ATLANTIS PRESS
Advances in Computer Science Research, volume 71
4th International Conference on Machinery, Materials and Information Technology Applications (ICMMITA 2016)

A kind of Prefetching Data Way to Hadoop MapReduce Environments

Hui Xia[1,a], Peng Wu[1,b]

[1] NO.253, North HuangHe Street, Shenyang Normal University, Liaoning 110034, China

[a]fruend_xia@126.com, [b]wu.peng@163.com

Keywords: Hadoop, MapReduce, Prefetching, Distributed Computing.

Abstract. In MapReduce environment, problem of data redundancy,a large number of tasks to be processing and mass data storage come up, in order to solve these problems, we put forward the way of data prefetching , preprocessing of hid the remote data access latency, by adjusting the allocation of resources to Reduce the business, we put forward the method of than before, caused by the Reduce tasks of remote data access performance problems caused by time delay and resource competition system, the hidden by prefetching data the method Reduce task of remote data access latency, and Reduce task control through the resource allocation, to Reduce resource competition caused the Reduce task, the experiment results show that with the default Hadoop graphs and Hadoop Online Prototype (HOP), compared with the method the system performance can be improved by more than 10%.

Introduction

The rapid development of Internet technology has led to the explosion of data [1], the massive data storage and processing of the problem. To solve this problem, the company take the lead in put forward the graphs model [2-3], the open source implementation HadoopMapReduce [4] has won the wide application in industry and academia [5-8].

Seo et al. [9], Ibrahim et al. [11] proposed to reduce the data access latency caused by Reduce task by reducing the amount of data to be copied and to solve the performance problem caused by delay. Reduce and Map tasks are required to be executed serially, and the performance gains from parallel execution of tasks are lost.Condie et al. [12] proposed to reduce the number of Reduce tasks waiting for data by using special transmission control protocol (TCP) connection copy data Time, but this will consume additional network bandwidth.The above methods did not reduce the data access latency and reduce resource competition to optimize system performance, therefore, can not solve the problem presented in the paper.

In order to solve the problem of system performance caused by the remote data access delay and resource competition caused by Reduce task, this paper proposes a data prefetching method based on pre-scheduling to hide the remote data access caused by Reduce task Delay, by Reduce the task and control the allocation of resources to reduce Reduce task caused by competition in the resource.

Reduce Performance issues caused by the task

MapReduce model based on the preparation of the job contains a large number of Map tasks and Reduce tasks, Map task processing operations of the original input data, resulting in <key, value> form of records, these records as intermediate results stored in the local node; Reduce tasks to deal with these Record, resulting in the final output of the job, but for each Reduce task, only the record containing the specific key in the intermediate result is processed.

During the execution, the Reduce task first copies the records corresponding to the specific keys from all the nodes, then sorts the records, and then performs the specified operations on the sorted records.As the number of Map tasks is very large and distributed among different nodes, Reduce tasks to complete the record copy, you must perform a large number of remote I / O operations; these operations will be in the process of introducing a large number of remote data access latency.

The Reduce task only copies the records from the processed Map task, that is, only after all Map tasks have been completed, the Reduce task can copy all the relevant records to the execution node. If the Map task has not been executed yet and no data is available for copying , The Reduce task enters the waiting state, but does not release the occupied resources such as memory and CPU during the waiting process, thereby causing unnecessary resource competition for other tasks performed on the same node, bringing the system performance Negative impact .

Data prefetching method in the process of the scheduling

To solve the problem caused by the Reduce tasks of remote data access latency and resource competition result in system performance problems, the author puts forward the data prefetching method based on the preliminary schedule. The method of graphs in the environment of existing task scheduling model for innovation, introduced the preliminary scheduling mechanism. This method defines the threshold value of prestressing scheduling and dispatching threshold, is used to control tasks into the time in the process of the scheduling and formal scheduling phase.

Definition of pre-scheduling threshold and the scheduling threshold.

