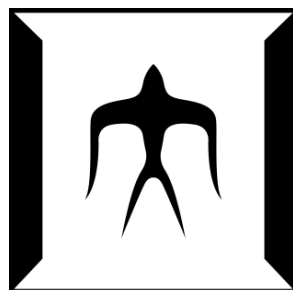


# TSUBAME3.0利用説明会

[www.t3.gsic.titech.ac.jp](http://www.t3.gsic.titech.ac.jp)



平成29年度版 (Rev.20170825)

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) 2017 GSIC All Rights Reserved.

# CONTENTS

- 歴史・概要・特徴
- ハードウェア・ソフトウェア
- 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
- データ移行について
- リンク一覧

# 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  
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年	2008年	2009年	2010年	2011年	2012年	2013年	2014年	2015年	2016年	合計
学術利用	HPCI	-	-	-	-	-	6	5	10	14	5	40
	JHPCN	-	-	-	4	6	5	11	10	10	12	58
	有償利用	-	-	1	4	9	14	17	22	23	25	115
産業利用	無償利用	11	15	15	8	10	12	21	17	13	15	137
	有償利用	成果公開	-	-	3	6	7	9	8	10	8	59
		成果非公開	-	-	2	7	6	4	10	12	13	64

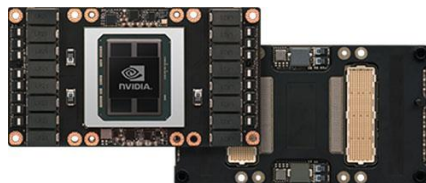
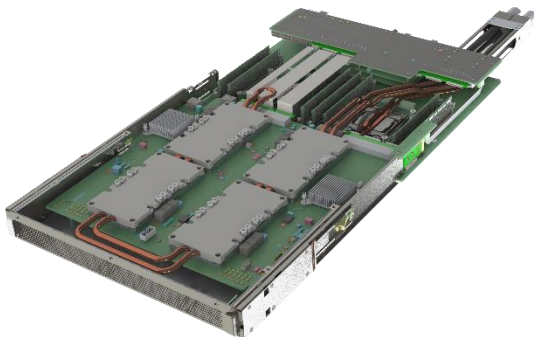
※ 2017年8月より運用開始  
Green500 世界1位(2017/6)

# TSUBAM3.0 概要

## Compute Node

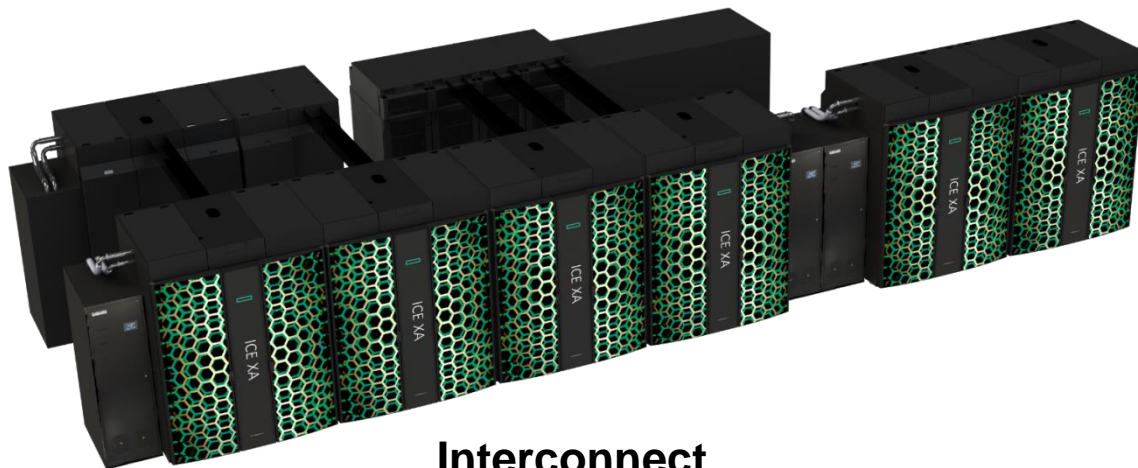
CPU: Intel Xeon E5-2680v4 (14core) × 2  
GPU: NVIDIA Tesla P100 × 4

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



## System

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



## Interconnect

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

## Operating System

SUSE Linux Enterprise Server 12 SP2

## Job Scheduler

UNIVA Grid Engine

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

# TSUBAME3.0の特徴

- 莫大な演算性能・巨大ストレージ
  - 計12PFlops(倍精度)の演算性能・・・TSUBAME2.5の2.1倍
    - AI分野で注目されている、半精度なら47PFlops!
  - 計16PBのストレージ容量・・・TSUBAME2.5の2.8倍
- 省エネスパコン世界一
  - 2017年6月Green500ランキングにおいて、世界一を受賞！
  - 超省エネ冷却技術
- クラウド技術による、柔軟な環境を提供 (準備中)
- TSUBAME2.5のソフトウェアはほぼそのまま稼働
  - おなじみ、Intel CPU + NVIDIA GPU + Linux OS の組み合わせ
  - ネットワーク I/F が InfiniBand から Omni-Path に変更された

