



Concurrency in ROOT

Some random thoughts

Benedikt Hegner

ROOT Retreat, 8.7.2014

Task Based Model

- **The task based model seems the most promising one for big software**
 - Essentially all HEP frameworks ended up developing for this model
- **Seems to me to be the best choice for ROOT as well**
 - Unless really needed, don't handle threads directly!

ROOT and other projects

- **ROOT parallelisation needs to compose with other HEP software**
 - From Geant4 we can learn what not to do!
 - It has to be thought about from ground up
- **Global resource management is a must**
 - Different resource pools for different sub-parts of the application does not work
 - Imagine a FWK, a ROOT, and a Geant4 pool competing
- **Options for such a management:**
 - Create tasks and pass them to the experiment FWK for execution
 - Use a resource pool given to ROOT explicitly

Understanding side-effects

- **Which parts of the API are interconnected?**
 - Side effects of a call to ROOT are often surprising to a naive user
 - If calling TA:foo and TB:bar - do they share resources and do they block each other?
- **Frameworks have a reserve-before-schedule model to resolve task-conflicts upfront**
 - e.g. never run two tasks at the same time using a thread-unsafe library
 - Any chance of resource requirement announcement in ROOT tasks?
- **Two problems to tackle:**
 - Document interconnections for users
 - Understand and minimize interconnections