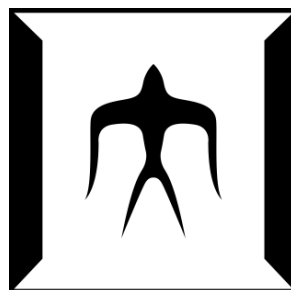


TSUBAME3.0利用講習会

www.t3.gsic.titech.ac.jp



令和5年度版(2023/11/14)

東京工業大学

学術国際情報センター

共同利用推進室

Copyright (C) 2010-2024 GSIC All Rights Reserved.

CONTENTS

- 歴史・概要
- ハードウェア・ソフトウェア仕様
- 利用開始とログイン
- 利用可能アプリケーション～module～
- 資源タイプ(計算ノード)
- ジョブの実行とスクリプト
- TSUBAMEポイントと課金
- リンカー覧

TSUBAMEの歴史

TSUBAMEの変遷

2006年	TSUBAME1.0	85TFlops/ 1.1PB	アジアNo1「みんなのスパコン」
2007年	TSUBAME1.1	100TFlops/ 1.6PB	ストレージ・アクセラレータ増強
2008年	TSUBAME1.2	160TFlops/ 1.6PB	GPUアクセラレータ680枚増強 (S1070)
2010年	TSUBAME2.0	2.4PFlops/ 7.1PB	日本初のペタコン (M2050)
2013年	TSUBAME2.5	5.7PFlops/ 7.1PB	GPUをアップグレード (K20X)
2017年	TSUBAME3.0	12PFlops/16.0PB	Green500 世界1位！ (P100)
2024年	TSUBAME4.0	67PFlops/44.2PB	来春稼働予定？ (H100)

共同利用推進室の事業 TSUBAME学外利用の窓口として

- 2007年 文科省 先端研究施設共用イノベーション創出事業(無償利用)
- 2009年 TSUBAME共同利用開始(有償利用)
- 2010年 文科省 先端研究施設共用促進事業、JHPCN 開始
- 2012年 HPCI(革新的ハイパフォーマンス・コンピューティング・インフラ)開始
- 2013年 文科省 先端研究基盤共用・プラットフォーム形成事業
- 2016年 東京工業大学 学術国際情報センター 自主事業化、
HPCI 産業利用(実証利用、トライアル・ユース)開始

利用区分 / 年度		2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	合計	
学術利用	HPCI	-	-	-	-	-	6	5	10	14	5	9	12	16	14	8	10	109	
	JHPCN	-	-	-	4	6	5	11	10	10	12	11	15	14	8	7	6	119	
	有償利用	-	-	1	4	9	14	17	22	23	25	23	27	25	28	30	28	276	
産業利用	無償利用/HPCI		11	15	15	8	10	12	21	17	13	15	8	3	3	1	1	1	154
	有償利用	公開	-	-	3	6	7	9	8	10	8	8	5	6	4	5	2	1	82
		非公開	-	-	2	7	6	4	10	12	10	13	16	19	19	20	14	12	164
合計		11	15	21	29	38	50	72	81	78	78	72	82	81	76	62	58	904	

利用区分

- 有償利用

共同利用：学術利用（成果公開のみ）

共同利用：産業利用（成果公開・成果非公開）

- 無償利用

HPCI/JHPCN による利用（学術・産業）

利用区分	利用者	制度		募集時期	申請および審査	成果	料金（税込）
学術利用	他大学 または 研究機関等	HPCI		年1回 10月頃	HPCI運用事務局 （高度情報科学技術研究機構）	公開	無償
		JHPCN		年1回 1月頃	JHPCN拠点事務局 （東京大学 情報基盤センター）	公開	無償
		TSUBAME学術利用		随時 募集中	東京工業大学 学術国際情報センター	公開	1口：110,000 円
産業利用	民間企業	HPCI	産業課題	年1回 10月頃	HPCI運用事務局 （高度情報科学技術研究機構）	公開	無償
			産業試行課題	随時 募集中			
		TSUBAME産業利用		随時 募集中	東京工業大学 学術国際情報センター	公開	1口：110,000 円
						非公開	1口：330,000 円

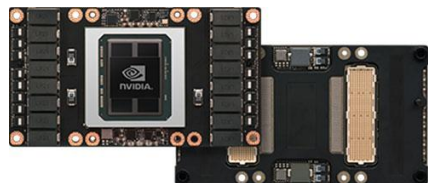
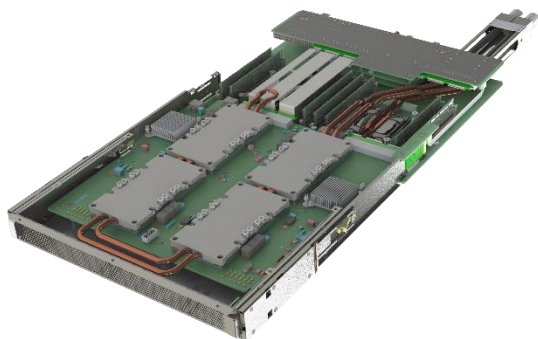
※ 2017年8月より運用開始
Green500 世界1位(2017/6)
Top500 国内10位 (2023/11)

TSUBAME3.0 概要

Compute Node

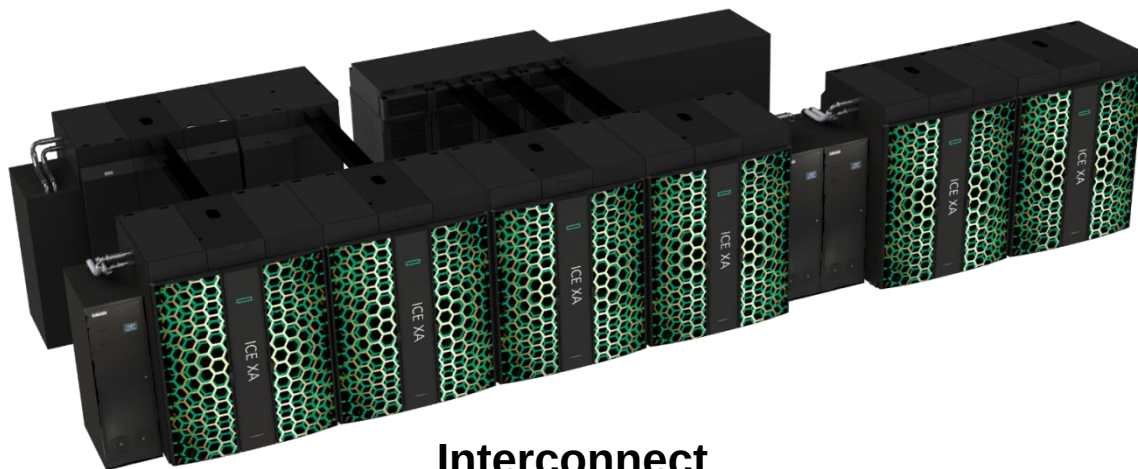
CPU: Intel Xeon E5-2680v4 (14core) × 2
GPU: NVIDIA Tesla P100 × 4

Performance: 22.5 TFLOPS
Memory: 256 GB(CPU)
64 GB(GPU)



System

540 nodes: 15120 CPU cores, 2160 GPUs
Performance: 12.15 PFLOPS



Interconnect

Intel Omni-Path HFI 100Gbps × 4
Topology: Fat-Tree

Operating System

SUSE Linux Enterprise Server 12 SP5

Job Scheduler

UNIVA Grid Engine
(Altair Grid Engine)

<https://www.gsic.titech.ac.jp/sites/default/files/T3SpecJ.pdf>

GPUスパコン計算ノード比較

資源提供機関のハードウェア情報 (Xeon+GPU)

