Google Summer of Code

Proposal

Hazelcast Jet execution engine support for Apache Gora(GORA-546)

01. Organization

Apache Software Foundation

02. Introduction

The Apache Gora™ open source framework provides an in-memory data model and persistence for big data. Gora supports persisting to column stores, key value stores, document stores, distributed in-memory key/value stores, in-memory data grids, in-memory caches, distributed multi-model stores, and hybrid in-memory architectures. Gora also enables analysis of data with extensive Apache Hadoop MapReduce™ and Apache Spark™ support[1].

Gora is specifically designed to give an abstraction for different noSQL technologies. Gora has brought the ORM(Object Relational Mapping) concept to noSQL data stores. Still it provides support for SQL databases too. At the moment gora supports following data stores,

1. Apache Hbase
2. Apache Cassandra
3. Apache Solr
4. MongoDB
5. Apache Accumlo
6. Apache CouchDB
7. Amazon DynamoDB
8. Infinispan
9. OrientDB
10. Aerospike

Gora provides a generic API to work with above datastores. Data storing, data persisting and querying can be done via Gora APIs on above data stores.

Apart from data stores gora provides support for mapReduce, Apache Spark, Apache Pig and Apache Flink.

On the other hand, Hazelcast Jet is an application embeddable, distributed computing engine built on top of Hazelcast In-Memory Data Grid (IMDG). With Hazelcast IMDG providing storage functionality, Hazelcast Jet performs parallel execution to enable data-intensive applications to operate in near real-time[2].

03. Why I selected this project,

I'm an undergraduate of Faculty of Information Technology, University of Moratuwa, Sri Lanka. I'm really interested in participating GSOC 2019. Furthermore i'm keen on contributing to apache projects. So I thought of selecting Apache gora. I'm planning to direct my career towards data science. So that also led me to selecting Apache gora.

I have experience on stream processors like wso2 stream processor and siddhi query language. And I have also contributed to implement stream processing capabilities into wso2 Ballerina Language. These factors led me to selecting the "Hazelcast Jet execution engine support for Apache Gora" project.

04. Problem in Brief

Apache gora supports mapReduce, Apache Spark, Apache Pig and Apache Flink at the moment. Apache Spark is considered to be the successor of mapReduce. On the other hand, Hazelcast jet is an emerging distributed computing engine which competes shoulder to shoulder with Apache spark and others.

According to hazelcast jet team, jet is much faster than Apache Spark and Apache Flink[3].

Jet keeps both computation and storage in-memory, as much similar to spark.

"The key difference between Jet and competing technologies such as Spark and Flink is that it's built on top of something called a "one-record-per-time architecture," the company said[4].

This means Jet can process incoming data records as soon as possible, whereas Spark accumulate records into micro-batches before processing them. As a result, Jet simply works faster, thereby lowering latency for the applications it powers, the company claims.

Company also claims that jet is very simple to setup and program in comparison of competing technologies.

So the combination of Gora and jet could result in an brilliant product. Through this Gora users can use jet very easily. On the other hand, jet could utilize the gora's ORM for various different datastores directly.

05. Proposed Solution

Jet accesses data sources and sinks via its connectors. They are a computation job's point of contact with the outside world[5].

As per the discussions carried out and per my studies, found out that jet provides means of integrating via jet sink and source builders(custom connectors)[6].

Here I could come up with a possible POC[7]. What this does is read the AccessLog table created via existing LogManager example, and feed the PageView objects to jet.

So to provide jet execution support for gora, we have to implement source and sink connectors for hazelcast jet. Jet provides means of writing custom connectors[6]. Through these connectors hazelcast jet can get data into its pipeline.

Following is a simple class diagram of proposed solution.

