Developing Pig on Tez

Cheolsoo Park
VP, Apache Pig
Netflix

Mark Wagner
Committer, Apache Pig
LinkedIn
What is Pig

- Apache project since 2008
- Higher level language for Hadoop that provides a dataflow language with a MapReduce based execution engine

```plaintext
A = LOAD 'input.txt';
B = FOREACH A GENERATE flatten(TOKENIZE((chararray)$0)) AS word;
C = GROUP B BY word;
D = FOREACH C GENERATE group, COUNT(B);
STORE D INTO './output.txt';
```
Pig Concepts

- LOAD
- STORE
- FOREACH ___ GENERATE ___
- FILTER ___ BY ___
Pig Concepts

GROUP ___ BY ___

- 'Blocking' operator
- Translates to a MapReduce shuffle
Pig Concepts

Joins:
• Hash Join
• Replicated Join
• Skewed Join
Pig Latin

A = LOAD 'input.txt';
B = FOREACH A GENERATE
   flatten(TOKENIZE((chararray)$0))
   AS word;
C = GROUP B BY word;
D = FOREACH C GENERATE group, COUNT(B);
STORE D INTO './output.txt';
Physical Plan

LOAD
FOREACH
GROUP BY
FOREACH
STORE

LOAD
FOREACH
LOCAL REARRANGE
GLOBAL REARRANGE
PACKAGE
FOREACH
STORE
Map Reduce Plan

Map
- LOAD
- FOREACH
- LOCAL REARRANGE
- PACKAGE
- FOREACH
- STORE

Reduce
- STORE
What's the problem

- Extra intermediate output
- Artificial synchronization barriers
- Inefficient use of resources
- Multiquery Optimizer
  - Alleviates some problems
  - Has its own
Apache Tez

- Incubating project
- Express data processing as a directed acyclic graph
- Runs on YARN
- Aims for lower latency and higher throughput than Map Reduce
Tez Concepts

- Job expressed as directed acyclic graph (DAG)
- Processing done at vertices
- Data flows along edges
Benefits & Optimizations

- Fewer synchronization barriers
- Container Reuse
- Object caches at the vertices
- Dynamic parallelism estimation
- Custom data transfer between processors
What we've done for Pig

- New execution engine based on Tez
- Physical Plan translated to Tez Plan instead of Map Reduce Plan
- Same Physical Plan and operators
- Custom processors run the execution plan on Tez
Along the way

- New pluggable execution backend
- Made operator set more generic
- Motivated Tez improvements
Group By

```
f = LOAD 'foo'
    AS (x:int, y:int);
g = GROUP f BY x;
h = FOREACH g GENERATE
    group AS r,
    SUM(f.y) as s;
i = GROUP h BY s;
```
Join

\[ l = \text{LOAD 'left' AS} \ (x, y); \]
\[ r = \text{LOAD 'right' AS} \ (x, z); \]
\[ j = \text{JOIN} \ l \ \text{BY} \ x, \ r \ \text{BY} \ x; \]
Group By

\[
f = \text{LOAD 'foo'} \quad \text{AS} \quad (x:\text{int}, y:\text{int});
g = \text{GROUP f BY x;}
h = \text{GROUP f BY y;}
i = \text{JOIN g BY group, h BY group;}
\]
f = LOAD 'foo' AS (x, y);
o = ORDER f BY x;
Performance Comparison

Map Reduce

Tez
How it started

Shared interests across organizations

• Similar data platform architecture.
  • Pig for ETL jobs
  • Hive for ad-hoc queries
How it started

Shared interests across organizations

• Hortonworks wants Tez to succeed.
Community meet-ups helped

- Twitter presented summer intern’s POC work at Tez meet-up.
- Pig devs exchanged interests.
Organizing team

Community meet-ups helped

- Tez team hosted tutorial sessions for Pig devs.
- Pig team got together to brainstorm implementation design.
Building trust

Companies showed commitment to the project

- Hortonworks: Daniel Dai
- LinkedIn: Alex Bain, Mark Wagner
- Netflix: Cheolsoo Park
- Yahoo: Olga Natkovich, Rohini Palaniswamy
Setting goals

Make Pig 2x faster within 6 months

- Hive-on-Tez showed 2x performance gain.
- Rewriting the Pig backend within 6 months seemed reasonable.
Acting as team

Sprint
- Monthly planning meetings
- Twice-a-week stand-up conference calls

Issues / discussions
- PIG-3446 umbrella jira for Pig on Tez
- Whiteboard discussions at meetings
Knowledge transfer

- Pig old timer Daniel Dai acted as mentor.
- Everyone got to work on core functionalities.
- Everyone became an expert on the Pig backend.
Sharing credit

• Elected as a new committer and PMC chair.

• Gave talks at Hadoop User Group and Pig User Group meet-ups.

• Speaking at ApacheCon and upcoming Hadoop Summit.
Further collaborations

Looking for more collaborations

- Parquet Hive SerDe improvements.
- Sharing experiences with SQL-on-Hadoop solutions.
Mind shift

“If we can’t hire all these good people, why don’t we use them in a collaboration?”

• Collaboration instead of competition.
Mind shift

“Why do we reinvent the wheel?”

- Share the same technologies while creating different services.
Believe in the Apache way