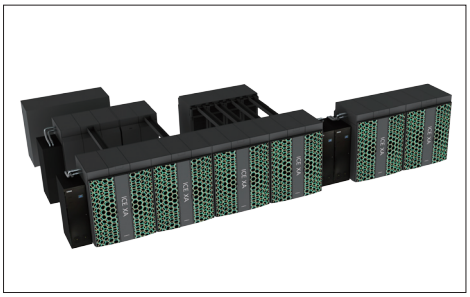
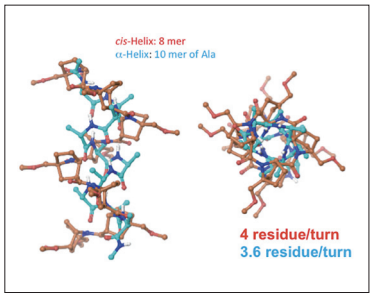
e-Science Journal
Global Scientific Information and Computing Center

Vol. 16

TSUBAME ESJ.





Overview of TSUBAME3.0, Green Cloud Supercomputer for Convergence of HPC, AI and Big-Data

ChainerMN: Scalable Distributed Deep Learning Framework

Creation of Chemical Structures Aimed for Drug Design, Facilitated by Efficient GPU Computing

http://www.gsic.titech.ac.jp/TSUBAME_ESJ

Overview of TSUBAME3.0, Green Cloud Supercomputer for Convergence of HPC, AI and Big-Data

Satoshi Matsuoka[1] Toshio Endo[1] Akira Nukada[1] Shinichi Miura[1] Akihiro Nomura[1]
Hitoshi Sato[2] Hideyuki Jitsumoto[1] Aleksandr Drozd[1]
1 Global Scientific Information and Computing Center, Tokyo Institute of Technology
2 Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology

TSUBAME3.0 is one of largest supercomputers in Japan, which enjoys 47.2PFlops performance in half-precision. It drives broad range workloads, not only in traditional HPC area, but also in big-data and AI. It is expected to achieve theoretical PUE of 1.033, as results of our long-time efforts in improvement of density and energy efficiency. This report introduces architecture of TSUBAME3.0, while mentioning issues appeared in previous TSUBAME systems and new operation strategies as their solutions.

Introduction 1

Global Scientific Information and Computing center (GSIC) at Tokyo Institute of Technology is operating the TSUBAME supercomputer series as "Everybody's supercomputer", which focus on high performance and easy-to-use. Since TSUBAME1.0 introduced in 2006, TSUBAME has been harnessed by around 2,000 users from Tokyo Tech, other research institutes, and industrial area. After partial adoption of GPU accelerators in TSUBAME1.2, GPUs are fully equipped in TSUBAME2.0 introduced in 2010. Owing to higher energy efficiency largely, TSUBAME2.0 achieved 30 times higher speed performance than TSUBAME1.0 within a similar power budget[1]. TSUBAME2.0 has ranked as No.4 in the Top500 supercomputer ranking and awarded "the Greenest Production Supercomputer in the World". From the peta-scale application aspect, 2PFlops performance has been observed in dendritic solidification simulation, which resulted in winning the ACM Gordon Bell Prize[2]. In 2013, all the GPUs in TSUBAME2.0 have been replaced to K20X GPUs, and the resultant system is called TSUBAME2.5, whose peak performance is 5.6PFlops (double precision).

GSIC has started the operation of the brand-new system in this series, called TSUBAME3.0, in August 2017. The proposition submitted by SGI Japan has been adopted in January; since then, GSIC, SGI Japan and related vendors have cooperatively promoted preparations towards the operation start. Figure 1 is an illustration of the entire TSUBAME3.0 system.

The target applications area of TSUBAME3.0 is not limited to typical HPC area, but includes big-data and artificial intelligence (AI) area. The main components of the system are 540 compute nodes, which are customized version of SGI ICE XA. The total numbers of CPUs and GPUs are 1,080 and 2,160, respectively, and the total performance is 12.15PFlops in double precision and 47.2PFlops in half precision (or more). Half precision is an expression of floating point values in 16bit, which is expected to be useful in big-data and AI area, and implemented by hardware in GPU accelerators adopted in TSUBAME3.0. As another feature, each compute node has a high-speed and large capacity (2TB) NVMe SSD. The total capacity of SSDs are 1.08PB. Also shared large-scale storage is more powerful than ever; the total capacity is 15.9PB and transfer speed is 150GB/s. This storage hierarchy can be harnessed for extremely high-speed big-data analysis.

TSUBAME3.0 has excellent power efficiency, largely owing to GPU accelerators of the latest generation. We have conducted performance measurement of the HPL benchmark with 144 compute nodes. When the parameters are optimized for better power efficiency, the performance of 1.998PFlops has been achieved with power consumption of 141.6kW. The power-performance ratio, 14.11GFlops, won world No.1 in the Green500 List in June 2017[3].

The cooling system of TSUBAME3.0 is also highly optimized for energy saving. We have used and analyzed indirect liquid cooling in TSUBAME2.0/2.5 and liquid submersion cooling in TSUBAME-KFC prototype supercomputer. Based on such knowledge, TSUBAME3.0 cooling system consists of warm water cooling for CPUs and GPUs, the main heat sources, and indirect liquid cooling for the rest parts. This achieves high power-efficiency, stability and maintainability.

This report overviews the TSUBAME3.0 system, including its cooling system and facility, and describes the operation strategy.

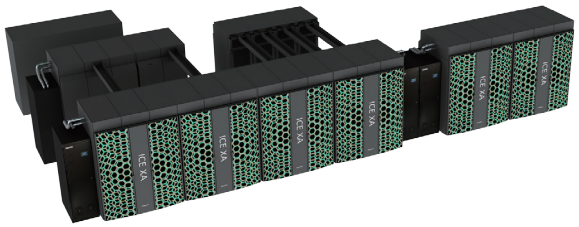


Fig. 1 Illustration of TSUBAME3.0 system

TSUBAME3.0 Architecture 2

2.1 Computing Nodes

TSUBAME3.0 includes 540 uniform compute nodes, each of which consists of the latest CPUs, GPU accelerators, large main memory, high-speed interconnect and fast SSDs. Two Intel Xeon E5-2680 V4 Processor (Broadwell-EP, 14 cores, 2.4GHz) are adopted as CPUs, and main memory capacity is 256GiB, which composed of eight DDR4-2400 ECC REG DIMM 32GB modules, and bandwidth of main memory is 154GB/s.

As GPU accelerators, four NVIDIA Tesla P100 for NVLink-Optimized Servers (16GB HBM2@732GB/s, 5.3TFLOPS@FP64, 10.6TFLOPS@FP32, 21.2TFLOPS@FP16) are equipped. They are fourth generation GPUs in the TSUBAME series, which succeed Tesla S1070 (GT200) in TSUBAME1.2, Tesla M2050 (Fermi) in TSUBAME2.0 and Tesla K20X (Kepler) in TSUBAME2.5. Each node is also equipped with four Intel Omni-Path Architecture based Host Fabric Interfaces(HFI). The injection bandwidth is 400Gbps in uni-direction and 800Gbps in bi-direction. As local storage, A NVMe-based high-speed SSD with 2TB capacity per node is used. High-speed interconnects and SSDs are especially important for big-data/AI usages.

Fig. 2 is a block diagram of a TSUBAME3.0 compute node. The node has the almost symmetric structure with even numbers of CPUs, GPUs and HFIs. In the figure, PCIe represents PCI-Express Gen3, whose transfer speed is 1GB/s per lane. A Broadwell-EP CPU has 40 PCI-Express lanes. Among of them, 16 lanes are connected to one of Omni-Path HFIs. Other 16 lanes are connected to a PLX PCI-Express switch, which is connected with two GPUs and another Omni-Path HFI. The rest lanes of CPU1 are connected to an NVMe SSD.

GPU accelerators on TSUBAME3.0 are Tesla P100 with SXM2 form factor. Each GPU has four 20GB/s data link, called NVLink, which are used for direct data transfer among GPUs.

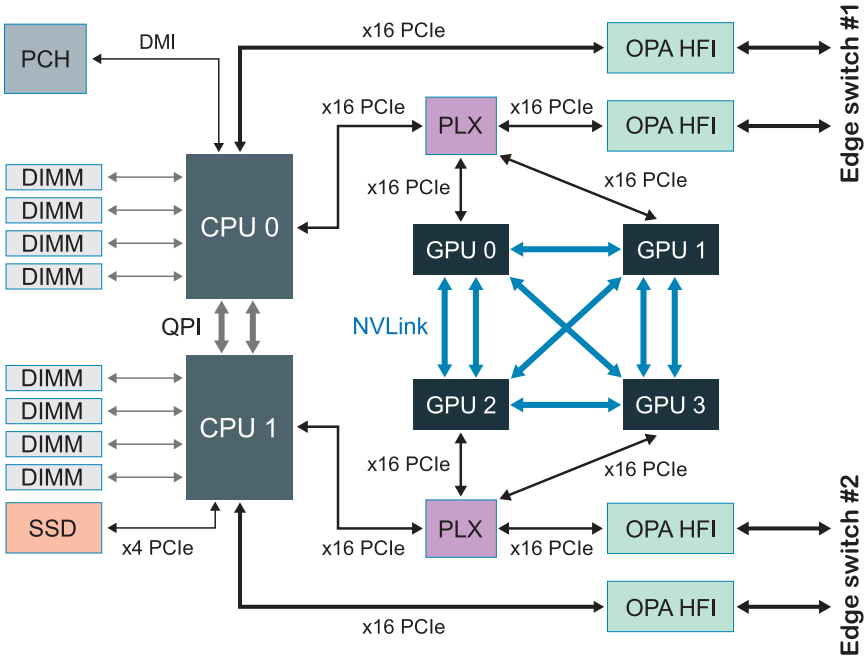


Fig. 2 Block diagram of a TSUBAME3.0 compute node

Table 1 shows comparison between a TSUBAME2.5 node and a TSUBAME3.0 node. The numbers shown are theoretical peak ones. The computation speed of GPUs is largely improved on TSUBAME3.0, whereas GPUs on TSUBAME2.5 have been upgraded in 2013. Especially the half precision (FP16) is highly improved. Also capacity and bandwidth of memory and the SSD are largely enhanced toward big-data applications.

