

GeoWave Monitor: Map-Based Accumulo

Status and Health

[GeoWave Monitor: Map-Based Accumulo Status and Health](#)

[Introduction](#)

[Proposed Solution](#)

[Architecture](#)

[Implementation Details](#)

[Table Stat](#)

[Tablet Server Stat](#)

[Tablet Stat](#)

[Namespace Operation](#)

[Geospatial Extent](#)

[Background Worker](#)

[REST Service](#)

[http://<host>/webapp-context/stat/listns](#)

[http://<host>/webapp-context/stat/listtable/{ns}](#)

[http://<host>/webapp-context/stat/table](#)

[http://<host>/webapp-context/stat/tablet/{table}](#)

[http://<host>/webapp-context/stat/tablet/{table}/{tabletId}](#)

[http://<host>/webapp-context/stat/geo/{table}](#)

[Web Application](#)

[Intergration](#)

[Result](#)

[Reference](#)

1. Introduction

Geowave is a tool to store, index and search Geographical data on top of the key-value pair data storage. Geowave is built on top of the Geoserver and work on top of the apache accumulo. Apache accumulo is one dimensional data representation using key value pair. Geo wave provide the way to Multi Dimensional data representation to accumulo from Geowave. Also with that it get the capabilities to do the Geo Graphical data operations to the Accumulo data.

Apache Accumulo provide the Accumulo overview page to provide Statistical data based on the Accumulo Data. It include the Table, Tablet and Tablet server based statistics. In this project,

idea is to provide same set of statistical information in a different manner. Using same Data, represent them as a Geospatial information will give and very descriptive and easy interpretation to the users. Thus, using Geo spatial information represent Stat related to the Tablet and Tablet information display on a Map.

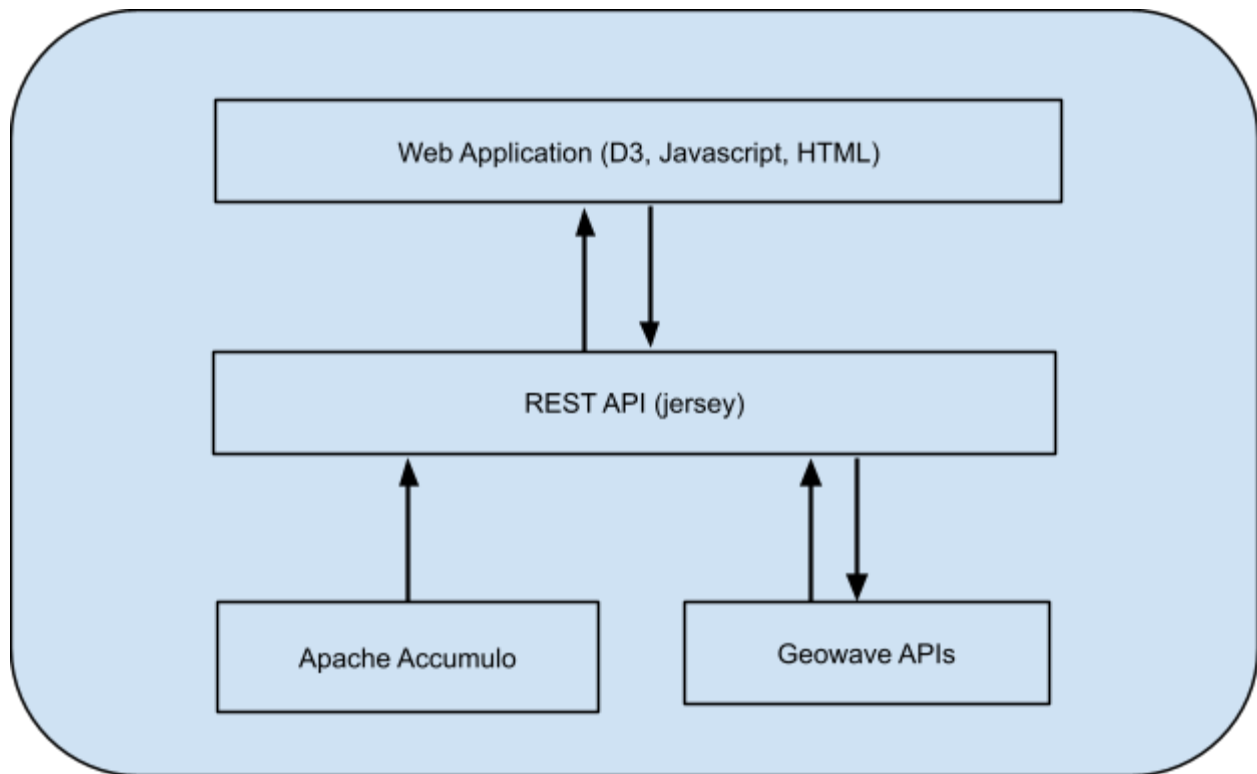
2. Proposed Solution

Solution to this scenario was implementing Map-based Monitor for Apache Accumulo. Application is run as web application and implemented as extension to the Geowave. Which connect to the Accumulo and related component to extract and fetch Stat related information using Existing or implemented REST api.

