Accelerating Process Mining using Relational Databases

Alifah Syamsiyah, Boudewijn F. van Dongen, Wil M.P. van der Aalst

Eindhoven University of Technology, Eindhoven, the Netherlands
A.Syamsiyah@tue.nl, B.F.v.Dongen@tue.nl, W.M.P.v.d.Aalst@tue.nl

Abstract. Given the abundance of event data, the challenge of process mining today is to enable *process mining in the large*. This research aims to address scalability problem in terms of memory use and time consumption. To this end, we use relational databases as the framework to both store event data and do process mining analysis. We conduct a pre-computation of intermediate structures during insertion time of the data. Finally, we implement the existing process mining algorithms to be compatible with relational database settings.

Keywords: process mining, big event data, relational database

This document contains the PhD research plan and is organized as follows. In Section 1 we define what will be accomplished by eliciting relevant research questions. In Section 2 we present the background knowledge. In Section 3 we explain the significance of the research contribution. Then in Section 4 we describe the method adopted in the research. Finally, in Section 5 we present what we have done so far.

1 Research Questions

This work is conducted to answer these following research questions:

- How to deal with tremendous event data in process mining?
- How to do process mining analysis with event data taken from databases?
- How to gain performance benefit from relational databases in terms of memory use and time consumption?

2 Background

Process mining is introduced as a research discipline that sits between machine learning and data mining on the one hand and process modeling and analysis on the other hand. It can be viewed as a means to bridge the gap between data science and process science. The goal of process mining is to turn event data into insights and actions in order to improve processes [12]. Given the

rapid development of event data, the challenge is to enable *process mining in the large*. This work will be focused on the use of relational databases as a storage of event data and as an engine to pre-compute process mining metrics.

There are some works related to the use of databases in process mining. XESame [15] is one of the tools to extract event data from databases. This work provides an interactive interface where users can select data from the database and then match it with XES elements. The downside of this work is the lack of direct access to the database since it is only considered as a storage place of data.

Another technique for extracting event data from databases was presented in [3]. This work uses two types of ontologies. The first is called domain ontology which gives a high level view of the data stored in the database. The second is called event ontology which contains the XES structure. Data in databases is extracted through these ontologies using a well-establish technology called Ontology-based Data Access [7]. Although this work is promising, the performance issues make it unsuitable for large databases.

RXES was introduced in [13] as the relational representation of the XES standard for event logs. The work presents the database schema as well as some experiments showing that RXES uses less memory compared to the standard approach. RXES puts the initial stone for direct access to the database, however, this research has no longer continued.

In addition to database approaches, some other techniques for handling big data in process mining have been proposed [2,8,9], two of them are decomposing event logs [1] and streaming process mining [5]. In decomposition, a large process mining problem is broken down into smaller problems focusing on a restricted set of activities. Process mining techniques are applied separately in each small problem and then they are combined to get an overall result. This approach deals with exponential complexity in the number of activities of most process mining algorithms [11]. In streaming process mining, process mining framework ProM is integrated with distributed computing environment Apache Hadoop. Hence we can analyze event data whose size exceeds the computers physical memory. Streaming process mining also provides online-fashioned process mining where the event data is freshly produced, i.e. it does not restrict to only process the historical data as in traditional process mining. However, neither decomposition nor streaming are directly applicable to existing process mining technique. Both approaches require some changes in the algorithms.

