PROCEEDINGS OF SCIENCE

Provenance model for Lattice QCD

Tanja Auge,[a] Gunnar Bali,[b] Meike Klettke,[a] Bertram Ludäscher,[c] Wolfgang Söldner,[b] Simon Weishäupl[b] and Tilo Wettig[b,*]

[a] *University of Regensburg, Faculty of Computer Science and Data Science, Germany*
[b] *University of Regensburg, Department of Physics, Germany*
[c] *University of Illinois at Urbana-Champaign, School of Information Sciences, USA*

E-mail: tilo.wettig@ur.de

Workflow management has become an important topic in many research communities. Here, we focus on the particular aspect of provenance tracking. We follow the W3C PROV standard and formulate a provenance model for Lattice QCD that includes the ensemble-generation and measurement parts of the Lattice QCD workflow. Since many important provenance questions in our community require extensions of this model, we propose a multi-layered provenance approach that combines prospective and retrospective elements.

The 40th International Symposium on Lattice Field Theory (Lattice 2023)
July 31st - August 4th, 2023
Fermi National Accelerator Laboratory

[*] Speaker

1. Introduction and motivation

Lattice QCD (LQCD) is a mature field of research that generates and analyzes a large amount (i.e., many PetaBytes) of data. It is therefore desirable to implement the so-called *FAIR Guiding Principles for scientific data management and stewardship* [1], which are concerned with data and metadata as well as workflows. FAIR stands for *Findable*, *Accessible*, *Interoperable*, and *Reusable*. While these concepts are largely self-explanatory, a detailed discussion can be found in [1]. The LQCD community has already taken steps towards implementing FAIR principles long before the term FAIR was established. About twenty years ago the *International Lattice Data Grid* (ILDG) was created to facilitate the sharing of gauge-field configurations [2]. Ref. [3] describes the ILDG metadata standard QCDml, which includes tracking information for ensemble generation. Ref. [4] introduced EspressoDB, a systematic workflow and data management tool for LQCD. In this contribution, we address the issue of workflow provenance and propose a provenance model for parts of the LQCD workflow [5].

In general, the term *provenance* refers to information that describes the production process of an end product. The importance of provenance in LQCD can perhaps best be understood by a real-world example. We stored a set of configurations at an external research institute, and due to file-system problems at this institute, silent data corruption occurred. If we had performed measurements on the corrupted configurations before the corruption was noticed, we would have obtained incorrect results. In this case, provenance information is useful in two respects: On the one hand, provenance can identify the (downstream) measurement results affected by the corrupted configurations. On the other hand, provenance can trace the incorrect measurement results back (upstream) to the corrupted configurations.

Going beyond this example, there are a number of important questions that provenance can help answer in an automated way, such as

Q1 Which datasets are affected by an error or bug?

Q2 How are datasets affected by modifying a parameter?

Q3 Who was involved in generating the data?

Q4 Which codes and experts are needed to repeat a workflow?

Q5 Which data/parameters are needed to (re-) produce a result?

These and other questions address different aspects of provenance, as we will discuss below. In Section 2 we briefly describe the W3C PROV standard on which our provenance model is based, in Section 3 we use this standard to construct a provenance model for LQCD workflows, and in Section 4 we propose an extension to the W3C PROV standard motivated by the requirements of LQCD. This contribution summarizes Ref. [5], to which we refer for a more detailed exposition.

2. W3C PROV concepts

Our work is based on the W3C PROV standard [7, 8], which can be formulated as a graph and is shown in Fig. 1. There are three types of nodes in the graph: *entities* that can be derived from other entities, *activities* that generate or use entities, and *agents* that perform/control activities or

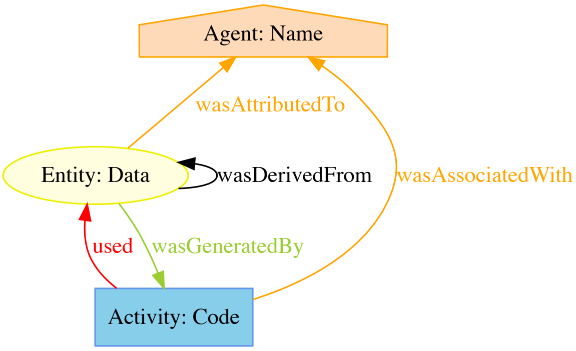


Figure 1: Basic concepts and relations of the W3C PROV standard (based on [6], figure from [5]).

produce entities. Examples are given for each case. The relations between the nodes are described by edges as shown in the figure. For example, an entity *wasGeneratedBy* an activity, or an activity *wasAssociatedWith* an agent. Note the conventions for the direction of the arrows.

