International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 3 (2015) pp. 7209-7214

http://www.ripublication.com

Real Time Data Analytics

Kiranmai munagapati[1] and D.Usha Nandhini[2]

[1]Student, ME Computer Science, Sathyabama University.
[2]Faculty of Computing, Sathyabama University
[1] kiranmai.munagapati@gmail.com,
[2] ushaduraisamy@yahoo.co.in

Abstract

Analyzing Big data is a challenging problem. Because it is a complex and unstructured data. Analyzing the data, drawing user behaviors from data, and projecting the behavior pattern will make the decision easy. There are only two paradigms for data processing: batch and stream. The prediction has to be in real time. By streams we can analyze the data in motion. The aim of this paper is to design a system that will analyze and display the real time pattern tracking of web application using streams technology. Design and implement the real-time application log tracking, deducing patterns, identifying anomalies which can be common application exceptions or security threats such as DDOS attacks**.**

Keywords- big data, motion, IBM infosphere, batch.

Introduction

According to IBM more than 2.5 quintillion bytes of data (2.5 billion gigabytes) are created every day. No one knows what is that data. Every action we do online like Facebook post, Twitter Tweet, Instragram photos etc... Generates Meta data. Companies can leverage this data to understand user behaviors what products they like (phone vs. android phones), what features they are expecting in a product. Understanding user behaviors is a very complex task because all this Meta data is unstructured. It's very difficult to convert billions of gigabytes unstructured data to structured. The existing computation power including super computers cannot handle such massive data. A new thinking and new paradigm is evolving to analyze the data. All these techniques together are called as BigData.Analyzing the data, drawing user behaviors from data, and projecting the behavior patterns will make the decision making easy. There are really only two paradigms for data processing: batch and stream. Batch processing is fundamentally high-latency (after so much time). So trying to look at a terabyte of data all at once, to do that computation in less than a

second with batch processing is impossible. Stream processing looks at smaller amounts of data as they arrive. Trying to look at a terabyte of data all at once, it is not possible to do that computation in less than a second with batch processing. Most of the data is time sensitive. Time sensitive data is useless after shelf life. For example consider a Pharmaceutical wants to track when a disease like H1N1 (Swine Flu) is breaking out. If the company identifies the demand for a certain medicine it can quickly increase its production so it makes more revenue. This prediction makes the company more effective. If the company identifies the patient's behavior after six months disease broke out it's not very useful for the company as it already lost the demand for these six month and also not enough medicine available for the patients. So In this case prediction has to be in near real time. Similarly a web application has so many security threats. Hackers like Anonymous always try to hack financial company websites for their economical or political threats.

Literature Survey

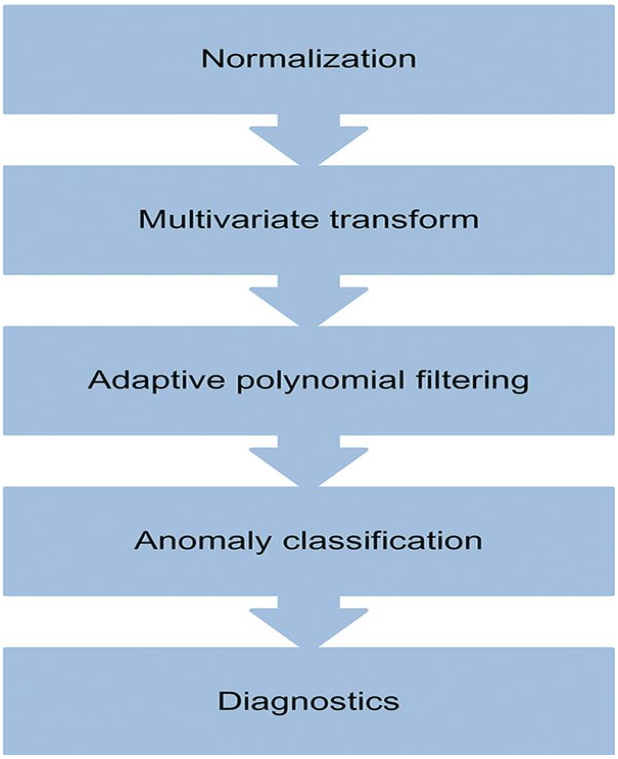
Anomaly detection is a recent challenging technique. Different anomalies are identified like DOS attack, worms etc. Lot of research work has done at data at rest and rather than data in motion.

