THE UNIVERSITY OF TOKYO

SCD/ITC

Supercomputing Division,
Information Technology Center, The University of Tokyo

Welcome

SCD/ITC, The University of Tokyo, Japan

The Supercomputing Division, Information Technology Center, The University of Tokyo (http://www.cc.u-tokyo.ac.jp/) was originally established as the Supercomputing Center of the University of Tokyo in 1965, making it the oldest academic supercomputer center in Japan. The Information Technology Center (ITC) was organized in 1999, and the Supercomputing Center became the Supercomputing Division (SCD) of the ITC, joining three other divisions at that time. ITC is also a core organization of the "Joint Usage/Research Center for Interdisciplinary Large-Scale Information Infrastructures" project, and a part of HPCI (the High-Performance Computing Infrastructure) operated by the Japanese Government. The three main missions of SCD/ITC are (i) providing services for supercomputer operations and supporting supercomputer users, (ii) doing research, and (iii) providing education and training. Currently, SCD/ITC consists of more than 10 faculty members. SCD/ITC is now operating five supercomputer systems, a Hitachi SR16000/M1 based on Power7 architecture with 54.9 TFLOPS of peak performance (Yayoi), a Fujitsu PRIMEHPC FX10 System (Oakleaf-fx) with 1.13 PFLOPS, another Fujitsu PRIMEHPC FX10 System (Oakbridge-fx) with 136.2 TFLOPS for long-time execution, Integrated Supercomputer System for Data Analyses & Scientific Simulations (Reedbush) by SGI with 1.80-1.93 PFLOPS, and the Manycore-based Large-scale Supercomputer System (Oakforest-PACS aka. PostT2K) by Fujitsu with 25 PFLOPS as JCAHPC.

Computational Science Alliance, the University of Tokyo

Experiences and knowledges of parallel programming are key advantages for the development of code for complicated, large-scale problems on massively parallel computers. At the University of Tokyo, we established the Computational Science Alliance (http://www.compsci-alliance.jp/) in 2015 with the collaboration of 13 departments including ITC. Primary purpose of the alliance is to provide interdisciplinary education program of HPC for CS&E with flexible and comprehensive classes and courses.

Joint Center for Advanced High Performance Computing (JCAHPC)

In 2013, Center for Computational Sciences, University of Tsukuba (CCS) and ITC agreed to establish the Joint Center for Advanced High Performance Computing(JCAHPC). JCAHPC consists of more than 20 faculty and staff members of CCS and ITC. Primary mission of JCAHPC is designing, installing and operating Oakforest-PACS system. In addition, CCS and ITC will develop system software, numerical libraries, and large-scale applications to for Oakforest-PACS system in collaboration made possible by the establishment of JCAHPC. JCAHPC is a new model for collaboration for research and development between supercomputer centers.

http://jcahpc.jp/

JCAHPC

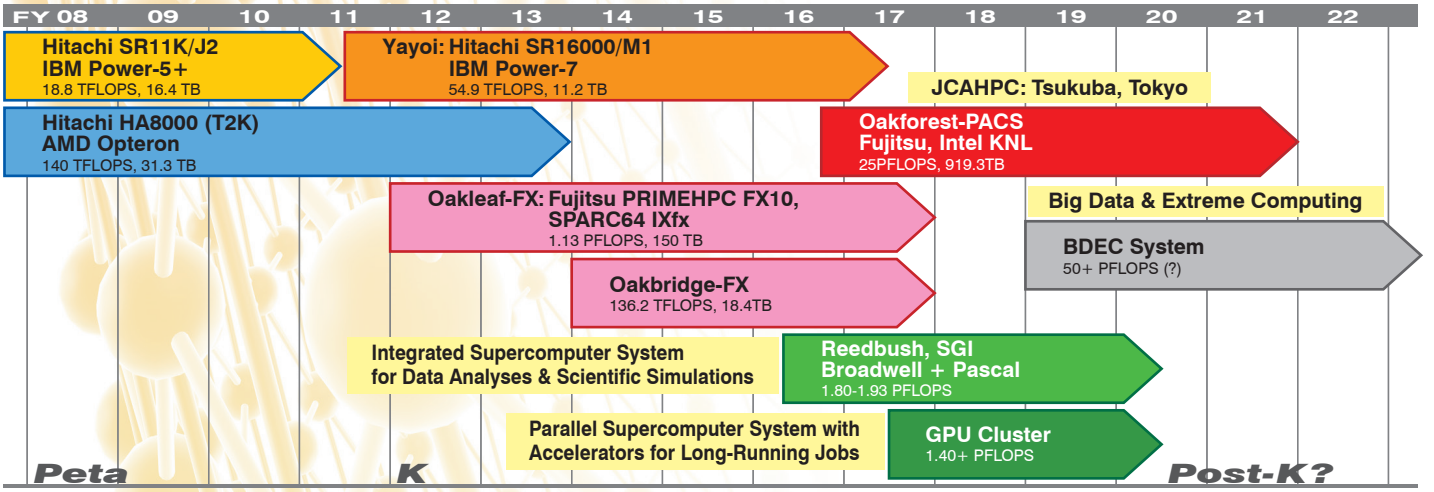
Services for Academia and Industry

There are more than 2,000 user on the three supercomputer systems operated by SCD/ITC, and 50% of them are from outside of the university. All of the systems are quite busy, and their average utilization ratio is approximately 90%. Providing services to support these users is one of our most important responsibilities. Hands-on tutorials for parallel programming are held about 10 times per year, and individual on-site consulting is also available. Up to 10% of the total computational resources of the Oakleaf/Oakbridge-FX systems, the Reedbush system, and the Oakforest-PACS system is open for users from industry.

