www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 10 October 2017, Page No. 22551-22558
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i10.04

Real Time Product Feedback Review and Analysis Using Apache Technologies and NOSQL Database

BiswaRanjan Samal[1], Mrutyunjaya Panda[2]

[1,2] P.G Department of Computer Science and Application, Utkal University
VaniVihar, Bhubaneswar- 751004, India

Abstract:
Whenever a feedback system comes into mind, it's always a demand of the e-commerce organizations to get the customer feedbacks in real time and to build some strong dashboards on top of these feedbacks/ratings. So that they can easily know the performance of any product at any point of time as well as they could able to take a decision, on what to do with the products those are getting very poor feedbacks. Which will result in a minimum impact on the tangible and intangible assets of the organizations.
For achieving the above goal it is very necessary for these organizations to adopt the right tool and implement the required environment which can deal with the real time big data ingestion, enrichment, indexing and have the power to perform simple as well as complex analysis algorithm on the stored data.
In this paper, we have collected Amazon Product Ratings for doing analysis and used Apache NiFi for ingesting real-time data into Apache Solr and have taken help of Banana Dashboard to show the real time analysis results in the form of attractive and user-friendly dashboards.

Keywords: Big Data; Apache NiFi; Apache Solr;MongoDB

Introduction

A Feedbacks is nothing but the experience that the customer shares after availing a service provided by the service provider. Here in our context service refers to the products and service provider refers to the e-commerce organizations who sales those products to the customers. The experience or feedbacks shared by the users represent their sentiments[8]. For example considering feedback in the form of ratings between 1-5, where 1 represents the worst experience (Negative Sentiment) and the 5 represents the best experience(Positive Sentiment).Feedbacks also may come in a textual format where users have represented their opinion with the help of the texts. In this scenario for knowing the type of the sentiment, it's mandatory to take help of Machine Learning Algorithms[12].

Because of rapidly growing competitions [9] in the e-commerce sector, customer feedback always plays a vital role for these organizations. Therefore it's always preferable to provide good service/product to the customers and collect the good feedbacks from them. If the product feedbacks can be known in real time then it can help the service providers to take a decision on time and take the necessary actions against the products which are gaining the negative feedbacks[9]. So that the customers will no more continue to get those bad products, which will be benefited for both the customer and service provider.
This is why we motivated towards doing a research which will provide a solution to the above problem. We have considered only the product feedbacks which are in the form of ratings between1 to 5.
While considering the feedback collection from the users, It's very likely possible that millions of feedbacks might be coming per second/minute throughout the globe. The process of feedback gathering may have a collection of different websites(Ex: Survey Monkey, etc) and mechanisms(Ex: email, SMS, etc) [2] which are getting stored across variety of databases[10], so it is very necessary to collect all those feedback data from these mediums/databases and ingest, enrich them as per the need in real time without any fault and security violence.
In a feedback system, data can be found at either of the three conditions such as Data in client site, Data in Motion and Data in rest. Data in client site refers to the medium by using which users are viewing or giving the feedbacks, Ex: Web Browser. We can impose security on these mediums by various means

such as making the communication socket secured, Ex: SSL [1]. Then it comes to the data in rest basically data in rest refers to the storage medium that has been used for storing the data, Ex: Databases and these storage mediums can be secured by means of applying some encryption or authentication. But if we consider about the data in motion this is the major concern that how to secure and track the data flow which is in motion it's quite difficult.

We know that we don't have any control on data in motion but at the same time, we should not allow any kind of data loss. Therefore here it comes a requirement for a data automation platform/framework which is capable of handling millions/billions of records as well as can ingest and enrich the data in real time with zero risks and fault. It also should be able to track the record of the data transmission and data flow between different stages of data in motion.

At the same time, this platform should be user-friendly, easy to use and the most important thing is that it should be open source. So the care should be taken while choosing any data automation platform and above said requirement should be met. That's why we choose Apache Nifi[3] as our Data Automation Tool.

While discussing the real-time analysis of data it's obvious to think how effectively we can store data and create indexing on it so that the huge amount of data will not hamper the performance of the application. To deal with this we have selected Apache Solr[4][5] as our Search Platform.

It's always a better idea to choose the visualization tool which supports best to the environment or database that you are using. That's why we have selected Banana Dashboard which runs on top of the Apache Solr[6][7].