# TSUBAME2.5からの移行のポイント

以下では、TSUBAME3.0をT3, TSUBAME2.5をT2と呼ぶ

- T3のアカウントは新規に作成が必要
- T2上のデータは、T3へ10月末までに各自でコピー
- スケジューラが異なる → ジョブ投入方法が変わる
- moduleコマンドに慣れる必要 (MPI, CUDA, ISVアプリ...)
- 各ソフトウェアのバージョン違いに注意
- T2の(旧)TSUBAMEポイントとT3の(新)TSUBAMEポイントは別もの、それぞれ購入する必要あり(学外の方はポイント移行対応)
- ジョブの最長実行時間 96時間 ⇒ 24時間。  
24時間超のジョブは、予約キューへの移行を想定
- 自動課金の廃止
- 定額キュー(T2のVキュー)廃止
- Windows HPC廃止

# TSUBAME3.0の計算ノード

## TSUBAME2.5との比較

	TSUBAME2.5	TSUBAME3.0
計算ノード数	1400台 HP SL390s	540台 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

# 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 × 240 (10月まで)	540 (現在 320)



# 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
  - 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
    - グループあたり最大300TB(予定)
- ローカルスクラッチ領域 (各ノード固有)
  - ノードあたり1.9TB
  - 最も高速だが、ジョブ終了時に消える

# 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
  - [www.t3.titech.ac.jp](http://www.t3.titech.ac.jp) 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 (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>

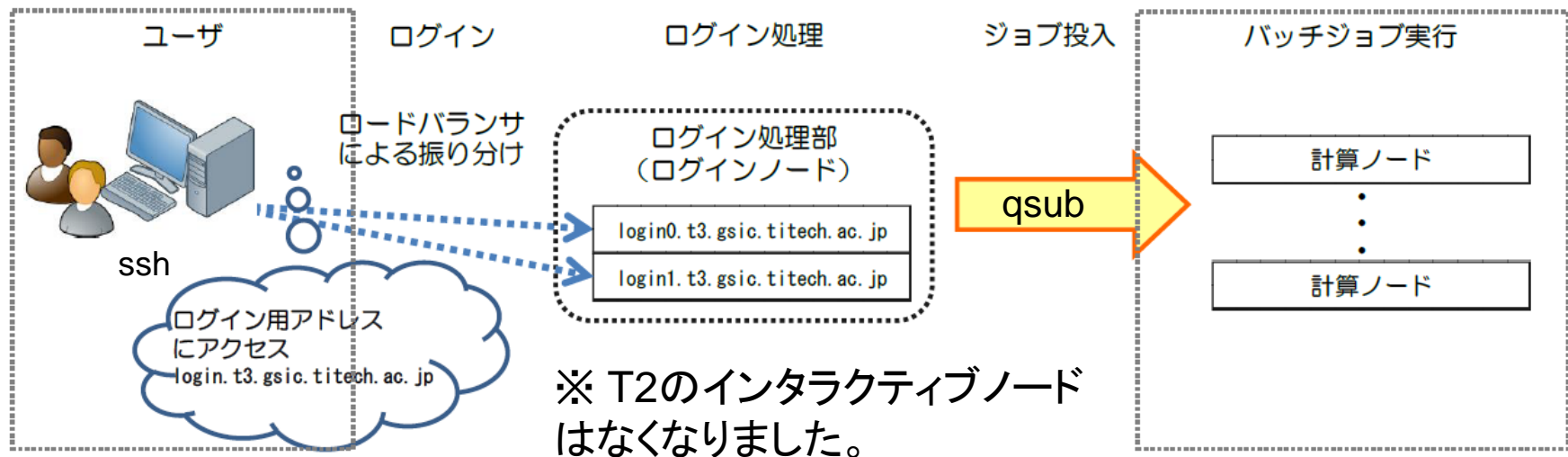
# 利用できるサービス

- 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 (準備中)
- 有償サービス
  - 研究目的の大規模計算(従量制、予約ジョブも)
  - 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(予定)

# 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  $\doteq$  1TSUBAME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

# 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`)

# 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 gnuplot`
  - 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

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/modules/modulefiles/compiler ----  
cuda/8.0.44(default)          intel/16.0.4.258          pgi/17.5  
cuda/8.0.61                   intel/17.0.4.196(default)
```

```
----- /apps/t3/sles12sp2/modules/modulefiles/mpi -----  
intel-mpi/17.3.196(default) openmpi/1.10.2-pgi2017  
mpt/2.16                      openmpi/2.1.1
```

```
----- /apps/t3/sles12sp2/modules/modulefiles/tools -----  
allinea/7.0.5                 intel-itac/17.3.030      papi/5.5.1  
intel-ins/17.1.3.510645      intel-vtune/17.4.0.518798 perfsuite/1.1.4
```

# 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 module

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/sles12sp2/modules/modulefiles/isv -----
abaqus/2017          comsol/53          lsdyna/R9.1.0      matlab/R2017a
amber/16             dytran/2017       lsprepost/4.3      nastran/2017.1
amber/16_cuda        gaussian16/A03     maple/2016.2       patran/2017.0.2
ansys/R18.1          gaussian16_linda/A03 marc_mentat/2017   schrodinger/Feb-17
avs/8.4              gaussview/6        mathematica/11.1.1

----- /apps/t3/sles12sp2/modules/modulefiles/free -----
a2ps/4.14            gromacs/2016.3    openfoam/4.1        tgif/4.2.5
cp2k/4.1             hadoop/2.8.0      paraview/5.0.1      tinkert/8.1.2
cudnn/5.1            hdf5/1.10.1       petsc/3.7.6/complex tmux/2.5
cudnn/6.0            imagemagick/7.0.6  petsc/3.7.6/real    visit/2.12.3
fftw/2.1.5           jdk/1.8.0_131     php/7.1.6           vtk/6.1.0
fftw/3.3.6           jdk/1.8.0_144     pov-ray/3.7.0.3     xpdf/3.04
gambss/apr202017r1   lammmps/31mar2017 python-extension/2.7
gimp/2.8.22          namd/2.12          r/3.4.1
gnuplot/5.0.6        nccl/1.3.4         texlive/20170704
```

