Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing

M. Zaharia et al. (2012)

Christopher Little

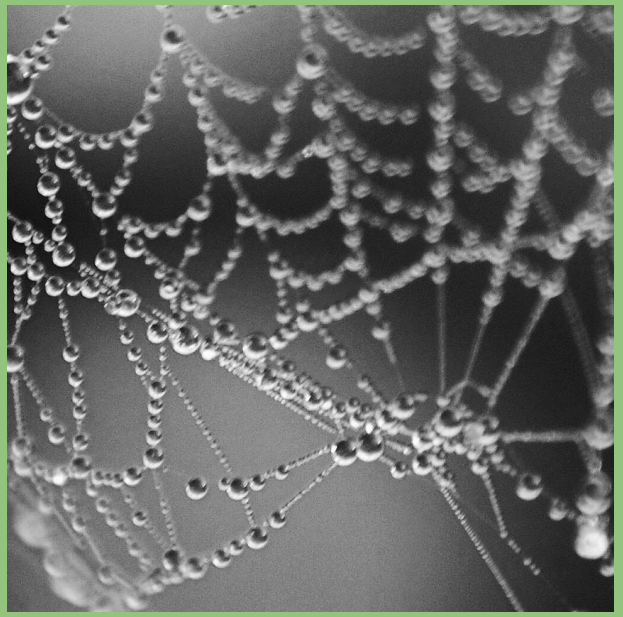
Outline

Context

Resilient Distributed Datasets

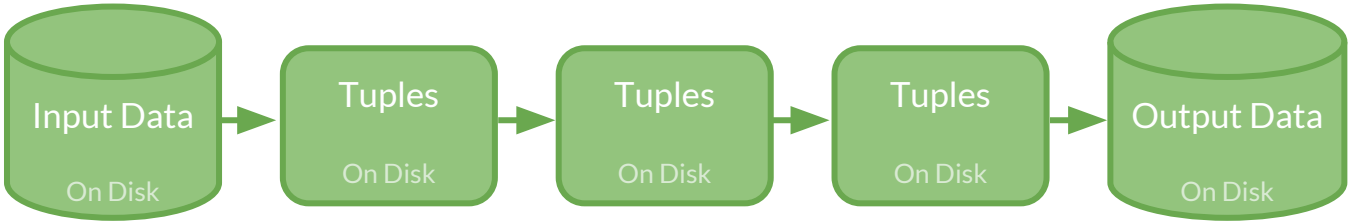
Spark

Evaluation

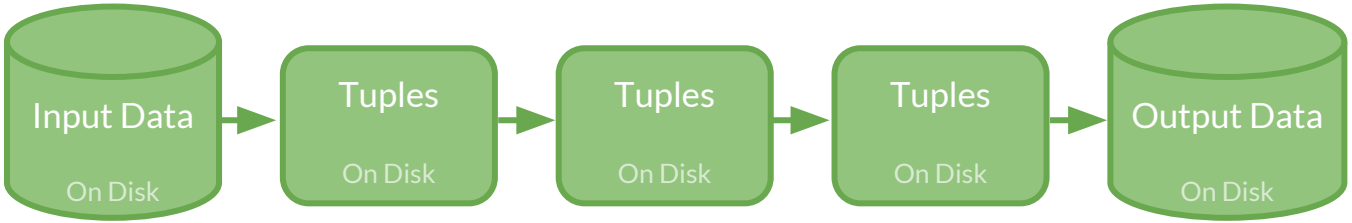


Context.