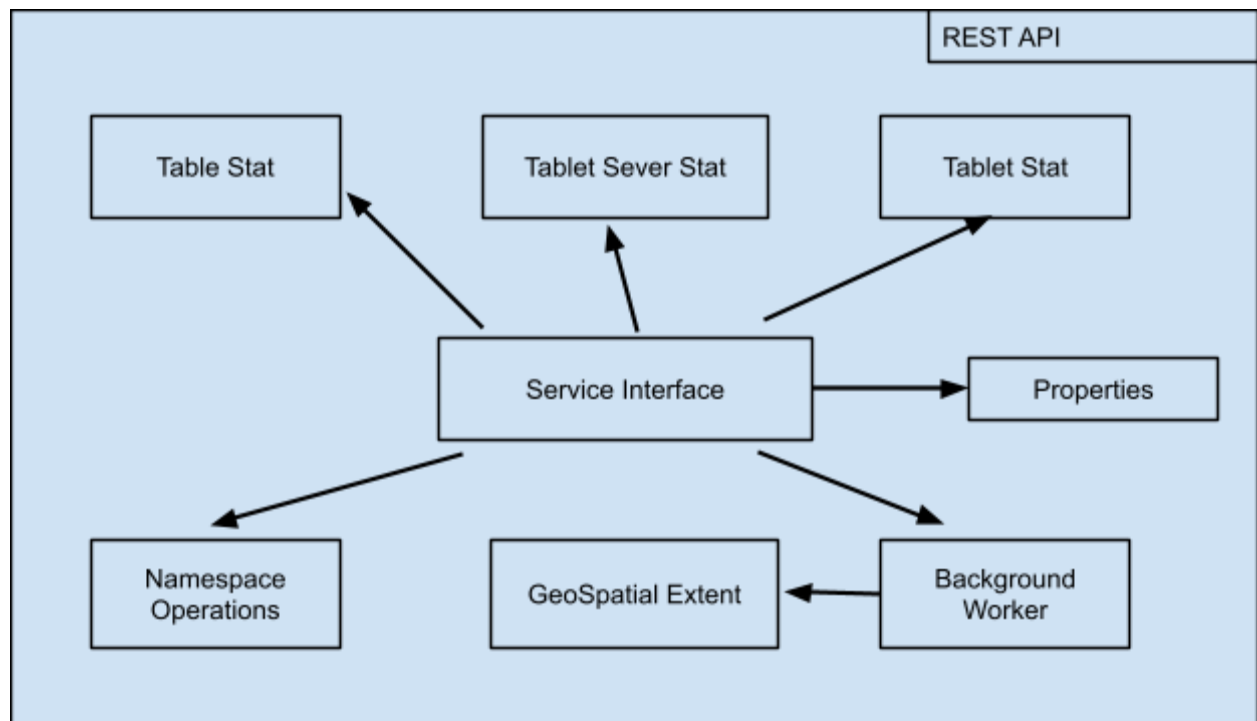
Fetch data is again reformatted and interpreted into Geographic information to display it on a map. Which should contain the Table and Tablet related Statistical data and vice verse. Application implemented as a two component containing web application and REST api. Web application use the implemented or existing REST api to fetch and reformat the data. Both of the component is integrated with the Geowave at the end.

3. Architecture

Architectural solution for the above solution is described as below.



Overview Architecture



REST API implementation Architecture

- **Table Stat**
 - Responsible to fetch and format the Table related Statistical information
- **Tablet Server Stat**
 - Responsible to fetch and format the Tablet Server related Statistical information
- **Tablet Stat**
 - Responsible to fetch and format the Tablet related Statistical information
- **Namespace Operation**
 - Namespace related table data operations. Table identification in a specific namespace etc.
- **Geospatial Extent**
 - responsible to extraction of the Geospatial features of a Tablet. Also finding the Convex hull of the tablet extent
- **Background Worker**
 - Background worker is daemon thread running periodically to compute the tablet convex Hull. Convex hull calculation is bit Time Cost operation it is running about once a 5min or similar time period.
- **Properties**
 - Properties related to Geowave and Accumulo related modules like Zookeeper etc.
- **Service Interface**
 - Which is the REST service interface. which expose the above implented functions as a Service via REST.