3 Significance

Relational database is one of the technologies used in big data computing. This research uses relational databases as the framework to enable process mining in the large. We argue that relational databases are the most suitable approach for process mining compared to other types of databases. The XES standard requires a relational representation between its elements, for example, an event must belong to a trace and a trace is part of a log. Therefore, aggregate-

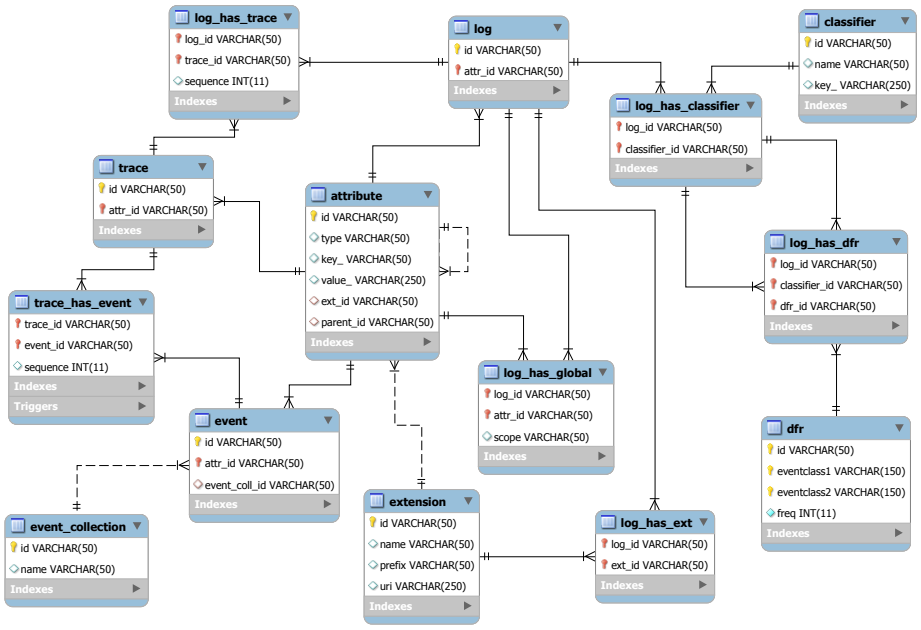

Fig. 1. DB-XES schema

oriented NoSQL databases [14] such as key-value store databases, document databases, and columned-oriented databases are not appropriate for XES event data. Relation-oriented NoSQL such as graph databases may be suitable, however, it does not provide supports for complex queries such as trigger.

Given the result from this research, process mining is able to handle big event data for discovering process models, doing conformance checking, and enhancing the process model. Moreover, process mining can be applied to the whole data to get insight from exceptional behavior.

4 Research Design and Methods

In this section we describe and motivate the method adopted in the research.

We first introduce a relational representation of XES, called DB-XES. Differently from normal process mining analysis which uses event log files, we use event data stored in relational databases. In other words, we move the location of data from files to databases. This provides scalability in terms of memory use due to the fact that memory is not bounded to the computer's disk size.

Second, we move some computations from analysis-time to insertion-time. We pre-compute intermediate structures of process mining algorithms in the database and keep the computed tables up-to-date of the insertion of new events. Using this approach, we maintain the intermediate structure to be always ready

and can be directly accessed by users whenever it is needed. This provides scalability in terms of time consumption since we cut the computation time inside process mining tools.

Figure 1 shows the DB-XES schema. As the XES structure [4], the schema contains main elements of event data, i.e. *log, trace, event,* and *attribute*. These elements are connected through table *log_has_trace* and *trace_has_event*. Global attributes, extensions, and classifiers are linked to the *log*. Furthermore, table *event_collection* is used to store the source of an event.

DB-XES also contains table *dfr* and *log_has_dfr*. This table is used to store *Directly Follows Relation (DFR)*, i.e. a pair of event classes *(a,b)* where a is directly followed by b in the context of a trace. DFR is one of the intermediate structures used by various process mining algorithms, such as Alpha Miner [10] and Inductive Miner [6].

For doing the experiment, we use real life event data from a company which contains 29,640 traces, 2,453,386 events, 54 different event classes, and 17,262,635 attributes. Then we extend this log in two dimensions, i.e. we increase (1) the number of event classes and (2) the number of traces, events and attributes. We extend the log by inserting copies of the original event log data with some modifications in the identifier, task name, and timestamp. We extend the number of event classes as a separate dimension since the growth of event classes gives exponential influences.

