Provenance for Lattice QCD workflows

Tanja Auge
University of Regensburg, Faculty of
Computer Science and Data Science,
Germany
tanja.auge@ur.de

Gunnar Bali
University of Regensburg,
Department of Physics, Germany
gunnar.bali@ur.de

Meike Klettke
University of Regensburg, Faculty of
Computer Science and Data Science,
Germany
meike.klettke@ur.de

Bertram Ludäscher
University of Illinois at
Urbana-Champaign, School of
Information Sciences, USA
ludaesch@illinois.edu

Wolfgang Söldner
University of Regensburg,
Department of Physics, Germany
wolfgang.soldner@ur.de

Simon Weishäupl
University of Regensburg,
Department of Physics, Germany
simon.weishaupl@ur.de

Tilo Wettig
University of Regensburg,
Department of Physics, Germany
tilo.wettig@ur.de

ABSTRACT

We present a provenance model for the generic workflow of numerical Lattice Quantum Chromodynamics (QCD) calculations, which constitute an important component of particle physics research. These calculations are carried out on the largest supercomputers worldwide with data in the multi-PetaByte range being generated and analyzed. In the Lattice QCD community, a custom metadata standard (QCDml) that includes certain provenance information already exists for one part of the workflow, the so-called *generation* of configurations.

In this paper, we follow the W3C PROV standard and formulate a provenance model that includes both the generation part and the so-called *measurement* part of the Lattice QCD workflow. We demonstrate the applicability of this model and show how the model can be used to answer some provenance-related research questions. However, many important provenance questions in the Lattice QCD community require extensions of this provenance model. To this end, we propose a multi-layered provenance approach that combines prospective and retrospective elements.

KEYWORDS

Workflow provenance, W3C PROV, Lattice QCD

1 INTRODUCTION

Provenance generally refers to "any information that describes the production process of an end product, which can be anything from a piece of data to a physical object" [17]. This is a challenge we also face in our application, Lattice QCD workflows. Let us give a brief introduction to this application. On the fundamental level, our understanding of nature rests on the Standard Model of elementary particles and their interactions. The Standard Model is formulated in terms of quantum field theories, including *Quantum Chromodynamics* (QCD). Currently many experimental and theoretical efforts are underway to search for physics beyond the Standard Model. These searches require supporting QCD calculations that must be carried out to high precision. The preferred tool for such calculations is the numerical simulation of QCD on a space-time lattice (*Lattice QCD*). Lattice QCD has evolved over more than four decades and is now a mature field with many hundreds of researchers all over the world. Similar to experiments, where one first collects data and later analyzes them, the Lattice QCD programme factorizes into three parts, *generation*, *measurement*, and *analysis* [4]. In the first part, ensembles of so-called *gauge-field configurations* are generated using the *Markov chain Monte Carlo method* and then stored to disk. In the second part, so-called *correlation functions* that are relevant for the specific physics programme are computed on these configurations, and the resulting data are also stored to disk. In the third part, the correlation functions are combined into the observables of interest. The first two parts are very compute-intensive and use the largest supercomputers worldwide. At present, typical data sizes for a given collaboration are about one PetaByte of ensembles and several PetaBytes of derived data [4].

Already twenty years ago the Lattice QCD community initiated the *International Lattice Data Grid* (ILDG) [20, 22, 23, 27] for the purpose of sharing ensembles of configurations (according to what is nowadays known as the *FAIR principles* — FAIR stands for **F**indable, **A**ccessible, **I**nteroperable, and **R**eusable, where Reusable includes provenance aspects [14, 37, 42]). For the generation part of the Lattice QCD programme, the ILDG metadata standard (QCDml

[27, 41]) also includes tracking information, but a full provenance concept has not been developed yet. In this paper we present a provenance model for the generation part using the W3C PROV standard (which did not exist when the ILDG metadata standard was created). Furthermore, we extend our model to include the measurement part of the Lattice QCD programme, for which no community effort on provenance has been made so far. We view this as an important step towards implementing the FAIR principles in research data management.

To illustrate the importance of provenance in the Lattice QCD context, let us discuss an example that occurred in practice. A set of configurations was stored at an external research institute. During the storage period, silent data corruption took place due to a file-system problem. Measurements based on the corrupted configurations could have been performed before the data corruption was noticed. In such a situation, provenance can trace incorrect measurement results back (upstream) to the corrupted configurations. Conversely, provenance can identify the (downstream) measurement results that may have been affected by the corrupted configurations.

