COLLEGE OF SCIENCE & ENGINEERING
Department of Computer Science

Frog-B-Data

Big Data is a Big Deal!

Sushant Ahuja, Cassio Cristovao, Sameep Mohta
Advisors: Dr. Antonio Sanchez, Dr. Donnell Payne

Motivation

- In the near future, Big Data is going to touch every business and every person on Earth.
- Currently **less than 0.5%** of all data collected is being analyzed and used. The amount of data collected has been increasing exponentially in the recent years.
- Data now streams through our phones, credit cards, computers, cars, trains and planes thus creating "**The Big Data Revolution**".

Objectives & Goals

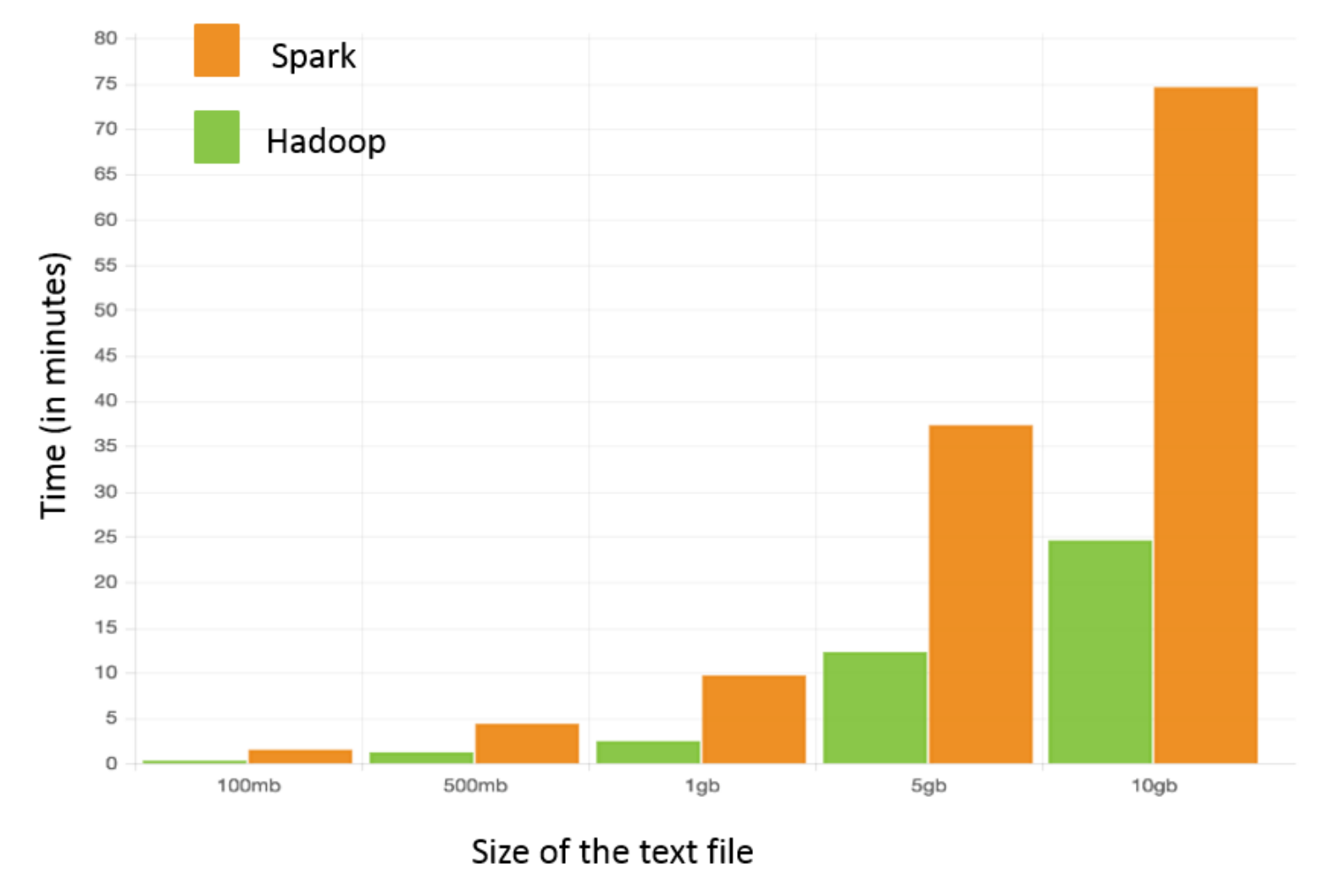
- **Performance** tests and **comparative** studies in different environments :
  - Single Node (Java, Hadoop & Spark environments)
  - Cluster (Hadoop & Spark environments)
- Gathering research data for **feasibility** validation
- **Predict data** through recommendation systems using data mining
- **Awareness** on Big Data at TCU College of Science and Engineering

