A Debate on MapReduce

Shuxiao Liu

Illinois Institute of Technology

A Debate on MapReduce

A paper which expressed criticisms on MapReduce was released by David J. DeWitt and Michael Stonebraker on January 17, 2008. In the paper, they compared MapReduce with DBMS, and drew a conclusion that MapReduce is a major step backwards.

Then, on January 22, 2008, Mark C. Chu-Carroll released a paper against former one. He showed his praise on both DBMS and MapReduce, and explained what is the main function of MapReduce is different from DBMS and the role MapReduce plays.

I will first summarize and discuss their opinions of these two paper. Then give my own opinion on this topic.

Paper One

In the first paper, the author who is a professor of computer science thinks MapReduce is a major step backwards. First, he put forward 5 critical points on MapReduce:

1. A giant step backward in the programming paradigm for large-scale data intensive applications
2. A sub-optimal implementation, in that it uses brute force instead of indexing
3. Not novel at all -- it represents a specific implementation of well known techniques developed nearly 25 years ago
4. Missing most of the features that are routinely included in current DBMS
5. Incompatible with all of the tools DBMS users have come to depend on

To explain his views above in detail, he introduces how MapReduce works. It consists of two programs – Map and Reduce. Map reads records, filters and/or transfer them, and then outputs a set of records of the form (key, data). A “split” function split these records into M disjoint buckets and output them. If there are N nodes in total, there are N*M output files.

Reduce program processes the records and writes them to an output file which is a part of the final answer. In author’s opinion, Map is similar to group-by, and also Reduce to aggregate function in SQL.

After giving the definition of MapReduce, the author explains his 5 critical points successively. MapReduce does not take advantage of any lessons from 40 years’ history of DBMS community, like it ignores the importance of schema, which can not keep garbage out of data sets. MapReduce does not provide indexes, which will cause access difficulties. MapReduce just make use of outdated techniques. MapReduce does not provide features which are necessary in DBMS. MapReduce has no tools provided by modern DBMSs or its own tools.

At the end, the author explains that MapReduce has its own value that it can be used as the basis for building scalable database systems.

Paper Two

In the Second paper, the author who is a Google employee holds an opposite view. He believes that MapReduce is an unreplaceable programming model with particular uses, while he admits DBMS is powerful.

In the beginning, Mark introduces MapReduce (almost same as paper one) and its advantages. MapReduce can use a bunch of cheap machines to resolve problems which are supposed to be resolved only by super machine.

Then, the author uses a metaphor to explain his understanding of both MapReduce and DBMS. He compares DBMS to hammer, which is definitely powerful, but in particular environment, it’s not as useful as a screwdriver—MapReduce.

According to the 5 points of David J. DeWitt, the author successively puts forward his objections. M/R is a more proper way to program large-scale data applications. Indexing is not that helpful when data isn't fundamentally relational. MapReduce is not supposed to be novel, because it is a well understood form of data parallel computing. MapReduce and DBMS are applicable to different problems, so MapReduce doesn't need DBMS's features or tools.

At the end, the author claims again that DBMS is powerful, but it is not he only tool in the world. It has its weak point which is the strong point of MapReduce.

Comparison

David J. DeWitt is known for his research on parallel databases, benchmarking, object-oriented databases, and XML databases, having great contributions to the database field. He compares MapReduce to DBMS in the whole paper and draw a conclusion that MapReduce is far worse than DBMS and should learn from DBMS.

Mark C. Chu-Carroll has Specialties on software configuration management, distributed systems, and worked for IBM, Google, Twitter and now Dropbox. He admits that RDBs are amazing tools, but are not designed for large-scale data. MapReduce can resolve that kind of problem.

My Opinion

Considering that David J. DeWitt has worked several decades on area of RDB, it is easy to understand that he can not accepted new things — MapReduce. I think he might choose a wrong object to compare. MapReduce is not design to replace RDB. They are applicable to different problems.

As a ITM student, I have both learned DBMS and MapReduce(this semester). I can clearly feel out that they are both great tools to solve different problems. For example, in week 3 assignment, I can get our result in a few minutes by using MapReduce.

I agree with Mark’s views. MapReduce is not a database model, but a large data computing technique. It is ridiculous to compare MapReduce to DMBS.

References

[1] " MapReduce: A major step backwards," By David DeWitt on January 17, 2008.

[2] " Databases are hammers; MapReduce is a screwdriver.," Mark C. Chu-Carroll on January 22, 2008