Definition 1. $T_{pre-scheduling} = \{x \mid x \in (0,1)\}, T_{pre-scheduling}$ is scheduled threshold, which denotes has completed the Map task number and the ratio of the total number of Map tasks, take a number between 0 and 1.

When the Map task number and the ratio of the total number of Map tasks as scheduling threshold, began to Reduce task scheduling in advance. After receiving the advance scheduling tasks, node for the prefetch data. The preliminary schedule, the smaller the threshold value is set to begin to prefetch the earlier, and vice versa. If the scheduling threshold is set too large, may Reduce task starts the required data are not prefetch to local node, reach by prefetching hidden remote access latency.

Definition 2. $T_{scheduling} = \{x \mid x \in (0,1], T_{scheduling} \geq T_{pre-scheduling}\}.T_{scheduling}$ is scheduled threshold, says it has completed the number of Map tasks and the ratio of the total number of Map tasks, take a number between 0 and 1.

When the ratio of the total number of Map tasks and task scheduling threshold, began to Reduce task scheduling. After receiving the scheduling tasks, node allocation of resources to tasks, allows tasks to perform. The scheduling, the smaller the threshold value is set to node earlier, for task allocation of resources, tasks have been caused by competition for resources the sooner, the greater the impact on the rest of the task; And vice versa. Therefore, by adjusting the scheduling threshold, can control the task from scheduling to dispatch phase transformation, and control the resource competition caused by the Reduce task.

Pre-scheduling.

As stated earlier, when the Map task number and the ratio of the total as scheduled threshold, began to Reduce task scheduling in advance, in the process of scheduling in advance, the scheduler to the node first booking resources necessary for the execution of the Reduce task. Then, according to the reservation, choose appropriate amount of the number of tasks, the state changed to assigned to make an appointment after its "scheduling" nodes. After receiving the assigned tasks, appointments node started from the intermediate results of the Map task output data prefetching for scheduling tasks.

The following defines the two rules are used to calculate the resources need to make an appointment.

Rules 1. If $R_{total} \quad DIV \quad N_{total} > 0$,then $\forall_{n_i} \in N, i \in [1, N_{total}], (R_{total} \quad DIV \quad N_{total})R_s \rightarrow n_i$,wherein R_{total} and N_{total} respectively denote the total number of Reduce tasks in the job and compute nodes in the cluster,N denotes a set of computing nodes, $\rightarrow$ denotes a resource reservation relationship, $(R_{total} \quad DIV \quad N_{total})R_s \rightarrow n_i$ indicates that the amount of resources reserved to the node n i is three $(R_{total} \quad DIV \quad N_{total})R_s$.

Rules 2. **If** $R_{total} \ \ MOD \ \ N_{total} \neq 0$,then $\exists S \subset N, \forall_{S_j} \in S, j \in [1, R_{total} \ \ MOD \ N_{total}]$, $R_s \to s_j$,S denotes a collection of $(R_{total} \ \ MOD \ N_{total})$ nodes.

The calculation process is as follows: Step 1: Calculate the amount of resources needed to make an average reservation to each node Step 2: Calculate the amount of resources required to reserve a Reduce task for each additional node Step 1 According to rule 1, The steps are calculated according to rule 2.For ease of description, $R_s = (Mem, Compt, Disk, O)$ denotes the amount of resources required to perform a reduce task,wherein, Mem, Compt, Disk and O represents the required amount of memory resources, computing resources, hard disk storage space and other resources respectively.

Prefetching.

In the pre-fetching process, each node first obtains the completed Map task information, and finds out the intermediate result of the Map task output in the pre-fetching process. In this way, From which the data associated with the pre-scheduled Reduce task is copied and stored locally at the node.Reduce tasks can read the data locally during execution, thus hiding the latency of accessing the data remotely.Each node prefetches data to the Reduce task pre-scheduled to this node according to the following model.

$$f(x,y) = \begin{cases} (x+1) \ \ MOD \ \ N_{total}, & y = 1, \\ (f(x,y-1)+1) \ \ MOD \ N_{total}, & y > 1, \end{cases} \quad (1)$$

Where x is the node number of the prefetch operation, y is the round number prefetching, $f(x,y)$ is the linked list number, N_{total} is the total number of compute nodes.

