Big Data Monitoring and Analysis with Cloud Service Architecture

Pooja Mankawade[1], Poonam Popale[2], Kira Patel[3], Priyanka Talekar[4]
[1, 2, 3, 4] Department of Information Technology
[1, 2, 3, 4] D. Y. Patil College of Engineering, Akurdi

Abstract- *Cloud Monitoring supervises big data that is being continuously produced from traces of applications, organizations and platforms. Big data needs to be analyzed and monitored due to its huge volume with effective architecture. Analysis and monitoring of data gives insights of the system's workload and makes sure that data is operating at optimum levels. Analysis process benefits data analysis, result visualization and query fetching in efficient way. More efficient, reliable and scalable architecture is needed to fetch the data, aggregate it and analyze it in different level of granularity due to its large amount of traces. Since, we are using Big Data so the System Architecture should be flexible with addition of new storage data types and store them. This is why we have built an additional DAO layer on the top layer. REST is an architectural style for communication between Software Systems. Each data point has it's own URL. In. this paper we are proposing a System Architecture combining all latest technologies and is built over Spring Framework where each entity is differentiated from other with the help of DAO Model where Database and Server are independent of each other. This Architecture reduces maintenance cost, increases user experience with the System, can be modified or updated easily. Elastic Search acts as a Shield to our Architecture to secure it and to avoid connection to third party.*

Keywords- tomcat Server, REST API , Big data , Elasticsearch, MongoDB, MySQL, DAO model, MongoConnector.

I. INTRODUCTION

Now a days cloud and its server is being very famous for data storage, because of which it is very easy to access the data from anywhere of your Geo-location. Cloud monitoring is a process of monitoring infrastructures, platforms, and custom metrics to help ensure that workloads are operating at minimum levels. It involves operations for querying and processing large amount of traces, so to overcome this we have developed service architecture consist of technologies like Angular.js, REST, Elasticsearch, DAO model, MySQL and MongoDB and this service architecture is developed in spring framework. In this service architecture the core part of architecture is Elasticsearch, it is recently launched popular search engine which is distributed and multitenant-capable full text search engine. Elaticsearch is focused on storing and searching within database. It has HTTP REST interface and java interface, because of that it can accept REST request as REST strictly use HTTP protocol for data communication. To minimize the coupling between client and server components in distributed application REST will be useful. RESTful model is consist of four major operations of HTTP like GET, POST, PUT, DELETE.

Angular.js is used in client side in the front end which sends minimized URL from REST request. We have to controllers in the architecture one of them is main controller and other one is base controller used for authentication and to check whether the data is in json format or not respectively. DAO model is an abstract interface to database. Database is consist of MySQL and MongoDB as dataset can be both structural and nonstructural data. The benefit of this architecture is that, in this architecture everything is independent. This makes this architecture fault tolerant. Database and servers are independent of each other. Even if server goes down or fails to work, database would be there to provide service as data in the database would not be harmed. JSON is used for data exchange. JSON is used instead of XML, as XML needs more coding than JSON, also JSON is simpler than XML and is human readable. JSON parser is used to convert the replied data to java script object which makes it simpler for client system. The flow of reply to the client can be shown as:

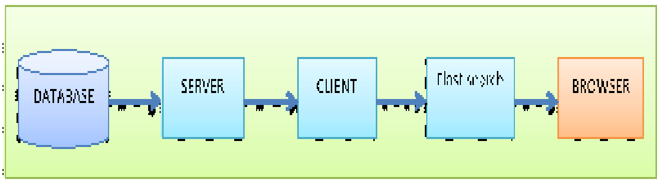

Figure 1. (Flow of Architecture)

Elasticsearch creates index and each index is provided with certain number of nodes. Index in Elasticsearch is used for clustering. Elasticserach can be acted as proxy. This is why, Elasticsearch makes the process to work faster. Queries from browser will go to the ElasicSearch and we will get quick response. In the future, if the data increases tremendously, it can be divided into small chunks and replicas will get created in Elasticsearch as well as in MongoDB. If data is not available in Elasticsearch then it can be directly fetched from MongoDB as Elasticsearch and MongoDB has direct connectivity between them with the help of mongo-connector. It is advantage for us that Elasticsearch accepts REST request. REST Client is used for testing purpose to test that if the request is in JSON format or not. We have used Tomcat server as a depiction of cloud.