# 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 (1/2), q\_node (1/4)...
  - 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	240	4	1.00
H	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
  - 540ノードから各資源タイプを割り当てる(現在320)

# 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-17lxx`

# 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
```

```
#$ -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)  
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

```
#!/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
```

s\_gpu を1個使用 (GPU利用の最小単位)

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

# 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=1:00:00
#$ -N openmp
. /etc/profile.d/modules.sh
module load cuda/8.0.44
module load intel/17.0.4.196
export OMP_NUM_THREADS=28
./a.out
```

← 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する場合

← ノード内に28スレッドを配置

# 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
#$ -l h_rt=1:00:00
#$ -N intelmpi
. /etc/profile.d/modules.sh
module load cuda
module load intel
module load intel-mpi
mpirun -ppn 8 -n 32 ./a.out
```

← 資源タイプ F を 4ノード使用

← Intel MPI 環境の設定

← 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

※ 1ノード25プロセスだと  
4ノードで 100並列の計算

# ジョブスクリプトの例(4)

## ・T2のスクリプトを T3 用に書き換える

例：Sキュー12コアx2、GPUx2、メモリ40GBを10分お試しで実行

```
$ t2sub -q S -l select=2:ncpus=12:gpus=2:mem40gb -l walltime=0:10:00 ...
```

```
$ cat t3.sh
```

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

```
#$ -cwd
```

```
#$ -l h_node=2          ※ h_node で 14コア、2 GPU、メモリ 120GB
```

```
#$ -l h_rt=0:10:00      ※ walltime を 10分に設定
```

```
#$ -N test
```

```
. /etc/profile.d/modules.sh
```

```
module load prog1 prog2  ※ 必要なモジュールをロード
```

```
$ qsub t3.sh          ※ -g で課題グループを設定せずに qsub を実行
```

# 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-17lxx ./job.sh

→成功すると、

Your job 3456 ("job.sh") has been submitted

のように表示されて、ジョブID(ここでは3456)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      prior    name          user            state submit/start at     queue  
jclass                      slots ja-task-ID  
-----  
-----  
3456 0.55500 job.sh      touko-t-aa      r      08/03/2017 12:17:41  
all.q@r8i2n7                                     7  
ノード名                                         r は実行中、qw 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。

# 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 3456

※ 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.o3456` と `job.sh.e3456`
- 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のHキューに相当するが、1日単位 → 1時間単位とし、使いやすくする予定
  - 通常ジョブの最大時間である、24時間以上利用可能に

