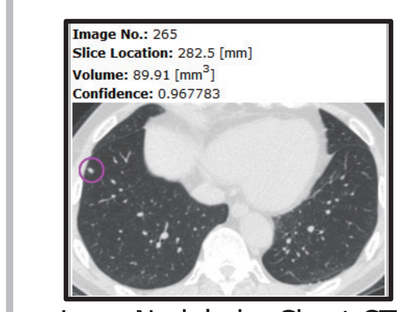
Large-scale Deep Learning using Supercomputer

Toshihiro Hanawa, hanawa@cc.u-tokyo.ac.jp

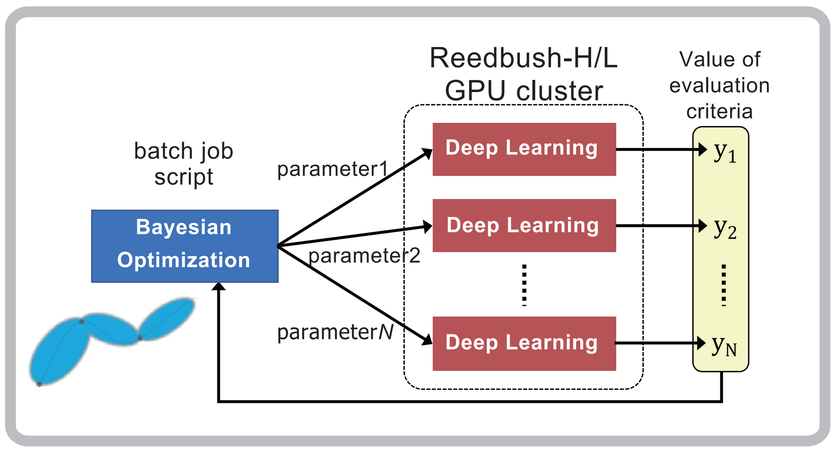
Deep Learning Training for Medical Images using Supercomputer based on Asynchronous Parallel Bayesian Optimization

We collaborate with Univ. Tokyo Hospital to realize high efficient medical image analysis using ITC' s supercomputers. Deep learning requires large amounts of computational power, and numerous hyper-parameter optimization has great influence on the performance of deep learning. Thus, we have been developing a framework for training deep learning with hyper-parameter optimization on the supercomputer system.

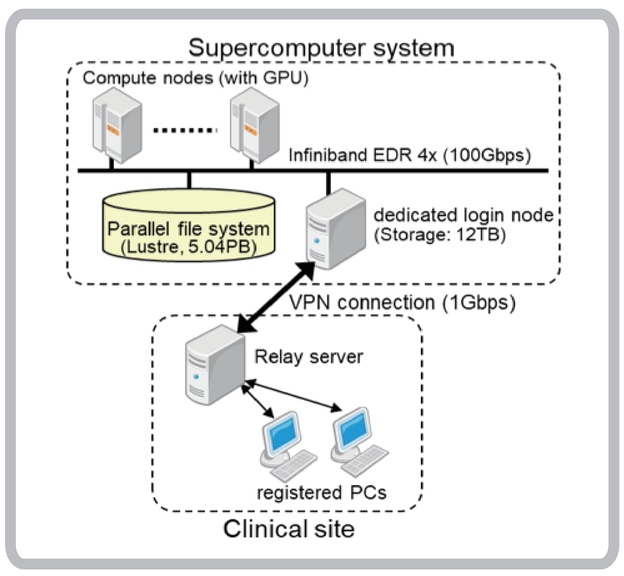


Lung Nodule by Chest CT
(○:detected lesion)