Measure	T2.5	T3.0	Ratio
CPU			
#cores×frequency (GHz)	35.16	72.8	2.07
Memory capacity (GiB)	54	256	4.74
Memory bandwidth (GB/s)	64	153.6	2.40
GPU			
#CUDA cores	8064	14,336	1.78
FP64 (TFlops)	3.93	21.2	5.39
FP32 (TFlops)	11.85	42.4	3.58
FP16 (TFlops)	11.85	84.8	7.16
Memory capacity (GiB)	18	64	3.56
SSD			
Capacity(GB)	120	2,000	16.67
READ (MB/s)	550	2,700	4.91
WRITE (MB/s)	500	1,800	3.60
Network			
Transfer speed (Gbps)	80	400	5.00

Table 1 Comparison between a TSUBAME2.5 node and a TSUBAME3.0 node. The aggregated number is shown within a node.

2.2 Interconnect

All compute nodes, storages and management nodes are connected via interconnect based on the Intel Omni-Path Architecture. As the more important characteristics of the interconnect, compute nodes are connected in a fat-tree topology with full bisection bandwidth. The tree topology consists of director switches, edge switches and compute nodes as leaves. Each chassis contains nine compute nodes and two edge switches. Each edge switch has 18 downlinks connected to nine nodes, and 18 uplinks connected to three director switches.

The design of the entire interconnect topology on supercomputer has a heavy impact especially on performance of large scale computations. TSUBAME3.0 adopts the full bisection fat tree, after its success in TSUBAME2.0. On this topology, all communication passes between arbitrary ports can be established without contention theoretically. Since communication in a certain job does not affect performance of other jobs, this topology is suitable to supercomputers in universities, which accommodate applications with wide variety in the scale and communication patterns. While the basic topology is common between TSUBAME2.0/3.5 and TSUBAME3.0, the former has used the static routing method with a fixed routing table, thus we have observed performance degradation caused by network contentions in certain communication patterns [4]. On the other hand, TSUBAME3.0, with the adaptive routing method that dynamically finds communication paths, such degradations are expected to be solved.

2.3 Storage

2.3.1 Global Storage Area

Shared storages with peta-scale capacity and high access speed are essential components in supercomputers. On TSUBAME2.5, a global storage system that is equally accessible from all compute nodes has largely contributed to convenience in the usage of the supercomputer. TSUBAME3.0 inherits this direction, while capacity and access speed are largely enhanced. Considering flexibility in the operation and localization of effects of failures, the TSUBAME3.0 global storage, shown in Fig.3, consists of three systems, based on Lustre parallel file system. Each system has 5.3PB effective capacity and the total capacity is 15.9PB.

Fig. 3 The global storage of TSUBAME3.0

2.3.2 Scratch Area

Modern supercomputers should support extremely frequent file I/O requests from compute nodes. It is especially important for recent workloads including big-data applications. Global file systems are often designed to optimize I/O performance for files with large sizes, however, some workloads including machine learning may include frequent I/O requests to a large number of small files. In such cases, parallel file systems like Lustre tend to suffer from bottleneck in meta data servers (MDS), which prevents performance improvement.
One of solutions is to introduce a Flash-based "Burst Buffer" system, which works as a shared cache system of global file system. In the design of TSUBAME3.0, we did not adopt this approach due to issues in cost performance of current Burst Buffers.

Instead, we provide local Flash SSDs attached to compute nodes for highly frequent file I/O. This approach is basically similar to that in TSUBAME2.5, however, it has been largely enhanced in the following aspects. First, both of the capacity and access speed are highly improved as shown in Table 1. Secondly, TSUBAME3.0 support temporary parallel file systems as follows. Since each SSD is equipped to one of compute nodes, it is only visible from the node without any special treatments. This situation is inconvenient for multi-node jobs. In order to improve usability, we have adopted a technology called BeeOND [6], which bundles several SSDs into a temporary parallel file system based on BeeGFS [5]. The created file system is alive during execution of a multi-node job; thus it can be used as a shared scratch area.

Preparation of Facility 3

In the design phase of TSUBAME3.0, our purposes included achievement of high performance system with high density and high power efficiency. For this purpose, we have required preparation of basic facility, since we noticed that the building of GSIC, which is more than 40 years old, was weak as a modern data center. Also we considered the possibility of parallel operation of TSUBAME3.0 and TSUBAME2.5, which presented other challenges in the floor space and the electric power limitation. In order to solve the above issues, we have conducted repair work of the GSIC building. Moreover, in order to reduce operation costs, the cooling system of TSUBAME3.0 is highly power efficient at the world's top level.

3.1 The Server Room

Modern supercomputers have tended to increase the number of nodes for performance improvement, which requires larger footprint area. On the other hand, TSUBAME series have introduced high density system design to reduce footprint. Especially, the replacement this time is faced with stricter condition since both TSUBAME3.0 and TSUBAME2.5 will be settled in the building. Not only for reducing footprint, higher density is also important for improvement of power/cooling efficiency, which has been and will be important in designing supercomputers. In TSUBAME3.0, we have planned to adopt direct liquid cooling as described later. While this method has an advantage in energy saving, it tends to increase the weights of racks, due to liquid pipes and coolant liquid itself. From the above discussion, we have considered that the weight per rack could be close to 1t/m^2. Since there was no room in the GSIC building that tolerates such a tough condition, we conducted repair work for the new machine room of TSUBAME3.0 as follows. The space has been originally used for TSUBAME2.5 storage system.

- To accommodate TSUBAME3.0 system and future possible extensions, the floor loading capacity of the entire room is 100t.
- To improve the floor loading capacity and reduce the system costs, the slab floor is adopted, instead of free access (raised) floor. Cables and pipes are put over the racks.
- To widen the space for cabling and piping, the original ceiling is removed.

By doing these, we have prepared the new server room of around 145m^2 area, as shown in Fig.4. As a result of the preparation, the room can accommodate the rack design of with high density, and the required area for the entire TSUBAME3.0 system including storages is less than that of TSUBAME2.5 as illustrated in Fig.5. There is also sufficient room for future possible system extension.

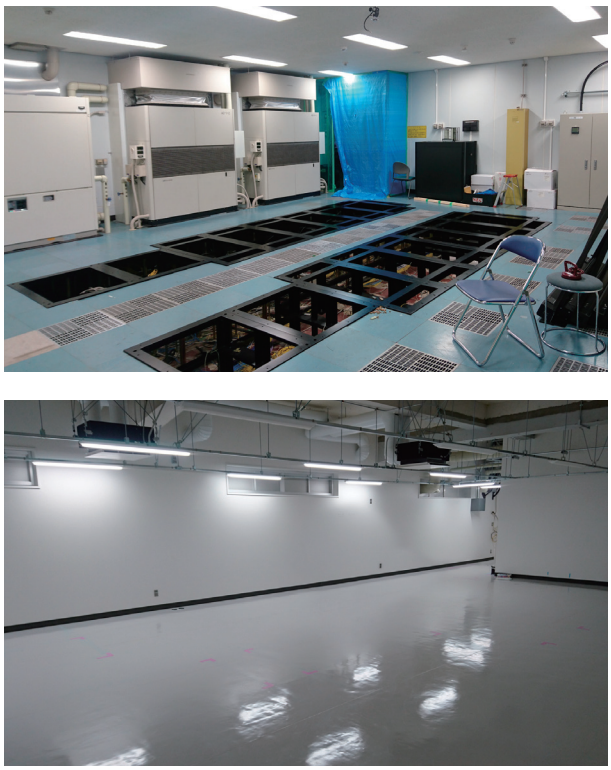


Fig. 4 TSUBAME3.0 server room
(Upper: before construction,
Lower: after construction)

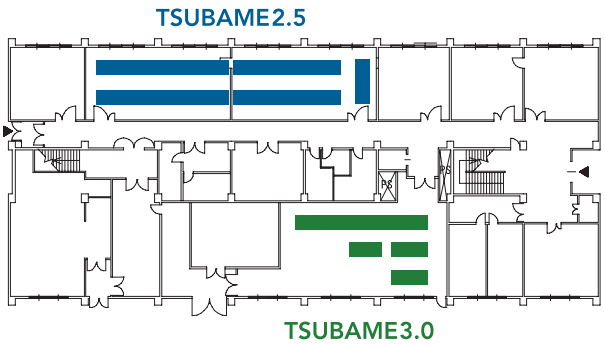


Fig. 5 Placement of TSUBAME3.0 and TSUBAME2.5

3.2 Power Supply Facility

As one of energy saving methods of supercomputer systems, we have considered to introduce power supply facility with higher voltage than 200V used in TSUBAME2. The candidates included 3-phase 4-wire 415V and 3-phase 3-wire 480V. Using the higher voltage generally reduces energy loss in wires and wiring costs. On the other hand, the new power facility should have generality to support commodity servers for management, network switches and so on. For this purpose, we adopted 3-phase 4-wire 415V, since we can get one-phase 240V connection from it easily, which supports almost IT equipment and reduces procurement costs. Based on above considerations including costs for new trances, discussions with the facility division of Tokyo Tech, and we have prepared power 3-phase 4-wire supply facility with 2MW capacity. The average power consumption of TSUBAME3.0 is less than 1MW finally, thus there is room for future extensions both in space and electric capacity.

