Hewlett Packard
Enterprise

PBSと利用方法のご紹介

2018年6月20日

目次

1. PBSProfessionalの概要

PBS Professionalとは

スーパーコンピュータなどのハイパフォーマンスコンピューティング（HPC）環境のジョブスケジューリングとワークロード管理を最適化し、システム効率と作業者の生産性を高めるミドルウェアです。

PBS Professional（**以下「PBS」とします**）は、柔軟性の高いワークロード管理システムである、PBS（Portable Batch System）のプロフェッショナルバージョンです。

PBSは、元々はNASAの航空宇宙関連コンピューティングリソースを管理するために開発されました。

それ以来、PBSは、スーパーコンピュータのワークロードの最先端を担い、Linuxクラスタ上のデファクトスタンダードとなっています。

オープンソースライセンス版のPBS について

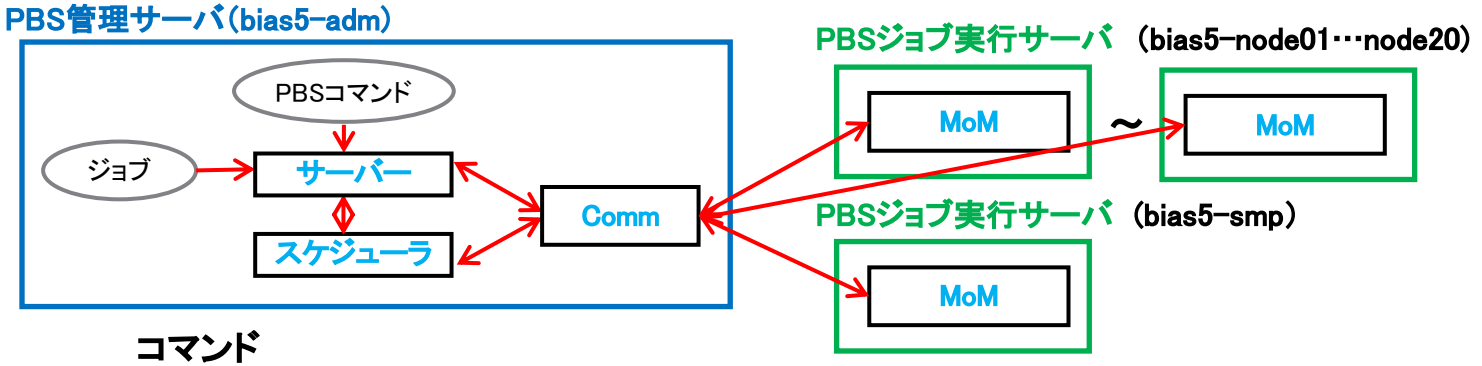
生物情報解析システムでは、オープンソースライセンス版のPBSを使用します。

同PBSは、HPC（ハイパフォーマンスコンピューティング）のOpenHPCプロジェクトの一環として、PBS開発元のAltair社より、2016年6月20日にオープンソース提供されたものです。

有償版のPBS バージョン14.1の機能をベースとしています。

PBSの構成

PBSは、以下に示すように一連のコマンドおよびシステムデーモン / サービスで構成されています。



コマンド

PBSにはジョブを投入、監視、変更、および削除する一連のコマンドが用意されています。
PBSコマンドには、一般ユーザーが実行できるコマンドや管理者権限またはオペレータ権限が必要なコマンドがあります。

ジョブ

ジョブは実行するコマンドやアプリケーションを記述しているシェルスクリプト形式のタスクです。

サーバー

サーバーは、PBSのジョブを管理します。
PBSコマンドがPBSサーバーと通信し、ジョブがサーバーに投入され、サーバーがジョブをキューイングして実行サーバに送信します。

スケジューラ

スケジューラは、PBSで指定されているポリシーに従って、ジョブを実行します。
スケジューラは、各ジョブの要件と使用可能なリソースを照合し、ポリシーに従ってリソースを割り当てます。

MoM(Machine-oriented Mini-serverの略語です)

ジョブ実行サーバーに送信されたジョブは MoMが管理します。 各実行サーバーでは、1つのMoMでジョブを管理します。
MoMは、各ジョブを起動して監視し、ジョブ投入者にジョブの出力を返します。

Comm(Communication Daemonの略語です)

