A Comparison of Approaches to Large-Scale Data Analysis

Pavlo, Andrew, Eric, Paulson, Alex, Rasin, Daniel, Abadi, DAvid, DeWitt, Samuel, Madden, Michael Stonebraker. A Comparison of Approaches to Large-Scale Data Analysis Web. 19 October 2016

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 SrivastavaBuilding a High-Level Dataflow System on top of Map-Reduce: The Pig Experience Web. 19 October 2016

Michael Stonebraker's ICDE Talk About his "10 Year Test of Time

Scott Heinrich

Database Management

10/ 20/16

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

The PIG system was created to be a middle ground, or the best of both worlds.

PIG offers high level composable data manipulation while at the same time encoding explicit dataflow graphs like in Map Reduce.

Pig is implemented in Java, and as such has Java's memory management issues

Java does not allow the developer to control memory allocation and deallocation directly.

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

User-defined functions (UDFs) is one way to include user code. But UDFs must be written in Java and must conform to Pig's UDF interface.

Another way to implement code is through streaming using Pig Latin Syntax.

Pig's memory manager maintains a list of all Pig bags created in the same JVM, using a linked list of Java WeakReferences.

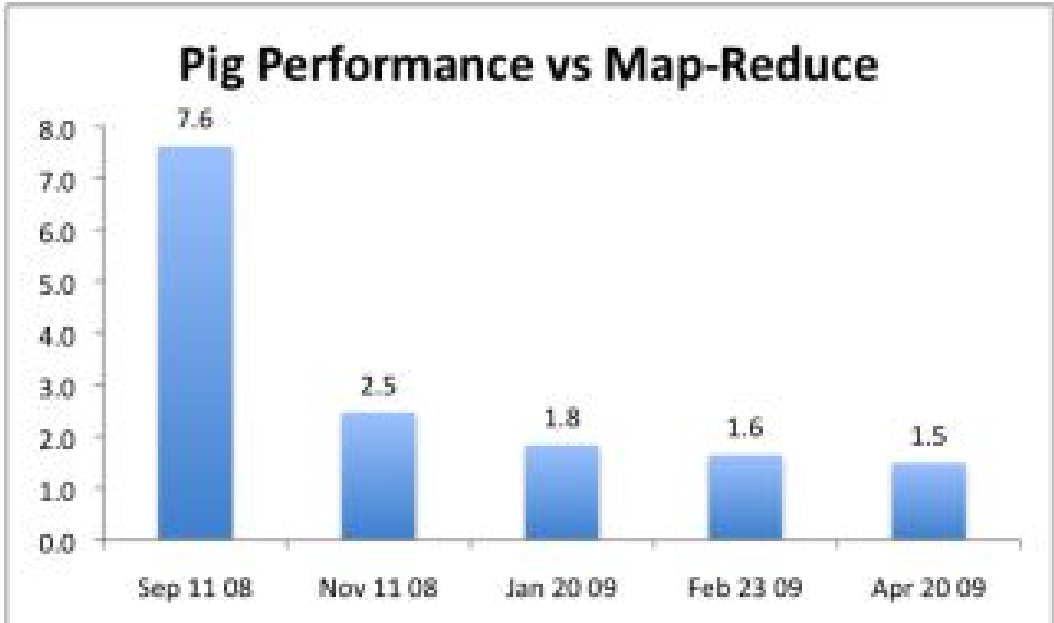
WeakReferences allows the garbage collector to discover bags no longer in use, while still getting rid of already "dead" bags.

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

PIG was created to be simple like Map-reduce, but at the same time include some user interactivity. While Map-reduce is simple, it has plenty of cons such as:

1) Data manipulation primitives Needed to be coded by hand
2) Lack of support for combined processing of multiple sets
3) Lack of support for N-step dataflows