Another important aspect is the reproducibility and replicability (as defined in [1]) of results published in a scientific article. On the one hand, we need provenance information to determine what part of the data was actually relevant for the published results. On the other hand, we also need provenance of the associated workflows. This will allow us (or others) to reconstruct a published result in a transparent way. For details we refer to [2, 3].

To integrate provenance into Lattice QCD workflows we need a sustainable provenance model. After analyzing other physical problems such as [21, 40] we decided to use the W3C PROV model [15], which defines "a data model, serializations, and definitions to support the interchange of provenance information on the Web." In [21], the authors "lay the foundation for making an automated provenance generation tool for astronomical/data-processing pipelines" but do not use the W3C PROV model, while [40] describes a W3C PROV model to make the software BACARDI, which provides a database with information about active and inactive objects orbiting the Earth, provenance-aware. The model is implemented using the Python library prov [9].

The provenance model towards which researchers naturally gravitate starts from a high-level conceptual workflow model and then specializes or instantiates details to answer provenance questions. Therefore, to generate our provenance model we first describe a generic Lattice QCD workflow (see Figure 2 below). In a second step, we analyze a concrete Lattice QCD workflow in more detail and develop an initial W3C PROV model for it (Figure 3). Finally, we propose a multi-layered model (Figure 4) that we envisage to be realized as a W3C PROV extension.

The structure of this paper is as follows. We first provide background information on provenance and the W3C PROV standard (Section 2). Next, we introduce our use case (Lattice QCD) and establish our W3C PROV-based provenance model (Section 3). We then discuss new ideas to extend our provenance model (Section 4). Finally, we summarize and give an outlook (Section 5).

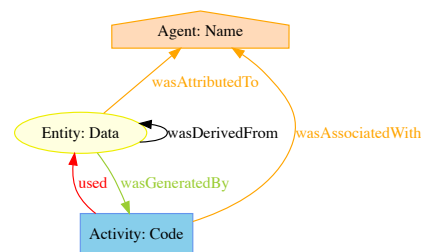*Contributions.* Our main contributions are as follows.

Figure 1: PROV core concepts and relations, based on [13].

- We demonstrate the applicability of provenance modeling for the data-intensive science field of Lattice QCD.
- We show how the W3C PROV model for Lattice QCD workflows can be used to answer common provenance-related research questions such as **Q1** to **Q5** (Section 3).
- We articulate the need for model extensions to support researchers who wish to employ provenance to make their research workflows more transparent and to collect and utilize more detailed provenance information.

2 BACKGROUND AND RELATED WORK

Provenance information encompasses metadata on entities, activities, and agents involved in a production process. In scientific workflows this provenance information is usually logged [24]. In the case of Lattice QCD, we are dealing with a complex workflow that generates and analyzes large amounts of data using supercomputers. For answering our provenance-related questions, which we will present in detail in Section 3, we employ the W3C PROV model to describe the corresponding provenance information. This model was developed ten years ago and defines a "core data model for provenance for building representations of the entities, people and processes involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness" [13]. The model distinguishes three core concepts: *entities*, *activities*, and *agents*. Entities are data or artifacts and can be *derived from* other entities. Activities can *generate* or *use* entities. Agents can perform or control activities or produce entities. Workflow provenance can be described by graphs whose nodes are entities, activities, or agents. Their relations are described by different edge types of the graph (Figure 1).

At the core of the W3C PROV standard lies the PROV data model (PROV-DM [32]). It defines concepts for expressing and exchanging provenance information and is realized by a family of related specifications, e.g., for provenance aimed at human consumption (PROV-N), a PROV ontology (PROV-O), and an XML schema (PROV-XML), see [13] for details. A direct precursor of the W3C PROV model was the Open Provenance Model (OPM [31]).

In the context of scientific workflows, provenance is a record of the derivation of a set of results. There are two different forms of provenance [44]: The W3C PROV model as well as OPM capture *retrospective provenance*, i.e., information about past workflow executions and data derivations. In contrast, *prospective provenance* captures the structure of a workflow and can be understood as a recipe for future workflow executions. Workflow graphs can also

serve as a high-level "summary" of what has happened in the past, i.e., despite their prospective nature, such graphs can also be used to describe past workflow executions at a higher, conceptual level. Since the W3C PROV model (and similarly OPM) were meant to provide a minimal model for retrospective provenance, questions that involve prospective provenance or hybrid provenance elements cannot be directly answered in this model [25, 38].

