A MapReduce Iteration Framework In Local Parallel and Message Synchronization

Ting Wang, Hualiang Zhang, Peng Zeng
Department of Industrial Control Network and System
Shenyang Institute of Automation Chinese Academy of Sciences
Shenyang, Liaoning, 110016, China
E-mail: wangting1@sia.cn

Keywords: MapReduce; Local Parallel; Message Synchronization; Parallel Layer; Barrier Grid

Abstract. With the development of large-scale distributed computing, Stand-alone operating environment to meet the demand of the time and space overhead of massive data based on. There is more attention to how to design the distributed algorithm for efficient cloud computing environment. The MapReduce model cannot solve the issue. In this paper, the redesign of the computing model of MapReduce, ensure the existing calculation models compatible with the old MapReduce operation. At the same time, the framework used the message synchronization mechanism to implement state data changing interaction tasks in Parallel Layer. Compared to the original MapReduce operation, greatly reduces the processing time of the MapReduce iterative algorithm.

Introduction

MapReduce [1] is a kind of processing large scale data sets distributed Google proposed parallel programming model, also is Is the current cloud computing core calculation model. Many scientific research institutions and companies are R & D design of a parallel processing system for massive data specification MapReduce based on their technology. The Apache Hadoop is an open source implementation of MapReduce, Is the current academia and industry in fact the massive data parallel processing standard. Hadoop can be easily deployed in a commercial machine cluster of common, but Hadoop only provides limited executive strategies for users, the reason is parallel programming environment simplifies the user, improve the level of abstraction. In the fact, a large number of algorithms are iterative process significant and have a dependency relation data. If we develop a MapReduce iterative application, we need to actively intervention implementation process. Through the external chain called many times to implement. There is some system or MapReduce system based on Hadoop, Such as HaLoop [2], Twister [3] etc. They try to complete the iteration in internal operations persistence reduces the cost of several rounds of intermediate results. In the realization of topology but with distributed memory and local cache to store the graph, the design method has some limitations in dealing with massive data.

The way to solve the problem of the source of a very important topic in the MapReduce programming specification: Mapper or Reducer there is no dependence, no interaction in the data section of different independent execution. This is a partial parallel computing "hypothesis". Parallel problem can be solved based on this model can be decomposed into several independent parts and they can be executed independently and asynchronously. There is no communication between the ideal model of the asynchronous parallel Mapper or Reducer, Shuffle processing data interaction between Mapper and Reducer depends only on the sync level. Therefore some demand for intermediate data interaction contains the iterative process of the parallel algorithm can only use chain called MapReduce operations to meet the demand of data interactive iteration, according to the convergence condition of iterative applications to decide when to terminate the call chain.

This paper breaks through the local MapReduce parallel computing based on the hypothesis, designs a synchronous parallel computing framework based on Hadoop message. The new framework will Map (Reduce) phase is divided into a number of synchronization in Parallel Layer, Parallel Layer asynchronous parallel tasks, Parallel Layer to complete data interaction between tasks using the message passing mechanism. The new framework using the graph node drive more
efficient support for a large class of iterative information needs and contains the iterative process, greatly reducing the iterative algorithm will not be in the original MapReduce processing mode price.

Related Works

Research on parallel programming model and Optimization for large-scale distributed current graph algorithm and Pregel [4] programming framework many results can provide support for the. Google parallel graph algorithms, according to the graph is characterized by multiple rounds of iterative design at the vertex information processing mode. But the platform is not based on Hadoop, restrictions in both openness and versatility, and all of its computational state is kept in memory, it is also a lack of effective support for large-scale data. AveryChing et al proposed a large graph processing framework based on Giraph Hadoop, the framework supports dynamic resource management, distributed coordination system work unit using ZooKeeper, and provides graphics library supports iteration, and the programming framework all time message using processing capacity of synchronous programming for improving iterative algorithm in the Hadoop platform, but in the design strategy and implementation details (especially the message processing) has many different, relative to the Giraph built on Hadoop multithreaded architecture, frame is designed in this paper adopts the design mode of invasion, the implementation of distributed coordination between nodes using communication, without the need of manual segmentation and distribution the input data, so the compatibility, easy to use and general characteristics of more advantages of Surfer[5] system provides the primitives of MapReduce and Propagation primitives, and the use of building blocks the to support large algorithm in the cloud, the main goal is to provide visual monitoring during operation, and not related to map features the specific implementation details of Lin et al in [6,7] graph algorithm on MapReduce realization for local optimization, proposed the Mapper composite And, to avoid repeated transmission and range partitioning graph topology optimization technology, but it is based on MapReduce scheduling multiple round. There are many parallel processing framework, such as Apache HAMA [8] and CMU GraphLab [9] also supports iteration, but the platform for specific problem areas. In addition to [10, 11] is a lot of the research hope from the parallel computing of Message Passing Interface (MPI) to enhance the processing ability of the MapReduce, but these researches have not given implementation based on Hadoop platform, the fault-tolerant and scalable, robust characteristics exist in the loss.

