Efficient Distributed Workload (Re-)Embedding*

Monika Henzinger
Stefan Neumann
Stefan Schmid
Faculty of Computer Science, University of Vienna
Vienna, Austria

ABSTRACT

Modern networked systems are increasingly reconfigurable, enabling *demand-aware* infrastructures whose resources can be adjusted according to the workload they currently serve. Such dynamic adjustments can be exploited to improve network utilization and hence performance, by *moving* frequently interacting communication partners closer, e.g., collocating them in the same server or datacenter. However, dynamically changing the embedding of workloads is algorithmically challenging: communication patterns are often not known ahead of time, but must be *learned*. During the learning process, overheads related to unnecessary moves (i.e., re-embeddings) should be minimized. This paper studies a fundamental model which captures the tradeoff between the benefits and costs of dynamically collocating communication partners on ℓ servers, in an online manner. Our main contribution is a distributed online algorithm which is asymptotically almost optimal, i.e., almost matches the lower bound (also derived in this paper) on the competitive ratio of any (distributed or centralized) online algorithm.

CCS CONCEPTS

• **Theory of computation → Online learning algorithms**; • **Networks →** *Network algorithms*;

KEYWORDS

Networked systems, Online algorithms, Competitive analysis, Graph partitioning

ACM Reference Format:
Monika Henzinger, Stefan Neumann, and Stefan Schmid. 2019. Efficient Distributed Workload (Re-)Embedding. In *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '19 Abstracts), June 24–28, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3309697.3331503

1 CONTEXT AND MOTIVATION

Along with the trend towards more *data centric* applications (e.g., online services like web search, social networking, financial services as well as emerging applications such as distributed machine learning [8, 10]), comes a need to *scale out* such applications, and distribute the workload across multiple servers or even datacenters. However, while such parallel processing can improve performance, it can entail a non-trivial load on the interconnecting network. Indeed, distributed cloud applications, such as batch processing, streaming, or scale-out databases, can generate a significant amount of network traffic [9].

At the same time, emerging networked systems are becoming increasingly flexible and thereby provide novel opportunities to mitigate the overhead that distributed applications impose on the network. In particular, the more flexible and dynamic resource allocation (enabled, e.g., by virtualization) introduces a vision of *workload-aware* infrastructures which optimize themselves to the demand [2]. In such infrastructures, communication partners which interact intensively, may be *moved* closer (e.g., collocated on the same server, rack, or datacenter) in an adaptive manner, depending on the demand. This "re-embedding" of the workload allows to keep communication local and reduce costs. Indeed, empirical studies have shown that communication patterns in distributed applications feature much locality, which highlights the potential of such self-adjusting networked systems [3, 6, 11].

However, leveraging such resource reconfiguration flexibilities to optimize performance, poses an algorithmic challenge. *First,* while collocating communication partners reduces communication cost, it also introduces a *reconfiguration cost* (e.g., due to virtual machine migration). Thus, an algorithm needs to strike a balance between the benefits and the cost of such reconfigurations. *Second,* as workloads and communication patterns are usually not known ahead of time, reconfiguration decisions need to be made in an *online* manner, i.e., without knowing the future. We are hence in the realm of online algorithms and competitive analysis.

This paper studies the fundamental tradeoff underlying the optimization of such workload-aware reconfigurable systems. In particular, we consider the design of an online algorithm which, without prior knowledge of the workload, aims to minimize communication cost by performing a small number of *moves* (i.e., migrations). In a nutshell, we consider a *communication graph* between n vertices (e.g., virtual machines) which can be perfectly partitioned among a set of ℓ servers (resp. racks or datacenters) of a given *capacity*. We assume that the *communication patterns*, which partition the communication graph, consist of n/ℓ vertices and that once the whole communication graph was revealed, each server must contain exactly one communication pattern.

The communication graph is initially unknown and revealed to the algorithm in an online manner, edge-by-edge, by an adversary who aims to maximize the cost of the given algorithm. The cost

*Authors are ordered alphabetically. For the full version of this paper see [7].

SIGMETRICS '19 Abstracts, June 24–28, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6678-6/19/06.
https://doi.org/10.1145/3309697.3331503

here consists of *communication cost* and *moving cost*: The algorithm incurs one unit cost if the two endpoints (i.e., communication partners) of the request belong to different servers. After each request, the algorithm can reconfigure the infrastructure and move communication endpoints from one server to another, essentially *repartitioning* the communication partners; however, each move incurs a cost of $\alpha > 1$.

In other words, this paper considers the problem of *learning a partition*, i.e., an optimal assignment of communication partners to servers, at low communication and moving cost. Interestingly, while the problem is natural and fundamental, not much is known today about the underlying algorithmic challenges, except for the negative result that no good competitive algorithm can exist if communication partners can change arbitrarily over time [1]. This lower bound motivates us, in this paper, to focus on the online *learning variant* where the communication partners are unknown but fixed. At the same time, as we will show, the problem features interesting connections to several classic problems. Specifically, the problem can be seen as a *distributed* version of online caching problems [13] or an *online* version of the k-way partitioning problem [12].

2 OUR CONTRIBUTIONS

We initiate the study of a fundamental problem, how to learn and re-embed workload in an online manner, with few moves. We make the following main contributions.

We present a distributed $O((\ell \log \ell \log n)/\varepsilon)$-competitive online algorithm for servers of capacity $(1+\varepsilon)n/\ell$, where $\varepsilon \in (0, 1/2)$. We allow the servers to have $\varepsilon n/\ell$ more space than is strictly needed to embed its corresponding communication pattern (which is of size n/ℓ); we denote this additional space as *augmentation*. Such augmentation is needed, as our lower bounds discussed next show. For the formal details of our model see [7].

