i-MAGNET : A Real-time Intelligent Framework for Finding Specific Needles From Needle Stacks

Faisal Zaman*, Sebastian Robitzsch*, Zhiguo Qu*, John Keeney‡, Sven van der Meer‡ and Gabriel-Miro Muntean*
*Dublin City University, The Rince Institute, Dublin, Ireland
Email: faisal.zaman@ieee.org, sebastian.robitzsch@dcu.ie and gabriel.muntean@dcu.ie
‡Ericsson, Network Management Lab, Athlone, Ireland
Email: (john.keeney, sven.van.der.meer)@ericsson.com

***Abstract*—Currently the volume of telecom network management data is expanding exponentially, mainly due to the explosive growth in the number of communicating devices along with the increase in heterogeneity of the networks. Such scale of data obsoletes the traditional approach of extracting offline analytics from the network traces governed by some pre-defined schemes. In order to increase the efficiency of the Operations Support System (OSS) and gain in-depth understanding of the generic relationship between network entities, the monitoring data needs to undergo large-scale deep analytics processing. In this paper we present i-MAGNET, an integrated analytics framework developed with the popular real-time stream processing paradigm Storm. The components of i-MAGNET intelligently micro-batch segments of incoming streams to enable high-throughput online analytics of management trace streams. Inter-dependence metrics (temporal and statistical) are exploited to extract contiguous event sub-sequences, which can then be independently examined as part of a network incident analysis system.**

I. INTRODUCTION

Telecom operators are currently facing the dilemma of controlling the spiralling cost of managing the ever-increasing network complexity due to the deployment of more and more Network Elements (NEs) for maintaining high quality communication. Trace streams emanating from these NEs are embedded with event-patterns indicative to network operation and performance status. However, in addition to the granular events making up the interesting patterns, huge volumes of silo and uninteresting events form the main bulk of the data in such streams. This noise and irrelevant data obfuscate the important events (defined as "needles" in this paper) and their patterns to the extent that OSSs and analytics systems struggle to find useful information about network behaviour in these streams. For example Fault Management (FM) data contains many more events and event-patterns that can realistically be classified as "noisy" than interesting events and patterns, i.e. event-patterns capturing anomaly behaviour. In addition to the lack of event veracity, the *volume* of the trace traffic generated by each NE, *variety* of the sources of each event and *velocity* of the trace streams make trace analytics extremely difficult. A pro-active prescriptive analytics system based on event-stream analytics should be able to provide solutions to a particular network incident **in-time**.

For event-stream analytics Event-based Stream Processing (ESP) is a de-facto technique which reduces events through event abstraction, but lacks in scalability [1]. StreamMapReduce tackled the problem of throughput by combining ESP with MapReduce and incrementally processes event-streams [2]. Scalable Advanced Massive Online Analysis (SAMOA) is a stream mining platform to deploy off-the-shelf machine learning algorithms on streaming data [3]. Continuous Stream Mining Engine (CSM Engine) is a Java based Complex Event Processing (CEP) Engine for real-time event correlation finding across multiple events based on pre-defined rules (abstraction), but lacks support for modelling event-patterns. A stepwise heuristic algorithm was employed to detect and remove noise events based on computing co-occurrence statistics of each pair of events to measure the inter-relationship between the events and infer whether events should be included or not [4]. In this context, the goal of the proposed i-MAGNET is to analyze the input event-stream in **near real-time** with minimal loss of pattern-related information so that online and offline pattern discovery and matching are simplified and more effective.

The proposed framework i-MAGNET is designed to integrate event filtering with Storm[1] (a real-time stream processing paradigm) to extract individual event sub-sequences from large volumes of event traces. Such individual self-contained event sub-sequences which must have statistical and temporal commonalities are more effective for Network Management Systems (NMSs) if extracted in real-time. From functional point of view i-MAGNET is composed of two main components, (i) a load balancer, and (ii) a data reducer. The **load balancer** dynamically balances the load of the incoming data via an arrival-rate based data distribution solution known as Kafka [5]. Once the data is loaded, the **data reducer** abstracts the trace stream by applying a metrics-based relation finder and reduces the size of the stream.

The remainder of this paper paper is structured as follows: i-MAGNET is described in detail in Section II, while also presenting i-MAGNET as a parallel algorithm for reducing data by abstracting trace data. Integration of i-MAGNET into an intelligent network management system E-Stream [6] is discussed in Section III. The evaluation of i-MAGNET using artificially generated event sequences is discussed in Section IV. The paper is concluded in Section V.