Here Onwards the paper is organized as follows. Section II contains some of the related works. In-depth information about the technologies used can be found in section III. Whereas demonstration of the adopted methodology is present in section IV. Experimental Results and discussions are available in Section V followed by Section VI featuring the conclusions and future works.

Related Works

In[10], customer comments about the products are taken as the feedbacks and then the attributes present in the feedbacks are analyzed if it's positive or negative then an overall score is assigned to the each attribute and the final score gets calculated and based upon it authors have built some of the visualization tools.

In[11] authors have collected car review data of 4 years and have developed a model named Pulse, by using which they can visualize the product in two dimensions such as topic and sentiment.

In[14] authors have collected the customer feedbacks from the twitter and have introduced a geo and feature based stream analysis technique which helped them for doing the term associations and developing some new visualizations.

1. USED TECHNOLOGIES

3.1 Apache Nifi

Apache NiFi is an Apache Software Foundations project, using which one can enable data flow automation between the systems[13]. It helps in moving and tracking of data. It is developed using the flow based programming having it's own web interface. It can co-ordinate between the producers of data and consumers of data. Like in a feedback system there may be several mediums of data storage are available which can act as data producers, and the database used to real-time analysis can act as a consumer for data. Some of the basic concepts are described below

- ***FlowFile:*** Each object which is moving through the system is represented by a FlowFile, and NiFi keeps metadata of attribute string with the content size.
- ***FlowFile Processor*:** The Processor does the actual work. Processor can route, transform, and enrich the data by accessing the incoming flow files and its contents. TheProcessor can also operate independently without the flow files.
- ***Connection*:** This is the linkage between the processors which acts as a queue. These queues can enable backpressure by having upper bounds on load.
- ***Flow Controller*:** It acts as a broker for transmission of flow files between the processors, for which it keeps track of how processes are connected, and manages the thread allocation accordingly
- ***Process Group*:** It's nothing but a set of processors with their connections. It receives data using input ports and sends them through output ports.

Various features of Apache NiFi is discussed below,

- It guarantees delivery even at in high scale.

- It supports buffering and back pressure.
- It can set priorities on how data can be retrieved from a queue
- It provides visual control over the data flows
- NiFi have the ability to securely exchange data in every stage using 2-way SSL.

3.2 Apache Solr

It is a Storage/Search engine which is optimized for searching text-based data of large volumes. It is an Open Source project under Apache Foundation and built on top of Lucene(full-text search engine)[11][4]. Because of it's searching and indexing feature it is widely used by the applications which have search requirements.

As because in our research we need a highly scalable storage engine along with the strong indexing and optimized searching facility, that's why we have used Apache Solr as our storage medium. Various features of Apache Solr can be found below.

Supports Restful APIs: Solr can store data in XML, JSON and .CSV format. Also One can use Restful services to communicate with the solr.

Full-Text Search Facility: All the facilities which are required for doing a full-text search is already available with the solr. Like Tokens, wild card, spell check, etc.

Ready for Enterprise: Organizations can deploy solr according to their need such as in small, large, distributed, cloud and stand alone systems.

Extensible: As solr is developed on top of the Lucene and Lucene is developed using the Java, so one can customize its components according to their need.

***Big Data compliant*:** It can be used as a storage for NOSQL data where one can distribute search tasks along the cluster.

***Admin Page*:** Solr provides a very user-friendly interface for its users where the users can manage logs, update, add, delete and search records.

***Text-Centric*:** Solr is good at searching the text data and it returns data according to the user's query relevantly.

3.3 Banana

It is a web-based data visualization tool developed using web technologies like BootStrap, AngularJs, CSS3[7] and D3.js. Using it one can create dashboards on top of the data stored in the Solr instance. These dashboards run in clients browser and it can fetch data from the Solr instances present locally or in the cloud.

As because we need to display information's in the form of dashboards and again the data for those dashboards would retrieve from the Solr Storage that's why we have selected the Banana as our data visualization tool.

3.4 MongoDB

MongoDB is an open source document based, cross platform NOSQL Database, having features like scalability, high availability and high performance. It is developed using C++ and stores data in JSON format[14][15]. It can be used to store the Big Data. We have used MongoDB for storing the JSON documents that are converted from CSV files which contain the customer reviews about the products. Beside above features MongoDB has following features,

1) Built-in failover and replication
2) Native sharding and horizontal scalability
3) Map Reduce support
4) Real Time aggregation on huge data

3.5 Hortonworks Sandbox

As we are dealing with the Big Data and it's different technologies, so it is necessary to have an environment setup for the big data. This environment can be set using cloud or by some tools, such as Hortonworks Sandbox. It is preconfigured with the minimum things that someone needs to deal with the big data[16][17]. It needs a virtual machine to run with at least 8GB of dedicated RAM.