Supercomputer Systems in SCD/ITC: Oakforest-PACS and Reedbush

SCD/ITC started the operation of two new systems in FY2016. First one is JCAHPC's Oakforest-PACS by Fujitsu, which consists of 8,208 nodes with Intel Xeon Phi processors, and started its full operation on December 1st, 2016. The Oakforest-PACS has been offered to researchers in Japan and their international collaborators through various types of programs operated by the High-Performance Computing Infrastructure (HPCI), by MEXT's Joint Usage/Research Centers, and by each of CCS and ITC under their original supercomputer resource sharing programs. It is expected to contribute to dramatic development of new frontiers of various field of studies, including computational science and engineering (CSE). The Oakforest-PACS will be also utilized for education and training of students and young researchers in both of CSE and high-performance computing (HPC).

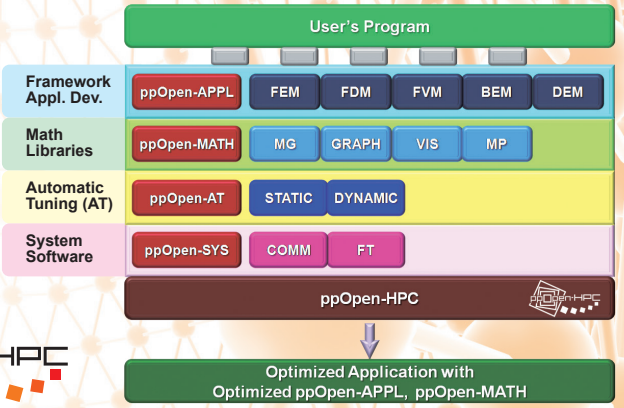
Second one is the Integrated Supercomputer System for Data Analyses & Scientific Simulations (Reedbush) by SGI with Intel Broadwell-EP and NVIDIA Tesla P100 (Pascal) at 1.80-1.93 PFLOPS. The Oakleaf/Oakbridge-FX systems will complete its mission in the end of March 2018. New system (BDEC) is expected to start its operation in April, 2019. Our current systems including Oakforest-PACS are designed for computational science and engineering, and are mostly used for that purpose. In the BDEC System, we plan to develop new types of users, such as Big Data, Deep Learning, etc. The Reedbush system is expected to be a pilot system for the BDEC System. Preliminary operation for compute nodes with Broadwell-EP only at 508 TF started on July 1st, 2016, and full operation started on March 1st, 2017.



International & Domestic projects

ppOpen-HPC & ESSEX-II

"ppOpen-HPC" is an open source infrastructure for development and execution of optimized and reliable simulation code on post-peta-scale (pp) parallel computers based on many-core architectures, and it consists of various types of libraries, which cover general procedures for scientific computation. Source code developed on a PC with a single processor is linked with these libraries, and the parallel code generated is optimized for post-peta-scale systems. The target post-peta-scale system is the Post T2K System. "ppOpen-HPC" is part of a five-year project (FY.2011-2015) spawned by the "Development of System Software Technologies for Post-Peta Scale High Performance Computing" funded by JST-CREST. The framework covers various types of procedures for scientific computations, such as parallel I/O of data-sets, matrix-assembly, linear-solvers with practical and scalable preconditioners, visualization, adaptive mesh refinement and dynamic load-balancing, in various types of computational models, such as FEM, FDM, FVM, BEM and DEM. Automatic tuning (AT) technology enables automatic generation of optimized libraries and applications under various types of environments. We release the most updated version of ppOpen-HPC as open source software every year in November (2012-2015), which is available at http://ppopenhpc.cc.u-tokyo.ac.jp/ppopenhpc/ . In 2016, the team of ppOpen-HPC joined ESSEX-II (Equipping Sparse Solvers for Exascale) project (Leading P.I. Professor Gerhard Wellein (University of Erlangen-Nuremberg), http://blogs.fau.de/essex/), which is funded by JST-CREST and the German DFG priority programme 1648 "Software for Exascale Computing" (SPPEXA) under Japan (JST)-Germany (DFG) collaboration until FY.2018. In ESSEX-II, we develop pK-Open-HPC (extended version of ppOpen-HPC, framework for exa-feasible applications), preconditioned iterative solvers for quantum sciences, and a framework for automatic tuning (AT) with performance model.



Robust and scalable preconditioner for Krylov subspace methods

As a part of ESSEX-II project, we investigate iterative solvers for ill-conditioned sparse linear systems derived from eigenvalue problems. In particular, this investigation aims at developing robust and scalable preconditioners for Krylov subspace methods. The ESSEX-II project addresses eigenvalue problems of quantum systems with strong links to optics and biology and to novel materials such as graphene and topological insulators. Then, we suppose that coefficient matrices have mathematical properties as below.

