A Parallel R Framework for Processing Large Dataset on Distributed Systems

Hao Lin
Shuo Yang
Samuel P. Midkiff

Nov. 17, 2013

This work is initiated and supported by Huawei Technologies

HUAWEI

PURDUE UNIVERSITY

Rise of Data-Intensive Analytics

Data Sources

Personal data for the internet

- query history
- click stream logging
- tweets
- …

Machine Generated Data

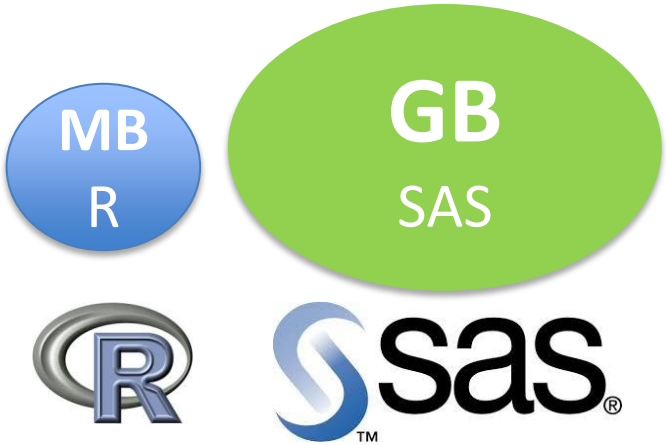
- Sensor networks
- Genome sequencing
- Physics experiments
- Satellite imaging data

HUAWEI PURDUE UNIVERSITY

New Challenges

Huge demand for data analysts, statisticians & data scientists

Traditional tools work on summary or sampled data



A good tool for large-scale data:

- Usability: stick to traditional semantics
- Performance: distributed parallelism
- Fault Tolerance: MapReduce

HUAWEI PURDUE UNIVERSITY

Usability

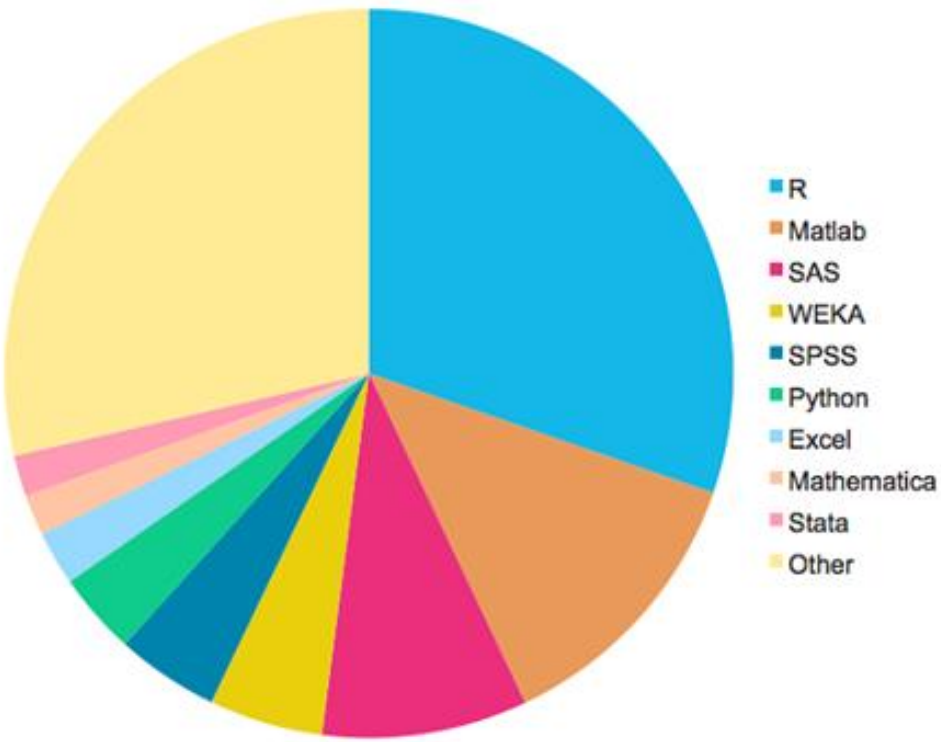
R used by about 30% of data analysts

Why R:

- Data structures
- Functional language
- Rich functionality
- Graphic visualization
- Open source

Why not R:

