7 Things to Know When Buying for an

Shoemakers:
Alekh Jindal, Jorge Quiané, Jens Dittrich

1. Why Shoes?

MapReduce has been subject of active research in many aspects:
- Analysis and optimization of MapReduce jobs
- High-level query languages
- Efficient execution of MapReduce jobs
- and many others...

However, data layouts have not been explored in depth.

2. Why Our Elephant Needs Different Shoes?

<table>
<thead>
<tr>
<th>DBMS</th>
<th>MapReduce</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Typically deployed over a small number of nodes</td>
<td>• Typically deployed over thousands of nodes</td>
</tr>
<tr>
<td>• Usually use 8KB data page size</td>
<td>• Data block size of 64MB by default (up to 1GB)</td>
</tr>
<tr>
<td>• Data is typically replicated at the table-level</td>
<td>• Data is replicated 3 times by default at the block-level</td>
</tr>
</tbody>
</table>

3. What is Wrong with Old Shoes?

Row-, Column-, and PAX stores:
- Many redundant reads
- High network cost
- Complex data block placement
- High tuple reconstruction cost

4. What Shoes Do We Propose?

- We keep the external-view of a data block intact:
  • Data blocks store the same data as before
  • Inside a block: any layout e.g. pax, column-group
- We exploit the default data block replication:
  • Different layouts for different data block replicas
  • Each replica optimized for different query sub-class

5. How Do We Design the Shoes?

- **Interestingness function:** average normalized mutual information between any pair of attributes
  \[ I(G) = \frac{1}{G} \sum_{i,j} I(A_i, A_j) \]
  \[ |G| > 1 \]
  \[ |G| = 1 \]
- We consider the column group packing as 0-1 Knapsack problem
  \[ \max \sum_{i=1}^{m} I(G_i) \cdot \kappa_i \quad \text{subject to:} \]
  \[ \sum_{i=1}^{m} \kappa_i \leq \kappa \quad \forall \kappa, \forall j \in X \land j \notin G, \forall G \in \mathcal{G} \]

6. How Do We Ride Our Elephant?

1. Create the Trojan Layout configuration file in HDFS, e.g. MyDatasetName_Layer-1_Layer-2_Layer-3
2. Upload input into HDFS as before
3. Supply referenced attributes in the job configuration
4. Supply the itemize UDF to transparently read Trojan Layouts
5. Route map tasks to the best-layouts

Further Information

**Paper:** Trojan Data Layouts: Right Shoes for a Running Elephant. Alekh Jindal, Jorge Quiané, Jens Dittrich. SoCC 2011.

**Talk:** Big Data Session, Talk 3 (10:30-12:00).
Friday, 28th October.

**Hadoop++ Project:** http://infosys.uni-saarland.de/hadoop++

**OctopusDB Project:** http://infosys.uni-saarland.de/octopusdb.php

7. How Were the Field Trials?

![Graph showing field trial results over Hadoop-Row and Hadoop-PAX]

**TPC-H Queries Accessing Lineitem**

- **Improvement Factor**
  - Redundant Attributes Read
  - Spans in Tuple Reconstruction

**Hadoop-Row**
- Redundant Attributes Read: 84
- Spans in Tuple Reconstruction: 14

**Hadoop-PAX**
- Redundant Attributes Read: 8
- Spans in Tuple Reconstruction: 10

**Hyrise Layout**
- Redundant Attributes Read: 100
- Spans in Tuple Reconstruction: 100

**Trojan Layout**
- Redundant Attributes Read: 14
- Spans in Tuple Reconstruction: 20

**Upload Time**

- HDFS: 80
- Trojan HDFS: 40

SAARLAND UNIVERSITY. COPYRIGHT © 2011.