Commは、サーバー、スケジューラ、MoMの各デーモンを仲介するものです。

2. ジョブを投入する

キューの選択

	分散処理用計算機クラスタ(bias5-node01～node20)			共有メモリ計算サーバ(bias5-smp)		
キュー名	small(default)	medium	large	smps	smpm	smpl
ジョブの特徴	短時間・並列多	中規模	長時間	中メモリ	大メモリ	最大メモリ
ジョブ実行サーバ	bias5-node01～ bias5-node20	bias5-node01～ bias5-node20	bias5-node01～ bias5-node20	bias5-smp (ldas-smp)	bias5-smp (ldas-smp)	bias5-smp (ldas-smp)
最大実行時間	6時間	72時間	no limit	no limit	no limit	no limit
ジョブで利用できる最大メモリ	96GB	96GB	96GB	500GB	1TB	3TB
キューで利用できるジョブ数	no limit	no limit	no limit	6	3	1
キューで利用できるCPU数	580	200	20	48	48	36
ユーザー人当たりのCPU数(サーバ全体)	480					
ユーザー人当たりのCPU数(キューあたり)	400	150	10	no limit	no limit	no limit
デフォルトCPU数	1	1	1	1	1	1
デフォルトメモリサイズ	3GB	3GB	3GB	250GB	500GB	1500GB

利用目的に合わせての上記の各キューをご利用ください。

ジョブの投入

$ qsub オプション スクリプト

qsubの実行例

①キューを指定しない、または、スクリプト内でキュー名を指定している場合

```
$ qsub test.sh
4720.bias5-adm  ← JOBIDが表示されます。
                  JOBIDの後ろには“.bias5-adm”が付与されます。
```

・スクリプト内でキュー名を指定していない場合、デフォルトの「smallキュー」に投入されます。

・スクリプト内でキュー名を指定している場合、指定されたキューに投入されます。

②qsubでキューを指定する場合、または、スクリプト内で指定しているキューと異なるキューに投入する場合

```
$ qsub -q large test.sh
4722.bias5-adm
```

・-qで指定したキューに投入されます。

qsubのオプション

qsubコマンドの主なオプションをご紹介します。 これ以外のオプションについては、man qsubを参照願います。

-e 標準エラーファイル名	標準エラー出力のファイル名を指定します。
-I	インタラクティブ・バッチジョブを投入します。
-j オプション	ジョブの標準出力ファイルと標準エラー出力ファイルを結合するように指定します。 oe : 標準出力ファイルに結合します。 eo : 標準エラー出力ファイルに結合します。
-J X-Y[:Z]	ジョブアレイ範囲（アレイジョブの実行時に使用します）
-l リソースリスト	ジョブで使用するCPU数、メモリ、実行サーバを指定します。
-m オプション	メールを送信する条件を設定します。 a, b, e のうち1つ以上の文字を指定します。（デフォルト -m a） a : ジョブが中止（aborted）された場合にメールを送信します。 b : 実行開始（begin）時にメールを送信します。 c : メールを送信しません（not）。 e : ジョブの終了時にメールを送信します。
-N ジョブ名	ジョブ名を指定します。
-o 標準出力ファイル名	標準出力のファイル名を指定します。
-q キュー名	キュー名を指定します。（指定が無い場合、デフォルトのsmallキューの指定となります。
-S インタプリンタ名	ジョブスクリプトのインタプリンタ名を指定します（/bin/sh等）
-v 変数リスト	環境変数等の設定をジョブに持たせる場合に使用します。 例）-v var1=10, “var2= ‘A,B’ ”
-V	ログイン環境の環境変数を全て設定した状態でジョブを実行します。

PBSのジョブ投入スクリプトについて

```
#!/bin/sh          ←①シェルの指定
#PBS -q medium     ←②qsub指示文
#                  ←③コメント
COMMAND_PATH=" $HOME"  ┐
RUNCOMMAND=" a.out"    ┘ ④実行するタスク（スクリプト本体）
cd $PBS_O_WORKDIR ←⑤ジョブ投入ディレクトリへの移動
$HOME/a.out   ←⑥実行コマンド
exit
```

