2017 IEEE 37th International Conference on Distributed Computing Systems

Evaluation of Deep Learning Frameworks over Different HPC Architectures

Shayan Shams*, Richard Platania*, Kisung Lee, and Seung-Jong Park
Division of Computer Science and Engineering
Center for Computation and Technology
Baton Rouge, LA 70803, USA
Email: {sshams2,rplatania,lee,sjpark}@cct.lsu.edu

***Abstract*—Recent advances in deep learning have enabled researchers across many disciplines to uncover new insights about large datasets. Deep neural networks have shown applicability to image, time-series, textual, and other data, all of which are available in a plethora of research fields. However, their computational complexity and large memory overhead requires advanced software and hardware technologies to train neural networks in a reasonable amount of time. To make this possible, there has been an influx in development of deep learning software that aim to leverage advanced hardware resources. In order to better understand the performance implications of deep learning frameworks over these different resources, we analyze the performance of three different frameworks, Caffe, TensorFlow, and Apache SINGA, over several hardware environments. This includes scaling up and out with single- and multi-node setups using different CPU and GPU technologies. Notably, we investigate the performance characteristics of NVIDIA's state-of-the-art hardware technology, NVLink, and also Intel's Knights Landing, the most advanced Intel product for deep learning, with respect to training time and utilization. To our best knowledge, this is the first work concerning deep learning bench-marking with NVLink and Knights Landing. Through these experiments, we provide analysis of the frameworks' performance over different hardware environments in terms of speed and scaling. As a result of this work, better insight is given towards both using and developing deep learning tools that cater to current and upcoming hardware technologies.**

I. Introduction

Deep learning has continued to thrive as the availability of capable computing resources becomes more common. Many fields are now able to make use of deep learning for their particular applications. These applications include biomedical image analysis, social media analysis, and many more. New deep learning techniques and applications are constantly being developed as they continue to see success through many domains. With this spurring of development, many different software and hardware have been developed that cater towards deep learning workloads. There are a few existing works that focus on the comparison of deep learning hardware performance from speed and scaling perspectives. In order to more effectively compare the variety of software tools, they should be benchmarked with these metrics over various hardware environments. In this work, we aim to do this using three deep learning frameworks, several HPC environments, and state-of-the-art hardware technologies.

The trending development of deep learning tools is providing users with more options that can cater specifically to their needs. Typically, these needs stem from available hardware resources. For instance, a user with access to a large-scale commodity cluster may aim to utilize a different deep learning tool than a user with access to a single multi-GPU machine. This isn't to say that one single tool isn't suitable for multiple environments. Rather, the different tools were created to include features that benefit certain hardware setups. Caffe is a popular deep learning framework that excels at single- or multi-GPU training on a single machine, making it more accessible to the general user [1]. On the other hand, Apache SINGA strives to provide scalable deep learning in a distributed environment [2], [3]. Combining multiple ideologies, TensorFlow looks to perform well on a variety of platforms, including scaled-up CPU or GPU machines, scaled-out clusters, and mobile devices [4]. There are several works that benchmark the performance of these and other frameworks with respect to batch training time [5], [6]. Caffe in particular has been thoroughly evaluated. However, works concerning the evaluation and comparison of TensorFlow, SINGA, or other distributed deep learning tools through scaling are limited. We aim to contribute to the already existing benchmarks with scaling results. The three aforementioned frameworks are used in this work. Many other tools have been developed with certain performance goals in mind, and we plan to explore these in future works [7], [8].

Since many of the recently developed software for deep learning were created with specific computing environments in mind, it is also important to consider the ongoing advancement of related hardware technologies and their impact on software performance. The spotlight tends to remain on the advancement of general purpose GPUs because of their high performance vector computation and deep learning's reliance on such. However, it is still important to visualize CPU and other computing hardware capability with respect to deep learning. It is interesting to see if the recent and upcoming technologies, such as Intel's Knight's Landing Xeon Phi processor with Omni-Path interconnect, can compete with GPU in deep learning. On top of this, GPU memory is limited,

*These authors contributed equally to this work.

1063-6927/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDCS.2017.259

IEEE computer society

and the size of data continues to grow. Hence, CPU or other technologies may see increased use in deep learning. GPU-related technologies are beginning to adapt to overcome this issue and the data transfer bottleneck it imposes on GPU-based training. Previous works have been done to evaluate hardware performance for deep learning, and we aim to add to these with the inclusion of new frameworks and hardware [9], [5], [6]. Notably, our work evaluates the impact of NVIDIA's recently developed NVLink technology for deep learning, which is as an alternative to traditional PCIe Gen3 for CPU to GPU and GPU to GPU data transfer, and also Intel's Knight Landings (KNL), which was very recently introduced to overcome the problem of limited memory on GPUs while having powerful and fast vector and tensor operations. This technology is unique in that it is Intel's first Xeon Phi accelerator that can replace a host processor, eliminating the need for PCIe transfer.