We have used Hortonworks Data Platform Sandbox along with Oracle Virtual Box and have installed Apache Solr Apache NiFi and Banana for achieving our research goal.

Proposed Methodologies

The step-by-step procedures for the proposed methodology adopted in this research are as follows:

4.1 Collecting product review datasets

As because we are going to work on the product reviews so for us the first step is to collect the product review data sets and those data sets should be large in size. So that we can very well observe the performance of various tools we have used. For our research, we have collected some of the product review data sets published by the amazon[18] over the internet in CSV format. Fig.1 represents our dataset size i:e 96,70,000 reviews stored in Apache Solr Database.

DOI: 10.18535/ijecs/v6i10.04

Fig.1 Representing total number of reviews

4.2 Data Enhancement

Collected reviews are not according to our need that's why we did some modifications, such as the

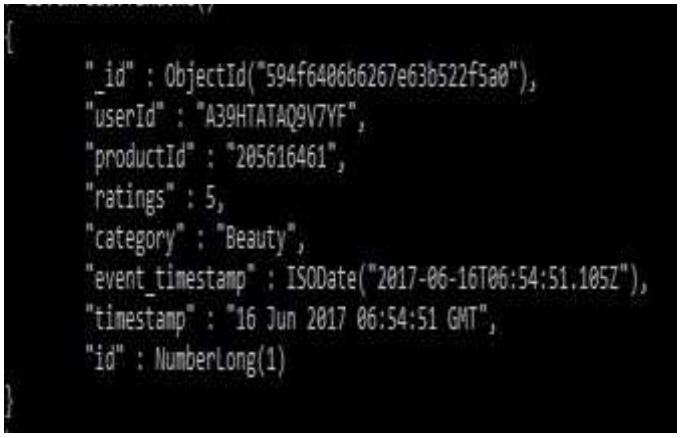


Fig.3 Modified review data in JSON format

available time stamp format and the timestamp. We have changed the timestamp

Fig.4 REST Service demonstration

according to our need and also have added the category field to each review, After modification, we convert the data from CSV to JSON and inserted those data into the MongoDB. Fig.2 represents a review in CSV format and Fig.3 represents the modified JSON format of that same review inserted in MonogDB.

	A	B	C	D
1	A39HTATAQ9V7YF	205616461	5	1369699200

Fig.2 Product review data in CSV format

Java language has been used for modifying the CSV data points, converting CSV to JSON and also for inserting data into MongoDB.

4.3 Implementing REST Service

For demonstrating a Real Time scenario we have implemented a REST Service using Java language. This Service can be called from any browser or by any web application for fetching the data from the MongoDB. Whenever the service is called it will return 1,00,000 records from the MongoDB.

Fig.4 shows the REST Service demonstration.

4.4 Creating Apache NiFi Template & Ingesting Data into Apache Solr

In this step, we have created a database in our Solr instance to save the data and Apache NiFi template for retrieving the data and sending it to the Solr. a template is nothing but a collection of some inter connected processor which automates the data flow from one system to another.

Fig.5 Designed Apache Nifi Template

DOI: 10.18535/ijecs/v6i10.04

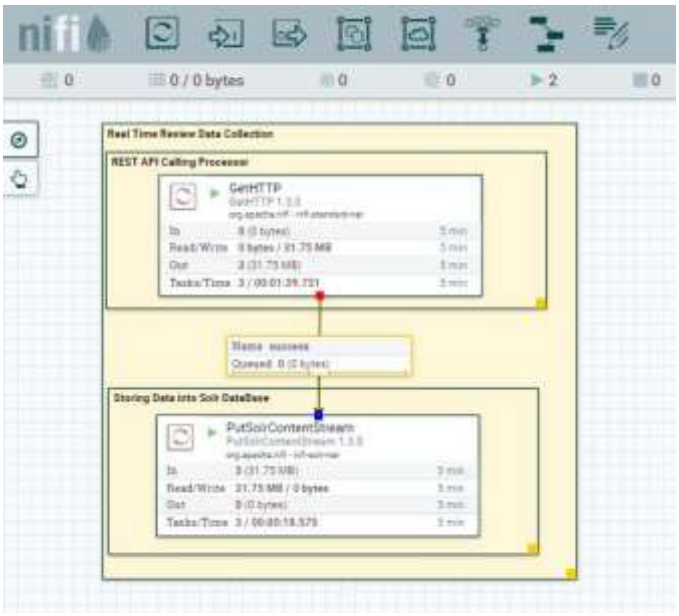


Fig.5 shows the Apache Nifi Template that we have designed for our research.

