Journal of Network and Computer Applications 46 (2014) 1–25

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

A comprehensive view of Hadoop research—A systematic literature review

Ivanilton Polato [a,b,*], Reginaldo Ré [b], Alfredo Goldman [a], Fabio Kon [a]

[a] Department of Computer Science, University of São Paulo, São Paulo, Brazil
[b] Department of Computer Science, Federal University of Technology - Paraná, Campo Mourão, Brazil

ARTICLE INFO

Article history:
Received 9 January 2014
Received in revised form
25 June 2014
Accepted 21 July 2014
Available online 1 August 2014

Keywords:
Systematic literature review
Apache Hadoop
MapReduce
HDFS
Survey

ABSTRACT

Context: In recent years, the valuable knowledge that can be retrieved from petabyte scale datasets – known as Big Data – led to the development of solutions to process information based on parallel and distributed computing. Lately, Apache Hadoop has attracted strong attention due to its applicability to Big Data processing. *Problem:* The support of Hadoop by the research community has provided the development of new features to the framework. Recently, the number of publications in journals and conferences about Hadoop has increased consistently, which makes it difficult for researchers to comprehend the full body of research and areas that require further investigation. *Solution:* We conducted a systematic literature review to assess research contributions to Apache Hadoop. Our objective was to identify gaps, providing motivation for new research, and outline collaborations to Apache Hadoop and its ecosystem, classifying and quantifying the main topics addressed in the literature. *Results:* Our analysis led to some relevant conclusions: many interesting solutions developed in the studies were never incorporated into the framework; most publications lack sufficient formal documentation of the experiments conducted by authors, hindering their reproducibility; finally, the systematic review presented in this paper demonstrates that Hadoop has evolved into a solid platform to process large datasets, but we were able to spot promising areas and suggest topics for future research within the framework.

Contents

* Corresponding author at: Department of Computer Science, Federal University of Technology - Paraná, Campo Mourão, Brazil. Tel./fax: +55 44 3518 1449.
E-mail address: ipolato@utfpr.edu.br (I. Polato).

http://dx.doi.org/10.1016/j.jnca.2014.07.022
1084-8045/

1. Introduction

One of the largest technological challenges in software systems research today is to provide mechanisms for storage, manipulation, and information retrieval on large amounts of data. Web services and social media produce together an impressive amount of data, reaching the scale of petabytes daily (Facebook, 2012). These data may contain valuable information, which sometimes is not properly explored by existing systems. Most of this data is stored in a non-structured manner, using different languages and formats, which, in many cases, are incompatible (Bakshi, 2012; Stonebraker et al., 2010).

Take, for instance, Facebook, which initially used relational database management systems (DBMS) to store its data. Due to the increasingly large volume of information generated on a daily basis (from a 15TB dataset in 2007 to a 700TB dataset in 2010) (Thusoo et al., 2010), the use of such infrastructure became impracticable. Specially because, most of its data is unstructured, consisting of logs, posts, photos, and pictures. One of the Facebook's largest clusters holds more than 100 PB of data, processing more than 60,000 queries a day (Facebook, 2012). Having achieved in September 2012 more than 1 billion active users, Facebook may be considered one of the largest and most valuable social networks.

Companies holding large amounts of user data started to be evaluated not just by their applications but also by their datasets, specially the information that can be retrieved from them. Big companies like Google, Facebook and Yahoo! have an aggregate value not only for their provided services but also for the huge amount of information kept. This information can be used for numerous future applications, which may allow, for example, personalized relationships with users.

The “Big Data” (Zikopoulos and Eaton, 2011; White, 2012) term is used to refer to a collection of large datasets that may not be processed using traditional database management tools. Some of the challenges involved when dealing with Big Data goes beyond processing, starting by storage and, later, analysis. Concerning data analysis and Big Data, the need for infrastructures capable of processing large amounts of data, within an acceptable time and on constrained resources, is a significant problem. Plausible solutions make use of parallel and distributed computing. This model of computation has demonstrated to be essential nowadays to extract relevant information from Big Data. Such processing is accomplished using clusters and grids, which use, generally, commodity hardware to aggregate computational capacity at a relatively low cost.

Although parallel and distributed computing may be one of the most promising solutions to store and manipulate Big Data, some of its characteristics may inhibit its use by common users. Data dependency and integrity, cluster load balancing and task scheduling are major concerns when dealing with parallel and distributed computing. Adding the possibility of an almost certain machine failure, the use of these concepts becomes non-trivial to inexperienced programmers. Several frameworks have been released to abstract these characteristics and provide high level solutions to end users (DeWitt et al., 2008; Battré et al., 2010; Malewicz et al., 2010; Isard et al., 2007); some of them were built over programming paradigms, such as MPI and MapReduce.

The MapReduce programming paradigm, now highly used in the context of Big Data, is not new. One of the first uses of this paradigm was on the LISP programming language. It relies basically on two functions, Map and Reduce. The first generates maps based on a given user defined function and the second groups Map outputs together to compute an answer. The paradigm is very useful when dealing with batch programs where data is manipulated in a sequential way. Recently the MapReduce paradigm attracted attention because of its applicability to parallel computing. Google's MapReduce composed initially of the GFS distributed filesystem (Ghemawat et al., 2003) and the implementation of MapReduce (Dean and Ghemawat, 2004, 2008), brought to the fore the use of the simple and consolidated functions Map and Reduce in parallel and distributed computing using Java and C++ libraries. This approach feeds the Reduce function with the Map function results. This enables parallelism since partitioned portions of data may be fed into different instances of Map tasks throughout the cluster. The results are gathered, used as inputs to the Reduce instances, and the computation is accomplished. The great novelty here is that the approach hides from users a lot of the complexity of parallelization and distribution. Users can focus on the functionality of their programs and the framework abstracts the complexity and controls the infrastructure.

Based on this novel approach, Doug Cutting, an employee of Yahoo! at the time, and Mike Cafarella, a professor at University of Michigan, developed Hadoop, later called the Apache Hadoop framework. It is an open source implementation of the Google's MapReduce approach. It uses the same idea from Google's: hiding complexity from users allowing them to focus on programming. Mostly known by its MapReduce implementation, Apache Hadoop also has an ecosystem composed of several applications ranging from data warehousing to a data flow oriented programming language. The Apache Hadoop framework provides solutions to store, manipulate and extract information from Big Data in several ways. The framework has evolved over the last few years and promotes data integrity, replication, scalability, and failure recover in a transparent and easy-to-use way. All these factors have made Apache Hadoop very popular both in academia and in industry.

The early adoption of Hadoop by the research community has provided rapid evolution and development of new features to the framework. Over the last five years, the framework received numerous contributions from researchers, most of them published worldwide in journals and conferences. The large number of publications makes it difficult for researchers to find specific topics or areas that still need further investigation, resulting in a need for an unbiased analysis of the scientific and technological research results obtained in this field in the past years. Thus, this paper is mainly directed to

- Researchers and graduate students willing to carry out new research around Hadoop, which can use the current paper as a guide to which areas have been covered in past research and which areas deserve more attention.