

CMS and ROOT I/O



Christopher Jones *FNAL*

Thread-safety

CMS is transitioning to a threaded-framework

Have temporary solutions to support ROOT I/O

Need better threading support in ROOT

TObject properly determines if on stack or heap

Open different TFiles on different threads safely
don't require preloading all known streamers

Resolve deadlock problems between ROOT's lock and dlopen lock

Want improvements for future

Efficient use of thread local storage

present ROOT implementation uses a mutexed linked list

Approved method for using multiple threads to read from one TFile

Approved method for using multiple threads to write to one TFile

Idea for Threaded Reads



Something to try next year

TTreeCache manages access to the file

keeps memory footprint smaller than multiple TFiles

TTree per simultaneous event read from TTreeCache

have to synchronize access to TTree within event

Synchronize when TTreeCache needs to read more

For future

Parallel unpacking different TBranches for the same event