MapReduce


Input Data
On Disk
Tuples
On Disk
Tuples
On Disk
Tuples
On Disk
Output Data
On Disk

MapReduce

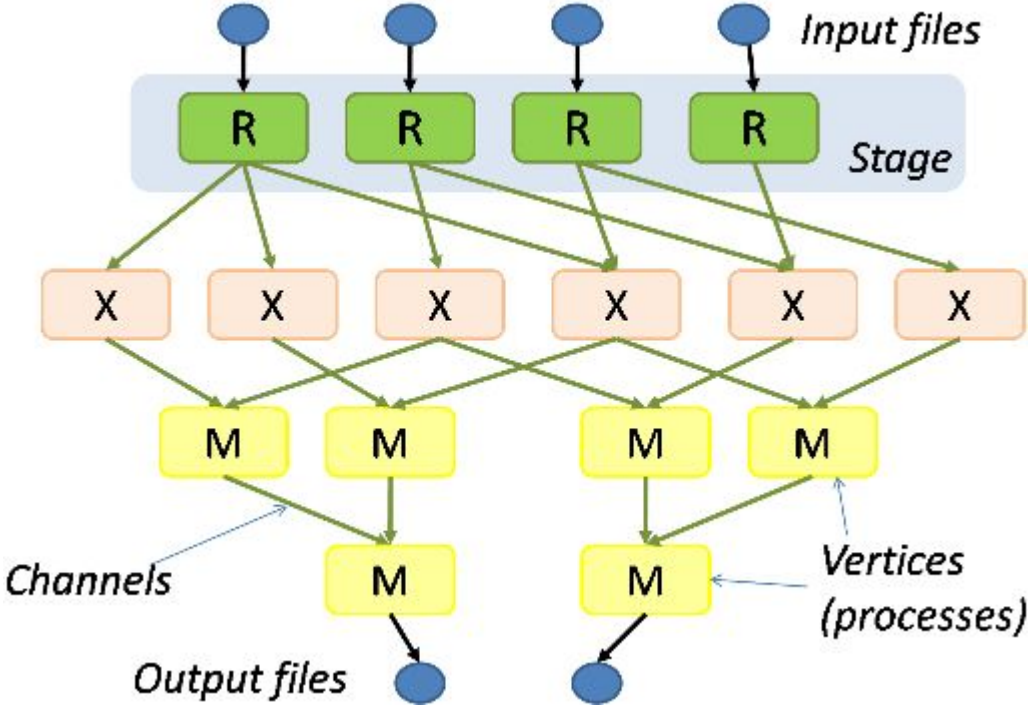


HaLoop

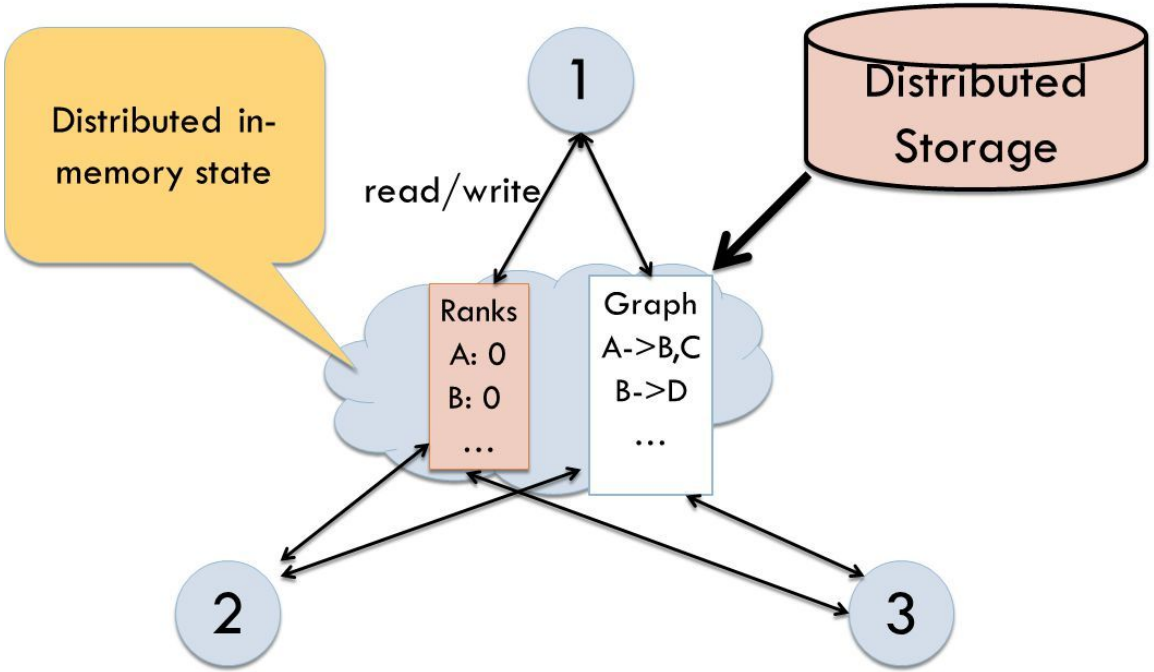
Google Pregel

APACHE GIRAPH

Dryad


