TSUBAME4.0利用講習会

https://www.t4.cii.isct.ac.jp/tsubame-kyodo

令和7年度版（2025/4/8)

東京科学大学

情報基盤センター

共同利用支援室

Copyright (C) 2010-2025 GSIC, CII 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
- リンク一覧

TSUBAME 性能向上の歴史

2006
TSUBAME1.0
85TFlops/1.1PB

アジアNo.1 !!
「みんなのスパコン」

2007
TSUBAME1.1
100TFlops/1.6PB

2007年 文科省 先端研究施設
共用イノベーション創出事業

2008
TSUBAME1.2
160TFlops/1.6PB

2009年 TSUBAME共同利用開始

2010/11/01
TSUBAME2.0
2.4PFlops/7.1PB
日本初のペタコン

2010年 JHPCN 開始

2013/9
TSUBAME2.5
5.7PFlops/7.1PB

2012年 HPCI開始

2017/8/1
TSUBAME3.0
12PFlops / 16PB

2024/4
TSUBAME4.0
66.8PFlops / 44PB

2016年 HPCI 産業利用開始
（実証利用、トライアル・ユース）

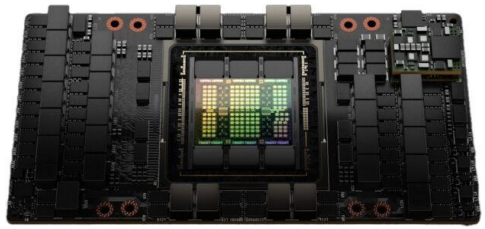
※ 2024年4月より運用開始
Top500 #36 (国内5位) (2024/11)
Green500 #30 (国内1位)

TSUBAME4.0 概要

Compute Node

CPU: AMD EPYC 9654 (96 core) × 2
GPU: NVIDIA H100 SXM5 HBM2e × 4

Performance: 278.5 TFLOPS
Memory: 768 GB(CPU)
94 GB(GPU)



System

240 nodes: 480 CPU sockets, 960 GPUs
Performance: 66.8 PFLOPS

Operating System

RedHat Enterprise Linux 9

Job Scheduler

Altair Grid Engine
(UNIVA Grid Engine)

Interconnect

InfiniBand NDR 200Gbps ×4
Full-bisection Fat-Tree

https://www.gsic.titech.ac.jp/sites/default/files/spec40j.pdf

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　４月稼働開始　(H100)

共同利用推進室 (2024年9月まで) 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

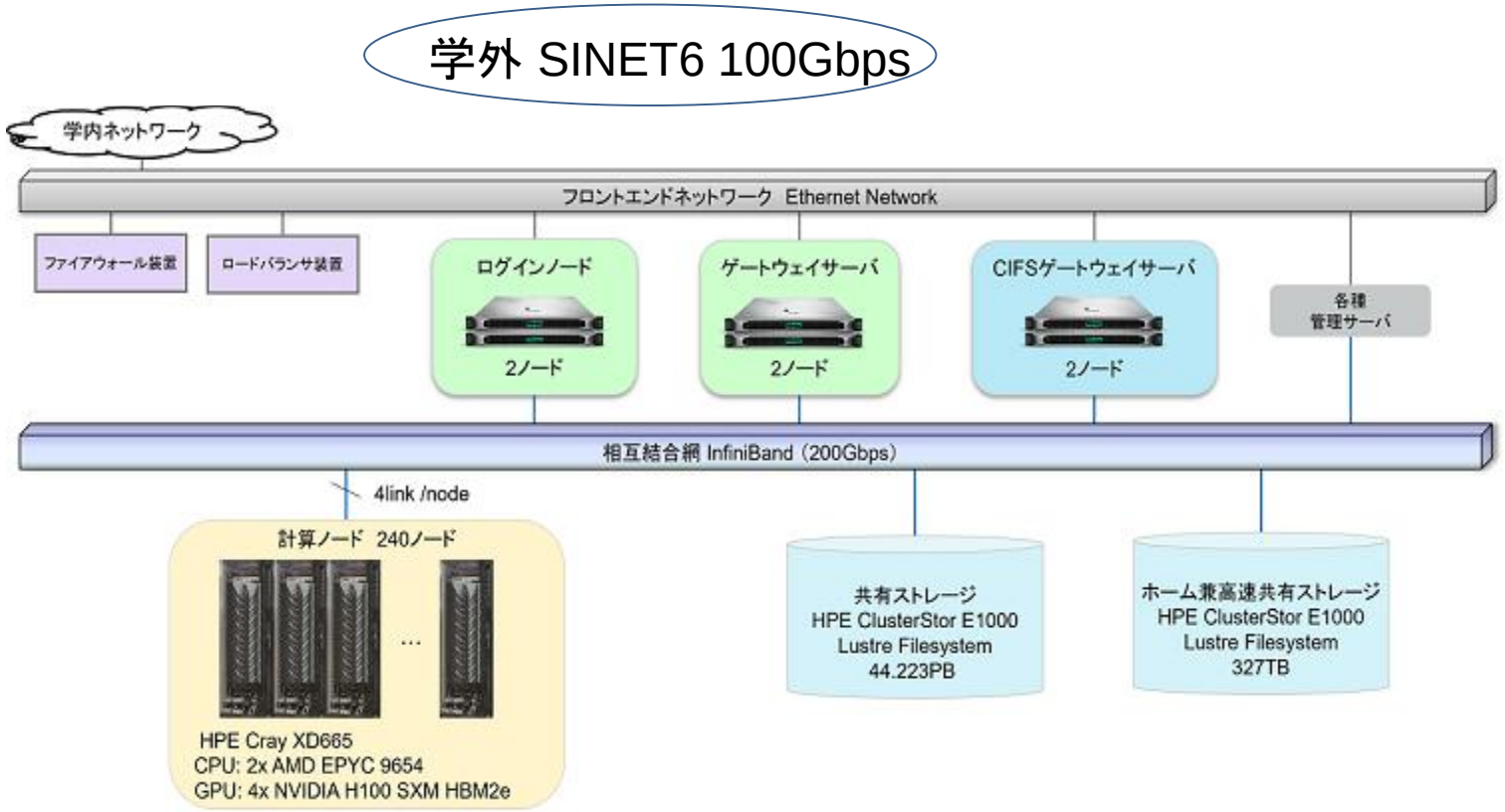
利用区分 ／ 年度		2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	合計
学術利用	HPCI	-	-	-	-	-	6	5	10	14	5	9	12	16	14	8	10	10	13	132
	JHPCN	-	-	-	4	6	5	11	10	10	12	11	15	14	8	7	6	7	18	144
	有償利用	-	-	1	4	9	14	17	22	23	25	23	27	25	28	30	28	26	54	356
産業利用	無償利用/HPCI	11	15	15	8	10	12	21	17	13	15	8	3	3	1	1	1	0	0	154
	有償利用 公開	-	-	3	6	7	9	8	10	8	8	5	6	4	5	2	1	0	14	96
	有償利用 非公開	-	-	2	7	6	4	10	12	10	13	16	19	19	20	14	12	10	24	198
合計		11	15	21	29	38	50	72	81	78	78	72	82	81	76	62	58	53	123	1080