現在準備中です

- 現時点ではジョブの実行時間は24時間までとなります。

# データ転送など

- 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 : lammmps

```
$ 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。

# 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]

# 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、容量はローカルスクラッチと共用
  - /beond ディレクトリ (bee gfs on demand) 2ノードで約 3.7TB

# 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口	3,600,000pt	200,000円

一口 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

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

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

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

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>

# データ移行概要

- ・データ移行はTSUBAME2.5のインタラクティブノードから行います。計算ノードからは実行できません。
- ・TSUBAME2.5のサービス終了(10月末日)とともに、既存のデータにはアクセスできなくなります。
- ・10月は駆け込み需要で転送性能が低下する可能性があります。できるだけ9月中に終わるよう、なるべく早くデータを移行してください。
- ・TSUBAME2.5 と TSUBAME3.0 では、ユーザー名、グループ名が異なります。十分にご注意ください。(共同利用のユーザー様)

ユーザー名	数字4桁に	グループ名	T3の区分名
17IABXXX	→ 17IA0XXX	t2g-17IAB	→ <b>tgh</b> -17IAB
17IBCXXX	→ 17IB0XXX	t2g-17IBC	→ <b>tgi</b> -17IBC
17ICDXXX	→ 17IB0XXX	t2g-17ICD	→ <b>tgj</b> -17ICD

# データ移行手順

- ・あらかじめ TSUBAME3.0ポータルにてパスワードを設定してください
- ・TSUBAME2.5 にログインします  
\$ ssh <T2\_UserName>@login-t2.g.gsic.titech.ac.jp
- ・作業している TSUBAME2.5 のインタラクティブノードの  
ノード名を記録しておきます(あとでログインする場合)  
\$ hostname  
t2a0061xx
- ・データ転送用ホスト t3-data を経由してファイルを転送します  
\$ rsync -av --progress --log-file=rsynclog.txt (一行です)  
/work1/t2g-17lxx <T3\_UserName>@t3-data:/gs/hs0/tgx-17lxx  
<T2のディレクトリ> <T3のディレクトリ>
- ・大量のデータには rsync を推奨。少量のデータなら sftp、scp でも可  
\$ sftp <T3\_UserName>@t3-data  
\$ scp <UserData> <T3\_UserName>@t3-data:~/<UserDirectory>
- ・screen を使用する場合は以下の「データ移行概要」を参照してください。  
<http://www.gsic.titech.ac.jp/sites/default/files/datacopy.pdf>



# 関連リンク

ログインノード	<a href="http://login.t3.gsic.titech.ac.jp">login.t3.gsic.titech.ac.jp</a>
共同利用推進室	<a href="http://www.gsic.titech.ac.jp/tsubame">http://www.gsic.titech.ac.jp/tsubame</a>
共同利用推進室 FAQ	<a href="http://www.gsic.titech.ac.jp/kyodou/FAQ">http://www.gsic.titech.ac.jp/kyodou/FAQ</a>
利用講習会資料	<a href="http://www.gsic.titech.ac.jp/kyodou/beginners_course">http://www.gsic.titech.ac.jp/kyodou/beginners_course</a>
TSUBAME2.5計算サービス	<a href="http://tsubame.gsic.titech.ac.jp">http://tsubame.gsic.titech.ac.jp</a>
TSUBAME3.0ウェブページ	<a href="http://www.t3.gsic.titech.ac.jp">http://www.t3.gsic.titech.ac.jp</a>
TSUBAME3.0利用 FAQ	<a href="http://www.t3.gsic.titech.ac.jp/faq">http://www.t3.gsic.titech.ac.jp/faq</a>
TSUBAME3.0利用状況	<a href="http://www.t3.gsic.titech.ac.jp/monitoring">http://www.t3.gsic.titech.ac.jp/monitoring</a>
TSUBAME3.0利用ポータル	<a href="https://portal.t3.gsic.titech.ac.jp/ptl">https://portal.t3.gsic.titech.ac.jp/ptl</a>
TSUBAME3.0利用の手引き	<a href="http://www.t3.gsic.titech.ac.jp/docs/TSUBAME3.0_Users_Guide.html">http://www.t3.gsic.titech.ac.jp/docs/TSUBAME3.0_Users_Guide.html</a>
TSUBAME3.0利用ポータル利用の手引き	<a href="http://www.t3.gsic.titech.ac.jp/docs/TSUBAME3.0_Portal_Users_Guide.html">http://www.t3.gsic.titech.ac.jp/docs/TSUBAME3.0_Portal_Users_Guide.html</a>
UNIX 入門	<a href="http://tsubame.gsic.titech.ac.jp/docs/guides/UNIX/UNIX.pdf">http://tsubame.gsic.titech.ac.jp/docs/guides/UNIX/UNIX.pdf</a>
チューニング資料	<a href="http://tsubame.gsic.titech.ac.jp/docs/guides/UNIX/tune.pdf">http://tsubame.gsic.titech.ac.jp/docs/guides/UNIX/tune.pdf</a>