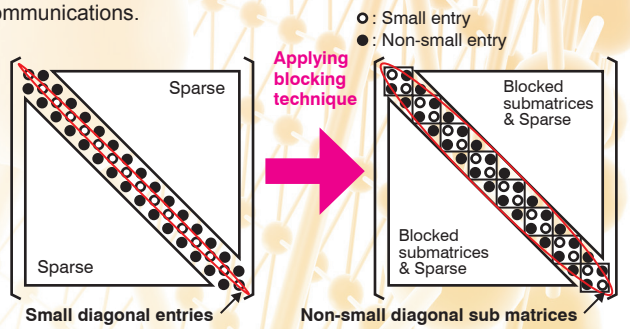
- Indefinite
- Small diagonal and non-small off-diagonal entries
- Ill-condition : High condition number

These properties are not favorable and very challenging for iterative linear solvers because the property of ill-conditioned systems decrease the convergence ratio of Krylov subspace methods. In addition, indefinite matrices also get worse the convergence ratio. To overcome these difficulties, we consider the use of preconditioning and regularization techniques for Krylov subspace methods.

We expect that the use of regularization techniques will improve the condition number of coefficient matrices. However, the technique ends up changing the linear system to be solved. Therefore, we consider that the regularization is only applied to preconditioner not to change the linear system. One of our purposes in this research is to find effective regularizations.

Preconditioners are generally used to improve the convergence ratio of Krylov subspace methods. We can classify the preconditioners in general use into the following three groups; 1) relaxation; 2) incomplete decomposition; and 3) Krylov subspace methods. In this study, we will investigate the effect of all preconditioners. We tentatively focus on the incomplete decomposition preconditioners because of the robustness and effectiveness to the convergence ratio. Then, we have to pay attention to possessing the computational precisions. Small diagonal and non-small diagonal entries are unfavorable for the incomplete decomposition preconditioners.

As one of the key techniques, we consider applying some blocking algorithms to the coefficient matrices. By using the blocking technique, diagonal sub-matrices can be included some non-small off-diagonal entries. The norm of the blocked diagonal submatrices, which corresponds to magnitude of diagonal entries in non-blocked case, can be not small. As a result, faster convergence will be expected because of improving computational precisions. In addition, the blocking technique has the good effects for not only the convergence but also the program optimizations. The technique allows the effective SIMDization of submatrix-vector multiplications. For the massively parallelization, the blocking algorithm with a re-ordering method reduces the communications.



JHPCN: Japan High Performance Computing & Networking plus Data Analysis and Information Systems

JHPCN (Japan High Performance Computing & Networking plus Data analysis and Information Systems) is carried out by the "Joint Usage/Research Center for Interdisciplinary Large-Scale Information Infrastructures," which consists of eight academic supercomputer centers in Japan: those of Hokkaido University, Tohoku University, the University of Tokyo, Tokyo Tech, Nagoya University, Kyoto University, Osaka University, and Kyushu University (Core Organization: UTokyo). The project was started in April of 2010. The total performance of the supercomputer systems involved is approximately 16 PFLOPS (July 2016). JHPCN promotes collaborative research projects using the facilities and human resources of these eight centers, including supercomputers, storage systems, and networks, and interdisciplinary projects utilizing multiple facilities are especially encouraged. Since 2013, the JHPCN centers are responsible for the operation of those joint research resources named the HPCI-JHPCN system as parts the High Performance Computer Infrastructure (HPCI) system. So far 35-40 projects have been accepted each year since 2010.

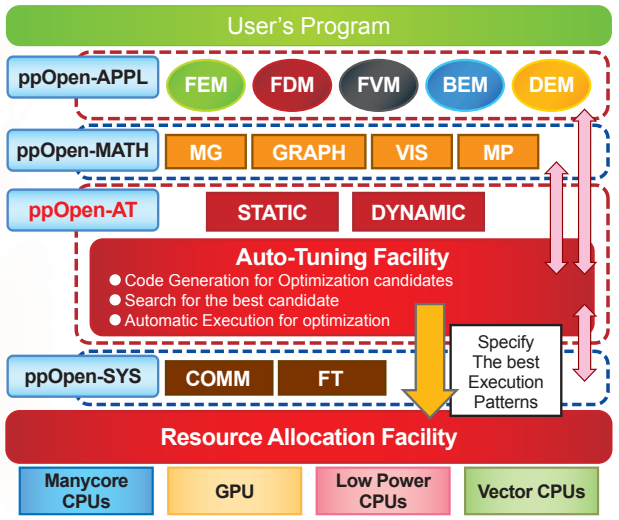
Scientific Computing

ppOpen-AT: An Auto-tuning Description Language for ppOpen-HPC

Computer architectures are becoming more and more complex due to non-uniform memory accesses and hierarchical caches. It is very difficult for scientists and engineers to optimize their code to extract potential performance improvements on these architectures.

