はじめての富岳

接続からプログラム実行まで

夏の電脳甲子園「SuperCon 2021」本選大会

やりたいこと

- 富岳にログインする
- 富岳でプログラムをコンパイル・実行する

富岳にログインする

ログインまでの流れ

- **やりたいこと　SSHで富岳のログインノードにログインする**
  SSH：通信を暗号化してリモートの計算機にログインする仕組み
  秘密鍵ファイルと、公開鍵ファイルの二つの鍵ファイルが必要
  （どちらの鍵も皆さんが用意する：あとで）
- **公開鍵は、あらかじめ富岳のログインノードに置く必要がある**
  どうやって置くか？→web ブラウザ（Chromeなど）で専用サイト（ポータルサイトという）にアクセスして置く
- **すみません、ポータルサイトにアクセスできません！**
  →アクセスするための証明書をwebブラウザに登録する必要があります

皆さんに配布したのは、その証明書と証明書を登録するためのパスフレーズです

アカウント（証明書やパスフレーズも）あなた専用です。
チーム内であっても、絶対に共有しないでください。

作業の順番

https://www.hpci-office.jp/fugaku/user-info/index.html にあるスタートアップマニュアルを参照

マニュアルはダウンロードしてかまいませんが、利用終了後には削除して下さい

1. 証明書を web ブラウザに登録　2.2.1節（Firefox）or 2.2.3節 (Chrome)
2. SSH の秘密鍵・公開鍵のペアを作る 2.4.1節
3. ポータルサイトにアクセスして、公開鍵を登録する　2.3節、2.4.2節
4. SSH　の端末ソフトを用いて、富岳のログインノードにログインする 2.4.3節

説明の都合上、以下の環境を想定します

※Mac や Linux でログインしている研究者もたくさんいますが、ここでは説明しません。スタートアップマニュアルには記述があるので、これらの環境からログインしたい方はそちらを参照して下さい。

OS: Windows10

ブラウザ： Firefox　または　Chrome

SSH端末ソフト：PuTTY、　ファイルコピーソフト：WinSCP

証明書を web ブラウザに登録

- 富岳のポータルサイトには、許可のある人（証明書を持っている人）しかアクセスできません。そのため、web ブラウザにあらかじめ証明書を登録します。
- 用意するもの：

  理研から来たもの：証明書、パスフレーズ　証明書を登録する際に使用

  ユーザーが用意するもの：（上記とは別の）パスワード

  「クライアント証明書利用時に入力するパスワード」、あとでブラウザが証明書にアクセスするために使用。
  お好きなものをどうぞ。

- 操作の詳細は、スタートアップマニュアルの2.2.1節（Firefox）or 2.2. 3 節 (Chrome)をご覧ください

SSH の秘密鍵・公開鍵のペアを作る

- 富岳にアクセスするための鍵（SSH鍵）を作ります。ログイン時のユーザー認証と、富岳との通信の暗号化に使います。 ペアになった秘密鍵（private key） と公開鍵（public key） を作ります。
- 用意するもの：PuTTY というフリーソフト
  SSH 秘密鍵用のパスフレーズ（お好きなものをどうぞ）
- PuTTY：本家から最新版をダウンロードして利用します
  https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
  から putty-64bit-0.76-installer.msi をダウンロードしてインストール。デフォルト（すべてYES）でOK。鍵作成プログラムPuTTYgen, 接続プログラムPuTTYなどがインストールされます。
- **重要：パスフレーズは必ず設定！秘密鍵は誰にも渡さない！**
- 作業の詳細は、スタートアップマニュアルの2.4.1節をご覧ください

ポータルサイトにアクセスして、公開鍵を登録する

- 用意するもの： 先ほど作った公開鍵、証明書、 クライアント証明書利用時に入力するパスワード
- ポータルサイトhttps://www.fugaku.r-ccs.riken.jp/にアクセス。
- 証明書にアクセスためのパスワードの入力が求められます、あなたが**6ページ**で設定したパスワードを入力してください。
  もしどの証明書を使うか訊かれたら、先ほど登録した証明書を選択します（発行元がR-CCSになっている証明書、おそらく一つしか登録されていないので迷わないはず）
- 公開鍵は、適当なエディタやメモ帳で開いて、コピペします。
- 重要: **秘密鍵を登録しない**
- 操作の詳細は、スタートアップマニュアルの2.3節（ポータルへのアクセス）、2.4.1節（公開鍵の登録）を参照してください。

富岳にログインする（ログインノードへのログイン）

- 用意するもの：
  SSH秘密鍵、秘密鍵用パスフレーズ、富岳のアカウント名
- PuTTYを用いて(PuTTYgenではない)
  login.fugaku.r-ccs.riken.jp
  にログイン。マニュアルにあるように、秘密鍵を用いて接続します。
- 詳細はスタートアップマニュアルの2.4.3節を参照してください。
  補足：X11 forwarding やAgent forwarding は無効でOK（有効にしてもOK）
- ここでログインするのは、ログインノードといってコンパイルなどの操作専用の計算機です。実際に計算する本体（計算ノードといいます）とは別です。 ログインノード名は、ランダムで login2 ～ login8 になります
- ログインノードではLinuxが動いていて、コマンドを入力して操作します。

