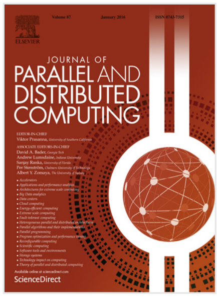
Accepted Manuscript



A cost minimization data allocation algorithm for dynamic datacenter resizing

Wuhui Chen, Incheon Paik, Zhenni Li, Neil Y. Yen

PII: S0743-7315(17)30098-9
DOI: http://dx.doi.org/10.1016/j.jpdc.2017.03.010
Reference: YJPDC 3652

To appear in: *J. Parallel Distrib. Comput.*

Received date: 20 January 2017
Revised date: 28 February 2017
Accepted date: 21 March 2017

Please cite this article as: W. Chen, I. Paik, Z. Li, N.Y. Yen, A cost minimization data allocation algorithm for dynamic datacenter resizing, *J. Parallel Distrib. Comput.* (2017), http://dx.doi.org/10.1016/j.jpdc.2017.03.010

A Cost Minimization Data Allocation Algorithm for Dynamic Datacenter Resizing

Wuhui Chen, Incheon Paik, Zhenni Li, and Neil Y. Yen, *Member, IEEE*

Abstract—Modern datacenters dynamically adjust the number of active servers in different geographic regions to adapt to the dynamic workloads from user requests and electricity price heterogeneity. One of the main challenges for datacenter resizing is that the heavy network traffic among datacenters causes significant deterioration of the overall performance and considerably increases the operational expenditure of datacenters. In this paper, we propose an efficient data allocation technique that considers both the static and dynamic characteristics of datacenters to enable more efficient datacenter resizing. We first formulate the optimal data allocation problem, propose a generic model for minimizing the communicating cost in datacenter resizing, and show that the data allocation problem is NP-hard. To produce feasible solution in polynomial time, we propose a heuristic algorithm considering the traffic flow in the network topology of datacenters by first transforming the data allocation problem into a chunk distribution tree (CDT) construction problem, and then reducing the CDT construction to a graph partitioning problem. The experimental results show that our efficient data allocation approach can improve the performance of MapReduce operations effectively with lower communicating and computing costs for datacenter resizing.

Index Terms—Big data processing, cost minimization, datacenter resizing, data allocation, MapReduce operation

1 Introduction

The data explosion in recent years has led to a fast rising demand for big data analytics in modern datacenters, which are usually distributed in different regions geographically, e.g., Google's 14 datacenters are found in over eight countries on four continents [1]. An important fact for the geographically distributed datacenters is that the electricity prices generally vary with location and time, i.e., they show spatial and temporal heterogeneity. As shown in Fig. 1 (a), the electricity prices fluctuate with time in both Mountain View and Houston, and can differ among all three places shown at any time during the day, e.g., $38.52 USD/MW-h and $48.14 USD/MW-h for Houston and Mountain View, respectively, at 1:00 pm [2]. Another fact is that the arrival times of MapReduce jobs follow a Poisson distribution. Fig. 1 (b) shows the number of arriving job requests per 10-minute interval during a one-day period. The workload reaches the first daily peak between 1:00 am and 4:00 am, and the second peak between 1:00 pm and 4:00 pm [3]. Motivated by such facts, a representative and promising way of reducing the operational expenditure of datacenters would be adjusting the number of activation servers in different datacenters, generally known as datacenter resizing [2], [4], to adapt to dynamic workloads from user requests and electricity price heterogeneity. An intuitive way to do this is to activate more servers at datacenters with a low electricity price or heavy workload. Fig. 1 (c) shows the number of servers each location should run to meet the workload requirements and quality-of-service constraints for all the front-end Web portal servers. It can be observed that the number of activation servers at each location varies with the electricity price and time in Google's geographically distributed datacenters [2].

Moreover, communicating resources have become increasingly important and even critical in big data processing, in particular for geographically distributed cloud datacenters, where large data volumes are frequently transferred among such centers because of input reads, intermediate data shuffles, and output writes during the map, shuffle, and reduce steps [5], [6], [7]. For example, in [6], the authors conduct an experimental and analytical study to identify some performance issues inherent to geographically distributed datacenters, and argue that mechanisms that can reduce the volume of intermediate data being transferred across distributed datacenters are critical for improving the performance of Hadoop cluster deployment. In addition to involving heavy computation loads, large-scale cloud systems have a large I/O data volume flow per CPU per second. For instance, typical outflow and inflow traffic ranges from 10 MB/s to 20 MB/s in the Hadoop cluster with 2000 computing nodes at *Taobao* [3]. Another example is *BigBench* [8], in which the geographically cross-datacenter traffic is about 706 GB/day, raising the cost of providing services. With growing data volumes and the scaling of clusters, network transfer becomes a potential bottleneck for the whole cloud computing system. Thus, to achieve high throughput in geographically distributed datacenters for big data processing, data transfers during the execution of tasks need to be minimized, because the CPU utilization of the server nodes would decrease when there is a large volume of pending traffic data transfers.

To lower network traffic for MapReduce workloads, data allocation, which brings data chunks (blocks) closer to data consumers (e.g., MapReduce operations), is seen

- *Wuhui Chen is with the School of Data and Computer Science, Sun Yat-sen University, China. E-mail: chenwuhui21@gmail.com.*
- *Incheon Paik, Zhenni Li and Neil Y. Yen are with the School of Computer Science and Engineering University of Aizu, Fukushima, Japan. E-mail: paikic@u-aizu.ac.jp, lizhenni2012@gmail.com, neilyyen@u-aizu.ac.jp.*