Suh and Lee *Hum. Cent. Comput. Inf. Sci.* (2018) 8:27
https://doi.org/10.1186/s13673-018-0150-9

Human-centric Computing and Information Sciences

RESEARCH Open Access

A survey of simulation provenance systems: modeling, capturing, querying, visualization, and advanced utilization

Young-Kyoon Suh[1] and Ki Yong Lee[2*]

*Correspondence: kiyonglee@sookmyung.ac.kr
[2] Department of Computer Science, Sookmyung Women's University, 100 Cheongpa-ro 47-gil, Yongsan-gu, Seoul 04310, South Korea
Full list of author information is available at the end of the article

Abstract

Research and education through computer simulation has been actively conducted in various scientific and engineering fields including computational science engineering. Accordingly, there have been a lot of attentions paid to actively utilize provenance information regarding such computer simulations, particularly conducted on high-performance computing and storage resources. In this manuscript we provide a comprehensive survey of a wide range of existing systems to utilize provenance data produced by simulation. Specifically, we (1) categorize extant provenance research articles into several major themes along with well-motivated criteria, (2) grasp and compare primary functions/features of the existing systems in each category, and (3) then ultimately propose new research directions that have never been pioneered before. In particular, we present a taxonomy of scientific platforms regarding provenance support and holistically tabulate the major functionalities and supporting levels of the studied systems. Finally, we conclude this article with a summary of our contributions.

Keywords: Provenance, Simulation, Systems, Platforms, Classification, Taxonomy

Introduction

In the past years, *big data* has revolutionized the way much of our data is collected, shared, and used. As a new kind of big data, a large volume of data is being actively generated in the field of computer simulation. Computer simulation has been increasingly used in many computational science and engineering disciplines, thanks to remarkably advanced IT infrastructure. As the users of computer simulation increase, online simulation platforms have appeared and lowered entry wall for those who are not familiar with command line interface (CLI) by providing an easy-to-use interface and conducting automated simulations. These platforms typically utilize high performance computing clusters and distributed/parallel computing infrastructure to support numerous users while performing computationally-heavy simulations [1–4].

For a completed simulation, a platform can keep the execution record of that simulation for future reference. For instance, suppose that at a specific time, a user chooses a simulation program (or tool), gives some inputs on the program, and requests the execution as provided. Once the platform finishes the simulation execution, it can store

information about (i) who asked the simulation, (ii) what simulation program and what version of the program was used, (iii) how long it took, (iv) what input values were provided, (v) what output values or files were produced, (vi) whether it succeeded or failed, (vii) what computing nodes were leveraged, and so forth. If the platform can utilize this sort of information about completed simulations well, users can view past records about previously-executed simulations at any time (as long as the platform can keep this vast amount of data). This source of information (about simulation) is typically called *provenance* [5–8].

Of course, there has been a rich body of existing research with a focus on the provenance utilization about simulation result, experimental and observational data, and scholarly articles. Many of the body aims at benefiting from provided provenance along with scientific data to grasp the validity of the data or reproducing the data. To this end, many provenance-aware simulation service systems have been realized and used. In this manuscript, we call a system that provides services utilizing simulation provenance *a simulation provenance service system* or *a provenance-driven service system*.

Unfortunately, there have been few research articles to discuss how simulation provenance service systems differ from one another. The most relevant survey would be the work of Herschel et al. [8]. Her survey presents a classification of a variety of applications on provenance whose concept is widely applied in "general" contexts. But her study lacks in provenance service systems on high-performance computing (HPC) simulations from computational science and engineering disciplines. Moreover, her survey does not cover a variety of matters emerging in terms of simulation platform administration. Finally, the coverage in her work is so broad that it is hard to grasp what specific provenance service systems are available for simulation users. Hence, it is not sufficient for her work to satisfy a need of better understanding the characteristics (weaknesses and strengths) of such provenance service systems so that computational scientists and engineers can better choose a system enough to fulfill their purpose, and developers can further improve existing systems with enhanced services (which we think are potentially promising).

To satisfy the need, this manuscript conducts a comprehensive survey of a wide range of provenance management systems proposed to support scientific simulations and workflows on HPC resources. More specifically, we divide these simulation provenance systems into several categories and then examine major features of a system in each category. Based on our understanding, we propose potentially promising ideas on which these systems can provide more advanced services based. The proposed ideas are expected to get the simulation service systems to function more efficiently as well as to assist a user to better use the platform via the enhanced services.

Especially, we also propose a general taxonomy of provenance-driven scientific platforms (prior to the discussion of the proposed ideas). This taxonomy is driven by well-motivated criteria in terms of functionalities. For instance, scientific platforms utilizing provenance can be grouped by whether (i) online simulations are supported, (ii) provenance can be collected for (successful or failed) simulation jobs, (iii) standardization of collected provenance is considered, (iv) reusing simulation results is supported, and (v) advanced provenance utilization such as mining is considered. We discuss the taxonomy in greater detail later in the article.

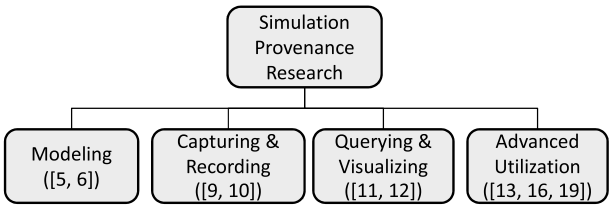

Fig. 1 A classification of major topics of provenance research. Existing provenance research can be divided into four major sectors: modeling, capturing and recording, querying and visualizing, and advanced utilization

Our contributions can be summarized as follows:

- We conduct an in-depth survey of simulation provenance systems on scientific simulations and workflows.
- We categorize these systems into several relevant areas.
- We investigate the goals and characteristics of these systems in each category.
- We provide a taxonomy of the systems along with a suite of well-motivated criteria.
- We perform a comparative analysis of the systems based on major features regarding aimed problems.
- We propose future research directions and ideas, suggesting advanced provenance services for better user convenience.

To the best of our knowledge, this is the *first* work to (1) present a reasonable categorization and a well-organized taxonomy for extant provenance service systems, (2) conduct a comparative analysis among these systems, and (3) propose future research directions for further engineering the systems.

The rest of this manuscript is organized as follows. The following section categorizes major research topics regarding provenance. After that, we investigate several modeling techniques on provenance data. In turn, we review what to capture from scientific experiments and simulations and how to record the captured information into provenance. Next, we discuss querying and visualizing provenance data and further examine advanced usage of provenance data related to data mining. We then proceed with a taxonomy of scientific platforms with respect to provenance support. Subsequently, we present future research directions and ideas. Finally, we conclude this survey by summarizing our discussions.

Categorization of simulation provenance research

So far there have been a number of research articles discussing provenance management in scientific applications and platforms. In this section we provide a categorization of several major research themes regarding provenance.

As depicted in Fig. 1, there are four categories of provenance research we identified: modeling, capturing and recording, querying and visualizing, and advanced utilization. We now elaborate on each category in the following.

Simulation provenance modeling

This category aims at developing a suitable model for provenance to support interoperability via standardization or to help querying and analyzing provenance more efficiently. There have been so far several models that were proposed [5, 6]. The goal of these models is to provide flexibility such that the model can be applied to a variety of applications. A provenance record complying with such a model is typically represented in the form of a relation (table) or graph. It is needed to continuously revise and improve the model as it is closely related to service performance at the time of developing a provenance service.

Simulation provenance capturing and recording

This category represents capturing and recording in an automated way provenance information from a scientific simulation program during its execution. Indeed, it could be a challenging task to extract provenance data without human intervention while the program in execution on a simulation platform. Also, it is obvious that where to store provenance records determines later retrieval and query performance. Recently, there has been a clustered study of automatically collecting provenance without altering such a program while charging little overhead on the platform [9, 10]. Currently provenance data is stored in a relational database management system (RDBMS), semi-structured, or unstructured database. A file system is also used as a provenance repository with limited query functionality.

Simulation provenance querying and visualizing

This category concerns mechanisms of accessing and visualizing provenance information more easily and more efficiently. In general, users can browse via a web interface and view via a visualization tool provenance information represented in the form of graph. In particular, VisTrails [11, 12] is a specialized platform for visualizing provenance information, and it allows users to query provenance data via QBE (Query-By-Example). Typically, SQL (Structured Query Language) is adopted for querying provenance data stored in an RDBMS. XQuery (XML Query) and SPARQL (Simple Protocol and Resource Description Framework Query Language) are also used as query language for provenance data in the format of XML (eXtensible Markup Language) and RDF (Resource Description Framework), respectively.

Advanced utilization of simulation provenance

This category addresses mining useful information or supplying informative service from provenance data that is typically useful to users or system administrators. Note that most of existing provenance service systems perform simple retrieval or allows plain browsing only. It would be interesting to uncover hidden patterns or draw new insights by applying data mining techniques on provenance data. So far, a rich body of existing literature has focused on (i) exploring a workflow pattern that frequently appears [13–15], (ii) applying provenance data to scheduling or optimizing simulations and workflows that are in execution [16–18], and (iii) estimating when to complete a specified workflow (or simulation) [19, 20]. Recently, there has been a growing need to apply a

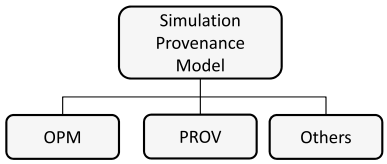

Fig. 2 A classification of provenance representation. Several representative models have been proposed. The PROV model has been widely adopted as almost a de-facto standard

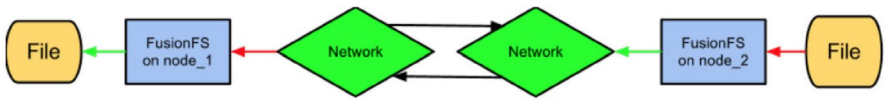

Fig. 3 An example of OPM used in a prior work [28]

variety of data mining techniques to develop frequent pattern mining and classification to provenance data.

We now begin our detailed survey by reviewing provenance modeling in the following section.

Simulation provenance modeling

In this section we discuss how provenance information has been structured, particularly for *interoperability*.

Figure 2 provides a classification of how to represent provenance data. As depicted in the figure, broadly speaking there have been around two major models that were proposed and widely used in scientific domains, in addition to several other models.

