




What's In Store For ROOT I/O

Philippe Canal

October 19, 2011

What's new in root/io

- TClonesArray extension
- gDirectory / gFile are now thread local 
 - Completely removing the dependency requires backward incompatible changes in TObject ...
- TMemFile
- TParallelMergingFile and parallelMergeServer
- CPU Performance enhancements
- Contributed
 - Write Sequence implementation (Chris Jones)
 - TEntryListArray (Bruno Lenzi)

TClonesArray

- New function TClonesArray::ConstructedAt
 - Always returns an already constructed object.
 - Construct the object only once, return it 'unchanged' the other times.

```
new(a[i]) TTrack(x,y,z,...);  
...  
a.Delete(); // or a.Clear("C")
```

- with the simpler and more efficient:

```
TTrack *track = (TTrack*)a.ConstructedAt(i);  
...  
a.Clear();
```

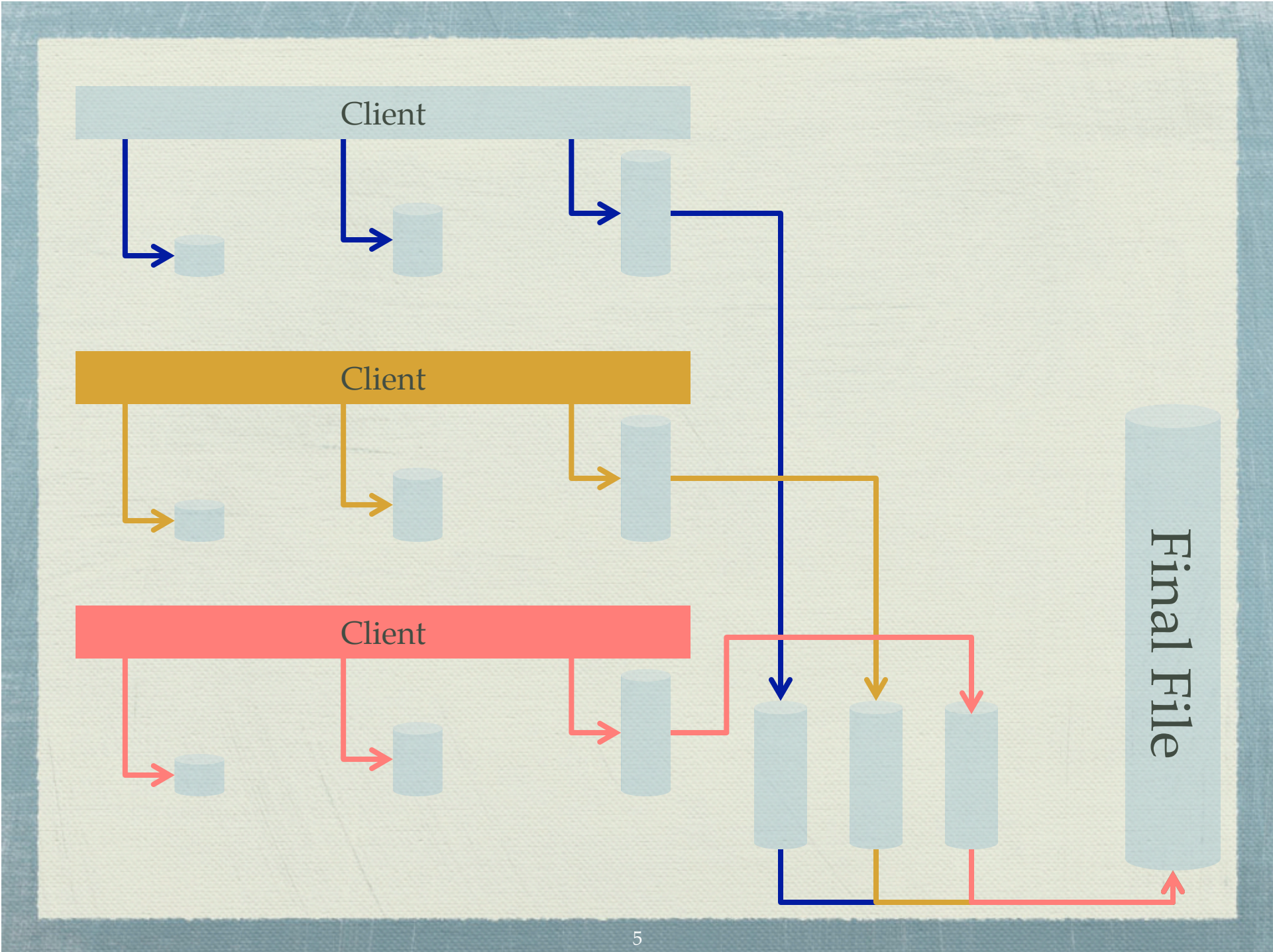
- even in case where the TTrack class allocates memory.