To address these limitations, several extensions to retrospective provenance standards have been developed over the years: ProvONE [6] extends the W3C PROV model by adding a workflow (i.e., prospective provenance) layer that is then linked to the retrospective layer to support hybrid queries combining both provenance types in scientific workflow applications. ProvONE "aims to provide the fundamental information needed to understand and analyze scientific workflow-based computational experiments" [6]. Similarly, its precursor D-PROV [30], and the related *Open Provenance Model for Workflows* OPMW [12], provide scientists with a vocabulary and relational structure for answering hybrid provenance questions. The Wf4Ever research object model includes a vocabulary for workflow execution provenance [39]. Numerous tools have been developed that capture computational provenance, see, e.g., [36] for a survey on provenance tools for scripts, and [5] for a tool for capturing workflow provenance. The idea of combining retrospective with prospective provenance to support hybrid queries has also been employed before, see, e.g., [43] or [35]. The latter reference combines the annotation-based prospective provenance modeling tool YesWorkflow [28] with a tool for capturing fine-grained retrospective provenance from Python scripts [34].

3 PROVENANCE FOR LATTICE QCD

In this section, we develop a provenance model for Lattice QCD using the vocabulary known in that community. A W3C PROV representation of our model is shown in Figure 3 below.

3.1 Workflow for Lattice QCD Calculations

As outlined in the introduction, a generic Lattice QCD workflow consists of three main parts. In the first part (*generation*), so-called *gauge field configurations* are generated by means of *Monte Carlo* techniques, usually employing the HMC algorithm [10]. At the end of this process a certain number of configurations becomes available, where each configuration consists of a fixed number of complex numbers. These configurations are stored on disk for subsequent analysis. In the second part (*measurement*), *correlation functions* and other derived data (collectively called *measurement data*) are computed from the configurations. These correlation functions contain information about physical observables. In the third part (*analysis*), the observables are computed from the correlation functions, which includes averaging over configurations, extrapolating to certain limits, and other activities. The outcome of these calculations can then be confronted with experimental results.

The first two parts of the workflow are illustrated in Figure 2 using gray frames. Here, we do not consider the third part for two reasons: the first two parts are much more compute-intensive, and the third part depends on the observable of interest and is thus much less generic than the first two. We leave the construction of a provenance model for the analysis part to future work.

The input for the generation part is given by simulation parameters which control both the HMC algorithm (*algorithmic parameters*) and the details of the physics (*physical parameters*). Note that the physical parameters also enter the measurement part. Typically, they are available in the *configuration metadata* that are produced during the generation part. Additional parameters (*measurement parameters*) enter the measurement part.
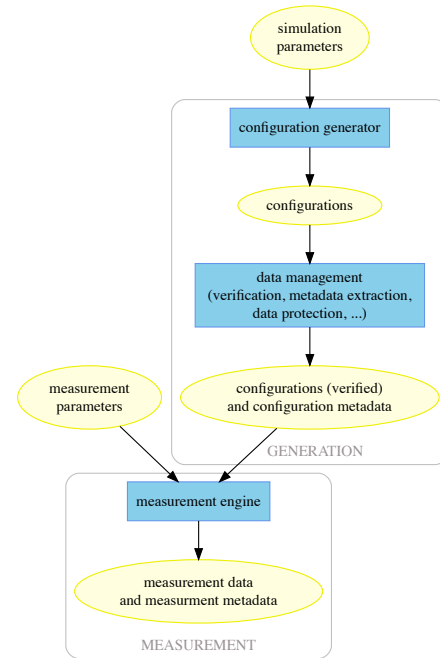
Figure 2: Representation of the first two parts of a generic Lattice QCD workflow.

Let us briefly discuss the activity labeled "data management" in Figure 2. Typical Lattice QCD calculations generate a huge amount of data [4], and thus data management has become of critical importance [23]. Collecting and processing metadata is obviously a mandatory task. Furthermore, to ensure correctness, data verification is necessary, which utilizes the metadata. Also, since generating the configurations is computationally very expensive, it is important to back up the data to prevent data loss. To comply with scientific standards, often data need to be archived for a certain period of time. Note that data management is equally important for both configurations and measurement data, i.e., the measurement part of Figure 2 also contains data management even though it is not shown explicitly.