資源提供 機関	計算資源名 ／機種名	システム全体		ノード単体			ノード間 ネットワーク	
		演算性能	ノード数	プロセッサ	演算性能	メモリ		
Xeon+GPU(NVIDIA)	産総研	ABCI ※1	56.5 PF	1208	Xeon Gold 6148 (20コア) x2 +Tesla V100 x 4	34.2 TF	384GiB	InfiniBand EDR
		※1 有償利用資源			Xeon Platinum 8360Y(Ice Lake) x2 +NVIDIA A100 x8	161 TF	512GiB	InfiniBand HDR
	東工大	TSUBAME3.0	12.15 PF	540	Xeon E5-2680v4, 2.4GHz(14コア)x2 +Tesla P100 x4, NVLink	22.5 TF	256GiB	Intel Omni-Path 100Gbps x4
	名大	「不老」TypeⅡ サブシステム CX2570 M5	7.49 PF	221	Xeon Gold 6230, 2.10-3.90 GHz(20コア)x2 +Tesla V100 x4	33.88 TF	384GiB	InfiniBand EDR 100 Gbps x2
	東大	Wisteria/BDEC (Aquarius : データ学習ノード群)	7.2 PF	45	Xeon Platinum 8360Y(Ice Lake,2.4GHz,36コア)x2 NVIDIA A100 x8	160 TF	512GiB	InfiniBand HDR (200Gbps) x4
	筑波大	Pegasus	6.1 PF	120	Intel Xeon +NVIDIA H100,PCIe	51.2 TF	128GiB	InfiniBand NDR200 (200Gbps)
	筑波大	Cygnus (Deneb node)	2.4 PF	80	Xeon Gold 6126(Skylake-SP,2.6GHz,12コア)x2 +Tesla V100x4,PCIe	30 TF	192GiB	InfiniBand HDR100 4x
	九大	ITO サブシステムB CX2570 M4 (2023/4～2024/2)	3.05 PF	128	Xeon Gold 6140 (Skylake-SP,2.3GHz,18コア)x2 +Tesla P100x4, NVLink	23.85 TF (2.65TF + 5.3TF*4)	384GiB	InfiniBand EDR 4x 100 Gbps

https://www.hpci-office.jp/materials/r05a_boshu_setsumeikai_hpci.pdf#page=6 より引用

利用開始とログイン

2008年



Tesla S1070 (Tesla GT200)
on TSUBAME1.2

2010年



Tesla M2050 (Fermi)
on TSUBAME2.0

2013年

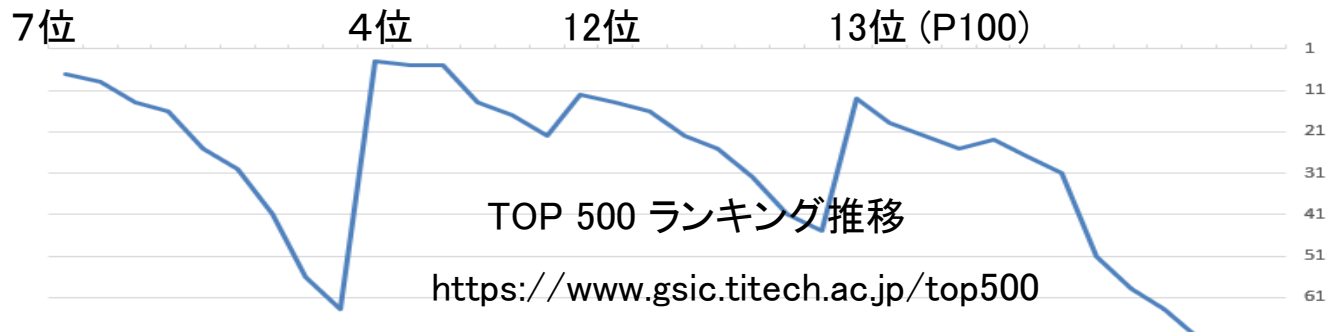


Tesla K20X (Kepler)
on TSUBAME2.5

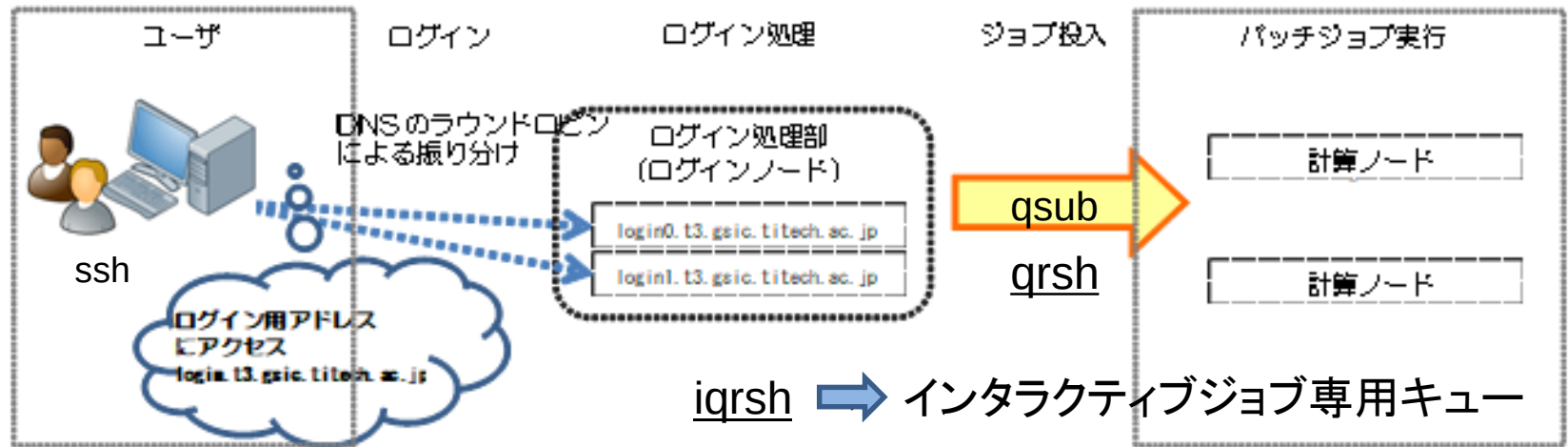
2017年



Tesla P100 (Pascal)
on TSUBAME3.0



計算機へのログイン



- SSHログイン: `ssh <username>@login.t3.gsic.titech.ac.jp`
 - どちらかのログインノードに振り分けられる
 - 原則、公開鍵認証方式のみ(パスワードは不可)
 - ログインノードではファイル編集、軽いコンパイルなど
 - GPU なし (module load cuda でCUDAコンパイルは可能)
 - HPCI ユーザーも同じログインノードを使用 (gsi ssh)
 - GUI (X Window) を利用する場合は `ssh -YC` にてログインする

TSUBAME3ポータル

- アカウント作成方法 (以下のいずれか)
 - 東工大学内では、東工大ポータル → TSUBAMEポータル
 - TSUBAMEポータル <https://portal.t3.gsic.titech.ac.jp/ptl/>
- 学外の方のアカウントは共同利用推進室にて発行
アカウント発行に際し本人のメールアドレスが必要
TSUBAME3.0ポータルにて
 - 公開鍵の設定 (ssh-keygen, Tera Term, PuTTY)

※ Windowsで利用可能なSSHクライアント https://www.t3.gsic.titech.ac.jp/windows_sshclients
 - パスワードの設定 (ログインパスワード)
 - ジョブ情報の確認 (ポイント消費など)
 - <https://www.gsic.titech.ac.jp/sites/default/files/Portal2022v1.pdf>

有償サービス

- 課題単位でグループを作成
課題採択: **TSUBAMEグループ** を割り当てる
- TSUBAMEポイントによるプリペイド従量制
 - 1ノード × 1秒 = 1 TSUBAMEポイント
 - 1口 = 1000ノード時間 = 1000 × 3600 TSUBAMEポイント
ポイントを消費し口数が不足した場合は追加購入可能。
- グループ共有の高速ストレージ (Lustre)
 - /gs/hs0/グループ名, /gs/hs1/グループ名
 - TB × 月単位 (1TB/月 36,000 TSUBAMEポイント)
 - ホームディレクトリ (25GB) の利用は無償