Linux ubuntu operating OS can be used to implement this architecture, the implementation can be executed in the JVM environment independent of the Operating System. The Architecture we are designing is yet not proposed and is advanced. The paper is entitled with the cloud keyword, but instead we can also use Apache Tomcat, GlassFish, Jboss for running on server as a cloud. The data is stored in elastic, MongoDB with replicas in both mongo and Elasticsearch, the advancement of using such new adaptive technologies is required as the data stored in cloud is not effective if the cloud server shuts down due to some unexceptional errors and also it is not cost effective. The architecture operates with the web services i.e REST APIs and Json's for the fast communication of data between the Architecture (client, server, database). The architecture is web based and is platform independent and is compatible with all browsers, it can be made advanced by making it compatible with the mobile devices.

II. LITRATURE SURVEY

There are some existing cloud monitoring services such as Cloud Watch, Azure Watch, Nimsoft, and Nimbus are delivered by cloud service providers as part of the Infrastructure as a Service (IaaS) model. They can be launched within an integrated cloud management console. To some extent, these services have a limitation in supporting analysis such as workload forecast and pattern matching beyond the simple built in aggregation.[1]

As we know big data can't be stored on a single machine and backup of big data is also an issue. Before we get into some of the ways it can be stored, it is important to note that this is a very active research area, and any system architecture should be able to cope with addition of new storage types. This is why we chose to build an additional interface on top of our storage layer. Another issue to keep in mind is the CAP theorem sometimes called "Brewer's Theorem", it states that a storage implementation could choose any two properties from Consistency, Availability and Partition tolerance, but not all three. Big data storage systems can be divided according to which property they sacrifice in order to achieve the other two. Those that give up consistency are called "eventually consistent". In such systems, the data may not be consistent for a bounded amount of time. [5]

Representational State Transfer (abbr. REST or ReST) is an architectural style for communicating between Software Systems. It uses the HTTP protocol for communication, thus offers a simple alternative to RPC, CORBA and SOAP/WSDL. Under REST each data point has its own URL (Universal Resource Locator, in a way restoring this term's original semantics). Client applications act on data points using their URLs and standard HTTP verbs, such as GET, POST and DELETE. Because of the ubiquity of HTTP support, systems that use "RESTful" architecture can be communicated from virtually any programming environment. First introduced by Roy Fielding in his doctoral thesis, REST is used today in many systems including Twitter, Flicker and Instagram in their public API. In the context of our proposal, using REST will allow us to abstract the storage layer and enable proxy servers and caches to speed up access times -as data points are rarely updated once published.[8]

Elasticsearch stores documents in an index allowing you to search them. The index is a container for documents. You can have multiple indexes in your cluster of Elasticsearch nodes. Documents are typed. A type has a list of fields that are in the documents of that type. ElasticUtils calls this a "mapping type" or a "doc type" since the word type is somewhat ambiguous depending on the context. Elasticsearch is concerned with storing and searching within the database. It is highly distributed search engine.[3]

A "sharded" relational database has at least one of its tables split between different machines. This early form of big data storage is still in use today. MongoDB is a NoSQL database which uses the "document model" to store data. In this model, loosely based around the JSON format, records consist of (key, value) pairs. Keys are strings, and values may be numbers, strings, arrays or objects. As objects are themselves sets of (key, value) pairs, this structure allows for unbounded nesting. Graph databases store graphs nodes and directed edges, with properties for both. As graph theory is central to computer science, these data stores have many uses. Numerous libraries with implementations of graph algorithms are available -some databases come with common algorithms built in.[5]

III. PROPOSED SYSTEM

As related to our base paper and considering Existing system in mind which is used to manage or handled Big Data monitoring, we are designing a Robust and Scalable architecture which will be independent of the platforms, minimal hardware's, operating systems and which can work over the Distributed Systems. We are using the advanced Technologies i.e., the group of advanced technologies to make our Software Architecture more Reliable and long lasting.

