CHRYSAOR: FINE-GRAINED, FAULT-TOLERANT MAPREDUCE

Pedro Sá da Costa
Fernando M. V. Ramos, Miguel Correia

Navigators

LaSIGE

inesc id lisboa

Ciências ULisboa

Motivation

- MapReduce is a tool for **processing massive datasets**

Motivation

- MapReduce is a tool for **processing massive datasets**
- Programming framework used by Google, or Yahoo (most likely here too)

Motivation

- MapReduce is a tool for **processing massive datasets**
- Programming framework used by Google, or Yahoo (most likely here too)
- Targeted a **single data center** and tolerates **crash faults**

Motivation

- MapReduce is a tool for **processing massive datasets**
- Programming framework used by Google, or Yahoo (most likely here too)
- Targeted a **single data center** and tolerates **crash faults**
- The **unprecedented data growth** brought the need to **scale-out computation** across clouds

Motivation

NEWS

Google: DRAM error rates vastly higher than previously thought

PCs will likely require error correction code in the future due to DRAM issues

By Lucas Mearian FOLLOW

Computerworld | Oct 8, 2009 4:51 PM PT

RELATED TOPICS

Data Center

Servers Google

A study released this week by Google Inc. and the University of Toronto showed that data error rates on dynamic RAM memory modules are vastly higher than previously thought and may be more responsible for system shutdowns and service interruptions.

MORE LIKE THIS

Tips for using RAS features for storage systems

Chapter 1: Network Overview

Report: Intel's Braidwood flash memory module could kill SSD market

Motivation

GAWKER

GCreep: Google Engineer Stalked Teens, Spied on Chats (Updated)

Adrian Chen

09/14/10 02:26PM Filed to: EXCLUSIVE

581.25K

We entrust Google with our most private communications because we assume the company takes every precaution to safeguard our data. It doesn't. A Google engineer spied on four underage teens for months before the company was notified of the abuses.

David Barksdale, a 27-year-old former Google engineer, repeatedly took advantage

Motivation

Internet outage disrupts Twitter and Spotify, reportedly due to cyber attack

LAUREN HOLTER @laurenholter | 10.21.16 | 10:34 am

EMAIL SHARE TWEET PIN IT REDDIT

Motivation

What are the advantages of using multiple clouds?

- It **increases resilience** by avoiding single points of failure.

Motivation

What are the advantages of using multiple clouds?

- It **increases resilience** by avoiding single points of failure.
- It can **improve performance** (data locality, computing and network diversity)

Motivation

What are the advantages of using multiple clouds?

- It **increases resilience** by avoiding single points of failure.
- It can **improve performance** (data locality, computing and network diversity)
- It can **improve security**, for instance by exploring the interaction between public and private clouds.

Motivation

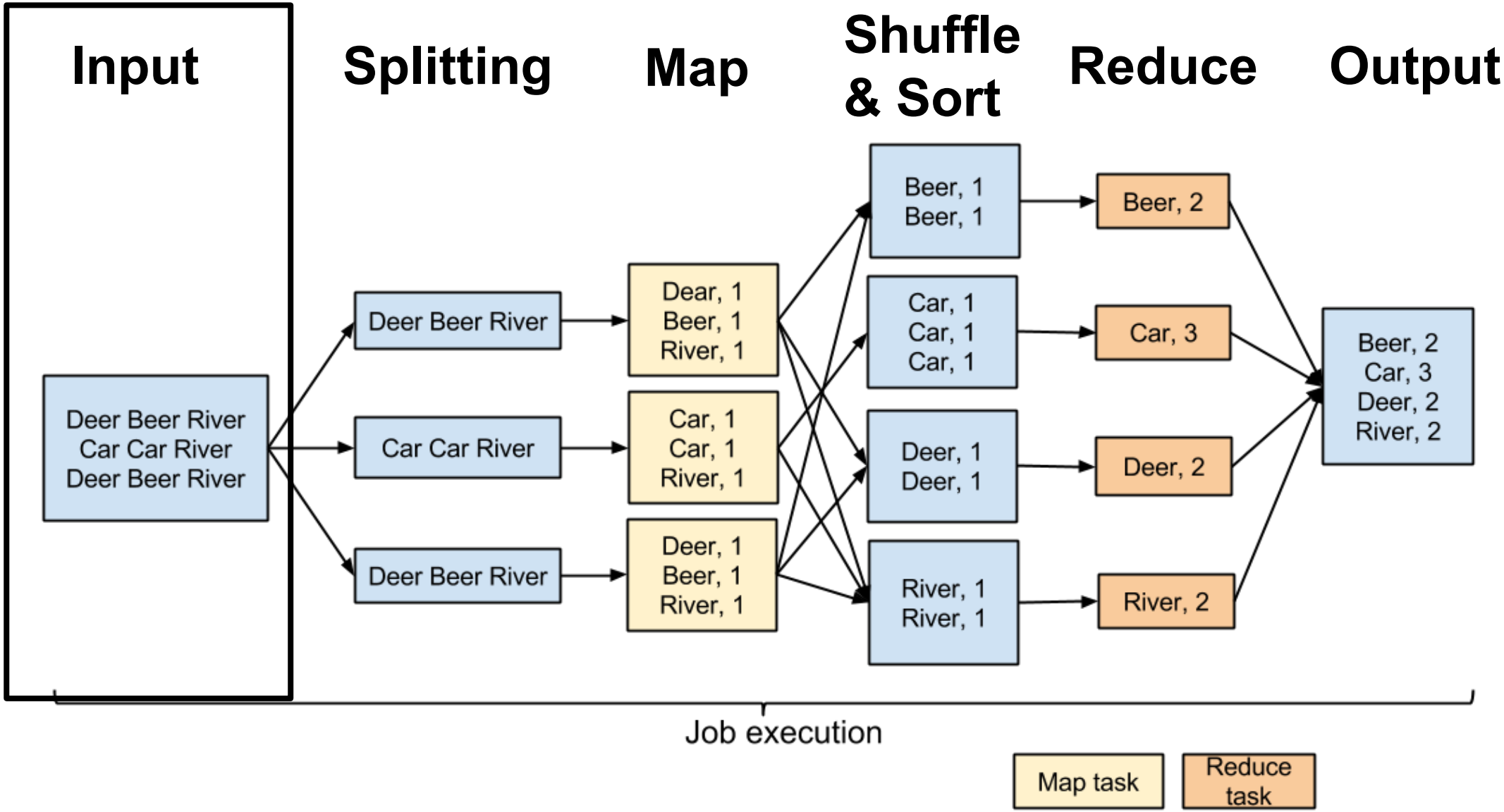
What are the advantages of using multiple clouds?

- It **increases resilience** by avoiding single points of failure.
- It can **improve performance** (data locality, computing and network diversity)
- It can **improve security**, for instance by exploring the interaction between public and private clouds.
- It may help in **reducing costs**

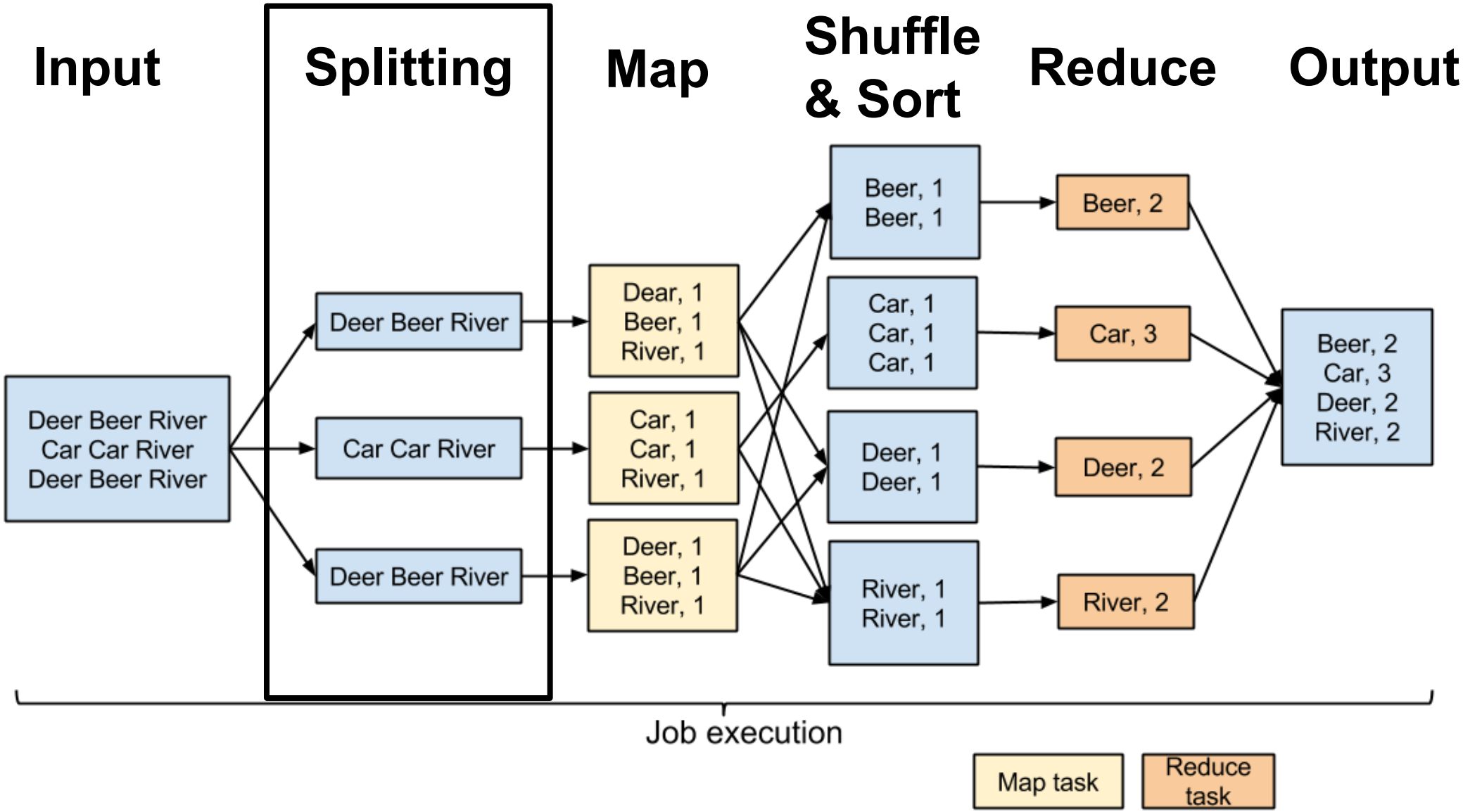
Outline

- **MapReduce**
- Contribution
- MapReduce Execution
  - MapReduce Job
  - Logical MapReduce Jobs
  - WordCount Example
- Evaluation
  - Experimental Setup
  - Experimental Results
- Conclusions