Open provenance model (OPM) [6] is one of the oldest data models for representing provenance. The goals of OPM can be summarized in the following. First, it aims at exchanging provenance across systems via a standardized model. Second, it supports provenance sharing and developing some tools based on the model. Third, it represents various types of provenance. Lastly, it provides a function to infer on provenance. A provenance record stored along with OPM is expressed as a directed graph [21]. OPM has been adopted in many different sectors—Web [5, 22], hydrologic science [23], network analysis [24], medical image [25], e-Science [26], computing systems [27, 28], and so forth—that demands provenance management. Especially, it became the predecessor of the W3C (World Wide Web Consortium) provenance (PROV) [5]. As illustrated in Fig. 3, OPM was used to represent provenance on network transmission when a simulation program was executed in the prior work [28].

PROV is a W3C standard model to represent, query, and exchange provenance data on the Web. It was a de-facto standard model established by the World Wide Web (WWW) Provenance Working Group to exchange provenance information in heterogeneous environments such as the Web. In 2013, PROV became standardized by W3C.

Among the PROV specifications, the PROV data model (PROV-DM) [29] represents provenance information with the three major *types*, that is, *entity*, *activity*, *agent*, and their *relationships*, that is, *generation*, *usage*, *communication*, *derivation*, *attribution*,

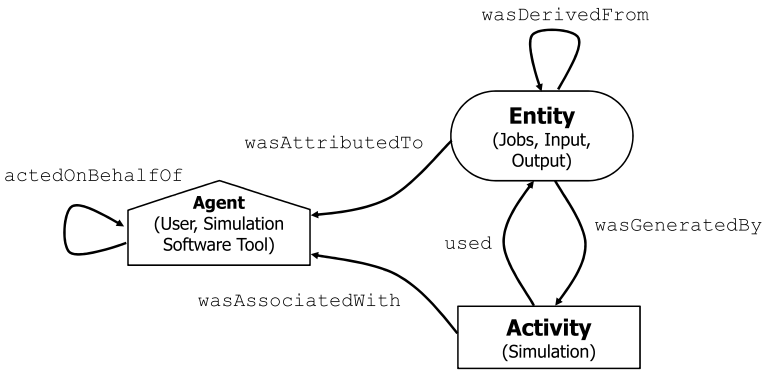

Fig. 4 An example of PROV-DM used in Suh's work [31]

association, and *delegation*. The types are linked along with their specific relations, expressed by arrows.

Thanks to the standardization of PROV, many simulation platforms are now considering its adoption for storing and managing their provenance. One concrete example is Pignotti et al. work [30]. They propose a novel approach to the representation and querying of agent-based simulation provenance. Their work uses PROV-DM to represent simulation provenance. Specifically, in their model an agent represents simulation source code, data, input/output parameters, library version, or compiler. An agent can be a user, an operating system, a certain hardware component, and a software tool. An activity corresponds to a specific action such as design, data collection, adjustment, and verification. A relation among these entities can be either *wasGeneratedBy*, *used*, *wasAttributedTo*, *wasInformedBy*, or *wasDerivedFrom*.

Another real example of PROV application comes from Suh's work [31]. The authors adopted PROV to collect and standardize provenance on executed simulations as shown in Fig. 4. More specifically, for a specified simulation provenance record Fig. 5 exhibits its PROV representation realized in the form of JSON (JavaScript Object Notation).

That said, it appears that most simulation platforms are still hesitant to represent their simulation provenance along with PROV. The reason boils down to some reasonable statements that (i) PROV may not be mature enough, (ii) there are some compatibility issues with legacy applications, or (iii) it is too verbose to apply to the platforms. More efforts are needed to draw more popularity on PROV.

Lastly, there are several other models used to represent provenance data in specific applications. For example, some researchers [32] proposed a provenance data model for scientific workflow, called ZOOM. The aim of using ZOOM is to support a general model for being capable of querying from various perspectives provenance data emerging from a variety of scientific research. In ZOOM, provenance data for scientific workflow are stored in an RDBMS, which enables a user to look inside the provenance data with SQL.

CRM^{1}_{dig} [33] is a model proposed for representing and querying provenance in e-Science. In particular, this model extends an ontology for CIDOC[2] CRM [34, 35], a

[1] Conceptual Reference Model.

[2] ICOM's International Committee for Documentation; ICOM is the abbreviation of International Council of Museums.

```
// Element definitions
"prov:agent": {"simulation SW":
{"prov:label": "2D_Comp_P"}},
"prov:entity":
 { "prov:Input": {"fileName":"NASAsc20714(2).msh"},
   "prov:job1":{"677eb1d5-14c8-437a-a270acb7aa8885c5"},
   "prov:job2":{"585a7cd7-0919-4824-af8efdf024b2d174"},
   "prov:Output": {"/EDISON/LDAP/zacwhee/
    394197c0-1134-41df-9a6b-2b49460aaec1/result"}},
"prov:activity": {"simulation ID":
 { "prov:label":
        "394197c0-1134-41df-9a6b2b49460aaec1"},
// Relation definitions
 "prov:actedOnBehalfOf": {
  "zacwhee": {"prov:delegate":"2D_Comp_P",
  "prov:responsible": "zacwhee"}},
 "prov:wasAssociatedWith": {"2D_Comp":
  {"prov:activity":"2D_Comp_P",
  "prov:agent":"zacwhee"}},
  "prov:wasAttributedTo":
  {"677eb1d5-14c8-437a-a270-acb7aa8885c5 ":
  {"prov:agent":2D_Comp_P","prov:entity": "Job1"},
  {" 585a7cd7-0919-4824-af8e-fdf024b2d174":
  {"prov:agent": "2D_Comp_P","prov:entity": "Job2"},
 " NASAsc20714(2).msh":
 {" prov:agent": " zacwhee ",
  "prov:entity": "prov:Input"}
 },
"prov:wasGeneratedBy": {
"prov:activity":"394197c0-1134-41df-9a6b-2b49460aaec1",
"prov:role": {"WasGeneratedBy"},"entity": "Output",
"prov:activity":"394197c0-1134-41df-9a6b-2b49460aaec1",
"prov:role": {"WasGeneratedBy"}, "entity": "job1",
"prov:activity":"394197c0-1134-41df-9a6b-2b49460aaec1",
"prov:role": {"WasGeneratedBy"}, "entity": "job2",
"prov:activity":"394197c0-1134-41df-9a6b-2b49460aaec1"},
"prov:used": {{"prov:activity":
"394197c0-1134-41df9a6b2b49460aaec1",
"prov:entity":"prov:Input"}}
```

Fig. 5 An example of PROV-JSON used in Suh's work [31]

standard for disseminating cultural contents. The model allows for more detailed depiction on a physical environment on scientific observation process. Users can query CRM provenance data as well [36].

Capturing simulation provenance

Figure 6 categorizes several mechanisms of capturing provenance in scientific domains.

First, a *model-based* approach is the most popular technique for capturing simulation provenance. As described earlier, OPM was adopted into many different scientific domains, and now PROV, backed by a privilege of W3C standard, seems to replace the role of OPM. One example to follow the model-based approach is WS-VLAM (Workflow System on Virtual Laboratory Abstract Machine) [37], which is a workflow management system for scientific workflow in a grid environment. This work proposed two ways of capturing provenance. One way is to use OPM as described earlier. Another is

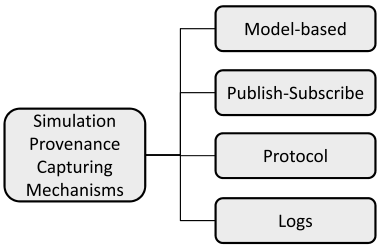

Fig. 6 Several mechanisms for capturing provenance. These four methods are typically exploited to effectively capture provenance in scientific workflows and simulations executed in different computing environments

to represent provenance data as *history-tracing* XML (or, HisT) and capture them. HisT stores provenance in a layered XML document form, in which each layer represents a task in a given workflow.

The second way of capturing simulation provenance is to leverage a *publish-subscribe* method. An example of using this approach is Tylissanakis et al. work [9], in which they proposed a system to collect and manage data provenance for multi-physics simulation workflows via notification used in Web Services Resource Framework (WSRF) [38]. The notification messages follow WS-Notification [39], which is a standard for exchanging messages among web services in the format of publish-subscribe. The authors' work exposes data and simulation models as WSRF services and if a provenance record is generated, then that record is delivered as a WS-notification message. In this way, simulation provenance in their work is collected non intrusively with an existing system. Another example of adopting the publish-subscribe model is from Simmhan et al. work [10]. Their approach proposes a general framework for capturing provenance for scientific workflows in an implementation-independent way. The proposed framework aims to minimize collection cost by exchanging notification messages among services composed of a scientific workflow. The human intervention is extremely limited in their framework, only up to the level of letting a workflow component send a provenance notification message.

Thirdly, a *protocol* to collect and record provenance data has drawn attention from the community. One example comes from Groth et al. work [40], in which they invent an implementation-neural protocol, named PReP (Provenance Recording Protocol), for capturing provenance in a grid environment. PReP, which is performed on a service-oriented architecture [41], consists of a series of steps, such as *negotiation*, *invocation*, *submission*, and *termination*. They claim that if a service or application complies with their proposed protocol, it can capture provenance in a standardized way.

Lastly, another way of capturing provenance is to utilize *logs*. Sun et al. [42] proposed a log-based approach for reservoir management workflows. In their work a workflow instance is extracted from an application's log. That captured instance is physically stored along with OPM. More specifically, their approach first finds the execution (or realization) pattern from each application's log file, then builds a workflow pattern from that pattern and finally reconstructs provenance by grasping which order of tasks were executed within that workflow instance.

In short, many different approaches were proposed for capturing simulation provenance, but nowadays PROV seems being established as a standard specification for scientific simulation provenance.

Simulation provenance querying and visualization

We now discuss how existing works approach querying stored provenance data. We also consider how they visualize the provenance data as answer for an issued query.

Querying

Kloss' work [43] proposed a framework to manage and query provenance data for scientific and engineering simulations. In their framework it is possible to issue a query on provenance data available in a provenance store, via query tools. The query is categorized into these questions: (1) what simulation cases concerned a given item, (2) what simulations corresponded to a given parameter, (3) what simulations were performed on a given simulation model, (4) what output was produced by a simulation with a given parameter, or (5) what differences were made between two simulations for the same input parameter(s).

SciProv [44] is an architecture for supporting a semantic query on provenance metadata in the context of e-Science. SciProv is in conjunction with scientific workflow management system and standardize provenance data along with OPM. In particular, SciProv takes advantage of semantic web [45, 46]; namely, it enables users to issue a semantic query on provenance data by utilizing ontology and semantic engine. In this ways it's definitely possible to extract semantic information from provenance data that are not explicitly stored. SciProv uses SPARQL [47] as a query language.

