PECARI
Pig Engineers Care About R Improvements

Tristan Ravitch, Evan Driscoll, and James Jolly

December 2, 2008
Outline

1. Introduction
2. Our Goal
3. Implementation
4. Future Work

Tristan Ravitch, Evan Driscoll, and James Jolly
PECARI Pig Engineers Care About R Improvements
What is Pig Latin?

Set-oriented data transformation language in which:
- Primitives filter, combine, split, and order data
- Users describe a sequence of transformations
- Set transformations are stateless
Advantages include:

- Set transformations massively parallel (Hadoop)
- Familiar programming style
- Support for semi-structured data
- Ad-hoc data analysis and manipulation
An Opportunity

Pig currently performs no optimizations.

- "the user knows best" philosophy
- Minimal knowledge of dataset
- Computing costs is hard
Outline

1. Introduction
2. Our Goal
3. Implementation
4. Future Work
Join Order Optimizations

Why these?
- High cost operations
- Non-trivial implementation
- Users adopt Pig to make joins simple
Example Query

```pig
a = LOAD 'input_A.txt' USING PigStorage('\t')
   AS (a_key, a_data);
-- ...
x = JOIN d BY d_key, c BY c_key;
y = JOIN x BY c_key, b BY b_key;
z = JOIN y BY b_key, a BY a_key;
STORE z INTO 'results.txt' USING PigStorage();
```
Logical Dataflow Reordering

\[ |A| = 5K, |B| = 50K, |C| = 500K, |D| = 5M \]

Smallest → largest vs. largest → smallest can significantly differ in execution speed
Outline

1. Introduction
2. Our Goal
3. Implementation
4. Future Work
Necessary Components

- A method to pick the best join order
- A method to compute join costs
- Relation cardinalities and key distributions
Distributed Sampling

- Hadoop stores data in HDFS
- Plenty of bandwidth and spare cycles

Each sampling daemon takes:

**Inputs**: filename, join attribute

**Output**: histograms
Estimating Join Cost

Examine *logical* plans by walking through the operator tree

- Gauge cardinality of inputs at load nodes
- Combine equidepth histograms at join nodes
What’s next?

- Performance numbers:
  - Time savings for various workloads
  - Overhead of sampling and optimization
- Evaluate bushy plans
Future Work

- Re-optimize on-the-fly
- Project early, rejoin later
- Informed DFS
- Distribute plan searches using MapReduce
Questions?

THAT CONCLUDES MY TWO-HOUR PRESENTATION. ANY QUESTIONS?

DID YOU INTEND THE PRESENTATION TO BE INCOMPREHENSIBLE, OR DO YOU HAVE SOME SORT OF RARE "POWER-POINT" DISABILITY?

ARE THERE ANY QUESTIONS ABOUT THE CONTENT?

THERE WAS CONTENT?