Thesis summary

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

1 Info

Title: Data center resource management for in-network processing
Graduand: Marco Micera
Supervisors: Prof. Fulvio Risso †, Prof. Patrick Eugster ‡, M.Sc. Marcel Blöcher ‡
Research areas: cloud computing, distributed systems, in-network processing

2 Thesis purpose

Nowadays there exist several In-Network Processing (INP) solutions that allow tenants to improve their application performance in terms of different metrics: DAIET [5] inventors claim to achieve an 86.9%-89.3% traffic reduction by performing data aggregation entirely in the network data plane. Other solutions like NetChain [2] and IncBricks [4] let programmable switches store data and process queries to cut end-to-end latency; CloudMirror [3] allows client applications to specify bandwidth and high availability guarantees.

For the time being, it seems that there is still no valid resource allocation algorithm that takes into account the presence of a network having a data plane that supports (partially or completely) INP. This thesis has mainly two goals: (i) model and evaluate an Application Programming Interface (API) through which applications can ask for INP resources and (ii) discuss the importance of a scheduler which can reject INP requests and propose their server-only equivalent when needed (e.g., high switch utilization).

2.1 Modeling INP resources

One of the two goals of this Master's thesis 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 able to ask 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 future ones. It may also 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 INP services as vertexes or links. Tenant applications could either use the same model used within the data center or a simplified one, adding another level of abstraction.

2.2 INP-aware Resource Managers

In order for everything to work, a network-aware placement algorithm in the Resource Manager should be able to consider INP and server resources conjunctly: this brings new challenges in the field of resource management as there are currently no Resource Managers (RMs) doing this. One problem that could arise is due to the fact that INP resources are typically very

† Computer Networks Group, Politecnico di Torino, Italy
‡ Distributed Systems Programming Group, Technische Universität Darmstadt, Germany

limited in a data center: this thesis argues the importance of an RM which is flexible enough to propose alternatives based on the current utilization of INP and server resources, since one kind of physical resource type can become the bottleneck for the other.

3 Personal contribution and obtained results

3.1 System design

3.1.1 Composites translation methods

3.2 Resource model

3.3 Generic groups

3.4 Simulation

3.4.1 Results

3.5 Conclusions

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 *Proceedings 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.