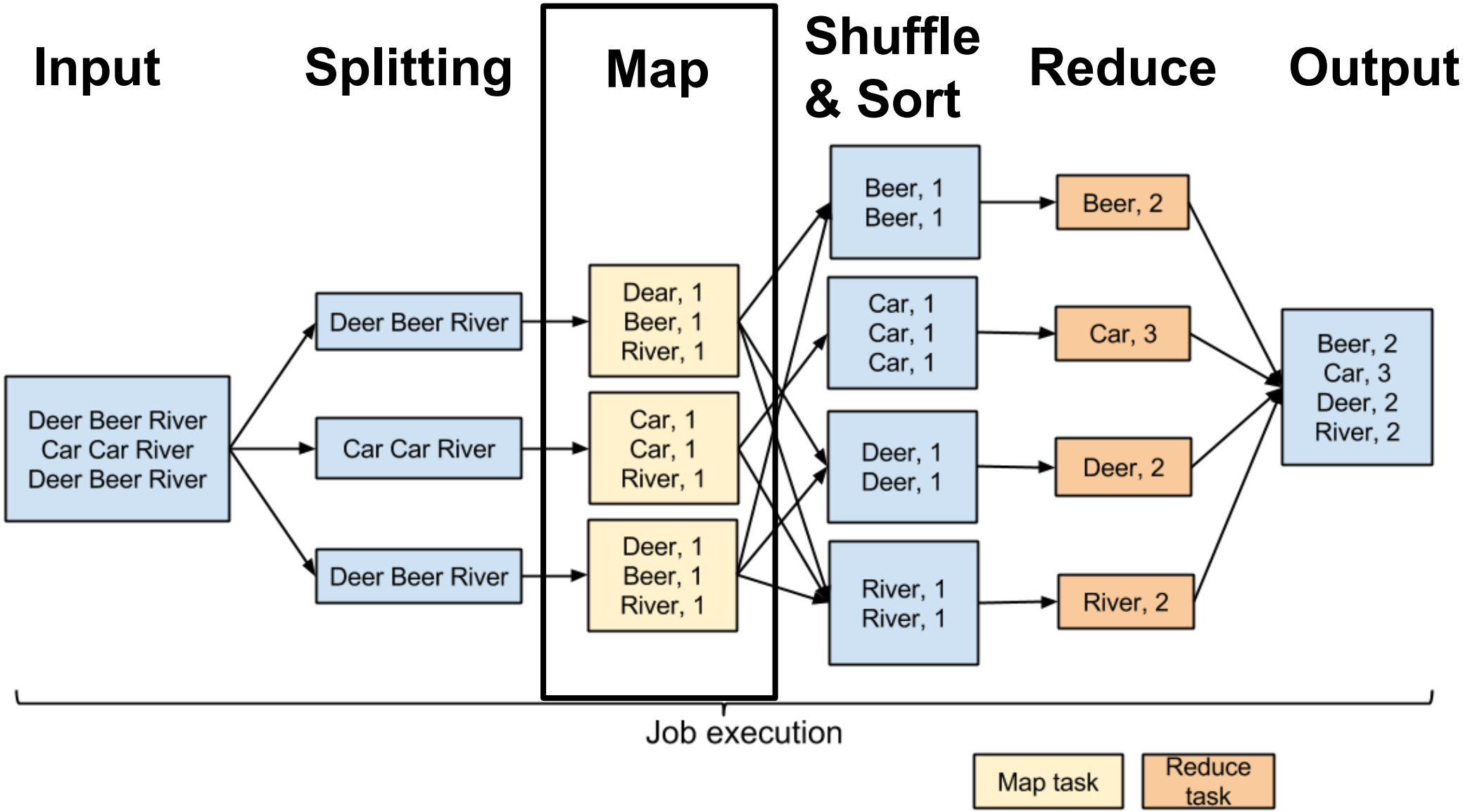
Hadoop MapReduce



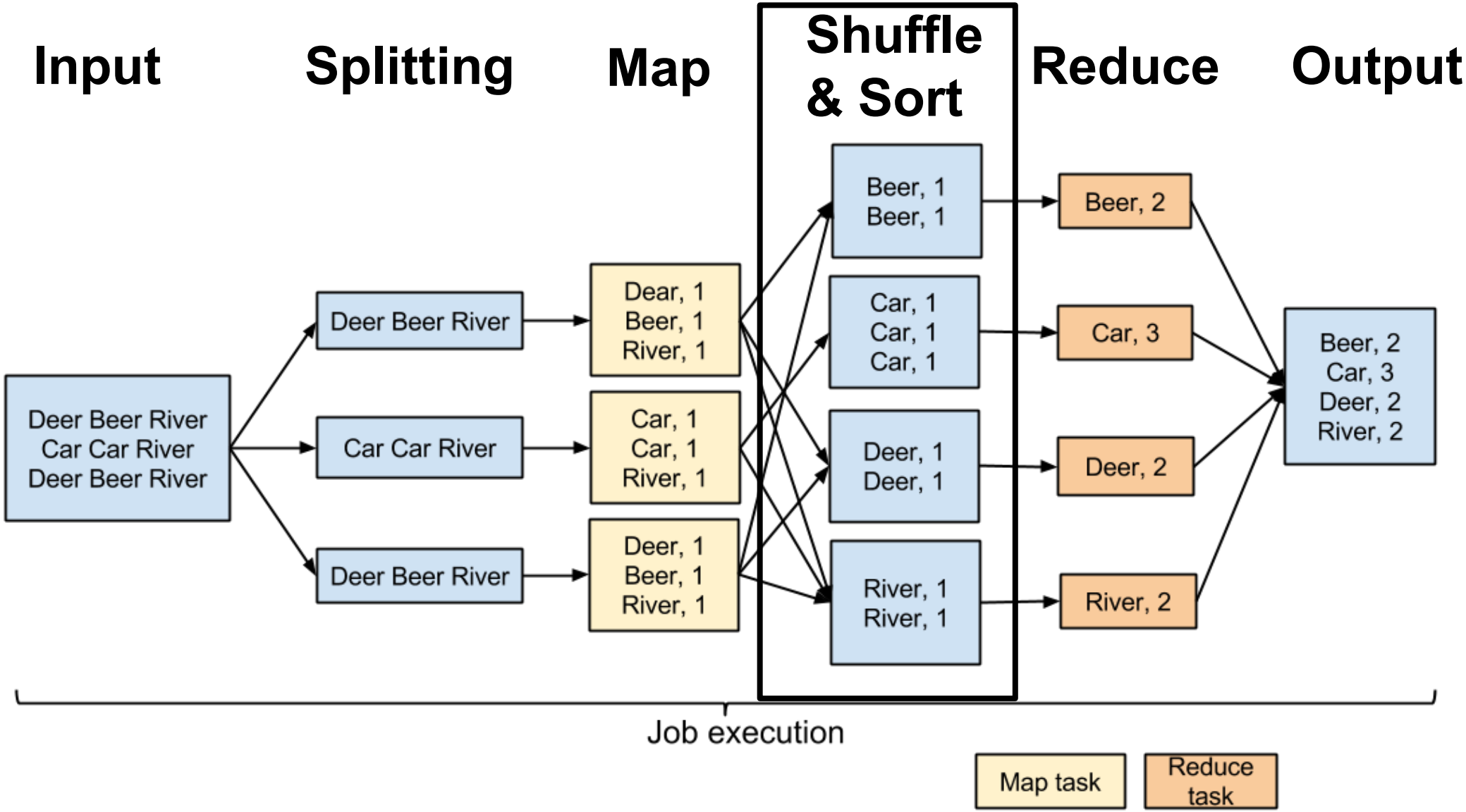
Hadoop MapReduce



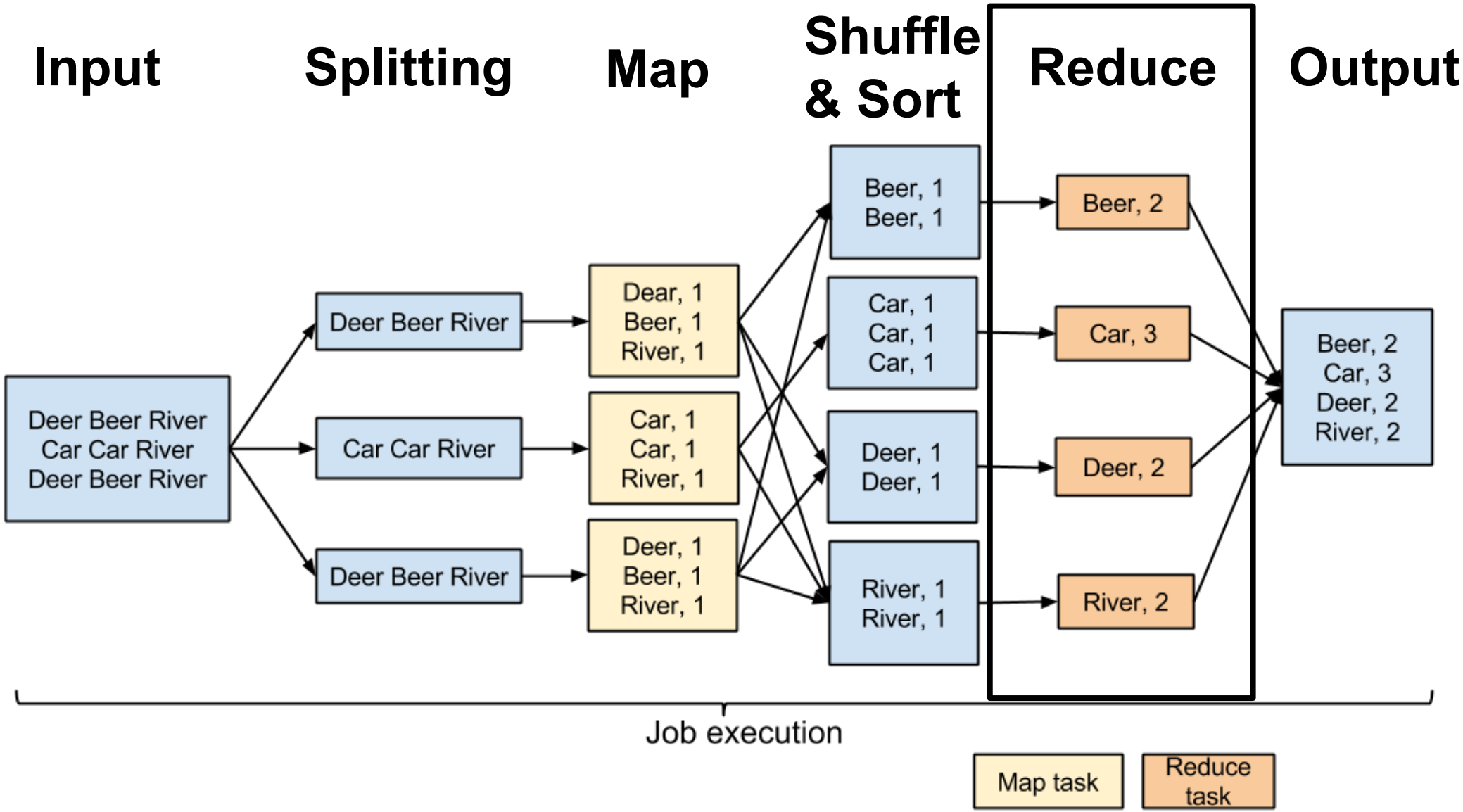
Hadoop MapReduce



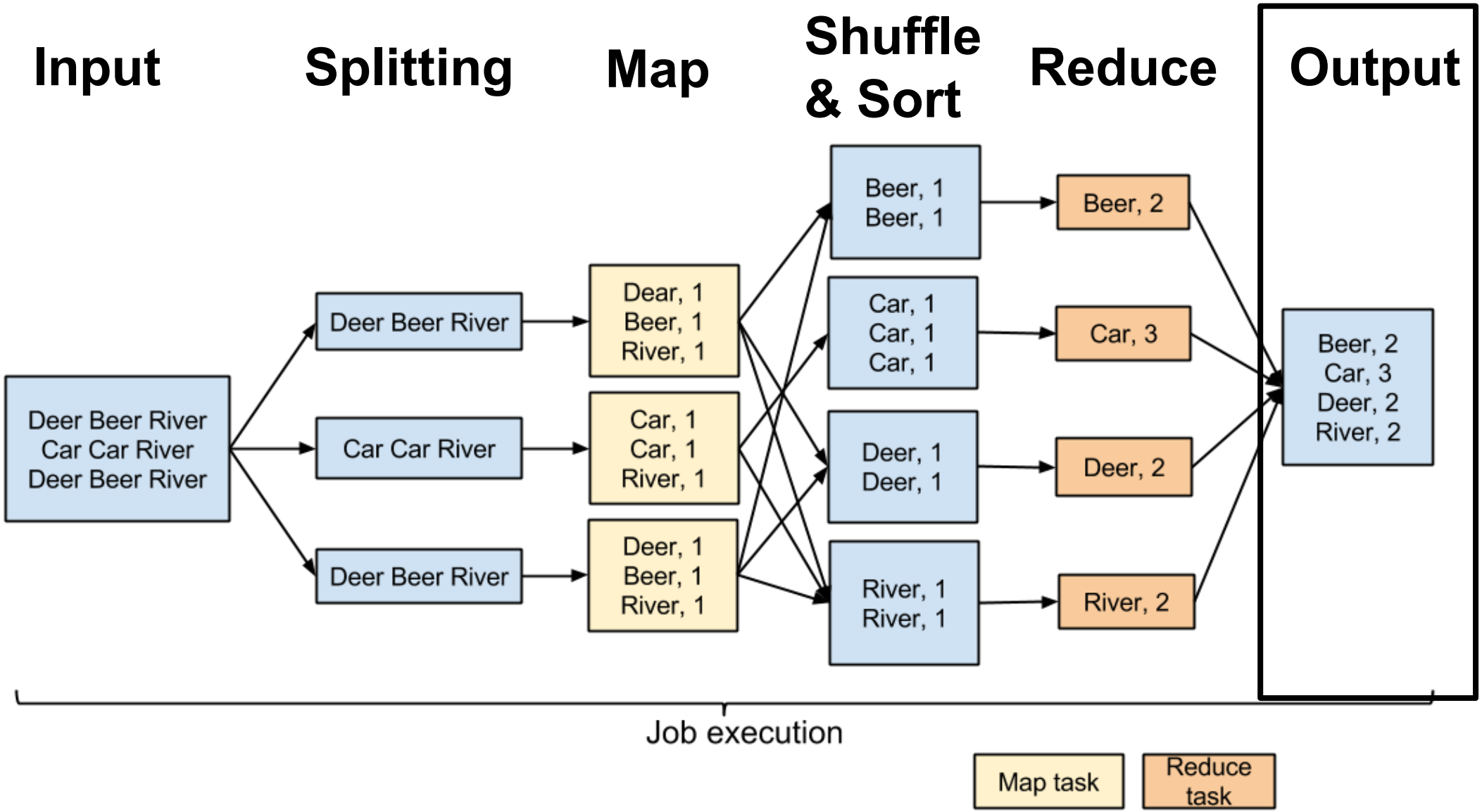
Hadoop MapReduce



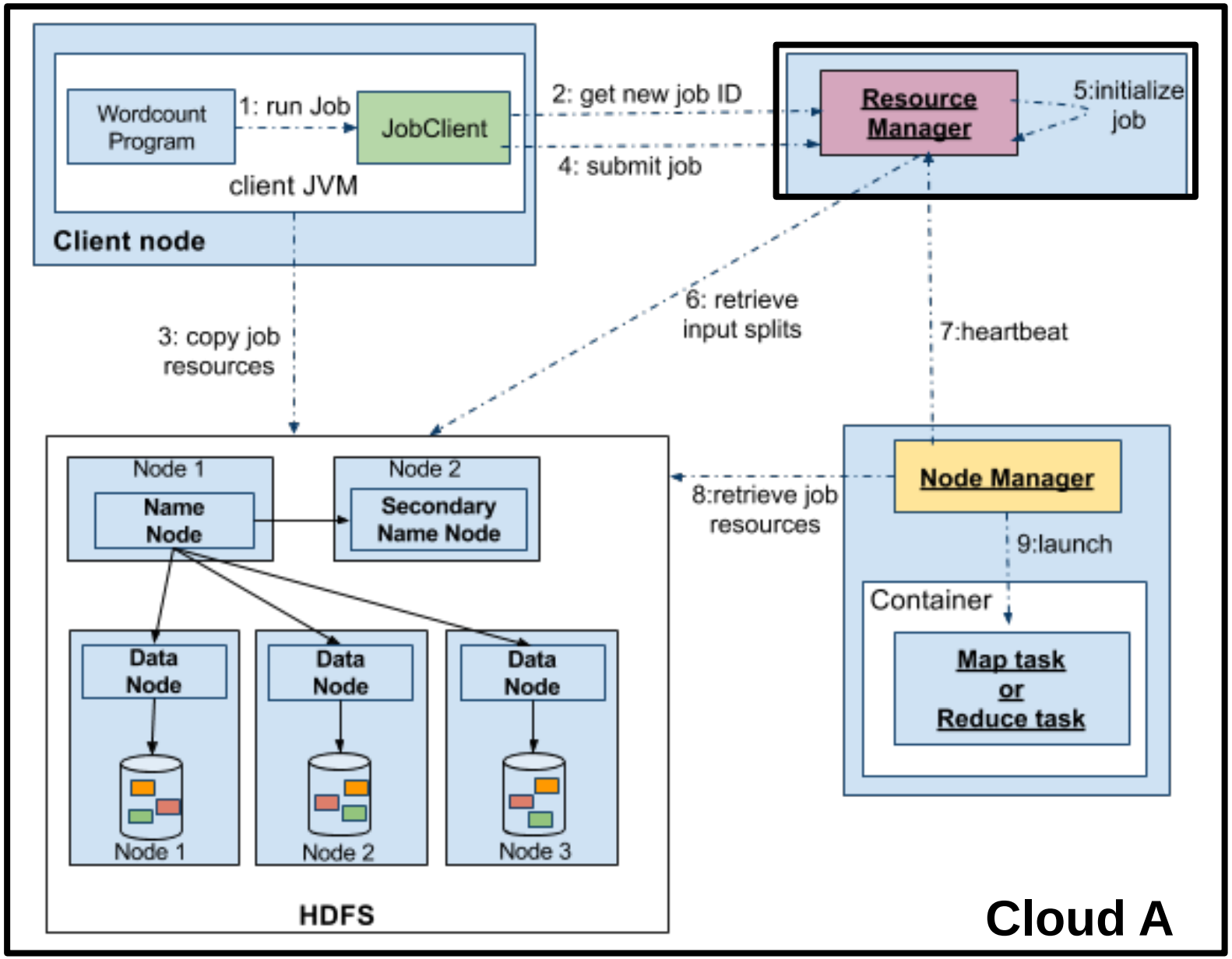
Hadoop MapReduce



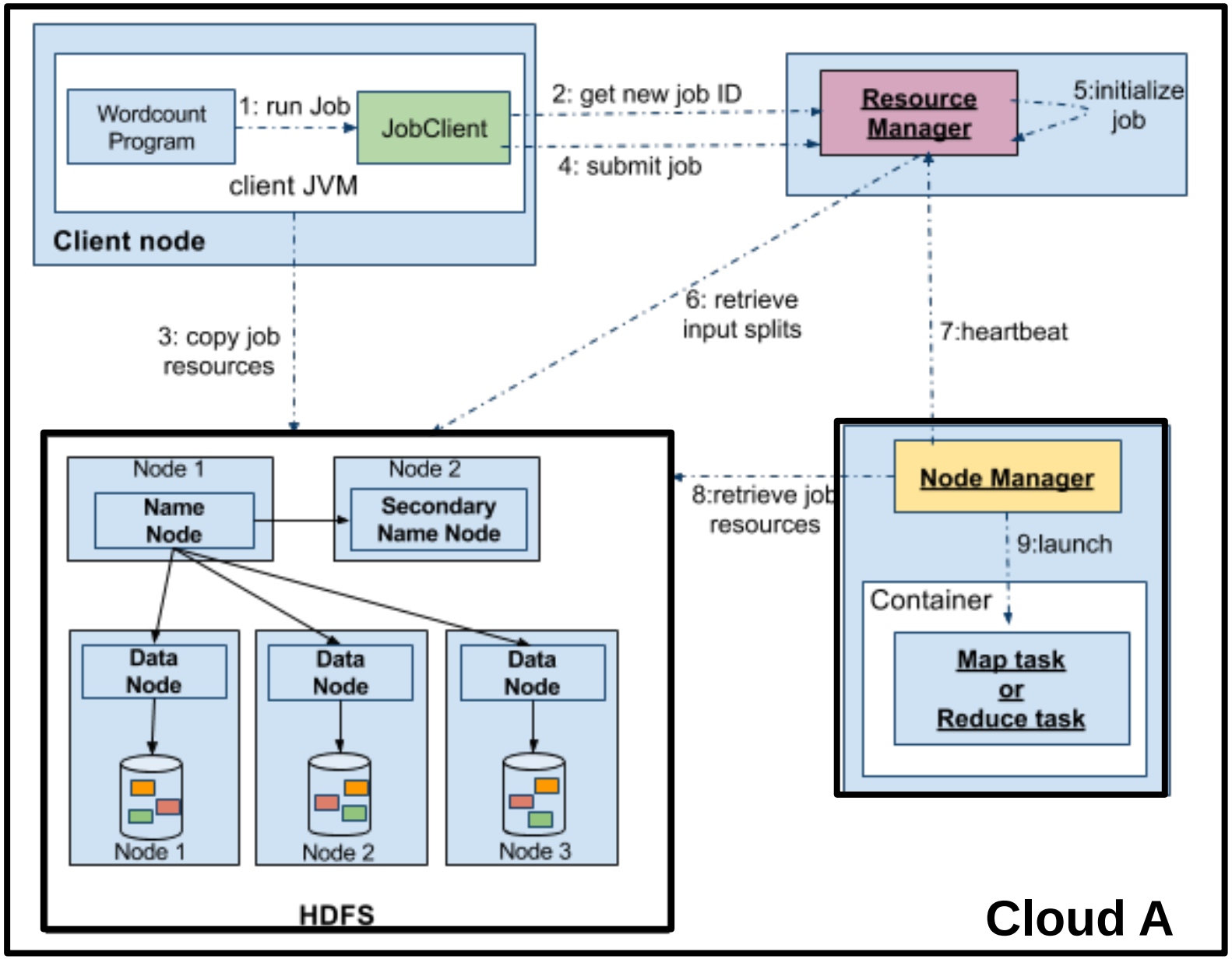
Hadoop MapReduce



Hadoop MapReduce



Hadoop MapReduce



Hadoop MapReduce

- Tolerates **crash faults** by re-launching tasks

Hadoop MapReduce

- Tolerates **crash faults** by re-launching tasks
- Add **checksums to detect data corruptions**

Hadoop MapReduce

- Tolerates **crash faults** by re-launching tasks
- Add **checksums to detect data corruptions**
- Vulnerable to **cloud outage**, **arbitrary** and **malicious faults**

Outline

- MapReduce
- **Contribution**
- MapReduce Execution
  - MapReduce Job
  - Logical MapReduce Jobs
  - WordCount Example
- Evaluation
  - Experimental Setup
  - Experimental Results
- Conclusions

Contribution

- Execute **replicated jobs** in different clouds

Contribution

- Execute **replicated jobs** in different clouds
- Tolerate **faults at the task level** and relaunch the faulty tasks.