II. i-MAGNET : A REAL TIME INTELLIGENT ANALYTICS FRAMEWORK

In i-MAGNET the incoming stream is first split into transactions (micro-batches or tuple-batches) based on the event burst-rate, and each tuple-batch is then distributed to

[1] https://storm.incubator.apache.org/

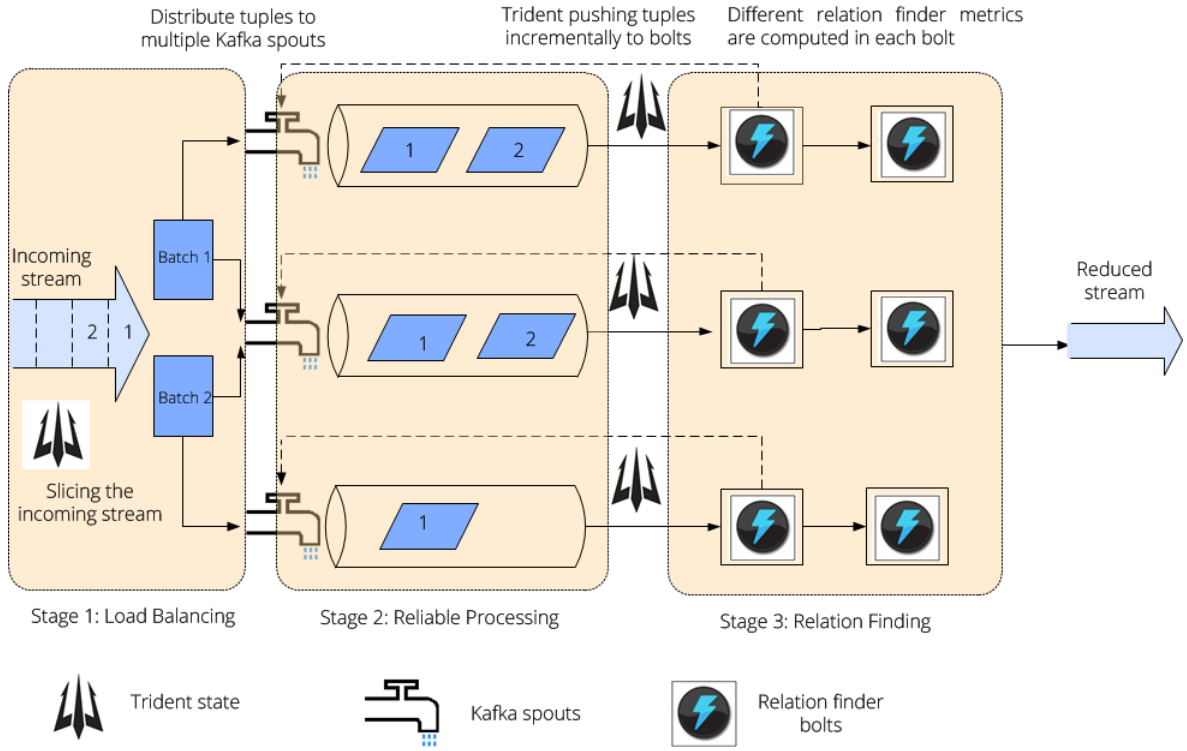

Fig. 1. The functional overview of i-MAGNET

parallel processors to compute the event inter-relationship metrics incrementally. In this way the framework can scale-up for processing large volumes of continuous data in a timely manner. The success of i-MAGNET is underpinned by reliably distributing the stream tuples and accurately identifying the relations between the event instances in each batch. Replicating the distribution and relation-metric computation increases the reliability and fault tolerance of the approach. The i-MAGNET workflow is presented in Figure 1 and consists of three stages: load balancing, reliable data processing, and relation finding. These three stages ensure low latency and higher accuracy, which are the pivotal features of a stream processing model [7]. In summary as a stream processing framework, i-MAGNET is capable of :

- **Rate-based diffusion of streams:** the stream is sliced into tuple-batches and the number of partitions are based on the stream burst rate
- **Fault tolerant processing:** each tuple-batch is replicated to maintain lossless processing of the streams
- **Incremental processing:** the distributed processing of each tuple-batch is pipelined with the relation finding process in order to maintain lower latency of the complete framework

i-MAGNET is implemented with two components: (a) the load balancer, and (b) the data reducer. Kafka generates data processing "spouts" based on the load and distributes the processing to each spout. The abstracted instances make the later task of pattern discovery more precise. The data reducer is designed to work in conjunction with Trident[2], which splits the incoming stream into micro-batches (also defined as tuple-batches) and incrementally executes the relation finder on each distributed micro-batch.

