

The Study of Hadoop Application across Multiple Data Centers

Aizhi Wu^{1, a}

¹College of Vehicles and Energy, Yanshan University, Qinhuangdao, 066004, China

^aemail: ysuwaz@163.com

Keywords: data-intensive computing; data center; Hadoop; hierarchical distributed computing; across multiple clusters

Abstract. Hadoop is a reasonable tool for cloud computing in big data and MapReduce paradigm may be a highly successful programming model for large-scale data-intensive computing application. However, traditional Hadoop and MapReduce have been deployed over local or tightly-coupled cloud resources with one data center. This paper focuses on the issue of Hadoop application across multiple data centers. A hierarchical distributed computing architecture of Hadoop is designed and proposed. The job submitted by user can be decomposed automatically into several subtasks which are then allocated and executed on corresponding cluster by location-aware manner. The presentation of the workflow shows the operating principle of this architecture.

Introduction

Nowadays, data-intensive computing typically uses modern data center architectures and massive data processing paradigms [1]. The requirements for data-intensive analysis of scientific data across distributed clusters or data centers have grown observably in recent years. This research is devoted to a study on the distributed data processing model across multiple data centers.

MapReduce paradigm and Hadoop implementation have emerged as successful model and framework for data-intensive high throughput applications. However, conventional MapReduce and Hadoop are developed to operate on single cluster environments and cannot be applied to distributed data processing across data centers. More and more data-intensive computing applications are widely (or even partially) distributed and have several distributed data centers. In these cases, the most efficient architecture for running Hadoop jobs over the entire data set becomes non-trivial.

In this paper, we try to employ the hierarchical methodology and propose a hierarchical distributed computing architecture based on Hadoop across multiple clusters. Socket is used to transfer these tasks to the corresponding data center, where the data processing is supposed to be run without data moving. Global HDFS (Hadoop Distributed File System) is presented to provide the catalog service of these data centers, which can record the meta-data and access path of each data center. The design and application of this architecture is presented and we hope this study may enlighten the attention of Hadoop global implementation.

Related Works

Cloud computing is a set of network enabled services to allow the centralized data storage and online access to computer services or resources. It provides scalable, quality guaranteed, normally personalized, convenient computing infrastructure on demand. So to say, Cloud computing is a reasonable and prospective way to solve many challenges in era of big data. The MapReduce paradigm and its open sourced implementation—Hadoop has been recognized as a representative enabling technique for Cloud computing [2].

In distributed computing frameworks based on Hadoop, one job committed to master node (name node) may be divided into several same tasks and then run on several slave nodes (data nodes). The data of this job is stored in HDFS which is distributed to data nodes of the same cluster. In the situation of distributed computing crossing clusters, there are three approaches to solve this problem [3]. One “local” approach is to collect the distributed data into a centralized location,

another “global” approach is to deploy a global MapReduce over all wide-area resources to process the data over a large pool of global resources, and the third approach is to set up multiple MapReduce clusters in different locations and then combine their respective results in a second-level MapReduce (or reduce-only) job. We tend to adopt the third way, by using data parallel processing paradigms on multiple clusters, simulations can be run on multiple computing centers concurrently without the need of copying the data.

Several research works have been investigated on this issue. In G-Hadoop [4], Gfarm and Torque were adopted to manage data files and resources for clusters, and a security framework in G-Hadoop has been studied. In P2P-MapReduce [5], the adaptive MapReduce framework provided a more reliable MapReduce middleware in dynamic Cloud infrastructures that can be effectively exploited to manage node churn, master failures, and job recovery in a decentralized but effective way. A Federated MapReduce [6] provided a transparent way to run original MapReduce jobs across multiple clusters without any extra programming burden. The Apache Hadoop MapReduce implementation may be upgraded for a multi-cluster environment with a decision algorithm that would prefer local computers to the remote [7]. The research on Hadoop (MapReduce) across data centers or clusters is rare, but we believe this issue is worthwhile. In our works, we try to employ Hadoop in the application of spatial data processing with high performance.

The Architecture Design of Hadoop across Multiple Clusters

With the rapid step of manufacture technology and observation technique, human can obtain massive spatial data of variant types from multiply sources. Researchers have devoted to collecting variant spatial data and providing spatial information service. Many spatial data centers have been built up all over the world for different purpose and diverse services. The processing and managing of spatial data must face the status of multiple data centers.