At the current stage, this work has limitation in the SQL query execution. The number of joins explodes and makes the query inefficient. Although the framework is still able to handle 10^8 number of traces, events, and attributes (the largest number used in the experiment), the need of optimizing the query still exists.

5 Research Stage

This research has been started since December 2015. In the first stage, we create a relational representation of XES called DB-XES. Then, using OpenXES as the interface, we create an access from DB-XES to ProM. Hence, any ProM plug-ins can work with DB-XES similarly as working with XES event log files.

In the next stage, we focus on enabling process discovery in large event data. We create a representation of the most common used intermediate structure, i.e. directly follows relations, in DB-XES. This structure is pre-computed and maintained to be up-to-date of the insertion of new events. Then, we conduct experiments using the state-of-the-art process discovery techniques, namely Inductive Miner. The result shows that the proposed solution gives performance benefit in terms of memory use and time consumption.

The experiment result is paving the way of applying other process mining techniques. In the current stage, we are implementing handover of work in DB-XES. The metrics have been translated into database tables, and some experiments are being run. In the following we briefly list the future research steps:

– Extend the approach with other advanced intermediate structures, such as the intermediate structures of declarative process mining.
– Apply the event removal feature in database while keeping the intermediate structures live under insertion and deletion of event data.
– Optimize the query performance through indexing and possibly apply more advanced big data technologies, such as Spark SQL.
– Implement conformance checking in the context of DB-XES.

References

1. W.M.P. van der Aalst. Decomposing Petri Nets for Process Mining: A Generic Approach. *Distributed and Parallel Databases*, 31(4):471–507, 2013.
2. A. Azzini and P. Ceravolo. Consistent process mining over big data triple stores. In *2013 IEEE International Congress on Big Data*, pages 54–61, June 2013.
3. Diego Calvanese, Marco Montali, Alifah Syamsiyah, and Wil MP van der Aalst. Ontology-driven extraction of event logs from relational databases. In *Business Process Intelligence 2015*. 2015.
4. C.W. Günther. XES Standard Definition. www.xes-standard.org, 2014.
5. Sergio Hernández, Sebastiaan J. van Zelst, Joaquín Ezpeleta, and Wil M. P. van der Aalst. Handling big(ger) logs: Connecting prom 6 to apache hadoop. In *BPM Demo Session 2015*.
6. Sander J. J. Leemans, Dirk Fahland, and Wil M. P. van der Aalst. Discovering block-structured process models from event logs - A constructive approach. In *PETRI NETS 2013*.
7. Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Journal on data semantics x. chapter Linking Data to Ontologies, pages 133–173. Springer-Verlag, Berlin, Heidelberg, 2008.
8. Hicham Reguieg, Boualem Benatallah, Hamid R. Motahari Nezhad, and Farouk Toumani. Event correlation analytics: Scaling process mining using mapreduce-aware event correlation discovery techniques. *IEEE Trans. Services Computing*, 8(6):847–860, 2015.
9. W. v. d. Aalst and E. Damiani. Processes meet big data: Connecting data science with process science. *IEEE Transactions on Services Computing*, 8(6):810–819, Nov 2015.
10. W. van der Aalst, T. Weijters, and L. Maruster. Workflow mining: discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, Sept 2004.
11. Wil M. P. van der Aalst. Distributed process discovery and conformance checking. In *FASE 2012*.
12. W.M.P. van der Aalst. *Process Mining: Data Science in Action*. 2011.
13. Boudewijn F. van Dongen and Shiva Shabani. Relational XES: data management for process mining. In *CAiSE 2015*.
14. Meenu Dave Vatika Sharma. Sql and nosql databases. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(8):20–27, August 2012.
15. H.M.W. Verbeek, J.C.A.M. Buijs, B.F. van Dongen, and W.M.P. van der Aalst. XES, XESame, and ProM 6. In P. Soffer and E. Proper, editors, *Information Systems Evolution*, volume 72, pages 60–75, 2010.