Journal of Computer Sciences and Applications, 2015, Vol. 3, No. 6, 177-180
Available online at http://pubs.sciepub.com/jcsa/3/6/13
© Science and Education Publishing
DOI:10.12691/jcsa-3-6-13

Big Data Processing: Big Challenges and Opportunities

Neelamani Samal, Nilamadhab Mishra[*]

Department of Computer Science & Engineering, Gandhi Institute for Education and Technology, Bhubaneswar, Odisha, India
*Corresponding author: nilamadhab76@gmail.com

Abstract With the rapid growth of emerging applications like social network, semantic web, sensor networks and LBS (Location Based Service) applications, a variety of data to be processed continues to witness a quick increase. Effective management and processing of large-scale data poses an interesting but critical challenge. Recently, big data has attracted a lot of attention from academia, industry as well as government. This paper introduces several big data processing techniques from system and application aspects. First, from the view of cloud data management and big data processing mechanisms, we present the key issues of big data processing, including definition of big data, big data management platform, big data service models, distributed file system, data storage, data virtualization platform and distributed applications. Following the Map Reduce parallel processing framework, we introduce some MapReduce optimization strategies reported in the literature. Finally, we discuss the open issues and challenges, and deeply explore the research directions in the future on big data processing in cloud computing environments.

Keywords: *big data, cloud computing, data management, distributed processing*

Cite This Article: Neelamani Samal, and Nilamadhab Mishra, "Big Data Processing: Big Challenges and Opportunities." *Journal of Computer Sciences and Applications*, vol. 3, no. 6 (2015): 177-180. doi: 10.12691/jcsa-3-6-13.

1. Introduction

Data processing is common part of processes inside every organization. Critical challenges of these days came with is well known character defined mostly for big data – velocity, variety, and volume. Even new technologies appeared, traditional data sources and processes require variety of different approaches. Current research and development in the field of data processing accommodates knowledge from different areas including algorithms, hardware, software, engineering, and social issues. Applications usually combine high-performance computers for computation, high-performance databases and cloud servers for data storage and management, and desktop computers for human-computer interaction Source for processing often come from models or observations based on different scientific, engineering, social, and cyber applications.

Massive sets of data in pet bytes (1015) or terabytes (1012) are available for analytical and transactional processing. Main application areas are medicine, large sensor networks, social networks, and other industrial bases sources of data. The common factor is existence of connections between data which on the other hand leads to increased complexity of datasets. In our paper we will defined some of our observations and selected experimental results to describe basic challenges of data processing. We are dealing with three different approaches: relational, semantic, and graph based. All of these require accommodation of different techniques. Section 2 reviews the architecture and the key concepts of big data processing.

Sections 3 and 4 present the classification of major distributed applications and optimization methods of the Map Reduce framework while Section 5 discusses several open issues and future challenges. Finally, Section 6 concludes this paper.

2. Big Data Management System

According to a recent survey by Gartner in 2010g, 47% of survey respondents rank data growth in their top three challenges, followed by system performance and scalability at 37%, and network congestion and connectivity architecture at 36%. Many researchers have suggested that commercial Data Base Management Systems (DBMSs) are not suitable for processing extremely big data. Classic architecture's potential bottleneck is the database server while facing peak workloads. One database server has restriction of scalability and cost [2], which are two important goals of big data processing. In order to adapt various large data processing models.

D. Kossmann et al.[3] presented four different architectures based on classic multi-tier database application architecture which includes partitioning, replication, distributed control and caching architecture. It is clear that alternative providers have different business models and target different kinds of applications: Google seems to be more interested in small applications with light workload whereas Azure is currently the most affordable service for medium to large services. Most of recent cloud service providers are utilizing hybrid architecture that is capable of satisfying their actual service requirements. In this section, we mainly discuss big data architecture from four key aspects: big data service models, distributed file system, non-structural and semi-structured data storage and data virtualization platform.

2.1. Big Data Service Model

As we all known, cloud computing is a kind of information and communication[9] technology, which delivers valuable resources to people as a service, such as Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS)[4]. There are several leading Information Technology (IT) solution providers that offer these services to the customers. Now, as the concept of the big data came

up, cloud computing service model is gradually transferring into big data service model, which are DaaS (Database as a Service), AaaS (Analysis as a Service) and BDaaS (Big data as a Service). The detailed descriptions are as follows: Database as a Service means that database services are available applications deployed in any execution[8] environment, including on a PaaS. But in the big data context, these would optimally be scale-out architectures such as No SQL data stress and in-memory databases.

Analysis as a Service would be more familiar with interacting with an analytics platform on a higher abstraction level. They would typically execute scripts and queries that data scientists or programmers developed for them.

Big data as a Service coupled with Big Data platforms are for users that need to customize or create new big data stacks, however, readily available solutions do not yet exist. Users must first acquire the necessary cloud computing infrastructure, and manually install the big data processing software. For complex distributed services, this can be a daunting challenge.