PBSのジョブスクリプトは、上記のもので構成されています。

①シェルの指定 ：ジョブスクリプトのシェルの種類を指定します。

②qsub指示文

ジョブスクリプトの先頭を #PBSで始めると、qsubコマンドのオプション指示として処理されます。

指示文は、#PBS文で指定する方法以外に qsubコマンド実行時のオプションで指定する事も可能です。

指示文で指定したオプションと異なるオプションを qsubコマンド実行時に指定した場合、**qsubコマンドのオプションが優先されます**。

③コメント： 一般のシェルスクリプトと同様、先頭が #のみであればコメントとみなされます。

④実行するタスク ： パスの設定、変数のセット等スクリプト本体の記述をします。

⑤ジョブ投入ディレクトリへの移動

qsubコマンドを実行したディレクトリに移動します。

$PBS_O_WORKDIRには qsub実行ディレクトリ名が設定されます。

デフォルトで PBSは、ジョブの終了時に標準出ファイルおよび標準エラーファイルを投入サーバの$PBS_O_WORKDIRに返します。

⑥実行コマンド：実行するアプリケーション、コマンドを指定します。

SGEのジョブ投入スクリプトをPBSの投入スクリプトに変更

SGEのジョブ投入スクリプトをPBS用に変更する方法をご紹介します。

SGEスクリプト	PBSスクリプト
#!/bin/sh	#!/bin/sh
	#PBS -l **ncpus=CPU数**
#$ -cwd	該当する指示文はありません、下記の「**cd ${PBS_O_WORKDIR}**」が相当します。
#$ -t 1-150	#PBS -J 1-150
#$ -q medium	#PBS -q medium
#$ -N blastjob	#PBS -N blastjob
#$ -S /bin/sh	#PBS -S /bin/sh
DB=/bio/db/blast/db/nr	DB=/bio/db/blast/db/nr
SEQDIR=.	SEQDIR=.
OUTPUTDIR=./_xml	OUTPUTDIR=./_xml
PROGRAM=blastx	PROGRAM=blastx
OTHER_OPT= “-outfmt 5 -evalue 0.001 -num_threads **2**	OTHER_OPT= “-outfmt 5 -evalue 0.001 -num_threads **$NCPUS**”
	cd ${PBS_O_WORKDIR}
$PROGRAM -db $DB -query ./nohit_longest.fasta.$SGE_TASK_ID	$PROGRAM -db $DB -query ./nohit_longest.fasta.$PBS_ARRAY_ID

注）上記の**ncpus = cpu数**で、変数 ncpus にジョブで使用するCPU数（**core数**）を必ず指定してください。
ncpusの指定があるとPBSは、指定CPU数の空きがあるノードにジョブをディスパッチし、PBS環境変数 **NCPUS** には指定されたCPU数がセットされます。

SMP並列(OpenMP)ジョブのスクリプト例(4並列の場合)

```
#!/bin/sh
# OpenMP sample jobscript
#
#PBS -l ncpus=4
#PBS -q large
#PBS -N TEST01
export OMP_NUM_THREADS=$NCPUS
cd ${PBS_O_WORKDIR}
./myprog  input_data
```

ncpus=cpu数で使用するCPU数(**core数**)を指定してください。

環境変数 **OMP_NUM_THREADS**には、**$NCPUS**(ncpusに指定したCPU数)をセットしてください。

MPI並列ジョブのスクリプト例（80並列の例、2ノードで40*2）

```
#!/bin/sh
# MPI sample jobscript
#
#PBS -l select=2:ncpus=40
#PBS -q medium
#PBS -N TEST02
export PATH=/usr/mpi/gcc/openmpi-3.1.0rc2/bin:$PATH
export LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-3.1.0rc2/lib64/:$LD_LIBRARY_PATH
cd ${PBS_O_WORKDIR}
mpirun -n $NCPUS -hostfile $PBS_NODEFILE ./myprog input_data
```

select=ノード数で使用する ノード数、**ncpus=cpu数**で使用するCPU数（**core数**）を指定してください。

実行する場合、必ず "mpirun -n **$NCPUS**" を記述してください.

PBS環境変数 **$PBS_NODEFILE**には、ジョブが実行されるノードのホスト名がPBSでセットされます。

3. キューとジョブの状況を確認する

キューの状況を確認(1/3)

$ qstat -Q オプション

qstat -Qの実行例

①全てのキューの一覧と現在の状況を確認する

$ qstat -Q

```
$ qstat -Q
Queue            Max   Tot Ena Str   Que   Run   Hld   Wat   Trn   Ext Type
---------------- ----- ----- --- --- ----- ----- ----- ----- ----- ----- ----
medium             0     7 yes yes     0     6     0     0     0     0 Exec
large              0     2 yes yes     0     2     0     0     0     0 Exec
mbgd               0     0 yes yes     0     0     0     0     0     0 Exec
small              0   273 yes yes   218    55     0     0     0     0 Exec
smps               0     1 yes yes     0     1     0     0     0     0 Exec
test               0     0 yes yes     0     0     0     0     0     0 Exec
smpm               0     3 yes yes     0     3     0     0     0     0 Exec
smpl               0     1 yes yes     0     1     0     0     0     0 Exec
mbgd-smp           0     0 yes yes     0     0     0     0     0     0 Exec
```