富岳と手元のPC間のファイルのコピー

- 用意するもの：WinSCPというフリーソフト、先ほど作ったSSH秘密鍵、秘密鍵のパスフレーズ
- WinSCPは本家 https://winscp.net/eng/download.php

からダウンロードする（○○**Downloadという広告が出ることがあるので、必ず「DOWNLOAD WINSCP 5.19.2 (10.9MB)」と書かれたボタンからダウンロードすること！**）

- インストールはデフォルト（すべて YES/はい）でOK。
- デフォルトで PuTTY に保存してあるセッションを自動で取り込んでくれます（インストール時に訊かれるので取り込むべし）。
- 取り込んだセッションを指定して（もし空欄ならユーザー名に富岳のアカウント名を入力して）接続。秘密鍵のパスフレーズを聞かれたら入力。

富岳と手元のPC間のファイルのコピー（続き）

- WinSCPを使えば、Windowsでのファイル操作と同じような操作で、富岳との間でファイルをコピーできます。
- **重要** **富岳には（どの秘密鍵でも）秘密鍵は絶対に置かないで下さい**
  見つけたらアカウントを停止します
- WinSCPで富岳上のファイルも操作できます。
  Linuxの端末作業に不慣れな人は、WinSCP経由でファイル操作をするとはかどります。　でもできればこの機会に Linux 端末の操作にも慣れてください
- ファイル名は日本語を避ける方が無難（半角スペースも避けるのが吉）。
  大文字小文字を区別するので注意。
- 表示は「コマンダー」「エクスプローラー」の2種類あるのでお好みで。
  表示→環境設定→外観　で変更可。

プログラムをコンパイル・実行する

今回の計算手順

1．ログインノードから計算ノードへログインする
2．計算ノードでソースファイルを編集＆コンパイル
　　emacs と vim が使えます
　　解説しませんが、ソースを手元の PC で編集しても構いません
3．計算ノードで実行
4．（適宜）ソースファイル等を手元のPCにコピーする

- 補足：富岳のようなスーパーコンピュータは、通常は以下のように使います
  1．ソースコードは手元のPC等で用意（ログインノードで編集することもある）
  2．ログインノードでコンパイル
  3．計算ノードで実行（バッチジョブとして実行）
     今回は計算ノードにログインしてユーザーが直接実行しますが、普段はスケジューラー経由で実行する（ユーザーは計算ノードにログインしない）ことがほとんど。
  4．ログインノードから手元のPC等に結果をコピーして解析
     （ログインノードで結果を確認することもある）

こちらの使い方については19ページへ

課題提出もバッチジョブ形式です

計算ノードにログインする

- 以下を入力すれば、計算ノードにログインできます。

赤枠：各自で変更可

```
pjsub --interact  -L rscgrp=supercon2021,node=1,elapse=3:00:00 --mpi proc=48
```

意味：ほとんどは（ここでは）おまじない、、、と思っていいのですが、皆さんが使えるノード数は1ノードなのでnode=1 を指定しています（富岳全体では158,978ノードあります）。elapse=で指定した時間（上では3時間）経過すると、計算ノードへの接続が自動的に切れます。この時間は各自で変更してOKですが、利用可能時間に収まるようにしてください。

MPI並列化を用いるときは、 --mpi proc= にMPIプロセス数を指定してください（ここでは最大値の48）

- ファイルはログインノードと同じものが見えます。
- 計算ノードからのログアウト（＝ログインノードに戻る）にはexit と入力。
- ログインノード・計算ノード共通：export LANG=ja_JP.UTF-8 と入力すると、メッセージを日本語にできます。
- ファイル名には日本語や半角空白を含めない方が無難。大文字小文字を区別するので注意。

コンパイル

- mkdir test_hello と入力して、test_hello (＝適当な名前)のディレクトリを作ります。 mkdir はディレクトリを作るコマンド
- cd test_helloと入力して、test_hello に移動します。
  cd は別のディレクトリに移動するコマンド。ログインした直後にいるディレクトリはホームディレクトリと呼ばれます。
- ソースファイルを用意します（ここではこちらで用意した hello.c をコピーしています)。

  cp /vol0003/ra020003/data/test_hello/hello.c ./
- ls （エルエス）と入力してファイル一覧を表示し、hello.cが表示されることを確認します。
- fcc hello.c と入力します。（C++ なら、FCC hello.cpp）
- もう一度 ls と入力し、a.out というファイルが表示されれば成功。