I. DESIGN

![Diagram of Partial parallel model]

This paper aims to inherit many characteristics of Hadoop based on the original model by introducing the message synchronization, using the message passing mechanism of Parallel and Layer synchronization to more efficiently support distributed graph algorithm.

**Local parallel model.** Parallel computing can be used in a variety of different parallel programming model representation and implementation, each model has a computing application to adapt. In order to simplify parallel application development to work, no parallel development experienced programmers can correctly and quickly write parallel applications, MapReduce restrictions on the programming model, the main part according to the local parallel computing
model of the MapReduce programming model. It is using this model limitation, MapReduce realizes automatic parallel processing parallel programs, and the automatic support such as input partition, parallel task scheduling and communication, fault tolerance, load balancing, parallel details provided in the interior, achieving high scalability and high parallel factor.

A number of parallel tasks can be mutually by dividing the input sets no communication cost independent execution. The following formal definitions of partial parallel programming model MapReduce. See from Figure 1 can be, the MapReduce job execution plan is divided into Map and Reduce in two stages, also known as an MR process, parallel task asynchronous internal each phase of the MR process (Map1~Map3 or Reduce1~Reduce3) are running in the ideal model of partial parallel.

On the Map stage, independent input fragment Split1~Split3 input data is automatically cut into equal size (the default value of Split is 64MB, and the underlying distributed storage system block size, this strategy avoids the data transmission of Split across block boundaries and may have caused the network cost), several key input segments is parallel to the Set. A MapReduce processing framework based on data localization optimization strategy will run the execute node Map1~Map3 distribution to the input segments where there is no dependent relationship. No communication between the Map1~Map3 implementation process, in accordance with the formal definition of partial parallel, intermediate results it produces is a new collection of key value pairs.

In the stage of Reduce, the intermediate results generated by the Map1~Map3 through the intermediate results generated by the output key partition operation after the partition of Part1~Part3 as input to Reduce1~Reduce3. There are no any dependencies without communication between Reduce1~Reduce3 implementation process, in accordance with the formal definition of partial parallel, output the final results of Reducer will automatically be persisted to floor HDFS.

Between Map phase and Reduce phase is synchronous serial, exist in a relatively transparent user implicit synchronization and communication process.

Reducer must wait until the last Mapper execution completed before starting execution. However, the Shuffle process data generated by Mapper and Mapper is two overlapping execution, namely arbitrary after the end of the Mapper, Reducer can Shuffle the intermediate results, which can shorten the length of the parallel and pipeline processing, improve processing efficiency. If a concurrent application of the MR process requires the use of multiple times, then an MR process and MR process are a chain of synchronous serial. Between communication and data interaction also occurs only in an MR process of Map phase and Reduce phase as well as the multi-pass MR process.

Message synchronization. The main challenge by message synchronization mechanism lies in the existing MapReduce parallel computing framework, between Mapper or Reducer does not support the message. The Map (Reduce) BSP model of internal support for message passing inspiration of parallel computing. BSP model relative to the existing MapReduce provides Abstract parallel high level, implementation synchronous control using Barrier Grid, using the message passing mechanism to complete the interaction task parallel data.

In this model, a parallel job consists of a series of Parallel Layer, a Phase Parallel. The message synchronization calculation is mainly composed of 3 ordered parts every Parallel Layer: (1) partial parallel asynchronous computation of each task. Parallel Layer independent parallel execution; (2) parallels tasks in Parallel communication. Layer before the end of the message passing mechanism to complete the data interaction; (3) synchronization synchronization waiting with a Parallel Layer all parallel tasks interact completed, then the entire parallel can down a Parallel Layer to move into the next round, parallel.

See the BSP model for the design of MapReduce parallel processing framework and the original MapReduce parallel processing framework to maintain consistency in the overall framework from Figure 1 can be observed from a macro point of view, if the Map phase of one of the MR process as a Parallel Layer, the Reduce phase as another Parallel Layer, Shuffle process two stages as Barrier Grid synchronization and communication process of Parallel between the Layer, but MapReduce itself is also in line with the message synchronization calculation, which is why the MapReduce job scheduling chain can be realized based on iterative interactive demand.

In BSP model, the intermediate state data by using the message passing mechanism between the parallel tasks can be interactively changed, parallel task message delivery between mission not separate dispersion behavior, but are considered as a whole and constraint between adjacent Parallel Layer. The sent message needs to be brought together in a Parallel Layer, on a Parallel Layer brings together news need to wait until the next Parallel Layer can be used in parallel task execution, the message can not cross the discontinuous Parallel Layer. Using message passing parallel logic can be used interactively, the same data continue to run in a new state. The iterative calculation can be abstracted as the processing logic the same, in the state of the data the new focus on expected
convergence for execution, so this mode is very suitable for distributed parallel graph algorithms are iterative demanding. And this is not across the Parallel Layer message passing model effectively simplifies a lot of parallel task message between maintenance cost.