2.2. Distributed File System

Google File System (GFS) is a chunk-based distributed file system that supports fault-tolerance by data partitioning and replication. As an underlying storage layer of Google's cloud computing platform, it is used to read input and store output of Map Reduce. Similarly, Hadoop also has a distributed file system as its data storage layer called Hadoop Distributed File System (HDFS), which is an open-source counterpart of GFS. GFS and HDFS are user level file systems that do not implement POSIX semantics and heavily optimized for the case of large files (measured in gigabytes). Amazon Simple Storage Service (S3) is an online public storage web service offered by Amazon Web Services. This file system is targeted at clusters hosted on the Amazon Elastic Compute Cloud server-on-demand infrastructure. S3 aims to provide scalability, high availability, and low latency at commodity costs. ES[11] is an elastic storage system of epic, which is designed to support both functionalities within the same storage. The system provides efficient data loading from different sources, flexible data partitioning scheme, index and parallel sequential scan. In addition, there are several general file systems that have not to be ad- dressed such as Moose File System (MFS), Kosmos Distributed File system (KFS)

2.3. Non-structural and Semi-structured Data Storage

With the success of the Web 2.0, most IT companies increasingly need to store and analyze the ever growing data, such as search logs, crawled web content and click streams collected from a variety of web services, which are usually in the range of petabytes. However, web data sets are usually non-relational or less structured and processing such semi-structured data sets at scale poses another challenge. Moreover, simple distributed file systems mentioned above cannot satisfy service providers like Google, Yahoo!, Microsoft and Amazon. All providers have their purpose to serve potential users and own their relevant state-of-the-art of big data management systems in the cloud environment. Bigtable [12] is a distributed storage system of Google for managing structured data that is designed to scale to a very large size (petabytes of data) across thousands of commodity servers. Big table does not support a full relational data model. However, it provides clients with a simple data model that supports dynamic control over data layout and format. PNUTS [13] is a massive scale hosted database system designed to support Yahoo! web applications.

2.4. Data Virtualization Platform

Data virtualization describes the process of abstracting disparate systems. It can be described as conceptual building of abstract layers of resources. In short, big data and cloud computing refer to a convergence of technologies and trends that are making IT infrastructures and applications more dynamic, more modular and more consumable. Currently, the technology of constructing virtualization platform is just in the primary phase, which mainly depends on the cloud data center integration technology.

3. Distributed Applications

In this age of data explosion, parallel processing is essential to perform a massive volume of data in a timely manner. In contrast, the use of distributed techniques and algorithms is the key to achieve better scalability and performance in processing big data. At present, there are a lot of popular parallel and distributed processing models, including MPI, General Purpose GPU (GPGPU), MapReduce and MapReduce-like. We will focus on the last two processing models.

3.1. MapReduce

MapReduce proposed by Google, is a very popular big data processing model that has rapidly been studied and applied by both industry and academia.[7] MapReduce has two major advantages: it hide details related to the data storage, distribution, replication, load balancing and so on. Furthermore, it is so simple that programmers only specify two functions, which are map function and reduce function. We divided existing MapReduce applications into three categories: partitioning sub-space, de- composing sub-processes and approximate overlapping calculations. While MapReduce is referred to as a new approach of processing big data in cloud computing environments, it is also criticized as a "major step backwards" compared with DBMS. As the debate continues, the final result shows that neither of them is good at what the other does well, and the two technologies are complementary.19 Recently, some DBMS vendors have integrated MapReduce front-ends into their systems including Aster, HadoopDB [14], Greenplum [15]. Mostly of those are still database, which simply provide a MapReduce front-end to a DBMS. HadoopDB is a hybrid system which efficiently takes the best features from the scalability of MapReduce and the performance of DBMS. Lately, J. Dittrich et al. proposed a new type of system named Hadoop++ which indicates that HadoopDB has also severe drawbacks, including forcing user to use DBMS, changing the interface to SQL and so on.

3.2. MapReduce-like

Many programmers feel uncomfortable with the MapReduce framework and prefer to use SQL as a high-level declarative language. Several projects have been developed to ease the task of programmers and provide high-level declarative interfaces on top of the MapReduce framework. The declarative query languages allow query independence from program logics, reuse of the queries and automatic query optimization features like SQL does for DBMS. We call them the MapReduce-like system. The Apache Pig [16] project is designed as an engine for executing data flows in parallel on Hadoop. It uses a language, called Pig Latin to express these data flows. It is built on top of Hadoop framework, and its usage requires no modification to Hadoop. The Apache Hive project is an open-source data warehousing solution built by the Facebook Data Infrastructure Team. It supports ad-hoc queries with an SQL- like query language called HiveQL. In recent two years, it has emerged some new distributed data processing systems, and even called beyond MapReduce. However, in essence these are all MapReduce's further improvements and outspreads.

