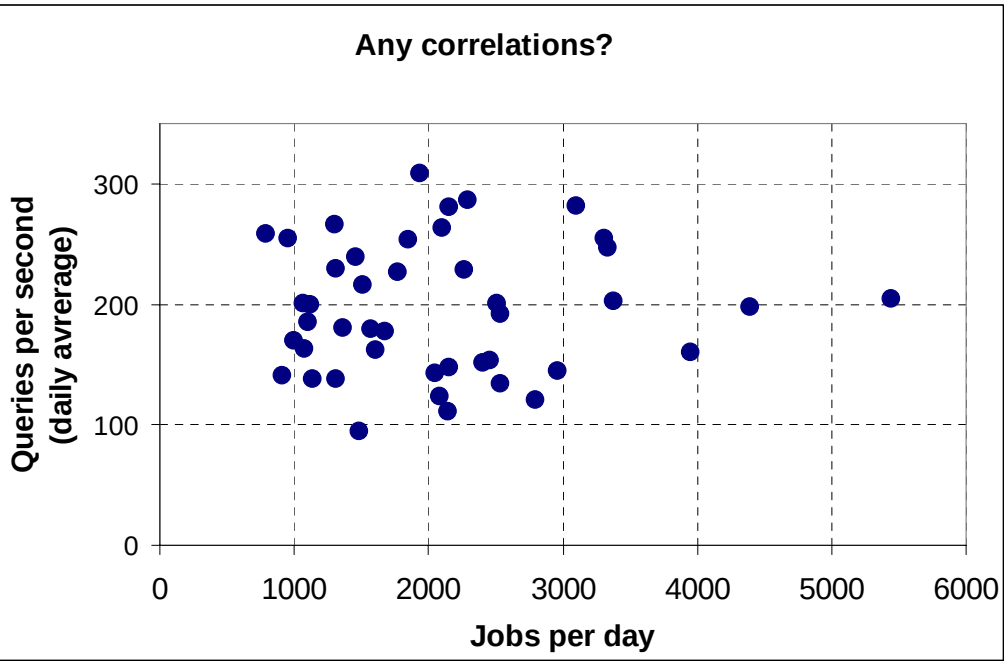
Trigger Menu Configuration DB

- Data complexity - low
  - Statistics for recent DB Release 8.7.2
    - 73 tables in 1 schema
    - Rows: 176,230
    - SQLite data volume: 14 MB
- Update frequency: “Static” i.e. upon request
- Usage:
  - JDBC tools are used to copy Oracle source to SQLite
  - Data reconstruction jobs do not access this database
- Replicated on the Grid via SQLite in validated DB Releases

Correlations: Grid Jobs vs. Database Server Load





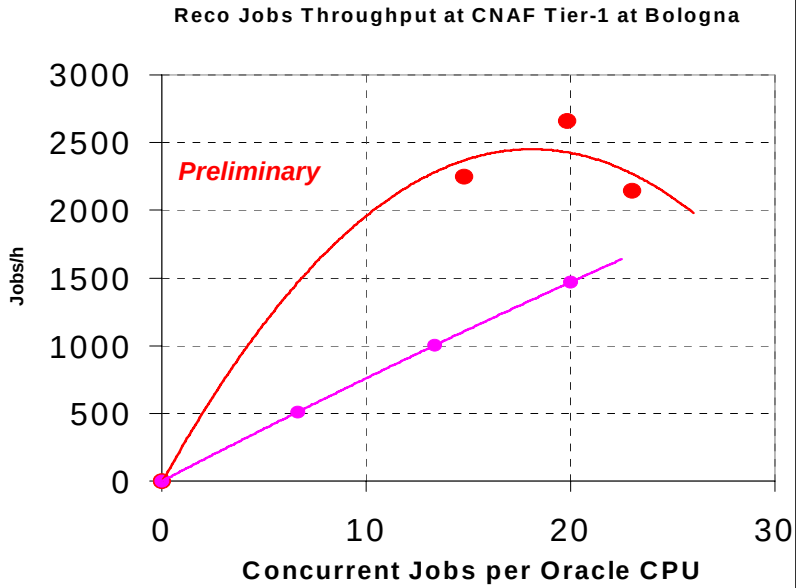


ATLAS Oracle Scalability Testing

- The goal of database scalability testing is to detect hardware limits of Oracle servers deployed at the Tier-1 sites, so that the server overload conditions can be safely avoided in a production environment
- First tests showed that Oracle capacities are sufficient for expected nominal jobs throughput
- Recent tests and operational experience in 2009 confirmed our expectations