[6,7] Lakhina uses the multiway subspace method to identify anomaly detection.[8] Principle component analysis (PCA) is a technique for identifying anomalies in network traffic.PCA transfer the original data into subspaces. Using PCA selects the m-dimensional subspace; normal data is projected into this subspace. Abnormal data is projected into the residual subspace.

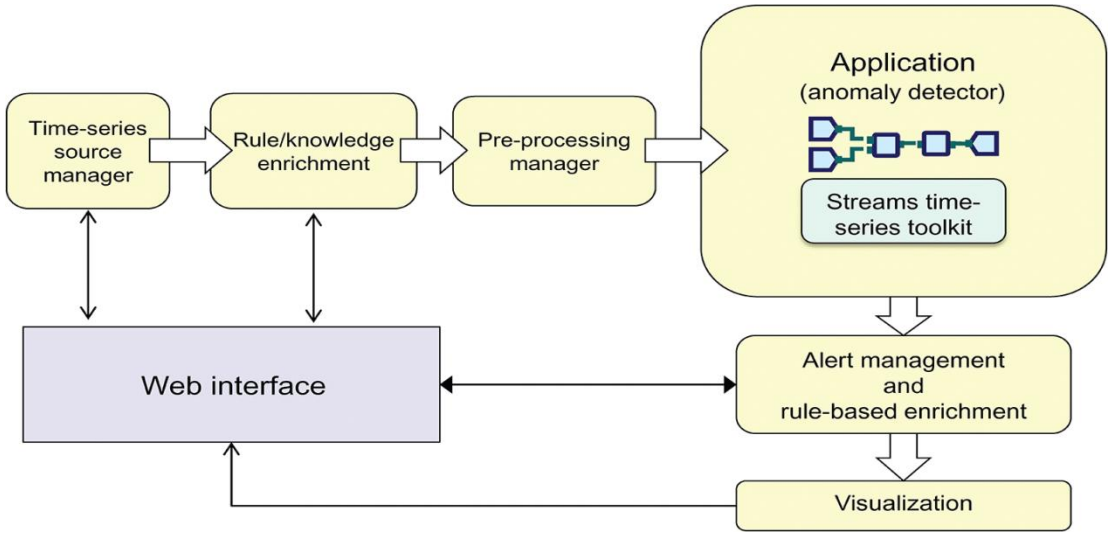
In real time scenario, all the data is not available at a time, the receiving data is faster than the data processing time. Online version algorithm[7] has been proposed sliding window for anomaly classification. But here the drawback is that window size is not optimal due to no guideline.

To overcome this SPIRIT algorithm[9] is proposed .In SPIRIT Given n numerical data streams, all of whose values we observe at each time tick t, SPIRIT can incrementally find correlations and hidden variables, which summaries the key trends in the entire stream collection.

Similarly KOAD (Kernel-Based Online Anatoly Detection)[10] uses recursive least-square regression technique to detect outliers. the target variable is arbitrarily defined as the sum of the values in the input vector. The regression is incremental. The incoming data are iteratively projected in a kernel-based subspace on which an online recursive least-square regression is performed. The distance to the subspace is used as a score to detect anomalies in real time.



The ACA processing steps. Incoming time series are normalized and then passed to a multivariate transform. An adaptive polynomial filter tracks data movements on the transformed space and produces residuals. The residual statistics are then used to detect anomalies. Diagnostics are performed to signal the faulty time series.



