Optimizing Files for Analysis

Brian Bockelman
22 June 2017

Idea #1:
One Basket Per Cluster

- With a simple tweak, we can force ROOT to continue growing basket sizes so there is one-per-cluster.
- Why?
    - We already suggest all TTree users logically think in terms of event clusters.
    - Would make the bulk IO APIs faster — unaligned basket boundaries cost speed!
    - Merging by event clusters becomes trivial.
    - Smaller chance of disk IO between cluster boundaries: more predictable performance.
    - Makes OptimizeBaskets trivial.
- Why not?
    - Potentially significant memory costs for “surprisingly large events.” Bad for highly variable event sizes (i.e., when average is 1MB/evt but max is 1GB/evt).

Idea #2:
More aggressively resize baskets

- We currently shrink baskets when they are 2x larger than the historical average.

- Idea: whenever we double a basket size while filling, resize it to N% (N=120?) of occupied size when a flush is performed.

  - Example: suppose we have a 1MB basket and we have to put in a 1.1MB object. We would immediately resize the basket to 2MB. The basket size would stay at 2MB.

  - With this change, at the event cluster boundary, we would shrink the memory usage to 1.1*1.2 = 1.32MB.

- **Goal**: tighten lower bound on the difference between “minimum memory size” and “used memory size”.

CMS Test

- Does this make a difference?
    - Started with a CMS real-data AOD file from 2017, from the SingleMuon primary dataset.
- Writing with CMSSW:
    - Default ROOT settings (20MB AutoFlush) -> peak RSS 837MB
    - 50MB AutoFlush -> peak RSS 1020MB.
    - 50MB AutoFlush, one basket per cluster -> peak RSS 1403MB.
    - 50MB AutoFlush with shrinking -> 1020MB. (no change!)
    - 50MB AutoFlush, one basket per cluster, with shrinking -> 1088MB.
- Reading with CMSSW:
    - 50MB AutoFlush -> 738MB
    - 50MB AutoFlush, one basket per cluster -> 827MB

CMS AOD Test - Compression

- Going into a detour about compression levels. Default for this file is LZMA-4 (resulting in a 2,816MB file) with 15MB auto-flush (CMS default).
    - CMSSW peaks at 840MB RAM to re-compress this file.
- With a 10MB auto-flush
    - None: 12,059MB file, 7.5 minutes
    - ZLIB-7: 3,150MB file, 11 minutes (compression @ 15MB/s), 792MB RSS, 0.021s/evt
    - ZLIB-9: 3,129MB file, 27 minutes (2.7MB/s), 797MB RSS, 0.11s/evt
    - LZMA-4: 2,925MB, 33 min (1.9MB/s), 814MB RSS, 0.15s/evt
    - LZMA-9: 2,891MB, 51 min (1.1MB/s), 821MB RSS, 0.26s/evt

CMS AOD Test - Compression

- Repeating with 20MB auto flush settings (timings are about the same):
  - ZLIB-7: 2,996MB, 859MB RSS
  - ZLIB-9: 2,974MB, 862MB RSS
  - LZMA-4: 2,739MB, 869MB RSS. -2.7% from baseline file size, +29MB RSS
  - LZMA-9: 2,701MB, 863MB RSS. -4.1% from baseline, +23MB RSS

CMS AOD Test - Compression

- Repeating with 30MB auto flush settings (timings are about the same):
    - ZLIB-7: 2,941MB, 888MB RSS,
    - ZLIB-9: 2,918MB, 890MB RSS
    - LZMA-4: 2,666MB, 931MB RSS.
        - -5.3% from baseline, +91MB RSS
    - LZMA-9: 2,626MB, 934MB RSS.
        - -6.5% from baseline, +94MB RSS

CMS AOD Conclusions

- Basket clustering:
    - Aggressive shrinking made the most difference when combined with one basket per cluster.
    - One basket per cluster - with shrinking - cost about 60MB at write time.
    - One basket per cluster cost about 89MB at read time.
        - Modest decrease in number of baskets (10%).
        - Conclusion: forcing one-basket-per-cluster has little advantage for CMS EDM.
- Next week - revisit the idea for ntuples.
- Compression updates:
    - For about 60MB of RSS at write time, one can decrease file sizes by about 5%.
    - Probably not going to increase LZMA settings until we have IMT enabled for writes.