Next Generation of Event Data Processing Frameworks

Preliminary Ideas for a New Project Proposal

September 2011

Motivation

- For the last 40 years HEP event processing frameworks have had the same structure
  - initialize; loop events {loop modules {…} }; finalize
  - O-O has not added anything substantial
  - It is simple, intuitive, easy to manage, scalable
- Current frameworks designed late 1990's
  - We know now better what is really needed
  - Unnecessary complexity impacts on performance
- Inadequate for the many-core era
  - Multi-process, multi-threads, GPU's, vectorization, etc.
  - The one job-per-core scales well but requires too much memory and sequential file merging

Vision

- Same framework for simulation, reconstruction, analysis, high level trigger applications
- Common framework for use by any experiment
- Decomposition of the processing of each event into 'chunks' that can **executed concurrently**
- Ability to process several [many] events concurrently
- Optimized scheduling and associated data structures
- Minimize any processing requiring exclusive access to resources because it breaks concurrency
- Supporting various hardware/software technologies
- Facilitate the integration of existing LHC applications code (algorithmic part)
- Quick delivery of running prototypes. The opportunity of the 18 months LHC shutdown

Universal Framework

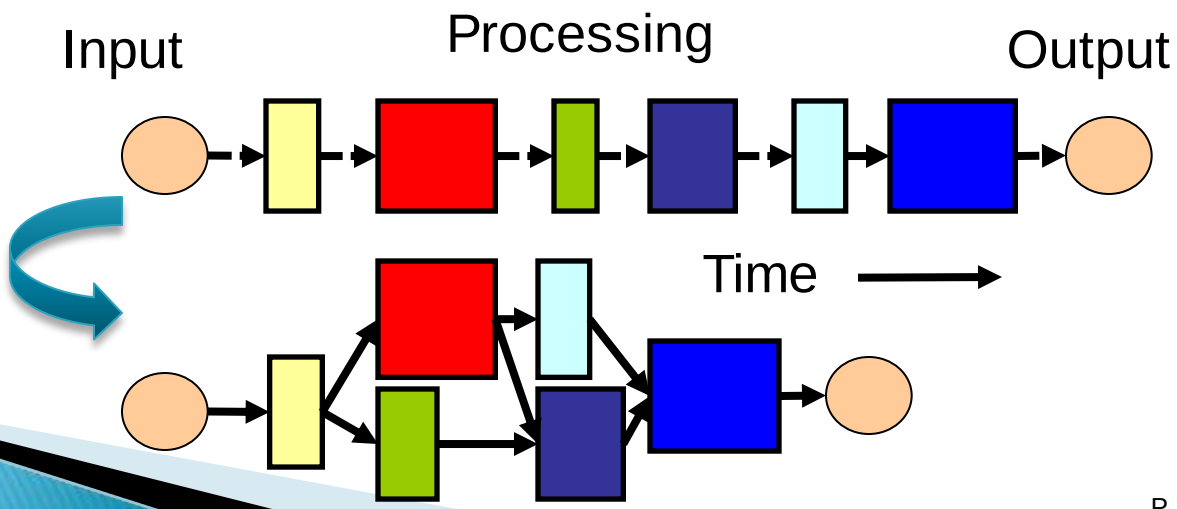
- Current frameworks used by LHC experiments supports all data processing applications
  - High-level trigger, reconstruction, analysis, etc.
  - Nothing really new here
- But, simulation applications are designed with a big 'chunk' in which all Geant4 processing is happening
  - We to improve the full and fast simulation using the set common services and infrastructure
  - See later the implications for Geant4
- Running on the major platforms
  - Linux, MacOSX, Windows

Common Framework

- Frameworks can be shared between experiments
  - E.g. Gaudi used by LHCb, ATLAS, HARP, MINERVA, GLAST, BES3, etc.
- We can do better this time :-)
  - Expect to work closely with LHC experiments
  - Aim to support ATLAS and CMS at least
- Special emphasis to requirements from:
  - New experiments
    - E.g. Linear Collider, SuperB, etc.
  - Different processing paradigms
    - E.g. fix target experiments, astroparticles