We propose an open source infrastructure for development and execution of optimized and reliable simulation code on large-scale parallel computers. We have named this infrastructure “ppOpen-HPC,” where “pp” stands for “post-peta.”

An auto-tuning (AT) capability is important and critical technology for further development of new architectures and maintenance of the overall framework. ppOpen-AT is an AT language for code optimization in five crucial numerical methods provided by the ppOpen-HPC project. The functions and software layers are shown in the figure below. New AT functions in ppOpen-AT are summarized as follows: (1) Directive optimizations for many core CPU, such as the Xeon Phi; (2) directive optimizations for OpenACC; (3) code optimizations for deep hierarchical memories by utilizing code selection; (4) other advanced optimizations for loop transformations.



High-productivity Framework for Stencil Applications

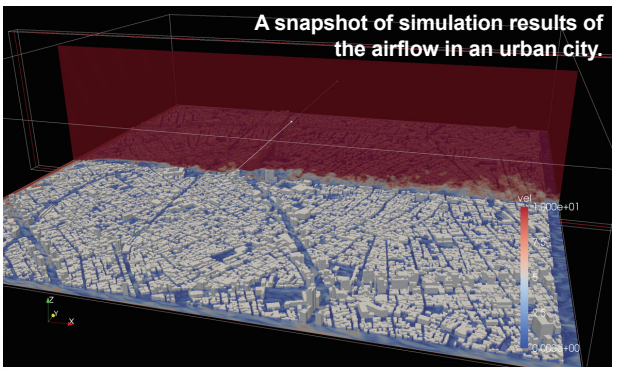
Stencil applications such as computational fluid dynamics are the major applications in high-performance computing. These applications have successfully obtained high performance on modern supercomputers equipped with accelerators such as GPU and Xeon Phi. Obtaining high-performance using thousands of accelerators often needs skillful programming.

We are currently developing the high-productivity framework. The framework is designed for stencil applications with explicit time integration running on regular structured grids. Our framework is implemented in C++ and CUDA languages. It automatically translates user-written stencil functions that update a grid point and generates both GPU and CPU codes. A stencil function can be defined as a C++ functor. The programmers write user code just in the C++ language, and it can be executed on multiple GPUs with the auto-tuning mechanism and the overlapping method to hide communication cost by computation. It can be also executed on multiple CPUs with OpenMP without any change of code. In addition, our framework provides a data structure that supports element-wise computations, which allow us to write GPU kernel codes as inline codes.

```
struct Diffusion3d {
__host__ __device__
float operator() (const float *f, const ArrayIndex &idx,
        float ce, float cw, float cn, float cs,
        float ct, float cb, float cc) {
  const float fn = cc*f[idx.ix()]
        + ce*f[idx.ix(1,0,0)] + cw*f[idx.ix(-1,0,0)]
        + cn*f[idx.ix(0,1,0)] + cs*f[idx.ix(0,-1,0)]
        + ct*f[idx.ix(0,0,1)] + cb*f[idx.ix(0,0,-1)];
  return fn;
}};
```

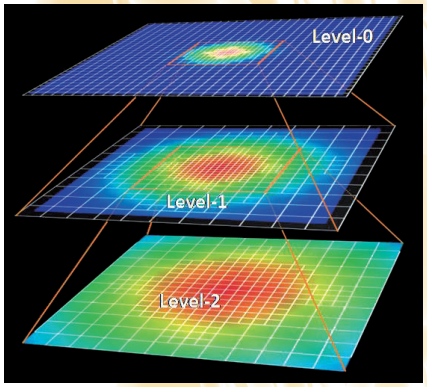
An example of a stencil function

We are introducing the mechanism for enabling the computations beyond the capacity of the GPU device memory into this stencil framework. We realize this by a combination of a temporal blocking method for locality improvement and an automatic swapping between GPU and CPU. The temporal blocking technique can suppress performance degradation caused by frequent memory swapping between GPU and CPU. The automatic swapping is based on a MPI/CUDA wrapper run-time library called HHRT. By using the framework-based approach, computation exceeding the capacity of the GPU device memory is realized without complicated modification of the structure of the time integration loop accompanying data movement between GPU and CPU. The framework-based application for the airflow in an urban city preserves 80% performance of the maximum performance obtained by the original version even with the twice larger than the GPU memory capacity.



A snapshot of simulation results of the airflow in an urban city.

Adaptive Mesh Refinement Technique for ppOpen-HPC



We have developed an adaptive mesh refinement (AMR) technique for ppOpen-HPC applications. The demands of multi-scale and multi-physics simulations will be met with the advent of post-peta scale super computer systems. To achieve such simulations with a reasonable cost of computer resources, the spatial and temporal resolutions have to be adjusted locally and dynamically, depending on the local scales of physical phenomena. In the AMR code, computational grids with different spacing are dynamically created in hierarchical layers according to the local conditions of phenomena. Fine grids suitable to the local domain which need high resolution are applied only there, and other regions are simulated by using moderate size grids. Therefore, increments to the numerical cost due to the localized region are not serious if the AMR technique is adopted.

An example of a computational domain with the adaptive mesh refinement technique.

