Machine Learning Based Adaptive Flow Classification for Optically Interconnected Data Centers

Nicolaas Viljoen[1], Houman Rastegarfar[2], Mingwei Yang[2], John Wissinger[2], and Madeleine Glick[2]
[1] *Netronome Systems, 2903 Bunker Lane, Santa Clara, CA 95054, USA*
[2] *College of Optical Sciences, University of Arizona, Tucson, AZ 85721, USA*
nick.viljoen@netronome.com, {houman, jwissinger, mglick}@optics.arizona.edu, mingweiyang@email.arizona.edu

ABSTRACT
We optimize flow placement for a hybrid network implementing an adaptive neural network classifier. We predict elephant flows with high accuracy on anonymized university network traffic. We also demonstrate the capability to perform highly complex actions at 40 Gbps using less than 5% of co-processor capacity. This shows that it is possible to implement intelligent actions such as a neural network in a data center using fully programmable NICs without handicapping the server CPU.
Keywords: networks, circuit-switched, optical interconnects.

1. INTRODUCTION

Traffic flows from different applications have differing requirements (e.g., throughput, latency). For best network performance the flows should be directed and scheduled to satisfy the demands while optimizing performance. Studies have shown that proper flow classification yields significant improvement in resource utilization [1], [2], [4]. In an all electrical data center network proper flow placement involves attempting to distribute elephants uniformly across links while in a hybrid electrical/optical data center higher performance is achieved when long lived, high bandwidth elephants are assigned to optical links and delay intolerant or control flows are placed on electrical links (Fig. 1).With an efficient classification algorithm it is feasible to allocate resources to flows according to their requirements and avoid overprovisioning (e.g., allocating an optical circuit to a short mouse flow) and underprovisioning (e.g., mapping a bulk data transfer to a resource-constrained electrical switch) problems [1]. Some data center architectures use application level load balancing by creating highly homogenized traffic flows between racks and pods [5]. This can give a significant advantage; however, this type of system may not adapt to unpredictable future loads.

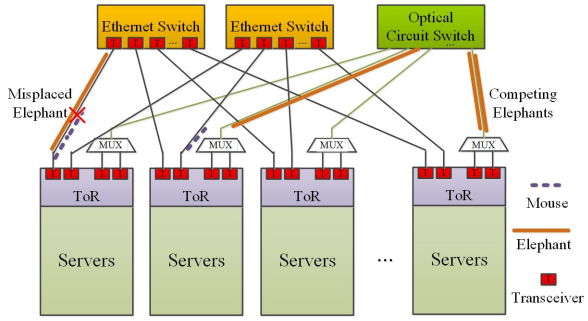

Figure 1. Hybrid electrical/optical data center: Misplaced elephants reduce performance.

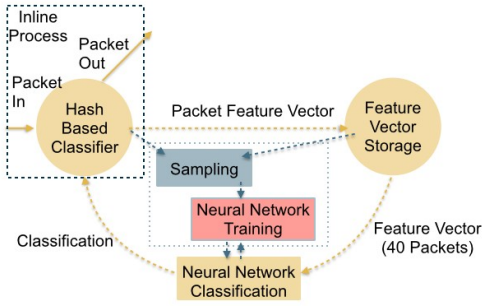

Figure 2. High level control algorithm: The hash based classifier is the only section that needs to operate at line rate.

We exploit the advantages of machine learning combined with recent advances in fully programmable network interface cards (NICs) to optimize scheduling for elephant and mouse flows in a data center network. Machine learning enables predictive and adaptive classification, allowing faster classification of elephants in unpredictable environments such as multi-tenant data centers. In this work we use a neural network based classifier to detect elephants with higher levels of accuracy than leading heuristics, while maintaining the flexibility to adjust to the network environment due to the adaptive and continuous learning of the system. Our classification is done at the edge in order to reduce the burden to the controller. This requires a virtual probe at the compute node for feature detection. Using the fully programmable NIC for this purpose as outlined previously allows this computational model to be scalable at high data rates [6]. There are a wide variety of definitions of elephant flows. To avoid classifying short lived bursty flows as elephants, our analysis considers flows larger than 100 MB as elephants [7]. We also demonstrate that it is possible to operate a hash based classifier on the fully programmable network processor at a data rate of 40 Gbps, which is crucial for the operation of our overall algorithm (Fig. 2).

2. MACHINE LEARNING-BASED FLOW CLASSIFICATION

Machine learning is a form of computational intelligence that provides computers with the ability to learn and adapt without being explicitly programmed. Neural networks have existed as a form of machine learning since the late 1950s. However, neural networks have become truly useful for perceptual problems over the past 10 years or so. The change is due to algorithmic breakthroughs and the increase in applicable computational power, such as single instruction multiple threaded GPUs or multiple instruction multiple threaded NPUs (Network Processing Units). The challenge in networking has been related to feature extraction in real time, allowing the predictive power of neural networks to be harnessed in a time frame which is beneficial. With recent advances in fully programmable NPU NICs, this is now possible using a many core, 1000+ threaded approach.