- Single threaded
- limited memory



From:
Survey by Revolution Analytics
http://r4stats.com/articles/popularity/

Performance - Spark Framework

Distributed Computing with Fault-tolerance

Developed at AMP lab, UC Berkeley

Flexible Programming Model

- DAG job scheduler

Performance

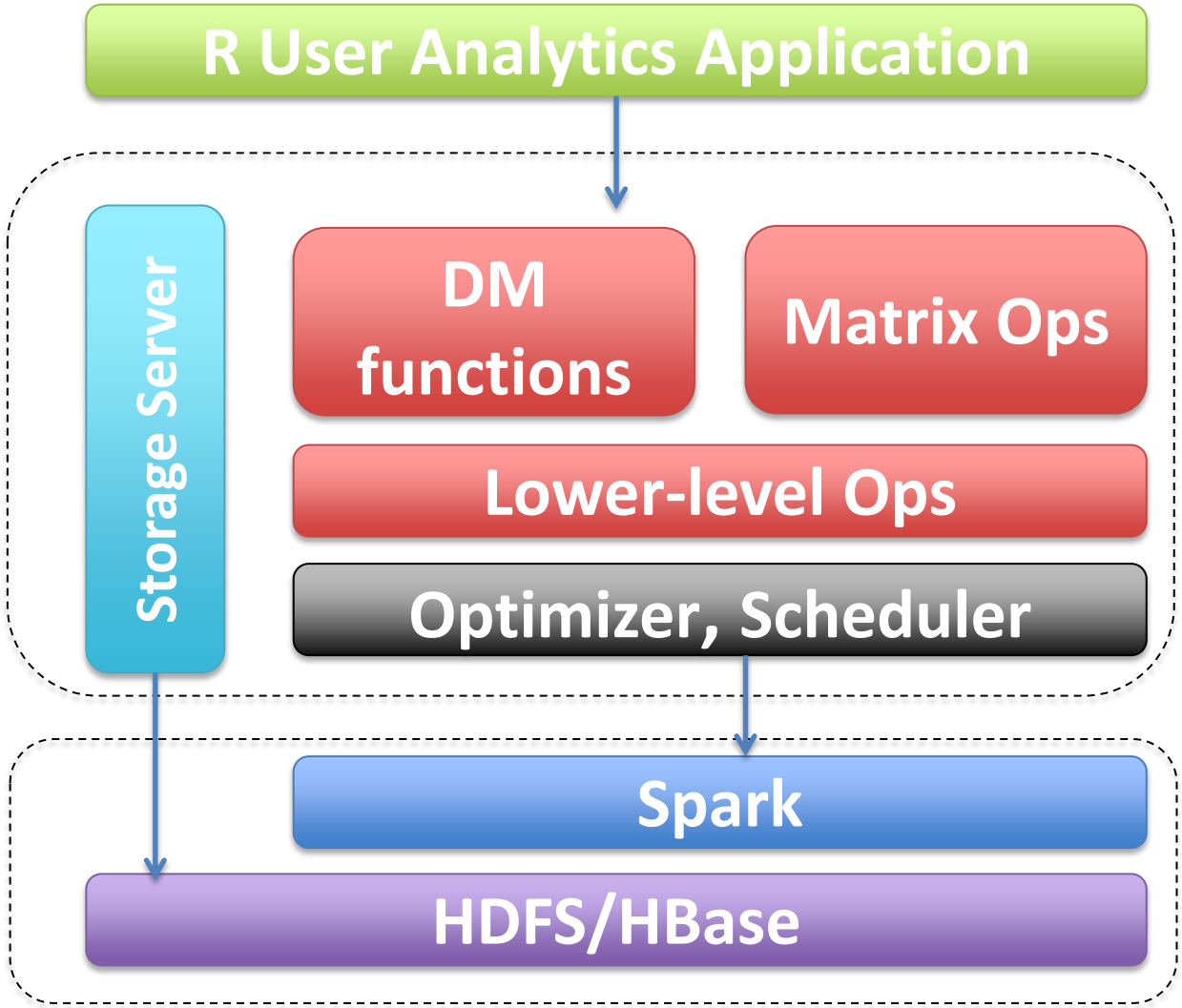
- In-memory
- Good for iterative algorithms