Input files
R
R
R
R
Stage
X
X
X
X
X
X
M
M
M
M
Channels
M
M
Vertices
(processes)
Output files

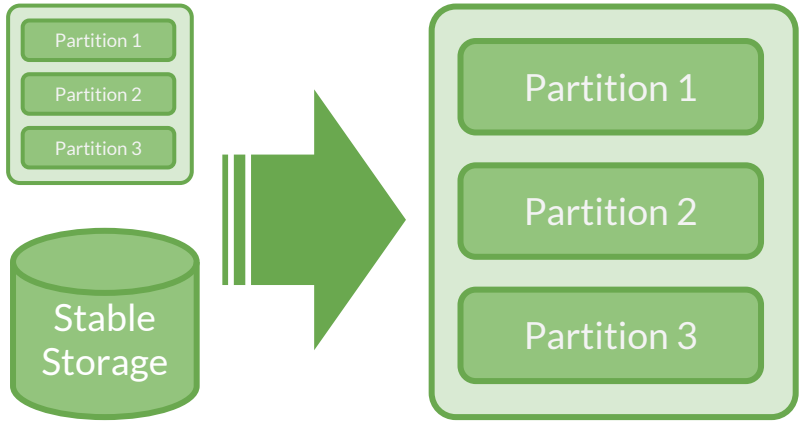
Distributed Shared Memory



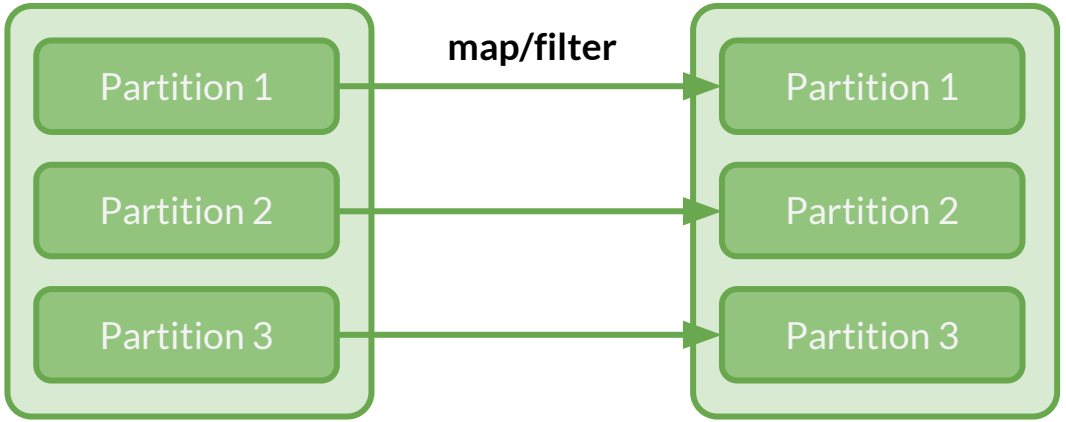
Resilient Distributed Datasets (RDD).

