Marimba:
A MapReduce-based Framework for Incremental Recomputations on Big Data

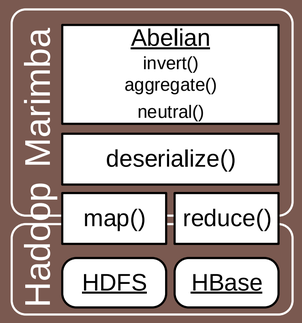
Johannes Schildgen (schildgen@cs.uni-kl.de)
TU Kaiserslautern, Germany

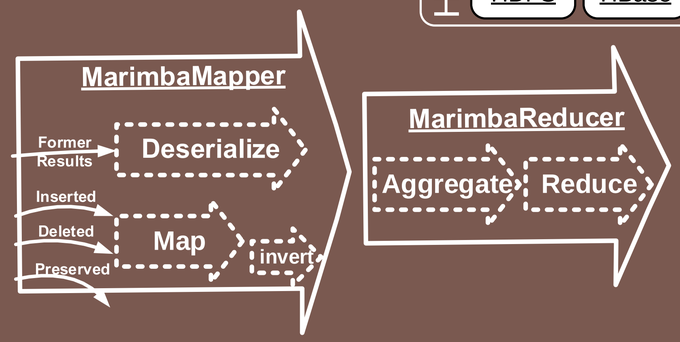
Background
Many MapReduce jobs for analyzing Big Data require many hours and have to be repeated again and again because the base data changes continuously. With simple MapReduce every job has to be executed from scratch and cannot benefit from previous computations.

Research Questions
- How to avoid full recomputations?
- Can a MapReduce job reuse its own result from a former computation?

What We Can Learn From Materialized Views
MapReduce jobs and materialized views in relational DBMS have a lot in common. A long running query is executed periodically to calculate a result. When this result can be computed by only analyzing changes the in base data (deltas) as well as the result of the former computation, a materialized view is called *self-maintainable*. Our approach to make MapReduce jobs self-maintainable is based on two popular concepts, Overwrite and Increment Installation:

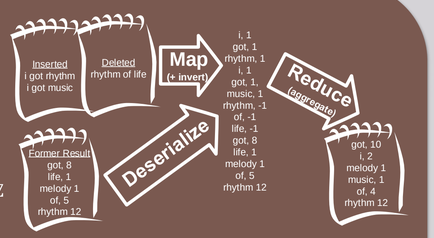
Marimba
- Simply write Map & Reduce functions,
- as intermediate value type use an Abelian type,
- the Marimba framework will only read inserted and deleted data,
- deleted values will be inverted,
- the changes will be aggregated on the previous result.
- Generic combiner (simply aggregates values)
- For Overwrite Installation, a Deserialize method has to be defined to convert the former result into Abelian objects.



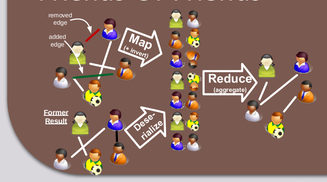


(dotted arrows: user-defined functions)

Examples
WordCount
- map(linenum, line): ∀word in line: emit(word, 1)
- invert: $\mathbb{Z} \to \mathbb{Z} := \cdot(-1)_{\mathbb{Z}}$
- aggregate: $\mathbb{Z} \times \mathbb{Z} \to \mathbb{Z} := +_{\mathbb{Z}}$
- identity element: $0_{\mathbb{Z}}$

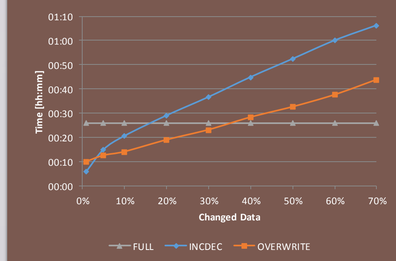


Friends-Of-Friends

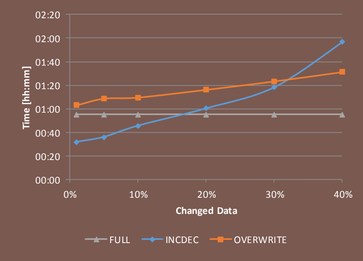


Other
- Reverse Web-Link Graph
- N-Grams
- PageRank
- ...

Results



WordCount



Reverse Web-Link Graph

ADBIS 2013 Advances in Databases and Information Systems

HiS

Technische Universität Kaiserslautern