

# **Geometry DB News.**

## **DB Release Status & Outlook.**

Vakho Tsulaia  
University of Pittsburgh

ATLAS Software&Computing Workshop  
CERN, 04/07/2011

# Outline

- **Geometry DB**
  - Optimizing number of queries
  - Geometry DB in Frontier
- **DB Releases**
  - Status
  - Few thoughts about the future of database releases

# Geometry DB. Number of queries

- Problem: Athena jobs issue **too many** queries to the Geometry DB
  - Typical reconstruction job sends **~2.5K** queries for reading **~250** data tables
  - **~90%** of these queries access **3 master tables** in the database in order to map a given global ATLAS tag to the tags of data tables (**root-2-leaf** mapping)
- Solution: introduce a special schema object in the database (see next slide), which will **cache** the root-2-leaf mapping for **all supported and locked ATLAS tags**.
  - Such cache is queried only once per job, thus the total number of queries goes down to **~300**

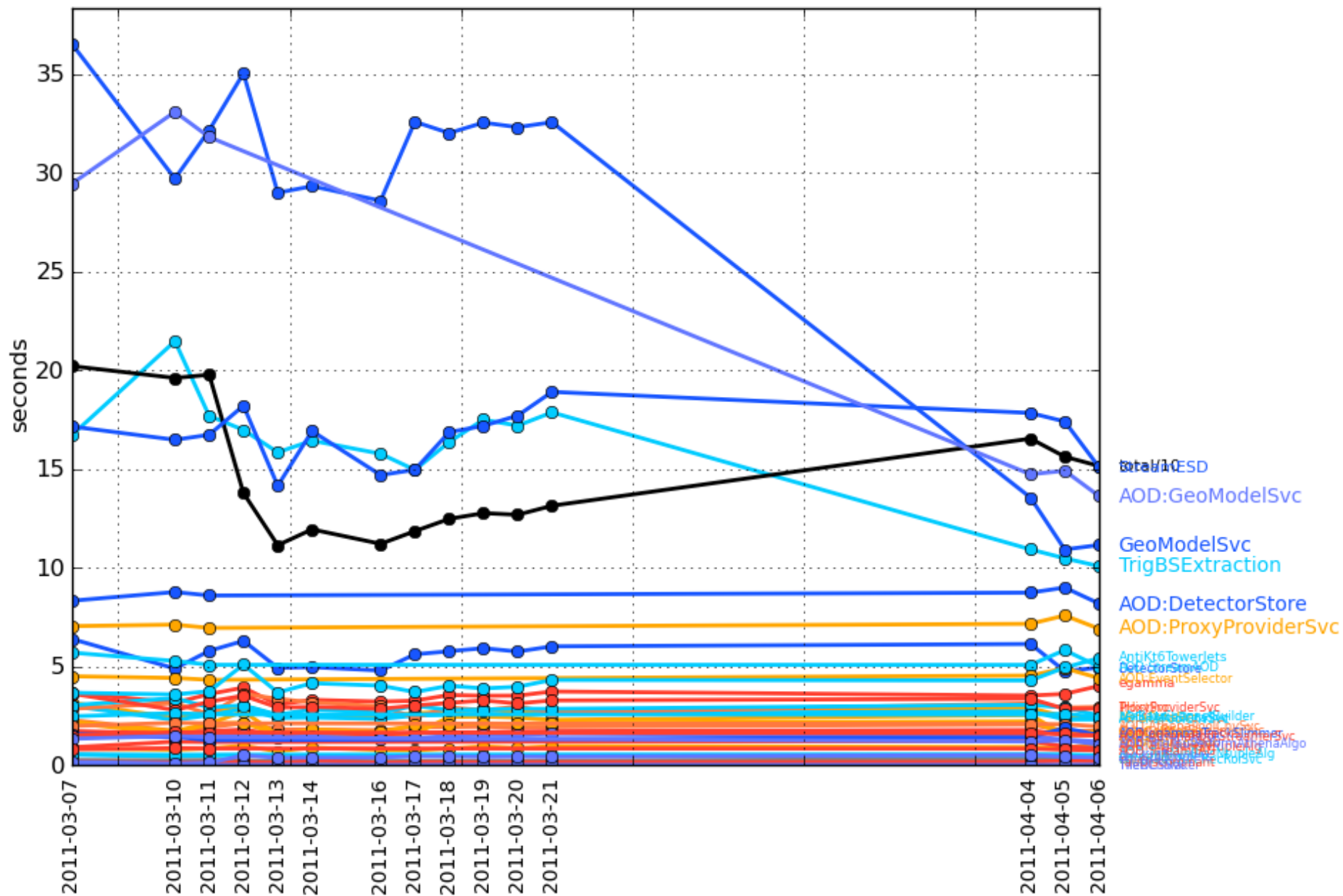
# Geometry DB. Number of queries. *Cache implementation*

