CHEP 2019

4-8 November 2019, Adelaide, Australia

Evaluation of the ATLAS model for remote access to database resident information for LHC Run 3

Elizabeth Gallas[1], Gancho Dimitrov[2]
on behalf of the ATLAS Collaboration

1. University of Oxford
2. CERN

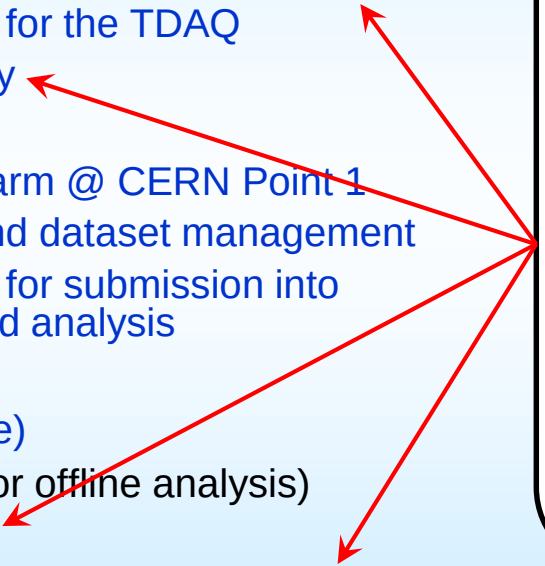
Overview

- ATLAS database usage
  - And the need for distributed database information
    - Non-event data (conditions and configuration) for offline analysis
    - The AMI factor
- Survey of evolution since Run 1 (over 10 years):
  - database & related file distribution
- Current system
  - Components
  - Developments
- Factors influencing future evolution (LHC Run 3 and Run 4)
- Summary and Conclusions

Overview – Databases in ATLAS

Database information is used extensively in ATLAS: every stage of data taking & analysis

- Configuration
  - PVSS – Detector Control System (DCS) Configuration & Monitoring
  - Trigger – Trigger Configuration (online and simulation)
  - OKS – Configuration databases for the TDAQ
  - Detector Description – Geometry
- File and Job management
  - Tier 0 – initial data processing farm @ CERN Point 1
  - Rucio / DDM – distributed file and dataset management
  - ProdSys – jobs characterization for submission into PanDa – production & distributed analysis
- Metadata catalogues such as
  - AMI (dataset selection catalogue)
- Conditions data (non-event data for offline analysis)
  - Conditions Database in Oracle
  - [POOL files in DDM (referenced from the database]
- …

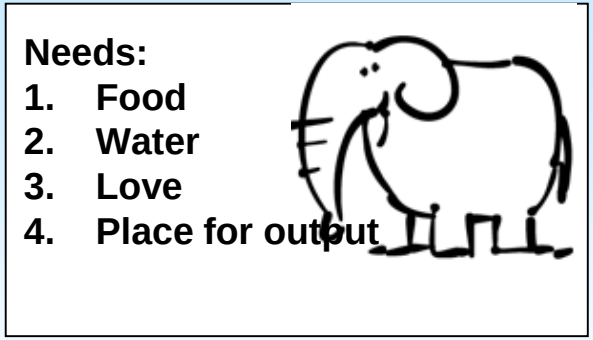


Database distribution needs
(just as true 10 years ago):
Event data processing generally requires the most up-to-date
- Configuration and
- Conditions information

It's interesting review how our distribution operations have changed in how we provide
- Database-resident data and
- File-based data it references

in order to plan for the future.

What do grid jobs need ?



What do your jobs need ?