- **GetHTTP**: This processor is used for calling the REST services or any web URL, we configured this processor to call our previously implemented REST service and scheduled it with 1-minute duration. So that the processor will call the REST service after every minute and gets the records from it and sends those records to the next processor.
- **PutSolrContentStream**: This processor is responsible for storing the incoming data into the Solr instance. It takes the Solr Database name and location of its configuration.

Fig.7 Graph showing overall total rating count

So the data flow from the MongoDB to the Apache Solr in real time through the REST service and it is automated, so no more manual work needed. The Data automation tool will take care of the data flow and provides us facilities that we can see the data

The algorithm of proposed methodology is presented in Table1 flow history and can know how much data comes into the processors and goes out from the processors and the status of transmission also.

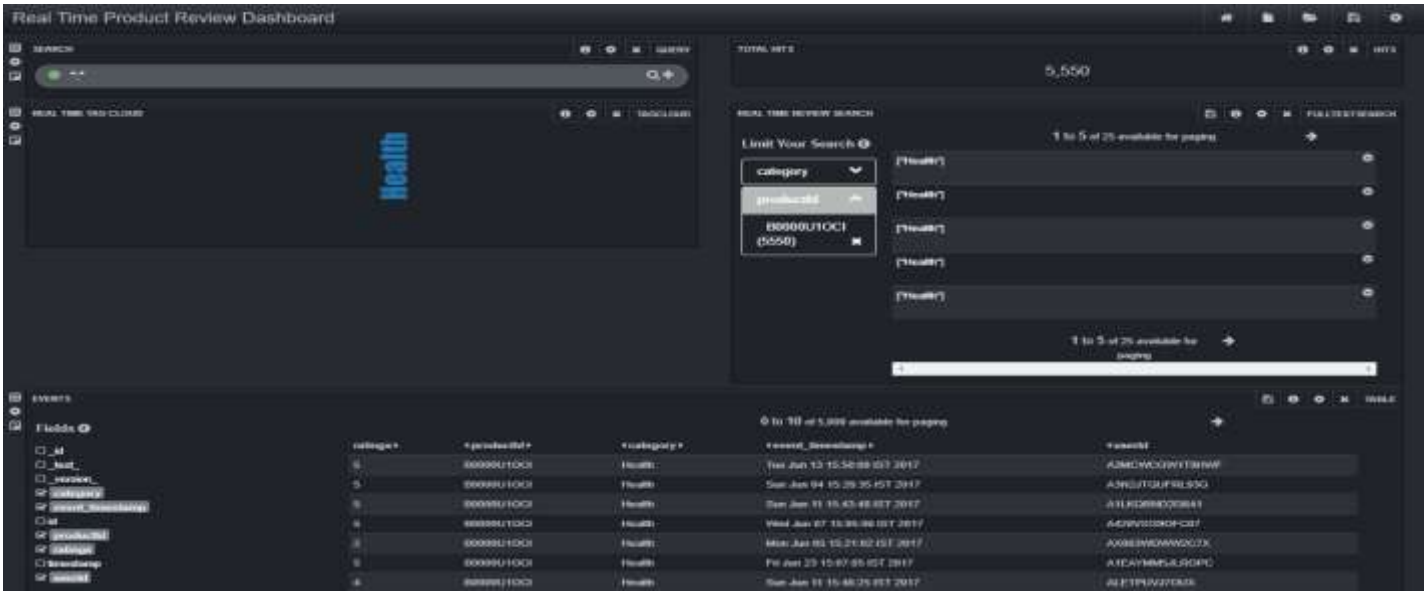
Fig.6 Represents the Data Flow History of GetHTP processor.

Fig.6 DataFlow history of GetHTTP processor

4.5 Building Banana Dashboard

As we have discussed that Banana dashboard is built on top of the data present in the Apache Solr[6][7], and it is a web-based visualization tool so we can access the Banana Tool in the browser by entering http://127.0.0.1:8983/solr/banana. Here we can select our Solr collection name and the type(Time Series Dashboard, Nontime series dashboard), theme(Dark, Light) of the dashboard we want to create.