○ジョブ数

R ： 実行中ジョブ数

Q ： キュー待ちジョブ数

キューの状況を確認(2/3)

②指定したキューの現在の状況を確認する

```
$ qstat -Q キュー名
```

```
$ qstat -Q smpm
Queue             Max   Tot Ena Str   Que   Run   Hld   Wat   Trn   Ext Type
----------------- ----- ----- --- --- ----- ----- ----- ----- ----- ----- ----
smpm                0     8 yes yes     1     7     0     0     0     0 Exec
```

キューの状況を確認(3/3)

③指定したキューの設定内容と現在の状況を確認する

```
$ qstat -Qf キュー名
```

```
$ qstat -Qf small
Queue: small
    queue_type = Execution
    total_jobs = 316
    state_count = Transit:0 Queued:141 Held:0 Waiting:0 Running:175 Exiting:0 B
        egun:0
    resources_max.mem = 96gb
    resources_max.walltime = 06:00:00
    resources_default.mem = 3gb
    resources_default.ncpus = 1
    default_chunk.Qlist = SMALL
    resources_available.ncpus = 580
    resources_assigned.mem = 537600mb
    resources_assigned.ncpus = 400
    resources_assigned.nodect = 175
    max_run_res.ncpus = [u:PBS_GENERIC=400]
    enabled = True
    started = True
```

紫　ジョブ数

Transit	別のキューに移動中ジョブ数
Queued	キュー待ちジョブ数
Held	ホールド(保留中)ジョブ数
Waiting	ジョブ実行待機中ジョブ数
Running	実行中ジョブ数
Exiting	終了中ジョブ数
Begun	開始中ジョブ数

緑　設定内容

resources_max.mem	ジョブで利用できる最大メモリ
resources_max.walltime	最大実行時間
resources_default.mem	デフォルトメモリサイズ
resources_default.ncpu	デフォルトCPU数
resources_available.ncpus	キューで利用できるCPU数

max_run_res.ncpus = [u:PBS_GENERIC=400]
　ユーザー一人当たりのCPU数（キューあたり）

赤　現在の状況

resources_assigned.mem	現在のキューの使用メモリ
resources_assigned.ncpus	現在のキューの使用CPU数
resources_assigned.nodec	現在のキューの実行ジョブ数

ジョブの状況を確認(1/10)

```
$ qstat オプション
```

qstatの実行例

ここでは、よく使うものをご紹介します、これ以外の使い方については、「man qstat」を参照してください。

①実行中、キュー待ちジョブを含む全てのユーザの全てのジョブを確認する

```
$ qstat -a
```

```
$ qstat -a

bias5-adm:

                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
362.bias5-adm   sgiadm   small    TESTJOB     37276   1   1    6mb 06:00 R 00:00
363.bias5-adm   sgiadm   smpm     TESTJOB    145771   1   1  114gb   --  R 00:00
364.bias5-adm   sgiadm   smpm     TESTJOB    145814   1   1  114gb   --  R 00:00
365.bias5-adm   sgiadm   smpm     TESTJOB    145858   1   1  114gb   --  R 00:00
366.bias5-adm   sgiadm   smpm     TESTJOB    145902   1   1  114gb   --  R 00:00
367.bias5-adm   sgiadm   smpm     TESTJOB    145946   1   1  114gb   --  R 00:00
368.bias5-adm   sgiadm   smpm     TESTJOB    145990   1   1  114gb   --  R 00:00
369.bias5-adm   sgiadm   smpm     TESTJOB    146033   1   1  114gb   --  R 00:00
370.bias5-adm   sgiadm   smpm     TESTJOB       --    1   1  114gb   --  Q   --
```

○ジョブのステータス

R ： 実行中

Q ： キュー待ち中

ジョブの状況を確認(2/10)

②全ユーザの実行中ジョブを確認する

```
$ qstat -r
```

```
$ qstat -r

bias5-adm:

                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
362.bias5-adm   sgiadm   small    TESTJOB     37276   1   1    6mb 06:00 R 00:00
363.bias5-adm   sgiadm   smpm     TESTJOB    145771   1   1  114gb   --  R 00:00
364.bias5-adm   sgiadm   smpm     TESTJOB    145814   1   1  114gb   --  R 00:00
365.bias5-adm   sgiadm   smpm     TESTJOB    145858   1   1  114gb   --  R 00:00
366.bias5-adm   sgiadm   smpm     TESTJOB    145902   1   1  114gb   --  R 00:00
367.bias5-adm   sgiadm   smpm     TESTJOB    145946   1   1  114gb   --  R 00:00
368.bias5-adm   sgiadm   smpm     TESTJOB    145990   1   1  114gb   --  R 00:00
369.bias5-adm   sgiadm   smpm     TESTJOB    146033   1   1  114gb   --  R 00:00
```