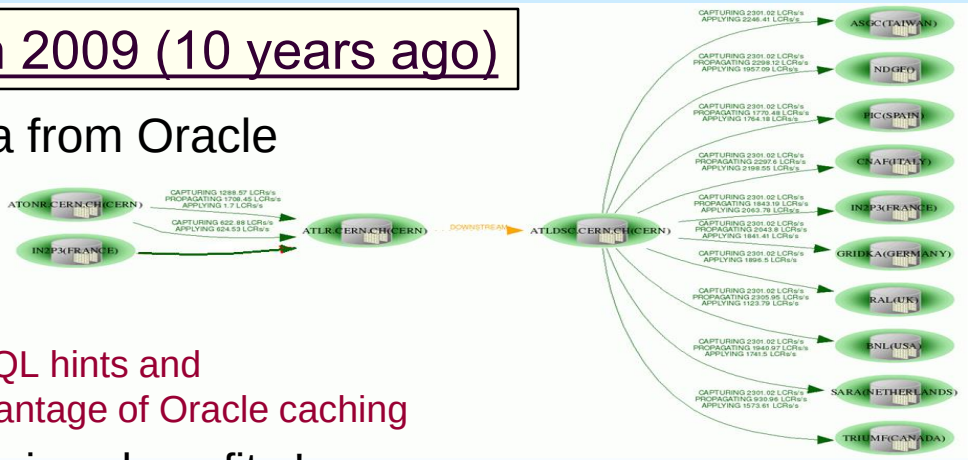
1. Data (Events)
2. Database (Configuration and Conditions)
3. Efficient I/O (sometime across a network), CPU
4. (A Purpose and a) Place for Output

To provide #2: Configuration & Conditions data (2 exclusive but related forms)

- Database resident data (Master copy resident in Oracle @ CERN)
  - Geometry, Trigger, & Conditions DBs (LCG COOL & CORAL infrastructure-based)
  - Athena is provisioned to read the database via COOL/CORAL methods
    - From the database (Oracle) directly or via Frontier / Squid
    - From SQLite files (packaged in a “DB Release”)
- Conditions Files:
  - Are referenced by “GUIDs” in Conditions DB data
  - These GUIDs point to distinct POOL files registered in Rucio / DDM
  - Athena is provisioned to request data via POOL token

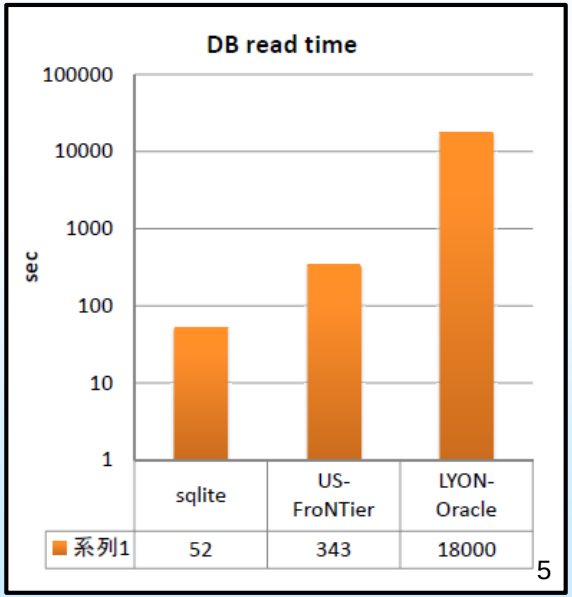
Database Replication: ‘Crossroads’ in 2009 (10 years ago)

- Pre-2009 plan: Grid jobs would read data from Oracle
  - To facilitate this, data was replicated to each of the 10 Tier-1 sites to minimize network distance to jobs
    - Also, COOL performance was optimized with SQL hints and built in IOV alignment in Athena would take advantage of Oracle caching

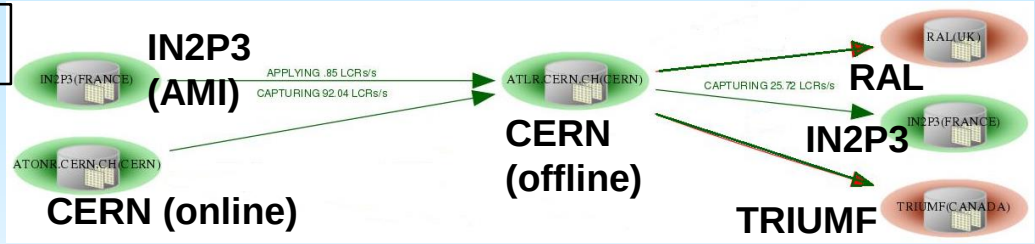


- But in 2009: Frontier evaluation showed obvious benefits ! (the Frontier / Squid system originally developed by CMS)
  - Colleagues in Tokyo ran local dedicated tests comparing DB access times using the 3 possible access methods:
    - Local SQLite files (in Tokyo, Japan) ~ 52 seconds
    - US-based Frontier site (at BNL, United States) ~ 343 seconds
    - Direct Oracle access (at IN2P3, France) ~ 18000 seconds !
  - While Oracle performance is excellent on LANs → VERY poor performance over a WAN ☹
  - Also, Frontier provides a buffer to queue requests, protecting the DB