In the prefetching process, the node first computes the value of $f(x,y)$ according to Eq. (1), and then reads the map task completion information from the linked list with the number $f(x,y)$. From the intermediate result indicated by the information *(x, y)* is calculated according to Eq. (1),and the value of $f(x,y)$ is calculated according to $f(x,y)$ to read the information in the list to start a new round of data prefetching,prefetching process shown in Fig.1

Experiment

The experimental environment consists of a 11-node Linux cluster, in which all nodes in the cluster are interconnected by two cascaded gigabit network switches.The default Hadoop system is implemented on the cluster, the Hadoop system with prefetch technology is implemented, HOP system.In the experiment, Hadoop Distributed File System (HDFS) file block size is set to 64MB and 128MB in the HDFS file block size in turn take these two values in the case of three systems running the same , And verify the validity of the method by comparing the execution of the job.

This paper mainly deals with the system performance problem caused by the remote data access latency and resource competition caused by the Reduce task. The execution time of the job is an important index to reflect the system performance. The shorter the execution time is, the better the system performance is. Time to evaluate the effectiveness of the prefetching method proposed in this paper.In order to be more general, the average execution time of the job is the evaluation index.Compared with the other two Hadoop systems, The shorter the average execution time in a Hadoop system, the more effective the method is in the table 1. Table 1 gives the job information used in the experiment.

The prefetch method based on pre-scheduling has been implemented in Hadoop-0.20.2 by comparing the execution of the pre-fetched Hadoop system and the HOP system in the default Hadoop system to verify the performance of this method HOP system is based on Condie et al to achieve the results of open-source systems.