System, tools & hardware

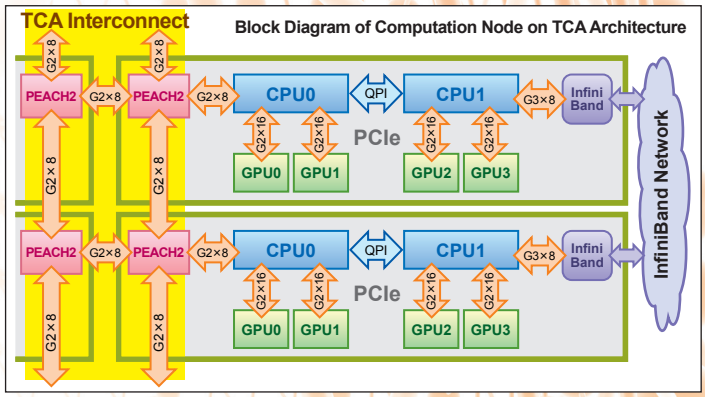
Tightly Coupled Accelerators(TCA)

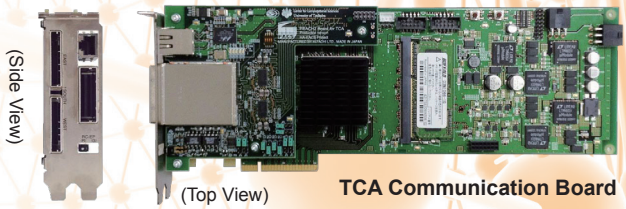
GPGPU is now widely used for accelerating scientific and engineering computing to improve performance significantly with less power consumption. However, I/O bandwidth bottleneck causes serious performance degradation on GPGPU computing. Especially, latency on inter-node GPU communication significantly increases by several memory copies. To solve this problem, TCA (Tightly Coupled Accelerators) enables direct communication among multiple GPUs over computation nodes using PCI Express. PEACH2 (PCI Express Adaptive Communication Hub ver. 2) chip is developed and implemented by FPGA (Field Programmable Gate Array) for flexible control and prototyping in cooperation with University of Tsukuba. PEACH2 board is also developed as an PCI Express extension board.

TCA provides the following benefits:

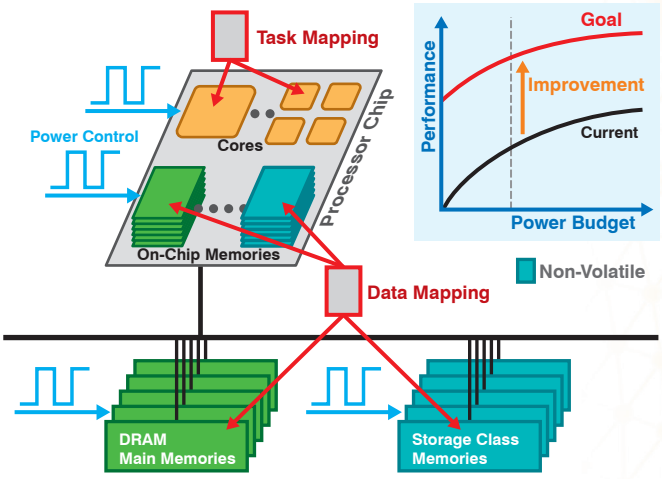
- Direct I/O among GPU memory over nodes
  - Reduce the overhead, obtain good scaling
- Shared PCI Express address space among multiple nodes
  - Ease to program

PEACH2 can transfer not only GPU memory but also host memory seamlessly since PEACH2 relies on the PCIe protocol. The DMA controller in the PEACH2 chip provides a chaining DMA function in order to transfer multiple data segments using the chained DMA descriptors automatically via hardwired logic, and also supports a block-stride transfer which can be specified with a single descriptor.





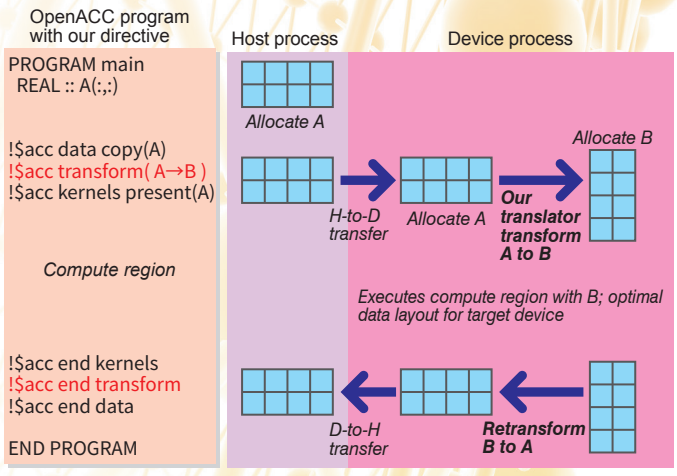
Energy-Efficient HPC Systems



Because power consumption is one of the most critical concerns for future HPC systems, the Information Technology Center has been working on developing power-performance optimization techniques for them from various perspectives. Particularly, we focus on improving energy-efficiency of future HPC nodes which need to operate under limited power budgets and efficiently convert them to performance for suppressing the total system power. To this end, we firstly try to design HPC nodes by use of emerging device technologies including 3D stacking and non-volatile memories, which is going to meet the increasing demands for very large memory space of various HPC applications with moderate power consumption of the memory systems. Secondly, we attempt to coordinate different kind of software/hardware power-performance knobs on the nodes depending on executed workloads to utilize given power budgets efficiently. More specifically, we coordinate the knobs including task mapping on different kind of cores, data mapping between on/off-chip volatile/non-volatile memories and several hardware power controls such as DVFS, power-gating and dynamic resource resizing on various components in the nodes. In addition, we also try to develop power-aware programming methodologies for such nodes.