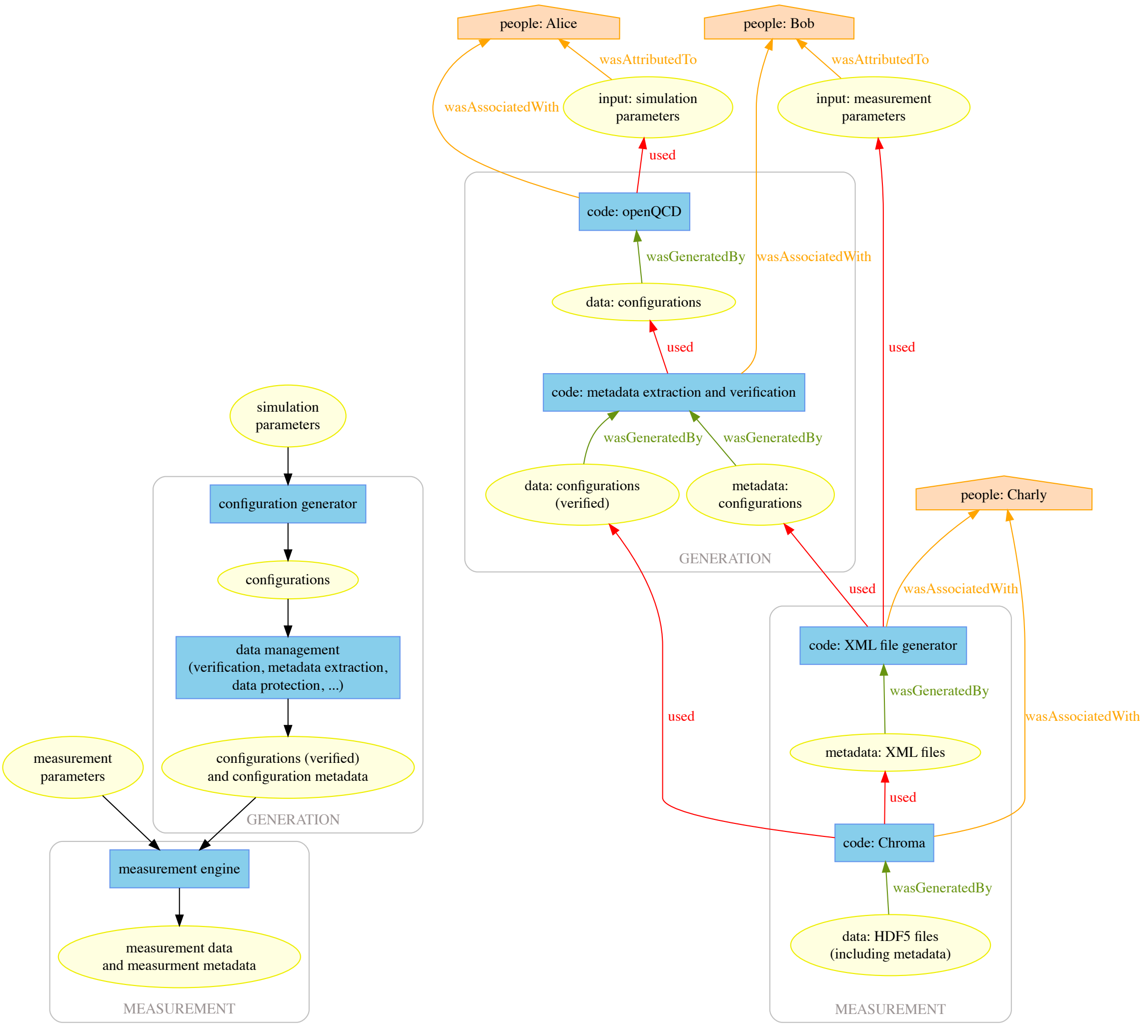
To prepare for the discussion below we distinguish two forms of workflow provenance: *retrospective provenance* is information about past workflow executions and data derivations, while *prospective provenance* captures the structure of the workflow and provides a recipe for future workflow executions.

3. Lattice QCD workflows with and without provenance

The LQCD programme can be roughly divided into three parts: *generation* (of gauge-field ensembles), *measurement* (i.e., computation of correlation functions), and *analysis* (i.e., computation of observables). Here, we focus on the first two parts, which are more compute-intensive and more generic than the analysis part. Fig. 2a shows a generic LQCD workflow for generation and measurement. We use the same shapes and color codes as in Fig. 1 to distinguish activities and entities. The simulation data include both physical parameters and algorithmic parameters. The activity labeled “data management” includes a number of steps to ensure correctness of the data, to backup and/or archive the data, and to process the metadata. Note that the measurement part also includes data management, which is not shown to avoid cluttering the figure.