3.2 Including Provenance

To better support researchers in their computational science work, questions like the following should be answerable by the data management system and cyberinfrastructure.

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 questions are just a few examples of provenance-related questions that computational scientists often need to answer and are similar to those raised in [40]. They are always *entity-*, *activity-* or *agent-*focused and are used, e.g., for defect detection, quality assurance, process validation, monitoring, statistical analysis, developer evaluation or information gathering.[1] Because of the broad range of questions, different provenance elements are required to answer the questions stated above. While questions **Q1** and **Q2** pertain mainly to the data level, questions **Q3** to **Q5** relate more closely to the underlying workflow that generated the data.

We follow the W3C PROV standard presented in [13] to define a provenance model for the specific Lattice QCD workflow implemented in our research group, see Figure 3. The four *activities* shown in the blue code boxes read the two sets of input parameters, manage the data, and generate the final HDF5 files. The seven *entities* shown in the yellow boxes represent input parameters, data, and metadata. The model is completed by three *agents* (orange boxes) who execute the activities and are responsible for the input parameters. In our case, the agents Alice and Bob define the input parameters and execute the first two activities openQCD and metadata extraction and verification, respectively. In the workflow, verified data and metadata are generated. Using these and the measurement parameters, Charly then executes the activities in the measurement part and generates the actual HDF5 files.

There are two main differences between Figures 2 and 3. First, the data management activity in Figure 2 also includes backup and archiving, which we omitted in Figure 3 in the interest of simplicity. Second, while the measurement sub-workflow in Figure 2 consists of a single activity, the corresponding sub-workflow in Figure 3 includes a second activity. This is due to the fact that in the actual implementation of the measurement engine, another intermediate step is required, i.e., the generation of suitable XML input files. In summary, the provenance graph in Figure 3 can be seen as a refinement of the more abstract workflow version in Figure 2.

Upon closer inspection, we notice that the PROV graph does not contain *w*-edges (*wasDerivedFrom*) of the form $(D_1) \overset{w}{\leftsquigarrow} (D_2)$, i.e., a data entity D_2 was derived from another entity D_1. Instead, Figure 3 contains chains of *u*-edges (*used*) and *g*-edges (*wasGeneratedBy*) of the form

$$(D_1) \overset{u}{\leftsquigarrow} [A] \overset{g}{\leftsquigarrow} (D_2),$$

i.e., a process (activity) A used data entity D_1 and generated data entity D_2. In many applications, including ours, the *used* and *wasGeneratedBy* relations are important to explicitly model the flow of data entities in and out of processing steps (activities), thereby supporting powerful provenance analysis queries. Although the standard does not assume that a *used-wasGeneratedBy* chain always implies a *wasDerivedFrom* relation, this is often the case in practice and in our Lattice QCD provenance model as well. For visual clarity, we omitted these *wasDerivedFrom* edges in Figure 3. An implementation of the model could construct these edges on demand using a
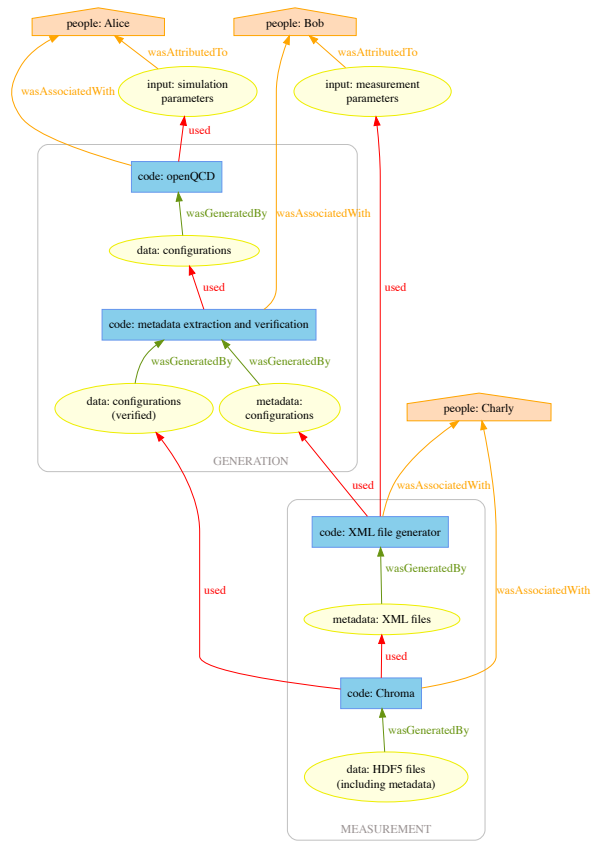
---

[1]For other computational science provenance questions see, e.g., [29] and [35].

