

Job Provenance—Insight into very large provenance datasets^{*}

Software demonstration

Aleš Křenek^{1,2}, Luděk Matyska^{1,2}, Jiří Šitera¹, Miroslav Ruda^{1,2},
František Dvořák¹, Jiří Filipovič¹, Zdeněk Šustr¹, and Zdeněk Salvet^{1,2}

¹ CESNET z.s.p.o., Žitkova 4, 160 00 Praha 6, Czech Republic

² Institute of Computer Science, Masaryk University,
Botanická 68a, 602 00 Brno, Czech Republic

email: First.Last@cesnet.cz

Abstract. Following the job-centric monitoring concept, Job Provenance (JP) service organizes provenance records on the per-job basis. It is designed to manage very large number of records, as was required in the EGEE project where it was developed originally. The quantitative aspect is also a focus of the presented demonstration. We show JP capability to retrieve data items of interest from a large dataset of full records of more than 1 million of jobs, to perform non-trivial transformation on those data, and organize the results in such a way that repeated interactive queries are possible. The application area of the demo is derived from that of previous Provenance Challenges. Though the topic of the demo—a computational experiment—is arranged rather artificially, the demonstration still delivers its main message that JP supports non-trivial transformations and interactive queries on large data sets.

1 Introduction

Provenances are usually used to provide insight in the history of a particular piece of data. In the context of computational experiments they also serve as a source of additional information when results are out of expected bounds, grouped in new ways or organized in patterns not encountered previously. Systems like Job Provenance (JP) [1, 2], which keep track of a potentially huge number of executed jobs [3–5], are able to either guarantee or to repute that the computational experiment itself was properly executed and correct data were used.

In the proposed demonstration, we will show how the JP can be used to manage large amount of provenance data in a parametric study that is able to easily generate millions of provenance records, thus getting easily out of the range of conventional analysis methods.

^{*} This work has been supported by Czech research intents MSM6383917201 and MSM0021622419. Job Provenance was developed in the EU EGEE-II project, INFISO-RI-031688.

2 Demonstration scenario

2.1 Evaluated computational experiment

The demonstration is based on the conclusions of the study [6], which compares normal aging vs. Alzheimer’s disease. Image processing workflow of this study was used for both previous Provenance Challenges¹.

A critical step in this study is measurement of volume of seven regions of interest in the brain. In [6] this step is done manually, by an expert human operator. Our demonstration assumes development of a software tool to replace this manual step with an automated procedure. Then the topic of the demo is a computational experiment of evaluating and calibrating the tool—sweep over available datasets and the whole space of possible parameter settings.

We extend the original workflow with the acquirement of annotations of the softmeaned image. It is implemented either as the human action (for reference), or by invoking the hypothetical automated tool we evaluate.

The parametric study sweeps over 100 sets of available patient images, as well as the following parameters of the processing:

- The *order of the warping model* (`-m` argument of `align_warp`) has a significant impact on the computational complexity, therefore it makes sense to examine an overall sensitivity to it. Moreover, as each of the four input images is acquired with different MRI scan settings, the sensitivity can be different, therefore all four instances of the parameter have to be examined independently. We sweep those in 4 distinct steps: 3, 6, 9, and 12.
- We assume that the core of the hypothetical automated measurement of the volume of the regions of interest in the brain is some kind of thresholding on voxel intensity. Therefore the *critical threshold* is the principal parameter to be calibrated. The main goal of the experiment is finding a working range of this threshold, applicable (i. e. discriminating dementia) to all available inputs. As the full range of the intensity is 0–4095, we run the parameter in 40 steps of 100.

With the described set of parameters and their steps, the overall number of the workflow instances per a single input data set is $4^4 \times 40 = 10240$, yielding more than 1 million of computations on all inputs altogether.

2.2 Visual form — the demo GUI

The demonstration is done using a simple graphical interface (Fig. 1), crafted specifically for this purpose. There are three logically distinct parts forming the interface:

- *Controls for job selection* allow to specify parameter ranges (Sect. 2.1) to query. The effect of changing the selection, e. g. restricting the threshold values, is rendered immediately by the other GUI components.