The remainder of the paper is organized as follows. First, we describe the deep learning frameworks and hardware used in this work. Following this, we evaluate the different frameworks with various metrics on differing hardware setups. Before concluding, a brief discussion regarding challenges is given. After conclusion, we talk about future works to which we intend to extend this work.

II. Deep learning frameworks

TABLE I: Deep learning frameworks for evaluation.

FRAMEWORK	VERSION	URL
Caffe	v1.0	http://caffe.berkeleyvision.org/
TensorFlow	v0.12	https://www.tensorflow.org/
Apache SINGA	v0.30	https://singa.incubator.apache.org/

For this work, we selected three frameworks for evaluation: Caffe from Berkeley, TensorFlow from Google, and Apache SINGA [1], [4], [3], [2]. These were selected based on a combination of popularity, performance, and distributed computing capability. Because of the difficulty in compiling the different frameworks over various hardware and developing similar models for benchmarking, we limited ourselves to only these. For example, ensuring correct compilation and execution of each framework over the P100 and Knight's Landing took much effort. Furthermore, producing code for identical deep learning models and training for each framework is difficult. In a future work, we plan to diversify our selection of frameworks.

A. Caffe

Caffe was developed at the Berkeley Vision and Learning Center and is one of the most popular deep learning frameworks currently available. It is made popular in part by its fast benchmark training time and ease of use for programming novices. Because of its popularity, the large community of users and developers have released many different flavors of Caffe, each attempting to add a new feature or improve an existing one. For instance, Intel released their own version with optimization for Xeon processors[1]. Additionally, there have been several releases that add MPI to the framework, consequently enabling scaled-out deep learning. In this work, we make use of the base version of Caffe from Berkeley for GPU comparisons and for evaluating Intel's Knights Landing (KNL) we use the Intel version of Caffe.

B. TensorFlow

TensorFlow was originally developed by the Google Brain Team at Google's Machine Intelligence research organization. It has since then become an open-source project. Deep learning models can be expressed with a data flow graph, where each vertex represents some computation, similar to a neuron in a neural network. Recently, distributed training was added to TensorFlow. This framework was selected for our work because of its fast community growth and distributed training capability. Furthermore, because it is still quite new, there is a lack of validated benchmarks available that measures the performance of TensorFlow over different hardware.

C. Apache SINGA

SINGA is a lesser known deep learning platform designed with distributed training in mind. The lack of popularity is likely due to the fact that it is an Apache project that is still in incubation. However, it offers desirable features in a deep learning tool. The deep learning model definition, which is done by defining layers, is similar to that of Caffe, allowing easy migration of model configurations from one framework to another. More importantly, it enables effortless scaling out and flexible client/server configuration for synchronous or asynchronous training. The authors have published promising results demonstrating the performance of scaling out and both synchronous and asynchronous training [3], [2]. In spite of its lack of popularity, we select SINGA in order to compare the scalability with other frameworks in multiple different distributed HPC environments.

III. HPC architectures

TABLE II: Hardware for deep learning evaluation.

HARDWARE	CORES	MEMORY
Intel E5-2680v2 Xeon Processor	20	64 GB
IBM Power8 Processor	20	256 GB
IBM Power8+ Processor with NVLink interface	20	256 GB
NVIDIA Tesla P100 with NVLink interface	3584	16 GB
Intel Phi 7230 processor(KNL)	64	96 GB

A. Deep learning with CPU

More often than not, training deep learning models with CPU is not ideal. It is well known that GPUs provide an extreme advantage in cases of vector computation, which makes up the majority of computation during training. In practice, the CPU typically acts as the parameter server, performing scalar

[1] Available at https://github.com/intel/caffe

updating of parameters received from GPUs and redistributing the updated values to the GPUs. However, it is still important to evaluate CPU-based training because not all users have access to GPUs with sufficient memory for training larger data. In Table II, the CPUs used in this work are described. The main difference between the Power8 and Power8+ is that the Power8+ supports NVLink, which will be described in the following subsection. We investigate the Intel's Knight's Landing performance on various deep learning frameworks. Intel Knight's Landing is interesting since they it can replace the host CPU, eliminating the need for data transfer over PCIe, while having acceptable number of cores and hyper threads to accommodate the needs of vector operations in deep learning. In addition, it can has access to a larger memory pool by using RAM, enabling training with a larger batch size and model and making faster convergence possible.