採択課題一覧	<a href="http://www.gsic.titech.ac.jp/node/60">http://www.gsic.titech.ac.jp/node/60</a>
HPCI産業利用	<a href="http://www.gsic.titech.ac.jp/hpci-sangyo">http://www.gsic.titech.ac.jp/hpci-sangyo</a>

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利用の手引き TSUBAMEポータル 編

2017/08/25

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) 2010-2017 GSIC All Rights Reserved.

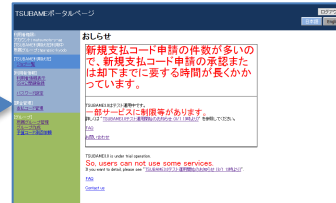
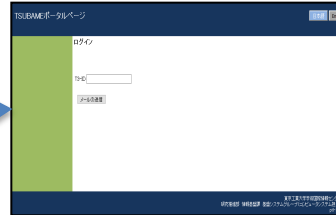
「TSUBAMEポータル利用の手引き」もご参照ください。

[http://www.t3.gsic.titech.ac.jp/docs/TSUBAME3.0\\_Portal\\_Users\\_Guide.html](http://www.t3.gsic.titech.ac.jp/docs/TSUBAME3.0_Portal_Users_Guide.html)

# TSUBAME3.0の利用開始手順



TSUBAME3.0利用者



TSUBAME3.0ポータル  
公開鍵の登録



1) Tsubameポータル  
にアクセスしメール送信

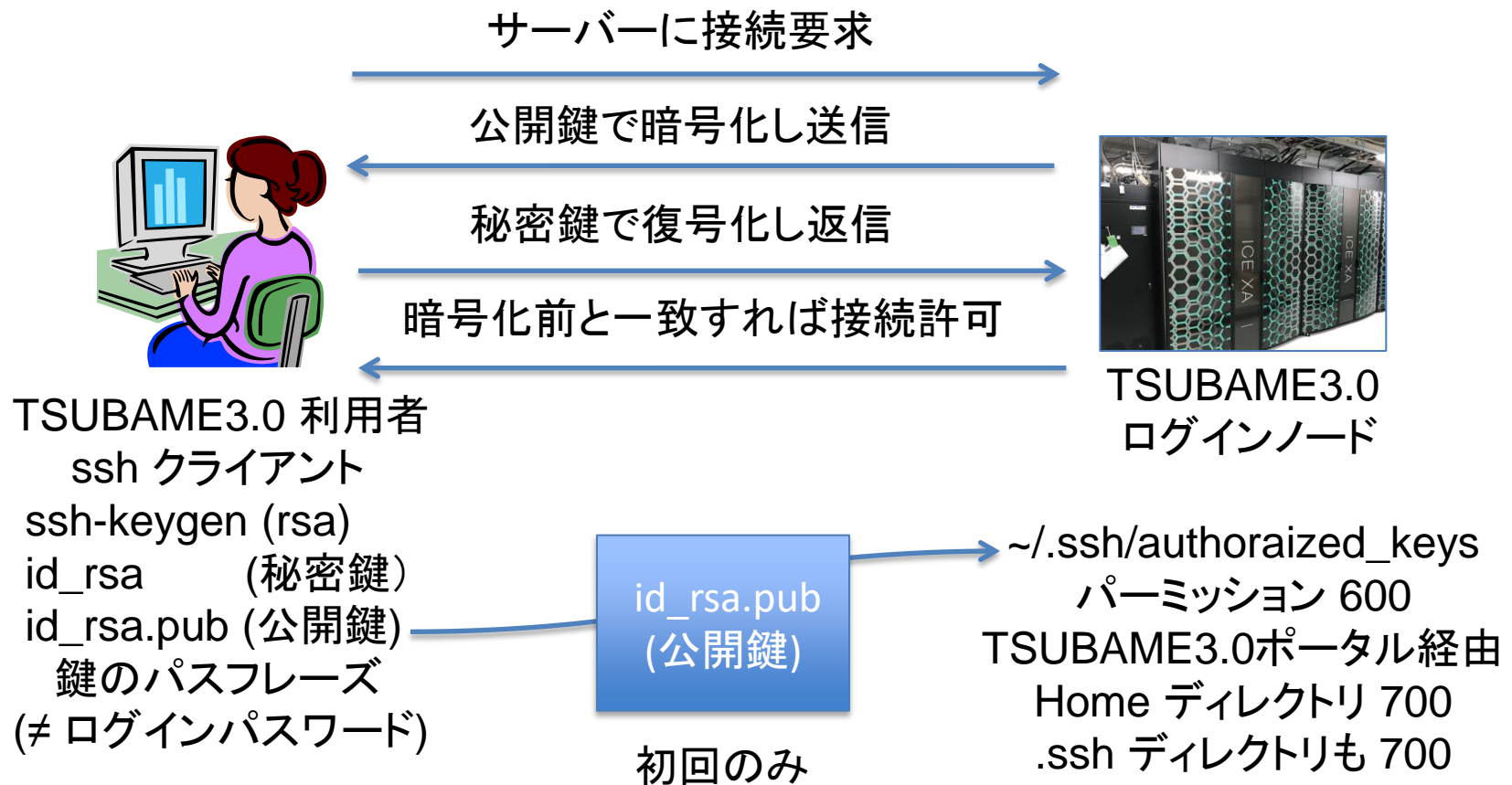
2) Webブラウザでアク  
セスし公開鍵を登録。

各種利用環境の設定と  
利用状況の確認

3) SSHによるログイン  
(公開鍵方式)

TSUBAME3.0  
利用開始

# 公開鍵認証によるアクセス



# TSUBAMEポータルページへのアクセス

TSUBAMEポータルページ

日本語English

ログイン

T3-ID

メールの送信

クリック→メールが届く→

TSUBAME3.0 ログインURL : TSUBAME3.0 Login URL - Mozilla Thunderbird

ファイル(F) 編集(E) 表示(V) 移動(G) メッセージ(M) ツール(T) ヘルプ(H)

受信 作成 チャット アドレス帳 前へ 次へ タグ

差出人 portal@t3.gsic.titech.ac.jp ☆

件名 TSUBAME3.0 ログインURL : TSUBAME3.0 Login URL

宛先 tokodai-taro@m.titech.ac.jp ☆

東工太郎様

TSUBAME3.0のログインURLをお知らせします。  
<[https://portal.t3.gsic.titech.ac.jp/ptl/login/d-Y91\\_1qheL4tLTuE0uG3qaoygvrZYtLUFq3j18klUI-](https://portal.t3.gsic.titech.ac.jp/ptl/login/d-Y91_1qheL4tLTuE0uG3qaoygvrZYtLUFq3j18klUI-)>

有効期限は30分です。

-----

TSUBAMEポータルページにアクセスします。

<https://portal.t3.gsic.titech.ac.jp/ptl/>

T3-ID に自分の TSUBAME3.0 のアカウントを入力し

メールの送信 をクリックします。

あらかじめ登録されたメールアドレスに URL が送られます。

メールで送られてきた URL をブラウザでアクセスすることで

TSUBAMEポータルページにアクセスできます。

# TSUBAMEポータルページのトップ

TSUBAMEポータルページ

ログアウト

日本語English

利用者権限:  
アカウント: matsumoto-y-ag  
[TSUBAME利用状況]利用中  
所属グループ: tea-gsic-kyodo

[TSUBAME利用状況]  
[ジョブ一覧](#)

[利用者情報]  
[利用者情報表示](#)  
[SSH公開鍵登録](#)  
[パスワード設定](#)

[課金管理]  
[支払コード管理](#)

[グループ]  
[所属グループ管理](#)  
[グループ作成](#)  
[予算コード承認依頼](#)

お知らせ

新規支払コード申請で、新規支払コードは却下までに要しています。

TSUBAME3.0はテスト運用中です。  
一部サービスに制限等  
詳しくは“[TSUBAME3.0テスト運用開始のお知らせ](#)”  
[FAQ](#)  
[お問い合わせ](#)

TSUBAME3.0 is under trial operation.  
So, users can not use  
If you want to detail, please see “[TSUBAME](#)”  
[FAQ](#)  
[Contact us](#)

初めにすること

◎ SSH公開鍵のアップロード  
○ パスワードの設定

その後も利用する項目

○ パスワード設定(半年ごとに変更)  
○ ジョブ一覧 (ジョブの確認)  
○ 予約システム (キューの予約: 予定)

利用しない項目

■ 利用者情報表示  
■ 支払コード管理  
■ 所属グループ管理  
■ グループ作成  
■ 予算コード承認依頼

# トップページ > SSH公開鍵登録

TSUBAMEポータルページ

ログアウト  
日本語 English

利用者情報:  
アカウント: matsumoto-y-ae  
[TSUBAME利用状況] 利用中  
所属グループ: tea-gsic-kyodo

[TSUBAME利用状況]  
ジョブ一覧

[利用者情報]  
利用者情報表示  
SSH公開鍵登録  
パスワード設定

[安全管理]  
支払コード管理

[グループ]  
所属グループ管理  
グループ作成  
予算コード承認依頼

### SSH公開鍵追加

SSH公開鍵をコード入力または、ファイルアップロードにて追加することができます。  
SSH公開鍵コードを入力して追加ボタンを押してください。

#### 公開鍵コード入力

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQADM6fvLZDcwCqJz2B2eyvT4L991jdE6EzN0YniG/KRpF1LU0DuZH  
TNYyUczzhZSrW12FAWwsDoekaBWUgyegQLGZFpn4LkM9IaV12Yv31437vTVH/h6gkj10rJQSVnRr19k3GbPb457FU  
R7S1Kd5BkThM658MowGdsr8j5KhLF10iq7nd1o9MGw3m40t+C3NN7xdwTRiRv3+jTSgAabW1qSCz5NGvH1DZzt r5  
v0uNXXCxp0NN5de0gIBbwCGaLeL9nfGvQTDRExvi16NvJQNM8zbS1c3ZtQrk5wEopfbAmgaB1sVgW9P29dabNr01q  
t6NQX8fck2Fwk9o1FMUQvail ymatsumo@hp800
```

