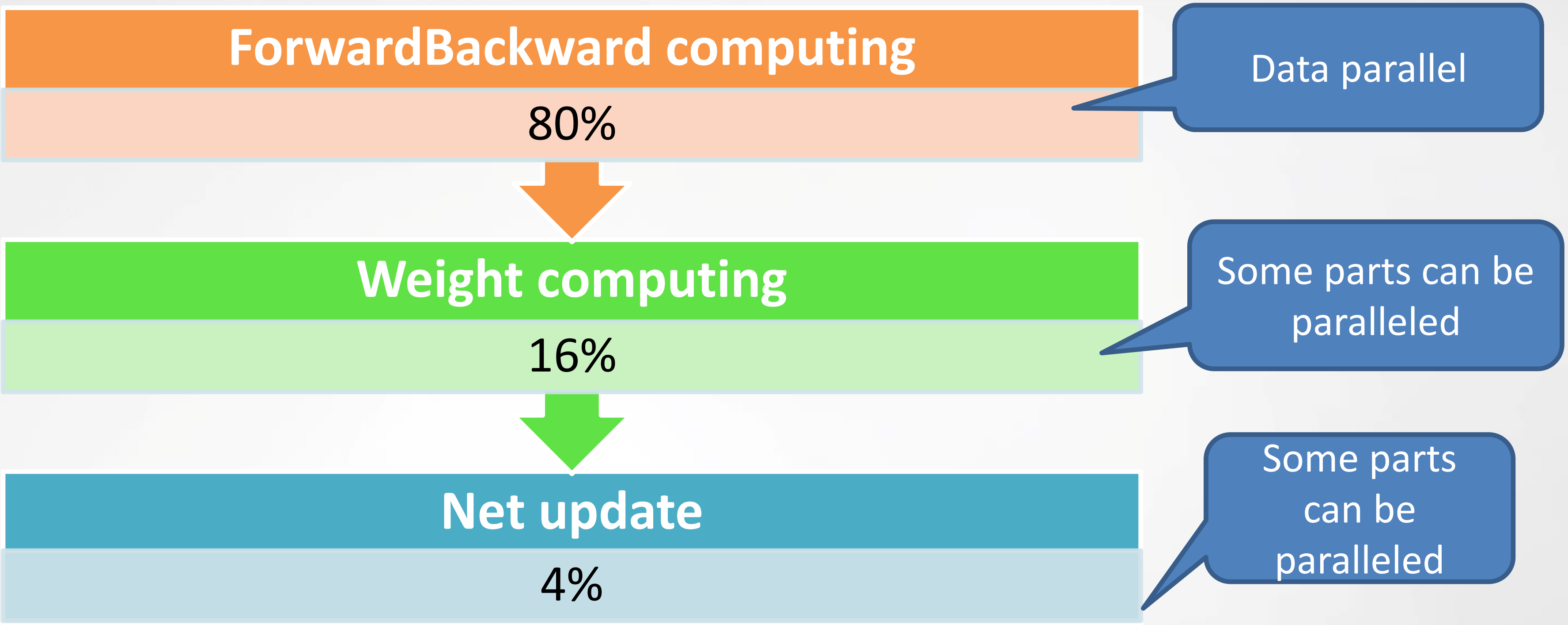
Caffe-MPI: A parallel Framework on the GPU Clusters