3.3. Application Challenge

As we all known, deploying big data applications on cloud environment is not a trivial or straightforward task. We need to exploit the cloud computing methods to process more areas of big data. There are several important classes of existing data processing and applications that seem to be more compelling with cloud environments and contribute further to its momentum in the near future, such as: Complex Multi-media Data: In the new cloud based multimedia-computing paradigm, users store and process their multimedia application data in a distributed manner, eliminating full installation of the media application software. Multimedia processing in the context of cloud environments imposes great heterogeneity challenges in content-based multimedia retrieval system,38 distributed complicated data processing, high cloud QoS support, media cloud transport protocol, media cloud overlay network and media cloud security, P2P cloud for multimedia services, and so on. Physical and Virtual Worlds Data: The power of people interacting with people in an online setting has driven the success or failure of many companies in the internet space. There are also many difficulties such as how to organize big data storage, and whether process it on real world or virtual world. We need to present a new architecture and implementation of a virtual cloud to fuse of cloud computing and virtual worlds. The large-scale of virtualized resources also need to be processed effectively and efficiently. Mobile Cloud Data Analytics: Smart phones and tablet remarkably started to carry sensors like GPS, Camera and Bluetooth etc. People and devices are all loosely connected and trillions of such connected components will generate a huge data ocean. They are generally relying on large datasets which is difficult to be stored on small devices with limited computing resources. Hence, these large datasets are more conveniently to be hosted in large datacenters and accessed through the cloud on their demand. Besides, dynamic indexing, analyzing and querying large volumes of high-dimensional spatial big data are major challenges.

4. MapReduce Optimization

Previous works have shown that MapReduce systems are inefficient in utilizing com puting resources. In this section, we present details of approaches about improving the performance of processing big data with MapReduce.

4.1. Data Transfer Bottlenecks

It is a big challenge that cloud users must consider how to minimize the cost of data transmission. Consequently, researchers have begun to propose variety of approaches. Map-Reduce-Merge[17] is a new model that adds a Merge phase after Re- duce phase that combines two reduce outputs from two different MapReduce jobs into one, which can efficiently merge data that is already partitioned and sorted (or hashed) by Map and Reduce modules. Map-Join-Reduce [17] is a system that extends and improves MapReduce runtime framework by adding Join stage before Reduce stage to perform complex data analysis [10] tasks on large clusters. The authors presented a new data processing strategy which runs filtering-join aggregation tasks with two consecutive MapReduce jobs. It adopts one-to-many shuffling [1] scheme to avoid frequent check pointing and shuffling of intermediate results. Moreover, dif- ferent jobs often perform similar work, thus sharing similar work reduces overall amount of data transfer between jobs. MRShare [18] is a sharing framework proposed by T. Nykiel et al. that transforms a batch of queries into a new batch that can be executed more efficiently by Merging jobs into groups and evaluating each group as a single query.

4.2. Index Optimization

Many researchers have implemented the traditional and optimized index structures on MapReduce to obtain better performance. In [19], T. Liu et al. built hybrid spill trees in parallel and implemented a scalable image searching algorithm which can be used efficiently to find near duplicates among over billions of images using MapReduce. However, the tree-based approaches have some problems. They did not scale due to traditional top-down search that overloaded the nodes near the tree root, and failed to provide full decentralization. Whereas Voronoi based index [20] made clusters highly scalable by its loose coupling and shared nothing architecture. Till now, Voronoi based index cannot process multidimensional data. Hence, the index structure which is simple, scalable and well be used for distributed processing mode is a best choice for the effective store and processing of the data. Later, Menonet al., presented a novel parallel algorithm for constructing suffix array and BWT of a sequence leveraging the unique features of MapReduce and reduced the end to end runtime from hours to mere minutes. [21] There are also some papers adapting inverted index, which is a simple but practical index structure and appropriate for MapReduce to process big data, such as in [22] etc. We did a large of research on large-scale spatial data environment and designed a distributed inverted grid index by combining inverted index and spatial grid partition with MapReduce model, which is simple, dynamic, scalable and fits for processing high dimensional spatial data.[23] While most kinds of large data are high dimensional, so in [24], J.Wang et al. designed a new system, epic, in which different types of indexes were built to provide efficient query processing for different applications.

4.3. Iterative Optimization

Classic parallel applications are developed using message passing runtimes such as MPI (Message Passing Interface) and PVM (Parallel Virtual Machine), where par allel algorithms are developed using above techniques to utilize the rich set of communication and synchronization constructs offered which are to create diverse communication topologies [9]. In contrast, MapReduce and similar high-level programming models support simple communication topologies and synchronization constructs. MapReduce also is a popular platform in which the dataflow takes the form of a directed acyclic graph of operators. However, it requires lots of I/Os and unnecessary computations while solving the problem of iterations with MapReduce.

