Final Project Proposal
Parallel and Real Time Computation of Statistics for Thousands of Data Streams

Background

I am working with a company specializing Big Data Analytics Development. An interesting application from our manufacturing customers is how to analyze the data collected from sensors in a real time manner. Consider the problem of monitoring thousands of time series events measured by sensors in a second, can we do some calculations among the data such as average of amount among a couple sensors, standard deviation, the correlations among them in real time?

Objectives

Establish a Hadoop and MapReduce computation platform to do parallel computations in applications.

Find out the parallelism of some statistical functions and model that can be computed quickly in the proposed MapReduce computation platform.

Leverage high performance kv store to persist historical data and improve the computation with them.

Computation Model

I am going to use "Slide Windows" model that we can define the interested slide window, with base windows that are shorter windows partitioned equally within slide window.

Given the length of the sliding window w and the current timepoint t, stat(s, slide(w)) will be computed in the sub-sequence s[t-w+1... t].

Suppose w = kb (equal partitions) where b is the length of a base window and k is the number of base windows within a slide window. Use S[0], S[1], .. to denote the sequence of base windows, then S[i] = s[(t-w)+ib+1...(t-w)+(i+1)b]

Let me use Sum of data within a slide window as example:
in a time series of t1, t2, t3, t4, t5, …

s[t1...t4] = S[t1]+S[t2]+S[t3]+S[t4]
s[t2..t5] = s[t1...t4] – S[t1] + S[t5]

Assume the duration between two continuous time-points are the same, S[t] is highly parallel if we calculate S[t] in MapReduce platform and store the former s[t1...t4], the value of s[t2...t5] is to expire the first one and add in the next. We can see the performance that when slide windows are from tens of base windows, we can get the next s[] value from the last s[] and one addition, one subtraction.

When we consider more complicated calculations such as the distance or correlations among multiple streams, we may come up similar formula as above. Then we can leverage the project to efficiently calculate values in real time.

Computation Platform

The computation platform will be a Hadoop/MapReduce ecosystem that includes pig, spark, hadoop,hive etc. I would like to build it up from scratch such that I can learn more about the details of them. I may go further to establish YARN and MapReduce2 depending upon the timing. My plan is to leverage docker system, seeking the available modules to build up the platform. The platform will be ubuntu linux.

Data Source

The data will be used are stock prices primarily from finance.yahoo.com, although I may consider to generate or simulate some automatically for demonstrating specific features purposes.

Charting

The charts to demonstrate the outcome results of the calculations will be from open sources in html and javascript. An open source of charting http://www.highcharts.com/ will be considered although I would rather leave it open in terms of flexibility.

Deliverables

All implementation will be in java. I will deliver:

- The documentation to record the details how I build up the docker system of MapReduce Platform.
- Sum, Average and Standard Deviation of data streams in slide windows will certainly be completed and demonstrated. The performance analysis may be included.
- Correlation or the distance analysis of data streams are a **"maybe"** depending upon the time.
- Fourier Transform has been a wide research topics in the MapReduce Computation ecosystem. This is a **"maybe"** implementation too. The idea is to transfer the time series data from time domain to the frequency domain and do parallel calculations against the coefficients in the frequency domain which is fast and efficient. Then convert them back to time domain that is expected. This approach can be achieved to get a very useful approximation of the results in many applications. I will discuss the model and how to apply the proposed computation platform to do the calculations for such applications in the project presentation.

Team!

I live in California, so may be hard to find a partner to work together, I assume it will be a one person project for now.

Reference

High Performance Discovery in Time Series, Dennis Shasha/Yunyue Zhu
http://www.amazon.com/High-Performance-Discovery-Time-Techniques/dp/0387008578/