# Assessing the Performance Impact of Scheduling Policies in Spark

Authors: MaryAshley Etefia, Henry Santer, Faculty Advisor: Ningfang Mi[1], Mentor: Zhengyu Yang[1]
Email: mve45340@uga.edu

## Background

→ Apache Spark is a real-time data processing framework.
→ Scheduling policies tell Spark when resources will be distributed and where resources will go.
→ Resilient distributed datasets (RDD) are fixed storage spaces operating in Spark applications.[1]
→ RDD dependency is the relationship between RDDs within the stages of a Spark application.[1]

## Goals

✓ To identify which workloads have a wide or narrow RDD dependency.

✓ To determine which scheduling policy is optimal for workloads that are running simultaneously.

## Procedure

1. Set up an Apache Spark cluster with 1 master and 4 worker nodes.
2. Submitted WordCount, K-Means, and PageRank applications separately and observed their DAGs.
3. I configured FAIR and FIFO scheduling policies and those same three workloads simultaneously for each scheduling policy.

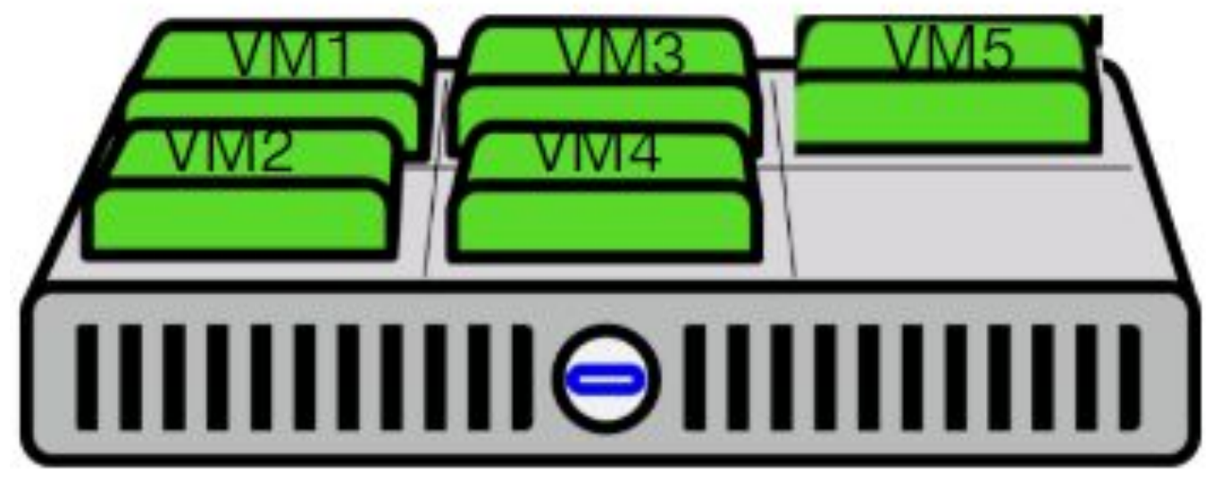


Figure #1

## Model + Graphs

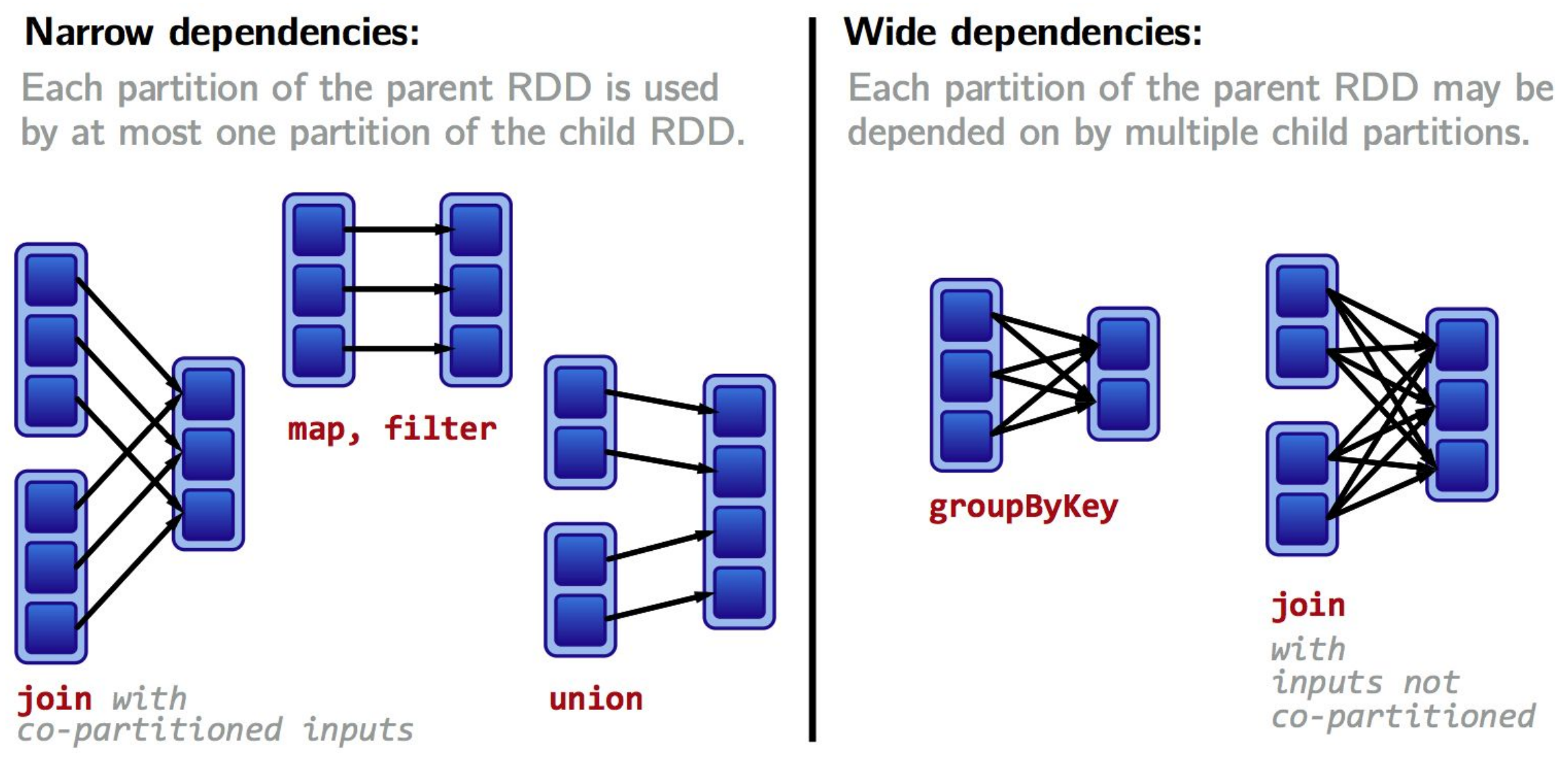
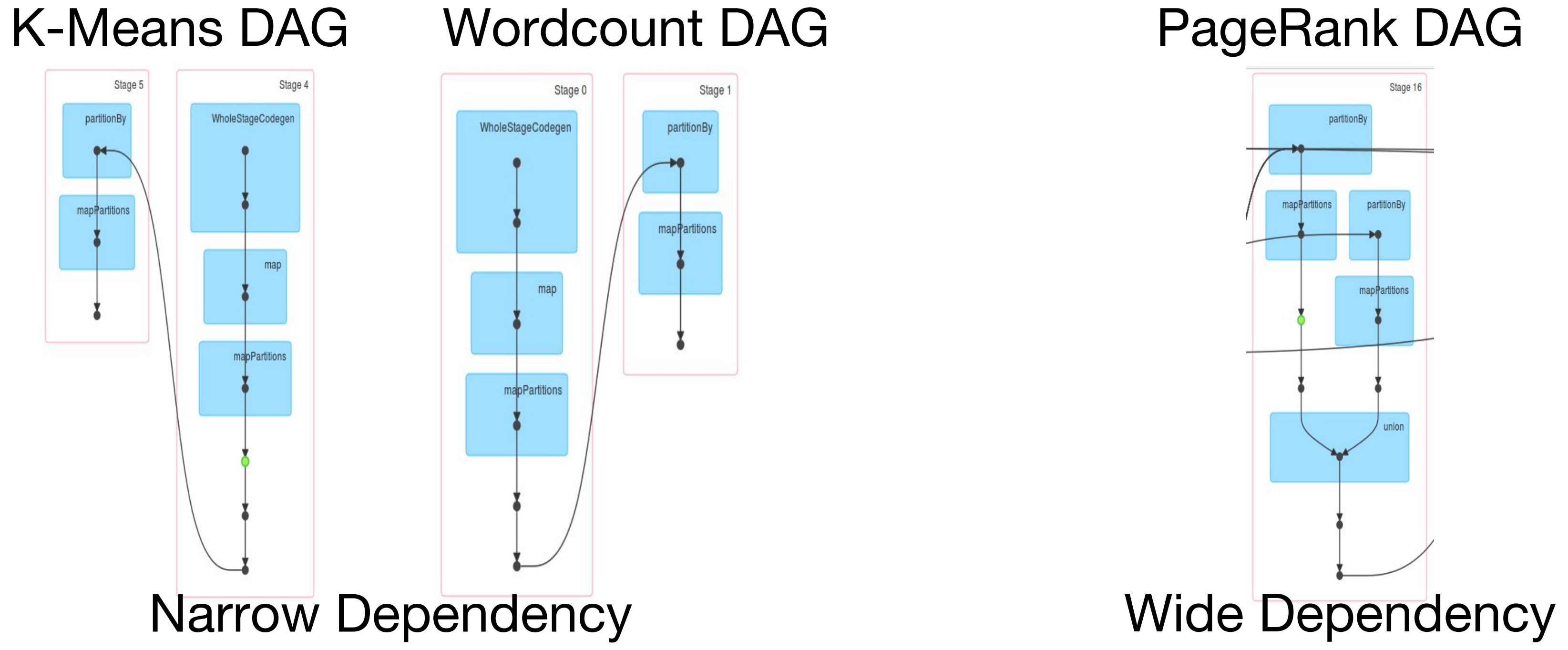
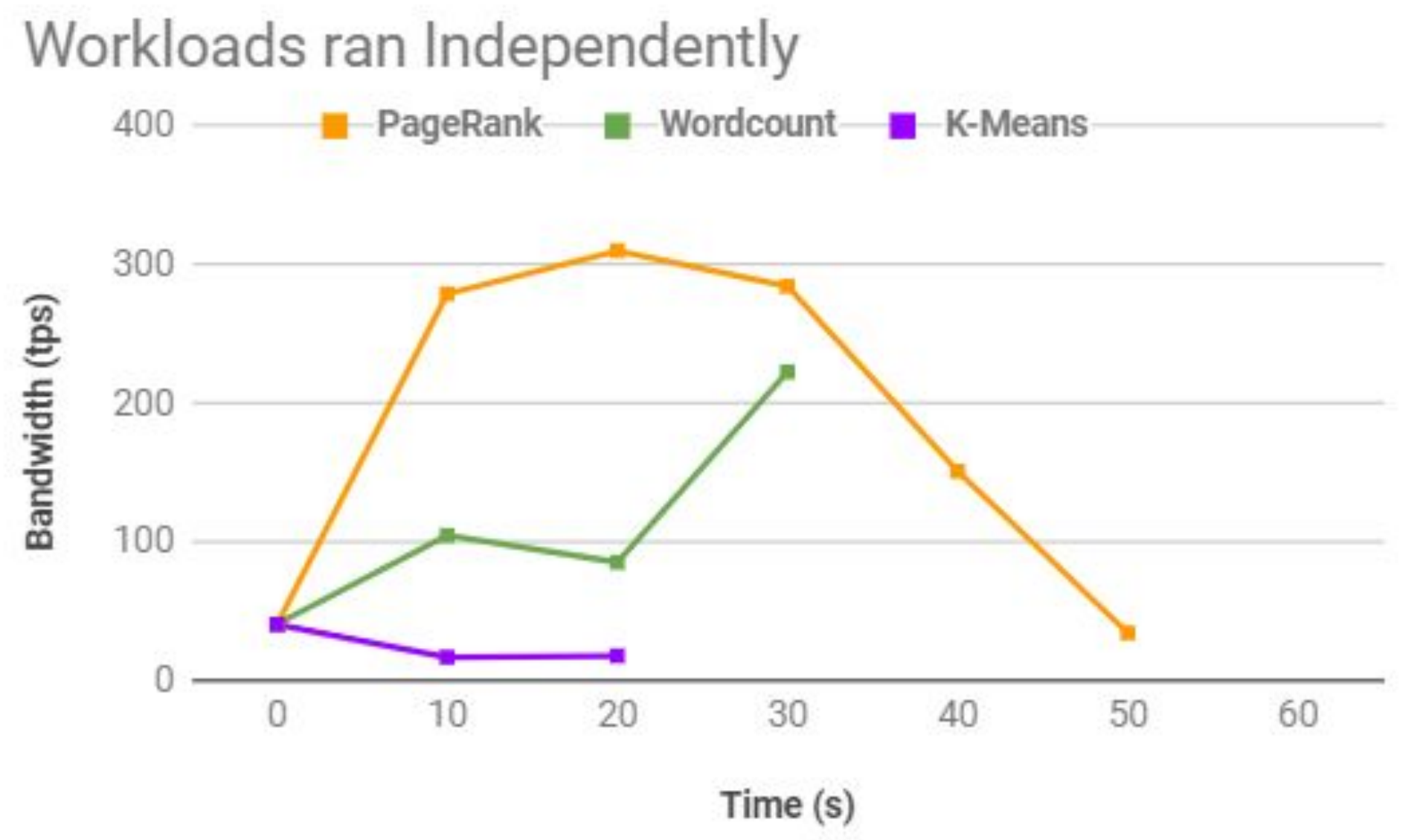