- Two options considered for implementation of the tag cache:  
*Materialized View* and *Ordinary Table*
- **Materialized View**
  - Does not work with Oracle Streams ... **rejected!** (seemed rather attractive otherwise)
- **Ordinary Table**
  - **HVS\_TAGCACHE** table introduced in the Geometry DB
  - Table initially loaded with caches for all existing supported locked global tags
  - Implemented a **stored procedure** for adding caches of new tags incrementally
  - The stored procedure is **called automatically** when a new **global tag gets locked** by the Geometry DB Web-based management utility

## Geometry DB. Number of queries. *Status*

- Geometry DB schema modifications were first successfully tested on INTR, then migrated to ATLR
- Updated **RDBAccessSvc**, which uses the HVS\_TAGCACHE table, is in **dev nightly** since ~2 weeks
- The effect of this optimization is clearly visible for the jobs, which read geometry data from Oracle (RTT, interactive jobs with nightlies). See ***next slide***
  - For SQLite the speedup is also there, but less visible (~20%)

CPU overhead contributions in RAW->ESD->AOD [seconds]



# Geometry DB and Frontier

- Several *proof-of-principle* tests for reading geometry data via **Frontier** done in the past.
  - The most important one done by **Andrea Valassi** in July 2010. **Several bugs fixed in CORAL**, which were preventing the correct reading of the geometry data via Frontier
- However, before rolling out the geometry DB access via Frontier to everyone it was necessary to **test** that it **does not add big overhead to the existing Frontier load caused by the Conditions data access**
- Such test was carried out last week (March-31) and it proved that the additional impact on Frontier caused by the geometry data is small
  - More details about the test in backup slides

