Caffe-MPI: A parallel Framework on the GPU Clusters

Shaohua Wu
Senior Software Engineer
wushh@inspur.com
Caffe-MPI

- What is Caffe-MPI?
  - Developed by Inspur
    - Open-source: https://github.com/Caffe-MPI/Caffe-MPI.github.io
  - Programmed by MVAPICH
  - Based on the Berkeley Vision and Learning Center (BVLC) single node version
  - A GPU Cluster version
  - Support 16+ GPUs to Train
Analysis of Caffe

- Caffe needs long training time for big data sets on a single node.
Caffe-MPI Architecture

• HPC Technology
  – Hardware arch: IB+GPU cluster+Lustre
  – Software arch: MPI+Pthread+CUDA

• Data parallel on GPU Cluster

<table>
<thead>
<tr>
<th>GPU Cluster Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GPU master node</strong></td>
</tr>
<tr>
<td><strong>GPU Slave Node</strong></td>
</tr>
<tr>
<td><strong>Storage</strong></td>
</tr>
<tr>
<td><strong>network</strong></td>
</tr>
<tr>
<td><strong>Software</strong></td>
</tr>
</tbody>
</table>
MPI Framework Design

- MPI Master-Slave model
  - Master Process: Multi Pthread Threads + CUDA Threads
  - Slave Process: CUDA Threads

Design of Master Process

• Master Process (0 process)
  – Three functions
    • Parallel read data and send data
    • Weight Computing and The parameter update
    • The parameter communication
Design of Slave Process

- Slave process
  - CPU
    - To receive training data from the master process
    - To send weight data (GPU-to-GPU)
    - To receive new net data (GPU-to-GPU)
  - GPU
    - ForwardBackward computing

- Slave Node
  - The number of Slave process = the number of GPU
Features of the Computing & Communication

- GPU parallel computing
- Computing & Communication asynchronous parallel
- Communication Optimization
  - GPU RDMA: Weight Data and Net data between GPUs

Total Time = \( \max(T_{\text{Read Data} + \text{Send Data}}, T_{\text{ForwardBackWord Computing} + \text{Weight Computing and Net Update} + \text{Net Send}}) \)
The Performance of Caffe-MPI

- Speed-up Ratio: \(16\text{GPU}/1\text{GPU}=10.45\times\)
- Scalability efficiency: 65%
Tuning 1: Change BatchSize

- Speed-up Ratio: 16GPU/1GPU = 10.74X
- Scalability efficiency: 67%
Tuning 2: Caffe-MPI+cuDNN

- 21% Performance improvement by cuDNN
- Speed-up: 16GPU vs. 1GPU = 12.66x
- Scalability: 79%

GoogleNet (Iterations = 4000, batch size = 64)
Tuning 3: Parallelizing Read and Send Data

- Parallelizing read training data from Lustre Storage and send data to different GPUs
  - GPU Cluster is divided into sub groups
  - Each group has a master node
  - Each master node read and send data in parallel with Multi Processes and Multi Threads

- Support large-scale GPU computing platform for large training data set
The Performance of Caffe-MPI

- Speed-up Ratio: 16GPU/1GPU = 13X
- Scalability efficiency: 81%
Caffe-MPI Plan

- Plan:
  - Support cuDNN 4.0
  - MPI tuning
    - Symmetric model
Conclusions

• Caffe-MPI
  – 13x performance improvements: 16 GPU vs. 1GPU
• Support 16+ GPU for large data sets
  – Improved master-slave model
• Open source
THANKS