Figure 3: Lattice QCD workflow as a W3C PROV model.

custom derivation rule; see [8] and [33] for further details on the interplay of these different relationships.

The provenance graph in Figure 3 contains specific instance-level information such as the names of the agents Alice, Bob, and Charly and abstract identifiers on the schema level such as XML file. In many cases adding such instance-level information satisfies the needs of the Lattice QCD community. For example, the calculations are usually carried out by the same people so that the individuals/concrete agents can be viewed as "part of the system."

Figure 3 lists the names of the individuals who regularly act as agents for a specific research group and time period. A more generic setup would indicate an agent of type Person. Using the names of individuals, question **Q3** can be answered directly from the provenance graph. In contrast, question **Q4** refers to activities and agents at the workflow level and requires prospective provenance. **Q5** is a similar question posed at the entity level.

Like **Q3**, question **Q1** is a retrospective provenance question. For example, we may realize that the output data are incorrect. In this case, the data derivation chains must be traced from the results back to the sources (*upstream* propagation). As another example, we may find an error in an activity or entity, such as the silent data corruption in the configurations mentioned above. Then the

erroneous activity or entity needs to be fixed, and subsequent entities need to be recomputed or corrected (*downstream* propagation). In both examples we have to perform dependency tracing along derivation chains in the provenance graph.

Question **Q2** can be interpreted in different ways. If we are solely interested in the dependency structure at the conceptual level (Figure 3), question **Q2** requires prospective provenance only. Alternatively, if we are interested in the effect of a parameter change on a dataset in a previous workflow run, we need retrospective provenance.

To summarize, questions **Q1** and **Q3** require retrospective provenance, questions **Q4** and **Q5** correspond to prospective provenance queries, and question **Q2** combines both kinds of provenance.

4 TOWARDS LAYERED PROVENANCE

Every execution of the workflow by ALICE, BOB, and CHARLY results in a provenance graph similar to the one shown in Figure 3. Since the overall provenance model structure remains the same in all cases, the provenance graph depicted in Figure 3 is really a *provenance template* graph, i.e., each *workflow run* (execution) 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). In turn, the provenance template in Figure 3 can be seen as a specialization of the workflow graph in Figure 2 that describes the general form of Lattice QCD workflows used by the community. For example, the generic steps *configuration generator* and *measurement engine* in the workflow are specialized to *code: openQCD* [26] and *code: Chroma* [11], respectively, which are the specific tools used by the physicists in our research group.

4.1 Provenance Templates vs Instances

To address the practical needs of our research scientists, while at the same time employing a standard model to facilitate data exchange and transparency, we propose to extend the W3C PROV model to include both instance-level provenance graphs and—linked to these—a template-level provenance graph. The relationship between template and instance graphs is a very natural one, as the latter can be viewed as isomorphic copies of the former, where schema-level elements have been replaced by object identifiers. In this way, the template graph can serve as an overview or a summary of the many instance graphs.

We further propose to add a workflow layer to this extension. In the resulting multi-layer provenance model (Figure 4), a community-wide *workflow graph* can be specialized to a *provenance template* (for individual research groups), which in turn will be *instantiated* whenever workflow runs are executed. These instances then include concrete values of all input parameters, the names and time stamps of the data files containing the configurations, the version numbers or git hashes of the codes, compiler versions and flags, the names of persons who executed the compilation, details of the machines running the calculations, etc.

