Son of Grid Engine の CentOS7 へのインストール方法

20190830 京都大学精神医学教室　宮田　淳

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 系があるが、後者は CentOS6 までの対応となっており、RHEL7 および CentOS7 にはそのままではインストールできない。

本マニュアルでは、RHEL7 および CentOS7 への sge8.1.9 のインストール方法について紹介する。なお宮田は RHEL も CentOS も初心者であるため、下記の記載にも不正確な部分が含まれるかも知れないことを、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、Mac への SGE のインストールを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、RHEL7.7 に、スタンドアローンとして 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://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FslSge

http://stab.st-andrews.ac.uk/wiki/index.php/Son_of_Gridengine

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 サーバー下にある場合には、DNS サーバーがこれを行うので、特に何も設定する必要はない。そうでない場合は/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 サーバー下なのでこの設定は不要であった。もし/etc/hosts を編集する場合は、テキストエディタ nano を用いて

```
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）。最下行に

<IP address> <ホスト名>

を書き込む。IP address とホスト名の間にはスペースを入れる様に。「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。たぶんこれで OK のはず。

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、ジョブ投入専用に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ー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
```

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_qmaster /etc/services
grep sge_execd /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、RHEL／CentOSではyumというコマンドを使う。yum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yumコマンドでまとめて行える。

　あるリポジトリからダウンロード・インストールするためには、まずそのリポジトリを安全なものとして、マシンに登録する必要がある。まず代表的なリポジトリであるEPELを登録する：

```
sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

そのうえで、sgeをCentOS7で使えるようにするために、下記のダウンロード・インストールを行う：

```
sudo yum install jemalloc-3.6.0 lesstif-0.95.2 munge-libs-0.5.11 libdb4-utils-4.8.30
```

jemallocというのはマルチコア・マルチスレッドCPUをより効率よく利用するためのプログラムらしい。詳細は割愛。

ついで下記のサイトから、CentOS7版のSGEをダウンロードする。

https://copr-be.cloud.fedoraproject.org/results/loveshack/SGE/epel-7-x86_64/00756477-gridengine/

ダウンロードするのは下記の5つである：

```
gridengine-8.1.9-2.el7.x86_64.rpm
```

gridengine-qmaster-8.1.9-2.el7.x86_64.rpm
gridengine-qmon-8.1.9-2.el7.x86_64.rpm
gridengine-execd-8.1.9-2.el7.x86_64.rpm
gridengine-guiinst-8.1.9-2.el7.noarch.rpm

「２．環境・条件」で挙げた 2 つのサイトでは、CentOS6 版の 8.1.9-1.el6 をダウンロードするよう指示している。しかしこれをインストールしても、あとで sudo yum update すると上記の CentOS7 版に置き換えられるので、上記をダウンロードした方が手間が省ける。この５つのうち、guiinst は GUI ベースのインストーラーであるが、このマニュアルでは使わない。試しに走らせてみたところ「ファイルがそこにありません」というようなエラーが出た。正常に走らせるためには若干の編集が必要のようである。またこのマニュアルの手動インストールにとって必要かどうかは不明である。
　次に yum でこれらをインストールするのであるが、ダウンロードしたディレクトリからこのまま
sudo yum install -y gridengine-8.1.9-2.el7.x86_64.rpm gridengine-devel-8.1.9-2.el7.noarch.rpm gridengine-execd-8.1.9-2.el7.x86_64.rpm gridengine-qmaster-8.1.9-2.el7.x86_64.rpm gridengine-qmon-8.1.9-2.el7.x86_64.rpm

とやると、下記のようなエラー表示が出る：
エラー: パッケージ: gridengine-8.1.9-2.el7.x86_64 (/gridengine-8.1.9-2.el7.x86_64)
　　要求: perl(XML::Simple)
****
yum can be configured to try to resolve such errors by temporarily enabling
disabled repos and searching for missing dependencies.
To enable this functionality please set 'notify_only=0' in /etc/yum/pluginconf.d/search-disabled-repos.conf
****
エラー: パッケージ: gridengine-8.1.9-2.el7.x86_64 (/gridengine-8.1.9-2.el7.x86_64)
　　要求: perl(XML::Simple)

つまり perl の XML::Simple というモジュールが必要ということなのだが、これが置かれていた RPMForge というリポジトリは現在、メンテナンスされていない。つまりどんな悪意のプログラムが置かれているかも分からない状態となっている。CentOS の Wiki でも利用しないよう勧めている。とはいえこれを使うしかない。方法はいくつかあるが、最も簡単と思われるのは、上記のエラーメッセージにあるとおり、/etc/yum/pluginconf.d/search-disabled-repos.conf ファイルを一時的に書き換えて、XML::Simple をダウンロード、インストールできるようにすることである。ターミナルで
sudo nano /etc/yum/pluginconf.d/search-disabled-repos.conf

でファイルを開き、notify_only=1 を notify_only=0 に書き換えて、「Control + X」で保存。ファイル名はそのままで OK なので「Y」。最後に Enter でターミナルに戻る。これで再び
sudo yum install -y gridengine-8.1.9-2.el7.x86_64.rpm gridengine-devel-8.1.9-2.el7.noarch.rpm gridengine-execd-8.1.9-2.el7.x86_64.rpm gridengine-qmaster-8.1.9-2.el7.x86_64.rpm gridengine-qmon-8.1.9-2.el7.x86_64.rpm

とすると、Enable all repositories and try again? と聞かれるので、「y」を選択。するとインストールが進む。途中で「これで良いか？」と聞かれるので再び「y」。最後に「Would you like to permanently enable these repositories? 」と聞かれるので、これには当然「N」を選択。インストールが完了し、/opt/sge に必要なファイルがインストールされている。最後に再び
sudo nano /etc/yum/pluginconf.d/search-disabled-repos.conf

で、notify_only=0 を notify_only=1 に戻して保存。

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（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/sge
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。
以下に示す以外は全てデフォルトのまま Enter で OK。

・Please enter valid user name では”sgeadmin”と入力する。
・Windows Execution Host のインストールは「n」（no）を選択。
・Grid Engine JMX MBean server は「n」を選択。
・Setup spooling では classic を選択。BerkeleyDB は大規模なクラスター用である。
・「設定したパラメータを変更しますか」には「n」を選択。
・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.。
・Shadow host(s) now は「n」を選択。

あとはガイドに従ってそのまま 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~/.bash_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
```

