Revisiting the MapReduce Paradigm: an R-Specific View

Norm Matloff and Alex Rumbaugh
University of California at Davis

UCB R Group
May 19, 2015
Updated, May 22

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Visual Summary

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Visual Summary

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Visual Summary

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Visual Summary

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Visual Summary

My view: Plain Old R can work better in many situations

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview

- When I was here one year ago, I speculated that Hadoop would start to lose popularity sometime in the future.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview

- When I was here one year ago, I speculated that Hadoop would start to lose popularity sometime in the future.
  - Too slow.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview

- When I was here one year ago, I speculated that Hadoop would start to lose popularity sometime in the future.
  - Too slow.
  - Not many ops.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview

- When I was here one year ago, I speculated that Hadoop would start to lose popularity sometime in the future.

  - Too slow.
  - Not many ops.

- That time seems to have begun.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview

- When I was here one year ago, I speculated that Hadoop would start to lose popularity sometime in the future.

  - Too slow.
  - Not many ops.

- That time seems to have begun.

- E.g. see *The Hadoop Honeymoon Is Over*,
  http://smartdatacollective.com/martynjones/
  318406/hadoop-honeymoon-over

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview

- When I was here one year ago, I speculated that Hadoop would start to lose popularity sometime in the future.
  - Too slow.
  - Not many ops.
- That time seems to have begun.
- E.g. see *The Hadoop Honeymoon Is Over*, http://smartdatacollective.com/martynjones/ 318406/hadoop-honeymoon-over
- There is a new kid on the block, Spark,

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview

- When I was here one year ago, I speculated that Hadoop would start to lose popularity sometime in the future.

  - Too slow.
  - Not many ops.

- That time seems to have begun.

- E.g. see *The Hadoop Honeymoon Is Over*,
  http://smartdatacollective.com/martynjones/
  318406/hadoop-honeymoon-over

- There is a new kid on the block, Spark, with an R interface, SparkR, a big improvement

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview

- When I was here one year ago, I speculated that Hadoop would start to lose popularity sometime in the future.
  - Too slow.
  - Not many ops.
- That time seems to have begun.
- E.g. see *The Hadoop Honeymoon Is Over*, http://smartdatacollective.com/martynjones/318406/hadoop-honeymoon-over
- There is a new kid on the block, Spark, with an R interface, SparkR, a big improvement
- But I will argue that for us R users, the utility of either Hadoop or SparkR is much more limited than many people realize.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview

- When I was here one year ago, I speculated that Hadoop would start to lose popularity sometime in the future.

  - Too slow.
  - Not many ops.

- That time seems to have begun.

- E.g. see *The Hadoop Honeymoon Is Over*,
  http://smartdatacollective.com/martynjones/
  318406/hadoop-honeymoon-over

- There is a new kid on the block, Spark, with an R interface, SparkR, a big improvement

- But I will argue that for us R users, the utility of either Hadoop or SparkR is much more limited than many people realize.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview

- When I was here one year ago, I speculated that Hadoop would start to lose popularity sometime in the future.
    - Too slow.
    - Not many ops.
- That time seems to have begun.
- E.g. see *The Hadoop Honeymoon Is Over*,
  http://smartdatacollective.com/martynjones/
  318406/hadoop-honeymoon-over
- There is a new kid on the block, Spark, with an R interface, SparkR, a big improvement
- But I will argue that for us R users, the utility of either Hadoop or SparkR is much more limited than many people realize.
- And I will present an alternative.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview of MapReduce

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview of MapReduce

- Parallel ops (cluster or multicore).

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview of MapReduce

- Parallel ops (cluster or multicore).
- Work flow: map/sort/reduce.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview of MapReduce

- Parallel ops (cluster or multicore).
- Work flow: map/sort/reduce.
- Example: word count.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview of MapReduce

- Parallel ops (cluster or multicore).
- Work flow: map/sort/reduce.
- Example: word count.
  - Map: Read a line, break into words, emit one record for each (with count 1).

