Status of the FCC experiments' software

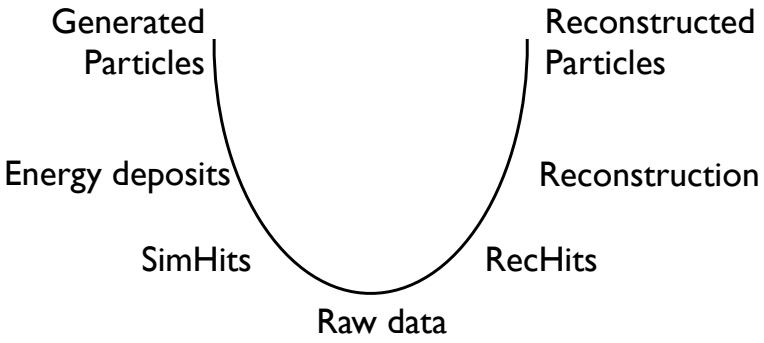
B. Hegner
PH-SFT
19.6.2014

Outline

- How to approach a new software project
- Development Environment
- Frameworks and Data Models
- Detector Description
- Simulation, Reconstruction, Analysis
- What next

Software Environment for FCC

- The FCC experiments are starting a SW effort from scratch, right?
  - Well, yes and no…
  - Problems have somehow been solved elsewhere already
  - One needs to pick and choose wisely
  - Only develop new things if it is worth it
- Fields to find solutions for
  - Development Environment
  - Core Framework
  - Simulation
  - Detector Description
  - Reconstruction
  - Data Model
  - Analysis



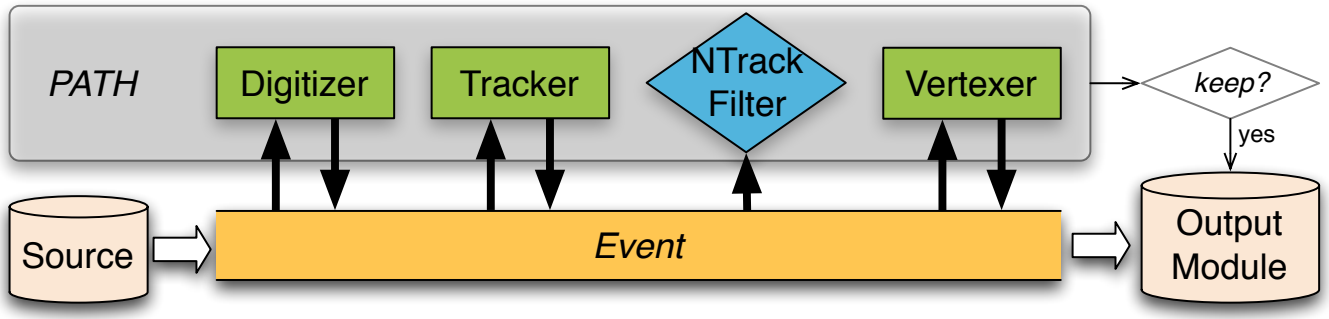
Very simplified view on data flow

Driving considerations

- Not many people and ambitious goals
  - Pragmatic start needed
  - FCC-* should share same software, wherever possible
- LHC software turned out to be quite complex and very specific
  - One has to start as simple as possible
- As time progresses move to more sophisticated solutions
  - Allow components to be replaced later on
  - Simulation as prime example
  - Flexibility
- Take advantage of effort of other people
  - Large choice of SW products to choose from in terms of generators, detector simulation, visualization, reconstruction, analysis…
  - Aim for, but don't blindly force, synergy with other projects

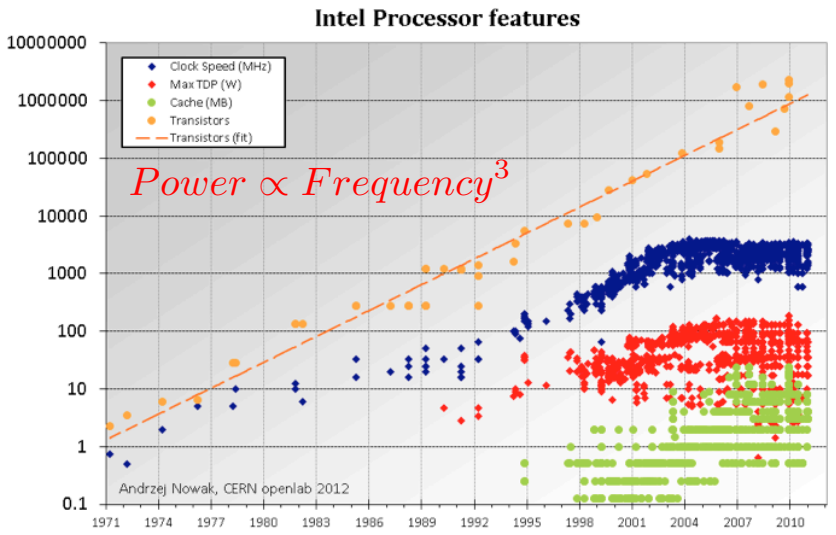
Why a Software Framework?

