

HA-PACSを使う

1. ネットワークに接続する
2. HA-PACSにログインする
3. プログラムを作成するための準備と諸注意
4. バッチジョブを実行し、GPUを動かす
5. プロファイラ（nvvp）を使う

HA-PACSに関する情報（利用の手引など）は
<https://www2.ccs.tsukuba.ac.jp/hapacs/>
から参照可能です。
アクセスの際には利用者とパスワードが必要です。

1. ネットワークに接続する

- 会場の無線LAN（Wi-Fi）に接続する
 - 接続先アクセスポイント eduroam
 - 利用に必要な情報 ユーザーIDとパスワード（配布しているもの）
- よくあるトラブルと対策など
 - 既にeduroamというネットワークに接続したことがある場合
 - 現在も有効なアカウントを持っている場合
 - 持っているアカウント情報を使って接続して構いません
 - 現在は有効なアカウントを持っていない場合
 - 接続情報を削除してから、配布した情報で接続してください
- 手持ちのノートPCで接続できることを確認してください

2. HA-PACSにログインする

- 事前にメールでお伝えしたとおり、配布されているアカウント情報を使ってsshでログインする
 - `ssh username@hapacs.ccs.tsukuba.ac.jp`
 - nvvp（可視化プロファイラ）を使う場合には-Yオプションもつける
- よくあるトラブルと対策など
 - ネットワークにはちゃんとつながっているか確認する
 - 事前にテストしたPCと今日使っているPCが違う場合
 - SSH認証鍵の配置（パーミッションなど）に問題が無いか確認する
- 手持ちのノートPCでログインできることを確認してください

3. プログラムを作成するための準備と諸注意

- ディレクトリの扱いについて
 - ログイン時のディレクトリ（/**home**/GPUSEMINAR/利用者名）にはログイン時に最低限必要なファイルのみを置く
 - 作業には/work以下のディレクトリ（/**work**/GPUSEMINAR/利用者名）を使う
 - **注意：バッチジョブを実行できるのは/workの中からのみ**
- コンパイルの為の準備
 - moduleコマンドを使ってコンパイルのための準備をする
 - 環境変数PATHなどが書き換わり、コンパイラなどが使えるようになる

moduleの切替

- CUDA Cを使う場合
 - `module load cuda/7.5.18`
 - `nvcc`が利用可能となることを確認する
 - `$ which nvcc`
`/opt/CUDA/7.5.18/cudatoolkit/bin/nvcc`
- CUDA FortranまたはOpenACCを使う場合（PGIコンパイラを使う場合）
 - `module load pgi/16.4`
 - `pgcc`や`pgfortran`が利用可能となることを確認する
- その他のオプション
 - `module list`でload済のものを確認可能
 - `module avail`で利用可能なmoduleを確認可能

4. バッチジョブを実行し、GPUを動かす

- PBS Professionalを用いてジョブを実行する
- 用意するもの
 - ジョブスクリプト（シェルスクリプトファイル）
- 使用する主なコマンド
 - qsub ジョブを投入する（実行させる、キューに入れられる）
 - qstat ジョブの実行状況を確認する
 - qdel ジョブを削除する（削除したいジョブのIDを引数に指定する）

ジョブの実行例 (1/2)

• ジョブスクリプトの中身

#PBS -S /bin/bash	使用するシェル
#PBS -A GPUSEMINAR	プロジェクト名
#PBS -q gpuseminarbatch	キュー名
#PBS -l select=1:ncpus=1	使用するリソース量
#PBS -l walltime=0:01:00	最大実行時間（最大30分まで、例では1分）
. /opt/Modules/default/init/sh	module初期化
module load cuda/7.5.18	module設定（CUDA Cの例）
cd \${PBS_O_WORKDIR}	ジョブを投げたディレクトリへ移動（必要）
./simple1.out	実行

その他の主なオプションなど

-N ジョブ名（qstatで表示される名前などに影響）

-o outfile

-e errfile 出力されるファイル名を任意に変更可能

-j oe エラー出力と標準出力をまとめる

ジョブの実行例 (2/2)

- 投入

- qsub job.sh

- 確認

- qstat 利用者名 状態 (R：実行中、Q：待ち状態、など)

- 末尾に

852512.pbs-1 job.sh user 0 R gpuseminarbatch
が表示されたと仮定すると、852512がジョブID

- qstat 852512

- 対象のジョブに関する様々な情報が表示される

- 削除

- qdel 852512

- 投入したジョブに問題があることがわかったときなどに使う

動作確認

- あらかじめ用意されているCUDAプログラム（ソースコード）を用いて、コンパイルとバッチジョブの実行を試す

- 手順

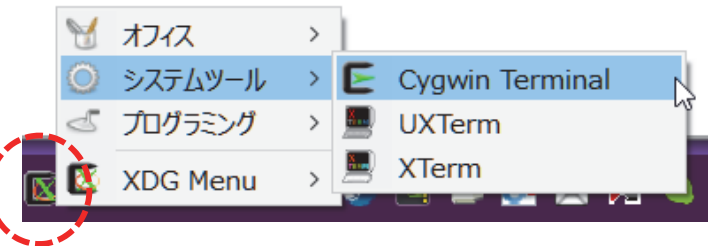
```
$ module load cuda/7.5.18
$ cd /work/GPUSEMINAR/利用者名
$ mkdir check
$ cd check
$ cp /work/GPUSEMINAR/ohshima/public/check/*.sh ./
$ nvcc -o check.out check.cu
$ qsub check.sh
$ cat check.sh.o*
initial: A = 0.000000
result: A = 1.000000
```

5. プロファイラ (nvvp) を使う

NVIDIA CUDA Visual Profiler

- 準備：Windowsユーザのみ

- Cygwinインストール時にxorg-serverとxinitをインストールしておく
- 「XWin Server」を起動する
 - タスクトレイにアイコンが2つ増える
- 緑の線の入った方のアイコンから「Cygwin Terminal」を起動する



- 使用

- HA-PACSへssh接続する際に-Yオプションを付ける
 - `ssh -Y username@hapacs.ccs.tsukuba.ac.jp`
- 起動時に-Xオプションを付けてインタラクティブジョブを起動する
 - `qsub_gpu -I -X -A GPUSEMINAR -q gpuseminar`
 - 終了はexit (Ctrlを押しながらdでも良い)
- module設定を行い、nvvpを起動する
 - `module load cuda/7.5.18`を設定したあとでnvvpコマンドを実行