¹ <http://twiki.ipaw.info>



Fig. 1. Application GUI

- *Occurrence diagram* of queried jobs is an approx. 20×20 array, where the x axis maps to the age of the subject while y is hippocampus volume². Each cell of the array is split up into “green” (non-demented) and “red” (demented) subject sections; the number of corresponding jobs (which processed data of subject of this age and dementia classification, and computed this volume) is visualized using color saturation. An attached color scale bar shows absolute numbers of jobs, up to several thousands per cell, typically. Array cells are selectable with a mouse.
- *Parameter value histograms* show the actual distribution (i. e. number of jobs again) of values of the five studied parameters corresponding to the *selected cells* of the occurrence diagram. Properties of these distributions can give clues for further changes in parameter selection.

2.3 Analysis step by step

In this section we describe the steps of the demonstration, as they are shown one after another, the phenomena observed in each one, as well as partial conclusions made.

Working range of threshold. We start with a full range of all the parameters, displaying all the jobs of the experiment. The resulting occurrence diagram is blurred, showing almost regular distribution of green and red color.

According to the conclusions of [6] there should be a clear horizontal separation between red and green regions. We select the intermediate region (hippocampus volume $6.3\text{--}6.6\text{ cm}^3$) on the diagram, which should be empty if [6]

² The region in brain which volume is related to dementia according to [6].

holds and the automated measurement works reliably, and we start examining distribution of the parameters there. The histogram of the threshold parameter shows low occurrence of mid-range values (1000–2500) while both lower and higher values occur rather frequently.

We conclude that the range of 1000–2500 is the working range of the threshold. The conclusion is confirmed by restricting the job selection to this range—a visible separation of the green and red regions appears.

Sensitivity to alignment parameters. Now the diagram shows also an anomaly—a strange sharp vertical bar (i. e. a failure to discriminate dementia) at the subject age of 82. For the time being we exclude it (ignore the region), and we focus on further improvement of the separation.

Further experiments with the restriction of the threshold parameter do not help anymore. Therefore we keep its range of 1000–2500, as well as the selection of hippocampus volume 6.3–6.6 cm³, and we focus on the warping order parameters. While the values of the third and fourth ones are distributed regularly, there is a visible domination of low values for the first and second ones. We deduce that there is a certain number of unaligned input images which require higher-order warping to get matched. If the warping order is restricted, the resulting softmeaned image is blurred, yielding the thresholding method to be unusable in general. Moreover, the sensitivity to the warping order is higher for the first two images. When the selection of these two values is restricted, the separation in the occurrence diagram improves visibly.

Defective input. Now we focus on the visible anomaly of the sharp vertical bar. Its strict vertical orientation indicates a fixed patient age, therefore suggesting that it may have occurred for a specific input data only. We select the central (i. e. failing) part of the vertical bar and query for occurrence of input files in this area. Domination of one input set indicates that the hypothesis is likely. Visualization of this specific four files reveals the reason—an image taken by error from a completely different experiment.

2.4 Batch job submission

After finding the defective input we query for all jobs that are affected by it, and mark them as invalid. The defect disappears from the diagram.

We retrieve the full specification of the affected jobs, and after replacing the reference to the defective input we submit the fixed specifications. The processing takes some time but we can observe its progress³.

³ The bottleneck is *execution* of the jobs on our testbed which accepts approx. 100 jobs per minute. Our measurements [5] show the raw JP input throughput is about 10× higher.

3 Experiment Setup

3.1 Job Provenance Service

JP [1] was developed to keep tracks of job execution in a Grid environment. Since references to input and output datasets as well as arbitrary application-specific attributes can be easily recorded with the jobs, the records gathered by JP form a natural provenance of the datasets. JP provides efficient means to store but also to search through such a provenance. Capabilities of JP were already demonstrated to data provenance community by participation on two previous Provenance challenges [2].

