

# Big Data Systems - Survey (IN4331)

SB Ramalingam Santhanakrishnan (4740270)  
S.B.RamalingamSanthanakrishnan@student.tudelft.nl

Konrad Kleeberger (4748476)  
k.kleeberger@student.tudelft.nl

Zhen Sun (4733983)  
Z.Sun-1@student.tudelft.nl

Chuting Zhu (4728572)  
C.Zhu-3@student.tudelft.nl

## ABSTRACT

Graph processing is a subset of big data processing and specialized systems are being built and improved upon to handle ever increasing demands from various domains, primary on the problems of social networks and search. The survey paper pivots around a benchmark metric called COST (cost of single thread), targeted at the graph analysis systems like GraphX, evaluating its necessity for the enormous level of scalability support for a few select problems, given that they are orders of magnitude faster, when executed on a single core commodity machine with appropriate optimizations. We also take a look at research on graph systems, published after the establishment of COST and conclude on what a decision maker should consider while evaluating such systems in general.

## 1. INTRODUCTION

(author?) defined three components for distributed systems to be scalable. The distance between the farthest nodes, the amount of resources and users it can handle and the number of organizations which want to administrate the system. Big Data can be seen as a numeric scalability problem as an increasing amount of data has to be handled without a compromising loss of performance. A systems can either be scaled vertically which means to add more resources like storage, RAM or CPU to an existing machine or scale horizontally as distributing the load over multiple machines.

While vertical scaling can reach a good performance on solving the big data problems, it has some drawbacks which are difficult to be overcome, such as the great risk of one-point failure and limited upgradeability in the future. Therefore, horizontal scaling becomes a great choice when some user request the scalability without too much investment. Dealing with the problem appears in parallelizing the computation, distributing the data and handling failures is a great obstacle for many researchers.

## 2. BIG DATA SYSTEMS AND THEIR SCALABILITY DEMANDS

This section discusses different Big data systems and the property of their design which affect scalability.

### 2.1 Batch systems

The first proposal of a big data system came from Google as they recognized that their developers constantly solved the same problems over and over again when dealing with large amount of raw data like data distribution, fault toler-

ance or load balancing. Therefore they implemented with MapReduce a framework which solves this problems and allows developers to focus on the problem solving on a higher level [1]. It is based on a distributed file system and uses the primitive steps of map and reduce as a programming model. Yahoo created with Hadoop an open source implementation of the Google ecosystem this also includes the MapReduce paper.

MapReduce is a batch processing system as it uses bounded or finite data sets. A map reduce process has to come to an end.

Google used it for indexing the documents of their web crawler which were data sets of around 20 terabyte. - stores everything to disk - optimizations blabla - for what kind of problems is it most suitable?

- how does it scale? - what parts make is bad/good for scaling

### 2.2 Streaming systems

### 2.3 Graph systems

With Pregel [2] Google introduced a framework for graph processing which handles basic distributed computing challenges and provides a programming model which is suitable for most graph algorithms. The algorithms are centered around a vertex and proceed in rounds which allows synchronization and easier programming in distributed systems. An open source implementation of the paper is Apache Giraph which was developed by Yahoo.

- how does it scale - what part makes it bad/good for scaling

## 3. LIMITATIONS OF PARALLELIZING

It is a known fact from Amdahl's law that parallelizing

## 4. CONTRIBUTION AND CORE CLAIMS OF THE CHOSEN PAPER

## 5. RESEARCH AFTER COST

## 6. FINAL TAKEAWAYS

## References

- [1] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

- [2] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 135–146. ACM, 2010.
- [3] B. C. Neuman. Scale in distributed systems. In *Readings in Distributed Computing Systems*, pages 463–489. IEEE Computer Society Press.