ILDG

Metadata working group report

ILDG

Overview

- Extending QCDml
  - Propagator formats
    - USQCD and ETMC
  - metadata?
- Using QCDml
  - Workflow as a tool for
    - Data provenance
    - metdata capture
  - FNAL group has already started to look at this

Propagator sharing

- Two groups already store propagators internally
  - USQCD
  - ETMC
  - UKQCD would in principle share with USQCD
    - if we actually had a machine. Sigh …
- Scope for a common format?
  - Not all propagators are the same
  - What about the source? $\longrightarrow$
  - What about the metadata?
- This work is already being done
  - Can we make a common format
    - *De facto* standard
  - ILDG adopt formats already in use

USQCD format

- Four formats
  - **`C1D12`**: One complex scalar source record and twelve solution records, one for each source spin and color. The solution records correspond to each source spin and color. The order of source spin and color should be sequential with color varying most rapidly
  - **`CD_PAIRS`**: Alternating source and solution for any number of pairs. The source in each case is a complex field
  - **`DD_PAIRS`**: Alternating source and solution for any number of pairs. The source in each case is a Dirac field
  - **`LHPC`**: [USQCD standard under development.]
- Source field included
- QIO records (Lime records underneath)

ILDG General QIO file organization

- Series of logical QIO records
  - File info
  - Record info plus payload
  - Record info plus payload
  - ... (unlimited)
- Record info plus payload: four LIME records
  - Private record info
  - User record info
  - Binary payload
  - Checksum for payload
- Each LIME record has a unique LIME type. Helps if non-QIO software reads the file.
- User record contains unconstrained XML record
  - metadata?

ETMC Format

- Extension to SciDAC format
  - **`DiracFermion_Sink`** no source, sinks
  - **`DiracFermion_Source_Sink_Pairs`** source, sink
  - **`DiracFermion_ScalarSource_TwelveSink`** source, 12 sinks
  - **`DiracFermion_ScalarSource_FourSink`** source, 4 sinks
- One record for each fermion field plus 2(3) others
  - In style of ILDG gauge config format
- **`etmc-propagator-format`**

```
<etmcFormat>
 <field>diracFermion</field>
 <precision>32</precision>
 <flavours>1</flavours>
 <lx>4</lx> <ly>4</ly> <lz>4</lz> <lt>4</lt>
 </etmcFormat>
```

ILDG ETMC

- Next is `scidac-binary-data`
- One record for each flavour
  - data layout is
  - `t,z,y,x,s,c`
- Also include
  - gauge configuration lfn, checksum and SciDAC checksum
  - Indentify configuration
- ETMC can read SciDAC propagators

Propagator Summary

- Many different propagators
  - need multiple formats
- This represents a methodology for writing propagators
- ETMC extension includes
  - data size/layout
    - In same style as ILDG gauge cfg format
  - identifiers for gauge cfg
    - minimal data provenance
- MDWG should consider adoption as ILDG standard
  - ETMC extensions recommended/required?
  - Metadata is very minimal
    - Could for ease of use
    - Poor for data provenance

Workflow

- Many different workflow tools exist
  - allow user to build, repeat, reuse a pattern of work
- Metadata capture and Data provenance
  - Recording what was done is an important part of scientific prudence
  - Workflow can help by recording *everything*
    - *automatically*
    - *systematically*
- UK attempting to obtain funding for proje
- Fermilab group already started work
  - Include Jim Simone's slides

