DIANA: Nebraska Activities

Brian Bockelman (PI)
David Abdurachmanov (Software Engineer)
Zhe Zhang (PhD Student)

Nebraska Team

Brian Bockelman

David Abdurachmanov

Zhe Zhang

Approach

- Vision: *Improve HEP analysis through improving ROOT IO (RIO)*.
- Goal: Over the lifetime of the DIANA project, deliver a **10x** speedup for common analysis activities.
- Methodology:
  - **“Fast IO” API**. Provide new, extremely fast APIs that target common analysis activities.
  - **File format improvements**. Improve compression ratio, decompression speed, and file layout.
  - **Code Engineering improvements**. Streamline and improve existing code quality. Marginal improvements to existing algorithms

Fast IO

- Recent activities:
  - **Bulk IO API**: Deserialize many events at once, avoiding overheads.
  - **Implicit Multithreading** (IMT): Contribute to wider ROOT effort to allow RIO libraries utilize multiple threads for single-threaded interfaces.
  - Processing models research: Have users express code using high-level semantics. Looking at functional-like interfaces, similar to many "big data" projects.
    - Better processing models allow us to utilize parallelism, decreasing processing time. Example in next slide.
- Next 6 months:
  - Finish off items in the "fast" RIO plan: more data types, high-level interface, .
  - Guide all this into the appropriate releases.