4. Implementation Details

a. Table Stat

this component is responsible for extracting statistical information associated with tables. Here it use Accumulo inbuilt implementation from "MasterMonitorInfo". Using it's "getMasterStats()" method it is possible to extract the accumulo related statistical information. It contain the most of the Statistic information including tablet,table and tablet server stats. To filter the table related statistic, here it use the "tableMap" property. It contain the all the tables related stat data. Next data again moved to the Jax-bean called "TableBean.java". Data is set into the "TableBean" after the data parsed to the appropriate data formats.

b. Tablet Server Stat

This component is responsible to extract the Tablet server related statistic data. Tablet servers are physical nodes which are storing data chunks (Tablet) of the tables. For this as well it uses the "MasterMonitorInfo" class. To filter the Tablet server related data it uses the "getTServerInfo()" method to filter it out. It contains the list of "TabletServerStatus" component by holding table statistics related to each tablet server. Using the above method it is not possible to get the tablet server statistic data directly. Thus what it has to do is, it needs to aggregate all the values of each table in each tablet server to get the final tablet server related statistic data. After the data are extracted it is formatted and moved to "TabletServerBean" object.

c. Tablet Stat

Tablet Stats component extracts the Tablet related statistics. Tablets are physical representations of the Logical tables. Logical tables are represented as collections of the Tablet by dividing the table row wise. These tablets are assigned to different tablet servers randomly. Statistics extraction implementation is done using both "MasterMonitorInfo" class and "TabletClientService.Client". These data are again formatted and represented as "TabletBean" class.

d. Namespace Operation

Namespace operation is responsible to retrieve the namespace related table information. Namespace is something specific to the Geowave. It is a standard Accumulo Table. But Geowave creates specific Tables which data representation is different from Accumulo data. Accumulo's initial data representation is still preserved here. But Geowave defines additional parameters and column families to support Geowave operations. In that case, to identify the Geowave related table here they use namespace. Namespace is identified by the string value. It is appended to the standard table name when created. If the table going to be created is "SPATIAL_VECTOR_IDX" and if the namespace is "namespace", the resulting table name would be "namespace_SPATIAL_VECTOR_IDX". In this module, we extract the namespace from a given table. For that, it uses the above strategy to use to compose the name. Other function is to return the list of tables corresponding to the given namespace.

e. Geospatial Extent

This module is the most of the important module in the application. Because which is the one who do the most important task of the project. What is does is finding the geospatial extent or Geospatial representation of the Tablet and formatting again it to show in map. It there main functions. Those are, finding the table range, getting geospatial extent based on range, calculate convex hull of geospatial extents.

First it find the all the split points of a table. Using that it find out the table of the table. Next using the tablet it find the key range of these each tablets. Next use this ranges to find the geospatial extent of the tablet. Then iterate all the rows of this tablet to find the geospatial extent using the "AccumuloUtils.decodeRow()" function. Geospatial extent contain the list of coordinates representing the corresponding tablet on a map. Next using these points find the convex hull of the tablet. Convex hull is calculated using the "com.vividsolutions.jts.algorithm.ConvexHull;" class and associated functions.

f. Background Worker

function of this module is to start background daemon thread to find the spatial extent and convex hull of each of the tablets of the tables. Purpose of making them as background thread is above mention calculation are time consuming functions. Because of this it calculate the convex hull periodically and hold it to server when new request instantly.

First it find out the all the tables of the Accumulo and which support Geowave operations. TheAI then It find the all the tablet convex Hull and move to the List. At the end this module contains the map holding above calculated list of convex Hull. Map keys are corresponding to the table name and values are corresponding to the list of tablet convex hull of that corresponding table.

This calculation is start up when wep application loaded in to the container. This module implemented class is "ServletContextListener" class and "Thread" class. It's "contextInitialized()" start to execute when it loaded it to container. This is the place it start to execute calculations. Above function is overridden and have implemented the above mentioned calculation. Also it contain the function "contextDestroyed()" which fire up when application destro or unloaded. All the execution interruption function are call inside this. In "contextInitialized()" method has defined the scheduler to define the time period to perform convex hull calculation.

g. REST Service

Above mentioned function are exposed as rest service. Then this REST service or API can be used to any other extending features. Web application is also implemented using this REST API. All the API is implemented and server as HTTP GET methods.

Here are REST api information in brief.

