DLI Accelerated Data Science Teaching Kit

Lecture 11.1 - Spark Overview

http://spark.apache.org

Not a modified version of Hadoop

Separate, fast, MapReduce-like engine

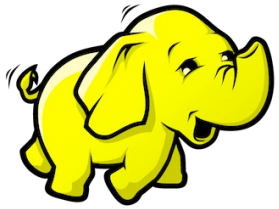
» **In-memory** data storage for very fast iterative queries

» General execution graphs and powerful optimizations

» Up to 100x faster than Hadoop MapReduce in memory

Compatible with Hadoop's storage APIs

» Can read/write to any Hadoop-supported system, including HDFS, HBase, SequenceFiles, etc.



Port of Apache Hive to run on Spark

Compatible with Hive data, metastores, and queries (HiveQL, UDFs, etc)

Similar speedups of up to 40x

Project History

Spark project started in 2009 at UC Berkeley AMP lab, open sourced 2010

Became Apache Top-Level Project in Feb 2014

Spark SQL started summer 2011

Built by 1000+ developers and people from 200 companies

Scale to 1000+ nodes in production

Used by many companies and organizations: Amazon, eBay, IBM, NASA, Yahoo, …

http://en.wikipedia.org/wiki/Apache_Spark

Why a New Programming Model?

MapReduce greatly simplified big data analysis

But as soon as it got popular, users wanted more:

- » More **complex**, multi-stage applications (e.g. iterative graph algorithms and machine learning)
- » More **interactive** ad-hoc queries

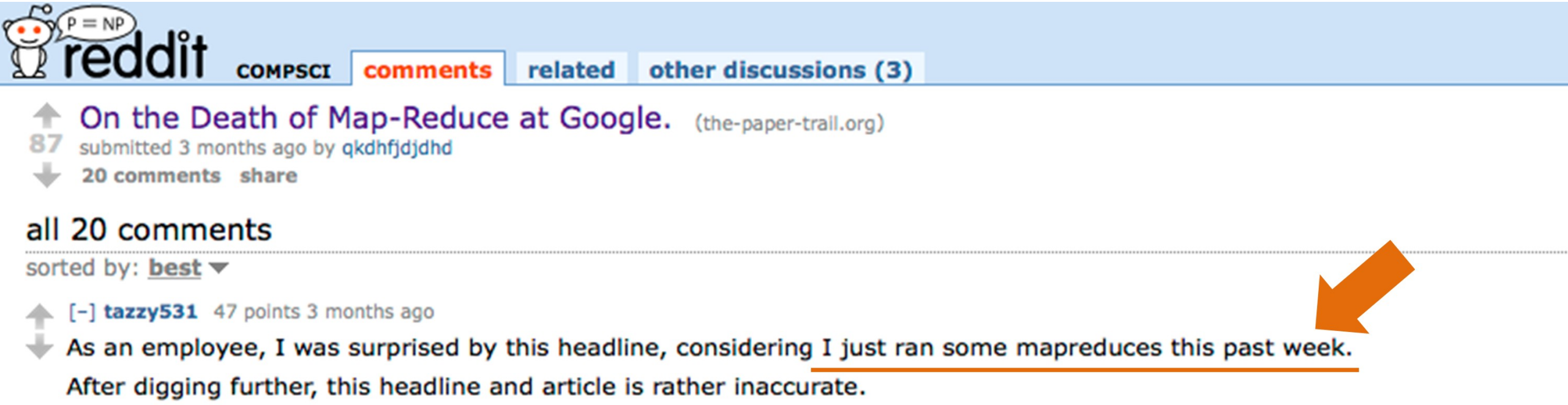
Require faster **data sharing** across parallel jobs

Is MapReduce dead? Not really.