In our proposed Architecture we are using the REST full web services to handle and process the requests and responses from server to clients, the requests includes GET, POST, PUT, DELETE and the responses are send to the client side with the proper and compact response codes and selected key along with it to verify the response coming from server in client-side easily without entering the whole data or without parsing the whole data, here we are introducing JSON (Light weight data transfer) to transfer the data in our whole Architecture i.e., from client to server and from server to database whereas an another option enabled is between database and client directly to minimize the response time for the repeated and to aviod unnecessary processing's done to server for the repeatations of data.

In front-end i.e., in client side we are AngularJS Framework which is built over Javascript language, due to its asynchronous behavior and framework support we can secure our data even from client side using authentication Api's.

Elasticsearch in today's world called as the fastest search tool or technology is used for secure and scalable search on user end, here again we can use Shield of Elasticsearch to secure our Architecture and avoid third party connections to our system.ES works in terms of Indexes which is again divided into nodes and is able to keep replicas of the searched data independent of the browser cache and same can be used for further data analyzing and Monitoring.

On the Database side we are using MySQL and MongoDB, the combination of Structured and Unstructured Databases to provide good user experience i.e., as per the requirements the selected databases will come in contact with the user reducing the cost and maintenance od databases although. MongoDB is specially used to handled the Unstructured data and whereas MySQL is used for storing the current user information to make sure that authenticated user is operating the database by cross verifying the details in both the databases.

We are proposing a Model of SA which is combination all the latest Technologies and is build over the spring framework where each entity is differentiated from other with the help of DAO model, where the database and server are independent of each other.

Hence we are about to develop a robust and rigid architecture which reduces the maintenance costs, increases user experience with the system and can be modified easily i.e., can be updated at any instance without referring the whole application which makes it unique in the Software Development Sector.

IV. SYSTEM ARCHITECTURE

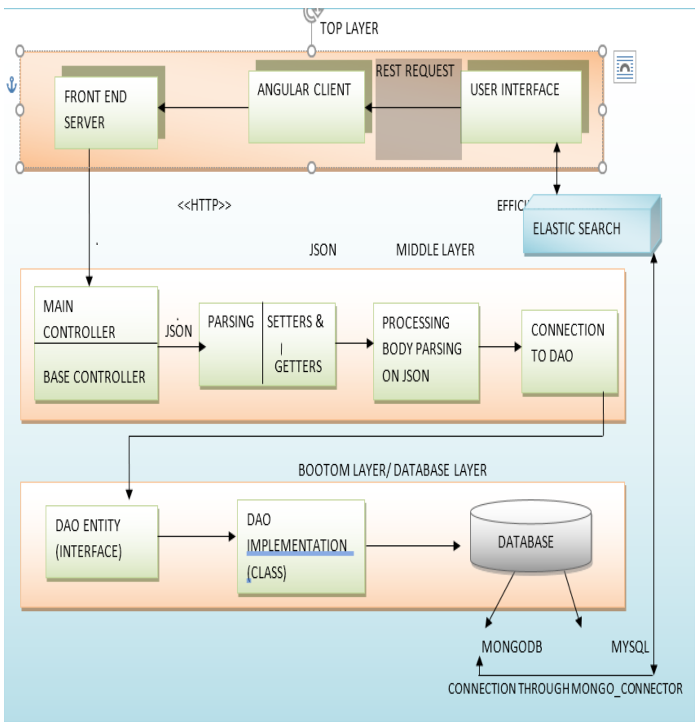

Figure 2. (Proposed System Architecture)

REST full web Services:

- RESTful web services are built to work best on the Web. Representational State Transfer (REST) is an architectural style that specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web. In the REST architectural style, data and functionality are considered resources and are accessed using Uniform Resource Identifiers (URIs), typically links on the Web. The resources are acted upon by using a set of simple, well-defined operations. The REST architectural style constrains an architecture to a client/server architecture and is designed to use a stateless communication protocol, typically HTTP. In the REST architecture style, clients and servers exchange representations of resources by using a standardized interface and protocol.