利用区分

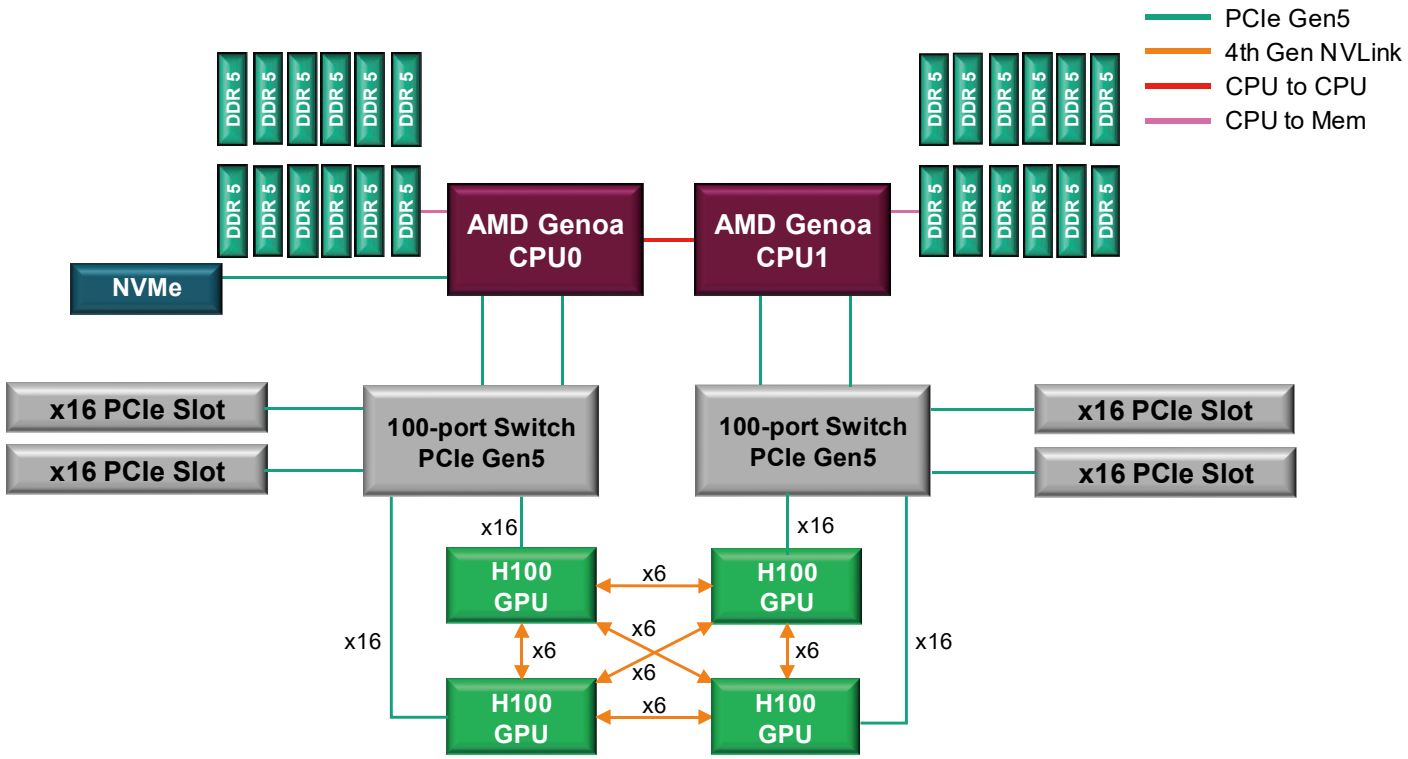
- 有償利用
  - 共同利用：学術利用（成果公開のみ）
  - 共同利用：産業利用（成果公開・成果非公開）
- 無償利用
  - HPCI/JHPCN による利用（学術・産業）

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	東京科学大学 情報基盤センター	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	東京科学大学 情報基盤センター	公開	1口：110,000 円
						非公開	1口：440,000 円