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview of MapReduce

- Parallel ops (cluster or multicore).
- Work flow: map/sort/reduce.
- Example: word count.
  - Map: Read a line, break into words, emit one record for each (with count 1).
  - Sort by word.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview of MapReduce

- Parallel ops (cluster or multicore).
- Work flow: map/sort/reduce.
- Example: word count.
    - Map: Read a line, break into words, emit one record for each (with count 1).
    - Sort by word.
    - Get counts by adding all the 1s for each unique word.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Overview of MapReduce

- Parallel ops (cluster or multicore).
- Work flow: map/sort/reduce.
- Example: word count.
  - Map: Read a line, break into words, emit one record for each (with count 1).
  - Sort by word.
  - Get counts by adding all the 1s for each unique word.
- Most famous example: Hadoop.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's Wrong with Hadoop

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's Wrong with Hadoop

- SLOW.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's Wrong with Hadoop

- SLOW. Like an elephant. :-)

  - Input to code must come from disk, output must be written to disk.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's Wrong with Hadoop

- SLOW. Like an elephant. :-)
  - Input to code must come from disk, output must be written to disk.
  - Awful for iterative algorithms.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's Wrong with Hadoop

- SLOW. Like an elephant. :-)

  - Input to code must come from disk, output must be
    written to disk.
  - Awful for iterative algorithms.
  - Sort phase (*shuffle*) is performed even if one's algorithm
    doesn't need it.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's Wrong with Hadoop

- SLOW. Like an elephant. :-)

  - Input to code must come from disk, output must be written to disk.
  - Awful for iterative algorithms.
  - Sort phase (*shuffle*) is performed even if one's algorithm doesn't need it.

- Difficult to install/configure.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's Wrong with Hadoop

- SLOW. Like an elephant. :-)
  - Input to code must come from disk, output must be written to disk.
  - Awful for iterative algorithms.
  - Sort phase (*shuffle*) is performed even if one's algorithm doesn't need it.
- Difficult to install/configure. Not everyone is a systems expert.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's Wrong with Hadoop

- SLOW. Like an elephant. :-)

  - Input to code must come from disk, output must be written to disk.
  - Awful for iterative algorithms.
  - Sort phase (*shuffle*) is performed even if one's algorithm doesn't need it.

- Difficult to install/configure. Not everyone is a systems expert. Even worse when also need to install R interface.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's Wrong with Hadoop

- SLOW. Like an elephant. :-)
  - Input to code must come from disk, output must be written to disk.
  - Awful for iterative algorithms.
  - Sort phase (*shuffle*) is performed even if one's algorithm doesn't need it.
- Difficult to install/configure. Not everyone is a systems expert. Even worse when also need to install R interface.
- Map and reduce ops too low-level.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's Wrong with Hadoop

- SLOW. Like an elephant. :-)

    - Input to code must come from disk, output must be written to disk.
    - Awful for iterative algorithms.
    - Sort phase (*shuffle*) is performed even if one's algorithm doesn't need it.

- Difficult to install/configure. Not everyone is a systems expert. Even worse when also need to install R interface.

- Map and reduce ops too low-level. "Build a house from matchsticks."

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What Hadoop Gets Right

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What Hadoop Gets Right

- Distributed file system (HDFS).

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What Hadoop Gets Right

- Distributed file system (HDFS).

- "Move the computation to the data," rather than *vice versa*.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What Hadoop Gets Right

- Distributed file system (HDFS).
- "Move the computation to the data," rather than *vice versa*.
- Thus reduce time-consuming network communication time.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What Hadoop Gets Right

- Distributed file system (HDFS).
- "Move the computation to the data," rather than *vice versa*.
- Thus reduce time-consuming network communication time.
- Redundancy/fault tolerance, very important if have a huge cluster.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Spark

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Spark

- Extended map/reduce paradigm.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Spark

- Extended map/reduce paradigm.
- Cacheability of intermediate results, i.e. no costly writes to disk.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Spark

- Extended map/reduce paradigm.

- Cacheability of intermediate results, i.e. no costly writes to disk.

- *Lazy* computation: programmer's several specified ops automatically combined into faster coalesced code.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Spark

- Extended map/reduce paradigm.
- Cacheability of intermediate results, i.e. no costly writes to disk.
- *Lazy* computation: programmer's several specified ops automatically combined into faster coalesced code.
- Shuffle often avoided.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Spark

- Extended map/reduce paradigm.
- Cacheability of intermediate results, i.e. no costly writes to disk.
- *Lazy* computation: programmer's several specified ops automatically combined into faster coalesced code.
- Shuffle often avoided.
- Runs on top of HDFS or other DFS,

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Spark

- Extended map/reduce paradigm.