B. Deep learning with GPU

Training evaluation using GPU is arguably more important than the CPU evaluation. GPUs outperform CPUs in vector computation, which makes up the majority of computation in deep learning. The GPUs utilized for our experiments are listed in Table II. Of particular interest is the Tesla P100 with NVLink. NVLink is a recently developed technology by NVIDIA that intends to provide faster data sharing between CPU to GPU and GPU to GPU. In the evaluation, we will go into detail on their performance over the P100 and in order to make a clear comparison and find out the effectiveness of the the new NVlink technology. We select and compare NVIDIA Tesla P100 GPUs, one with support of NVLink and the other with PCIe Gen3 connection to CPU.

TABLE III: Deep learning models and datasets for experiments.

MODEL	DATASET	URL
AlexNet [10]	ILSVRC'12	image-net.org/challenges/LSVRC/2012
VGG-19 [11]	ILSVRC'12	
GoogLeNet [12]	ILSVRC'12	
LeNet [13]	MNIST	yann. lecun. com/exdb/mnist
ConvNet	CIFAR-10	www.cs.toronto.edu/ kriz/cifar.html

IV. Evaluation

We divide the evaluation into three parts. First, we investigate the performance of different CPU, GPU, and Knight's Landing models on training time. Second, we analyze training performance on a single node while scaling up the number of CPU and GPU. The frameworks evaluated for this are Caffe (both Berkeley and Intel versions) and TensorFlow. Lastly, we look at scaling out with multiple nodes. For this, we focus on TensorFlow, SINGA, and the Intel version of Caffe, since they were designed with scaling in mind, and the base version of Caffe does not scale out. Several neural network models and datasets are used throughout these experiments and are described in Table III. They are all commonly used in deep learning experiments and benchmarks. All timings were averaged over 1000 iterations and ignore the first 100 iterations which often exhibit some startup overhead. In other words, the timings from iterations 101 to 1100 are used.

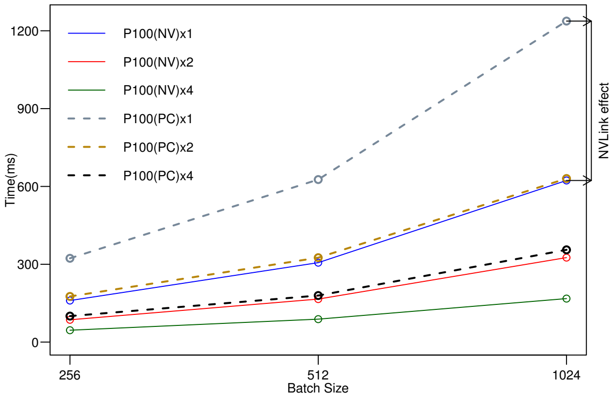

Fig. 1: Scaling up with Caffe using AlexNet for GPU (P100 with and without NVLink) while increasing number of GPUs and batch size.

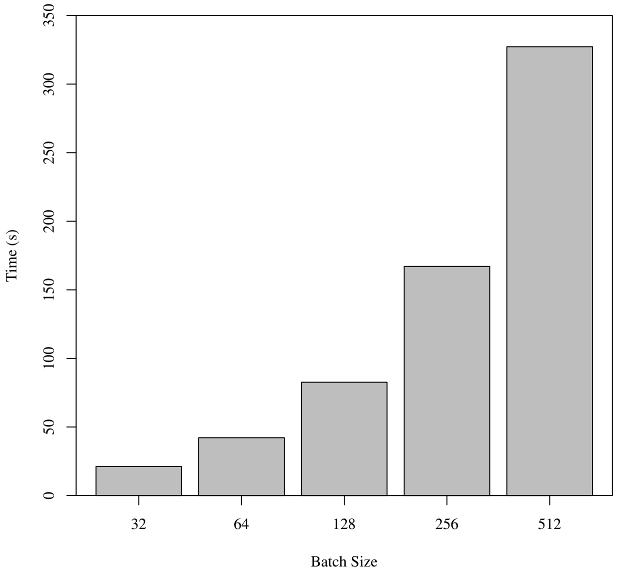

Fig. 2: Scaling up with Caffe using GoogLeNet for CPU while increasing the batch size.

