NSS News

Recent Papers and Posters

• Our poster on Cutting Packet Fat in Shallow VNF Chain Processing by Swati Goswami, Nodir Kodirov, Ivan Beschastnikh to appear at NSDI'19 [1]


• Data parallel training is widely used for scaling distributed deep neural network (DNN) training. However, the performance benefits are often limited by the communication-heavy parameter synchronization step. In this paper, we take advantage of the domain specific knowledge of DNN training and overlap parameter synchronization with computation in order to improve the training performance. We make two key observations: (1) different parameters can afford different synchronization delays and (2) the optimal data representation granularity for the communication may differ from that used by the underlying DNN model implementation. Based on these observations we propose a new mechanism called Priority-based Parameter Propagation (P3), which, synchronizes parameters at a finer granularity and schedules data transmission in such a way that the training process incurs minimal communication delay. We show that: P3 can improve the training throughput of ResNet-50, Sockeye and VGG-19 by as much as 25%, 38% and 66% respectively.


• The recent popularity of deep neural networks (DNNs) has generated considerable research interest in performing DNN-related computation efficiently. However, the primary focus is usually very narrow and limited to (i) inference - i.e. how to efficiently execute already trained models and (ii) image classification networks as the primary benchmark for evaluation. Our primary goal in this work is to break this myopic view by (i) proposing a new benchmark suite for DNN training, called TBD 1, which comprises a representative set of eight DNN models and covers six major machine learning applications: image classification, machine translation, speech recognition, object detection, adversarial networks, reinforcement learning, and (ii) performing an extensive performance analysis of these models on three major deep learning frameworks (TensorFlow, MXNet, CNTK) across different hardware configurations (single-GPU, multi-GPU, and multi-machine). We present a new toolchain for performance analysis for these models that combines the targeted usage of existing performance analysis tools, careful selection of performance metrics, and methodologies to analyze the results. We also build a new set of tools for memory profiling in three major frameworks. These tools can shed light on precisely how much memory is consumed by different data structures (weights, activations, gradients, workspace) in DNN training. Using our tools and methodologies, we make several important observations and recommendations on where future DNN training research and optimization should be focused.


• Building correct implementations of distributed systems continues to elude us. Solutions consist of abstract modeling languages such as TLA+, PLusCal, which specify models of systems and tools like Coq, and SPIN which verify correctness of models but require considerable amount of effort, or transparent model checkers like MODIST, CMC and CHESS which suffer from state space explosion, rendering them impractical to use as they are too slow. We propose Dara, a novel hybrid technique that combines the speed of abstract model checkers with the correctness and ease-of-use of transparent model checkers. Dara utilizes tests as well as a transparent model checker to generate logs from real executions of the system. The generated logs are analyzed
to infer a model of the system which is model-checked by SPIN to verify user-provided invariants. Invariant violations are reported as likely bug traces. These traces are then passed to a replay engine which tries to replay the traces as real executions of the system to remove false positives. We are currently evaluating Dara's efficiency and usability.


  Recent networking research has identified that data-driven congestion control (CC) can be more efficient than traditional CC in TCP. Deep reinforcement learning (RL), in particular, has the potential to learn optimal network policies. However, RL suffers from instability and over-fitting, deficiencies which so far render it unacceptable for use in datacenter networks. In this paper, we analyze the requirements for RL to succeed in the datacenter context. We present a new emulator, Iroko, which we developed to support different network topologies, congestion control algorithms, and deployment scenarios. Iroko interfaces with the OpenAI gym toolkit, which allows for fast and fair evaluation of different RL and traditional CC algorithms under the same conditions. We present initial benchmarks on three deep RL algorithms compared to TCP New Vegas and DCTCP. Our results show that these algorithms are able to learn a CC policy which exceeds the performance of TCP New Vegas on a dumbbell and fat-tree topology. We make our emulator open-source and publicly available: https://github.com/dcgym/iroko


  P4 is a domain-specific language for implementing network data-planes. The P4 abstraction allows programmers to write network protocols in a generalized fashion, without needing to know the configuration specifics of the targeted data-plane. The extended Berkeley Packet Filter (eBPF) is a safe virtual machine for executing sand-boxed programs in the Linux kernel. eBPF, and its extension the eXpress Data Path (XDP), effectively serve as programmable data-planes of the kernel. P4C-XDP is a project combining the performance of XDP with the generality and usability of P4. In this document, we describe how P4 can be translated into eBPF/XDP. We review the fundamental limitations of both technologies, analyze the performance of several generated XDP programs, and discuss problems we have faced while working on this new technology

**Recent Talks**


Fabian Ruffy: Linux Network Programming with P4, 2018 Linux Plumbers Conference [6], Video [8]

Gleb Naumenko: Efficient Transaction Relay for Bitcoin, 2018 Scaling Bitcoin (WIP Section) [9], Video [10]

**Internships**

**Summer 2019**

Summer 2018

Fabian Ruffy: VMware Research [12]

Gleb Naumenko: Blockstream [13]

References

[1] https://www.usenix.org/conference/nsdi19
[8] https://www.youtube.com/watch?v=r_P5pKNFNHY
[10] https://www.youtube.com/watch?v=2MxO-oa_1LM&feature=youtu.be&t=11m50s
[12] https://research.vmware.com/
Article Sources and Contributors