OpenACC Extension for Performance Portability

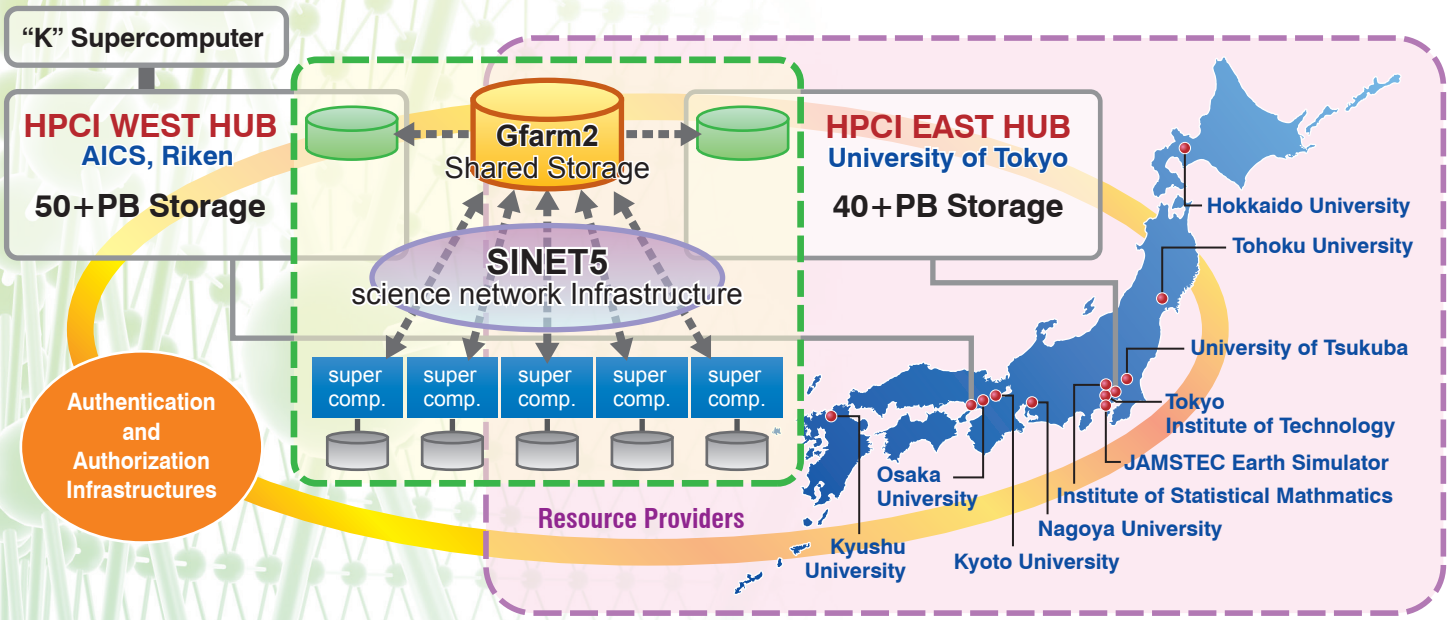
OpenACC is gaining momentum as an implicit and portable interface in porting legacy CPU-based applications to heterogeneous, highly parallel computational environment involving many-core accelerators such as GPUs and Intel Xeon Phi. OpenACC provides a set of loop directives similar to OpenMP for the parallelization and also to manage data movement, attaining functional portability across different heterogeneous devices; however, the performance portability of OpenACC is said to be insufficient due to the characteristics of different target devices, especially those regarding memory layouts, as automated attempts by the compilers to adapt is currently difficult. We are currently working to propose a set of directives to allow compilers to have better semantic information for adaptation; here, we particularly focus on data layout such as Structure of Arrays, advantageous data structure for GPUs, as opposed to Array of Structures, which exhibits good performance on CPUs. We propose a directive extension to OpenACC that allows the users to flexibility specify optimal layouts.



Supercomputers in SCD/ITC

HPCI: High Performance Computing Infrastructure

The HPCI is intended to be an environment enabling a user to easily use the “K” supercomputer and other top-level computation resources in Japan. Also it is expected to match user’s needs and computation resource for accelerating an HPC scenario that includes exploratory research, large-scale research, and industrial use. The HPCI has twelve computational resource providers, of which nine are supercomputing centers of universities and three are governmental research institutes. And these resources suppliers are loosely connected with SINET5, the high speed academic backbone network. SCD/ITC participates in this project as a hub resource provider in the Kanto region (HPCI EAST hub). The HPCI EAST Hub consists of a 40+ PB storage system.