→ The decision was therefore taken by ATLAS to deploy a Frontier / Squid infrastructure world-wide



Database replication: then and now



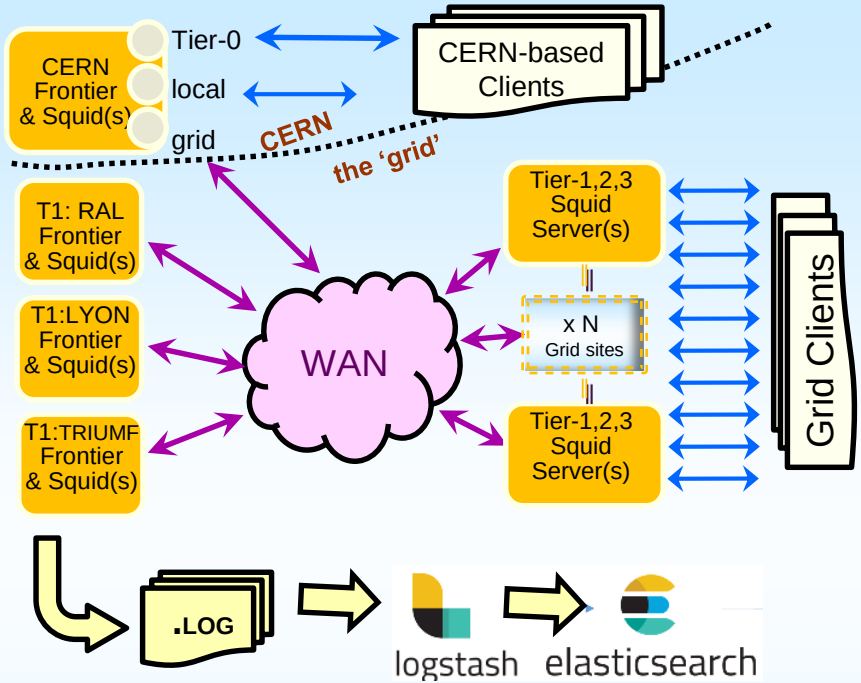
- Since the decision in 2009
  - 7 (of the 10) Tier 1 sites previously hosting Oracle replicas have shifted contributions toward storage and CPU
  - This has consolidated replication to the remaining sites, turning the focus toward improved Frontier / Squid deployment
- Currently: 3 Tier 1 sites are hosting Conditions replicas
  - RAL (United Kingdom)
    - phasing out Oracle support in 2020
  - TRIUMF (Vancouver, Canada)
    - Provides a Conditions replica in North America
  - IN2P3 (Lyon, France)
    - Provides essential conditions CERN failover
    - Inversely: replicates its AMI database to CERN

- Current smooth operations rely on the remaining replicas to provide
  - Geographical distribution
  - Load balancing
  - Failover redundancy

As for the Future …

- During LS2, Oracle sites upgrade to 19c
  - A corresponding upgrade in Oracle Streams replication is also needed.
- Oracle licensing changes are expected on the Run 3 timescale
  - Adjustments in site-wise resources will balance licence boundaries and cost with requirements.
- Frontier / Squid infrastructure continues evolve in terms of deployment, failover and monitoring (next slide …)

Frontier / Squid evolution



- Frontier / Squid deployment includes
  - Frontier launchpads at Oracle sites
  - Squid servers across the grid to serve the requests of clients

- In recent years, an Elasticsearch repository has been deployed at the University of Chicago
  - Collecting information from Frontier Launchpad site logs for the purpose of studying these transactions
- This repository has served many uses already (as described other CHEP presentations)
  - With regard to this presentation, we can analyze the overall load carried by each of the Oracle sites
    - For example in the recent period
      - CERN satisfied ~55% of requests while IN2P3 ~ 26% and TRIUMF ~17%.
  - A reconfiguration of the repository is underway, so we plan more in-depth studies using this data in future
    - For example: it would be useful to study the advantages of geographic distribution to the performance seen by clients

Files based Conditions distribution (and evolution)

We have 2 categories of file-based data with distribution requirements:

1. "Conditions files": File-based conditions data referenced in-line by GUID in the Conditions Database
   - Subsystem-wise data which was determined to be better suited to POOL files than to store in-line in the database
2. "DB Release" files: An aggregation of both SQLite database replicas packaged along with the associated above Conditions files referenced by the database
   - Such files allow event processing jobs to execute without network access such as on HPCs.

