Hadoop’s Optimization Framework for Map Reduce Clusters

Aman Lodha

Department of Information Technology, Savitribai Phule Pune University India

Abstract: Most widely used frameworks for developing MapReduce-based applications is Apache Hadoop. But developers find number of challenges in the Hadoop framework, which causes problem to management of the resources in the Map Reduce cluster, that will optimize the performance of MapReduce applications running on it. In the proposed system we consider the constraints in the resource allocation process in the MapReduce programming model for large-scale data processing for speed up performance. For that purpose we propose the novel technique called Dynamic approach for performing speed up of the available resources. It contains the two major operations: they are slot utilization optimization and utilization efficiency optimization. The Dynamic technique has the three slot allocation techniques they are Dynamic Hadoop Slot Allocation (DWSA), Speculative Execution Performance Balancing (SEPB), and Slot Pre-scheduling. It achieves a performance speedup by a factor of over the recently proposed cost-based optimization (CBO) approach. In addition performance benefit increases with input data set size.

Keywords: MapReduce, HDFS, Dynamic hadoop slot allocation, slot prescheduling etc.

1. Introduction

MapReduce is a popular computing framework for large-scale data processing. Practical experience shows that inappropriate configurations can result in poor performance of MapReduce jobs, however, it is challenging to pick out a suitable configuration in a short time. Also, current central resource scheduler may cause low resource utilization, and degrade the performance of the cluster. Appropriate configuration settings could reduce execution time of jobs by using cluster resources efficiently and avoiding unnecessary disk I/Os. Previous configuration tuning works can be categorized into three groups: following best practices and MapReduce tuning guides, offline configuration tuning, and online configuration tuning. Online tuning systems search appropriate configuration by dynamically assigning test configurations to running tasks in the job. However, there are multiple drawbacks in current online approach. Firstly, the searching strategies for finding optimal configuration take little consideration to characteristics of MapReduce; secondly, they neglect efficient resource utilization in the whole system; Thirdly, after a desirable configuration is achieved, the job uses the same configuration afterwards. However, the configuration might not be suitable for latter tasks because of data skew. Inappropriate configuration can cause a task being killed due to out of memory error. While tuning configuration parameters improves task performance, using cluster resources efficiently can also achieve significant performance improvement.

A. MapReduce Framework:

The mapreduce framework consists of two steps namely Map step and reduce step. Master node takes large problem input and slices it into smaller sub problems and distributes these to worker nodes. Worker node may do this again and leads to a multi-level tree structure.

2. Literature Survey

Zhendong Bei, Zhibin Yu, Huiling Zhang, Wen Xiong, Chengzhong Xu, Senior, “RFHOC is an automated performance tuning approach” that adjusts the Hadoop configuration parameters for an application running on a given cluster to achieve optimized performance. The model takes Hadoop configurations as input and outputs a performance
prediction. In a subsequent step, we then use the performance prediction models for each phase as part of a genetic algorithm to search for the optimum Hadoop configuration for the application of interest. This system automatically tunes the Hadoop configuration parameters and build analytical models based on oversimplified assumptions, affecting the overall model’s accuracy and ultimately the achievable performance improvements.[1]

Xiaoan Ding, Yi Liu, Depei Qian, “JellyFish: Online Performance Tuning with Adaptive Configuration and Elastic Container in Hadoop Yarn”, this paper proposes an online performance tuning system, JellyFish, to improve performance of MapReduce jobs and increase resource utilization in Hadoop YARN. JellyFish continually collects real-time statistics to optimize configuration and resource allocation dynamically during execution of a job. During performance tuning process, JellyFish firstly tunes configuration parameters by reducing the dimensionality of search space with a divide-and-conquer approach and using a model-based hill climbing algorithm to improve tuning efficiency; secondly, JellyFish re-schedules resources in nodes by using a novel elastic container that can expand and shrink dynamically according to resource usage, and a resource re-scheduling strategy to make full use of cluster resources. The Proposed system does not support multiple resource rescheduling strategy and multi-tenant Hadoop environment. [4]

Dili Wu and Aniruddha Gokhale A, “Self-Tuning System based on Application Profiling and Performance Analysis for Optimizing Hadoop MapReduce Cluster Configuration” in this paper the PPABS framework comprises two distinct phases the Analyzer, which trains PPABS to form a set of equivalence classes of MapReduce applications for which the most appropriate Hadoop configuration parameters that maximally improve performance for that class are determined, and the Recognizer, which classifies an incoming unknown job to one of these equivalence classes so that its Hadoop configuration parameters can be self-tuned. Experimental results comparing the performance improvements for three different classes of applications running on Hadoop clusters deployed on Amazon Ee2 show promising results. Limitation of system despite its popularity, however, application developers face numerous challenges in using the Hadoop framework, which stem from them having to effectively manage the resources of a MapReduce cluster, and configuring the framework in a way that will optimize the performance and reliability of MapReduce applications running on it.[2]

Chi-Ou Chen, Ye-Qi Zhuo, Chao-Chun Yeh, Che-Min Lin, Shih-wei Liao, “Machine Learning-Based Configuration Parameter Tuning on Hadoop System” In this paper, he focus on optimizing the Hadoop MapReduce job performance by tuning configuration parameters, and then we propose an analytical method to help system administrators choose approximately optimal configuration parameters depending on the characteristics of each application. approach has two key phases: prediction and optimization phase. The prediction phase is to estimate the performance of a MapReduce job, whereas the optimization phase is to search the approximately optimal configuration parameters strategically by invoking the predictor repeatedly. In our evaluation results, our work can help system administrators to improve the performance about 2X to 8X better than traditional methods. This current system is evaluated in cluster with the same nodes capability. [3]

3. Proposed System

4. System Analysis

After the analysis of the data, Hadoop is installed on a windows Machine and the sample data is inserted into the repository. And later on, the Map-Reduce code is executed along with an initialization cluster. We can browse all the files in the default URL using the localhost. We need to specify all the paths for running the map-reduce like the input directory, output directory and the cluster initialization directory along with the algorithm we used. Hadoop provides a reliable output in mapping and analyzing
of the data. This Data can be employed for further representation.

5. Result Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Before Processing</th>
<th>After Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of the File</td>
<td>15000 BYTES</td>
<td>14000 BYTES</td>
</tr>
<tr>
<td>No. of Records</td>
<td>300</td>
<td>70</td>
</tr>
<tr>
<td>No. of Attribute</td>
<td>60</td>
<td>55</td>
</tr>
</tbody>
</table>

6. Conclusion

The proposed efficient data aware MapReduce framework is powerful for big data management. Results show that there is substantial improvement in performance of Hadoop jobs by reducing completion time and storage overhead using efficient data aware caching for big data application.

7. References


