Reducing number of queries to the Geometry DB

Vakho Tsulaia
University of Pittsburgh

ATLAS S&C Workshop
CERN, 02-Dec-2010

Number of Queries

- Typical Athena job needs to retrieve data from **~250 tables** in the Geometry DB for building detector geometry in GeoModel
- But the total **number of queries** is **~2500**!
  - Huge fraction of them are auxiliary queries, which are executed only for obtaining Global Tag to Child Tag mapping
  - 3 tables, which implement HVS in the Geometry DB, are heavily accessed by these queries
- All such queries can be avoided altogether by introducing a special '**Tag Lookup table**' in the database:
  - Should contain (**Global Tag, Child Tag**) pairs for ALL child tags linked to the given global tag
  - This table should be **queried only once** per job and the **result set cached** by the RDBAcccessSvc

Changes in the Database

- Need to define structure for the Tag Lookup table in Oracle
- Need to come up with tools and strategy for loading data to this table
- One possible scenario:
  - Implement a stored procedure in Oracle, which will generate lookup for given global tag (**DONE**)
  - At the first stage generate lookups for all currently **supported ATLAS tags** using this stored procedure
  - Make lookup generation part of the tag locking procedure for future ATLAS tags

Changes in the Client Code

- Extend functionality of the RDBAccessSvc (local implementation exists):
  - At early stage of the job retrieve lookup information for the given global geometry tag and cache it in memory (std::map)
  - Use the lookup table as primary source for obtaining Global Tag – Child Tag mapping
  - Fail over previous mechanism of deducing Child Tag information if
    - **Lookup table for the given global tag does not exist yet (new global tags, which have not been locked yet)**
    - **Child tag has been requested either with subsystem overrider (not supported by production job, OK for testing/development) or with explicit tag for the leaf node**
- Such RDBAccessSvc can work transparently with any global tag with or without lookup table

Test

- Build tag lookup table for **ATLAS-GEO-16-00-00** in Oracle and SQLite
- Use two versions of the RDBAccessSvc (dev nightly and updated one)
- Run simple test job, which does nothing besides GeoModelSvc initialization ('Reco Geometry' with all subsystems involved)
- Monitor performance metrics of the GeoModelSvc initialization

	Dev Nightly	New code
Total # of queries	**2535**	**290**
CPU SQLite (sec)	**9.9**	**7.7**
CPU Oracle (sec)	**32.3**	**10.4**

Next steps

- **Short term**

  1. More testing for the new code. Then put everything into release (maybe too late for 16.5.0)
  2. Implement DB infrastructure
     - **Build lookup tables for all currently supported geometries**
     - **Extend existing tag locking procedure by adding tag lookup builds for global geometry tags only**

- Steps 1. and 2. are decoupled, thus implementation order not important

Next *possible* steps

- **Longer term** (after done with the previous stage)
  - Make Geometry DB available in FroNTier
  - I have almost no experience with FroNTier, hence I don't know how much effort will be required …
  - … on the other hand
    - **We already have quite a few FroNTier experts in ATLAS, hence getting support and assistance should not be too hard**
    - **Andrea Valassi already successfully tested Geometry DB access via FroNTier last summer**

Next *possible* steps

- **And when we have Geometry DB fully functional in FroNTier**
  - Use FroNTier as default access mechanism for the Geometry Database
    - **The same way we do it for the Conditions DB**
    - **Get rid of the 'hybrid' access mode**
  - Keep MC Database Releases for
    - **MC productions on the Grid**
    - **'Private usage' (like 'Traveling laptop' mode)**
  - Drop the dependency of distribution kits on Database Releases
    - **Will that work for the trigger DB??**
    - **What about various configuration files distributed within DB Release??? (tnsnames.ora, dblookup.xml)**

Final remarks

- Please consider the two previous slides a 'raw' proposal
- Hopefully this will be subject for a discussion
- We need to agree on feasibility of such strategy and then work out a detailed plan
- Feedback and criticism more than welcome