A. CPU and GPU performance

Before considering scaling up or out, it is important to see the performance of a single compute resource. Our first

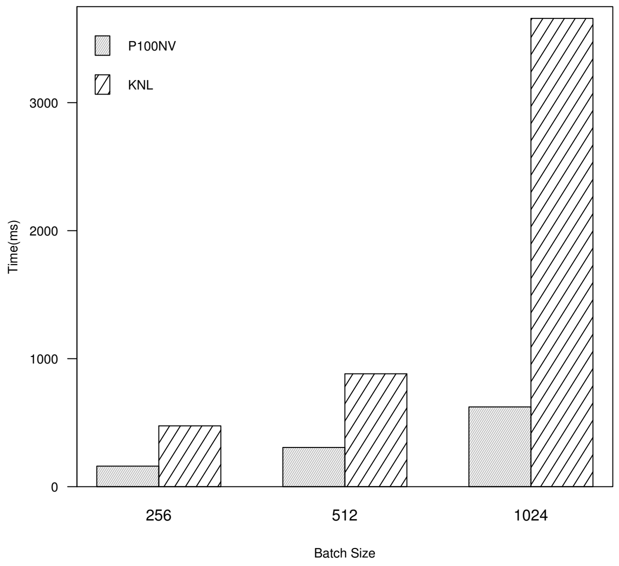

Fig. 3: Scaling up batch size on P100 with NVLink and KNL using Alexnet with Caffe.

TABLE IV: Benchmark for average time to train one batch with Caffe and AlexNet. The ILSVRC'12 dataset was used with a batch size of 256

HARDWARE	AVERAGE TIME (s)
Intel E5-2680v2 Xeon	51.76
IBM Power8 (SMT=OFF)	47.66
IBM Power8 (SMT=4)	79.28
IBM Power8 (SMT=8)	106.56
IBM Power8+ (SMT=OFF)	37.98
IBM Power8+ (SMT=4)	58.83
IBM Power8+ (SMT=8)	60.49
NVIDIA Tesla K20X	1.61
NVIDIA Tesla P100 (NVLink)	0.15
NVIDIA Tesla P100 (PCIe)	0.32
Intel's Knights Landing	0.88

experimental consideration is the affect of CPU and GPU model on the time taken to train a batch of images. In Table IV, we provide the training time per batch with various CPU and GPU configurations. These results were gathered using Caffe and the AlexNet model with the ImageNet Large Scale Visual Recognition Comptetion 2012 (ILSVRC'12) dataset. Each scenario consists of only one worker (whether it be CPU or GPU) and has a batch size of 256. As expected, the two configurations using GPU give the best performance, and the P100 GPU with NVLink vastly outperforms CPU and KNL. Later in this section, we will investigate this performance more closely in order to visualize the effect of NVLink on training time. There is an interesting trend with the CPUs; there is performance degradation with any configuration where simultaneous multi-threading (SMT) is on. In other words, the number of threads exceeding the number of cores hurts performance. As it is shown and expected in IV, GPU and Intel's Knights landing outperform CPU remarkably, so the paper will concentrate more on accelerator hardware than benchmarking CPU.

Previously, Table IV demonstrated the added benefit of the P100 and NVLink over the PCIe. To analyze the data transfer speedup provided by NVLink more closely, we provide Table V, which describes the data transfer rate of NVLink against PCIe Gen3. The DeepBench toolkit[2] was used to gather these results. We used the All-Reduce technique, which relies on keeping the parameters on all instances of the model across 4 GPUs synchronized by making sure all instances of the model have the same copy of the gradients before taking an optimization step. We measured the time spent to synchronized parameters for four P100 GPUs, with and without NVlink, in one node. As it is shown in Table V, except for the smallest case of 100000 floats, NVLink outperforms PCIe by an approximate factor of two and the difference is much more obvious when the amount of data is increased. As a result, it is obvious that NVlink contributes to faster training and shorter time to completion very effectively.

TABLE V: Data transfer speeds using P100 with and without NVLink.

NUM.FLOATS	BYTES	NVLINK	PCIE
100000	400000	0.238(msec)	0.186(msec)
3097600	12390400	1.427(msec)	2.861(msec)
4194304	16777216	1.914(msec)	3.858(msec)
6553600	26214400	3.088(msec)	6.029(msec)
16777217	67108868	7.519(msec)	16.358(msec)

B. Scaling up a single node

After laying the groundwork for different CPU and GPU performance, we will see how well different framework and hardware combinations scale up. In practice, many users utilize a single machine with either multiple CPU cores or GPUs as workers. Here, we we will see how different frameworks, Caffe and TensorFlow in particular, scale up with respect to CPU and GPU. Starting with Figure 1, we have the training time for Caffe with Alexnet on P100 GPUs while scaling up the batch size.In the figure, P100(NV) and P100(PC) represent P100 with NVlink and P100 with PCIe connection respectively and the number after (x) shows number of GPU used in the benchmark. We scale up from one to 4 GPUs across one node with batch sizes from 256 to 1024 images. As it is expected, increasing the batch size increases the computation time linearly and increasing the number of GPUs decreases the training time linearly. Figure 1 illustrates Caffe shows very good scalaliblity. Another highlighted feature the Figure 1 is the NVLink speedup. The marked gap shows the difference in training time for P100 with NVlink and P100 with PCIe for

[2] Available at https://github.com/baidu-research/DeepBench

batch size of 1024 images. The mentioned difference is the speed up solely caused by having faster communication link between CPU and GPUs.

