

Optimization of Task Scheduling Algorithms in Heterogeneous Environment

HaiLan Pan

School of Economics and Management, Shanghai Polytechnic University,
2360Hao, JinHaiLu, PuDong Shanghai 201209 China
panhailan@sspu.edu.cn

Keywords: cloud workflow; heterogeneous; scheduling algorithm; HDFS; MapReduce

Abstract. Workflow system is the best choice for projects with complex business logic, and cloud workflow technology has emerged because many applications need to join large data processing. This paper first introduces the workflow and MapReduce technology briefly, and then proposes to use agent mode to solve the engine centralization problem of traditional cloud workflow architecture. Finally, this paper describes the workflow scheduling model and algorithm based on MapReduce, which makes full use of time fragmentation by using task segmentation to improve efficiency.

Introduction

With the widespread use of computers and mobile devices, as well as the rapid development of Internet technology support, people's work, life and entertainment have undergone tremendous changes. All these have led to the explosive growth of information and data, so cross-disciplinary, interdisciplinary large-scale collaborative data research has become a hot spot. At present, general application system is difficult to achieve multi-party collaboration and complex logic process, and the workflow system has the characteristics of business and logic separation, which can solve the problem of multi-application cooperative work. To make data collection, processing and storage more convenient, many companies have chosen a cloud computing platform to solve large-scale parallel programming, simplify resource management, which can not only provide powerful high-performance computing power and mass data storage capacity for major business systems, but also create the basic conditions for collaborative development[1].

Related technical analysis

Workflow technology. Workflow technology is the calculation model of workflow, that is, how to express the appropriate model and implement calculation in the computer for workflow' s the logic and rules. The main problem which workflow can solve is that in order to achieve a business goal, using computers transfer documents, information or tasks in a number of participants according to achieve a predetermined rule automatically.

MapReduce model based on Hadoop. Hadoop will divided work into a number of small tasks to perform, these small tasks are divided into two categories: map tasks and reduce tasks. Users only need to customize the map and reduce functions, and not worry about complex processing details[2,3]. A brief introduction to the MapReduce model process flow is as follows:

A) The related data of Mapreduce tasks will be divided into small blocks of data by Hadoop, the block size tends to data size of HDFS block whose default value is 64MB, and then the task map will be assigned to run in the storage of small input fast node , This is known as the implementation of data localization optimization. The input data of map function is key/value pairs, and intermediate data set generated after processing is key/value pairs too.

B) Then sorting the intermediate data set, which is mainly merge sort. Followed by the copy process, that is Reduce requests to read each maps' output of the intermediate data. The sort and

copy phases are often referred to as shuffle stages. Then according to the reduce rules in the reduce side will merge the Intermediate data pair.

C) Finally, the output of the Reduce function is stored in HDFS for later processing or user accessing.

Cloud workflow framework

There are many good platforms/frameworks in the cloud environment, but this paper chooses Hadoop platform as the sub-platform of the workflow management system in the cloud environment, and focuses on the processing of MapReduce type task in Hadoop and File synthesis processing. Because MapReduce framework owns good parallelism and HDFS has a good function of managing the underlying files.

Architecture approach of the workflow model. At present, the two main workflow model architecture methods are service orchestration and service choreography. The workflow engine in the service orchestration framework is centralized, and the tasks and related data in the process are monitored and scheduled through the central control. If the big data is processed, it is easy to cause the bottleneck. The workflow engine in the service choreography framework is distributed. By tracking the information sequence of the tasks in the process, the common goal can be achieved by the point-to-point interaction, and the bottleneck can be avoided, but which is difficult to realize[4]. This paper adopts a centralized service orchestration architecture.

Traditional Workflow Architecture. Fig. 1 depicts the overall architecture of a traditional cloud workflow deployed in a cloud computing environment. This includes the user access layer, the workflow engine layer, and the underlying cloud.