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



ForwardBackward computing

80%

Data parallel

Weight computing

16%

Some parts can be paralleled

Net update

4%

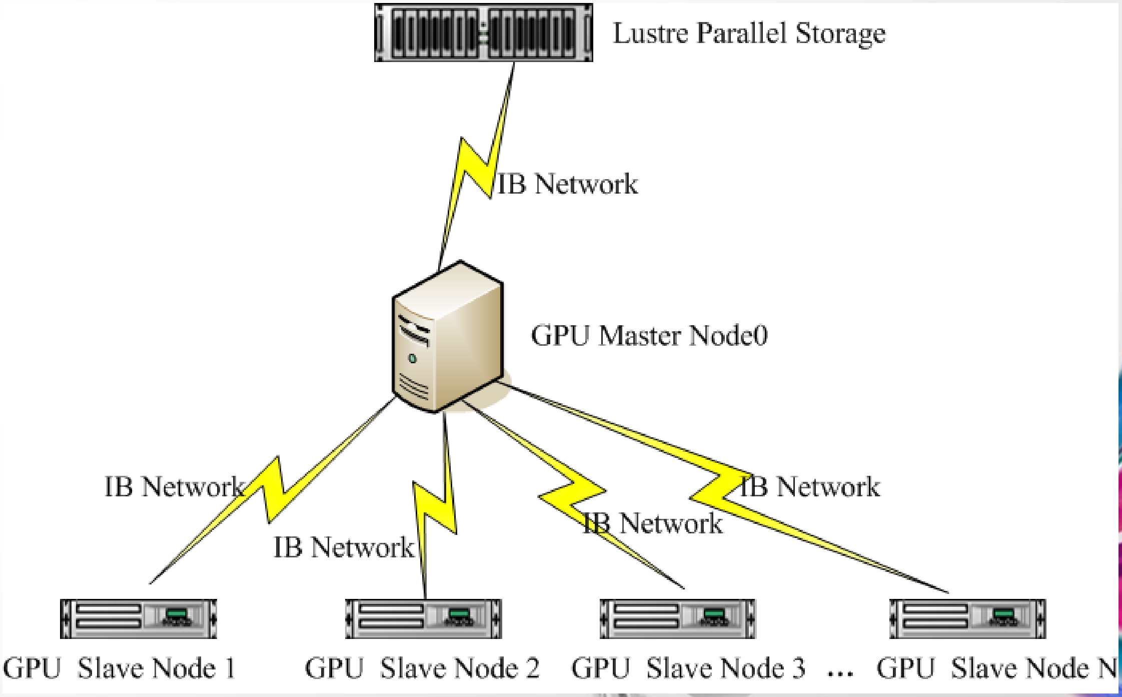
Some parts can be paralleled

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

Caffe-MPI Architecture

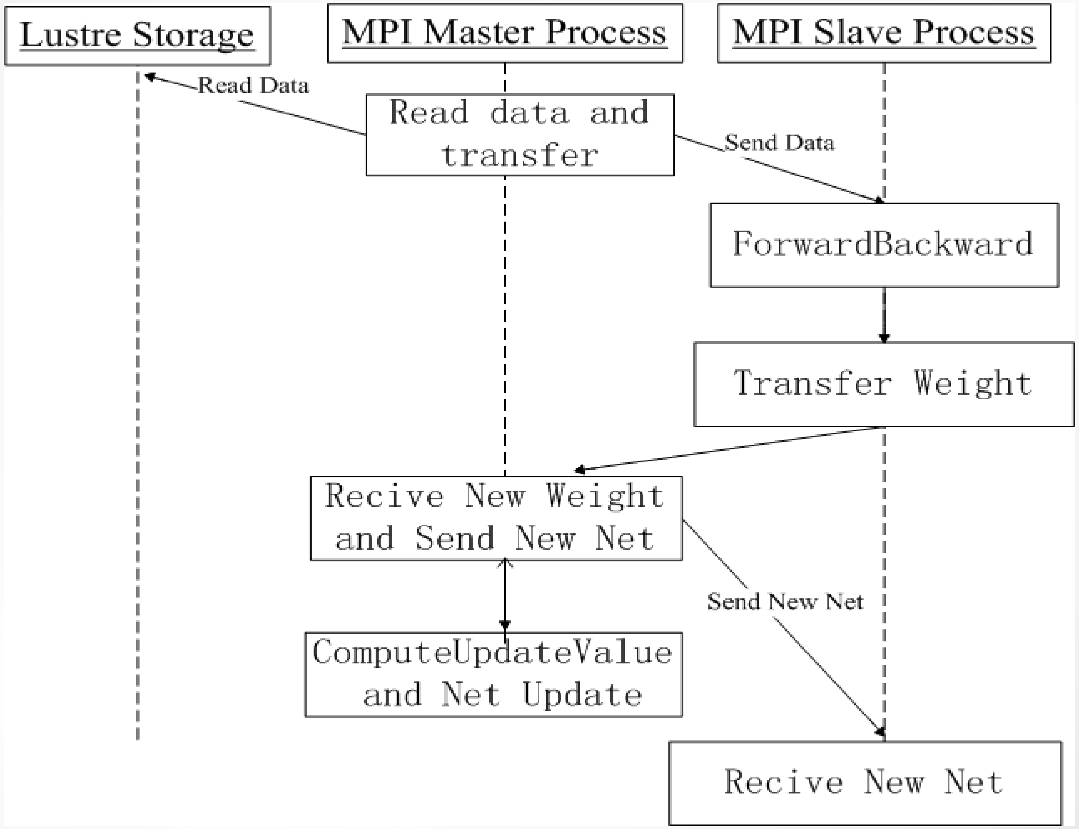
- HPC Technology
  - Hardware arch： IB+GPU cluster+Lustre
  - Software arch： MPI+Pthread+CUDA
- Data parallel on GPU Cluster

GPU Cluster Configuration	
GPU master node	Multi GPUs
GPU Salve Node	Multi GPUs
Storage	Lustre
network	56Gb/s IB
Software	Linux/Cuda7.5/Mvapich2



