Magnet: Real-time Trace Stream Analytics Framework for 5G Operations Support Systems

Sebastian Robitzsch, Faisal Zaman, Sven van der Meer, John Keeney and Gabriel-Miro Muntean

***Abstract*— The era of petabyte data has arrived as the digital big data universe continues its expansion towards exascale with massive volumes of data generated by diverse distributed sources. The size of big data makes it very difficult gaining insight into the data meaning. In industrial applications, in order to explore both data meaning and the complex relationship between data components, big data needs to be processed and reduced enabling further deeper analysis in a timely manner. In this paper an integrated data analytics framework is presented designed to extract the set of instances exhibiting statistical dependency from massive volume of data in a pre-defined quasi real-time manner. The parallel computing model of MapReduce is enhanced to realise *Magnet*. The solution presented in this paper is applicable to the telecommunications market where it optimises next-generation network management systems for heterogeneous radio access technologies.**

***Index Terms*—Operations Support System, Big Data, 5G, Real-time, MapReduce, Stream Analytics, Pattern Matching, Data Mining**

I. Introduction

THE continuous increase in the number of mobile devices and users on one hand and advent of ubiquitous communication technologies, development of innovative networking applications and user demand for high quality rich media content on the other hand are behind the latest massive amounts of data generated and exchanged, whose volume continues to grow at an exponential pace. The astounding increase in the volume of data generated is shown for instance by the fact that the amount of information created from the dawn of human civilization to 2003 (i.e., 5 exabytes) is now generated in just a couple of days. Additionally the data digital universe is experiencing a two-fold expansion every two years since 2012 so that the annual global IP traffic is expected to exceed the zettabyte (1000 exabytes) threshold by the end of 2016, and reach 1.6 zettabytes per year by 2018 according to forecasts by leading industry forums. This huge amount of information (lately labelled as “data explosion”, derived from “big data”) leverages myriad opportunities for its analysis and use. However such exploration poses significant challenges, mostly as this massive volume of data, represented by heterogeneous and diverse dimensionality data components, is too big for the processing capacity of the traditional analytical tools. So far, in order to be able to cope with this influx of big data business enterprises constantly scale-up the analytic performance by employing incremental upgrades to the existing solutions. However, these solutions have severe limitations, especially in terms of handling streams of big data transferred over large bandwidth networks and originating from multiple sources and therefore proposal of novel approaches is required.

Such big data is usually bursty in nature and can be handled by spreading the incoming data into multiple windows; this will also facilitate distributing the data processing. Alternatively, based on the data incoming rate, resources can be allocated dynamically [1]. Variability in the data is also caused by larger proportion of irrelevant, redundant and noisy information coming from various sources. This in turn makes very difficult the extraction of meaningful knowledge from the data within a limited time. Simple light weight data projection techniques can serve the purpose of reducing the data dimension, but is capable of handling limited volumes of data only. Parallelizing the projection can solve the problem, but there is a possibility of higher approximation error. Event-based Stream Processing (ESP) is advantageous for reducing the number of events (which are distinct data instances labelled as such by the operators of ESP) with considerable low latency, but the scalability is constrained [2].

This paper addresses these challenges by proposing *Magnet*, an integrated scalable data analytics framework designed to extract the set of instances exhibiting statistical dependency from large amounts of data in a given time period. *Magnet* consists of two major components: a load balancer and a data reducer. The load balancer dynamically balances the load of the incoming data via an arrival rate-based adaptive window solution. Once the data is loaded, the data reducer approximates the data by applying an on-the-fly parallel random projection technique and finding correlated instances. Correlated instances do not necessarily imply a pattern of instances, but make the task of pattern discovery more precise. The data reducer is designed to work in conjunction with MapReduce (MR) [3] a ubiquitous parallel computing paradigm to process large data. Basically MR performs its tasks in batches with high latency, while *Magnet’s* low latency is the result of using micro-batches of the input data and pipelining the projection and correlation computation jobs.

This paper has been submitted for review on 01 June 2015. The work described was originally funded by Enterprise Ireland Innovation Partnership Programme with Ericsson Ireland under grant agreement IP/2011/0135.

S. Robitzsch is with InterDigital, Ltd., London, United Kingdom (e-mail: sebastian.robitzsch@interdigital.com).

F. Zaman is with Adaptive Mobile Security, Dublin, Ireland (e-mail: faisal.zaman@adaptivemobile.com).

S. v. d. Meer and J. Keeney are with Ericsson, Athlone, Ireland (e-mail: john.keeney@ericsson.com, vdmeer@ieee.org).

G.-M. Muntean is with Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Glasnevin, Dublin 9, Ireland (e-mail: gabriel.muntean@dcu.ie).

