Weekly Report – 51

Xiufeng Liu

University of Waterloo, CA
xiufeng.liu@uwaterloo.ca

1 This Week

At the beginning of this week, I first investigated the SciDB on how it can fit our benchmark requirements. I designed a three-dimensional data model for ESSEX power dataset, import the data set, and learn how to do the basic data analysis in SciDB. In the later of this week, I focus on the study of the use of KDB+, a column-oriented in-memory database.

2 Next Week

Learn the q programming language, and try to write the program for the analysis data model.

3 KDB+

kdb+ is the single-platform, high-volume, and high-performance database which supports billions of real-time records analysis [7]. kdb+ has an enterprise 64-bit version, and an 32-bit trial version. However, a full-64bit version can be made available for academic users. Kdb+ is an in-memory column-oriented database based on the concept of ordered lists, which makes it extremely fast to do data analytics. Kdb+ is able to handle millions of records per second, billions per day accumulated in in-memory databases, and billions per day records stored in databases on disk. It is now widely used by the companies in financial services industry, including Goldman Sachs, Morgan Stanley, Deutsche Bank, etc. [8], to capture real-time ticks and do analysis.

Unlike conventional row-based DBMS, kdb+ is column-oriented database which data is stored in vectors of ordered lists, instead of the set as in RDBMS. Therefore, multiple identical values can co-exists in a list. Since kdb+ is an in-memory database, it requires a lot of RAM if keeps all the column data of a massive table in memory. Therefore, a big table can be splayed (partitioned), and only the necessary column data is read and kept in memory. This minimizes the memory foot print and improves the performance greatly during data analysis. If the size of one-column data still cannot fit into memory, the data can be further partitioned horizontally, for example horizontally partition based on the date of time series data. The data file for each partition is stored under a separate folder in file system.

Q language servers as the query language for kdb+, which was developed by Arthur Whitney. Q is a vector-based functional language built for querying and transforming data through anonymous functions, *lambdas*. Since q is interpreted, users can enter commands straight into the console, do not have to wait for compilation, and the results can be returned instantaneously. Q is a strong typed language,

supporting a rich SQL dialet, and the data types: scalar including date, datetime, minute, second, and time; lists; dictionaries and tables. It also supports IPC for both of asynchrononous and synchronous.

References

1. Ten Million Meters Scalable to One Hundred Million Meters for Five Billion Daily Meter Readings. Sept. 2011.
2. J. Yang, Y. Zhai, D. Xu, et al. "SMO Algorithm applied in time series model building and forecast". In *Proc. of ICMLC*, 2007:2395-2400, 2012.
3. P. G. Brown. "Overview of SciDB: Large Scale Array Storage, Processing and Analysis". In Proc. of SIGMOD, pp. 963–968, 2010.
4. SciDB User's Guide. `http://scidb.org/HTMLmanual/13.3/scidb_ug/index.html`
5. SciDB `http://www.scidb.org/` as of 2013-12-07.
6. SciDBR. `https://github.com/Paradigm4/SciDBR` as of 2013-12-07.
7. KDB+. `www.kx.com` as of 2013-12-07.
8. KDB+ Customers. `http://kx.com/end-user-customers.php`