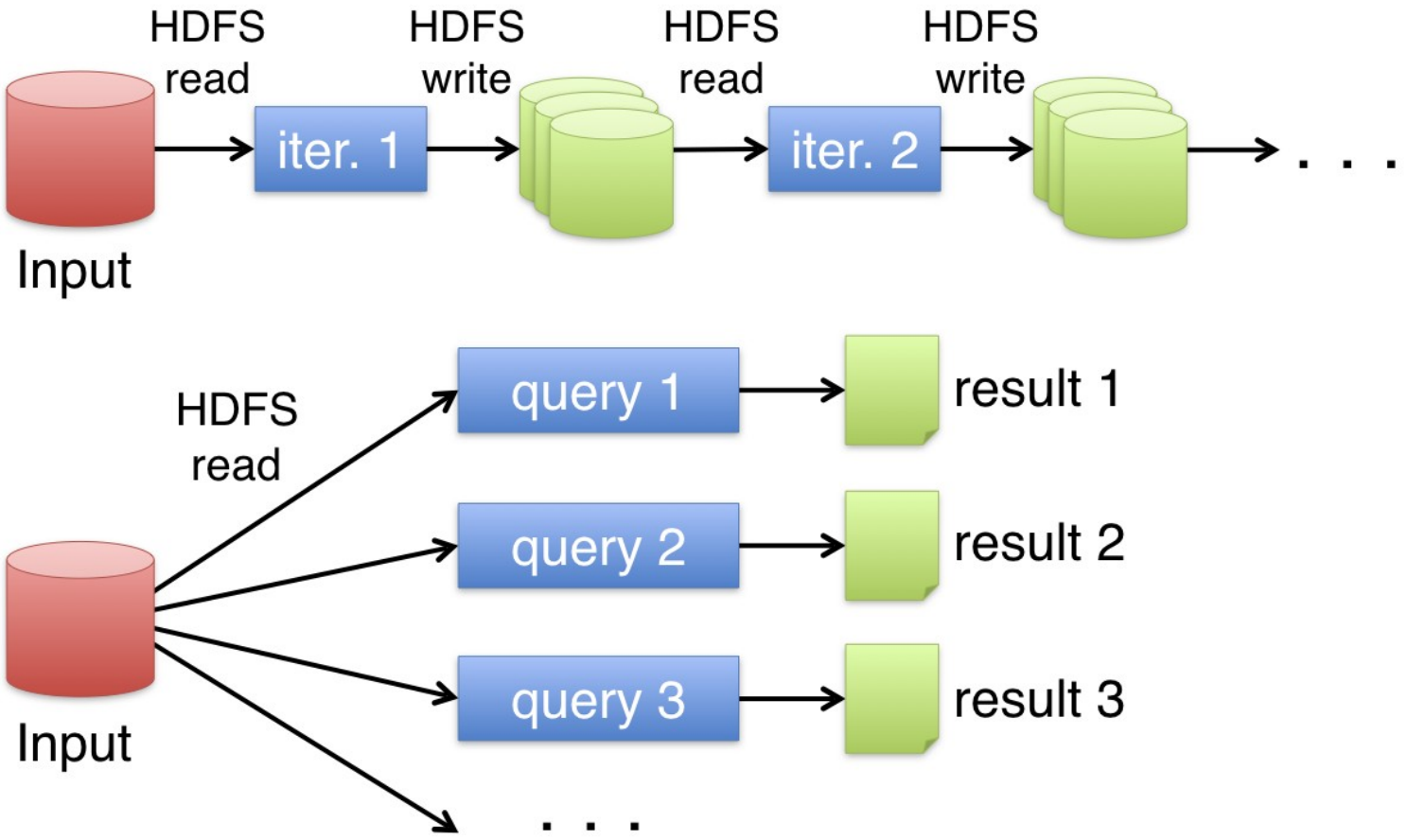
Google Dumps MapReduce in Favor of New Hyper-Scale Analytics System

http://www.datacenterknowledge.com/archives/2014/06/25/google-dumps-mapreduce-favor-new-hyper-scale-analytics-system/

http://www.reddit.com/r/compsci/comments/296aqr/on_the_death_of_mapreduce_at_google/

