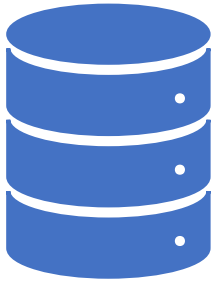
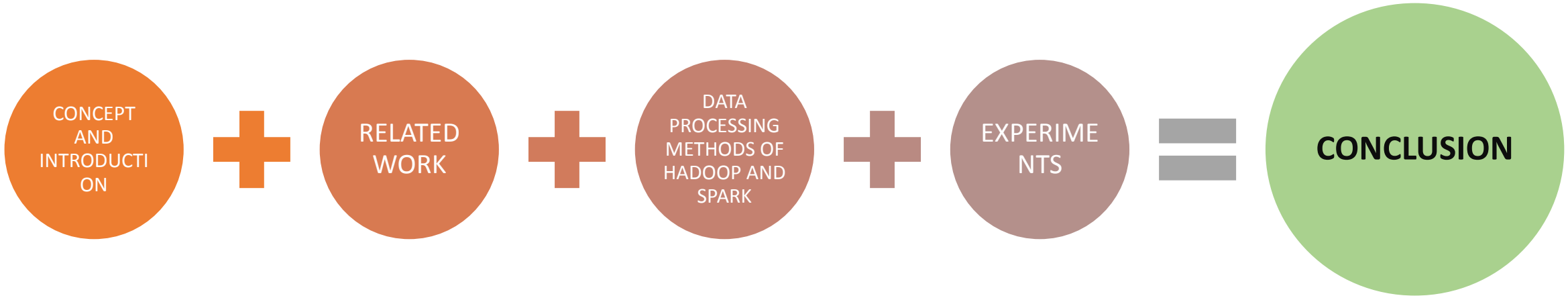
Impact of Memory Size on Bigdata Processing based on Hadoop and Spark

of Seunghye Han, Wonseok Choi, Rayan Muwafiq, Yunmook Nah

Presenting…

Shaista Gulnaar




CONCEPT AND INTRODUCTION
RELATED WORK
DATA PROCESSING METHODS OF HADOOP AND SPARK
EXPERIMENTS
CONCLUSION

OUTLINE

Hadoop - Hadoop Distributed File System and MapReduce processing. It stores intermediary data on Hadoop Distributed File System, which is a disk-based distributed file system

Spark stores intermediary data in the memories of distributed computing nodes as Resilient Distributed Dataset.

how memory size affects distributed processing of large volume of data, by comparing the running time of K-means algorithm of HiBench benchmark on Hadoop and Spark clusters, with different size of memories allocated to data nodes.

Our results show that Spark cluster is faster than Hadoop cluster as long as the memory size is big enough for the data size.

But, with the increase of the data size, Hadoop cluster outperforms Spark cluster. When data size is bigger than memory cache, Spark has to replace disk data with memory cached data, and this situation causes performance degradation.

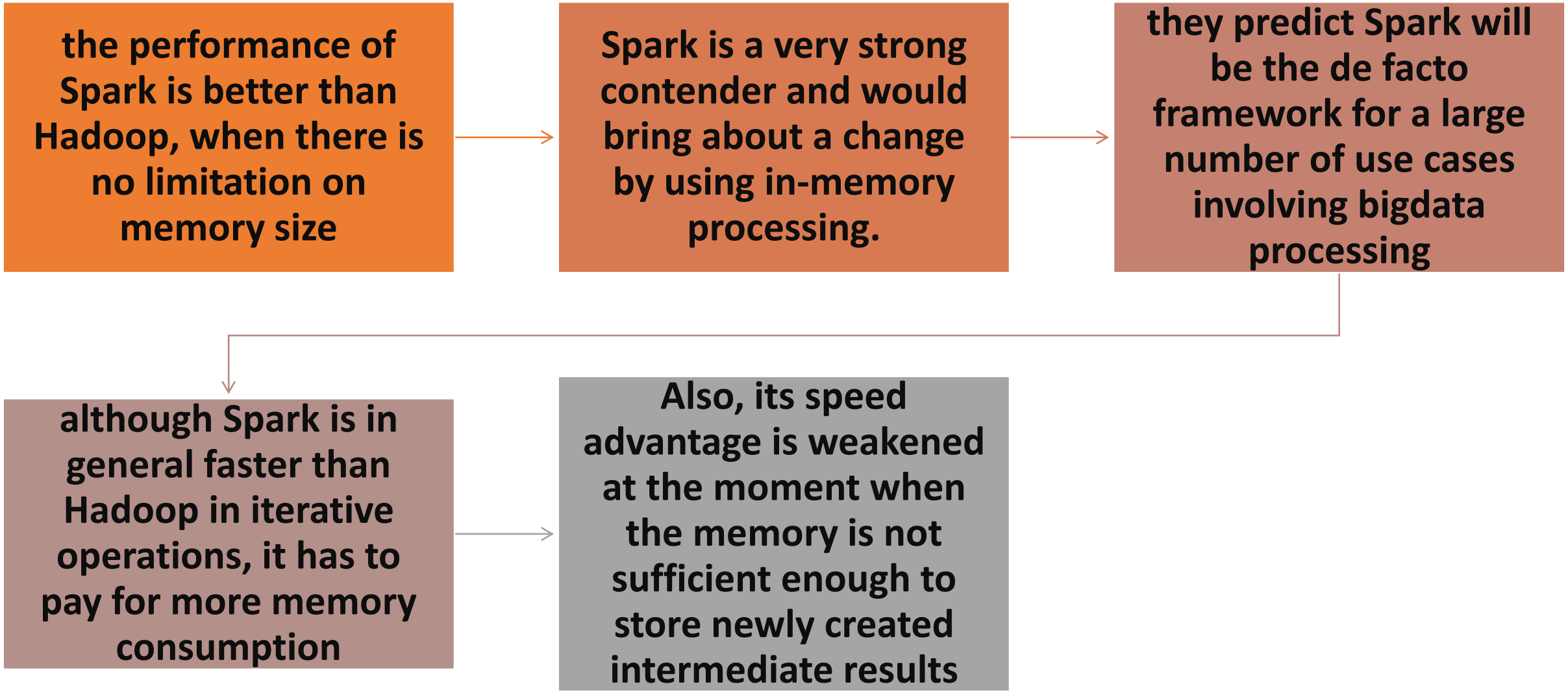
CONCEPT

need to manage very large volume of data produced by such devices is ever increasing rapidly