MPI Framework Design

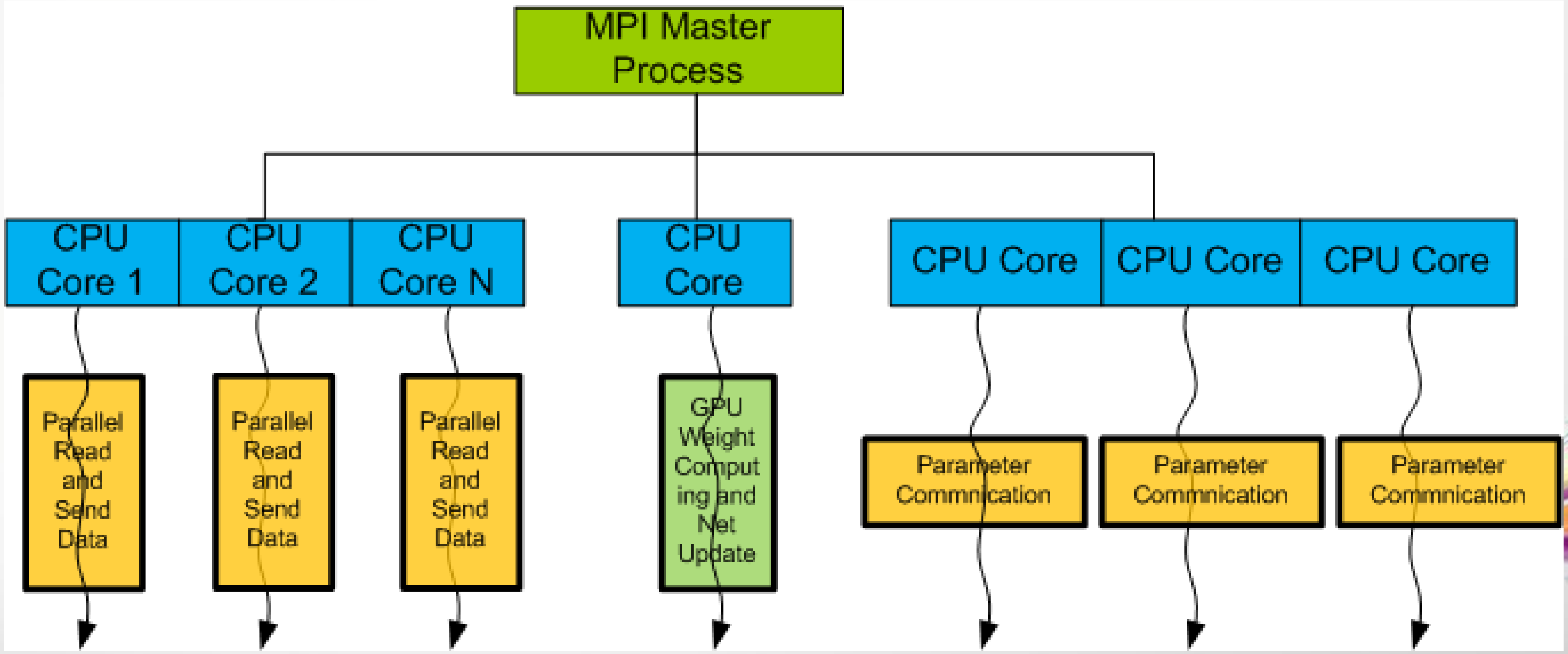
- **MPI Master-Slave model**
  - Master Process: Multi Pthread Threads+CUDA Threads
  - Slave Process: CUDA Threads



Reference： Q Ho, J Cipar, H Cui, J. K Kim, S Lee, P. B. Gibbons, G. A. Gibson, G. R. Ganger, E. P. Xing. More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server.

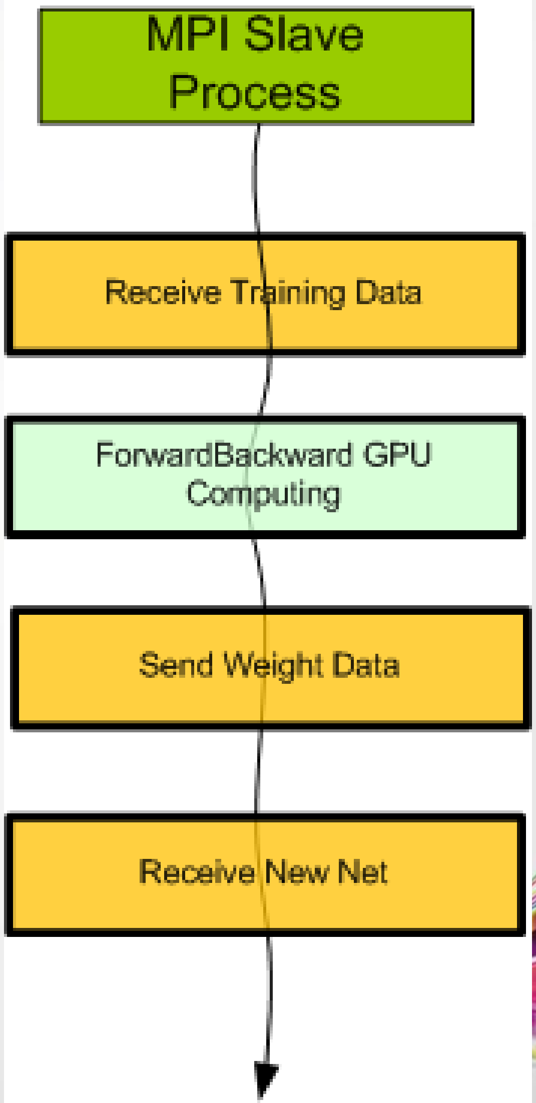
Design of Master Process

- Master Process (0 process)
  - Three functions
    - Parallel read data and send data
    - Weight Computing and The parameter update
    - The parameter communication