A. Load Balancer

Communication traces are bursty in nature and can exhaust the ingesting capability of the system by straining the I/O and computational resources. Also, in order to reduce the response period to resolve network incidents, trace analytics should be provided to the network manager with minimal delay. The load balancer operates to mitigate these challenges by combining *rate based micro-batching* and *reliable pipeline parallelism* criteria.

1) Rate Based Micro-batching: The underlying concept of rate-based micro-batching is to control the processing load of event-storms by distributing the stream-tuples (events) into micro-batches (or tuple-batches) based on the stream burst rate. It should be noted that the number of micro-batches needs to be increased in case of acute stream bursts (event storms). With this technique, rather than directly using individual stream events, each tuple-batches can contain a large volume of data, which improves throughput through Storm [8] in the next step. In i-MAGNET the stream burst-rate computation and micro-batching is executed by Trident. The advantage of the Trident is that, each tuple-batch is tagged with a separate ID and the operations on each tuple-batch is stateful.

2) Reliable Pipeline Parallelism: To achieve scalability in i-MAGNET each tuple-batch is incrementally distributed and replicated for data reduction. In this way each tuple-batch is replicated several times resulting in greater accuracy in the data reduction process. i-MAGNET uses Kafka to achieve this by distributing the tuple-batches into multiple spouts. To optimize reliability and scalability, the number of the spouts generated should be equal to or larger than the number of tuple-batches

[2]https://storm.incubator.apache.org/documentation/Trident-tutorial.html

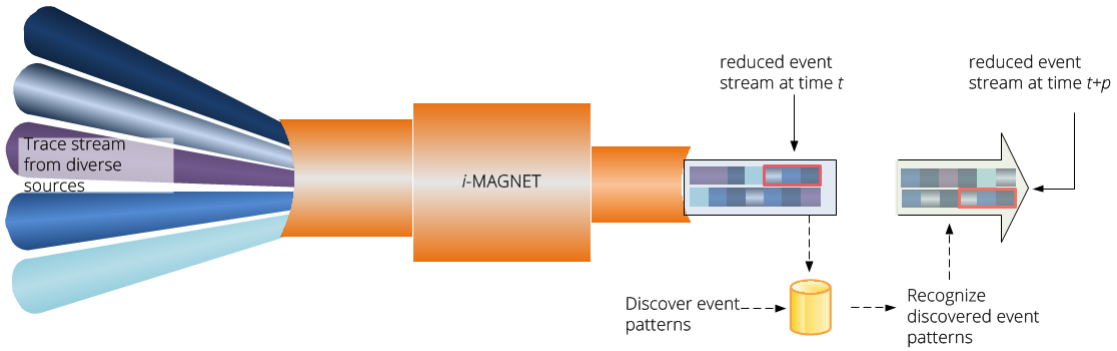

Fig. 2. i-MAGNET in E-Stream

[9]. The other advantage of combining Kafka and Trident is that, increasing the number of tuple-batches increases the consuming rate of each Kafka spout, thus maintaining low latency even in the case of event storms.

B. Data Reducer

Network incidents are usually preceded by sequences of concomitant symptomatic events. The temporal inter-relationships of these events (symptoms, incidents and effects) in different sequences is approximately similar in terms of intra-temporal distance. Therefore statistically, if the similar events appear together in similar temporal sequences, then they should exhibit significantly higher inter-dependence or correlation. On the other hand, other events, such as periodic reporting events or configuration events appear in a standalone manner and are unrelated to other network events, and so show low correlations with other events around them. The task of the data reducer is to analyze the events traces to "find" and "define" the inter-relationships between events in order to detect noisy events of limited value, and then remove these events from the trace stream.