- Initially one has to be very pragmatic
- Start with simple buildings blocks and make them gradually more sophisticated
- However, one has to ensure their interoperability
- A good framework hides complexity
  - With slightly higher costs at the beginning than putting first pieces together directly…
  - … it allows gradual evolution of code



Recent Challenges to Software Frameworks

- The "power" wall changed how CPU technology evolved
  - No increase in clock frequency any more
  - Number of cores doubles every two years
  - Instruction parallelization increases



- To take full advantage of existing hardware one has to develop software properly
  - Think of parallel execution right from the start
  - Doing it afterwards is very hard (as can be seen in the LHC experiments)

- Gaudi (*) chosen as FCC framework
  - Production quality (used by multiple experiments already)
  - Designed for flexibility
  - Experts based at CERN
  - PH-SFT's GaudiHive project ensures its future-proofness
    - Preparing it for parallelization

(*) http://proj-gaudi.web.cern.ch/proj-gaudi/

Development Environment

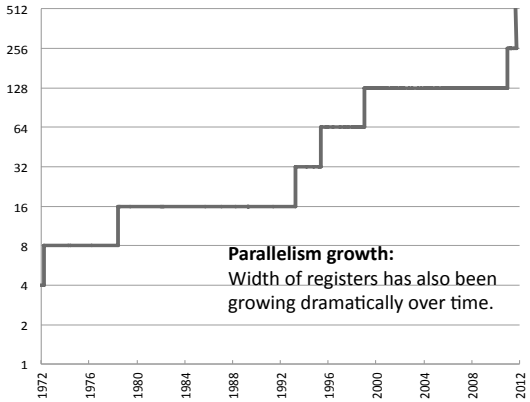
- Success of development is only guaranteed if tools and procedures are in place
- Communication
  - fcc-experiments-sw-dev@cern.ch
  - Every two weeks meeting with quick status reports
    https://indico.cern.ch/category/5666/
  - Please join, if you are interested!
- Repository
  - A common git containing everything done
    (https://git.cern.ch/web/?p=fcc-experiments-sw.git;a=summary)
  - Will eventually move to github
- Testing
  - Offer to re-use infrastructure of PH-SFT
- JIRA for bug/request tracking
  - https://sft.its.cern.ch/jira/browse/FCC
- Releases
  - Go for frequent snapshots (~ 2 weeks)
  - Space in /afs/cern.ch/exp/fcc
- Twiki for Documentation
  - https://twiki.cern.ch/twiki/bin/viewauth/FCC/WebHome

Will stay with agile and slim setup

Single Instruction – Multiple Data (SIMD)

- Processors supporting Single Instruction, Multiple Data (SIMD) can execute *one instruction* on *multiple data*
  - e.g. calibrate multiple jets in one step
- Successive SIMD instruction sets exist (MMX, SSE, SSE2, … , AVX) with ever increasing register size
  - AVX (since 2011) allows four double precision floating point values
- The technique to use this potential is called "vectorization"

Vector Parallelism: Register Width



Vector Instruction sets e.g.: SSE4.x, AVX, FMA

The Data Model

- The Data Model defines common data structures for tracks, jets, etc.
- It is one of the most central pieces of the SW
  - Every algorithmic code and every physicist is exposed to it
  - Changing it afterwards is costly, if not impossible
- A good data model is essential for being efficient in development and runtime
- The LHC experiments have very complex data models
  - First of all, they worked
  - Fairly hard to adapt to new technologies like **vectorization**
  - Not future-proof

- If there is one component to really spend time on, it is the data model
- Work on this is ongoing

Detector Description

Detector Description is an essential ingredient

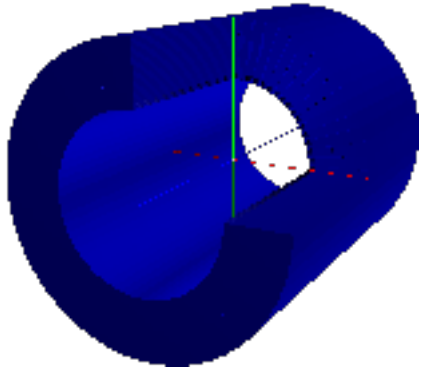
- There will be plenty of detector design iterations
- Important to stay flexible and don't couple SW to a particular detector design

AIDA triggered a new project DD4hep (*)

- Covering simulation, reconstruction, alignment in a consistent way
- Covering the **entire lifecycle of an experiment** (from initial design ideas until final detector)
- Provides straight path to Geant4 via GDML and generic detector constructors, sensitive elements etc...
- Being used in ILC/CLIC

