Chapter 62
"Saksham Model" Performance Improvisation Using Node Capability Evaluation in Apache Hadoop

Ankit Shah
Shankersinh Vaghela Bapu Institute of Technology, India

Mamta C. Padole
https://orcid.org/0000-0002-0695-5970
The Maharaja Sayajirao University of Baroda, India

ABSTRACT

Big Data processing and analysis requires tremendous processing capability. Distributed computing brings many commodity systems under the common platform to answer the need for Big Data processing and analysis. Apache Hadoop is the most suitable set of tools for Big Data storage, processing, and analysis. But Hadoop found to be inefficient when it comes to heterogeneous set computers which have different processing capabilities. In this research, we propose the Saksham model which optimizes the processing time by efficient use of node processing capability and file management. The proposed model shows the performance improvement for Big Data processing. To achieve better performance, Saksham model uses two vital aspects of heterogeneous distributed computing: Effective block rearrangement policy and use of node processing capability. The results demonstrate that the proposed model successfully achieves better job execution time and improves data locality.

INTRODUCTION

In the current digital era, several terabytes of data is generated on daily basis, due to the advances in High Performance Computing, IoT devices, Sensors, Entertainment and Communicating devices and variety of applications. Big Data is a term coined for such a huge magnitude of data. Due to high speed

DOI: 10.4018/978-1-7998-5339-8.ch062

data generation, it is difficult to handle storage and processing of big data, using individual computing systems. But, the data, its processing and analysis have become vital from all perspectives of human life. For big data processing, distributed computing is the widely adopted approach, by researchers and scientists.

Distributed computing allows us to break big data processing tasks or jobs among multiple computing devices. Big data processing jobs can be executed on homogeneous or heterogeneous distributed systems. On distributed systems, we may need to dispense these jobs on thousands of machines for computing and finally are required to collect the results. In order to attain rapid outcomes, various tools and techniques need to be adopted for storage, processing and analysis of big data in distributed environment. These tools are required to handle various issues like fault tolerance, reliability, scalability, performance issues and many more. In the prevailing times, Apache Hadoop seems to be promising choice that is capable of handling the enormous amount of data, using HDFS combined with MapReduce.

The paper discusses features of Apache Hadoop and how it manages storage and scheduling of jobs. The existing approach in Hadoop comprises of some limitations, which are resolved using Block Rearrangement and Node Labeling for storage and scheduling, thus, improving performance.

The paper is structured as follows. Section 2 describes briefly about Apache Hadoop and the HDFS block placement policy. Section 3 describes the literature review on related work done for Hadoop performance improvement using different approaches. Section 4 gives an insight into the motivation of our work. Section 5 and 6 explains our proposed approach and Saksham: Block Rearrangement algorithm. Section 7 shows the experimental setup of Grid'5000. Sections 8 explain the results of the proposed approach compared with Hadoop default specifications. Finally, in section 9, the paper is concluded and future enhancements are mentioned.

APACHE HADOOP

Apache Hadoop (Hadoop.apache.org, 2018) is an open-source framework especially developed for the purpose of distributed computing for big data. Hadoop has become widely popular due to its adaptability of commodity hardware. Hadoop has a better edge in terms of performance in homogeneous environment rather than the heterogeneous one (Dean and Ghemawat, 2008). Hadoop comprises of three important components: Hadoop Distributed File System (HDFS), Yet Another Resource Negotiator (YARN) and MapReduce.

1. HDFS (Shvachko et al., 2010): It allows to split the dataset holding big data into multiple blocks and stores them to various datanodes in the distributed file system. Namenode maintains the metadata for the distributed blocks.
2. YARN (Vavilapalli et al., 2013): It separates the resource management layer and processing components layer. YARN is responsible for managing resources of Hadoop cluster.
3. MapReduce (Dean and Ghemawat, 2008): It is a programming framework on top of YARN, responsible for the processing of big data that enables enormous scalability across thousands of computing devices run on a Hadoop cluster.