コンパイル（続き）

ダウンロードしたマニュアル・手引書は、利用終了後に削除すること

- **コンパイルの仕方**

  C言語　fcc [オプション] ソースファイル名
  C++　　FCC [オプション]　ソースファイル名

  MPI プログラムの場合は　mpifcc, mpiFCC　を使う

- **性能評価は、問題で指定されたオプションでコンパイルして行います。**
  デバッグ時以外は、同じオプションの指定をお勧めします

- 参考：よく使うオプション

  -Ofast　　富岳向け最適化。利用推奨。
  -fopenmp OpenMP を利用する。
  -Nclang　今回は**常に指定する**。Clang　モードでコンパイルで、最適化の癖が異なる。
  -O0　　オーゼロ、最適化禁止。デバッグ用。
  -Rpass=.* –ffj-lst=t　最適化情報の表示。チューニング時に役立つ。

詳細は以下のリンク先にある「Ｃ言語使用手引書」「Ｃ＋＋言語使用手引書」に記載があります
分量が多いので注意　https://www.fugaku.r-ccs.riken.jp/docs/manuals/tcsds-1.2.31

プログラムを実行する

- （もしログインノードにいるなら）計算ノードにログインします。
- cd (ディレクトリ名)で、実行ファイル（a.out）のある場所に移動します。
- ./a.out と入力します。
- （hello world プログラムであれば）画面に
  hello, world!
  と出力されます。
- （おまけ）計算ノードもLinuxなので、cat /proc/cpuinfo と入力すると、あなたが計算ノードで使っている cpu (A64FX)の情報が表示されます。さっきの hello world は、この cpu で動いたんですよ！

プログラムを実行する（続き）

- MPIプログラムの実行は次のようにします。
  mpiexec –np プロセス数　./a.out

プロセス数は　pjsub … –mpi proc=xxx　で指定したxxx を超えないようにしてください（xxx と同一が望ましい）。

バッチジョブのスクリプト（20ページ）では、実際に使うプロセス数を #PJM –mpi “proc=… ” に指定してください。

- OpenMPで用いるスレッドの数を変更したいときは、
  export OMP_NUM_THREADS=24
  のように入力します(ここでは24スレッドに変更)。

- 現在のスレッド数の指定を知るには、
  echo $OMP_NUM_THREADS
  と入力します。何も表示されない（空行のみ）の場合はデフォルト値で、上記の MPI xxx プロセス時に利用可能な最大スレッド数になります。proc=1 だと 48, proc=48 だと1です。

バッチジョブの使い方

- コンパイルはログインノードで行います（計算ノードで行ってもよい）
  コンパイルコマンドがログインノードと計算ノードで異なるので注意

場所	MPIなし: C/C++	MPIあり: C/C++
ログインノード	fccpx/FCCpx	mpifccpx/mpiFCCpx
計算ノード	fcc/FCC	mpifcc/mpiFCC

- 次ページの myjob.sh　（ファイル名は適当でOK）　を用意して
  pjsub myjob.sh
  としてスケジューラに実行を依頼します
- 実行状況は、pjstat で確認できます
- 実行が終わると、myjob.sh.1234567.out　のようなファイルに結果が出力されます。中身は less myjob.sh.1234567.out で表示できます。

バッチジョブの使い方（続き）

myjob.sh の例です。MPIプロセス数、OMP のスレッド数は適宜変えてください

```
#!/bin/bash
#PJM --rsc-list "node=1"
#PJM --rsc-list "rscgrp=supercon2021"
#PJM --rsc-list "elapse=0:06:00"
#PJM -mpi "proc=4"
#PJM -j

# do not generate empty output files
export PLE_MPI_STD_EMPTYFILE=off

export OMP_NUM_THREADS=12
echo $OMP_NUM_THREADS

date

mpifcc -Ofast -fopenmp -Nclang hello.c
mpiexec --stdin inputfile --stdout outputfile.${PJM_JOBID} ./a.out
```

6分でジョブを打ち切る

MPI 4プロセス。 mpi を用いないときはこの行を削除またはコメントアウト

-j で標準エラー出力を標準出力に統合

（空ファイル生成を防ぐためのおまじない）

OpenMPのスレッド数は各プロセス12に設定

（echo で12を表示：省略可）

（開始時刻を出力：省略可）

コンパイル（課題提出時には必須、ソースファイル名は適宜変更）

プログラムの実行 （mpiexec -np 4 --stdin inputfile --stdout outputfile.${PJM_JOBID} ./a.out でも可）

inputfile の中身を標準入力に読み込み、標準出力は outpufile.${PJM_JOBID}に出力する。${PJM_JOBID} には job の番号が入る。 mpiを用いないときは、単に./a.out < inputfile > outputfile.${PJM_JOBID} とする。

/vol0003/ra020003/data/batch_scripts に課題提出用の雛形があります。

資料は以上です

- 皆さんが富岳にログインできるのは、期間中の9時から17時までです。
  17時になったら、自動で接続が切れます。
  （最終日26日は9時から13時まで、13時で接続が切れます）
- 期間中は、discordでチューターが皆さんの質問に答えます。
- それでは富岳を満喫しましょう。 Enjoy!