Resilient Data Sets

- Recover from loss and failures

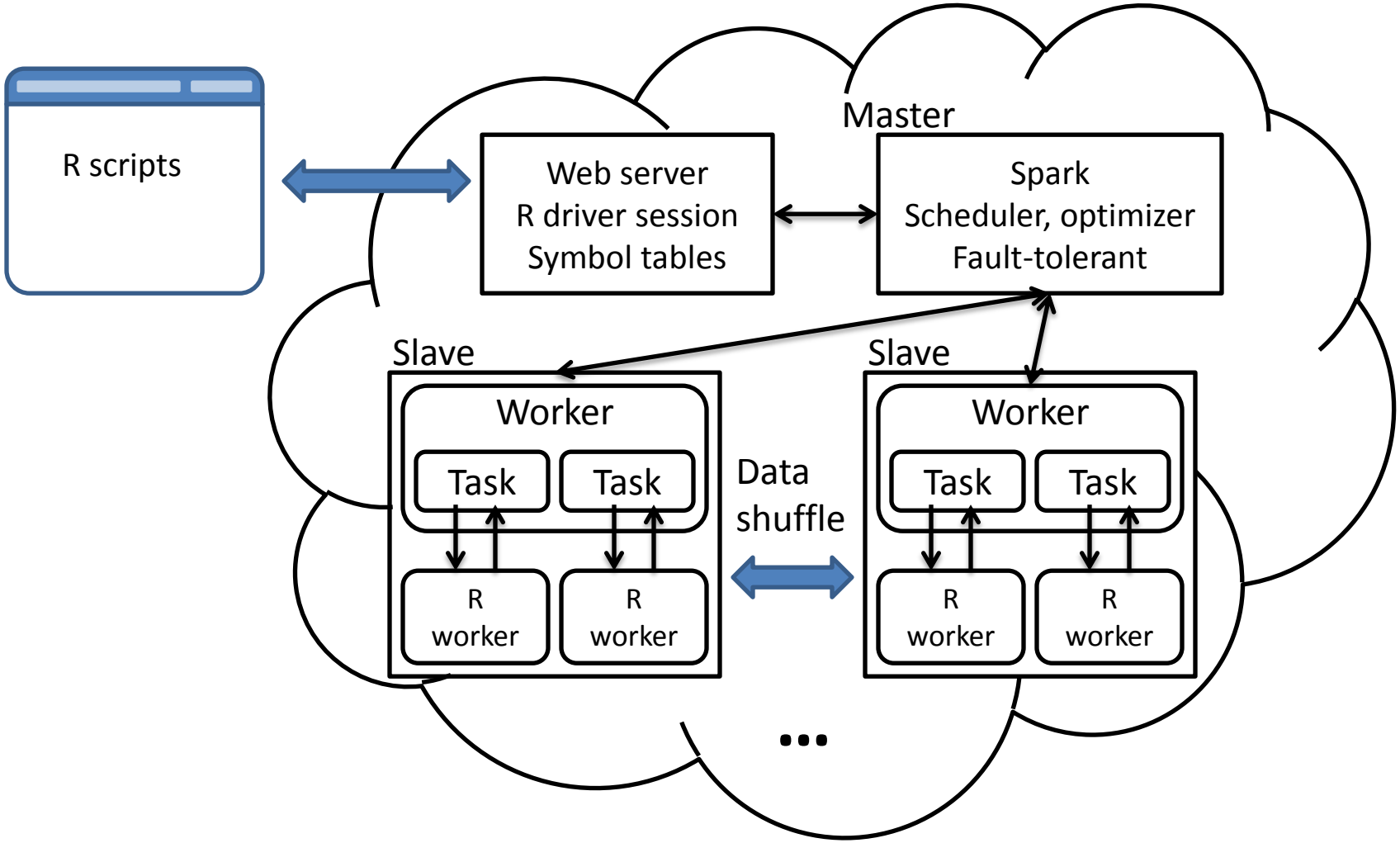
HUAWEI PURDUE UNIVERSITY

RABID Package Structure



RABID as an R extension package

Runtime Overview



Programming Model

R List – Most general data structure

- Collection to store elements of any & different types
- similar to Python tuples
- Very general and flexible

BigList – distributed list structure

- Extended R List to be distributed
- Override R list functions to support BigList
- Building blocks for higher level structures and functions

HUAWEI PURDUE UNIVERSITY

RABID Example (1)

Sample APIs

```
library("Rabid")
```

Load the Rabid package

```
DATA <- rb.readLines("hdfs://…")
```

Read text into a BigList

```
DATA <- lapply(DATA, as.numeric, cache=TRUE)
```

Apply function in parallel

```
func <- function(a) { ... }
DATA2 <- lapply(DATA, func)
```

Apply UDF in parallel

```
DATA3 <- aggregate(DATA2, by=id, FUN=mean)
```

Aggregate by user specified keys

```
centroids <- as.list(sample(DATA, 16))
```

Transform back to R list

HUAWEI PURDUE UNIVERSITY

RABID Example (2)

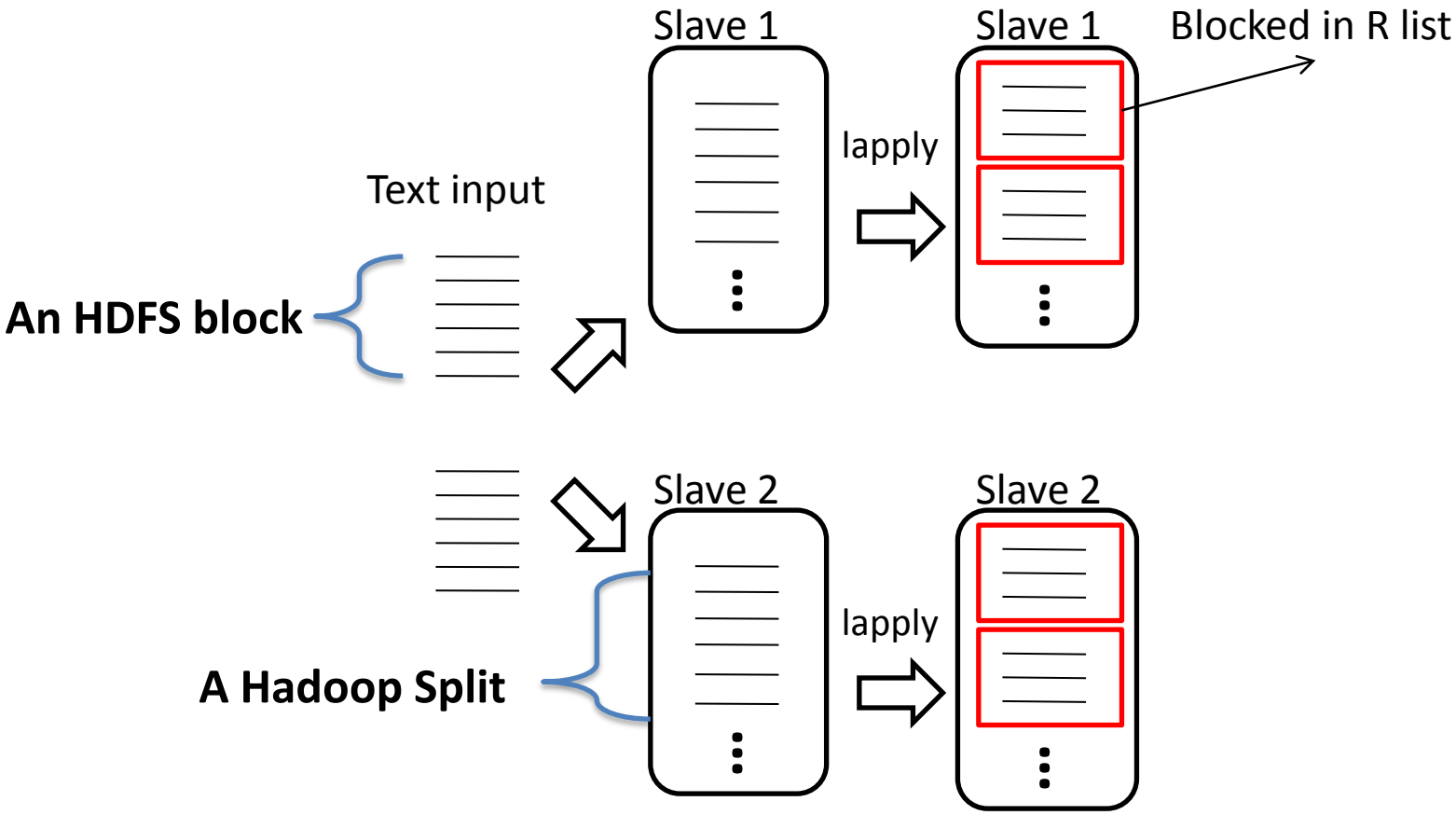
Sample APIs

```
library("Rabid")

mat <- rb.read.matrix("hdfs://…", cache=F)

mat1 <- mat + mat
mat2 <- mat %*% mat
t(mat)
cor(mat)
```