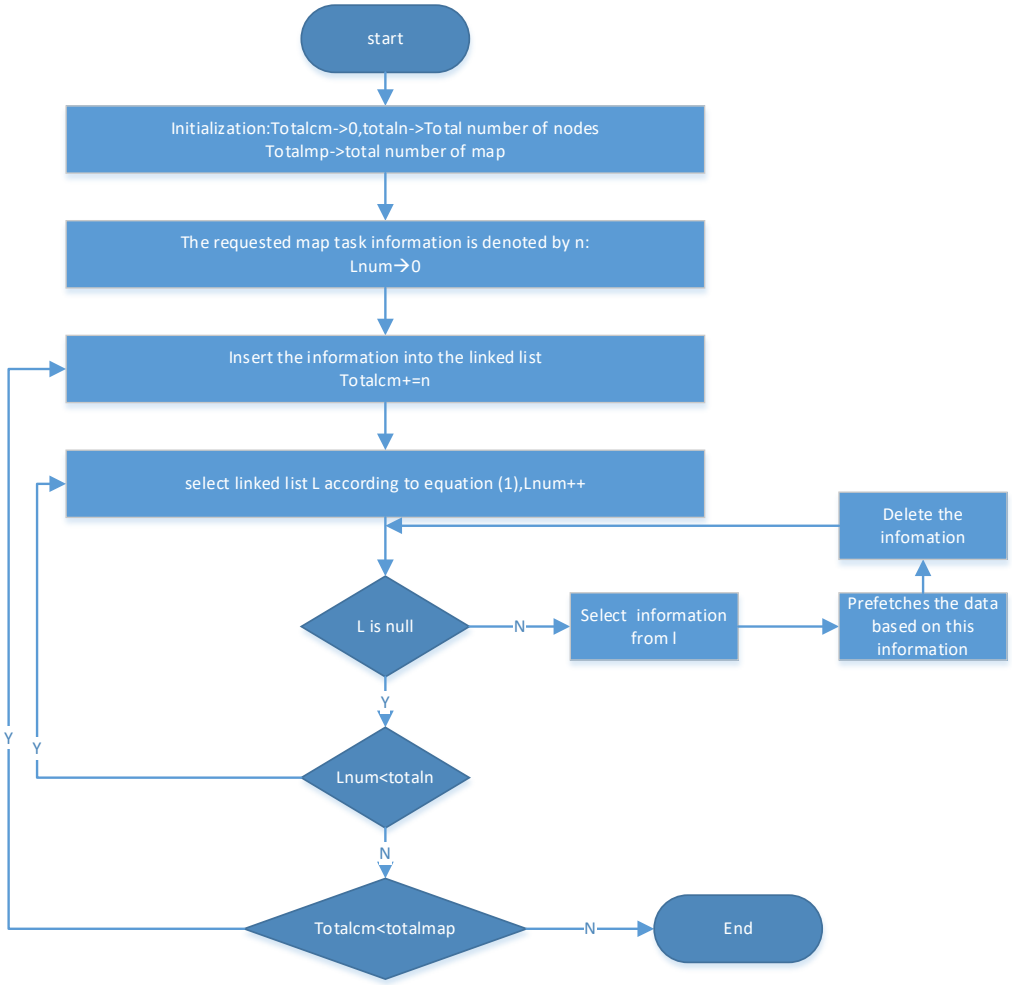

Fig 1. Prefetching process flow chart

Table 1 Job Information

Job number	Input /GB	Map tasks	Reduce tasks	HDFS File block/MB
1	110.28	183	1	64
2			5	
3			10	
4	24.74	183	1	128
5			5	
6			10	

As shown in Table 1, jobs 1, 2, and 3 are used to evaluate the effectiveness of the prefetch method in the case of HDFS file blocks of 64 MB. Fig. 1 shows the Hadoop implementation of the prefetch method for the three jobs in the default Hadoop system. The average execution time in the system and the HOP system is shown in Fig. 1. In most cases, the average execution time of the job in the Hadoop that implements the prefetch method is less than the average execution time in the other two systems. The Hadoop system in the province, when compared to the situation in the HOP system, the best execution time is reduced by 13% and 29%, respectively, in the case of the Hadoop system implementing the method in the text, the average execution of the job 2 Time reduced by 10% and 34% respectively, and the average execution time of job 3 decreased by 6% and 36%, respectively.

Figure 2 shows the average execution of the three jobs in the Hadoop system and the HOP system that implement the prefetch method on the default Hadoop system Time, as shown in Figure 3. Compared with the other two systems, on average, the average execution time for job 4 decreased by

6% and 17%, the average execution time for job 5 decreased by 13% and 15%, respectively, and the average execution time for job 6 decreased by 10% and 7%, respectively.

During the experiment, it was found that the average execution time of a job in a default Hadoop system is shorter than the average execution time in a Hadoop system that implements a prefetch method. Figure 5 depicts the number of occurrences of such a situation. Figure 5 shows that, when the HDFS file block to 64MB, there were six operations such cases; when the HDFS is 128MB, there have been three such cases, which shows that the method in the HDFS file block larger, stable.

Summary

In order to solve the system performance problem caused by remote data access latency and resource competition caused by Reduce task, a data prefetching method based on pre-scheduling is proposed.The task scheduling mode of Hadoop system is innovated,The scheduling process is divided into two stages: pre-scheduling and formal scheduling, pre-fetching data for the Reduce task, hiding the remote access latency of the data using the pre-scheduling stage, adjusting the resource allocation by the formal scheduling phase, controlling the resource competition caused by the Reduce task . This method has been implemented in Hadoop-0.20.2, and the experimental results show that this method can improve the system performance