- Cacheability of intermediate results, i.e. no costly writes to disk.

- *Lazy* computation: programmer's several specified ops automatically combined into faster coalesced code.

- Shuffle often avoided.

- Runs on top of HDFS or other DFS, so retain "move the computation to the data" philosoophy.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Spark

- Extended map/reduce paradigm.

- Cacheability of intermediate results, i.e. no costly writes to disk.

- *Lazy* computation: programmer's several specified ops automatically combined into faster coalesced code.

- Shuffle often avoided.

- Runs on top of HDFS or other DFS, so retain "move the computation to the data" philosoophy.

- Typically way faster than Hadoop.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Spark

- Extended map/reduce paradigm.
- Cacheability of intermediate results, i.e. no costly writes to disk.
- *Lazy* computation: programmer's several specified ops automatically combined into faster coalesced code.
- Shuffle often avoided.
- Runs on top of HDFS or other DFS, so retain "move the computation to the data" philosoophy.
- Typically way faster than Hadoop.
- Has various high-level ops, not just map and reduce.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Spark

- Extended map/reduce paradigm.

- Cacheability of intermediate results, i.e. no costly writes to disk.

- *Lazy* computation: programmer's several specified ops automatically combined into faster coalesced code.

- Shuffle often avoided.

- Runs on top of HDFS or other DFS, so retain "move the computation to the data" philosoophy.

- Typically way faster than Hadoop.

- Has various high-level ops, not just map and reduce.

- Elegant, sophisticated fault-tolerance mechanism.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Drawbacks to Spark

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Drawbacks to Spark

- Still have installation/configuration headaches, even worse than Hadoop.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Drawbacks to Spark

- Still have installation/configuration headaches, even worse than Hadoop. Ditto for SparkR.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Drawbacks to Spark

- Still have installation/configuration headaches, even worse than Hadoop. Ditto for SparkR.

- High-level ops are abstract, steep learning curve.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Drawbacks to Spark

- Still have installation/configuration headaches, even worse than Hadoop. Ditto for SparkR.

- High-level ops are abstract, steep learning curve. (Where have we heard that before?)

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Drawbacks to Spark

- Still have installation/configuration headaches, even worse than Hadoop. Ditto for SparkR.

- High-level ops are abstract, steep learning curve. (Where have we heard that before?)

- Not clear that SparkR has much advantage over Plain Old R (POR).

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Drawbacks to Spark

- Still have installation/configuration headaches, even worse than Hadoop. Ditto for SparkR.

- High-level ops are abstract, steep learning curve. (Where have we heard that before?)

- Not clear that SparkR has much advantage over Plain Old R (POR). See next slide.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's in It for R Users?

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's in It for R Users?

- Granted, Hadoop/Spark have automatic fault tolerance, and an efficient sort, both important.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's in It for R Users?

- Granted, Hadoop/Spark have automatic fault tolerance, and an efficient sort, both important.

- But many apps don't need a sort, and many Hadoop users have small clusters (Hadoop Wiki, https://wiki.apache.org/hadoop/PoweredBy).

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's in It for R Users?

- Granted, Hadoop/Spark have automatic fault tolerance, and an efficient sort, both important.

- But many apps don't need a sort, and many Hadoop users have small clusters (Hadoop Wiki, https://wiki.apache.org/hadoop/PoweredBy).

- So, POR seems preferable for many users.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's in It for R Users?

- Granted, Hadoop/Spark have automatic fault tolerance, and an efficient sort, both important.

- But many apps don't need a sort, and many Hadoop users have small clusters (Hadoop Wiki, https://wiki.apache.org/hadoop/PoweredBy).

- So, POR seems preferable for many users.

  - No Java/database/configuration issues.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's in It for R Users?

- Granted, Hadoop/Spark have automatic fault tolerance, and an efficient sort, both important.

- But many apps don't need a sort, and many Hadoop users have small clusters (Hadoop Wiki, https://wiki.apache.org/hadoop/PoweredBy).

- So, POR seems preferable for many users.
  - No Java/database/configuration issues.
  - No need to learn new abstractions.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's in It for R Users?

