様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　６ 年　 ６ 月 １４ 日現在

機関番号：８２６２６

研究種目：若手研究

研究期間：2021～2023

課題番号：２１Ｋ１７７５１

研究課題名（和文）Scalable Hybrid-parallelism Design for Mega-Size Deep Learning Model

研究課題名（英文）Scalable Hybrid-parallelism Design for Mega-Size Deep Learning Model

研究代表者

Nguyen Truong (Nguyen, TRUONG)

国立研究開発法人産業技術総合研究所・情報・人間工学領域・研究員

研究者番号：６０８３５３４６

交付決定額（研究期間全体）：（直接経費）　　3,600,000円

研究成果の概要（和文）：大規模なデータセットを使用した大規模なディープラーニングのトレーニングでは、3D 並列処理 (データ + パイプライン + テンソル) が標準になることがわかりました。このトレーニング プロセスを高速化する方法を提案しました。I/O 時間を短縮するために、ローカル シャッフルとペアワイズデータ交換およびモデル交換を使用して、モデルの精度を維持します。計算時間を短縮するために、トレーニング中に重要でないサンプルを削除します。ネットワークアーキテクチャと集団通信を共同設計することで、通信時間を短縮します。論文8件、ポスター2件を発表し、賞を2つ獲得しました。

研究成果の学術的意義や社会的意義
Our research helps to support the research and development of big models. It brings a groundbreaking new solution with the requirements of the urgent AI, e.g.,ChatGPT. It can be ultimately contributing to the advancement of AI models, particularly foundational models, in the context of social 5.0.

研究成果の概要（英文）：We deal with memory capacity limitation when training a large model by separating the model into multiple smaller parts (published a Q1 journal-TNSM23). We also found that 3D parallelism (data+pipeline+tensor) becomes standard in training large-scale Deep Learning with large datasets. We proposed the methods to speed up this training process. To reduce the I/O time, we use local shuffling along with a pair-wise data exchanging and a model exchanging to maintain the accuracy of the model. We published 3 papers (IPDPS22a, CCGRID23, CANDAR23), a poster (HPCAsia24), and achieved 2 best paper awards. To reduce the computing time, we eliminate to process the non-important samples during the training (published at a A* conference - Neurips23). We reduce the communication time by co-design network architecture and collective communication. We published 2 rank A paper (IPDPS22b, CCGRID24), a Q1 journal (JPDC23) and a poster (HPCAsia23).

研究分野：High performance computing

キーワード：Distributed Training　 Large Model　 Large dataset　Large scale system

※科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属します。

１．研究開始当初の背景

Deep Learning (DL) proved their effectiveness in a variety of science and engineering applications such as language processing, speech, and visual recognition. Trends in DL show an increase in training dataset sizes as well as the introduction of bigger/deeper models to improve accuracy. In addition, applying DL in new domains, such as health care and scientific simulations, introduces bigger datasets and more complex DNN models. Those trends make the DL processing computationally expensive for a single compute node. Therefore, large-scale parallel training/inference on High-Performance Computing (HPC) systems or clusters of GPUs is becoming increasingly common to achieve faster training/inference time.

There are two prominent strategies for parallelizing DL training/inference: data and model parallelism. *Data parallelism* duplicates the DL model and partitions the dataset onto multiple processes, e.g., cores/GPUs. Each Processing Element (PE) performs the forward and backward propagation for its local portion of data in mini/micro batches to compute the local gradients of weights, iteratively. Next, all the PEs have to share their local gradients to obtain the averaged global gradients before the weight update phase, i.e., a collective Alleduce operation is performed. *Model parallelism* splits a DNN model across its depth or its width into composite partitions, where each composite partition is assigned into one Processing Element. All the PEs process the same input data samples (in minibatch).

To efficiently deploy a DNN model in a large-scale distributed system, one of the critical challenges are to figure out the optimal large-scale parallelization strategy. Practically, data parallelism becomes a common choice because it is simple and sufficient. However, scaling data parallelism with a larger number of computing nodes, e.g., 1000s of GPUs, can be limited by the large communication overhead for gradient exchange among computing nodes at each iteration. In addition, with the explosive increase of the size of datasets and the DL model (mega-size DL model), training/inference DL has to deal with the memory capacity limits. A notable case is in the area of language processing NLP at which models are increasingly approaching O(100B) parameters (or weight), e.g., 175 billion in GPT-3. In the other hand, as the sample size (in datasets) becomes bigger, e.g., higher dimension, higher resolution images, etc., the memory capacity also limits the number of data samples (or minibatch size) that can be map into GPUs to process. Hence, restricting the scaling of data parallelism by both memory issue and communication overhead. In this context, model parallelism could be a solution, yet there exists an upper limit on the number of PEs for model parallelism when scaling. Thus, it restricts the parallelization and speed-up degree.

