SuperB and its computing requirements

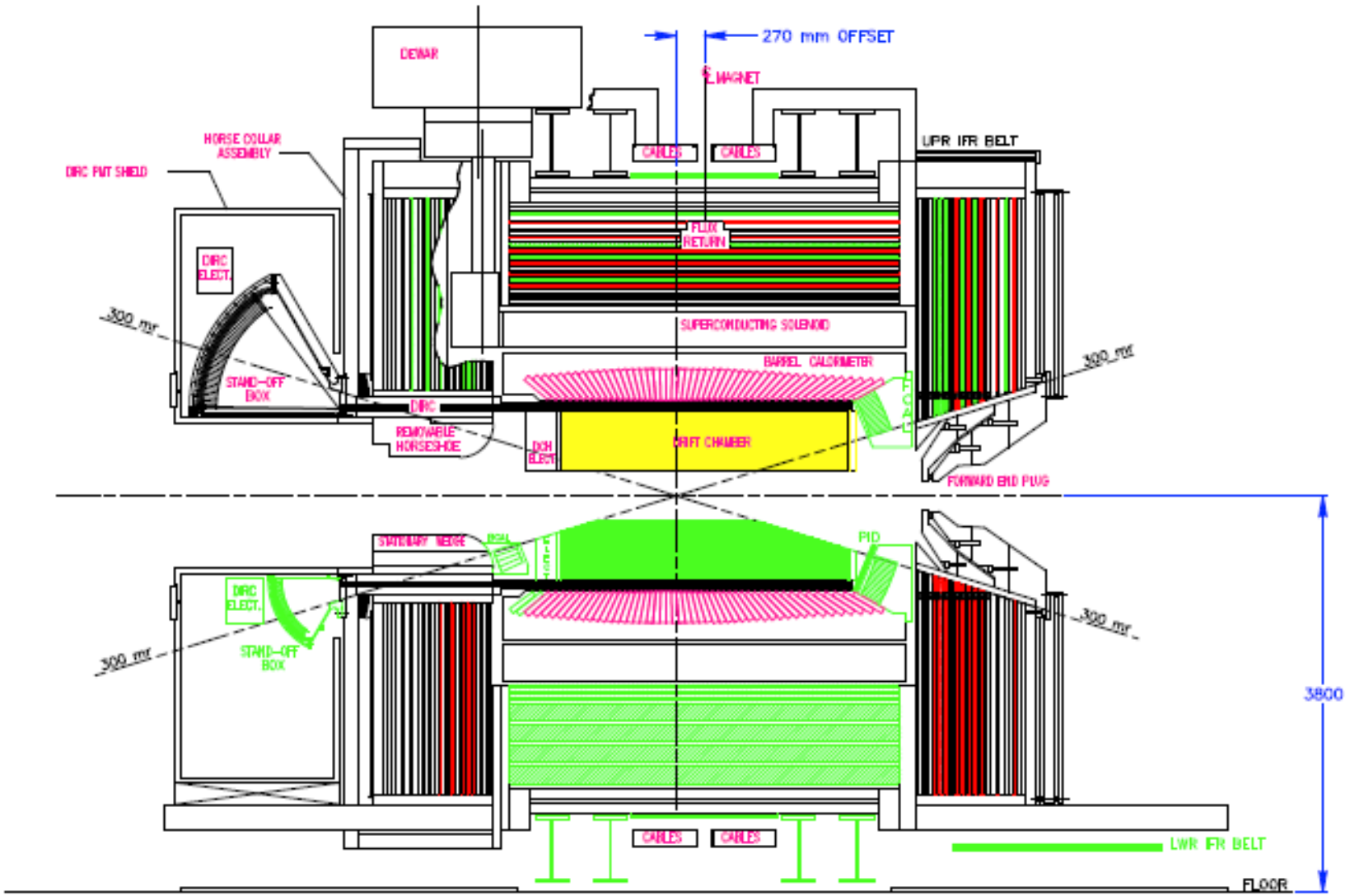
Fabrizio Bianchi
University of Torino and INFN-Torino