Structure of RDDs

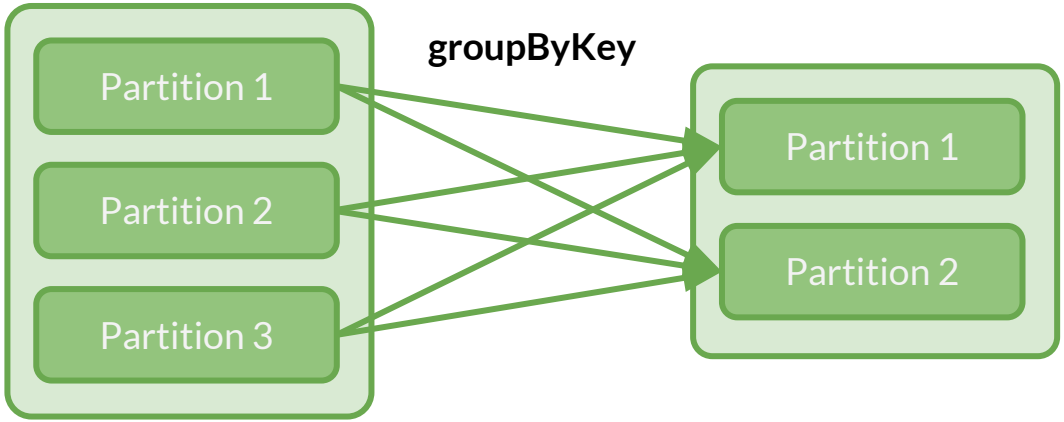
Structure of RDDs



Structure of RDDs



Structure of RDDs



Spark.

Spark example

```
lines = spark.textFile( "hdfs://..." )
errors = lines.filter( _.startsWith( "ERROR" ) )

errors.persist()

// Count errors mentioning MySQL:
errors.filter( _.contains( "MySQL" ) )
      .count() // Nothing computed until now

// Return the time fields of errors mentioning
// HDFS as an array (assuming time is field
// number 3 in a tab-separated format):
errors.filter( _.contains( "HDFS" ) )
      .map( _.split( '\t' )( 3 ) )
      .collect( )
```

Scheduling

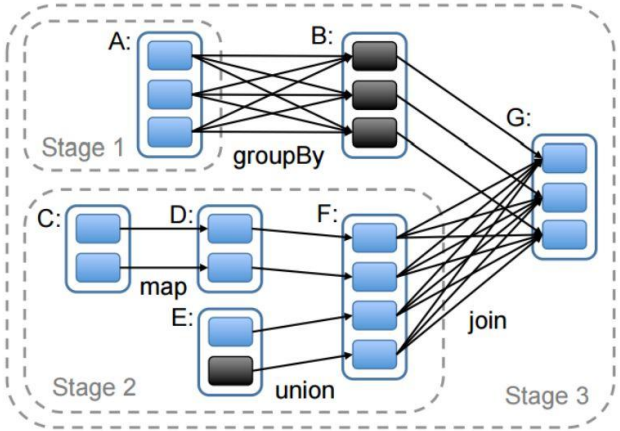


Figure 5: Example of how Spark computes job stages. Boxes with solid outlines are RDDs. Partitions are shaded rectangles, in black if they are already in memory. To run an action on RDD G, we build build stages at wide dependencies and pipeline narrow transformations inside each stage. In this case, stage 1's output RDD is already in RAM, so we run stage 2 and then 3.

Evaluation.