User access layer as a workflow portal, is the information exchange channels between user and workflow engine. Workflow engine is the core part of workflow management system. Workflow execution system can be composed of one or several workflow engines, and different workflow engines work together to accomplish workflow task. The underlying cloud, which is based on cloud resources, provides a range of cloud support functions including transaction processing, storage, and so on.

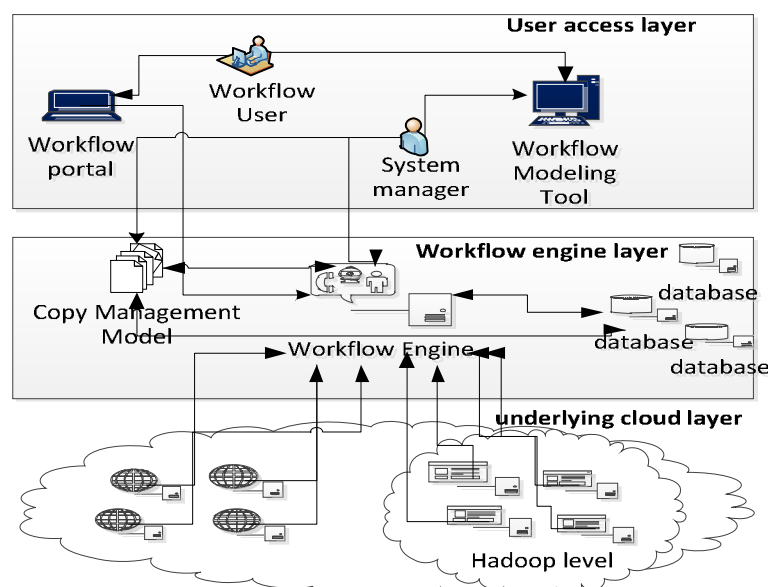


Fig. 1 the overall structure of the traditional cloud workflow

Middle Layer Proxy Structure. From Fig. 1 above, the workflow engine layer is the most heavily burdened, all the control flow and data flow have to go through the central engine, which is a typical service orchestration architecture. This paper will join the middle agent layer between the workflow engine layer and the underlying cloud to support the control by the central engine, the data between the agents can be transmitted directly to each other so that the system can avoid bottlenecks, and for

MapReduce based cloud services, it can be assigned to Hadoop model on underlying cloud , and let it give full play to the advantages of distributed management.

As shown in Fig.2, The way of using proxy nodes can reduce the transmission of intermediate data, thereby reduce the pressure on the central engine to avoid bottlenecks.

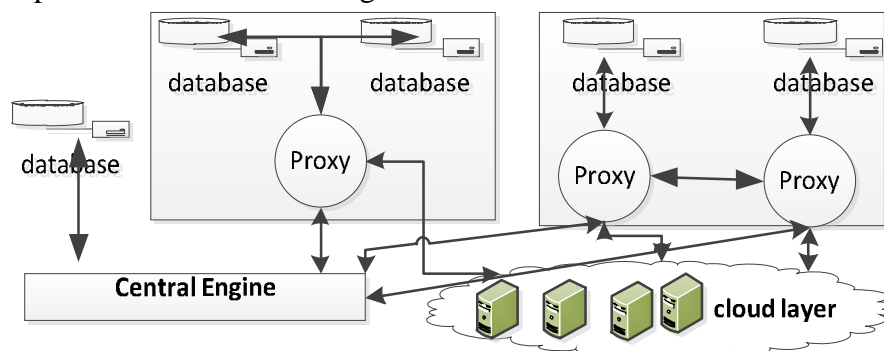


Fig. 2 the middle layer proxy structure

MapReduce workflow scheduling model and algorithm

The task scheduling in workflow management system includes task decomposition, resource location, resource selection and optimization. Scheduling is often performed by selecting resources that satisfy scheduling constraints in a given set of resources.

