International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137

Volume – 02, Issue – 03, March – 2017, PP – 59-62

Management of Huge File Attachments and Speedup the Web Applications Response using Hybrid Database Architecture

Saundarya S[1],
Mrs. M. Maheswari, Associate Professor[2]
[1](Computer Science and Engineering, Panimalar Engineering College/ Anna University, India)
[2](Computer Science and Engineering, Panimalar Engineering College/ Anna University, India)

Abstract: Now a days, the web applications especially e-governance applications are becoming slower because of the heavy load at the backend especially due to the file uploads and downloads, and not processing it efficiently. Thus in this research, we introduce a new hybrid database architecture for efficiently managing the file attachments and thereby improving the speed of the web applications' response. Also, by using this architecture as the back end, all three factors of CAP theorem: Consistency, Availability and Partitioning Tolerance can be attained. Thereby a biggest limitation of web servers called data-aging can be eliminated. That is, there is no need for removing old data from secondary storage to allow the associated media to be reused for future backups, anymore. Though there are several Bigdata management techniques, our proposed system is differentiated by its completely open source components and by its much simpler process for configuring the backend unlike other Bigdata tools like hadoop, hive, etc.

Keywords: Bigdata Analtics, Efficient Backend Processing, Hybrid Database Architecture, Innovative Backend Datastore, eGovernance Huge Dataset Management

1. INTRODUCTION

The main aim of our project is to develop a Hybrid and Innovative Database Architecture for speeding up the Web Applications' response for the e-Governance Projects in which there is a requirement of huge and unlimited file attachments and no Data Aging, by using efficient techniques and components. Today, most of the sites are based on RDBMS and data aging is never in the agenda of users. Most applications fall under the category of e-Governance which run on 24x7 basis. Data size, as on date, is in TB's in applications like Transport, e-District. The number of applications having TB database size is bound to increase. Database Server's role is crucial for quicker response to the end user (mostly Citizen) compared to role of Application-Server or Web-Server. Database Server performance is proportional to the number of CPU's and size of the RAM. Increasing CPU and RAM Size every year is not possible due to restriction in hardware. Users are not convinced in increasing CPU/RAM as option. Users always quote the performance of sites like Google, IRCTC and Social Media. RDBMS by nature cannot have a Multi-Master Cluster except in commercial products at a huge cost.

2. OBSERVATIONS

In the existing system, almost all the e-governance web applications involves the following: Upload photo / scanned-documents as part of application which are then verified by concerned officials; These documents, which are stored in the database, are critical since the local officials do not retain any paper at their office or Common Service Centers while accepting application from Citizens. Data size, as on date, is in TB's in applications like Transport, eDistrict. The number of applications having TB database size is bound to increase. Database Server's role is crucial for quicker response to the end user (mostly Citizen) compared to role of Application-Server or Web-Server. Database Server performance is proportional to the number of CPU's and size of the RAM. Increasing CPU and RAM Size every year is not possible due to restriction in hardware. Users are not convinced in increasing CPU/RAM as option. Users always quote the performance of sites like Google, IRCTC and Social Media. RDBMS by nature cannot have a Multi-Master Cluster except in commercial products at a huge cost. PostgreSQL is the widely used community based RDBMS as on date in all eGovernance projects.

3. EXISTING SYSTEM

It is essential to upload photo / scanned-documents as part of application. These are verified by concerned officials. These documents, which are stored in the database, are critical since the local officials do not retain any paper at their office or Common Service Centers while accepting application from Citizens. On

demand by any authority at any time, it should be available to produce the uploaded documents to the higher authority. Two options are exercised while handling the attached-documents or photos. They are, either to store them in the Database or to store them in the File System. Both approach have pros and cons which can be overcome by proposed system.

4. FILE SYSTEM APPROACH FOR ATTACHMMENTS

It is difficult to monitor the availability of space in the file system when user is uploading document; this may lead to loss of data. Creating and maintaining Replica of the uploaded document is very challenging. There are few commercial tools available for creating Replica. Generally rsync / lsync are used in NIC. There are some technical challenges in using these commands to maintain the Replica. Always there are security issues in this approach. s The uploaded documents can be deleted or manipulated by anyone who have access to the system. Auditing is cumbersome. There is a possibility of file corruption / truncation due to OS related issues during rebooting in case of abrupt shutdown particularly with physical hard disk partitions. SAN has better facility to recover unlike the physical hard disk partitions.

5. DATABASE APPROACH FOR ATTACHMENTS

PostgreSQL stores the database as part of bytea / text field-type. The data stored is not kept with table. The data spreads across pg_largeobject & pg_toast and set of tables. As the table grows, there is definite slowdown during insertion. Deletion/ Updation of records will fragment the table and severely slow down the access. Careful planning is essential to remove old records and keep it for archive. Aggressive vacuum (reclaim space due to deletion of records) is essential when the records are accumulated for longer period. This requires andmore CPU resources. Effective aging is the prudent way. Every year, few days the website had to be shutdown and remove the processed records and attachments from active database. The records & attached-documents which were removed in the earlier step should be kept as part of the separate archived-database which can be mounted on-demand for any display / reference at a later time. RDBMS by nature is bound to Single Box and hence only vertical scalability is possible which is restricted by the server's hardware configuration. Multi-Master setup feature with PostgreSQL is still evolving. Hence we have to go for Distributed approach using NoSQL store for storing and rendering the attached documents which consume 90% of the space for any citizen-centric applications.

