TSUBAME3.0利用講習会

www.t3.gsic.titech.ac.jp



平成29年度版(Rev.20180223) 東京工業大学 学術国際情報センター 共同利用推進室

TSUBAME3.0利用講習会

CONTENTS

□ 歴史・概要
 □ ハードウェア・ソフトウェア仕様
 □ 利用法(ログイン)
 □ 利用可能アプリケーション~module~
 □ 資源タイプ
 □ ジョブの実行とスクリプト
 □ 課金情報
 □ リンクー覧

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)

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 2009年 TSUBAME共同利用開始 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

利用	1区分 /	年度	2007年	2008年	2009年	2010年	2011年	2012年	2013年	2014年	2015年	2016年	合計
	ŀ	IPCI	-	-	-	-	-	6	5	10	14	5	40
学術利用	٦٢	IPCN	-	-	-	4	6	5	11	10	10	12	58
	有償利用		-	-	1	4	9	14	17	22	23	25	115
	無	賞利用	11	15	15	8	10	12	21	17	13	15	137
産業利用	右償利田	成果公開	-	-	3	6	7	9	8	10	8	8	59
	11頁利用	成果非公開	-	-	2	7	6	4	10	12	10	13	64

HPCI 産業利用(実証利用、トライアル・ユース)開始

2016年 東京工業大学 学術国際情報センター 自主事業化、

※ 2017年8月より運用開始 Green500 世界1位(2017/6) Top500 世界13位(2017/11)

TSUBAM3.0 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



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



Operating System Job Scheduler

SUSE Linux Enterprise Server 12 SP2 UNIVA Grid Engine

http://www.gsic.titech.ac.jp/sites/default/files/T3SpecJ.pdf

TSUBAME3.0利用講習会

TSUBAME2.5との比較

	TSUBAME2.5	TSUBAME3.0
計算ノード数	1400台 HP SL390s	<mark>540台</mark> SGI ICE XA
CPU	12コア Westmere (Xeon X5670 2.93GHz 6core ×2)	28コア Broadwell (Xeon E5-2680 v4 2.4GHz 14core × 2)
総コア数/GPU	16,800/4200	15,120/2160
Memory	54 GB	256 GB
GPU	Tesla K20X ×3 (GPUあたり、1.3TFlops, 6GBメモリ)	Tesla P100 × 4 (GPUあたり、5.3TFlops, 16GBメモリ)
ローカルストレージ	50GB SSD	2TB NVMe SSD
ネットワーク	40Gbps QDR IB × 2	100Gbps Omni-Path × 4
	TSUBAME3.0利用講習会	

TSUBAME2.5 vs 3.0 比較表

項目	TSUBAME2.5	TSUBAME3.0
OS	SLES 11 SP3 x86/64bit	SLES12 SP2 x86/64bit
ジョブ管理	PBS (Portable Batch System) t2sub, t2stat, t2del	UGE (Univa Grid Engine) qsub, qstat, qdel, qrsh
ネットワーク	InfiniBand (Mellanox)	Omni-Path (Intel)
CUDA	7.5 (K20Xx3) GPU direct	8.0 (P100x4) NVlink
ログインノード	login-t2.g.gsic.titech.ac.jp (2)	login .t3.gsic.titech.ac.jp (2)
インタラクティブ	20 (10)	なし (qrsh)
計算資源	キュー (S, G, U, H, X) 12 core	資源タイプ (F, H, Q, G) 28 core
環境設定	切り替えシェルスクリプト	module コマンド
外部へのアクセス	t2a006170	すべてのノード
転送用ノード	io-t2	なし(同上)
計算ノード	Sキュー×400 など	540(現在520)

利用開始とログイン

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) 東工大ポータル → TSUBAME3ポータル

- <u>www.t3.titech.ac.jp</u>からリンク → アカウント名を入力すると メールが飛ぶ → URLをクリック
- ・ 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メールアドレスが必要 TSUBAME3.0ポータルにて
 - 公開鍵の設定(WSL, ssh-keygen, Tera Term, PuTTY)
 - 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)
 - http://www.gsic.titech.ac.jp/sites/default/files/Portal2017t3.pdf

計算機へのログイン



- SSHログイン: ssh <username>@login.t3.gsic.titech.ac.jp
 > ログインノードの一つへ振り分けられる
 - 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、コンパイルなど
 - GPU なし (module load cuda でCUDAコンパイル可能)
 - 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 -Y にてログインする