At present, most applications put the entire MapReduce type transaction on the Hadoop platform, and the scheduling of the workflow task based on the priority constraint is separated from the scheduling of the MapReduce task under the Hadoop platform. However, scheduling can be affected by the characteristics of the underlying distributed resources and the relationships among tasks in the process. So this paper will be marked in the allocation of tasks to distinguish, so that MapReduce type of transaction in the Hadoop platform to deal with other transactions are assigned to other platforms, as shown in Fig. 3.

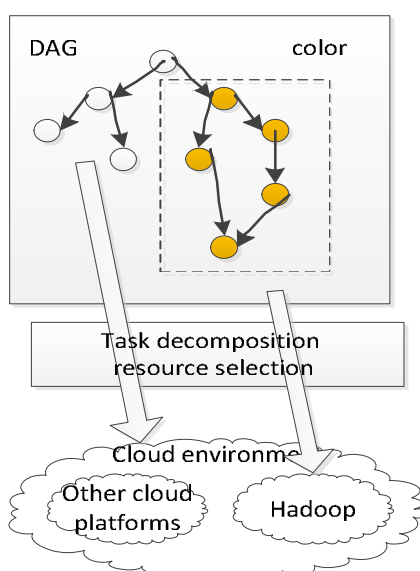


Fig. 3 Colored selection diagram

Hadoop environment deployment. Hadoop is a MapReduce computing function module of workflow management system extension. In addition to its framework deployment, Hadoop should consider the integration and communication with the workflow system, and finally make the workflow system have the ability to handle the cloud affairs of MapReduce class. As an extension framework is

usually seen as independent of the main system. Workflow management system in the central engine integrated scheduling task tasks and Hadoop framework MapReduce tasks.

HDFS in Hadoop has two types of nodes, and operates in Master / Slave mode of operation, that is, a Namenode (Master) and multiple Datanodes (Slaves). There are two control-type nodes, a Jobtracker, and a series of Tasktracker during job execution. Jobtracker coordinates all tasks running on the system by scheduling tasks on Tasktracker; Tasktracker runs the, While periodically sending the running information to Jobtracker. Normally, one machine is selected as the Name Node in the cluster and the other is the Job Tracker, which is Masters. The remaining machines act as Data Nodes and as Task Takers, which are Slaves.

Optimization of the scheduling algorithm. The algorithm realizes that in the whole process, if some subprocess or the whole flow is MapReduce kind of business, mark it with the color (different color mark different affairs class), in order to distinguish the general business-intensive type task. As shown in Figure 3, the workflow engine in the workflow analysis, decomposition, according to the (color) tag selected MapReduce class transaction transferred to the Hadoop platform processing, and other types of things are assigned to the cloud environment, the other basis Processing nodes (or other platforms, do not do here to expand). Workflow engine will be combined with Hadoop HDFS feedback back to the node real-time information to make a comprehensive scheduling decisions. Hadoop is assigned to the task of the bottom of the management and fault-tolerant aspects are also dependent on Hadoop. In general, this enables the cloud workflow system to take full advantage of Hadoop's advantages in distributed systems.

Model language description. The model description language used in this paper is BPEL4WS (Simple Business Process Execution Language for Web Services). A workflow composed of MapReduce tasks can be described as DAG graph $W = (J, E, D, C, T)$ Each node is a job task, each MapReduce type of task can be Hadoop platform MapReduce parallel computing framework to complete. The scheduler allocates tasks partitioned by job into appropriate slots according to the task class.

Algorithm scheduling process. For DAGs, which are generally composed of indivisible job tasks, [6] shows that HEFT performs better than most other heuristic algorithms in most cases, especially for large, highly parallel DAG graphics. The model corresponding to this algorithm is designed to deal with large-scale data-intensive tasks, the choice of reference to the HEFT algorithm. The model is characterized by: DAG in each job can be split into several tasks, part of the tasks can meet the priority constraints in the case, is scheduled to slot the idle time gap, even if the gap is not enough to accommodate The whole job. Through this process, we can achieve higher efficiency and make the entire project process execution time shorter. The scheduling process of the whole algorithm consists of two parts:

A) The first sub-process is intended to sort the job task nodes in the DAG to generate a priority queue, provided that the precedence relation constraint must be satisfied. By starting from the egress, traverse from bottom to top to recursively calculate the value of each task, and then sort the task value by non-incremental way to generate job scheduling sequence priority_queue.