Contribution

- Execute **replicated jobs** in different clouds
- Tolerate **faults at the task level** and relaunch the faulty tasks.
- Use **digests in the tasks** to validate computation

Contribution

What do we want?

- A system that **tolerates the above-mentioned classes** of faults at reasonable cost

Contribution

What do we want?

- A system that **tolerates the above-mentioned classes** of faults at reasonable cost
- That requires **minimal modifications** to the users' applications

Contribution

What do we want?

- A system that **tolerates the above-mentioned classes** of faults at reasonable cost
- That requires **minimal modifications** to the users' applications
- and it **does not change** Hadoop source code

Outline

- Contribution
- **MapReduce Execution**
  - MapReduce Job
  - Logical MapReduce Jobs
  - WordCount Example
- Evaluation
  - Experimental Setup
  - Experimental Results
- Conclusions

Logical Job

- Cannot **pause** a job with MapReduce API

Logical Job

- Cannot **pause** a job with MapReduce API
- One logical job will be **dedicated to the map tasks**, and the other to the **reduce tasks**

Logical Job

- Cannot **pause** a job with MapReduce API
- One logical job will be **dedicated to the map tasks**, and the other to the **reduce tasks**
- Create the concept of **Identity map tasks**

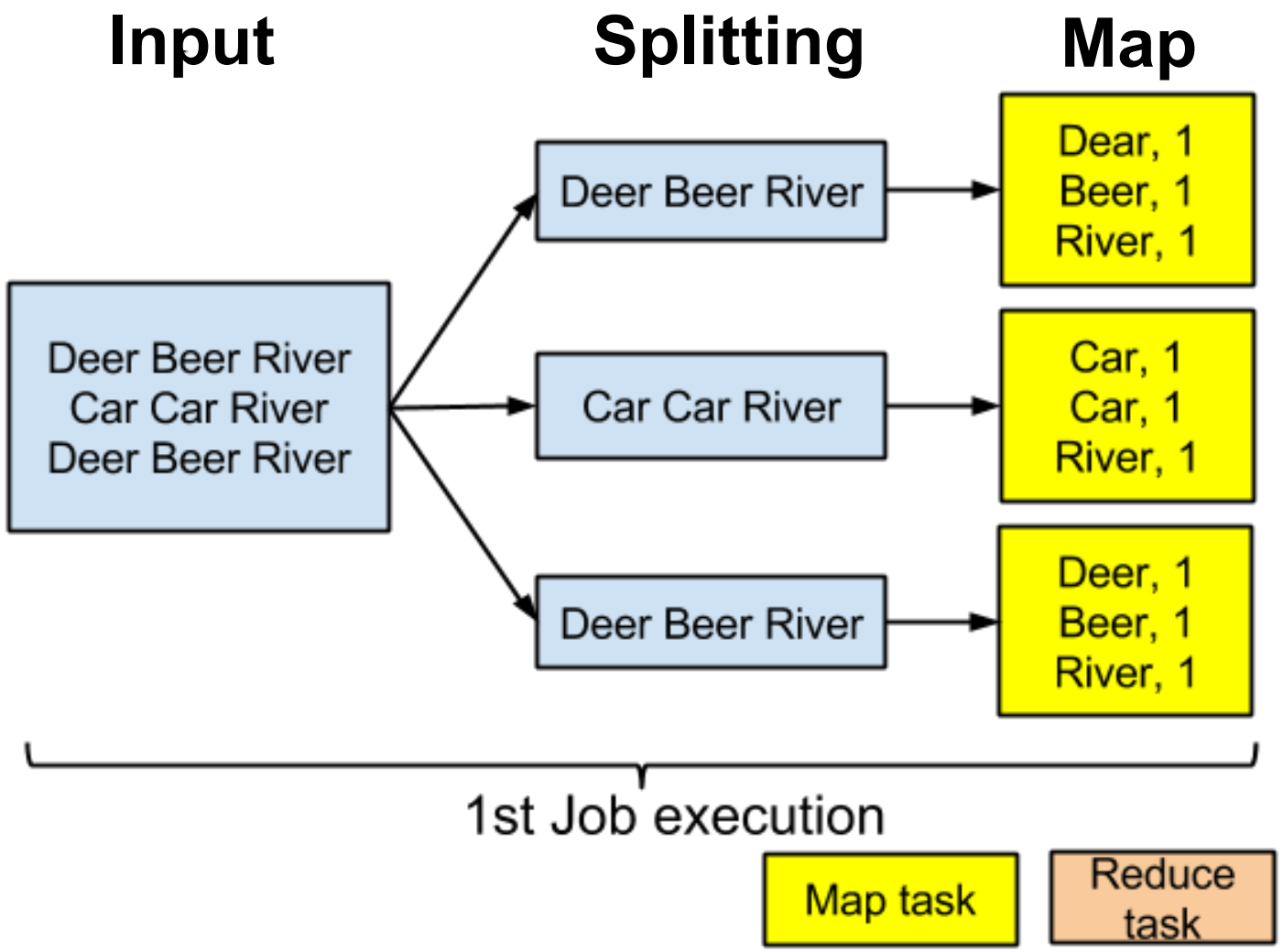
Logical Job

- Cannot **pause** a job with MapReduce API
- One logical job will be **dedicated to the map tasks**, and the other to the **reduce tasks**
- Create the concept of **Identity map tasks**
- **Hadoop viewpoint**, each logical job is a complete MapReduce job

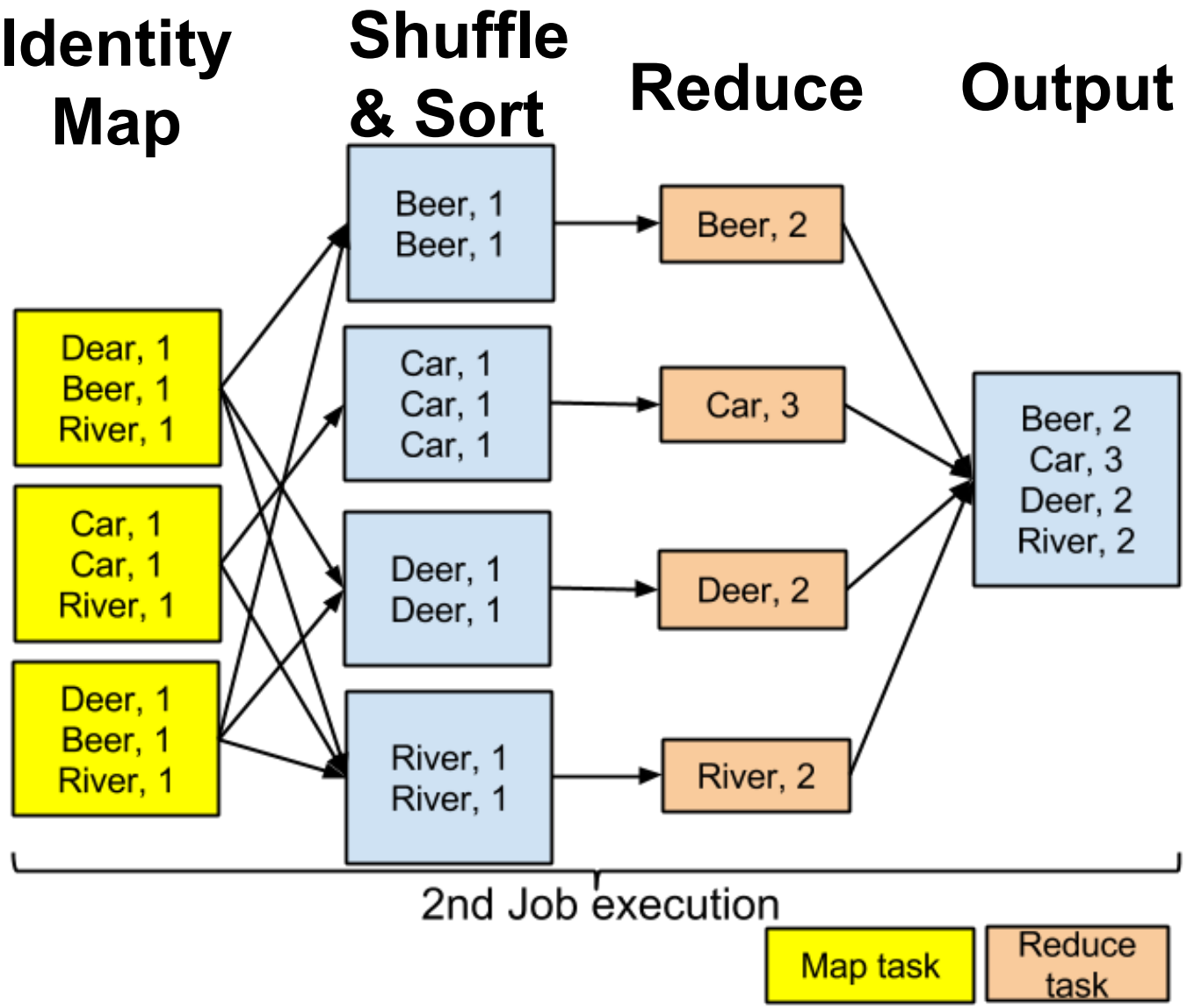
Logical Job

- Cannot **pause** a job with MapReduce API
- One logical job will be **dedicated to the map tasks**, and the other to the **reduce tasks**
- Create the concept of **Identity map tasks**
- **Hadoop viewpoint**, each logical job is a complete MapReduce job.
- **Chrysaor viewpoint**, one logical job corresponds to the map tasks of the Chrysaor job, and the other to the reduce tasks