3.3 Cooling Facility

In order to reduce operation costs of supercomputers, we need to reduce energy consumption. The reduction is important not only for computing equipment, but for the cooling facility. For example, even in TSUBAME2 with optimized cooling system, 22% electric power of the entire system is consumed in the cooling system. The electric fee of TSUBAME2 exceeds $1M per year, thus more than $200K is paid for cooling.

To make cooling systems energy efficient, GSIC has promoted research including operation and evaluation of air/liquid hybrid cooling introduced in TSUBAME2 and liquid submersion cooling in TSUBAME-KFC[7].
Through these studies, we have learned that hybrid cooling using air requires coolant water with low temperature (10℃ or less) for indirect cooling, thus it needs chillers with compressors, which becomes the major source of power consumption. This issue is mostly solved by direct cooling as in TSUBAME-KFC, since the coolant liquid can be in high temperature (30℃ or higher). On the other hand, there are other issues on maintenance costs and racking space in liquid submersion cooling.

From the above discussions, TSUBAME3.0 uses direct liquid cooling without intervention of air for CPUs and GPUs, which are major heat sources. For this purpose, water pipes go inside each node, and the water is cooled by cooling towers on the rooftop of GSIC building. This method

is expected to achieve free cooling without compressors throughout the year. The expected average PUE (power usage efficiency) is estimated to be 1.033, which means the power consumption of cooling system is around 3% of the whole system, and helps energy saving tremendously.

Operation 4

The one of major focuses in operation of TSUBAME series has been usability to broaden the horizons of users, including who are using PCs or small scale clusters for their computations. TSUBAME3.0 introduces several new services to improve the high usability.

4.1 Dynamic Node Partitioning based on Container Technologies

On TSUBAME, there are large number of running jobs with wide variety in the aspect of computing resources; some jobs use only CPUs, while other mainly use GPUs. The number of used processors and nodes are also different. To accommodate these jobs efficiently, node partitioning is a good approach. In TSUBAME2.0/2.5, each of 400 nodes have been partitioned into "CPU part" and "GPU part" by using VM technology. However, since the partition method and node set to be divided have been static, we have observed unused computing resources. As another issue, since VM technology is not so matured when the operation of TSUBAME2.0 has been started, three GPUs on a single node cannot be partitioned.

TSUBAME3.0 introduces more flexible and dynamic partitioning method based on modern resource grouping and container technologies, Docker. Each partition can have direct access to GPUs and Omni-Path HFIs efficiently. Fig. 6 shows an example of flexible node partition. Since the partitions can be created shortly, they are dynamically invoked for each submitted job by cooperation with the job scheduler. With this method, static partitioning in TSUBAME2 is not needed, and any nodes can be partitioned, which leads to efficient usage of computing resources in the entire system.

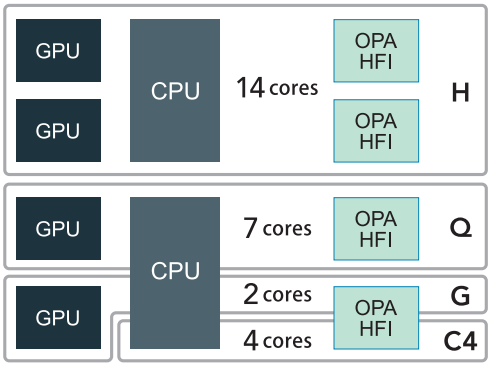


Fig. 6 An example of node resource partition on TSUBAME3.0

4.2 Efficient Resource Usage

In the operation of TSUBAME2.0/2.5, GSIC has made several improvements in operation. For example, we have modified operation policy of TSUBAME2 in order to make the backfilling mechanism of job scheduler more efficient [8]. Here the usage fee of a job not only depends on the consumed time but on the specified limitation time. TSUBAME3.0 adopts this method for better resource utilization.

On the other hand, there still remained several issues that may degrade resource usage; one of them is the static node partitioning described in the previous section.

Another issue comes from node reservation system. In addition to the typical usage via job scheduler, users can reserve nodes during the specific dates beforehand. In TSUBAME2, the granularity of date specification has been one day (10:00 to 9:00 next day). In TSUBAME3, the granularity will be one hour, thus users feel free to use the node reservations (As of writing this manuscript, the reservation system is under construction).

Moreover, in TSUBAME2, some special nodes are used for node partition and other some nodes are used for node reservation. On the other hand, there is no specific role assigned to TSUBAME3.0 nodes; all nodes can be used for any purposes. Thus the system is flexible to the changes of system load and job mixtures in order to keep better resource usage.

4.3 Providing System Image per Job with Containers

TSUBAME2.0/2.5 has been operated for more than six years, which is longer than the previous plan. On such systems with long lives, some users become aware of staleness of OS kernels, standard libraries, and so on. While it was mitigated by the OS upgrade to SUSE Enterprise Linux 11 SP3 in 2013,

we still observed difficulties in installation new applications especially in AI area, thus some users download and build such software including standard libraries [9]. This situation with staleness is also undesirable from the aspect of security, while TSUBAME nodes use up-to-date security patches. This issue mainly comes from backward incompatibility in update of OS and/or standard libraries. OS update also raises heavy costs in verification tests of all running ISV applications.

TSUBAME3.0 harnesses the container technology, Docker, in order to divide the system image per node and the bare-metal image (As of writing this manuscript, this mechanism is under construction). With this mechanism, users can choose system images for their usages, such as an image with the latest libraries, an image where a specific ISV application has been verified, and so on. This mechanism will be useful to reproduce the past experimental results that depend on libraries of the previous version. From the aspect of security, it is difficult to support system images created by end-users directly. Instead, GSIC will provide several useful images to users.

4.4 External Network Connections

In TSUBAME2, computing nodes are basically isolated from Internets. Thus users send their programs and data from outside via interactive nodes. TSUBAME3.0 computing nodes are selectively allowed to communicate to the external network. The connection is 100Gbps via SINET5, thus there is flexibility in using external storage or computing services.

By harnessing this high network bandwidth, we are planning to make partial global storage directly reachable from external network via NFS or CIFS.

4.5 Releasing Monitoring Information

We make monitoring information of TSUBAME3.0 open via the web, as done in TSUBAME2 [10]. Users can view real-time information on node usage, storage usage, the scheduler and power consumption, etc. Also information on system troubles are available. Currently the trouble information is maintained by operators as CSV files internally, however, there is an issue in statistical analyses. We are planning to make all information machine-readable and more structured as open data. Additionally, we are preparing to provide information on each job to its owner user, which would be useful for users to improve their applications.

4.6 Paperless Account Application and External Usages

Users from Tokyo Tech can use paperless account application system from the "Tokyo Tech portal" web page as in TSUBAME2. Also external users can apply to TSUBAME3.0 usage via several systems. One of them is HPCI, which is an infrastructure to connect major supercomputer systems in Japan.

Technically, a single TSUBAME2.5 user might have several user accounts with several attributes. On TSUBAME3.0, those accounts are unified per user, which would help the usability.

4.7 Grand Challenge System

TSUBAME3.0 also inherits the "grand challenge" from TSUBAME2, which provides extremely large computing resources to a few groups, selected via peer-reviews [11]. On TSUBAME2, 20 groups have used the whole computing resources for 12 to 24 hours, and 14 groups have used 1/3 computing nodes for one week. TSUBAME3.0 also provides this system towards new computing challenges.

Summary 5

This report described the overview of TSUBAME3.0, which was started its operation on August 2017. TSUBAME3.0 inherits number of advantages of TSUBAME series, including high usability and power efficiency. Moreover, it adopts new features not only for traditional HPC applications but big-data/AI area.

Acknowledgment

The design of TSUBAME3.0 is based on research projects by GSIC, including MEXT "Ultra Green Supercomputing and Cloud Infrastructure Technology Advancement", "Energy Optimizations of Supercomputing and Cloud Infrastructure for Achievement of Smart Community", JST CREST (JPMJCR1303, JPMJCR1501).

References

[1] Satoshi Matsuoka, Toshio Endo, Naoya Maruyama, Hitoshi Sato, Shin'ichiro Takizawa. The Total Picture of TSUBAME 2.0. GSIC, Tokyo Tech, TSUBAME e-Science

Journal, No.1, pp. 2—4 (2010)
[2] T. Shimokawabe, T. Aoki, T. Takaki, A. Yamanaka, A. Nukada, T. Endo, N. Maruyama, S. Matsuoka. Peta-scale Phase-Field Simulation for Dendritic Solidification on the TSUBAME 2.0 Supercomputer. IEEE/ACM SC11, 11pages (2011)
[3] TOP500 Supercomputer Sites: Green500 List for June 2017, https://www.top500.org/green500/lists/2017/06/
[4] Akihiro Nomura, Toshio Endo, Satoshi Matsuoka. Performance Evaluation of Multi-rail InfiniBand Network on TSUBAME2.0 (in Japanese). IPSJ SIG report, Vol. 2012-HPC-137, No. 3, pp. 1-5 (2012)
[5] Fraunhofer Center: BeeGFS - The Leading Parallel Cluster File System, https://www.beegfs.io/
[6] Fraunhofer Center: BeeOND: BeeGFS On Demand, https://www.beegfs.io/wiki/BeeOND
[7] T. Endo, A. Nukada, S. Matsuoka. TSUBAME-KFC: a Modern Liquid Submersion Cooling Prototype towards Exascale Becoming the Greenest Supercomputer in the World . IEEE ICPADS 2014, pp.360-367 (2014)
[8] Akihiro Nomura, Atsushi Sasaki, Shin'ichi Miura, Toshio Endo, Satoshi Matsuoka. Improvement of Scheduling Efficient on TSUBAME2 and Visualization of Users Behavior (in Japanese). IPSJ SIG report, Vol. 2015-HPC-150, No. 2, pp. 1-7 (2015)
[9] GSIC, Tokyo Tech. Experimental Services on TSUBAME2.5, http://tsubame.gsic.titech.ac.jp/labs
[10] GSIC, Tokyo Tech. TSUBAME2.5 Monitoring Portal, http://mon.g.gsic.titech.ac.jp/
[11] GSIC, Tokyo Tech. TSUBAME Grand Challenge System, http://www.gsic.titech.ac.jp/GrandChallenge

ChainerMN: Scalable Distributed Deep Learning Framework

Takuya Akiba, Keisuke Fukuda
Preferred Networks, Inc.

