CARTILAGE: Flexible Hadoop Skeleton

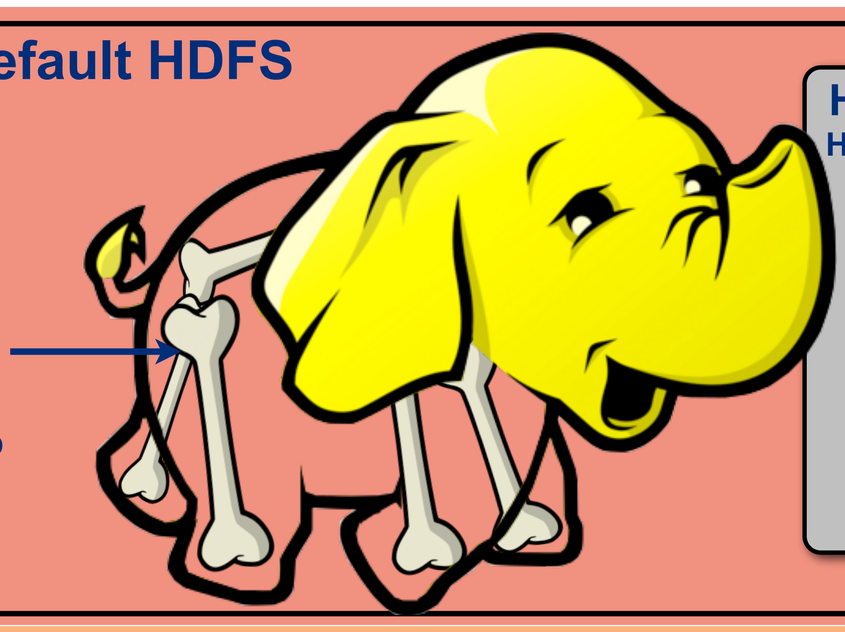
Alekh Jindal
CSAIL, MIT

Jorge Quiané
Data Analytics Group, QCRI

Samuel Madden
CSAIL, MIT

معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
Member of Qatar Foundation

Default HDFS



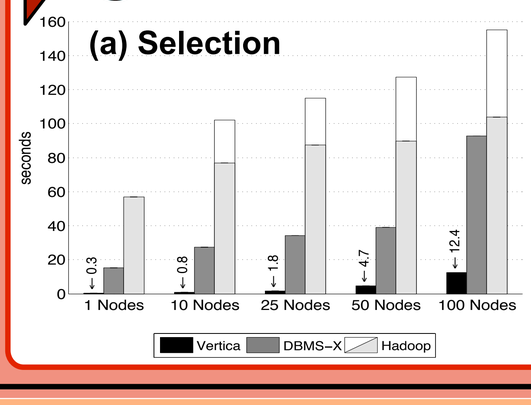
HDFS
Hard-coded Upload Pipeline

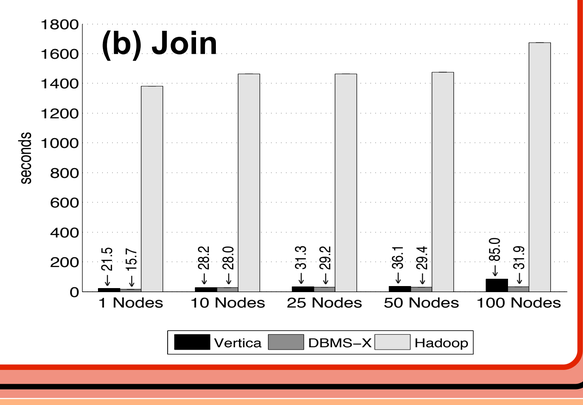
Store
Block Replica Placement
Replicator
Physical Partitioner
Parser
Input Dataset