Problems & Solutions

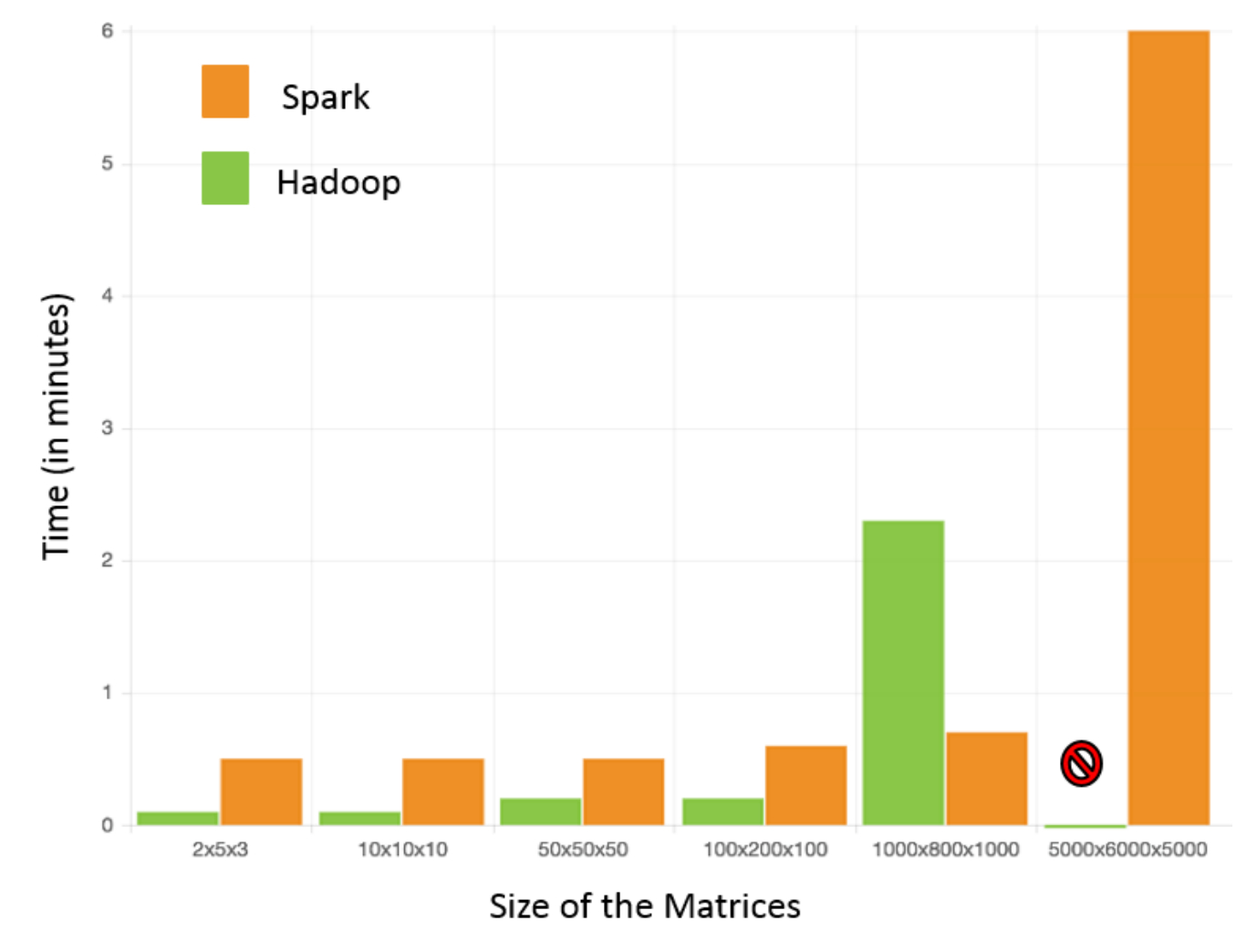
- Java Virtual Machine has a **limited heap memory** to process data
- **Clustering** is not feasible within Eclipse
- Apache Hadoop and Spark solve both these problems by **distributing the data** and its **processing** among different nodes of a cluster