２．研究の目的

The objective of this research is to **enable training/inference mega-size DL models on large-scale distributed HPC systems in the magnitude of days**. We aim at figuring out the optimal large-scale parallelism strategies when deploying a given DL model on an HPC system.

３．研究の方法

To address these above challenges, we consider several research topics includes:

- (A) **Hybrid-parallelism design**: study the limitation of different parallelism strategies and find novel fine-grained hybrid parallelism strategies for each type of specific applications
- (B) **Method to reduce the training time**:
  - **(B.1)** Reduction of the I/O time by optimizing the time of data movement from storge system to the memory of the computing node.
  - **(B.2)** Reduction of the computing time by eliminating non-important samples during the training process.
  - **(B.3)** Reduction of communication time by a co-design between communication algorithm and system architecture to mitigate the communication overhead.

４．研究成果

A. Parallelism design

As mentioned, each of data and model parallelism strategies has its own limitations. In this project, we anticipate that hybrid parallelism strategies would have a central role in the scaling of DL training, especially for mega-size DL model and scientific simulations that more than often deal with high-resolution datasets. A hybrid parallelism is a combination of two (or more) strategies. The hybrid parallelism inherits the small memory requirement from model parallelism while breaking the scaling limitation so that enable training/inference with a much larger parallelization and speed-up degree. As more datasets from HPC start to be analyzed by DL frameworks, this type of hybrid parallel strategies will become more and more relevant because data parallelism will simply be not enough. Our work published in the Q1 journal (TNSM2023) investigate the combination of data parallelism with the tensor parallelism. Tensor parallelism is the common form of model parallelism which splits a DNN model across its width. The result show that this 2-D hybrid parallelism (data + tensor) is effective in term of accuracy. However, 2-D parallelism is not enough for recent big models, e.g., which has trillions of parameters. In this context, 3-D parallelism (data + pipeline + tensor) becomes the straightforward approaches. In which pipeline parallelism is another form of model parallelism where the DNN models is spitted across its depth. Figure 1 illustrates the combination of model parallelism (horizontally split the DL model) model that is implemented on top of data parallelism. (data plus model parallelism). In which, 4 GPUs are arranged into 2 groups of 2 GPUs. This hybrid strategy implements the model parallelism inside each group and data parallelism between groups.