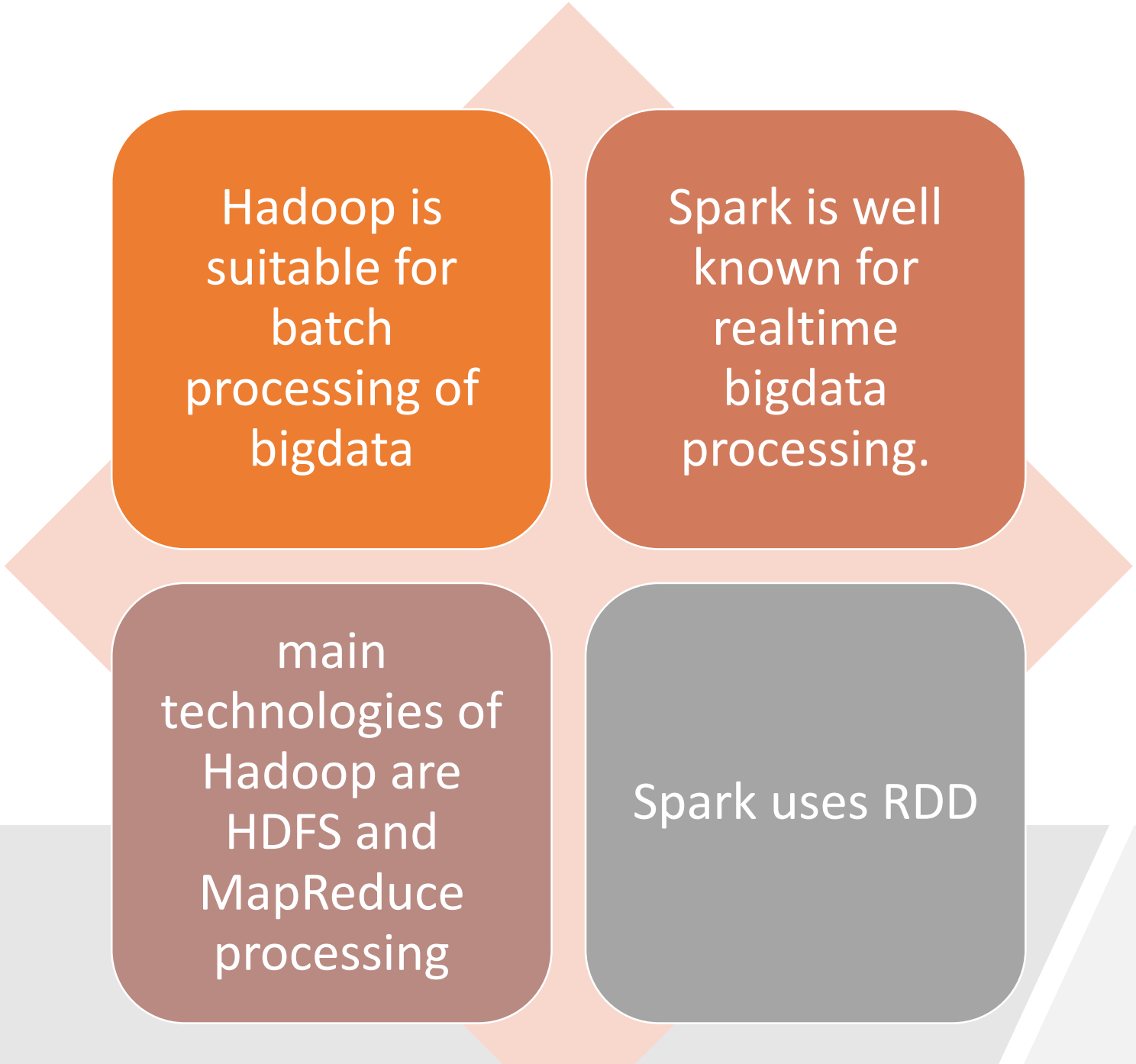
Hadoop technologies are disk-based, thus having physical limitations in processing speeds

Spark is a main-memory-based open source big data processing platform with the concept of RDD (Resilient Distributed Dataset), which generates input data set within main memory, thus enabling very fast in-memory processing of data compared with disk-based solutions.