1 on the vertical chart is the performance of Map Reduce

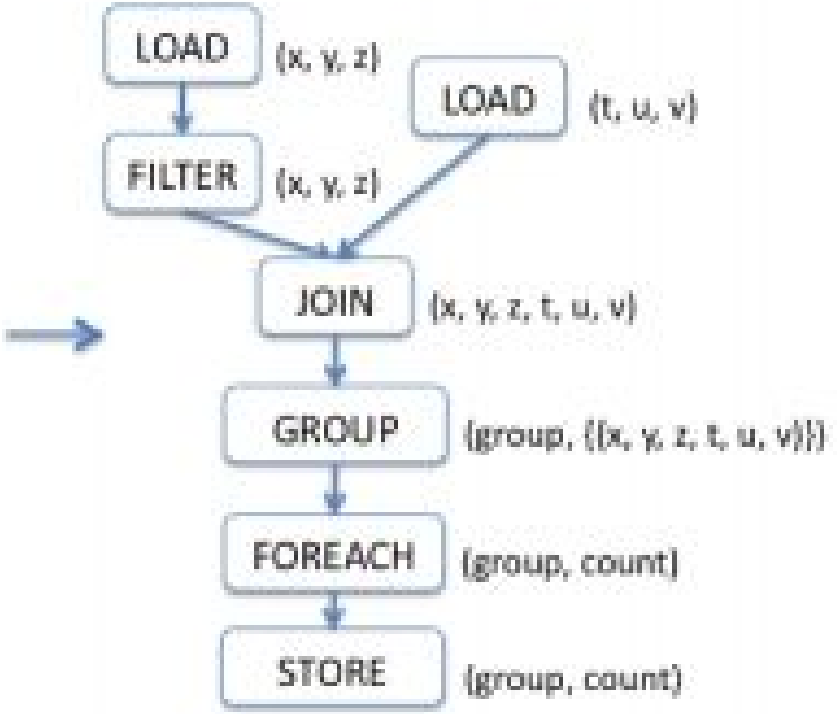


PIG Translation

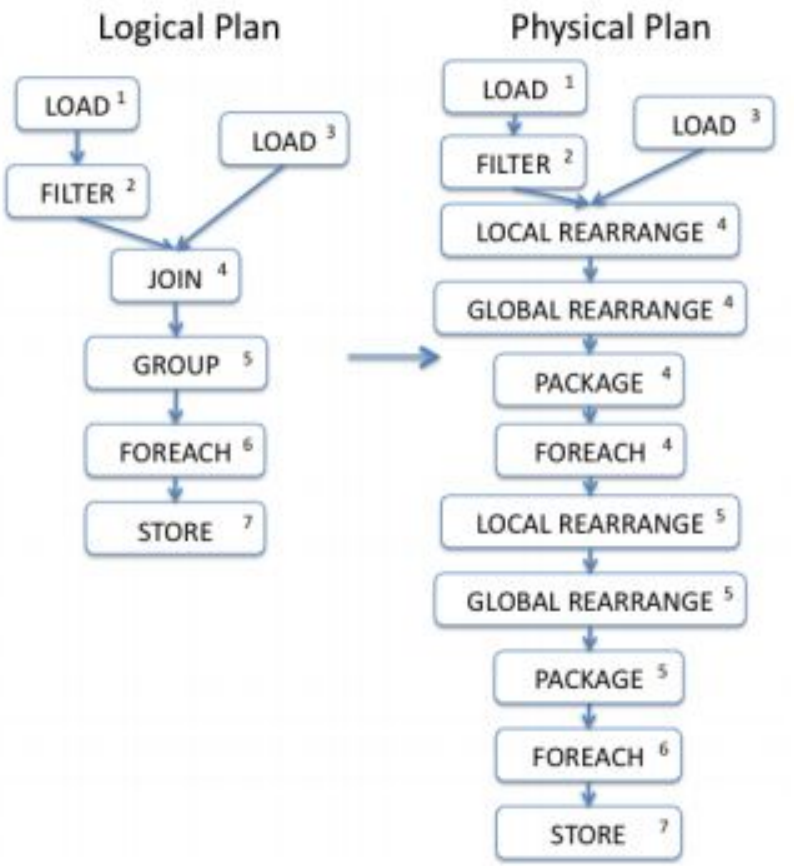
Pig Latin

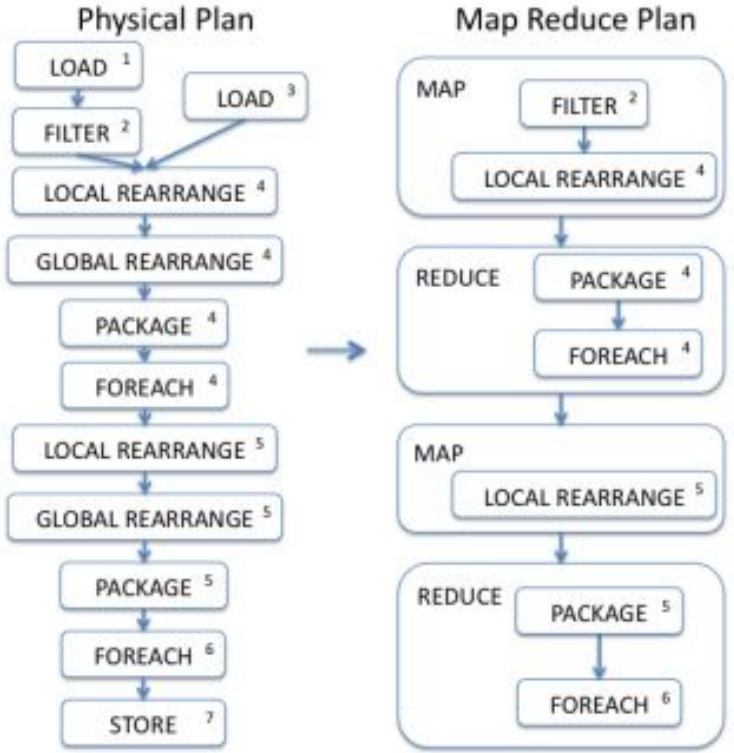
```
A = LOAD 'file1' AS (x, y, z);
B = LOAD 'file2' AS (t, u, v);
C = FILTER A by y > 0;
D = JOIN C BY x, B BY u;
E = GROUP D BY z;
F = FOREACH E GENERATE
    group, COUNT(D);
STORE F INTO 'output';
```

Logical Plan



PIG Translation Cont'd





A Comparison of Approaches to Large-Scale Data Analysis: Summary

This paper is compares Map-Reduce(MR) to Parallel Database Management Systems (DBMS).

The goal is to see how MR performs alongside two different types of (DBMS)
MR is compared with Vertica and DBMS-X.

DBMS is an alternative to MR, and has a few benefits over it.
DBMS has standard relation table and SQL support.
DBMS has a higher upkeep in manpower, and cost.

MR is considered more user friendly
MR-Hadoop is used by Yahoo
HIVE on top of Hadoop is used by Facebook

A Comparison of Approaches to Large-Scale Data Analysis: Details

MR is considered a “new way of thinking” when it comes to programming large distributed systems.

MR doesn’t require transformation or loading

MR doesn’t require schemas.

DBMS-X
Hash partitioned, sorted and indexed beneficially
Compression enabled
Difficult to work with

Vertica
Default parameters, except hints that only one query runs at a time
No secondary indices, tables sorted beneficially

A Comparison of Approaches to Large-Scale Data Analysis: Analysis

When it came down to testing, MR-Hadoop gets out performed by both Vertica and DBMS-X. On average, DBMS-X was at least 3 times faster than MR, and Vertica was 2 times faster respectively.