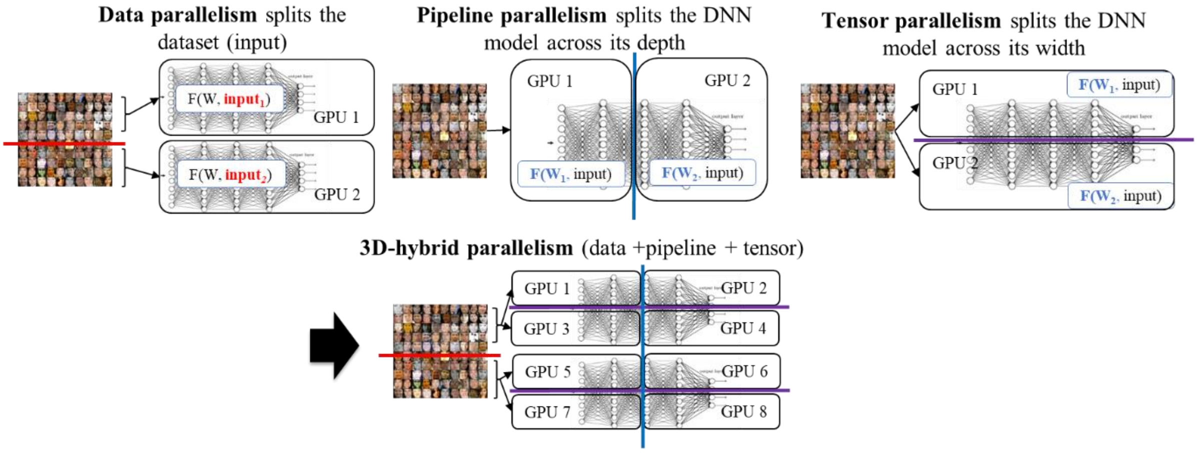

Data parallelism splits the dataset (input)
F(W, input$_1$)
GPU 1
F(W, input$_2$)
GPU 2
Pipeline parallelism splits the DNN model across its depth
GPU 1
GPU 2
F(W$_1$, input)
F(W$_2$, input)
Tensor parallelism splits the DNN model across its width
GPU 1
F(W$_1$, input)
GPU 2
F(W$_2$, input)
3D-hybrid parallelism (data +pipeline + tensor)
GPU 1
GPU 2
GPU 3
GPU 4
GPU 5
GPU 6
GPU 7
GPU 8

Figure 1: *Parallelism strategies for training a large Deep Learning model.*

In the following, we study the methods to reduce the training/inference time for this 3-D parallelism approach.

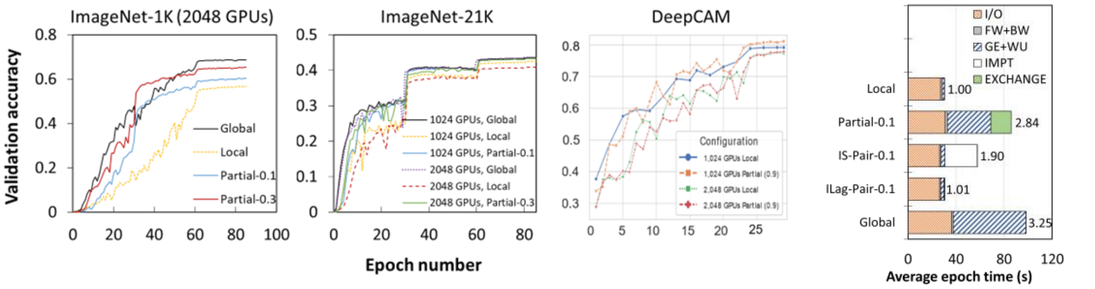
B. Method to reduce the training time.

B.1. Reduction of I/O time

Distributing the training of a neural network in a data (and hybird) parallelism fashion over compute nodes of a supercomputer requires loading the input samples on compute nodes so that each node can process a subset of the samples at each training epoch. This is either done by storing the entire dataset on computing node-local storage, or by each node reading a subset of the samples from the parallel file system (PFS). As datasets become larger, storing the entire dataset on local storage becomes impossible since they exceed storage capacities. Similarly, reading from the parallel files system puts enormous pressure on the storage nodes because many compute nodes read terabytes of data simultaneously. Moreover, to improve generalization, distributed neural network training shuffles the data at each epoch so that nodes can randomly access input samples, which further increases the I/O requirements of deep learning applications.

In this context, an alternative way is to partition the dataset among computing nodes, i.e., each node uses the same part of the dataset for all the epochs (known as local shuffling). Our work published in IPDPS2022 (rank A conference) showed that the local shuffling could not achieve similar validation accuracy as the default global shuffling strategy in large-scale training. Thus, we proposed a novel partial-local shuffling strategy that randomly exchanges only a proportion of the dataset among computing nodes in each epoch. Through extensive experiments on up to 2,048 GPUs of ABCI, the partial-local shuffling strategy then achieves similar accuracy as global shuffling while only requiring storing up to 0.03% of the whole dataset (as shown in Figure 2.a). In addition, for data sets that do not fit locally in the first place, partial-local shuffling can improve accuracy compared to the local only access. This opens the doors for leveraging the potential of locality in large scale training of large datasets and addresses the DL I/O challenge at its root: avoid I/O when possible.

However, exchanging the samples randomly between computing nodes leads to a personalized all-to-all communication pattern which is sensitive to network congestion when scaling up. In this context, we then propose an exchange strategy that is scalable. That is we avoid network congestion by managing the communication pattern in a pair-wise manner (instead of an all-to-all pattern). We pair the worker that holds the most important sample with the worker that holds the least importance of samples, and so on To reduce the overhead of computing the importance of samples in an epoch, we propose to reuse the training losses in the previous epoch (lagging loss). Our proposed Partial Shuffling has accuracy as good as the conventional global shuffling while achieving training time as fast as local shuffling (Figure 2.b). We presented this work in the HPCAsia2024 poster session.