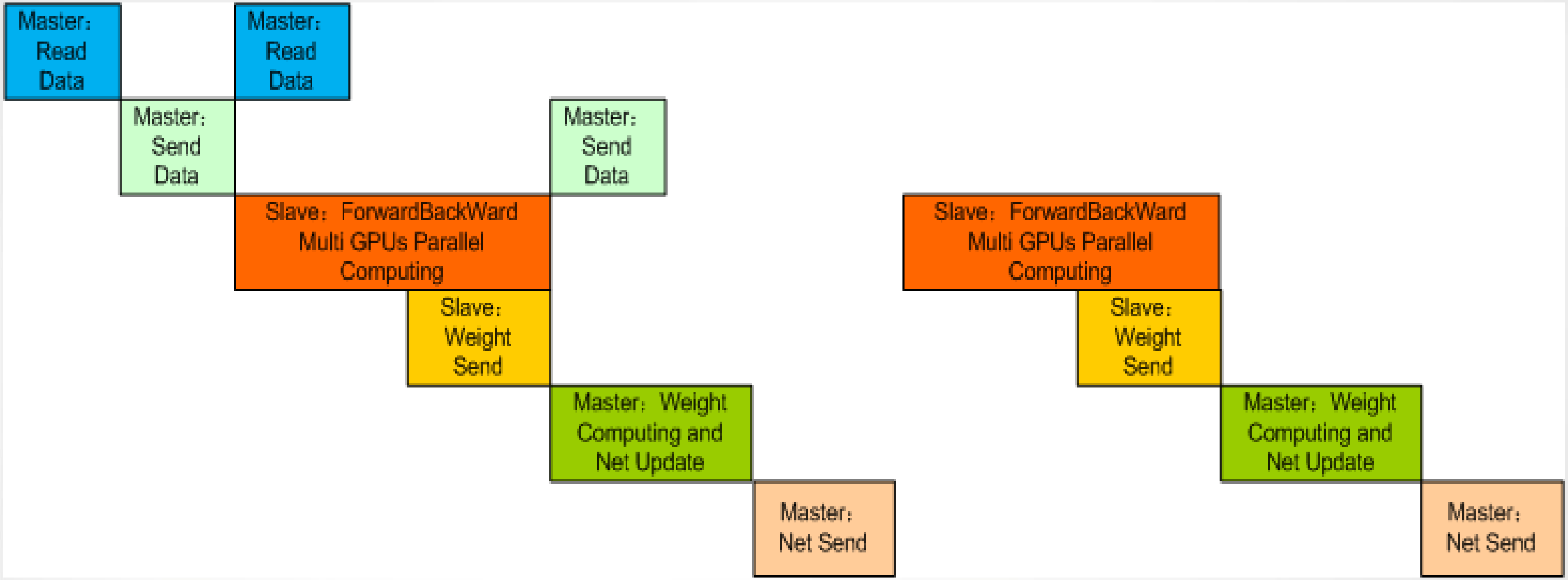


Design of Slave Process

- 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



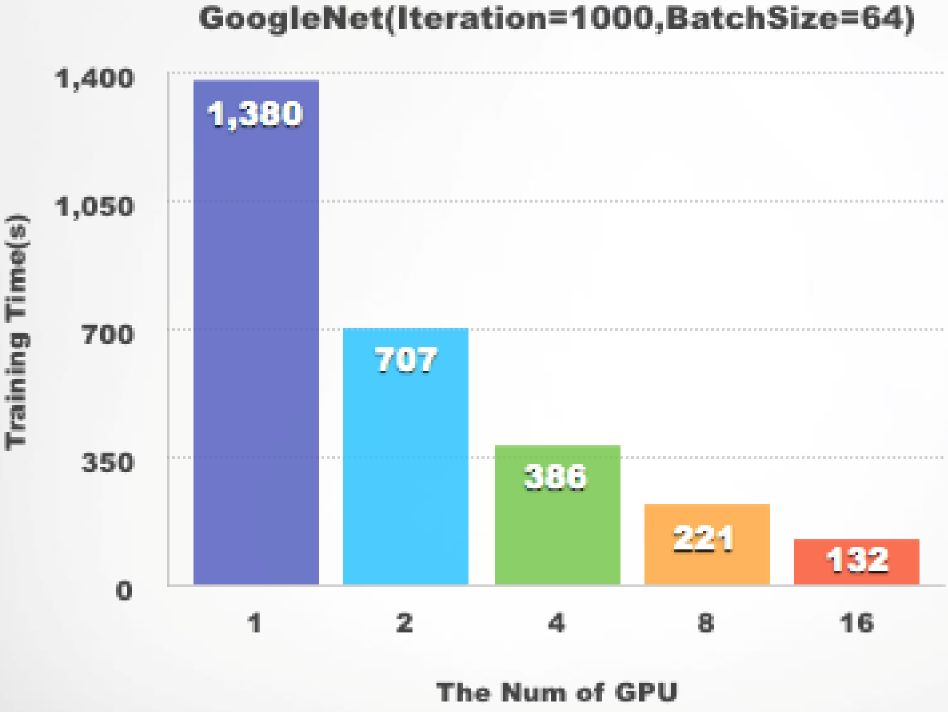
Master：Read Data
Master：Send Data
Master：Read Data
Master：Send Data
Slave：ForwardBackWard Multi GPUs Parallel Computing
Slave：Weight Send
Master：Weight Computing and Net Update
Master：Net Send
Slave：ForwardBackWard Multi GPUs Parallel Computing
Slave：Weight Send
Master：Weight Computing and Net Update
Master：Net Send