- Both are:
  - Registered in Rucio/DDM (the ATLAS distributed file and dataset management)

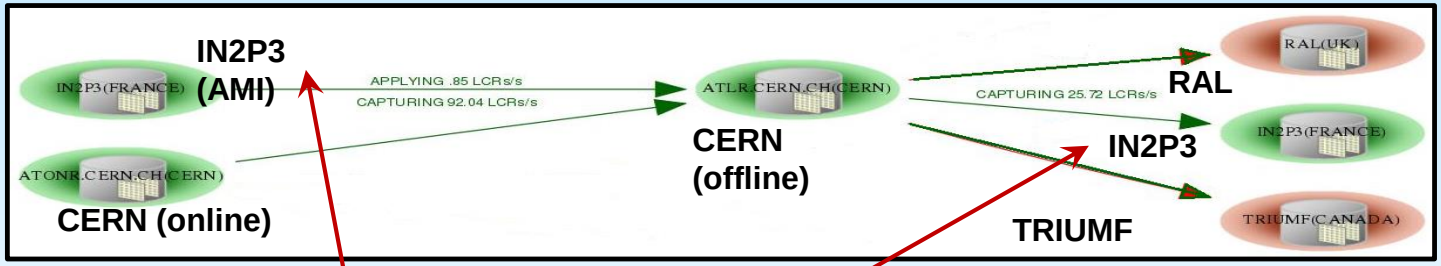
Originally:

- conditions files were distributed to sites via POOL file registration
  - Sites needed a 'hotdisk' to make files available to local jobs
  - A corresponding PFC (Pool File Catalog) was needed
- DB Release files were distributed to sites running the job (MC)
  - This eventually became unsustainable due to cumulative evolving MC conditions

Currently, these file systems are readily available via CVMFS

- Big improvement over maintaining file systems and updating the catalogs.
- DB Release production has been streamlined – to contain only the latest 'best knowledge' MC conditions data
  - A manageable data volume on grid nodes

The AMI factor



Note: The IN2P3 (Lyon, France) site is both a data source and destination site

- Destination: Conditions & Configuration data
- Source: AMI (ATLAS Metadata Interface)

AMI: a long history in ATLAS and CHEP!

CHEP 2013: 10 year anniversary retrospective

- Has continued to evolve with latest technologies and extend for new use cases
- Proven resilient, robust, scalable, adaptable
- Mature ecosystem:
  - Shared with multiple experiments

Developed by our colleagues in Grenoble

- Master AMI database @ IN2P3 (Lyon)

For ATLAS, the main component provides dataset-level metadata services

ATLAS AMI data is replicated to CERN providing

- Fall-back/secondary services
- Data source for other CERN-based applications for example other ATLAS metadata repositories
  - Run-level (COMA)
  - Event-level (EventIndex)

Summary and Conclusions

- In ATLAS: The basic requirements for the distribution of database-resident data and associated files remains unchanged.
  - Some perspective is gained in reviewing the evolution of the components of systems described here
    - Many of which are described in this (and previous) CHEP Conferences
  - Oracle replication on the grid has been consolidated to just a few dedicated sites
    - The Frontier / Squid deployment has been adjusted accordingly
    - Studies of request logs are proving very fruitful to inform future database design.
- Through Run 3, Oracle is the current chosen DBMS for database-resident conditions and configuration data
  - This follows naturally from the decision to continue to use CORAL and COOL infrastructure through Run 3
  - But this an open issue for Run 4 with the proposed deployment of a new system for conditions-related storage (CREST): **a RESTful and more cache-efficient system.**
    - With such changes we expect a considerable shift in the future database distribution model.
- The move toward RESTful database services and cache-compatible payloads are probably the most significant factors in the evolution of database distribution schemes for the experiments of the future.