One of the keys for deep learning to have made a breakthrough in various fields was to utilize high computing powers centering around GPUs. Enabling the use of further computing abilities by distributed processing is important not only to make the deep learning bigger and faster but also to tackle unsolved challenges. This article focuses on the distributed processing of deep learning and describes the design, implementation and evaluation of ChainerMN, the distributed deep learning framework we have developed.

1 Introduction

It has been learned recently that deep learning can achieve far better predicting performance than existing methods in image recognition, natural language processing, speech recognition and many other fields where machine learning is being applied. The basic technology of neural networks used in deep learning has a long history dating back to the 1950's. As we entered the 2010's, the neural network technology with its long history have made the breakthrough as "deep learning" as described above because it is thought to have successfully combined all the advances of algorithms, large-scale data and high computing powers. Even today, it would be difficult to achieve an outstanding predicting performance by deep learning if one of the three lacks. In this article, we focus on one of the three pillars supporting deep learning: Computing performance.

It has become a standard approach to use highly efficient GPUs for training in many deep-learning tasks. Nevertheless, the training process is still time-consuming even with the latest GPUs because models have also grown massive and complex as GPUs have improved significantly. Taking a long time on training means you have a limited number of times to do trial and error for models and parameters needed to achieve high accuracy, making it difficult to produce a good predicting performance. It also means there is a limit to the usable data size. Thus, using a number of GPUs in parallel is crucial in accelerating calculation.

We once conducted a visual representation learning of a chemical compound using Neural Fingerprint [1] and its advanced method as the TSUBAME trial use by industries in fiscal 2015 with the aim of applying deep learning in a drug development field. As a problem of machine learning, it comes down to the binary classification on the presence of activation. It took 10-20 hours per task, which is shorter than learning a typical model of general object recognition using CNN. However, because there were 72 tasks, if hyperparameter tuning is included, a computing performance equal to or greater than them was required. This is one of the reasons we took on this as a subject of TSUBAME2. With each of the 72 tasks being independent, we were able to conduct the experiment just by a simple parallel execution. Please refer to our trail use report for details of the experiment.

On the other hand, we have also conducted research to make non-independent, single learning even faster and implemented them. ChainerMN described in this article is one of such initiatives. ChainerMN has been developed as an add-on package to provide a distributed learning function to Chainer by installing it. In the course of developing ChainerMN, we took the following features into consideration:

1. **Flexibility**: Chainer is a flexible framework based on its Define-by-Run approach and ChainerMN is designed not to ruin the flexibility aspect. This allows for easy distributed learning even in complex use cases such as dynamic neural networks, generative adversarial networks and reinforced deep learning.

2. **High speed**: We selected technologies assuming practical workloads in deep learning from the very beginning of designing ChainerMN as well as exercised ingenuity with respect to implementation so that hardware performance is fully utilized. This has resulted in the highest performance shown in a comparison experiment with other frameworks as detailed hereinbelow.

This article is about ChainerMN, first explaining the basic elements of distributed deep learning, followed by the implementation of ChainerMN. Finally, we will present the results of our evaluation experiment.

Fundamentals of Distributed Deep Learning 2

2.1 Basic deep learning

We can express the prediction by neural networks against input data x as $f(x;\theta)$ where θ is parameter for neural networks.

Learning in neural networks using backpropagation and stochastic gradient descent or its variations is an iterative algorithm. Each iteration is composed of the following three steps:

1. Forward computation
2. Backward computation
3. Optimization

In the forward-computation step, first the prediction $f(x;\theta)$ is calculated against a input data point x. Then, the loss E is calculated to represent the difference from the correct output y for x. Here, the cross entropy and other indicators may be used.

In the backward-computation step, $g = \frac{\partial E}{\partial \theta}$, the gradient of the parameter θ in the direction of decreasing the loss E, is calculated. Gradients for all parameters are calculated using the chain rule while going backward from the output layer to the input layer.

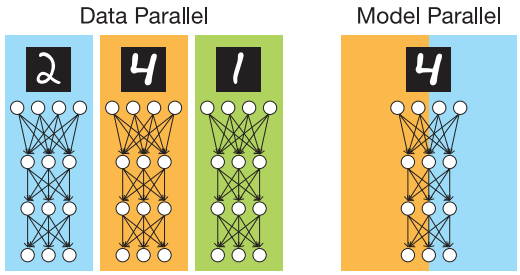
In the optimize step, the parameter θ is updated using the gradient g. The simplest rule is to update θ to $\theta - \eta g$ where η is parameter called a learning rate.

In practice, instead of using a single training example in an iteration, the forward and backward calculations are performed simultaneously against multiple training examples and optimization is executed using the average of gradients against all the examples. The input examples used in an iteration is called a minibatch while its size is called a batch size. A typical batch size ranges from several tens to several hundreds.

Please note that the above description is based on a standard supervised learning. Nonetheless, in case that neural networks are applied to other frameworks such as unsupervised learning and semi-supervised learning, the parallelizing method we will explain below is applicable and ChainerMN is also usable.

2.2 Two parallelization approaches

There are generally two types of approaches to speed up deep learning by distributed processing: data parallel and model parallel. In the data-parallel approach, each worker has a model replica and calculate gradients of different minibatches. Workers use these gradients to update the model in a collaborative manner. In the model parallel approach, each worker has a portion of the model and work in cooperation with others to do the calculation for one minibatch.



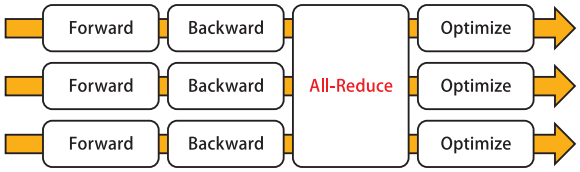
The model-parallel approach was actively used in the days when GPU memory was small. At present, the model parallel is rarely used in its basic form as the data parallel approach is being generally used. In the meantime, some issues with the data paralleled approach have surfaced while a research on a new form of the model parallel is underway. The model parallel and the data parallel can be used at the same time as well.

2.3 Synchronized and asynchronous in data parallel

In this subsection, we will focus on the dataparallel approach which is commonly used now. The data-parallel approach is roughly divided into synchronized and asynchronous types and we explain about the former first.

Each iteration in synchronized, data-parallel deep learning is composed of the following four steps:

1. Forward computation
2. Backward computation
3. All-Reduce communication
4. Optimization



This has an additional step "All-Reduce" to the regular iteration described earlier. In this step, workers communicate with each other to find the average of gradients calculated by individual workers and distribute the average. All workers update the model using the gradient they have obtained through the communication. If we define the batch size processed by each worker as *b* and the number of workers as *n*, the gradient obtained through communication is equivalent to the gradient in the batch size bn. This means gradients are calculated using more training data in one iteration, improving the gradient quality and accelerating the learning process.

Asynchronous type, on the other hand, uses special workers called a parameter server. The parameter server controls model parameters. Normal workers send gradients to the parameter server once the gradients are obtained by forward and backward calculations. The parameter server receives and uses the gradients to update the model. Workers receive new model parameters and calculate gradients again.

Implementation of ChainerMN 3

3.1 API Design

Chainer is a framework with its define-by-run feature. Define-by-run is a model that takes advantage of the flexibility of script languages where learning models and computational flows are defined at runtime. A define-and-run approach, on the other hand, is a model that pre-defines a structure of networks, after which data is input and calculation is done. While potentially easier to optimize performance, this approach is said to lack flexibility.

Chainer provides programming models that enable you to define complex neural networks flexibly or make modifications during runtime thanks to its define-by-run approach. This lets researchers and engineers work on new models or complex models through trial and error with ease and therefore is suitable for research and development of machine learning.

Upon development, we designed the API with the objective of making it easily portable from existing Chainer programs without putting limitations on the flexibility of Chainer.

ChainerMN is designed to add a communication function to existing Chainer programs by adding additional component to Optimizer. On top of this, basic porting can be done just by adding the Scatter process which distributes data for data parallel computations. Other parts i.e. Iterator, Updater and Evaluator do not need to be changed in basic use cases. Because of this API design, it allows various Chainer programs to be ported with less efforts while making the most of the advantage given by define-by-run.

3.2 Parallel computation model of synchronized data parallel

The data-parallel, synchronized computational model has been adopted by the current ChainerMN. Each computational model is described already.

First, we decided to use the data-parallel approach because existing deep learning applications would easily be extensible and faster learning process through data parallel was highly expected. Roughly speaking, data parallelization is tantamount to increasing a minibatch size in a normal deep learning application and has its advantage of being applicable without having to make major changes in algorithms and codes of existing programs. However, caution should be exercised regarding accuracy of trained models as stated later.

Whether the synchronized or asynchronous type is desirable is also a nontrivial question since different kinds of strategies have been taken in each implementation and results would vary depending on tasks or settings. The paper [3] shows experimental results that the asynchronous type is less stable in terms of convergence whereas it is faster to converge in the synchronization. In addition, Allreduce is available in the synchronized type and we can also benefit from the optimized, proven, group communication mechanism of MPI while in the asynchronous model the implementation scheme that uses a parameter server is used in general. Based on these points, we decided to adopt the synchronized type for ChainerMN at first.

