

## 1. はじめに

FSL やそれを利用した HCP パイプラインでは、計算を高速化するために CPU あるいは GPU を介した並列分散処理を行う。CPU の並列分散化には、Sun Microsystems が開発していたオープンソースの Sun Grid Engine (SGE) が使われる。しかし同社は 2010 年に Oracle が買収し、ソースコードがクローズドになってしまった（現在は Univa が引き継いでいるが、やはりクローズドである）。

これを受けて、リバプール大のグループが Sun Grid Engine のオープンソースの最終版である 6.2u5 をベースに、オープンソースでの開発を継続し、Son of Grid Engine として公開している。

<https://arc.liv.ac.uk/trac/SGE/wiki>

以下、本マニュアルでは Son Grid Engine も SGE と呼ぶ。上記サイトでは 2016 年に最新版の 8.1.9 が公開されている。対応 OS は Linux のみで、Debian/Ubuntu 系、RedHat/CentOS 系がある。

本マニュアルでは、Ubuntu18.04LTS への SGE 8.1.9 のインストール方法について紹介する。なお宮田は Linux 初心者であるため、下記の記載にも不正確な部分が含まれるかも知れない。その場合はご容赦ください。また本マニュアルの作成に関しては、前京都大学脳機能総合研究センターの吉田英史先生にご指導いただいた経験が活かされています。この場を借りて御礼申し上げます。

なお宮田は（もちろん吉田先生も）本マニュアルの内容に関して一切の責任を負いません。あくまで自己責任でこのマニュアルをご使用下さい。

## 2. 環境・条件

以下では、Ubuntu18.04 に、スタンドアロンとして SGE をインストールした際の手順を記す。主に以下のウェブサイトを参照した。

<https://arc.liv.ac.uk/trac/SGE/wiki>

<https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FslSge>

<https://root-forum.cern.ch/t/couldnt-find-font-adobe-helvetica-medium-r-10-iso8859-1/21619>

基本事項として、SGE はもともと、ネットワーク内にあるたくさんのマシンをつないでクラスター化し、並列分散処理により計算力を増幅する目的で設計されている。そのためには各マシンがネットワーク上で互いに連絡し合う必要があり、この時に使われるのがホスト名（マシンのネットワーク上での名前）である。またこの理由により、マシンはネットワークにつながっている必要がある。ホスト名の確認は下記のコマンドで行う：

```
uname -n
```

一方、ネットワーク上の各マシンにはそれぞれ IP アドレスが割り当てられている。こちらはコンピュータの世界における住所である。SGE が正しくクラスター内のマシンと連絡出来るためには、誰（ホスト名）がどこ（IP アドレス）に住んでいるのかが分からないといけない。ネットワークが DNS サーバー下であり、DNS サーバーがこれを行っている場合、特に何も設定する必要はない。

そうでない場合、`/etc/hosts` にホスト名と IP アドレスの対応を記述する必要がある。宮田の環境は DNS サーバー下であるが、この設定が必要であった。以下のコマンドで IP address を確認。またはアプリ一覧>設定>ネットワーク>歯車アイコンで確認できる。

```
hostname -i
```

`/etc/hosts` の編集には、テキストエディタ nano を用いる。ターミナルから  
`sudo nano /etc/hosts`

と打ち込むと、`/etc/hosts` ファイルが root 権限で開かれる（つまり編集できる）。このとき、宮田の環境では

```
127.0.0.1    localhost
127.0.1.1    <ホスト名>
```

となっていた。この2段目が SGE のインストールの邪魔をするので、以下のように修正。  
<IP address> <ホスト名>

IP address ととホスト名の間にはスペースを入れることを忘れずに。「Control + X」で保存、ファイル名はそのままなので「Y」、そして Enter で終了。マシンを再起動して変更を反映させる。

### 3. アカウント作成

上記の様に、もともと SGE はネットワークを介して別のマシンにジョブを投入する。これは本来 root 権限で行う必要があるが、それはセキュリティリスクでもある。そのため SGE では、ジョブ投入専用 root 権限を持たないアカウントを作成し、それを各マシンに（安全かつジョブ投入権限のあるアカウントとして）登録するという構造を採用している。これはスタンドアロンで設定するときも同様である。

下記のコマンドで、アカウント sgeadmin を作成する：

```
sudo useradd --home /opt/sge --system sgeadmin
```