Woodman et al. [48] proposed a system to support storing the current and old version of data of workflow and services to extend a range of more useful queries. The proposed system, developed based on e-Science Central [49], supports a more variety of queries by integrating provenance data (automatically collected) into performing prior versions of workflows and services. Their query made it possible to support the following type of queries: whether or not a workflow will produce the same result (1) *if it is re-executed*, (2) *if we use a different version of the data, workflow, or service*, and (3) *if we replace the service with the older version*.

Zhao et al. work [50] aims at providing various types of queries for a variety of provenance information about links on data items linked each other. The provenance information includes not only when and who to generate but also when to update it. Also, the information can include more detailed contents such as the presence of former links, what links they were if any, and why they were removed. To represent links by which data items are connected, the authors used a "named graph." To query such a graph, the proposed work uses SPARQL.

In the same line, there was a proposed system, called TripleProv [51–53], for tracking and querying provenance over linked data on the Web. As a sort of a RDF database system, TripleProv supports deriving provenance information transparently and automatically for given queries.

One of the most recent works is from Wylot et al. [54], who proposed a system to more efficiently record, track, and query provenance on general RDF data. In their work, they

defined a provenance-enabled query as a query to derive a provenance polynomial for input data and derivation process associated with the output data for given provenance scope and workload query. They proposed a technique of representing the provenance-enabled query as an equivalent SPARQL. Therefore, users can more efficiently execute a variety of queries on triple data represented by RDF.

Unfortunately, those works are orthogonal to simulation provenance, which is the focused scope of this article. But it is worth to consider applying their querying approach to simulation provenance data, in the sense that both camps try to seek for the best approach for querying provenance records more efficiently.

Finally, PROV-AQ (Provenance Access and Query) [55] is a W3C standard specification proposed to obtain the information about provenance on the Web. The specification introduces mechanisms for accessing and querying provenance. Specifically, it elaborates on a protocol based on HTTP (Hypertext Transfer Protocol) for provenance access and on how to locate a SPARQL service endpoint. But it seems that not many follow-up works use PROV-AQ yet.

Now we discuss visualizing provenance data.

Visualization

Chen et al. proposed several visualization techniques for large-scale provenance graph [56]. Their work included interactive browsing, manipulation, and analysis functions for large-scale provenance graph. The proposed techniques were implemented using Cytoscape [57–59] and used some visualization technology based on (i) incremental loading for provenance annotation, (ii) customized (hierarchical or time-based) layout for provenance display, (iii) visual style such as color, size, and transparency, and (iv) abstract view to eliminate unnecessary nodes or cluster neighboring nodes.

Prov-Vis [60] is a well-known visualization tool for large-scale scientific workflow. In Prov-Vis users can query provenance data and receive the summary of the queried result. The summary information is used for visualization. Prov-Vis allows scientists to walk through executed workflow instances and visually check the generated output. Prov-Vis is in conjunction with two scientific workflow engines, SciCumulus [61] and Chiron [62], which can query the provenance data and pass the query results to Prov-Vis for visualization.

Jensen et al. [63] proposed a tool for visualizing the provenance captured from an existing NASA instrument ingest pipeline. In their work the provenance is collected through the Karma [10, 64] provenance system and represented as an OPM-based graph. Enabling users to easily browse and manipulate the captured graph, the tool can further assist the users to compare the provenance graphs each other and readily grasp the relationship information among provenance data on each file or process. Especially, the users can visually view the chain of representing the process of data translation.

Lastly, one system [65] was proposed to query and visualize provenance data produced from scientific workflow used in ocean observation. The authors invented (1) a specialized query processor for the ocean observation area and exploited (2) VisTrails [11, 12, 66] for visualization. The feature of the proposed system was to integrate the two.

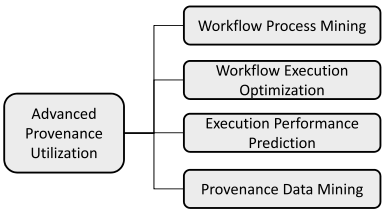

Fig. 7 Advanced provenance utilization

Advanced utilization of provenance

In the past years a lot of attentions have been paid to draw more advanced utilization of provenance data. The advanced utilization does neither mean simply browsing nor retrieving provenance information by simple predicates. That aims at better utilizing by deriving hidden information or mining execution patterns from accumulated provenance data. More specific application areas are (1) workflow process mining, (2) workflow execution optimization, (3) execution performance prediction, and (4) provenance data mining, as depicted in Fig. 7.

Workflow process mining

This area is specialized in locating a "partial workflow" that is frequently executed, by analyzing provenance data. A discovered partial workflow can be used to (i) grasp common tasks that are frequently executed, (ii) design new workflows, and (iii) recommend a workflow. A simulation platform can benefit from these use cases.

One example comes from Naseri et al. work [67], which proposed a novel approach of mining a workflow model from provenance information. Their work took advantage of Bayesian Structure-Learning (BSM) method [68, 69] for conducting mining a workflow model. That BSM method was used tor compute a probability that a certain task and another were executed together, build a skeleton of workflow, and then determine the order of tasks. Their approach was implemented on provenance data collected by the Taverna workflow system [13, 14].

The second example is a system called FlowRecommender [70]. This system aims at discovering a frequently-appearing task order from provenance data and then recommending that order to reflect it into workflow design. The system recommends to a user a frequent workflow model that is reconstructed by computing a frequency and a probability of each task sequence.

Similarly, another recommendation service was developed by De Oliveira et al. [71]. Their work focused on finding a composition of services or programs that are frequently executed together in order to utilize that composition for a new workflow design. They applied collaborative filtering to provide service or program recommendation [72]. Using the VisTrails workflow system [11], they implemented their system.

Another approach [73] focused on extracting from detailed provenance data key abstractions by finding a common partial workflow based on execution provenance. The common part can be utilized for reuse, circumstance understanding, and new workflow design.

Besides, Zeng et al. [74] proposed a method for mining a workflow model by considering not only task order but also data access order based on provenance data. Silva et al. [75] presented a technique for mining a declarative model—a model to describe via a logic language (such as Declare [15, 76]) interactions among actions in a certain process-based on provenance data stored along with PROV. DeBoer et al. [77] proposed a more efficient substructure mining technique for network provenance graphs. The ideas of these work could be applied to simulation provenance service platforms, so that simulation tools that are frequently executed can be recommended, or a frequent workflow template can be provided, or a frequent simulation tool or workflow can be identified on the platforms for better user convenience.

Workflow execution optimization

This area aims to improve the execution performance of a workflow that is currently requested, by utilizing provenance on previously-executed workflows. This provenance information typically includes (i) the elapsed time of a service, (ii) input/output parameters, (iii) success or failure, (iv) computing resources used for execution, and so forth. Taking advantage of these pieces of information, we can expect a subsequent execution of a workflow to be further optimized.

Leveraging provenance data, Missier [78] made an attempt to improve the workflow execution through data mining and machine learning techniques. His work provides a function to dynamically update a workflow in execution by utilizing provenance data. Namely, when a certain task of a given workflow instance should be executed, the best application can be recommended for that task, based on the analysis (such as speed/success or fail) on the past runs of applications. Then, a dynamic update is performed for the recommended application to carry out the task.

Altintas et al. [79] proposed an approach, called "smart rerun," for rerunning a subsequent execution of a workflow more efficiently in the Kepler workflow system [16, 80], based on collected provenance information. When users change some parametric values and re-execute a workflow, the proposed approach avoided rerunning the tasks that should be executed again, regardless of the change of the values. This smart rerun took advantage of the provenance information capturing parametric values that were entered in the past and tasks that were executed previously.

There was an approach [17] to analyzing provenance data and then predicting future demands on grid and cloud computing resources. The authors applied pattern matching algorithms on the most recent provenance data to foresee subsequent requests. That is, their work predicts a next demand by matching similar patterns to a recent usage pattern based on provenance data.

Besides, Li's team [18] proposed a statistical technique for inferring and categorizing with high accuracy users' behaviors based on provenance logs (collected by Progger [81]).

As done in these works, simulation platforms can improve their performance by reusing existing simulation results or allocate in advance necessary computing resources through provenance data analysis.

Execution performance prediction

This area pursues better estimating completion time or resource consumption on future computing jobs by utilizing provenance data. Specifically, workflow execution time and I/O pattern prediction can be an example of interest in this area.

Dai et al. [82] proposed a technique for predicting I/O traffic incurred by the system executing an application based on provenance records. The authors assumed that the future I/O pattern could be predicted in the case of iteratively executing the same application for several datasets. Their approach proceeded with (1) similar application clustering, (2) I/O pattern understanding, and (3) I/O patter prediction.

Malik's group [19] suggested a method of predicting the execution time of a computing job on Grid infrastructures, via machine learning methods. For model training, they utilized provenance data in association with job execution. Their model was based on MLP [83] was trained through feature selection by PCA [84].

There was another work [20] to estimate the execution time of the medical data applications based on provenance and performance data collected in the e-Science Central workflow system [49]. In this work, the authors invented their own prediction model based on provenance information including data size, algorithm settings, and execution time.

Like these works, simulation platforms can increase users' convenience by considering (i) predicting the execution time of long-running simulations and (ii) coping proactively with simulations that require huge computing resources through the useful utilization of provenance data.

Utilization of provenance data mining

There have been quite a few researches conducted to apply mining techniques to provenance data. Chen proposed in his doctoral dissertation [85] a novel method to represent provenance graph data more efficiently in order to apply a variety of data mining algorithms. His method used graph partitioning and feature extraction techniques for compressing large-scale provenance graphs. To assess the effectiveness of the proposed provenance representation method, he tested traditional mining techniques, such as "k-means clustering," "random forests," and "apriori algorithm," for the discovery of abnormal or variant workflows, the understanding of different types among workflows, and the grasping of frequent pattern in workflow execution, respectively.

Macko et al. [86] presented an approach to applying "local clustering" to a large volume of provenance graph data that is growing over time. The authors' technique aims to mine high-level, meaningful information on a local cluster basis from the detailed provenance data.

Brady's team did research on finding significant components (i.e., objects or resources) comprising simulations by studying provenance data. His team computed the frequencies of the components included in the simulations via a cosine method. In this way, the authors utilized the frequencies for the optimization of subsequent simulations.

Another team [87] proposed a technique of approximately summarizing provenance information associated with data that was growing in size and becoming more complicated. To produce approximately summarized provenance, their technique leveraged

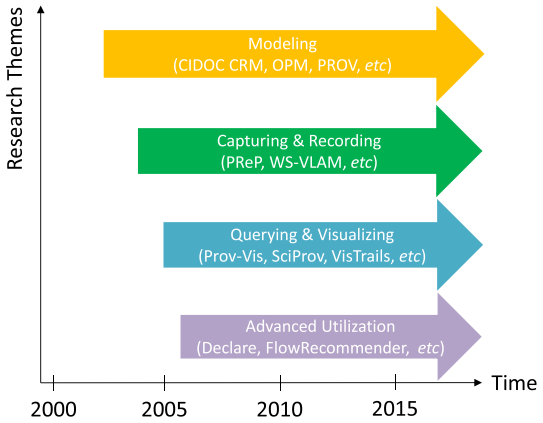

Fig. 8 A general trend of provenance research themes

semantic information among data and the usage of the provenance while compromising some information loss.