TSUBAME4.0 構成図



TSUBAME4.0 ノード構成概要



https://www.t4.cii.isct.ac.jp/docs/handbook.ja/#compute_node

1 CPU = 8 core x 12 = 96 cores

各GPUは6本のNVLinkにて接続

TSUBAME計算ノード比較

項目	TSUBAME1.2	TSUBAME2.5	TSUBAME3.0	TSUBAME4.0
演算性能	77.48 TFlops	5.76 PFlops	12.15 PFlops	66.8 PFlops
計算ノード数	655台 SunFire X4600	1400台 HP SL390s	540台 HPE (SGI) ICE XA	240台 HPE Cray XD6500
CPU	16コア (AMD Opteron 2.4GHz 2core ×8)	12コア Westmere (Xeon X5670 2.93GHz 6core ×2)	28コア Broadwell (Xeon E5-2680 v4 2.4GHz 14core ×2)	192 コア (AMD EPYC 9654 2.4GHz 96core ×2)
総コア数/GPU	10,480/680	16,800/4,200	15,120/2,160	46,080/960
メモリー	32/64/128GB	54 GB	256 GB	768 GB
GPU	S1070 (Tesla x 4) x 170 = 680	Tesla K20X ×3 (1.3TFlops, 6GB)	Tesla P100 ×4 (5.3TFlops, 16GB)	H100 HBM2e ×4 (66.9TFlops, 94GB)
ローカル ストレージ	N/A	50GB SSD	2TB NVMe SSD	2TB NVMe SSD
ネットワーク	10Gbps x 2 SDR Infiniband	40Gbps x 2 QDR Infiniband	100Gbps x 4 Omni-Path	200Gbps x 4 NDR Infiniband

GPUスパコン計算ノード比較

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	
JCAHPC	**Miyabi-G 演算加速ノード**	78.8 PF	1,120	NVIDIA Grace CPU (72コア、3.0GHz) + NVIDIA H100	70.4 TF	120GiB	InfiniBand NDR200
東京科学大学	**TSUBAME4.0**	66.8 PF	240	AMD EPYC 9654 (2.4GHz,96コア) x 2 + NVIDIA H100 (94GB) x 4	278.5 TF	768GiB	InfiniBand NDR200 x4
九大	**玄界ノードグループB**	10.1 PF	38	Xeon Platinum 8490H (1.9GHz,60コア) x 2 + NVIDIA H100x4(SMX5)	265.0 TF	1,024GiB	InfiniBand NDR400 x2
筑波大	**Pegasus**	8.1 PF	150	Xeon Platinum 8468 + NVIDIA H100,PCIe	54.2 TF	128GiB＋2,048GiB	InfiniBand NDR200
名大	**「不老」TypeⅡ** サブシステム CX2570 M5	7.5 PF	221	Xeon Gold 6230, 2.10-3.90 GHz(20コア) x 2 + NVIDIA V100 x 4	33.9 TF	384GiB	InfiniBand EDR100 x2
東大	**Wisteria/BDEC-01(Aquarius：** データ・学習ノード群)	7.2 PF	45	Xeon Platinum 8360Y(2.4GHz,36コア)x2 + NVIDIA A100 x 8	160.0 TF	512GiB	InfiniBand HD200 x4
阪大	**SQUID GPUノード**	6.8 PF	42	Xeon Platinum8368 (2.4GHz,38コア)x2 + NVIDIA A100 x 8	161.8 TF	512GiB	InfiniBand HDR200
北大	**次期スパコン Grand Chariot 2** （2025/7予定～）	6.6 PF	24	Xeon Gold 6548Y (2.5GHz,32コア)x2 + NVIDIA H100 x 4	272.7 TF	512GiB	InfiniBand NDR x 2
産総研	**ABCI3.0 (XD670)**	415 PF	766	Xeon 8558 (48コア) x 2 + NVIDIA H200 SXM5 (141GB) x 8	541.8 TF	2,048GiB	InfiniBand NDR x 8

https://www.hpci-office.jp/application/files/1717/2722/4253/r07a_boshu_setsumeikai_hpci.pdf#page=6 より引用