3.3 Implementation and Optimization

A communication pattern as a HPC application isrelatively simple because ChainerMN, at the time of writing this article, uses the data-parallel synchronized type model. Roughly speaking, auxiliary parts include Allreduce, a major process to replace gradients which are learning and evaluation results, and Scatter which arranges necessary data before learning.

Allreduce is the area that especially requires speed because it is called in every learning iteration and needs to process a large amount of data. We attempted to improve speed by using not only MPI but also NCCL library developed by NVDIA. NCCL is a highly-optimized communication library which provides a faster Allreduce process between NVIDIA GPUs within nodes. By hierarchically combining NCCL which performs communication among GPUs within nodes and MPI which performs communication among nodes, we have successfully implemented a faster Allreduce process.

(Note: NCCL library version 2 was released as we were writing this paper. NCCL2 library supports communication among nodes using Infiniband, which realizes a highly optimized Allreduce regardless of MPI.)

Evaluation 4

4.1 Important notes regarding evaluation

One of the factors making distributed deep learning difficult is that improving throughput does not necessarily mean better learning efficiency. Take the data-parallel approach for example. The more you increase the number of GPUs, the larger the batch size gets. However, if the batch size gets large enough to have an adverse effect on learning, the accuracy of the model will start to decrease gradually. Two causes of this have been pointed out. First, in case of learning for the same number of epochs, the number of iterations will decrease and the model may not mature. It is also known that as dispersion of gradients gets smaller, it likely leads to local solution of poor quality called sharp minima resulting in a model with bad generalization ability [2].

It can be said that benchmark results reporting throughput only without considering these challenges facing distributed deep learning are insignificant. With the simple Allreduce being the major communication pattern, a superficially high performance and scalability can be achieved easily by increasing a batch size or reducing a synchronization frequency. This kind of setting, however, generates no useful learning result. Although our GPU memories had a plenty of margin this time also, we made the setting to limit the batch size handled by each GPU so that the model could achieve a sufficient accuracy.

4.2 Evaluation results

The figure 1 shows the performance evaluation using a model called ResNet-50 [4] trained on ImageNet dataset which is being widely used as a benchmark of image recognition. In this evaluation, we compared ChainerMN, MXNet, CNTK and TensorFlow on a computing environment with 1-128 GPUs. For the detailed settings and supplemental figures of this experiment, please refer to our blog post [5].

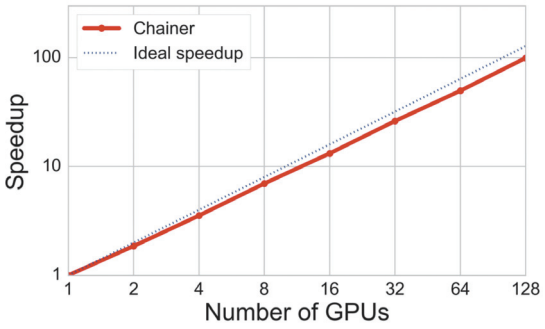


Fig. 1 Training Speedup for ImageNet Classification (ResNet-50)

While Chainer did not perform as well as MXNet and CNTK with 1 GPU, it was the fastest with 4 GPUs or more. Because Chainer is a defineby- run style framework written in Python, CUDA kernel issuance and other processes in CPU are prone to become a bottleneck, which we believe is unfavorable to Chainer as opposed to C++ based MXNet and CNTK. Nevertheless, Chainer was the fastest with 4 GPUs or more thanks to its hierarchical communication of NCCL and MPI combined.

The possible reason for TensorFlow's low performance is due to its client-server model using gRPC. A client-server model has a higher overhead than the collective communication Allreduce and we also observed a phenomenon that gRPC performance became lower with huge messages during the experiment [6].

As stated above, we believe any deep learning benchmark is meaningless without evaluating evaluation.

Figure 2 represents computation time and accuracy when distributed learning was conducted using ChainerMN. The figure shows ChainerMN was able to maintain accuracy with 128 GPUs as the learning time became faster, in comparison with 8 GPUs.

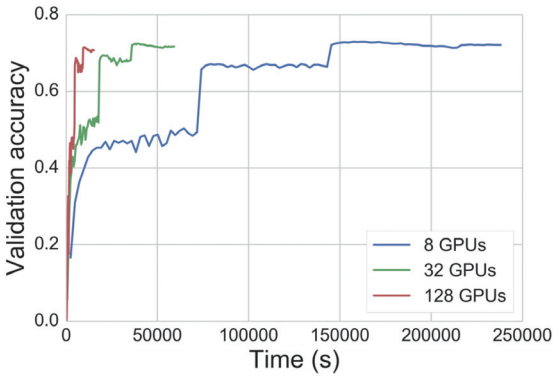


Fig. 2 Achieved Accuracy for ImageNet classification

Conclusions 5

In this article, we described the fundamentals of distributed deep learning and the design, implementation and evaluation of ChainerMN. Chainer and ChainerMN are designed to have both high flexibility and scalability with its primary object of accelerating research and development in deep learning. We will continue making improvements by tackling challenges such as model parallel, communication and computation overlaps, asynchronous computation among workers, optimized communication by compressed gradients, and failure resistance.

While deep learning has already become essential in some applications, the global competition in its research and development continues to intensify as the technology is ever making rapid progress. In order to advance the R&D activities efficiently, it is vital to have a storage that stores and utilizes learning data with its size reaching up to several terabytes at times, an accelerator that makes vast linear algebra operations faster as well as robust computing infrastructure including low-delay, broadband networks to exchange a huge amount of gradient data.

TSUBAME3.0 is a big-data supercomputer with the best computational efficiency in the world, a firstrate computing power, high-speed interconnection and a strong network file system and local storage. We believe by having TSUBAME3.0 and Chainer/ChainerMN combined together we can have a big advantage when competing globally in the research and development.

References

[1] Dahl, G. E., Jaitly, N., & Salakhutdinov, R. (2014). Multi-task neural networks for QSAR predictions. arXiv preprint arXiv:1406.1231

[2] Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P.. On largebatch training for deep learning: Generalization gap and sharp minima. ICLR'17.

[3] Pan, X., Chen, J., Monga, R., Bengio, S., & Jozefowicz, R. Revisiting Distributed Synchronous SGD. ICLR'16 workshop track.

[4] Kaiming He et al. "Deep Residual Learning for Image Recognition", CVPR 2016.

[5] https://research.preferred.jp/2017/02/chainermn-benchmark-results/

[6] https://github.com/tensorflow/tensorflow/issues/ 6116

Creation of Chemical Structures Aimed for Drug Design, Facilitated by Efficient GPU Computing

Yuko Otani* Tomohiko Ohwada*
* Graduate School of Pharmaceutical Sciences, The University of Tokyo

New chemical structures are crucial for new functions such as biological activities, which are relevant to drug discovery. The question is *how we can create new chemical structures a priori in terms of three-dimensional coordinates*. Moreover *how can we define the structures in solution?* We demonstrate here our experimental efforts to synthesize two new molecular scaffolds which take "defined" 3D structures such as helices based on artificial amino acids and a structured lipid mediator. In order to design the 3D structures (conformations), the population of conformers, which is affected by the environmental conditions such as solvents must be taken into account. Medicinal chemists are now forced to change their intuitions to free-energy based pictures of chemical structures and molecular motions. Combination of the GPU computing with the experiments, particularly synthetic chemistry is required to create new chemical structures.

1 Introduction

New chemical structures are crucial for new functions such as biological activities which are relevant to drug discovery. The question is *how we can create new chemical structures a priori in terms of three-dimensional coordinates*. Moreover *how can we define the structures in solution?* We demonstrate here our experimental efforts to synthesize two new molecular scaffolds, which take "defined" 3D structures.

Helical molecules based on artificial amino acids can literally resemble natural α-helix composed of α-amino acids. However, structural features such as pitch, diameter, and the number of residues per turn of these new helical molecules are completely different from those of natural α-helix. Such different features can be utilized as functional peptide mimics, e.g. artificial helix composed of smaller number of repeat units can cover similar length with α-helix composed of larger number of α-amino acids. This is one of the experimental strategies to create new chemical diversity.

Our another focus is to create biological functional analogues (i.e., non-lipid analogues) of lipid mediators. Our target lipid mediator is a flexible molecule which contains many rotatable single bonds. This molecule can activate at least three receptor subtypes. We try to define the structure responsible for activation of the respective receptor. In order to consider the 3D structures (conformations), we need to take into account all possible conformations to characterize a single chemical molecule. We need to know the population of conformers, which may be affected by the environmental conditions such as solvents.

Medicinal chemistry essentially studies design and optimization of multiple interactions of biomolecules such as proteins, DNA, RNA and others, with organic compounds such as medicines and antibodies. Therefore, we need to consider multiple combinations of accessible conformational space of molecules in order to characterize optimal interactions. Chemists had been often proposing chemical structures with their intuitions. However, nowadays chemists are forced to consider free-energy based pictures of chemical structures and molecular motions. Efficient GPU computing is required to establish such free-energy based pictures of molecules, which will be related to the experimental phenomena. Particularly medicinal chemists are forced to establish non-linear relationships between conformational populations and biological activities to design/create new chemical structures.

We demonstrate here our experimental efforts to synthesize new molecules which take "defined" 3D structures. We are trying to combine the experimental events with GPU computation, which were/are carried out on the TSUBAME 2.5 supercomputing system. We will demonstrate here some of such combinations.