Utilizing reconstructed provenance information, another article [88] suggested an automated annotation technique for given data. Given a history of changes to data, this technique kept track of the provenance about the data and associate with the original data the metadata attached to the final output data. In particular, this reconstruction was performed via A* search algorithm and a heuristic function based on input and output data.

Besides, another example [89] is from applying data mining to astrological provenance data for the purposes of workflow type classification or clustering, interrelationship exploration, and outlying pattern discovery.

For better user service, simulation platforms can utilize data mining techniques used in this rich body of the existing literature, in order to find event patterns that frequently occur in simulations, detect abnormal simulations, or understand general characteristics of simulations.

A general trend of provenance research

Figure 8 visualizes how provenance research has so far evolved. The community began to work on provenance modeling in the first place. As mentioned before, CIDOC, OPM, and PROV were considered the most representative modeling techniques. The community then paid their attention to capturing and recording provenance via model-based, publish-subscribe, protocol, and logs. The popular examples were PReP (by protocol), WS-VLAM (by model-based), etc. Next, the community made a lot of efforts to query and visualize provenance data. Some of the representative examples were Prov-Vis, SciProv, VisTrails, and so forth. Lastly, the community exerted to apply more advanced utilization to provenance data. We considered some examples such as FlowRecommender, Declare, etc.

The following section discusses a taxonomy for the examined systems.

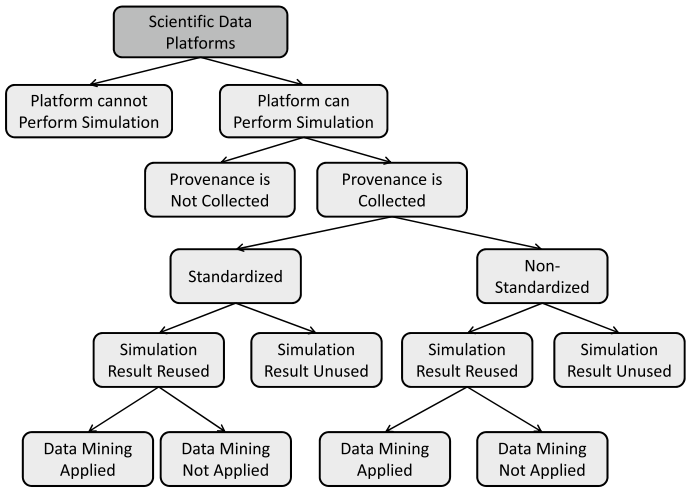

Fig. 9 A proposed taxonomy of simulation platforms regarding provenance

A taxonomy of scientific platforms over provenance support

In this section we propose a taxonomy for categorizing various scientific platforms in terms of provenance utilization. As depicted in Fig. 9, the taxonomy is structured along with well motivated criteria.

In the tree, the root node represents a set of scientific data platforms. Some platforms can perform simulations (the left child of the root) online while the others cannot (the right child of the root). If the platforms can conduct online simulations, then some of them (the right child at the third level) can collect provenance information from simulations that are successfully completed or failed for some reasons, or some others (the left child at the third level) may discard or ignore the information. Such a provenance can be collected in the form of standardization (the left child at the fourth level) or not (the right child at the fourth level). At the fifth level, the platforms capturing standardized or nonstandard provenance can be further divided into ones (the left child of each of the two nodes at the fourth level) that can reuse or ones (the right child of each of the two nodes at the fourth level) that do not reuse existing simulation results. Finally, if there exist the platforms which can utilize simulation results again, they can be divided into those that can or do not apply data mining to their provenance data (the left and right children of each of the left nodes at the fifth level, respectively).

Table 1 compares different simulation service platforms in terms of their application domains and provenance support. The top row of Table 1 represents features to identify specific characteristics of those platforms. Most of the features are from the taxonomy in Fig. 9. In particular, the subrow appearing below from 'Modeling' through 'Data Mining' indicates how strongly the associated feature is supported. The support level can be seen as a sort of *ranking*. The leftmost column in the table represents a number of different platforms examined in our survey.

When it comes to the ranking, here are our criteria associated with each feature. First, let's consider the "modeling." If a given system follows PROV (or OPM) (standard), some other languages such as XML or a relational model like a table [(semi-) structured], or a regular file (unstructured), then the support level of the system is

Table 1 Major features and their overall levels of simulation platforms to support provenance

Platforms	Application domain	Provenance subject	Preservation (where to record)	Modeling (how to represent)	Querying (how to access)	Visualization (how to show)	Result reuse	Reproduction	Data mining
				(Support level = [*high* | *medium* | *low*])					
EDISON [3]	Scientific simulation (domain-neutral)	Simulation	RDBMS	Table (*med*)	SQL (*high*)	None (*low*)	Limited (Manual) (*med*)	Possible (*high*)	None (*low*)
myGrid [90]	Biology/bioinformatics	Workflow	RDBMS	XML/HTML/RDF (*med*)	SQL (*high*)	Graph (*high*)	Limited (Manual) (*med*)	Possible (*high*)	None (*low*)
Taverna [14]	Biology/bioinformatics	Workflow	RDBMS	XML (Scufl)+RDF (*med*)	SQL (*high*)	Graph (*high*)	Unknown (*low*)	Unknown (*low*)	None (*low*)
Chimera [91]	Physics/astronomy	Workflow	RDBMS	VDL (*med*)	SQL+VDL (*high*)	Graph (*high*)	Unknown (*low*)	Unknown (*low*)	Unknown (*low*)
CMCS [92]	Chemistry	Workflow	File system	XML/RDF (*med*)	Via a browser (*low*)	Graph (*high*)	Unknown (*low*)	Unknown (*low*)	None (*low*)
PASOA [93–98]	e-Science	Service	Memory/RDBMS/file system	Unknown (*low*)	Java-based API/XQuery (*high*)	Unknown (*low*)	Unknown (*med*)	Possible (*high*)	None (*low*)
ESSW [99]	Earth science	Workflow	RDBMS	XML (*med*)	SQL (*high*)	Graph (*high*)	Unknown (*low*)	Unknown (*low*)	Unknown (*low*)
Kepler [16]	General science	Workflow	File system	MoML (XML) (*med*)	File search (*low*)	Unknown (*low*)	Unknown (*low*)	Possible (*high*)	Unknown (*low*)
Kepler Distributed Provenance Framework [100]	Kepler-extension	Workflow based on MapReduce	RDBMS (MySQL)	Table (*med*)	API/SQL (*high*)	Unknown (*low*)	Unknown (*low*)	Unknown (*low*)	Unknown (*low*)
RAMP [101]	Distributed system	Workflow based on MapReduce	File system/key-value store	File (*low*)	API (*med*)	Unknown (*low*)	None (*low*)	None (*low*)	Unknown (*low*)
HadoopProv [102]	Distributed system	Workflow based on MapReduce	File system/key-value store	File (*low*)	API (*med*)	Graph (*high*)	None (*low*)	None (*low*)	Unknown (*med*
Pig Lipstick [103]	Distributed system	Workflow based on MapReduce	Pig latin	OPM (*high*)	Graph-based API (*med*)	Graph (*high*)	Unknown (*low*)	Unknown (*low*)	Unknown (*low*)
Karma [104]	Weather forcast	Dynamic workflow	RDBMS	XML (*med*)	SQL (*high*)	Graph (*high*)	Unknown (*low*)	Unknown (*low*)	Unknown (*low*)
Pegasus [105–107]	Distributed system	Workflow	VDS/RDBMS	OWL (*med*)	SPARQL/SQL (*high*)	Unknown (*low*)	Unknown (*low*)	Unknown (*low*)	Unknown (*low*)
REDUX [108]	Windows system	Workflow	RDBMS	Table (*med*)	SQL (*high*)	Unknown (*low*)	Unknown (*low*)	Possible (*high*)	Unknown (*low*)
Swift [109–111]	Distributed system	Workflow	File system	File (*low*)	File Search (*low*)	Unknown (*low*)	Unknown (*low*)	Unknown (*low*)	Unknown (*low*)

Table 1 (continued)

Platforms	Application domain	Provenance subject	Preservation (where to record)	Modeling (how to represent)	Querying (how to access)	Visualization (how to show)	Result reuse	Reproduction	Data mining
				(Support level = [*high* | *medium* | *low*])					
VisTrails [11, 12, 66]	Computing system	Workflow	RDBMS	Python Object (XML/Table) (*med*)	System-defined query (*med*)	Graph (*high*)	Unknown (*low*)	Unknown (*low*)	Unknown (*low*)
PASS [112]	Computing system	Linux process	Berkeley DB	File (*low*)	Graph-based query (nq) (*med*)	Graph (*high*)	None (*low*)	None (*low*)	Unknown (*low*)
ES3 [108]	Computing system	Linux process	DBMS	XML (*med*)	SQL (*high*)	Graph (*high*)	None (*low*)	None (*low*)	Unknown (*low*)
CloudProv [113]	Cloud system	Real-time application	Database	File (*low*)	API (*med*)	Unknown (*low*)	Unknown (*low*)	Unknown (*low*)	Unknown (*low*)
Milieu [114]	Scientific simulation	Workflow	Database	Table (*med*)	SQL (*high*)	Unknown (*low*)	Unknown (*low*)	Possible (*high*)	Unknown (*low*)
Sumatra [115, 116]	Scientific simulation	Program	File system	(CSV) File (*low*)	File search (*low*)	Possible (*med*)	Unknown (*low*)	Possible (*high*)	Unknown (*low*)
e-Science Central [117, 118]	Scientific simulation	Workflow/scientific data	PaaS	OPM (*high*)	Web-based interface (*low*)	Graph (*high*)	Possible (*med*)	Possible (*high*)	None (*low*)

considered as *high*, *medium*, or *low*, respectively. Concerning the "querying," if the system uses a high-level query language such as PROV-AQ, SQL, and SPARQL, a querying method based on an API (Application Programming Interface), or a regular file search, the respective level is *high*, *medium*, or *low*. As far as "visualization" is concerned, the level is *high*, *medium*, or *low* if the system can show provenance in the form of graph, if another form of visualization is possible, or if unknown or no tool exists, respectively. For the "result reuse," if the system supports an automated technique or a manual (or possible) method, then the level corresponds to *high* or *medium*, respectively, and if not known or not possible, then *low*. Considering the "reproduction," if the system can reproduce simulation results via provenance, the level is *high*, and otherwise, *low*. Lastly, if "data mining" is applied in the system, the level is *high*, and otherwise, *low*.

