Building a High-Level Dataflow System on top of Map-Reduce: The Pig Experience

Alan F. Gates, Olga Natkovich, Shubham Chopra, Pradeep Kamath, Shravan M. Narayanamurthy, Christopher Olston, Benjamin Reed, Santhosh Srinivasan, Utkarsh Srivastava Yahoo!, Inc.
Analysis by John Kryspin
5/8/2015

Main Idea of Pig

- Easy to use high-level dataflow system
- Open Source
- System allows for less human errors because more functions are available from get go
- Uses both SQL-stylized code and Map-Reduced execution techniques

How Is Pig Implemented

- Used on top of Map-Reduce
- Offers high-level data manipulation constructs
- Programmer is in control
- Pig’s constructs can be woven in with user-provided executables
- Large level of customizability

Analysis of Pig

- Allows for larger scalability
- Easy to use
- Good customization options
- Easy debugging
- Is generally a better system to use than just Map-Reduction methods and is easier on the database programmer
- Proven to work well (Yahoo)

Main idea of Comparison Paper

- Parallel SQL DBMS are much faster than MapReduce once a Parallel SQL DBMS is tuned
- MapReduce had two functions so it is very simple and easy to understand
- Functions like joins are not readily available in MapReduce and there is no indexing
- MapReduce is a brute force system that wastes energy

How is MapReduce Implemented

- Has two functions *map* and *reduce*
- Map takes in data and parses through it and creates an output for reduce to take in
- Reduce takes in each key value pair and creates an output that summarizes all the data depending on what you wanted to do

Analysis of MapReduce

- Not as fast as Parallel DBMS
- Missing simple functions
- No indexing
- Easy to understand
- Many ways/options on how to use it like Hadoop
- SQL is more powerful once understood

Comparison

- Pig creates an environment using MapReduce that takes in SQL like code which makes it more scalable and easier to perform certain operations on
- The underlying system of Pig is still MapReduce so it is slower than other options
- Pig offers a better programming experience

Stonebraker Talk Main Ideas

- Every system will eventually have its own database system that works best for it
- No one database system will work best on everything
- More and more options will be released and the best will rise in usage
- What we thought in the 80’s was all incorrect about how there should be one sole database system that works best of everything

Conclusion

- Pig is good for scalability/ease of use but is slower than a Parallel DBMS because it uses MapReduce
- Pig may rise in popularity due to its specific applications it is *not* “one size fits all”
- Pig allows for custom user executions embedded within MapReduce techniques which creates versatility