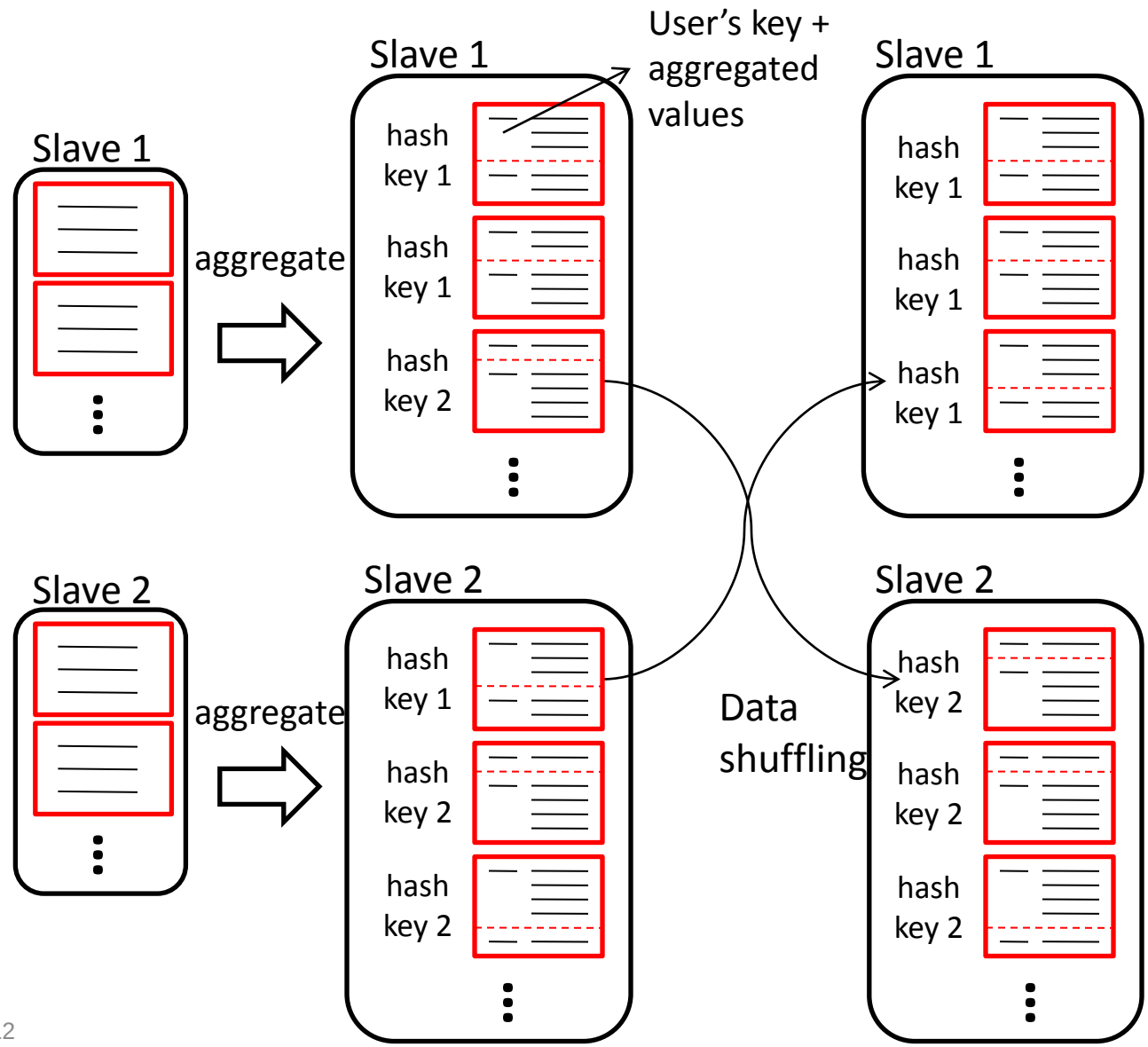

HUAWEI PURDUE UNIVERSITY

Data Blocking for lapply()