登録するSSH公開鍵をアップロードしてください。

SSH公開鍵ファイル:  参照... ファイルが選択されていません。

追加  
アップロード

- ・ 学外ネットワークからTSUBAME3.0 への SSH接続は **公開鍵認証のみ**となります。従ってSSH公開鍵アップロードが必要です。
- ・ 公開鍵を作成し、TSUBAMEポータルページからアップロードします。
- ・ アップロードした公開鍵は  
~/.ssh/authorized\_keys ファイルの末尾に追加されます。
- ・ Mac の ssh-keygen を用いた場合は、ファイル id\_rsa.pub の内容をアップロードします。  
.ssh は、コマンド + SHIFT + “.” で出現

公開鍵の作成は教育用電子計算機システムのウェブページにある「2. 公開鍵の作成」(<http://edu.gsic.titech.ac.jp/?q=node/46>)の PuTTYgenでの作成方法をご参照ください。  
TSUBAME利用手引き(学外からlogin) [http://tsubame.gsic.titech.ac.jp/login#key\\_auth](http://tsubame.gsic.titech.ac.jp/login#key_auth)



# トップページ > パスワード設定

TSUBAMEポータルページ

ログアウト

日本語English

利用者権限:  
アカウント: matsumoto-y-ag  
[TSUBAME利用状況]利用中  
所属グループ: tea-gsic-kyodo

[TSUBAME利用状況]

[ジョブ一覧](#)

[利用者情報]

[利用者情報表示](#)  
[SSH公開鍵登録](#)

[パスワード設定](#)

[課金管理]

[支払コード管理](#)

[グループ]

[所属グループ管理](#)  
[グループ作成](#)  
[予算コード承認依頼](#)

TSUBAMEパスワードの設定

新パスワード

新パスワード再入力

登録

TSUBAMEのパスワードを設定します。  
(TSUBAMEのコマンドラインでは変更できません)  
パスワードは以下のルールに従う必要があります。

- 8文字以上32文字以下
- 数字, 英字の小文字と大文字それぞれを一回以上使うこと
- アカウント名や姓名を含まないこと
- 英単語など容易に憶測できる文字列でないこと
- 3回以上同じ文字を連続して使用しないこと
- アカウント名に含まれる連続する3文字を使用しないこと
- 同じ文字をパスワード長の半分以上の回数使用しないこと

注) TSUBAMEポータルのパスワードは TSUBAME のログインパスワードです。  
※ パスワードを忘れた場合は再度メールを送信して最初から設定できます。

# 公開鍵・秘密鍵の作成

## 【公開鍵の作成例】

Mac にて ターミナルソフトを起動します。[移動] → [ユーティリティ] → [ターミナル]

```
Mac — bash — 80x24
pcm062099:~ guest440$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/usr0/guest440/.ssh/id_rsa):
Created directory '/home/usr0/guest440/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/usr0/guest440/.ssh/id_rsa.
Your public key has been saved in /home/usr0/guest440/.ssh/id_rsa.pub.
The key fingerprint is:
5b:5b:f7:c7:0a:a9:76:61:aa:2e:77:82:06:b1:af:9c guest440@pcm062099
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
|      .
|     o S . . .
|    o  o 000 o
|   o .. .00. +
|  . .= o +... ..
| Eo +o=.. .
+-----+
pcm062099:~ guest440$
```

← ssh-keygen と入力する

← リターンする

← パスフレーズをつける

← 同じものを2回入力する

(5文字以上の任意の文字列)

**注) パスフレーズは表示されません**

※作成された鍵 (/home/usr0/guestNNN/.ssh ディレクトリ)