TSUBAME3.0ソフトウェア

- OS: SUSE Linux Enterprise Server (SLES) 12 SP5
 - Docker コンテナ (<https://helpdesk.t3.gsic.titech.ac.jp/manuals/handbook.ja/jobs/#container>)
 - Singularity 対応 (<https://helpdesk.t3.gsic.titech.ac.jp/manuals/handbook.ja/freesoft/#singularity>)
- スケジューラ: Univa (Altair) Grid Engine (8.6.11)
- コンパイラ: gcc (4.8.5), Intel (23.0) oneAPI, nvhpc (23.1)
- MPI: OpenMPI, Intel MPI, SGI MPT (Message Passing Toolkit)
- CUDA 11.0.3 (default) 12.1.0 が利用可能
- プログラミングツール: Intel Vtune, PAPI, Arm Forge...
- 多種ISVアプリ(後述)

moduleコマンド(後述) による切り替え

moduleコマンドについて

- 利用するソフトウェアに関する環境設定を、
module コマンドを用いて設定する
 - 例: `module load intel` → Intelコンパイラ
 - `module load intel/21.5.0` のようにバージョン指定も可能
 - 例: `module load python-extension/3.10.2`
- 用意されているモジュールの一覧: `module avail`
- モジュールによっては、さらに依存モジュールのロードが必要。現在のモジュールは `module list` で確認する
 - 例: gromacsモジュールはintel-mpiモジュールに依存
- moduleコマンド自体が動かないとき(後述)は
 - `./etc/profile.d/modules.sh` ←先頭は「ドット・スペース」

現在インストールされているモジュール(1)

コンパイラ、MPI、開発ツール 関連のモジュール。\$ module available

必要な環境に応じた module を load し、プログラムをコンパイルする。

コンパイラ: gcc 12.2.0、Intel 23.0.0、nvhpc 23.1、clang 14.0.6

MPI: OpenMPI、Intel MPI、SGI MPI (MPT) OpenACC は nvc -acc にて利用

例1) gcc + OpenMPI の場合 : module load cuda openmpi

例2) Intel + IntelMPI の場合 : module load intel cuda intel-mpi

/apps/t3/sles12sp2/modules/modulefiles/compiler					
cuda/10.0.130	cuda/11.6.1	intel/16.0.4.258	intel/21.1.1	nvhpc/22.2	pgi/20.1(default)
cuda/10.1.105	cuda/11.6.2	intel/17.0.4.196	intel/21.2.0	nvhpc/23.1(default)	pgi-llvm
cuda/10.2.89	cuda/12.1.0	intel/17.0.5.239	intel/21.5.0	pgi/17.10	pgi-nollvm
cuda/11.0.194	cuda/8.0.44	intel/18.0.1.163	intel/23.0.0(default)	pgi/17.5	
cuda/11.0.3(default)	cuda/8.0.61	intel/19.0.0.117	nvhpc/20.9	pgi/18.1	
cuda/11.1.1	cuda/9.0.176	intel/19.1.0.166	nvhpc/21.2	pgi/18.7	
cuda/11.2.146	cuda/9.1.85	intel/19.1.1.217	nvhpc/21.3	pgi/19.1	
cuda/11.5.0	cuda/9.2.148	intel/19.1.3.304	nvhpc/21.7	pgi/19.7	
/apps/t3/sles12sp2/modules/modulefiles/mmpi					
ibmmpi/v9.01.04.03		intel-mpi/21.2.0		openmpi/2.1.2-opa10.9-t3-thread-multiple	
intel-mpi/17.3.196		intel-mpi/21.5.1		openmpi/2.1.2-opa10.9-thread-multiple	
intel-mpi/17.4.239		intel-mpi/21.8.0(default)		openmpi/2.1.2-pgi2017	
intel-mpi/18.1.163		mpt/2.16		openmpi/2.1.2-pgi2018	
intel-mpi/19.0.117		openmpi/1.10.2-pgi2017		openmpi/2.1.2-thread-multiple	
intel-mpi/19.6.166		openmpi/2.1.1		openmpi/3.1.4-opa10.10(default)	
intel-mpi/19.7.217		openmpi/2.1.2		openmpi/3.1.4-opa10.10-t3	
intel-mpi/19.9.304		openmpi/2.1.2-opa10.9		openmpi/4.0.7	
intel-mpi/21.1.1		openmpi/2.1.2-opa10.9-t3		openmpi/4.0.7-t3	
/apps/t3/sles12sp2/modules/modulefiles/tools					
forge/22.0.2	intel-ins/21.2.0	intel-itac/21.1.1		intel-vtune/20.3.0.612611	
forge/22.1.3(default)	intel-ins/22.0.0	intel-itac/21.2.0		intel-vtune/21.1.1	
intel-ins/17.1.3.510645	intel-ins/23.0.0(default)	intel-itac/21.5.0		intel-vtune/21.2.0	
intel-ins/17.1.4.527006	intel-itac/17.3.030	intel-itac/21.8.0(default)		intel-vtune/22.0.0	
intel-ins/18.1.1.535159	intel-itac/17.4.034	intel-vtune/17.4.0.518798		intel-vtune/23.0.0(default)	
intel-ins/19.0.0.569751	intel-itac/18.1.017	intel-vtune/17.5.0.526192		papi/5.5.1	
intel-ins/20.0.0.603904	intel-itac/19.0.018	intel-vtune/18.1.0.535340		perfboost/2.16	
intel-ins/20.1.0.604266	intel-itac/20.0.015	intel-vtune/19.0.2.570779		perfsuite/1.1.4	
intel-ins/20.3.0.604771	intel-itac/20.1.024	intel-vtune/20.0.0.605129			
intel-ins/21.1.1	intel-itac/20.3.036	intel-vtune/20.1.0.607630			

※ gcc, gfortran 等は module load しなくても利用できます。gcc は 10.2.0、11.2.0 も利用可能です。

※ 2022年春のメンテナンス後のバージョン一覧 <https://www.t3.gsic.titech.ac.jp/changes2022>

現在インストールされているモジュール(2)

アプリケーションのモジュールの一覧 \$ module available 続き

※一部のISVのプログラムは学外の方にはご利用になれません。

```
----- /apps/t3/sles12sp2/modules/modulefiles/isv -----
abaqus/2017                comsol/53                gaussian16/B01_cpu        marc_mentat/2021.2        nastran/2018.1
abaqus/2017_explicit        comsol/53a              gaussian16/B01_gpu        marc_mentat/2021.4        nastran/2018.2
abaqus/2020                comsol/53a_u1           gaussian16/C01_cpu        marc_mentat/2022.1        nastran/2018.2.1
abaqus/2021                comsol/53a_u2           gaussian16/C01_gpu (default) marc_mentat/2022.3        nastran/2019.0
abaqus/2022                comsol/53a_u3           gaussian16/C01_nbo7       marc_mentat/2022.4        nastran/2020.0
abaqus/2023                comsol/53a_u4           gaussian16/C01_nbo7_cpu   mathematica/11.1.1        nastran/2021.0
actran/19.1                comsol/54               gaussian16/C01_nbo7r27    mathematica/11.2.0        nastran/2021.1
actran/2020                comsol/54_u1            gaussian16/C01_nbo7r27_cpu mathematica/11.3.0        nastran/2021.2
actran/2021                comsol/54_u3            gaussian16_linda/A03      mathematica/12.0.0        nastran/2021.4
actran/2021.1              comsol/54_u4            gaussian16_linda/B01      mathematica/12.1.0        nastran/2022.1
actran/2022.1              comsol/55               gaussview/6               mathematica/12.1.1        nastran/2022.3
amber/16                   comsol/55_u1            gaussview/6.1            mathematica/12.2.0        nastran/2022.4
amber/16_cuda              comsol/55_u2            lsdyna/R10.1.0           mathematica/13.1.0        patran/2017.0.2
amber/16up10               comsol/55_u3            lsdyna/R10.2.0           mathematica/13.2.1 (default) patran/2018.0
amber/16up10_cuda          comsol/56               lsdyna/R11.1.0           matlab/R2017a            patran/2019.0
amber/16up12_cuda          comsol/56_u1            lsdyna/R9.1.0            matlab/R2017a_u3         patran/2019fp1
amber/18up12               comsol/56_u2            lsdyna/R9.3.1            matlab/R2017b            patran/2020.0
amber/18up17               comsol/60_u0            lsprepost/4.3            matlab/R2017b_u9         patran/2021.0
amber/18up5                comsol/60_u1            lsprepost/4.5            matlab/R2018a            patran/2021.1
amber/20up0                comsol/60_u2            maple/2019.1              matlab/R2018a_u6         patran/2021.2
amber/20up08 (default)     comsol/61               maple/2019.2.1            matlab/R2018b            patran/2021.4
amber/22up0                comsol/61_u2 (default)   maple/2020.1              matlab/R2019a            patran/2022.1
amber/22up0_at22up3        dytran/2017             maple/2021.1              matlab/R2019b            patran/2022.3
ansys/R20.1                dytran/2018             marc_mentat/2017          matlab/R2020a            patran/2022.4
ansys/R20.2                dytran/2021             marc_mentat/2017.1        matlab/R2020b            schrodinger/2020-2 (def.)
ansys/R21.1                dytran/2021.1           marc_mentat/2018.1        matlab/R2021a            schrodinger/2021-1
ansys/R22.1                dytran/2021.2           marc_mentat/2019          matlab/R2021b            schrodinger/Feb-17
ansys/R22.2                dytran/2022.1           marc_mentat/2019.1        matlab/R2022a            nastran/2017.1
ansys/R23.1 (default)     dytran/2022.2           marc_mentat/2020          matlab/R2022b            nastran/2017.1
avs/8.4                   gaussian16/A03           marc_mentat/2020.1        matlab/R2023a (default)
avs/8.5                   gaussian16/B01           marc_mentat/2021.1
```