Further block the data for better efficiency of transferring and processing



Data Blocking for aggregate()



Distributing Computation

Computations are abstracted as R functions, which are serialized to the nodes and evaluated

R has a scoping rule for searching free variables in an enclosing environment

We need to ship the functions together with the values of free variables in its environment

```
z <- 1
func1 <- function() {
    y <- 2
    func2 <- function(x) {
        x + y + z
    }
}
```


HUAWEI PURDUE UNIVERSITY

Merging Deferred Operations

Two major overheads of each RABID operation:

1) data transferring

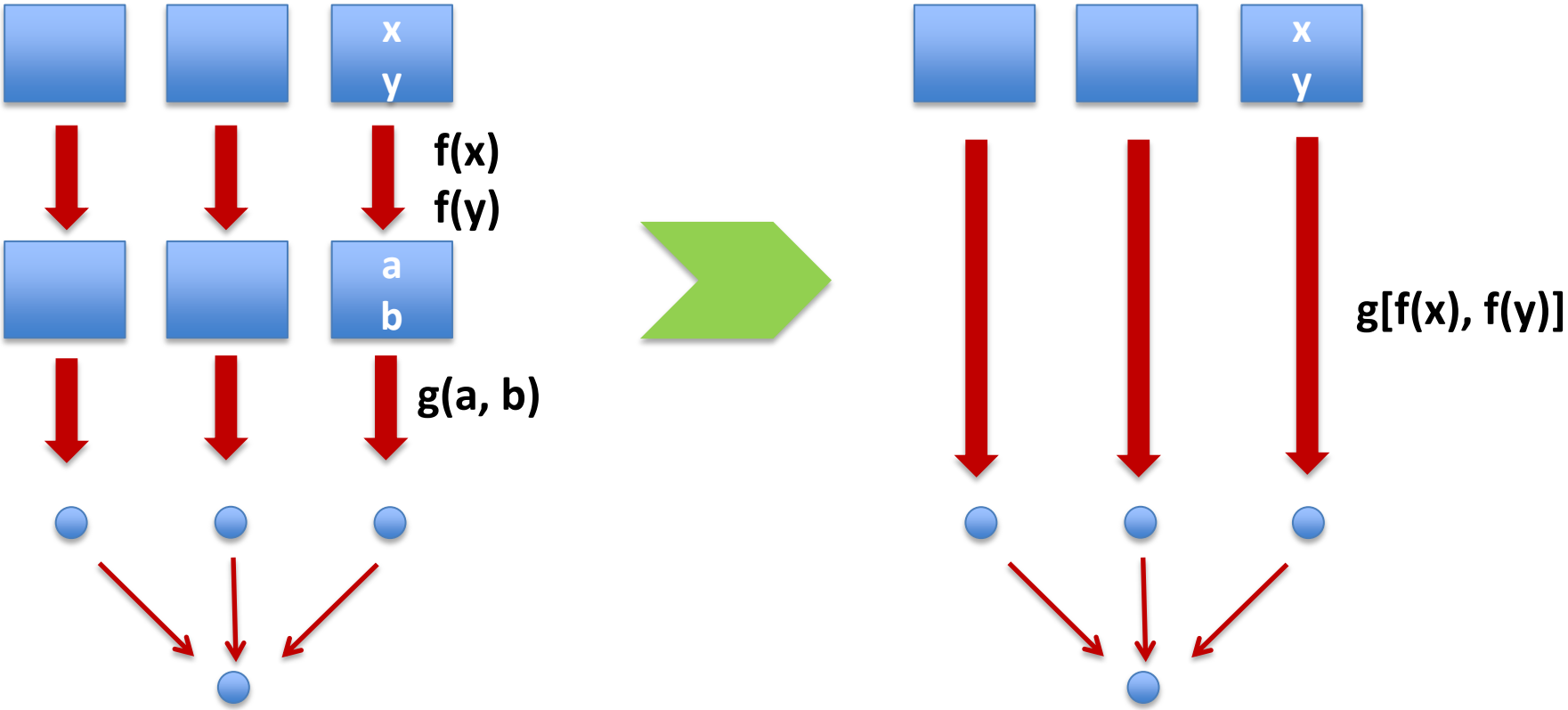
2) serialization/deserialization

Merge adjacent deferred operations into one reduces the overheads

What kind of operations can be merged: non-aggregation

HUAWEI PURDUE UNIVERSITY

Merging Deferred Operations



Fault Tolerance

- Take the advantage of Spark's fault tolerance feature at the worker side
- Detect user code errors that terminate R worker sessions; catch the error and stop Spark job immediately
- Zookeeper at the master side

HUAWEI PURDUE UNIVERSITY

Applications & Benchmarking

Compare RABID with Hadoop & RHIPE

- RHIPE: R and Hadoop Integrated Programming Environment, developed at Purdue University

Logistic Regression:

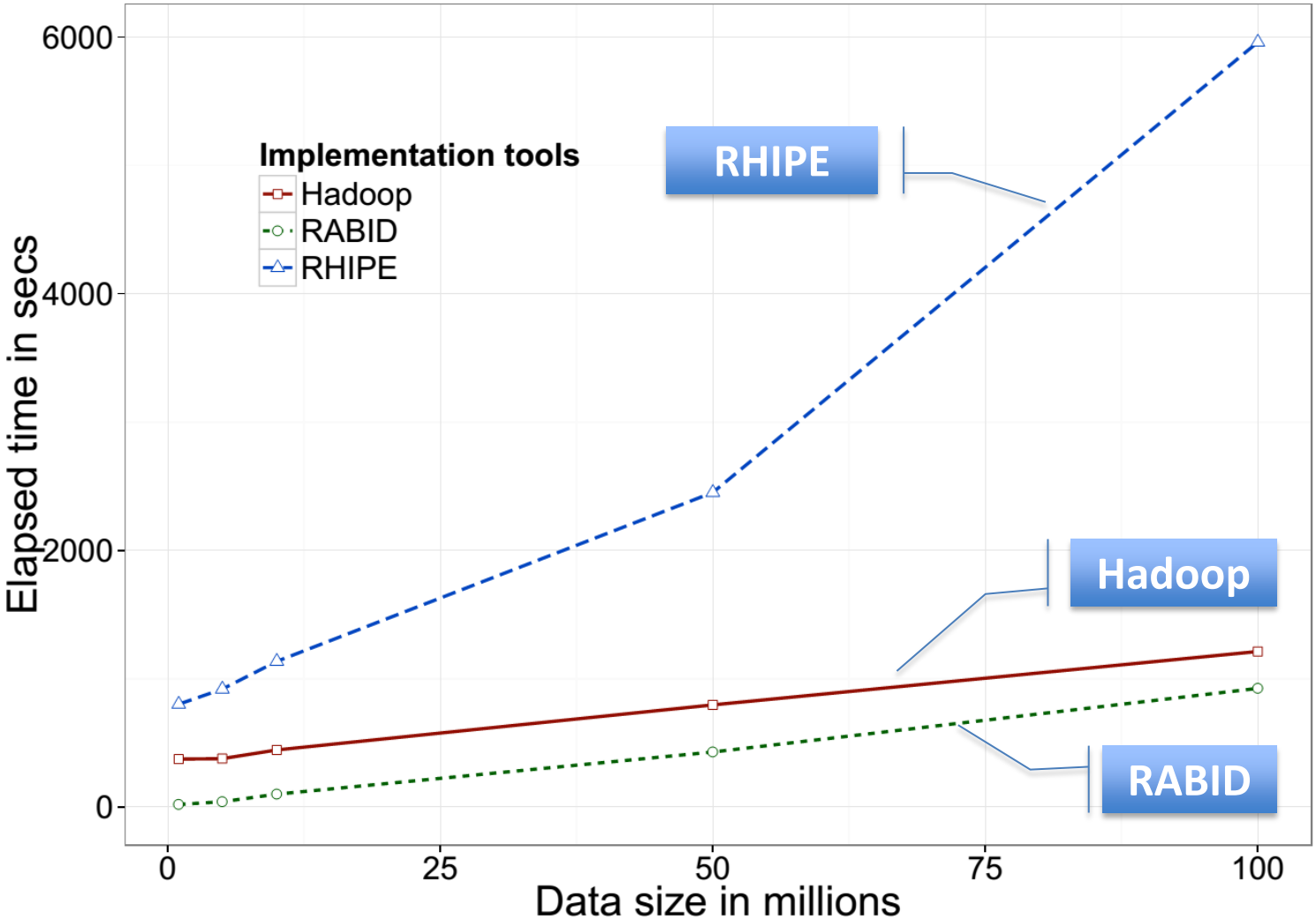
- 10 worker nodes
- 1 ~ 100 million records
- RABID uses 1/6 LOC of Hadoop

Movie Clustering (K-Means):

