International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-7, Issue-6S3, April 2019

Monitoring your Fleets on the Go with RTDFMS [Real time distributed Fleet Monitoring System]

Alok Srivastava, Aditya Tandon, Prashant Mani, Lalbihari Barik

Abstract: ***Fleets are an indispensable part of human life starting from an early phase. There was always the need for a good carrier whether it was Stone Age or the age of modernized advanced technology of today. Sheep's and mules are still the means of transportation in the remote hilly areas where there is no facilitation of good transportation by roads. However, in recent the means of transportation and communication has been evolved so abruptly that almost none of the area is left where we do not have vehicles.Vehicles have touched every part of life. Whether it is the school bus which drops or carries your children or your daily schedule of life - going to Office, visiting relatives or buying goods for daily usages, no part of life is left untouched of it. All because of this, the number of means of transportation are increasing day by day. More than 1.2 billion vehicles are running currently on the road at a time and so as the minimum number of peoples riding the vehicle. So in this contrast, for a public fleet servicing organization or any corporate organization, it has become mandatory to keep concern for the safety of peoples and their lives. For private organizations and companies, tracking of fleets have become mandatory in order to make the productivity of workers smoother and trustworthy.***

Index terms: ***RTDFMS, Hadoop, Big Data, Apache Storm, Spark, Elastic Search.***

I. INTRODUCTION

Imagine that out of 1.2 billion of vehicles, it is required to monitor Hundred Thousand of vehicles. How do you monitor them? Let us consider a simple use case.

Revised Manuscript Received on December 22, 2018.

Alok Srivastava, Assistant Professor, ECE Department, MVN University, Palwal

Aditya Tandon, Research Scholar, Amity University, Noida, UP

Prashant Mani Associate Professor, Department of ECE, SRM Institute of Science and Technology, Ghaziabad

Lalbihari Barik, Asst. Prof., Department of Information Systems, Faculty of Computing and Information Technology in Rabigh, King, Abdulaziz University, Kingdom of Saudi Arabia

Every vehicle is enabled with GPS monitoring system with video capturing functionality of periodic 5 minutes and Geo Location capturing functionality with periodic interval of 15 seconds. Then the number of videos captured per hour is 12 and number of Geo Location events captured per hour is 240. Let us consider, an average size of captured video is 2MB and average size of Geo Location information is 40 Bytes. Then 24MB of disk space is
required per hour for the captured videos and additional 9600 bytes space is required for event data. If the vehicle is monitored per day for 12 hours, 288 MB of video data and 225 KB of Geo location data will be stored at file system. For Hundred Thousand vehicles – 28,125 GB of disk space is required for storing the video data and 21.46 GB of disk space is required for storing Geo Event information. So, approximately 28,150 GB of disk space is required per day for this solution to work.

Now after gathering the statistics, we are primarily concerned of two major factors:

- **Scalability**: Would the system work seamlessly when number of vehicles sending Geo information increases to 80,000 from the initial toll of 20,000.
- **Performance**: How would the system behave under severe load? Would it respond promptly upon the user request or would it hang for hours and crashed.

Well, seems an interesting problem. I approached a traditional way. Using a relational DB - MYSQL and Apache Tomcat server performed exceptionally well, when the numbers of concurrent requests to the server were around 20,000.

When I increased the load beyond it, there are unprecedented results. The server becomes unresponsive and the response time appears too high. Posting a single request takes more than seconds of time. Some requests get failed with exceptions. Taking all those factors in account, it became mandatory for me to be thoughtful about an alternative approach that could handle number of concurrent requests beyond this limit.

II. RELATED WORK

Many of the fleet monitoring solutions exists in the market. They do the excellent jobs on monitoring vehicles based on GPS based solutions. How-ever none of the organizations have solution that would meet the demand of high traffic load on server. The current solutions run on a single traditional high end HTTP servers like – Apache tomcat, Jetty, IBM HTTP server etc. working with relational databases like MYSQL or Oracle. When the server faces concurrent requests greater than certain threshold [twenty thousand requests per unit time], they start showing abrupt behavior with failures and delays.

The Microsoft white paper – Developing Fleets and Assets tracking solutions with Web Maps [17] specifies that it is difficult for management to continuously monitor field operations to ensure that vehicles are operating at maximum efficiency. Some of these vehicle management applications use traditional Geo Graphic information system (GIS) products to display vehicles and asset locations, which add to the complexity and cost. Traditional GIS systems are good for spatial data creation, maintenance, and analysis but they are not designed to handle the larger real time data streams involved with fleet management.