Proposed System
Figure1:STAM overall architecture

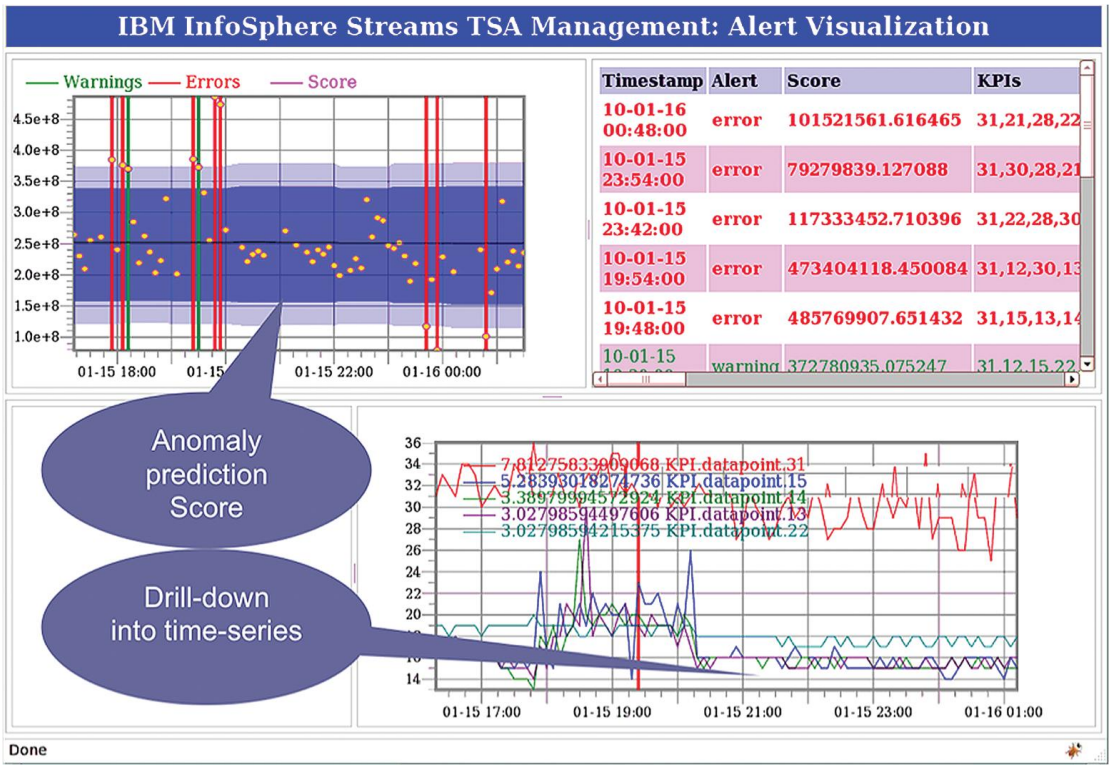


Figure2:STAM web-based interface

SYSTEM OVERVIEW:

The above figure depicts the system architecture. The system is having a web interface, it will interact with the user and forward the request to the other components for handling request. Time series source manager communicate with the web interface for loading the data.

Time series source manager forward the data to the pre-processing module for joining and aligning data in real time properly. Time stamp is assigned to the time series data, then the data is aligned and synchronized. The aligned data is send to the application module .Application module handles several real time applications. These applications run on the IBM InfoSphere Streams middleware.

The output of the application module is return back to the Web Interface via a real-time visualization system. Alert management is used to display an alert if any anomaly is detectd in the application.web interface provides the functions like source selection,types of analytics, parameter selection etc.Once the analytic is chosen, the output is visualize to the user through visualization module**.**

IBM® InfoSphere® Streams:

IBM® InfoSphere® Streams[1] is a platfor which analyze the data in motion.it allows user developed applioctions to quickly ingest, analyze and correlate information as it

arrives from thousands of real-time sources. The solution can handle very high data throughput rates, up to millions of events or messages per second.