Our research is focused specifically on the telecoms industry as it has dealt with large amounts network data for decades and is preparing to cope with big data, mostly due to the unprecedented rise in network control information generated by the next generation mobile networks. The purpose of *Magnet* is to serve specific business needs, making the big telecom network data a "service" rather than a "technique" by integrating the analytical results to pattern discovery to enable predictions of the network scenarios and respective solutions. *Magnet* performance is tested on artificially generated telecoms network trace data. The input consists of an online stream generated by the in-house developed emulator OpenMSC[1]. The contribution of this work can be summarized as follows:

- Developed a parallel algorithm for scaled and refined approximation of data
- Developed an analytical framework *Magnet* capable of handling streams of data
- Evaluated *Magnet* on abstracting the symptomatic events leading to cell congestion from an artificially simulated trace file.

The principle of how *Magnet* works and where the MP design has been improved are described in Section II. This is followed by describing the realization of *Magnet* in the E-Stream architecture in Section III. The evaluation of the integrated *Magnet* solution is presented in Section IV followed by a discussion on how *Magnet* fits in the context next generation Operations Support Systems (OSSs). The work is concluded in Section VI.

II. The *Magnet* Algorithm

The success of *Magnet* relies on its ability to scale the processing of massive data volumes, approximate the data with high accuracy and introduce a limited time delay. In *Magnet*, parallelizing the process of approximation and correlation computation is the source of scalability; replicating the parallelization process incurs minimal approximation error and adaptive slicing of the stream integrated with pipelining the processes facilitate stream processing.

The workflow of *Magnet* is presented in Figure 1 and consists of three stages: dynamic load balancing, refined data approximation, and correlation finding. These three stages ensure low latency and low error, which are the pivotal features of a stream processing model [4]. In summary *Magnet* is capable of:

- Rate-based diffusion of stream: *Magnet* splits the stream based on the arrival rate and spreads the split stream among smaller slices.
- Fault tolerant processing: *Magnet* replicates each slice of the split stream to maintain better approximation accuracy.
- Incremental processing: *Magnet* supports pipeline parallelism of the tasks.

Conventional data approximation techniques incur significant number of distortions while projecting data points onto lower dimensions. In *Magnet* the Data Approximation (DA) was realised using MapReduce where data partitioning and re-processing of each partition has been optimised, as described with further details in [5]. This optimisation involves an extension of the approximation process by processing each partition multiple times according to statistical theory. Consequently, the approximated results will be more accurate. This is why the chosen MapReduce approach can be described as enhanced.

In *Magnet*, DA approximates a micro-batch of streams and starts correlation finding on that micro-batch rather than waiting for the whole stream-batch to be processed. Sequential processing is at the core of this hybridisation of batch processing and real-time processing. This design fully serves our objective of maintaining low latency and high throughput processing. Spark-based stream processing is similar to the proposed approach where the stream is split into discretised small batches but processed in-memory which allows a more interactive and faster processing of batches by design; however, at the time E-Stream started MapReduce still provided a much more stable environment compared to Spark which was key towards a future integration into Ericsson's xStream system [6]. Thus, MapReduce was favoured over Spark.