We show that there are inherent limitations of what online algorithms can achieve in our model: We derive a lower bound of $\Omega(1/\varepsilon + \log n)$ on the competitive ratio of any deterministic online algorithm given servers of capacity at least $(1+\varepsilon)n/\ell$. This lower bound has several consequences: (1) To obtain $O(\log n)$-competitive algorithms, the servers must have $\Omega(n/(\ell \log n))$ augmentation. (2) If the servers have $\Omega(n/\ell)$ augmentation (e.g., each server has 10% more capacity than the size of its communication pattern), our algorithm is optimal up to an $O(\ell \log \ell)$ factor. Thus, our results are particularly interesting for large servers, e.g., in a wide-area networking context where there is usually only a small number of datacenters where communication partners can be collocated (e.g., $\ell = 20$): if each datacenter ("server") has augmentation $0.1 \cdot n/\ell$, our algorithm is optimal up to constant factors.

The distributed algorithms we present not only provide good competitive ratios but they are also highly efficient w.r.t. the network traffic they cause. In fact, we show that for $\ell = O(\sqrt{\varepsilon n})$ servers, running the algorithms introduces only little overhead in network traffic and that this overhead is asymptotically negligible.

While the previous algorithms require exponential time, we also present polynomial time algorithms at the cost of a slightly worse competitive ratio of $O((\ell^2 \log n \log \ell)/\varepsilon^2)$.

As a sample application of our newly introduced model we present a distributed union find data structure [5, 14] (also known as disjoint-set data structure or merge-find data structure): There are n items from a universe which are distributed over ℓ servers; each server can store at most $(1+\varepsilon)n/\ell$ items and each item belongs to a unique set. The operation *union* allows to merge two sets. In our setting, we require that items from the same set must be assigned to the same server. To reduce the network traffic, our goal is to minimize the number of item moves during union operations. For example, when two sets are merged which are assigned to different servers, then the items of one of the sets must be reassigned to another server. We compare against an optimal offline algorithm which knows the initial assignment of all items and all union operations in advance. We obtain the same competitive ratios as above. We believe that this distributed union find data structure will be useful as a subroutine for several problems such as merging duplicate websites in search engines [4].

We also show that our algorithms solve an online version of the k-way partition problem.

The formal theorem statements and the proofs of all of the above mentioned results can be found in [7].

ACKNOWLEDGMENTS

We are grateful to Rachit Agarwal and the anonymous reviewers for their useful comments. The research leading to these results has received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement No. 340506. Stefan Neumann gratefully acknowledges the financial support from the Doctoral Programme "Vienna Graduate School on Computational Optimization" which is funded by the Austrian Science Fund (FWF, project no. W1260-N35).

REFERENCES

[1] Chen Avin, Andreas Loukas, Maciej Pacut, and Stefan Schmid. 2016. Online Balanced Repartitioning. In *DISC*.
[2] Chen Avin and Stefan Schmid. 2018. Toward Demand-Aware Networking: A Theory for Self-Adjusting Networks. In *CCR*.
[3] Theophilus Benson, Aditya Akella, and David A. Maltz. 2010. Network Traffic Characteristics of Data Centers in the Wild. In *IMC*. 267–280.
[4] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. 1997. Syntactic Clustering of the Web. *Computer Networks* 29, 8-13 (1997), 1157–1166.
[5] Bernard A. Galler and Michael J. Fischer. 1964. An improved equivalence algorithm. *Commun. ACM* 7, 5 (1964), 301–303.
[6] Monia Ghobadi, Ratul Mahajan, Amar Phanishayee, Nikhil Devanur, Janardhan Kulkarni, Gireeja Ranade, Pierre-Alexandre Blanche, Houman Rastegarfar, Madeleine Glick, and Daniel Kilper. 2016. ProjecToR: Agile Reconfigurable Data Center Interconnect. In *SIGCOMM*. 216–229.
[7] Monika Henzinger, Stefan Neumann, and Stefan Schmid. 2019. Efficient Distributed Workload (Re-)Embedding. *POMACS* 3, 1 (2019), 13:1–13:38.
[8] Mu Li, David G. Andersen, Jun Woo Park, Alexander J. Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J. Shekita, and Bor-Yiing Su. 2014. Scaling Distributed Machine Learning with the Parameter Server.. In *Proc. USENIX OSDI*, Vol. 14. 583–598.
[9] Jeffrey C. Mogul and Lucian Popa. 2012. What we talk about when we talk about cloud network performance. *CCR* (2012).
[10] Mohammad Noormohammadpour and Cauligi S Raghavendra. 2017. Datacenter Traffic Control: Understanding Techniques and Trade-offs. *IEEE Communications Surveys & Tutorials* (2017).
[11] Arjun Roy, Hongyi Zeng, Jasmeet Bagga, George Porter, and Alex C. Snoeren. 2015. Inside the Social Network's (Datacenter) Network. In *SIGCOMM*. 123–137.
[12] Ethan L. Schreiber, Richard E. Korf, and Michael D. Moffitt. 2018. Optimal Multi-Way Number Partitioning. *J. ACM* 65, 4 (2018), 24:1–24:61.
[13] Daniel D. Sleator and Robert E. Tarjan. 1985. Amortized efficiency of list update and paging rules. *Commun. ACM* 28, 2 (1985), 202–208.
[14] Robert Endre Tarjan and Jan van Leeuwen. 1984. Worst-case Analysis of Set Union Algorithms. *J. ACM* 31, 2 (1984), 245–281.