EDISON [3] is an online simulation platform developed to support various tools (or *science apps* [31]) from several computational science and engineering disciplines (including computational fluid dynamics, nano physics, computational chemistry, structural dynamics, computer-aided optimal design, computational medicine, urban environment, and computational electromagnetics as of March 2018). If a user runs a simulation on the EDISON platform, the provenance information in association with that simulation is stored as a tuple in a relation (or a table) managed by an RDBMS. Thus, the provenance can be queried by SQL. To visualize the completed simulation's results, EDISON launches a visualization tool that it has, or it is possible to connect to an external visualization tool such as ParaView [119]. Little visualization is supported on the provenance data. Reusing simulation results seems possible but limited. It is possible to reproduce simulation results based on the stored provenance, but it appears that provenance data mining is not applied to the platform yet although its interest is growing [31].

myGrid [90] is a simulation platform for biology or bioinformatics. Provenance is recorded for an executed workflow. The provenance record, which can be represented XML, HTML, and RDF, is stored in an RDBMS and thus is queried by SQL. It is possible to visualize stored provenance data in RDF via an external tool. It seems that workflow execution results can be reused. Also, the provenance information can be utilized to reproduce the existing results. But it is not known whether provenance data mining is supported in that platform.

Taverna [14] is a simulation workflow system used in bioinformatics. Extending an existing workflow engine, it adds a function of automatically collecting provenance while running a given workflow. The provenance information is captured in the form of XML(Scufl [14])/RDF and stored into an RDBMS (or MySQL [120]). A graph is used for provenance visualization. Unfortunately, it is unknown about whether to reuse simulation results and reproduce workflow execution based on provenance. Obviously, mining techniques were not applied to its provenance data.

Chimera [91], used in physics and astrology, is a prototype system to implement virtual data grid (VDG)—collaboration environment—in which data objects are generated and shared. The system captures each step of data conversion as provenance. Specifically, if a user describes in virtual data language (VDL) a workflow of performing a series

of tasks on a given dataset, and subsequently the workflow gets executed, then the system represents in VDL and stores into an RDBMS a provenance record of when the conversion was performed on which dataset at what time. A user can query the collected provenance via SQL or VDL. The query result is returned in the form of a graph consisting of nodes representing applications and edges indicating input/output data. The stored provenance data can be utilized to optimize reproduction tasks. However, it is not certain whether simulation results by the workflow can be reused, reproduced, and further be utilized for data mining.

Collaboratory for Multi-scale Chemical Science (CMCS) [92] is a system for collaboration and data management used in chemistry. In this system provenance is stored in Scientific Annotation Middleware (SAM) to keep track of which scientific data was generated by which process. Unfortunately, CMCS does not support automated extraction of provenance. Thus, an application of a workflow itself should capture the provenance information, or a user needs to input the provenance manually via a web portal when the workflow is executed. A user can query the stored provenance and view the provenance in the form of a graph in a specialized browser. It is not known regarding simulation result reuse. Also, it is not certain whether the system could support reproduce the result. Applying data mining to the provenance data is not considered.

Provenance Aware Service-Oriented Architecture (PASOA) [93–98] is a platform to support provenance across e-Science. The platform uses a standardized protocol, called PReP (Provenance Recording Protocol), to collect, store, and infer provenance. Each service, belonging to a workflow, needs to capture its provenance individually. This sort of provenance is stored in memory, an RDBMS, or a file system. A user can query provenance data via Java-based query API or XQuery. Little known is how to visualize the provenance. We do not know simulation results can be reused, and it seems that provenance data mining is not applied in the platform. It, however, is possible to reproduce an executed workflow based on the provenance data.

Earth System Science Workbench (ESSW) [99] is a simulation system to support metadata management and data storage in earth science. Provenance in ESSW is collected to record the holistic information about converting data obtained from satellite. This data conversion is represented in a workflow script, which connect data flows and then generates provenance data. A script writer needs to store into an RDBMS the provenance data via a library provided by ESSW. It is possible to query the data via SQL. The provenance can be viewed in the form of a graph in a web browser. Simulation result reuse and reproduction as well as provenance data mining are not known for the system.

Kepler [16] is a scientific workflow system, adding provenance framework to manage links indicating the lineage of data generated by workflows. This provenance information is expressed and stored as file in Modeling Markup Language (MoML), a variant of XML. The stored provenance is searchable via a file system, but a high-level query language is not supported for the search. It is possible to reproduce experiment process based on the provenance data in the Kepler system. But it is little known about the provenance visualization, simulation result reuse, and data mining.

Kepler Distributed Provenance Framework [100] is a system to extend the Kepler system to support provenance on MapReduce-based workflow. A (relational) data model was proposed to represent the provenance within a MapReduce job. The

framework provides functions of storing and querying (via an API) provenance along with the proposed model. The stored provenance information is distributed across MySQL clusters. It is almost unknown about the advanced utilization of provenance including simulation result reuse, reproduction, and data mining.

Reduce and Map Provenance (RAMP) [101] is a system to extend Hadoop, and capture and keep track of provenance on a workflow consisting of MapReduce jobs. It is possible to automatically collect detailed provenance via wrapped Hadoop API while being nonintrusive to users and Hadoop. The collected provenance is stored as a file in a Hadoop File System (HDFS) and can be queried via a Hadoop API. There is little known about the visualization as well as provenance data mining. It seems that this work is orthogonal to simulation result reuse and reproduction.

HadoopProv [102] is very similar to RAMP. HadoopProv adds to Hadoop collecting and analyzing provenance on MapReduce jobs. Its goal is to minimize provenance collection cost. One thing differing from RAMP is to answer a provenance query by generating a resulting graph for the visualization.

Pig Lipstick [103] is a system not only to keep track of provenance but to visualize a workflow executed in Pig Latin [121]. This system uses OPM to capture provenance on the workflow. The provenance is represented as a graph. It is obvious that this work is independent of simulation result reuse and reproduction. It is not known whether provenance data mining is applied to the data.

Karma [104] is a workflow system for weather forecast. Especially, this system supports *dynamic workflow*, meaning one whose execution path is changed along with an external event. The system collects from workflow logs and stores execution provenance into a central database server. Even though the provenance is represented in XML, the final form is in the tuple format. The provenance can be visualized as a graph via a tool provided by the system. The advanced utilization including simulation result reuse, reproduction, and data mining is not known.

Pegasus [105–107] is a workflow engine to automatically convert a given high-level workflow specification into to a concrete execution plan in a distributed environment. Provenance in Pegasus is collected by using virtual data system (VDS), expressed in OWL, and finally stored into an RDBMS. The collected provenance can be queried via SPARQL and SQL. It is little known about the visualization of the provenance and the advanced utilization.

REDUX [108] extends Windows Workflow Foundation engine and adds to it a function of automatically collecting logs about workflow execution. For the logs, provenance data about the executed workflow are collected. REDUX stores in an RDBMS the provenance data in the form of a relation, which can be queried by SQL. The engine can also reproduce an executed workflow by executing queries to find all steps involving data generation. However, we do not know about the advanced utilization as well as visualization.

Swift [109–111] is a system to support SwiftScript as script language combined with high performance execution system. The system collects provenance consisting of a various piece of information such as program name, parameters, start time, end time, elapsed time, termination status, and execution machine. This provenance is used for

workflow scheduling and optimization. Unfortunately, there is little known about provenance visualization and data mining and simulation result reuse and reproduction.

VisTrails [11, 12, 66] is a system developed to support workflow and provenance management. The main goal of VisTrails is to enable a user to query via a very intuitive interface based on QBE and recycle provenance data. The provenance data can be internally represented by a Python object, which is convertible to XML or a table. The converted data are stored into an RDBMS. Retrieved provenance data can be visualized in the form of a graph. It seems not known about whether or not to reuse simulation results and to apply provenance data mining.

PASS [112], running at a file system level, is a system to capture as provenance data a variety of execution statistics of a Linux process, including the name of that program, what was the input to that program, and what files were produced. Provenance collection is performed by the Linux kernel. Berkeley [122] serves as a preservation repository for the provenance data, which are represented as a graph. It is possible to query the provenance graph via a tool named 'nq' or via various languages supported by Berkeley DB. This system is orthogonal to simulation result reuse or reproduction. We do not know whether data mining is applied to the provenance data. PASS is in wide use for collecting provenance in the cloud.

ES3 [108] is a system aiming at extracting provenance information from an arbitrary application running on Linux. Its provenance extraction method is to monitor the interaction between the target application and its running environment. The interaction is stored as a log record into ES3 database, which represents the provenance information by a graph. ES3 is also not related to simulation result reuse and reproduction. It is not known about whether provenance data mining is applied.

CloudProv [113] is a framework to incorporate, model, and monitor provenance about data coming in real time in the cloud environment. The proposed framework provides public API that can be used to develop an application for sharing and incorporating provenance data. The collected provenance data is stored in provenance database by collection manager via the API. It is possible to query the provenance data via the API. Simulation result reuse and reproduction are not applicable to the CloudProv framework. We do not know about the applicability of mining techniques to their provenance data.

Milieu [114] is a framework to collect provenance about scientific experiment on HPC systems. The provenance data is stored in a separate database and can be queried by SQL. Users can reversely trace the production process of certain data based on the provenance data, which can be referred for reproducing scientific simulation and experiments. There is little clue about simulation result reuse and provenance data mining.

Sumatra [115, 116] is a provenance management and trace tool to support numerical simulation or analysis for the purpose of reproduction. The system allows for reproducing simulation results by storing and managing via Python API execution provenance including code (or program) version, parameter files and options, and the information of the platform to run the code. The simulation results can be annotated and browsed via a web interface. The provenance data are stored as a CSV file, which can be searchable in a file system. It is possible to reuse and reproduce simulation results. However, it is uncertain about whether provenance data mining is applied to the tool.

Lastly, e-Science Central [117, 118] is a cloud-based computing platform to support Software as a Service (SaaS) and Platform as a Service (PaaS) for scientific data management, analysis, and collaboration. Through this platform, scientists can not only upload and share their simulation results and execute workflow, but also query and view provenance information about each data item. OPM is used to represent the provenance data, and a graph to visualize the data in the platform. Unfortunately, provenance data mining is of little interest in this platform.

In short, many of the above systems (or platforms) can collect provenance and thereby provide users with richer execution statistics about simulation and workflow (, plus executed processes), but the advanced use of provenance leading with data mining seems not drawing much attention from those systems yet in spite of potential benefits for the users.