First Oracle Scalability Tests in 2007

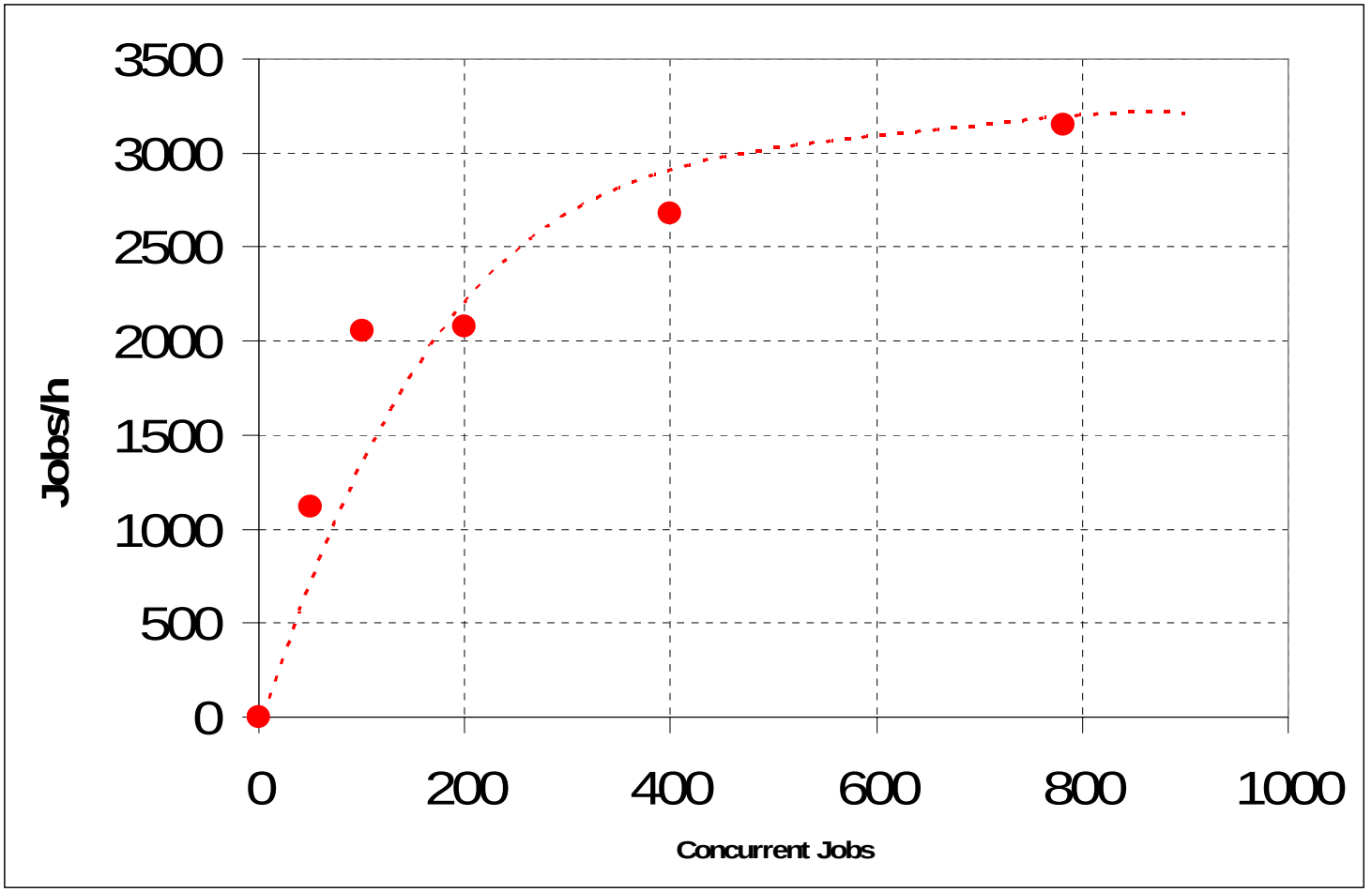
- Test jobs read realistic Conditions DB data workload at random
- We estimate that ATLAS daily reconstruction and/or analysis jobs rates will be in the range from 100,000 to 1,000,000 jobs/day
- For each of ten Tier-1 centers that corresponds to the Conditions DB access rates of 400 to 4,000 jobs/hour