- A RESTful web service exposes a set of resources that identify the targets of the interaction with its clients. Resources are identified by URIs, which provide a global addressing space for resource and service discovery.
- Resources are manipulated using a fixed set of four create, read, update, delete operations: PUT, GET, POST, and DELETE. PUT creates a new resource, which can be then deleted by using DELETE. GET retrieves the current state of a resource in some representation. POST transfers a new state onto a resource.
- Resources are decoupled from their representation so that their content can be accessed in a variety of formats, such as HTML, XML, plain text, PDF, JPEG, JSON, and others. Metadata about the resource is available and used, for example, to control caching, detect transmission errors, negotiate the appropriate representation format, and perform authentication or access control.
- Every interaction with a resource is stateless; that is, request messages are self-contained. Stateful interactions are based on the concept of explicit state transfer. Several techniques exist to exchange state, such as URI rewriting, cookies, and hidden form fields. State can be embedded in response messages to point to valid future states of the interaction.
- RESTful Web Services are basically REST Architecture based Web Services. In REST Architecture everything is a resource. RESTful web services are light weight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications.

JSON Data Carrier over Network:

- JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.
- JSON is a syntax for storing and exchanging data.
- We can also convert any JSON received from the server into JavaScript objects.
- JSON is easy to integrate in JavaScript applications, but XML isn't. This makes JSON a preferred data format with many mobile application developers.
- JSON and XML are different encoding mechanisms for describing structured data. JSON tends to be a more efficient encoding mechanism, so a typical JSON message will be smaller than the equivalent XML message.
- JSON Web Services let you access portal service methods by exposing them as a JSON HTTP API. Service methods are made easily accessible using HTTP requests, both from JavaScript within the portal and from any JSON-speaking client
- Json Parsing is very easy as compared to XML parsing i.e time effiecient by giving fast response and reducing network load due to its light weight Nature.

Spring Framework:

- The Spring Framework is an application framework and inversion of control container for the Platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Technology platform. Although the framework does not impose any specific programming model, it has become popular in the Technical community as an alternative to, replacement for, or even addition to the Enterprise application.
- Spring is light weight framework because of POJO model.
- Spring framework is said to be a non-invasive means it doesn’t force a programmer to extend or implement their class from any predefined class or interface given Spring API, in struts we used to extend Action Class right that’s why struts is said to be invasive.
- Spring Framework is well known for its loose coupling behavior.
- Below is spring framework architecture for understanding the core body part and its working accordingly.

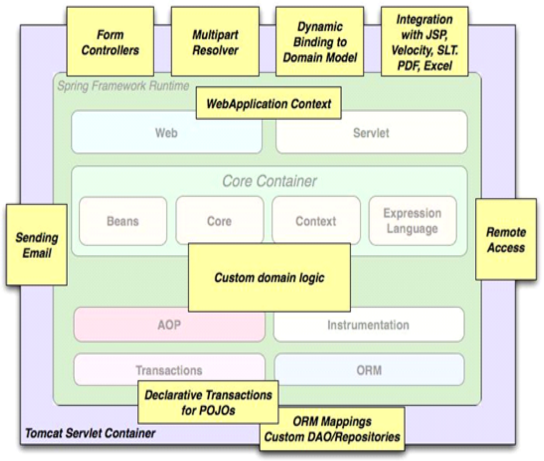

Figure 3.

Elasticsearch:

- Elasticsearch is a search engine based on Lucene. It provides a distributed, multitenant-capable full-text search engine with an Http web interface and schema-free Json documents.
- Elasticsearch is able to achieve fast search responses because, instead of searching the text directly, it searches an index instead.
- In Elasticsearch, an index may store documents of different "mapping types". You can associate multiple mapping definitions for each mapping type. A mapping type is a way of separating the documents in an index into logical groups.
- Unlike Solr, Elasticsearch is schema-free.
- Elasticsearch has been compared to Apache Solr and offers several notable features:
- Provides a SCALABLE search solution.
- Performs near-real-time searches.
- Provides support for multi-tenancy.
- Streamlines backup processes and ensures data Integrity
- An index can be easily recovered in a case of a server crash.
- Uses Javascript Object Notation (JSON) and Java application program interfaces.
- Automatically indexes JSON documents.
- Each index can have its own settings.
- Can search and index document files in diverse formats.