As our initial analysis of the provenance-related queries **Q1** to **Q5** from Section 3.2 has shown, to answer all such questions requires a combination of retrospective provenance elements (as provided by the W3C PROV standard) and prospective elements (as given
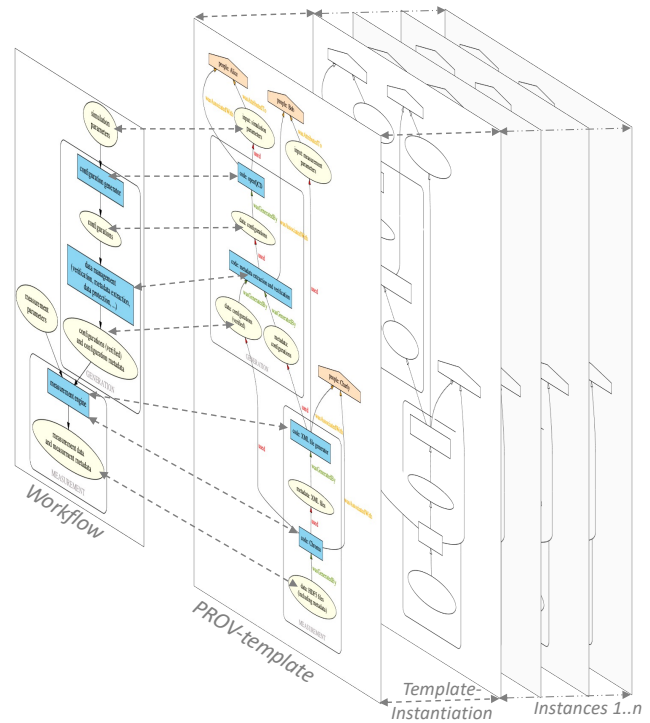
Figure 4: Layers of the proposed hybrid provenance model: Elements of a conceptual workflow (left) naturally map to elements in a W3C PROV template model (middle). Each workflow execution creates PROV-compatible retrospective provenance graphs (right), i.e., instances $1, \ldots, n$ **of the provenance template in the middle.**

by a workflow or a provenance template graph). After extending our model in this way, it should satisfy the following desiderata:

(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 proposed multi-layer model (Figure 4) can be implemented in different ways. We hope to bring together both communities, i.e., our colleagues from the Lattice QCD community and the provenance research community, to co-develop suitable W3C PROV extensions. Our current plan is to explore and evaluate existing standards, in particular W3C PROV and RDF [7]. The latter would also allow us to embed multi-layered provenance models into a more general *knowledge graph*/LOD (Linked Open Data) framework, leveraging again existing standards, tools, and namespaces. Using these, the Lattice QCD community can determine how far they want to go with modeling and formalizing the semantics for the different types, e.g., to create controlled vocabularies and/or formal ontologies in OWL, agreed upon by the community. We reach out to the provenance community to get a head-start in our

model-building efforts. Since we are not the first ones to identify the need for *hybrid provenance* models (see, e.g., [5, 6, 12, 39, 44] among many others), we hope to build on existing efforts rather than reinventing the wheel.

4.2 Implementation & Evaluation Plan

In order to evaluate the efficacy and practicality of the proposed model, we plan to implement a prototype for the Lattice QCD community. Since a considerable amount of provenance information is already captured by existing QCD workflows in log files, we will first develop a Python-based provenance harvesting tool. The harvested information then needs to be mapped to a suitable provenance store (e.g., a relational or graph database) that implements our model. Additional provenance information can be found in the attributes and dataspace objects of the HDF5 datasets and in the file and folder names. As described in [29], this provenance information will then be extracted based on the applicable conventions. Finally, provenance information that is required in our model, but not (yet) available through harvesting, will have to be recorded through other means, e.g., using a light-weight provenance recorder, through instrumentation of the code, or by writing additional information to log files.

Our development efforts will be informed by a survey on collecting and managing provenance from scripts [36] and practical experience with tools that capture and integrate prospective and retrospective provenance information [28, 34, 35, 43]. For exporting interoperable provenance in W3C PROV-compliant form we will use the Python prov library [18]. We will also explore the option to cast our extended provenance model as a W3C standards-compliant knowledge graph. This would allow us to leverage additional standards and tools based on popular triple stores and graph query languages such as SPARQL [16] or Cypher [19].

5 SUMMARY AND FUTURE WORK

Lattice QCD is an important field of particle physics that generates and analyzes huge amounts of data. We have proposed a provenance model for Lattice QCD workflows based on the W3C PROV standard. Starting from a generic workflow (Figure 2) we have derived a PROV template (Figure 3) that allows us to answer a number of typical provenance-related questions. To address a more complete set of provenance questions (see, e.g., **Q1** to **Q5**) we have proposed a layered model that provides the necessary information. It consists of a workflow layer, a provenance template layer, and an instance layer (Figure 4).

