Proposal for a Master’s thesis

**Scalable Online Monitoring**

Supervisors: Dr. Dmitriy Traytel  
Professor: Prof. David Basin  
Issue date: 21 Aug 2015

Prerequisites

- Familiarity with linear temporal logic (e.g. from the FMFP course)  
- Some familiarity with Big Data frameworks (e.g. MapReduce, Hadoop, Spark)

**Introduction**

We consider the (Online) Monitoring Problem: Given a stream of time-varying data, called events, and a property formulated in a specification language, check whether the property is satisfied for every prefix of the stream. If the property is not always satisfied, the monitor should report the violating events. In contrast, the Offline Monitoring Problem assumes the complete data given as an event log at once, rather than as an evolving event stream. The Online setting is more difficult to handle and also subsumes the Offline setting.

As an example, consider a property from the domain of financial services: "total amount of money withdrawn by any customer in the last 30 days does not exceed 5000 CHF except if the customer has previously received a special credit limit". Suppose we wish to monitor this property on a stream of order and cash transfer events produced by a large bank. Already this simple property requires the specification language to be expressive. It requires aggregations ("total amount"), quantification ("any customer"), Boolean connectives ("except if"), and qualitative ("has previously received") and quantitative ("30 days") temporal connectives. A specification language that encompasses all those features is metric first-order temporal logic (MFOTL)—an expressive extension of linear temporal logic (LTL).

There is a large amount of prior work (see [2,3] for an overview) on monitoring MFOTL properties. A result of this work is the MonPoly tool, which works both in the offline and online setting on moderate quantities of data. Using the Google’s MapReduce framework [4], MonPoly successfully monitored several billion events in the offline setting by distributing the workload among 1000 machines [1]. The core idea there was to split the log into multiple slices and evaluate the entire formula on these slices in parallel.
Objectives

The MonPoly tool is reasonably efficient, but it needs help with scalability. In this master thesis we want to study the following question:

Can we replace MapReduce with a framework that supports the online setting, while otherwise retaining the data slicing strategy from [1] to obtain a scalable online monitoring algorithm?

Concretely, we would like to marry MonPoly with existing Big Data frameworks such as Apache Spark [5]. MonPoly will thereby be used as a black box for monitoring small portions of the input stream. Slicing the stream will require a mild extension of our previously used methodology [1] to the online case. The slicing and the combination of the results must be managed within the Big Data framework.

Tasks

This project can be subdivided into the following tasks:

1. Study the relevant literature on scalable monitoring in the offline setting [1].
2. Extend the event log slicing methodology developed in [1] to the online setting.
3. Study the landscape of Big Data frameworks that support the online setting. Pick the most suitable framework for implementing the previously developed stream slicing.
4. Implement a scalable online monitoring algorithm combining stream slicing and the MonPoly tool.
5. Evaluate the implementation in synthetic case studies.
6. Write the final report and prepare the presentation.

Deliverables

The following deliverables are due at the end of the project:

Final report  The final report should consist of an introduction; a theoretical background section; one or more sections describing the modelling, implementation and evaluation; and a conclusion. The report may be written in English or German. Two copies of the report must be delivered to the supervisor.

Source code  Complete development including easily reproducible case studies.

Presentation  At the end of the project, a presentation of 20 minutes must be given during an InfSec group seminar. It should give an overview and discuss the most important highlights of the work.
References