Briefly, we give the scaling performance with respect to CPU. Figure 2 has the training time for Caffe with Googlenet with scaled batch size from 32 to 512 images. While the CPU performance is significantly behind GPU in terms of speed, CPU still exhibits good scaling.

As mentioned before, Intel has released Intel's Knights Landing to overcome the limited GPU memory problem which limits the batch size while having fast vector and matrix operations by providing 64-72 number of cores, depending on the model, and 4 threads per core. In addition, KNL is Intel's first Xeon Phi accelerator that can replace a host processor eliminating the need for PCIe transfer which in turn contribute to better performance.In Figure 3, we compare time for one training iteration for P100 with NVlink and KNL by scaling up the batch size in Alexnet with Caffe. Figure 3 shows although KNL is showing acceptable speed, it is outperformed by P100 by almost factor of 3 regarding the performance.

To compare Caffe and TensorFlow more closely, we evaluate their training times with two larger, popular networks, VGG-19 and GoogLeNet (Inception V1). Figure 4 gives the time per training iteration using a batch size of 128 with the GoogLeNet (Inception V1) model. Both frameworks exhibit good scaling, but Caffe performs each iteration almost two times faster than TensorFlow. In moving to a larger network, we have VGG-19 results in Figure 5. The frameworks' performance is much closer in this case, however TensorFlow began to exhibit some scaling issues with four nodes.

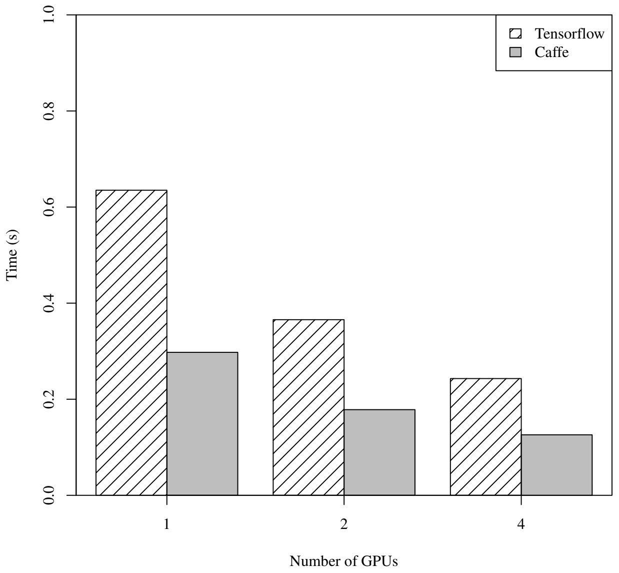

Fig. 4: Scaling up batch size on P100 with NVLink using GoogLeNet with TensorFlow and Caffe.

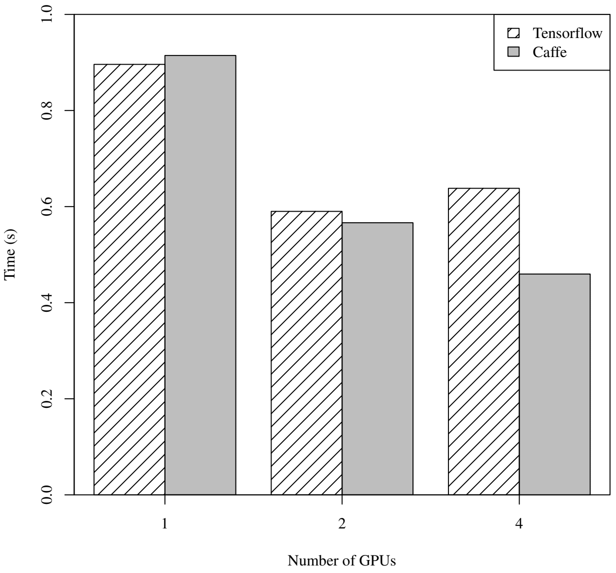

Fig. 5: Scaling up batch size on P100 with NVLink using VGG-19 Net with TensorFlow and Caffe.