To be able to answer provenance-related questions such as **Q1** to **Q5** in the introduction, we now need to add provenance information. Note that we require different provenance elements to address the different questions. Questions **Q1** and **Q2** are more data-related, while questions **Q3** to **Q5** are more workflow-related. In Fig. 2b we show the W3C PROV model we created for the specific LQCD workflow used by our research group. The four activities (blue code boxes) read the input parameters, manage the data, and generate the final HDF5 files. The seven entities (yellow boxes) represent input parameters, data, and metadata. The model further includes three agents (orange boxes) who execute the activities and are responsible for the input parameters. To keep the figure simple, we have dropped backup and archiving in the generation part. Also, we have split the measurement activity of Fig. 2a into two activities to better reflect the actual workflow.

Again, the directed edges describe the relationships between the agents, activities and entities. However, we note that one edge type is missing. Since the flow of data entities is unique, *w*-edges



(a) Generic workflow without provenance. **(b)** Specific workflow formulated as a W3C PROV model.

Figure 2: Representation of the first two parts (generation and measurement) of a Lattice QCD workflow (both figures from [5]).

(*wasDerivedFrom*) are simulated by a chain of u-edges (*used*) and g-edges (*wasGeneratedBy*).

Let us see how questions Q1 to Q5 can be answered using the provenance model. Questions Q1 and Q3 require retrospective provenance, questions Q4 and Q5 are prospective provenance queries, and question Q2 combines both types of provenance. Question Q3 can be answered directly from the provenance graph using the name of the person. Similarly, questions Q4 and Q5 can be answered by inspecting the provenance graph and identifying the corresponding activities, entities, and agents. Question Q1 is more complex. If we realize that the output data are incorrect, the data derivation chains must be traced from the results back to the sources (upstream propagation). If we find an error in an activity or entity, such as the silent data corruption mentioned above, the erroneous activity or entity needs to be fixed, and subsequent entities need to be recomputed or corrected (downstream propagation). In both cases we need to trace the dependencies along the derivation chains in the provenance graph. Finally, question Q2 requires prospective provenance if we are only interested

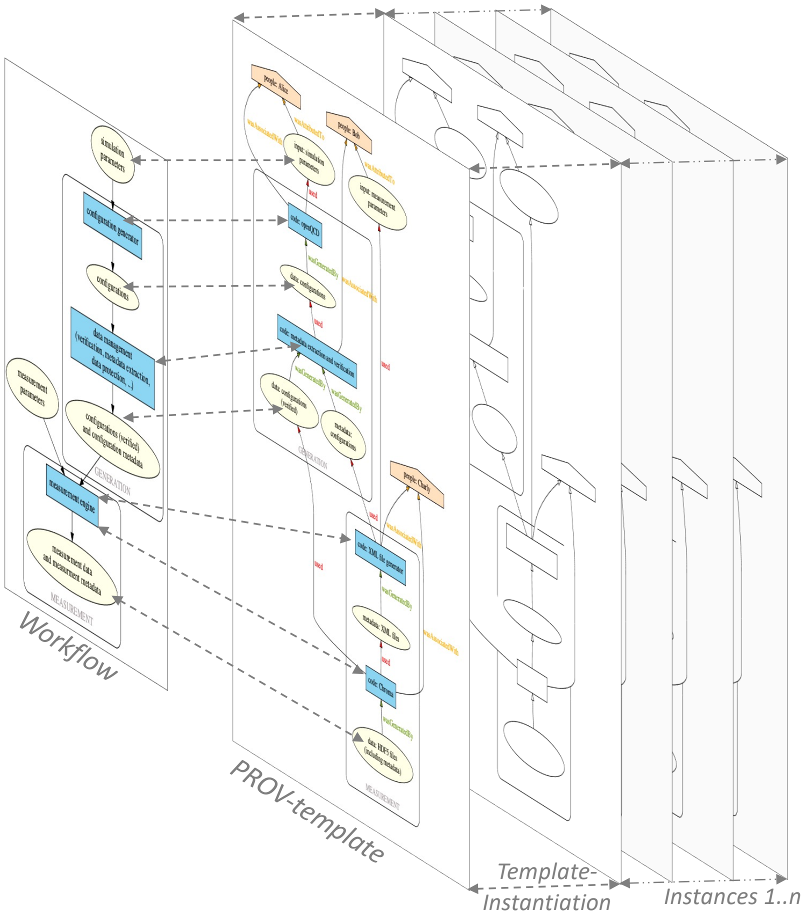


Figure 3: Proposed hybrid provenance model: The conceptual workflow (left) naturally maps to a W3C PROV template model (middle). Every workflow execution creates PROV-compatible retrospective provenance graphs (right), i.e., instances $1, \ldots, n$ of the provenance template in the middle (figure from [5]).

in the dependency structure at the conceptual level, while we need retrospective provenance if we are interested in the effect of a parameter change on a dataset in a previous workflow run.

