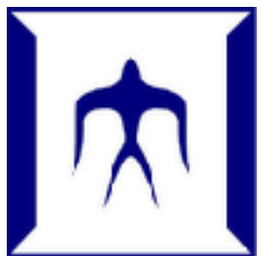
High Performance Scientific Computing

High Performance Scientific Computing 2018
Rio Yokota
rioyokota@gsic.titech.ac.jp



Course schedule

	Course schedule	Required learning
Class 1	How to use TSUBAME	Login to Tokyo Tech's supercomputer TSUBAME and learn how to use libraries and the job scheduler
Class 2	Shared memory parallelization	Use pthreads and OpenMP to achieve shared memory parallelization
Class 3	Distributed memory parallelization	Use MPI to achieve distributed memory parallelization
Class 4	SIMD parallelization	Use SSE, AVX, and AVX512 to achieve SIMD vectorization
Class 5	GPU programming	Use OpenACC, CUDA, and OpenCL to program GPUs
Class 6	Multi-GPU programming	Combine CUDA and MPI to use multiple GPUs on TSUBAME
Class 7	Cache blocking	Use BLISLAB and CUBLAS as an example to practice cache blocking
Class 8	Numerical libraries	Understand how LAPACK, SCALAPACK, and FFTW work, and learn to use them appropriately

Course schedule

Class 9	Fast linear solvers	Understand how to choose the appropriate solvers in PETSc and Trilinos
Class 10	I/O libraries	Use NetCDF, HDF5, MPI-IO to read and write on large parallel file systems
Class 11	Parallel debugger	Use CUDA-GDB, Valgrind, TotalView to debug parallel code
Class 12	Parallel profiler	Use gprof, VTune, PAPI, Tau, Vampire to profile parallel code
Class 13	Performance primitives	Learn how to use performance primitives such as ModernGPU and MapReduce
Class 14	Graph partitioning	Use METIS and ParMETIS to partition a large graph in parallel
Class 15	Deep Learning	Use ChainerMN to train a large neural network on a parallel computer

How to use TSUBAME

High Performance Scientific Computing 2018
Rio Yokota

How to create an account

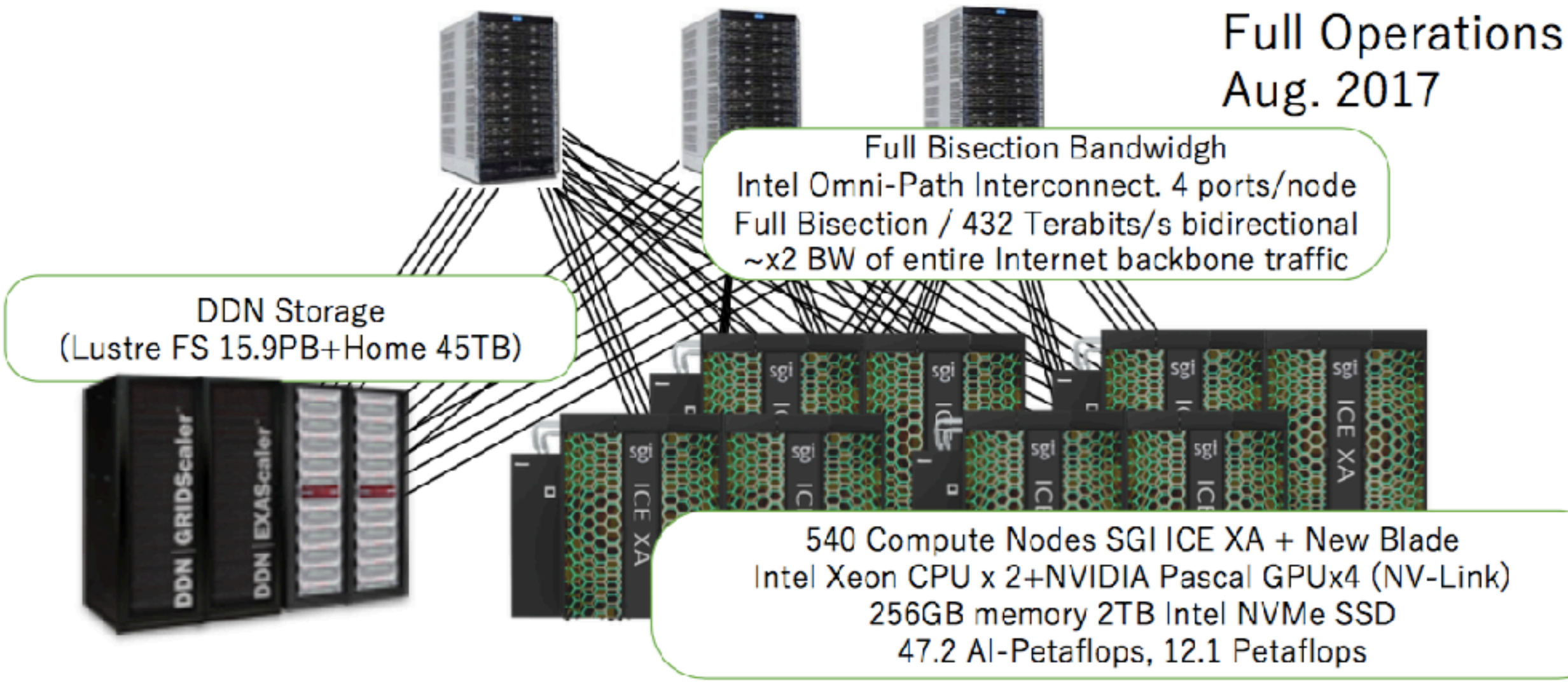
1. Login to Tokyo Tech Portal
2. Click “TSUBAME 3 Portal”
3. Create new account