results in

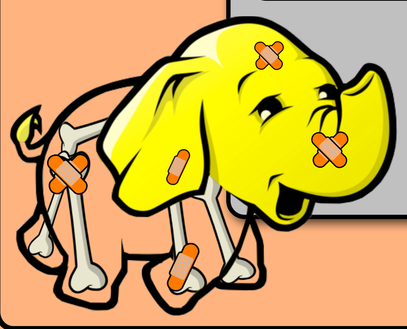
Poor Performance

1. Inflexible Upload Plan
2. Not suitable for many applications





Current Practice: modify the Hadoop code



HDFS
Hard-coded Upload Pipeline

Replica 1 | Replica 2 | Replica 3
New | New | New
Block Replica Placement
Replicator
Physical Partitioner
Parser
Input Dataset

results in

Some existing proposals:
- RCFile (ICDE'11)
- CIF (VLDB'11)
- TrojanLayouts (SOCC'11)
- HAIL (VLBD'12)

Issues

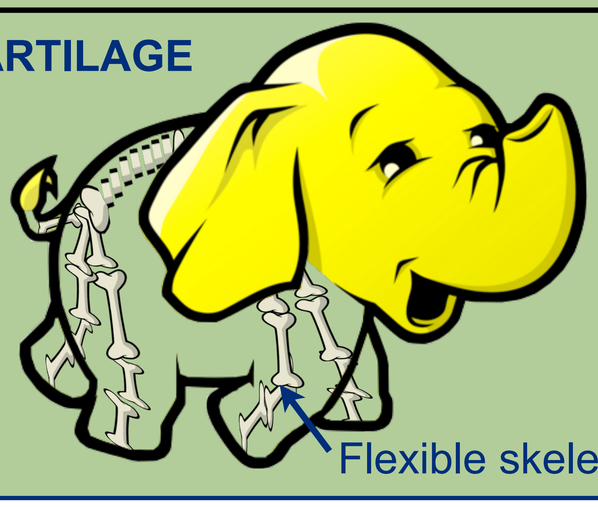
1. Improved performance only for specific workloads
2. Still a hard-coded upload plan
3. Deep changes to Hadoop
4. Hard to use

bringing us to...

Research Challenges

1. How to adapt storage to a large variety of workloads?
2. How to provide flexibility without code changes?
3. Flexibility vs Ease-of-Use vs Efficiency
4. How to preserve fault-tolerance?

CARTILAGE



Idea

1. Introduce a declarative *upload plan*
2. Decouple users datasets from physical files
3. Allow for flexible query processing