Narrow dependencies:
Each partition of the parent RDD is used by at most one partition of the child RDD.
map, filter
join with co-partitioned inputs
union
Wide dependencies:
Each partition of the parent RDD may be depended on by multiple child partitions.
groupByKey
join with inputs not co-partitioned

Figure #2

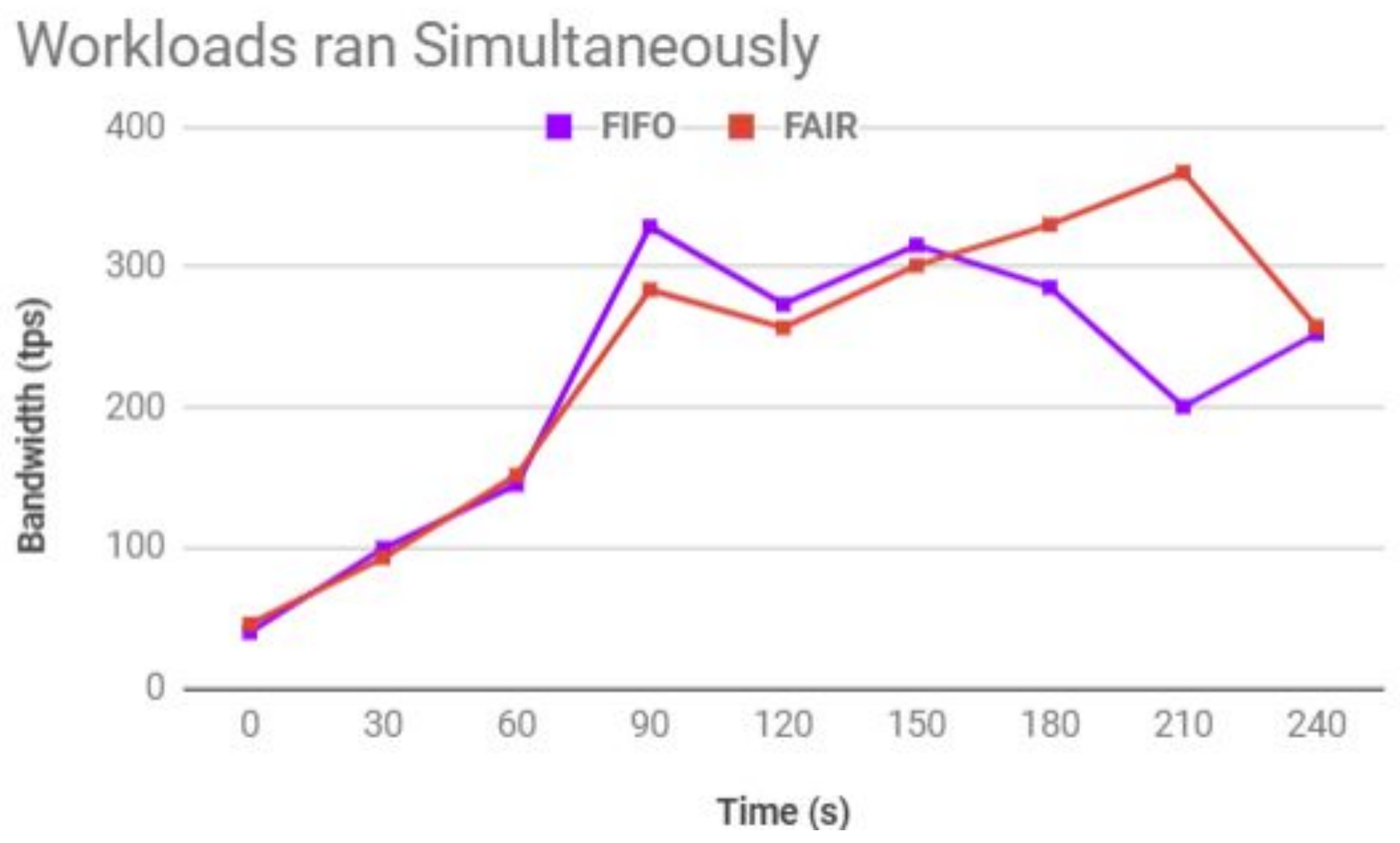


Narrow Dependency

Wide Dependency



Workloads ran Independently



Workloads ran Simultaneously

## Conclusion

After observing each of their Directed Acyclic Graphs (DAG), I found that PageRank has a wide RDD dependency and K-Means and Wordcount have a narrow RDD dependency. PageRank has a wide dependency because it is evident that data is shuffled across multiple RDDs. Also, I have concluded that Spark's FAIR scheduling policy is optimal for a combination of workloads such as PageRank, K-Means, and Wordcount, since the average bandwidth for the FAIR policy was slightly higher than the FIFO policy.

## Future Works

Scaling the performance results of low-level Spark workloads to datasets handled by data centers could be investigated in the future. Potential researchers could also investigate some aspects of scaling like vertical and horizontal sharding.