※ アプリケーションのバージョンアップによりプログラムのバージョンが更新されていることがありますのでご注意ください。

現在インストールされているモジュール(3)

アプリケーションのモジュールの一覧 \$ module available 続き ※フリーソフトウェア

```
----- /apps/t3/sles12sp2/modules/modulefiles/free -----
a2ps/4.14
alphafold/2.0.0(default)
alphafold/2.1.1
alphafold/2.2.0
alphafold/2.3.0
apptainer/1.1.6
autoconf/2.71
automake/1.16.4
caffe/1.0
chainer/4.3.0
chainer/5.2.0
clang/10.0.0
clang/11.1.0
clang/13.0.1
clang/14.0.6(default)
clang/9.0.0
cmake/3.21.3
code-server/3.9.0
cp2k/2023.1
cp2k/4.1
cp2k/4.1-libint
cp2k/5.1_cuda
cp2k/5.1-libint
cp2k/7.1.0
cp2k/8.1.0
cp2k/9.1.0
cudnn/5.1
cudnn/6.0
cudnn/7.0
cudnn/7.1
cudnn/7.3
cudnn/7.4
cudnn/7.6
cudnn/8.1
cudnn/8.3
cudnn/8.8.1
deepmd-kit/2.1.5
dmtcp/2.5.2
emacs/28.2
ffmpeg/4.2.2
fftw/2.1.5
fftw/3.3.10
fftw/3.3.6
fftw/3.3.8
fftw/3.3.9
firefox/76.0.1
gamess/apr202017r1
gcc/10.1.0
gcc/10.1.0-cuda
gcc/10.2.0
gcc/10.2.0-cuda
gcc/11.2.0
gcc/11.2.0-cuda
gcc/12.2.0(default)
gcc/12.2.0-cuda
gcc/6.5.0
gcc/8.3.0
gcc/8.3.0-cuda
gimp/2.10.4
gimp/2.8.22
gnuplot/5.0.6
gnuplot/5.2.4
gnuplot/5.4.1
gnuplot/5.4.3
gromacs/2016.3
gromacs/2018.1
gromacs/2019.4
gromacs/2020.4
gromacs/2021.3
gromacs/2022
gromacs/2023(default)
gromacs/4.6.7
gromacs/master-20230413
gromacs-deepmd/2020.2
hadoop/2.8.0
hdf5/1.10.1
hdf5/1.12.0
hdf5-parallel/1.10.5
hdf5-parallel/1.12.0
help2man/1.40
hpci/1.0
imagemagick/7.0.11-1
imagemagick/7.0.6
intel-python/2.7.14(default)
intel-python/3.6.5
jupyterlab/2.1.0(default)
jupyterlab/2.1.0-py383
jupyterlab/3.0.9
jupyterlab/3.0.9-py383
lammps/10feb2021
lammps/23jun2022_u1
lammps/23jun2022_u2
lammps/23jun2022_u3(default)
lammps/29sep2021_u2
lammps/31mar2017
lammps/3mar2020
lammps/9oct2020
llvm/10.0.0
llvm/11.1.0
llvm/3.9.1
llvm/9.0.0
mesa/13.0.3
mpifileutils/0.9.1
namd/20220414
namd/2.12
namd/2.12-20180711
namd/2.13
namd/2.13-20191210
namd/2.14
namd/2.14b1
namd/2.14b2
namd/2.15alpha1
namd/3.0b2
nccl/1.3.4
nccl/2.1
nccl/2.12.7
nccl/2.1.4
nccl/2.17.1(default)
nccl/2.2.13
nccl/2.4.2
nccl/2.8.4
netcdf-parallel/4.7.0
netcdf-parallel/4.7.4
ninja/1.10.0
novnc/1.3.0
openfoam/10.0
openfoam/4.1
openfoam/6.0
openfoam/7.0
openfoam/8.0
openfoam/9.0
openfoam-esi/v1906
openfoam-esi/v1912
openfoam-esi/v2012
openfoam-esi/v2112
openfoam-esi/v2212
openjdk/1.8.0.242
parallel/20211022
paraview/0_5.2.0
paraview/5.0.1
paraview/5.10.0
paraview/5.10.0-egl
paraview/5.11.0(default)
paraview/5.11.0-egl
paraview/5.4.0
paraview/5.8.0
paraview/5.8.0-egl
paraview/5.9.0
paraview/5.9.0-egl
petsc/3.14.5/complex
petsc/3.14.5/real
petsc/3.7.6/complex
petsc/3.7.6/real
petsc/3.9.3/complex
petsc/3.9.3/real
php/7.1.6
pov-ray/3.7.0.3
python/3.10.2
python/3.11.2(default)
python/3.6.5
python/3.8.3
python/3.9.2
python-extension/2.7(def)
python-extension/3.4
qt/5.12.10
quantumespresso/6.8
quantumespresso/7.1
r/3.4.1
singularity/3.6.3
singularity/3.6.4
singularity/3.7.1
singularity/3.8.6
singularity/3.8.7(default)
tensorflow/1.12.0
tensorflow/1.9.0
tensorflow/2.4.1
tensorflow/2.8.0
tensorrt/6.0.1.8
tensorrt/7.2.3.4
texlive/20170704
tgif/4.2.5
tinker/8.10.1
tinker/8.10.5
tinker/8.1.2
tinker/8.7.2
tinker/8.8.3
tmux/2.5
tmux/2.7
tmux/3.1
tmux/3.2
tmux/3.3(default)
turbovnc/2.2.2
v8/6.7.17-7
visit/2.12.3
vmd/1.9.3
vtk/6.1.0
vtk/8.0.0(default)
xpdf/3.04
```

現在インストールされているモジュール(4)

HPCIで整備されたアプリケーションの一覧

HPCIシステムへの国プロソフト利用環境整備プロジェクトにより整備されたソフトウェアです。

これらのソフトウェアはHPCIユーザ以外の方もご利用いただけますが東工大ではサポートしておりません。

HPCIユーザはHPCIヘルプデスクへお問い合わせください。

<https://www.hpci-office.jp/pages/helpdesk/>

HPCI以外のユーザについては各ソフトウェアのコミュニティへ直接お問い合わせください。

/apps/t3/sles12sp2/modules/modulefiles/hpci-apps			【国プロアプリ】	
abinit-mp/1.10	frontistr/5.0b	hphi/3.0.0	ntchem2013/11.1	salmon/1.0.0
abinit-mp/1.15	frontistr/5.1.1	hphi/3.1.2	ntchem2013/12.1.1 (default)	salmon/1.2.0
abinit-mp/1.22 (default)	frontistr/5.2 (default)	hphi/3.3.0	openmx/3.8.3	salmon/1.2.1
abinit-mp/2.4	genesis/1.3.0_cpu	hphi/3.4.0	openmx/3.8.5	salmon/2.0.0
akaikkr/cpa2002v010_cpu	genesis/1.3.0_gpu	hphi/3.5.0 (default)	openmx/3.9.2 (default)	salmon/2.0.2 (default)
akaikkr/cpa2002v010_gpu	genesis/1.5.1_cpu	modylas/1.0.4	phase0/2018.01.01	smash/2.2.0
alamode/1.3.0	genesis/1.5.1_gpu	modylas/1.1.0b (default)	phase0/2019.02	smash/3.0.0 (default)
frontflowblue/8.1	genesis/1.7.0_cpu	mvmc/1.2.0	phase0/2021.01 (default)	
frontistr/5.0a	genesis/1.7.0_gpu (default)	ntchem2013/10.1	phonopy/2.12.0	