In the design and implementation of distributed computing architecture of Hadoop across multiple spatial data centers, there are mainly three problems need to be discussed and tackled. (1) How to obtain the view of all data in multiple data centers, which may decide the allocation of computing task to proper data center, (2) How to decompose the job into tasks and dispense them to corresponding clusters, which can synergistically control each Hadoop execution engine in multiple clusters, (3) How to deliver tasks transparently across different administrative domains, which need efficient and general way to keep communication among different clusters. In this paper, we try to employ the hierarchical methodology and propose a hierarchical distributed computing architecture based on Hadoop across multiple clusters.

This architecture represents a master/slave communication model. The master node is the central entity in this architecture and it is responsible for accepting jobs submitted by user, splitting the jobs into smaller sub-jobs and distributing these sub-jobs to the certain slave nodes where required data are located. The master node is also in charge of managing the metadata of all files available in every data nodes (global HDFS view of multiple HDFS). The slave node is installed on each participating cluster and enables it to run sub-jobs allocated by the job adapter in the master node (we can also call it as global name node).

Global HDFS provides a global view of all files in HDFSs of every cluster. In HDC-Hadoop data must be managed in a location-aware manner in order to provide the required location information for the task adapter on master node. Global HDFS just records meta-data of all HDFS files obtained from name nodes, such as data size, file format, access path and so on. When new file is input or existing file is modified, related name node (records the meta-data of HDFS in one cluster) must send the updated meta-data to global HDFS.

Users write the MapReduce program for the desired job based on the data files of global HDFS. After compiling and decomposing, the file names in Map or Reduction function may be transfer to certain locations of files in HDFS. Then the program is allocated and executed on one or several slave nodes where the required data files stored. If all the files in MapReduce program are located in one cluster (data center), the Map and Reduce programs may be transferred to corresponding cluster (slave node). Other else, while the files in program are located in several clusters, the

program may be recompiled to Map-Map-Reduce program or Map-Reduce-Map- Reduce program, and the program may be executed step by step in different clusters. The whole work flow is composed of five steps, shown as Figure 1.

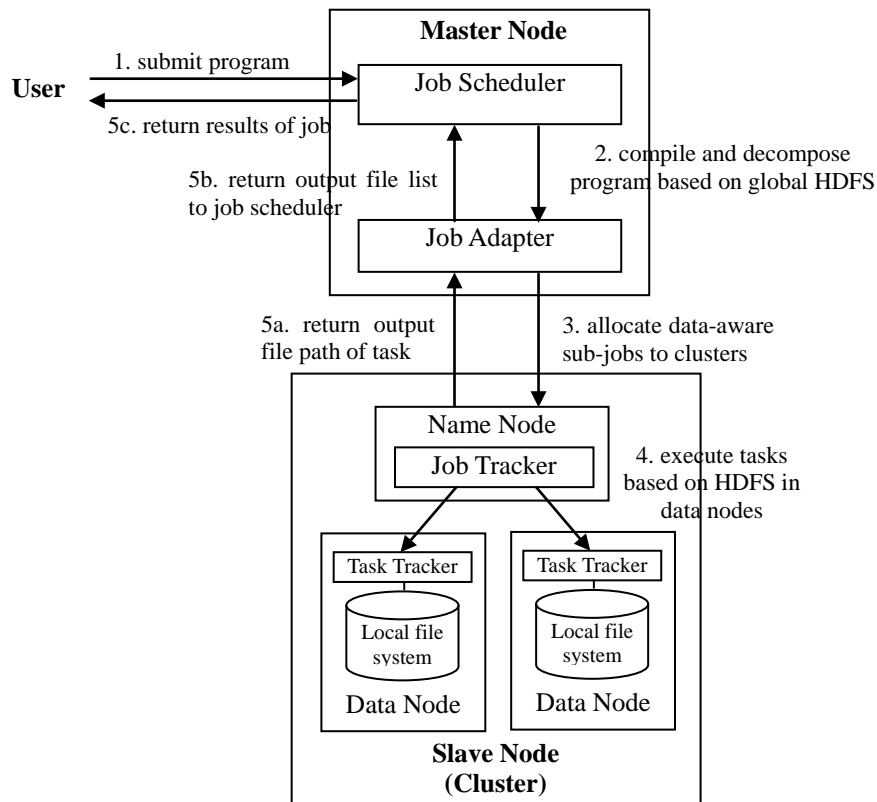


Fig.1. Job execution flowchart in hierarchical distributed computing architecture based on Hadoop

1) User submits a MapReduce program and configuration (including parameters, input files and additional resources) to master node (global name node). The job scheduler may set a job ID for the new job and push it in the queue of jobs.