2 Creation of ordered structures, different from naturally occurring molecules

2.1 Length-dependent convergence to organized structures of nitrogen- pyramidalized bicyclic β-proline oligomers.

Peptides and proteins take ordered structures by using hydrogen-bonds. However such ordered structures are collapsed when they are exposed to water, because external water makes hydrogen bonding with amino acid residues. Therefore, tertiary amides, which lack hydrogen atom on the amide nitrogen atom, can create helical molecules, which are robust even in water environment. However, tertiary amides contain amide *cis-trans* isomerization. The monomeric

α-proline amide (N-acetyl-L-proline-NHMe) favors *trans*-amide conformation with a *trans:cis* ratio of approximately 73:27 in D_2O. On the other hand, β-peptides, because they can form stable regular structure and are stable to proteolytic degradation. Oligopeptides of non-natural β-proline and its analogues can also adopt ordered structures. However, control of the amide *cis-trans* equilibrium in oligomers of β-proline analogs is often less effective than in α-proline oligomers. However, peptide homooligomers of β-amino acids bearing the bicyclic 7-azabicyclo[2.2.1] - heptane skeleton (7-azabicyclo[2.2.1]heptane-2- carboxylic acid, 1, Figure 1), which is a conformationally constrained β-proline analog [1-8], have been synthesized (dimer 2, trimer 3, tetramer 4, and pentamer 5, Figure 1(a)).

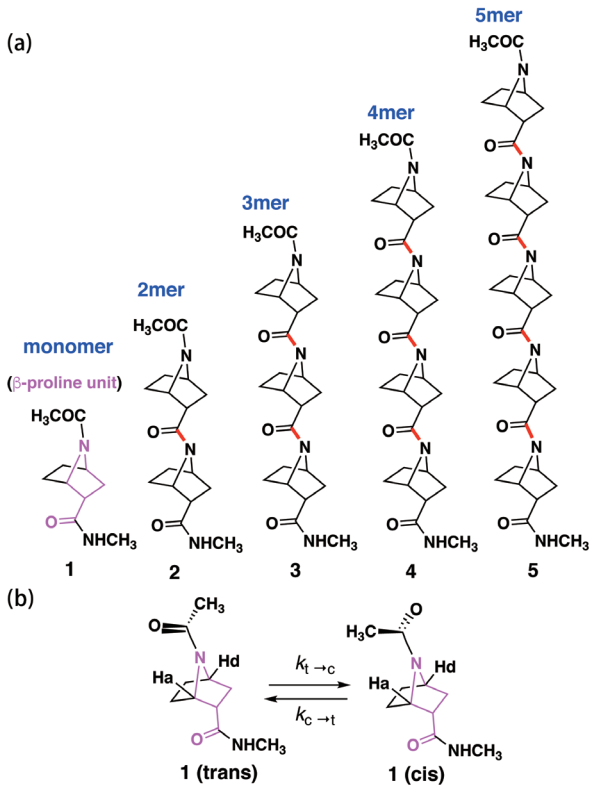


Fig. 1 (a) Designed β-proline analogue and oligomers. (b)Amide *cis-trans* isomerization.

A derivative of dimer 2 with a tBuOCO (Boc) group at the N-terminal and a N-methylamide at the C-terminal exhibits *trans-cis* equilibrium with a slight preference for the *trans* form (*trans*:*cis* = 55:45 in $CDCl_3$) (Figure 1(b)). However, detailed solution structural analysis of oligomers longer than the dimer was hampered by line-broadening of NMR signals due to slow interconversion between amide rotamers. Furthermore, the bicyclic amide takes a nonplanar structure in solution, and the amide is tilted to either side of the amide plane (formed by the amide nitrogen and two bridgehead carbons). The experimental circular dichroism (CD) spectra of the hydrochloride salts of unprotected oligopeptides (2, 3, 4, and 5) in methanol were also reported [3] (see Figure 3 (a)). They showed characteristic and intense CD with the minimum at around 198 nm and the maximum at around 217 nm, and an isodichroic point at around 206 nm. Furthermore, the intensity per residue at the maximum increased length-dependently. These observations suggested that thermodynamically stable regular structure could be induced as the oligomer is elongated. However, further investigation was needed to enable the assignment of major structures in solution.

We estimated the rotational barrier of the monomeric amide of bicyclic β-proline by NMR spectroscopy and performed MD simulation of homooligomers with an umbrella sampling method to accelerate amide bond rotation and to enable sampling of a wide range of conformations [9]. We found that as the oligomer is elongated, the ratio of *trans*-amide bonds increases, which implies that extended helical structures are stabilized. This scenario is consistent with the observed CD spectra, in which the intensity of the signal per residue increases as the oligomer is elongated.

The EXSY spectrum of N-Ac-1-NHMe indicated that rotational barriers of N-Ac-1-NHMe in CD_3OD were obtained as $\Delta G^{\ddagger}_{t\rightarrow c,\,300\,K} = 17.9 \pm 0.5$ kcal/mol and $\Delta G^{\ddagger}_{c\rightarrow t,\,300\,K} = 17.7 \pm 1.9$ kcal/mol. The rotational barrier of N-Ac-1-NHMe in D_2O estimated by line-shape analysis was similar to that in CD_3OD : $\Delta G^{\ddagger}_{t\rightarrow c,\,300\,K} = 17.9 \pm 0.4$ kcal/mol. Thus, the solvent effect on the amide equilibrium and the rotational barrier is small. The barrier is smaller than the reported value of the rotation barrier of acetyl-L-proline-NHMe, which has $\Delta G^{\ddagger}_{t\rightarrow c,\,298\,K} = 20.40$ kcal/mol. This is due in part to the reduced double bond character of the amide bond in N-Ac-1-NHMe, which is expected to take a nitrogen-pyramidal amide structure as in other related bicyclic compounds.

Each conformer was classified with respect to the dihedral angle ω along the amide linkage and depicted as a combination of *t (trans)* and *c (cis)* from the N-terminal position in the cases of oligomers longer than the dimer. Populations of all conformers arising from different combinations of amide rotamers are calculated by MD simulation with an umbrella sampling method (Figure 2(a)).

It is suggested that the rotamers with high contents of *trans*-amide are more highly represented throughout the oligomers. For example, the most populated conformers in methanol were *t* for 2 (57%), *tt* for 3 (40%), *ttt* for 4 (28%) and *tttc* for 5 (23%) (Figure 2(a) and (b)). Therefore, The ratio of *trans*-amide among all amide bonds, averaged over all conformers, increased gradually with increasing oligomer length in methanol: the percentages of trans-amide bond in 2, 3, 4, and 5 are 57%, 65%, 68% and 70%, respectively (Figure 1a).

The CD spectra of all possible rotamers of oligomers 2, 3, 4, and 5 were calculated by using the TDDFT method at the level of IEF-PCM (methanol) - B3LYP/TZVP and population-averaged for each rotamer according to the probability obtained from MD simulation (Figure 3(b)).

The obtained calculated CD spectra shows similar shape to the experimental CD spectra, and the intensity per residue at both minimum and maximum increased with increasing oligomer length. Therefore, this study implied that increase in molecular freedom, that is, nitrogen pyramidalization may enhance generation of ordered structures. This is a new paradigm to establish ordered structures.

2.2 Hydrogen Bonding to Carbonyl Oxygen of Nitrogen-Pyramidalized Amide

The amide bond is a key linkage in proteins, peptides and peptide mimics, serving to connect two neighbouring amino acids. Most amide bonds are planar, but nonplanar amide structures have been suggested to occur even in proteins and peptides. Although the magnitude of nonplanarity found in proteins and peptides is not large, some nonplanar amides with distinct ground states have been reported. In such non-planar amides, the nitrogen atom gains a partial sp^3-character (i.e., nitrogen-pyramidalization), and at the same time bond-twisting occurs. This phenomenon results in increased electron density at the nitrogen atom, and decreased electron density at carbonyl oxygen, as compared with the situation in a planar amide. While hydrogen-bonding to the pyramidalized electron-rich nitrogen atom has been experimentally and computationally investigated, there has been little study on the possibility of hydrogen bonding to the electron-deficient carbonyl oxygen atom of non-planar amides.

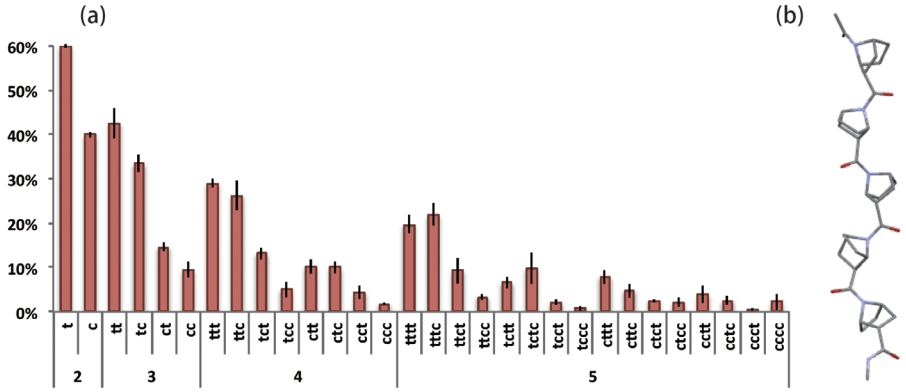


Fig. 2 (a) Probability of each conformer in dimer 2, trimer 3, tetramer 4 and pentamer 5 in methanol.
(b) A representative *tttt* structure of 5.