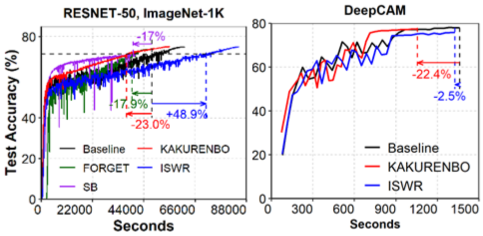
(a) *Training accuracy at large scale* (b) *Training time (512GPUs)*

Figure 2: *Example result in research B.1*

Another alternative approach is to allow exchanging the models between computing node before aggregating the model at the end of each epoch. This approach is presented in CCGRID23 paper (Best paper award Finalist) and CANDAR23 paper (Best Paper Award).

B.2. Reduction of computation time

In this work, we propose a method for hiding the least-important samples during the training of deep neural networks to increase efficiency, i.e., to reduce the cost of training. We build our hypothesis on the following observations on the effect of sample quality on training: (a) biased with-replacement sampling postulates that not all samples are of the same importance and a biased, with-replacement sampling method can lead to faster convergence and (b) data pruning methods show that when select samples are pruned away from a dataset, the predication accuracy that can be achieved by training from scratch using the pruned dataset is similar to that of the original dataset. Our hypothesis is that if samples have a varying impact on the learning process and their impact decreases as the training progresses, then we can in real-time, adaptively, exclude samples with the least impact from the dataset during neural network training.



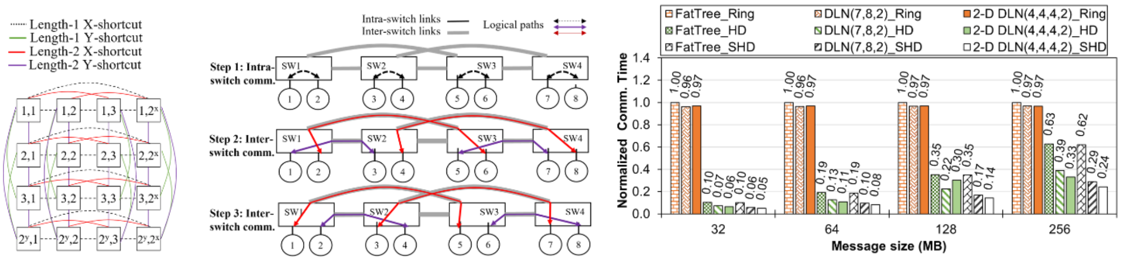
	Dataset	Metrics	Baseline	ISWR	FORGET	SB	KAKUR.
Up stream	Fractal-3K	Loss	3.26	3.671	3.27	4.18	3.59
		Time (min) Impr.	623 -	719 (+15.4%)	533 (-14.4%)	414 (-33.5%)	529 (-15.1%)
Down stream	CIFAR-10	Acc. (%) Diff.	95.03 -	95.79 (+0.76)	95.85 (+0.82)	93.59 (-1.44)	95.28 (+0.25)
	CIFAR-100	Acc. (%) Diff.	79.69 -	79.62 (-0.07)	79.95 (+0.26)	76.98 (-2.71)	79.35 (-0.34)

Figure 3: *Example result in research B.2. (LEFT) Proposed method, KAKURENBO, reduces up to 23% of total training time while maintaining the same accuracy. (RIGHT) Our method is also useful in fine tuning the big model.*

Specifically, we dynamically hide samples in a dataset to reduce the total amount of computing and the training time, while maintaining the accuracy level. Our proposal, named KAKURENBO, is built upon two pillars. First, using combined information about the loss and online estimation of the historical prediction confidence of input samples, we adaptively exclude samples that contribute the least to the overall learning process on a per-epoch basis. Second, in compensation for the decrease in the number of SGD steps, we derive a method to dynamically adjust the learning rate and the upper limit on the number of samples to hide to recover convergence rate and accuracy. We evaluate performance both in terms of reduction in wall-clock time and degradation in accuracy. Our main results are two-fold: first, we show that decaying datasets by eliminating the samples with the least contribution to learning has no notable negative impact on the accuracy and convergence and that the overhead of identifying and eliminating the least important samples is negligible. Second, we show that decaying the dataset can significantly reduce the total amount of computation needed for DNN training. We also find that state-of-the-art methods such as importance sampling algorithm, data pruning, or sample hiding techniques performs poorly on large-scale datasets. To the contrary, our method can reduce training time by 10.4% and 22.4% on ImageNet-1K and DeepCAM, respectively, impacting Top-1 accuracy only by 0.4% (as shown in Figure 3). Our work published in Neurips2023 (rank A* conference).

