Making Database Applications Shareable

Boris Glavic
Illinois Institute Of Technology
bglavic@iit.edu

Tanu Malik
University of Chicago
tanum@ci.uchicago.edu

Quan Pham[1]
University of Chicago
quanpt@gmail.com

Motivation. Sharing and repeating applications that involve interactions with a database is currently a painful and cumbersome process. In this poster we discuss how to make sharing such applications trivial through the combination of three techniques and systems we have recently developed: **LDV (Light-weight database virtualization)** [1] is a tool for sharing and repeating applications accessing a relational database. LDV monitors the execution of an application including its database access and creates a reproducibility package that can be shared and reexecuted on a different machine. In addition to a provenance graph, such a package contains all dependencies of the application (libraries, binaries, and data files) and the relevant slice of the database required for reexecution. Running a packaged application requires neither any manual installation nor setting up a database. We use LDV as a client application for creating provenance graphs and packaging applications and as a tool for reexecuting packages. **GProM (Generic Provenance Middleware)** [2] is a database independent provenance middleware for computing provenance of queries, updates, and transactions over multiple database backends. We use GProM to compute fine-grained provenance for SQL commands to determine what data needs to be included in a repeatability package. That is, GProM is integrated with LDV to wrap the database interfaces used by an application. **PROVaaS** (http://provaas.org/) is a web platform (RESTful service) for storing and querying provenance documents. Clients can upload provenance described in PROV (http://www.w3.org/TR/prov-overview/). PROVaaS supports automatic inference of links between submitted PROV graphs (e.g., to identify that two entities in two PROV graphs represent the same file). The clients can query the provenance of an entity or activity across multiple graphs. In this project we use PROVaaS as a centralized repository for PROV graphs created by LDV.

Poster Description. We will give an overview (see Figure 1) of how we have "glued" the three systems together to create a novel infrastructure that is more than the sum of its components. The poster will give an introduction to each of the involved systems, show the architecture of our combined solution, and showcase example applications with overlapping data (screenshots) that are repeated on different machines with a wide variety of OS distributions and installed software (including database versions that conflict with the packaged database). The poster will also show how files and data that are used by multiple application are linked by PROVaaS.

[1]Work done during the author's Ph.D. study at University of Chicago.

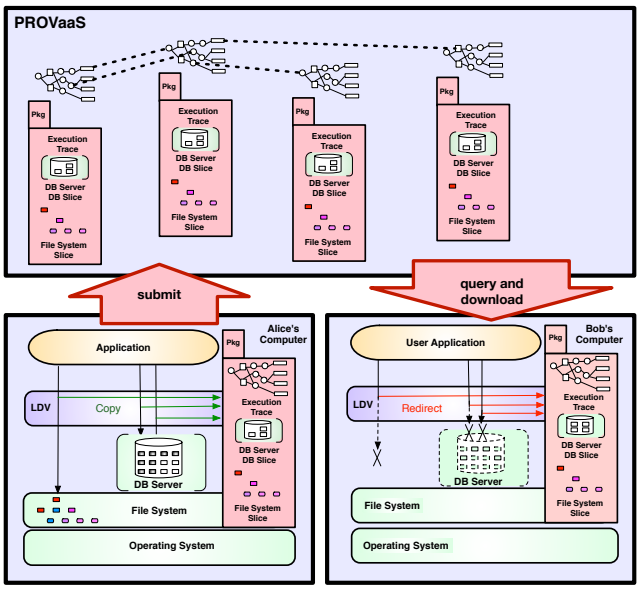

Fig. 1. System Overview

Conclusions and Future Work. By combining three provenance systems and techniques, we will develop the next generation reproducibility framework for database applications. In future work we would like to extend PROVaaS to store not only provenance but complete LDV packages and use the collective provenance information gathered by PROVaaS to be able to trade computation for storage. To reproduce any data produced by a package we have the choice of either caching that data (using storage) or to reproduce it by (partially) executing the package (using computation). Thus, we treat the provenance graphs created by LDV as recipes for producing data (files or database content). Furthermore, we want to enable users to query packages, applications, and data stored in PROVaaS based on provenance, data content, and additional metadata (e.g., user executing the application). Finally, to make use of provenance graphs for applications of the size we are targeting we need to be able to automatically generate concise summarizations of provenance.

References

[1] Q. Pham, T. Malik, B. Glavic, and I. Foster, "LDV: Light-weight Database Virtualization," in *ICDE*, 2015.

[2] B. Arab, D. Gawlick, V. Radhakrishnan, H. Guo, and B. Glavic, "A generic provenance middleware for database queries, updates, and transactions," *TaPP*, 2014.