Problem statement

Marco Micera
Politecnico di Torino
marco.micera@gmail.com

Introduction

This is the problem statement for my Master's thesis, which will be written at the Technische Universität Darmstadt, Germany, during a six-month Erasmus+ exchange and under the supervision of Prof. Patrick Eugster †, M.Sc. Marcel Blöcher † and Prof. Fulvio Risso ‡.

1 In-network processing: a definition

Within this project, in-network processing (INP) refers to the technique that exploits network switches to modify and/or store data packets, without involving any kind of higher-layer devices. Therefore, approaches that make use of middle-boxes do not fall within our definition of INP. This is why INP is different from *active networking* and *Network Function Virtualization* (NFV).

2 Problem

Using INP to keep scaling data centers' performance seems a promising idea: Daiet [5] inventors claim to achieve an 86.9%-89.3% traffic reduction, hence reducing servers' workload; NetChain [2] can process queries entirely in the network data plane, eliminating the query processing at servers and cutting the end-to-end latency to as little as half of an RTT.
Current data center Resource Managers (RMs) (e.g., Apache YARN [7], Google Omega [6]) are not completely network-unaware: for instance, some of them are capable of satisfying affinity rules. CloudMirror [3] even provides bandwidth guarantees to tenant applicaitons. Still, current RMs do not consider INP resources.

† Distributed Systems Programming Group, Technische Universität Darmstadt

‡ Computer Networks Group, Politecnico di Torino

As a consequence, tenant applications cannot request INP services while asking for server resources.

3 Modeling INP resources

This Master's thesis goal consists in investigating how to model INP resources and how to integrate them into RMs.
In order to offer INP services to a tenant application, the latter should be capable of asking for INP resources through an API. To do that, INP resources must be modeled not only to support currently existing INP solutions such as [5] [2] [4] [1], but also to support future ones. It might be convenient to derive a single model to describe both server and INP resources.
Classic tenant application requests can often be modeled as a key-value data structure. CloudMirror [3] requires a *Tenant Application Graph* (TAG) as an input, which is a directed graph where each vertex represents an application component and links' weights represent the minimum requested bandwidth. One possible model could be based on a TAG, describing network resources and/or INP services as vertexes or links. Tenants applications could either use the same model used within the data center or a simplified one, adding another level of abstraction.
Finally, a network-aware placement algorithm in the Resource Manager should then be able to allocate the requested resources accordingly.

References

[1] R. L. Graham, D. Bureddy, P. Lui, H. Rosenstock, G. Shainer, G. Bloch, D. Goldenerg, M. Dubman, S. Kotchubievsky, V. Koushnir, et al. Scalable hierarchical aggregation protocol (sharp): a hardware architecture for efficient data reduction. In *Proceed-*

ings of the First Workshop on Optimization of Communication in HPC, pages 1–10. IEEE Press, 2016.

[2] X. Jin, X. Li, H. Zhang, N. Foster, J. Lee, R. Soulé, C. Kim, and I. Stoica. Netchain: Scale-free sub-rtt coordination. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 35–49, Renton, WA, 2018. USENIX Association.

[3] J. Lee, Y. Turner, M. Lee, L. Popa, S. Banerjee, J.-M. Kang, and P. Sharma. Application-driven bandwidth guarantees in datacenters. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 467–478, New York, NY, USA, 2014. ACM.

[4] M. Liu, L. Luo, J. Nelson, L. Ceze, A. Krishnamurthy, and K. Atreya. Incbricks: Toward in-network computation with an in-network cache. In *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '17, pages 795–809, New York, NY, USA, 2017. ACM.

[5] A. Sapio, I. Abdelaziz, A. Aldilaijan, M. Canini, and P. Kalnis. In-network computation is a dumb idea whose time has come. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, HotNets-XVI, pages 150–156, New York, NY, USA, 2017. ACM.

[6] M. Schwarzkopf, A. Konwinski, M. Abd-El-Malek, and J. Wilkes. Omega: flexible, scalable schedulers for large compute clusters. In *SIGOPS European Conference on Computer Systems (EuroSys)*, pages 351–364, Prague, Czech Republic, 2013.

[7] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, B. Saha, C. Curino, O. O'Malley, S. Radia, B. Reed, and E. Baldeschwieler. Apache hadoop yarn: Yet another resource negotiator. In *Proceedings of the 4th Annual Symposium on Cloud Computing*, SOCC '13, pages 5:1–5:16, New York, NY, USA, 2013. ACM.