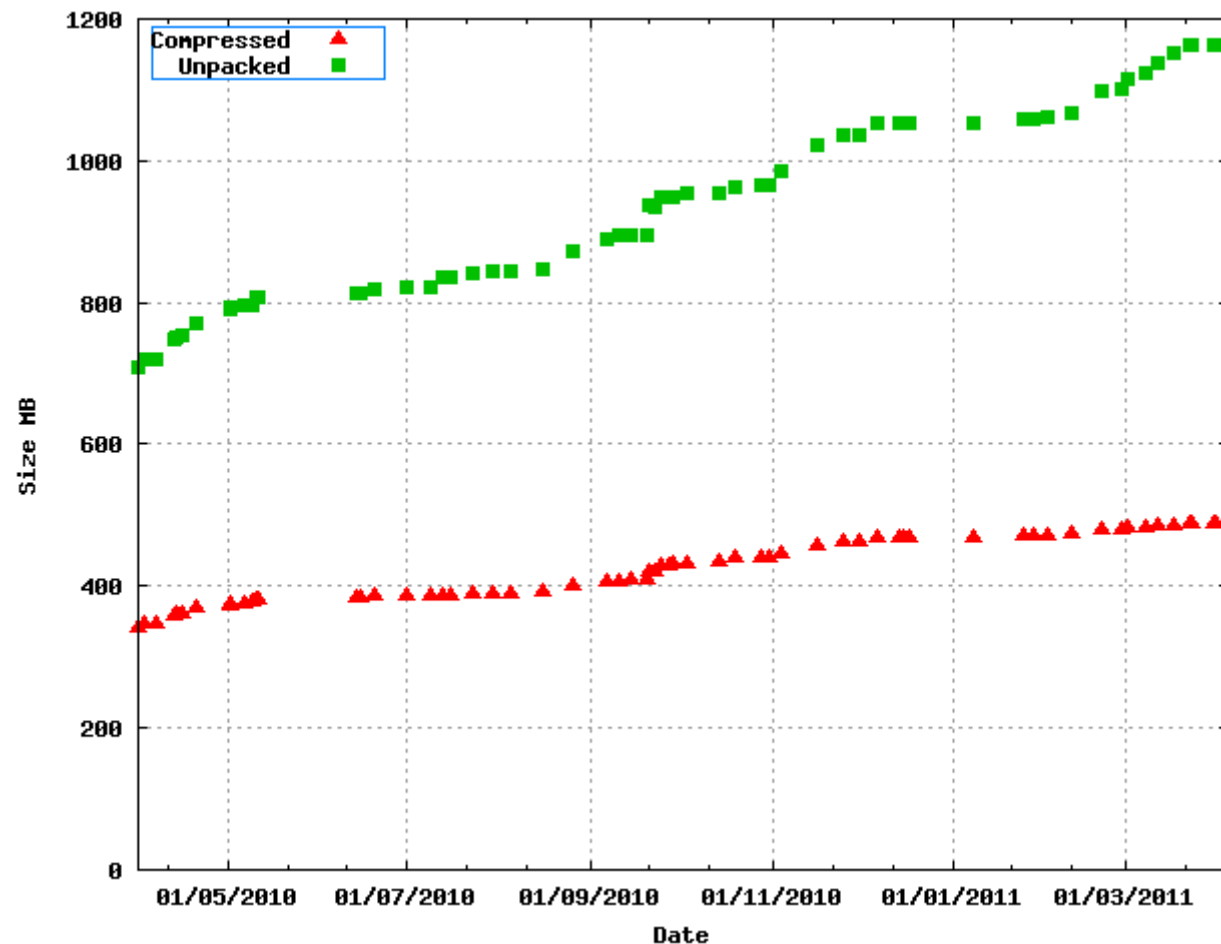
# Geometry DB and Frontier. Status

- For the moment the Geometry DB is available via Frontier **only at CERN**
- The next step is to put Geometry DB into **Oracle streams** replication from T0 to T1s. As a result it will become available via Frontier worldwide
  - Next week
- Then we need decide on the strategy of the Geometry DB reading from Frontier (**Frontier vs SQLite**) and make appropriate **changes** in the software (**AtlasAuthentication, DBReplicaSvc**)
  - It's not unrealistic to have this done by **rel 17**



# DBRelease. Status

- **Stable infrastructure.** No changes since the introduction of 'current' database releases last Summer
- Some statistics for the period **from April 1<sup>st</sup> 2010 to April 1<sup>st</sup> 2011:**
- Total of **64** DB Releases built
- Size (packed/unpacked) grew from **343/708** MB to **489/1163** MB
  - **Conditions DB** (SQLite + POOL files) contributes to **90%** of the size
  - Geometry and Trigger DB have ~5% each



## DB Release. Outlook

- **Having the geometry DB available in Frontier offers several possibilities to rethink the overall strategy of DB Release usage in ATLAS**
- We can drop the current *hybrid* database access model in analysis jobs
  - The analysis jobs should be able to read both Geometry and Conditions from Frontier
  - No more need to download & unpack 1.2G DB Release only for getting access to 50MB Geometry DB replica

## DB Release. Outlook - 2

- We can also switch from DB Releases to Frontier in **physics validation tasks**
  - No need to wait until new DB Release is made/validated/distributed on the Grid. May take up to 1 day
  - No need to go through the same procedure again if some problems have been identified with either Geometry or Conditions
- This would require, however, the availability of the **Trigger MC DB** in Frontier
  - Joerg seems to be rather positive about this idea
- Not sure whether the prompt replication of **POOL conditions files** can be an issue here...