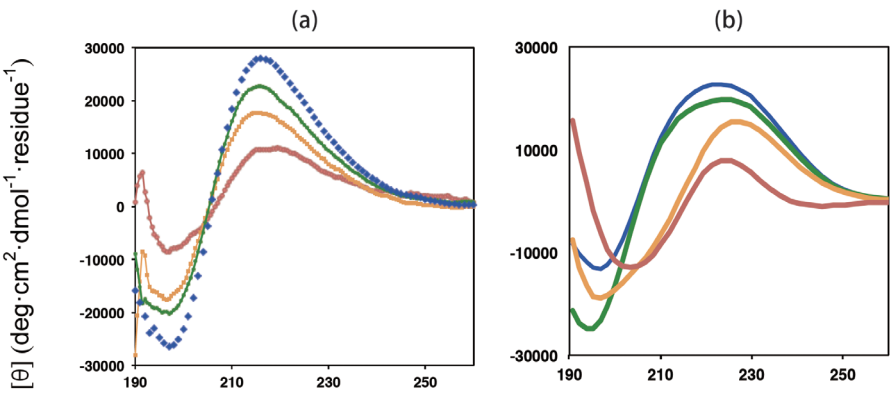


Fig. 3 (a) Experimental and (b) simulated CD spectra of oligomers. (2: red, 3: orange, 4: green, 5: blue)

7-Azabicyclo[2.2.1] heptane amides are chemically stable and intrinsically nonplanar (Figure 4 and also Figure 1), and substitution at the bridgehead position of bicyclic β-proline derivatives (7-azabicyclo [2.2.1] heptane-2-carboxylic acids) can bias amide *cis-trans* isomerization toward either *cis* or *trans*, depending on the position of the bridgehead substituent (see section 2.3.). Homooligomers of optically active derivatives take helical structures with all-*cis* or all-*trans* amide bonds. Nonplanarity of amide structures in homooligomers can be detected in crystal structures. In solution, there is an equilibrium between two conformers, *anti*, i.e., the carbonyl group is tilted toward the opposite side of the C-terminal group with respect to the plane of the nitrogen atom and two bridgehead carbons, and *syn*, in which the carbonyl group is tilted toward the same side of the C-terminal group. But, because of the small energy difference between the *anti* and *syn* conformers and low inversion barrier, it has been difficult to detect the direction of pyramidalization in solution. We demonstrated the presence of hydrogen-bonding to the carbonyl group of nitrogen-pyramidalized amides by means of crystallographic and NMR analyses and vibrational circular dichroism (VCD) spectroscopy [8]. Our results indicated that such hydrogen-bonding is strong enough to switch the preferred direction of nitrogen-pyramidalization of ground-state amide bonds of bicyclic β-proline derivatives (Figure 4).

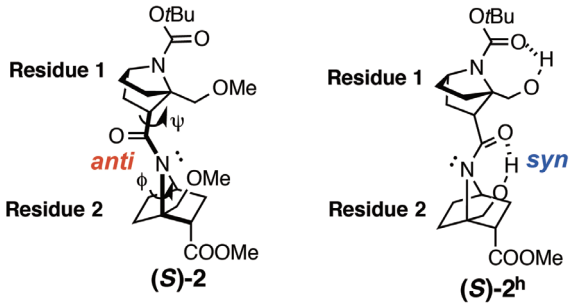
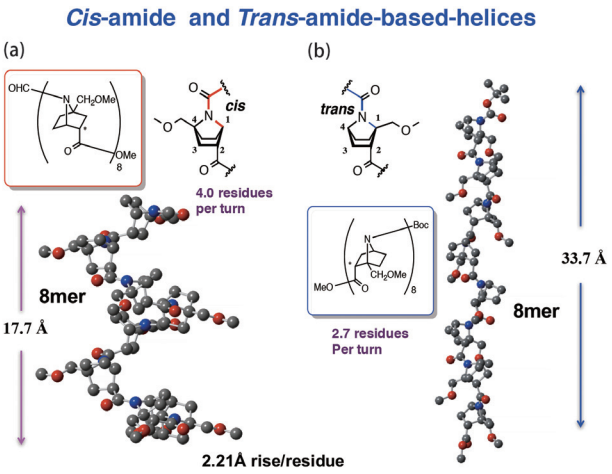


Fig. 5 Non-natural helix structures generated by restricted amide *cis-trans* isomerization.
(a) *cis*-amide helix. (b) *trans*-amide helix.

Directional preference of nitrogen- pyramidalization of non-planar amides has been little discussed so far, probably because amides with a distinct non-planar ground state are too unstable and the magnitude of non-planarity of protein and peptide amides is only marginal. We confirmed that bicyclic β-proline derivatives were a good structural platform to observe such effects.

2.3 Complete control amide *cis-trans* isomerization generate helical molecules

We have reported the synthesis and structural analysis of homooligomers of 7-azabicyclo [2.2.1] heptane-2-*endo*-carboxylic acid, a bridged β-proline analogue (see section 2.1.). Our further study of bicyclic oligomers with a substituent installed at the remote C4-bridgehead position (Figure 5(a)) revealed that a bridgehead methoxymethyl substituent completely biased the amide *cis-trans* equilibrium to the *cis*-amide structure [5]. Homooligomers take helical structures.



These helical structures were generated independently of the number of residues and irrespective of the solvent. This complete selectivity is assumed at least partially to stem from steric repulsion between the bridgehead substituent and the neighboring residue. Thus, we expected that introduction of a substituent at the C1-bridgehead position adjacent to the carboxylic acid moiety, rather than the remote C4-bridgehead position (Figure 1(b)), would tip the *cis-trans* amide equilibrium towards *trans*-amide structure without the aid of hydrogen bonding. This expectation was realized [6]. Such homooligomer takes trans amide and the overall structure provide a helical structure, which is different from the helix based on cis-amide linkage (Figure 5(b)). In Figure 5, the energy-minimized structure of the *trans*-amide-octamer as obtained by Monte Carlo conformation search followed by DFT geometry optimization in simulated water. The energy-minimized structure bears all-*trans* amides and

takes a left-handed helical structure. The helix has about 2.7 residues per turn and 4.2Å rise per residue. These parameters are different, but not much, from the case of PPII (3 residues per turn and 3.1Å per residue). This is also different from the structure of the *cis*-amide type oligomer substituted at the C4-bridgehead position (Figure 5(a)); the (*S*)-oligomer take a left-handed *cis*-amide helix with about 4 residues per turn and a 2.2Å rise per residue (Figuer 6). These structural features of these artificial helix allow us to utilize short oligomers as new scaffolds for create modulators of protein-protein interaction (PPI) (unpublished data).

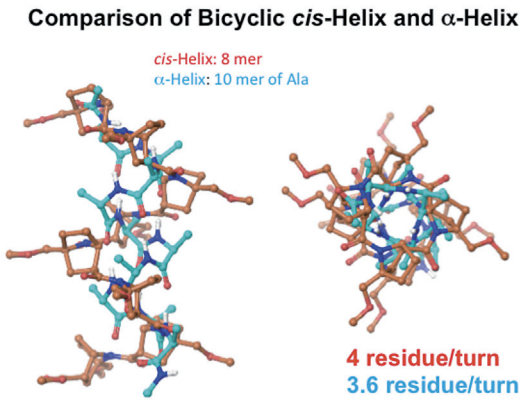


Fig. 6 Comparison of non-natural *cis*-amide helix and natural *α* -helix.

Medicinal chemistry of lipid mediators 3

3.1 Conformational constrain in lipid mediator

Lysophosphatidylserine (LysoPS), which is derived from phosphatidylserine by enzymatic deacylation, has lipid mediator-like actions, and induces multiple cellular responses both *in vitro* and *in vivo*, including mast cell degranulation, neurite outgrowth in PC12 cells, suppression of proliferation of T lymphocytes, migration of fibroblasts and tumor cells and engulfment of apoptotic cells by macrophages (Figure 7).

Recent studies to find ligands of orphan G-protein-coupled receptors (GPCRs) identified three LysoPS-specific receptors, namely P2Y10 (LPS_2), A630033H20 (LPS_{2L}, (LPS_2-like)) and GPR174 (LPS_3) in addition to the previously identified LysoPS receptor (GPR34 (also known as LPS_1)). Orthologs of GPR34, P2Y10 and GPR174 have been found in human, mouse, rat and zebrafish, and they respond to endogenous LysoPS with EC_{50} values at submicromolar level. Therefore, there appear to be three LysoPS receptors of pharmaceutical significance. We designed and synthesized a number of conformationally constrained LysoPS analogues by embedding the glycerol moiety into 2-hydroxymethyl-3-hydroxytetrahydropyran and related skeletons (Figure 7), and we examined the selective receptor-activating ability of the resulting derivatives[10-14]. We have discovered that lysophosphatidylserine analogues containing the conformationally constrained glycerol show potent and selective activity towards GPR34 and P2Y10.

Comparision of saturated and unsaturated cyclic and benzene analogeus of the glycerol moiety showed that compounds, which include increased planarity of the ring moiety, have more potent agonistic activities toward P2Y10 than that of compound which bears a saturated ring structure as a glycerol moiety[12], which means that ring planarity is probably involved in the agonistic activity for P2Y10 and GPR34 with potency and receptor selectivity (Figure 7). Molecular simulations will provide supporting insight for the proposed idea.

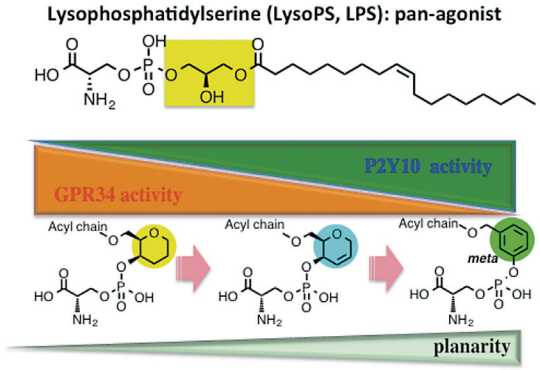


Fig. 7 Structure of bioactive lipid mediator, lysophospahtidyserine (LysoPS) and glycerol analogues.
Planarity is crucial for subtype selectivity.