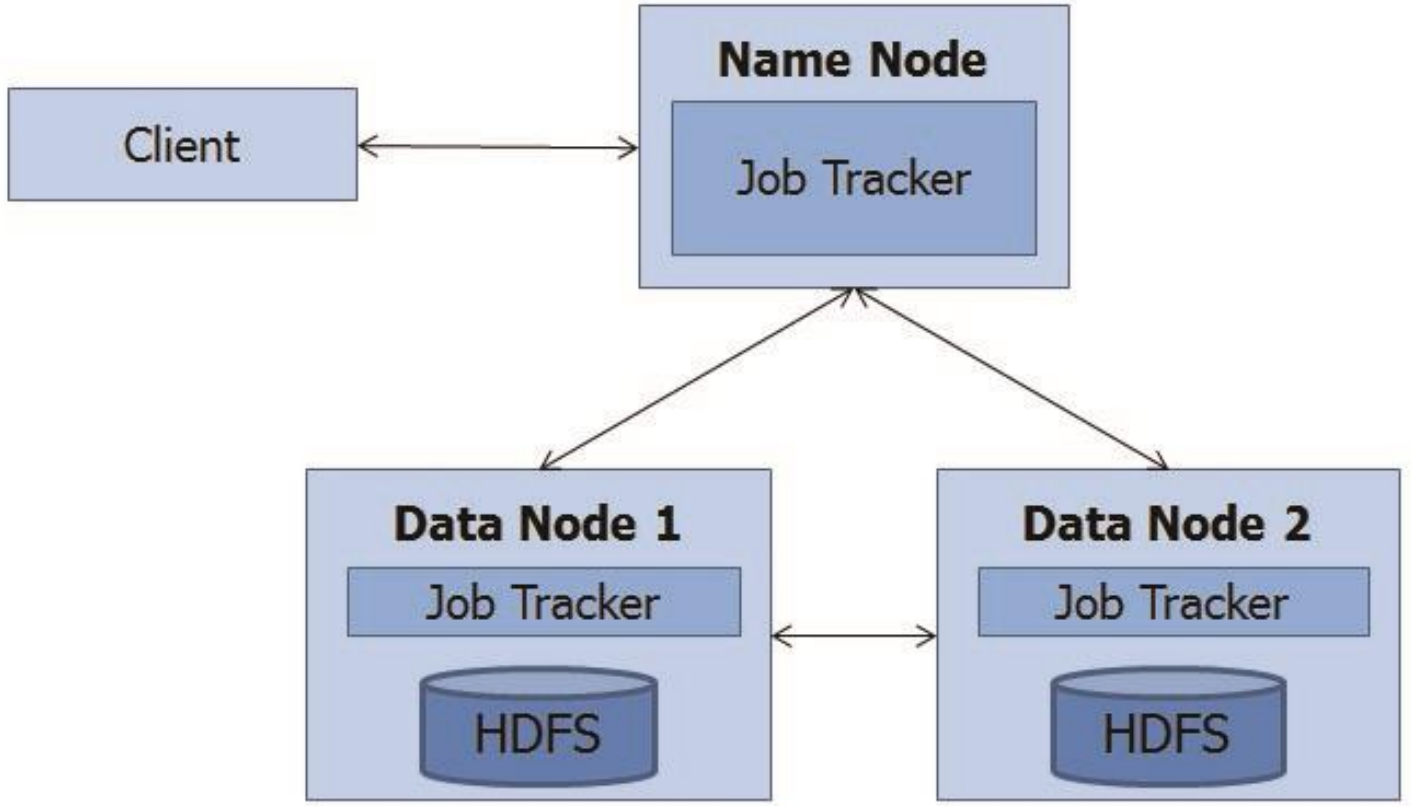
INTRODUCTION



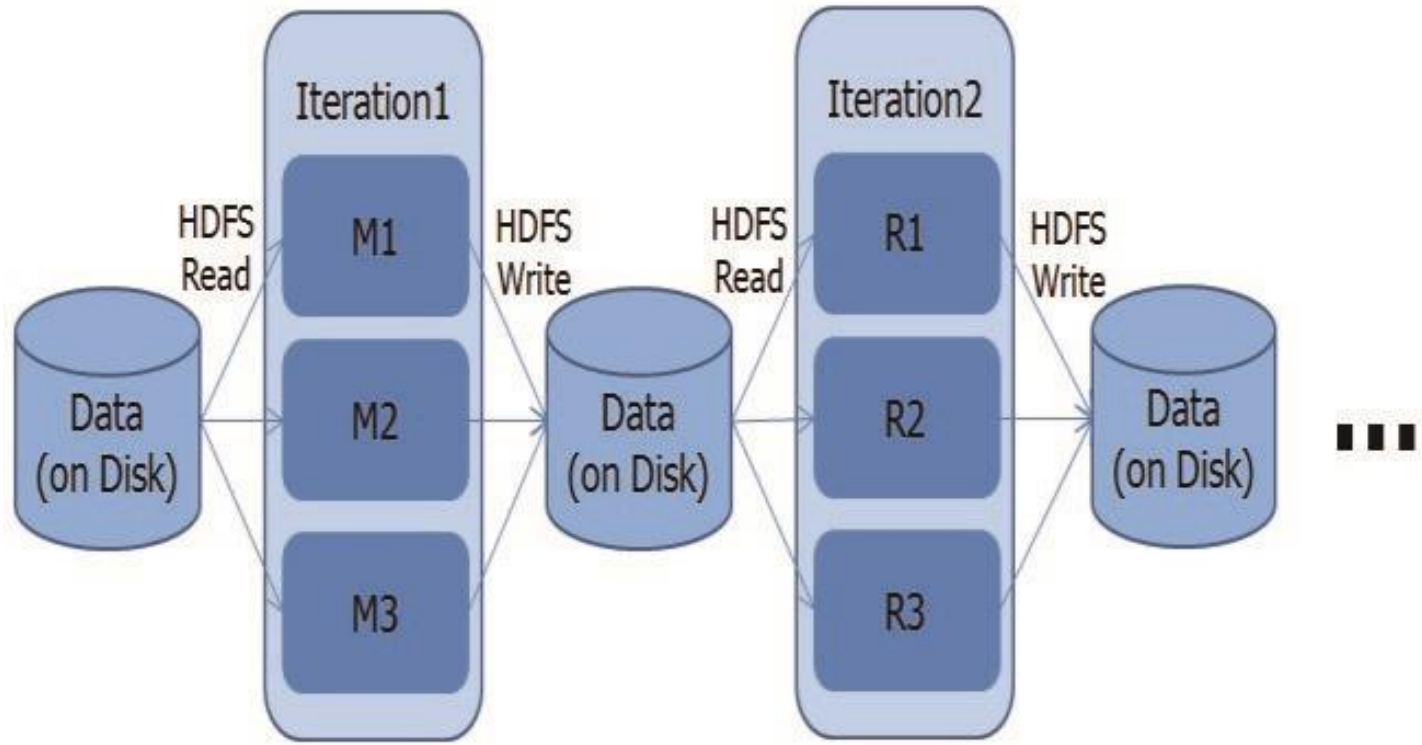
RELATED WORK


Hadoop is suitable for batch processing of bigdata
Spark is well known for realtime bigdata processing.
main technologies of Hadoop are HDFS and MapReduce processing
Spark uses RDD

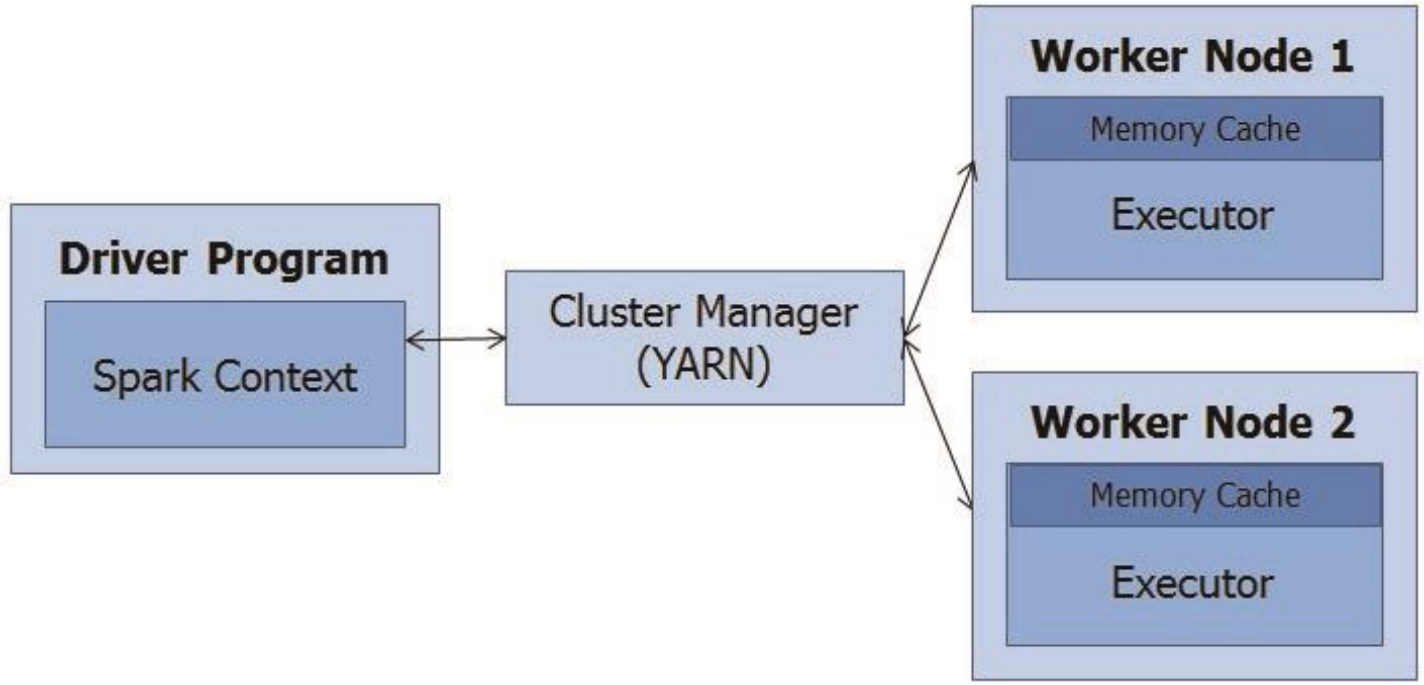
Hadoop cluster with one master node and two slave nodes