C. Scaling out with multiple nodes

In this section, we will provide some initial analysis of scaled-out deep learning with TensorFlow, SINGA and Intel Caffe over Intel's Knight Landings using Omni-Path interconnect and P100 using InfiniBand. First, we observe TensorFlow, Caffe, and SINGA results for scaling out GPUs with the LeNet architecture and MNIST dataset in Figure 6. This shows the number of images trained per seconds as more nodes are added. Each node has one P100 GPU worker with a mini-batch size of 64 images. The network interconnect used is 56 Gbps InfiniBand. Performance is depicted in terms of number of images trained per millisecond. At this point, distributed training is a very experimental feature in TensorFlow. Regardless, it provides good scaling as more nodes are added. While SINGA certainly benefits from scaling out, it lags behind the other two frameworks severely in terms of training throughput. The Intel version of Caffe also shows good scalability while providing the best throughput in terms of number of images trained per millisecond.

Since Intel Caffe was designed with particular effort towards utilizing Intel processors and coprocessors (i.e., Knight's Landing), we investigate its performance with respect to scaling out over nodes with Knight's Landing (KNL) processors. These results include TensorFlow's performance in order to have a point of comparison. SINGA had compilation issues, so it is not included in this result. Figure 7 shows TensorFlow and Caffe results for scaling out with LeNet and MNIST mini-batch size of 64 images on each KNL. The number of images trained per millisecond is reported as more KNL machines are added. Each machine has one KNL. The network interconnect

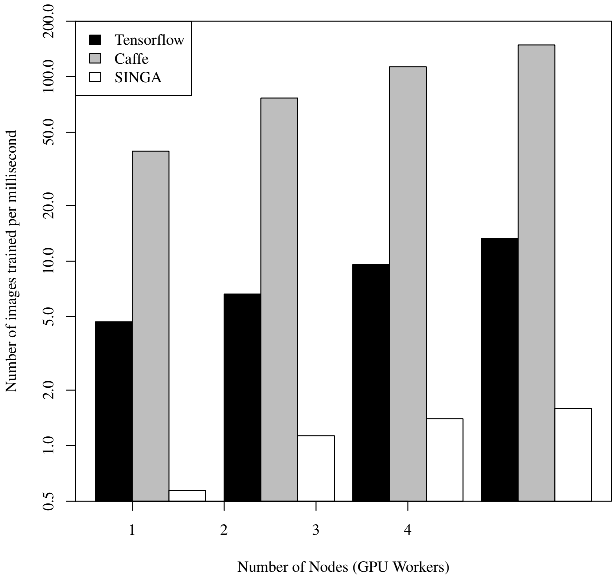

Fig. 6: TensorFlow and SINGA scale out results. Each node has one P100 GPU worker and each worker has a mini-batch size of 64.

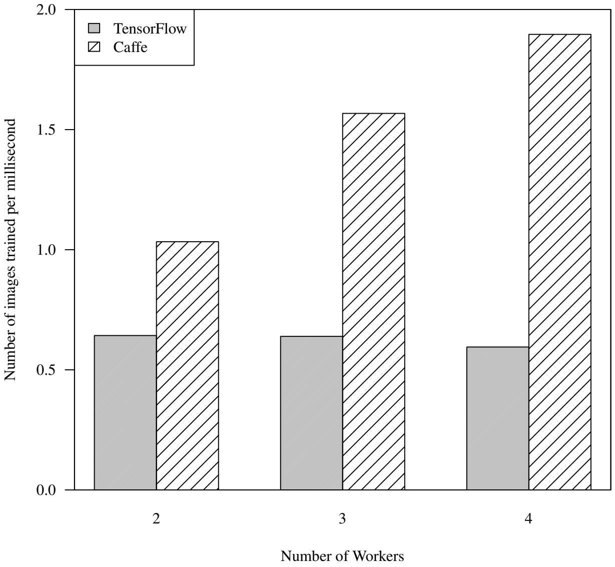

Fig. 7: Caffe and TensorFlow scale out results. Each node is one KNL worker and each worker has a mini-batch size of 64.

used is Intel Omni-Path. These results show the benefit of leveraging a particular hardware technology when developing a deep learning tool. Caffe outperforms TensorFlow in terms of image throughput as well as scalability. In fact, TensorFlow exhibits poor scalability, likely due to it not being developed with KNL in mind.

It is evident, based on Figures 6 and 7, that GPU outperforms KNL. We can more directly observe this in Figures 8 and 9, which provide omparisons of scaling out Caffe and TensorFlow, respectively, using KNL with Omni-Path and P100 with InfiniBand. Again, the LeNet model and MNIST dataset are used with a mini-batch size of 64 images. With regards to both Caffe and TensorFlow, the P100 provides drastically more throughput than the KNL.

D. Discussion

Efficiently scaling up single-node training can become challenging, particularly when using GPUs. GPU training introduces an additional overhead in transferring data to and from the CPU or other GPUs. This bottleneck limits the scalability of multiple-GPU machines. In this work, we used a newly developed technology, NVLink, to speed up data transfer. Figure 1 demonstrated the effect of having NVlink by comapring the time between P100 GPU with and without NVLink with the same batch size.