FSLSGe のウェブサイトでは、Debian/Ubuntu ユーザーは useradd ではなく adduser コマンドを使うと良い、と書かれているが、必ず useradd を使うこと。adduser で上記のオプションで作成すると、あとで /opt/sge の所有者・グループの設定でエラーが出て、解決が大変である。

これはログイン画面にも現れない簡易的なアカウントである。後述の qmaster、execd どちらもこの sgeadmin から走らせる必要がある。

### 4. ポートの設定

上記の様にネットワークを介して情報をやりとりするために、Linux の各ディストリビューションでは SGE 専用のポートがデフォルトで設定されている。

```
grep sge /etc/services
```

として、下記の出力があるかどうかを確認しておく。

```
sge_qmaster 6444/tcp      # Grid Engine Qmaster Service
sge_qmaster 6444/udp      # Grid Engine Qmaster Service
sge_execd   6445/tcp      # Grid Engine Execution Service
sge_execd   6445/udp      # Grid Engine Execution Service
```

### 5. SGE のダウンロード・インストール

Linux ではリポジトリと呼ばれる「ソフトウェア置き場」からソフト（パッケージと呼ばれる）をダウンロード・インストールするのが普通である。リポジトリというのはちょうど Mac の App Store や Android の Google Play の様なものにあたる。リポジトリからのダウンロードやインストールには、Ubuntu では apt というコマンドを使う。apt を使うと、「ソフト A をインストールするにはソフト B とソフト C が事前にインストールされている必要がある」というような「依存関係」を自動で解決してくれる（つまり B も C もダウンロード・インストールしてくれる）。入手したソフトのアップデートも apt コマンドでまとめて行える。

Ubuntu では、gridengine という SGE のパッケージが用意されており、公式のリポジトリからダウンロード・インストール可能になっている。しかし Ubuntu18.04 版 (8.1.9+dfsg-7build1) はバグがあり、正常にインストール出来ない。従って、上記の SGE サイトからダウンロードする。