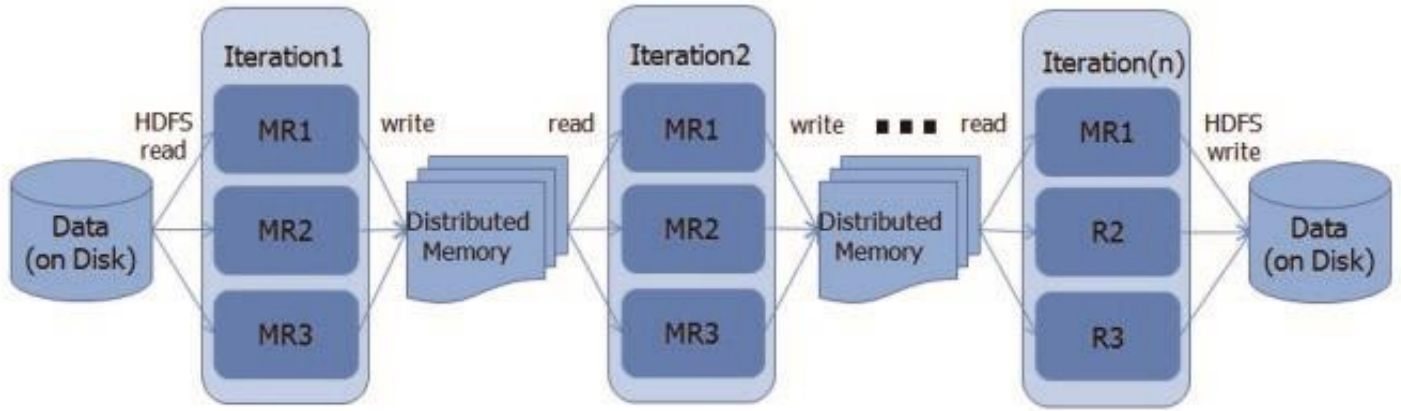
Iterative processing using Hadoop



Spark cluster



Iterative processing using Spark RDD

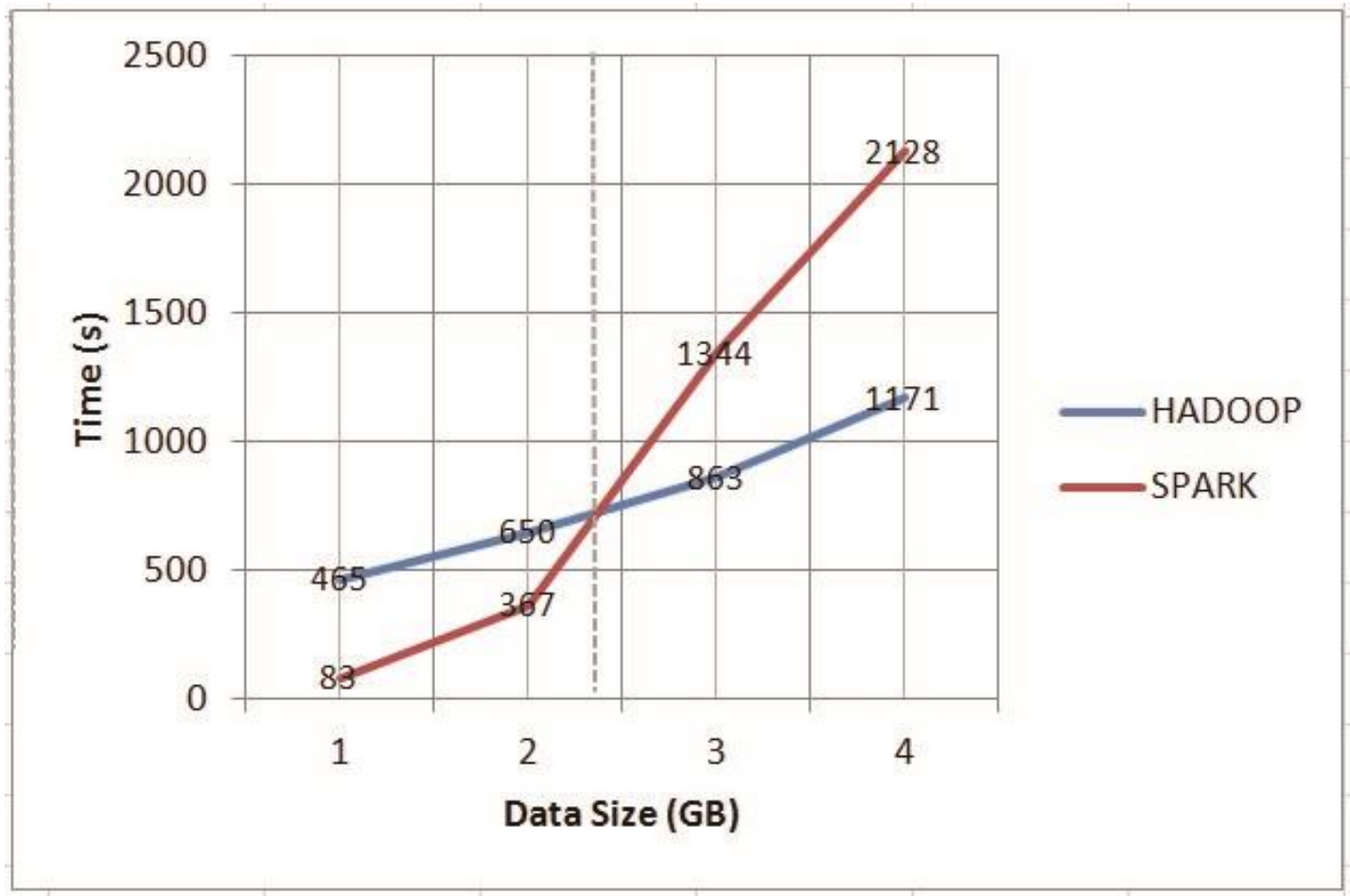


EXPERIMENTAL DETAILS

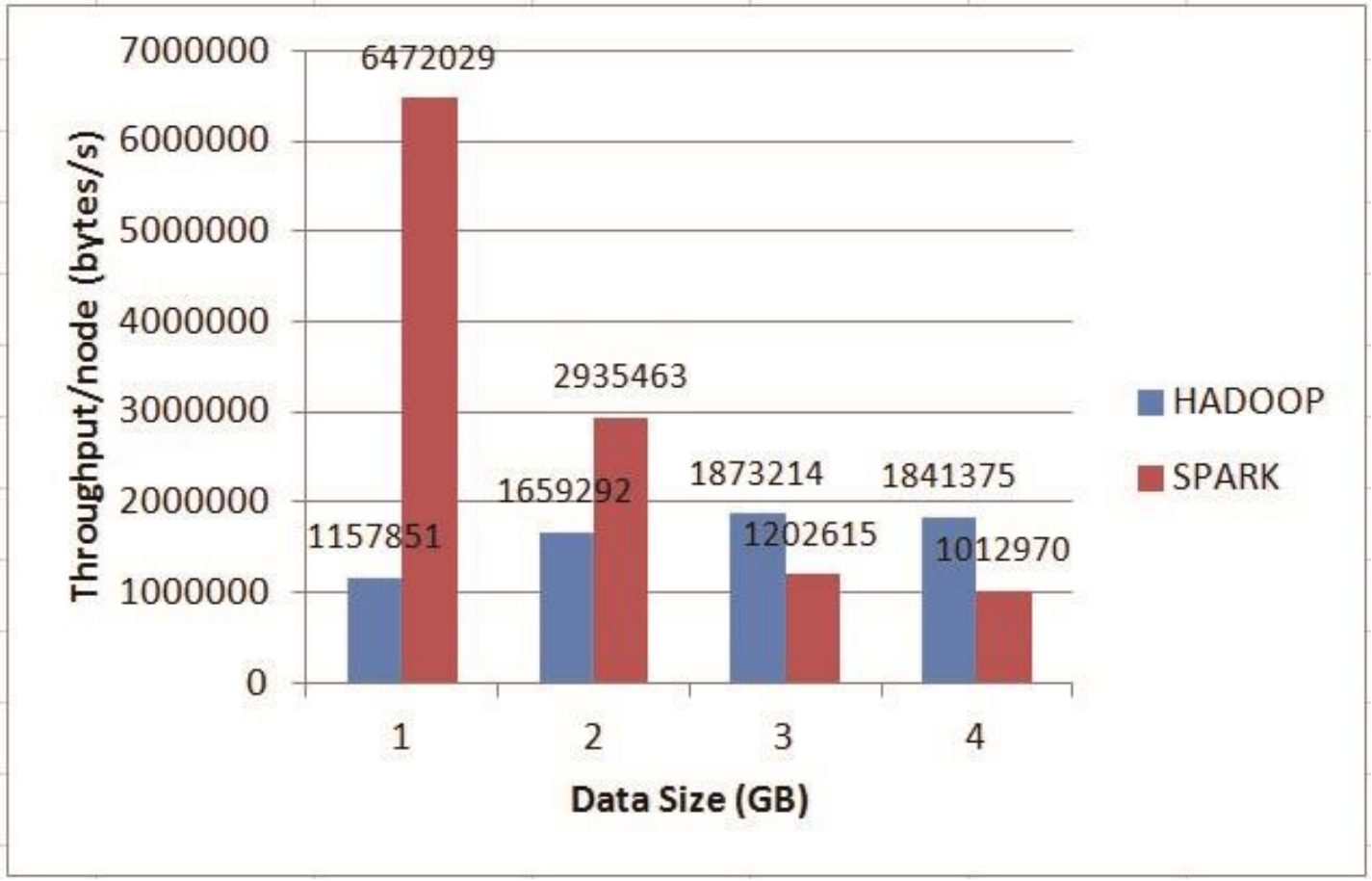
Table 1. Experimental platform