To guarantee message to overall pooled form orderly transfer in Parallel Layer, we must rely on efficient synchronization mechanism of Parallel Layer. In the BSP model, the synchronous wait for Parallel Layer is achieved by using Barrier Grid [11]. Barrier Grid is the global coarse grain size controllable synchronization mechanism, using Barrier Grid a parallel tasks can be divided into a plurality of continuous loosely synchronized Parallel Layer, Parallel in Figure 1 Layer0 and Parallel Layer1. Its message is guaranteed converge only within a Parallel Layer, and passed in the adjacent successor Parallel Layer.

In this paper, the improved parallel processing framework to support the Parallel Layer inside the Map or Reduce phase, this design mode based on calculation, need through iterative times MapReduce external chain calls can be achieved before, is now available in an MR process, using the internal Map stage (stage Reduce) synchronous execution of multiple Parallel Layer can be completed. Complex message transfer control of the new operation system, the parallel program developers only need to use the original MapReduce application development experience can write parallel applications more efficient in the improved parallel framework. The parallel frame improved effectively reduces the external iteration takes a lot of time. But, compared with the original Hadoop parallel computing framework for [12], parallel computing framework to support the improved BSP model also introduces some new cost: First of all, coarse-grained Barrier Grid synchronization makes the overall execution time of a single Parallel Layer is sensitive to the single most slow to complete the task, the inconsistent state task completion time unusual problems can make use of [1] Speculative Execution Hadoop mechanism, using the backup task redundancy execution effectively, and the computing and communication of multiple tasks a Parallel Layer concurrency is overlapping execution, so the price can further be a Parallel Layer asynchronous parallel multiple tasks amortization; secondly, a new cost model of Barrier Grid synchronization is a potential may result in decreased performance bottleneck, however, actually improved in the Barrier model Grid synchronization is implicit Barrier Grid model in multiple MR inter-procedural Map and Reduce phase synchronous transfer into multiple Barrier Grid Map and Reduce stages of a MR in the synchronous, so nature does not add any new synchronization costs.

### Implement

Precursor set and Successor set represent the nodes in parallel computing based on dependencies between tasks, and message passing. We based on the adjacency list through appropriate expansion deformation, abstract used in the improved framework has to enter the key figure of representation, as shown in Figure 2.

![Fig. 2 Model framework of Key-Value pair](image)

Figure 2 input keys can be arbitrary content (the default button is the offset line of text at the beginning), input values are divided into 5 basic parts: a node identifier, the direct precursor of collection, direct successor set, metadata and the current node state. The metadata is the real key information graph element represents, metadata contains nodes the metadata and the side of the two parts. For example, in the traffic network graph, nodes represent the road intersection, where nodes metadata may include position coordinates, the region and the node name and other information; the edges represent connectivity between node section, the data may include the link length and section capacity, one-way traffic and other information. The current node status is with the current state value iteration update, the node in the middle state on behalf of the iteration, when the iterative convergence, intermediate state value becomes the graph operation result status value.

Each iteration consists of processing the same logic, which comprises the following main steps:

**Function 1 node drive start process.** As mentioned earlier, figure after pretreatment is divided into an input collection of key value pairs, each pair is calculated as the central node, the user design Map processing logic according to the application requirements of the intermediate state key information contained in the first iteration computing nodes the value of:

**Interactive processing between 2 node state.** Using message passing mechanism, the intermediate state node value according to the nodes in the graph adjacency relationship of transmission, adjacent relation is the direct precursor of the collection node and direct successor set completely;
**Function iterative processing node drive.** The new intermediate state and a parsing Parallel Layer messages obtain the adjacency node values in the adjacent nodes, intermediate state the new value of the original input key set and graph topology on the set, Map processing logic application user design calculation intermediate nodes in the iteration the value of the set.

**Iteration termination detection.** According to the termination condition determines the iterative application is to continue performing iterative processing returns to step B, or to stop the iterative return results. The system can terminate condition specifies two iterations, a comparison of results between Parallel Layer is adjacent to the error is less than the specified threshold, two iterations set an upper limit.

By the description of the core content, calculation model of MapReduce based on the graph is the node of the asynchronous parallel calculation and synchronous message passing based on adjacent edges.

**Conclusions and Future Works**

In this paper, based on the present algorithm MapReduce implementation performance problem, by introducing the message synchronization model to realize an improved support for Barrier Grid message passing parallel computing framework based on Hadoop open source. Through the iteration process is internalized into the Map or Reduce section of the Parallel Layer, effectively reduce the previous round scheduling overhead, provides an efficient computational model for the design of distributed graph algorithms. However, how to solve the degradation information interaction frequent dense graph performance problems and more widely in machine learning, clustering algorithm is implemented under the platform remains to be further studied.

**Acknowledgment**

The authors acknowledge the financial support of the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant No.XDA06020602.

**References**


Vehicle, Mechatronics and Information Technologies
10.4028/www.scientific.net/AMM.380-384

A MapReduce Iteration Framework in Local Parallel and Message Synchronization
10.4028/www.scientific.net/AMM.380-384.2237