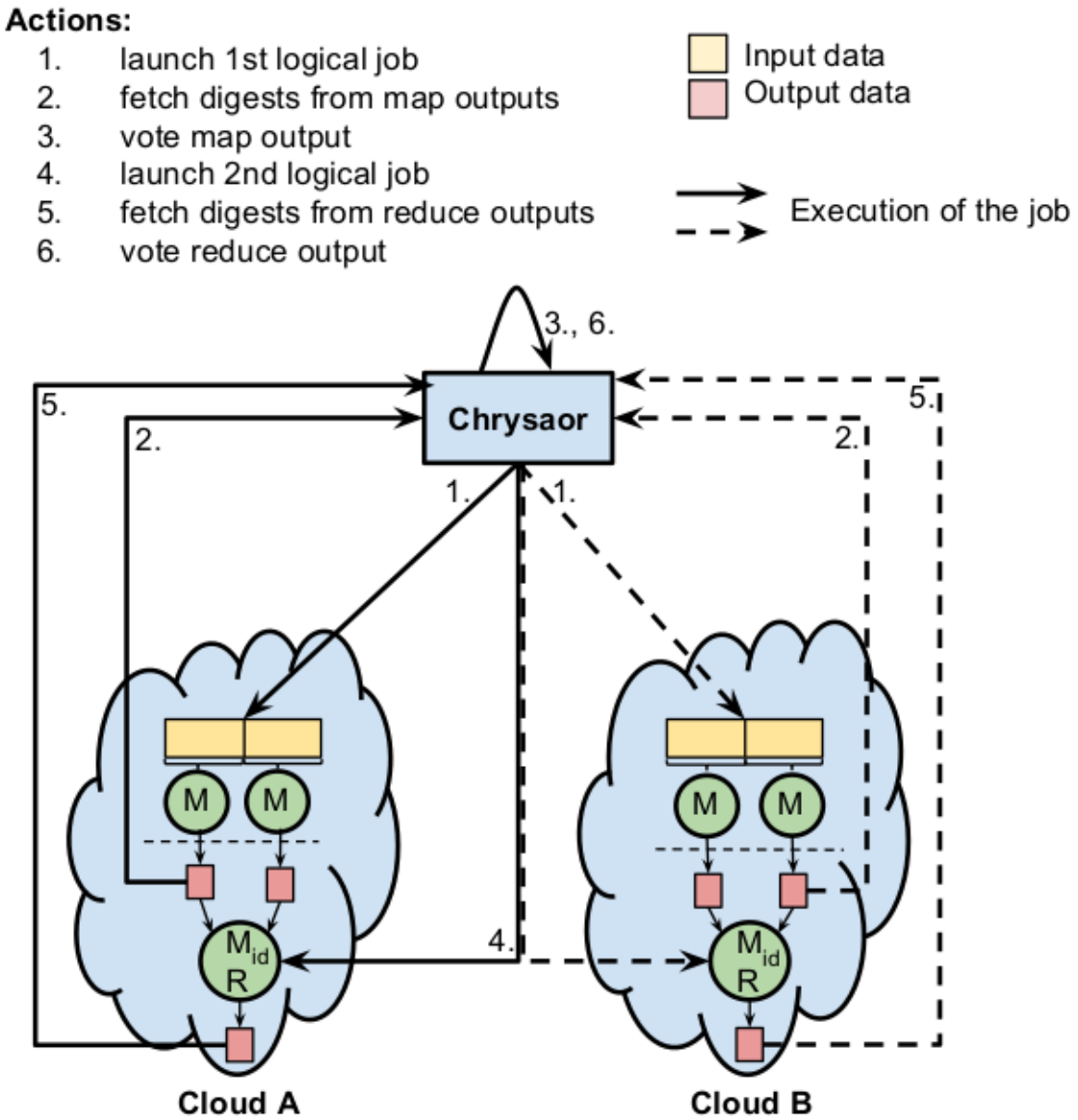
Logical Job


Input
Splitting
Map
Deer Beer River
Car Car River
Deer Beer River
Deer Beer River
Car Car River
Deer Beer River
Dear, 1
Beer, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Beer, 1
River, 1
1st Job execution
Map task
Reduce task

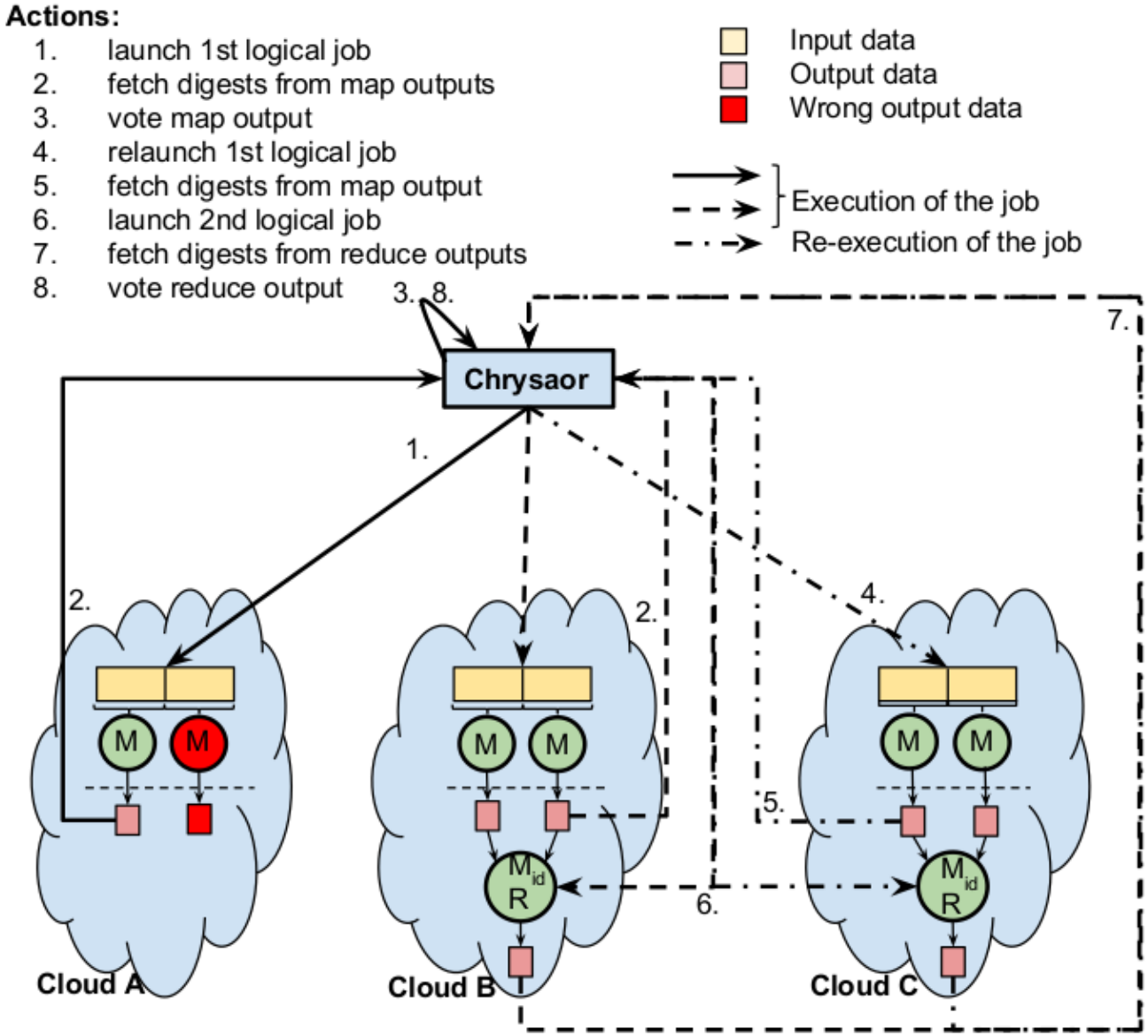
Logical Job


Identity Map
Shuffle & Sort
Reduce
Output
Dear, 1
Beer, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Beer, 1
River, 1
Beer, 1
Beer, 1
Car, 1
Car, 1
Car, 1
Deer, 1
Deer, 1
River, 1
River, 1
Beer, 2
Car, 3
Deer, 2
River, 2
Beer, 2
Car, 3
Deer, 2
River, 2
2nd Job execution
Map task
Reduce task

Logical Job (Bird's Eye View)



Logical Job (With Malicious Faults)



WordCount Example

```
public static void main(String[] args) throws Exception {
        // first map tasks
        JobConf conf = new JobConf(MyWordCount.class);
        conf.setJobName("wordcount");
        (…)
        job.setMapperClass(MyMap.class);
        job.setReducerClass(MyReducer.class);
        conf.setClass("mapreduce.job.map.identity.class",
MyFullyIndentityMapper.class, Mapper.class);
        (...)
        JobClient.runJob(job);
}
```

WordCount Example

```
public static class MyFullyIndentityMapper extends
  Mapper<Object, Text, Text, IntWritable>{

  public void map(Object key, Text value, Context context)
throws IOException, InterruptedException {
    StringTokenizer itr = new
StringTokenizer(value.toString());
    while (itr.hasMoreTokens()){
      word.set(itr.nextToken());
    }
  }
}
```

Outline

- MapReduce
- Contribution
- MapReduce Execution
  - MapReduce Job
  - Logical MapReduce Jobs
  - WordCount Example
- **Evaluation**
  - Experimental Setup
  - Experimental Results
- Conclusions

Experimental Setup

- Evaluate the performance using Hadoop Gridmix Benchmark

Experimental Setup

- Evaluate the performance using Hadoop Gridmix Benchmark
- Amazon EC2 testbed

Experimental Setup

- Evaluate the performance using Hadoop Gridmix Benchmark
- Amazon EC2 testbed
- Setup 3 clouds located in different sites