利用開始とログイン

2008年 2010年 2013年 2017年 2024年

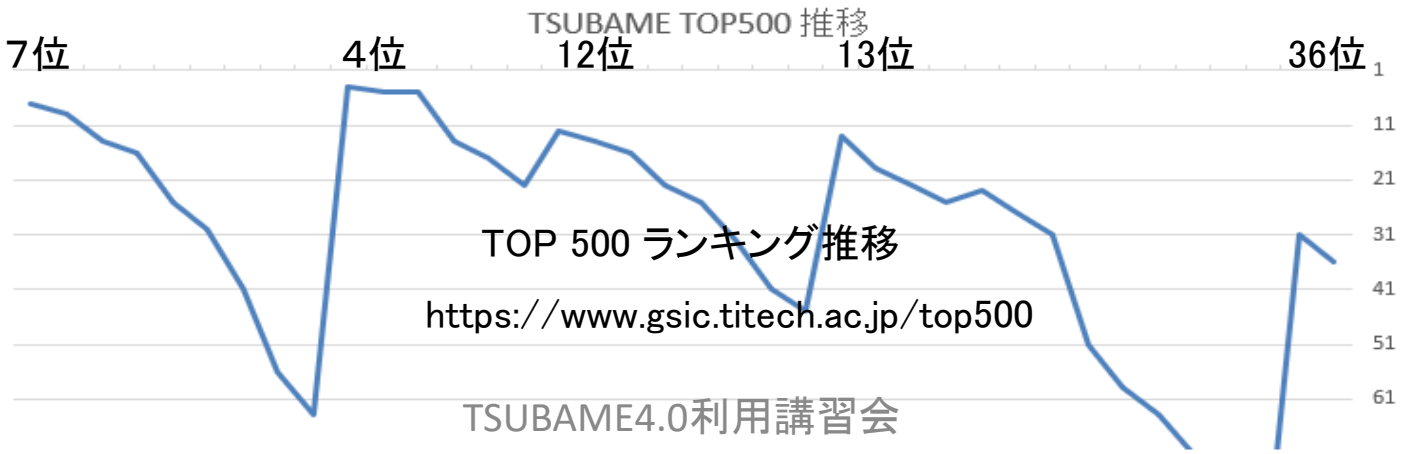
Tesla S1070 (Tesla GT200)
on TSUBAME1.2

Tesla M2050 (Fermi)
on TSUBAME2.0

Tesla K20X (Kepler)
on TSUBAME2.5

Tesla P100 (Pascal)
on TSUBAME3.0

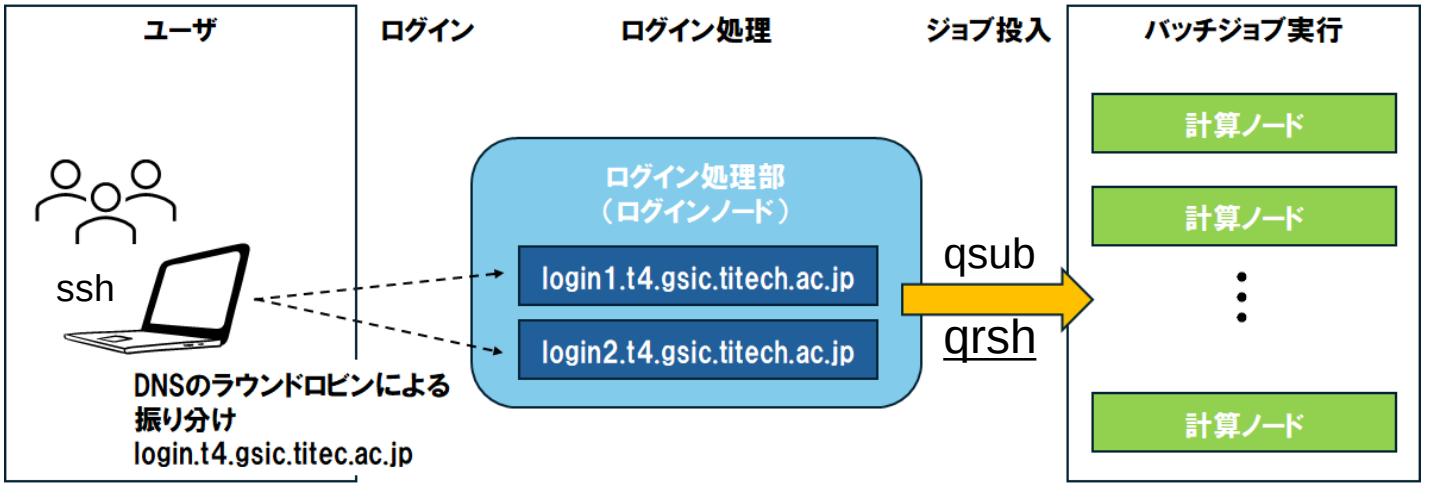
NVIDIA H100 SXM5
on TSUBAME4.0



TOP 500 ランキング推移

https://www.gsic.titech.ac.jp/top500

計算機へのログイン



iqrsh → インタラクティブジョブ専用キュー

- SSHログイン: ssh <username>@login.t4.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

TSUBAME4ポータル

- **アカウント作成方法 (以下のいずれか)**
  - 東京科学大学ポータル → TSUBAMEポータル
  - TSUBAMEポータル https://portal.t4.gsic.titech.ac.jp/ptl/
- 学外の方のアカウントは共同利用支援室にて発行

  アカウント発行に際し本人のメールアドレスが必要

  TSUBAME4.0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.t4.cii.isct.ac.jp/docs/faq.ja/general/#sshclients_win

  - 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.t4.cii.isct.ac.jp/sites/default/files/2025-04/Portal2025v1.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時間＝ 1 TSUBAMEポイント
  - １口＝ 400ノード時間＝ 400 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。
- グループストレージ （課題代表者にて設定可能）
  - HDD /gs/bs/グループ名 大容量ストレージ（TB 単位 100TB）
  - SSD /gs/fs/グループ名 高速ストレージ（GB 単位 3TBまで）
  - HDD 大容量ストレージ： 1TB/年 ６ TSUBAMEポイント
  - SSD 高速ストレージ： 1TB/年 24 TSUBAMEポイント
  - ホームディレクトリ (25GB) は無償

TSUBAME4.0ソフトウェア

- OS : Red Hat Enterprise Linux 9.4
- スケジューラ : Altair Grid Engine 2023.1.1
- コンテナ : Apptainer (旧 Singularity)
- コンパイラ: （※ Intel の icc, ifort は icx, ifx に）

  gcc 11.4.1, oneAPI 2025.0.0, nvhpc 25.1, AOCC 4.1.0
- MPI : Intel MPI 2021.11, OpenMPI 5.0.7-gcc
- CUDA 12.8.0 (ドライバ 570.124.06)
- プログラミングツール: Intel Vtune, PAPI, Linaro Forge…
- その他商用アプリ(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係する環境設定は、
  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/2024.0.2 のようにバージョン指定も可能
- 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依存モジュールもロードされる
  現在のモジュールは module list で確認する
  - 例：module load gromacs　で関連モジュールもロード
    Loading requirement: cuda/12.3.2 openmpi/5.0.2-gcc
- T4 では modules.sh の実行は不要となりました。
  - . /etc/profile.d/modules.sh ← 不要です