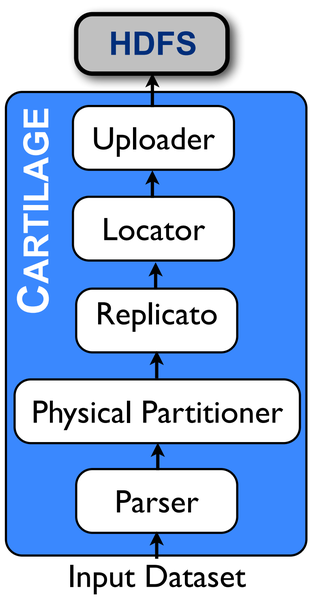
results in

Benefits

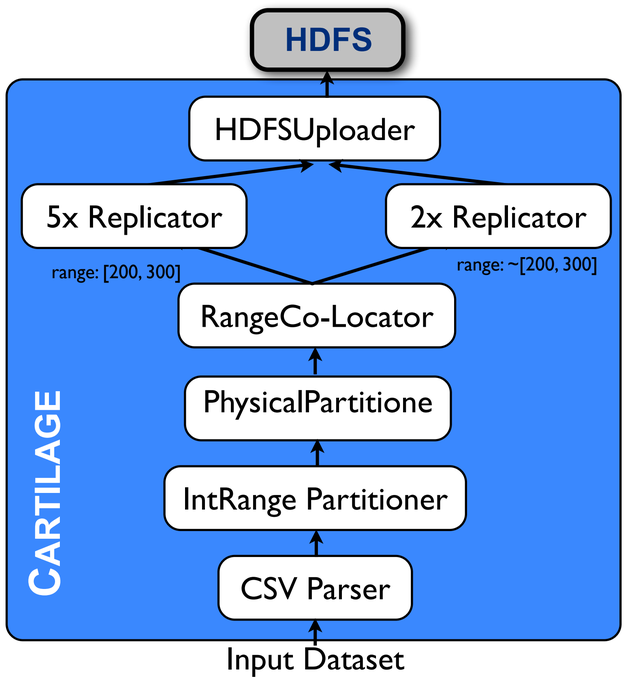
1. Flexible Upload Plan
2. Easy implementation of any data layout
3. Enable storage heterogeneity
4. Allow for new applications