FCC-hh working on DD4hep description

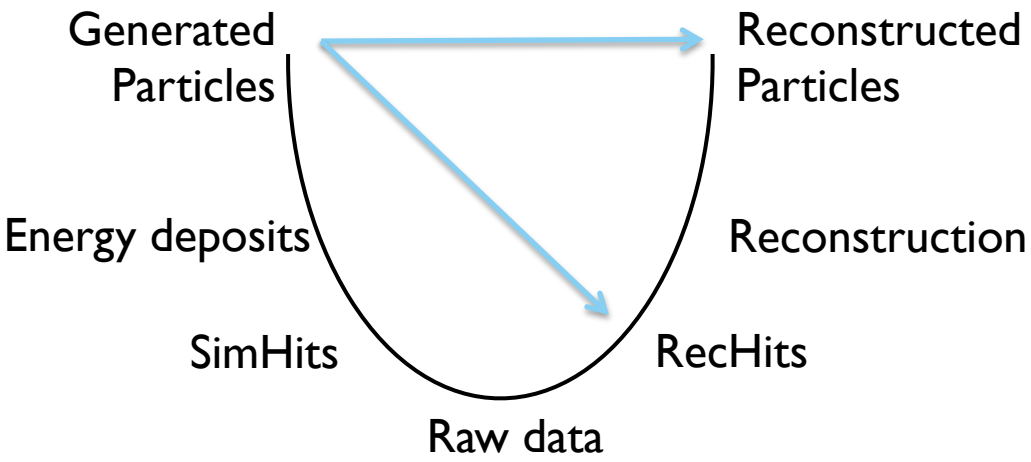
- Needed start help by authors, but status very promising
- Steep and painful learning curve



(*) http://aidasoft.web.cern.ch/DD4hep

Simulation

- At different stages different level of detail required
  - generator smearing vs. fast sim vs. full sim



Different levels of detail =
Doing short cuts in the full workflow

Simulation II

- FCC choices are
  - Delphes (*)
  - Custom fast simulation
  - Geant4
- Interfacing it to the same framework is the way to progress
- Generators trivially covered – HepMC as input standard
  - Code example in place
- Lots of work, but rather clear what to do
- First visible milestone for new SW would be reproducing existing results done w/ Delphes previously

(*) https://cp3.irmp.ucl.ac.be/projects/delphes

Integrating Delphes

- (Desired) milestone for technical work is mid of August
- However, Delphes is a small framework of its own
  - Own data formats
  - Own event store
  - Own configuration system
- Authors provided tutorial to us
- Nprovide an interface to use Delphes as a library rather than as framework
  - Currently simple all or nothing approach
  - Unfortunately it makes many of the very nice features not directly accessible
  - Iteration w/ developers on this did not start yet, but from past experience expect productive collaboration

Reconstruction and Analysis Code

- Reconstruction
  - Apart from fast jet no global solution around, but many individual solutions one can select from
    - E.g. multiple particle flow implementations
  - Requires assessment of existing code
  - Whatever is chosen needs to be adapted to common data model

- Analysis
  - Allow multiple paradigms to do analysis
    - C++ and Python
  - Many (n-tuple based) solutions exist
    - People come with their code from different experiments
  - Common solution very desirable, but hard to achieve
  - Need to collect requirements and needs
  - Ongoing efforts to interface existing code from CERN CMS: https://github.com/cbernet/heppy

Next concrete steps

Software stack of external SW (ROOT, Geant4, Generators)

- Infrastructure in place and candidate build in active use

Geometry Description (DD4hep)

- FCC-hh test setup in place

Example workflows

- Currently setting up example workflow from reading HepMC up to final histogram

Data Model

- Ongoing work on initial data model

Simulation

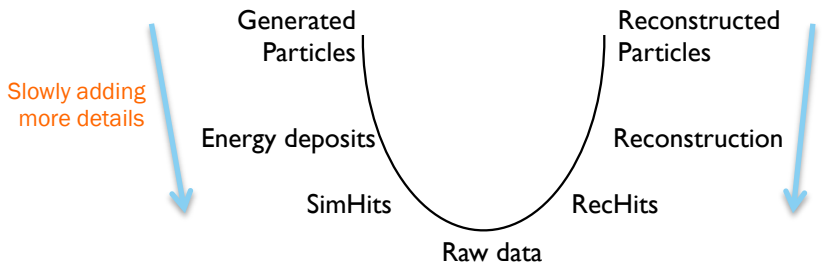
- Interface Delphes (testing volunteers welcome!)
- Interface Geant4

Reconstruction

- Need to add pieces according to progress in simulation

Training

- A small coding sprint early July for new developers
  - Date to be decided
  - If interested, can do a little hands-on before
- Small end-user training after summer break in early September



Work items being collected in https://sft.its.cern.ch/jira/browse/FCC

Currently less than 1 FTE working on it, thus physics milestones and timescales define how pragmatic every item has to be tackled

Please join fcc-experiments-sw-dev@cern.ch if you are interested in rapid progress!

The PH-SFT Group at CERN

- The SFT group plays a major role in development and maintenance of common scientific software for the physics experiments
  - ROOT
  - Geant4
  - Generator Services
  - CernVM
  - …
- Helps experiments by providing a software stack
  - External libraries (approx. 110)
- Several group members have direct responsibilities in software projects of the LHC experiments.
- We do R&D for future HEP software
  - GeantV
  - GaudiHive
- Recently started effort to support post-LHC experiments
  - Our idea is to propose a 'turn key' solution covering most of the experiments' needs
  - Currently one staff and one doctoral student (integral ~0.5 FTE) contributing to the FCC efforts