Geometry DB Access Optimizations

Vakho Tsulaia
University of Pittsburgh

- Reducing number of queries to the Database
- Various optimizations on the client side
  - Smart pointers to RDBRecordsets (memory)
  - RDBQuery objects (CPU)

ATLAS S&C Workshop
CERN, 01-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 some kind of auxiliary queries, which are executed for mapping global ATLAS tag to tags of its children nodes
- All such queries can be avoided altogether by introducing a special '**tag lookup table**' in the database:
  - Contains (Global Tag, Child Tag) pairs for ALL child tags linked to the given global tag
  - This table is queried only once per job and the result set is cached by the RDBAcccessSvc

Number of Queries

- I have built such lookup table for one global tag ATLAS-GEO-16-00-00, updated code of the RDBAccessSvc package (not in SVN yet) and looked at performance metrics of the GeoModelSvc initialization

SQLite access:

	Dev Nightly	New code
Total # of queries	**2539**	**293**
CPU sec	**9.9**	**7.7**
VMem MB	62.56	62.75
NAlloc M	6.5	6.3
Malloc MB	434	406

Number of Queries

Oracle access:

	Dev Nightly	New code
Total # of queries	**2535**	**290**
CPU sec	**32.3**	**10.4**
VMem MB	98.9	98.9
NAlloc M	6.5	6.3
Malloc MB	336	316

Machine/platform: *Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz 2MB cache / i686-slc5-gcc43-opt*

Release: *Dev nightly rel_4 (25-Nov)*

Job options: *GeoModelSvc/jobOptions.BuildGeometry.py*

Other optimizations

- **Memory**. Migration to shared pointers to RDBRecordsets
  - Allows for releasing RDBRecordsets from memory when they are no longer needed
  - Core infrastructure put in place ~1 year ago. Migration of all major clients finished recently.
  - Total gain in VMem **~20MB**
- **CPU**. Wider usage of the RDBQuery objects
  - RDBQuery allows clients to bypass RDBRecordsets if the data will be finally kept in user defined structures (avoids unnecessary copying)
  - Initially developed for LAr, interface updated recently for wider usage
  - **10%** speedup of MuonGeoModel initialization after migration to RDBQuery

Next steps

- **Number of Queries**
  - More testing for the code. Then put everything into release
  - Implement DB infrastructure
    - **Build lookup tables for all currently supported geometries**
    - **Develop automated procedure of building lookups for new global tags**
- **Memory optimizations for Reco jobs**
  - Release Muon Dead Material description after it has been used by the Tracking Geometry