Parallelizing CNNs on Xeon Phi coprocessors

Ji Kim (jyk46), Jason Setter (jls548), Ben Shulman (bgs53)

Background

Convolutional Neural Networks (CNNs) are state-of-the-art classifiers for many classification tasks, particularly in computer vision. CNNs are neural networks where levels consist of convolutional layers, non-linearities (typically ReLU), and subsampling. At the end a fully-connected layer (or multiple) map final inputs to classifications.

As with any neural network, the accuracy of CNNs depends heavily on how well the network is *trained*. Training may or may not be supervised, but always involves exposing the network to a large number of labeled instances to determine suitable coefficients for each layer in the CNN. Instances are first forward propagated to calculate probabilities and then the loss is used to back-propagate the gradient and update the CNN.

Previous Work

There is a rich foundation of work on parallelizing CNNs to improve the performance of the training and classification phases of CNNs. Much of this work focuses on the development of parallel libraries for general-purpose CPUs and GPUs to streamline the development of next-generation CNNs such as Theano,[1] Caffe,[2] and Torch[3]. Another focus of the previous work is on optimizations for parallelizing CNNs for GPUs as well as specialized hardware for accelerating machine learning algorithms[4]. However, there has been much less work on parallelizing CNNs for Many Integrated Core (MIC) architectures such as the Intel Xeon Phi. One relevant example is a preliminary analysis of the potential speedups one could achieve using the Intel Xeon Phi by Viebke and Pllana[5]. In addition, there is no work to the authors' knowledge that explores the energy tradeoffs of utilizing GPUs or MICs for parallelizing CNNs.

Objectives

The primary objective of this project is to parallelize and optimize a reasonably complex implementation of a CNN for the Intel Xeon Phi accelerator boards on the Totient cluster. Effectively utilizing these accelerators is expected to provide significant speedups to both the training and the classification phases of the CNN.

[1] http://deeplearning.net/software/theano/

[2] http://caffe.berkeleyvision.org

[3] http://torch.ch

[4] Y. Chen, et al. DaDianNao: A Machine-Learning Supercomputer. MICRO 2014.

[5] A. Viebke, S. Pllana. The Potential of the Intel Xeon Phi for Supervised Deep Learning. HPCC 2015. *arXiv preprint arXiv:1506.09067*.

The secondary objective of this project is to provide first-order insights into the tradeoffs associated with parallelizing CNNs for MICs versus GPUs. Although there has been a wide range of work on parallelizing CNNs for GPUs with success in achieving impressive speedups, there is less consensus on the performance and energy impacts of parallelizing CNNs for MICs. As such, most of the CNN implementations used today are heavily optimized for GPUs. However, one challenge with this approach is *scaling-out* computation for datacenters due to the relatively high power cost of GPUs[6]. It would be interesting to see if optimizing CNNs for MICs would be able to achieve comparable speedups with higher energy-efficiency to address this challenge.

The summary of the project goals is as follows:

- Identify an existing CNN implementation for study
- Verify the serial/parallel CNN implementations on the compute nodes (Intel Xeon E5)
- Parallelize and tune CNN for the accelerator boards (Intel Xeon Phi)
- Develop first-order energy models for accelerator boards
- Evaluate and compare performance and energy between the parallel implementations of the CNN on the compute nodes, the accelerator boards, and GPUs (from literature)

[6] K. Ovtcharov, O. Ruwase, J. Y. Kim, J. Fowers, K. Strauss, E. Chung. Toward Accelerating Deep Learning at Scale Using Specialized Logic. Hot Chips 2015.