詳細につきましてはこちらをご参照ください。

<https://www.t3.gsic.titech.ac.jp/hpci-apps>

https://www.hpci-office.jp/pages/appli_software

その他、使用実績はあるがサポートしていないソフトウェア:

VASP は大学でサイトライセンスを取得できないため、所属組織にてライセンスを取得する必要があります。

https://www.vasp.at/registration_form/ Makefile = <https://www.vasp.at/wiki/index.php/Makefile.include>

Licenses are only issued to well defined research groups under the direction of a single chair, professor or working group leader in one single physical location.

ジョブの実行についての概要

- ジョブスケジューラは UNIVA (Altair) Grid Engine
- ジョブの性質にあわせて、資源タイプを選択
 - f_node (フル), h_node (ハーフ), q_node (クォーター)...
 - s_gpu、q_core、s_core ...
- ジョブの投入は **qsub** コマンドを用いる
 - 「ジョブスクリプト」を用意する (vi, vim, emacs など...)
- 予約キューの利用
 - 1時間、1ノード単位からの予約、24時間以上利用可能
- ssh による計算ノードへの直接ログイン
 - qsub で割り当てた f_node のみ直接 ssh でログイン可能

資源タイプ一覧

資源タイプ	タイプ名	CPUコア数	メモリ(GB)	GPU数	課金係数
F	f_node	28	235	4	1.00
H	h_node	14	120	2	0.50
Q	q_node	7	60	1	0.25
I	共有対話型	7	60	1	0.20
G	s_gpu	2	30	1	0.20
C4	q_core	4	30	N/A	0.20
C1	s_core	1	7.5	N/A	0.06

- MPIジョブ等では、f_node=4、q_node=10 のように1ジョブで複数資源を利用可能
 - 異種混在は不可、最大で72ノード割り当て可能
 - 520ノードから各資源タイプを割り当てる
 - 最大利用可能資源量は→ <https://www.t3.gsic.titech.ac.jp/resource-limit>

計算ノードのインタラクティブ利用

- 計算ノードにて対話的な実行を試したい場合など、
インタラクティブな利用が可能（-l = ハイフン 小文字のエル）

```
qrsh -l [資源タイプ] -l h_rt=[利用時間] -g [グループ]
```

- 例: `qrsh -l q_node=1 -l h_rt=0:10:00`（お試し利用）

→ 計算ノードが割り当てられ、Linuxコマンドが実行できる。

※ この例では `q_node` なので、7コア1GPU 利用可能。

- 10分以上利用する時は、`-g` オプションにてTSUBAMEグループを指定する。 `h_rt` には適切な wall time を設定する。

- 例: `qrsh -l f_node=2 -l h_rt=1:00:00 -g tgx-23lxx`

※ 複数ノードを割当てた際は `cat $PE_HOSTFILE` にて計算ノードを確認できる

- `f_node` 以外を `qrsh` で割り当てた場合もX転送が可能。

例: `qrsh -l s_core=1,h_rt=0:10:00`

計算ノードのインタラクティブ利用

- インタラクティブジョブ専用キューによる利用

q_node 相当の資源を共有し対話的に利用可能

```
iqrsh -l h_rt=[利用時間] -g [TSUBAMEグループ]
```

- 7コア、1GPU、メモリ60GB を、最大7名で共有し対話的に利用する。
- 実行可能な資源がない場合、ジョブは投入できない。
- メモリの内容は混雑状況に応じて SSD にスワップされる。
- 1ユーザーあたり一度に実行可能なジョブは1ジョブのみ。
- 最大利用時間は24時間。10分以内の無償利用はなし。
- ローカルスクラッチ領域(SSD)も共有されている。
- Singularity によるコンテナジョブは利用可能。
- デバッガや可視化ツール、Jupyter Lab 等の対話型利用を想定。
- プロセッサを占有する計算は通常の計算ノードを利用すること。

ジョブの投入の概要

1. ジョブスクリプトの作成

- ジョブの最長実行時間は24:00:00(延長なし)
- お試しだと 00:10:00 (10分間 2ノードまで無料)
- 24時間以上実行する場合は予約システムを利用

2. qsub を利用しジョブを投入

3. qstat を使用しジョブの状況を確認

4. qdel にてジョブをキャンセル

5. ジョブの結果を確認

※詳細はこちら → <https://helpdesk.t3.gsic.titech.ac.jp/manuals/handbook.ja/jobs/#jobscript>

Step 1. ジョブスクリプト

- 下記のような構成のファイル(ジョブスクリプト)をテキストエディタなどで作成 (vi など TSUBAME上で編集)
 - 拡張子は .sh

```
#!/bin/sh
```

```
#$ -cwd
```

```
#$ -l [資源タイプ] =[個数]
```

```
#$ -l h_rt=[経過時間]
```

```
#$ -p [プライオリティ]
```

```
[module の初期化]
```

```
[プログラミング環境のロード]
```

```
[プログラム実行]
```

- ← 現在のディレクトリで下記を実行する (あったほうがよい)
- ← 資源タイプ × 個数を利用 (必須)
- ← 実行時間を0:10:00などと指定 (必須)
- ← スケジューラにとっての優先度(なくても可) 省略時は -5、-4 が中間、-3 が最優先

-cwd, -l, -p等は、このスクリプトに書く代わりに、qsubのオプションとすることも可能。他のオプションについては、利用の手引き4.2.2を参照 -g はここには記述できない。

ジョブスクリプトの例(1)

- 例: Intelコンパイラ+CUDAでコンパイルされたプログラム a.out を実行したい

```
#!/bin/sh
#$ -cwd
#$ -l s_gpu=1
#$ -l h_rt=0:10:00
#$ -N GPU
. /etc/profile.d/modules.sh
module load cuda
module load intel
./a.out
```

※ -l は ハイフン 小文字のエル

※ ./etc は “ドット スペース /etc”

s_gpu を1個使用 (GPU利用の最小単位)

実行時間を10分(お試し利用)に設定

ジョブに名前をつけることも可能

「module」を利用可能にする

「cuda」と「intel」必要なモジュールを load
一行にも書ける module load cuda intel

プログラムを実行

```
module load nvhpc
```

※ 旧PGIコンパイラは
NVIDIA HPC Toolkit となりました。

※ nvhpc のオプションは -ta=tesla,cc60

もしくは nvfortran -Mcuda=cuda8.0,cc60

-gencode=arch=compute_60, code=sm_60

ジョブスクリプトの例 (2)

- OpenMP による、ノード内並列ジョブの例

```
#!/bin/sh
#$ -cwd
#$ -l f_node=1
#$ -l h_rt=0:10:00
#$ -N openmp
. /etc/profile.d/modules.sh
module load cuda/10.2.89
module load intel/19.1.3.304
export OMP_NUM_THREADS=28
```

```
./job_pre
./job_main
./job_post
```

```
for i in data1 data2 data3 ...
do
    ./program ${i}
done
```

資源タイプ F を 1ノード使用

バージョンを明示的に指定

ノード内に28スレッドを配置

複数の処理の記述も可能

入力データの数だけループの例

※ CPU/GPUの指定 <https://www.t3.gsic.titech.ac.jp/node/326>

ジョブスクリプトの例(3)