SuperB Computing R&D Workshop
Ferrara, March 9-12, 2010

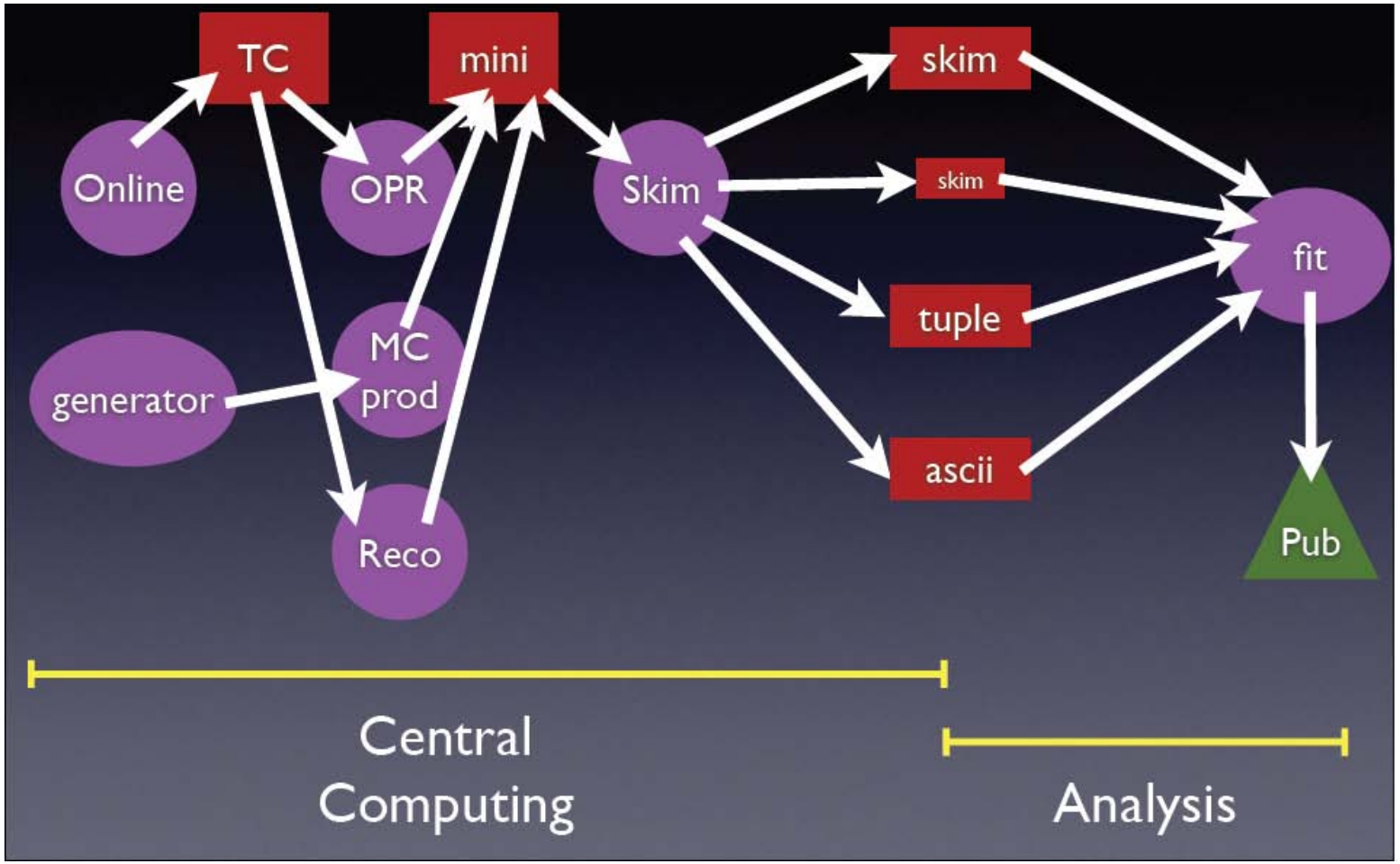
Outline

- The detector.
- The size of the problem.
- Hardware requirements.
- Distributed computing.
- Code requirements.