A trace consists of time-stamped sequential events, and an efficient way to leverage off-the shelf analytical techniques is to transform events from the time domain to the frequency domain by examining the frequency of occurrence of each different event type during a time window. The **window** in this context is the size of the tuple-batch received by a single Kafka spout. The reducer then continues by analysing the dependency between the events. In the current implementation events are filtered out based on the degree of inter-dependence, which is computed using an automated threshold-based correlation algorithm. A preliminary version of the algorithm can be found in [4]. Steps of the algorithm are implemented in the following bolts:

- `TransactionSplitFunction`: Split the received events into equal sized transactions
- `OccurrenceCountAggregator`: Count the occurrences of each event within each transaction
- `EventFrequencyAggregator`: Aggregates the frequency of each event over all the transactions
- `CorrelationCalculationFunction`: Calculates the correlation between each pair of events
- `MaxCorrelationCalculationAggregator`: Finds the maximum correlation of each event using Equation 1. $r'_{m_a} \vee r'_{m_b}$ scans through all the pairwise correlation values r_{m_a,m_b} and checks each event pair $< a, b >$ for the maximum correlation coefficient r'_m for each event when all correlations with all other events are considered.

$$r'_{m_a} \vee r'_{m_b} = \max_{\forall m_a, m_b} (r_{m_a,m_b}) \tag{1}$$

- `CorrelationFilter`: Calculates the correlation threshold and filters out events with maximum correlation lower than the threshold.

III. INTEGRATING i-MAGNET IN E-STREAM

E-Stream [6] aims to build a modular system capable of discovering network incidents (patterns) by analysing the telecom network trace data and then using these patterns to predict future network incidents so that corrective actions can be recommended before or as the incident occurs. For accurate discovery of event patterns and associations it is necessary to design a more intelligent data collection mechanism to extract useful information. i-MAGNET forms this part of the tool-chain, so the timeliness of each system is of key importance for the end-to-end approach as a whole.

i-MAGNET provides the functionality of intelligent and scalable data reduction through filtering out redundant events from the trace while preserving the data-information. Removing noise from the raw data stream significantly increases the probability of accurately finding relevant patterns of events in the reduced stream. Also sequential pattern discovery techniques which are inherently resource-intensive can run on the reduced event stream with significantly lesser complexity. Figure 2 illustrates how i-MAGNET is integrated in the E-Stream system to reduce the incoming trace stream and assist in event pattern discovery and recognition.

IV. EVALUATION OF i-MAGNET

This section presents the evaluation of i-MAGNET. The objective is to select potential events by measuring the degree of inter-dependence (correlation) between events. In the current settings, events with lower correlation are considered as 'noise' and filtered out from the stream. The system is evaluated using traces of events drawn from an emulation of mobile telecom control plane communication.

A. Data Preparation

As generic dimensionality reduction techniques generally work on numeric values, it is necessary to translate the information contained in the simulated control plane events

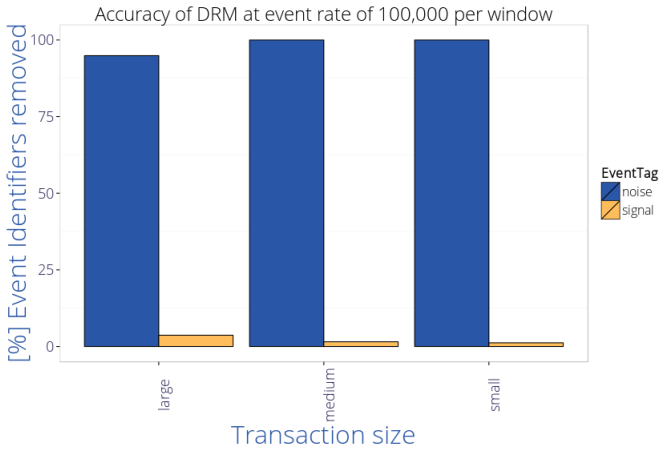

Fig. 3. Accuracy of i-MAGNET in detecting and removing noisy events

into a number format. For this purpose, the open source OpenMSC real-time emulator [10] is used to generate a stream of simulated LTE call set-up and call release control plane event sequences. These simulated sequences include events from various User Equipment (UE), Evolved NodeB (eNB), Mobile Management Entity (MME), Packet Data Network (PDN) Gateway (P-GW), Serving Gateway (S-GW) and Home Subscriber Server (HSS) nodes. OpenMSC was configured to emulate sequences for 10 UEs connected to 5 eNBs, while at the same time generating uncorrelated noise events following a uniform distribution. This results in an Event Identifier (EventID) rate of 206 EventIDs per second on average with a signal to noise ration of 1:6.