-rw-----	id_rsa	←秘密鍵
-rw-r--r--	id_rsa.pub	←公開鍵

# 公開鍵認証によるアクセス準備

## 【用意するもの】

- a. TSUBAMEへ接続するための機器(例:PC)
- b. SSHが利用できるソフトウェア(例:cygwin)
- c. 公開鍵
- d. 秘密鍵
- e. 秘密鍵を利用するためのパスフレーズ

## 【手順】

	内容	講習会環境
①	PCを用意する	センターの学生用端末 (Mac )
②	ssh が利用できるソフトウェアを用意し、①のPCへインストールする (Windows環境: bash、Tera Term、PuTTY など)	実習室の Mac にあらかじめインストールされているターミナルソフトウェアを使用します。
③	公開鍵、秘密鍵を作成する (Tera Term でも作成可能)	ssh-keygen コマンドを実行し、公開鍵と秘密鍵を作成します
④	秘認証を利用するためのパスフレーズを設定する	上記設定の際に秘密鍵に「パスフレーズ」を設定します
⑤	公開鍵をアップロードする アップロードには「 <a href="#">TSUBAMEポータル</a> 」を使用。	TSUBAME に公開鍵をアップロードします アップロードには「 <a href="#">TSUBAMEポータル</a> 」を使用。
⑥	ssh にて接続確認を行う login.t3.gsic.titech.ac.jp にアクセスできることを確認。 ssh のポート 22 が許可されていることを確認。	Mac のターミナル から ssh にて TSUBAME に接続します。 接続する際は、④で設定した「パスフレーズ」を入力します

※重要 ①～⑥については、自社環境にて再度実行していただく必要があります。

# ssh によるログイン

ターミナルソフトにて操作を行います。

\$ **ssh ログイン名@login.t3.gsic.titech.ac.jp**

← ssh ログイン名@ログインノード

The authenticity of host 'login-t2.g.gsic.titech.ac.jp (131.112.4.49)' can't be established.

RSA key fingerprint is be:2c:b4:06:47:b3:f2:4a:f4:f6:f0:80:87:ff:ae:f5.

Are you sure you want to continue connecting (yes/no)? yes

← “yes” と入力します

Warning: Permanently added 'login-t2.g.gsic.titech.ac.jp, 131.112.4.49' (RSA) to the list of known hosts.



← 先ほど設定したパスフレーズを入力  
注) パスフレーズは表示されません

Last login: Wed Jan 7 12:34:56 2015 from 192.168.34.xxx

ログイン名@t2a0061xx:~>

← 無事にインタラクティブノードにログインできました

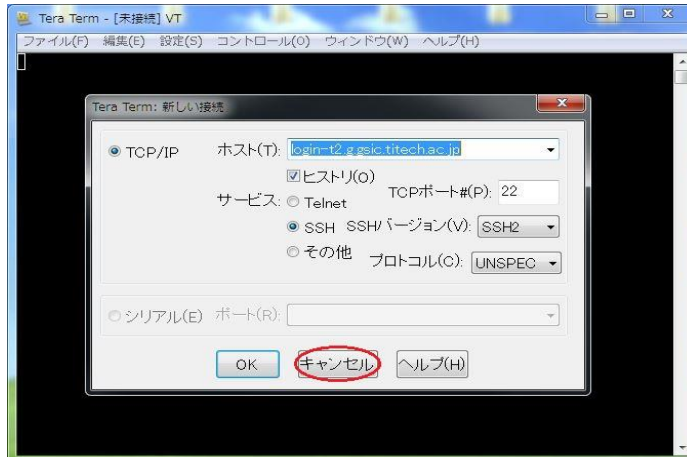
HPCIのアカウントでログインする場合も

**ログイン名@login.t3.gsic.titech.ac.jp** でログインできます。

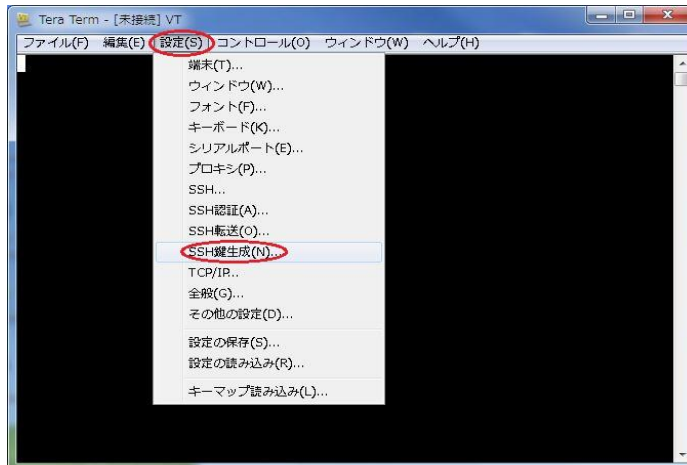
# 公開鍵・秘密鍵の作成

Tera Term による公開鍵・秘密鍵の作成の例

1. Tera Term を起動します。(例: スタートメニュー → [すべてのプログラム] → [Tera Term] を選択)
2. [新しい接続] 画面で [キャンセル] を選択します。



3. [設定] のメニューから [SSH鍵生成(N)] を選択し、鍵生成の画面を表示させます。



# 公開鍵・秘密鍵の作成

4. [TTSSH: 鍵生成] の画面で [生成(G)] を選択し鍵を生成します。



5. 鍵のパスフレーズを設定し、公開鍵および秘密鍵を保存します。



上下に同じパスフレーズを入力します

← 任意のフォルダに保存してください  
(id\_rsa.pub および id\_rsa )