5. Conclusions and Future Work

Big data is the “new” business and social science frontier. The amount of information and knowledge that can be extracted from the digital universe is continuing to expand as users come up with new ways to massage and process data. Moreover, it has become clear that “more data is not just more data”, but that “more data is different”. “Big data” is just the beginning of the problem. Technology evolution and placement guarantee that in a few years more data will be available in a year than has been collected since the dawn of man. If Facebook and Twitter are producing, collectively, around 50 gigabytes of data per day, and tripling every year, within a few years (perhaps 3-5) we are indeed facing the challenge of “big data becoming really big data”. We – as a global society – are evolving from a data-centric to a knowledge-centric community. Our knowledge is widely distributed and equally widely accessible. One program that is addressing this problem is The Federal Semantic Interoperability Community of Practice (SICoP) which supports an evolving model: Citizen-Centric Government – Systems That Know; Advanced Analytics – Systems That Learn; and Smart Operations – Systems That Reason. These systems will require big data. The data will not be stored in one or even a few locations; it will not be just one or even a few types and formats; it will not be amenable to analysis by just one or a few analytics; and there will not be just one or a few crosslinkages among different data elements. Thus, it is an exemplar of some of the issues we have addressed in this paper. Solving the issues and challenges addressed in this paper will require a concerted research effort – one which we expect to evolve over the next several years

References

[1] American Institute of Physics (AIP). 2010. College Park, MD, (http://www.aip.org/fyi/2010/)

[2] Ayres, I. 2007. Supercrunchers, Bantam Books, New York, NY.

[3] The State of the Art in Distributed Query Processing DONALD KOSSMANN, University of Passau, ACM Computing Surveys, Vol. 32, No. 4, December 2000.

[4] The Apprenda Library (https://apprenda.com/library/paas/iaas-paas-saas-explained-compared/).

[5] Felten, E. 2010. “Needle in a Haystack Problems”,https://freedom-to-tinker.com/blog/felten/needle-haystackproblems/

[6] Fox, B. 2011. “Leveraging Big Data for Big Impact”, Health Management Technology, http://www.healthmgttech.com/.

[7] https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html.

[8] Gantz, J. and E. Reinsel. 2011. “Extracting Value from Chaos”, IDC’s Digital Universe Study, sponsored by EMC.

[9] Jacobs, A. 2009. “Pathologies of Big Data”, Communications of the ACM, 52(8):36-44.

[10] JASON. 2008. “Data Analysis Challenges”, The Mitre Corporation, McLean, VA, JSR-08-142

[11] Kaisler, S. 2012. “Advanced Analytics”, CATALYST Technical Report, i_SW Corporation, Arlington, VA

[12] https://en.wikipedia.org/wiki/BigTable

[13] https://en.wikipedia.org/wiki/Pnuts

[14] https://cs.uwaterloo.ca/~kmsalem/courses/.../Chalamalla-HadoopDB.pdf

[15] https://en.wikipedia.org/wiki/Greenplum

[16] https://pig.apache.org/

[17] www.cs.rutgers.edu/~zz124/cs671.../srikanth_mapreducemerge.pdf. Map-Reduce-Merge: Simplified Relational Data. Processing on Large. Clusters. Hung-chih Yang, Ali Dasdan. Yahoo! Ruey-Lung Hsiao, D. Sto Parker.

[18] http://www.journalofcloudcomputing.com/content/3/1/12. Improving the performance of Hadoop Hive by sharing scan and computation tasks Tansel Dokeroglu1, Serkan Ozal1, Murat Ali Bayir2, Muhammet Serkan Cinar3 and Ahmet Cosar1.

[19] Liu et al. "An Investigation of Ptractical Approximate Nearest Neighbor Algorithms", 2004. Carnegie-Mellon University, pp. 1-8.

[20] www.elsevier.com/locate/jcss , Journal of Computer and System Sciences 77 (2011) 637-651.

[21] Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis, IJCAI-07 1606 , Evgeniy Gabrilovich and Shaul Markovitch Department of Computer Science Technion—Israel Institute of Technology, 32000 Haifa, Israel {gabr,shaulm}@cs.technion.ac.il.

[22] https://en.wikipedia.org/wiki/MapReduce.

[23] Applied Spatial Data Analysis with R Authors:Roger S. Bivand, Edzer Pebesma, Virgilio Gómez-Rubio.

[24] A twelve-analyzer detector system for high-resolution powder diffraction P. L. Lee, D. Shu, M. Ramanathan, C. Preissner, J. Wang, M. A. Beno, R. B. Von Dreele, L. Ribaud, C. Kurtz, S. M. Antao, X. Jiao and B. H. Toby. J. Synchrotron Rad. (2008). 15, 427-432.