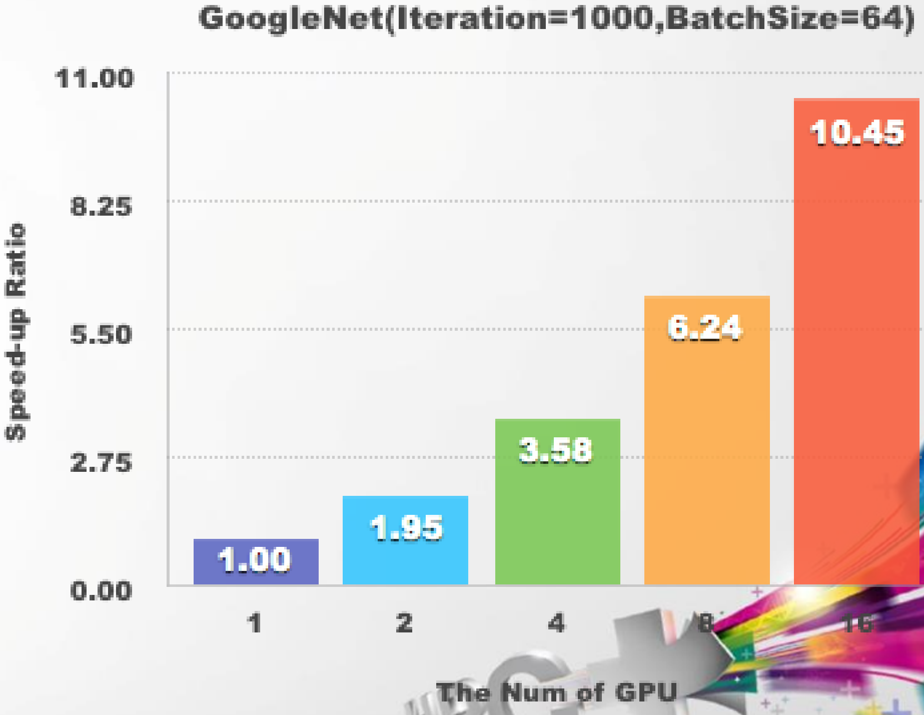
- GPU parallel computing
- Computing & Communication asynchronous parallel
- Communication Optimization
  - GPU RDMA：Weight Data and Net data between GPUs

Total Time=max($T_{\text{Read Data+Send Data}}$,$T_{\text{ForwardBackWord Computing+ Weight Computing and Net Update+ Net Send}}$)

The Performance of Caffe-MPI

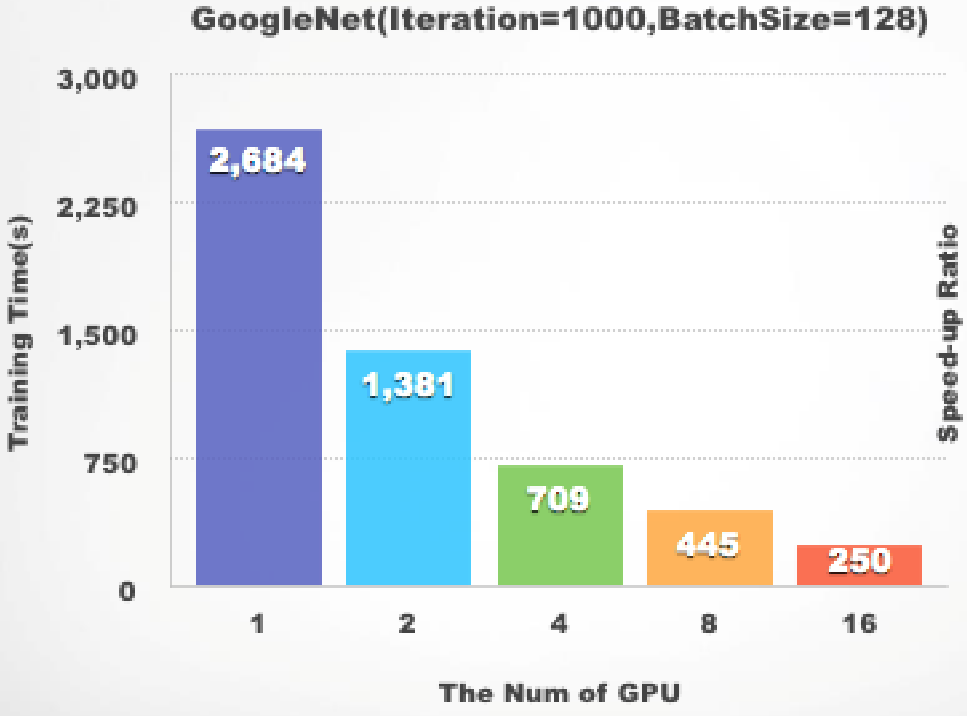
- Speed-up Ratio：16GPU/1GPU=10.45X
- Scalability efficiency：65%