Physical Node	***CPU***	INTEL CORE I5-4690 CPU @ 3.50GHZ
	Memory	16GB
	HDD	500GB
	Network	1Gbps Ethernet
OS	Ubuntu 15.04	
Apache Spark	1.6.1	
Hadoop	2.7.0	
JDK	1.7.0_79	
Benchmark	HiBench v4.0	

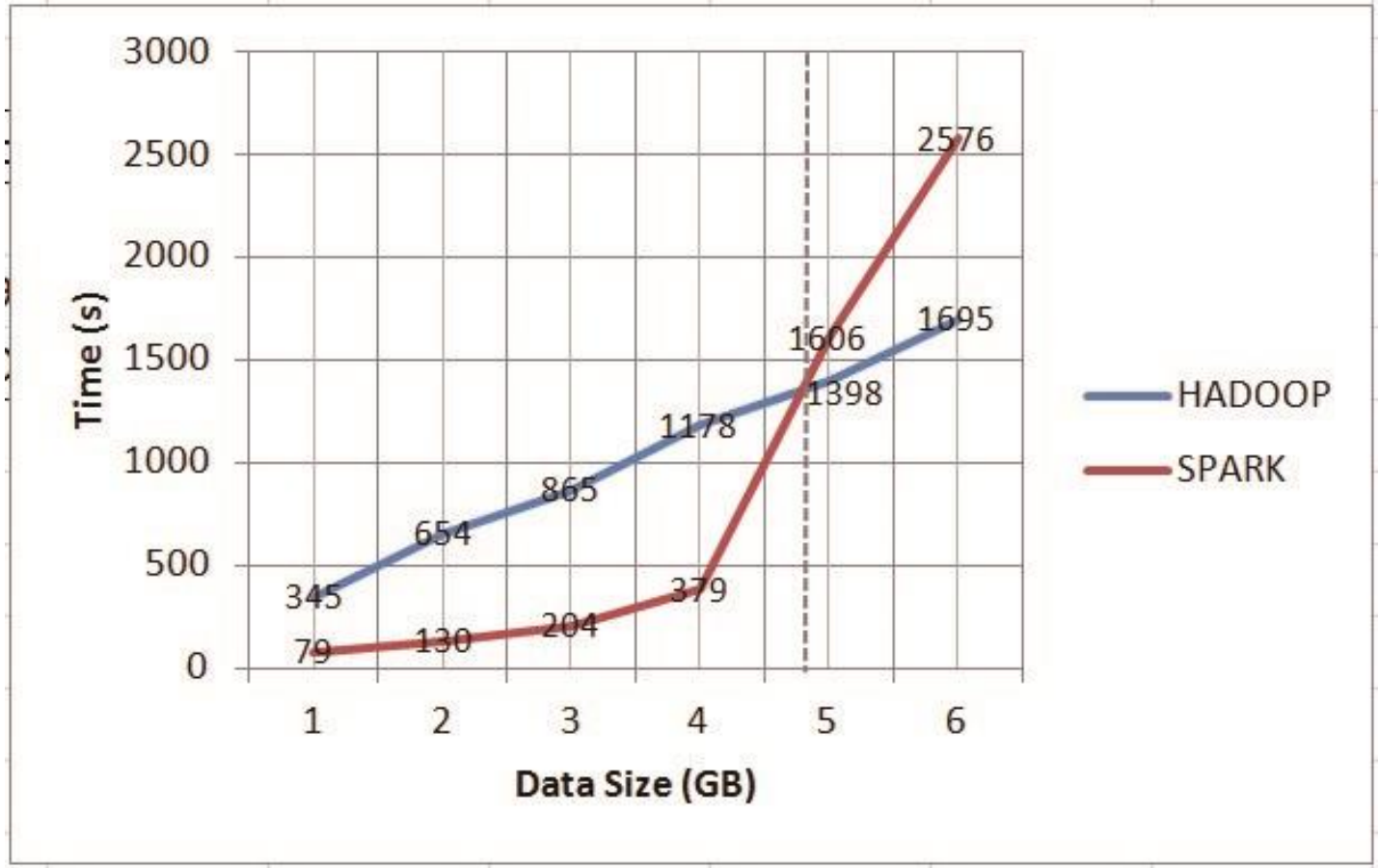
Data processing time with 4GB memory per node.