ジョブの状況を確認(3/10)

③自分の実行中ジョブを確認する

```
$ qstat -r -u Myユーザ名
```

```
$ qstat -r -u sgiadm

bias5-adm:

                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
362.bias5-adm   sgiadm   small    TESTJOB     37276   1   1    6mb 06:00 R 00:00
363.bias5-adm   sgiadm   smpm     TESTJOB    145771   1   1  114gb   --  R 00:00
364.bias5-adm   sgiadm   smpm     TESTJOB    145814   1   1  114gb   --  R 00:00
365.bias5-adm   sgiadm   smpm     TESTJOB    145858   1   1  114gb   --  R 00:00
366.bias5-adm   sgiadm   smpm     TESTJOB    145902   1   1  114gb   --  R 00:00
367.bias5-adm   sgiadm   smpm     TESTJOB    145946   1   1  114gb   --  R 00:00
368.bias5-adm   sgiadm   smpm     TESTJOB    145990   1   1  114gb   --  R 00:00
369.bias5-adm   sgiadm   smpm     TESTJOB    146033   1   1  114gb   --  R 00:00
```

・qstatの表示は、デフォルトで全ユーザの表示になります。
その為、自分のジョブを確認する場合、左記のように
"-u Myユーザ名" の指定が必要となります。

・以下のエイリアス設定を行うと、自分のジョブのみが表示されます。

Bシェル系

.bashrc .profile等に以下を追加します

```
alias qstat="qstat -u $USER"
```

Cシェル系

.cshrc等に以下を追加します

```
alias qstat "qstat -u $USER"
```

ジョブの状況を確認(4/10)

④指定したキューの自分の実行中ジョブを確認する

```
$ qstat -r -u Myユーザ名 キュー名
```

```
$ qstat -r -u sgiadm small

bias5-adm:
                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
362.bias5-adm   sgiadm   small    TESTJOB     37276   1   1     6mb 06:00 R 00:00
```

ジョブの状況を確認(5/10)

⑤自分のジョブが実行されているサーバ名(ノード名)を確認する

```
$ qstat -r -u  Myユーザ名 キュー名 -n -1
```

```
$ qstat -r -u sgiadm -n -1

bias5-adm:

                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
8354.bias5-adm  sgiadm   medium   NODEJOB     11254   1  40    3gb 72:00 R 00:01 bias5-node19/0*40
8355.bias5-adm  sgiadm   medium   NODEJOB     61264   1  40    3gb 72:00 R 00:00 bias5-node20/0*40
                                                                                 注)
```

注) ①　　/②*③
bias5-node19/0*40
①ジョブが実行しているサーバ名
②ジョブがサーバのcpuに割り当てられたシーケンス番号（意識する必要はありません）
③ジョブが使用しているcpu数

ジョブの状況を確認(6/10)

⑥キュー待ちとなっている自分のジョブを確認する

```
$ qstat -i -u Myユーザ名
```

```
$ qstat -i -u sgiadm

bias5-adm:

                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
9625.bias5-adm  sgiadm   large    NODEJOB        --   1  20    3gb   --  Q   --
9629.bias5-adm  sgiadm   smps     SMP_JOB        --   1   2   20gb   --  Q   --
```

ジョブの状況を確認(7/10)

⑦キュー待ちとなっている原因を確認する

```
$ qstat -f ジョブID
```

＊例：ユーザー人当たりのCPU数のリミットを超過した場合

```
$ qstat -f 9625
Job Id: 9625.bias5-adm
    Job_Name = NODEJOB
    Job_Owner = sgiadm@bias5-login
    job_state = Q
    queue = large
                    :
                    : (省略)
    Resource_List.mem = 3gb
    Resource_List.ncpus = 20
    Resource_List.nodect = 1
    Resource_List.place = pack
    Resource_List.select = 1:mem=3gb:ncpus=20
                    :
                    : (省略)
    comment = Not Running: Queue large per-user limit reached on resource ncpus
    etime = Thu Jun 14 19:53:52 2018
    Submit_arguments = -q large testNode.sh
    project = _pbs_project_default
```

ジョブの状況を確認(8/10)

⑧アレイジョブの投入～状況を確認する(1/3)