Scalable distributed training is arguably more challenging than scaling up in a single node because it is subject to the same CPU and GPU data transfers but with an additional, more severe bottleneck in the network. However, it is still important to push towards efficient distributed training in order to mitigate larger models and the constant increase in available data for training. In order to make this advancement, deep learning software should make clever use of model and data parallelism, have a strong communication architecture in place, and leverage state-of-the-art technologies.

Certain hardware technologies are already being well leveraged by deep learning tools, as shown in this paper. However, as exhibited by Intel Caffe in Figures 7 and 8, it shows tremendous speedup over TensorFlow when using the Knight's Landing but remains slower than when using the P100. In other words, Intel Caffe, developed with utilizing Intel Xeon processors and coprocessers, is still faster with P100 over the KNL. This is likely due to the early state of development of deep learning over KNL, especially when CUDA and CUDNN, libraries utilized by Caffe for deep learning over GPU, are much more established.

V. Conclusion

In this work, we evaluated three different deep learning tools, Caffe, TensorFlow, and SINGA, over a variety of hardware setups. Analysis was provided in terms of speed and scaling, promoting a better understanding of these tools and their performance in different scenarios. Of particular interest was the performance analysis using NVIDIA's NVLink technology over PCIe and using Intel's Knights Landing hosts with Intel Omni-Path interconnect.

As a result of these experiments, we have the following observations:

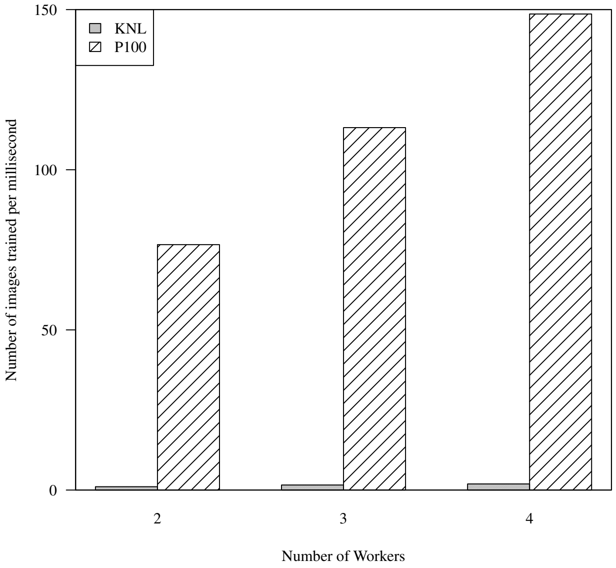

Fig. 8: Scale out result with Intel Caffe over P100 via InfiniBand and KNL via Omni-Path. For P100 results each node has one P100 GPU worker and for KNL result each node is one KNL worker and each worker has a mini-batch size of 64.

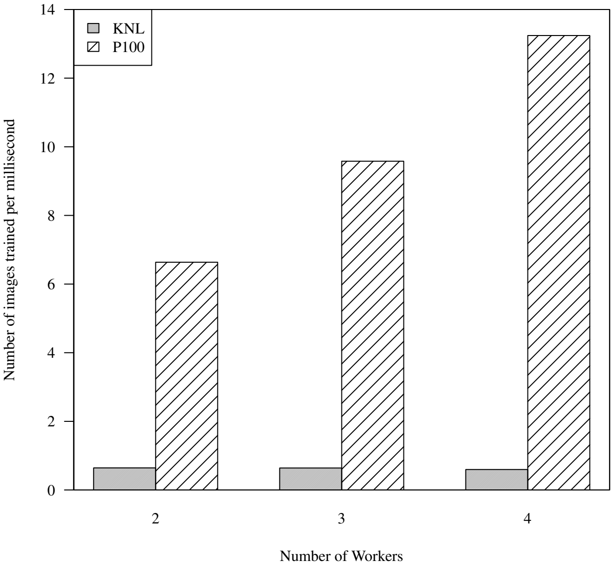

Fig. 9: Scale out result with TensorFlow over P100 via InfiniBand and KNL via Omni-Path. For P100 results each node has one P100 GPU worker and for KNL result each node is one KNL worker and each worker has a mini-batch size of 64.