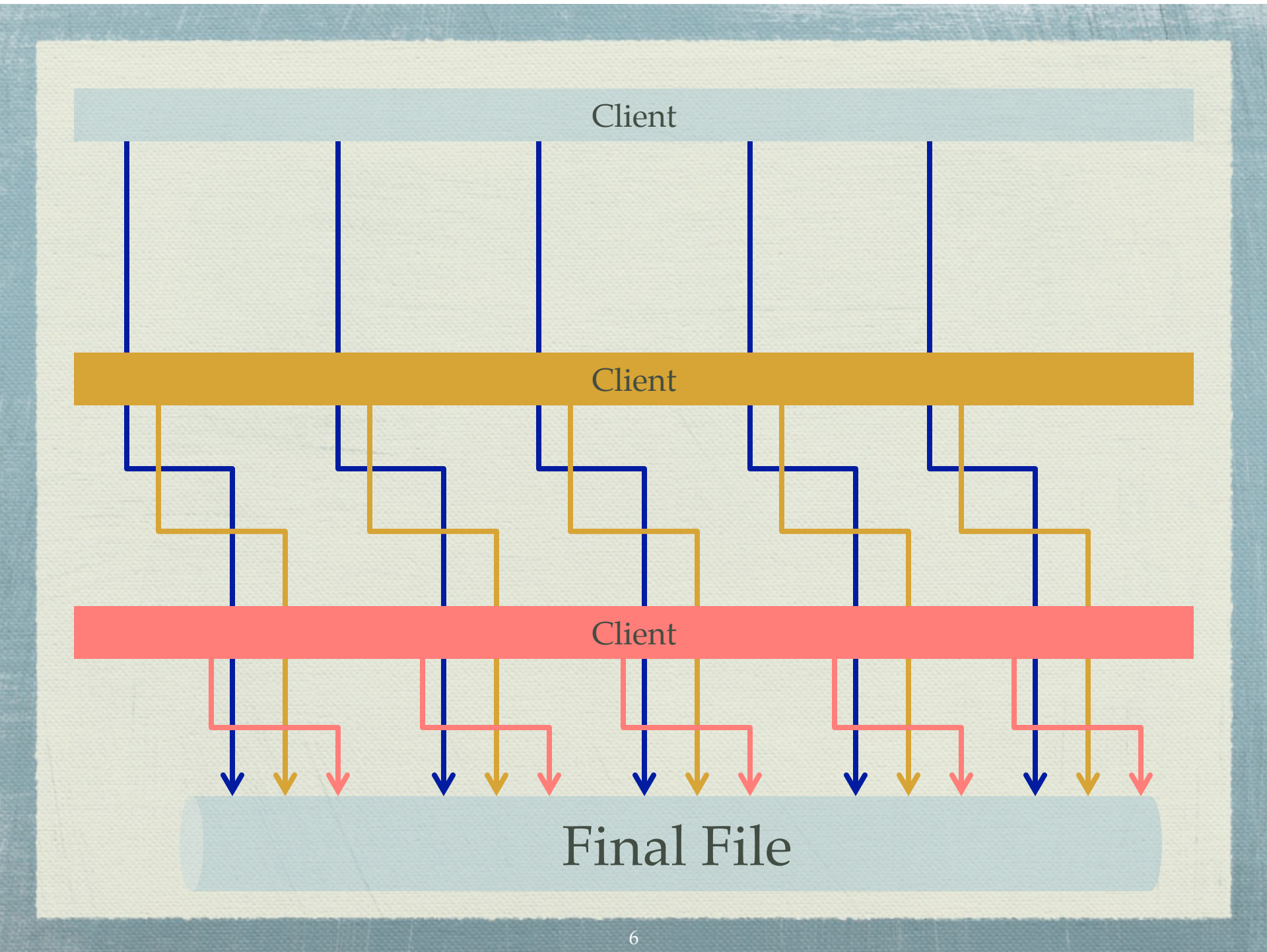
Parallel Merge

- New class TMemFile
 - A completely in memory version of TFile.
- New class TParallelMergingFile
 - A TMemFile that on a call to Write will
 - Upload its current content to a parallelMergerServer
 - Reset the TTree objects to facilitate the new merge.

```
TFile::Open("mergedClient.root?pmerge=localhost:1095","RECREATE");
```

- New daemon parallelMergeServer
 - Receive input from local or remote client and merger into request file (which can be local or remote).
 - Fast merge TTree. Re-merge all histogram at regular interval.





Contributions

- TEntryListArray (Bruno Lenzi)
 - TEntryListArray is an extension of TEntryList, used to hold selected entries and subentries (sublists) of a TTree. It is useful in case of a TTree with containers (vectors, arrays, ...). A typical example is the case when each entry of a TTree corresponds to physics events and each subentry (index of arrays) represents a particle.
- Write Sequence implementation (Chris Jones)
 - Implementation of the infrastructure to speed up the write part of the Streaming as it has already been done in the read part.
 - Next is to implement the individual Write Streaming actions.
 - Necessary step toward the implementation of the I/O customization write rules

Ideas bubbling up

- Disclaimers: those ideas might or might not come to fruition 😊
- Write only once files (Hadoop)
 - At the possible expense of file size, write the directory information at the end rather than the beginning.
 - Lose the ability to detect truncated files ; not forward compatible.
- In TBasket compress each entry individually (for *large* basket)
 - Also *copy* the compression dictionary from one basket to the next.
 - Allow for fast sparse reads. Copy allows for better compression.
- Find a way to avoid storing the byte count and version number for deep hierarchy