Large Variety of Upload Plans

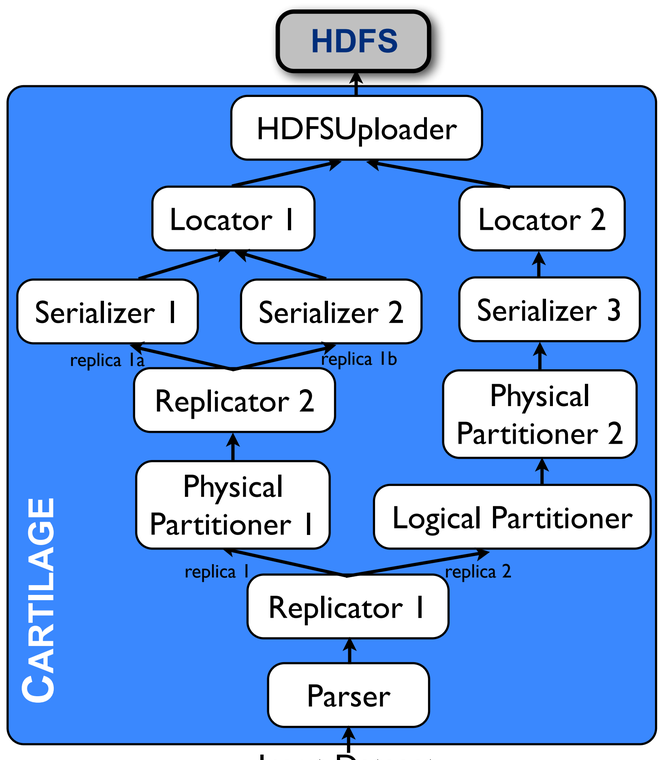
(a) Emulating HDFS



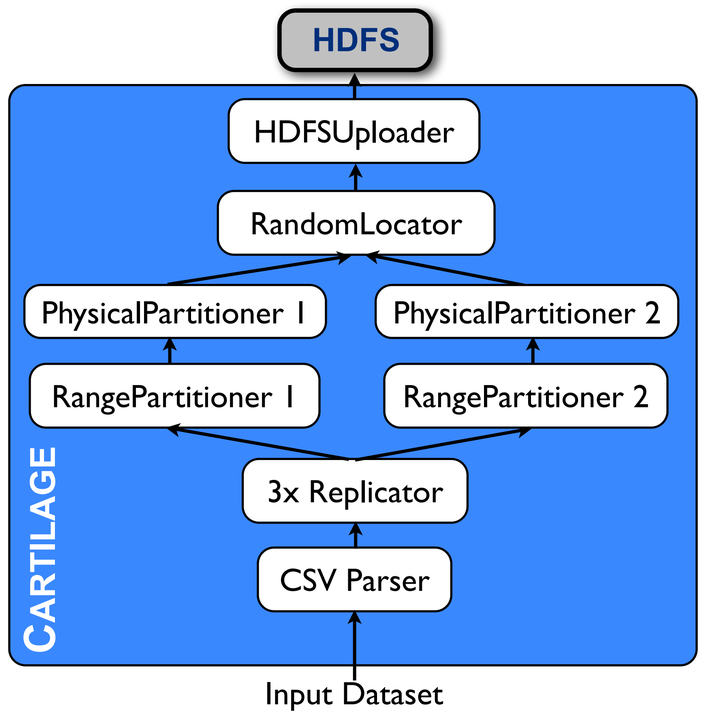
(b) Flexible data replication



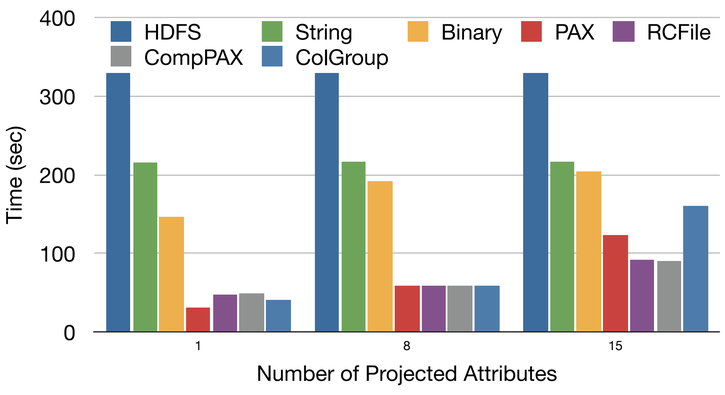
(c) Heterogenous storage



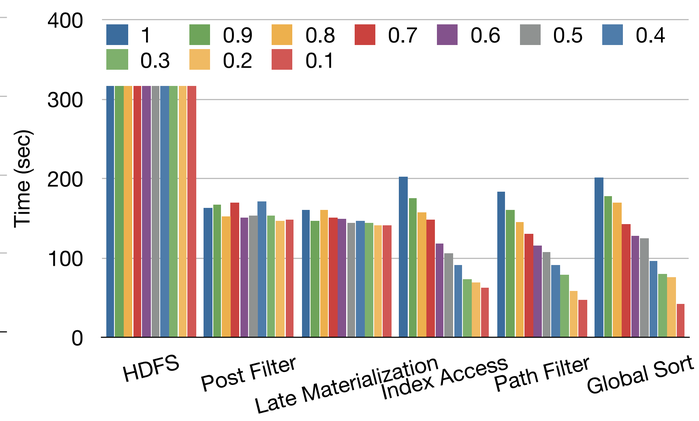
(d) Heterogenous partitioning



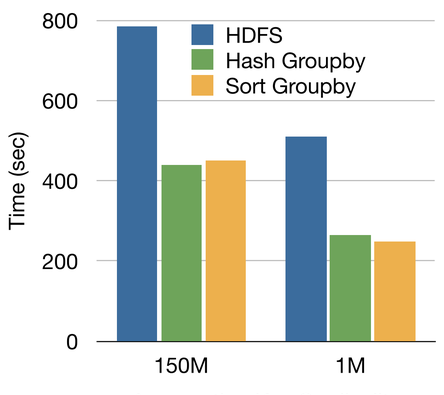
Improved Performance



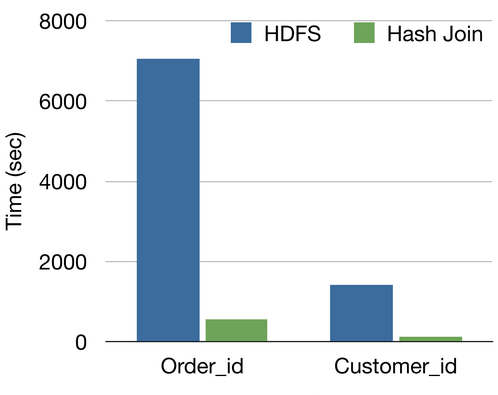
(a) Projection



(b) Selection



(c) Aggregation



(d) Join