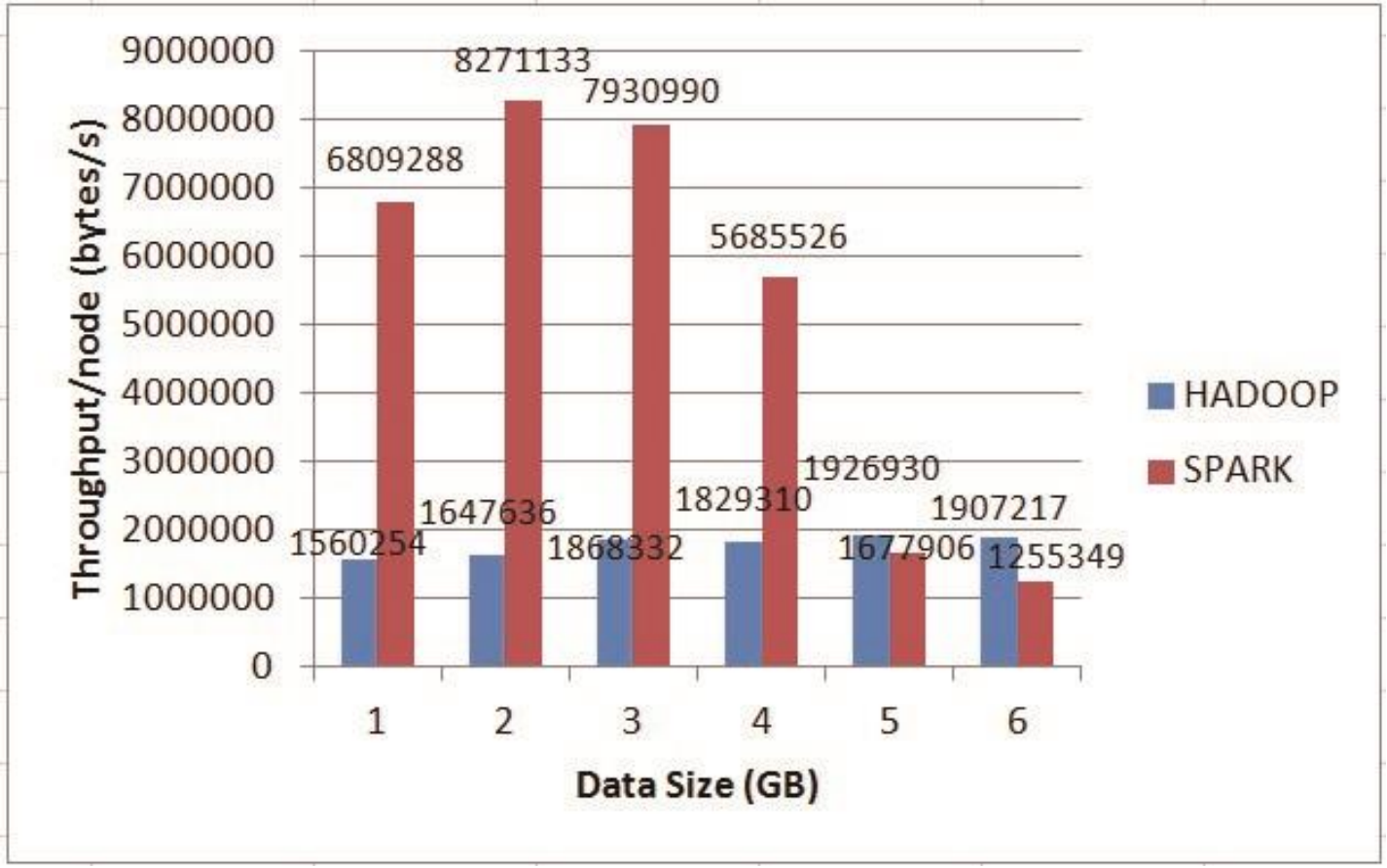
Data processing rate with 4GB memory per node.



Data processing time with 8GB memory per node.