 - Idea is to record a 'flavor' once per buffer / (io operation) per top level class and to associate a fully unrolled sequence of actions to this flavor

Meanwhile in another ...

- Cling released in July 2011
 - Fully functional C and C++ interpreter (including C++11)
 - Uses Just-In-Time compilation
 - Still a few issues to solve (e.g. reloading of code)
 - Distributed with v5.32
- Browsers and HTML5
 - Provide ROOT file access (ROOT-IO.js) and graphics directly in the browser

Meanwhile in another ...

- Tablets and Smart Phones
 - Currently 200 Million iOS devices, 25 Millions iPads
 - All of ROOT (except device graphics) ported to iOS (little endian, 32-bit, ARM)
 - Graphic being implemented now.
- Improvements in Graphics
 - Move to native graphics on Mac OS X
 - New more powerful TeX engine
- Continued development in Stats, Math, Proof

Release Schedule

- ROOT Release v5-32-00
 - Version v5-32-rc1 will be released Nov 1, 2011
 - Version v5-32-rc2 will be released Nov 15, 2011
 - Version v5-32-00 will be released Nov 29, 2011
- ROOT Release v5-34 (Beta of cling based dictionary)
 - May 2012
- ROOT Release v6 (Cling based dictionary)
 - November 2012

Current Priorities (v5.34)

- Bug Fixes / Support
- *I/O Customization: Nested Objects (several weeks)*
- Explore changing the on-file byte format to little endian (days)
- Explore other small change in file format to reduce size (days)
- Update fast-merging to leverage the TTreeCache (days)
- Upgrade SetAddress and SetBranchAddress (days - focused)
- Continue optimization of TBranch::GetEntry (days)

v5.34 Contributions

- Allow more than one TTreeCache per file (automatically) - Peter
- Resolve the issue of the TTreeCache startup time – Peter
- Reimplementation of OptimizeBaskets – Brian
- Fast Merging sort by cluster and branches - Brian
- Testing of parallel prefetching – Brian
- I/O Customization: Write Rules – Chris
- Test environment - Ilija/Wahid