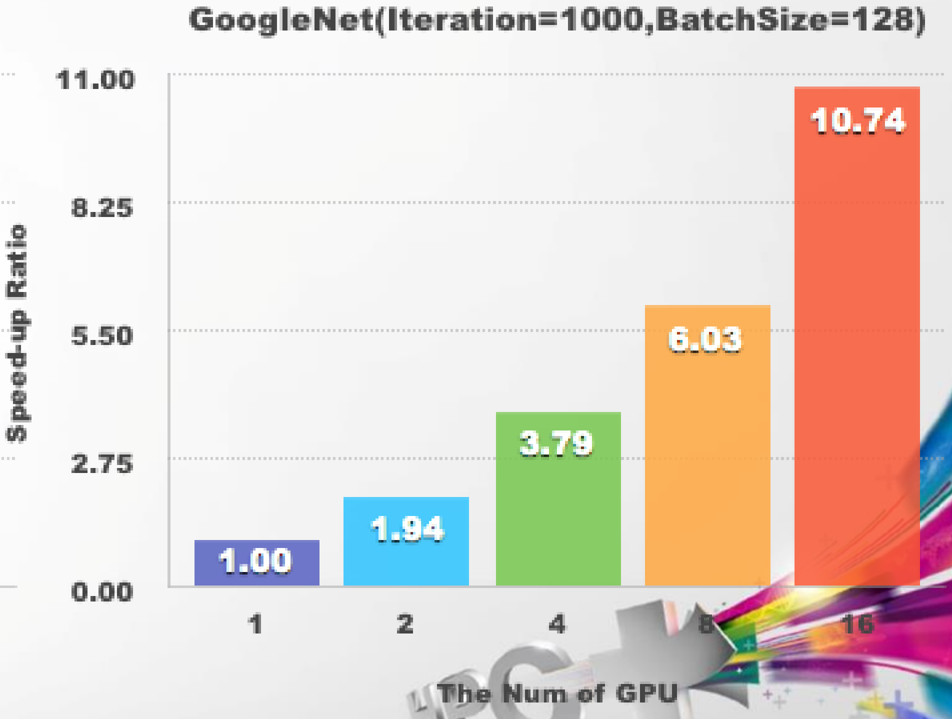




Tuning 1： Change BatchSize

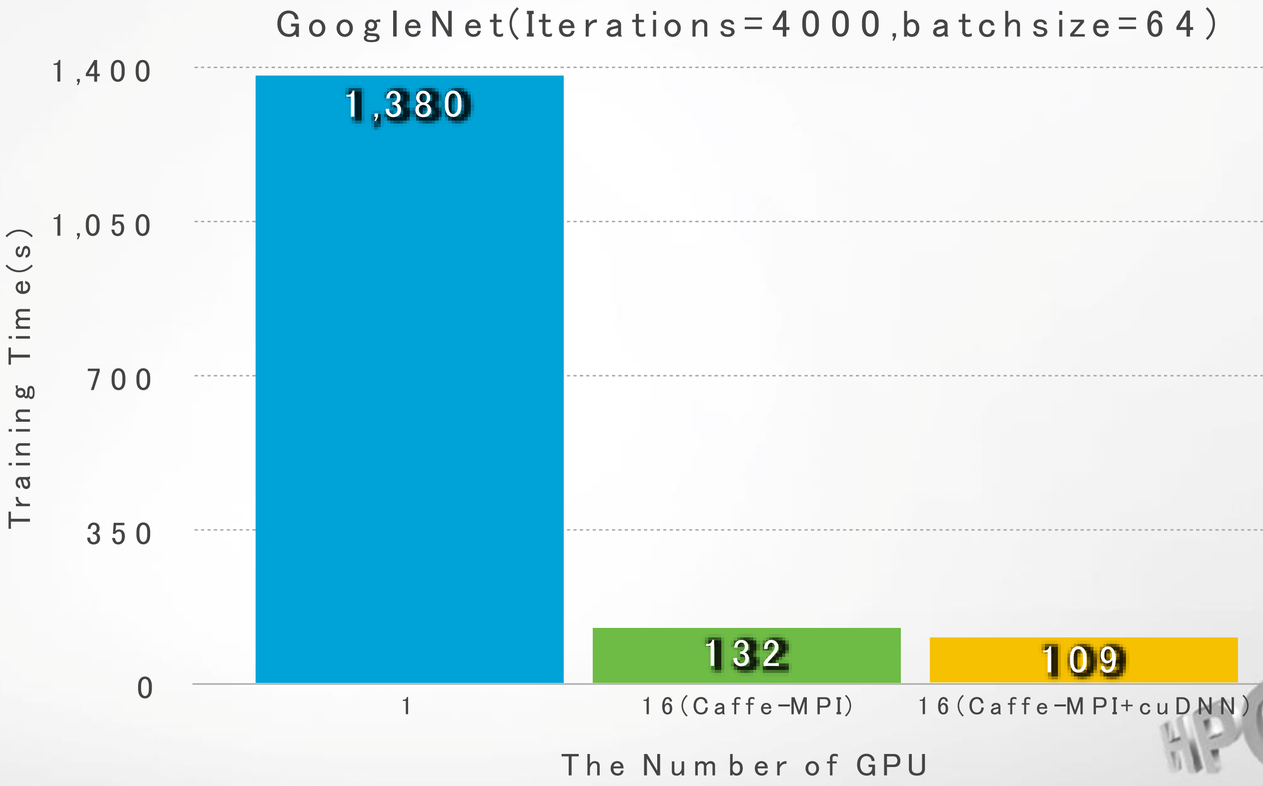
- Speed-up Ratio： 16GPU/1GPU=10.74X
- Scalability efficiency： 67%