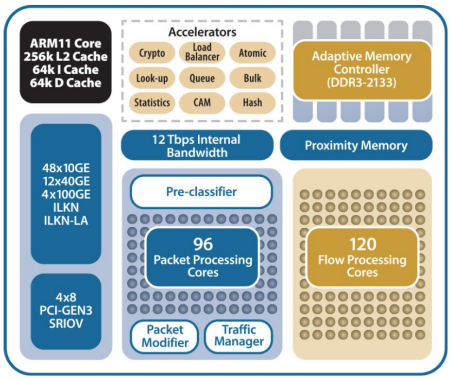

Figure 3. Architecture of NFP 6xxx on NIC: 120 flow processing cores allow multiple instructions to be implemented on different data simultaneously.

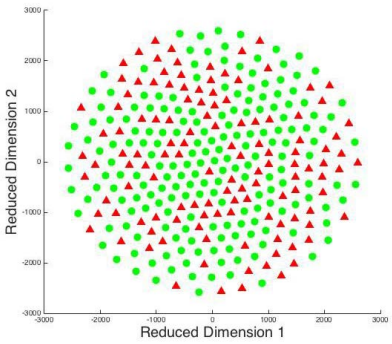

Figure 4. tSNE Diagram: There is a level statistical variability in the dimensionally reduced training set, indicating that there are patterns in our data. Red triangles are elephants, green circles are mice.

As shown in Fig. 2 the algorithm consists of 3 key parts: 1) The **Hash Based Classifier** which checks whether the packets belong to a classified flow. This is required to run at high speed with low latency. 2) The **Feature Vector Storage** which stores the flow features for packets from as yet unclassified flows. Feature vectors are required to have enough information to allow the machine learning algorithm to make a flow classification decision. 3) The **Neural Network** which classifies complete feature vectors. The feature vector is based upon previous work [8]. It includes the five-tuple (source IP address, destination IP address, source port, destination port, transport protocol), packet sizes and a set of intraflow timings within the first 40 packets of a flow. The data was normalized by approximate non-linear whitening and also attempting to ensure internal covariate shift was reduced. This was to attempt to improve training speed and avoid the disappearing gradient problem when using gradient descent backpropagation [9, 10]. The tSNE dimensional reduction (Fig. 4), which probabilistically maps high dimensional distributions to lower dimensional ones using Kullback Liebler divergence, shows a noticeable level of statistical variability between the classes [11].

The type of neural network used is a fully connected multi-layer perceptron (MLP) with 4 hidden layers. MLPs are relatively easy to implement in high dimensional situations without base knowledge of intermediate features. MLPs have high levels of true negative classification which is important to ensure mice do not flood the optical connection [12]. Due to the nature of mouse and elephant distribution, which has an overwhelming amount of mice, there is a class imbalance problem. This is overcome by training with a non-proportional amount of mice and elephant flows. To ensure adaptability we use an internal self-supervised teaching mechanism to update the classifier model. The next step in this work is developing a model based on deep belief networks, this has shown significant promise in initial testing.

3. ADAPTIVE FLOW CLASSIFICATION

To assess the MLP based algorithm we compare its effectiveness to a common bandwidth based classification algorithm which is checking for any flows taking more than approximately 10% of the bandwidth in a second and classifying these as elephants [14]. The data used was collected within an anonymized university network. Its overall characteristics closely matched previously described data center traffic with 4% of traffic as elephants which contain 94% of the data [13]. The data was collected at 1 Gbps line rate over 24 hrs with 20 minutes sampled per hour. It encapsulated a variety of traffic patterns and distributions. Different types of traffic were dominant at different times of day. The captured traffic was replayed into the classifiers and was classified as mouse or elephant traffic. Classification is performed using information within the headers of the first 40 packets of the flow. Results of the classification sys- tem are shown as confusion matrices in Fig. 5. We compare effectiveness at flow classification and byte classification, defined as the amount of bytes sent on the correct

path. These results show that the MLP based system performs 22% better in terms of predictively detecting elephants and even more so in terms of byte allocation, where it has a 26% advantage. This gives a significant advantage in terms of being able to load balance a hybrid electrical/optical network. The second aspect to note is the consistency of the performance of the MLP. A problem for the heuristics is that they are not able to adjust to changing situations. The performance of the MLP is relatively consistent as shown by Fig. 6 with a variance of 139 in correct true positive percentage compared to 280 for the heuristic. The MLP system is designed to be a continuously learning and incrementally updatable classifier that can respond to changes in traffic patterns that occur over just a few seconds. In our experiment we show model updates every hour because of the way the data was collected. We note that there is never more than a one hour period of traffic performance 5% under the mean (Fig. 6). For example between 1600 and 1700 hrs, there is a reduction in MLP accuracy. However, by the next hour, the system has retrained itself. This shows the flexibility of the MLP to adjust to the conditions in the network.

