Revisiting Performance in Big Data Systems: An Resource Decoupling Approach

Chen Yang[1], Qi Guo[2], Xiaofeng Meng[1], Rihui Xin[1] and Chunkai Wang[1]
[1]School of Information, Renmin University
[2]Institute of Computing Technology, Chinese Academy of Sciences
[1]{yang_chen, xfmeng, xinrihui, wangchunkai}@ruc.edu.cn, [2]guoqi@ict.ac.cn

Big data systems for large-scale data processing are now in widespread use. To improve their performance, both academia and industry have expended a great deal of effort in the analysis of performance bottlenecks. Most big data systems, as Hadoop and Spark, allow distributed computing across clusters. As a result, the execution of systems always parallelizes the use of the CPU, memory, disk and network. If a given resource has the greatest limiting impact on performance, systems will be bottlenecked on it. For a system designer, it is effective for the improvement of performance to tune the bottleneck resource. The key point for the aforementioned scenario is how to determine the bottleneck resource. The nature clue is to quantify the impact of the four major components and identify one causing the greatest impact factor as the bottleneck resource.

The current work[1] as a white-box analysis tries to extract the impacted time from the in-system execution stages using a given resource and evaluate the impact of resources on performance. However, it only works on I/O components, and results are error-prone. The main reason is the mismatch between the resource usage and the impacted execution stages. Firstly, the big data systems are complex, so it is difficult to find all factors that may be impacted by a given resource, such as the neglect of time blocked by major page faults. Secondly, different resources are parallelized due to asynchronism such that it is difficult to separate the impact of a given resource, such as the extraction of time blocked by network I/O from shuffle time. Therefore, the evaluation of the impact of resources is not easy through a white-box analysis.

To overcome this issue, we treat this problem from a completely different perspective that is the black-box analysis. When a given resource is upgraded, the performance will be improved. Based on this observation, we present a **resource decoupling approach** to evaluate the impact of resources. The main idea is that we decouple the resources by upgrading them and solve their impact factors by an intensiveness degree model later. We run the system and observe the speedup after upgrading a single resource, so that the performance improvement is only related to the given resource, and the resource impact has thereby been isolated in the speedup collected. We repeat this process for all other resources, so they will be decoupled dynamically. Essentially, the speedup is a combination of all the impacts of a given resource on performance, so that we do not focus on the exact match between the resource usage and the execution stages within systems. Furthermore, an intensiveness degree model addressed by us evaluates impact factors of four major resources on performance. Our model uses the actual speedup as an input and solves the quantitative intensiveness degrees, which are comparable to each other. Thus, a resource resulting in the maximum will be determined as the bottleneck.

We test Spark 1.6.2 on two benchmarks (BDbench and TPC-DS) with the on-disk and in-memory workloads (caching data in memory or not). The intensiveness degree ranking is 0.57 (CPU), 0.23 (memory), 0.22 (disk) and 0.04 (network), so the CPU is the bottleneck resource. The CPU intensiveness degree will be reduced due to the high overhead of memory access when running in-memory workloads, causing them to be occasionally slower than on-disk workloads.

Resource decoupling, which is not complicated for practice, treats the big data system as a black-box and only collects the speedup, so that it is general, not only for a specific system. We also build a prototype system DRes that implements the resource decoupling approach to automatically find the bottleneck resource. It might be valuable for the cloud server vendors to help users choose an appropriate cluster environment by providing a small and upgradeable cluster to quantify the application bottlenecks before purchase. The experimental findings might also help users to tune Spark.

SoCC '17, September 24–27, 2017, Santa Clara, CA, USA

ACM ISBN 978-1-4503-5028-0/17/09. . . $15.00
https://doi.org/10.1145/3127479.3132685

ACKNOWLEDGMENTS

This research was partially supported by grants from the Natural Science Foundation of China (No. 61532016, 61532010, 61379050, 91224008); the National Key Research and Development Program of China (No. 2016YFB1000602, 2016YFB1000603); Specialized Research Fund for the Doctoral Program of Higher Education(No. 20130004130001), and the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University(No. 11XNL010).

REFERENCES

[1] Kay Ousterhout, Ryan Rasti, Sylvia Ratnasamy, Scott Shenker, and Byung Gon Chun. Making sense of performance in data analytics frameworks. In *12th USENIX Symposium on Networked Systems Design and Implementation*, pages 293–307, 2015.