The Pig Experience

By Alan F. Gates, Olga Natkovich, Shubham Chopra, Pradeep Kamath, Shravan M. Narayanamurthy, Christopher Olston, Benjamin Reed, Santhosh Srinivasan, Utkarsh Srivastava

Analysis and Etc. by Evan Mastriano

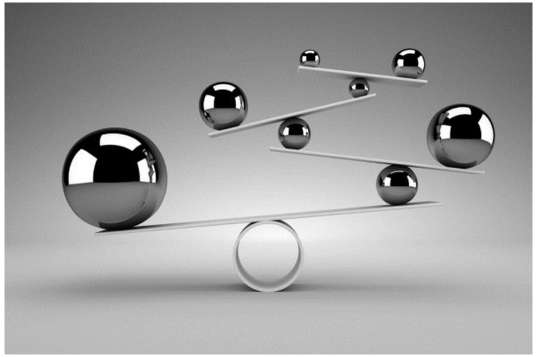
Pig - a middle ground for data manipulation

SQL is too advanced

Map-Reduce is too simple

Offers SQL-style manipulation constructs mixed with Map-reduce-style functions

Not purely declarative, so easily used and tweaked



Pig Implementation

Use of *Pig Latin* to turn dataflow programs into sets into an execution

No optimized storage structures (indexes or column groups)

3 modes of user interaction: Interactive, Batch, and Embedded

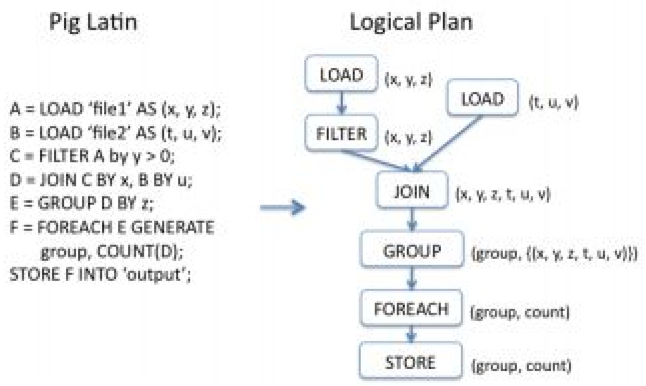
Features a nested data model, supporting standard and complex/non-normalized data

Translates query plan into a Map-Reduce execution plan

A pull iterator model is used as the execution proceeds

Benchmark was produced to improve and optimize performance, called *Pig Mix*

Used by Yahoo in roughly 60% of ad-hoc Hadoop jobs


Pig Latin

```
A = LOAD 'file1' AS (x, y, z);
B = LOAD 'file2' AS (t, u, v);
C = FILTER A by y > 0;
D = JOIN C BY x, B BY u;
E = GROUP D BY z;
F = FOREACH E GENERATE
      group, COUNT(D);
STORE F INTO 'output';
```

Pig Analysis

Effective for it’s goals

Provides an easier method of data manipulation without sacrificing too many previous constructs

Essentially added features to Max-Reduce to look and act slightly more advanced while providing the same results in the end

Plain SQL seems like the most efficient method, but this seems viable if SQL isn’t an option

The Comparison Paper

Map-Redux versus SQL Database Management Systems

Major differences:

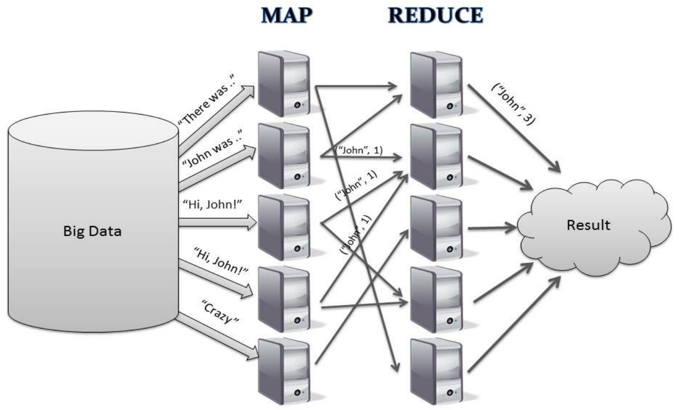
- Data Conformity
- Indexing and Compression Optimizations
- Programming Models
- Data Distribution
- Query Execution Strategies

It's Implementation

Map-Reduce

Two Functions - Map and Reduce

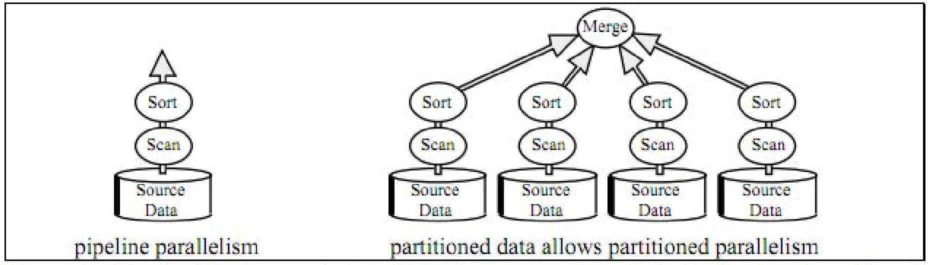
Map reads through and produces pairs that reduce orders and sorts



Parallel DBMSs

Tables partitioned over the nodes in a cluster

System optimizer translates SQL commands to output certain tables



Comparison Analysis

It seems to be up to personal preference

Both methods allow the same things to be accomplished (but in different ways)

While the SQL that enables Parallel DBMSs is very direct, it's also an acquired knowledge

The Map-Reduce method is far simpler, but requires more work on the processing end as it is far less efficient

Overall Comparison of Ideas

Pig sits in the middle of these two methods

It adds on top of Map-Reduce to provide advanced manipulation tools

Still too unorganized and weak to rival either extreme, but a promising niche manipulation system

Stonebraker's Ideas

Relational Database Management Systems are *not* the answer

- Too many variations, such as streaming or column-store

"One size fits none"

- If you don't specialize in something, you're good for nothing

Streaming can be added to an OLTP engine to create an easy engine

Graphing analytics split between column store, array engine, or special graph engines

Engines directed at specific applications create the best market (and row stores don't do that)

Pig V. The World

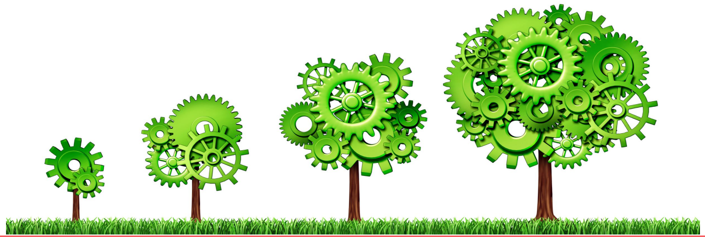
Pig seems to need to find a real niche in the market to succeed

It strengthens the Map-Reduce system by moving it closer to SQL DBMSs, but those aren’t the answer

Doesn’t seem like a big enough jump to give it a real “specialization” for a space in the market place

The most promising aspect of Pig is it’s continued support and growth

It can grow to adapt to new market areas and can develop its niche eventually



Works Cited

A Comparison of Approaches to Large-Scale Data Analysis by 7 people

Building a High-Level Dataflow System on top of Map-Reduce: The Pig Experience by 9 people

Stonebraker's ICDE 2015 Talk by Michael Stonebraker