- Granted, Hadoop/Spark have automatic fault tolerance, and an efficient sort, both important.

- But many apps don't need a sort, and many Hadoop users have small clusters (Hadoop Wiki, https://wiki.apache.org/hadoop/PoweredBy).

- So, POR seems preferable for many users.

  - No Java/database/configuration issues.
  - No need to learn new abstractions.
  - No forced shuffle.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's in It for R Users?

- Granted, Hadoop/Spark have automatic fault tolerance, and an efficient sort, both important.

- But many apps don't need a sort, and many Hadoop users have small clusters (Hadoop Wiki, https://wiki.apache.org/hadoop/PoweredBy).

- So, POR seems preferable for many users.
  - No Java/database/configuration issues.
  - No need to learn new abstractions.
  - No forced shuffle.
  - POR at least as expressive as SparkR, and already familiar.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

What's in It for R Users?

- Granted, Hadoop/Spark have automatic fault tolerance, and an efficient sort, both important.

- But many apps don't need a sort, and many Hadoop users have small clusters (Hadoop Wiki, https://wiki.apache.org/hadoop/PoweredBy).

- So, POR seems preferable for many users.
  - No Java/database/configuration issues.
  - No need to learn new abstractions.
  - No forced shuffle.
  - POR at least as expressive as SparkR, and already familiar.

- We've developed Snowdoop as an alternative:

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop Overview

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop Overview

- Pure POR!

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop Overview

- Pure POR! Uses the portion of R's **parallel** package adapted from the old **Snow**.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop Overview

- Pure POR! Uses the portion of R's **parallel** package adapted from the old **Snow**.
- Retains the DFS philosoophy.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop Overview

- Pure POR! Uses the portion of R's **parallel** package adapted from the old **Snow**.
- Retains the DFS philosoophy.
- Includes a distributed sort; could be optimized.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop Overview

- Pure POR! Uses the portion of R's **parallel** package adapted from the old **Snow**.

- Retains the DFS philosoophy.

- Includes a distributed sort; could be optimized.

- So simple, it's embarrassing, tough to call it a "package."

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Word Count

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Word Count

```
# prep: use SD filesplit(), load SD at nodes

# ndigs is number of digists in file suffix

fullwordcount <- function(cls, basename, ndigs) {
  counts <- clusterCall(cls, wordcensus, basename, ndigs)
  addlistssum <- function(lst1, lst2)
    addlists(lst1, lst2, sum) # SD
  Reduce(addlistssum, counts)
}

wordcensus <- function(basename, ndigs) {
  fname <- filechunkname(basename, ndigs) # SD
  words <- scan(fname, what="")
  tapply(words, words, length, simplify=FALSE)
}
```

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Word Count, cont'd.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Word Count, cont'd.

Test:

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Word Count, cont'd.

Test:

```
library(partools)
cls <- makeCluster(2)
setclsinfo(cls)  # SD
clusterEvalQ(cls, library(partools))
filesplit(cls, "x")  # SD
fullwordcount(cls, "x", 1)  # SD
```

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Output

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Output

```
> fullwordcount(cls,"x",1)
$a
[1] 2

$chuck
[1] 2

$Could
[1] 2

$How
[1] 1

$much
[1] 1

$wood
[1] 1

$woodchuck
[1] 2

$If
[1] 1

$ 'Wood?'
[1] 1
```

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. Hadoop Speed

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. Hadoop Speed

Disclaimer: No serious experiments done yet, just some small, very preliminary simulations.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. Hadoop Speed

Disclaimer: No serious experiments done yet, just some small, very preliminary simulations.

k-Means Clustering:

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. Hadoop Speed

Disclaimer: No serious experiments done yet, just some small, very preliminary simulations.

k-Means Clustering:

- Antonio, author of the R-Hadoop interface **rmr**, told me that the k-Means example is "just an example," not claimed to be fast.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. Hadoop Speed

Disclaimer: No serious experiments done yet, just some small,
very preliminary simulations.

k-Means Clustering:

- Antonio, author of the R-Hadoop interface **rmr**, told me
  that the k-Means example is "just an example," not
  claimed to be fast.