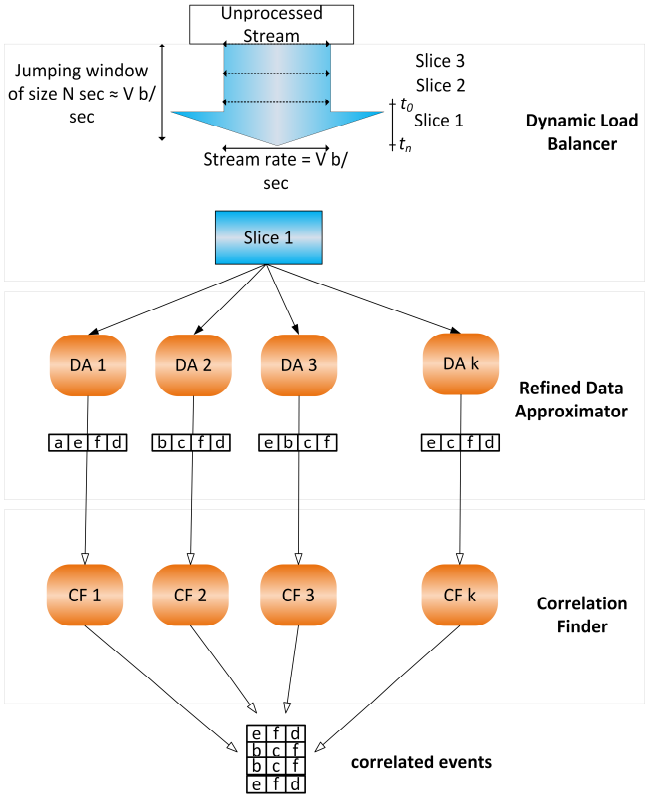

Figure 1: Internal workflow of *Magnet*

[1] OpenMSC is available at www.github.com/seronline/OpenMSC

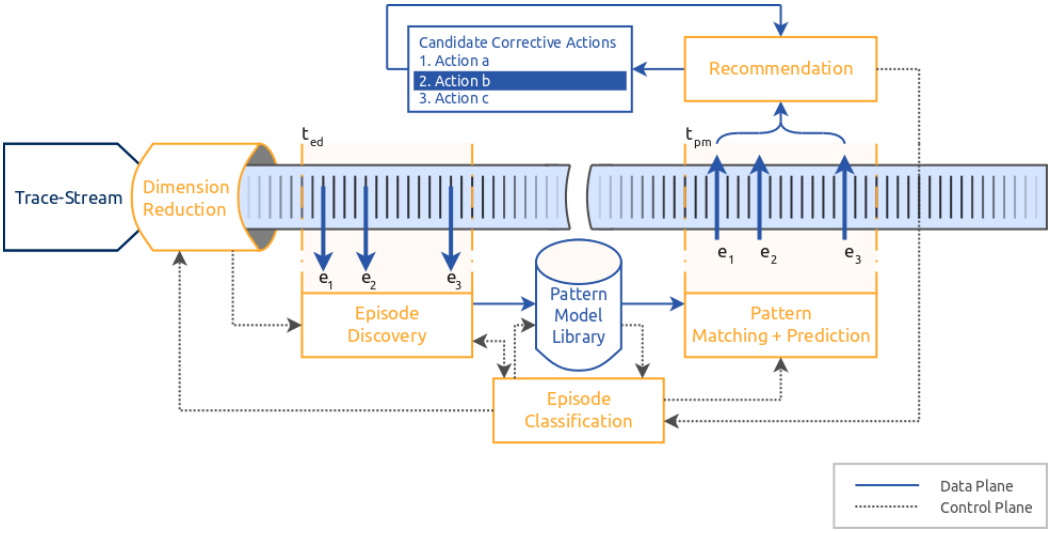

Figure 2: System design of the E-Stream project

A. *Dynamic Load Balancing*

The concept of dynamic load balancing is based on controlled processing of the incoming data stream. This entails feeding the processors with a manageable data volume, while the I/O is confronted with the event storm. In this mechanism the incoming data volume is controlled adaptively with the data arrival rate. In *Magnet* the data arrival rate (also defined as stream burst rate) is estimated first and based on that the window length is derived (which is the volume of incoming data stream), leading to seamless interaction between streaming rate and the capacity of the processors (i.e., buffer size).

B. *Refined Data Approximation*

The purpose of Refined Data Approximation (RDA) is to increase the efficiency of load balancing by further reducing the events, while maintaining lower estimation errors. RDA proceeds on-the-fly and therefore has no storage requirements. The theoretical basis of RDA is underpinned by the seminal Johnson-Lindenstrauss (JL) lemma [7].

In event-based streams distinct events exist as inputs rather than continuous values and this requires a careful re-configuration of the approximator. Random hash-based indexing is utilised to project events to lower dimensions. To maintain the on-the-fly processing of the input events we simply recompute the entries by Gaussian random hashing. Following the JL lemma and the law of large numbers we repeat the on-the-fly approximation several times and parallelize the whole procedure to scale the processing performance. The underlying idea is to partition the data and run the data approximation several times on each partition, and in this way we are replicating the projection several times to refine the approximation.

C. *Data Intensive Correlation Finder*

Correlation finder focuses on identifying any correlation between the data instances in order to capture the dependency relation between the instances. This correlative structure between the events can be detected by analysing the eigen-values of large dimensional random matrices and as a 'by-product' the events with poor spectral condition can be filtered out. These events can be ignored, as noisy events. This technique is more data intensive in a sense that it analyses the eigen-space (spectrum) of the covariance matrix of the observed event set and identifies eigen-states coming from random noise using the known eigenvalue distribution of random matrices. This results in a decomposition of the covariance matrix: a part containing useful information from events with potential correlative structure and another part capturing the random noise. We designed the correlation finder to work with RDA in a pipeline parallel framework. In this framework the correlation finder starts processing the projected data from RDA before all the data is projected. A more detailed description and evaluation of the correlation finder can be found in [5].

D. *Realisation of the Enhanced MapReduce*

Magnet aims to constitute a modular system capable of discover network events patterns by analysing the telecom trace data and predict network incidents from the event patterns to provide corrective actions. For accurate discovery of event associations it requires designing a more intelligent data collection mechanism to extract useful information only. And for responding to network incidents pro-actively, this information should be forwarded to the next working modules within minimal time delay. In E-Stream, *Magnet* provides the functionality of smart and scalable data collection.

