1 Summary

Apache Spark is the work of the open-source community led by researchers at UC Berkeley. It aims to simplify the complex nature of scalable data processing stemming from sequences of myriad different processing steps with different computing systems. To accomplish this, Spark introduces a unified programming model and engine for big data applications. The Spark programming model is similar to MapReduce but extends it with an abstraction called "Resilient Distributed Datasets," or RDDs. This is the basis behind Spark's claim that it uses a unified engine.

2 Strengths of the paper

Industry certainly sees the impact of this work. While it's used by most of the top companies, including Netflix, Yahoo, and eBay, the team that started this project at Berkeley founded the successful startup, Databricks, in 2013. Furthermore, it has become the largest open source community in big data (>1000 contributors from 250+ organizations). It's always inspiring to see projects born in academia "growing up" to become profitable and widely used as they mature.
The main benefits of Spark are speed, ease of use, and a unified engine to simplify previously complex workflows tied to big, messy data and therefore, boost developer productivity. The underlying technologies and architecture are therefore crucial for those who wish to understand the nuances of the large scale workflow implemented by its myriad users.

3 Weakness of the paper

The authors point out the main limitation of RDDs is increased latency due to synchronization in each communication step. However, they then note that this latency is often not a factor. Discussing an example where this latency would be a factor and how Spark would deal with it may have reinforced the point made.

4 Future work opportunities

§ Ongoing work mentions the Dataframes declarative API that most Spark programmers use as their standard abstraction for passing data. The process of achieving tight integration of this API with Spark is something I'd be interested in learning more abut.

5 Extra

Here's an awesome analogy[1] about Spark, Hadoop and MapReduce: *Single cook cooking an entree is regular computing. Hadoop is multiple cooks cooking an entree into pieces and letting each cook cook her piece. Each cook has a separate stove and a food shelf. The first cook cooks the meat, the second cook cooks the sauce. This phase is called "Map". A the end the main cook assembles the complete entree. This is called "Reduce". For Hadoop the cooks are not allowed to keep things on the stove between operations. Each time you make a particular operation, the cook puts results on the shelf. This slows things down. For Spark the cooks are allowed to keep things on the stove between operations. This speeds things up.*

[1] https://www.quora.com/What-is-Apache-Spark-and-how-does-it-compare-to-Hadoop-MapReduce