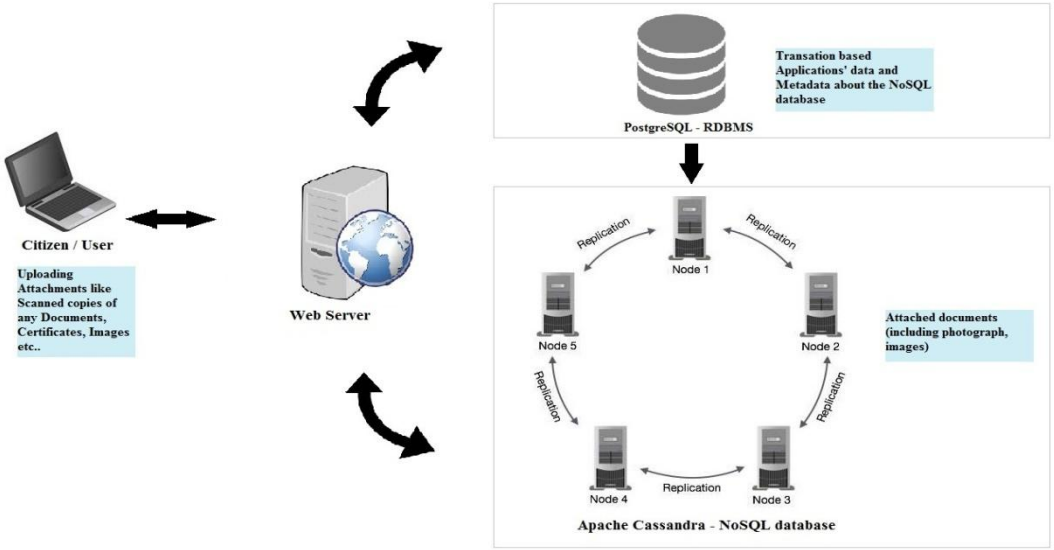
6. PROPOSED SYSTEM

Considering the above facts, we are presenting a demonstration of having hybrid-sample-application where in the Transaction-part is in RDBMS and attached documents (including photograph, images) are on Apache Cassandra Cluster. There are many NoSQL Databases which can be considered as alternatives to store the attached documents; they do not have the issue of fragmentation due to deletions / updations & related degradation of performance which are common in RDBMS. Apache Cassandra NoSQL Database is unique since all other NoSQL stores require Hadoop environment in every node. The environment include HDFS (File system) ,Zoo-Keeper (Monitor the cluster) , Pig/Hive like API to query the database. Hadoop-setup, as mentioned above, is too difficult to setup & manage with limited resources and manpower. Apache Cassandra-cluster setup using physical hard disk partitions can be done in minutes. All the participating nodes in the Cluster should have NTP configured. There are drivers (in many languages including Java, PHP, etc) available in the net to process the data with all features like parallel-querying, Spark-real-time-processing, etc. We can easily add / remove / repair the Apache-Cassandra-Cluster live.

7. APACHE CASSANDRA NoSQL DATASTORE

Apache Cassandra is available as Open Source. Cassandra uses a protocol called GOSSIP for data-synchronization among the participating cluster-nodes. The basic requirement to start a Cluster at a node is simply JRE-8 for the Cassandra version 3.7 and above, which is open source. Both Open JRE and SUN JRE are compatible. The other unique feature with Cassandra is it has almost SQL like commands to fetch records except JOINS. Since in our proposed architecture, this NoSQL store will be used only to store the attached documents in which there is no need for any JOIN query. Cassandra has all other features of RDBMS like databases, tables, views, functions and data-types which are compatible with those available in RDBMS unlike other NoSQL Databases. Every table must have one primary key. All the columns except primary key will be stored with time of updating since, there is always a possibility of stale record returned. This is part of the eventual consistency. It has all security features like users, roles, even SSL and other secured-access-schemes. One instance of Cassandra can have multiple databases assigned to different users (multi-tenant). Flexible schema per table is possible.

8. HYBRID DATABASE ARCHITECTURE



9. CAP THEOREM

The CAP theorem, also named Brewer's theorem after computer scientist Eric Brewer, states that it is impossible for a distributed computer system to simultaneously provide more than two out of three of the following guarantees: Consistency (Consistency in database systems refers to the requirement that any given database transaction must change affected data only in allowed ways. Any data written to the database must be valid according to all defined rules, including constraints, cascades, triggers, and any combination thereof.), Availability (availability is the condition wherein a given resource can be accessed by its consumers at any given tim), Partition tolerance(). i.e. any Database be it RDBMS or NoSQL-Store has at-least two features out of three which is mentioned above. But in our proposed system, we have overcame this drawback by having Hybrid Database Architecture (Any RDBMS, for instance postgreSQL for storing Transaction based information and the metadata and Appache Cassandra NoSQL datastore for storing attachments) we can attain all the three guarantees namely, Consistency, Availability, Partition tolerance.