JP consists of two services — JP *Primary Storage* (JPPS) which keeps all job records permanently, and JP *Index Server* (JPIS), which is created, configured, and populated on demand, according to particular users’ needs. This feature is highly exploited in the described demonstration.

One of the most important aspects of JP is the capability to be a core of application specific job management tools. As we shown in three independent studies ([3–5]), JP helps the user to see a grid job as a scientific experiment and to focus on the application layer. In the mentioned studies JP also acts as a generic engine to build a custom job management GUI. Such a graphical application is relatively thin layer on top of JP and can be highly customized for a particular need of experimental scientist (to support specific workflows and views). One of these studies was also focused on overall throughput, where the target load of millions jobs per day was successfully achieved ([5]).

3.2 Job implementation

Unlike in Provenance Challenges, the internal structure of the image processing workflow is not important in our demonstration. Therefore we understand the whole workflow as a single job furthermore.

For the purpose of the demo (1 million of jobs in a reasonable time), the actual payload of the jobs is faked — the jobs refer to 100 pre-computed images, and the principal result (hippocampus volume) is generated pseudo-randomly in a distribution that allows “discovering” the phenomena described in Sect. 2.3. This artificial approach does not affect the main message of the demo — JP is able to deal with millions of provenance records, whatever was the way of obtaining them.

A core of the distribution is the formula

$$threshold \cdot warp_sensitivity + threshold^3 \cdot (1 - warp_sensitivity)$$

where *warp_sensitivity* is a number in the range 0–1 expressing how much a specific data set is affected by a given warping parameter settings. For its lower values the cubic term prevails, therefore the working range of the threshold, where the computed hippocampus volume is close to the real one, is fairly wide. On the contrary, for higher values of *warp_sensitivity* the formula is almost linear, hence requiring a specific threshold setting to yield the right result.

Job execution is monitored by L&B [7], application-specific tags (patient id, threshold, warping parameters, computed hippocampus volume etc.) are recorded in terms of L&B *user tags*. Full job records, including the application-specific attributes, are stored into JP shortly after their termination.

3.3 Testbed

The computations are run on a 16 CPU cores machine⁴, hosting multiple virtual machines and being managed by PBS. L&B server and both JP services were run on common “off-the-shelf” machines.

Due to the simplified job payload the limiting speed factor turned to be the processing of jobs in PBS. Besides the need of careful setup of L&B in order to avoid disk congestion, as well as a known but addressable bottleneck of not reusing an open ftp connection [5], we did not observe any serious performance problems.

4 Related JP Extensions

4.1 Direct JPIS database access

The described parametric study became a pilot application for a new interface to JP Index Server—direct SQL database access. Unlike the web-service interface of JPIS that was used in previous demonstrations, this time the GUI communicates directly with the database engine underlying JPIS. Structure of the database tables that are meant to be accessed directly is documented and it will become another JPIS public interface. Compared to the limited (by intention) querying functionality of the WS interface this approach gives the user the full capability of SQL and it lets her optimize the queries. In the specific case we benefit of the GROUP BY clause counting occurrences of age/hippocampus volume quickly.

On the other hand, certain performance and security issues emerge. Ill-specified queries can generate unacceptable load on the database. As this access mode is intended mostly for single user JPIS instances, we don't consider the performance issues serious this time. However, the standard, fine-grain access control layer of JPIS, implemented on the WS interface, is bypassed, allowing the user to see all JPIS data. The emerging security problems must be addressed, probably by implementing the finer access control on JPIS too.

We are also considering an OGF-DAIS⁵ compliant interface that would combine the portability of WS access with the expressiveness of SQL.

⁴ <http://meta.cesnet.cz/en/resources/hardware.html#manwe4>

⁵ <http://forge.gridforum.org/projects/dais-wg>

4.2 Application-specific JP type plugin

The application also demonstrated the use of the *type plugin* concept in JP. JPIS database can store, besides literal values of the attributes, also their shrunk “database” form. This approach does not imply any general restriction on full attribute values (they can contain even large binary data) while still allowing efficient queries on the database form executed directly by the SQL engine, e. g. to index the columns appropriately. In general there is no 1:1 mapping between full and shrunk values, therefore further filtering on the full values must be performed once they are retrieved with an SQL query. However, the result set of the query is not so large typically.

