Elastic and Fault Tolerant Event Stream Processing using StreamMine3G

André Martin, Do Le Quoc
Technische Universität Dresden - Dresden, Germany
Email: andre.martin@se.inf.tu-dresden.de, do@se.inf.tu-dresden.de

Abstract—The massive amount of new data being generated each day by data sources such as smartphones and sensor devices calls for new techniques to process such continuous streams of data. Event Stream Processing (ESP) addresses this problem and enables users to process such data streams in (soft) realtime allowing the detection as well as a quick reaction to relevant situations.

In this tutorial, we will introduce the participants to ESP techniques as well as ESP systems such as Storm, Apache S4 and StreamMine3G. We will cover aspects such as programming models, fault tolerance as well as elasticity and cloud support of these platforms.

Keywords—event stream processing, esp, cep, fault tolerance, mapreduce, deterministic execution, elasticity

I. MOTIVATION - WHY ATTENDING THIS TUTORIAL?

The cloud computing paradigm enables users to store and process practically any kind of data at low costs as well as computational resources can be acquired and released at any time. During the last decade, a number of cloud storage services have evolved such as Amazon S3, a popular cloud storage service as well as Amazon Elastic MapReduce (EMR) targeting the processing of the previously stored data. Although Amazon EMR allows the processing of huge amounts of data, it does not allow for processing of continuous streams of data in (soft) real time.

In this tutorial, we want to take a look at ESP systems, a technique which enables users to process unbounded streams of data. Processing data in real time imposes new challenges to researchers such as providing fault tolerance as well as adjusting such systems dynamically to fluctuations in the incoming event stream.

Within the two EU Projects STREAM [1] and SRT-15 [2], the Systems Engineering Group at TU Dresden developed its StreamMine3G ESP platform which supports fault tolerance as well as elasticity. Through the development of StreamMine3G, the group has gained extensive knowledge in designing and building ESP systems as well as problems associated with it such as fault tolerance and elasticity. We would like to share this experience with the participants of our tutorial.

II. TUTORIAL CONTENTS

The tutorial will be offered as a half day event in a lecture style accompanied with hands on exercises. Each session will start with some introductory slides introducing the audience into the specific area followed by hands on exercises. Currently, there are the following sessions planned with breaks in between and room for advanced topics.

A. Introduction in ESP Systems

The first session will introduce the participants to the fundamentals of ESP systems such as the architecture, events streams and topologies as well as extensions to ESP systems so called Complex Event Processing (CEP). This includes an introduction into the programming models [3] of two commonly used ESP systems: Storm [4], Apache S4 [5] and the in-house developed StreamMine3G. We will furthermore cover statefull as well as stateless operators as well as implementation aspects for data sources and sinks. The session will conclude with hands on exercises by...
setting up and executing small examples applications using StreamMine3G.

B. Fault Tolerance in ESP Systems

The second session will cover various fault tolerance concepts such as active [6] and passive replication [7]. We will investigate requirements to the architecture of an ESP system as well as the running applications in order to provide various properties such as precise recovery. In the hands on part, we will implement a sample application including state serialization concepts to make operators resilient to crashes of the systems and verify recovery techniques by actively crashing our ESP platform.

C. Elasticity in ESP Systems

This session will cover challenges in building elastic stream processing systems such as providing mechanisms to migrate operators to new nodes to allow a system to expand or shrink depending on the fluctuation of the incoming data stream. We will take a closer look at StreamMine3G’s manager component which allows the implementation of various policies with regards to cost models of cloud providers such as Amazon EC2. Participants are invited to implement simple migration strategies in StreamMine3G to lower resource consumption when running StreamMine3G in cloud environments such as Amazon EC2.

D. Performance Optimizations in ESP Systems

Within the forth session, we will cover various aspects such as event batching, delegation for batching as well as state access to improve throughput by lowering lock contention as well as latency in an ESP system.

E. Selected Advanced Topics

The last session will be scheduled on demand in case there is interest and sufficient time. This allows us to look into topics such as integration of existing data streams such as Twitter firehose, how to provide a deterministic and replayable input to the system through some log-stage mechanism, and research concerning operator scheduling and placement.

III. BIOGRAPHY OF THE LECTURERS

The tutorial will be given by André Martin and Do Le Quoc. André Martin is a forth year PhD student at the Systems Engineering Group of TU Dresden and his research is centered around fault tolerance as well as elasticity in MapReduce and ESP Systems. He has six years of experience in big data systems such as MapReduce and ESP and is the creator of the StreamMine3G ESP system which will be used within the tutorial. Do Le Quoc is a first year PhD student at the Systems Engineering Group of TU Dresden. His research interests include network management, anomaly detection, deep packet inspection, cloud computing, and big data analytics. He received his master degree in computer science from Pohang University of Science and Technology (POSTECH), Korea. He also worked at R&D center of DASAN Networks company, Seoul, Korea before starting his Ph.D. studies at the Systems Engineering Group of TU Dresden, Germany.

IV. AUXILIARY INFORMATION

The tutorial can easily be followed without active participation, although it would be a lot less fun. We therefore recommend participants to bring their laptops and prepare a virtualisation environment (VirtualBox is recommended). We will provide the users with an off-the-shelf ready VM with the StreamMine3G software and all dependend services installed through an USB stick.

REFERENCES