- Apache SINGA may exhibit good scaling, but it leaves a lot to be desired in terms of training time compared to Caffe and TensorFlow.
- We have seen that P100 GPUs with NVLink consistently provide the best performance, in terms of the training time and communication speed, and significantly outperform Intel's Knight Landing.
- We have investigated the effect of Omni-Path versus InfiniBand as interconnect and showed that with even faster communication link, KNL is still behind the P100 GPUs in terms of performance and scalability.
- The experimental results obtained from the analysis of the different frameworks are as follows:
  1) Computation time: We have observed that Caffe is faster than other examined frameworks and it is outperforming TensorFlow. Larger networks, especially VGG-19, show the two frameworks much closer in timing, indicating that TensorFlow's performance suffers with smaller networks like LeNet. Both Caffe and TensorFlow outperform SINGA by a large margin.
  2) Scalability results: All three frameworks, Caffe, TensorFlow, and SINGA, scale-up with multiple GPUs on one node and scale-out with multiple nodes linearly, while again Caffe outperforms TensorFlow, which confirms our previous observation. However, when utilizing KNL and Omni-Path, Tensorflow does not scale well. This is not unexpected since it was not designed to run over KNL or other similar processors and coprocessors.

To our best knowledge, this is the first work evaluating NVLink and Intel's Knights Landing for deep learning benchmarks. In future works, we plan to add more frameworks for benchmarking and expand our hardware environment with an aim towards scaling out. As deep learning continues to thrive, it will continue to be necessary to evaluate the tools developed over state-of-the-art hardware.

A. Future work

One aspect in which we plan to extend this work involves adding more novel hardware that may be of interest to evaluate includes other Xeon Phi co-processors (e.g., Knight's Corner) and FPGAs. On top of adding additional hardware technologies, we plan to include additional popular deep learning models and frameworks. Specifically, we aim to evaluate more frameworks capable of distributed training. With this, further experiments and more detailed analysis towards scaling out deep learning will be done.

Acknowledgments: This work was funded by NIH P20GM103458-10, NIH P30GM110760-03, NIH P20GM103424, NSF award 1341008 (CC-NIE Integration: Bridging, Transferring and Analyzing Big Data over 10Gbps

Campus-Wide Software Defined Networks), National Science Foundation under grants MRI-1338051 and CC-NIE-1341008, LA Board of Regents under grant LEQSF(2016-19)-RD-A-08 and an IBM faculty award. The computing resources were made available thanks to resource allocation grants from XSEDE, Louisiana Optical Networking Initiative (LONI), and Louisiana State University High-performance Computing (HPC). The authors would like to thank IBM and Dell for their generosity in allowing them to use their compute resources.

References

[1] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22Nd ACM International Conference on Multimedia*, ser. MM '14. New York, NY, USA: ACM, 2014, pp. 675–678. [Online]. Available: http://doi.acm.org/10.1145/2647868.2654889

[2] W. Wang, G. Chen, T. T. A. Dinh, J. Gao, B. C. Ooi, K.-L. Tan, and S. Wang, "SINGA: Putting deep learning in the hands of multimedia users," in *ACM Multimedia*, 2015.

[3] B. C. Ooi, K.-L. Tan, S. Wang, W. Wang, Q. Cai, G. Chen, J. Gao, Z. Luo, A. K. H. Tung, Y. Wang, Z. Xie, M. Zhang, and K. Zheng, "SINGA: A distributed deep learning platform," in *ACM Multimedia*, 2015.

[4] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Józefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2016. [Online]. Available: http://arxiv.org/abs/1603.04467

[5] S. Shi, Q. Wang, P. Xu, and X. Chu, "Benchmarking state-of-the-art deep learning software tools," *CoRR*, vol. abs/1608.07249, 2016. [Online]. Available: http://arxiv.org/abs/1608.07249

[6] S. Bahrampour, N. Ramakrishnan, L. Schott, and M. Shah, "Comparative study of caffe, neon, theano, and torch for deep learning," *CoRR*, vol. abs/1511.06435, 2015. [Online]. Available: http://arxiv.org/abs/1511.06435

[7] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *CoRR*, vol. abs/1512.01274, 2015. [Online]. Available: http://arxiv.org/abs/1512.01274

[8] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[9] G. Lacey, G. W. Taylor, and S. Areibi, "Deep learning on fpgas: Past, present, and future," *CoRR*, vol. abs/1602.04283, 2016. [Online]. Available: http://arxiv.org/abs/1602.04283

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[13] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[14] R. Platania, S. Shams, C.-H. Chiu, N. Kim, J. Kim, and S.-J. Park, "Hadoop-based replica exchange over heterogeneous distributed cyberinfrastructures," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 4, pp. e3878–n/a, 2017, e3878 cpe.3878. [Online]. Available: http://dx.doi.org/10.1002/cpe.3878

[15] A. K. Das, S. J. Park, J. Hong, and W. Chang, "Evaluating different distributed-cyber-infrastructure for data and compute intensive scientific application," in *2015 IEEE International Conference on Big Data (Big Data)*, Oct 2015, pp. 134–143.