<https://arc.liv.ac.uk/downloads/SGE/releases/8.1.9/>

ダウンロードするのは下記の3つである：

sgc-common\_8.1.9\_all.deb

sgc\_doc\_8.1.9\_all.deb

sgc\_8.1.9\_amd64.deb（このファイルだけ他と離れて一番下にあるので、忘れずに）

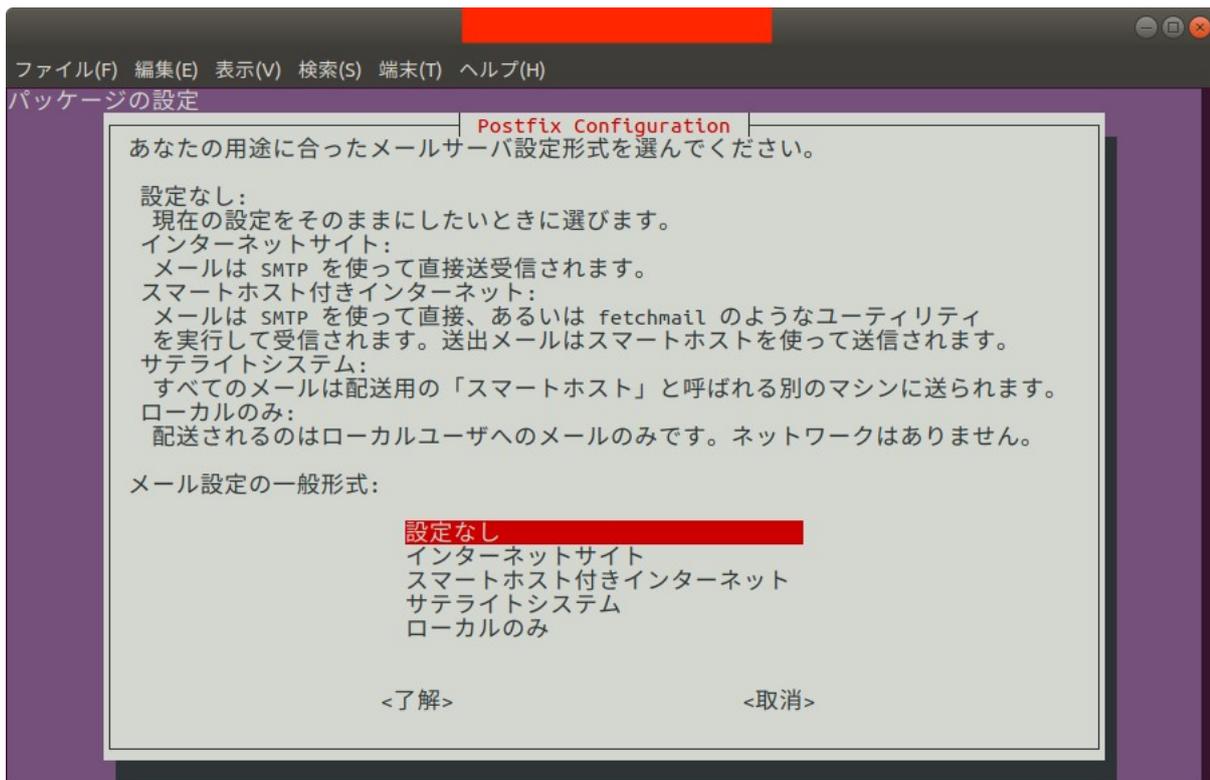
なお、同じ.deb ファイルでも下記はダウンロード不要

sgc\_8.1.9\_amd64.deb

ダウンロードしたフォルダに移動し、apt でこれらをインストールする。

```
sudo apt install ./sgc-common_8.1.9_all.deb ./sgc-doc_8.1.9_all.deb ./sgc_8.1.9_amd64.deb
```

すると下記のダイアログが出る：



設定なしを選んで Enter。途中「続行しますか」と聞かれたら Y と答えてインストールが終了する。

## 6. Qmaster および execd の設定

Qmaster (queue master) と execd (execution daemon) を設定する。どちらも Unix 系 OS において、メモリ上に常駐するプログラム (デーモン daemon) である。execd が実際のジョブを実行する下位プログラム (子プロセス) であり、qmaster は execd が実行するジョブの順番 (キュー queue) を管理する上位プログラム (親プロセス) である。

### 6. 1 Qmaster の設定

ターミナルから下記のコマンドを入力する：

```
export SGE_ROOT=/opt/sgc
```

```
cd $SGE_ROOT
```

```
sudo ./install_qmaster
```

これで対話モードに入る。sudo を付けず、./install\_qmaster だけだとマシン起動時に SGE を起動するスクリプトが作成されないようなので注意。

- 1) 最初に出てくる「Grid Engine admin user account」では、sgeadmin を指定する。このときダイアログの表示にふたとおりあるようである。
  - a) 「sgeadmin でインストールしますか」と聞かれる場合はデフォルトの Y で Enter。
  - b) 「root 以外のアカウントでインストールしますか」と聞かれる場合は Y と答えて、Please enter valid user name で” sgeadmin”と入力する。

あとは全てデフォルトのまま Enter で OK だが、以下を確認しておくこと。

- 2) Windows Execution Host のインストールは「n」 (no) を選択。
- 3) Grid Engine JMX MBean server は「n」を選択。
- 4) Setup spooling では classic を選択。BerkeleyDB は大規模なクラスター用である。
- 5) 「設定したパラメータを変更しますか」には「n」を選択。
- 6) ホストリストのファイルには「n」を選択し、次の対話では何も入力せずに Enter。
- 7) Shadow host(s) now は「n」を選択。

ガイドに従って enter を押し続けると対話モードが終了し、通常のターミナル画面に戻る。

## 6. 2 SGE の環境変数の設定

/root/.bashrc と ~/.profile に SGE の環境変数などの設定を記入する。下記で/root/.bashrc を開く：

```
sudo nano /root/.bashrc
```

最下行に下記を記入する。

```
# SGE settings
export SGE_ROOT=/opt/sge
export SGE_CELL=default
if [ -e $SGE_ROOT/$SGE_CELL ]
then
. $SGE_ROOT/$SGE_CELL/common/settings.sh
fi
```

~/.profile にも同様に記入する。なお、SGE\_ROOT は/opt/sge だが <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FslSge> には/usr/sge と間違って記述されているので注意！

アカウントをログアウト→ログインして変更を反映させる。

## 6. 3 ユーザーアカウントの追加

上述したようにネットワークを介してジョブを割り当てる際、root 権限が必要であるが、SGE ではこれを避けるためにユーザーアカウントを信頼できるアカウントとして登録する。これはルートユーザーになって行う必要がある。下記コマンドを入力。

```
sudo su
```

でルートユーザーになると、コマンドプロンプトが#に変わる。この状態で以下のコマンドを入力：

```
qconf -am <username>
```

下記のような表示が出れば OK。

```
root@XXXXXXXX added <username> to manager list
```

<username>は、ターミナルから `users` または `who` コマンドで表示されるものである。ログイン画面に現れるユーザ名とは異なるので注意。

終わったら、`exit` と打ち込んで通常のユーザーに戻る。

## 6. 4 Execution host の設定

複数台のマシンからなるクラスターの場合、`execution host` として、このコンピュータを登録する必要がある。

```
qconf -ah <ホスト名>
```