現在インストールされているモジュール

コンパイラ、MPI、開発ツール 関連のモジュール。$ module available　必要に応じたバージョンのモジュールを load して使用します。

コンパイラ: gcc 11.4.1, 14.2.0、Intel onAPI 2024.0.2, 2025.0.0、nvhpc 24.1、AOCC 4.1.0　MPI:　Intel MPI、OpenMPI、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/t4/rhel9/modules/modulefiles/compiler ----------------------------------------
aocc/4.1.0   cuda/12.0.0  cuda/12.3.2  intel/2024.0.2(default)  nvhpc/21.3          nvhpc/24.11_cuda11_gcc14
cuda/11.8.0  cuda/12.1.0  gcc/14.2.0   intel/2025.0.0           nvhpc/24.1(default)
------------------------------------------ /apps/t4/rhel9/modules/modulefiles/mpi -------------------------------------------
intel-mpi/2021.11  openmpi/5.0.2-gcc  openmpi/5.0.2-intel  openmpi/5.0.2-nvhpc
---------------------------------------- /apps/t4/rhel9/modules/modulefiles/tools ----------------------------------------
forge/23.1.2  intel-dnnl/3.3.0  intel-dnnl/3.6.0  intel-ins/2024.0  intel-itac/2022.2  intel-vtune/2024.0  intel-vtune/2025.0
---------------------------------------- /apps/t4/rhel9/modules/modulefiles/isv -----------------------------------------
abaqus/2024                                ansys/R24.1                mathematica/14.0(default)  VASP/6.4.2/5.0.2-nvhpc
amber/22up05_ambertools23up06_cpu          comsol/62_u2               mathematica/14.1           VASP/6.4.3/5.0.2-nvhpc(default)
amber/22up05_ambertools23up06_gpu          comsol/62_u3               mathematica/14.2           VASP5/5.4.4.pl2/5.0.2-nvhpc
amber/22up05_ambertools23up06_py310(default)  gaussian/16C2_cpu(default)  matlab/R2024a(default)  VASP6.5/6.5.0/5.0.2-nvhpc
amber/24up01_ambertools24up02              gaussian/16C2_cpu_NBO      matlab/R2024b
amber/24up02_ambertools24up03              gaussview/6.1              schrodinger/2024-1
---------------------------------------- /apps/t4/rhel9/modules/modulefiles/free -----------------------------------------
alphafold/2.3.2             cudnn/9.0.0                  hdf5-parallel/1.14.3/nvhpc24.1  openfoam-esi/v2312        spack/0.21.2
alphafold2_database/202411  deepmd-kit/2.2.9             imagemagick/7.1.1-29            openfoam/11.0             tensorrt/8.6.1.6
alphafold2_database/202503  ffmpeg/6.1.1                 jupyterlab/4.1.4                openjdk/1.8.0             tgif/4.2.5
alphafold3_database/202411  fftw/3.3.10-gcc              lammps/2aug2023_u3              openjdk/11.0.22(default)  tinker/8.10.5
alphafold3_database/202503  fftw/3.3.10-intel            miniconda/24.1.2                openjdk/21.0.2            tmux/3.3
autoconf/2.72               fftw/3.3.10-nvhpc            namd/3.0                        papi/7.1.0                turbovnc/3.1.1
automake/1.17               gamess/Jun302023R1           namd/3.0.1                      paraview/5.12.0(default)  VESTA/3.5.8
cmake/3.28.3                gromacs/2023-plumed          namd/3.0b6(default)             paraview/5.12.0-egl       visit/3.1.4
code-server/4.22.1          gromacs/2023.5               nccl/2.20.5                     petsc/3.20.4-complex      vmd/1.9.4
colabfold_database/202411   gromacs/2024(default)        netcdf-parallel/4.9.2/gcc11.4.1 petsc/3.20.4-real
colabfold_database/202503   gromacs/2024.2-plumed        netcdf-parallel/4.9.2/nvhpc24.1 pov-ray/3.7.0.9
cp2k/2024.1                 hadoop/3.3.6                 ninja/1.11.1                    quantumespresso/7.3.1
cudnn/8.9.7                 hdf5-parallel/1.14.3/gcc11.4.1  novnc/1.4.0                  R/4.4.0
---------------------------------------- /apps/t4/rhel9/modules/modulefiles/gsic -----------------------------------------
apptainer-olderenv  experimental  jupyterrun
```

※ Python は module load しなくても 3.9.18 が利用できます。※ **VASP の利用にはライセンスの所有が条件となります。**

※ HPCI の国プロソフトは現在準備中です。 ※ Gaussian/GaussView は学外からも有償で利用できます。

※ 有償の商用アプリの実行にはポータルでの登録が必要です。 https://www.t4.cii.titech.ac.jp/fare_overview

現在インストールされているモジュール

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、

本学のTSUBAME4.0サポート窓口ではサポート対応しておりません。

HPCIユーザはHPCIヘルプデスクへ直接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/t4/rhel9/modules/modulefiles/hpci-apps ----------------------------
abinit-mp/v1r22          ffvhc-ace/0.1             genesis/2.1.4.mixed_cpu   modylas/1.1.0         phase0/2024.01
akaikkr/cpa2021v02_cpu   ffx/03.01.01              genesis/2.1.4.mixed_gpu   mvmc/1.3.0            phonopy/2.27.0
akaikkr/cpa2021v02_gpu   frontistr/5.6             genesis/2.1.4.single_cpu  ntchem/24.10.mpi      salmon/2.2.1_cpu
alamode/1.5.0            genesis/2.1.4.double_cpu  genesis/2.1.4.single_gpu  ntchem/24.10.mpiomp   salmon/2.2.1_gpu
ffb/9.0                  genesis/2.1.4.double_gpu  hphi/3.5.2                openmx/3.9.9          smash/3.0.2
--------------------------------------- /apps/t4/rhel9/modules/modulefiles/hpci-apps ----------------------------
```

詳細につきましてはこちらをご参照ください。

https://www.t4.cii.isct.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/sign_in/registration_form/　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.

参考：VASPのビルド手順　https://www.t4.cii.isct.ac.jp/docs/faq.ja/apps/#vasp_build

ジョブの実行についての概要

- ジョブスケジューラは Altair Grid Engine (UGE)
- ジョブの性質にあわせて、資源タイプを選択
  - node_f (フル), node_h (ハーフ), node_q (クォーター)…
  - gpu_1、gpu_h、cpu_160、cpu_80、cpu_40 …
- 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 node_f のみ直接 ssh でログイン可能

TSUBAME4.0 資源タイプ一覧

資源タイプ	CPUコア数	GPU数	メモリ(GB)	ローカルスクラッチ領域(GB)	課金係数
node_f	192	4	768	1920	1.00
node_h	96	2	384	960	0.50
node_q	48	1	192	480	0.25
node_o	24	1/2	96	240	0.125
gpu_1	8	1	96	240	0.20
gpu_h	4	1/2	48	120	0.10
cpu_160	160	0	368	960	0.60
cpu_80	80	0	184	480	0.30
cpu_40	40	0	92	240	0.15
cpu_16	16	0	36.8	96	0.06
cpu_8	8	0	18.4	48	0.03
cpu_4	4	0	9.2	24	0.015