Data processing rate with 8GB memory per node



Spark memory structure

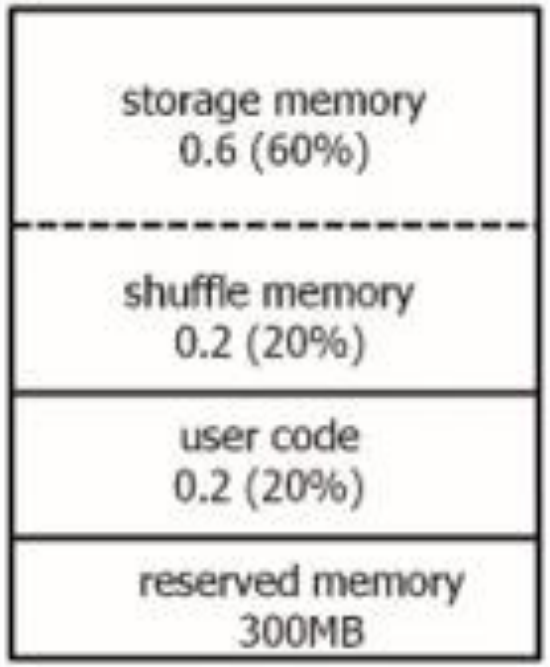
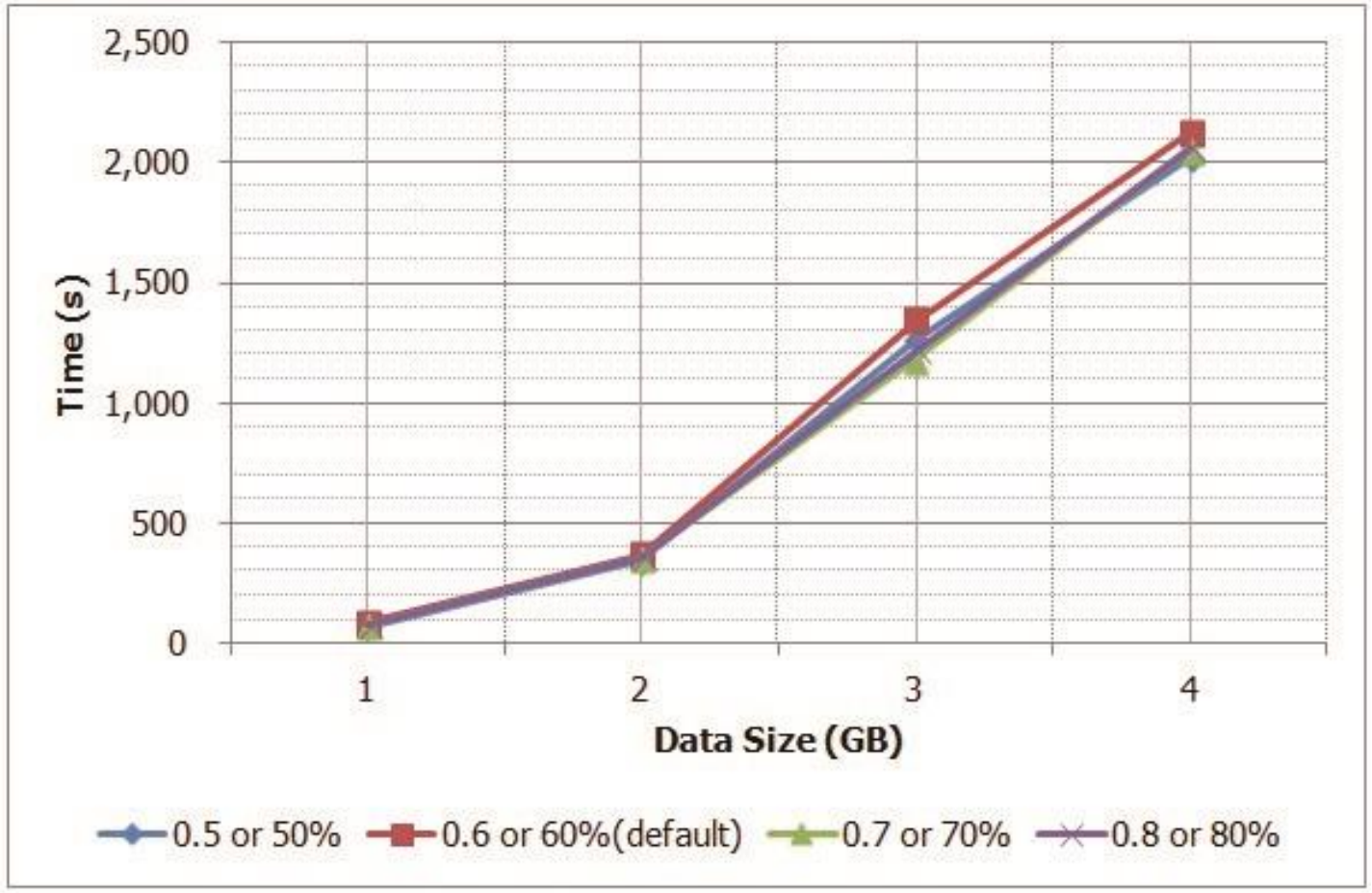
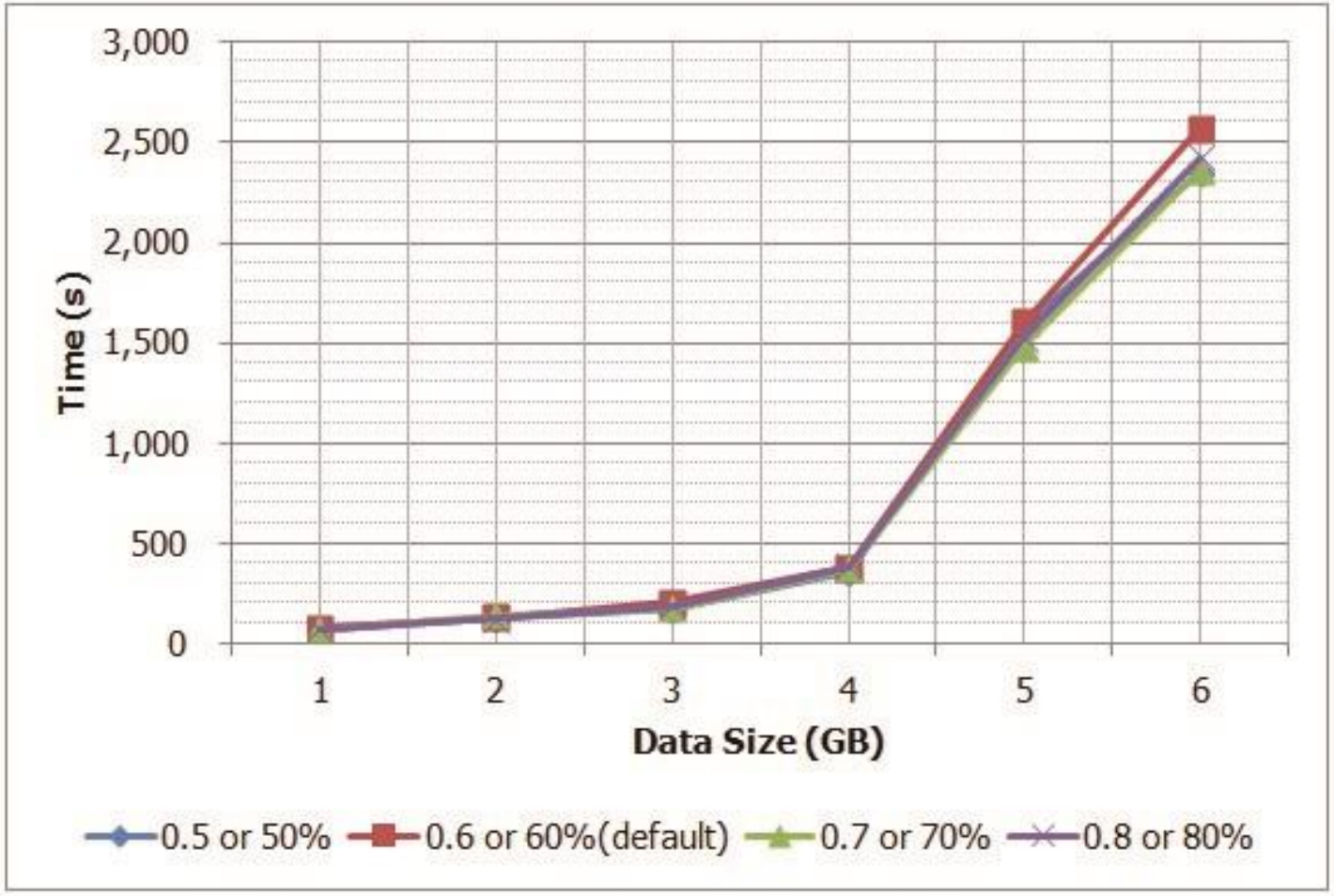


Figure 9. Spark memory structure.

