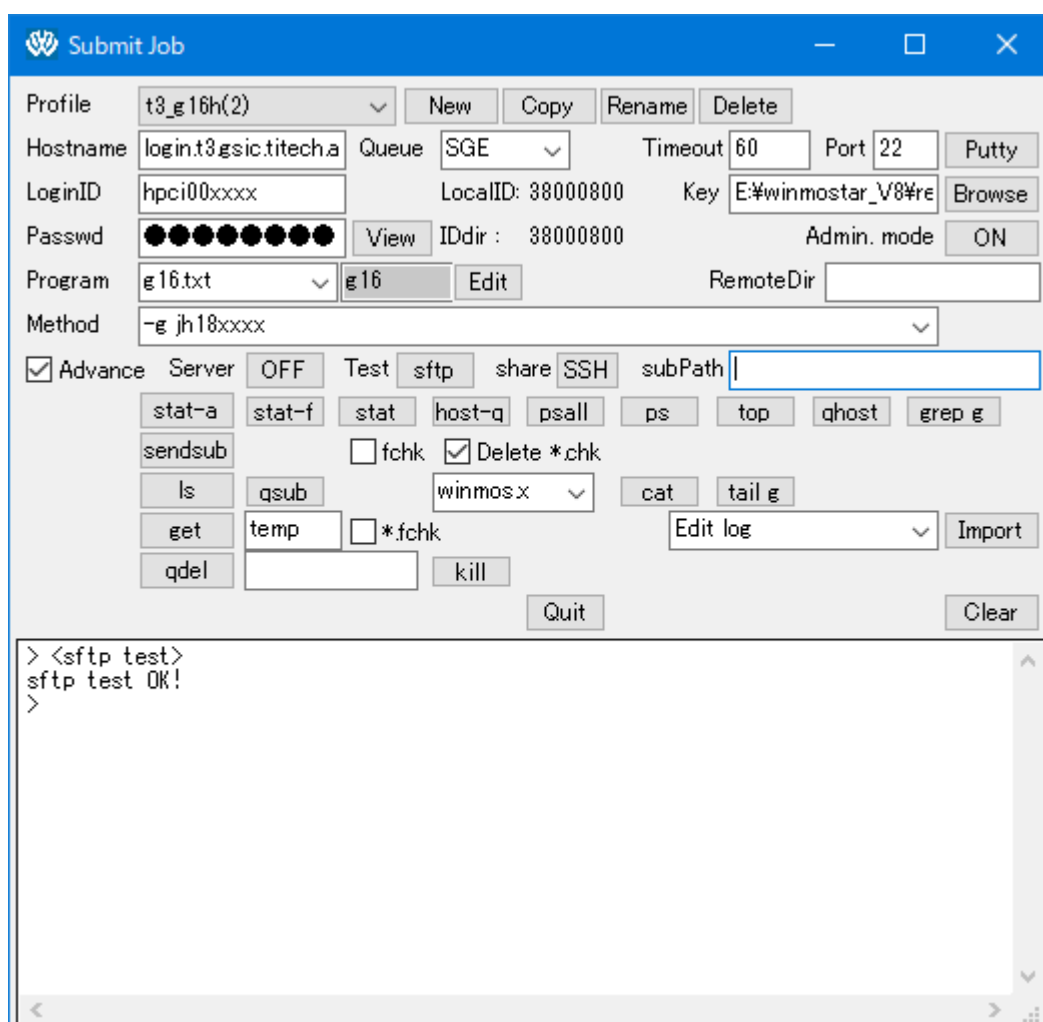


Winmostar Submit Job 機能マニュアル (TSUBAME3.0-Gaussian16 編)

1. リモートジョブ投入で Submit Job の画面が立ち上がる。
2. Submit Job の初期画面で、Server の [SET] を押すと Server の設定ができるようになる。
(Admin mode の [ON] を押し Password で “0204” を入力すると、Server の設定以外に管理者用の設定も可能になるが、通常は必要ない。)
3. Advance にチェックするとボタンが増えるが、Advance のチェックがなくても通常の操作はできる。TSUBAME3.0 では Queue で SGE を選択する。



Hostname(login-t3.g.gsic.titech.ac.jp)や LoginID、Passwd の設定を行う。Key は、フルパスでプライベートキーを設定する。Method でグループの設定 (-g) をする。

Program で g16.txt を選択し、[Edit]ボタンで開いたメモ帳の内容を以下のように書き換えて保存する。

```
#!/bin/bash
#$ -cwd
#$ -l f_node=1
#$ -l h_rt=0:30:00
#$ -p -5
./etc/profile.d/modules.sh
module load gaussian16
g16 < $0 > $1.log
```

#\$ -l f_node=1 の行は資源タイプ f_node のノードを 1 個確保する。この場合は [TSUBAME3.0 利用の手引き](#)に従って、入力データの%nproc(%nprocshred)は使用物理 CPU コア数の 28 とする。h_node の場合は%nproc=14、q_node では%nproc=7 にする。

4. Test の横の[sftp]ボタンでテストする。

最初の接続時は、以下のようなメッセージがでるので、y を入力する、

If you want to carry on connecting just once, without adding the key to the cache, enter "n".