Where to go now: our suggestions for future research directions

This survey has so far reviewed a rich body of existing literature involving simulation provenance data management. A number of papers and articles have focused on (i) building provenance data modeling in a standardized form, (ii) capturing and recording detailed provenance in a better-organized way, (iii) achieving helpful visualization and easy querying, and (iv) exploring advanced utilization of provenance data. As exhibited in Table 1, we also have conducted a comparative analysis of the studied platforms by the overall levels of their features.

In the meantime, we have found that there lie more research opportunities that can not only provide better simulation service but also further expand provenance research area. The opportunities represent "our own" future direction. Note also that some of the opportunities (such as predicting simulation execution time and detecting some abnormal simulation execution) are being realized in our "ongoing" research. In this section, we *propose* the opportunities as future research directions.

The opportunities concern seven different types of provenance-driven simulation services in the following.

Simulation execution time estimation

This type of service is to estimate how long simulation with given parameters will be conducted based on past provenance records. By utilizing the past execution data as a training set, we can build up "a machine learning model" for estimated simulation completion time. Specifically, we first identify what attributes from the provenance data involve estimating simulation time. We then select a prediction model. Using the existing provenance records, we train the model. For the new request, the trained model is asked to estimate when to finish the simulation with parameters specified in the request. Then the model produces execution prediction time. As more provenance data are available, the trained model can keep up-to-date by retraining, and accordingly the accuracy of estimated time will improve as well.

Abnormal simulation execution detection

This type of service aims at detecting any simulation whose execution provenance is *very* different from those of other simulations. The service can treat such an abnormal simulation as an *outlier* and alert it to a user that initiated that simulation. Whether a given simulation is abnormal is determined when (i) its output significantly deviates from others for the same or similar input parameter values, (ii) its completion time is substantially longer than usual for the same or similar input parameter values or, (iii) its input and output parameter values are considerably different from those of the other simulations. The abnormal simulations can be detected by applying to the provenance data "outlier detection" techniques (from data mining), in which an outlier refers to a much larger or smaller value compared to neighboring values, or a very rare value in a given distribution. In general, the degree of outlying values can be defined by "a statistical metric" (e.g., $> 3 \times$ standard deviations) or as "a user-specified threshold". The value of this service comes from the fact that the user can be quickly alerted on the outlierness of his/her simulation.

Event-aware simulation input parameter exploration

This service is intended to explore an input parameter set(s) causing an unusual "event" for a specified simulation. Such an event, for instance, refers to failed simulation, abnormal termination, or long-running simulation. When a user enters input parameter values into a simulation, the service can predict whether this simulation results in one of the events. To realize the service, we search for all simulation provenances related to the event that a user selects and extract a combination of input parameter values that most often appear along with the chosen event. To extract such information, we may use the "frequent pattern mining" [123] technique. As a great number of input parameters may be involved in a given simulation, it is essential to select such an efficient mining technique.

Prediction of termination status of simulation

This service pursues predicting whether a given simulation results in (i) a success, (ii) a failure, (iii) an abnormal exit, or (iv) a long-running status. Thanks to the service, before conducting a simulation a user may be informed in advance of the terminal status of the simulation. The service corresponds to "classification" [124] that determines the kind of the status through provenance analysis over input parameter values and termination status associated with the values. To perform the classification, the service can employ one of a variety of mining algorithms like *decision support tree*, *support vector machine*, *random forests*, *neural networks*, etc.. In particular, the selected algorithm must be able to classify the status into more than three categories.

Discovery of frequent pattern of simulation

This provenance service concerns uncovering hidden correlation that frequently occurs for every parameter participating in simulations. As an example, the service can find any association rule such that when the value of a certain field was 'A', it was more likely for another field to be 'B', or more generally, when the value of an input parameter was 'A', it was more likely for another input parameter to be 'B'. Through the service, a user can

find out a hidden execution pattern of simulations in which given that a value was 'X' for an input parameter, the value of another input parameter was 'Y' in most cases. The most relevant technique for realizing this service is frequent pattern mining mentioned earlier.

Simulation parameter recommendation

This service is to recommend input parameter values to obtain the output parameter values that we desire to know for a given simulation. The service can assist a user to figure out which input parameters can be specified for the very first simulation or for the simulation with a specific goal. In the service, we should find input parameter values associated with specified output parameter values as opposed to predicting the output values for the input values. Hence, this problem is totally different from regression analysis. To implement the service, we should develop a new mining algorithm, which may be more challenging than any other service.

Simulation provenance clustering

Finally, this service concerns grouping simulations with similar provenance. Instead of investigating the provenance of each simulation, this service intends to provide summarization of overall simulations. More specifically, we may obtain representative execution patterns of simulations by finding large clusters of simulations. In contrast, we may discern unusual execution patterns of simulations by finding very small clusters of simulations. This service can be implemented via clustering algorithms such as *k-means* [125], *hierarchical clustering* [126], *DBSCAN* [127], or *expectation-maximization algorithm* [128]. Furthermore, the similarity between provenances can be evaluated using the similarity between graphs [129].

By considering these suggested provenance services, we expect that simulation platforms can further improve the performance and elevate user convenience.

Conclusion

In this article we conducted a comprehensive survey of a rich body of existing literature discussing simulation provenance service systems. The main goals of this survey lie in (1) categorizing extant research articles into several major themes along with well-motivated criteria, (2) analyzing primary features of existing provenance systems, and (3) then ultimately better understanding how HPC simulation service systems can benefit from active provenance utilization. As our efforts to satisfy these goals, we provided a novel categorization consisting of four representative research themes as illustrated in Fig. 1. We then introduced many articles in the respective theme and delivered the key ideas of those articles. In Table 1 we also carried out an extensive, sold analysis of a number of different simulation service platforms in regard to provenance support. We finally proposed *several* research opportunities to further pioneer provenance research from new perspectives.

The following things could be considered as future work. It would be great if the evaluation results from the existing literature could be compared and contrasted with new empirical data that might be obtained independently. It would also be more interesting to investigate what metrics are useful to assess the performance of the studied systems in

terms of their efficiency and effectiveness. Moreover, it would be more valuable if some (common) benchmarks could be used in the performance evaluation.

Nevertheless, the attempts made by our survey are expected to contribute to ultimately enhancing the performance of the present simulation service platforms. From our study we have come to have a deep faith in (i) that this area still remains very charming for platform developers and researchers and (i) that simulation users will greatly benefit from advanced provenance service realized by addressing the challenges successfully.

Authors' contribution

YKS and KYL carried out a comprehensive survey of diverse simulation provenance service systems, proposed promising future research directions that are not yet much pioneered and drafted the manuscript. Both authors read and approved the final manuscript.

Author details

[1] School of Computer Science and Engineering, Kyungpook National University, 80 Daehak-ro, Bukgu, Daegu 41566, South Korea. [2] Department of Computer Science, Sookmyung Women's University, 100 Cheongpa-ro 47-gil, Yongsan-gu, Seoul 04310, South Korea.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2018R1C1B6006409) and by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. 2018-0-00269, A research on safe and convenient big data processing methods).

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Received: 1 June 2018 Accepted: 27 August 2018
Published online: 14 September 2018

References

1. McLennan M, Kennell R (2010) HUBzero: a platform for dissemination and collaboration in computational science and engineering. Comput Sci Eng 12:48–53
2. Klimeck G, McLennan M, Brophy SP, Adams GB III, Lundstrom MS (2008) nanohub.org: Advancing education and research in nanotechnology. Comput Sci Eng 10(5):17–23
3. Suh Y-K, Ryu H, Kim H, Cho KW (2016) EDISON: a web-based HPC simulation execution framework for large-scale scientific computing software. In: Proceedings of the 16th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGrid), IEEE, Piscataway, pp 608–612
4. Pardamean B, Baurley JW, Perbangsa AS, Utami D, Rijzaani H, Satyawan D (2018) Information technology infrastructure for agriculture genotyping studies. J Inf Process Syst 14(3):655–665
5. W3C PROV: PROV-Overview. https://www.w3.org/TR/prov-overview/. Accessed Jan 28 2018
6. Moreau L, Freire J, Futrelle J, McGrath RE, Myers J, Paulson P (2008) The open provenance model: an overview. In: International provenance and annotation workshop, Springer, Berlin, pp 323–326
7. Moreau L, Clifford B, Freire J, Futrelle J, Gil Y, Groth P, Kwasnikowska N, Miles S, Missier P, Myers J (2011) The open provenance model core specification (v1. 1). Future Gener Comput Syst 27(6):743–756
8. Herschel M, Diestelkämper R, Ben Lahmar H (2017) A survey on provenance: What for? What form? What from? Int J Very Large Data Bases (VLDB Journal) 26(6):881–906
9. Tylissanakis G, Cotronis Y (2009) Data provenance and reproducibility in grid based scientific workflows. In: Proceedings of the 2009 workshops at the grid and pervasive computing conference, IEEE, Piscataway, pp 42–49
10. Simmhan YL, Plale B, Gannon D (2006) A framework for collecting provenance in data-centric scientific workflows. In: Proceedings of the international conference on web services, IEEE, Piscataway, pp 427–436
11. Bavoil L, Callahan SP, Crossno PJ, Freire J, Scheidegger CE, Silva CT, Vo HT (2005) Vistrails: enabling interactive multiple-view visualizations. In: IEEE visualization (VIS), IEEE, Piscataway, pp 135–142
12. Freire J, Silva C The official website for VisTrails. https://www.vistrails.org/index.php/Main_Page. Accessed Feb 5 2018
13. Oinn T, Addis M, Ferris J, Marvin D, Senger M, Greenwood M, Carver T, Glover K, Pocock MR, Wipat A (2004) Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics 20(17):3045–3054
14. Apache Taverna: Apache Taverna. https://taverna.incubator.apache.org/. Accessed Mar 2 2018
15. Montali M, Pesic M, van der Aalst WM, Chesani F, Mello P, Storari S (2010) Declarative specification and verification of service choreographiess. ACM Trans Web 4:1–62

