Future Generation Computer Systems 89 (2018) 31–43

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A locality-aware shuffle optimization on fat-tree data centers

Jihe Wang [a], Danghui Wang [a], Meikang Qiu [b,c,*], Yao Chen [d,e], Bing Guo [f]

[a] School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, SX 710072, China
[b] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China
[c] Department of Electrical Engineering, Columbia University, New York City, NY, 10027, USA
[d] Advanced Digital Sciences Center, University of Illinois at Singapore Pte Ltd, Singapore
[e] School of Computers, Guangdong University of Technology, Guangzhou, China
[f] Computer Science College, Sichuan University, Chengdu, SC 610064, China

HIGHLIGHTS

- Reduce core bandwidth consumption by scheduling.
- Use a similarity-based distance to evaluate the dynamic distance between leaf-nodes.
- Use the distance to schedule workers to avoid high intensity usage of resources.

ARTICLE INFO

Article history:
Received 23 November 2017
Received in revised form 21 May 2018
Accepted 10 June 2018
Available online 25 June 2018

Keywords:
Locality
Shuffle
Fat-tree network
Data-center
Node distance

ABSTRACT

Shuffle is a data exchanging phase that is always inserted between two adjacent computations to deliver intermediate results in data centers. It generates a burst of traffic that exhausts the network bandwidth, debasing the availability of the core layer facilities in fat-tree topologies. Previous researches follow either *flow inhibition* or *infrastructural upgrading* to achieve a high utilization of core network resources. However, dynamic pressure from shuffle burst introduces more unpredictable usage of core network that disturbs the global locality-based optimization on task schedule. In this work, we reduce the core bandwidth consumption by scheduling the location of adjacent computing workers based on our proposed distance model that uses a similarity-based distance to evaluate the dynamic distance between fat-tree leaf-nodes. Task assignment further utilizes this distance to schedule workers to avoid high intensity usage of core network resources. This design improves the performance of shuffle phase in popular on-data-center algorithms as well as maintains infrastructural inexpensiveness of their fat-tree topology. The proposed models are evaluated on a semi-physical simulation test platform and compared to state-of-the-art solutions, such as Space Shuffle and Scalable Shuffle. The results show that our design achieves an up to 18% speedup on shuffle procedure and a 23% extension of network capacity. In addition, a significant mitigation of congestion can be obtained on the bottleneck of core network.

1. Introduction

Computation in data centers generally create a large number of shuffle phases to deliver intermediate results among servers for successive processes [1]. For example, MapReduce [2] and Spark [3,4] are shuffle-intensive frameworks that burst a large amount of data fed from mappers to reducers for further classified processes. As a kernel phase of large scale distributed computing, highly concurrent shuffle arouses a comprehensive network activity with high consumption of bandwidth and energy [5–8], damaging the availability of network facilities. Therefore, network architectures in data centers always invoke fat-tree topology [9] to fairly deliver the transfer load and restrain the infrastructural cost [10]. However, in fat-tree network, the shuffle burst deteriorates the core layer network much more than periphery network. Hence it is critical to efficiently utilize the core layer resources during shuffle burst, otherwise, the core layer could suffer not only intermittent congestion but also expensive capacity expanding.

Previous shuffle optimization isolates shuffle activities from the topology of the infrastructure to claim a single-goal solution, either *flow inhibition* or *infrastructural upgrading*. Their pros and cons are depicted as Fig. 1. For *flow inhibition*, many methods focus on how

* Corresponding author at: College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China.
E-mail addresses: wangjihe@nwpu.edu.cn (J. Wang), wangdh@nwpu.edu.cn (D. Wang), mq2203@columbia.edu (M. Qiu), yao.chen@adsc.com.sg (Y. Chen), guobing@scu.edu.cn (B. Guo).

https://doi.org/10.1016/j.future.2018.06.016
0167-739X/

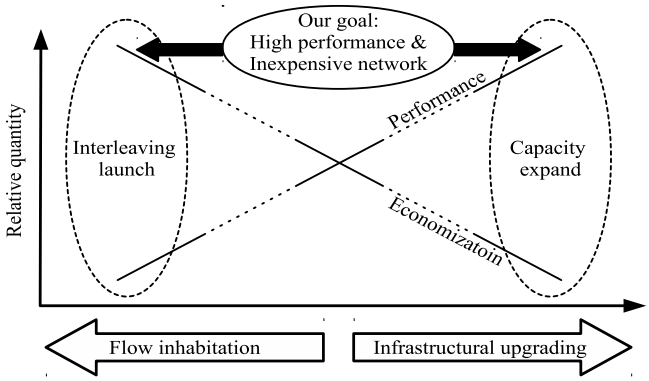

Fig. 1. The optimizing dimension of shuffle-oriented network for data centers.