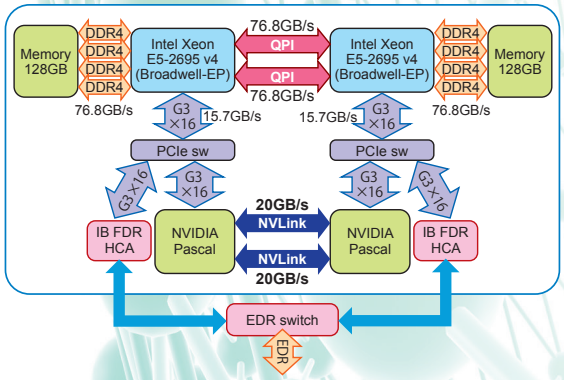
Reedbush (SGI Rackable system)

Reedbush is the first supercomputer system introduced accelerators in SCD/ITC, and total peak performance is expected to around 2 PFlops. We begin computing service only with Reedbush-U (CPU only) in July 2016, and service with full system including Reedbush-H (with GPU) in March 2017. This system is installed at Asano campus (our main campus) and operated by SGI. This system has following missions:

- Supplementary computational resource for reducing congestion of Oakleaf/Oakbridge-FX systems
- Development and promotion for new users, such as Big Data, and Deep Learning researchers
- Pilot system towards the Post FX10 system

	Reedbush-U	Reedbush-H
Peak performance	508.03 TFlops	1287.4-1418.2 TFlops
Number of nodes	420	120
Total memory size	105 TByte	30 TByte
Compute node	SGI Rackable C2112-4GP3	SGI Rackable C1100 series (under development)
CPU	Intel Xeon E5-2695v4 (Broadwell-EP, 18 core, 2.1 GHz) x2 socket 1209.6 GFlops	
Memory	256 GB (DDR4-2400 x4ch x2), 153.6 GB/sec	
GPU	None	NVIDIA Tesla P100 (Pascal, 4.8-5.3 TFlops, 16 GB, 720 GB/sec) x 2
Interconnect	InfiniBand EDR 4x (100 Gbps)	InfiniBand FDR 4x 2 link (56 Gbps x2)
Interconnect topology	Full-bisection Fat Tree	
Parallel file system	Lustre File System (DDN SFA14KE x3) 5.04 PB, 145.2 GB/sec	
File cache system	Burst buffer (DDN IME14K x6) 209 TB, 436.2 GB/sec	

Compute node of Reedbush-H



Yayoi (HITACHI SR16000 M1)

Yayoi (SR16000 M1) consists of Power7 computational nodes. Each node has four Power7 processors and 200GB of shared memory. Each of eight nodes are connected to each other via a fast network. We began providing computing service (only for the “personal course”) in October, 2011 as the successor system of the SR11000. This system is expected to achieve research outcomes for many existing programs which require large shared memory.

Entire	Theoretical peak	54.906 TFLOPS
	Main memory	11200 GB
	Number of nodes	56
Node	Theoretical peak	980.48 GFLOPS
	CPUs (cores)	4 (32)
	Main memory	200 GB
	Memory bandwidth	512 GB/s
Processor	Processor	IBM Power7 (3.83GHz)
	Cache memory	L2: 256 KB/Core L3: 32 MB/Processor
	Theoretical peak pre core	30.64 GFLOPS/core

Supercomputers in SCD/ITC

Oakleaf-fx (FUJITSU PRIMEHPC FX10)

Oakleaf-fx (PRIMEHPC FX10) is the first PFLOPS supercomputer system in SDC/ITC. We began computing service with it in April, 2012. The system has 4800 compute nodes with SPARC64IXfx CPUs and all nodes are connected by 6-Dimension Mesh/Torus Interconnects (Tofu). Well-balanced computational performance and power consumption are achieved. Thanks to the compatible architecture with the K computer, great contributions are expected for computer science in Japan. We provide two kinds of computing services: a "personal course" for individual researchers, and a "group course" for research groups. We also provide various types of special services such as services for educational purposes, services for young users, services for commercial users, and a program called the “large-scale HPC challenge.”

Entire	Theoretical peak	1.13 PFLOPS
	Main memory	150 TB
	Number of nodes	4,800
Node	Theoretical peak	236.5 GFLOPS
	Processor	Fujitsu SPARC64 IXfx (1.848 GHz)
	CPU (core)	1 (16)
	Main memory	32GB
	Cache memory	L2: 12 MB (shared by 16 cores)
	Main bandwidth	85 GB/s

Oakbridge-fx (FUJITSU PRIMEHPC FX10)

The Oakbridge-fx is another PRIMEHPC FX10 system for long-time execution. The system has 576 nodes that is the same architecture as Oakleaf-fx’s one. The Oakleaf-fx users can also use this system with job class: long, which allows to use 24-576 nodes for up to 1week.

