Bio

Boris Glavic is an Assistant Professor of Computer Science at the Illinois Institute of Technology where he leads the IIT database group (http://www.cs.iit.edu/~dbgroup/). Before coming to IIT, Boris spend two years as a PostDoc in the Department of Computer Science at the University of Toronto working at the Database Research Group under Renée J. Miller. He received a Diploma (Master) in Computer Science from the RWTH Aachen in Germany, and a PhD in Computer Science from the University of Zurich in Switzerland being advised by Michael Böhlen and Gustavo Alonso. Boris is a professed database guy enjoying systems research based on solid theoretical foundations. His main research interests are provenance and information integration. He has build several provenance-aware systems including Perm (relational databases), Ariadne (stream processing), GProM (database provenance middleware), Vagabond (see http://cs.iit.edu/~dbgroup/research/index.html), and LDV (database virtualization and repeatability).

Abstract:

Provenance for database queries, information about how the outputs of a query where derived from its inputs, has recently gained traction in the database community resulting in the development of several models and their implementation in prototype systems. However, currently there is no system or model that supports transactional updates limiting the applicability of provenance to databases which are never updated. In this talk, I introduce reenactment, a novel declarative replay technique for transactional histories, and demonstrate how reenactment can be used to retroactively compute the provenance of past updates, transactions, and histories. The foundation of this research are MV-semirings, our extension of the well-established semiring provenance model to updates and transactions running under multi-versioning concurrency control protocols. In this model, any transactional history (or part thereof) can be simulated through a query, i.e., any state of a relation R produced by a history can be reconstructed by a query. We call this process reenactment. More formally, the reenactment query for a transactional history H is equivalent (in the sense of query equivalence) to the history under MV-semiring semantics. These formal underpinnings are the basis of an efficient approach for computing provenance of past transactions using a standard relational DBMS without having to compute and store provenance during transaction execution. I will show how reenactment queries can be constructed from an audit log, a log of past SQL operations, and how queries with MV-semiring semantics can be encoded as standard relational queries. A naive implementation would either require replay of the complete history from the beginning or proactive materialization of provenance while transactions are run. However, as long as a transaction time history is available, reenactment can be started from any past database state. Since most modern DBMS support audit logs and time travel (querying transaction time histories) out of the box and these features incur only moderate overhead on transaction execution, this approach enables efficient provenance computation for transactions on-top of standard database systems. I present encouraging experimental results based on our implementation of these techniques in our GProM (Generic Provenance Middleware) provenance database middleware. Furthermore, I sketch additional use cases for reenactment including post-mortem debugging of transactions (exposing the internals of a faulty transaction execution), historical what-if queries, and lazy materialization of versions in a data curation platform.