≒T3のf_node

- 計算機資源は、node_f=1、node_q=4 のように指定する。

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 (-I = 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 node_q=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 node_q なので、48コア1GPU 利用可能。(TSUBAME3.0 の1ノード相当)

- 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 node_h=2 –l h_rt=1:00:00 –g tg*x*-25I*XX*

※ 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

- node_f 以外を qrsh で割り当てた場合もX転送が可能。

  例 : qrsh -l cpu_4=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**

  node_o 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]
```

- 24コア、メモリ96GB 、1/2GPU、最大12名で共有し対話的に利用。
- 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。
- 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://www.t4.cii.isct.ac.jp/docs/handbook.ja/jobs/#submit

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

← module の初期化は不要となりました。

-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 gpu_1=1
#$ -l h_rt=0:10:00
#$ -N GPU
module load cuda
module load intel
./a.out
```

※ -l は ハイフン 小文字のエル

gpu_1 を1個使用 (GPUを1つ利用)

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

「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,cc90
もしくは nvfortran –Mcuda=cuda12.0,cc90
-gencode=arch=compute_90, code=sm_90

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 node_f=1
#$ -l h_rt=0:10:00
#$ -N openmp

module load cuda/12.3.2
module load intel/2024.0.2
export OMP_NUM_THREADS=192

./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192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

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 node_q=4
#$ -l h_rt=0:10:00
#$ -N intelmpi

module load cuda
module load intel
module load intel-mpi
mpiexec.hydra -ppn 1 -n 4 ./a.out
```

資源タイプ Q を 4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 1プロセスで 4並列

※ 1ノード 1プロセス (1 GPU)
4ノード 4並列の計算の例

- OpenMPIでは、

9行目: module load openmpi

10行目: mpirun –npernode 1 –n 4 -x LD_LIBRARY_PATH ./a.out

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 node_q=2
#$ -l h_rt=0:10:00
#$ -N HyBrid

module load cuda
module load intel
module load intel-mpi
export OMP_NUM_THREADS=8
mpiexec.hydra -ppn 6 -n 12 ./a.out
```