Although performance-wise, MR-Hadoop failed miserably, it was by far the easier to setup, and use during the testing. MR also minimizes work lost when hardware failure occurs. Higher Level Interfaces such as PIG and HIVE can improve upon MR's faults.

DBMS executed queries by having the nodes scan local tables and extract the necessary files. Executed queries are then merged, which uses less resources than MR

Comparing the Two Papers

A Comparison of Approaches to Large-Scale Data Analysis

- Compares Similar Systems for Business and Big Data Analytics
- Talked about advantages and disadvantages of each system through a number of tests.
- Discussed the results of the tests, and why the architecture of each system effected their result.

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

- Compares PIG to Map-Reduce Hadoop
- Goes over how PigLatin Text gets converted to Map-Reduce.
- Discussed the issues primarily for PIG, and how over time it has evolved and changed for the better.

Stonebraker Summary

Original goal was to have a relational database that could be used to solve all types of questions. “One size fits ALL”

No place for streaming applications within traditional row stores of RDBMS Making Row stores obsolete. “All for NONE”

As these markets expand and evolve, it opens up more opportunity for new ideas

NVRAM, Big Main Memory, Processor Diversity (Nvidia GPU’s, Numa), Higher Network Speeds, are advancing at staggering rate which allows for new architecture to emerge. This new architecture will take advantage of these new technologies. Some of these new technologies are predicted within the next 5-10 years (INTEL).

Advantages & Disadvantages of PIG

Advantages

- Allows Users to input their own code through UDFs or Pig Latin Syntax
- Open Sourced. Has had various update already
- Supports N-Step Dataflows unlike MR.

Disadvantages

- Falls short when compared to MR
- Memory Management issues due to Java implication