この時のホスト名は「2. 環境／条件」の `uname -n` で調べたものである。しかし今回、**スタンドアローンの1台のマシンで設定する場合には、この作業は不要**であった（すでに登録されています、と表示される）。

次に Qmaster と同様に、`execd` の設定を対話モードで行う。

```
cd $SGE_ROOT
```

```
sudo ./install_execd
```

すべてデフォルトのまま Enter し、対話モードは終了。qmaster 同様、インストールは `sudo` で行うこと。

なお上述の FSLsge のサイトでは、「デフォルトキューを作成しますか」には「n」を選択するように、と書かれている。そうすると「7. キューの設定」のところで `all.q` が見つからない。FSL で SGE を使うだけならこれでもいいのかもしれないが、念の為「y」でデフォルトキュー `all.q` を作成しておいた方が無難のように思われる。こうすることで FSL 以外でも SGE を使える（と思われる）。

## 6. 5 Submission host の設定

この PC が job submission 出来るように設定する。これも root user で行う。

```
qconf -as <ホスト名>
```

## 6. 6 sgeadmin での起動の確認

`sgeadmin` アカウントの作成の項目で述べたように、qmaster、execd とともに `sgeadmin` アカウントから起動する必要があるが、上記の手順のどこかを間違っていると、`username` アカウントから起動されていることがある。下記で確認する：

```
ps aux|grep sge
```

以下の様な表示が出る

```
sgeadmin 3721 0.1 0.0 340384 65192 ?    Sl 11:14 0:49 /opt/sge/bin/lx-amd64/sge_qmaster
<username> 12668 0.0 0.0 223272 14396 ?    Sl 13:03 0:23 /opt/sge/bin/lx-amd64/sge_execd
<username> 70772 0.0 0.0 112732 976 pts/1  S+ 19:48 0:00 grep --color=auto sge
```

この場合、`sge_execd` を `sgeadmin` から起動し直す必要がある。qmaster、execd の停止は

```
sudo $SGE_ROOT/default/common/sgeexecd stop
```

```
sudo $SGE_ROOT/default/common/sqemaster stop
```

で行う。重要なこととして、停止は必ず `execd`→`qmaster` の順に行う。前者が子プロセス、後者が親プロセスであるためである。これで停止しない場合には、

```
sudo kill <process ID>
```

で停止する。上記だと 3721、12668、70772 がそれぞれの process ID である。起動では逆に qmaster、execd の順に起動する。コマンドは以下：

```
sudo $SGE_ROOT/default/common/sgemaster start
sudo $SGE_ROOT/default/common/sgeexecd start
```

ps aux|grep sge で、以下の様にどちらも sgeadmin が起動していることを確認。

```
sgeadmin 71420 0.2 0.0 311712 36664 ?    SI  20:11  0:00 /opt/sge/bin/lx-amd64/sge_qmaster
sgeadmin 71559 0.3 0.0 215084 10152 ?    SI  20:12  0:00 /opt/sge/bin/lx-amd64/sge_execd
<username> 71577 0.0 0.0 112728  976 pts/1   S+  20:12  0:00 grep --color=auto sge
```

## 7. キューの設定

下記の手順で、キューの設定を行う。通常、クラスターマシンには複数のユーザーが次から次へとジョブを投入する。それらのジョブに順番（キュー）を設定し、実行の優先順位を決定するという仕組みである。デフォルトでは SGE は all.q という一種類のキューしか持たないが、FSL では以下の 4 種類のキューを設定する必要がある。

```
verylong.q - priority => 20
long.q - priority => 15
short.q - priority => 10
veryshort.q - priority => 5
```