- i. **http://<host>/webapp-context/stat/listns**
 - This GET method return all the namespaces geowave created
- ii. **http://<host>/webapp-context/stat/listtable/{ns}**
 - GET method, return all the table of the **ns** namespace
- iii. **http://<host>/webapp-context/stat/table**
 - GET method return statistic of all the Tables accumulo contain
- iv. **http://<host>/webapp-context/stat/tablet/{table}**
 - GET method, return all the tablet statistics information of the given **table**
- v. **http://<host>/webapp-context/stat/tablet/{table}/{tabletId}**
 - GET method, return specified table statistics. Table name and the tablet Id should be provided.
- vi. **http://<host>/webapp-context/stat/geo/{table}**
 - GET method return all the geo spatial extents of the given **table**

h. Web Application

Web application is simple web application written using HTML and javaScript. It use the D3js as library to implementing Graphs. It contain the World map which use to display the mapped tabled geospatial extent on this map.

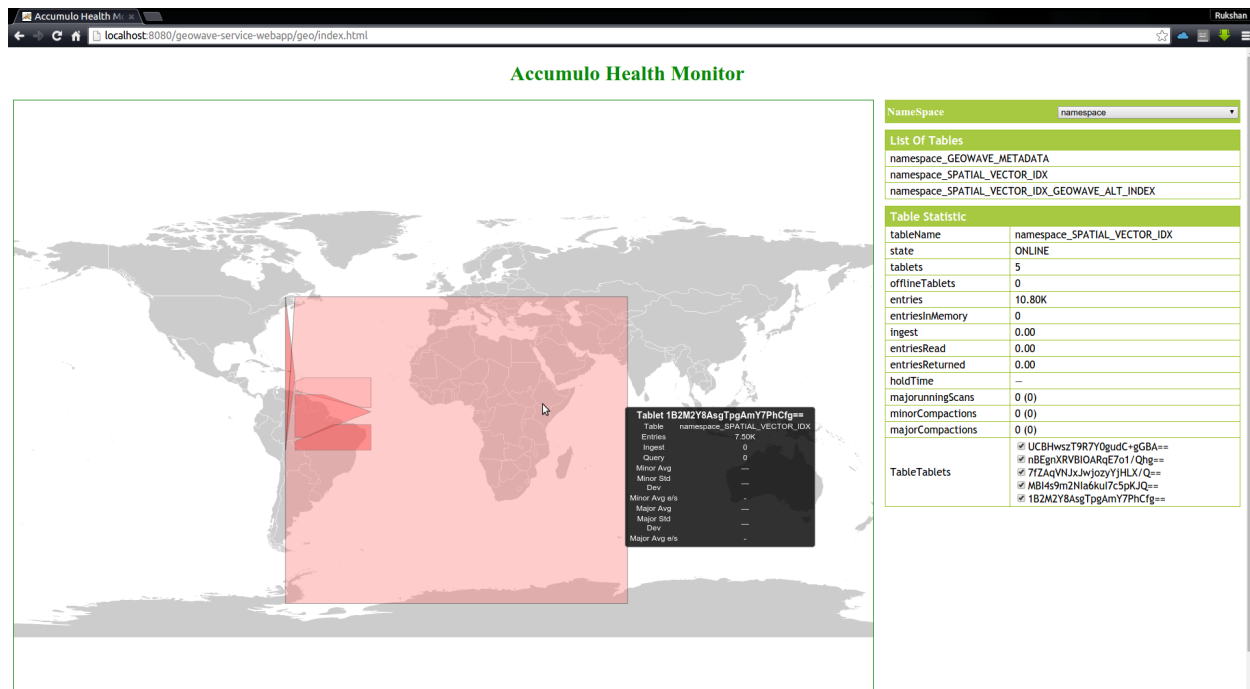
which is responsible to display Table and tablet statistic data. It is possible to see the Tablet information filtering according to the any namespaces. when click the one of the table name is shows the all the tablet extent on the map. it is possible to hide and show each of this table on the map. And when mouse hover on each tablet, it will popup with the tablet statistics.

Here all the interaction are done via the REST api. This web application is completely running at the browser. when it need any statistic or geospatial extent information, it make REST call to REST module to fetch them and display on the client web app.

i. Intergration

Above mentioned REST module and web application are work together to implement Geowave monitor. REST module which is java web-app implemented using jersey[3], is deploy inside the geowave web-app module. Also UI interface is put inside the webapp module. But it can be disassemble any time and host on anywhere else. Because it is not tightly couple with Geowave and it just use the REST api. After the build the geowave web-app module is it possible to deploy on the java web app container and run.

5. Result



Current view of the web application interface

6. Reference

- [1] <http://ngageoint.github.io/geowave/documentation.html>
- [2] https://accumulo.apache.org/1.7/accumulo_user_manual.html
- [3] <https://jersey.java.net/documentation/latest/getting-started.html>
-