4. Towards layered provenance

The generic workflow in Fig. 2a can be viewed as a template. If we disregard the provenance information in Fig. 2b for a moment, this graph is then an instance of Fig. 2a in which the generic elements are specialized to the activities and entities relevant to our research group. Similarly, we can view the provenance graph of Fig. 2b as a *provenance template graph*. Every workflow execution then generates its own provenance instance graph, in which schema-level elements (e.g., `data:HDF5 files`) are replaced by references to concrete instance objects (e.g., `X251r000n1000_run3.hd5`).

Based on this observation we propose to extend the W3C PROV model to include both a *template-level* provenance graph and the corresponding *instance-level* provenance graphs including concrete values of all input parameters, names, timestamps, version numbers, etc. In addition, we propose a workflow layer from which the provenance template graph is derived. This proposal is shown in Fig. 3. A model based on this proposal should satisfy the following requirements:

1. The community-level workflow structure should be linkable to provenance template graphs (research group level).
2. The instance-level provenance graphs of the hundreds or thousands of runs (with varying parameter settings etc.) should be automatically linked to a provenance template.
3. A domain-aware provenance model should allow users to distinguish different types of data, e.g., using namespaces.

The next step is to develop a concrete implementation. There are several options to do so. Here, we restrict ourselves to an outline of what a possible LQCD prototype could look like. Existing LQCD workflows already capture a large amount of provenance information in log files. This information could be collected by a Python-based provenance harvesting tool and then deposited in a provenance store, i.e., a suitable database. In many cases provenance information can also be obtained from file or folder names. The remaining provenance information required by our model but not yet available through harvesting would have to be generated by, e.g., a light-weight provenance recorder, code instrumentation, or writing additional information to log files.

5. Summary and outlook

We have taken first steps to bring together the LQCD and provenance communities to define comprehensive standards for data management, including provenance metadata, and to co-develop W3C PROV extensions suitable for LQCD. Concretely, we have proposed a provenance model for the generation and measurement parts of the LQCD workflow based on the W3C PROV standard. We started from a generic workflow (Fig. 2a) from which we derived a PROV template (Fig. 2b). We then proposed a layered model consisting of a workflow layer, a provenance template layer, and an instance layer (Fig. 3). This model allows us to answer many provenance-related questions such as **Q1** to **Q5**.

In future work we plan to refine our initial model proposal and to implement a prototype that will allow us to evaluate its efficacy and practicality. We also plan to apply our provenance model to the third part (analysis) of the overall LQCD workflow, which is less generic and more collaboration-specific.

Acknowledgments

We thank Sara Collins, Christoph Lehner, Nils Meyer and Stefan Solbrig for stimulating discussions. This work was supported in part by DFG project 460248186 (PUNCH4NFDI).

References

[1] M.D. Wilkinson et al., *The FAIR Guiding Principles for scientific data management and stewardship*, *Scientific Data* **3** (2016) 160018.

[2] A.C. Irving, R.D. Kenway, C.M. Maynard and T. Yoshie, *Progress in building an International Lattice Data Grid*, *Nucl. Phys. B Proc. Suppl.* **129** (2004) 159 [hep-lat/0309029].

[3] C.M. Maynard and D. Pleiter, *QCDml: First milestone for building an International Lattice Data Grid*, *Nucl. Phys. B Proc. Suppl.* **140** (2005) 213 [hep-lat/0409055].

[4] C.C. Chang, C. Körber and A. Walker-Loud, *EspressoDB: A scientific database for managing high-performance computing workflows*, *J. Open Source Softw.* **5** (2020) 2007 [1912.03580].

[5] T. Auge, G. Bali, M. Klettke, B. Ludäscher, W. Söldner, S. Weishäupl and T. Wettig, *Provenance for Lattice QCD workflows*, *WWW '23 Companion: Companion Proceedings of the ACM Web Conference 2023* (2023) 1524 [2303.12640].

[6] Y. Gil et al., *PROV Model Primer*, 2013.

[7] L. Moreau et al., *The open provenance model core specification (v1.1)*, *Future Gener. Comput. Syst.* **27** (2011) 743.

[8] P. Groth and L. Moreau, *PROV-Overview*, 2013.