Priority はキューの優先順位で、20 が最低で、0 が最高である。順位の高いキューから優先的に実行される。これらの設定は以下をターミナルにコピペすることで、一度に行える。ただし Microsoft Word や LibreOffice Writer からだと改行コードがおかしいので、一旦テキストファイルにコピペした上でターミナルにコピペすると良い：

```
# change defaults for all.q
qconf -sq all.q |\
    sed -e 's/bin\/csh\/bin\/sh/' |\
    sed -e 's/posix_compliant\/unix_behavior/' |\
    sed -e 's/priority          0/priority 20/' >\
    /tmp/q.tmp
qconf -Mq /tmp/q.tmp

# add other queues
sed -e 's/all.q/verylong.q/' /tmp/q.tmp >\
    /tmp/verylong.q
qconf -Aq /tmp/verylong.q

sed -e 's/all.q/long.q/' /tmp/q.tmp |\
    sed -e 's/priority *20/priority 15/' >\
    /tmp/long.q
qconf -Aq /tmp/long.q

sed -e 's/all.q/short.q/' /tmp/q.tmp |\
    sed -e 's/priority *20/priority 10/' >\
    /tmp/short.q
qconf -Aq /tmp/short.q

sed -e 's/all.q/veryshort.q/' /tmp/q.tmp |\
    sed -e 's/priority *20/priority 5/' >\
    /tmp/veryshort.q
qconf -Aq /tmp/veryshort.q
```

最下行だけ実行されずに残るので、Enter して設定完了。なお各行を説明すると

```
# change defaults for all.q
qconf -sq all.q \<
    sed -e 's/bin\/csh/bin\/sh/' \<

    sed -e 's/posix_compliant/unix_behavior/' \<

    sed -e 's/priority 0/priority 20/' >\  
    /tmp/q.tmp
qconf -Mq /tmp/q.tmp

# add other queues
sed -e 's/all.q/verylong.q/' /tmp/q.tmp >\  
    /tmp/verylong.q
qconf -Aq /tmp/verylong.q

sed -e 's/all.q/long.q/' /tmp/q.tmp \  
    sed -e 's/priority *20/priority 15/' >\  
    /tmp/long.q
qconf -Aq /tmp/long.q

sed -e 's/all.q/short.q/' /tmp/q.tmp \  
    sed -e 's/priority *20/priority 10/' >\  
    /tmp/short.q
qconf -Aq /tmp/short.q

sed -e 's/all.q/veryshort.q/' /tmp/q.tmp \  
    sed -e 's/priority *20/priority 5/' >\  
    /tmp/veryshort.q
qconf -Aq /tmp/veryshort.q
```

説明用のコメント行。  
all.q の設定情報の呼び出し  
シェルを/bin/csh から/bin/sh に変更。現在、  
SGE ではデフォルトで/bin/sh になっている  
ので、なくても OK。  
シェルの開始モードを posix\_compliant から  
unix\_behavior に変更  
priority を 0 (最高) から 20 (最低) に変更  
/tmp/q.tmp として上記を保存  
/tmp/q.tmp を開く (以下の作業のため)

コメント行。  
all.q を verylong.q に変更  
/tmp/verylong.q と名前をつけて保存  
verylong.q をキューとして追加

all.q を long.q に変更  
priority を 15 に変更  
/tmp/long.q として保存  
long.q をキューとして追加

all.q を short.q に変更  
priority を 10 に設定  
/tmp/short.q として保存  
short.q をキューとして追加

all.q を veryshort.q に変更  
priority を 5 に設定  
/tmp/veryshort.q として保存  
veryshort.q を追加

スラッシュとバックスラッシュが多くて見づらい。いつか書き直したい。  
上記の設定が反映されたか確認のため、ターミナルから  
qstat -f

として、以下のような表示が出れば OK。

queuename	qtype	resv/used/tot.	load_avg	arch	states
all.q@<host name>	BIP	0/0/48	0.32	lx-amd64	
long.q@<host name>	BIP	0/0/48	0.32	lx-amd64	
short.q@<host name>	BIP	0/0/48	0.32	lx-amd64	
verylong.q@<host name>	BIP	0/0/48	0.32	lx-amd64	
veryshort.q@<host name>	BIP	0/0/48	0.32	lx-amd64	