TSUBAME3.0ソフトウェア

- OS: SUSE Linux Enterprise Server (SLES) 12 SP2
 Dockerコンテナによる仮想化 (準備中)
- スケジューラ: Univa Grid Engine
- コンパイラ: gcc, Intel, PGI
- MPI: OpenMPI, Intel, SGI MPT (Message Passing Toolkit)
- CUDA 8.0
- プログラミングツール: Intel Vtune, PAPI, Allinea Forge...
- 多種ISVアプリ(後述)

moduleコマンド(後述)による切り替え

利用できるサービス

• 無償サービス

- ホームディレクトリ (25GBまで)
 小規模の計算試験(2 資源タイプ 10分間まで)
 インタラクティブ、デバッグ専用ノードの利用 (?)
 GUI 専用ノード準備中(?)
- 有償サービス
 - 研究目的の大規模計算(従量制、予約キュー?)
 高速ストレージ利用(グループ利用、月額制)
 - 追加ISVアプリケーション利用(?)
 - アプリ別課金システム(?)

有償サービス

- 課題単位でグループを作成 (TSUBAMEグループ)
- 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
 - TSUBAME2.5 の時より桁数は多く見える (3600倍)
- グループ共有の高速ストレージ
 - /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)

TSUBAME3.0のストレージ

- ホーム(全ノードから共有)
 - /home
 - ファイルシステム:NFS
 - 1ユーザあたり25GB、無料で利用可能
- 高速ストレージ(全ノードから共有)
 - /gs/hs0, /gs/hs1, /gs/hs2
 - ファイルシステム:Lustre
 - それぞれ 4.8PB
 - グループによる購入が必要(1TB/月 10ノード時間相当) - グループあたり最大 300TB
- ローカルスクラッチ領域 (各ノード固有)
 - /scr
 - ノードあたり 1.9TB
 - 最も高速だが、ジョブ終了時に消える

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を、前もってmoduleコマンドで行う
 - 例: module load intel → Intelコンパイラ
 - module load intel/16.0.4.258 のようにバージョン指定も可能
 - 例: module load python-extension/3.4
 - TSUBAME2.5ではシェルスクリプトで環境を切り替え
- 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
 たとえば、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「ピリオド・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 関連のモジュールです。 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 4.8.5(*)、Intel 17.0.4、PGI 17.5 MPI: OpenMPI、Intel MPI、SGI MPI (MPT) 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/mod	lules/modulefiles/compiler
cuda/8.0.44(default)	intel/16.0.4.258	pgi/17.5(defaut)
cuda/8.0.61	intel/17.0.4.196(default)	pgi/17.10
	/apps/t3/sles12sp2/i	modules/modulefiles/mpi
intel-mpi/17.3.196(defa	ult) openmpi/1.10.2-pgi	2017 openmpi/2.1.2-pgi2017
mpt/2.16	openmpi/2.1.1(defa	ult)
	-/apps/t3/sles12sp2/m	odules/modulefiles/tools
allinea/7.0.5	intel-vtune/17.4.0.5187	98 perfsuite/1.1.4
intel-ins/17.1.3.510645	papi/5.5.1	
intel-itac/17.3.030	perfboost/2.16	

※ gcc は module load しなくても利用できます。最新版が必要な場合はご自分でご用意ください。

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 の一覧です。 使用するプログラムに応じた module を load します。 ISVのプログラムによっては学外の方はご利用になれません。

※ 一部のソフトウェアは P100 GPU に対応しておりませんので CPU でお使いください。

	/apps/t3/sles12	sp2/modules/modulefiles/isv	
abaqus/2017	ansys/R18.2	gaussview/6	mathematica/11.2.0
amber/16	avs/8.4	lsdyna/R9.1.0	matlab/R2017a(default)
amber/16_cuda	comsol/53	lsprepost/4.3	matlab/R2017b
amber/16up10	dytran/2017	maple/2016.2	nastran/2017.1
amber/16up10_cuda	gaussian16/A03	marc_mentat/2017	patran/2017.0.2
ansys/R18.1(default)	gaussian16_linda/A03	<pre>mathematica/11.1.1(default)</pre>	schrodinger/Feb-17
	/apps/t3/sles12	sp2/modules/modulefiles/free	
a2ps/4.14	gromacs/2016.3(default)	nccl/1.3.4	texlive/20170704
cp2k/4.1	gromacs/4.6.7	openfoam/4.1	tgif/4.2.5
cudnn/4.0	hadoop/2.8.0	paraview/0_5.2.0	tinker/8.1.2
cudnn/5.1	hdf5/1.10.1	paraview/5.0.1	tmux/2.5
cudnn/6.0	<pre>imagemagick/7.0.6</pre>	petsc/3.7.6/complex	visit/2.12.3
cudnn/7.0	jdk/1.8.0_131	petsc/3.7.6/real	vtk/6.1.0(default)
fftw/2.1.5	jdk/1.8.0_144	php/7.1.6	vtk/8.0.0
fftw/3.3.6	lammps/31mar2017	pov-ray/3.7.0.3	xpdf/3.04
gamess/apr202017r1	llvm/3.9.1	python-extension/2.7 (de	fault)
gimp/2.8.22	mesa/13.0.3	python-extension/3.4	
gnuplot/5.0.6	namd/2.12	r/3.4.1	

