Efficient approaches for merging local-histograms to global-histogram

Zhe Zhang, Xusheng Xiao, Ye Jin
Possible thesis advisor: Steffen Heber

September 20, 2010

1 Introduction

Recently years, in many nature scientific research areas, such as global climate research, it becomes not only more globalized but also more commonly that original data gathered are in tera-scale or peta-scale. Original local data's measurements, like range or precision, vary within different areas or countries.

Histogram is a graphical representation in statistic that is also used in Visualization of Data Mining. It is easy to generate local histogram for a single chunk data with consistent measurements. However, scientists who focus on global area research, like those climate ones studying global warming, will need global statistic results, for example: global histogram. It will be unrealistically time and resource-consuming to go through whole bunch of data again only for those global statistic results.

Some smart scientists has already published their concept that pre-process the data while they are generated in memory and have not been stored into file-system, and hence there will be less need to read-out those peta-scale source data and do the analysis.[2] Based on their thought, we plan to implement and test the correctness ad performance of our methods that generate global-histogram from local-histograms via both serial programming and Parallel programming using MPI. Also we will compare our methods' performance with others[1].

2 Application

The application will be related with data statistics in global climate. Specifically with in Generate Global temperature, humidity distribution-histogram by merging local-grid ones using MPI Parallel Programming.

3 Intuition thought, Method and its benefits

The Intuition thought is that using Parallel Programming to generate the peta-scale global histogram's generation from local histograms (results) but not from original data to reduce or eliminate the I/O burden of re-read and write the data file-systems.
There are three step:

1. Split souce data into multi-part evenly and Sort data

   This step is to simulate the multiple local data sources, and prepare(sort) the data for later manipulations of generating local histogram via different sort algorithm .

2. Generate local-histogram based on the single part well sorted data

   This step is using MPI to communicate among processes first to pass global information package. Then based on the global information package that all processes received, each process calculate local histogram via self-defined hist-function() in C program language, which produce the same output as using R.

3. Merge those local-histograms into one global-histogram

   This step use MPI to pass the local result of local-histograms among all processes, and merge them to global-histogram.

4 Data sources

1. **GIS free online database**: $http://data.geocomm.com/$

   Reasons: In this database, the maps are divided into hierarchical levels, from states to country to global. This kind of raw data confront the problem of how to generate global histogram without going through the whole huge chunk data again?

2. **Microarray Data**: $http://www.cse.buffalo.edu/faculty/azhang/Teaching/Project2.rar$

   Reasons: In the past few years, microarray technology has become one of the foremost tools in biological research. The emergence of this technology has empowered researchers in functional genomics to monitor gene expression profiles of thousands of genes (perhaps even an entire genome) at a time. However, mining microarray data also presents great challenges to Bioinformatics research. This data source will provide us a chance to analyze microarray data using data mining techniques, such as clustering and classification.

References

[1] Chad Jones, Kwan liu Ma, Allen S, and Lee Roy. Visual interrogation of gyrokinetic particle simulations. 2007.

[2] Fang Zheng, Hasan Abbasi, Ciprian Docan, Jay Lofstead, Scott Klasky, Qing Liu, Manish Parashar, Norbert Podhorszki, Karsten Schwan, and Matthew Wolf. Predata-preparatory data analytics on peta-scale machines.