Creation of Chemical Structures Aimed for Drug Design, Facilitated by Efficient GPU Computing

Summary 4

Experimental chemists are now forced to change their molecular views as molecules are now regarded to have free energy-based pictures and molecular movements (trajectories). New chemical structures are crucial for new functions such as biological activities which are relevant to drug discovery. These questions *how we can a priori create new chemical structures and how we can define the structures in solution* are not new, but have been longstanding. While sampling methods of conformation space and parameterization of force fields need to be improved, time-efficient GPU computation can realize the world in which such calculations become indispensable tools in experimental laboratories. These situations remind experimental chemists of the deep consideration of what is "similarity of structures" and how to do a priori structural hopping, which may lead to solve the question "how to create new chemical structures". Intensive collaborations of computational/informatics/IT scientists and structural biologists with experimental chemists are key for further development. Very recently we witness such collaborations which is so powerful to understand molecular mechanism of the biological events [13,14].

Acknowledgements

Computations were performed in "High Performance Computing Infrastructure (HPCI)" project (hp150177)(TO). A part of computations were performed on the TSUBAME2.5 supercomputer at Tokyo Institute of Technology as a subject of the TSUBAME encouragement program for female users (16IJ0003)(YO). This work was supported by JSPS KAKENHI Grant Number JP 26104508 (YO) and 26293002 (TO).

References

[1] An Evaluation of Amide Group Planarity in 7-Azabicyclo[2.2.1]heptane Amides. Low Amide Bond Rotation Barrier in Solution. Yuko Otani; Osamu Nagae; Yuji Naruse; Satoshi Inagaki; Masashi Ohno; Kentaro Yamaguchi; Gaku Yamamoto; Masanobu Uchiyama; Tomohiko Ohwada *Journal of the American Chemical Society*, **2003**, *125*, 15191-15199.

[2] Alpha,alpha-Disubstituted Amino Acids Bearing a Large Hydrocarbon Ring. Peptide Self-Assembly through Novel Hydrophobic Recognition Tomohiko Ohwada; Daisuke Kojima; Tatsuto Kiwada; Shiroh Futaki; Yukio Sugiura; Kentaro Yamaguchi; Yoshinori Nishi; Yuji Kobayashi *Chemistry-A European Journal*, **2004**, *10*, 617-626.

[3] Oligomers of beta-Amino Acid Bearing Nonplanar Amides form Ordered Structures Yuko Otani, Shiroh Futaki, Tatsuto Kiwada, Yukio Sugiura, Atsuya Muranaka, Nagao Kobayashi, Masanobu Uchiyama, Kentrao Yamaguchi and Tomohiko Ohwada *Tetrahedron*, **2006**, 62. 11635-11644.

[4] Nonplanar Structures of Thioamides Derived from 7-Azabicyclo[2.2.1]heptane. Electronically Tunable Planarity of Thioamides Tetsuharu Hori; Yuko Otani; Masatoshi Kawahata; Kentaro Yamaguchi; Tomohiko Ohwada. *Journal of Organic Chemistry*, **2008**, 73, 9102-9108.

[5] Water-stable Helical Structure of Tertiary Amides of Bicyclic β-Amino Acid Bearing 7-Azabicyclo[2.2.1]heptane. Full Control of Amide Cis-Trans Equilibrium by Bridgehead Substitution. Masahiro Hosoya; Yuko Otani; Masatoshi Kawahata; Kentaro Yamaguchi; Tomohiko Ohwada *Journal of the American Chemical Society*, **2010**, 132, 42, 14780-14789.

[6] Secondary Structure of Homo-thiopeptides Based on a Bridged β-Proline Analogue: Preferred Formation of Extended Strand Structures with Trans-Thioamide Bonds Yuko Otani, Tetsuharu Hori, Masatoshi Kawahata, Kentaro Yamaguchi and Tomohiko Ohwada *Tetrahedron*, **2012**, 68 (23), 4418-4428. Special Issue (Symposium in print) on Chemistry of Foldamers

[7] Robust Trans-Amide Helical Structure of Oligomers of Bicyclic Mimics of β-Proline: Impact of Positional Switching of Bridgehead Substituent on Amide Cis-Trans Equilibrium. Siyuan Wang; Yuko Otani; Xin Liu; Masatoshi Kawahata; Kentaro Yamaguchi; Tomohiko Ohwada *Journal of Organic Chemistry*, **2014**, *79* (11), 5287-5300.

[8] Hydrogen Bonding to Carbonyl Oxygen of Nitrogen-Pyramidalized Amide-Detection of Pyramidalization Direction Preference by Vibrational Circular Dichroism Spectroscopy. Siyuan Wang, Tohru Taniguchi, Kenji Monde, Masatoshi Kawahata, Kentaro Yamaguchi, Yuko Otani, Tomohiko Ohwada *Chemical Communications*, **2016**, *52*, 4018 – 4021.

[9] Molecular Dynamics Study of Nitrogen-Pyramidalized Bicyclic β-Proline Oligomers: Length-Dependent Convergence to Organized Structure Yuko Otani, Satoshi Watanabe, Tomohiko Ohwada, and Akio Kitao *Journal of Physical Chemistry B*, **2017**, *121*, 100-109.

[10] Synthesis and Evaluation of Lysophosphatidylserine Analogs as Inducers of Mast Cell Degranulation. Potent

Activities of Lysophosphatidyl- threonine and Its 2-Deoxy Derivative Masazumi Iwashita, Kumiko Makide, Taro Nonomura, Yoshimasa Misumi, Yuko Otani, Mayuko Ishida, Ryo Taguchi, Masafumi Tsujimoto, Junken Aoki, Hiroyuki Arai, Tomohiko Ohwada *J. Med. Chem.,* **2009**, *52*, 5837-5863.

[11] Structure-Activity Relationships of Lysophosphatidylserine Analogs as Agonists of G-Protein-Coupled Receptors GPR34, P2Y10, and GPR174 Masaya Ikubo, Asuka Inoue, Sho Nakamura, Sejin Jung, Misa Sayama, Yuko Otani, Akiharu Uwamizu, Keisuke Suzuki, Takayuki Kishi, Akira Shuto, Jun Ishiguro, Michiyo Okudaira, Kuniyuki Kano, Kumiko Makide, Junken Aoki, and Tomohiko Ohwada *Journal of Medicinal Chemistry*, **2015**, *58*, 4204-4219.

[12] Conformational Constraint of the Glycerol Moiety of Lysophosphatidylserine, an Emerging Lysophospholipid Mediator, Affords Compounds with Receptor Subtype Selectivity Sejin Jung, Asuka Inoue, Sho Nakamura, Takayuki Kishi, Akiharu Uwamizu, Misa Sayama, Masaya Ikubo, Yuko Otani, Kuniyuki Kano, Kumiko Makide, Junken Aoki, Tomohiko Ohwada *Journal of Medicinal Chemistry*, **2016**, *59*, 3750-3776.

[13] Probing the Hydrophobic Binding Pocket of G-Protein-Coupled Lysophosphatidylserine Receptor GPR34/LPS1 by Docking-Aided Structure-Activity Analysis Misa Sayama, Asuka Inoue, Sho Nakamura, Sejin Jung, Masaya Ikubo, Yuko Otani, Akiharu Uwamizu, Takayuki Kishi, Kumiko Makide, Junken Aoki, Takatsugu Hirokawa, and Tomohiko Ohwada *Journal of Medicinal Chemistry*, **2017**, *60*, 6384-6399.

[14] Structural insights into ligand recognition by the lysophosphatidic acid receptor LPA_6 Reiya Taniguchi, Asuka Inoue, Misa Sayama, Akiharu Uwamizu, Keitaro Yamashita, Kunio Hirata, Masahito Yoshida, Yoshiki Tanaka, Hideaki E. Kato, Yoshiko Nakada-Nakura, Yuko Otani, Tomohiro Nishizawa, Takayuki Doi, Tomohiko Ohwada, Ryuichiro Ishitani, Junken Aoki, Osamu Nureki *Nature*, **2017**, *548*, 356–360.

● **TSUBAME e-Science Journal vol.16**

Published 11/13/2017 by GSIC, Tokyo Institute of Technology ©
ISSN 2185-6028
Design & Layout: Kick and Punch
Editor: TSUBAME e-Science Journal - Editorial room
Takayuki AOKI, Toshio WATANABE, Yuki ITAKURA
Address: 2-12-1-E2-6 O-okayama, Meguro-ku, Tokyo 152-8550
Tel: +81-3-5734-2085 Fax: +81-3-5734-3198
E-mail: tsubame_j@sim.gsic.titech.ac.jp
URL: http://www.gsic.titech.ac.jp/

TSUBAME

International Research Collaboration

The high performance of supercomputer TSUBAME has been extended to the international arena. We promote international research collaborations using TSUBAME between researchers of Tokyo Institute of Technology and overseas research institutions as well as research groups worldwide.

Recent research collaborations using TSUBAME

1. Simulation of Tsunamis Generated by Earthquakes using Parallel Computing Technique
2. Numerical Simulation of Energy Conversion with MHD Plasma-fluid Flow
3. GPU computing for Computational Fluid Dynamics

Application Guidance

Candidates to initiate research collaborations are expected to conclude MOU (Memorandum of Understanding) with the partner organizations/ departments. Committee reviews the "Agreement for Collaboration" for joint research to ensure that the proposed research meet academic qualifications and contributions to international society. Overseas users must observe rules and regulations on using TSUBAME. User fees are paid by Tokyo Tech's researcher as part of research collaboration. The results of joint research are expected to be released for academic publication.

Inquiry

Please see the following website for more details.
http://www.gsic.titech.ac.jp/en/InternationalCollaboration

GSIC
Global Scientific Information and Computing Center