Teletrack a pioneer in fleet monitoring solutions [18] specifies that it's almost impossible to track within a fleet that's why tracking like GPS tracking software's are good to invest to ensure that safety remains priority items for these businesses reducing costly violations and increasing road safety. The solution provided by them are too costly and no assurance of 100% traceability of fleets on the road. They would charge in dollars for the day for the tracking of one vehicle. This cannot be an affordable solution to them, who are economically less stable to pay such a high cost; e.g. [Public servicing organizations]. It brings the necessity to cut down the cost with an optimized implementation of the technological advancements that would ensure the solution to work with commodity hardware, with nearly no fragility, and very low maintenance cost.

III. IMPLEMENTATION

. There are two major bottlenecks found in the implementation of fleet monitoring solution with existing traditionally followed approach.

- Handling of large number of concurrent requests and processing in real time.
- Storage of data to some persistent store that is scalable and efficient on query processing and returning back the response.

Both of these problems can be efficiently handled with Real time distributed fleet monitoring system [RTDFMS]. Let us deep dive into implementation approach.

We must approach distributed publish-subscribe message system that is designed to be fast, scalable, and durable for handling the large number of concurrent requests. So, for this purpose, a distributed, partitioned, commit log service known as Kafka is the best suitable candidate.

A. Kafka

Kafka maintains feeds of messages in categories known as topics. The processes that publish messages to the Kafka topic are called producers. The processes that subscribe topic and processes the feeds of messages are known as consumers. Kafka runs as cluster comprised of one or more number servers which are known as brokers.

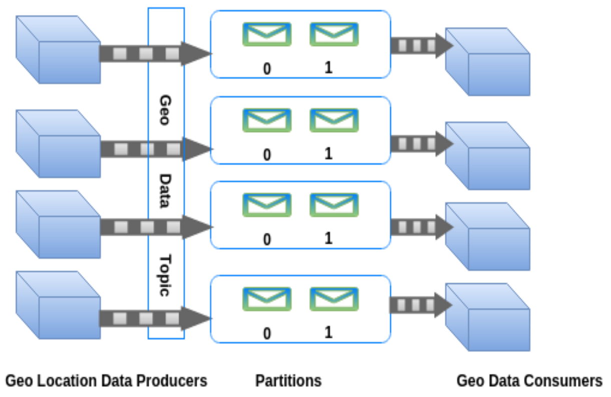

Fig.1 Architecture of Kafka

So at high level the producers send the Geo Location data over the network to the Kafka cluster and the consumer consumes them. The above diagram shows the high level of architecture of Kafka. Well, Kafka has solved our problem by holding the millions of real time streaming tuples in a queue, which were been failing previously because of large number of concurrent requests. Now, the next big challenge is: How do we consume and process these tuples mostly in real time. We are required to have a distributed real time computation system that makes it easy to reliably process unbounded streams of data. The most reliable and an efficient real time computation engine, an Apache Storm is prominent choice to process millions of tuples per second.

B. Apache Storm

Apache Storm is a scalable, fault tolerant open source distributed computation system that enables us to process high volume of data in real time as the Hadoop [19] does it for the batch processing system. Our use case scenario fits the best with the storm as there are large of volume of concurrent Geo location tuples obtained from GPS enabled vehicles at a time. It integrates with the queuing service like Kafka to subscribe the messages and process them in real time.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

A real time computation on Storm is carried out with concept called topologies. A topology comprises of graph of computation. Each node in topology consists of processing logic, and connection link between various nodes that indicates the flow of data across the nodes. The core abstraction in Storm is the stream, an unbounded sequence of tuples. The stream of data is transformed into a new stream in a distributed and reliable using the using the spouts and bolts. A spout is a source of stream. The Geo Location data from Kafka messaging service is read by spout. The bolt consumes the input stream sent through the spout or another bolt. The bolt comprises of the processing logic and other many functions like filtering of tuples, streaming aggregations, streaming joins, communicate with database and persist the

Result to the database, and so on.

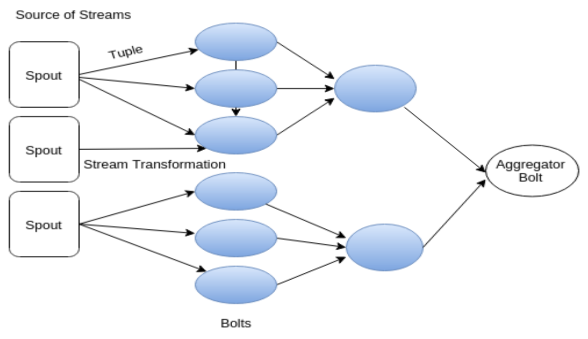

Fig. 2 A conceptual architecture of storm

Our next concern is storage of large volume of video data in the range of Tera bytes to Peta Bytes and its efficient retrieval. The response time for retrieval must be seem less irrespective of how much data is stored on the cluster based on certain key constraints. The obvious choice for this is the selection of NOSQL document store database. Based on various performance parameters, flexibility, and key requirement for query search response time to be seem less, Elastic search is found most reliable.