Entire	Theoretical peak	136.2 TFLOPS
	Main memory	18 TB
	Number of nodes	576
Node	Same as Oakleaf-FX	

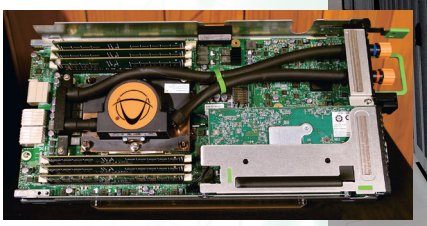
Oakforest-PACS (FUJITSU PRIMERGY)

Oakforest-PACS is the first supercomputer introduced by JCAHPC (Joint Center for Advanced HPC) which is established by SCD/ITC and Center for Computational Sciences, U. Tsukuba (CCS).The system consists of 8,208 nodes of Intel Xeon Phi (Knights Landing) as host processor, and Omni-Path Architecture provides 100 Gbps interconnection. Additionally, the system employs the parallel file system with 26 PB, and fast file cache system with 940 TB, over 1.5 TB/sec BW. This system is installed at Kashiwa campus and operated by Fujitsu. Full operation of Oakforest-PACS has started on December, 2016.

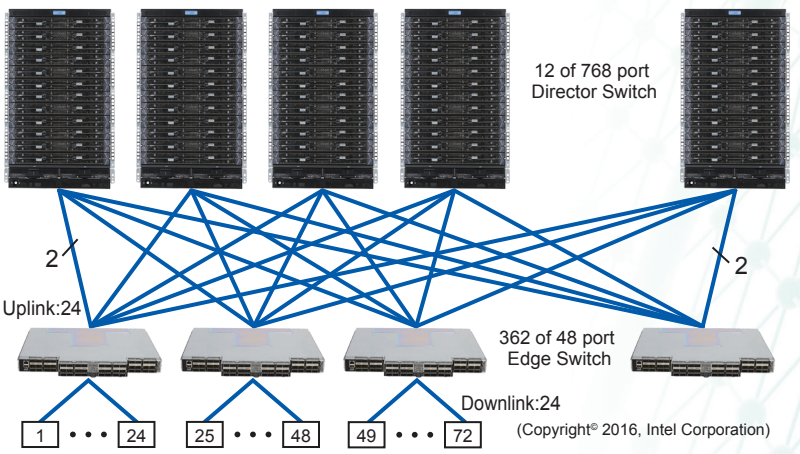
Oakforest-PACS will be offered to researchers in Japan and their international collaborators through various types of programs operated by HPCI, by MEXT’s Joint Usage/Research Centers, and by each of CCS and ITC. It is expected to contribute to dramatic development of new frontiers of various field of studies, including computational science and engineering (CSE). This system will be also utilized for education and training of students and young researchers in both of CSE and high-performance computing (HPC). Both of CCS and ITC will continue to make further social contributions through operations of the Oakforest-PACS.

Entire	Theoretical peak		25 PFLOPS
	Main memory		128TB (High BW)+ 770 TB (Low BW)
	Number of nodes		8,208
Node	Theoretical peak		3.05 TFLOPS
	Processor		Intel Xeon Phi (Knights Landing) 7250
	CPU (Core)		68
	Main memory		16 GB (High BW) + 96 GB (Low BW)
	Main bandwidth		490 GB/sec (High BW, effective) + 115 GB/sec (Low BW)
	Cache memory		L2: 1 MB / tile (2 cores)

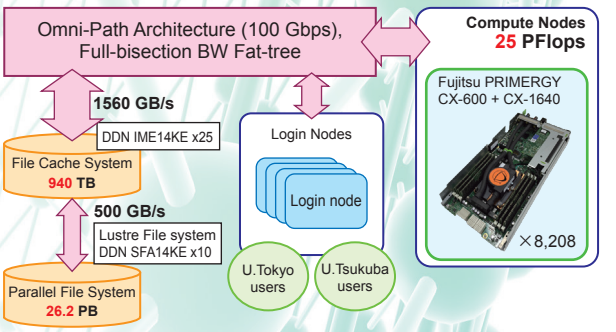
Power consumption			4.2 MW (including cooling)
Number of racks			102
Cooling system	Compute node	Type	Warm-water cooling Direct cooling (CPU) Rear door cooling (except CPU)
		Facility	Cooling tower & Chiller
	Others	Type	Air cooling
		Facility	PAC



Full bisection bandwidth Fat-tree by Intel® Omni-Path Architecture



(Copyright© 2016, Intel Corporation)



Nov. 2016 Rank		
TOP500	#6 (#1 in Japan)	13,554.6 TFLOPS
Green500	#6 (#2 in Japan)	4985.7 MFLOPS/W
HPCG	#3 (#2 in Japan)	0.3855 PFLOPS

SCD/ITC

Supercomputing Division,
Information Technology Center, The University of Tokyo

東京大学情報基盤センター
INFORMATION TECHNOLOGY CENTER, THE UNIVERSITY OF TOKYO
2-11-16 Yayoi, Bunkyo, Tokyo 113-8658, JAPAN TEL:03-5841-2710 FAX:03-5841-2708(G3) http://www.itc.u-tokyo.ac.jp/en/

2017.06