Optimization of CAFFE

Kumudha K.N. (12588), MSc(Engg), CSA

December 10, 2015

1 Optimizations

1.1 Compiler flags

The following compiler Flags were used -

1. -DTIME : To print the time taken by the train operation
2. -O2 : Used when loops use floating point calculations
3. -fopenmp : Enables the parallelizer to generate multi-threaded code
4. -o : to specify the output filename
5. -ipo : Enable interprocedural optimization
6. -fma : when specified, the compiler may generate FMA instructions for combining multiply and add operations where applicable

2 Performance

We performed the optimizations as specified in Section 1, and got the training time of $\boldsymbol{MNIST}$ network to **278.091** seconds for 10000 iterations and Number of cores used is 16. Whereas, the original code took 767.852 seconds on 16 cores.
Similarly, Training time of $\boldsymbol{CIFAR-10}$ changed from 259.360 seconds to **188.411** seconds on a 16 core machine The performance of running on the GPU was

- MNIST - 398.005 seconds for 10000 iterations
- CIFAR10 - 51.8234 seconds for 5000 iterations

2.1 Processor Details

Performance was measured with a system with the below configuration -
Mcastle1 machine
Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz
48 cores
L1 cache: 32K data and instruction
L2 cache: 256K
L3 cache: 30720K

Mcastle2 machine
AMD Opteron(tm) Processor 6386 SE
64 cores
L1d cache: 16K data and 64K instruction
L2 cache: 2048K
L3 cache: 6144K

2.2 Evaluation

Figure 1 shows the speedup obtained for MNIST network, for a batch size of 64

Figure 2 shows the speedup obtained for CIFAR-10 network

We also compared our resulkts against the training time obtained by GPUs to have a sense of the amount of optimization still achievable. The compariasion is in Figure 3

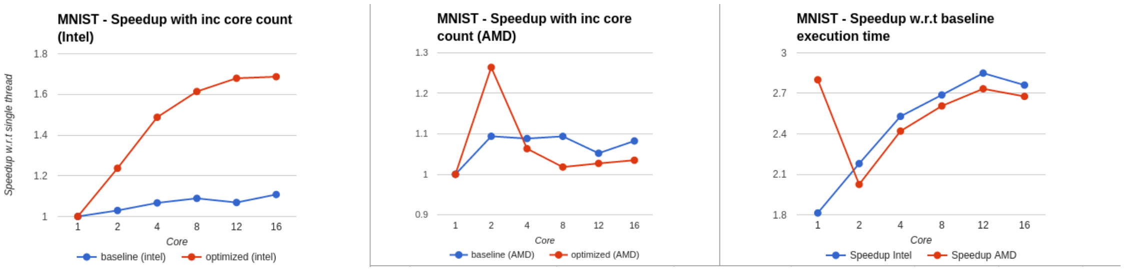

Figure 1

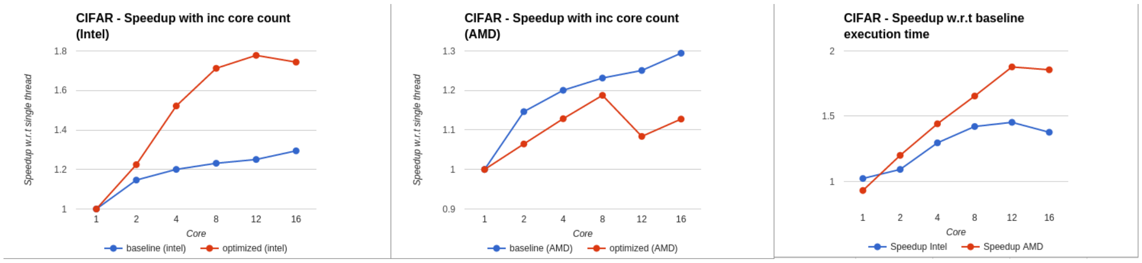

Figure 2

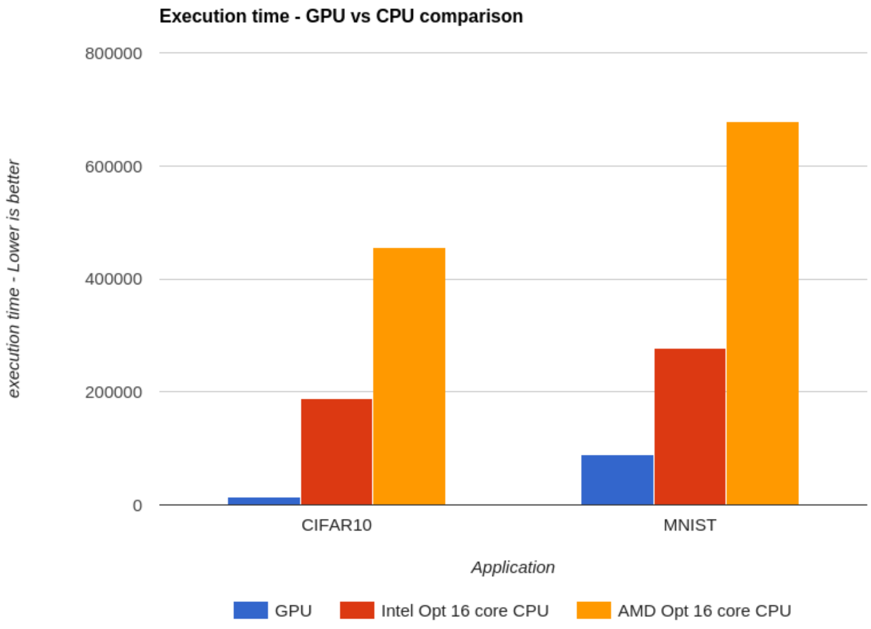

Figure 3