Workflow Project

Fermilab, IIT and Vanderbildt

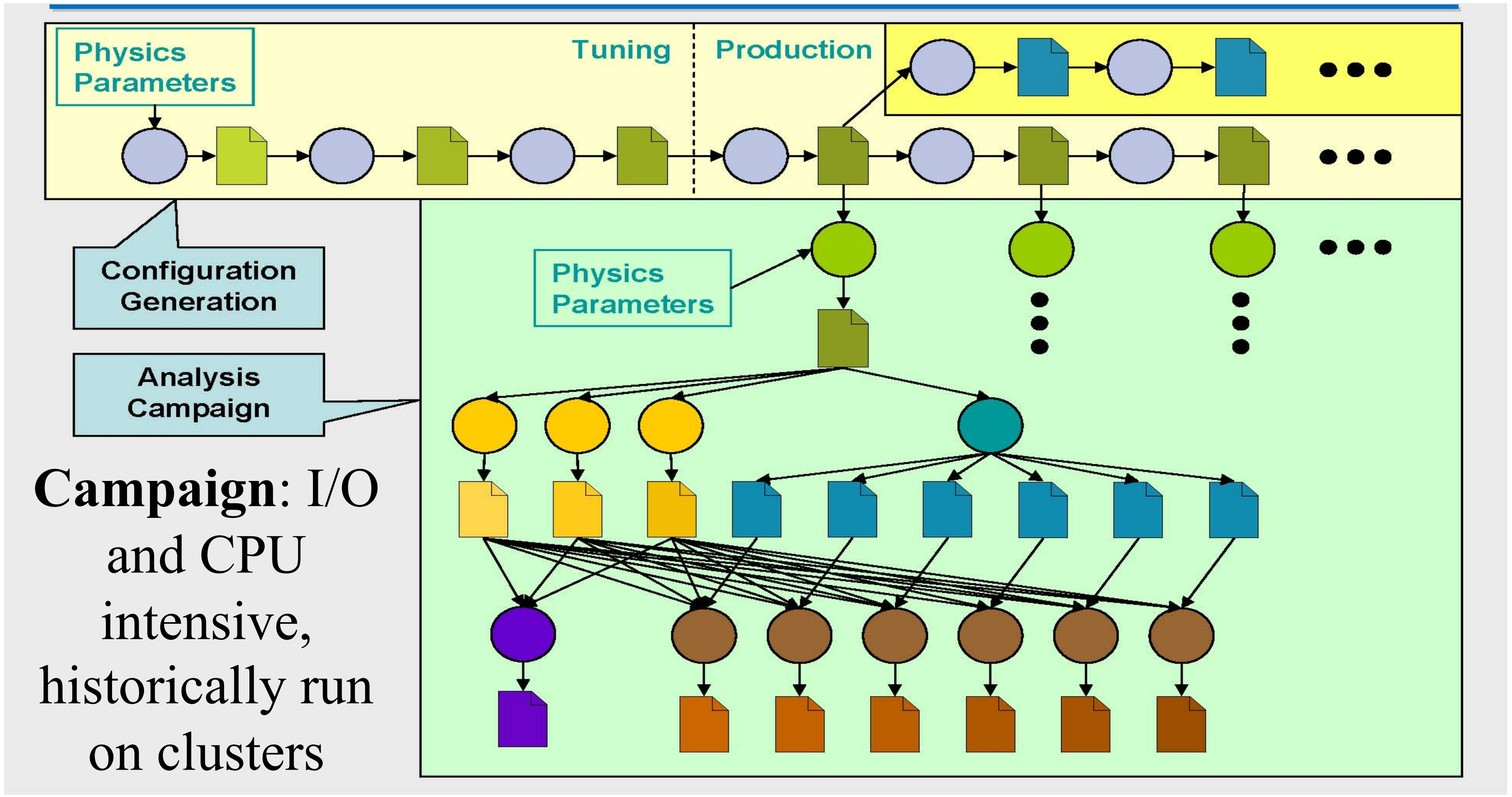
SciDAC2: A National Computing Infrastructure for Lattice Quantum Chromodynamics

LQCD computing issues

- Detection and re-running of failed jobs is an enormous burden on scientists.
- “Smart” rerun logic greatly complicates custom scripts.
- Reuse hampered by custom scripts with hard-coded information, e.g. directory names, parameters etc.
- Provenance record may consist solely of millions of disorganized log files.
- Poor data management impedes use/sharing of data.
- Issues become even worse with bigger computers and larger projects.

Typical Workflows

Confgen: Simpler in structure, simpler I/O, LCF application, shared products



Campaign: I/O and CPU intensive, historically run on clusters because of small jobs that run

Workflow Project Purpose

- Purpose is to create a system for
  - Designing *workflows* that describe *campaigns (physics processes)*,
  - Instantiating these *workflows* with files and physics parameters,
  - Executing the instantiated *workflows* on LQCD resources,
  - And recording workflow results
- Major Requirements
  - Describe physics independent of execution resource constraints
  - Store workflow definitions and application configuration parameters
  - Monitor and keep execution statistics
  - Track provenance
  - Detect and record job failures – fault tolerance
  - Support recovery/restart of applications
  - *Reuse existing tools where possible*

LQCD-Specific Components

- Required to be usable and understandable by scientists
  - Domain specific terminology and relationships
  - Capture the process of doing the science
  - Reduce learning curve for conducting complex science
  - Provides a documentation trail for all collaborators
- Focus
  - Parameterization of applications
  - Run-time history of participants and workflows
  - Provenance for data products
  - Secondary product data storage
- Major aspects
  - Management of parameter sets (physics and system related)
  - Management of workflow templates and participants
  - Persistent workflow state
  - Fault tolerance (cluster reliability project)
  - User query facilities

A Prototype

- Pieces tackled:
  - Parameterization of applications
  - Run-time history of participants and workflows
  - Provenance for data products
  - Secondary product data storage
- RDBMS for information
- Participants for running example applications
- Ruby on Rails
  - Active record object relational mapping pattern
  - Web user interface for configuration, control, and monitoring.
- Now: OpenWFEru (Ruote) as workflow engine
  - Business Process Model (BPM) engine
- Soon: evaluate Pegasus

Community Interactions

- Conferences and Workshops
  - L. Piccoli *et al*, "The LQCD Workflow Experience: What We Have Learned," *SC07 Proceedings* (2007)
  - L. Piccoli *et al*, "Tracking LQCD Workflows," *Lattice 2008 Proceedings* (2008)
  - A. Dubey *et al*, "Using Runtime Verification to Design a Reliable Execution Framework for Scientific Workflows," *EASe'09*.
  - L. Piccoli *et al*, "Lattice QCD Workflows: A Case Study", SWBES08 at E-Science 2008.
  - L. Piccoli *et al*, "LQCD Workflow Execution Framework: Models, Provenance, and Fault-Tolerance," *CHEP'09*.
- Existing Tools
  - Swift - U.Chicago
  - Askalon – Innsbruck
  - (Also met with Pegasus and Kepler teams)