資源タイプ Q を 2ノード使用

Intel MPI 環境の設定

1プロセスに 8スレッドを配置

ノードあたり MPI 6プロセス、
全部で12プロセスを使用する

- OpenMPI だと、

  9行目：module load openmpi

  11行目：mpirun –npernode 6 –n 12 –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 tg*x*-25I*XX* ./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 tg*x*-25I*XX* –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
    202406 0.55256 t4job      ux01234      r     06/05/2024 12:34:56 all.q@r20n1
-------------------------------------------------------------------------------
```

r は実行中、qw は待機中
Eqw は実行されません。

ノード名

- 主なオプション　ジョブステータスが「Eqw」となり実行されない。
  https://www.t4.cii.isct.ac.jp/docs/faq.ja/scheduler/#status_eqw

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

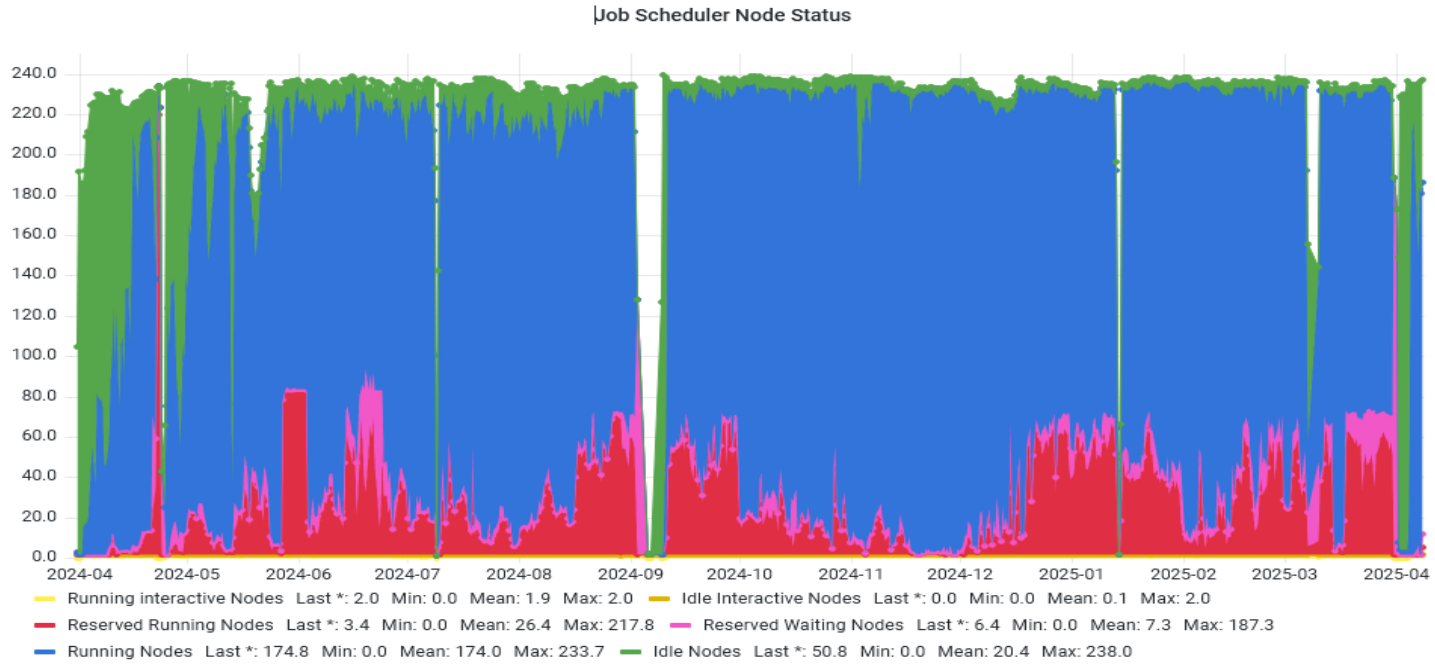
モニタリング情報

TSUBAME4.0 モニタリングページ

https://www.t4.cii.isct.ac.jp/monitoring/d/kK13bVxIk/dashboard-list?orgId=3

・ジョブモニタリング https://www.t4.cii.isct.ac.jp/monitoring/d/gfS9vcbIz/job-scheduler-node-status?orgId=3&from=now-7d&to=now

・マシンモニタリング https://www.t4.cii.isct.ac.jp/monitoring/d/MhvV9pAIk/compute-nodes?orgId=3&from=now-7d&to=now



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
　　共同利用支援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
t4-user-info コマンドにより知ることができます。

例 : $ t4-user-info group point　　TSUBAMEポイントを表示

例 : $ t4-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。

https://www.t4.cii.isct.ac.jp/docs/faq.ja/portal/#return_point

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

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 （ポータルにて権限を登録）
  - **1時間、1ノード単位**からの予約が可能
  - 24時間以上のジョブ実行したい場合は予約して利用する
  - 予約可能資源数 (資源タイプ node_f, node_h, node_q, node_o)

	4月～9月（閑散期）	10月～3月（繁忙期）
予約可能最大ノード数	70ノード	20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	3360ノード時間	960ノード時間

  - 予約時期によって課金係数が異なる

	4月～9月	10月～3月
実行開始24時間前（直前の予約を防ぐため）	5.00	10.00
実行開始14日前～1日前まで（14日前頃を推奨）	1.25	2.50
上記以外の時期（2週間以上前）	2.50	5.00

・計算ノードの予約　https://www.t4.cii.isct.ac.jp/docs/handbook.ja/jobs/#reservation

・ノード予約について https://www.t4.cii.isct.ac.jp/docs/portal.ja/node_reservation/

・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80%。予約開始前までは50%返却されます。

・予約時の注意: https://www.t4.cii.isct.ac.jp/docs/faq.ja/scheduler/#reservation_troubleshoot

・予約状況を調べるには　t4-user-info compute ars コマンドを用いる

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)
- 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
    ・大容量ストレージ領域 /gs/bs (HDD) 最大 100TB まで (1TB 単位)
    1TB あたり年間 6.0 ポイント消費（1ヶ月あたり 0.5 ポイント）
    ・高速ストレージ領域 /gs/fs (SSD)　最大 3TB まで (100GB 単位)
    100GB あたり年間 2.4 ポイント消費（1ヶ月あたり 0.2 ポイント）
  - /gs/bs/[グループ名] もしくは /gs/fs/[グループ名] としてアクセス
  - 使用量は lfs quota –g tg*x*-25I*XX* /gs/{bs|fs} (-h) 、
    “ t4-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
    - /local/${JOB_ID} スクラッチ ディレクトリ (SSD NVMe 1.92TB)
    - ジョブ終了時に消える
    - https://www.t4.cii.isct.ac.jp/docs/handbook.ja/jobs/#storage
  - ディレクトリ名は、ジョブごとに異なる

→ 環境変数 $TMPDIR 、$T4TMPDIR (MPI用) にて参照する

- たとえば Cプログラムでは、
  getenv(“TMPDIR”) などでディレクトリ名の文字列を取得する