C. Elastic search

Elastic search is highly scalable, full text search and analytics engine that allows to store, search, and analyze high volume of data in near real time. It is a distributed search engine and document store that uniformly balances the load across multiple nodes in a cluster environment. A cluster comprises of one more node that holds entire data and provides federated indexing and search capabilities across all nodes. It comprises of the following basic components.

a)Index: It is collection of documents having similar characteristics. In our use case Geo-Location is an index for geo location data.

b)Type: Each index can have one or more than one types that indicates a logical category of an index. In our case, we can specify the type as location so that location information for multiple states goes to its own logical partition. This improves the search time drastically with query constraints filter such as type = Location XYZ.

c)Document: It is a basic unit of information that can be indexed to a cluster. The document is expressed in the JSON (Java script object Notation) format there by it is much flexible to operate on as it is a ubiquitous Internet data interchange format.

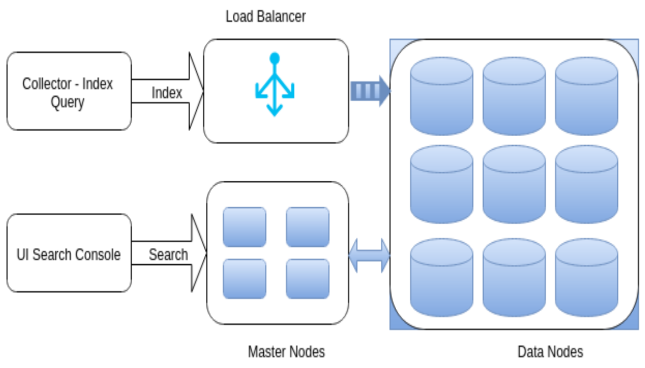

Fig.3 High Level Architecture of Elastic Search

D. ARCHITECTURE

The overall [RTDFMS] system works by integrating all the above components coordinating with each other. The fleet enabled with GPS system would send the Geo event information periodically with certain interval of time to the messaging queuing system Kafka [10]. The Kafka queuing service would acknowledge the message and keep it in a distributed queuing cluster. The messages from Kafka queue are consumed in parallel through one or more spouts of real time processing engine Storm [14]. The Storm processes these messages in the form of tuples and persists to the distributed storage engine Elastic Search [7]. The Elastic search store is exposed with various REST APIs provided by [RTDFMS] that are used to query, retrieve or update Geo information in order to display or manipulate various information events at the front end UI console.

IV.BENEFITS

Easily Scalable: The system would scale uniformly when the load to system increases with effortless addition of additional nodes on the cluster. Hundreds of thousands of vehicles can be monitored through the single unified system.

- ➢ Performance efficient: Irrespective of the high volume of data in Tera-bytes, the system would response in near real time.

- The system would be reliable irrespective of time span for the decades of year without the loss of performance.
- The maintenance of system would be significantly lower.
- Cost effective.
- The System would meet the demand of large enterprises and maximize productivity.
- Service revenue is increased.
- Improvement on customer service.