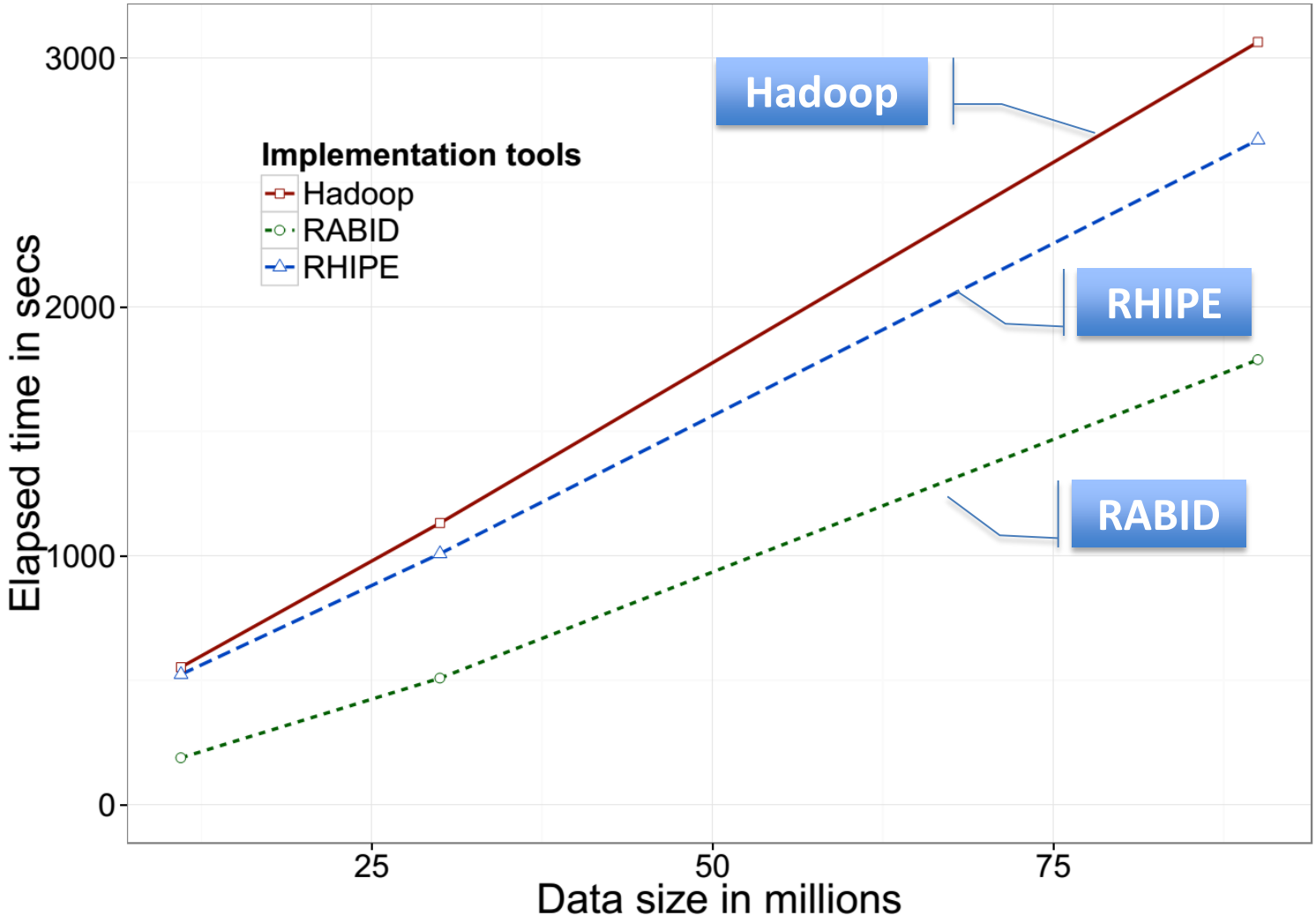
- 10 worker nodes
- 11 ~ 90 million ratings
- RABID uses 1/8 LOC of Hadoop

HUAWEI PURDUE UNIVERSITY

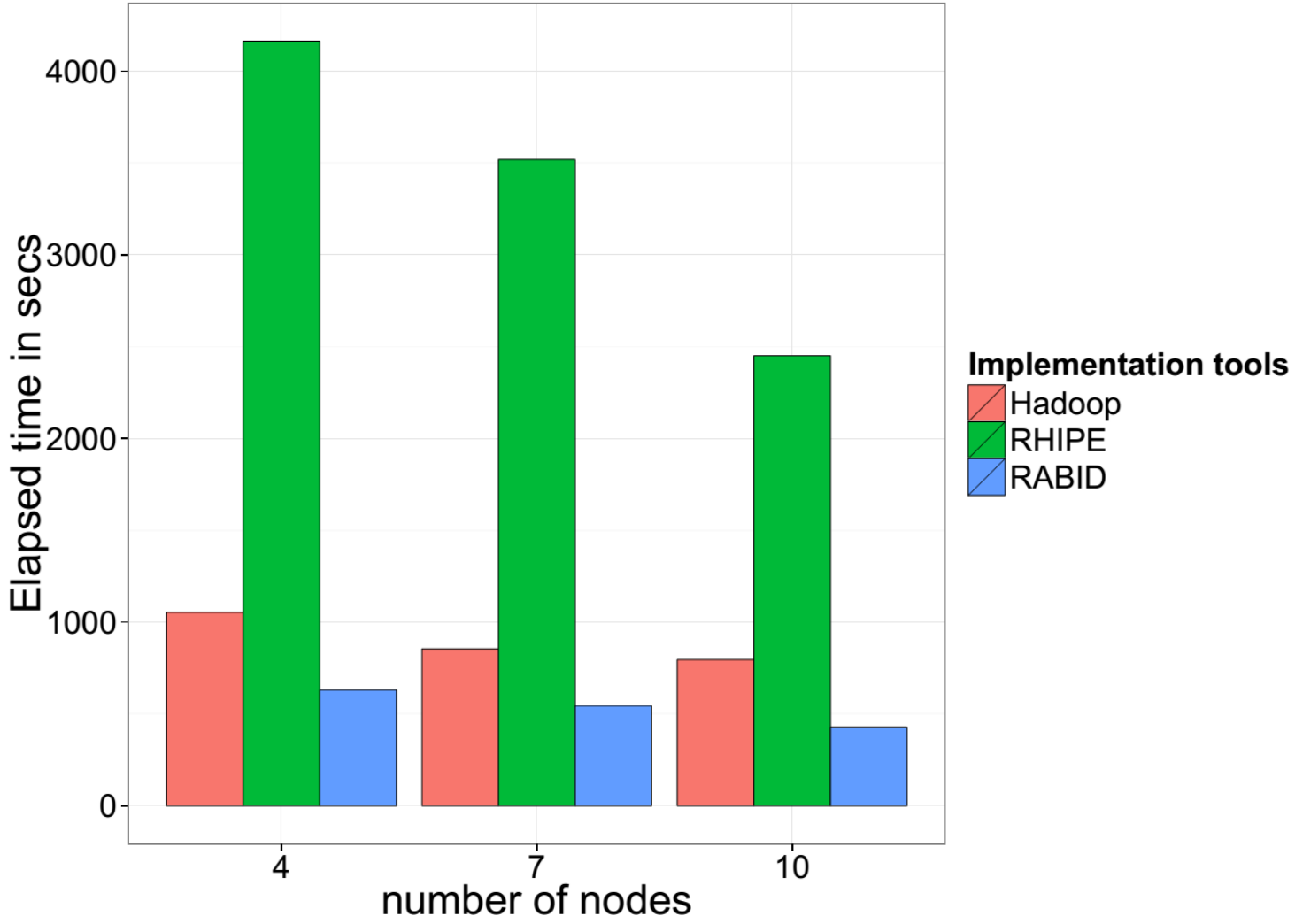
Logistic Regression Runtime over Data Size