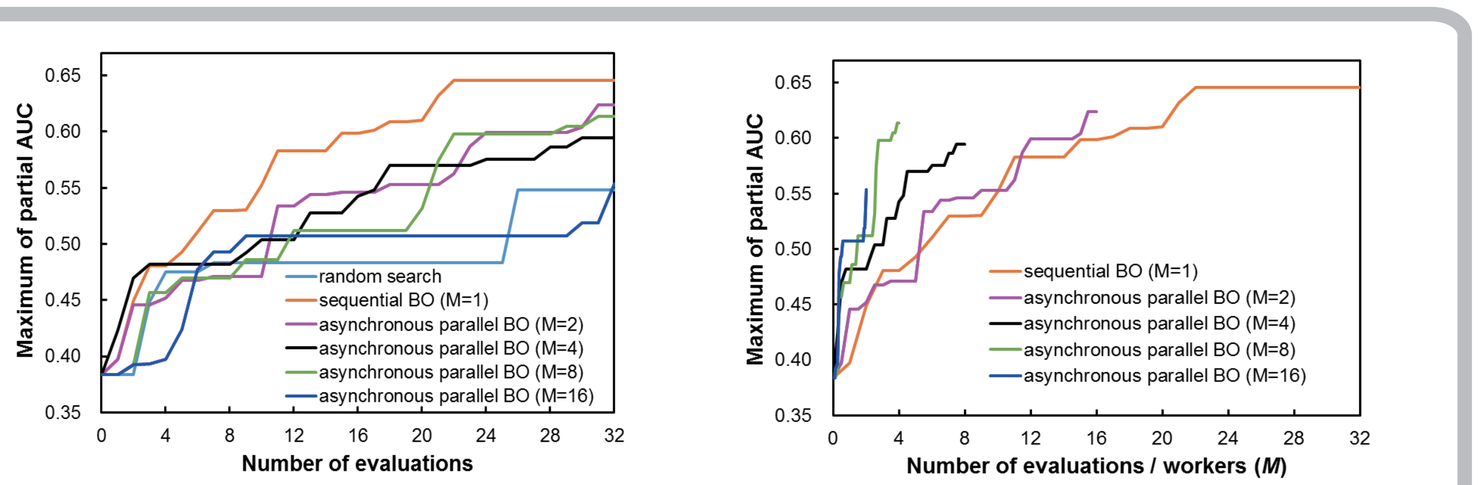
Examples of Lung Nodule detection
[Courtesy: Dr. Y. Nomura
(UTokyo Hospital)]



Automated Hyper-parameter Tuning
by Bayesian Optimization on Reedbush



Configuration of Medical Image Analysis System
with Reedbush system



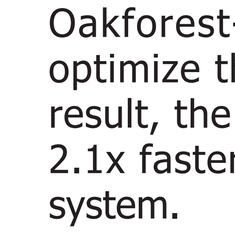
History of partial AUC of validation data where each value shows maximum in past evaluations (Left)
Horizontal axis indicates the number of evaluations divided by the number of workers (Right)

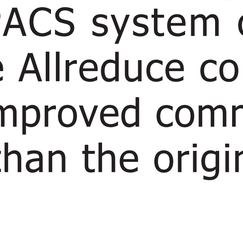
Performance Improvement of Deep Learning Training on Oakforest-PACS

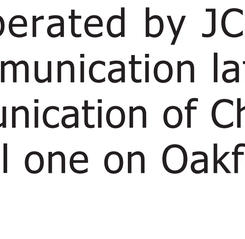
To utilize deep learning training on large-scale many core cluster, we conduct performance evaluation of large-scale deep learning framework ChainerMN on Oakforest-PACS system operated by JCAHPC, and optimize the Allreduce communication latency. As a result, the improved communication of ChainerMN is 2.1x faster than the original one on Oakforest-PACS system.

Process of Synchronous Data-parallel Deep Learning

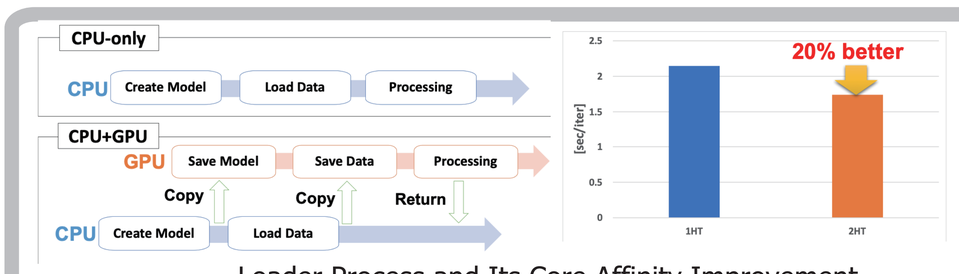
The four steps of synchronous data-parallel deep learning, which is the standard method of parallelism, is illustrated. In All-Reduce step, workers communicate with each other to find the average of gradients. Each worker optimizes the model by the average of gradients.