- MPIによる、複数ノード並列の例 (Intel MPI)

```
#!/bin/sh
#$ -cwd
#$ -l f_node=2
#$ -l h_rt=0:10:00
#$ -N intelmpi
. /etc/profile.d/modules.sh
module load cuda
module load intel
module load intel-mpi
mpiexec.hydra -ppn 4 -np 8 ./a.out
```

資源タイプ F を 2ノード使用

ノードリストは次の変数から取得

`$PE_HOSTFILE`

```
cut -c 1-6 $PE_HOSTFILE > nodelist
cat $PE_HOSTFILE | awk '{print $1 " slots="$2}' > nodelist
```

Intel MPI 環境の設定

ノードあたり 4プロセスで 8並列

- OpenMPIでは、

9行目: module load `openmpi`

10行目: mpirun `-npernode` 4 `-n` 8 `-x` LD_LIBRARY_PATH ./a.out

※ 1ノード 4プロセス (4 GPU)
2ノード 8並列の計算の例

ジョブスクリプトの例(4)

- ハイブリッド並列の例 (Intel MPI)

```
#!/bin/sh
#$ -cwd
#$ -l f_node=2
#$ -l h_rt=0:10:00
#$ -N HyBrid
. /etc/profile.d/modules.sh
module load cuda
module load intel
module load intel-mpi
export OMP_NUM_THREADS=7
mpiexec.hydra -ppn 4 -np 8 ./a.out
```

← 資源タイプ F を 2ノード使用

← Intel MPI 環境の設定

← 1プロセスに 7スレッドを配置

← ノードあたり MPI 4プロセス、
全部で8プロセスを使用する

- OpenMPI だと、
9行目: module load openmpi
11行目: mpirun -npernode 4 -n 8 -x LD_LIBRARY_PATH ./a.out

ステップ2: qsubによるジョブ投入

`qsub -g [TSUBAMEグループ] ジョブスクリプト名`

- [TSUBAMEグループ] は、ジョブスクリプト内ではなく
`qsub -g [TSUBAMEグループ]` として指定する。
 - 省略した場合は、お試し実行扱いとなり、2ノード10分まで

例: `$ qsub -g tgx-23lXX ./job.sh`

→ 成功すると、

`Your job 1234567 ("job.sh") has been submitted`

のように表示され、ジョブID(ここでは1234567)が分かる

- 予約ノードへのジョブの投入は `qsub -ar 予約番号` とする

例: `$ qsub -g tgx-23lXX -ar 予約番号 ./job.sh`

※) AR : Advance Reservation (実際のジョブの長さは10分間短くすること)

ステップ3: ジョブの状態確認

qstat [オプション]

例: qstat

→ 現在の自分のジョブ情報を表示

```
job-ID      prior    name          user            state submit/start at     queue  
jclass                                slots ja-task-ID
```

```
-----  
1234567 0.55500 job.sh      tokyo-t-aa  r      05/03/2019 12:17:41  
all.q(r8i2n7) ← ノード名
```

r は実行中、qw は待機中

Eqw は実行されません。

ジョブステータスが「Eqw」となり実行されない。

<https://www.t3.gsic.titech.ac.jp/node/65>

• 主なオプション

オプション	説明
-r	ジョブのリソース情報を表示します。
-j (JOBID)	ジョブに関する追加情報を表示します。

qstat -u “*” : 全てのジョブを表示します。

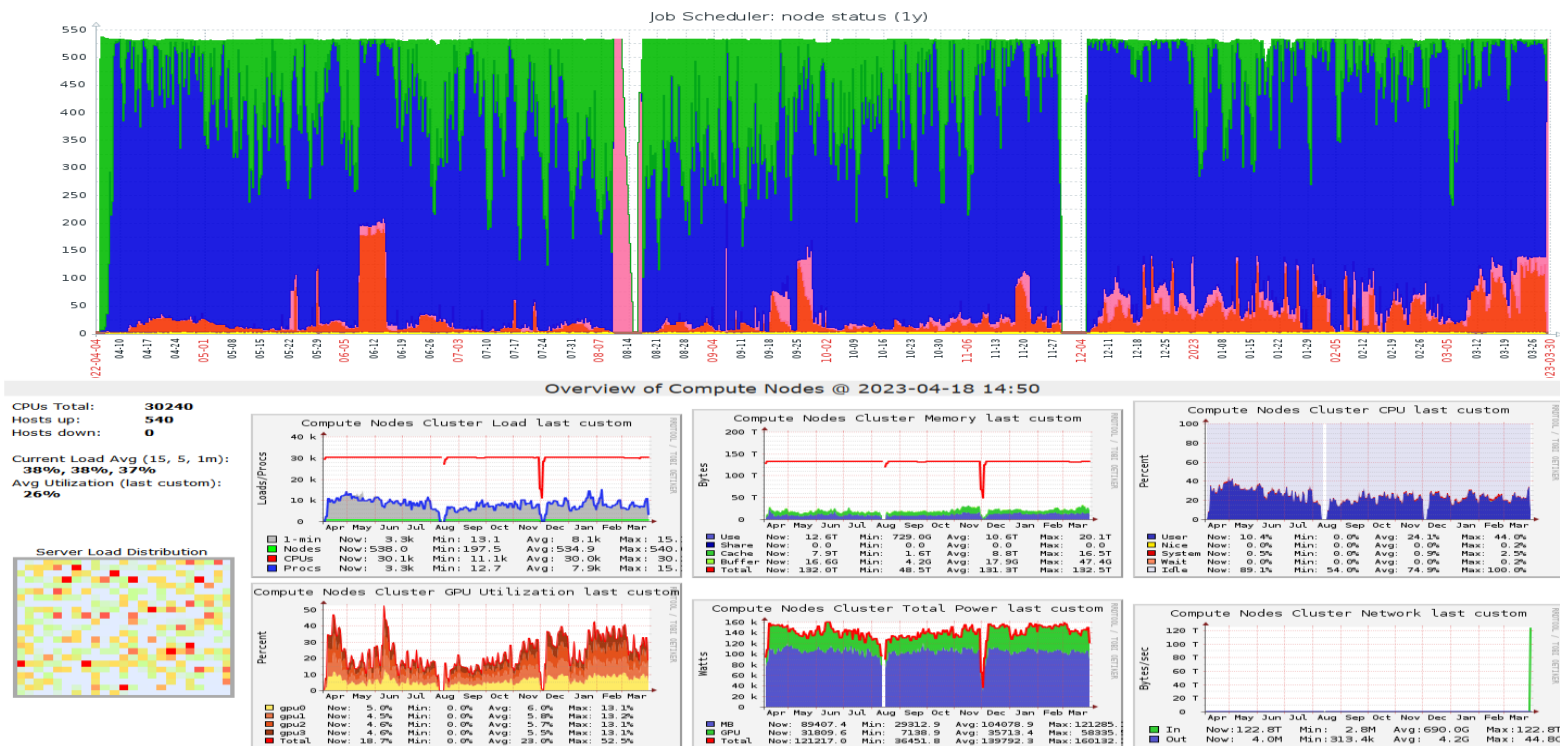
qacct -j job-ID : ジョブの詳細を表示します。

ステップ3: ジョブの状態確認

モニタリング情報

TSUBAME3.0 モニタリングページ <https://www.t3.gsic.titech.ac.jp/monitoring>

- ・ジョブモニタリング (ジョブの混雑具合 : **アイドルノード** **実行ノード** **予約ノード**)
- ・マシンモニタリング (各ノードの状況 : <http://pm1.t3.gsic.titech.ac.jp/ganglia/>)



ステップ4: ジョブを削除するには

qdel [ジョブID] ※ジョブIDは数字のみ

例: qdel 1234567 (前述の Eqw の例など)