- Our simulations (though not for large n, only 10^5) indicate
  that Snowdoop is about 100X faster than Hadoop.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. Hadoop Speed

Disclaimer: No serious experiments done yet, just some small, very preliminary simulations.

k-Means Clustering:

- Antonio, author of the R-Hadoop interface **rmr**, told me that the k-Means example is "just an example," not claimed to be fast.

- Our simulations (though not for large n, only 10^5) indicate that Snowdoop is about 100X faster than Hadoop.

- Same simulations show Snowdoop gives about a 50% speedup over R's serial **kmeans()** function, with 5 nodes.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. Hadoop Speed

Disclaimer: No serious experiments done yet, just some small, very preliminary simulations.

k-Means Clustering:

- Antonio, author of the R-Hadoop interface **rmr**, told me that the k-Means example is "just an example," not claimed to be fast.

- Our simulations (though not for large n, only 10^5) indicate that Snowdoop is about 100X faster than Hadoop.

- Same simulations show Snowdoop gives about a 50% speedup over R's serial **kmeans()** function, with 5 nodes. Note: **kmeans()** is written in C, not R.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. Hadoop Speed

Disclaimer: No serious experiments done yet, just some small, very preliminary simulations.

k-Means Clustering:

- Antonio, author of the R-Hadoop interface **rmr**, told me that the k-Means example is "just an example," not claimed to be fast.

- Our simulations (though not for large n, only 10^5) indicate that Snowdoop is about 100X faster than Hadoop.

- Same simulations show Snowdoop gives about a 50% speedup over R's serial **kmeans()** function, with 5 nodes. Note: **kmeans()** is written in C, not R.

sorting:

- This should be Hadoop's forte'.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. Hadoop Speed

Disclaimer: No serious experiments done yet, just some small, very preliminary simulations.

k-Means Clustering:

- Antonio, author of the R-Hadoop interface **rmr**, told me that the k-Means example is "just an example," not claimed to be fast.

- Our simulations (though not for large n, only 10^5) indicate that Snowdoop is about 100X faster than Hadoop.

- Same simulations show Snowdoop gives about a 50% speedup over R's serial **kmeans()** function, with 5 nodes. Note: **kmeans()** is written in C, not R.

sorting:

- This should be Hadoop's forte'.
- Yet we are finding Snowdoop 2X faster.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. SparkR Speed

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. SparkR Speed

- Haven't done a comparison yet.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. SparkR Speed

- Haven't done a comparison yet.

- But SparkR has no right to be faster than Snowdoop for nonsort apps.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. SparkR Speed

- Haven't done a comparison yet.

- But SparkR has no right to be faster than Snowdoop for nonsort apps.

- None of the apps in MLlib, Spark's machine learning (aka statistics) library uses sorting.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. SparkR Speed

- Haven't done a comparison yet.

- But SparkR has no right to be faster than Snowdoop for nonsort apps.

- None of the apps in MLlib, Spark's machine learning (aka statistics) library uses sorting.

- A BARUG speaker from Spark said that they haven't compared timings to non-MapReduce platforms.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. SparkR Speed

- Haven't done a comparison yet.
- But SparkR has no right to be faster than Snowdoop for nonsort apps.
- None of the apps in MLlib, Spark's machine learning (aka statistics) library uses sorting.
- A BARUG speaker from Spark said that they haven't compared timings to non-MapReduce platforms.
- With both Hadoop and Spark, is it really a matter of "It's not important how well the dog could walk on his hind legs, but that he could do it at all"?

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Snowdoop vs. SparkR Speed

- Haven't done a comparison yet.
- But SparkR has no right to be faster than Snowdoop for nonsort apps.
- None of the apps in MLlib, Spark's machine learning (aka statistics) library uses sorting.
- A BARUG speaker from Spark said that they haven't compared timings to non-MapReduce platforms.
- With both Hadoop and Spark, is it really a matter of "It's not important how well the dog could walk on his hind legs, but that he could do it at all"?

Stay tuned.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Obtaining Snowdoop

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Obtaining Snowdoop

- Still under development.

Revisiting the
MapReduce
Paradigm: an
R-Specific
View

Norm Matloff
and Alex
Rumbaugh
University of
California at
Davis

Obtaining Snowdoop

- Still under development.

- In our **partools** package, on CRAN and GitHub.