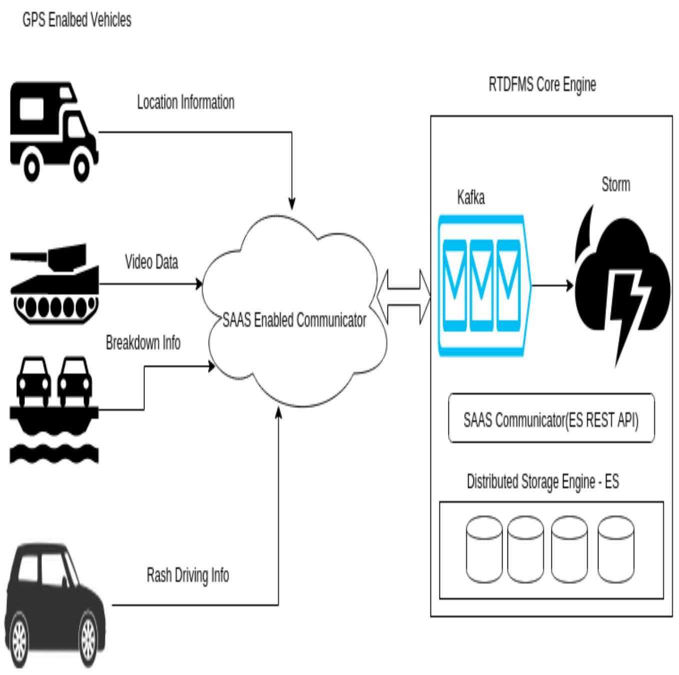

Fig.4 High Level Architecture of overall implementation of RTDFMS

V.CONCLUSION

Real Time Distributed Fleet monitoring system [RTDFMS] would be an advanced technological transition from the traditional fleet monitoring system to a highly scalable, distributed, efficient, and robust fleet monitoring solution that would easily meet the monitoring demand of tremendous volume of fleet tracking information. It would provide the significant value to large enterprises helping them to reduce cost and maximize the productivity.

ACKNOWLEDGMENT

The research leading to these result, authors would like to thank Guru Technology Pvt Ltd (www.nepguru.com) for funding.

REFERENCES

1. [1] Leo f, Top 6 Use Cases to Help You Understand Big Data Analytics, http://www.blue-granite.com/blog/top-6-use-cases-to-help-you-understand-big-data-analytics, Blue Granite, 26 Aug, 2015
2. [2] Prof. Arun D, Big data: The future is in analytics,
3. http://geospatialworld.net/Magazine/MarticleView.aspx?aid=30512, Geo spatial world, April 2013
4. [3] Chris B, Fleet Manager = Big Data Manager, http://www.autorentalnews.com/blog/auto-focus/story/2013/05/fleet-manager-big-data-manager.aspx, AutoRental News, 23 May, 2013
5. [4] Cell phone bus tracking applications developed, http://www.metro-magazine.com/Article/Story/2009/04/Cell-phone-bus-tracking-applications-developed.aspx,Metro Magazine. April, 2009
6. [5] Clinton G, Zachary T, Elastic search – the Definitive Guide, O'REILLY Media Inc, 30 Jan, 2015
7. [6] Eesn S, Our experience of creating large scale log search system using elastic search, https://dzone.com/articles/our-experience-creating-large, Dzone/Performance Zone, 07 May, 2013
8. [7] elastic.co, Getting started with elastic search, https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html, 2015
9. [8] Thomas C, Court Asked To Disallow Warrantless GPS Tracking,
10. http://www.informationweek.com/architecture/court-asked-to-disallow-warrantless-gps-tracking/d/d-id/1077257, Information Week, 4 March, 2009
11. [9] Leibiusky J, Eisbruch G, Simonassi D. Getting Started with Storm, pages 1-3, 3 Aug, 2012
12. [10] Apache Kafka, Getting started to Kafka, http://kafka.apache.org/documentation.html, Version 0.8.2, 2015
13. [11] John G, Alan S, Real-Time Muni Arrival Information Just a Free Call Away,
14. http://www.mtc.ca.gov/news/press_releases/2005/rel316.htm, Metropolitan Transportation Commission, Oakland, California , 13 July, 2005
15. [12] Quinton A, Storm Real-time Processing, Efficiently process unbounded streams of data in real time, Cookbook, Packt Publishing Ltd, 1 Jan, 2013
16. [13] Sean T, Peter P, Matthew J, Storm Applied: Strategies for real-time event processing, 12 Apr, Manning Publications Company, 2015
17. [14] Apache Storm, Apache Software Foundation, Storm Documentation- Introduction to Storm http://storm.apache.org/documentation.html, 0.9.4, 2015
18. [15] Jacob K, Macy's begins iBeacon shopping test, will send alerts to your iPhone when you enter stores, http://www.theverge.com/2013/11/21/5129336/macys-apple-ibeacon-support-herald-union-square-stores-shopkick, The Verge, 21 Nov, 2013
19. [16] Jonathan L, Gabriel E, and Dario S, Getting started with storm, O'REILLY Media Inc., 2012
20. [17] Developing Fleets and Assets tracking solutions with Web Maps], http://download.microsoft.com/bing/maps/develop/fleet-asserts.pdf, 2015
21. [18] Caroline A, How GPS Tracking Software Helps Improve CSA Scores and Increase Safety, http://www.teletrac.com/fleet-management-software/topics/gps-tracking-safety, 2015
22. [19] What is Apache Hadoop, http://hadoop.apache.org/index.pdf, Apache software Foundataion documentation, 2014