B.3. Reduction of the communication time

Training a Deep Learning model on High-Performance Computing systems is becoming a de-facto standard in deep learning. One of the key factors limiting the growth of large-scale training is the collective communication overhead between computing nodes or processing elements (PEs), e.g., an Allreduce operation of data parallelism and tensor parallelism. With the continual increase in model sizes, e.g., 100s GB, and the number of PEs, e.g., 1,000s of GPUs, communication becomes a major bottleneck. In this context, we aim at finding a network topology and its corresponding collective algorithm that features bandwidth optimality in O(log(P)) steps with a minimal (or without) network contention. We proposed the use of a family of network topologies that exploit small-world network models, e.g, Distributed Loop Network (DLN) and collective algorithms named Shifted Halving-Doubling (SHD) which improve the utilization of all the inter-switches links of the DLN topology. We then generalize the SHD algorithm and propose 2-D DLN which considers both the communication performance and the cost when implemented in a server room. We published this work in HPCAsia2023.



(a) Proposed 2D-DLN network topology

(b) Illustration of our Shifted Halving-Doubling Allreduce Algorithm.

(b) Normalized communication time of different [network]_[allreduce algorithm] to those of FatTree_Ring with 512 processing elements, e.g., GPUs.

Figure 4: *Example result in research B.3.*

We also optimize the communication inside one computing node. We propose to use the Kautz network for inter-memory network using switchless OPTWEB FPGA and multi-port collective communications to mitigate the influence of the startup latency on the execution time. Based on our experimental results with OPTWEB of custom Stratix10 FPGA cards, SimGrid simulation results show that our collective communication is 7x faster than that of Dragonfly with 272 FPGAs. Our works are published in two conference papers (IPDPS2022, CCGRID2024) and a Q1 journal (JPDC2023).

５．主な発表論文等

〔雑誌論文〕　計10件（うち査読付論文　8件／うち国際共著　10件／うちオープンアクセス　2件）

１．著者名 Nguyen Quan、Pham Hieu H.、Wong Kok-Seng、Le Nguyen Phi、Nguyen Truong Thao、Do Minh N.	４．巻 21
２．論文標題 FedDCT: Federated Learning of Large Convolutional Neural Networks on Resource-Constrained Devices Using Divide and Collaborative Training	５．発行年 2024年
３．雑誌名 IEEE Transactions on Network and Service Management	６．最初と最後の頁 418～436
掲載論文のDOI（デジタルオブジェクト識別子） 10.1109/TNSM.2023.3314066	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

１．著者名 Pham Quang Ha、Nguyen Nang Hung、Nguyen Thanh Hung、Pham Huy Hieu、Nguyen Phi Le、Nguyen Truong Thao	４．巻 -
２．論文標題 SEM: A Simple Yet Efficient Model-agnostic Local Training Mechanism to Tackle Data Sparsity and Scarcity in Federated Learning	５．発行年 2023年
３．雑誌名 Eleventh International Symposium on Computing and Networking (CANDAR)	６．最初と最後の頁 120-126
掲載論文のDOI（デジタルオブジェクト識別子） 10.1109/CANDAR60563.2023.00023	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

１．著者名 Truong Thao Nguyen, Balazs Gerofi, Edgar Josafat Martinez-Noriega, Francois Trahay, and Mohamed Wahib	４．巻 -
２．論文標題 KAKURENBO: Adaptively Hiding Samples in Deep Neural Network Training	５．発行年 2024年
３．雑誌名 37th Conference on Neural Information Processing Systems (NeurIPS 2023)	６．最初と最後の頁 1-23
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 該当する

１．著者名 Nguyen Quan、Pham Hieu H.、Wong Kok-Seng、Le Nguyen Phi、Nguyen Truong Thao、Do Minh N.	４．巻 21
２．論文標題 FedDCT: Federated Learning of Large Convolutional Neural Networks on Resource-Constrained Devices Using Divide and Collaborative Training	５．発行年 2024年
３．雑誌名 IEEE Transactions on Network and Service Management	６．最初と最後の頁 418～436
掲載論文のDOI（デジタルオブジェクト識別子） 10.1109/TNSM.2023.3314066	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 該当する