If you do not trust this host, press Return to abandon the connection.

Store key in cache? (y/n)

このテストが OK になれば、5. ~ 10. を行う。この時は、Server を OFF にしても良い。通常は Server を OFF にしておく方が、誤って設定を変えてしまう恐れがない。

5. SSH アクセス数の制限を回避するには、[TSUBAME3.0 の SSH アクセス数制限について](#)を参考にして Putty の設定をしてから share[SSH]を押す。この方法は、アクセス数の制限がない場合でも通信の応答性が良くなる効果がある。

6. [stat-a]でマシンの状況を確認する。

7. [sendsub]で TSUBAME にデータが転送されて、ジョブが投入され、Your job xxxxxxxx has been submitted と表示される。

filename.gjf の計算は RemoteDir で設定したディレクトリで実行されるが、RemoteDir が空白の場合は、Linux 機の LoginID のホームディレクトリ/Windows のユーザーID/g16/filename/のディレクトリで実行される。Windows のユーザーID は、画面上の LocalID で表示されている。Windows のユーザーID に漢字が含まれている場合は、下の IDdir で表示される 16 進数を含む文字列で代用される。例えば、"12 千田 AB"は" 1290E79363AB"になる。

Submit Job 画面や Winmostar の画面を終了してもよいが、再度立ち上げた場合は、filename.gjf を呼び出して[ls]することで、計算結果の出力ファイル(filename.log)を確認することができる。

同一ディレクトリでのジョブの二重投入は異常終了する。Filename または Windows のユーザーID 等が異なれば、同一ディレクトリにはならないので問題はない。

8. [ls]で、その他の出力ファイルの存在も確認できる。winmos.x はジョブの起動を示し、winmos.o はジョブの標準出力、winmos.e はエラー出力を示している。

[stat-a]等で実行状況を確認する。

9. 出力ファイルが確認できたら、[get]する。途中結果を[get]することもできる。

[import]の左のボックスに[get]後の動作（ファイルを開く等）を指定することができる。PC 上に

一度[get]したファイルは、再度[get]しなくても[import]で別の動作をすることができる。

計算→Import→Animation 等で、様々な図示ができる。MO 表示は、計算→Import→MO,UV,Charge...で行なう。

1 0. ジョブのキャンセルは、[qdel]の横の窓に[stat-a]で確認したジョブ ID を入力して[qdel]を押す。

1 1. Gaussian16 から追加された%cpu を使う場合は、f_node では%ncpoc=28 の代わりに%cpu=0-27 と指定すれば良い ([Gaussian 利用の手引](#))。しかし、f_node や q_node 等のように論理分割されたノードを用いる場合は、ジョブ実行時に動的に割り当てられる CPU コアを指定する必要があるため事前にデータに書くことができない。そこで、g16.txt を以下のようにすると、%nproc で始まる行は、動的に割り当てられた%cpu に置換されたデータで計算することができる。

```
#!/bin/bash
#$ -cwd
#$ -l h_node=1
#$ -l h_rt=0:30:00
#$ -p -5
rm $1_*. *
INPUT_ORG=$0
INPUT=$1_temp.gif
CPU="%cpu=`numactl -s | awk '/physepubind:/ { print $(2) "-" $(15) }`"
cat ${INPUT_ORG} | awk -v c="${CPU}" '{if (tolower(substr$(0),0,6))=="%nproc") {print c} else {print $(0)}}' > ${INPUT}
./etc/profile.d/modules.sh
module load gaussian16
g16 < ${INPUT} > $1.log
```

GPU を利用するには、%gpucpu も追加されるように以下のような g16.txt にする ([GPU の利用](#))。

```
#!/bin/bash
#$ -cwd
#$ -l h_node=1
#$ -l h_rt=0:30:00
#$ -p -5
rm $1_*. *
INPUT_ORG=$0
INPUT=$1_temp.gif
CPU="%cpu=`numactl -s | awk '/physepubind:/ { print $(2) "-" $(15) }`"
GPU="%gpucpu=0-1=`numactl -s | awk '/physepubind:/ { print $(2) "," $(3) }`"
cat ${INPUT_ORG} | awk -v c="${CPU}" -v g="${GPU}" '{if (tolower(substr$(0),0,6))=="%nproc")
```

```
{print c "¥n" g} else {print ${0}}' > ${INPUT}
./etc/profile.d/modules.sh
module load gaussian16
g16 < ${INPUT} > $1.log
```