

Parallel computation method for incremental cloud services based on Hadoop

Li Yonghong¹, Zhou Na² and Zhao Guofeng²

1.Earthquake Administration of Shandong Province, Jinan 250014, China

2.China Earthquake Networks Center, Beijing 100045, China

Keywords: Parallel computation ; incremental cloud services ; Hadoop

Abstract. Performance of Hadoop platform is closely related to the task scheduler. Based on the analysis of the existing Hadoop platform's job scheduling algorithm, I propose a multi-queue job scheduling optimization algorithm in the paper. Based on the actual test and analysis of Hadoop platform, it can be seen that the optimization algorithm proposed in this paper can effectively allocate the node resources according to the degree of demand for the node resource. It can achieve the resource sharing among multiple queues. And at the same time, it can also effectively avoid the ping-pong effect brought by resource competition at the same time. Optimization algorithm has greater improvements in execution efficiency than Hadoop default algorithm.

Introduction

With the rapid development of science and technology, the large data age comes as scheduled. The emerging data access methods such as Internet of Things and Cloud Computing have made the data volume increase geometrically and exponentially. Traditional data theory research and practical application methods are also affected by the large data age. At present, the four typical characteristics of large data age are: large data amount, wide variety, low value density, and fast processing speed [1]. At present, the large data age has come quickly, but theoretical research on large data is still weak. How to extract the useful data from a large amount of redundant information among the massive access data is a practical and valuable research [1, 2]. Driven by the application background, a large number of scholars gradually generate a strong interest in big data extraction algorithm, promoting the field to become a research hotspot. Through the four characteristics in the large data age, we can deduce that the operating efficiency of algorithm is the most important factor in large data theory research, which is the key factor affecting the practical application effect of algorithm.

Hadoop parallel computation solution

Mapreduce parallel computation model under traditional Hadoop platform is a parallel computation strategy. Grouping individual reduction strategy is adopted in the mode [9], namely operation of replacing group members by typical individuals will be adopted to reduce the data processing amount (see Fig. 2). In the typical individual operation method, item identifier list (pID) is designed in operation process to map the ID in process to the group IDS (mID), and then carry out overall reducing strategy for the group. For overall individuals, the time complexity of reduce operation algorithm is $\log_2(p)$. But for grouping plan of typical individual operation, the time complexity of reduce operation algorithm is $\log_2(p')$, of which $p' = \max_k(|P_k|)$, and P_k is the typical individual quantity in process subset. So we expect $p' \sim p$ during the algorithm realization process and this condition can be achieved by controlling grouping quantity and selecting conditions of typical individuals.

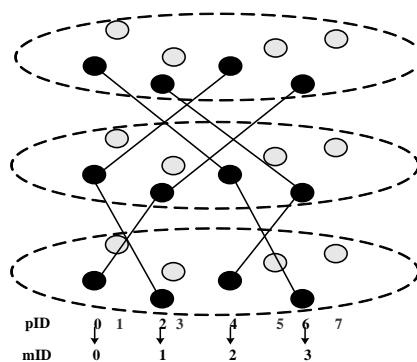


Fig. 1 Data-reducing Strategy

The group-based typical individual reduction strategy proposed in literature [2] can realize the information communication by defining MPI communication function in the group. However, since list of operational process identifiers is changing as iterations are performed, namely object group which carry out reduction is changing continuously, list of operational process identifiers shall be changed in the iteration and it may increase the communication overhead through MPI communication function. As the increasing of iteration, the center of gravity will nearer the center of cluster. So we shall firstly consider only on center of gravity and one cluster mode of process subset. In order to solve the high time occupation rate of MPI communication function, we shall replace the traditional MPI communication mode by synchronized group member management agreement and based on the group reduction operation, we can carry out parallel independent computation in process subset. It is suitable for the inserting of parallel algorithm.