# データ移行概要

- データ移行はTSUBAME2.5のインタラクティブノードから実施します
  - 計算ノードからはデータ移行できません
- TSUBAME2.5のサービス終了(10月末日)とともに、既存のTSUBAME2.5のファイルにはアクセスできなくなります
- 10月は駆け込み需要で転送性能が低下します。9月中に終わるよう、可能な限り速やかにデータを移行してください
  - データの転送性能を過信しないでください。平均数十MB/sec程度であると見積もってください
- TSUBAME2とTSUBAME3では、ユーザ名、グループ名が異なることがありますので、注意してください

# 事前準備

- TSUBAME3.0ポータルから、ユーザ・グループを、ご自身で作成してください。
- TSUBAME3.0のデータ移行先のグループディスクを事前にご登録ください。
- TSUBAME3.0ポータルを使用して、パスワードを設定してください。
- 本資料では便宜上、以下のように表記します。
  - TSUBAME2のユーザ名: <T2\_UserName>
  - TSUBAME3のユーザ名: <T3\_UserName>



# Step 1: TSUBAME2.5にログイン

- 作業サーバにTSUBAMEアカウントのログイン名でSSH接続

```
$ ssh loing-t2.g.gsic.titech.ac.jp -l <T2_UserName>
```

- 作業しているTSUBAME2のインタラクティブノードを記録する

```
$ hostname  
t2a00xxxx
```

この結果をメモしておくこと

# Step 2: screen起動

- screenを起動

```
$ screen -S <適当な名前>
```

screenを利用することでSSHセッションが解除されてもプロセスがkillされことなく作業を継続実行できました再接続することで画面を再開することができる。

```
# すでに存在するセッション一覧を表示
$ screen -ls
There is a screen on:
5270. (指定した適当な名前) (Detached)
```

```
# 存在するセッションに再接続する。
$ screen -x <指定した適当な名前>
```

```
# 作業を維持したままscreenから抜ける。
Ctrl-a + Ctrl-d (Ctrlを押しながらa、dと順番に押す。)
```

```
# screenを終了する。
$ exit
```

screenを起動した  
同一のノードでのみ有効

# Step3: データ移行コマンド

- rsyncコマンドを実行

一行で記述のこと

```
$ rsync -avE --no-p --progress --log-file=<ログファイル名>  
<T2上の同期元> <T3-UserName>@t3-data:<T3上の同期先>
```

Enter passphrase for key 'XXXXXXXX': (不明な場合は無視してEnter)

T3-UserName@t3-data's password: (T3のパスワードを入力してEnter)

例: T3ユーザであるtoukou-t-aaがT2上の/work0/t2g-XXX/YYYのディレクトリをT3上の/gs/hs0/t2a-ZZZの下にコピーし、そのログファイル名をrsynclog.txtとする場合

```
$ rsync -av --progress --log-file=rsynclog.txt  
/work0/t2g-XXX/YYY toukou-t-aa@t3-data:/gs/hs0/t2a-ZZZ
```

# Step4: データ移行中

- Screenの仮想コンソールを抜ける
  - メッセージ出力に関わらずCtrl-a + Ctrl-d (Ctrlを押しながらa、dと順番に押す。)
- 必要に応じてSSHセッションを切断する
  - SSHセッションを切断しても、データ移行は継続します。

```
$ exit
```

- rsyncの状況を確認する方法
  - ログファイルを確認する（簡易確認方法）

```
$ tail <ログファイル名>
```

- Screenの仮想コンソールに再接続する（詳細確認方法）

```
$ ssh loing-t2.g.gsic.titech.ac.jp -l <T2_UserName>
```

```
$ ssh <Step1で確認したノード名>
```

```
$ screen -x <指定した適当な名前>
```

```
~ メッセージ確認 ~
```

```
必要に応じて上記の手段でscreenの仮想コンソールから抜ける
```

# Step5:データ移行終了

- 終了判断

- 以下のようなメッセージがログファイルもしくはscreen上の仮想コンソール上に最後に出力されていれば終了と判断できる

```
sent VVVV bytes  received WWWW bytes  XXXXX bytes/sec  
total size is YYYYYY  speedup is Z.ZZZ
```

- 終了処理

- screen上の仮想コンソールの終了

```
$ ssh loing-t2.g.gsic.titech.ac.jp -l <T2_UserName>  
$ ssh <Step1で確認したノード名>  
$ screen -x <指定した適当な名前>  
~ メッセージ確認 ~  
$ exit
```

# データ移行のテクニック

- rsyncの途中でrsyncプロセスが死んだ場合  
(意図しないノードの故障時)でも、同じコマンドを実行することで途中から再開されます
- 大量のファイルを移行する場合は、ディレクトリを分割し、複数のrsyncを同時に実行することで、効果的に移行できます
- 1週間以上かかるようなrsyncコマンドの実行は避けてください
  - GSICがシステム構成を適宜最適化しますが、すでに実行中のプロセスには効果を与えません
- 可能な限り早急に移行を始めてください
  - 期限(10月末)までにデータ移行が終わらない場合は、データが消えます。バックアップはありません。
- データ移行はユーザの責任の元実施してください。移行によるトラブルはGSICは一切責任を負いません
  - ログを必ず確認してください