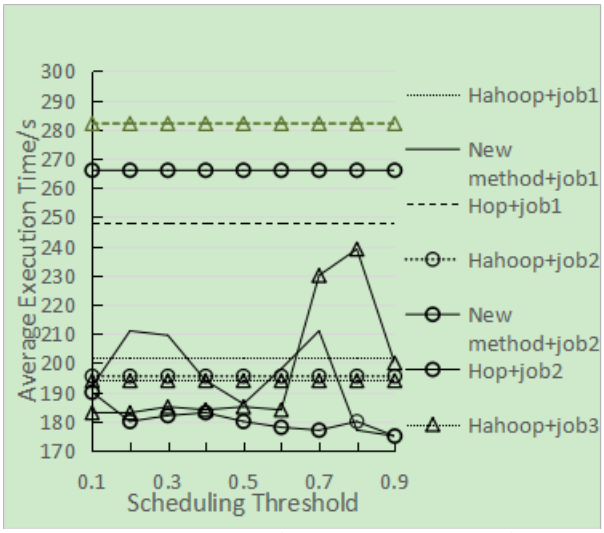

Fig 2. Average execution time for a job with HDFS block of 64MB

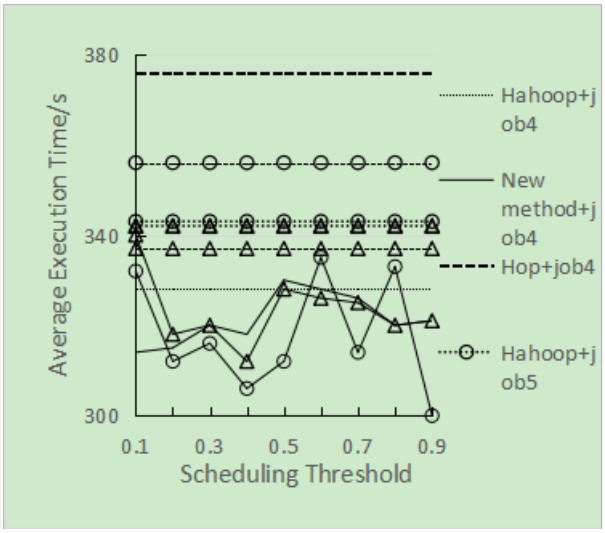

Fig 3. Average execution time for a job with HDFS block of 128MB

References

[1] Gantz J,Reinsel D. The Digital Universe Decade-are You Ready? [DB/OL]. [2012-12-26]. http://www.emc.com/collateral /demos/microsites/idc -digi -taluniverse/iview.htm.

[2] Dean J,Ghemawat S. Mapreduce:Simplified Data Processing on Large Custers[J]. Communications of the ACM,2008,51(1):107-113.

[3] Ghemawat S,Gobioff H,Leung S.The Google File System[C]//Proceedings of the 19th ACM Symposium on Operating Systems Principles.NewYork:ACM,2003:29-43.

[4 The Apache Software Foundation.Welcome to Hadoop Mapreduce! [DB/OL]. [2012-12-26]. http://hadoop.apache.org/mapreduce/.

[5] Menon A. BigData@Facebook [C]//Proceedings of Workshop on Management of BigData Systems. New York: ACM,2012:31-32.

[6] Lattanzi S,Moseley B,Suri S,et al.Filtering:a Method for Solving Graph Problemsin MapReduce [C]//Proceedings of the23rdACM Symposium on Parallelism in Algorithms and Architectures. NewYork:ACM,2011:85-94.

[7] Shao B,Wang H,Xiao Y.Managing and Mining Large Graphs: System sand Implementations[C]//Proceedings of the ACM SIGMOD International Conference on Management of Data.NewYork:ACM,2012:589-592.

[8] ChenY,Alspaugh S,Katz R.Interactive Analytical Processing in BigData Systems:a Cross -industry Study of MapReduce Workloads[C]//Proceedings of the VLDB Endowment:5. NewYork: ACM, 2012:1802-1813.

[9]SeoS, JangI, WooK,et al. HPMR:Prefetching and Preshuffling in Shared Mapreduce Computation Environment[C]//Proceedings of IEEE International Conferenceon Cluster Computing. Piscataway: IEEE,2009:1-8(528917).

[10] Ibrahim S,JinH,LuL,etal.Leen:Locality/Fairness-aware Key Partitioning for Mapreduce in the Cloud[C]//Proceedings of the IEEE International Conferenceon Cloud Computing Technology and Science.Piscataway:IEEE,2010:17-24.