※ アプリケーションのバージョンアップにより module 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。

ジョブの実行

ジョブの実行についての概要

- ジョブスケジューラはUNIVA Grid Engine(UGE)
- 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)...
 - TSUBAME2.5ではSキュー、Gキューなどとして使用
- ・ジョブの投入は qsub コマンド
 - 「ジョブスクリプト」を用意する必要
 - TSUBAME2.5 と文法が異なる
- 予約キューがより柔軟に(準備中)
 1時間単位
- qrsh コマンドによるインタラクティブ利用が可能 計算ノードにログインして利用できます

資源タイプー覧

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	240	4	1.00
Н	h_node	14	120	2	0.50
Q	q_node	7	60	1	0.25
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、現在は最大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
 - TSUBAME3.0利用講習会

計算ノードのインタラクティブ利用

「プログラムの編集・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

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)
- →計算ノードにログインし、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分以上利用する場合は、-gオプションにてTSUBAMEグループを指定する。
- 例: qrsh –l f_node=1 –l h_rt=1:00:00 –g tgx-17IXX
- f_node 以外の qrsh でX転送を利用する方法

qrsh -l s_core=1,h_rt=0:10:00 -pty yes -display \$DISPLAY -v TERM /bin/bash

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,お試しだと 0:10:00(時間延長オプションはなくなりました)
- 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

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
 - 拡張子は.sh

#!/bin/sh

#\$ -cwd

- #\$ -| [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) 省略時は -5, -4 が中間、-3 が優先度高

-cwd, -l, -p等は、このスクリプトに書く代わりに、qsubのオプションとしてもok 他のオプションについては、利用の手引き4.2.2を参照

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



module load cuda pgi

※ PGI のオプションは -ta=tesla,cc60 もしくは pgfortran -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	
#\$ −I f_node=1	— 資源タイプ F を 1ノード使用
#\$ -I h_rt=1:00:00	
#\$ −N openmp	
./etc/profile.d/modules.sh	
module load cuda/8.0.44	ジョンを明云的に指定する場合
module load intel/17.0.4.196	
export OMP_NUM_THREADS=28	— ノード内に28スレッドを配置
./a.out	

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=4 <	── 資源タイプ F を 4ノード使用
#\$ -l h_rt=1:00:00	
#\$ -N intelmpi	ノードリストは次の変数から取得
./etc/profile.d/modules.sh	\$PE_HOSTFILE
module load cuda	例: cat \$PE_HOSTFILE awk …
module load intel	
module load intel-mpi	Intel MPI 環境の設定
mpirun –ppn 8 –n 32 ./a.out	― ノードあたり8プロセスで32並列

OpenMPIでは、
 9行目: module load openmpi
 10行目: mpirun –npernode 25 –n 100 ./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	
#\$ −I f_node=4	── 資源タイプ F を 4ノード使用
#\$ -l h_rt=1:00:00	
#\$ −N HyBrid	
. /etc/profile.d/modules.sh	
module load cuda	
module load intel	
module load intel-mpi	── Intel MPI 環境の設定
export OMP_NUM_THREADS=28	― ノード内に28スレッドを配置
mpirun –ppn 1 –n 4 ./a.out 🗸	└── ノードあたり MPI 1プロセス、
	〜 全部で4プロセスを使用する

• OpenMPI だと、

9行目: module load openmpi

11行目: mpiexec.hydra –npernode 1 –n 4 ./a.out

TSUBAME3.0利用講習会

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

 -g [TSUBAMEグループ] については、ジョブスクリプト 内ではなく、ここで指定

- 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-17IXX ./job.sh

→成功すると、

Your job 123456 ("job.sh") has been submitted

のように表示され、ジョブID(ここでは123456)が分かる

ステップ3:ジョブの状態確認

qstat [オプション]

例: qstat

→ 自分の現在のジョブ情報を表示

job-ID jclass	prior	name	user slots ja	state -task-ID	submit/start	at queue	
123456 all.q@r8i2	0.55500 17	job.sh ド名	touko-t-a	a r	08/03/2017 12 r は実行中、qw	 :17:41 7 / は待機中	-