MongoDB:

- MongoDB uses a document-oriented data model.
- MongoDB is a cross-platform and open-source document-oriented database, a kind of NoSQL database. As a NoSQL database, MongoDB shuns the relational database's table-based structure to adapt JSON-like documents that have dynamic schemas which it calls BSON.
- Ad hoc queries - supports search by field, regular expression searches, and range queries.
- Indexing - any field in the BSON document can be indexed.
- Replication - provides high availability via replica sets which consists of two or more copies of the original data.
- Load balancing - sharding is the method used to allow MongoDB to scale horizontally, meaning that data will be distributed and split into ranges and then stored in different shards which can be located in different servers. Shard keys are used to determine how the data will be distributed.
- Aggregation - MapReduce can be applied to enable batch processing of data as well as perform aggregation operations.
- MongoDB provides auto-sharding for horizontal scale out. Native replication and automatic leader election supports high availability across racks and data centers. And MongoDB makes extensive use of RAM, providing in-memory speed and on-disk capacity.
- MongoDB became one of the most popular NoSQL databases, being used as the backend for many major websites including eBay, Craigslist, SourceForge and The New York Times.

DAO:

- Data access object (DAO) is an object that provides an abstract interface to some type of database or other persistence mechanism. By mapping application calls to the persistence layer, the DAO provides some specific data operations without exposing details of the database.
- In the real systems, there usually is DAO for each entity as they are typesafe.
- Data Access Object Pattern or DAO pattern is used to separate low level data accessing API or operations from high level business services.
- DAO allows user to make server independent of the database, which in future avoids many crashes related to server due to Databases complexities.

V. CONCLUSION

In this paper, we proposed an architecture that integrates Elastic Search and search based clusters by REST request to support the acquisition of cloud monitoring data in more efficient way. This architecture is based upon spring framework which benefits and is effective in organizing the middle tier applications. Due to spring framework, seventy percent of work is reduced. The search based cluster built with the help of Elastic Search enables indexing of large size of data, and thus makes this architecture suitable to work on ever accumulating data such as the traces produced from data centers.

This paper is architecture based in which the proposed architecture is vital part of the model. DAO layer enhances the allowance to create a nice abstraction layer of the actual storage system. Use of mongo-connector reduces the overhead of data to pass through every layer of architecture as database is directly connected to elastic search. There is always a scope for improvement of this architecture in various ways. To implement these changes, there is a need to be introduced to various upcoming new technologies which are compatible with this architecture. We can evaluate the performance and fault tolerance of this architecture by experiment. This architecture can be called as fault tolerant because everything including database and server is independent. Data optimization is the key point of this architecture.

V. ACKNOWLEDGEMENT

We take this opportunity to thank our project guide and Head of the Department (Information Technology) Mr. A. J. Patankar for their valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this research paper. We are also thankful to all the staff members of the Department of Information Technology of D. Y. Patil college of Engineering, Akurdi for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the institute for providing the required facilities, Internet access and important books.

REFERENCES

[1] Samneet Singh and Yan Liu- " A Cloud Service Architecture for Analyzing Big Monitoring Data", TSINGHUA SCIENCE AND TECHNOLOGY ISSN 1007-0214 05/10 pp 55-70 Volume 21, Number 1, February 2016.

[2] Eliot Horowitz, MongoDB CTO and Co-Founder- "MongoDB Architecture Guide", A MongoDB White Paper, 2015.

[3] Ronit Salunkhe, Sandeep Telang, Prachi Shrigondekar, Amruta Tanpure- "Review of RESTful Service Using MEAN Stack for Real Time Big Data Architecture", International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 4, Issue 11, November 2016.

[4] Creative commons BY-SA 3.0- "REST Elasticsearch Dropwizard",5th of may 2015.

[5] Adamczyk , Paul Patrick H. Smith, Ralph E. Johnson, and Munawar Hafiz –" REST and Web Services: In Theory and in Practice",2011.

[6] "HPSTRA Elastic search White Paper", Software AG, 7th of October 2015.

[7] Pon Prabakaran Shanmugam- "Real-Time Data Access Using Restful Framework For Multi-Platform Warehouse Environment", Wipro applying thought, 2016.

[8] Deepak Kumar- "Best Practices For Building RESTful web services", Infosys, 2015.