以下のサンプルスクリプトをアレイジョブとして投入して、状況を確認する例をご紹介します。

```
$ cat testarray.sh
#!/bin/sh
# test jobscript
#
#PBS -N ArrayExample
#PBS -q smps
#PBS -J 1-5
cd ${PBS_O_WORKDIR}
sleep 600
```

アレイジョブの投入

```
$ qsub testarray.sh
516[].bias5-adm
```

ジョブの状況を確認(9/10)

⑨アレイジョブの投入～状況を確認する(2/3)

アレイジョブの状況確認

◇通常のqstat

```
$ qstat -u sgiadm
Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
516[].bias5-adm sgiadm   smps     ArrayExamp     --   1   1   29gb   --  B   --
```

◇アレイジョブのみを確認する（オプション -J）

```
$ qstat -J
Job id            Name             User             Time Use S Queue
----------------  ---------------- ---------------- -------- - -----
516[].bias5-adm   ArrayExample     sgiadm                 0 B smps
```

◇サブジョブを含む全てのアレイジョブを確認する（オプション -t）

```
$ qstat -t
Job id            Name             User             Time Use S Queue
----------------  ---------------- ---------------- -------- - -----
516[].bias5-adm   ArrayExample     sgiadm                 0 B smps
516[1].bias5-adm  ArrayExample     sgiadm          00:00:00 R smps
516[2].bias5-adm  ArrayExample     sgiadm          00:00:00 R smps
516[3].bias5-adm  ArrayExample     sgiadm          00:00:00 R smps
516[4].bias5-adm  ArrayExample     sgiadm          00:00:00 R smps
516[5].bias5-adm  ArrayExample     sgiadm          00:00:00 R smps
```

S ： job state

B ： アレイジョブ

Time Use ： CPU時間

ジョブの状況を確認(10/10)

⑩アレイジョブの投入～状況を確認する(3/3)

◇ジョブの詳細を確認する（オプション -f）

```
$ qstat -f "516[1]"  ← ジョブIDを文字列リテラルで囲んでください
Job Id: 516[1].bias5-adm
    Job_Name = ArrayExample
    Job_Owner = sgiadm@bias5-login
    resources_used.cpupercent = 0
    resources_used.cput = 00:00:00
    resources_used.mem = 4136kb
    resources_used.ncpus = 1
    resources_used.vmem = 357328kb
    resources_used.walltime = 00:04:11
    job_state = R
    queue = smps
                :
                : (省略)
    Resource_List.mem = 29gb
    Resource_List.ncpus = 1
    Resource_List.nodect = 1
    Resource_List.place = pack
    Resource_List.select = 1:mem=29gb:ncpus=1
                :
                : (省略)
    array_id = 516[].bias5-adm
    array_index = 1
    project = _pbs_project_default
```

4. ジョブの履歴を確認する

ジョブの履歴を確認(1/3)

```
$ tracejob オプション ジョブID
```

tracejobコマンドで、指定したジョブの履歴が確認できます。

tracejob の実行例

①当日の指定ジョブの履歴を確認する（オプション指定無しで tracejobコマンドを実行すると、当日の情報表示となります）

```
$ tracejob ジョブID
```

```
$ tracejob 6278

Job: 6278.bias5-adm

06/13/2018 11:26:30  L    Considering job to run
06/13/2018 11:26:30  S    enqueuing into small, state 1 hop 1
06/13/2018 11:26:30  S    Job Queued at request of sgiadm@bias5-login, owner = sgiadm@bias5-login, job name = TESTJOB, queue = small
06/13/2018 11:26:30  S    Job Run at request of Scheduler@bias5-adm on exec_vnode (bias5-node10:mem=3145728kb:ncpus=4)
06/13/2018 11:26:30  S    Job Modified at request of Scheduler@bias5-adm
06/13/2018 11:26:30  L    Job run
06/13/2018 11:36:30  S    Obit received momhop:1 serverhop:1 state:4 substate:42
06/13/2018 11:36:30  S    Exit_status=0 resources_used.cpupercent=0 resources_used.cput=00:00:00 resources_used.mem=4000kb
resources_used.ncpus=4 resources_used.vmem=357328kb resources_used.walltime=00:10:00
```

ジョブの履歴を確認(2/3)

②指定された日数分まで遡って指定したジョブの履歴を確認する(当日より過去に終了したジョブの履歴を確認)

```
$ tracejob -n日数 ジョブID
```