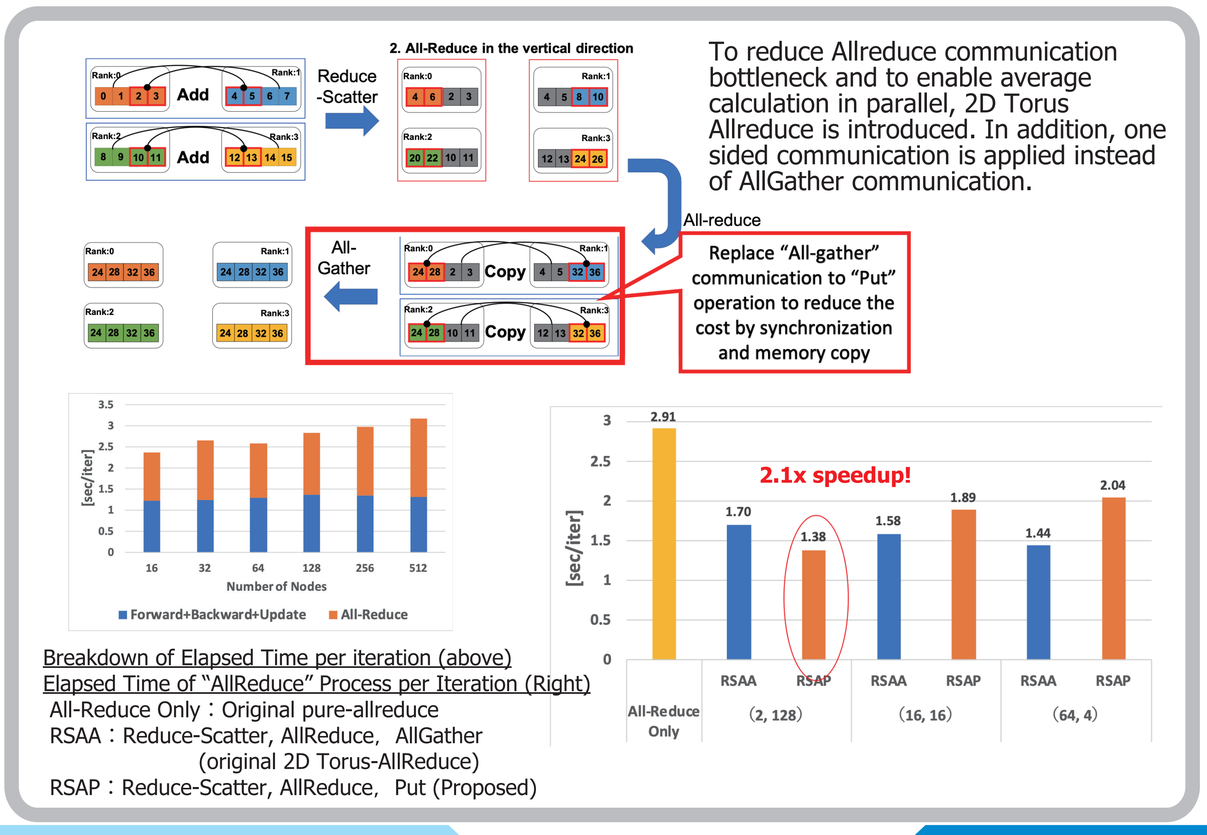


Software	Version
Intel Python	3.6.3
Intel MPI	2018.1.163
MPI4py	3.0.0
Chainer	5.0.0
iDeep4py	2.0.0



Loader Process and Its Core Affinity Improvement

CPU+GPU: GPU computes asynchronously and CPU is dedicated for data load.
CPU-only: CPU have to manage both of load and compute. HyperThreading can help to improve performance on KNL like CPU+GPU manner.



Breakdown of Elapsed Time per iteration (above)
Elapsed Time of "AllReduce" Process per Iteration (Right)
All-Reduce Only : Original pure-allreduce
RSAA : Reduce-Scatter, AllReduce, AllGather
(original 2D Torus-AllReduce)
RSAP : Reduce-Scatter, AllReduce, Put (Proposed)

Acknowledgment: This works are supported in part by "Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures" and "High Performance Computing Infrastructure" in Japan (jh170036, jh180073, jh190047-DAH).
References: [1] Y. Nomura, T. Hanawa, et al., "Development of training environment for deep learning with medical images on supercomputer system based on asynchronous parallel Bayesian optimization," ISC' 19 poster
[2] Y. Nomura, et al., "Preliminary development of training environment for deep learning on supercomputer system," Intl. J. CARS 2018;13(Suppl.1):S105-S106.