MapReduce (MR) is a special data-based programming model which has been established as a standard practice for processing large amounts of data in parallel. It includes two major phases: *map*: in which all the input data is transformed by a single argument in parallel and *reduce*: in which all the transformed input data is grouped based on some multiple arguments. The niche of MR usage in parallelisation and scalability is the consequence of statelessness of the mapping algorithms (mappers) and independent processing of different reducing solutions (reducers). The executions of the tasks take place sequentially, which limits the scope of MR in terms of processing continuous large scale data (stream). If the data going through MR is kept reasonably small and processing is

done incrementally, MR handles streaming data with reduced delay response. Following this observation, the basic workflow of MR was enhanced to enable *Magnet* processing of event streams by micro splitting the trace stream described next.

E. Micro-splitting of the Trace Stream

Stream micro-splitting starts with slicing the incoming data stream into small batches which are computed based on buffer size (processing power) and then the data collected in each batch is submitted to the MR. These small batches are part of the jumping windows in which the results of the batches are accumulated. We consider the deployment of 'jumping' windows instead of the 'sliding' windows as the latter requires more processing power for accumulating the outputs. The length of the jumping window is controlled dynamically according to the stream burst rate and the size of the slices depends on the minimal data volume the mappers should accumulate before sending it to the reducer (which actually performs the processing). The advantage of such micro splitting of streams is that after receiving a specific time slice from every mapper, the reducer starts the combination process and merges the result with the previously merged results.

F. Incremental Processing

One of the sources of high latency in MR is that the commencement of one process needs to wait for finishing of the process started earlier. The enhanced MR includes a modification of the basic MR functionality to perform parallel process execution. Micro splitting the stream facilitates pipelined parallelism between the components of *Magnet*; this procedure can also be defined as incremental processing. In this improved solution, the reducer does not need to wait until the map phase finishes the task. The reducer needs to compute the aggregated slice value only after receiving the data corresponding to the same slice from all mappers. After this is performed, it calls the user-defined merge() function to merge the slice results with the jumping window results.

III. INTEGRATING MAGNET IN E-STREAM

E-Stream aims to constitute a modular system capable of discover network events patterns by analysing the telecom trace data and predict network incidents from the event patterns to provide corrective actions. For accurate discovery of event associations it requires the design of a more intelligent data collection mechanism to extract useful information only. And for responding to network incidents pro-actively, this information should be forwarded to the next working modules within minimal time delay. In the E-Stream context, *Magnet* provides the functionality of smart and scalable data collection, processing and pattern identification of any root-cause relationship possibly available in the trace stream. The presented framework follows the E-Stream system design, as described in [8, 9]. E-Stream defined the five modules:

- Dimension Reduction Module (DRM)
- Episode Discovery Module (EDM)
- Episode Classification Module (ECM)

TABLE I
STRUCTURE OF A SINGLE EVENT IDENTIFIER (EVENTID) WITH TOTAL LENGTH OF 19 DIGITS

Field	Numeric Length
Source Network Element	5
Destination Network Element	5
Protocol Type	2
Primitive Name	2
Information Element	2
Value	3

- Pattern Matching Module (PMM) and
- Recommender System Module (RSM)

Magnet combines DRM, EDM, ECM and PMM in a single solution which can be directly mapped to the architecture defined in E-Stream. As depicted in Figure 3, the trace stream is first minimised in spatial size to reduce the computational complexity of discovering episodes (sequence of EventIDs that indicate potential patterns of interest) in the next step which are then further classified according to their correlation.

The illustrated pattern model library in Figure 2 represents the linking storage element between the episode discovery and the future pattern model matching mechanisms akin to a commonly known relational database implementation.

IV. EVALUATION

A. Emulation Set-up

In order to test and evaluate *Magnet* two high end 24 core servers were utilized with 128 GB RAM each running Ubuntu server 12.04. In order to guarantee a scalable environment where no single module is permitted to utilise all computational resources available on the physical machine, DRM, ECM, PMM and RSM were set up in a virtualised environment using Kernel Virtual Machines (KVMs).

The integrated E-Stream test-bed is only as good as the environment in which it is presented. Without a real continuous stream of trace data, the implemented modules would not be able to be tested against their scalability and the real-time requires. That is why OpenMSC was developed: a highly configurable C-based emulator which generates a stream of integer numbers, denoted as EventIDs, that represent the control plane communication data of a telecommunication network. OpenMSC allows specifying a single success use case, e.g., the set-up of a call from a mobile phone, and an arbitrary number of failure use cases[2]. Each use case is referred to as a communication description with communication descriptors being represented by integer numbers. More precisely, a descriptor denotes a single primitive exchange between a source and a destination NE and is part of an entire use case. Multiple communication descriptors are referred to as a communication description (the entire use-case). In comparison to discrete or continuous event simulators which produce a sequence of events using an internal simulator clock to model the system, OpenMSC generates each EventID based on the actual system time.