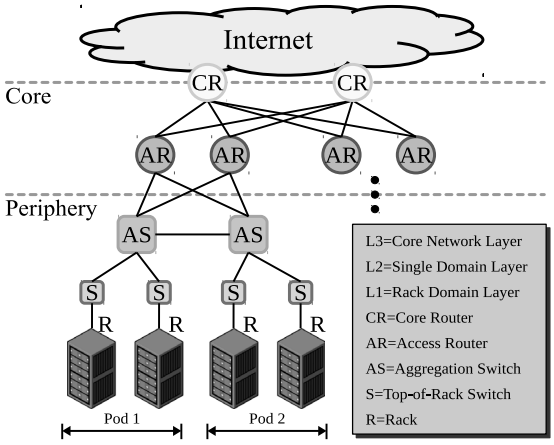

Fig. 2. Network architecture of fat-tree data centers.

to shrink the throughput volume to relieve the traffic congestion at bottlenecks. For example, Hadoop MapReduce [2] provides optional data compression to mitigate the network pressure. Other researches focus on interleaving shuffle to avoid burst data transmission over the upper-bound of the available bandwidth [11–13]. Those runtime solutions can achieve better economization and scalability, however, the transfer performance is significantly affected by the low compression speed and peak flow whittling. For *infrastructural upgrading*, many dedicated topologies are proposed to improve those bottleneck spots. For example, the work in [14] enhances the top-layer connectivity by an incremental deployment. However, the change of core infrastructure is always expensive and the performance-oriented optimization lacks of consideration on cost [9,15].

In this paper, we propose *vertical asymmetry* for rearrangement of the traffic load on fat-tree topology. *vertical asymmetry* depicts a phenomenon that core network provides narrower bandwidth but bears heavier traffic than periphery network. The structural asymmetry generates a busy core network surrounded by many idle periphery devices, thereby each of those shuffle tasks can only share a small part of core bandwidth. In the worst case, the intermediate results could paralyze the core facilities while the periphery bandwidth, on the contrary, is far from depletion. The unbalancing workload assignment on fat-tree is the major issue that leads to a weak tolerance [16] to large scale concurrency of shuffle tasks.

The goal of our work is to transpose the traffic workload vertically on fat-tree network so that huge shuffle workload only makes small impact on core network. To achieve this goal, we present a new shuffle scheduling method to suppress the huge volume of traffic on core network, including a *distorted distance model* to identify weight of each shuffle traffic, a *locality-aware shuffle assignment* to expel those heavy traffic from core network to periphery network (also called edge in our statement), and an *asymmetrical acceleration* to improve the utilization of peripheral bandwidth. In addition, a set of software methods are leveraged to improve the efficiency of both core and periphery facilities, avoiding expensive expansion of hardware.

We evaluate our approaches with a prototypical verification and performance comparison with stat-of-the-art design. Experiment results show that our design can achieve an up to 18% speedup on shuffle procedure and a 23% extension on network capacity. With our method, the concurrent transportation levers a hotter periphery networks and a cooler core network under a medium workload, providing sufficient capacity tolerance for heavier workload. Our core contributions include:

1. We address an asymmetric workload phenomenon in fat-tree data centers and setup an evidential causal relationship between the asymmetry and exhausted bandwidth on core network.
2. Distorted distance, a new dynamic data locality concept, is proposed to mitigate the over utilization of the core network by expelling the remote traffic to periphery network to improve the efficiency of the expensive bandwidth of the core network.
3. A centralized task management method is proposed to stimulate those shuffle tasks which have strong data coupling to be risen in a local region in the data centers. The results shows that this method is an efficient software strategy to enlarge the overall capacity of data centers without hardware upgrading.

The rest of this paper is organized as follows. Section 2 presents a brief background of shuffle and fat-tree. Section 3 introduces the motivation and methodology of this work. The distance model and task scheduling details are discussed in Section 4. Our kernel optimizing strategies are presented in Section 5. Experiment results of the evaluation is shown in Section 6, and the related work is presented in Section 7. Finally, Section 8 concludes the paper and shows our future work.

2. Background

In this section, we present a short review on fat-tree topology and shuffle progress in data centers, followed by a study on *vertical asymmetry* under common configurations.

2.1. Shuffle transfer in fat-tree data centers

In MapReduce framework, one dynamic set of computing node is duty to execute map functions, called MapTrackers and another similar set is dedicated for reduce function, called ReduceTackers. Organizing tens of thousands of computing nodes needs a flexible network topology for large-scale data exchange among nodes [17]. Fat-tree topology provides the advantages of *balance flow over network*, *rearrangeable non-blocking*, and *cheap infrastructure*, providing multiple configuration to enhance the transfer ability and economization. Fat-tree topology invokes full connection and associated connection to construct a strong core network and aggregation network layers separately, as shown in Fig. 2. However, the bandwidth provision of the core layer depends on the capability of the switches and the allocation of that capability, lead to a tolerant fragility when a large number of data flow pass through the core network in a short interval. Even worse, this kind of data burst is a