ここ3日間に終了した JOBID 227の履歴を確認

```
$ tracejob -n3 227

Job: 227.bias5-adm

06/06/2018 10:41:39  L    Considering job to run
06/06/2018 10:41:39  S    Job Queued at request of mbgdadm@bias5-login, owner = mbgdadm@bias5-login, job name = mbgd.csh, queue = mbgd
06/06/2018 10:41:39  S    Job Run at request of Scheduler@bias5-adm on exec_vnode (bias5-node01:ncpus=4:mem=4194304kb)
06/06/2018 10:41:39  S    Job Modified at request of Scheduler@bias5-adm
06/06/2018 10:41:39  L    Job run
06/06/2018 10:41:39  S    enqueuing into mbgd, state 1 hop 1
06/06/2018 10:41:50  S    Obit received momhop:1 serverhop:1 state:4 substate:42
06/06/2018 10:43:24  S    Post job file processing error
06/06/2018 10:43:24  S    Obit received momhop:1 serverhop:1 state:5 substate:53
06/06/2018 10:43:24  S    Exit_status=0 resources_used.cpupercent=0 resources_used.cput=00:00:00 resources_used.mem=4148kb
resources_used.ncpus=4  resources_used.vmem=373484kb resources_used.walltime=00:00:11
06/07/2018 10:45:19  S    dequeuing from mbgd, state 9
```

ジョブの履歴を確認(3/3)

$ qstat -x オプション

qstat -x で 24時間以内に実行されたジョブの実行履歴が確認できます。

注）24時間以上過去のジョブについては、qstat -x では確認ができません。

qstat x の実行例

```
$ qstat -x -u sgiadm

bias5-adm:
                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
342.bias5-adm   sgiadm   small    TESTJOB     72074   1   1    6mb 06:00 F 00:10
343.bias5-adm   sgiadm   smpm     TESTJOB    130917   1   1  114gb   --  F 00:10
344.bias5-adm   sgiadm   smpm     TESTJOB    130961   1   1  114gb   --  F 00:10
345.bias5-adm   sgiadm   smpm     TESTJOB    131005   1   1  114gb   --  F 00:10
346.bias5-adm   sgiadm   smpm     TESTJOB    131049   1   1  114gb   --  F 00:10
347.bias5-adm   sgiadm   smpm     TESTJOB    131153   1   1  114gb   --  F 00:10
348.bias5-adm   sgiadm   smpm     TESTJOB    131197   1   1  114gb   --  F 00:10
349.bias5-adm   sgiadm   smpm     TESTJOB    131242   1   1  114gb   --  F 00:10
```

5. ジョブを削除する

ジョブの削除

```
$ qdel ジョブID...ジョブID
```

qdelの実行例

```
$ qstat -a -u sgiadm

bias5-adm:
                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
414.bias5-adm   sgiadm   small    TESTJOB     77441   2  80     6mb 06:00 R 00:00

$ qdel 414
```

＊qdel 415 416 417 のように複数のジョブIDを指定することも可能です。

ジョブの一括削除(1/4)

```
$ qdel `qselect オプション -u Myユーザ名`
```

・qdelコマンドにジョブを抽出する qseletコマンドを組み合わせて一括削除を行います。
・誤って想定していないジョブまでを削除しないようにqdelを行う前にqselectで該当ジョブを確認後、一括削除することをお勧めします。

①自分の全てのジョブを一括削除する

```
$ qstat -a -u sgiadm -n -1    ← 全てのジョブを確認

                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
6017.bias5-adm  sgiadm   large    NODE_JOB    64369   1   4    3gb   --  R 00:01 bias5-node09/2*4
6018.bias5-adm  sgiadm   large    NODE_JOB      --    1   4    3gb   --  Q   --   --
6019.bias5-adm  sgiadm   smpm     SMP_JOB    109960   1   1   10gb   --  R 00:00 bias5-smp/0
6020.bias5-adm  sgiadm   smpm     SMP_JOB       --    1   1   10gb   --  Q   --   --

$ qselect -u sgiadm    ← 一括削除の事前確認
6017.bias5-adm
6018.bias5-adm
6019.bias5-adm
6020.bias5-adm

$ qdel `qselect -u sgiadm`  ← 全てのジョブを削除

$ qselect -u sgiadm
→ ジョブの表示が無いことを確認
```

ジョブの一括削除(2/4)