16. Altintas I, Berkley C, Jaeger E, Jones M, Ludascher B, Mock S (2004) Kepler: an extensible system for design and execution of scientific workflows. In: Proceedings of the 16th international conference on scientific and statistical database management (SSDBM), IEEE, Piscataway, pp 423–424
17. Caron E, Desprez F, Muresan A (2010) Forecasting for grid and cloud computing on-demand resources based on pattern matching. In: Proceedings of the second international conference on cloud computing technology and science, IEEE, Piscataway, pp 456–463
18. Li X, Joshi C, Tan AYS, Ko RKL (2015) Inferring user actions from provenance logs. In: Trustcom/BigDataSE/ISPA, 2015, vol 1. IEEE, Piscataway, pp 742–749
19. Malik MJ, Fahringer T, Prodan R (2013) Execution time prediction for grid infrastructures based on runtime provenance data. In: Proceedings of the 8th workshop on workflows in support of large-scale science, ACM, New York, pp 48–57
20. Hiden H, Woodman S, Watson P (2016) Prediction of workflow execution time using provenance traces: practical applications in medical data processing. In: Proceedings of the 12th international conference on eScience, IEEE, Piscataway, pp 21–30
21. Danger R, Joy RC, Darlington J, Curcin V (2012) Access control for OPM provenance graphs. In: International provenance and annotation workshop, Springer, Berlin, pp 233–235
22. Freitas A, Knap T, O'Riain S, Curry E (2011) W3P: building an OPM based provenance model for the web. Future Gener Comput Syst 27(6):766–774
23. Shu Y, Taylor K, Hapuarachchi P, Peters C (2012) Modelling provenance in hydrologic science: a case study on streamflow forecasting. J Hydroinf 14(4):944–959
24. Ebden M, Huynh TD, Moreau L, Ramchurn S. Roberts S (2012) Network analysis on provenance graphs from a crowdsourcing application. In: International provenance and annotation workshop, Springer, Berlin, pp 168–182
25. Glatard T, Lartizien C, Gibaud B, Da Silva RF, Forestier G, Cervenansky F, Alessandrini M, Benoit-Cattin H, Bernard O, Camarasu-Pop S (2013) A virtual imaging platform for multi-modality medical image simulation. IEEE Trans Med Imaging 32(1):110–118
26. Jung IY, Eom H, Yeom HY (2011) Multi-layer trust reasoning on open provenance model for e-Science environment. In: IEEE 9th International symposium on parallel and distributed processing with applications (ISPA), IEEE, Piscataway, pp 294–299
27. Gehani A, Tariq D (2012) SPADE: support for provenance auditing in distributed environments. In: Proceedings of the 13th international middleware conference, Springer, New York, pp 101–120
28. Zhao D, Shou C, Malik T, Raicu I (2013) Distributed data provenance for large-scale data-intensive computing. In: IEEE international conference on cluster computing (CLUSTER), IEEE, Piscataway, pp 1–8
29. Belhajjame K, B'Far R, Cheney J, Coppens S, Cresswell S, Gil Y, Groth P, Klyne G, Lebo T, McCusker J et al (2013) PROV-DM: The PROV Data Model
30. Pignotti E, Polhill G, Edwards P (2013) Using provenance to analyse agent-based simulations. In: Proceedings of the joint EDBT/ICDT 2013 workshops, ACM, New York, pp 319–322
31. Suh Y-K, Ma J (2017) SuperMan: a novel system for storing and retrieving scientific-simulation provenance for efficient job executions on computing clusters. In: 2017 IEEE 2nd international workshops on foundations and applications of Self* Systems (FAS* W), IEEE, Piscataway, pp 283–288
32. Cohen-Boulakia S, Biton O, Cohen S, Davidson S (2008) Addressing the provenance challenge using ZOOM. Concurr Comput Pract Exp 20(5):497–506
33. Doerr M, Theodoridou M (2011) CRM_{dig}: a generic digital provenance model for scientific observation. TaPP 11:20–21
34. Doerr M (2003) The CIDOC conceptual reference module: an ontological approach to semantic interoperability of metadata. AI Mag 24(3):75
35. Doerr M, Ore C-E, Stead S (2007) The CIDOC conceptual reference model: a new standard for knowledge sharing. In: Tutorials, posters, panels and industrial contributions at the 26th international conference on conceptual modeling, vol 83. Australian Computer Society, Inc, Australia, pp 51–56
36. Theodoridou M, Tzitzikas Y, Doerr M, Marketakis Y, Melessanakis V (2010) Modeling and querying provenance by extending CIDOC CRM. Distrib Parallel Databases 27(2):169–210
37. Gerhards M, Sander V, Matzerath T, Belloum A, Vasunin D, Benabdelkader A (2011) Provenance opportunities for WS-VLAM: an exploration of an e-Science and an e-Business approach. In: Proceedings of the 6th workshop on workflows in support of large-scale science, ACM, New York, pp 57–66
38. OASIS: OASIS Web Services Resource Framework (WSRF) TC. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf. Accessed Mar 11 2018
39. OASIS: OASIS Web Services Notification (WSN) TC. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn. Accessed Mar 11 2018
40. Groth P, Luck M, Moreau L (2004) A protocol for recording provenance in service-oriented grids. In: International conference on principles of distributed systems (OPODIS), vol 3544. Springer, Berlin, pp 124–139
41. Erl T (2005) Service-oriented architecture: concepts, technology, and design. Prentice Hall PTR, Upper Saddle River
42. Sun F, Zhao J, Gomadam K, Prasanna VK (2010) Provenance collection in reservoir management workflow environments. In: Proceedings of the 7th international conference on information technology: new generations, IEEE, Piscataway, pp 82–87
43. Kloss GK, Schreiber A (2006) Provenance implementation in a scientific simulation environment. In: International provenance and annotation workshop, Springer, Berlin, pp 37–45
44. Gaspar W, Braga RM, Campos F (2011) SciProv: an architecture for semantic query in provenance metadata on e-Science context. In: ITBAM, Springer, Berlin, pp 68–81
45. Berners-Lee T, Hendler J, Lassila O (2001) The semantic web. Sci Am 284(5):34–43
46. Lopez V, Fernández M, Motta E, Stieler N (2012) Poweraqua: supporting users in querying and exploring the semantic web. Semant web 3(3):249–265

47. Pérez J, Arenas M, Gutierrez C (2009) Semantics and complexity of sparql. ACM Trans Database Syst (TODS) 34(3):16
48. Woodman S, Hiden H, Watson P, Missier P (2011) Achieving reproducibility by combining provenance with service and workflow versioning. In: Proceedings of the 6th workshop on workflows in support of large-scale science, ACM, New York, pp 127–136
49. Hiden H, Watson P, Woodman S, Leahy D (2011) e-Science central: cloud-based e-Science and its application to chemical property modelling. Relatório Técnico CS-TR-1227, School of Comp. Sci. Newcastle University
50. Zhao J, Klyne G, Shotton D (2008) Provenance and linked data in biological data webs. In: Proceedings of the WWW2008 workshop on linked data on the web (LDOW 2008)
51. Wylot M, Cudre-Mauroux P, Groth P (2014) TripleProv: efficient processing of lineage queries in a native RDF store. In: Proceedings of the 23rd international conference on world wide web, ACM, New York, pp 455–466
52. Wylot M, Cudre-Mauroux P, Groth P (2015) Executing provenance-enabled queries over web data. In: Proceedings of the 24th international conference on world wide web, International World Wide Web Conference Committee, Geneva, pp 1275–1285
53. Wylot M, Cudré-Mauroux P, Groth P (2015) A demonstration of TripleProv: tracking and querying provenance over web data. Proc VLDB Endow 8(12):1992–1995
54. Wylot M, Cudre-Maroux P, Hauswirth M, Groth P (2017) Storing, tracking, and querying provenance linked data. IEEE Trans Knowl Data Eng 29:1751–1764
55. W3C PROV: PROV-AQ: Provenance Access and Query. https://www.w3.org/TR/prov-aq/. Accessed Mar 13 2018
56. Chen P, Plale B, Cheah Y-W, Ghoshal D, Jensen S, Luo Y (2012) Visualization of network data provenance. In: Proceedings of the 19th international conference on high performance computing (HiPC), IEEE, Piscataway, pp 1–9
57. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13(11):2498–2504
58. Smoot ME, Ono K, Ruscheinski J, Wang P-L, Ideker T (2010) Cytoscape 2.8: new features for data integration and network visualization. Bioinformatics 27(3):431–432
59. Lopes CT, Franz M, Kazi F, Donaldson SL, Morris Q, Bader GD (2010) Cytoscape web: an interactive web-based network browser. Bioinformatics 26(18):2347–2348
60. Horta F, Dias J, Elias R, Oliveira D, Coutinho A, Mattoso M (2013) Prov-Vis: Large-scale scientific data visualization using provenance. In: Proceedings of the international conference on high performance computing, networking, storage and analysis, Denver
61. de Oliveira D, Ogasawara E, Baião F, Mattoso M (2010) Scicumulus: a lightweight cloud middleware to explore many task computing paradigm in scientific workflows. In: IEEE 3rd international conference on cloud computing (CLOUD), IEEE, Piscataway, pp 378–385
62. Ogasawara E, Dias J, Silva V, Chirigati F, Oliveira D, Porto F, Valduriez P, Mattoso M (2013) Chiron: a parallel engine for algebraic scientific workflows. Concurr Comput Pract Exp 25(16):2327–2341
63. Jensen S, Plale B, Aktas MS, Luo Y, Chen P, Conover H (2013) Provenance capture and use in a satellite data processing pipeline. IEEE Trans Geosci Remote Sens 51(11):5090–5097
64. Simmhan YL, Plale B, Gannon D, Marru S (2006) Performance evaluation of the Karma provenance framework for scientific workflows. In: International provenance and annotation workshop (IPAW'06), Springer, Berlin, pp 222–236
65. Howe B, Lawson P, Bellinger R, Anderson E, Santos E, Freire J, Scheidegger C, Baptista A, Silva C (2008) End-to-end eScience: integrating workflow, query, visualization, and provenance at an ocean observatory. In: Proceedings of IEEE fourth international conference on eScience, IEEE, Piscataway, pp 127–134
66. Callahan SP, Freire J, Santos E, Scheidegger CE, Silva CT, Vo HT (2006) VisTrails: visualization meets data management. In: Proceedings of the 2006 ACM SIGMOD international conference on management of data, ACM, New York, pp 745–747
67. Naseri M, Ludwig SA (2013) Extracting workflow structures through Bayesian learning and provenance data. In: Proceedings of the 13th international conference on intelligent systems design and applications, IEEE, Piscataway, pp 319–324
68. De Campos CP, Zeng Z, Ji Q (2009) Structure learning of Bayesian networks using constraints. In: Proceedings of the 26th annual international conference on machine learning, ACM, New York, pp 113–120
69. Campos CP, Ji Q (2011) Efficient structure learning of Bayesian networks using constraints. J Mach Learn Res 12:663–689
70. Zhang J, Liu Q, Xu K (2009) FlowRecommender: a workflow recommendation technique for process provenance. In: Proceedings of the eighth Australasian data mining conference, vol 101, Australian Computer Society, Inc, Australia, pp 55–61
71. De Oliveira FT, Murta L, Werner C, Mattoso M (2008) Using provenance to improve workflow design. In: International provenance and annotation workshop, Springer, Berlin, pp 136–143
72. Schafer JB, Frankowski D, Herlocker J, Sen S (2007) Collaborative filtering recommender systems, vol. 4321. 2nd edn. Springer, Berlin, Lecture Notes in Computer Science, pp 291–324
73. Garijo D, Corcho O, Gil Y (2013) Detecting common scientific workflow fragments using templates and execution provenance. In: Proceedings of the seventh international conference on knowledge capture, ACM, New York, pp 33–40
74. Zeng R, He X, van der Aalst WM (2011) A method to mine workflows from provenance for assisting scientific workflow composition. In: IEEE world congress on services, IEEE, Piscataway, pp 169–175
75. Silva MF, Baião FA, Revoredo K (2014) Towards planning scientific experiments through declarative model discovery in provenance data. In: Proceedings of IEEE 10th international conference on eScience, vol. 2. IEEE, Piscataway, pp 95–98

