Abstract—Currently the volume of telecom network management data is expanding in exponential scale, mainly due to the explosive increase in the number of communicating devices along with the increasing 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 gaining in-depth understanding of the generic relationship between network entities, the monitoring data needs to undergo large-scale deep analytics. 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 subsequences, 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-expanding network complexity due to the deployment of more and more Network Elements (NEs) to maintaining high quality communication for consumers. The main management issues for operators are: the volume of the trace traffic generated by each NE, the variety of the sources of the data, and finally the velocity of the trace stream arrival. These issues make it extremely complex to effectively analyse the data in a timely manner.

Many operational decisions are hard coded in Operations Support System (OSS) application to respond to ‘severe’ incidents with higher priorities, however, with increasingly dynamic and changing network environments these decisions fail to evolve. In order to analyse massive amounts of management data in an efficient and timely manner an integrated analytics system is required. The system should be capable of spreading the load of the high velocity continuous data for controlled processing of the data. It should also be able to extract meaningful information by abstracting from the low-level trace information by reducing noisy and redundant information and aggregating dependent events. In this way it should be possible to efficiently and effectively harness such massive data to extract real-time actionable intelligence about the managed network.

The proposed framework \(i\)-MAGNET, is designed to integrate event filtering with Storm\(^1\) (a real-time stream processing approach) 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, and must be extracted in a real-time manner. 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\(^2\). Kafka generates data processing “spouts” based on the load and distributes the processing to each spout. 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 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.

An Event-based Stream Processing (ESP) approach was selected to reduce events\(^2\), while maintaining scalability and low latency. A step-wise heuristic algorithm\(^3\) was employed to detect and remove noise events, while still maintaining important events. This algorithm is based on computing co-occurrence statistics for each pair of events to measure the inter-relationship between the events and infer whether events should be included with each other or not.

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\(^4\) is discussed in Section III. The evaluation of \(i\)-MAGNET using artificially generated event sequences is discussed in Section IV, before the paper concluded in Section V.

II. \(i\)-MAGNET: A Real-time Intelligent Analytics Framework

Telecom trace streams emanating from the NEs contain information on the operation of the network, and embedded in this data is patterns of events indicating network operation and performance. However, in addition to the granular events that make up these interesting patterns, huge volumes of noise and

\(^1\)https://storm.incubator.apache.org/

\(^2\)https://storm.incubator.apache.org/documentation/Trident-tutorial.html
uninteresting events form the main bulk of the data in these streams. This noise and irrelevant data obfuscate the important events and their patterns to the extent that traditional OSS systems and analytics systems struggle to find useful information about network behaviour in these streams without extensive filtering informed by expert domain knowledge. 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 interesting anomaly behaviour. The goal of i-MAGNET, is to reduce the input data stream with minimal time delay and with minimal loss of pattern-related information, so that online and offline pattern discovery and matching become simplified and more effective.

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 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 [5]. 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. The following sections illustrate the functionality and workflow of these components.

### 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 [6] 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.
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 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 [3]. 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.

\[
r'_{m_a} \lor r'_{m_b} = \max_{m_a, m_b} (r_{m_a, m_b})
\]  

\( < a, b > \) for the maximum correlation coefficient \( r'_{m_i} \) for each event when all correlations with all other events are considered.

- **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 [4] aims to build a modular system capable of discovering network incident patterns by analysing the telecom network trace data, and then using these patterns to predict future network incidents so that alleviating 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 only. This system, \( i\)-MAGNET, forms part of this 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 by 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. Consequently, running resource-intensive pattern discovery techniques on the filtered reduced stream is computationally much cheaper than from raw trace stream. 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 into a number format. For this purpose, the open source OpenMSC real-time emulator [8] is used to generate a stream of simulated
i-MAGNET for identifying and extracting interesting sequences of frequent concomitant events from high-volume data streams. One of the key features of the framework is its reliable and scalable distributed approach to perform with accuracy detection of isolated and periodic events. These isolated and periodic events can then be treated as noise events and removed from the data stream to ease later processing, for example to mine the stream for patterns containing important incidents. The incoming stream is partitioned into tuple-batches and multiple copies of the partitioned data are processed in parallel to support both increased 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 relation finding process. The i-MAGNET framework is realized following the stream computing paradigm Storm; with the parallelization and incremental computation of tuple-batches modified to increase the performance of the framework.

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 large input data stream and at the same time keeping the data fragments with potential correlated structure. Within E-Stream, i-MAGNET enhances the effectiveness of the other algorithms that discover and match patterns in network management events by providing only the important event sequences. In this way the i-MAGNET approach is able to partly deliver on its goal making better use of the huge volumes of under-exploited management data available in modern telecom networks, in particular assisting in pattern discovery to enable real-time stream-based predictive analytics of the network conditions.

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 data streams. One of the key features of the framework is its reliable and scalable distributed approach to perform with accuracy detection of isolated and periodic events. These isolated and periodic events can then be treated as noise events and removed from the data stream to ease later processing, for example to mine the stream for patterns containing important incidents. The incoming stream is partitioned into tuple-batches and multiple copies of the partitioned data are processed in parallel to support both increased 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 relation finding process. The i-MAGNET framework is realized following the stream computing paradigm Storm; with the parallelization and incremental computation of tuple-batches modified to increase the performance of the framework.

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 large input data stream and at the same time keeping the data fragments with potential correlated structure. Within E-Stream, i-MAGNET enhances the effectiveness of the other algorithms that discover and match patterns in network management events by providing only the important event sequences. In this way the i-MAGNET approach is able to partly deliver on its goal making better use of the huge volumes of under-exploited management data available in modern telecom networks, in particular assisting in pattern discovery to enable real-time stream-based predictive analytics of the network conditions.

ACKNOWLEDGMENT

This work was funded by Enterprise Ireland Innovation Partnership Programme with Ericsson Ireland under grant agreement IP/2011/0135 [9].

REFERENCES