10. MODULES

10.1 Configure the No-SQL database and PostgreSQL database (Back-end)

To configure th no-SQL datastore to manage file attachments, first the apache Cassandra should be installed in a server node and while installing, the number of replicating nodes with its IP address is configured. It may even be altered after the installation process also. The modification process of specifying the additional nodes is done at the seeds /etc/cassandra/cassandra.yaml as follows:

```
seed_provider:
- class_name: org.apache.cassandra.locator.SimpleSeedProvider
parameters:
- seeds: "server_ip,server_ip_2,...server_ip_n"
```

For configuring PostgreSQL database for storing the metadata and Non-Transaction Based data(in order to maintain concistency), the postgreSQL executable file is downloaded from online and installed in the main server; but all the eGovernance applications are already using PostgreSQL as the backend. So, for maintaining Backward compatabilty, we are not changing any table structure that is already present in the existing system. But adding one more table with exactly the same structure except the column with datatype : bytea(i.e. the file attachments which will be stored on Appache Cassandra No- SQL store).

10.2 Server Deployment

According to the number of nodes to be replicated, the number of servers are deployed with additional one server with PoostgreSQL database. For instance, we shall take five nodes of Appache Cassandra No-SQL datastore with one PostgreSQL database as given in the architecture diagram. So there are Six Servers that are now available to respond to the citizens when they query. The biggest advantage of using No-SQL datastore is availability at any point of time. But when we consider the situatuion when citizen is trying to upload a detail

with some attachments within and at that point of time and all the five nodes of Cassandra failed to respond, there arise the same situation "unavailability" again. Thus, to overcome this problem, we are directing the attachments also to the postgreSQL when none of the Cassandra node is available for storing the documents attached. Or else, if it has not been directed like this, then there occurs a inconvenience to the citizens i.e. they have to once again enter all their details and then upload it once again just because a single attachment has not get uploaded.

10.3 Front-end Web-Application Development

The front is a simple webpage that contains some text, date and other datatype fields along with bytea fields that should be input by the citizens when they are to upload some information, and there is an option of 'show details' that when clicked by the citizen, must produce the respective details along with some bytea type fields as requested. So both for input and showing output, the webpage should be designed. This webpage is designed using the Web Application development languages like HTML5, CSS3, JSP. AJAX is also used for adding better user interface experience.

10.4 User Input Analysis

After the user has entered all the details and clicked the option 'upload' the input fields are analysed. All the fields other than the fields that needs huge space to get stored compared to simple fields like 'bytea' , 'text'(unbounded text size permitted) are directed to PostgreSQL whereas, those exceptional fields are directed to the No-SQL apache datastore. These two database tables are linked and referred by the Primary key of both tables. So whenever a citizen or even some higher authority queries for some information, both the metadata from RDBMS and attachments from Cassandra are retrieved by matching the primary key of the RDBMS table. Thus JOIN queries are made possible which would not have been if Cassandra alone is used to store both the metadata and the attachments.

11. BENEFITS

Fast uploading and retrieval of voluminous attachments (scanned images, photos, documents, etc) since the availability is been enhanced through this new hybrid database architecture. Enable the Web Applications to support both Consistency and Availability simultaneously. Flexible schema per table is possible. Thus it ensures low latency and high performance. Highly scalable since the number of nodes can be increased at an point of with less number of steps than the existing system. As everything is OpenSource, it is Cost Effective i.e. only the server deployment is the only thing that should be bought and other software requirements are open source and freely available in the internet. There will be no Single point of failure of the service.

12. CONCLUSION

Thus, the proposed new Hybrid Database Architecture will be used to manage the voluminous attachments uploaded / downloaded through Web Applications. User can upload large number of attachments (JPEG, PNG, PDF, mp3, mp4, etc.) without any size limitations. Response for the Users' Queries through Web Applications will be faster at the same time ensuring Consistency and Availability. So, when the Web Applications especially the e-Governance application will be made to response faster even with huge number of users requesting for http sessions especially during the announcement of some news or results or issues that is expected by huge number of citizens at the same time, where everyone will be trying to access the same site at a same point of time.

REFERENCES

Journal Papers:

[1]. Olga Kurasova, Virginijus Marcinkevicius, Viktor Medvedev, Aurimas Rapecka and Pavel Stefanovic, Strategies for Big Data Clustering, Institute of Electrical and Electronic Engineer, 15 December 2014.

[2]. Saurabh Arora, Inderveer Chana, A survey of clustering techniques for big data analysis, Institute of Electrical and Electronic Engineer, 15 December 2014.

[3]. Ajay Jangra, Vandhana Bhatia, Upsana Lakhinaza, Niharika Singh, An efficient storage framework design for cloud computing: Deploying compression on de-duplicated No-SQL DB using HDFS .

Books:

[4]. Pramod J. Sadalage, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence (Paperback – Import, 8 Aug 2012).

