

Results of meeting with Intel CnC Team

Marc Paterno
Christopher D. Jones
29 February 2012

What we said

- We presented a very high-level view of HEP event-processing frameworks.
- Description was sufficiently generic to cover (almost?) all our frameworks.
- We concentrated on explaining the features we thought had not previously been made clear to the CnC team.
- We had a long discussion (more than an hour) to clarify the issues involved.
- The slides we used for that presentation are posted to the Indico site.

What they said: the easy part

- They are clearly interested in the type of problem we're solving, and are willing to work with us on an attempt to use CnC in the framework design.
- They already have many of the features we need:
 - “Mags” to label computed data products, to distinguish between instances of the same class.
 - Most (all?) data elements are collections.
 - Created data are immutable.
 - “Modules” never communicate directly.
 - If modules are internally thread-safe, parallel computation is guaranteed to be safe.
 - One can use TBB parallelism inside CnC programs.
 - Nested CnC parallelism is planned, but not (fully?) implemented.
 - They support reductions, which will be important (e.g. for combining histograms).

What they said: the hard part

- Some elements are challenging: primarily, loading a “framework configuration” at runtime, rather than determining it a compile-time.
- The CnC team are working on a workflow analogous to ours (encoding the processing graph we used as an example in our explanation), in which the graph will be described at *runtime* not at compile time. This requires some work on their part.
- It is unclear how they will deal with loading modules from dynamic libraries; we are hoping this will be solved in their demonstration.

The workflow we presented

- Circles represent data products.
- Rectangles represent processing modules.
- Any or all data products may be written to an output file.