In the paper, two-level scheduling strategy is introduced in multi-queue job scheduling. Each user group has two job queues: job waiting queue and job operating queue. In the job waiting queue, job shall be sorted according to priority and submission time, and then it will be placed into job operating queue for scheduling after initialization. A certain resource quota shall be allocated for each job queue. Quota management method is the same as the Capacity scheduling algorithm. But preemption mechanism is added to the resource quota recovery, as shown in Fig. 2. When the job operating queue has not reached its quota, and the quantity of waiting tasks in the queue is bigger than the total number of recycled resource, corresponding resource recycling objects will be created and they will be added into the resource recycle list. Objects for resources recycling include resource quantity (difference value between the quota value and current held resource quantity) needed to recycle and resource recycling timeout information. In the scheduling algorithm, resource recycling list of each operating queue will be checked and resource recycling objects who are timeout recycled will be scheduled.

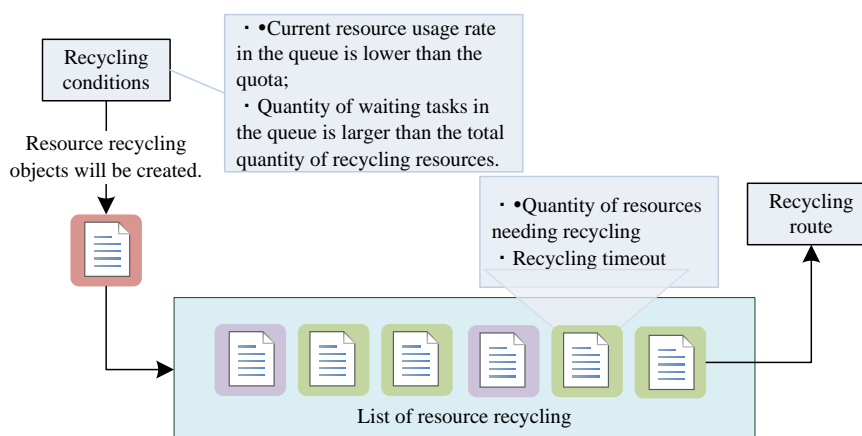


Fig.2 Preemption-supported Resource Recycling Process

As preemption is supported in queue resource quota management, it is possible to result in frequent preemption and recycling resources among multiple queues, resulting in a ping-pong effect and waste of system resources. Therefore, we classify queues into two types: shared queues and non-shared queues. Shared queues can only use free resources of other shared queues. Switching of

operating queue between shared queue and non-shared queue is based on the minimum resource quota and resource sharing threshold. By classification of queue type and reasonable setting of resource sharing threshold, we can effectively avoid the “ping-pong effect” .

Algorithm performance analysis

This experiment mainly compares the algorithm performance from the precision and efficiency of the evaluation algorithm. Eight true datasets are adopted to generate a larger mixed dataset as the test dataset. The eight datasets are composed of 10000 vectors, dimension of test dataset at each node meets geometric multiple of 2 (2, 4, 8, 16) and test datasets are all subject to [0, 1] normal distribution. In literature [3], cluster cost is adopted to evaluate the accuracy of K-means II algorithm. The method is used in the experiment to carry out simulation experiments for three algorithms, namely K-means II , FPCM and PGR-PFCM, and all experiments shall operate for 30 times to compute the average value. The simulation results are shown in Fig. 3-4.

Costs of test datasets for the three algorithms are shown in Fig. 3 (iteration step). It can be seen that in low dimension, costs of the three algorithms have few difference. With the increasing of data dimension, computation costs of the three algorithms are increasing rapidly, and the mutual cost difference is increasing, reflecting algorithm referred in the paper has performance advantage in high dimension dataset. Comparison of operating time for the algorithms is shown in Fig. 3. In the low dimension data, as there is few difference of costs by different algorithms, while PGR-PFCM algorithm has added agreement list operation, resulting in the increasing of operating time. But as the increasing of data dimension, growth rate of computing time for PGR-PFCM algorithm is shorter than the other two algorithms. Operating efficiency advantage of PGR-PFCM algorithm is shown in Fig. 3-4. For cluster success rate of the three algorithms under various dimension datasets (each algorithm shall operate for 30 times to compute the average value), it can be seen average cluster success rate of PGR-PFCM is the highest among the three algorithms, K-means II algorithm rates secondly and success rate of FPCM algorithm is the lowest.