In future work, we plan to apply our provenance model to the third part (*analysis*) of the overall Lattice QCD workflow. As indicated earlier, this part is less generic and more tailored to the specific physics programme of a particular collaboration.

We have argued in Section 3.1 that data management is important in all parts of the Lattice QCD workflow. Therefore, it is desirable to build upon the Lattice QCD community efforts in ILDG and PUNCH4NFDI to define more comprehensive standards for data management, which also need to include provenance metadata.

We plan to engage the Lattice QCD and provenance communities to refine our initial model proposal and, subsequently, to implement a prototype that will allow us to evaluate its efficacy and practicality.

REFERENCES

[1] ACM. 2020. Artifact Review and Badging - Current. https://www.acm.org/publications/policies/artifact-review-and-badging-current

[2] Tanja Auge, Moritz Hanzig, and Andreas Heuer. 2022. ProSA Pipeline: Provenance Conquers the Chase. In *ADBIS (Short Papers) (CCIS, Vol. 1652)*. Springer, 89–98.

[3] Tanja Auge and Andreas Heuer. 2019. ProSA - Using the CHASE for Provenance Management. In *ADBIS (LNCS, Vol. 11695)*. Springer, 357–372.

[4] Gunnar Bali et al. 2022. Lattice gauge ensembles and data management. *Proceedings of Science (PoS)* LATTICE (2022), 203. arXiv:2212.10138

[5] Flavio Costa et al. 2013. Capturing and querying workflow runtime provenance with PROV: a practical approach. In *EDBT/ICDT Workshops*. ACM, 282–289.

[6] Víctor Cuevas-Vicenttín et al. 2016. ProvONE: A PROV Extension Data Model for Scientific Workflow Provenance. http://jenkins-1.dataone.org/jenkins/view/Documentation%20Projects/job/ProvONE-Documentation-trunk/ws/provenance/ProvONE/v1/provone.html.

[7] Richard Cyganiak, David Wood, and Markus Lanthaler. 2014. RDF 1.1 Concepts and Abstract Syntax. https://www.w3.org/TR/rdf11-concepts/

[8] Saumen C. Dey et al. 2012. Datalog as a Lingua Franca for Provenance Querying and Reasoning. In *TaPP*. USENIX Association.

[9] DLR. 2022. *Integrating Provenance-Awareness into the Space Debris Processing System BACARDI.* https://github.com/DLR-SC/bacardi-prov-model

[10] Simon Duane et al. 1987. Hybrid Monte Carlo. *Phys. Lett. B* 195 (1987), 216–222.

[11] Robert G. Edwards and Bálint Joó. 2005. The Chroma software system for lattice QCD. *Nucl. Phys. B Proc. Suppl.* 140 (2005), 832. arXiv:hep-lat/0409003

[12] Daniel Garijo, Yolanda Gil, and Oscar Corcho. 2014. OPMW-PROV: The Open Provenance Model for Workflows. https://www.opmw.org

[13] Yolanda Gil et al. 2013. PROV Model Primer. https://www.w3.org/TR/prov-primer/

[14] GoFair. 2023. FAIR Principles. https://www.go-fair.org

[15] Paul Groth and Luc Moreau. 2013. PROV-Overview. https://www.w3.org/TR/2013/NOTE-prov-overview-20130430/

[16] Steve Harris, Andy Seaborne, and Eric Prud'hommeaux. 2013. SPARQL 1.1 Query Language. https://www.w3.org/TR/sparql11-query/

[17] Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. 2017. A survey on provenance: What for? What form? What from? *VLDB J.* 26, 6 (2017), 881–906.

[18] Trung Dong Huynh. 2020. prov 2.0.0. https://pypi.org/project/prov/

[19] Neo4j Inc. 2021. What is openCypher? https://opencypher.org

[20] A.C. Irving et al. 2004. Progress in building an International Lattice Data Grid. *Nucl. Phys. B Proc. Suppl.* 129 (2004), 159–163. arXiv:hep-lat/0309029

[21] Michael A. C. Johnson et al. 2021. Astronomical Pipeline Provenance: A Use Case Evaluation. In *TaPP*. USENIX Association.