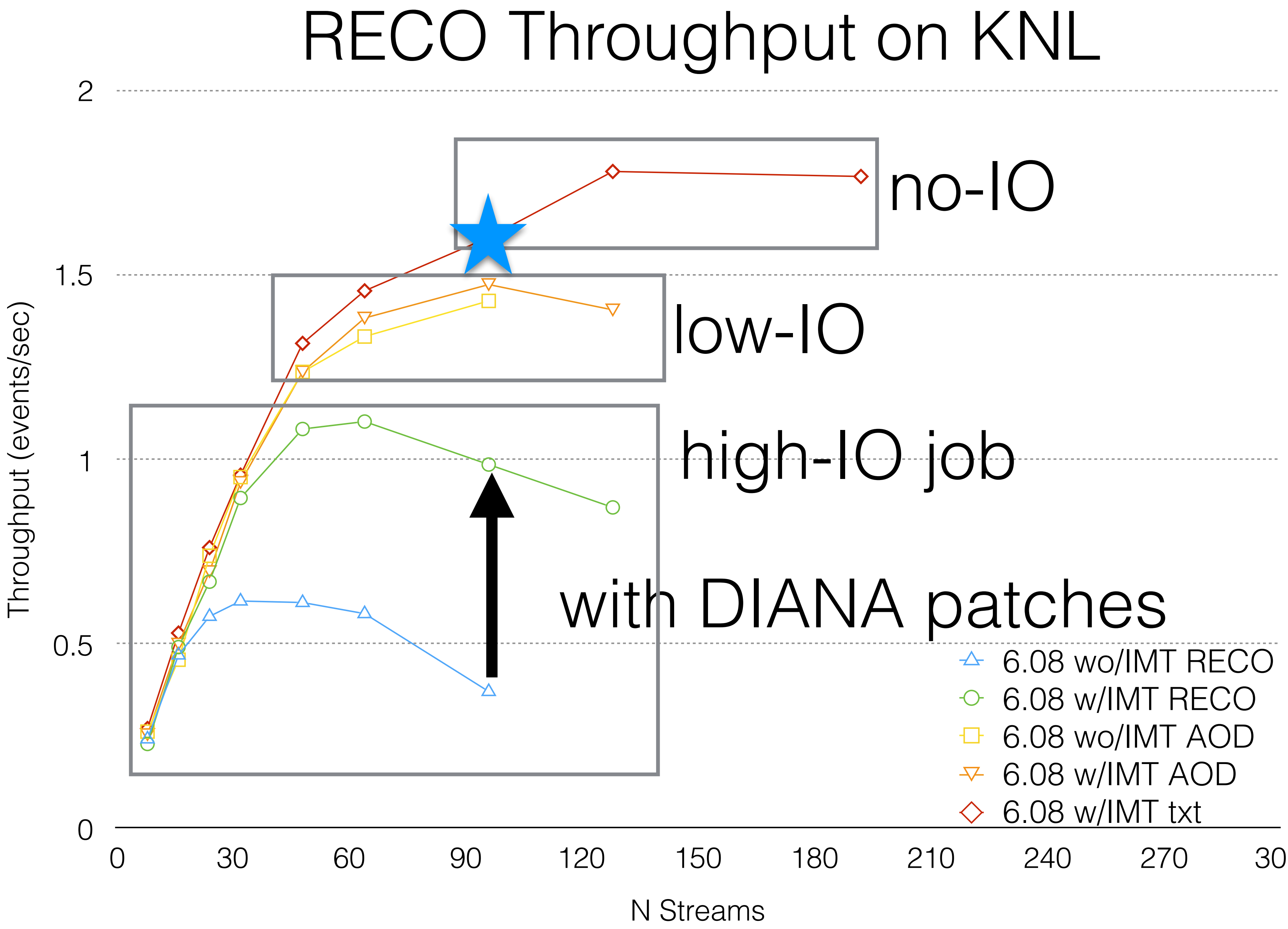
You say potato, I say…

- Both code samples below may appear to do the same thing:
  - The loop forces a specific ordering and level of parallelism. Not much room for improvement.
  - The library call does not! **We could parallelize this and make it faster**.
- Similar idea in ROOT: encourage users to write at a higher-level.

```
long64_t my_sum = 0;
for (auto entry : my_array) {
   my_sum += entry;
}
```

```
long64_t my_sum = sum(my_array);
```

Example IMT Improvements (CMS)



File Format Improvements

- Recent activity:
  - Broad investigation into alternate compression algorithms (LZ4) and methods (improving access to random events).
  - Prototype little-endian file format for ROOT (potential speed improvement).
- Next sixth months:
  - Study interaction between ROOT files and kernel / filesystem / hardware.
  - Change endianness in production code (stretch goal).

Engineering Improvements

- Recent activity:
  - `zlib` speedups: compression often saves 4x disk space, but has significant CPU costs.
    - We are finalizing a set of patches that provides 4x decrease in compression times and 20% decrease in decompression.
    - Intel QAT hardware (DavidA). Performs compression in-silicon for a potentially-large speedup; hit unexpected technical difficulties.
  - Reduce impact of latency: reshaping and generalizing some CMS-specific patches that reduce sensitivity to latency.
- Next 6 months:
  - Upstreaming zlib changes to ROOT and zlib itself.

Benchmarking Work

- To measure progress, we have a deliverable (not started) to create a 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.
    - Allows fast copy for integral types and C-style arrays of types.

Impact

- Impact has mostly been limited to the local community:
  - Two CHEP papers (Abdurachmanov & Zhang) by DIANA team.
  - `zlib` improvements put into CMSSW test builds.
  - DIANA forum presentation on ROOT TTreeProcessor. Trying to engage more the CERN summer student and staff on this approach.
  - CMS tests of the IMT branches.
- Expanding impact:
  - Working to get various test / prototypes into ROOT 6.10 and 6.12.
  - Re-establishing the weekly ROOT IO meetings and periodic workshops.

Collaborations

- Internal:
  - The high level work on Histogrammar and Femtocode mirrors the ROOT-based improvements in data frames. The .
- External:
  - DavidA is a significant presence inside CERN OpenLab, meaning we are able to test and benchmark on a wide variety of hardware. (*Note: majority of this work is under NDA, making it difficult to discuss in public forums.*).
  - Looking for other areas to make DavidA & DIANA contributions visible in 2017 (eg., Intel KNL community or QAT work above).

Other plans for >6 months

- Expand the “fast IO” project to more complex data structures; eventually switching to ‘zero copy IO’.
- Based on results of the fast IO project, start tackling more complex object types (likely still limited to PODs).
- More active push 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.

Questions?