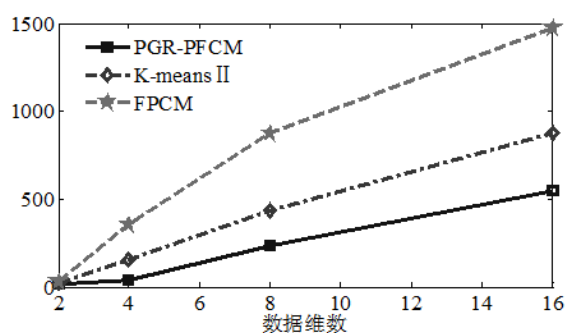


Fig. 3 Computing Cost Comparison of Algorithms

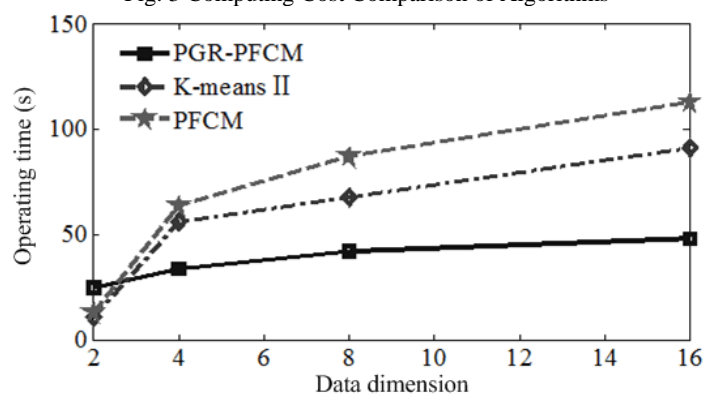


Fig. 4 Computing Time Comparison of Algorithms

Conclusion

In this paper, I carry out research on the job scheduling algorithm of Hadoop platform, and propose a multi-queue scheduling optimization algorithm which supports the job type classification. The optimization algorithm supports the node to allocate different types of work according to current workload, so as to improve the resource utilization of the nodes; free resources in operating queue is allowed to be used by other operating queues; when it is needed by the original operating queue, the resource can be recycled timely, namely task preemption is supported during the recycling process; logical division of list for shared queues and non-shared queues are adopted to avoid Ping-pong effect. Performance test results of Hadoop platform shows that compared with system default algorithm, the optimization algorithm has significantly improvements in execution efficiency of job scheduling and execution stability.

Acknowledgement

Shandong province key research and development project "cloud computing environment seismic waveform data management software development" (item number 2015GSF120012).

Reference

- [1] Yang R Z, Xiao W Q, Kai-Xi H U, et al. Method for Thirty-years Climate Normals Generation Based on Cloud-computation Platform[J]. Computing Technology & Automation, 2013.
- [2] Ibidunmoye O, Hern&#, Ndez-Rodriguez F, et al. Performance Anomaly Detection and Bottleneck Identification[J]. Acm Computing Surveys, 2015, 48(1):1-35.
- [3] Karun A K, Chitharanjan K. Locality Sensitive Hashing based incremental clustering for creating affinity groups in Hadoop — HDFS - An infrastructure extension[C]// International Conference on Circuits, Power and Computing Technologies. 2013:1243-1249.
- [4] K"Amf M, W. Kantelhardt J. Hadoop. TS: Large-Scale Time-Series Processing[J]. International Journal of Computer Applications, 2013, 74(17):1-8.
- [5] Wang Y, Luo J, Song A, et al. OATS: online aggregation with two-level sharing strategy in cloud[J]. Distributed & Parallel Databases, 2014, 32(4):467-505.