[2] See the appendix for a detailed message sequence chart

All communication descriptors are specified in an MSC file which follows the same terminology as the mscgen[3] tool to plot MSCs. The concept of OpenMSC is based on the assumption that each User Equipment (UE) in the network follows the same control plane procedures, as specified in advance through a MSC configuration file. Additionally, each UE repeats this action within a given time-frame. OpenMSC lets the researcher set parameters such as number of UEs and number of Base-Stations (BSs) in the network. As mentioned, a particular communication descriptor is represented as a single numeric EventID due to the requirement of most data mining algorithms which are able to work with numbers only. Therefore, OpenMSC translates each communication descriptor entity into a single unique integer representation, then concatenates these representations in a standardised way, as shown in Table 1.

Each communication descriptor has a source NE and a destination NE which exchange information using primitives that are standardised by 3GPP. As primitive names can be the same across multiple protocols, OpenMSC distinguishes between the protocol type and primitive name. Each primitive has various Information Elements (IEs) which hold particular values. The total length of the generated EventID is 19 due to the maximal length of the data type unsigned long long in all modern programming languages. The source and destination NEs are represented by a five digit long integer values. The integer representations are being allocated on an iterative basis where an unknown NE will receive a number which is incremented by one, when compared to its predecessor NE. The only exceptions are UEs and BSs, as they are treated differently by OpenMSC. In case of a BS, OpenMSC calculates a BS EventID ID_{bs} by:

$$ID_{bs} = 100 \cdot BS(m) \quad (1)$$

with $m \in Z$ and $m > 0$. If the total number of BSs, n_{bs}, equals 50 the five digit long numerical representation of the second BS ($m = 2$) is 00200. The numerical UE representation is calculated as follows:

$$ID_{ue} = 100 \cdot BS(m) + UE(n) \quad (2)$$

with m and $n \in Z$ and m and $n > 0$. For instance, let $m = 2$ and $n = 45$, the corresponding UE identifier ID_{ue} would be 00245. All other NEs receive a unique integer in the range of 1 through 99. This ensures that the five digit long numerical representation of a NE is always unique.

The only limitation by generating the EventID as described above is the total number of NEs, primitives and IEs that can be used in the entire emulation while ensuring a unique integer representation for every piece of information. However, with 999 BSs, 99 UEs at each BS, and further 99 NEs (not BS or UE), it is ensured that OpenMSC still provides enough flexibility to generate data-streams emulating rather large networks.

The MSC used for this example is shown in Figure 6 (Appendix) which comprises the two communication descriptions Success and Failure which consist of nine and two communication descriptors, respectively. There are three NEs chosen, and two protocol types, i.e., Radio Resource Protocol (RRC) and S1 Application Protocol (S1AP). The configuration used in OpenMSC to generate the data stream in the testbed is given in the Appendix section too. When providing the two input files, as given in the Appendix, OpenMSC generates an EventID rate per second of approximately 210. The stream is filled up with random EventIDs representing noise; the generation of noise EventIDs follows a normal distribution.

[3] The open source tool is available at http://www.mcternan.me.uk/mscgen

B. Results

This section presents the results of the E-Stream integrated test-bed where Magnet has been fully implemented using the modular E-Stream system described in Section III and the algorithmic approach from Section II.

1) Stream Reduction

In the emulation the dimension reduction algorithm is computed on 100000 events per window and transactions of different lengths. Transactions lengths are varied to check the sensitivity of the noise reduction, as described in [5]. 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 the dimensionality reduction successfully removes 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 is that 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.

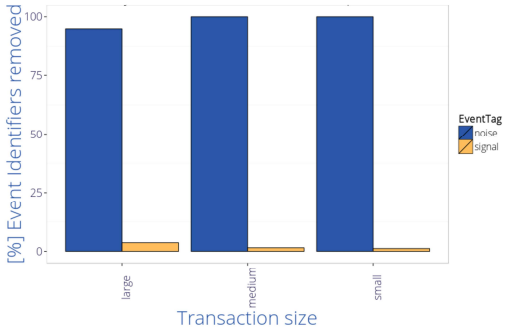

Figure 3: Accuracy of detecting and removing noisy events from trace stream

2) Detecting Episodes

The detection of sequences of EventIDs (i.e., episodes) is the next step in the trace analysis. The emulated use case consist of 20 different episodes pre-defined in the OpenMSC stream data out of which 10 are successful call set-ups and releases and 10 failure cases. The successful episodes consist of 37 unique EventIDs and the 10 failure pattern models of 27 unique EventIDs. Note, episodes that form either the full success or failure case are denoted as pattern models.