Test Results

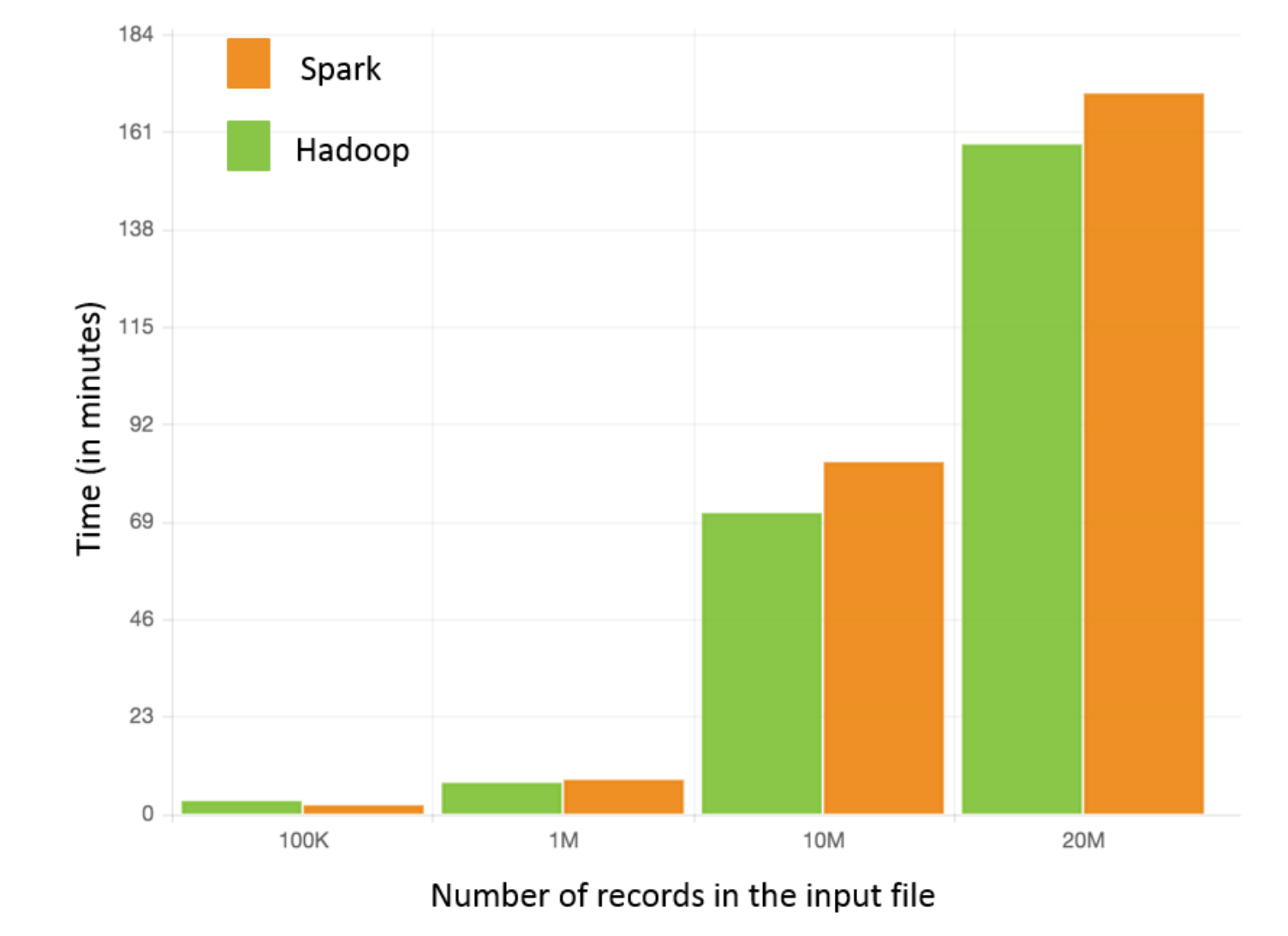
Word Count



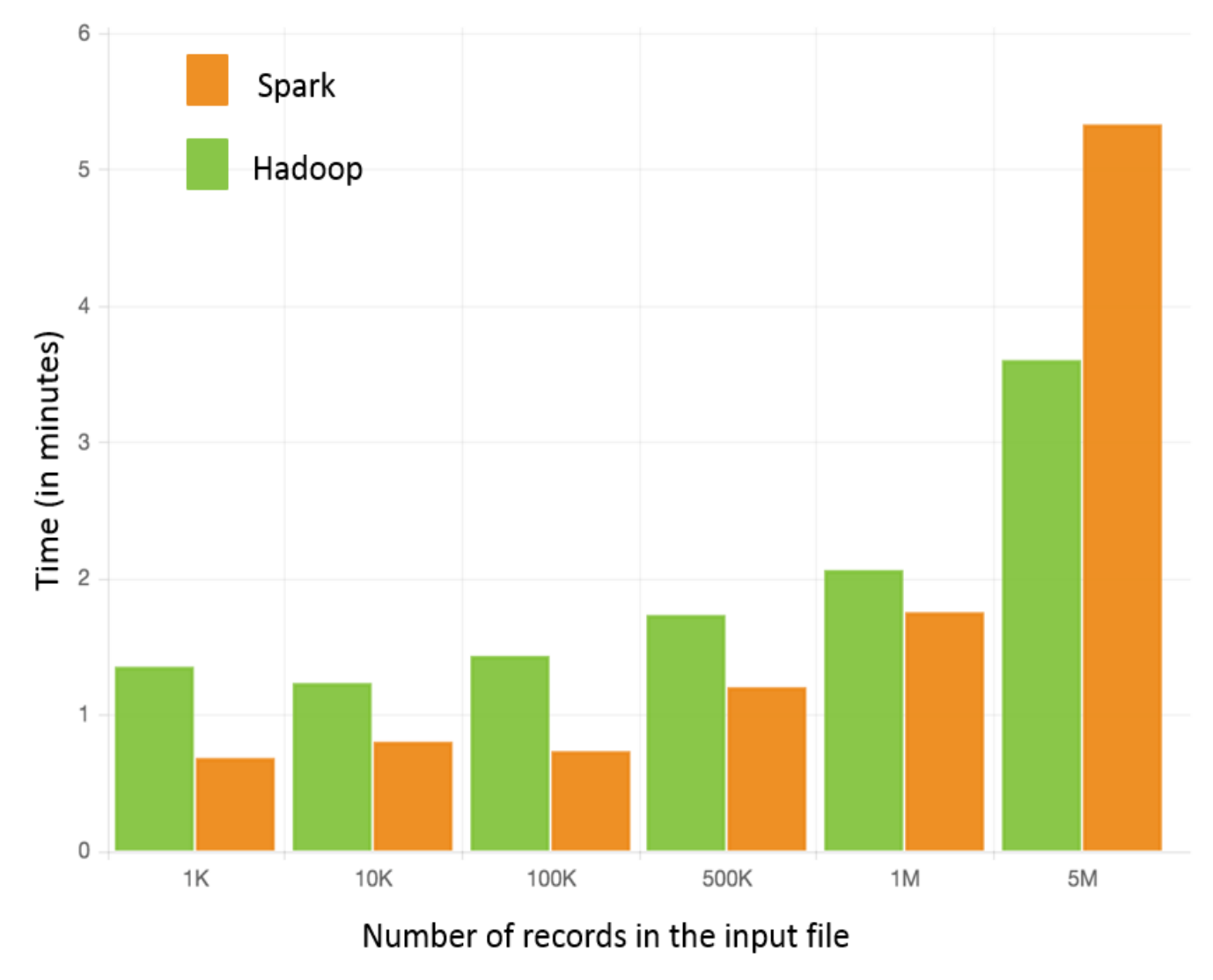
Matrix Multiplication



Movie Recommender



K-Means Clustering



Recommender

Hadoop

Co-occurrence on Mahout

Co-occurrence Matrix (CM)