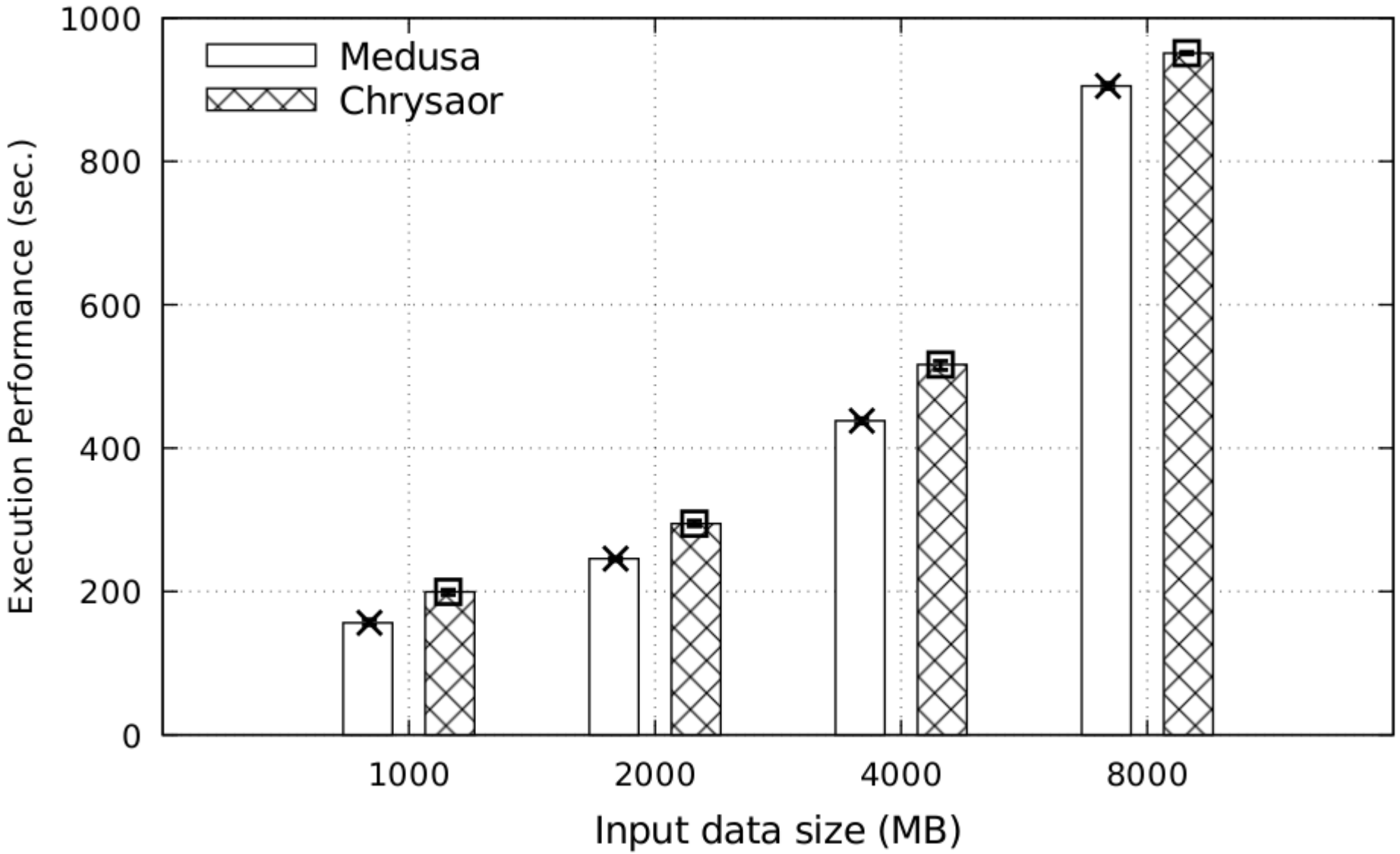
Experimental Setup

- Evaluate the performance using Hadoop Gridmix Benchmark
- Amazon EC2 testbed
- Setup 3 clouds located in different sites
- Each cloud is composed by 1 Resource Manager, and 4 Node managers

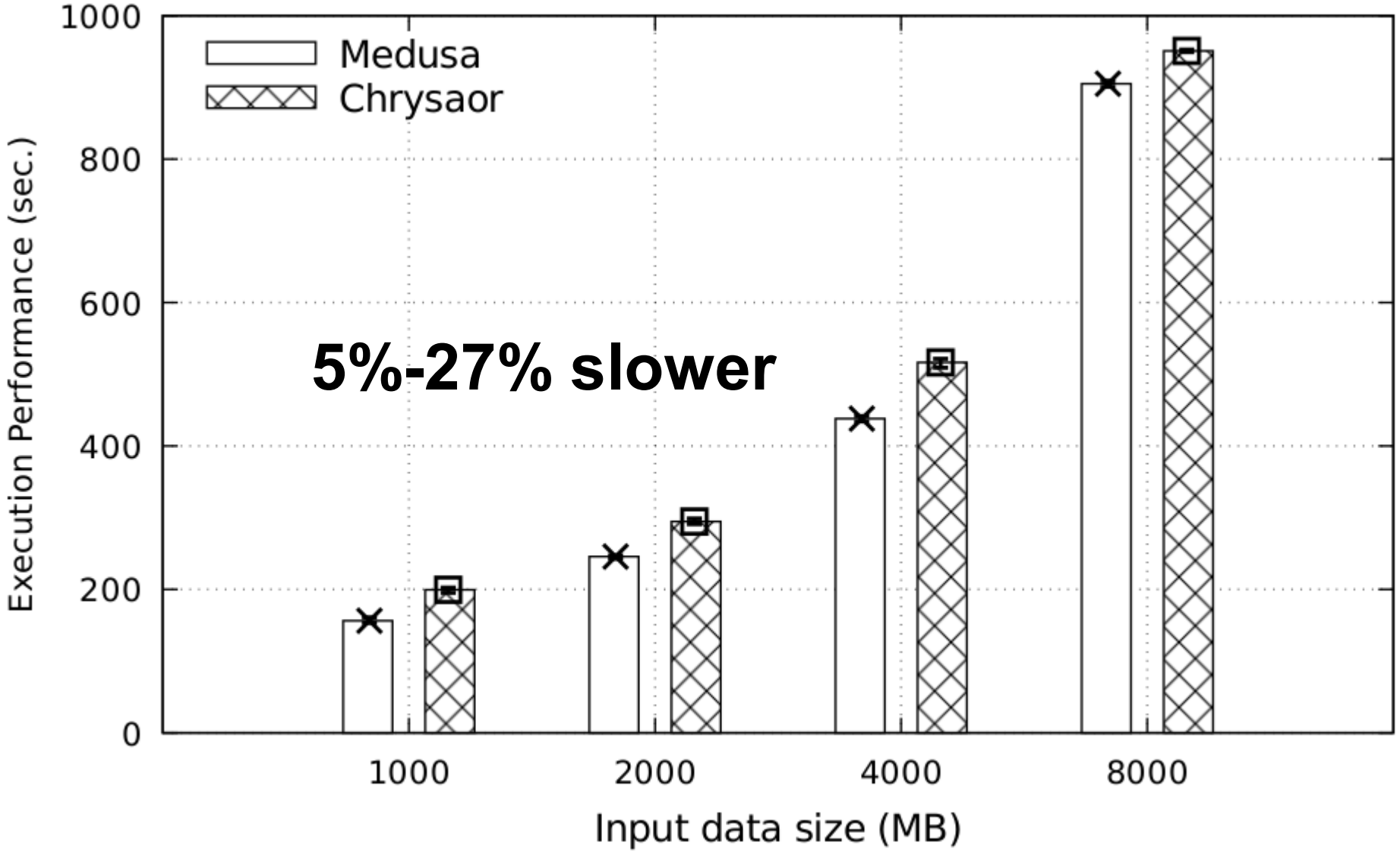
Experimental Setup

- Evaluate the performance using Hadoop Gridmix Benchmark
- Amazon EC2 testbed
- Setup 3 clouds located in different sites
- Each cloud is composed by 1 Resource Manager, and 4 Node managers
- *f=1*