2) Job in queue may be split into small location-aware sub-jobs based on required data. Based on meta-data in global HDFS, if all files in the MapReduce program are located in one cluster, the program may be transferred directly to corresponding cluster (slave node). Else if the files in the MapReduce program are located in different clusters, the program may be recompiled to Map-Map-Reduce or Map-Reduce-Map-Reduce program and organized into several sequential sub-jobs of different clusters. The file names in Map and Reduce functions based on global HDFS are translated into local file path of the HDFS in clusters.

3) Based on global HDFS, sub-job may be assigned properly to the name node of related cluster in data-aware manner. This allocation adopts socket communication method, which can inter-transfer message, program and file based on TCP (Transmission Control Protocol). And then a job execution can be run on the cluster based on local file system.

4) After the job is localized on the cluster, the *JobTracker* in name node splits into smaller tasks. When the *TaskTracker* in data node receives a new task, it localizes the tasks executable and resources in local file system. Then the job is executed by spawning a new JVM (Java Virtual Machine) on the computed node and running the corresponding task with the configured parameters.

5) After all tasks of a sub-job are completed in one cluster, the name node may return the result files paths to the job adapter in master node. The adapter checks whether this sub-job is a simple job (all required files located in one cluster, that is to say there is only one sub-job corresponding to a user job). If it is a simple job then register the new files in global HDFS and return the file name list to the job scheduler. Else if this sub-job has other subsequent or relevant sub-jobs, the output of this sub-job may be used as the input of other sub-jobs which may then be allocated and executed on

corresponding cluster. Till all the sub-jobs of one user job are completed, the output files may be register in global HDFS and returned the file name list to job scheduler. Finally, job scheduler returns the results to corresponding user by job ID.

Based on this architecture, distributed computing job across multiple clusters or data centers can be implemented.

Conclusion

The requirements for data-intensive computing across distributed clusters and multiple data centers have grown significantly in recent years. However, original MapReduce paradigm and Hadoop platform are developed to operate on single cluster environments cannot be implemented in large-scale distributed data processing across multiple cluster and data centers. The goal of this research is to apply Hadoop framework in the large-scale distributed computing across multiple data centers. In this paper, a hierarchical distributed computing architecture is designed and presented. This architecture is based on master/slave communication model and global HDFS is proposed to provide global view of data sets across distributed clusters. The job execution flowchart of this architecture is illuminated and shows the feasibility of this architecture.

Acknowledgement

In this paper, the research was supported by the Science and Technology Research and Development Plan of Qinhuangdao in China (No. 201401A010) and the Doctoral Foundation of Yanshan University (No. B718).

References

- [1] Lizhe Wang, Jie Tao, Holger Marten, Achim Streit, Samee U. Khan, Joanna Kolodziej and Dan Chen. MapReduce Across Distributed Clusters for Data-intensive Applications [C]. IEEE International Parallel & Distributed Processing Symposium, Shanghai, China, 2012. 2004-2011.
- [2] Lu Huang, HaiShan Chen, TingTing Hu. Research on Hadoop Cloud Computing Model and its Applications [C]. The 3rd International Conference on Networking and Distributed Computing, Hangzhou, China, 2012. 59-63.
- [3] Michael Cardosa, Chenyu Wang, Anshuman Nangia, Abhishek Chandra and Jon Weissman. Exploring MapReduce Efficiency with Highly-Distributed Data[C]. The 2nd International Workshop on MapReduce and its Applications, New York, USA, 2011. 27-33.
- [4] Lizhe Wang, Jie Tao, Rajiv Ranjan, Holger Marten, Achim Streit, Jingying Chen, Dan Chen. G-Hadoop: MapRecuce across distributed data centers for data-intensive computing [J]. Future Generation Computer Systems, 2013 29(3) 739-750.
- [5] Fabrizio Marozzo, Domenico Taliaa and Paolo Trunfio. P2P-MapReduce: Parallel data processing in dynamic Cloud environments [J]. Journal of Computer and System Sciences, 2012 78(5) 1382-1402.
- [6] Chun-Yu Wang, Tzu-Li Tai, Jui-Shing Shu, Jyh-Biau Chang and Ce-Kuen Shieh. Federated MapReduce to Transparently Run Application on Multicluster Environment [C]. IEEE 3rd International Congress on Big Data, Anchorage, USA 2014, 296-302.
- [7] I. Tomasic, A. Rashkovska and M. Depolli. Using Hadoop MapReduce in a Multicluster Environment [C]. The 36th International Convention on Information & Communication Technology Electronics & Microelectronics, Opatija, Croatia, 2013. 345-350.