	The Dark Knight	Star Wars	Casino Royale	The Godfather
The Dark Knight	4	5	3	4
Star Wars	5	5	0	0
Casino Royale	3	0	3	0
The Godfather	4	0	0	4

User Matrix (UM) | Recommendation Matrix (RM)

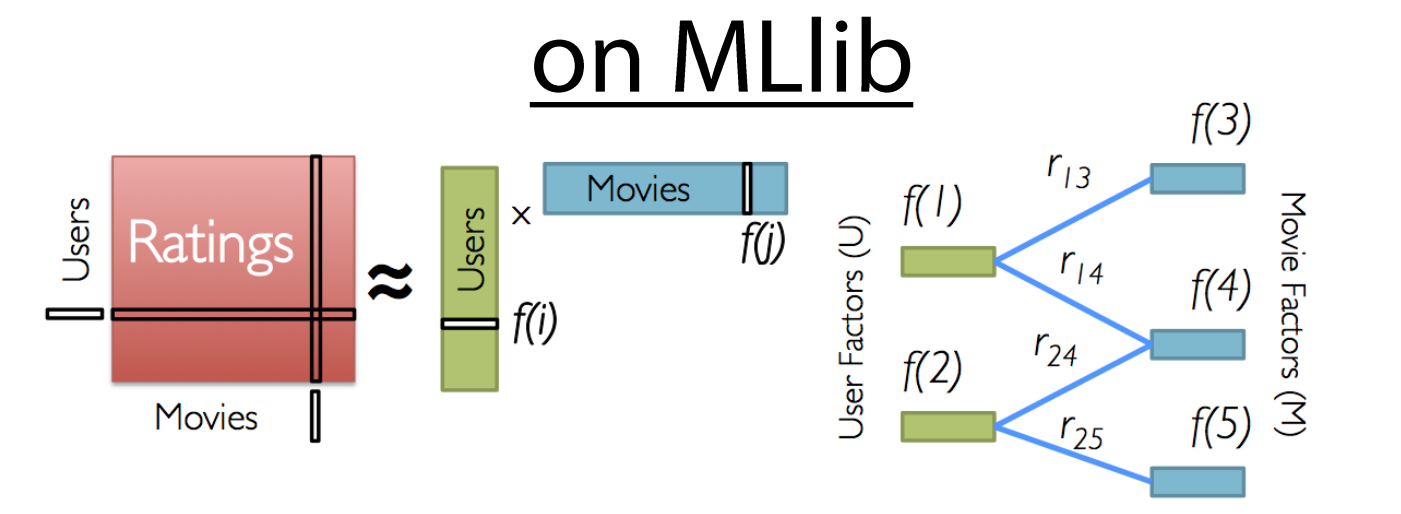
Sushant		R
0	Recommended →	**3**
0		0
1	↔ Already watched	3
0		0

RM = CM * UM

A **co-occurrence matrix** is generated for the movies, then the matrix is multiplied by the **User matrix** and the resulting matrix are the recommendations.

