Enhance Performance of Mapreduce Job on Hadoop Framework using Setup and Cleanup

Abstract

MapReduce is an effective programming model for large-scale data-intensive computing applications. Hadoop is an open-source implementation of MapReduce which has been widely used. The communication overhead from the big data sets’ transmission affects the performance of Hadoop greatly. In consideration of data locality, Hadoop schedules tasks to the nodes near the data locations preferentially to decrease data transmission overhead, which works well in homogeneous and dedicated MapReduce environments. However, due to practical considerations about cost and resource utilization, it is common to maintain heterogeneous clusters or share resources by multiple users. Unfortunately, it’s difficult to take advantage of data locality in these heterogeneous or shared environments [1]. To improve the performance of MapReduce in heterogeneous or shared environments, a data prefetching mechanism is proposed. In this paper, we can fetch the data to corresponding compute nodes in advance. It is proved that the proposal of this paper reduces data transmission overhead effectively with theoretical analysis. We also work on applying similar prefetching mechanisms to other phases in MapReduce, and researching on predicting the execution nodes of tasks in
cluster computing to improve performance and the result are clearly shows that proposed system will takes a less execution time as compared to existing mapreduce job.

References

4. “Improve the MapReduce Performance through complexity and performance based on data placement in Heterogeneous Hadoop Cluster ” Rajashekar M. Arasanal, Daanish U. Ruman Department of Computer Science University of Illinois at Urbana-Champaign.
7. Improving MapReduce Performance Using Smart Speculative Execution Strategy Qi Chen, Cheng Liu, and Zhen Xiao, Senior Member, IEEE 0018-9340/13/$26.00 © 2013 IEEE


Index Terms

Computer Science

Information Sciences

Keywords

Big data, Hadoop, Mapreduce, performance, prefetching mechanism, setup & cleanup