

ジョブの投入方法(jsub)

(最終更新日: 2024/12/3 ステップジョブのメールについて注意追加)

ジョブの投入は jsub コマンドを使って行います。必要なインプットファイルは scp, sftp を用いて ccfeep 上に配置するなどしてください。(scp, sftp に関しては[クイックスタートガイド](#)の情報をご確認ください。)

なお、Gaussian ジョブについては g16sub/g09sub という専用コマンドを用意しております。[Gaussian](#) を使う場合は、まず [g16sub/g09sub](#) の方をご確認ください。(一応 jsub でも投入は可能ですが面倒です)

- [基本的な実行方法](#)
- [ジョブスクリプトの作成](#)
 - RCCS 導入のアプリを使う場合 => [サンプルジョブの実行方法](#)
 - ヘッダー(リソース定義)のサンプルが欲しい => [ジョブ投入に関するTips](#)
 - [1 - インタープリター\(shebang\)の指定\(必須\)](#)
 - [2 - リソースの定義\(select; 必須\)](#)
 - [3 - 計算時間の指定\(walltime; 必須\)](#)
 - [4 - オプションパラメータの指定](#)
- [ジョブの依存関係\(-W depend=afterok:\(ジョブID\), -W depend=afterany:\(ジョブID\)\)](#)
- [一連のジョブ実行\(ステップジョブ; --step もしくは --stepany\)](#)
- [ジョブスクリプト中の変数を投入時に指定する\(-v 変数名=値\)](#)
- [ジョブの終了まで待つ\(-W block=true\)](#)
- [特殊な利用方法について](#)
 - [CPUコアはそれほど必要無いもののメモリが多く必要な場合](#)
 - [1つのジョブ内で複数の計算を実行したい場合](#)

基本的な実行方法

後述するジョブスクリプト(job.sh)を用意した上で ccfeep にて以下のように実行します。(実際に入力するのは \$ より後の jsub job.sh の部分だけです。)正常に実行できれば、ジョブの ID とサーバ名が表示されます。

```
$ jsub job.sh
6169323.ccpbs1
$
```

jsub コマンド自体の実行はすぐに終了しますが、投入されたジョブがすぐに実行されるとは限りません。混雑時にはしばらく待つ必要があります。投入したジョブは ccfeep からログアウトした後も残ります。

表示された数字のジョブのID(上記例では 6169323)はジョブを削除したりする場合に必要になります。ただし、jsub の実行時に表示されたものを常に覚えておく必要ではありません。通常は投入後に jobinfo コマンドで表示されるものを確認する程度で十分です。

キューの指定は不要ですが、jsub -q H job.sh のように H キューを指定しても問題ありません。

jsub コマンドのオプションについては jsub --help を実行することで確認できます。投入されたジョブの状況については [jobinfo コマンド](#) で確認することができます。投入済のジョブを取り消したい、強制的に終了させたい場合は [jdel コマンド](#) を使います。

ジョブスクリプトの作成

RCCS で導入したアプリケーションで、ジョブとして実行することを想定されるものについてはジョブスクリプトのサンプルを用意しています。その標準的な置き場所などについては、[サンプルジョブの実行方法のページを参照ください](#)。独自のアプリを実行する場合にこれらサンプルをジョブスクリプトのテンプレートとして利用することもできるかもしれません。

以下に一例として 32 MPI 並列、個々のプロセスでは OpenMP で 2 並列を行うジョブのジョブスクリプトを示します。ジョブは最大で 24 時間実行することになります。実行するプログラムは自身でビルドした my-program で、RCCS の用意した Open MPI 4.1.6 を使います。(以下のサンプルで ≪ で示した注釈部分を実際のファイルに書くとエラーになりますので、書かないようにお願いします。)

```
#!/bin/sh ≪ 1 インタープリター(shebang)の指定
#PBS -l select=1:ncpus=64:mpiprocs=32:ompthreads=2 ≪ 2 リソースの指定
#PBS -l walltime=24:00:00 ≪ 3 計算時間指定
≪ 4 オプションパラメータを指定可能

# ジョブ投入時の作業ディレクトリへ移動します。(推奨; 状況によっては省略可能)
# 計算ホストでのジョブ実行開始時点では通常ホームディレクトリが作業ディレクトリになります
# このコマンドを実行することでジョブ投入時のディレクトリに移動できます
cd ${PBS_O_WORKDIR}
```

```
# 必要であれば module 等の環境設定を行います
module -s purge
module -s load openmpi/4.1.6

# 以下、計算を実行するために必要な設定を追加します
export PATH="/home/users/${USER}/bin:$PATH"

INPUT=myinp.inp
OUTPUT=myout.out

# そしてプログラムを実行します。
mpirun -np 32 my-program -i ${INPUT} -o ${OUTPUT}
```

1 - インタープリター(shebang)の指定(必須)

2 - リソースの定義(select; 必須)

3 - 計算時間の指定(walltime; 必須)

4 - オプションパラメータの指定

ジョブの依存関係(-W depend=afterok, -W depend=afterany)

一連のジョブ実行(ステップジョブ; --step もしくは --stepany)

ジョブスクリプト中の変数を投入時に指定する(-v 変数名=値)

ジョブの終了まで待つ(-W block=true)

特殊な利用方法について

CPUコアはそれほど必要無いもののメモリが多く必要な場合

1つのジョブ内で複数の計算を実行したい場合