Figure 4 depicts the result of the episode discovery process where the dark blue colour represents the success pattern models and the blue colour represents the failure pattern

models. The discovered episodes are displayed with orange dots. After repeatedly running the episode discovery all the pre-defined pattern models including success and failure pattern models can be found in the collection of episodes cumulatively discovered and identification by their correlation. The results presented in Figure 4 indicate that this solution is capable of discovering the pattern models in an effective manner.

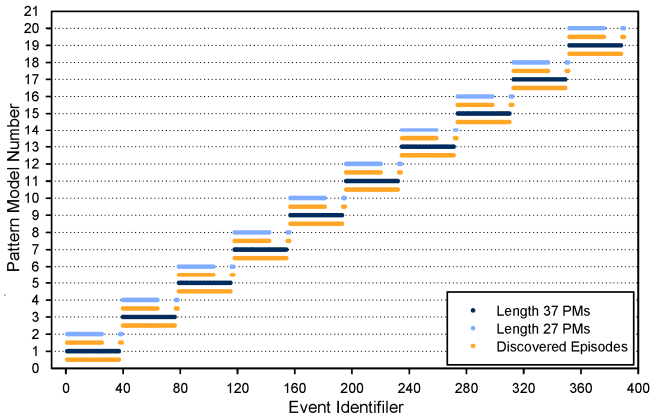

Figure 4: Successful discovery of all episodes (pattern models) defined in OpenMSC

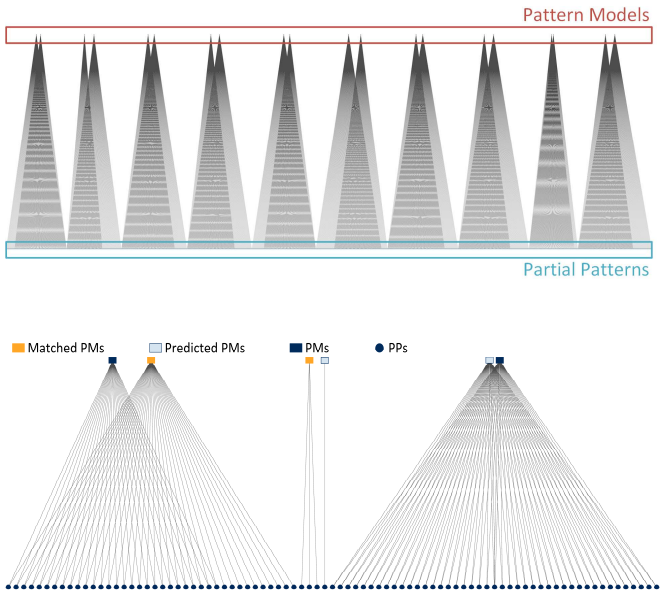

Figure 5: Pattern tree of classified episodes (a) and visualisation of the matched and predicted pattern models in the data stream (b)

3) Episode Classification

The objective of the real-time classification of discovered episodes in the previous step is presented here. The classification of all discovered episodes of all lengths utilises the graph theory theorem of building an acyclic unidirectional tree of episodes. The top level episodes (pattern models) are the top of the pyramids in Figure 5.a, while the next level of episodes (partial patterns) of the acyclic graph are displayed below and connected with a black line.

The discovered pattern model tree is then populated in the pattern model library which is the input to the pattern matching task. It can be reported that when scaling up the stream rate, the processing of thousands of episodes in order to build the pattern tree was always finished in real-time utilising. It can be concluded that all pattern models and their corresponding partial patterns were discovered by *Magnet*.

4) Pattern Matching

This section presents the last step of the *Magnet* framework: the matching of the discovered pattern models in the online trace stream. The objective is to match and predict the pattern models from the reduced event streams on the basis of the pattern model information provided by the pattern model library. The matching approach is evaluated using traces of events drawn from OpenMSC, the E-Stream emulator to generate a continuous stream of mobile telecom control plane communication events. For visualisation purposes the matching function is designed to show matched and predicted pattern models only together with their corresponding first level partial patterns. The prediction itself is based on the assumption that if a non-top level pattern model has been found, it must have been actually there and was probably only lost due to the slicing of the stream *Magnet* must undertake. The visualiser, depicted in Figure 5.b illustrates orange squares for matched pattern models, pale blue squares for predicted pattern models, dark blue squares for pattern models available in the patter model library and dark blue circles for Level 1 pattern models.

V. Discussion

This last section aims at putting the conducted research into perspective addressing questions around applicability, scalability and ease of integration into existing OSSs. For this discussion the main objective of *Magnet* is of significant importance, i.e.: the system should be agnostic to the use case in which it should discover root-cause relationships of existing and potential misbehaviours. With this objective in mind, *Magnet* (the final solution of the E-Stream project) is a fundamental step forward, as part of Ericsson's global R&D initiative in the area of management of next generation OSSs, being ahead of the state of the art in this space. *Magnet*, as result of a collaborative Dublin City University-Ericsson R&D effort, complements Ericsson's industrial-driven *xStream* [6] approach, which is currently embedded in a new enterprise network management product.