[22] Bálint Joó and C. M. Maynard. 2006. Progress in building the International Lattice Data Grid. *PoS* LAT2006 (2006), 028. https://doi.org/10.22323/1.032.0028

[23] Frithjof Karsch, Hubert Simma, and Tomoteru Yoshie. 2022. The International Lattice Data Grid - towards FAIR Data. In *39th Intl. Symposium on Lattice Field Theory*. arXiv:2212.08392 [hep-lat]

[24] Meike Klettke and Uta Störl. 2022. Four Generations in Data Engineering for Data Science. *Datenbank-Spektrum* 22, 1 (2022), 59–66.

[25] Chunhyeok Lim et al. 2010. Prospective and Retrospective Provenance Collection in Scientific Workflow Environments. In *SCC*. IEEE Computer Society, 449–456.

[26] Martin Lüscher and Stefan Schaefer. 2013. Lattice QCD with open boundary conditions and twisted-mass reweighting. *Comput. Phys. Commun.* 184 (2013), 519–528. https://doi.org/10.1016/j.cpc.2012.10.003 arXiv:1206.2809 [hep-lat]

[27] C. M. Maynard and D. Pleiter. 2005. QCDml: First milestone for building an International Lattice Data Grid. *Nucl. Phys. B Proc. Suppl.* 140 (2005), 213–221. https://doi.org/10.1016/j.nuclphysbps.2004.11.116 arXiv:hep-lat/0409055

[28] Timothy McPhillips et al. 2015. YesWorkflow: A User-Oriented, Language-Independent Tool for Recovering Workflow Information from Scripts. *Intl. Journal of Digital Curation* 10, 1 (2015), 298–313. https://doi.org/10.2218/ijdc.v10i1.370

[29] Timothy M. McPhillips et al. 2015. Retrospective Provenance Without a Runtime Provenance Recorder. In *TaPP*. USENIX Association.

[30] Paolo Missier et al. 2013. D-PROV: Extending the PROV Provenance Model with Workflow Structure. In *TaPP*. USENIX Association.

[31] Luc Moreau et al. 2011. The Open Provenance Model core specification (v1.1). *Future Gener. Comput. Syst.* 27, 6 (2011), 743–756.

[32] Luc Moreau et al. 2013. PROV-DM: The PROV Data Model. https://www.w3.org/TR/2013/REC-prov-dm-20130430/

[33] Luc Moreau et al. 2015. The rationale of PROV. *J. Web Semant.* 35 (2015), 235–257.

[34] Leonardo Murta et al. 2014. noWorkflow: Capturing and Analyzing Provenance of Scripts. In *IPAW (LNCS, Vol. 8628)*. Springer, 71–83.

[35] João Felipe Pimentel et al. 2016. Yin & Yang: Demonstrating Complementary Provenance from noWorkflow & YesWorkflow. In *IPAW (LNCS, Vol. 9672)*. Springer, 161–165.

[36] João Felipe Pimentel et al. 2019. A Survey on Collecting, Managing, and Analyzing Provenance from Scripts. *ACM Comput. Surv.* 52, 3 (2019), 47:1–47:38.

[37] PUNCH4NFDI. 2023. A consortium in the NFDI (National Research Data Infrastructure). https://www.punch4nfdi.de

[38] Yogesh Simmhan, Paul Groth, and Luc Moreau. 2011. Special Section: The third provenance challenge on using the open provenance model for interoperability. *Future Gener. Comput. Syst.* 27, 6 (2011), 737–742.

[39] Stian Soiland-Reyes et al. 2013. Wf4ever Research Object Model. http://wf4ever.github.io/ro/

[40] Martin Stoffers et al. 2022. Integrating Provenance-Awareness into the Space Debris Processing System BACARDI. In *2022 IEEE Aerospace Conference (AERO)*. 1–12. https://doi.org/10.1109/AERO53065.2022.9843783

[41] The ILDG Metadata Working Group. 2004–2013. Specification of the QCDml Standard. https://www2.ccs.tsukuba.ac.jp/ILDG/

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

[43] Qian Zhang et al. 2017. Revealing the detailed lineage of script outputs using hybrid provenance. *Intl. Journal of Digital Curation (IJDC)* 12, 2 (2017), 390–408.

[44] Yong Zhao, Michael Wilde, and Ian T. Foster. 2006. Applying the Virtual Data Provenance Model. In *IPAW (LNCS, Vol. 4145)*. Springer, 148–161.