TSUBAME4.0からは BeeOND はなくなりました。
高速SSD領域としては /gs/fs を利用してください。

※ TSUBAME3.0 vs TSUBAME4.0 比較表
https://www.t4.cii.isct.ac.jp/docs/handbook.ja/comparison/

TSUBAMEポイントについて

・グループ区分: tgh-, tgi-, tgj-（課題ID）

TSUBAME4.0 (成果公開 : h, i)	1口	400 TSUBAMEポイント	110,000円 （税込）
TSUBAME4.0 （成果非公開 : j ）	1口	400 TSUBAMEポイント	440,000円 （税込）

１口は 400ノード時間の計算機資源量です。
400 ノード × 1 時間 = ノード時間で計算されます。
TSUBAMEポイントを知るには TSUBAMEポータル
もしくは “ t4-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

= (利用ノード数 × 資源タイプ係数 × 優先度係数 ×
(0.7 × max(実際の実行時間(秒), 300) ＋0.1 × 指定した実行時間(秒))) ÷3600

資源タイプ	F	H	Q	O	G1	G2	C1	C2	C3	C4	C5
係数	1.00	0.50	0.25	0.125	0.20	0.10	0.60	0.30	0.15	0.06	0.03

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

※実行時間が5分間未満でも、5分(300秒)分のポイントが消費されます。

グループストレージの使用ポイント

- HDD /gs/bs は 1TB、1年あたり 6ポイント (6ノード時間相当) を課金
- SSD /gs/fs は 100GB、1年あたり 2.4ポイント (2.4ノード時間相当) を課金

※利用課金詳細: https://www.t4.cii.isct.ac.jp/fare_overview

データ転送など外部へのアクセス

- TSUBAME4.0 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)
- TSUBAME4.0 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: lammps
```
$ git clone https://github.com/lammps/lammps
```

例２: gromacs
```
$ git clone https://github.com/gromacs/gromacs
```

例３: PyTorch
```
$ git clone https://github.com/pytorch/pytorch
```

- 商用アプリソフトでは学外のライセンスサーバーを直接参照してください。
- 外部からの計算ノードの見え方
  https://www.t4.cii.isct.ac.jp/docs/faq.ja/general/#ipaddr

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、

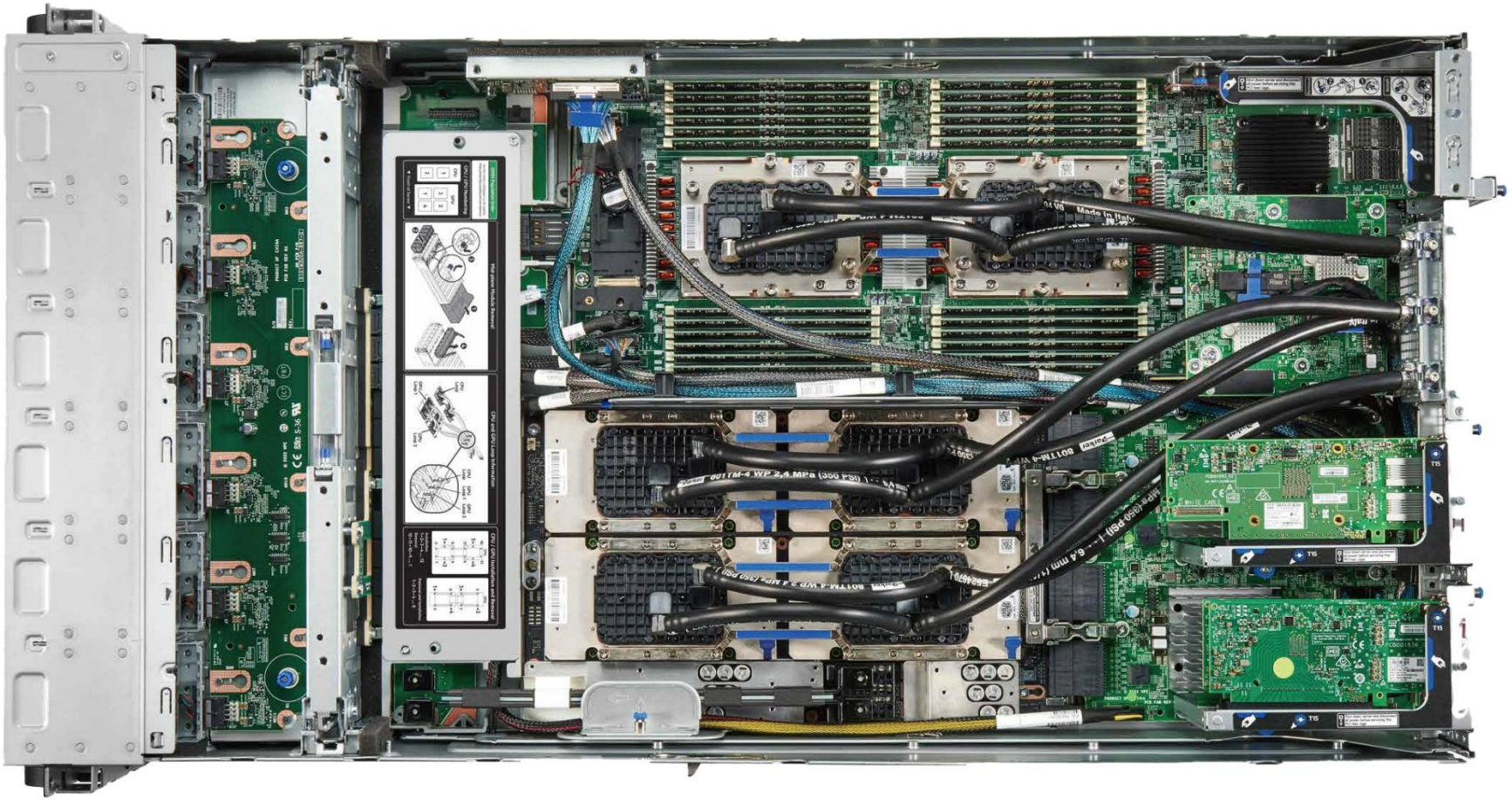
  tsubame-kyodo@cii.isct.ac.jp まで

  お気軽にお問い合わせください。

関連リンク

ログインノード	login.t4.gsic.titech.ac.jp
共同利用支援室	https://www.t4.cii.isct.ac.jp/tsubame-kyodo
共同利用支援室 FAQ	https://www.t4.cii.isct.ac.jp/tsubame-kyodo/FAQs
利用講習会資料	https://www.t4.cii.isct.ac.jp/node/182
TSUBAME4.0ウェブページ	https://www.t4.cii.isct.ac.jp/
TSUBAME4.0利用 FAQ	https://www.t4.cii.isct.ac.jp/manuals
TSUBAME4.0利用ポータル	https://portal.t4.gsic.titech.ac.jp/ptl/
TSUBAME4.0利用状況	https://www.t4.cii.isct.ac.jp/monitoring/d/kK13bVxIk/dashboard-list?orgId=3
TSUBAME4.0利用の手引き	https://www.t4.cii.isct.ac.jp/docs/handbook.ja/
TSUBAMEポータル利用手引き	https://www.t4.cii.isct.ac.jp/docs/portal.ja/
Open OnDemand 利用手引き	https://www.t4.cii.isct.ac.jp/docs/ood/
採択課題一覧	https://www.t4.cii.isct.ac.jp/tsubame-kyodo/AdoptedProjects
HPCI産業利用	https://www.gsic.titech.ac.jp/node/861.html
Linux基礎	https://www.t4.cii.isct.ac.jp/sites/default/files/2024-12/T4_seminar_Linux_2.pdf
並列プログラミング	https://www.t4.cii.isct.ac.jp/sites/default/files/2024-12/parallel_programming_2.pdf
GPU入門	https://www.t4.cii.isct.ac.jp/sites/default/files/2024-06/Intro_to_GPU_programming.pdf
GPUハンズオン	https://www.t4.cii.isct.ac.jp/sites/default/files/2024-06/hands-on_openacc.pdf
H100アーキテクチャ	https://developer.nvidia.com/ja-jp/blog/nvidia-hopper-architecture-in-depth/
マルチGPUプログラミング	https://www.cc.u-tokyo.ac.jp/events/lectures/124/20191016-2.pdf
AMD EPYC 9654	https://pc.watch.impress.co.jp/docs/news/1454879.html
HPCIセミナー資料	https://www.hpci-office.jp/events/seminars/seminar_texts

TSUBAME4.0計算ノード



TSUBAME4.0計算ノード外観

TSUBAME4.0 全体図