While the existing Magnet system as a research prototype might run into performance issues, Ericsson has viable software that can use and deploy the *Magnet* ideas in a commercial, scalable and high-performance environment. The related Ericsson works [6,11,12] focus on real-time evaluation of large-scale data streams demonstrating that the *Magnet*'s objectives of designing a solution which is agnostic to the actual stream's content while persistently discover root-causes is feasible. To this extend the choice of MapReduce as the agnostic stream processing platform on which *Magnet* has been realised can be explained by a rather pragmatic software engineering viewpoint which prefers platform stability and integrablity as long as the overall objectives can be met.

VI. Conclusion

This paper introduces an innovative integrated scalable data analytics framework *Magnet* proposed for extracting the correlative structure between events (data instances). One of the key functionalities of *Magnet* is to reduce data through a refined approximation process which is based on randomization and rough preservation of the statistical relationship between data instances. The approximation process is applied to multiple copies of partitioned data in order to support both increased scalability and high accuracy of the data approximation process. This also results in faster correlation computation. Further scalability is introduced by means of pipelining the approximation and correlation computation processes. *Magnet* framework is realized in the MapReduce model proposed for handling massive amounts of data. Fundamental enhancements are proposed for MapReduce to handle pipelining of the two processes and process continuous flows of big data (streams). *Magnet* evaluation was performed and the experimental results show that the framework is capable of reducing significant portion of the large input data stream and at the same time keeping the data fragments with potential correlative structure. Removing the bulk of the events through randomization and then keeping the set of events linked through statistical or temporal dependency, has enabled the reduction at such scale. The processing time of the framework is highly encouraging and recommends its utilisation as a continuous analytics system. From an application point of view *Magnet* enhances the effectiveness of the algorithms looking for patterns in data by providing only the correlated instances. The other aspect of the *Magnet* is that it can also be implemented in any stream computing model for building a real-time stream analytics system.

VII. Appendix

For completeness of this publication, the authors provide the configuration file of OpenMSC so that interested follow up readers can benchmark their solution against E-Stream:

```
openmscConfig: {
  numOfBss = 1;
  numOfUesPerBs = 10;
  ueActivity-Dist = "exponential";
  ueActivity-Dist-Lambda = 0.2;
  cdOverlap = false;
  informationElements = ( {
      ieName = " SIRErrorValue";
      ieDist = "gaussian";
      ieDistMu = "80.0";
      ieDistSigma = "5.0";
    },{
      ieName = "ErrorCode";
      ieDist = "constant";
      ieDistValue = "1";
    },{
      ieName = "SuccessCode";
      ieDist = "constant";
      ieDistValue = "0";
    } );
    noise = {
      uncorrelated = ( {
      distOccurrence = "uniform_real";
      distOccurrenceMin = "0.001";
      distOccurrenceMax = "0.01";
      eventIdRangeMin = "1";
      eventIdRangeMax = "500";
    });
  };
};
```

Furthermore, the MSC used as the second required input file for OpenMSC is also provided in Figure 6.

Figure 6: Emulated message sequence chart

ACKNOWLEDGMENT

The authors would like to thank (in no particular order) Zhuo Wu, Anderson Simiscuka, Gabriel Hogan, Dr. Zhiguo Qu, Dr. Zhenhui Yuan, and Dr. Conor McArdle for their contributions. The support of Enterprise Ireland Innovation Partnership with Ericsson is gratefully acknowledged.

REFERENCES

[1] Y. W. Ahn, A. M. K. Cheng, J. Baek, M. Jo, and H.-H. Chen, "An auto-scaling mechanism for virtual resources to support mobile, pervasive, real-time healthcare applications in cloud computing," IEEE Network, vol. 27, no. 5, pp. 62–68, Sep. 2013

[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. pp. 48–58, 2011

[3] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2008

[4] 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. 2002

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

[6] S. Achuthan, and J. O'Meara,, "A System for Monitoring Mobile Networks using Performance Management Events". IFIP/IEEE International Symposium on Integrated Network Management, 2013

[7] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," in Conference in modern analysis and probability New Haven, Conn., 1982), ser. Contemporary Mathematics. American Mathematical Society, vol. 26, pp. 189–206, 1984

[8] S. Robitzsch, F. Zaman, Z. Qu, J. Keeney, S. van der Meer, G.-M. Muntean, "E-Stream: Towards Pattern Centric Network Incident Discovery and Corrective Action Recommendation in Telecommunication Networks", IEEE Information Management, IFIP/IEEE International Symposium on Integrated Network Management, 2015

[9] F. Zaman, G. Hogan, S. van der Meer, J. Keeney, S. Robitzsch, G.-M. Muntean, "A recommender system architecture for predictive telecom network management," Communications Magazine, IEEE, vol.53, no.1, pp.286,293, January 2015