②指定したキューの自分の全てのジョブを一括削除する

```
$ qstat -a -u sgiadm -n -1  ← 全てのジョブを確認

Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
6024.bias5-adm  sgiadm   large    NODE_JOB    76158   1   4    3gb   --  R 00:00 bias5-node08/3*4
6025.bias5-adm  sgiadm   large    NODE_JOB    74874   1   4    3gb   --  R 00:00 bias5-node09/1*4
6026.bias5-adm  sgiadm   large    NODE_JOB      --    1   4    3gb   --  Q   --   --
6027.bias5-adm  sgiadm   smps     SMP_JOB    113391   1   2   20gb   --  R 00:00 bias5-smp/0*2

$ qselect -u sgiadm -q large   ← largeキュージョブ一括削除の事前確認
6024.bias5-adm
6025.bias5-adm
6026.bias5-adm

$ qdel `qselect -u sgiadm -q large`   ← largeキュージョブの一括削除

$ qstat -a -u sgiadm -n -1

Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
6027.bias5-adm  sgiadm   smps     SMP_JOB    113391   1   2   20gb   --  R 00:01 bias5-smp/0*2
```

→ largeキューのジョブ表示が無いことを確認

ジョブの一括削除(3/4)

③自分の全てのキュー待ちジョブを一括削除する

```
$ qstat -a -u sgiadm -n -1 ← 全てのジョブを確認

Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
6031.bias5-adm  sgiadm   smpm     SMP_JOB    117014   1   2   20gb   --  R 00:01 bias5-smp/3*2
6032.bias5-adm  sgiadm   smpm     SMP_JOB       --    1   2   20gb   --  Q   --   --
6033.bias5-adm  sgiadm   smps     SMP_JOB    117482   1   2   20gb   --  R 00:00 bias5-smp/0*2
6034.bias5-adm  sgiadm   smps     SMP_JOB       --    1   2   20gb   --  Q   --   --

$ qselect -s Q -u sgiadm  ← キュー待ちジョブ一括削除の事前確認
6032.bias5-adm
6034.bias5-adm

$ qdel `qselect -s Q -u sgiadm`  ←キュー待ちジョブの一括削除

$ qstat -a -u sgiadm -n -1
                                                          Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
6031.bias5-adm  sgiadm   smpm     SMP_JOB    117014   1   2   20gb   --  R 00:05 bias5-smp/3*2
6033.bias5-adm  sgiadm   smps     SMP_JOB    117482   1   2   20gb   --  R 00:00 bias5-smp/0*2
```

→ キュー待ちジョブの表示が無いことを確認

ジョブの一括削除(4/4)

④自分の全ての実行中ジョブを一括削除する

```
$ qstat -a -u sgiadm -n -1  ← 全てのジョブを確認

Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
6036.bias5-adm  sgiadm   smps     SMP_JOB    117570   1   2   20gb   --  R 00:06 bias5-smp/2*2
6038.bias5-adm  sgiadm   large    NODE_JOB    17021   1  10    3gb   --  R 00:00 bias5-node11/0*10
6039.bias5-adm  sgiadm   large    NODE_JOB       --   1  10    3gb   --  Q   --   --

$ qselect -s R -u sgiadm  ← 実行中ジョブ一括削除の事前確認
6036.bias5-adm
6038.bias5-adm

$ qdel `qselect -s R -u sgiadm` ← 実行中ジョブの一括削除

$ qstat -a -u sgiadm -n -1

Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
6039.bias5-adm  sgiadm   large    NODE_JOB    19549   1  10    3gb   --  R 00:00 bias5-node11/0*10
```

→ ジョブ 6036, 6038が削除され 6039が実行中となったことを確認

6. ジョブ投入時の注意事項

メモリを指定してジョブを投入する時のメモリ指定方法

PBS指示文 ncpus=cpu数の次に ″:″を付けて、mem=形式で使用するメモリサイズを指定してください。
（メモリの指定が無い場合、前述の「キューの選択」にある、デフォルトメモリサイズが指定されたものとされます）

```
#PBS -l ncpus=cpu数:mem=XXXX
```

メモリの単位として tb, gb, mb, kb, b が指定可能です（1gb=1024mb, 1mb=1024kb, 1kb=1024bとなります）

単位の指定がない場合は、b(バイト) が指定されたものとして扱われます。

また、小数点は使用できません。
従って、1.5GBのメモリを指定したい場合は、″mem=1536mb″のように指定します。

bias5-node01～20で 40並列を超えるジョブは投入できません

例）48並列のジョブ投入

48並列のジョブを投入する為にスクリプトで #PBS -l ncpus=48を指定、または qsub -l ncpus=48で投入するとクラスタ 1ノードあたり 40cpuの為、
以下のようにキュー待ちとなり、このジョブは流れません（qdelで削除が必要となります）

```
$ qstat -s -1 15368

                                                                Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
15368.bias5-adm sgiadm   small    NODEJOB1       --    1  48    3gb 06:00 Q   --    Not Running: Insufficient amount of resource: ncpus
```

Hewlett Packard
Enterprise

Thank you