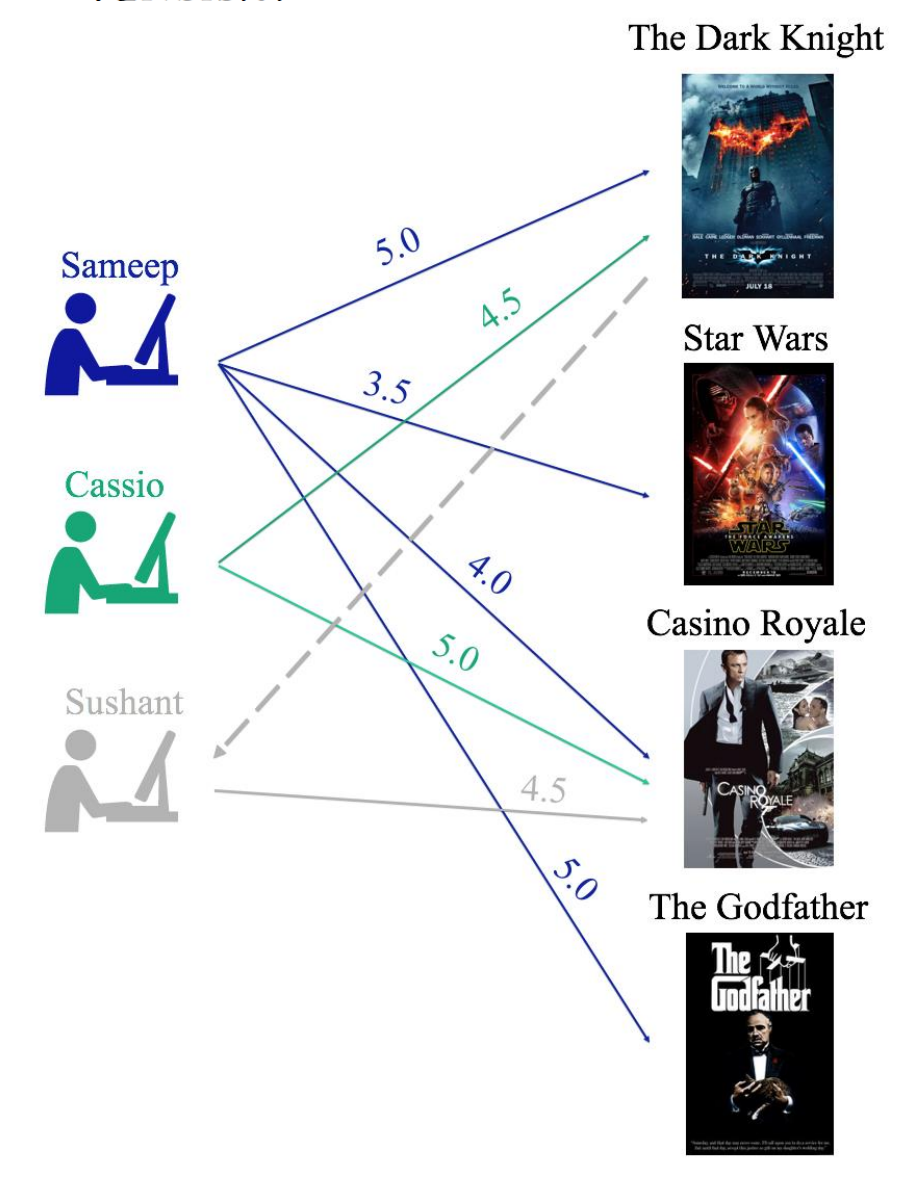
Spark

Alternative Least Square (ALS) on MLlib



Iterate:

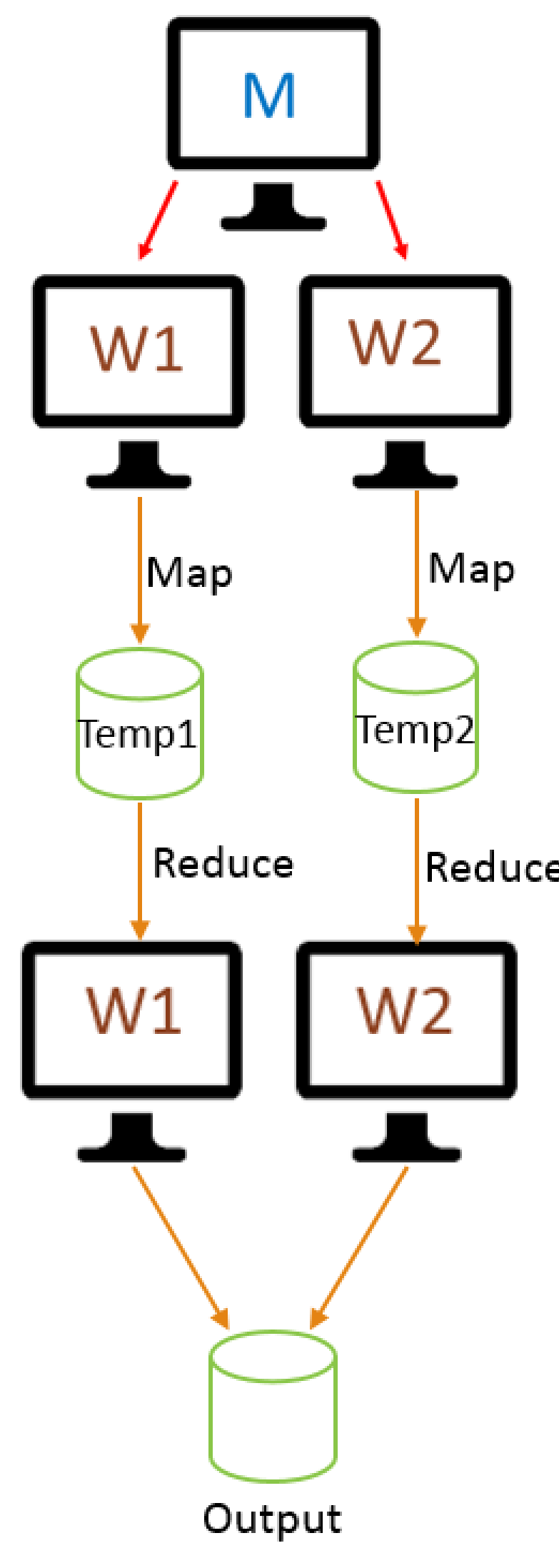
$$f[i] = \arg\min_{w \in \mathbb{R}^d} \sum_{j \in \mathrm{Nbrs}(i)} \left(r_{ij} - w^T f[j]\right)^2 + \lambda \|w\|_2^2$$



Sushant was recommended **The Dark Knight** based on what other users with similar taste also watched (**Collaborative Filtering**).

Hadoop

- **Open-source** software framework
- **Map/Reduce** – Style of data processing, developed by Google, for Big Data
- The go-to framework for **large-scale**, data-intensive deployments
- Uses **HDFS** – **breaks** up input data, stores it on compute nodes allowing **parallel processing** on blocks of data
- Needs 3rd party machine-learning library, **Mahout** for recommendations and clustering, and **Maven** for dependencies