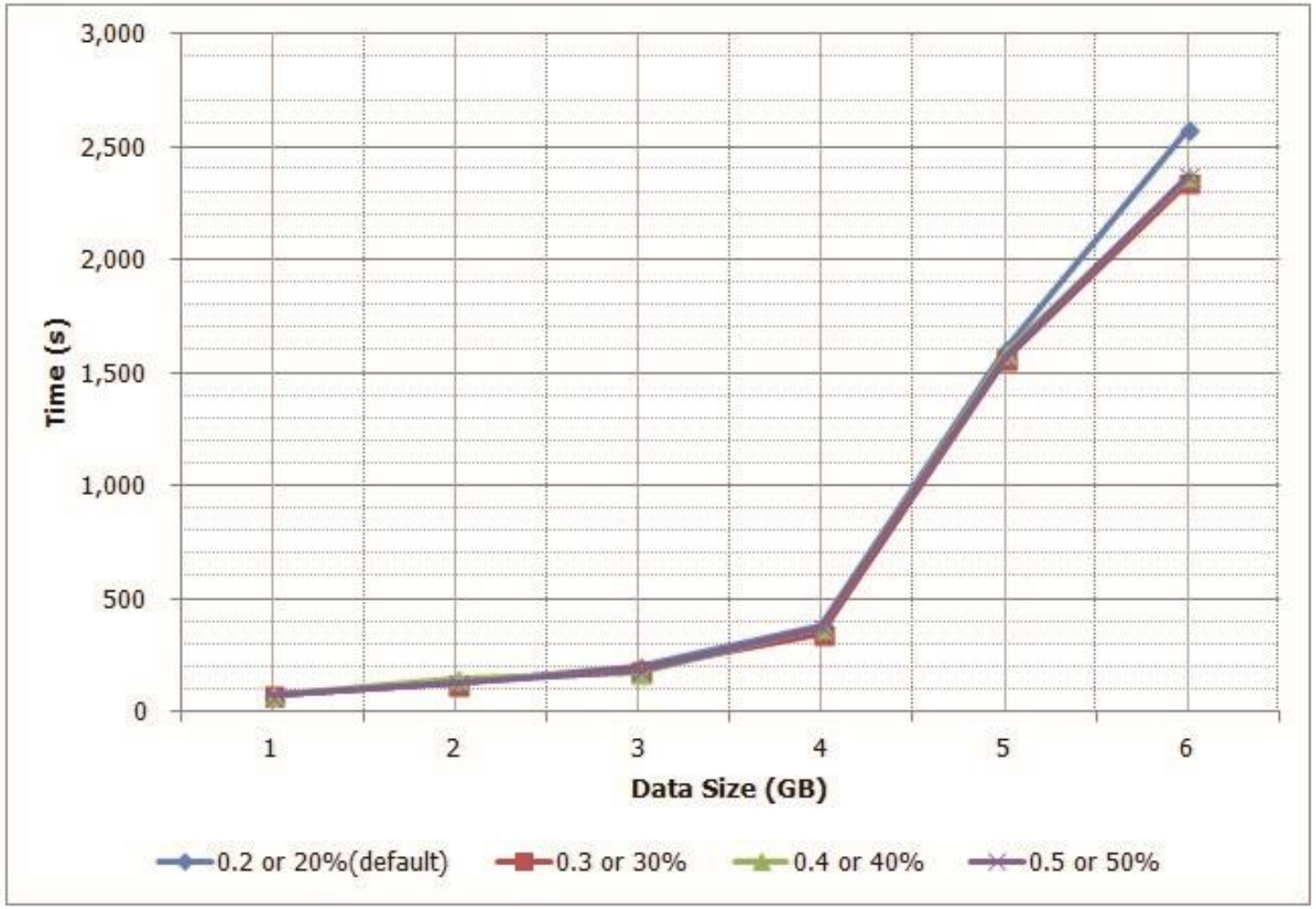
Spark processing time with different storage memory size (4GB memory per node).



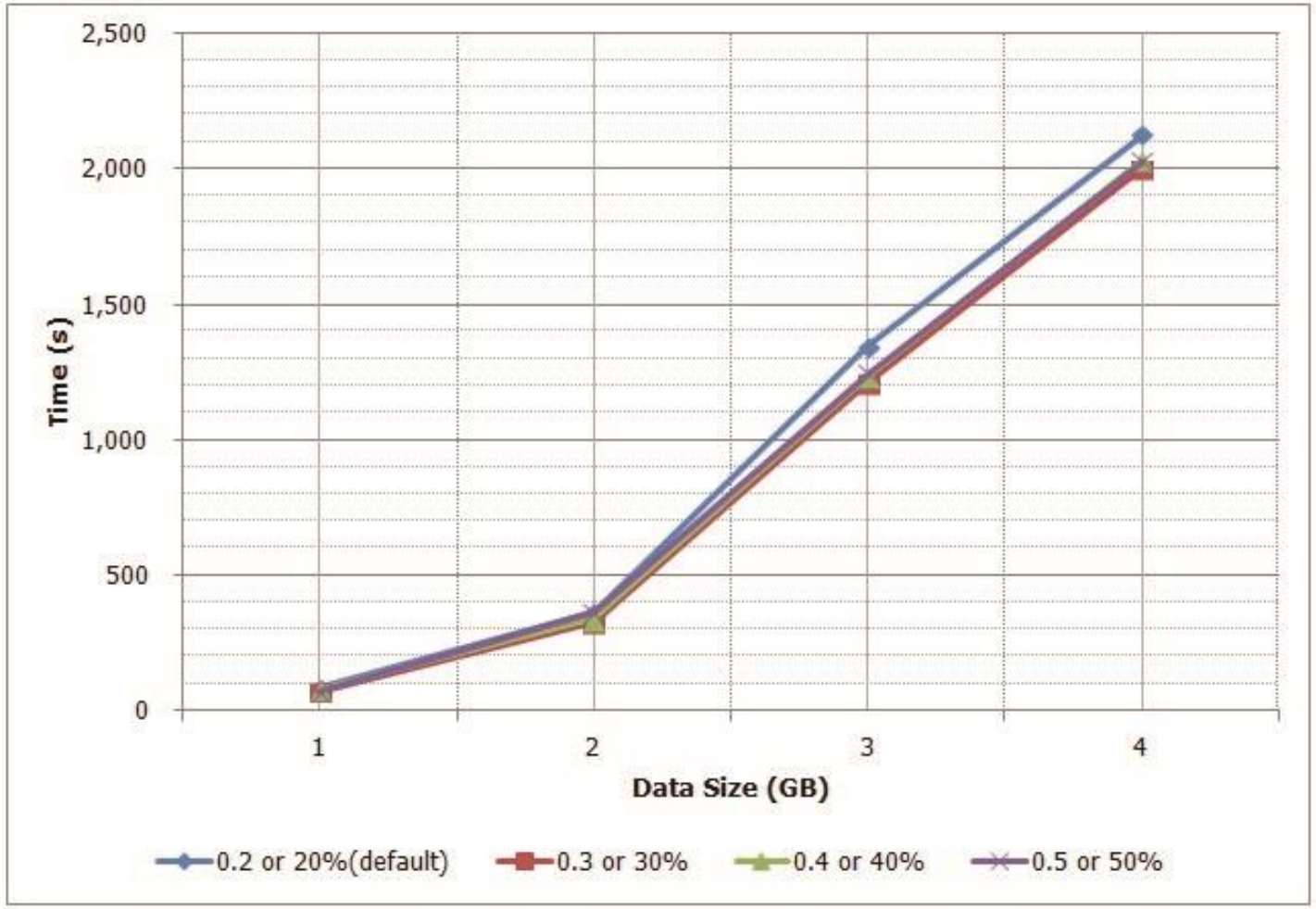
Spark processing time with different storage memory size (8GB memory per node).



Spark processing time with different shuffle memory size (4GB memory per node).



Spark processing time with different shuffle memory size (8GB memory per node).



Compared the running time of K-means algorithm of HiBench benchmark on Hadoop and Spark clusters, with different size of memories allocated to data nodes, to show how memory size affects distributed processing of large volume of data.

Our results show that Spark cluster is faster than Hadoop cluster as long as the memory size is big enough for the data size

But, with the increase of the data size, Hadoop cluster outperforms Spark cluster. With the increase of the data size, Spark cluster requires more time and its data processing throughput decreases rapidly. When data size is bigger than memory cache, Spark has to replace disk data with memory cached data, and this situation causes performance degradation.

CONCLUSIONS

Related with K-means algorithm processing, Spark is better than Hadoop when total input data size is smaller than 33.5% of total memory size assigned to whole worker nodes

while Hadoop is better than Spark when the total data size is greater than 33.5% of total memory size

CONCLUSIONS

FUTURE WORK

- Additional experiments considering further parameters, such as node numbers, will be helpful to find out more performance-influencing factors.

THANK YOU ☺