Streaming Analytics over Real-Time Big Data

Ranjitha P

Abstract- A portal is developed using open source tool called Liferay for water management and city management using data acquired from sensors deployed in overhead water tanks and across the city at different locations. The parameters captured from sensors are water level and parameters captured sensors deployed across the city include dust, UV, temperature, light, humidity, sound and air quality. Data generated by these sensors amounts nearly megabytes of data per day and gigabytes on an annual rate. Real time analytics help in monitoring and management of water resources and rate of pollution across the city. Visualizations are provided in the form of a time series graph. The visual representations are plotted using d3.js graphs in real-time, hence allowing the users to take corrective decisions with respect to water usage and managing pollution across the city.

Index Terms: D3.js, stream processing, siddhi CEP, big data analytics, liferay.

I. Introduction

Sensing elements are growing rapidly in all areas of day-to-day life leading to large volumes of data generation. Data generated in this manner is considered as streaming data which is continuous in nature. This leads to the notion of Big Data as it refers to the velocity aspect of Big Data. Though a large amount of data is generated, extracting meaningful and useful contents from such streams is highly motivated in order to make timely decisions. These timely decisions can be achieved only when the data obtained is processed, analyzed and visualized as and when it arrives (i.e., in real-time). Visualizing and analyzing such huge streams is referred to as Big Data Visualization and Analytics.

a) *Big Data Processing*

There are two types of Big Data processing – 1. Batch based stored data processing 2. Real-time data stream processing [6]. Batch based processing handles historical data while the Real-time stream processing handles on-the-fly data. Batch based processing is used when we first store the data and then perform analytics as a later part, while real-time data stream processing performs analytics as and when the data arrives irrespective of whether the data is stored or not.

The key data processing approach for handling real-time streaming data is Complex Event Processing (CEP) [5]. It infers events or patterns from multiple sources of data and identifies meaningful events among them and responds with possible quick decisions. In this paper, we are considering events to be the flow of water in various overhead water tanks within a campus and data generated from sensors which capture dust, light, sound, humidity, air quality, UV and temperature that are deployed across the city. Events here refer to the change of state in generation of data.

The primary functionality is to match queries with events and generate response immediately. In this methodology, stored queries are run over dynamic data streams. In general, it is just the reverse procedure of traditional databases. Hence such a processing plays a crucial role in Big Data analytics. Siddhi – CEP, an open source complex event processing engine which is under Apache license is used [7].

b) *Big Data Analytics*

Applying Big Data analytics on such large data sets enables to uncover hidden patterns, correlations and preferences. Such an analysis improves the performance and efficiency. It thus brings real value to the data. It deals with what attributes of real-time data has to be captured for analyzing and visualizing. The critical parameters are given utmost importance. It deals with collecting, organizing and analyzing large data sets to discover patterns.

c) *Big Data Visualization*

Visualizations are always more understandable and appealing to the end users than continuous raw data streams. Smarter visualizations result in smarter and quicker decisions. These data visualizations can be in the form of charts, graphs, maps, etc. which can be placed within a portal or a web page. In this paper, we have considered the visualizations for parameters captured from the sensors such as water level, salt/chlorine content, PH, dissolved salts and temperature which are plotted through d3.js graphs in real-time within a Liferay portal based on the location of the buildings.

II. Related Works

The prevalent use of sensors has led to Big Data which traverses enterprise data processing pipelines in a streaming fashion leading to online analytic capabilities. Statistical anlysis and data mining can be used for real-time systems by implementing over the same system [1]. Timely analytics over Big Data made up of huge data sets is the key factor. Sensors that generate Big Data are from various fields such as intelligent transportation with road and vehicle sensors, financial market trading and surveillance, crowd control,

Author: Department of CSE, MSRIT, Bangalore, India.
e-mail: ranjithaph@gmail.com

military decision making, large scale emergency response and early warning of natural disasters [2].