InfoSphere Streams helps you:

- **Analyze data in motion**—provides sub-millisecond response times, allowing you to view information and events as they unfold.
- **Simplify development of streaming applications**—uses an Eclipse-based integrated development environment (IDE).
- **Extend the value of existing systems**—integrates with your applications, and supports both structured and unstructured data sources.

Analyze data in motion

- Supports analysis of continuous data including text, images, audio, voice, video, web traffic, email, GPS data, financial transactions, satellite data and sensor logs.
- Includes toolkits and accelerators for advanced analytics, including a telco event data accelerator that analyzes large volumes of streaming data from telecommunications systems in near real time and a social data accelerator for analyzing social media data.
- Distributes portions of programs over one or more nodes of the runtime computing cluster to help achieve volumes in the millions of messages per second with velocities of under a millisecond.
- Allows you to filter and extract only relevant data from unimportant volumes of information to help reduce data storage costs.
- Scales from a single server to thousands of computer nodes based on data volumes or analytics complexity.
- Provides security features and confidentiality for shared information.

Simplify development of streaming applications

- Allows you to build applications with drag operators, and dynamically add new views to running applications using data visualization capabilites such as charts and graphs.
- Enables you to create, edit, visualize, test, debug and run Streams Processing Language (SPL) applications.
- Provides composites capability to increase application modularity and support large or distributed application development teams.
- Allows you to nest and aggregate data types within a single stream definition.
- Enables applications to be built on a development cluster and moved into production without recompiling.

Conclusion

The system takes the user input data and produces the output in real time pattern. Once the output is given to the system, we have to chose the type of analytics as system listed different analytics types. Once the type chosen system visualizes the

data in that form and alert displays if anomaly is detected. Pattern displays the real time data, so the we can easily identifies where attacks are getting and what particular time we are getting. Once those are identified we can shut down their ip address.

REFERENCES

[1]. A. Biem, E. Bouillet, H. Feng, A. Ranganathan, A. Riabov, O. Verscheure, H. Koutsopoulos, and C. Moran, "IBM InfoSphere Streams for scalable, real-time, intelligent transportation services," in Proc. SIGMOD Conf., 2010, pp. 1093–1104.

[2]. L. Amini, H. Andrade, R. Bhagwan, F. Eskesen, R. King, P. Selo,Y. Park, and C. Venkatramani, "SPC: A distributed, scalable platform for data mining,["in Proc. Workshop DM-SSP, 2006,pp. 27–37.

[3]. B. Gedik, H. Andrade, K.-L. Wu, P. S. Yu, and M. Doo, "SPADE:The system S declarative stream processing engine," in Proc. ACM SIGMOD, 2008, pp.1123–1134.

[4]. D. Yankov, E. Keogh, and U. Rebbapragada, "Disk aware discord discovery: Finding unusual time series in terabyte sized datasets,"Knowl. Inf. Syst., vol. 17, no. 2, pp. 241–262, Nov. 2008.

[5]. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey, ACM Comput. Surveys" vol. 41, no. 3, p. 15, Jul. 2009

[6]. A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows", ACM SIGMETRICS Perform. Eval. Rev., vol. 32, no. 1, pp. 61–72,Jun. 2004.

[7]. A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in Proc. Conf. Appl., Technol.,Archit., Protocols Comp. Commun., 2005, pp. 217–228.

[8]. I. Jolliffe, Principal Component Analysis, 2nd ed. New York, NY, USA: Springer-Verlag, 2002.

[9]. S. Papadimitriou, J. Sun, and C. Faloutsos, "Streaming pattern discovery in multiple time-series,"in Proc. 31st Int. Conf. Very Large Databases, 2005, pp. 697–708.

[10]. 14. T. Ahmed, M. Coates, and A. Lakhina, "Multivariate online anomaly detection using kernel recursive least squares," in Proc. 26th IEEE INFOCOM, 2007, pp. 625–633