20x

Spark claims to be up to 20x faster than Hadoop in iterative applications

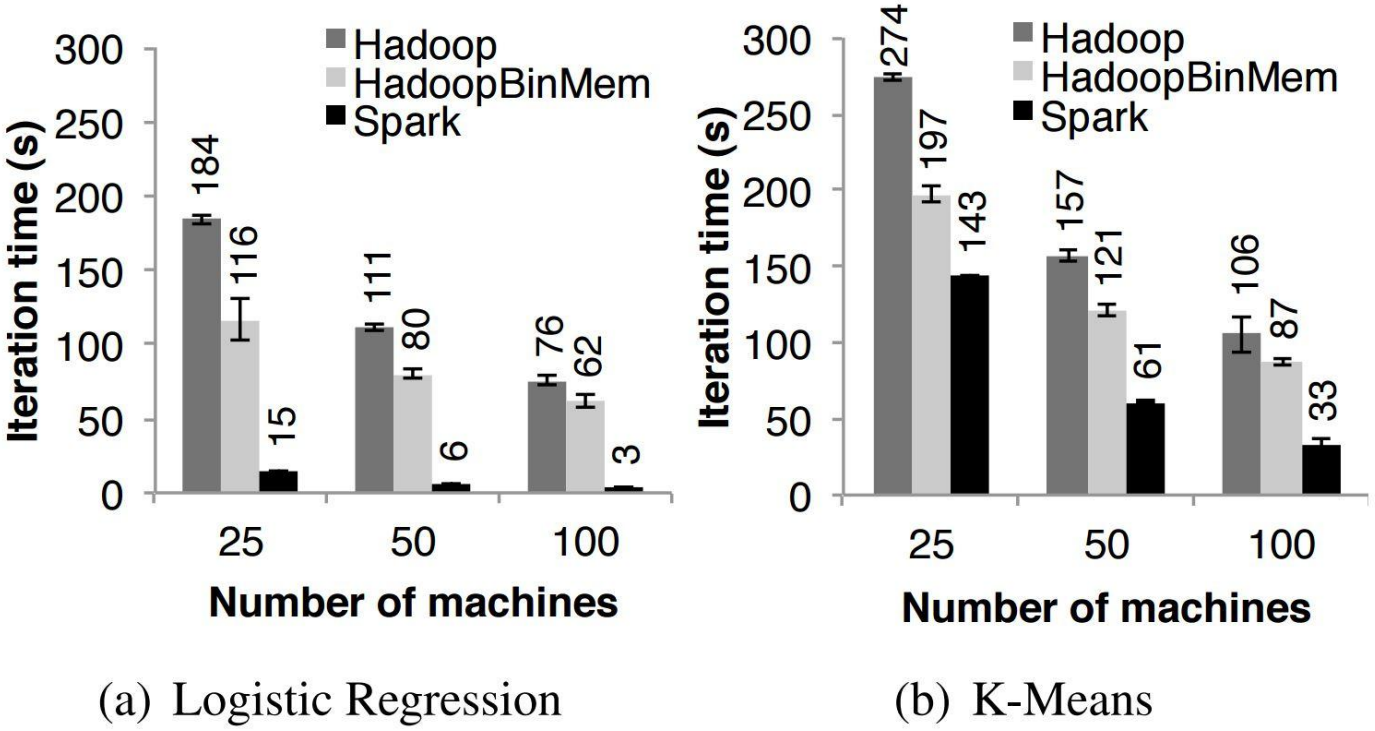


Figure 8: Running times for iterations after the first in Hadoop, HadoopBinMem, and Spark. The jobs all processed 100 GB.