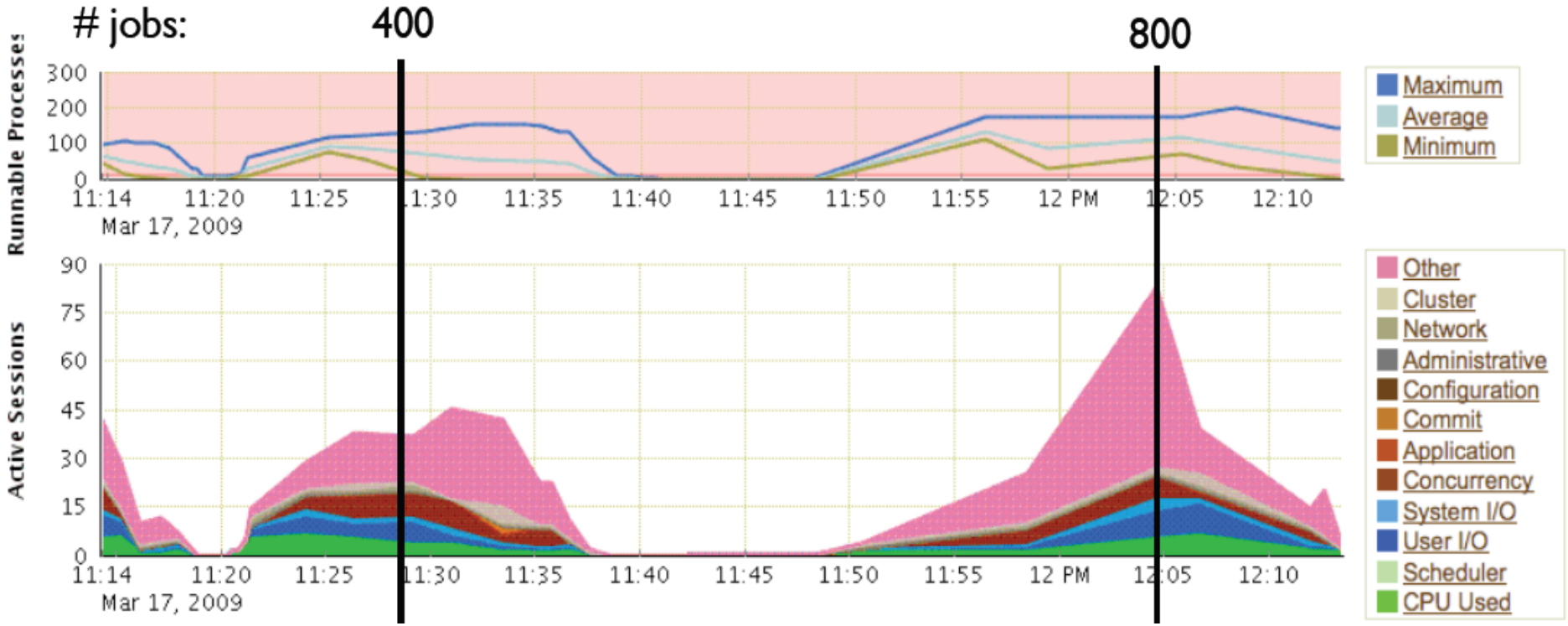
- Thus, preliminary results from the first scalability test were promising
  - We got initial confirmation that ATLAS capacities request to WLCG (3-node clusters at all Tier-1s) is close to what will be needed for reprocessing in the first year of ATLAS operations

2009 ATLAS Scalability Test Results at PIC

- 2009 scalability test results were in agreement with our previous findings:
  - Oracle capacities are sufficient for expected average jobs throughput



A Glimpse into the Many-Core Computing Era:

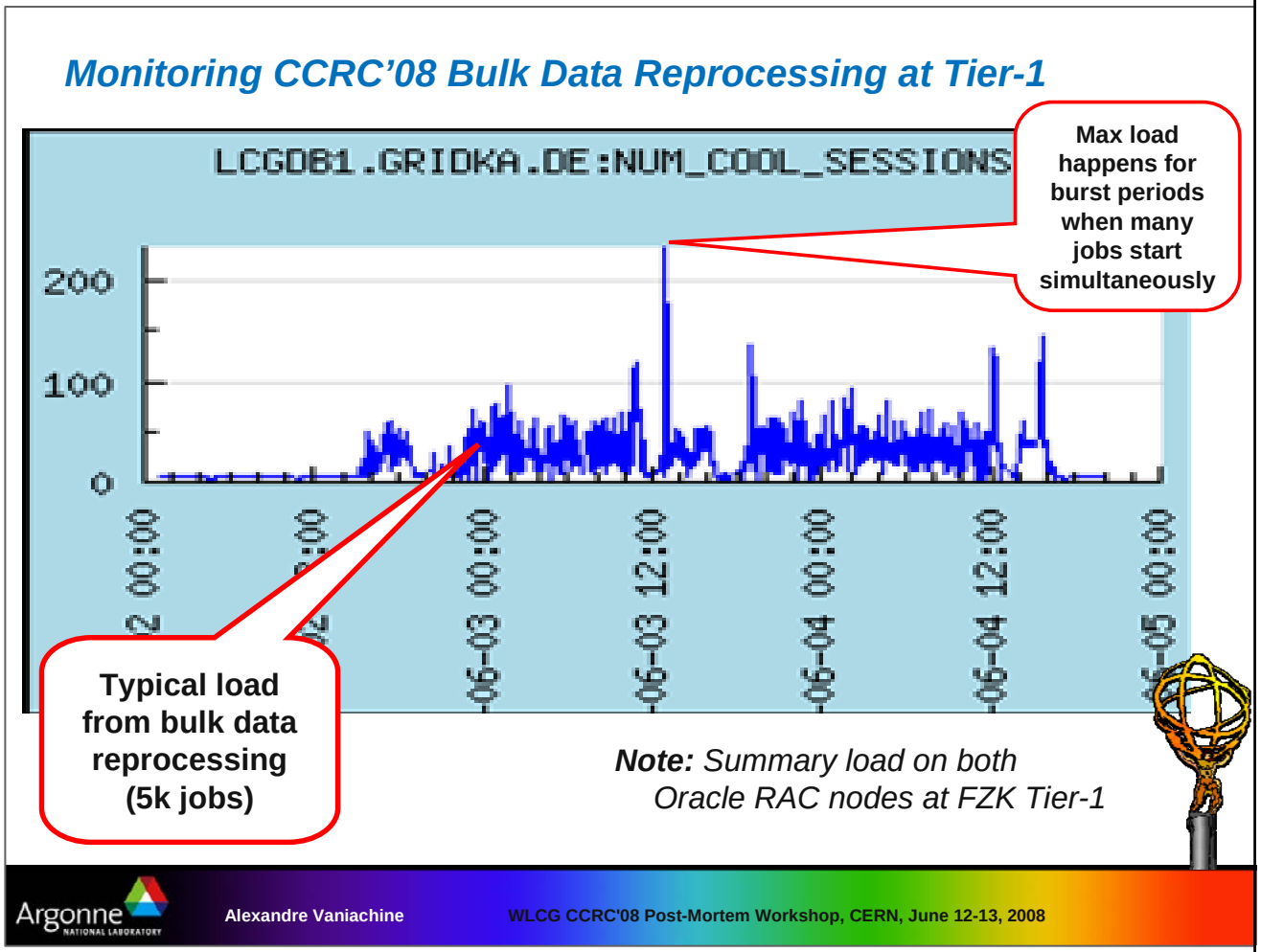


- Oracle scalability is limited by "Other" (inter-cluster traffic that provides memory locks)
- I expect that thanks to a balanced architecture the SciDB will be free from such bottlenecks

Tests by X. Espinal

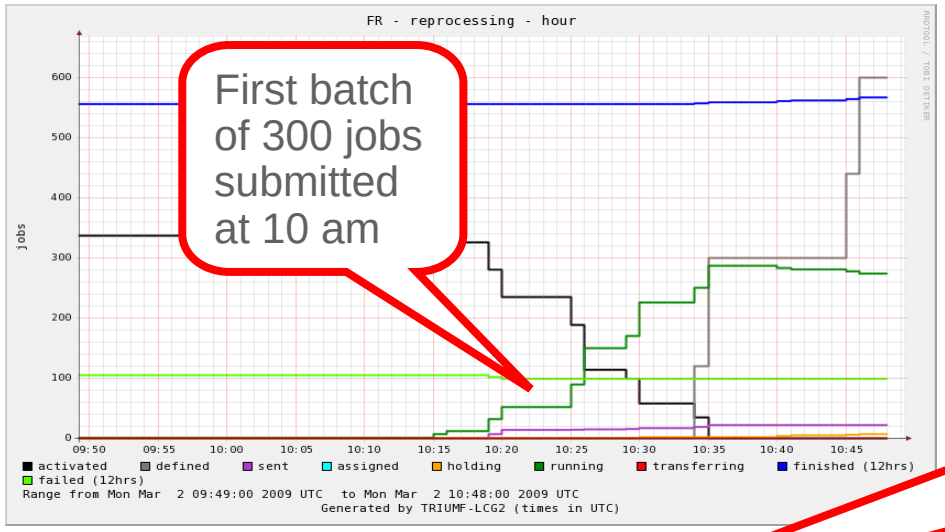
Pilot Query: Throttling Jobs Submission on the Grid

- Instabilities at Tier-1 sites may result in peak database access loads when many jobs are starting at once
- This may create overload of Oracle servers and degrade Oracle Streams replication worldwide
- To eliminate such incidents ATLAS is developing a tool to prevent Oracle overload in Grid computing:
  - "Pilot Query"



Pilot Query: Proof-of-principle Demonstrated

- Throttling Oracle server load on the Grid (at the Tier-1 site in Lyon)



First batch of 300 jobs submitted at 10 am

- Development of the next generation Pilot Query system is complete and ready for testing

Monitoring shows Oracle load limited by the Pilot Query technology

Because we set ATLAS application-specific Oracle load limit at 4