B. Experiment Details

In the experiments the correlation algorithm is computed on 100,000 events per window and transactions of different lengths (see Section II-B describing the terms window and transaction). Transactions lengths are varied to check the sensitivity of the noise reduction. The transaction lengths are derived from the frequency distribution of the events to maintain a balanced spectrum of events in each transaction. As can be seen from Figure 3, for small and medium transactions i-MAGNET successfully removed all noisy events, however, for larger transactions it incorrectly removes a higher portion of interesting correlated events. It should be noted that the trade-off with smaller transactions, that is with smaller transactions the correlation matrix needs to be calculated over higher number of samples causing higher computational complexity.

In smaller transactions the spiky occurrences of correlated events (co-occurring in large numbers) can be well separated from the unimportant noisy events. With larger transactions the record of the noisy events increases, causing to score higher correlation.

V. CONCLUSION

This paper introduces an intelligent scalable dimension reduction framework i-MAGNET for identifying and extracting interesting sequences of frequent concomitant events from high-volume event streams. One of the key features of the framework is its reliable and scalable distributed approach to accurately detect isolated unimportant events and remove them from the event stream. The framework implemented in Storm first partitions the event stream into tuple-batches and then enables multiple copies of the tuples to be processed in parallely in order to maintain scalability and reliability of the data analysis process. To achieve even lower latency the sharing process of the tuple-batches is pipelined with the computation of the relation finding process. A brief evaluation of i-MAGNET was conducted to demonstrate the accurate detection of unimportant events, leaving only important monitoring events which impact network performance. Experimental results show that the framework is capable of reducing significant portions of the event stream and at the same time keeping the events with potential correlative structure.

From operational point of view a dynamic resource sharing program can increase the capacity of i-MAGNET significantly. Functionally, periodic event (silo) filtering and temporally homogeneous distanced event clustering should increase the accuracy of faulty event-pattern finding as the output of i-MAGNEt is utilised by sequential pattern finding algorithms. Moreover, adaptive learning (concept drift) capability from evolving network scenarios shall increase the reliability of i-MAGNET further.

ACKNOWLEDGMENT

This work was funded by Enterprise Ireland Innovation Partnership Programme with Ericsson Ireland under grant agreement IP/2011/0135 [11].

REFERENCES

[1] L. Golab and M. T. Özsu, "Issues in data stream management," *ACM SIGMOD Record*, vol. 32, no. 2, pp. 5–14, Jun. 2003.

[2] A. Brito, A. Martin, T. Knauth, S. Creutz, D. Becker, S. Weigert, and C. Fetzer, "Scalable and Low-Latency Data Processing with Stream MapReduce," in *2011 IEEE Third International Conference on Cloud Computing Technology and Science*. IEEE, Nov. 2011, pp. 48–58.

[3] G. De Francisci Morales, "Samoa: A platform for mining big data streams," in *Proceedings of the 22Nd International Conference on World Wide Web Companion*, ser. WWW '13 Companion. International World Wide Web Conferences Steering Committee, 2013, pp. 777–778.

[4] F. Zaman, S. Robitzsch, Z. Wu, J. Keeney, S. van der Meer, and G. Muntean, "A heuristic correlation algorithm for data reduction through noise detection in stream-based communication management systems," in *2014 IEEE Network Operations and Management Symposium, NOMS 2014, Krakow, Poland, May 5-9, 2014*, 2014, pp. 1–8.

[5] J. Kreps, N. Narkhede, J. Rao *et al.*, "Kafka: A distributed messaging system for log processing," in *Proceedings of the NetDB*, 2011.

[6] F. Zaman, G. Hogan, S. van der Meer, J. Keeney, and G. Muntean, "A recommender system architecture for predictive telecom network management," *IEEE Communications Magazine*, 2014, accepted.

[7] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and Issues in Data Stream Systems," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS '02*. New York, New York, USA: ACM Press, 2002, pp. 1–16.

[8] P. Goetz and B. O'Neill, *Storm Blueprints: Patterns for Distributed Real-time Computation*. Packt Publishing, 2014.

[9] N. Marz and J. Warren, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*. Manning Publications Company, 2014.

[10] S. Robitzsch, "OpenMSC - An Open Source MSCgen-Based Control Plane Trace Emulator for Communication Networks," 2014. [Online]. Available: http://openmsc.blogspot.com

[11] Dublin City University and Ericsson, "E-Stream Project." [Online]. Available: http://estream-project.com