76. Pesic M, Schonenberg H, Van der Aalst WM (2007) Declare: full support for loosely-structured processes. In: 11th IEEE international enterprise distributed object computing conference (EDOC), IEEE, Piscataway, p 287
77. DeBoer D, Zhou W, Singh L (2013) Using substructure mining to identify misbehavior in network provenance graphs. In: First international workshop on graph data management experiences and systems, ACM, New York, p 6
78. Missier P (2011) Incremental workflow improvement through analysis of its data provenance. In: TaPP
79. Altintas I, Barney O, Jaeger-Frank E (2006) Provenance collection support in the Kepler scientific workflow system. In: International provenance and annotation workshop, Springer, Berlin, pp 118–132
80. Ludäscher B, Altintas I, Berkley C, Higgins D, Jaeger E, Jones M, Lee EA, Tao J, Zhao Y (2006) Scientific workflow management and the Kepler system. Concurr Comput Pract Exp 18(10):1039–1065
81. Ko RK, Will MA (2014) Progger: an efficient, tamper-evident Kernel-space logger for cloud data provenance tracking. In: Proceedings of the 7th international conference on cloud computing (CLOUD), IEEE, Piscataway, pp 881–889
82. Dai D, Chen Y, Kimpe D, Ross R (2014) Provenance-based prediction scheme for object storage system in HPC. In: Proceedings of the 14th IEEE/ACM international symposium on cluster, cloud and grid computing, IEEE, Piscataway, pp 550–551
83. Alpaydin E (2010) Introduction to machine learning, 2nd edn. The MIT Press, Cambridge
84. Wold S, Esbensen K, Geladi P (1987) Principal component analysis. Chemom Intell Lab Syst 2(1–3):37–52
85. Chen P (2016) Big data analytics in static and streaming provenance. Ph.D. thesis, Indiana University
86. Macko P, Margo D, Seltzer M (2013) Local clustering in provenance graphs. In: Proceedings of the 22nd ACM international conference on information and knowledge management, ACM, New York, pp 835–840
87. Ainy E, Bourhis P, Davidson SB, Deutch D, Milo T (2015) Approximated summarization of data provenance. In: Proceedings of the 24th ACM international on conference on information and knowledge management, ACM, New York, pp 483–492
88. Groth P, Gil Y, Magliacane S (2012) Automatic metadata annotation through reconstructing provenance. In: Semantic web in provenance management workshop
89. Borne K (2009) Scientific data mining in astronomy. arXiv preprint arXiv: 0911.0505
90. Stevens RD, Robinson AJ, Goble CA (2003) myGrid: personalised bioinformatics on the information grid. Bioinformatics 19(suppl–1):302–304
91. Foster I, Vockler J, Wilde M, Zhao Y (2002) Chimera: a virtual data system for representing, querying, and automating data derivation. In: Proceedings of the 14th international conference on scientific and statistical database management, IEEE, Piscataway, pp 37–46
92. Pancerella C, Hewson J, Koegler W, Leahy D, Lee M, Rahn L, Yang C, Myers JD, Didier B, McCoy R (2003) Metadata in the collaboratory for multi-scale chemical science. In: International conference on Dublin core and metadata applications, Pancerella, Shillington, pp 121–129
93. Miles S, Wong SC, Fang W, Groth P, Zauner K-P, Moreau L (2007) Provenance-based validation of e-Science experiments. Web Semant Sci Serv Agents World Wide Web 5(1):28–38
94. Moreau L, Groth P, Miles S, Vazquez-Salceda J, Ibbotson J, Jiang S, Munroe S, Rana O, Schreiber A, Tan V (2008) The provenance of electronic data. Commun ACM 51(4):52–58
95. Groth P, Miles S, Moreau L (2009) A model of process documentation to determine provenance in mash-ups. ACM Trans Internet Technol (TOIT) 9(1):3
96. Groth P, Moreau L (2009) Recording process documentation for provenance. IEEE Trans Parallel Distrib Syst 20(9):1246–1259
97. Miles S, Groth P, Branco M, Moreau L (2007) The requirements of using provenance in e-Science experiments. J Grid Comput 5(1):1–25
98. Miles S, Groth P, Munroe S, Moreau L (2011) PrIMe: a methodology for developing provenance-aware applications. ACM Trans Softw Eng Methodol (TOSEM) 20(3):8
99. Frew J, Bose R (2001) Earth system science workbench: a data management infrastructure for earth science products. In: Proceedings of the thirteenth international conference on scientific and statistical database management (SSDBM), IEEE, Piscataway, pp 180–189
100. Crawl D, Wang J, Altintas I (2011) Provenance for MapReduce-based data-intensive workflows. In: Proceedings of the 6th workshop on workflows in support of large-scale science (WORKS'11), ACM, New York, pp 21–30
101. Ikeda R, Park H, Widom J (2011) Provenance for generalized map and reduce workflows. In: Proceedings of the fifth biennial conference on innovative data systems research (CIDR), Asilomar, pp 273–283
102. Akoush S, Sohan R, Hopper A (2013) HadoopProv: towards provenance as a first class citizen in MapReduce. In: TaPP
103. Amsterdamer Y, Davidson SB, Deutch D, Milo T, Stoyanovich J, Tannen V (2011) Putting lipstick on pig: enabling database-style workflow provenance. Proc VLDB Endow 5(4):346–357
104. Cheung K-H, Hager J, Pan D, Srivastava R, Mane S, Li Y, Miller P, Williams KR (2004) KARMA: a web server application for comparing and annotating heterogeneous microarray platforms. Nucleic Acids Res 32(suppl–2):441–444
105. Deelman E, Blythe J, Gil Y, Kesselman C, Mehta G, Patil S, Su M-H, Vahi K, Livny M (2004) Pegasus: mapping scientific workflows onto the grid. In: Grid computing, Springer, Berlin, pp 11–20
106. Deelman E, Singh G, Su M-H, Blythe J, Gil Y, Kesselman C, Mehta G, Vahi K, Berriman GB, Good J (2005) Pegasus: a framework for mapping complex scientific workflows onto distributed systems. Sci Program 13(3):219–237
107. Deelman E, Vahi K, Juve G, Rynge M, Callaghan S, Maechling PJ, Mayani R, Chen W, da Silva RF, Livny M (2015) Pegasus, a workflow management system for science automation. Future Gener Comput Syst 46:17–35
108. Barga RS, Digiampietri LA (2008) Automatic capture and efficient storage of e-Science experiment provenance. Concurr Comput Pract Exp 20(5):419–429
109. Wilde M, Hategan M, Wozniak JM, Clifford B, Katz DS, Foster I (2011) Swift: a language for distributed parallel scripting. Parallel Comput 37(9):633–652

110. Gadelha LM Jr, Clifford B, Mattoso M, Wilde M, Foster I (2011) Provenance management in Swift. Future Gener Comput Syst 27(6):775–780
111. University of Chicago Computation Institute: The Swift Project. www.ci.uchicago.edu/swift. Accessed Mar 5 2018
112. Macko P, Chiarini M, Seltzer M (2011) Collecting provenance via the Xen Hypervisor. In: TaPP
113. Hammad R, Wu C-S (2014) Provenance as a service: a data-centric approach for real-time monitoring. In: 2014 IEEE international congress on big data (BigData Congress), IEEE, Piscataway, pp 258–265
114. Cheah Y-W, Canon R, Plale B, Ramakrishnan L (2013) Milieu: lightweight and configurable big data provenance for science. In: Big data (BigData Congress), 2013 IEEE International Congress, IEEE, Piscataway, pp 46–53
115. Davison A (2012) Automated capture of experiment context for easier reproducibility in computational research. Comput Sci Eng 14(4):48–56
116. Davison AP, Mattioni M, Samarkanov D, Teleńczuk B (2014) Sumatra: a toolkit for reproducible research. In: Implementing reproducible research. CRC Press, Boca Raton, pp 57–79
117. Hiden H, Woodman S, Watson P, Cala J (2013) Developing cloud applications using the e-Science central platform. Phil Trans R Soc A 371(1983):20120085
118. Watson P, Hiden H, Woodman S (2010) e-Science central for CARMEN: science as a service. Concurr Comput Pract Exp 22(17):2369–2380
119. Ayachit U (2015) The Paraview guide: a parallel visualization application
120. Oracle Corporation: MySQL: The World's Most Popular Open Source Database. https://www.mysql.com/. Accessed Mar 22 2018
121. Olston C, Reed B, Srivastava U, Kumar R, Tomkins A (2008) Pig latin: a not-so-foreign language for data processing. In: Proceedings of the 2008 ACM SIGMOD international conference on management of data, ACM, New York, pp 1099–1110
122. Olson MA, Bostic K, Seltzer MI Berkeley DB (1999) In: USENIX annual technical conference, FREENIX track, pp 183–191
123. Han J, Cheng H, Xin D, Yan X (2007) Frequent pattern mining: current status and future directions. Data Mining Knowl Discov 15(1):55–86
124. Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques. Morgan Kaufmann Publishers Inc., San Francisco
125. Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY (2002) An efficient k-means clustering algorithm: analysis and implementation. IEEE Trans Pattern Anal Mach Intell 24(7):881–892
126. Murtagh F (1983) A survey of recent advances in hierarchical clustering algorithms. Comput J 26(4):354–359
127. Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. SIGKDD 96:226–231
128. Moon TK (1996) The expectation-maximization algorithm. IEEE Signal Process Mag 13(6):47–60
129. Hao F, Sim DS, Park DS, Seo HS (2017) Similarity evaluation between graphs: a formal concept analysis approach. J Inf Process Syst 13(5):1158–1167