## DB Release. Outlook - 3

- Should we consider **removal of database releases from software kits?**
  - But keep the possibility for the users to easily install and use DB Releases if needed
    - This would require some CMT 'magic', hopefully doable
- **Pros:**
  - **Disk space.** Example 16.6.3 uses 8.9G on the releases area, out of which 1.2G (13%) is taken by DB Release 15.1.2
  - Once this is done, we can consider **removing obsolete** geometry and conditions **tags** from the database release without worrying too much about **backwards compatibility**
- **Cons:**
  - ... any?

## DB Release. Outlook - 4

- I am **not** proposing to decommission database releases
- Database releases (both MC and Conditions) play the key role in the production system (reprocessing, MC production)

# Summary

- We can benefit from the geometry database availability in Frontier as early as in release 17
- This should allow us to reconsider database release usage strategies in several areas, like
  - Physics validation
  - Analysis jobs
  - Software kit distribution
- DB Release usage by the production system should remain unchanged

# Backup

# Testing Geometry DB access in Frontier

- Test carried out March 31, 2011, when T0 resources were available because of the LHC downtime
- **Main purpose:** to prove that by reading geometry data via Frontier we don't add big overhead to the existing load on Frontier servers caused by conditions data access
- **Strategy:** run two identical rounds of reconstruction jobs and compare results
  - **Run 1.** Conditions from Frontier, Geometry from SQLite
  - **Run 2.** Conditions and Geometry from Frontier
- However, jobs in the Run 1 were unexpectedly slow, so we reran the Run 1 after Run 2 was over (referred to as **Run 1'** in the next slides)



# Testing Geometry DB access in Frontier (2)

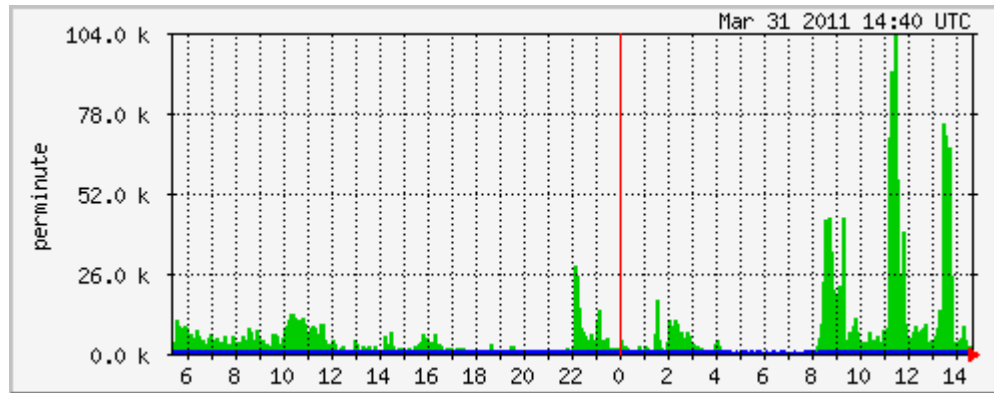
- Job configuration details
  - **Input dataset:**  
data11\_2p76TeV.00178229.physics\_Background.merge.RAW  
(837 files, 1934 events ==> 2.3 events/file on average).
  - **Job:**  
custom configuration, modified version of AMI x103 tag.
  - **S/w version:**  
AtlasProduction-16.6.2.6 with patch of RDBAccessSvc for optimizing number of queries to the geometry DB
  - **Frontier access:**  
FRONTIER\_SERVER="(proxyurl=http://atlassquid1.cern.ch:3128)  
(serverurl=http://atlasfrontier1.cern.ch:8000/atlr)"