To handle such a huge data, visual analytics is important because humans have the ability to gain quick insight and take quicker decisions. Computation of results obtained from such huge data has to be precise [3]. Dynamic Visual Analytics is the process of integrating knowledge discovery and interactive visual interfaces to facilitate data stream analysis and provide situational awareness in real-time [8].

D3.js is a web visual presentation tool for big data that provides graphical statistics of data flow. D3 achieves visualization through data loading, data binding, analytic transformation elements and excessive element. D3 uses CSS3, HTML and SVG (Scalable Vector Graphics) on the web. Data visualization technology views each data item as a single pixel element and hence large number of data sets constitute image of the data. Data visualization conveys information by using graphical tools and zoom features [9].

III. Methodology

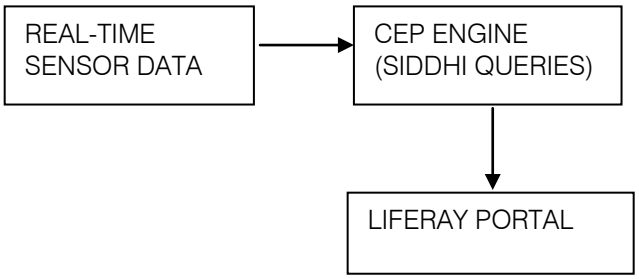

Figure 1 : High Level Design

In order to sense the usage of water and pollution level within a city, water level sensors and sensors that capture dust, light, sound, humidity, air quality, UV and temperature are deployed into water tanks and across the city.

Constant power supply and stable WIFI connection are made available for the duration of the project. These sensors use less bandwidth and electricity.

An API is created for these time-series data streams. At first, a source is created using POST method. POST can be empty or have an application/JSON body. Then the source info is fetched using the GET method. Data is sent after authorization modeled after Heroku's token based authorization. Later data is read in pages either in ascending chronological order or descending chronological order with latest results or entries being visualized. Also an option to visualize historical data is provided through the facility to choose from calendar option.

Figure 2 : Flow Design of Methodology

Figure 2. shows the flow of data along the different portlets.

According to the high level design in Figure 1, sensors generate real-time data which serves as the input to the CEP over which the data is processed and executed. After processing, the outcomes of the various queries are visualized in the form of D3.js graphs within a portal technology called Liferay.

IV. Input

The input to obtain the data visualization is a stream of data which can be either in CSV (Comma Separated values) or JSON format. And data can be either offline (Stored or historical data) or online (real-time data). The timestamps chosen are 1/min for a data generated from water level sensors and 1/10secs for data generated from sensors deployed across the city to counter pollution. Based on the location of the sensor selected, the visualizations are made available to the end users. These visualizations should be compatible on desktops, tablets and mobile phones.

V. Processing

Siddhi CEP queries reside in the CEP engine. The incoming data stream is run against appropriate Siddhi queries and the results are generated and the same queries can be visualized in D3.js graphs. These queries process event patterns. Based on the event that arises in the incoming data stream, that particular query gets executed for the visualization.

Sample Siddhi Query –

```
From DataStream #window. External
Time (timeStamp,200)
insert into Window Stream
```

In this query, we are detecting the rate drop based on time window where rate drop > 200.

VI. Output

The output will be a graph plotted against time stamp and water level. These graphs are plotted using D3.js which is a javascript library for manipulating documents [4]. The streaming data is captured in real-time and the graph also moves/changes along with time as the new data arrives. This graph will be a part of the portal technology called as Liferay which is a web based technology that provides personalization, single sign on, responsive and content management. Within a portlet, there will be a group of associated portlets. In this scenario, the various portlets are login portlet, Google

map portlet for location selection, chart portlet for graph visualization.

The user logs in through the login portlet and selects a location in a Google map portlet as well as the chart settings he wants to visualize. Based on the selection made in the Google map portlet, the water usage or the level of dust, sound, light, UV, humidity, air quality or temperature of that particular location will be visualized in the graph portlet.