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 "*" にて全てのジョブを表示します。

TSUBAME3.0利用講習会

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 123456

※ 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。

ステップ5: ジョブ結果の確認

- ジョブが(printfなどで)出力した結果は通常、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.o123456 と job.sh.e123456
- 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

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利用
 - T2は1日単位 → T3では1時間単位での予約が可能に
 - 予約機能により24時間以上のジョブが利用可能になります
 - 詳細につきましては下記手引きをご参照ください

•6.1.3. ノードの予約設定 (現在使用できません)

http://www.t3.gsic.titech.ac.jp/docs/TSUBAME3.0_Portal_Users_Guide.html

- TSUBAME2.5とTSUBAME3.0の主な違いについて

http://www.t3.gsic.titech.ac.jp/node/162



- TSUBAME3.0 ではログインノードおよび各計算ノードから
 外のネットワークへの直接のアクセスが可能となりました。
- 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: lammps

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

例2: namd

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

- ・ TSUBAME2.5 の t2a006170、io-t2 に相当するノードはなくなりました。
- ベンダーソフトなどで学外のライセンスサーバーを直接利用可能です。

ストレージの利用(1)

- ホームディレクトリ
 - 各ユーザに与えられる。25GBまで無料で使用可能 /home/?/\$username
- 高速ストレージ
 - TSUBAMEグループ管理者が購入すると、グループにディ レクトリが与えられる(課題代表者が購入可能)
 - TSUBAME2.5の/work0,/work1に相当
 - 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ファイルまで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-17IXX /gs/hs0 にて参照可能

ストレージの利用(2)

- ローカルスクラッチ領域
 - ノードごと・ジョブごとに一時利用できる領域
 - ・ TSUBAME2.5の/scrに相当
 - ・ ジョブ終了時に消える
 - ・ノードあたり約1.9TB、ストレージの中で最高速
 - ディレクトリ名は、ジョブごとに異なる
 - →環境変数 \$TMPDIR を見る必要
 - 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
- 共有スクラッチ領域
 - 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)
 - /beeond ディレクトリ (BeeGFS On Demand) 2ノードで約 3.7TB
 #\$ -v USE_BEEOND=1 とジョブスクリプト内に記述する

課金について: TSUBAME2.5と3.0の違い

・グループ区分: tgh-, tgi-, tgj-(共同利用)

TSUBAME2.5(公開)	1口	3,000pt	120,000円
(非公開)	1口	3,000pt	480,000円
TSUBAME3.0(公開)	1口	3,600,000pt	100,000円
(非公開)	1□	3,600,000pt	200,000円

ーロ 3000ポイント から 1000ノード時間相当へ ポイントはノード秒で表示することになりました。 2.5から 3.0 へのポイント変換レートは 1 : 0.4 です。

ポイントの消費式

ジョブ毎の使用ポイント =ceil(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))

資源タイプ	F	Н	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 =利用月数×利用可能容量(TB)×36,000(10ノード時間相当)

詳細は、TSUBAME3.0ウェブサイト上の規約をご覧ください http://tsubame.gsic.titech.ac.jp/sites/default/files/T3kiyaku.pdf

関連リンク

ログインノード

共同利用推進室 共同利用推進室 FAQ 利用講習会資料

TSUBAME2.5計算サービス TSUBAME3.0ウェブページ TSUBAME3.0利用 FAQ TSUBAME3.0利用状況 TSUBAME3.0利用ポータル login.t3.gsic.titech.ac.jp

http://www.gsic.titech.ac.jp/tsubame http://www.gsic.titech.ac.jp/kyodou/FAQ http://www.gsic.titech.ac.jp/kyodou/beginners_course

http://tsubame.gsic.titech.ac.jp http://www.t3.gsic.titech.ac.jp http://www.t3.gsic.titech.ac.jp/faq http://www.t3.gsic.titech.ac.jp/monitoring https://portal.t3.gsic.titech.ac.jp/ptl

TSUBAME3.0利用の手引き

http://www.t3.gsic.titech.ac.jp/docs/TSUBAME3.0_Users_Guide.html TSUBAME3.0利用ポータル利用の手引き http://www.t3.gsic.titech.ac.jp/docs/TSUBAME3.0 Portal Users Guide.html

UNIX 入門 チューニング資料 http://tsubame.gsic.titech.ac.jp/docs/guides/UNIX/UNIX.pdf http://tsubame.gsic.titech.ac.jp/docs/guides/UNIX/tune.pdf

採択課題一覧 HPCI産業利用 http://www.gsic.titech.ac.jp/node/60 http://www.gsic.titech.ac.jp/hpci-sangyo

TSUBAME3.0利用講習会

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。

TSUBAME2.5 アーキテクチャ



TSUBAME3.0 アーキテクチャ