B) The second sub-process with the help of HEFT's plug-in idea, combined with the MapReduce parallelization mechanism. It dispatches the job according to the priority_queue obtained earlier, and assigns each partition task of the corresponding job to the appropriate processor slot.

$$t_earliest_start_time[n_i, s_i] = \max\{ready_Time(j_j), slot_availTime(s_i)\} \quad (1)$$

$$t_earliest_finish_time[n_i, s_i] = t_earliest_start_time[n_i, s_i] + w(t_i, s_i) \quad (2)$$

$$readyTime(j_j) = \max_{nm \in pred(n_i)} (actual_finish_time(n_m) + c_{m,i}) \quad (3)$$

The actual earliest start time and end time are calculated by the algorithm Eq. 1,Eq. 2 and Eq. 3 to calculate the actual completion time. If all the job nodes in the DAG are scheduled, the schedule length

of the entire project flow is equal to the actual completion time of the last task in the egress. Because of the parallel strategy of Map Reduce, the smallest unit of scheduling is not a relatively large job, but a task that is cut out. In general, task execution time is significantly shorter than the job.

The Advantage of Algorithm. In the case of priority constraints, a special feature of the algorithm is that these tasks are flexibly inserted into the idle timestamps of the processing node slots. In this way, the efficiency of the scheduler will be significantly improved, because it is even in the slot space gap is not enough to accommodate the entire job, but can choose part of the task inserted into the gap, so that the time debris to make it more fully .

According to the processing type of the task, the scheduling mechanism allocates it to the appropriate processing node, and improves the utilization rate of the machine with the help of the plug-in idea, which makes the overall scheduling length shorter than other mechanisms. The advantages of this algorithm: it can shorten the completion time of the whole process, improve the efficiency of task execution, improve the utilization of equipment and reduce the waste of resources.

Conclusions

This paper investigates the scheduling algorithm of cloud workflow, and explores the method and advantage of using distributed heuristic scheduling algorithm to deal with the bottleneck caused by large data flow. It provides the idea for the processing of large and complicated process, especially improves heuristic algorithm to achieve efficient and controllable function. The downside is that there is no valid data to verify the validation of this algorithm, and we will further prove the effectiveness of the algorithm in the future.

Acknowledgements

This paper is sponsored by management science and engineering project of shanghai polytechnic university (No. XXKPY1606).

References

- [1]Altintas I, Barney O, Jaeger-Frank E. Provenance collection support in the kepler scientific workflow system. Springer Berlin Heidelberg, (2006), p. 118-132.
- [2]De Nicola A, Missikoff M , Smith F. Towards a method for business process and informal business rules compliance. Journal of Software: Evolution and Process , No.3 Vol.24(2012), p. 341-360.
- [3]Jehn K A, Rispens S, Thatcher S. M. The effects of conflict asymmetry on work group and individual outcomes. Academy of Management Journal, No.3 Vol.53(2010), p. 596-616.
- [4]Liu H, Abraham A, Hassanien A E. Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm. Future Generation Computer Systems, No.8 Vol.26(2010), p. 1336-1343.
- [5]Ming G, Li H. An Improved Algorithm Based on Max-Min for Cloud Task Scheduling. Springer Berlin Heidelberg , (2012), p. 217-223.
- [6]Topcuoglu H, Hariri S, Wu M y. Performance-effective and low-complexity task scheduling for heterogeneous computing. Parallel and Distributed Systems, IEEE Transactions on, No.3 Vol.13(2002), p. 260-274.