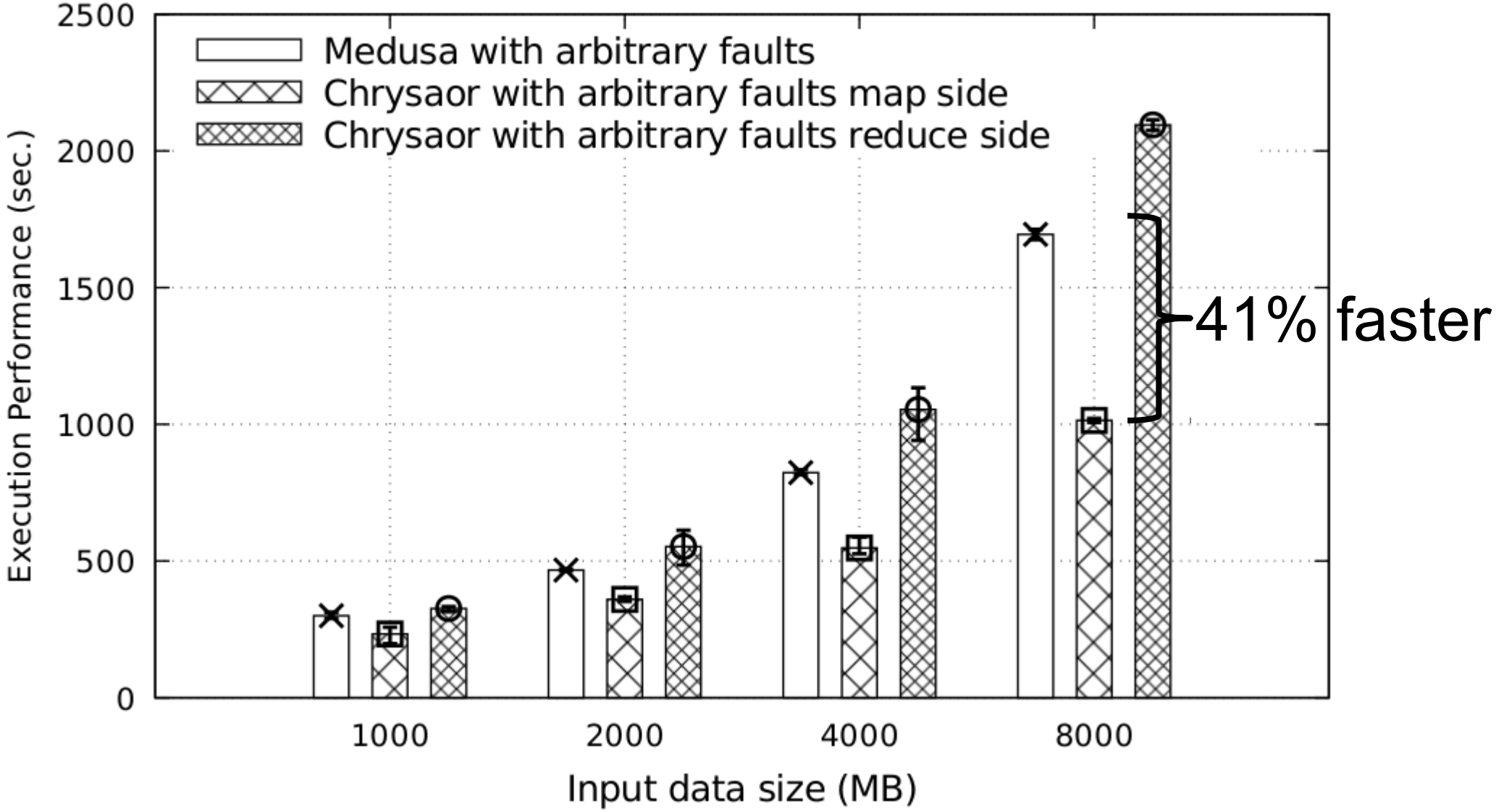
WordCount Performance



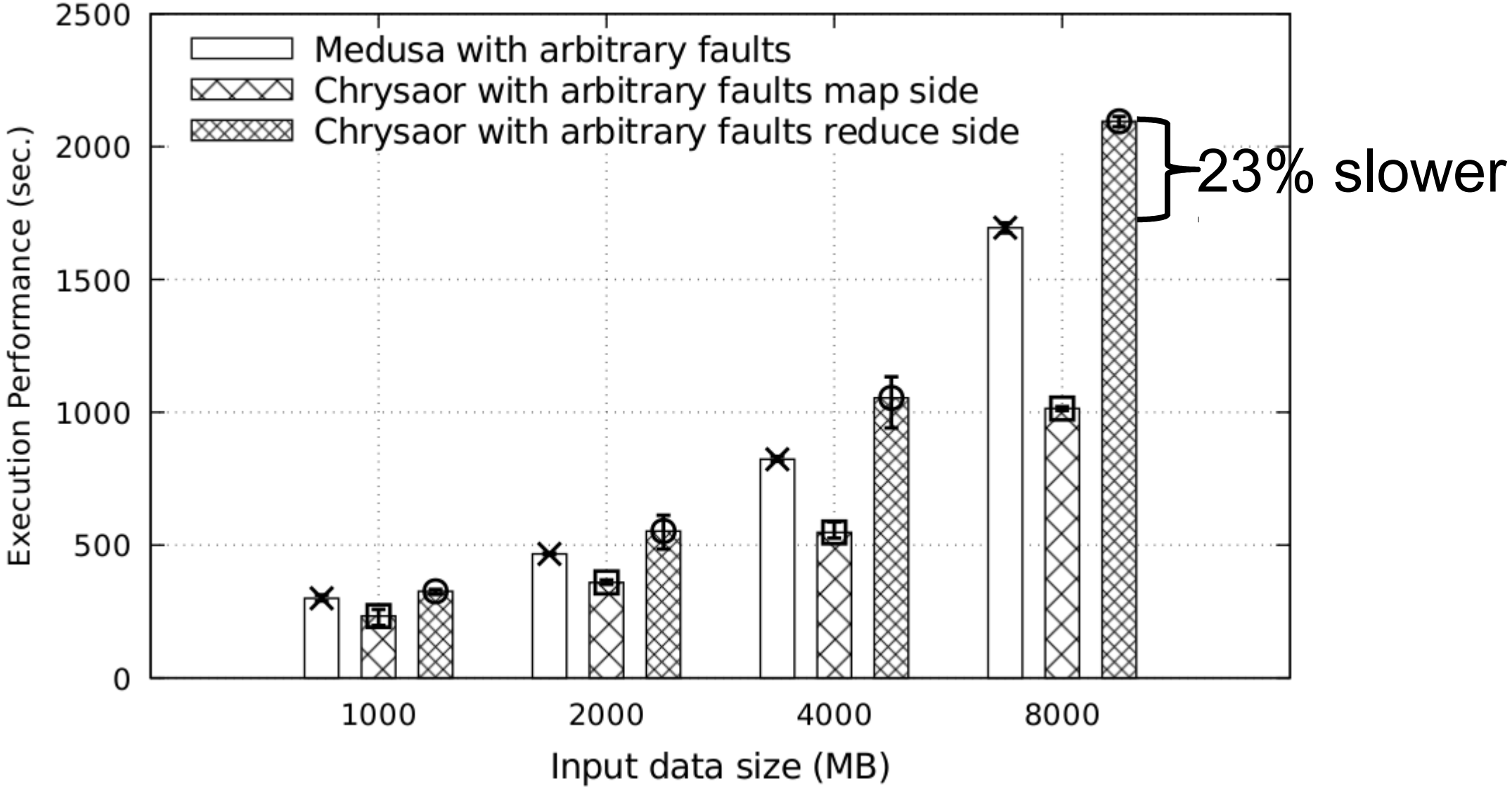
WordCount Performance



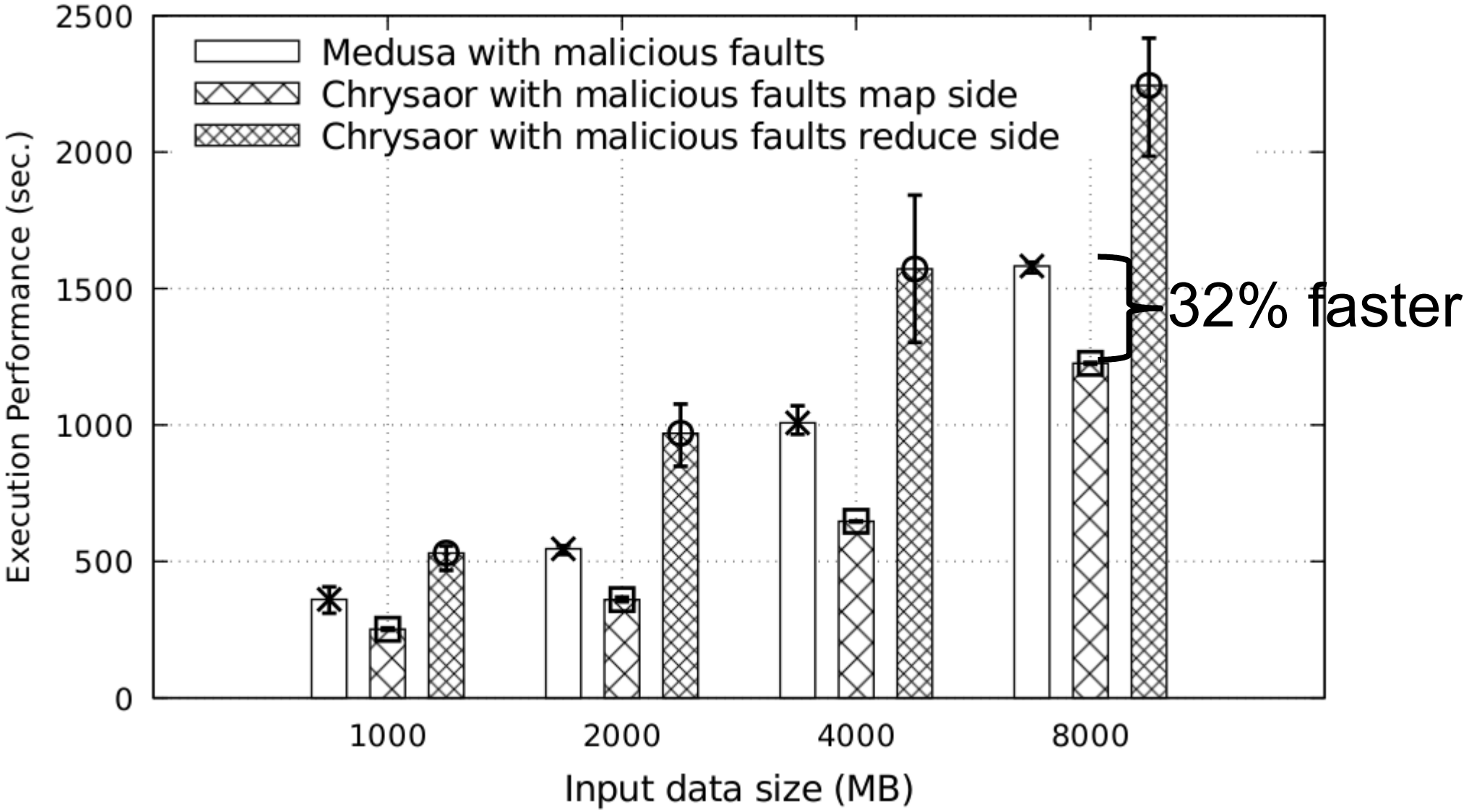
WordCount Performance



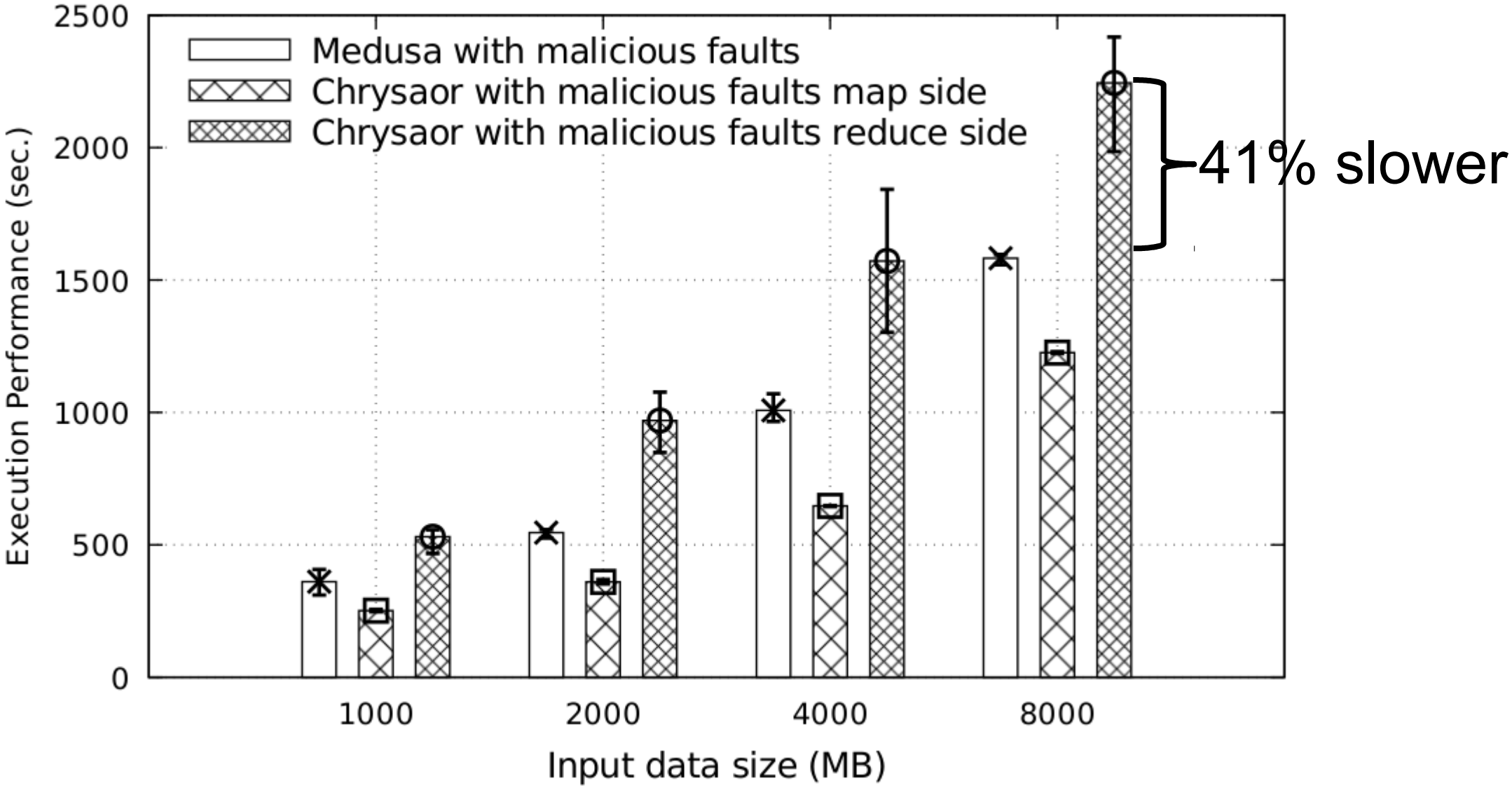
WordCount Performance



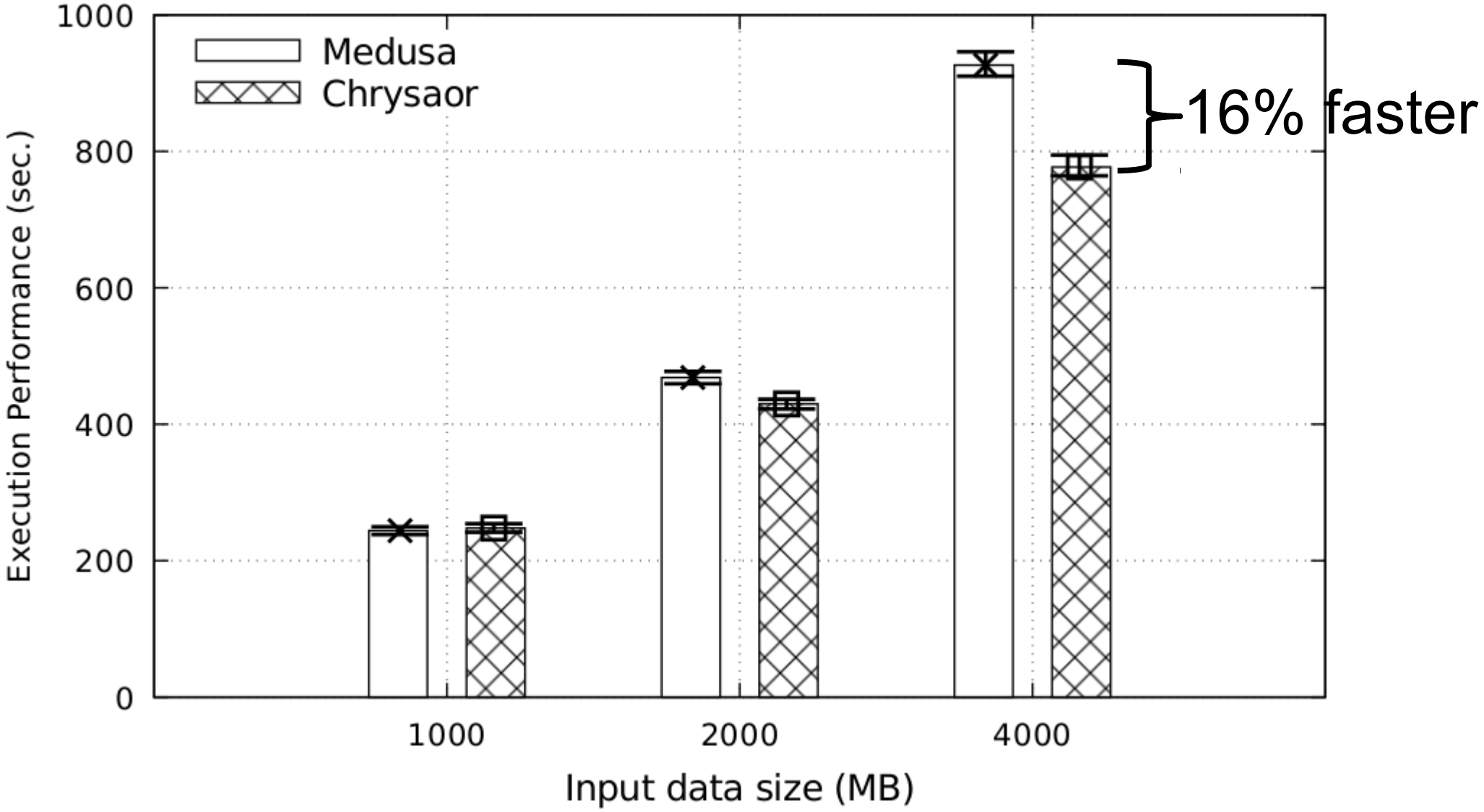
WordCount Performance



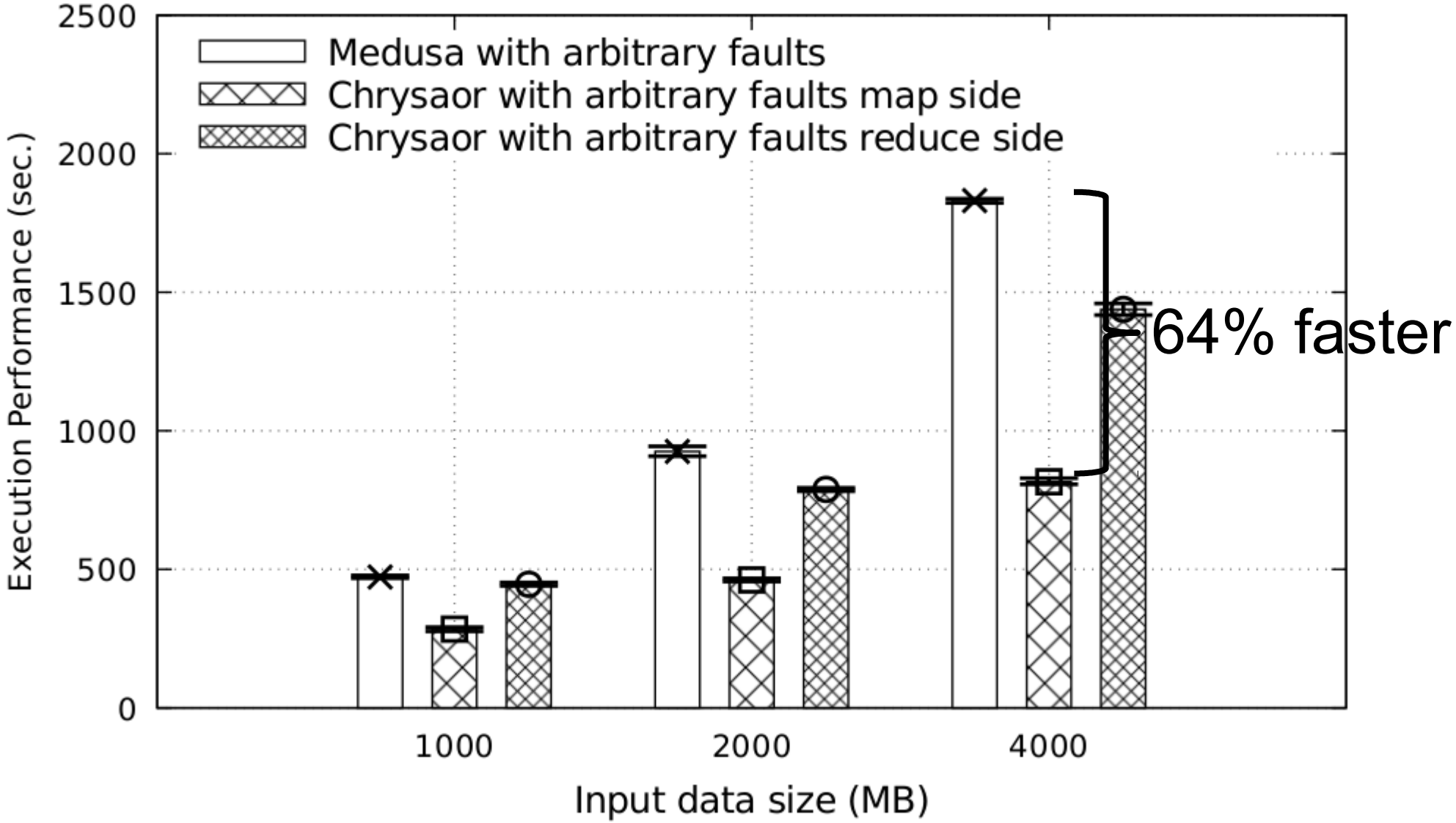
WordCount Performance



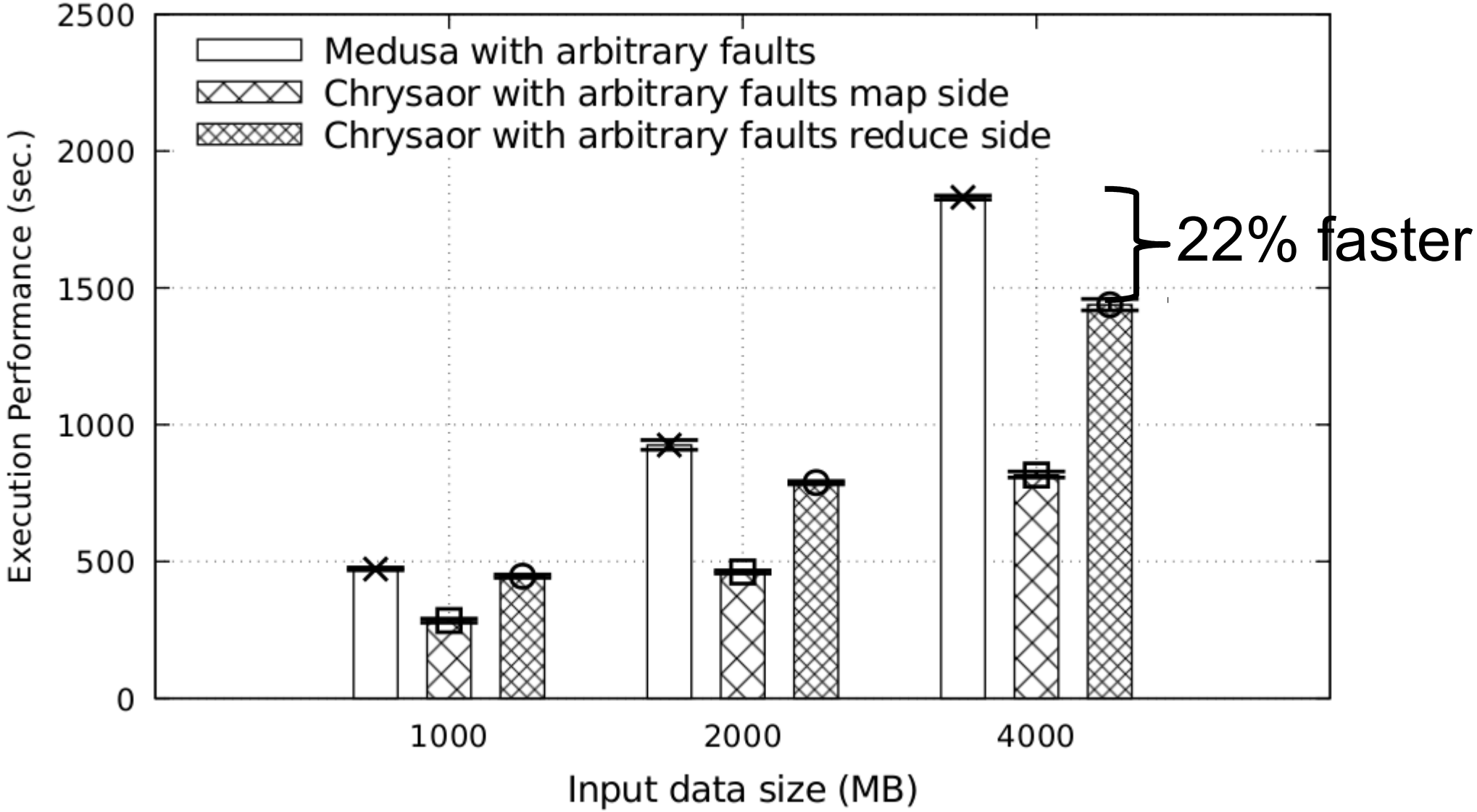
Sort Performance



Sort Performance



Sort Performance



Conclusion

- Hadoop fault tolerance mechanism **cannot deal with arbitrary faults**

Conclusion

- Hadoop fault tolerance mechanism **cannot deal with arbitrary faults**
- Chrysaor application to **scale out MapReduce applications**

Conclusion

- Hadoop fault tolerance mechanism **cannot deal with arbitrary faults**
- Medusa application to **scale out MapReduce applications**
- It achieves **great level of fault tolerance** at a resoanable cost

Questions?

Acknowledgements

Navigators

LaSIGE

inesc id lisboa

Ciências ULisboa