[10] S. Achuthan, and L. Fallon, "Load balanced telecommunication event consumption using pools", IFIP/IEEE International Symposium on Integrated Network Management, 2015

[11] L. Fallon, and D. O'Sullivan, "SECCO: A test framework for controlling and monitoring end user service sessions", IEEE Network Operations and Management Symposium (NOMS), 2014

Sebastian Robitzsch had been a postdoctoral researcher at Dublin City University, Ireland, for two years where he led E-Stream, a national funded collaborative project with Ericsson Ireland in the area of real-time data mining OSS solutions in mobile networks. In the past he has been with T-Systems, Germany; Fraunhofer FOKUS, Germany; and Nokia Research Centre, Finland, working on research issues spanning from interference and self-configuration techniques in 802.11-based multi-antenna mesh networks, heterogeneous radio access networks to system architecture design for trace analytics and recommender systems for next-generation OSSs. His recent research efforts focus on the softwarisation of network functions following existing SDN and NFV paradigms in order to allow a decoupling of infrastructure, service and content providers for a more versatile communication network. How to bring SDN and NFV to the monitoring task of large-scale networks has been one of his more recent proposal efforts. He received his Ph.D. from University College Dublin, Ireland, in 2013 and an M.Sc. equivalent (Dipl.-Ing. (FH)) from the University of Applied Sciences Merseburg, Germany. Currently, he is with InterDigital Europe, Ltd. working on ICN-related Horizon 2020 funded projects.

Faisal Zaman is a data scientist at Adaptive Mobile Security. Before, he was a Post-doctoral researcher with the Performance Engineering Laboratory and Network Innovations Centre, Rince Institute, Dublin City University, Ireland. He received his PhD in Information Science from Kyushu Institute of Technology in 2011. In his previous tenure as a Post-doctoral researcher in Kyushu Institute of Technology, he analysed time series data for weather forecasting and micro-array data for gene classification. He also worked as a Statistical Programmer in Shafi Consultancy Ltd and lead analytical teams to analyse medical trial data.

He is program committee member of several data mining conferences. He has published 30 articles, conference proceedings, books, book chapters, conference papers, and technical reports. He has experience in supervising PhD and M.Sc. level students.

Sven van der Meer received his PhD in 2002 from Technical University Berlin. He joined Ericsson in 2011 where he is currently a Master Engineer leading a team that will enhance the capabilities of Ericsson's OSS products. Most of his current time is dedicated to design and build advanced policy and predictive analytics systems. In the past, Sven has worked with Fraunhofer FOKUS (Berlin, Germany), Technical University Berlin (Germany) and the Telecommunication Software and Systems Group (TSSG, Ireland), leading teams and projects, consulting partners and customers, and teaching on university level. He is actively involved in the IEEE CNOM community as standing member of programme committees (IM, NOMS, CNMS, and APNOMS amongst others) and has helped to create and organise successful workshop serieses (MACE, MUCS, and ManFed.Com amongst others). He also contributed to standardisation organisations, namely the OMG and the TM Forum. He has published in more than 100 articles, conference proceedings, books, book chapters, conference papers, and technical reports. He has supervised and evaluated 6 PhD and more than 30 M.Sc. students.

John Keeney is a Senior Researcher in LM Ericsson Ireland, working in the Network Management laboratory in Ericsson's Software Research Campus in Athlone. His research focus is on monitoring and managing complex systems, especially telecoms systems, with a particular focus on knowledge extraction, event stream processing, and performance analysis. His work in Ericsson centres on online analytics and optimisation of Radio Access Network performance to inform the next-generation of Operation System Support (OSS) concepts for Ericsson's OSS product unit.

Gabriel Miro-Muntean received the Ph.D. degree from Dublin City University, Dublin, Ireland, for research in the area of quality-oriented adaptive multimedia streaming in 2003. He is an Associate Professor with the School of Electronic Engineering at Dublin City University (DCU), Dublin, Ireland. He is a Co-Director of the DCU Performance Engineering Laboratory and Consultant Professor with Beijing University of Posts and Telecommunications, China. He was Principal Investigator in E-Stream, a collaborative project with Ericsson Ireland in the area of real-time data mining OSS solutions in mobile networks. His research interests include quality-oriented and performance-related issues of adaptive multimedia delivery, performance of wired and wireless communications, energy-aware networking, and personalised e-learning. He has published over 250 papers in prestigious international journals and conferences, has authored three books and 18 book chapters and has edited six other books. He is an Associate Editor of the IEEE Transactions on Broadcasting, Associate Editor of the IEEE Communications Surveys and Tutorials, and reviewer for other important international journals, conferences, and funding agencies. He is a member of ACM, IEEE and IEEE Broadcast Technology society.