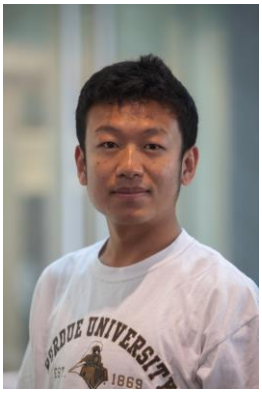
Carnegie Mellon University
Computer Science Department

THESIS PROPOSAL



Jinliang Wei

Efficient and Programmable Distributed Shared Memory Systems for Machine Learning Training

Friday, October 5, 2018 – 3:00 p.m. – NSH 3305

Machine learning training involves frequent and often sparse updates to a large number of numerical values called model parameters. Many distributed training systems have resorted to using distributed shared memory (DSM) (e.g. Parameter Server) for efficient sparse access and in-place updates. Compared to traditional programs, machine learning applications tolerate bounded error, which presents opportunities for trading off learning progress for higher computation throughput. In this thesis, I develop efficient and programmable distributed learning systems, by exploiting this trade-off in the design of distributed shared memory systems, along with parallelization and static and dynamic scheduling.

Thanks to this tolerance to bounded error, a machine learning program can often be parallelized without strictly preserving data dependence. Parallel workers may thus observe inconsistent model parameter values compared to a serial execution. More frequent communication to propagate updates and fresher parameter values may reduce such inconsistency, while incurring higher inter-machine communication overhead. I present a communication management mechanism that automates communication using spare network bandwidth and prioritizes messages according to their importance in order to reduce error due to inconsistency while retaining high computation throughput.

When each model update reads and writes to only a subset of model parameters, it is possible to achieve an efficient parallelization while preserving critical data dependence, exploiting sparse parameter access. Existing systems require substantial programmer effort to take advantage of this opportunity. In order to achieve dependence-preserving parallelization without imposing a huge burden on application programmers, I present a system Orion that provides parallel for-loops on distributed shared memory and parallelizes loops with loop-carried dependence.

At last, I propose to explore dynamic scheduling for dynamic control flow in dataflow systems such as TensorFlow. In TensorFlow, the computation graph is statically partitioned and assigned with computation devices. Static device placement is suboptimal as the operators' load can no longer be determined statically due to dynamic control flow. A suboptimal static device placement may result in imbalanced load and extra communication. It is promising to address the deficiency of static device placement by dynamically scheduling operations based on their load at runtime.

Thesis Committee:
Garth A. Gibson, Co-chair
Eric P. Xing, Co-chair
Phillip B. Gibbons
Vijay Vasudevan, Google Brain

Thesis Summary: http://www.cs.cmu.edu/~jinlianw/papers/thesis_proposal.pdf