Name: Prashanth Mallyampatti
Unity ID: pmallya
Student ID: 200250501

CSC724- Advanced Distributed Systems
Paper Review

TensorFlow: A System for Large-Scale Machine Learning

Summary:

In this paper a system for Large-Scale machine learning has been discussed. Machine Learning has been a topic of research and has accelerated the growth of many fields. To target the fact that creation of more sophisticated models, the availability of large datasets, and the development of software that enables resources to train over these datasets, TensorFlow has come up with a methodology aiming for flexibility.

The use of GPU-enabled servers for fast training, flexibility of allowing experimentation on new training models and system-level optimizations are the key points of TensorFlow. DistBelief a previous system of TensorFlow uses the parameter server architecture. DistBelief operated on directed graph of computation layers, has stateless worker nodes which do most of the computation. The parameter server processes maintain the current version of the parameters of the model. Due to the structure of directed acyclic graph, it has many disadvantages, such as – new programming languages aren't supported or else new layers have to be designed for them, without modifying the parameter server its hard to experiment with new optimization methods, and has a poor performance for new algorithms specifically recurrent and adversarial neural networks. To address these TensorFlow focuses on new models, training then on a large dataset and moving them to production. The TesnsorFlow execution model uses dataflow graphs to represent simpler operators in machine learning. Each operation represented by a vertex represents a unit of local computation. The edges called Tensor represents the output from the vertex or input to the vertex. On the user side, the user selects a subgraph that represents all possible computations and executes it. An operation can contain a mutable state that is read or write every time it executes. Multiple invocations can occur concurrently where each invocation is counted as a step, hence the dataflow is simplified to a distributed execution and is optimized for executing large subgraphs repeatedly with low latency. Each operation resides CPU, TPU or GPU in a particular task and this device is responsible executing a kernel for each operation. TensorFlow allows multiple kernel execution for a single operation. This paper also discusses Extensibility case studies, wherein TensorFlow enables the user to experiment on hard coded features in DistBelief. Some of them being- Differentiation and optimization, training on large datasets, synchronous replica coordination and the important one fault tolerance. The concept of checkpointing helps the developers to check back in case of failures at a particular point and user level checkpointing for algorithms running for days.

Coming to the implementation, TensorFlow library is implemented in C++ for portability and performance, with wide platform compatibility. TensorFlow was evaluated on various aspects- Single machine, synchronous replica, Image classification and Language modeling, each of which showing promising results and with less overheads.

Strong Points:

1) The main strength of this paper is that it provides a flexible system capable of performing ML computations with a good performance.
2) It supports large scale dataset training and inference which supports a large number of GPU trainings together and can run in either synchronous or asynchronous mode.
3) TensorFlow achieves high GPU utilization by using graph's dependency structure, which is used to issue a sequence of kernels to the GPU without waiting for intermediate results.

Weak Points:

1) Evaluations only on Nvidia GPU's have been discussed, its performance might differ with other GPU's.
2) Some more ML jobs could have been run and evaluated upon.