Hadoop Map/Reduce

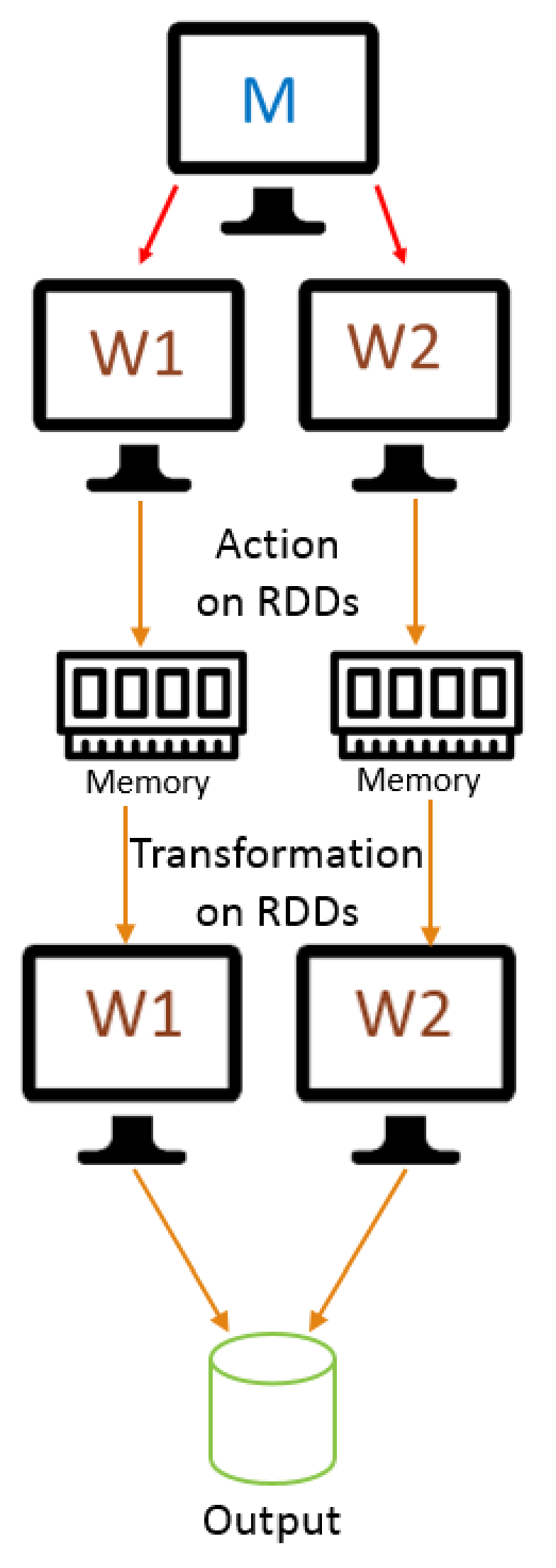
Map Phase:
- Splits input data-set into independent chunks
- Processed in parallel manner – map tasks
- Framework sorts the output of map tasks

Reduce Phase:
- Takes input from map function
- Performs summary operation
- Output stored in HDFS

HDFS

- Hadoop Distributed File System.
- Java-based file system that provides scalable and reliable data storage, distributes data among a cluster of nodes in blocks.
- Highly fault-tolerant, suitable for large data-sets.

Spark



- **Open source** Big Data processing framework designed to be fast and general-purpose
- Supports **Map** and **Reduce** functions.
- Does not have its own distributed system, can use HDFS or others
- **Lazy** (on demand) evaluation
- **In-memory** storage and computing.
- Offers **APIs** in Scala, Java, Python, SQL
- Built in libraries like **Mllib**
- Uses Apache **Maven** to build dependencies

Resilient Distributed Dataset (RDD)

- Special data structure for Spark, a read-only multi-set of data items distributed over a cluster of machines
- Availability of RDDs facilitates the implementation of interactive data analysis as well as iterative algorithms
- Represents an immutable, partitioned collection of elements

Conclusion and Future Work

- During this research, we gained vast amount of new knowledge on Big Data: both its significance and dependability in the real world. We performed a comparative analysis based on 4 tests to compare the efficiencies of Hadoop and Spark framework. Even though both have their own strengths and weaknesses, our test results indicate that while Hadoop processes larger files efficiently, Spark performs complex mathematical calculations much quicker than Hadoop.
- That being said, Spark is not here to replace Hadoop or vice versa because of their ability to work with each other to solve complex problems.
- For future work, we would suggest to improve the code of our test programs to make them better in terms of efficiency and performance on both frameworks.

References

- **Apache Hadoop :** http://hadoop.apache.org/
- **Apache Mahout :** http://mahout.apache.org/
- **Apache Mllib :** http://spark.apache.org/mllib/
- **Apache Maven :** https://maven.apache.org/
- **Apache Spark :** http://spark.apache.org/
- Ellen Friedman, Robin Anil, Sean Owen and Ted Dunning. *Mahout in Action*. Greenwich: Manning Publications Co. , 2011.
- Nick Pentreath. *Machine Learning with Spark*. Birmingham: Packt, 2015.

Acknowledgements

We would like to express our appreciation to Dr. Antonio Sanchez, who provided insight and expertise throughout the project. We would also like to pay our gratitude to Dr. Donnell Payne for motivating our team and helping us keep up with our schedule during the course of this research.