Topology Augmentation Meets Machine Learning

Extended Abstract

Saim Salman
Brown University
saim_salman@brown.edu

Theophilus Benson
Brown University
theophilus_benson@brown.edu

1 ABSTRACT

For any augmented architecture design [3–7, 9, 11, 13, 15], one has to decide the optimal placement of the flexible links at each time-step i.e., the optimal schedule. Customarily: (a) This problem is formalized as an optimization problem. (b) Due to the impracticality of solving this optimization at realistic data-center scale, i.e., 100 switches or above, greedy heuristics are employed to create an approximate solution. (c) Each heuristic is intricately tied to the application patterns and specific characteristics of the hardware; therefore does not generalize across novel patterns and provides sub-optimal performance. Thus as a community, we are forced to revisit and redesign these heuristics when the application pattern or network details changes – even a minor change. For example, while c-through [15] and FireFly [7] solve broadly identical problems, they leverage different heuristics to account for low-level differences.

This tight coupling of the "scheduling" heuristics with the hardware and the application patterns causes the heuristic to be dependent on both of these – with any new innovation, each new design develops an associated heuristic. In this paper, we argue for replacing the domain-specific rule-based heuristics with a learning-based approach. Learning based approaches have an advantage over other approaches that they are able to learn an optimal schedule while adapting to changes in the application patterns, network dynamics, and low-level network details. This enables the hardware and applications to evolve independently as the tight coupling with the heuristics is broken.

In this paper, we propose a framework, called DeepConfig , that simplifies the process of designing heuristics for a broad range of topology augmentation designs by learning the heuristic itself. Unlike a traditional OCS development (Figure 1), where the algorithms are bundled with the hardware, in our approach (Figure 1), the hardware and algorithms are independent and, thus, allowed to evolve independently.

The key insight underlying DeepConfig is that intermediate features generated from the parallel convolutional layers using network data, e.g., traffic matrix, allows us to generate an intermediate representation of the network's state, which is able to capture patterns specific to the hardware and the workload. Moreover, while labeled production data crucial for deep learning is unavailable, empirical studies [8] show that modern data center traffic is highly predictable in the case of long flows and thus amenable to offline learning with network simulators and

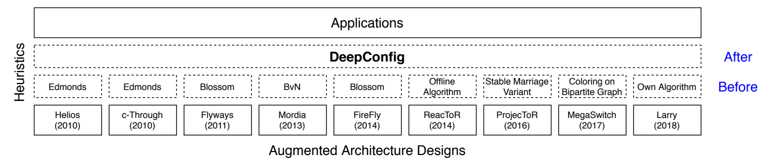

Figure 1: Replacing Heuristics with DeepConfig

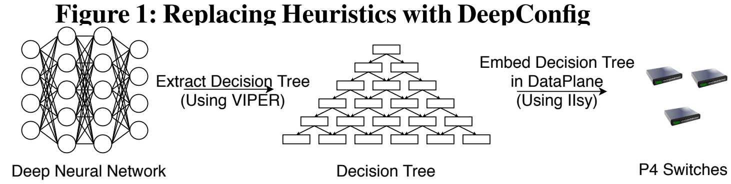

Figure 2: DeepConfig Pipeline

historical traces. DeepConfig will build on this insight by using reinforcement learning (RL) that learns through experience and makes no assumptions on how the network works. Rather, the models are trained through the use of a reward signal which "guides" them towards an optimal solution and thus do not require real world supervised data, and, instead, they can be trained using simulators. We've already published this learning based approach in a workshop paper: [14].

While DeepConfig provides certain benefits over hand-curated heuristics for augmented architectures it comes with its own problems that hinder its adoptability by the general networking community. (i) The underlying Deep Neural Network (DNN) is a blackbox and hence network operators would be hesitant in leveraging a technique that would be hard to debug in production and (ii) due to the model inference times (which is in the order of milliseconds) it would be hard to operate DeepConfig for hardware with reconfiguration time on the order of nano-seconds which is becoming more common.