# Testing Geometry DB access in Frontier (3)

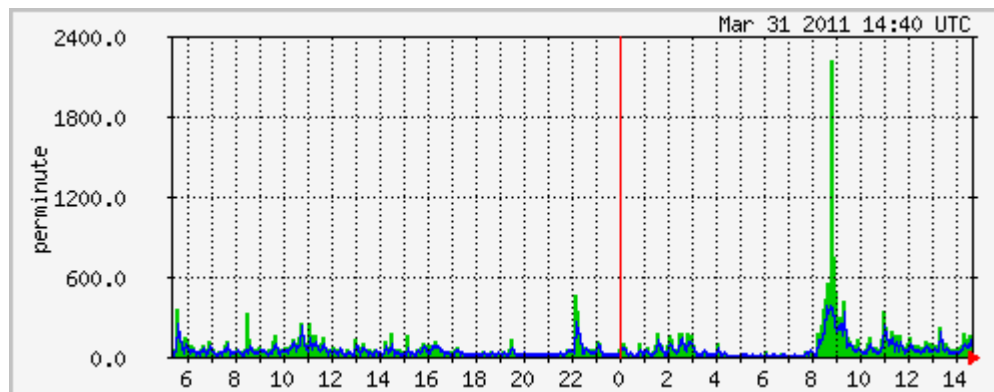
- Results (figures)
  - **Average job execution wall times** (without stagein/stageout, taken care of by the Tier-0 wrapper):  
(Run1) 1923s (Run2) 276 s (Run1') 269 s
  - **Number of queries to Frontier per job:**  
(Run1) ~2.4 K (Run2) ~2.9 K (Run1') ~2.4 K
  - **Data transfer from Frontier per job:**  
(Run1) ~15.2 MB (Run2) ~16.2 MB (Run1') ~15.2 MB

# Testing Geometry DB access in Frontier (4)

- HTTP Hits/Requests



atlassquid1.cern.ch

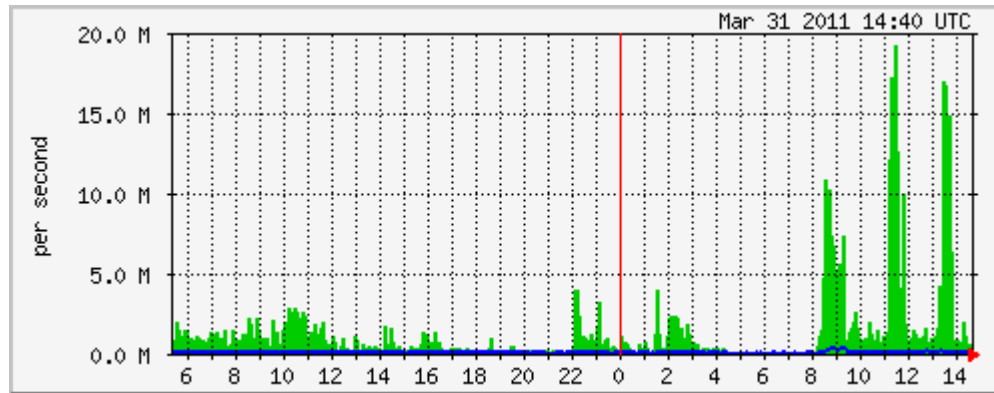


atlasfrontier1.cern.ch

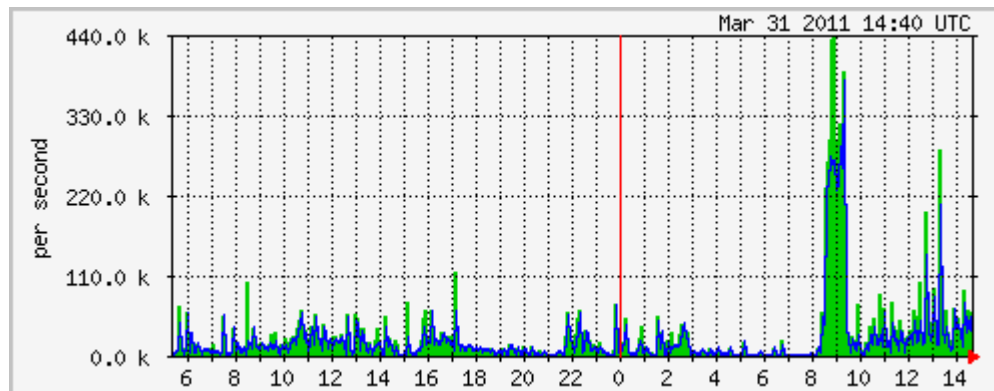
R1 R2 R1'

# Testing Geometry DB access in Frontier (5)

- Server traffic volume (In/Out)



atlassquid1.cern.ch



atlasfrontier1.cern.ch

R1 R2 R1'

# DB Release size breakdown, version 14.9.1