The Detector



Possible outline of SuperB processing (from D. Brown presentation)



The size of the problem

- Design Luminosity: 10^{36} cm^{-2} s^{-1}
- Integrated luminosity: up to 12 ab^{-1}/year

- HLT accepted cross section: 25 nb
- Accepted event rate: 25 KHz
- Raw Data event size: 75 KB
- Raw Data logging rate: ~ 2 GB/s

- Raw Data size: 875 TB/ab^{-1}
- Micro Data size: 42 TB/ab^{-1}
- Mini Data size: 80 TB/ab^{-1}
- Micro MC size: 100 TB/ab^{-1}
- Mini MC size: 150 TB/ab^{-1}
- Skim expansion factor: 2 (was 3-4 in BaBar, indexing vs deep copies)
- User analysis data 200 TB/ab^{-1}

Hardware Requirements (scaled from BaBar)

- Storage of raw data, mini, micro, skims, user analysis data (ntuples, fit results).
  - Enough storage to keep multiple processing of the data sample
- Reconstruction of raw data: in parallel with data taking, scales with peak luminosity.
  - Essential to fully monitor the quality of data
  - CPU estimate: 22 MSpecInt2000 / 10^{36} cm^{-2} s^{-1}
- Simulation and skimming of new data: in parallel with data taking, scales with peak luminosity.
  - CPU estimate: 160 MSpecInt2000 / 10^{36} cm^{-2} s^{-1}
- Capability of re-process, re-simulate and re-skim every year the data collected in previous years: scales with integrated luminosity
- Analysis capability: scales with integrated luminosity
  - CPU estimate: 160 MSpecInt2000 / ab^{-1}
  - Factor 2 reduction wrt BaBar
    - Optimization of analysis code

Data storage requirements

- Data volume will be huge: hundreds of PB.

- Storage resources will be geographically distributed.
- Users should be able to ignore the geographic location of data.

  Requirements on:
  - Storage systems.
  - Databases.
  - Job submission system.
  - Code design.

- Most SuperB applications will be I/O intensive.
- Avoid I/O bottlenecks. Requirements on:
  - Storage system performances.
  - Code design.

Distributed computing

- A must for SuperB: CPU and storage resources will be in different locations.
- Ideally you want something that:
  - Optimize usage of resources (storage/CPU/bandwidth).
  - Is transparent to users. They only have to care about:
    - Submitting jobs.
    - Define input data.
    - Specify were they want the output to show up.
- Is GRID middleware good enough for SuperB ?
  - Use cloud instead ? Or something else ?
- Impact of distributed computing on:
  - Databases technology choice.
  - Database applications design.

Impact of new CPU architectures

- Recently the single CPU performances have not increased significantly.
- Computing has been characterized by the advent of "multicore CPUs".
  - GPUs have hundreds of cores.
- Efficient exploitation of multi/many cores is a must for SuperB computing
- Programming paradigm must shift to parallel
  - Algorithms must be reformulated
  - Coding must follows new guidelines
  - Need to build the appropriate expertise in code developers

Code requirements

- SuperB will have a large community of code developers and users.
- Their efforts must be efficiently integrated.
  - Collaborative tools, releases building tools.
  - Training of developers.
- Quality of the code:
  - Must be monitored.
  - Quality standards must be defined and enforced.
- New coding paradigm and tools are needed to efficiently exploits new CPU architectures.

Code Quality Requirements: BaBar experience

- BaBar was the first HEP experiment to use c++ and OOAD.
  - Very limited expertise in HEP community.
  - Many tools not yet existing (STL…).
- Quality requirements were not stringent.
  - Compile on 2 compilers.
  - Run.
  - Limited memory leaks were accepted.
- Limited effort in offline code profiling and optimization.
  - Mostly in reconstruction and simulation code.
  - Recent optimization efforts on analysis code resulted in CPU time gain of a factor between 1.5 and 10.

Code & Data quality requirements for SuperB

- Checks on code quality:
  - Compile and run on different (>=2 ?) platforms.
  - No memory leaks.
  - Aggressive profiling and optimization of the code.
  - Code validation between platforms and releases.
  - Frequent (nightly ?) build of the code.

- Checks on data quality:
  - First check during data taking.
  - Additional checks on reconstructed data, not much after data taking, to provide feedback on detector performances.
    - Need automated tools.
    - Limit the human visual inspection of histograms.
  - MC data and skimmed data need to be checked too.