To solve these issues, we take advantage of recent works: (i) Meng et al. [12] leveraged VIPER [1] to be able to extract the policy from multiple Deep-RL systems including Pensieve [10] showing that there was only a 2% performance degradation from the original DNN and (ii) Xiong et el. [16] showed how one can embed a decision tree in a P4 [2] switch. We propose, that DeepConfig will take advantage of both these recent works, and extract the policy learned during training in the shape of a decision tree thereby minimizing size and inference time while making the policy much more interpretable and embedding this extracted decision tree using P4 switches to be able to accommodate hardware with reconfiguration time on the order of microseconds. We present an overview of this pipeline in Figure-2.

REFERENCES

[1] Osbert Bastani, Yewen Pu, and Armando Solar-Lezama. 2018. Verifiable reinforcement learning via policy extraction. In *Advances in neural information processing systems*. 2494–2504.

[2] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, et al. 2014. P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 87–95.

[3] Andromachi Chatzieleftheriou, Sergey Legtchenko, Hugh Williams, and Antony Rowstron. [n.d.]. Larry: Practical Network Reconfigurability in the Data Center.

[4] Li Chen, Kai Chen, Zhonghua Zhu, Minlan Yu, George Porter, Chunming Qiao, and Shan Zhong. [n.d.]. Enabling Wide-Spread Communications on Optical Fabric with MegaSwitch.

[5] Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. [n.d.]. Helios: A Hybrid Electrical/Optical Switch Architecture for Modular Data Centers. In *Proceedings of ACM SIGCOMM 2010*.

[6] Daniel Halperin, Srikanth Kandula, Jitendra Padhye, Paramvir Bahl, and David Wetherall. [n.d.]. Augmenting Data Center Networks with Multi-gigabit Wireless Links. In *Proceedings of ACM SIGCOMM 2011*.

[7] Navid Hamedazimi, Zafar Qazi, Himanshu Gupta, Vyas Sekar, Samir R. Das, Jon P. Longtin, Himanshu Shah, and Ashish Tanwer. [n.d.]. FireFly: A Reconfigurable Wireless Data Center Fabric Using Free-space Optics. In *Proceedings of ACM SIGCOMM 2014*.

[8] Sangeetha Abdu Jyothi, Carlo Curino, Ishai Menache, Shravan Matthur Narayanamurthy, Alexey Tumanov, Jonathan Yaniv, Ruslan Mavlyutov, Inigo Goiri, Subru Krishnan, Janardhan Kulkarni, and Sriram Rao. 2016. Morpheus: Towards Automated SLOs for Enterprise Clusters. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. USENIX Association, GA, 117–134. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/jyothi

[9] He Liu, Feng Lu, Alex Forencich, Rishi Kapoor, Malveeka Tewari, Geoffrey M Voelker, George Papen, Alex C Snoeren, and George Porter. 2014. Circuit Switching Under the Radar with REACToR.

[10] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural Adaptive Video Streaming with Pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication - SIGCOMM '17*. https://doi.org/10.1145/3098822.3098843

[11] William M Mellette, Rob McGuinness, Arjun Roy, Alex Forencich, George Papen, Alex C Snoeren, and George Porter. 2017. RotorNet: A Scalable, Low-complexity, Optical Datacenter Network. (2017).

[12] Zili Meng, Minhu Wang, Mingwei Xu, Hongzi Mao, Jiasong Bai, and Hongxin Hu. 2019. Explaining Deep Learning-Based Networked Systems. *arXiv preprint arXiv:1910.03835* (2019).

[13] George Porter, Richard Strong, Nathan Farrington, Alex Forencich, Pang Chen-Sun, Tajana Rosing, Yeshaiahu Fainman, George Papen, and Amin Vahdat. 2013. *Integrating microsecond circuit switching into the data center*. Vol. 43. ACM.

[14] Saim Salman, Christopher Streiffer, Huan Chen, Theophilus Benson, and Asim Kadav. 2018. DeepConf: Automating data center network topologies management with machine learning. In *Proceedings of the 2018 Workshop on Network Meets AI & ML*. 8–14.

[15] Guohui Wang, David G Andersen, Michael Kaminsky, Konstantina Papagiannaki, TS Ng, Michael Kozuch, and Michael Ryan. [n.d.]. c-Through: Part-time optics in data centers. In *Proceedings of ACM SIGCOMM 2010*.

[16] Zhaoqi Xiong and Noa Zilberman. 2019. Do Switches Dream of Machine Learning? Toward In-Network Classification. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*. 25–33.