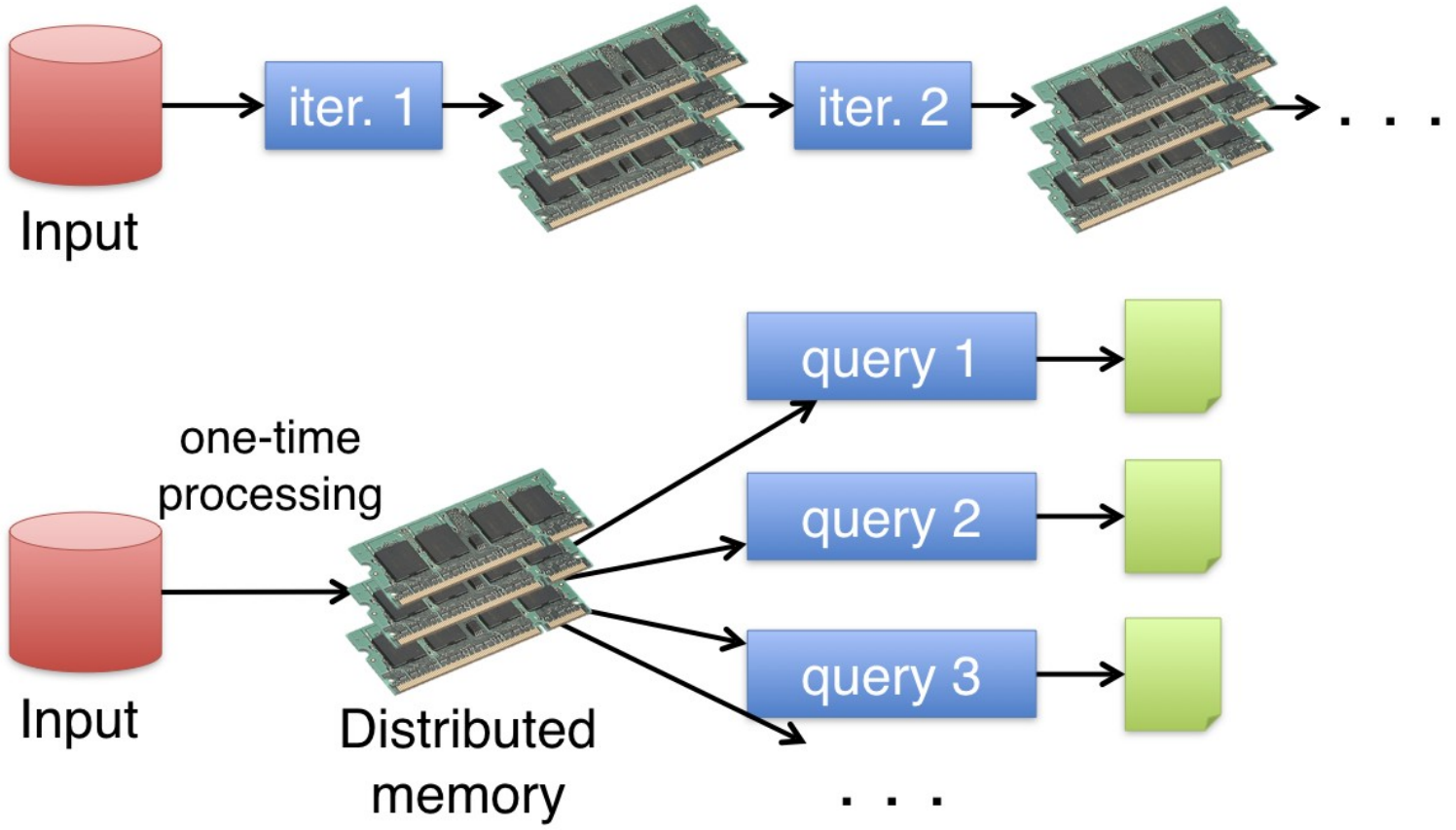


Data Sharing in Hadoop MapReduce



Slow due to replication, serialization, and disk IO

Data Sharing in Spark



10-100× faster than network and disk

Spark Programming Model

Key idea: *resilient distributed datasets (RDDs)*

- » Distributed collections of objects that can be cached in memory across cluster nodes
- » Manipulated through various parallel operators
- » Automatically rebuilt on failure

Interface

- » Supported languages: Java, **Scala**, Python, R
- » Can be used *interactively* from Scala, Python console

DLI Accelerated Data Science Teaching Kit

Thank You

We thank Dr. Matei Zaharia for sharing teaching materials for Spark.