DIANA: Nebraska Activities

Brian Bockelman
David Abdurachmanov
Recent Activities

• Compression work:
  • Intel QAT hardware (DavidA). Does compression in-silicon for a potentially-large speedup: getting access to hardware from ROOT has been surprisingly difficult.
  • zlib libraries improvements (DavidA). Combines and improves on several zlib patchsets; selects correct optimizations at runtime.
  • Alternate compression mechanisms (Zhang). Alternate compression libraries, per-event compression, etc.
• ROOT fast IO project (Bockelman). Deserialize many events at once, avoiding overheads.
• TTreeProcessor (Bockelman).  https://indico.cern.ch/event/586607/contributions/2370356/attachments/1371340/2079997/TTreeProcessor-Introduction.pdf
• Implicit Multi-Threading (IMT) for writing output. Allows compression to proceed in parallel when flushing multiple baskets.
Impact

• Impact has mostly been limited to the local community:

  • Two CHEP papers (Abdurachmanov & Zhang) by DIANA team.
  
  • zlib improvements put into CMSSW DEVEL build.
  
  • DIANA forum presentation on ROOT TTreeProcessor. Trying to engage more the CERN summer student and staff on this approach.
  
  • Working with CMS to get the IMT branches in shape for release.
Throughput on KNL

Throughput (events/sec)

RECO Throughput KNL; N threads = 1.5 * N streams
Expanding Impact of Past Work

• Basically, focussed around getting merge requests from the last few months merged (PR240, PR146, PR134, PR136).

• With Philippe, pushing to have a dedicated ROOT IO biweekly meeting (more frequent complement to the biannual ROOT IO workshops). Gather a bit more energy around the topic.
Development Work in the next 6 months

- Library of LHC Run II & analysis file samples.
  - Want to build simple benchmarks around this.
  - Important! Explicitly mentioned deliverable to NSF. Already have collected recipes to generate MC for LHC experiments.
- Upstream zlib contributions to Cloudflare and/or Mark Adler.
- Finish off items in the ROOT IO plan (https://docs.google.com/document/d/1iJfDdkOdR2zUvsnR_EffSCwKDg6_EElsmZb9VO5dL_E/edit#).
  - Allows fast copy for integral types and C-style arrays of types.
- Mature the TTreeProcessor into a framework ready for inclusion to ROOT.
Ideas for how we can collaborate within the DIANA

• The TTreeProcessor approach has significant overlap with work Jim is doing.

  • I’d like to see it serve as a C++ foundation to equivalent work in higher-level languages.

• Working with CERN’s OpenLab, DavidA has significant experience with alternate architectures & GPUs - is there any overlap with the Cincinnati team?

  • Note: majority of this work is under NDA, making it difficult to discuss in public forums.
Plans for >6 months

- Unlike TBranch::SetAddress, the “new”-style TTreeReader interface knows the leaf class at compile time.
  - This means IO could be done at event cluster boundaries, but deserialization could be inlined with user code (one less copy between main memory and caches).

- Work through merging of the various “sticky” IO format improvements (such as reduction of object overhead).

- Based on results of the fast IO project, start tackling more complex objet types (likely still limited to PODs).

- Get more involved in the ROOT7 interface discussions.

- Farther out - reworking merge algorithms based on Linux block layer feedback.
  - DavidA has a working solution for monitoring block-layer activity: we now need to measure effect of various merge layouts.