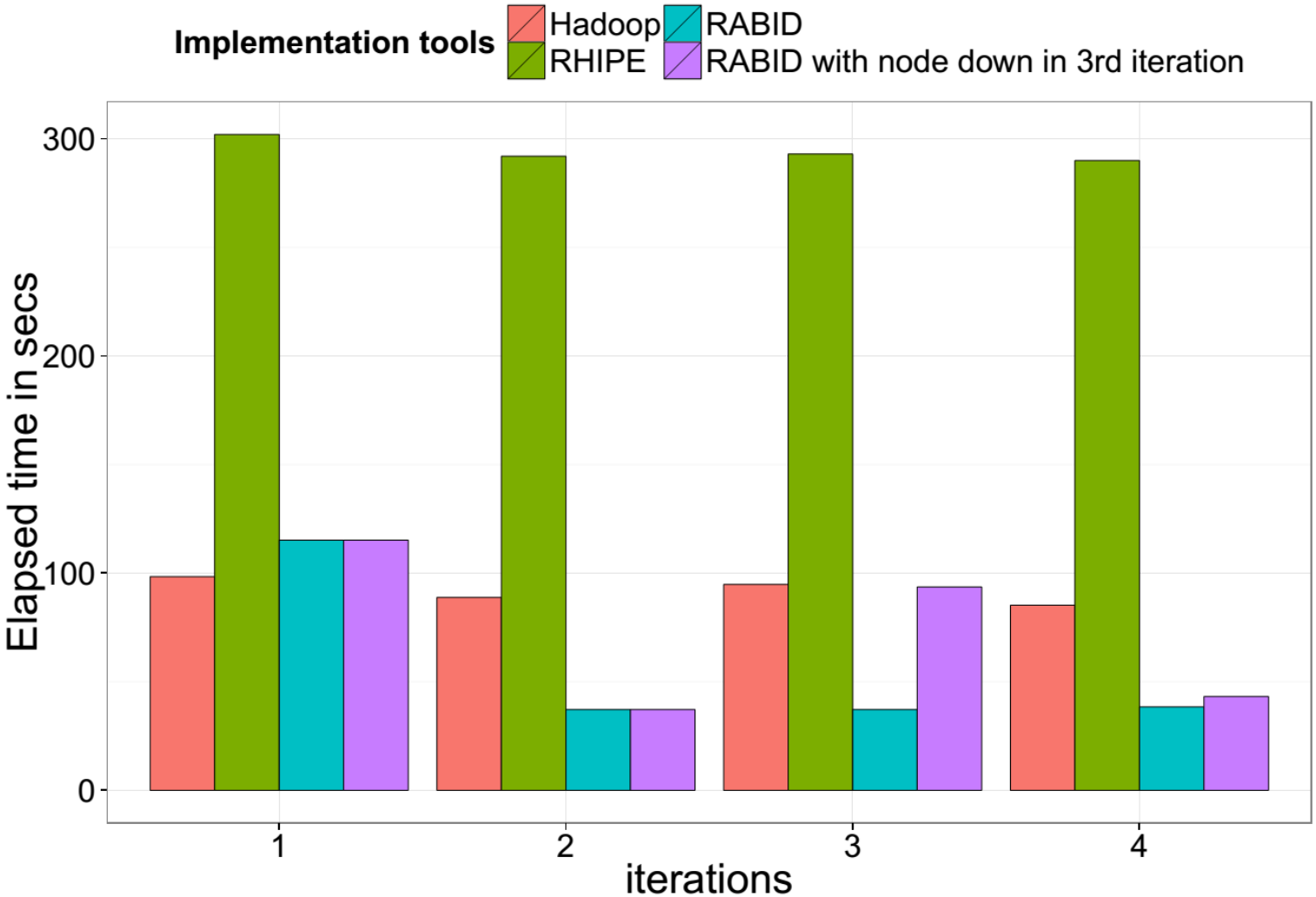
K-Means Movie Clustering Runtime



Logistic Regression Runtime over # nodes



Logistic Regression Runtime on iterations



Conclusions

- RABID provides R users with a familiar programming model that scales to large cloud based clusters, allowing larger problems sizes to be efficiently solved.
- Preliminary results show RABID outperforms Hadoop and RHIPE on our benchmarks
- RABID is cloud-ready to be used as a service

HUAWEI PURDUE UNIVERSITY

Future Work

- Optimizing data transferring between R session and Spark
- Trade-off between fault tolerance and performance
- Benchmarking more applications and at a larger scale

HUAWEI PURDUE UNIVERSITY

Thank You!

Also thank Prof. Michael Franklin and Matei Zaharia at UC Berkeley, for discussing the ideas over Spark project

Contact: haolin@purdue.edu
shuo.yang@huawei.com
smidkiff@purdue.edu

Nov 17, 2013

HUAWEI

PURDUE UNIVERSITY