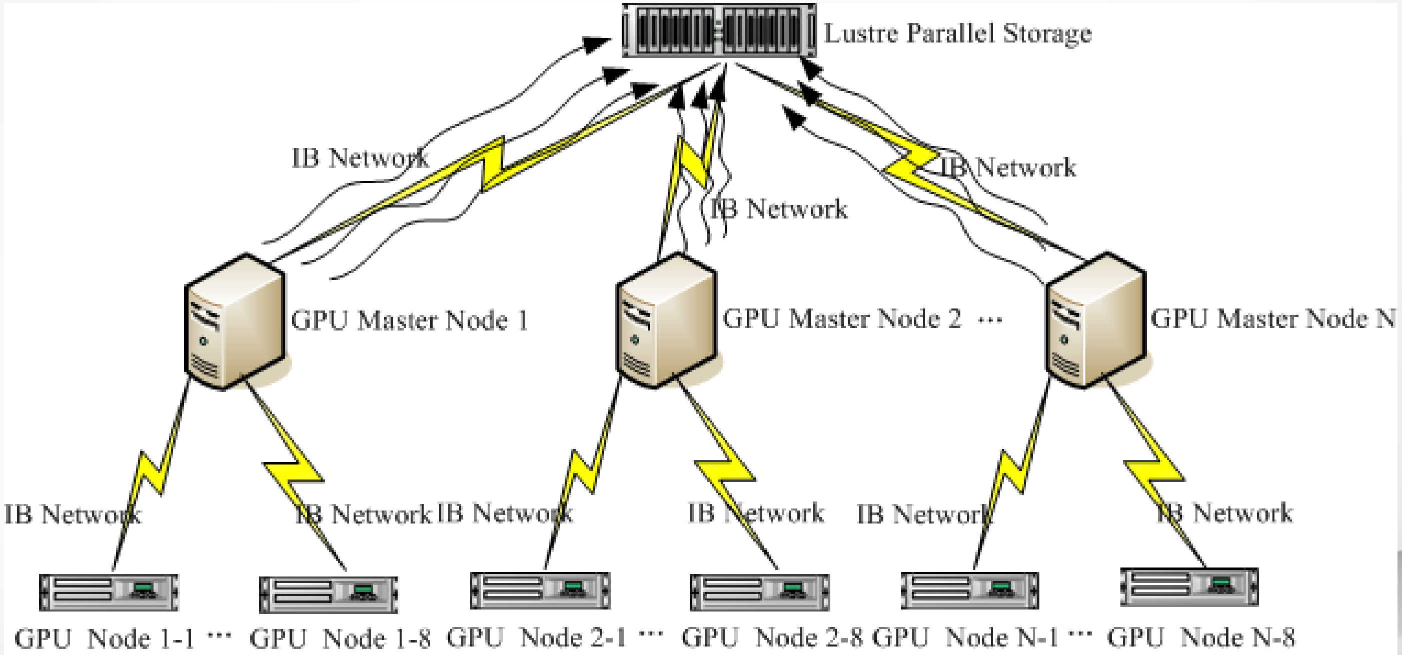
Tuning 2：Caffe-MPI+cuDNN

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



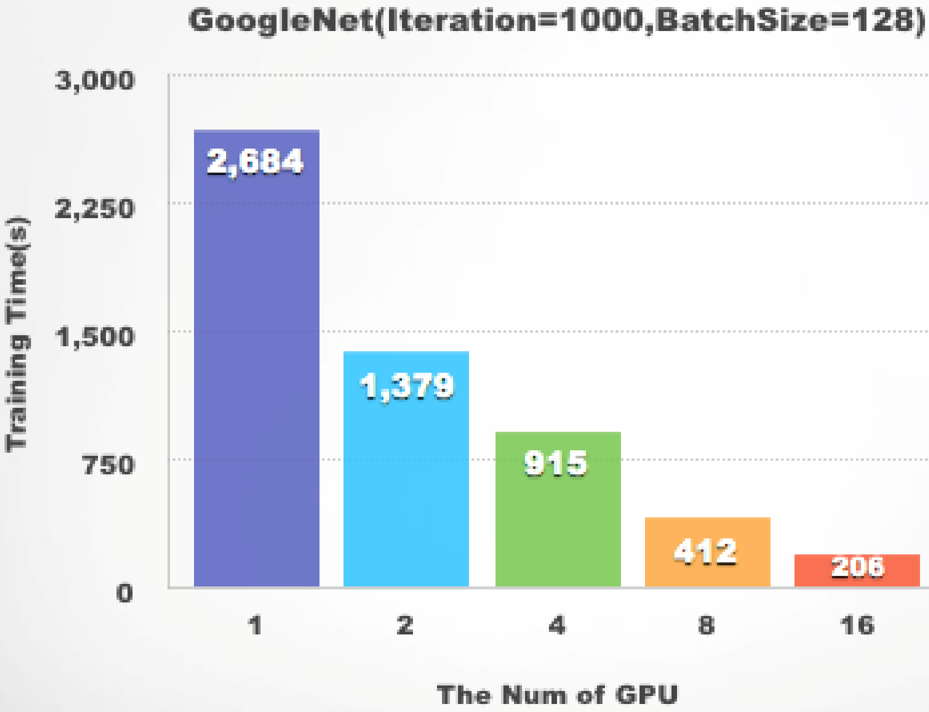
Tuning 3: Parallelizing Read and Send Data

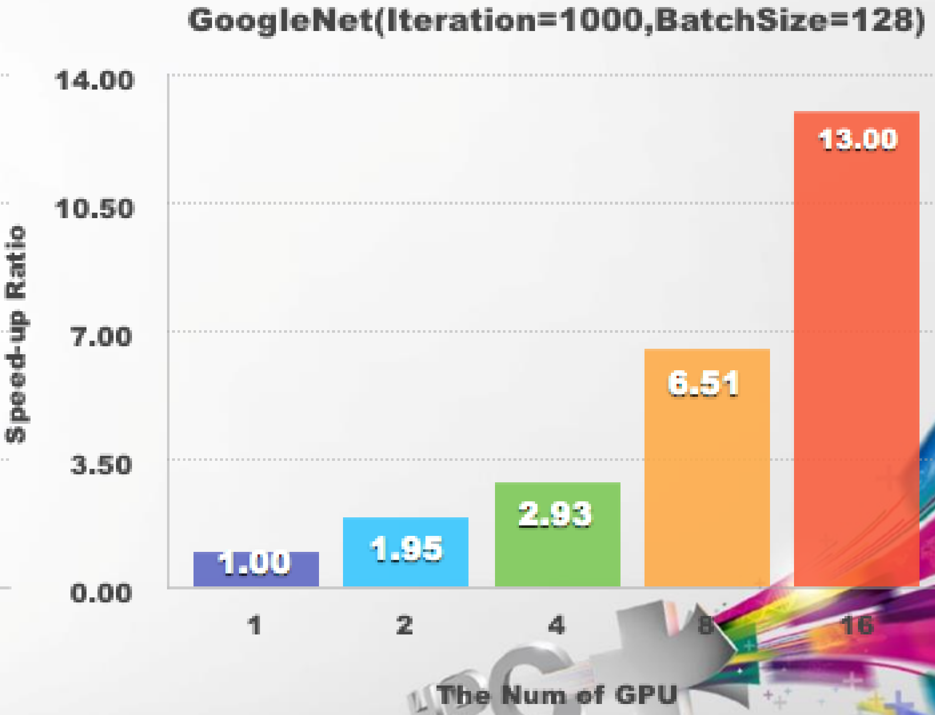
- Parallelizing read training data from Lustre Storage and send data to different GPUs
  - GPU Cluster is divided into sub groups
  - Each group has a master node
  - Each master node read and send data in parallel with Multi Processes and Multi Threads
- **Support large-scale GPU computing platform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