※ なんらかの原因でジョブが削除できないときは
共同利用推進室までご連絡ください。

※ TSUBAMEポイント、グループディスクの利用状況は
t3-user-info コマンドにより知ることができます。

例: \$ t3-user-info group point TSUBAMEポイントを表示

例: \$ t3-user-info disk group グループディスクの表示

注意: ジョブをキャンセルしても仮ポイントは**すぐには清算されません**。

3日以内に返却されます <https://www.t3.gsic.titech.ac.jp/node/151>

ステップ5: ジョブ結果の確認

- ジョブが(`printf`などで)出力した結果は、下記のファイルに格納される
 - 標準出力 → `[ジョブスクリプト名].o[ジョブID]`
 - 標準エラー出力 → `[ジョブスクリプト名].e[ジョブID]`たとえば、`job.sh.o1234567` と `job.sh.e1234567`
- ジョブ投入時に `-N [ジョブ名]` をつけておくと、
`[ジョブ名].o[ジョブID]` となる
- `-o [ファイル名]`, `-e [ファイル名]` オプションでも指定可
- `-j y` によりエラー出力を標準出力に書き出す(ファイル1つに)
- `-m abe -M <メールアドレス>` 結果をメールにて通知する
- `qacct -j job-ID` ジョブの詳細を表示する

結果ファイルの説明 <https://www.t3.gsic.titech.ac.jp/node/139>

計算ノードの予約利用

- 計算ノードを、開始時刻・終了時刻を指定して予約

- 1時間、1ノード単位からの予約が可能
- 24時間以上のジョブを予約して利用可能
- 予約可能資源数 (資源タイプ f_node, h_node, q_node)

	4月～9月(閑散期)	10月～3月(繁忙期)
予約可能最大ノード数	270ノード	135ノード
予約可能時間	168時間(7日間)	96時間(4日間)
最大確保予約枠	12,960ノード時間	6,480ノード時間

- 予約時期によって課金係数が異なる
 - 5.00 倍 実行開始24時間以内 (直前の予約を防ぐため)
 - **1.25 倍 実行開始14日前～1日前まで** (14日前頃の予約を推奨)
 - 2.50 倍 上記以外の時期(2週間以上前)
- 計算ノードの予約 https://helpdesk.t3.gsic.titech.ac.jp/manuals/portal.ja/node_reservation/
- ノード予約について <https://www.t3.gsic.titech.ac.jp/node/162> (キャンセルは24時間前)
- 予約後 **5分以内**にキャンセルすればポイントは**全て**返却されます。(予約不成立とする)
予約の5分後～開始24時間までは**半分**。予約開始24時間以内では返却されません。
- 予約時の注意: <https://www.t3.gsic.titech.ac.jp/node/263>
- 予約状況を調べるには t3-user-info compute ars コマンドを用いる

データ転送など外部へのアクセス

- ・ TSUBAME3.0 ではログインノードおよび各計算ノードから外部のネットワークへ直接アクセスできます。(SINET6)
- ・ TSUBAME3.0 にインストールされているソフトウェアでも git などを用いて最新版のソースを参照することが可能です。

例1: lammmps

```
$ git clone https://github.com/lammps/lammps
```

例2: gromacs

```
$ git clone https://github.com/gromacs/gromacs
```

例3: namd

```
$ git clone https://charm.cs.illinois.edu/gerrit/namd.git
```

例4: 最新の TensorFlow をインストールする

```
$ module load cuda python
```

```
$ pip install --user tensorflow
```

- ・ ファイル転送について補足 <https://www.t3.gsic.titech.ac.jp/node/96>
- ・ ISVアプリなどでは学外のライセンスサーバーを直接利用可能です。
- ・ 外部からの計算ノードの見え方 <https://www.t3.gsic.titech.ac.jp/node/244>

ストレージの利用 (1)

- ホームディレクトリ
 - 各ユーザごとに、25GBまで無料で利用可能
/home/?/[ユーザー名] (\$HOME)
- 高速ストレージ(グループディスク Lustre file system)
 - 課題グループのメンバーでアクセスするストレージ領域
(必要に応じて共同利用推進室にて割り当てます)
 - 1TB × 1か月で 36,000ポイント (10ノード時間 相当)
 - 1TB あたり 2,000,000ファイル のファイル数制限あり
 - 年度末まで一括購入されます(月単位での購入はできません)
 - /gs/hs0/[グループ名] もしくは /gs/hs1/[グループ名]
 - 使用量は `lfs quota -g tgx-23lXX /gs/hs0 (-h)` もしくは
“ `t3-user-info disk {group|home}` ” コマンドにより表示される

ストレージの利用 (2)

- ローカルスクラッチ領域（単一ノード）
 - ノードごと・ジョブごとに一時利用できる領域
 - /scr スクラッチ ディレクトリ (SSD NVMe 2TB)
 - ジョブ終了時に消える
 - ノードあたり約 1.9TB 【グループディスクよりも高速】
 - ディレクトリ名は、ジョブごとに異なる
 - 環境変数 \$TMPDIR、\$T3TMPDIR (MPI用) にて参照する
 - たとえば Cプログラムでは、
getenv("TMPDIR") などディレクトリ名の文字列を取得
- 共有スクラッチ領域（複数ノード）
 - 複数の f_node の領域を共有し1つのジョブで利用可能
 - ジョブ内での共有ストレージ（ジョブ終了時に消える）
 - /beond ディレクトリ (BeeGFS On Demand) 2ノードで約 3.7TB
 - #\$ -v USE_BEEOND=1 をジョブスクリプト内に記述する

TSUBAMEポイントについて

- ・グループ区分: tgh-, tgi-, tgj- (課題ID)

TSUBAME3.0 (成果公開 : h, i)	1口	3,600,000 TSUBAMEポイント	110,000円 (税込)
TSUBAME3.0 (成果非公開 : j)	1口	3,600,000 TSUBAMEポイント	330,000円 (税込)

1口は1000ノード時間の計算機資源量です。
1000 ノード × 3600 秒 = ノード秒で計算されます。
TSUBAMEポイントを知るには TSUBAMEポータル
もしくは “ t3-user-info group point ” コマンドにて

ポイントの消費式

ジョブ毎の使用ポイント

$$= \text{ceil}(\text{利用ノード数} \times \text{資源タイプ係数} \times \text{優先度係数} \times 0.7 \times \max(\text{実際の実行時間(秒)}, 300) + 0.1 \times \text{指定した実行時間(秒)})$$

資源タイプ	F	H	Q	G	C4	C1
係数	1.00	0.50	0.25	0.20	0.20	0.06

優先度	(デフォルト) -5	-4	-3
係数	1.00	2.00	4.00

グループストレージの使用ポイント

$$= \text{利用月数} \times \text{利用可能容量(TB)} \times 36,000 \text{ (10ノード時間相当)}$$

※ 課金の詳細につきましては下記をご参照ください。

http://www.somuka.titech.ac.jp/reiki_int/reiki_honbun/x385RG00001339.html#e000000198

関連リンク

ログインノード	login.t3.gsic.titech.ac.jp
共同利用推進室	https://www.gsic.titech.ac.jp/tsubame
共同利用推進室 FAQ	https://www.gsic.titech.ac.jp/kyodou/FAQ
利用講習会資料	https://www.gsic.titech.ac.jp/kyodou/beginners_course
TSUBAME3.0ウェブページ	https://www.t3.gsic.titech.ac.jp
TSUBAME3.0利用 FAQ	https://www.t3.gsic.titech.ac.jp/faq
TSUBAME3.0利用状況	https://www.t3.gsic.titech.ac.jp/monitoring
TSUBAME3.0利用ポータル	https://portal.t3.gsic.titech.ac.jp/ptl
TSUBAME3.0利用の手引き	https://helpdesk.t3.gsic.titech.ac.jp/manuals/handbook.ja/
TSUBAMEポータル利用の手引き	https://helpdesk.t3.gsic.titech.ac.jp/manuals/portal.ja/
採択課題一覧	https://www.gsic.titech.ac.jp/node/60
HPCI産業利用	https://www.gsic.titech.ac.jp/hpci-sangyo
Linux基礎	https://www.t3.gsic.titech.ac.jp/sites/upload/T3_seminar_Linux.pdf
TSUBAME利用法	https://www.t3.gsic.titech.ac.jp/sites/upload/T3_usage.pdf
GPU入門	https://www.t3.gsic.titech.ac.jp/sites/upload/TSUBAME3_GPU_Computing_2023_Spring.pdf
マルチGPUプログラミング	https://www.cc.u-tokyo.ac.jp/events/lectures/124/20191016-2.pdf
HPCIセミナー資料	https://www.hpci-office.jp/events/seminars/seminar_texts