Uploading your public key

1. ssh-keygen (this generates id_rsa.pub)
2. Login to TSUBAME 3 Portal
3. Click “Register SSH public key”
4. Copy & Paste your public key

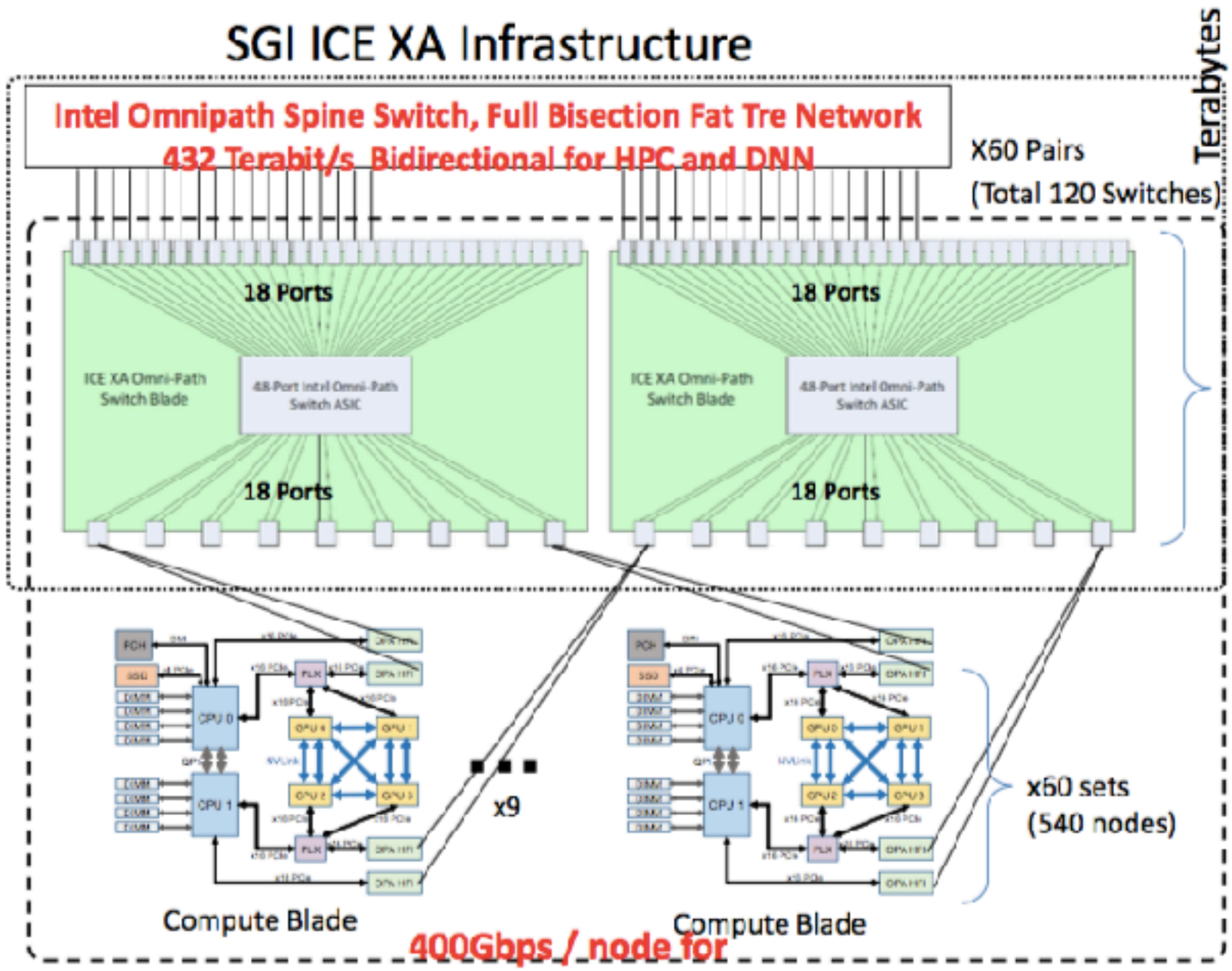
Login and setup

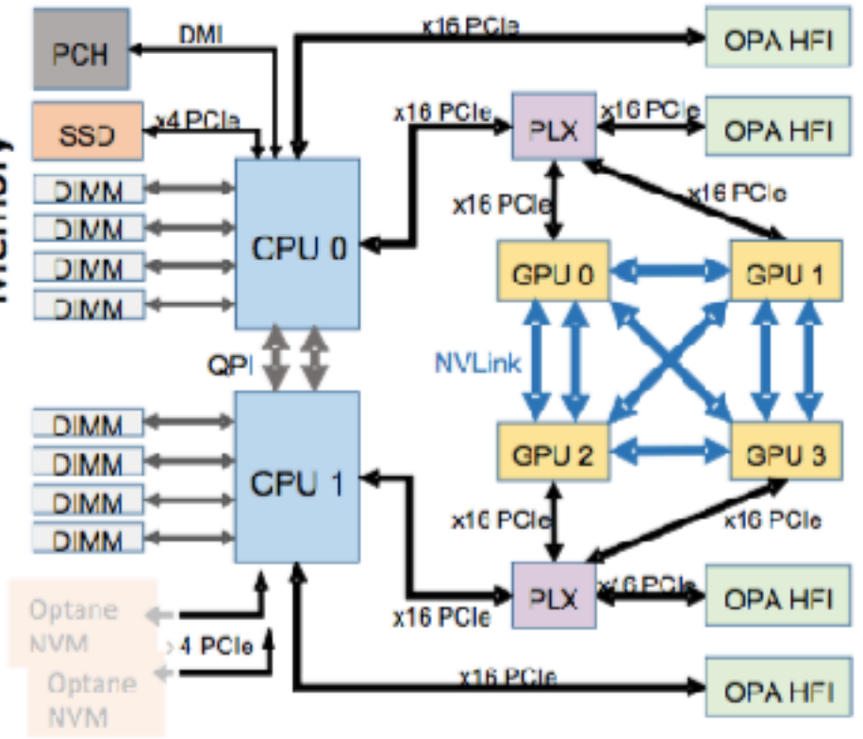
1. `ssh username@login.t3.gsic.titech.ac.jp`
2. `module avail`
3. `module load cuda intel intel-mpi`
4. `module list`



System configuration

TSUBAME3.0 Compute Node SGI ICE-XA, a New GPU Compute Blade Co-Designed by SGI and Tokyo Tech GSIC





Ultra high performance & bandwidth "Fat Node"

- High Performance: 4 SXM2(NVLink) NVIDIA Pascal P100 GPU + 2 Intel Xeon 84 AI-TFLops
- High Network Bandwidth – Intel Omnipath 100GBps x 4 = 400Gbps (100Gbps per GPU)
- High I/O Bandwidth - Intel 2 TeraByte NVMe
  - > 1PB & 1.5~2TB/s system total
  - Future Octane 3D-Xpoint memory Petabyte or more directly accessible
- Ultra High Density, Hot Water Cooled Blades
  - 36 blades / rack = 144 GPU + 72 CPU, 50-60KW, x10 thermals c.f. IDC

Job script

job.sh

```
#!/bin/sh
#$ -cwd
#$ -l f_node=1
#$ -l h_rt=0:01:00
./a.out
```

test.c

```
#include <stdio.h>

int main() {
  printf("hi\n");
}
```

1. `gcc test.c`
2. `qsub -g tga-hpc-lecture job.sh`
3. `qrsh -g tga-hpc-lecture -l f_node=1 -l h_rt=0:01:00`
4. `qstat`
5. `qdel`

```
error file
output file
```

```
 r   running
qw   waiting in the queue
 h   on hold
 d   deleting
 t   transition during job-start
 s   suspended
 S   suspended by the queue
 T   has reached the limit of the tail
 E   error
```