１．著者名 Pham Quang Ha、Nguyen Nang Hung、Nguyen Thanh Hung、Pham Huy Hieu、Nguyen Phi Le、Nguyen Truong Thao	４．巻 -
２．論文標題 SEM: A Simple Yet Efficient Model-agnostic Local Training Mechanism to Tackle Data Sparsity and Scarcity in Federated Learning	５．発行年 2023年
３．雑誌名 Eleventh International Symposium on Computing and Networking (CANDAR)	６．最初と最後の頁 120-126
掲載論文のDOI（デジタルオブジェクト識別子） 10.1109/CANDAR60563.2023.00023	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

１．著者名 Kien Trung Pham, Thao Nguyen Truong and Michihiro Koibuchi	４．巻 -
２．論文標題 A Bandwidth-Optimal All-to-All Communication in Two-Dimensional Fully Connected Network	５．発行年 2024年
３．雑誌名 24th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing	６．最初と最後の頁 1-7
掲載論文のDOI（デジタルオブジェクト識別子） 10.1109/CCGrid59990.2024.00010	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

１．著者名 Truong Thao Nguyen, Kien Trung Pham, Hiroshi Yamaguchi, Yutaka Urino, Michihiro Koibuchi	４．巻 -
２．論文標題 Effective Switchless Inter-FPGA Memory Networks	５．発行年 2023年
３．雑誌名 Journal of Parallel and Distributed Computing	６．最初と最後の頁 -
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

１．著者名 Nang Hung Nguyen, Duc Long Nguyen, Trong Bang Nguyen, Thanh-Hung Nguyen, Hieu Pham, Truong Thao Nguyen, Phi Le Nguyen	４．巻 -
２．論文標題 CADIS: Handling Cluster-skewed Non-IID Data in Federated Learning with Clustered Aggregation and Knowledge Distilled Regularization	５．発行年 2023年
３．雑誌名 23rd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing	６．最初と最後の頁 249-261
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1．著者名 Truong Thao Nguyen, Francois Trahay, Jens Domke, Aleksandr Drozd, Emil Vatai, Jianwei Liao, Mohamed Wahib, Balazs Gerofi	4．巻 0
2．論文標題 Why Globally Re-shuffle? Revisiting Data Shuffling in Large Scale Deep Learning	5．発行年 2022年
3．雑誌名 36th IEEE International Parallel & Distributed Processing Symposium	6．最初と最後の頁 1-12
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1．著者名 Kien Trung Pham, Truong Thao Nguyen, Hiroshi Yamaguchi, Yutaka Urino, Michihiro Koibuchi	4．巻 0
2．論文標題 Scalable Low-Latency Inter-FPGA Networks	5．発行年 2022年
3．雑誌名 36th IEEE International Parallel & Distributed Processing Symposium	6．最初と最後の頁 1-12
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

〔学会発表〕　計2件（うち招待講演　0件／うち国際学会　2件）

1．発表者名 Truong Thao Nguyen, Yusuke Tanimura
2．発表標題 Efficient Sample Exchanging for Large-Scale Training Distributed Deep Learning with Local Sampling
3．学会等名 International Conference on High Performance Computing in Asia-Pacific Region 2024（国際学会）
4．発表年 2024年

1．発表者名 Truong Thao Nguyen, Peng Chen, Yusuke Tanimura
2．発表標題 Efficient Allreduce Algorithm for Large-Scale Deep Learning on Distributed Loop Networks
3．学会等名 International Conference on High Performance Computing in Asia-Pacific Region 2023（国際学会）
4．発表年 2023年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

We achieved 2 paper awards including CCGRID23-Best Paper Finalists Award (https://ccgrid2023.iisc.ac.in/awards/) and CANDAR23-Best paper Award (https://is-candar.org/candar23/best_outstanding_papers)

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考
研究協力者	ゲロフィ　バラス (GEROFI BALAZS)	国立研究開発法人理化学研究所・計算科学研究センター・上級研究員	He used ABCI points (managed by PI) for experiments of DeepCAM models/dataset in our IPDPS22 paper.

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関