Sample output –

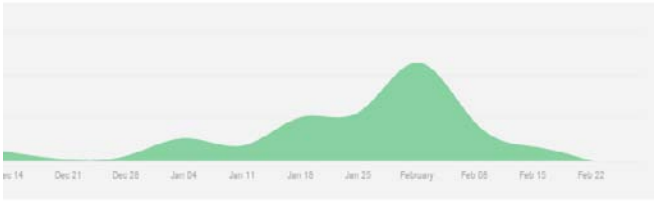

Figure 3 : Sample Graph

Time series graph plays a prominent role in these visualizations. The graph visualization gets highlighted with a spot at a particular point in the graph when aggregate queries are run.

VII. Implementation

The following results in Figure 3 and Figure 4 represent the time series visualizations. Here the graph is plotted against timestamp and data. The data can be water level or data from dust sensor, sound sensor, humidity sensor, light sensor, temperature sensor, air quality sensor or UV sensor.

Different visualizations can be realized based on what parameters of the data are captured by the sensors thus helping the users to smartly manage water and pollution level within the city. The main goal is to perform analytics on data-in-motion.

The challenges include fault-tolerance in the cluster where application is deployed, handling continuous data flow in mission critical applications with very minute disruption and resource wastage when the data input rate is non-uniform.

The most appropriate visualization is made available to the end users as part of the portal. The visualization obtained should be easily understandable so as to take quicker decisions.

