Network Scheduling in the Dark

Vojislav Đukić[1], Sangeetha Abdu Jyothi[2], Bojan Karlaš[1],
Muhsen Owaida[1], Ce Zhang[1], Ankit Singla[1]

[1]*ETH Zurich* [2]*University of Illinois at Urbana–Champaign*

Motivation. Advance knowledge of future events in a dynamic system can often be used to take actions that improve system performance. In data center networks, such knowledge could potentially benefit many problems, including routing and flow scheduling, circuit switching, packet scheduling in switch queues, and transport protocols.

Indeed, past work on each of these topics has explored this, and in many cases, claimed significant improvements [1–3]. Nevertheless, little of this work has achieved deployment in data centers, which largely use techniques that are agnostic to traffic information, such as shortest path routing with randomization, and first-in-first-out queueing at switches.

A significant roadblock for traffic-aware scheduling is that in practice, traffic characteristics can be hard to ascertain accurately in a timely fashion. In particular, past work on network flow and packet scheduling has assumed advance knowledge of *flow sizes.* In tightly controlled environments, developing an API for applications to expose such information is plausible, even though it could require changes to a large number of applications. However, even in such environments, the application itself may not know such information *a priori* – data analysis jobs, for instance, may start sending out the results of a computation before the execution finishes and the final size of the result is known. Further, for public cloud data centers, this API approach would require having their customers modify their applications.

Contribution. We thus examine both simple heuristics and learning methods to determine flow sizes in advance and evaluate their accuracy and utility. Our system, *Flux*, leverages behavioural patterns of cloud applications. It uses Gradient Boosting Decision Trees (GBDT) to correlate CPU, memory, disk, and network utilization to future network traffic. Flux entails no modifications to applications.

We tested Flux using various workloads – data processing on Spark, a Web workload on Apache Tomcat, training neural networks on TensorFlow – achieving high accuracy in flow size prediction, with R^2 values ranging from 73% to 97%.[1].

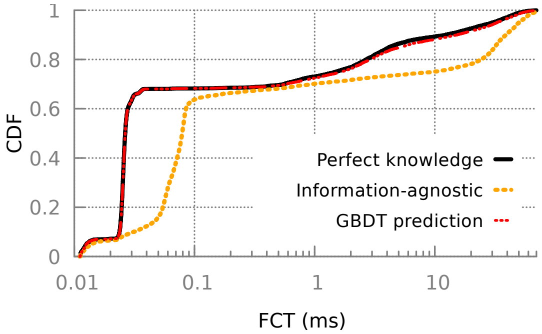

Figure 1: *Flow completion time (FCT) for pFabric for the SGD workload. Due to the log-scale, small visual differences are significant.*

We also explore the use of predicted flow sizes in three known network scheduling techniques [1–3], finding that Flux can reduce average flow completion times by 1.1× to 10.5× compared to information-agnostic techniques, even after accounting for the inaccuracies in our estimates. Figure 1 shows an example result, using the predicted flow sizes for packet scheduling using pFabric [1], for a Stochastic Gradient Descent Spark workload. Our early results show promise across workloads with substantial variation in underlying data and run configurations, indicating that our framework can learn which underlying system characteristics are predictive of traffic for different applications and workloads.

Conclusion. Our results indicate that accurate-enough flow size estimation is possible and it can be used to provide significant speedups when combined with existing network scheduling techniques. Thus, we are presently investigating the generality and limits of this approach.

REFERENCES

[1] M. Alizadeh, S. Yang, M. Sharif, S. Katti, N. McKeown, B. Prabhakar, and S. Shenker. pFabric: Minimal near-optimal datacenter transport. In *ACM SIGCOMM Computer Communication Review*, volume 43, 2013.

[2] P. X. Gao, A. Narayan, G. Kumar, R. Agarwal, S. Ratnasamy, and S. Shenker. pHost: Distributed Near-optimal Datacenter Transport over Commodity Network Fabric. CoNEXT '15, NY, USA, 2015. ACM.

[3] J. Perry, A. Ousterhout, H. Balakrishnan, D. Shah, and H. Fugal. Fastpass: A centralized zero-queue datacenter network. In *ACM SIGCOMM Computer Communication Review*, volume 44, 2014.

SoCC'18, October 11-13, Carlsbad, California USA

ACM ISBN 978-1-4503-6011-1/18/10.
https://doi.org/10.1145/3267809.3275458

[1] $R^2 = 1$ if the model produces perfect predictions, and $R^2 = 0$ if the model makes a prediction of zero value, always predicting the mean.