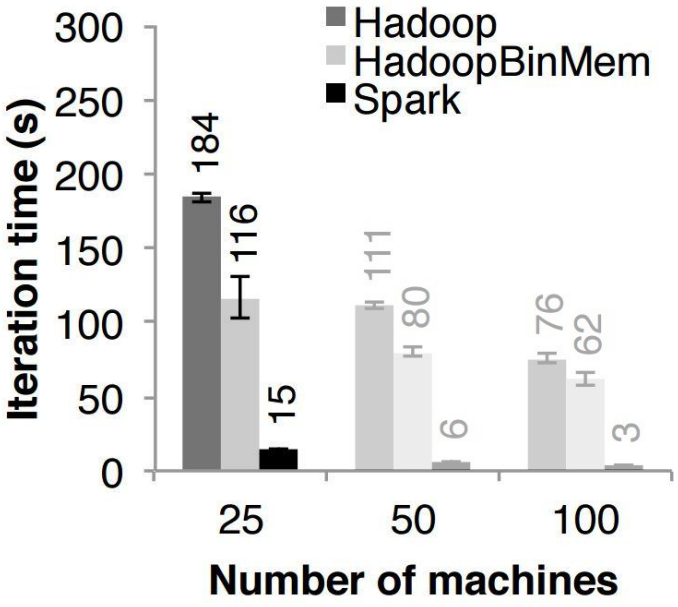


Figure 8
(logistic regression using 100 GB)

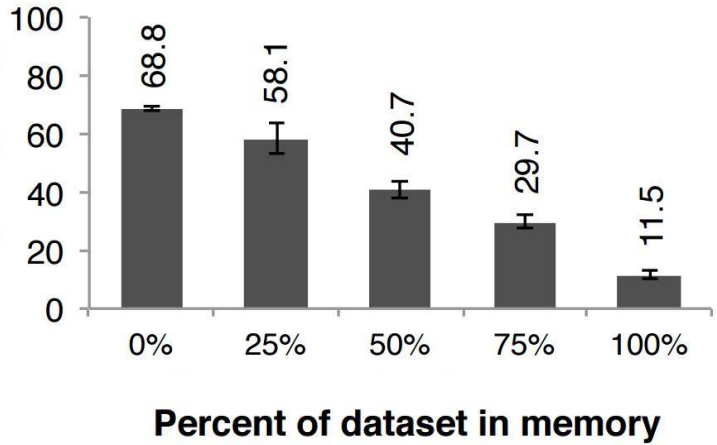


Figure 12
(logistic regression using 100 GB on 25 nodes)

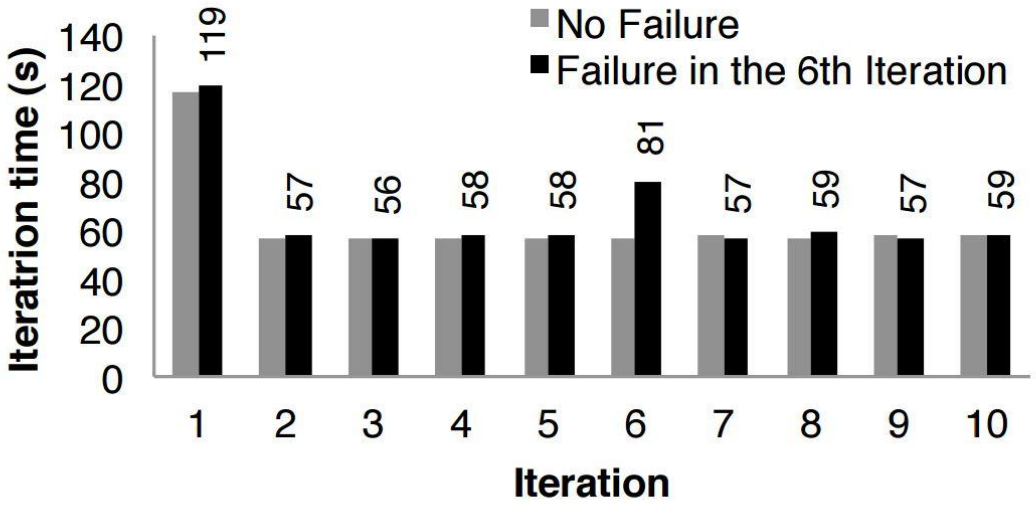


Figure 11: Iteration times for k-means in presence of a failure. One machine was killed at the start of the 6th iteration, resulting in partial reconstruction of an RDD using lineage.

Spark promises a lot, but the evidence presented here is insufficient.

(but it *could* live up to their claims)