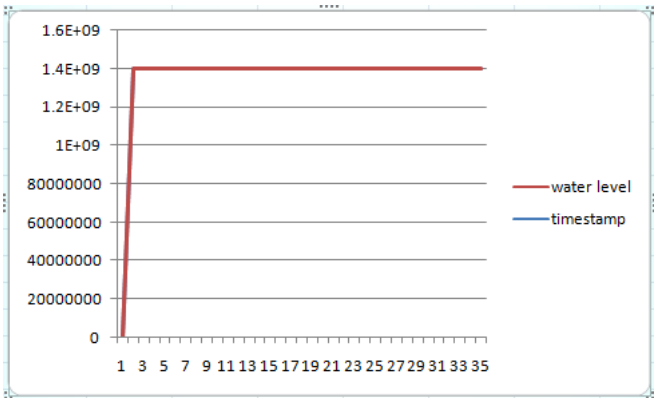

Figure 3 : Results for Smart Water Management

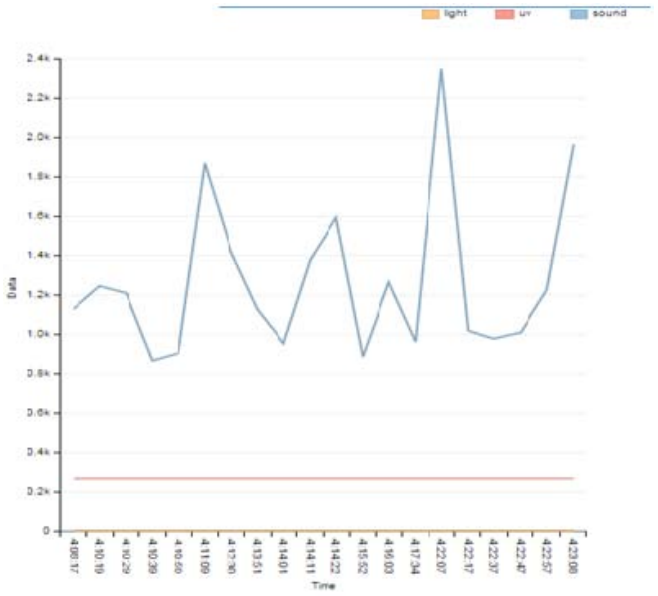

Figure 5 : Results for Sense City

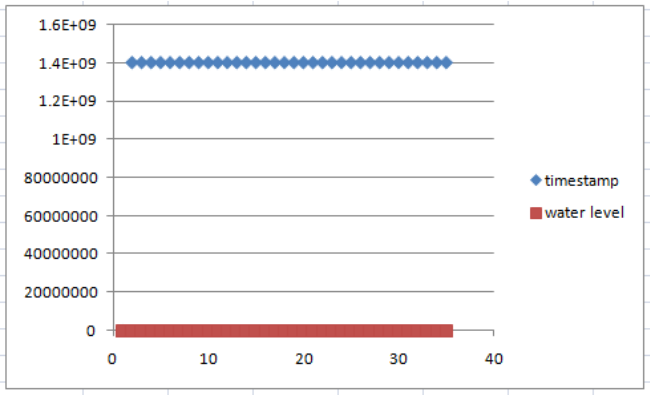

Figure 6 : Scatter Plot Results

VIII. Future Work

As part of future work, we extend the Big Data visualization techniques to different formats.

a) ARIMA Algorithm

Autoregressive Integrated Moving Average is a model that understands the data and predicts the future

points in series which is known as forecasting. It is referred to as ARIMA (p, d, q) or ARIMA (AR, I, MA).

Where p – auto regression

d - Integration

q – Moving average

This concept is used to predict what the future water levels will be based on the current water levels for a particular time series of data.

b) Scatter plot representation of water usage at different buildings within the campus. These are plots of data points on a horizontal and a vertical axis to show the usage of water at different buildings. At the same time building with isomorphic water usage can be grouped under the same cluster on the scatter plot.

c) To implement a similar Big Data analysis for different sensors that capture dust, airquality, sound, light, temperature and humidity.

IX. Conclusion

As the sensors generate huge amount of data, it is considered as Big Data by the velocity aspect of it. It is very important to analyze, process and visualize Big Data as it help in making meaningful and quicker decisions. This is very important in mission critical applications. Processing the data on-the-fly is achieved with the help of Siddhi CEP. Visualizations are more appealing and understandable. D3 graphs serve the purpose of visualization. As per the case of smart water management within a campus, allows us to make better usage of water and thus avoid wastage.

References Références Referencias

1. Ismail Ari, Erdi Olmezogullari, Omer Faruk Celebi, Cloud Computing Research Group and Avea Labs, Computer Science Department, Ozyegin University, Istanbul, TURKEY. “Data Stream Analytics and Mining in the Cloud.” 2012 IEEE 4th International Conference on Cloud Computing Technology and Science.
2. Nader Mohamed, UAE University, UAE, Jameela Al-Jaroodi, University of Pittsburgh, USA. “Real-Time Big Data Analytics: Applications and Challenges.” 2014 IEEE International Conference On High Performance Computing and Simulation (HPCS).K. Elissa, “Title of paper if known,” unpublished.
3. Jaegul Choo and Haesun Park , Georgia Tech. “Customizing Computational Methods For Visual Analytics With Big Data.” 2013 IEEE Computer Graphics and Applications.
4. “D3-Data Driven Documents.” Internet: http://d3js.org/
5. ”Siddhi”, https://github.com/wso2/siddhi
6. http://www.nec.com/en/global/rd/research/cl/bdpt.html [Big Data Processing].
7. Sriskandarajah Suhothayan, Isuru Loku Narangoda, Subash Chaturanga, Siddhi: A Second Look at Complex Event Processing Architectures, ACM GCE Workshop 2011.
8. Sabri Hassan, Johannes S¨anger, G¨unther Pernul, Department of Information Systems, University of Regensburg, Regensburg, Germany. “SoDA: Dynamic Visual Analytics of Big Social Data.” Big Data and Smart Computing (BIGCOMP), 2014 International Conference.
9. Fan Bao, Dalian Maritime University, Dalian, China , Jia Chen. “Visual framework for big data in d3.js.” Electronics, Computer and Applications, 2014 IEEE Workshop.

Global Journals Inc. (US) Guidelines Handbook 2015

www.GlobalJournals.org