## 8. 動作確認

GUIでジョブ・キューを確認・操作できる Qmon を起動する。ターミナルから qmon

と打ち込み、Enter。正常であれば、下のようなウィンドウが立ち上がるはずである。



一方、Ubuntu18.04 では下記のようなエラーが出る場合が多い。

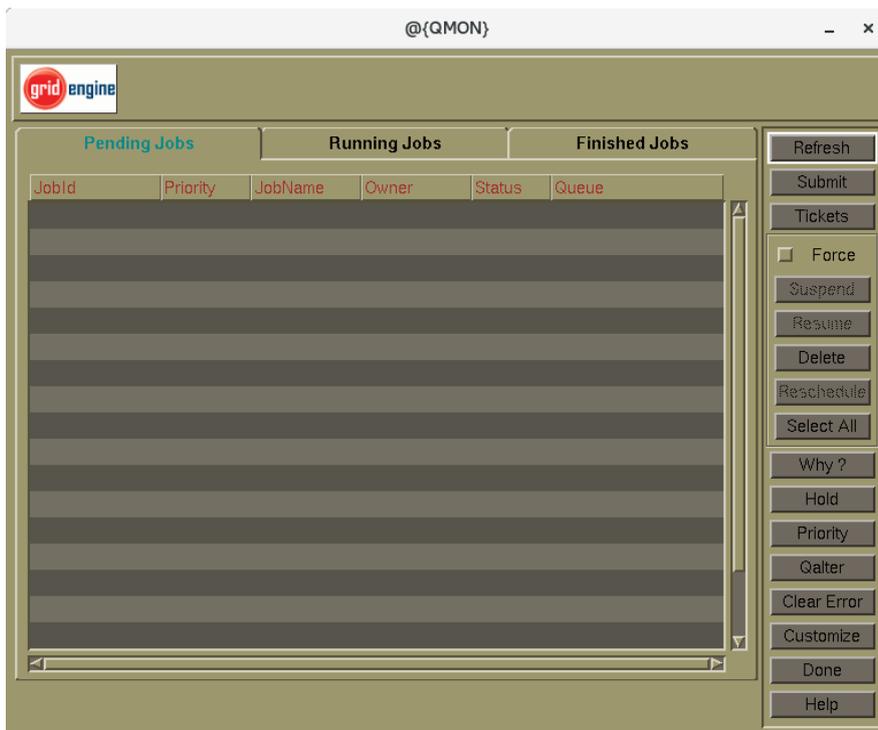
```
-----
long.q@ubuntu          BIP  0/0/4      0.30    lx-amd64
-----
short.q@ubuntu         BIP  0/0/4      0.30    lx-amd64
-----
verylong.q@ubuntu     BIP  0/0/4      0.30    lx-amd64
-----
veryshort.q@ubuntu    BIP  0/0/4      0.30    lx-amd64
-----
$ qmon
Warning: Cannot convert string "-adobe-helvetica-medium-r--14-*-*-*p-*-*-*" to type FontStruct
Warning: Cannot convert string "-adobe-helvetica-bold-r--14-*-*-*p-*-*-*" to type FontStruct
Warning: Cannot convert string "-adobe-helvetica-medium-r--20-*-*-*p-*-*-*" to type FontStruct
Warning: Cannot convert string "-adobe-helvetica-medium-r--12-*-*-*p-*-*-*" to type FontStruct
Warning: Cannot convert string "-adobe-helvetica-medium-r--24-*-*-*p-*-*-*" to type FontStruct
Warning: Cannot convert string "-adobe-courier-medium-r--14-*-*-*m-*-*-*" to type FontStruct
Warning: Cannot convert string "-adobe-courier-bold-r--14-*-*-*m-*-*-*" to type FontStruct
Warning: Cannot convert string "-adobe-courier-medium-r--12-*-*-*m-*-*-*" to type FontStruct
Warning: Cannot convert string "-adobe-helvetica-medium-r--10-*-*-*p-*-*-*" to type FontStruct
X Error of failed request: BadName (named color or font does not exist)
Major opcode of failed request: 45 (X_OpenFont)
Serial number of failed request: 862
Current serial number in output stream: 873
$ cd
```

この場合、qmon を表示させるためにフォントを追加する必要がある。ターミナルから下記を行う：

```
sudo apt install xfonts-75dpi xfonts-100dpi
```

インストール、マシンを再起動。再びターミナルから qmon

と打ち込んで、上記の Qmon のメインコントロールウィンドウが出ることを確認。一番左上のボタンでジョブコントロールパネルを呼び出す。



次に、テストスクリプトを作成する。テキストファイルに下記を記入：

```
#!/bin/bash
qsub <<CMD
#!/bin/bash
#$ -V
#$ -cwd
#$ -j y
#$ -N my_log
/bin/hostname
CMD
```

拡張子を sh にして、適当に名前をつけて保存する。ここでは qsubtest.sh とする。これをホームディレクトリに置く。

```
cd
```

でホームディレクトリに移動。

```
sh qsubtest.sh
```

で上記のテストスクリプトを実行。Qmon のジョブコントロールパネルで my\_log\*\*\*ジョブを確認。「Pending」→「実行中」→「完了」に移動していれば OK である。ホームディレクトリに my\_log\*\*\* という名前のファイルが出来上がっていることが確認できれば、動作確認終了である。