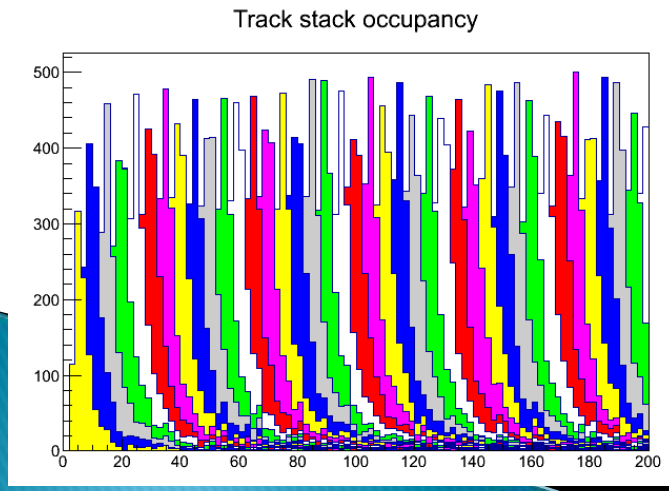
Concurrent 'chunk' processing

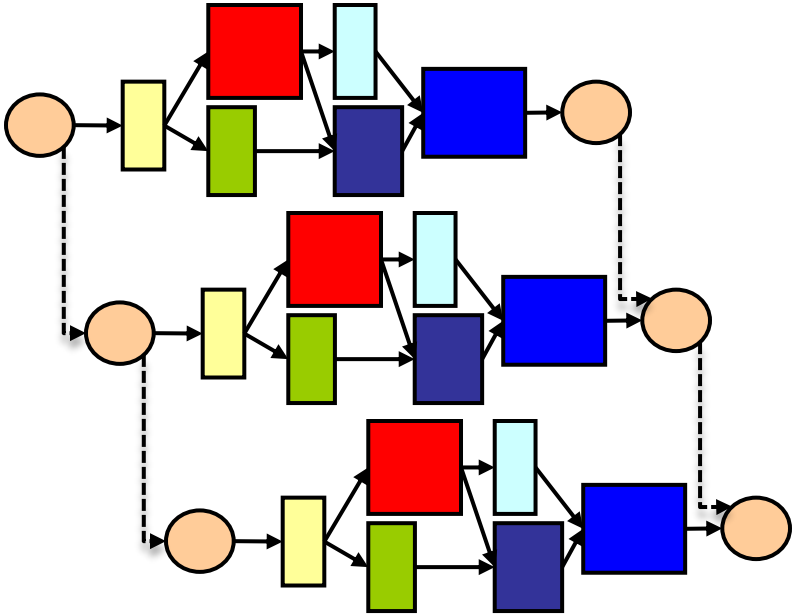
- Framework with the ability to schedule concurrent tasks
  - Full data dependency analysis would be required (no global data or hidden dependencies)
- Not much gain expected with today's designed 'chunks'
  - See CMS estimates at CHEP'10 (*)
  - Algorithm decomposition can be influenced by the framework capabilities
- 'Chunks' could be processed by different hardware/software
  - CPU, GPU, threads, process, etc.



Many Concurrent Events

- Need to deal with tails of sequential processing
  - See Rene's presentation (*)
- Introducing Pipeline processing
  - Never tried before!
  - Exclusive access to resources can be pipelined e.g. file writing
- Need to design a very powerful scheduler





Why the Framework managing the concurrency?

- Concrete algorithms can be parallelized with some effort
  - Making use of Threads, OpenMP, MPI, GPUs, etc.
  - But difficult to integrate them in a complete application
    - E.g. MT-G4 with Parallel Gaudi
  - Performance-wise only makes sense to parallelize the complete application and not only parts
- Developing and validating parallel code is difficult
  - 'Physicists' should be saved from this
  - In any case, concurrency will limit what can be done and not in algorithmic code
- At the Framework level you have the overall view and control of the application

Project

- Collaboration of CERN with FNAL, DESY and possible other Labs
  - Start with small number of people (at the beginning)
  - Open to people willing to collaborate
  - Strong interactions with ATLAS and CMS (and others)
    - E.g. Instrumentation of existing applications to provide requirements
  - Strong collaboration with Geant4 team
- Quick delivery of running prototypes (I and II)
  - First prototype in 12 months :-)
- Agile project management with 'short' cycles
  - Weekly meetings to review progress and update plans

R&D Activities

- We need to evaluate some of the existing and technologies and design partial prototypes of critical parts
  - See next slide
- The idea would be to organize these R&D activities in short cycles
  - Coordinating the interested people to cover all aspects
  - Coming with conclusions (yes/no) within few months

Proposed R&D activities

- Investigate current LHC applications to gather requirements
  - Dependencies, data access patterns, opportunities for concurrency, etc.
- Investigate design and implementations of state -of-the-art concurrent frameworks
  - Scheduling (static, dynamic, adaptive), memory model, I/O
- Prototype framework elements
  - Identify 'exemplar' algorithms to be parallelized
  - Data structures and memory allocation strategies
  - New languages (C++11,…) and libraries (OpenCL,…)
- Understanding I/O

Project Timeline