DOI: 10.18535/ijecs/v6i10.04



.

Fig.9 Final Banana Dashboard

Table1 Algorithm overflow of the proposed approach

Step1: Start
Step2: Product Review data set collection
Step3: Data Enhancement
Step3.1: Converting data from CSV to JSON
Step4: Inserting data into MongoDB
Step5: Implementing REST Service
Step6: Implementing Apache NiFi template for Data Flow Automation
Step7: Collecting data from MongoDB and ingesting into Solr
Step8: Building Real-Time Dashboards using Banana Visualization Tool
Step9: Finish

Experimental Results and Discussion

Fig7 represents that the time at which we have taken the snapshot till that, 5,620,000 number of records are already available in the Solr database, and it shows the total number of ratings for each rating criteria(1-5).

The dashboards will automatically get refreshed with the available data time to time. In Fig8 we have tried to analyze the current performance of Health products. And finally, in Fig9 we did analysis on, which product category got how many reviews, Top 10 products who got more reviews, Top 10 timestamps at which more reviews came, rating analysis along with a table where the user can search for the products.

As we said before, the Banana Dashboard has auto refresh feature so the data and dashboard both are auto-synced with the available data in Solr.

Conclusions and Future Works

In this paper we have demonstrated how to process big data(customer feedbacks) in real time and extract the useful information's from it and also demonstrated how to represent the information's in the form of Analytics. It will help the stakeholders and management peoples related to e-commerce for making the appropriate decision at perfect time.

Here we have used the available Open Source technologies and tools for processing the data sets available in the internet. The experiment is conducted on single node machine. In future we hope to extend our experiment to the multi-node environment and will try to merge some Artificial Intelligence along with the Machine Learning techniques to make it more powerful.

References

1. Zikopoulos, Paul, and Chris Eaton. Understanding big data: Analytics for enterprise class hadoop and streaming data. McGraw-Hill Osborne Media, 2011.
2. The Digitization of Word of Mouth: Dellarocas, Chrysanthos. "The digitization of word of mouth: Promise and challenges of online feedback mechanisms." Management science 49.10 (2003): 1407-1424.Promise and Challenges of Online Feedback Mechanism
3. Hughes, James N., et al. "A survey of techniques and open-source tools for processing streams of spatio-temporal events." Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming. ACM, 2016.
4. Shahi, Dikshant. "Apache Solr: An Introduction." Apache Solr. Apress, 2015. 1-9.
5. http://lucene.apache.org/solr/
6. Tan, Biying, et al. "Clairvoyant-push: A real-time news personalized push notifier using topic modeling and social scoring for enhanced reader engagement." Big Data (Big Data), 2015 IEEE International Conference on. IEEE, 2015.
7. https://docs.lucidworks.com/display/SiLK/Banana.
8. BiswaRanjan Samal, Mrutyunjaya Panda, HumanBeing Character Analysis from Their SocialNetworking Profiles A Semisupervised Machine Learning Approach, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 14, No. 5, May 2016 .
9. Srinivasan, Srini S., Rolph Anderson, and Kishore Ponnavolu. "Customer loyalty in e-commerce: an exploration of its antecedents and consequences." Journal of retailing 78.1 (2002): 41-50.
10. Oelke, Daniela, et al. "Visual opinion analysis of customer feedback data." Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on. IEEE, 2009.
11. http://lucene.apache.org/solr/
12. BiswaRanjan Samal, Mrutyunjaya Panda,Anil Kumar Behera,Performance Analysis of Supervised Machine Learning Techniques for Sentiment Analysis,2017 IEEE 3rd International Conference on Sensing, Signal Processing and Security (ICSSS)

13. https://nifi.apache.org/
14. Wei-Ping, Zhu, L. I. Ming-Xin, and Chen Huan. "Using MongoDB to implement textbook management system instead of MySQL." Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on. IEEE, 2011.
15. https://www.mongodb.com/what-is-mongodb
16. Kannadasan, R., et al. "An Integrated Approach for Configuring Hadoop Clusters by Ambari on Horton Sandbox."
17. https://hortonworks.com/tutorial/hadoop-tutorial-getting-started-with-hdp/
18. http://jmcauley.ucsd.edu/data/amazon/