A JP type plugin is a library, linked into JPIS at run-time, performing the “full to shrunk” attribute value mapping. In addition, declarations of SQL column type for a specific attribute can be defined, and full-value comparison function provided.

Specifically the plugin for this experiment data rounds the “age” attribute to the nearest even value, truncates “hippocampus volume” into buckets of size 0.3, and it transforms real value of “clinical dementia rating” to boolean. Then a single query

```
select age,volume,cdr,count(*) ... group by age,volume,cdr
```

populates directly all the cells in the occurrence diagram (Sect. 2.2) within approx. 1–2 s.

4.3 Configuration extensions and database schema changes

The original database schema of JPIS allows multiple values of a single attribute for a single job. Therefore the attributes are stored in separate tables. However, on approx. 1 million of our job records, the “group by” query shown in Sect. 4.2 accessing multiple tables ran more than 1 minute, not being acceptable for interactive use.

Therefore we further extended the configurability of JPIS to distinguish between *single-* and *multiple-value* attributes. The latter ones are stored as before, however, the shrunk database form (Sect. 4.2) of the former ones are aggregated all in a single table, allowing more efficient queries. Our core “group by” query speeds up by a factor of almost 100.

5 Highlights and Conclusions

We show a specific usage of Job Provenance, a generic customizable system focused on work with huge number of provenance records. On the scenario of a hypothetical parametric study, involving more than a million of computational jobs, JP capabilities of interactive queries over such number of records are clearly demonstrated.

The described queries yield execution of fairly simple SQL statements processing approx. 1 million of tuples, so that their interactivity is not a surprising result nowadays. Our main message is demonstrating the capability of JP to record full information on job execution, to harvest the data required for the specific application (they represent a tiny fraction of the primary data), and to reprocess and make them available in a form suitable for interactive work (reasonable sized SQL database in this specific case). The demonstration also became a pilot application of the new direct JPIS database access interface.

From the application point of view, the queries represent non-straightforward transformations of the parametric space, therefore they can reveal unforeseen behaviour, pattern, and other phenomena that might have remained hidden with a straightforward visualization of the experiment results. Similarly, eventual defects, incorrect inputs etc., which would distort the experiment outcome, are also detected.

References

1. František Dvořák et al. gLite job provenance. In Luc Moreau and Ian Foster, editors, *Proc. International Provenance and Annotation Workshop (IPAW'06)*, volume 4145 of *LNCS*. Springer, 2006.
2. Aleš Křenek et al. gLite job provenance—a job-centric view. *Concurrency and Computation: Practice and Experience*, 20(5), 2007. DOI: 10.1002/cpe.1252.
3. Aleš Křenek et al. Multiple ligand trajectory docking study— semiautomatic analysis of molecular dynamics simulations using EGEE gLite services. In *Proc. Euro-micro Conference on Parallel Distributed and network-based Processing*, 2008.
4. Jaroslava Schovancová et al. VO AUGER large scale Monte Carlo simulations using the EGEE grid environment. 3rd EGEE User Forum, Clermont-Ferrand, France, 2008.
5. Aleš Křenek et al. Experimental evaluation of job provenance in ATLAS environment. *J. Phys.: Conf. Series*, 2007. Accepted.
6. Denise Head et al. Frontal-hippocampal double dissociation between normal aging and Alzheimer's disease. *Cerebral Cortex*, 15(6):732–739, 2005. DOI:10.1093/cercor/bhh174.
7. Luděk Matyska et al. Job tracking on a grid—the Logging and Bookkeeping and Job Provenance services. Technical Report 9/2007, CESNET, 2007. <http://www.cesnet.cz/doc/techzpravy>.