Memorandum to Management: Final Code Walk

AUTHORS: Lal Birali and Kevin Enright

I. INTRODUCTION

The project that our group has been working on for the past several weeks is the design of a distri buted KV system to store mountains of data, that are distributed over several "nodes". This under taking starts in Milestone One, where the goal is to read in data from an .SOR file, where data co mes in from a specific source format, parsed and then stored into a data frame. We also allow for simple command line arguments for basic querying. We have a Makefile, that will can build, run tests, and run valgrind on our project via the make build , make tests , and make val commands. In Milestone Two, we were able to split our DataFrame into n number of nodes that was specifie d by input into the command line. For Milestone Four we added code that will eventually support client/server architecture and were able to serialize our objects using the boost library. For the C W, we wish to demonstrate the Trivial Application which works on one node. We wish to presen t our work in terms of what we had hoped to achieve.

II. ARCHITECTURE/IMPLEMENTATION

As of now, what has been implemented, as shown in the src/ folder is a SorParser.h file that read s in a file, and returns a Data Frame. This Data Frame is then passed into our Test class that also resides in a main.cpp file. Basic tests are run to ensure data integrity and quality. After this, the D ata Frame currently supports this functionality, as provided by a previous assignment:

- Adding rows
- Adding columns
- Setting values
- Getting values
- Getting number of rows and columns
- Setting column names and rows
- Setting an index row
- Map operations that perform aggregations
- Filter operations that includes data based on some provided test

Our Data Frame is also now split into n number of nodes specified by command line argument a nd these new DataFrames are stored in our KV Store.

We added in code to support messaging between clients that allows our server to send serialized dataframes to nodes. (We will eventually set up the nodes to be the only things to contain those s ections of the dataframe and delete all other instances but did not have time to do so.

III. USE CASES

Work in progress:

If make val or make test is run from the command line, then should see all tests pass along with p rompts for user input. Make val is currently showing us losing 400millon bytes, and we aren't rea lly sure how.

IV. OPEN ISSUES:

- --Switching the KV Store to store serialized dataframes instead of actual dataframes
- --Refactor our code to allow for easier testing
- --
  Switch our command line querying to use the nodes to get information instead of the orig inal dataframe.
- --Deserialize dataframes on the client side and allow for querying of the nodes.
- Implement networking

V. STATUS

The parsing of the file and storage of current data is occurring accurately. We are able to split sm aller datasets into KV stores. Our next phase to kill bugs and fix the server/client logic to support querying over that and then switch our KV store to store serialized dataframes instead of actual dataframes. We also thought about using threading, but due a lack of time since M5, we were un able to make much tangible process, so we reverted back to just making Trivial work.

(Apologies for the incompleteness of this assignment, one of us had a death in the family and we have been trying to get as much done as possible given the horrible circumstances)