

The BaBar Software Legacy in SuperB

David Brown, LBNL

SuperB Computing Workshop
16 December 2008

BaBar Software Reuse

- Advantages of re-using BaBar code in SuperB
 - Many man-years of labor saved
 - Continuity for user community
 - Battle-tested!
- Disadvantages and risks
 - Maintainability
 - work-hardened, poorly-documented code
 - Modern coding standards not enforced
 - Performance
 - Designs reflect obsolete technology (RW, F77, ...)
 - Algorithms not compatible with multi-core

Improving BaBar Software

- Migrate to newer technology
 - Framework, eventstore, release tools, ...
 - upgrade or replace external packages
- Simplify designs to reflect actual usage
- Improve critical algorithms
- Define and enforce standards
 - 0-leak tolerance
- Reverse-engineer and rewrite some blocks
- Review all software

Specifics

- Rewrite lfr reco, Dirc reco, Track pattern recognition, MC truth
- Modernize Kalman Fit, Emc reco, Beta
 - multi-core parallel processing
- Replace Matrix, algebra, geometry, framework, build system, release system
 - LHC/root/OpenSource?
- Persistence, event, bookkeeping, conditions?

Development Strategy

- Always maintain a working copy of the code
 - TDR work, physics studies, ...
- Make continuous adiabatic improvements
 - computing professionals? computing students?
- Factorize the projects
 - Allow small groups/individuals to contribute
 - Prioritize according to impact
- Central coordination

When to start? Now!

- Many projects are short-term, limited scope
 - good for summer students
 - can have immediate impact (simulation, TDR studies)
- Larger projects need long lead-time (Pete)
- Establish an active computing community integrated with the rest of SuperB activities

Conclusions

- Reuse of BaBar software is essential for the success of SuperB
 - similar scale investment as hardware
- BaBar software needs Repair and Upgrade for SuperB
 - factor of ~100 more data!
 - Moore's law will not give us all of it
- Computing Planning Group is preparing for the work