

Tracking LQCD Workflows

Luciano Piccoli^{1,2}, James N. Simone², James B. Kowalkowski², Xian-He Sun¹

Donald J. Holmgren², Nirmal Seenu², Amitoj G. Singh²

¹Illinois Institute of Technology, Chicago, IL, USA 60616

²Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL, USA 60510

Introduction

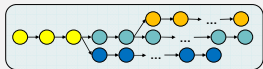
LQCD configuration generation and analysis campaigns require coordination of physics parameters, cluster parameters, binaries, input and output files.

- Python, Perl and shell scripts are currently used to execute configuration generation and campaigns.
- Secondary data, error messages and status are stored in log files generated by these scripts.
- Outputs are generated mainly as files from workflow tasks.

Workflow systems are capable of providing the coordination required by LQCD campaigns.

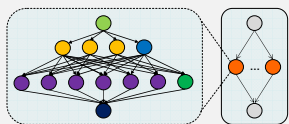
The Environment

- Tens of users running hundreds of complex workflows (campaigns) concurrently.
- Campaigns are composed of identical embarrassingly parallel sub-workflows running on different sets of inputs.
- Campaign running time may span several months.
- Campaigns execute thousands of MPI jobs.
- Large input and output files, from hundreds of MBytes to a several GigaBytes in size.
- Campaigns run on dedicated clusters with Infiniband and Myrinet (qcd, kaon and pion at Fermilab).
- Typical workflows:
 - Configuration Generation workflow (prototype)



- MILC asqtad configuration generation.
- Tadpole factor, u0, is first self consistently set during tuning phase of production.
- Series may fork to speed up production of configurations.
- Forks may be run at different facilities.

- Two-point analysis campaign sub-workflow

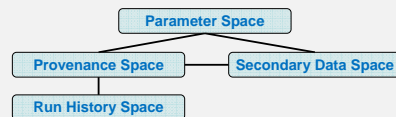


Workflow Systems

- **Advantages**
 - Generating same results by recomputing same workflows (reproducible research).
 - Data products can be regenerated as needed (storage space vs. computing capability).
 - Standardize the specification of workflows (use of same description language).
 - Searchable record of data products and execution history.
 - Simplify recovery from execution failures.
- **Features missing from current workflow systems**
 - Tracking of outputs in forms other than files (e.g. error messages and checksums)
 - Decoupling workflow description and parameters. Tracking all inputs provided to a workflow not restricted to input files, but input parameters and environment settings. (e.g. a workflow described as a script has a hardcoded value for the error tolerance. This parameter should be tracked by the system).
 - Efficient usage of computing and storage resources. Current workflow systems focus on a single workflow instance, while LQCD workflows have interdependencies (e.g. files shared between workflows).

Data Tracking Components

- Managed data are divided into four spaces for tracking and to minimizing cross-references.



- **Parameter Space:** Archives all parameters used as input for a workflow, including physics parameters (e.g. quark masses), algorithmic parameters (e.g. convergence criteria) and execution parameters (e.g. number of nodes used).
- **Provenance Space:** Keeps information regarding inputs and outputs for each workflow participant.
- **Secondary Data Space:** Used for storing secondary information generated by workflow participants. (e.g. plaquette values from each iteration).
- **Run History Space:** Archives detailed information about execution, including algorithm versions used and outputs generated.

Prototype Features

- Workflow independent framework to enable tracking of:
 - Provenance and location of generated data.
 - Values and history of secondary products.
 - Workflow input parameters.
 - History of execution and environment used by participants and workflows.
- Allows advanced queries about workflow history and status:
 - Examples of SQL query used to retrieve data that follows certain criteria:
 - Which configuration files were generated using algorithm su3_rmd version 1.2?
 - What are the configuration files generated with error smaller than 10E-5.

Prototype Implementation

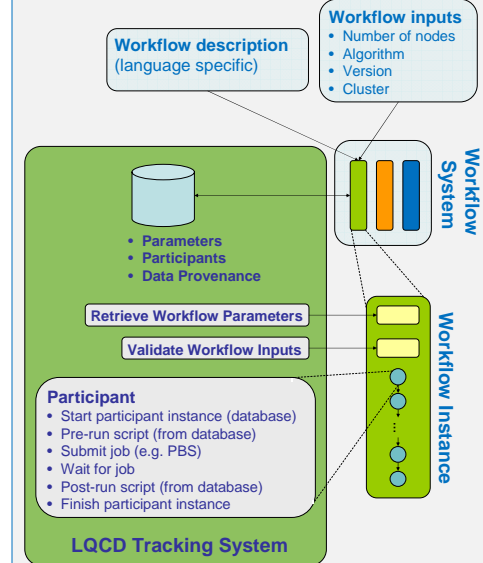
Prototype implementation currently handles configuration generation workflows. It requires some components to be configured, which are part of the provenance and secondary data spaces. Most of these components can be used for any type of workflows. This model provides facilities for the addition of workflow specific components. The following is a detailed description of the prototype model components, including components specific to configuration generation:

- **Parameter Space**
 - **Parameter:** represents basic types (string, integer, float) saved as name-value pairs; can be organized into arrays.
 - **Parameter Set:** collection of parameters.
- **Provenance Space:**
 - **Configuration File:** represents a configuration file generated as an output by a workflow. Contains a reference to the actual file location and file properties.
 - **Analysis Series:** contains a sequence of configuration files used for analysis campaigns.
- **Secondary Data Space:**
 - u0: sequence of u0 values generated during the tuning phase until a production value is reached.
- **Run History Space**
 - **Participant Type:** describes the type of participants, for example a known type of algorithm.
 - **Participant:** contains information about the implementation of a participant type. Participants are versioned.
 - **Participant Instance:** history of each participant run.
 - **Cluster:** details about the computers used for workflow processing.

Workflow Integration

- **Modeling**
 - Workflow systems have the concept of participant or actor that represents an atomic activity. They are connected to other activities through data and/or control dependencies.
 - The workflow participants or actors invoke the binary code corresponding to the participant abstraction in the database.

- **Execution**



- **User interface**
 - Select workflow inputs and parameters.
 - Monitor the execution of workflows.
 - Create new configuration parameters.
 - Monitor generated files and secondary data.

Improvements

- **Include campaign history in the tracking database:** expand database to include running and concluded campaigns. Should contain pointers to code repository where workflows are kept.
- **User interface:** expand functionality available to users through web access.