

# DIANA IO Update

Brian Bockelman, Zhe Zhang,  
for the DIANA team

# DIANA - A Reminder

- DIANA/HEP (NSF-ACI #1450323) aims to improve the analysis infrastructure in HEP.
  - This can be accomplished along many routes - improved collaboration software, statistical tools, interoperability with other data-intensive ecosystems.
  - But one very obvious way to contribute is making ROOT IO faster!
- Here, we report on the current activities.
  - Still have an open position: <http://diana-hep.org/pages/jobs.html>. Willing to consider locating at UNL or FNAL.

# A note on collaborating

- (Ignoring the code itself,) It's tough to get students / collaborators started on ROOT:
  - There's a known set of coding conventions (<https://root.cern.ch/coding-conventions>) that are not automatically checked when pull requests are sent. **Waste of reviewer time to have to worry about these** (esp. if the reviewer is many timezones away).
  - While roottests is useful, there is no CI integration with pull requests. **Waste of reviewer time to double-check integration test results** (assuming a sufficiently simple patch...).
    - Better to note integration tests fail when it is a PR than after it has been merged!
- Some simple automation should save everyone's time and ease the on-ramp of new contributions!
  - When we ran into similar problems with contributing to CVMFS, they ran a subset of their builds, unit tests, and code convention checks in Travis-CI. Made contributing much easier: can **focus on the code review itself**.

# Current Activities

- Improved compression:
  - Testing LZ4: actually, two groups are doing this. In practice, not yet a huge improvement over ZLIB-1 (possibly due to smaller buffer sizes in ROOT?).
  - “Random Access Compression” (RAC): Allows access to a single event from a buffer. Works well for sparse reads with many events per buffer.
  - “External Compression”: Simple comparisons of performance when compression is done in the TFile layer, unaware of TTree-level knowledge.
- All three items are going to be written up in a CHEP paper.
  - Probably RAC will be worth merging: others are more about documenting the phase space.

# Current Activities

- Porting of CMS's "lazy-download" (called "buffered read").
  - Rounds reads up to 128MB chunks that are then buffered on local temporary disk. Very useful for high-latency, non-repetitive (cache-unfriendly) analysis.
  - Pull request posted; tests pass; new test added.
- Porting of CMS's improvements to TTreeCache for handling of cache misses.
  - Stalled out at about 50-75% complete.
- Migration of serialization code from big endian to little endian.
  - Pull request posted; tests pass.
  - Some work left to validate schema evolution; added new member to TKey.

# Where we are going...

- The end-goal of this work is to do zero-copy IO for sufficiently simple objects (Plain Ol' C structs or similar).
  - I.e., switch of endian-ness likely has no performance effect unless we can reduce memory copies.
  - Intermediate work is to remove unnecessary memory copy of TTreeCache for decompression.
- Aim is to give user a cluster-at-a-time instead of event-at-a-time per top-level branch.
  - When combined with a lambda/functor interface, I hope we'd even get the compiler to do vectorization of IO.

# Working with Community

- We'd like to update the "standard set of test files" from each experiment.
  - Seems most of the tested files correspond to LHC Run I: would like to see updates to Run II.
    - And gather some samples from analysis / ntuples, not just framework files.
  - Particularly, CMS will require extra effort to either port-or-remove custom serialization code.
  - Would also like to get some sample files from neutrino community.
- Goal is to be able to track IO performance changes across versions.