~/.bash_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！
source で両ファイルの変更を反映する。

```
. /root/.bashrc
. ~/.bash_profile
```

またはターミナルを一旦終了し、再度起動する。通常、これで反映するはずである。下記コマンドで環境変数 SGE_CELL が設定されているか確認：

```
echo $SGE_CELL
```

default と表示されない場合、変更が反映されていない。この場合はアカウントをログアウト→ログインして反映させる。

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 root ユーザーになって行う必要がある。

```
sudo su
qconf -am <username>
```

と入力すると、

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

と表示されれば無事追加できている。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/sgemaster 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 ?   Sl   20:11   0:00 /opt/sge/bin/lx-amd64/sge_qmaster
sgeadmin 71559 0.3 0.0 215084 10152 ?   Sl   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 では以下の独自のキューを設定する：
verylong.q - priority => 20
long.q - priority => 15
short.q - priority => 10
veryshort.q - priority => 5

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。Word ファイル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、各行が何をしているか解説すると、

`# change defaults for all.q`	説明用のコメント行。
`qconf -sq all.q |\`	all.q の設定情報の呼び出し
`sed -e 's/bin\/csh/bin\/sh/' |\`	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。
`sed -e 's/posix_compliant/unix_behavior/' |\`	シェルの開始モードを posix_compliant から unix_behavior に変更
`sed -e 's/priority 0/priority 20/' >\`	priority を 0（最高）から 20（最低）に変更
`/tmp/q.tmp`	/tmp/q.tmp として上記を保存
`qconf -Mq /tmp/q.tmp`	/tmp/q.tmp を開く（以下の作業のため）
`# add other queues`	コメント行。
`sed -e 's/all.q/verylong.q/' /tmp/q.tmp >\`	all.q を verylong.q に変更
`/tmp/verylong.q`	/tmp/verylong.q と名前をつけて保存
`qconf -Aq /tmp/verylong.q`	verylong.q をキューとして追加
`sed -e 's/all.q/long.q/' /tmp/q.tmp |\`	all.q を long.q に変更
`sed -e 's/priority *20/priority 15/' >\`	priority を 15 に変更
`/tmp/long.q`	/tmp/long.q として保存
`qconf -Aq /tmp/long.q`	long.q をキューとして追加
`sed -e 's/all.q/short.q/' /tmp/q.tmp |\`	all.q を short.q に変更
`sed -e 's/priority *20/priority 10/' >\`	priority を 10 に設定
`/tmp/short.q`	/tmp/short.q として保存
`qconf -Aq /tmp/short.q`	short.q をキューとして追加
`sed -e 's/all.q/veryshort.q/' /tmp/q.tmp |\`	all.q を veryshort.q に変更
`sed -e 's/priority *20/priority 5/' >\`	priority を 5 に設定
`/tmp/veryshort.q`	/tmp/veryshort.q として保存
`qconf -Aq /tmp/veryshort.q`	veryshort.q を追加

設定が終わったら、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。下のようなウィンドウが立ち上がる。

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。