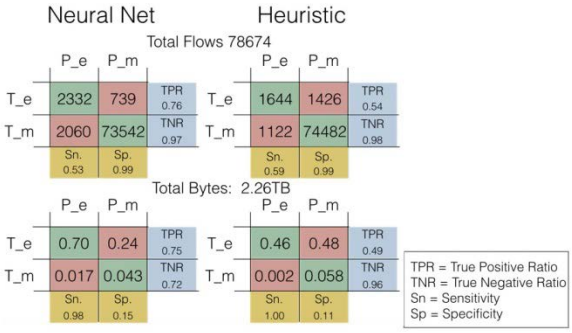

Figure 5. Confusion Matrix: The MLP based classifier performs better at elephant classification-TPR, while maintaining high TNR-mice classification. This allows the MLP based classifier to direct over 50% more bytes correctly.

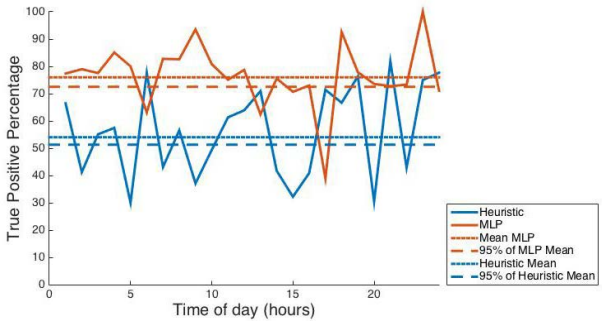

Figure 6. Adaptability: The MLP based classifier shows less variance and consistently adjusts itself when traffic profile changes.

To show that we are able to run this system without significantly affecting the server CPU utilization, we implement a hash based classifier at a throughput of 40 Gbps with packets that are smaller than that of the mean packet size in a data center [13]. To test this we have connected up two fully programmable NICs in a bidirectional configuration. On ingress, packets were classified based on protocol header contents in a hash table and an action assigned, e.g. add tunneling headers for matched flows. The reciprocal operation was enacted on the egress. An IXIA network traffic generator was used successfully to send 256 byte packets bidirectionally at 40Gbps. This highlights the use of hash based classification with associated complex actions at line speed. To implement this less than 5% of the processor capacity on the fully programmable NIC was required.

4. CONCLUSION

In conclusion we propose using machine learning combined with virtual probes based on intelligent NICs at the edge for improved elephant flow identification and adaptability to changing application requirements. Our initial results show improved classification and flow direction. We also demonstrate the capability to perform highly complex actions at 40 Gbps using less than 5% of co-processor capacity. This shows that it is possible to implement intelligent actions such as a neural network in a data center using fully programmable NICs without handicapping the server CPU.

ACKNOWLEDGEMENT

This work was supported by the NSF Center for Integrated Access Networks (CIAN) under grant # EEC-0812072. We would also like to thank Karl Newell and Jason Sullivan of the UITS for technical support and hardware provision and Gavin Stark, Stuart Wray, Pablo Cascon, Rolf Neugebauer and Dinan Gunawardena of Netronome Systems.

REFERENCES

[1] H. Rastegarfar *et al.*, "TCP flow classification and bandwidth aggregation in optically-interconnected data center networks," submitted to *IEEE J. Opt. Commun. Netw., 2016.*
[2] M. Al-Fares *et al.*, in *Proc. NSDI'10*, Apr. 2010, paper 19.
[3] A. R. Curtis *et al.*, in *Proc. IEEE INFOCOM 2011*, Apr. 2011, pp. 1629-1637.
[4] N. Farrington *et al.*, *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 339-350, Oct. 2010.
[5] A. Roy *et al.*, in *Proc. ACM Conference on Special Interest Group on Data Communication*, 2015.

[6] N. Viljoen, T. Tofigh, and B. Sullivan, in *Proc. ONS 2016 The Need for Complex Analytics from Forwarding Pipelines*, 2016.
[7] A. Greenberg *et al*., *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 51-62, Oct. 2009.
[8] G. J. Stark, N. J. Viljoen, and N. Viljoen, U.S. Patent Application 13/675,620.
[9] X. Glorot and Y. Bengio, in *Proc. International Conference on Artificial Intelligence and Statistics*, 2010.
[10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint*:1502.03167 (2015).
[11] L. van der Maaten, *The Journal of Machine Learning Research* 9.2579-2605 (2008): 85.
[12] J. Wang *et al*., *Pattern Recognition* 45.3 (2012): 1136-1145.
[13] T. Benson *et al*., in *Proc. 10th ACM SIGCOMM Conference on Internet Measurement*, 2010.
[14] S. Shirali-Shahreza and Y. Ganjali, in *Proc. IEEE ICC Workshops*, Jun. 2013, pp. 1335-1339.