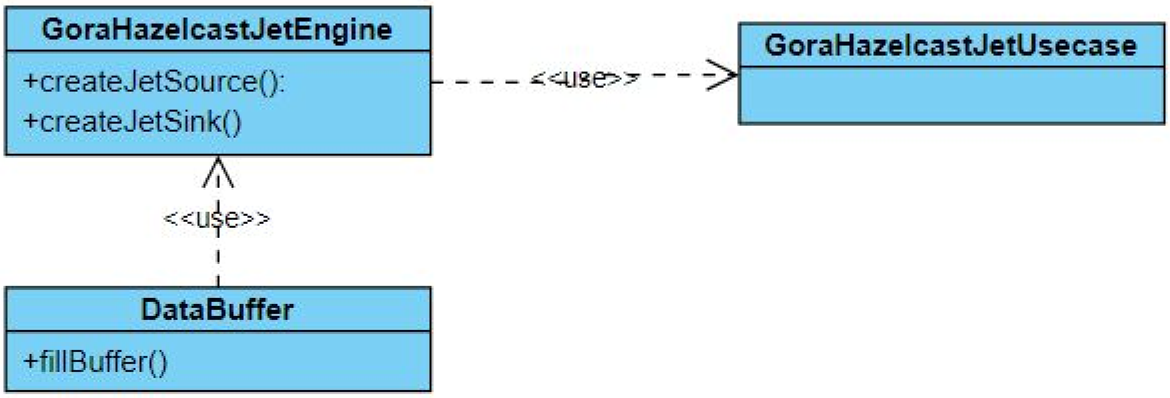

Diagram: High level class diagram of proposed solution

Here the DataBuffer class should be designed to read and write to different data stores. Jet repeatedly calls 'fillBuffer' method whenever it needs more data items.

Finally an end user should be able to read data from a gora datastore and do jet computations on it and write back to gora datastores.

06. Schedule & Timeline

Time Period	**Scheduled work**
Community Bonding **May 6 - May 27**	● Looking into the problem deeply. ● Communicating with mentor and Gora community to build relationships. ● Discussing further about implementation details.
Coding period **27th May – 23rd June**	● Creating the basic layout for the new module. ● Implementing jet source connector.
Evaluation phase 1 -> 24th June – 28th June	
Coding period **29th June – 21st July**	● Addressing comments from evaluation. ● Implementing sink connector.
Evaluation phase 2 -> 22nd July – 26th July	
Coding period **27th July – 18th August**	● Addressing comments from evaluation. ● Implementing Classic word count example via jet using gora datastores. ● Writing LogAnalytics tutorial in jet. ● Documentation.
Final Week **19th August – 26th August**	● Submit final work

07. Commitment

I’m very interested and very keen to work on this project. I plan to allocate most of my time to this project. So there will be definitely more than 30+ hours per week dedicated to this project as required by GSOC rules.

There is one exception, I have an examination period from 8th July to 28th july. But I’m confident that I will be able to allocate time to the project as well in this period.

I am interested in becoming a Apache committer in future. So I think this will be an added advantage to that. I'm also hoping to develop my community bonding via this project.

08. Open source contributions

I have worked in WSO2 Ballerina project. You can see the PRs here

https://github.com/ballerina-platform/ballerina-lang/pulls?q=is%3Apr+author%3ALahiruJayasekara+is%3Aclosed

I have contributed to FOSSASIA open event organizer app. You Can see the PRs here

https://github.com/fossasia/open-event-orga-app/pulls?q=is%3Apr+is%3Aclosed+author%3ALahiruJayasekara

Here is the above mentioned POC created for this project[7].

09. Contact info

Name - M.L.P.Jayasekara
Email - mlpjayasekera@gmail.com
Github - https://github.com/LahiruJayasekara

10. References

1. http://gora.apache.org/index.html
2. https://jet.hazelcast.org/introduction/
3. https://hazelcast.com/resources/jet-0-4-vs-spark-flink-batch-benchmark/
4. https://siliconangle.com/2017/02/07/hazelcast-launches-lightweight-distributed-data-processing-engine-rival-apache-spark/
5. https://docs.hazelcast.org/docs/jet/latest/manual/#source-sink-connectors
6. https://docs.hazelcast.org/docs/jet/latest/manual/#source-sink-builder
7. https://github.com/LahiruJayasekara/gora/blob/poc-hazelcast-jet/gora-tutorial/src/main/java/org/apache/gora/tutorial/log/HazelcastJetPOC.java