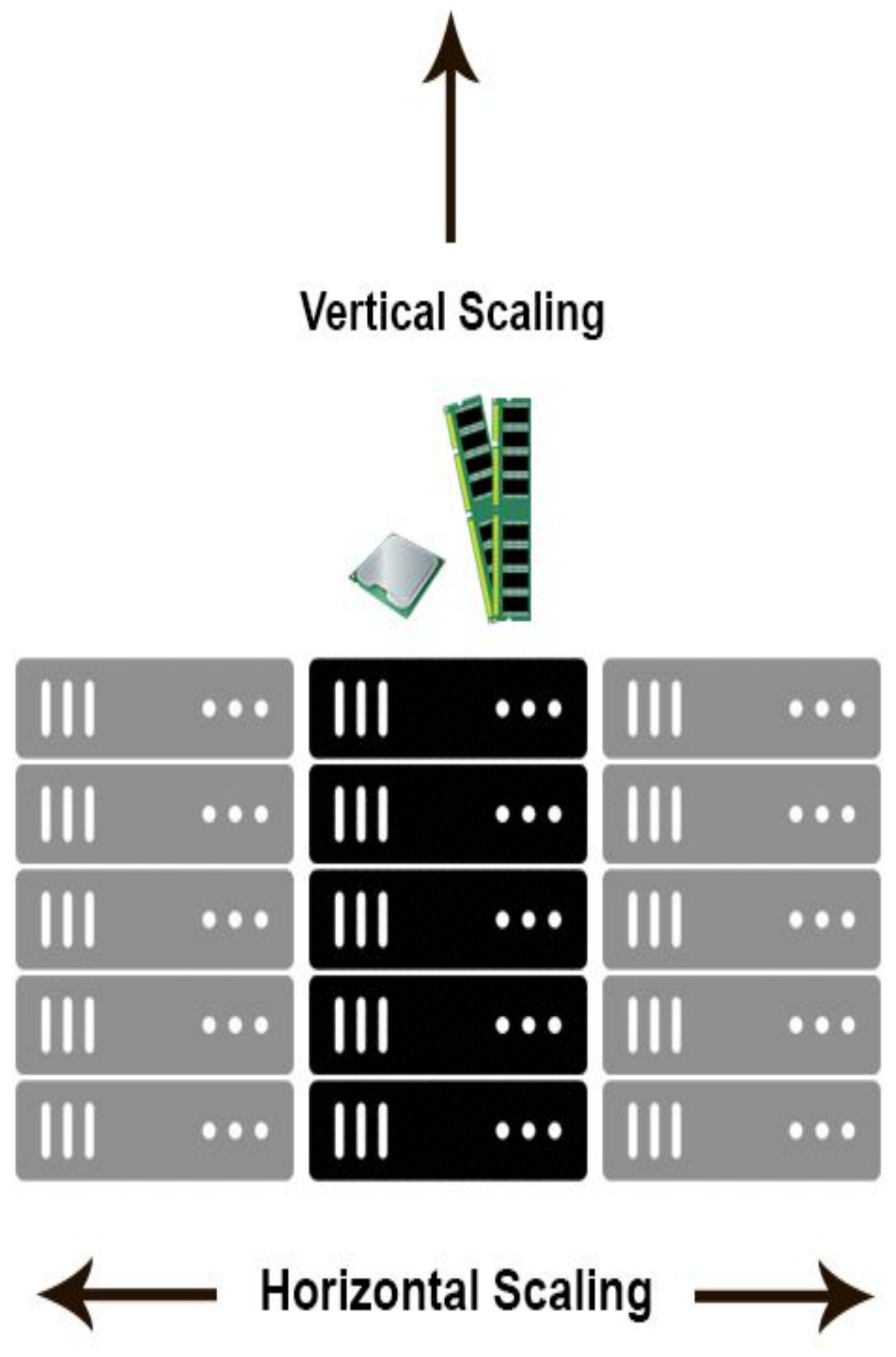


Figure #3

## Acknowledgements

NSF REU Award 1559894
Dr. Ningfang Mi
Dr. Zhengyu Yang

Mahsa Bayati
Danlin Jia
Henry Santer

References: 1. Gao, Han et al. "AutoPath: Harnessing Parallel Execution Paths for Efficient Resource Allocation in Multi-Stage Big Data Frameworks." *2017 26th International Conference on Computer Communication and Networks (ICCCN)* (2017): 1-9.