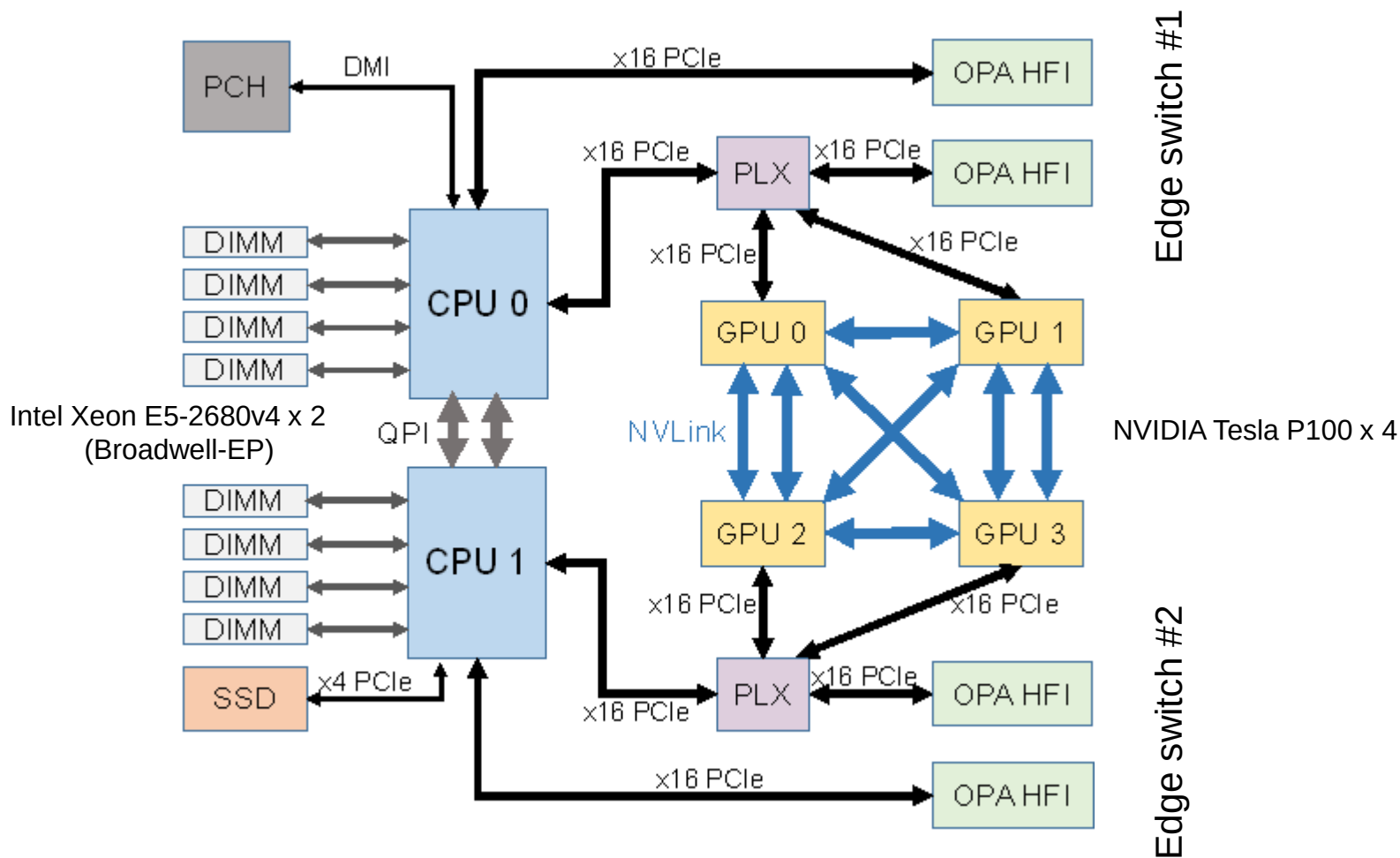
不明なことがありましたら以下のアドレスへ

- 共同利用制度の有償利用の利用者及び、
- HPCI実証利用、トライアルユース利用者は
課題ID、もしくはユーザーIDを添えて、

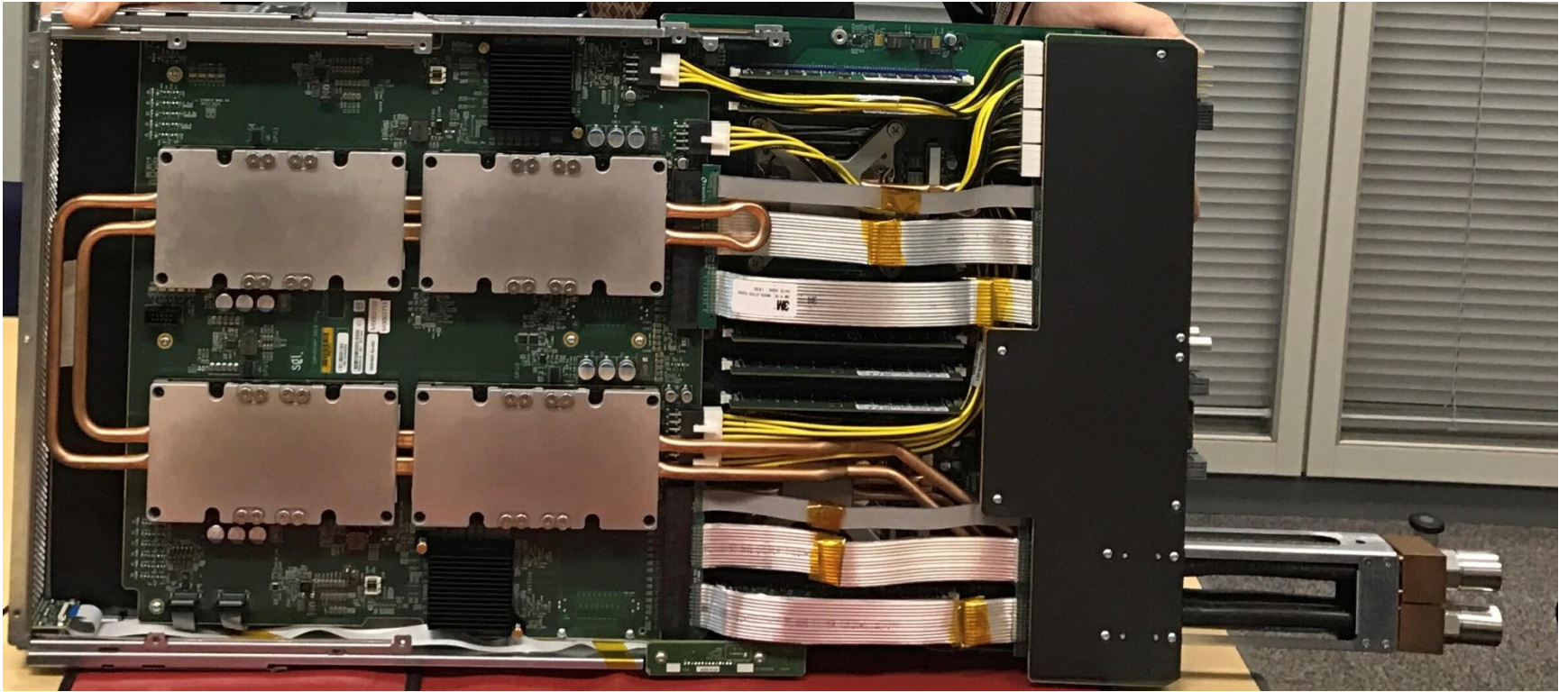
kyoyo@gsic.titech.ac.jp まで

お気軽にお問い合わせください。

TSUBAME3.0 アーキテクチャ



TSUBAME3.0 計算ノード



GPU x 4 (P100)

CPU x 2 (Xeon 2.4GHz)

冷却水↑

TSUBAME3.0 冷却システム系統図

32℃の自然大気冷却水による高効率高温冷却

