

NQS の使い方

—UNIX バッチ処理（計算機をブン回す人のために）—

総合情報処理センター 三 上 秀 秋

mikami@cc.hirosaki-u.ac.jp

コンピュータに仕事をさせる方法を処理方式で分けると会話型処理と一括処理に分けられます。会話型処理は計算機と会話しながら、仕事を進めて行く方法でメールやニュース、短時間で可能な計算処理などがあります。一括処理はバッチ処理とも言いますが、コンピュータに処理する手順が予め決まっています、その手順書（シェル、コマンド）を計算機に投入（入力）すると後は端末を切断しても計算機が自動的に処理する方法です。当然ながらバッチ処理は、長時間の計算やコマンド（シェル）を実行する場合に便利です。

NQS「Network Queuing System」はUNIX上でバッチ処理を行うためのシステムであり、当センターではUP4800(owani)のローカルバッチとACOSからowaniへ、owaniからACOSへのネットワークを通じたジョブの投入がNQSの機能を使用して利用できます。

1. NQSの概要

NQSでは、処理する単位をリクエストと呼んでいます。バッチ処理を行うリクエストはバッチリクエストです。このバッチリクエストをNQSに投入することにより、ホスト計算機は処理を実行し、結果はネットワークを通じて返します。このバッチリクエストの実体は、ワークステーションの場合はシェルスクリプトであり、ACOSの場合はバッチ処理を実行するJCL (Job Control Language) にあたります。

また、利用者からのリクエストを受け取り、一旦貯めておくところをNQSではキューと呼んでいます。さらにキューはローカルバッチを行うバッチキュー、ネットワークを通じて他ホストにバッチを実行させるパイプキューに分けられます。

当センターのバッチキューの種類

ホスト名	キューの種類	キュー名	備 考
owani	バッチ パイプ	owani acos	ローカルバッチ用 owaniからACOSを使う
LANPO(ACOS)	パイプ	owani	ACOS からowani を使う

2. 設定

NQSを使う場合は、次のような簡単な設定が必要です。但し、UP4800(owani) をローカルバッチで使う場合は必要ありません。

ACOSからUP4800(owani)、又はUP4800(owani) からACOSを使う場合の設定

①UP4800(owani) のファイル「.rhosts」に次の一行のテキストを入れておきます。

```
lanp0 userid ————— 「userid」は、「owani」の利用者ID
```

②ACOSのファイル「.NQSUSER」に次の一行のテキストを入れておきます。

```
OWANI userid1 unerNO userid2 —— userid1は「owani」の利用者ID  
userNOは「owani」の利用者番号  
userid2は「ACOS」の利用者ID
```

```
「owani」の利用者番号は「id」コマンドで調べることができます  
%id  
uid=userNO(userid) gid=groupNO(groupid)
```

3. NQSの主要コマンド

①バッチジョブの投入

qsub コマンドにより、バッチジョブを投入できます。

```
%qsub [option] [shell-script-file]
```

主要オプション	-a	指定時間後にリクエストを実行
	-eo	標準エラー出力を標準の保存先に出力する
	-lm	プロセスごとのメモリサイズ制限の設定
	-lt	プロセスごとのCPU時間制限の設定
	-o	指定目的地へ標準出力を向ける
	-q	指定キューへリクエストを向ける

例1. ホストowaniからバッチキューowaniを使う

```
%qsub -eo シェルスクリプトファイル名  
Request 0070.owani submitted to queue: owani.
```

キューの既定値がowaniなので-qオプションは必要なし

例2. ホストowaniからパイプキューacosを使う

```
%qsub -eo -q acos シェルスクリプトファイル名
Request 0075.owani submitted to queue: acos.
```

例3. ホストLANPO (ACOS) からパイプキューowaniを使う

```
SYSTEM ?qsub -eo シェルスクリプトファイル名
Request 0079.LANPO submitted to queue: owani.
```

ACOSのキューはowaniだけなので-qオプションは必要なし

②バッチリクエストの状態表示

qstat コマンドはキューの状態を表示できます。

```
%qstat [-a] [-l] [-m] [-u username] [-x] [queue-name ...]
```

主要オプション

- a すべてのリクエストを表示
- l リクエストを長い形式で表示
- m リクエストを中間形式で表示
- u user-name

user-name が所有するリクエストのみ表示

例 owaniからqstatコマンドを入力

```
%qstat

owani@owani: type=BATCH; [ENABLED, RUNNING] pri=30
  0 exit; 2 run; 0 stage 0 queued; 0 wait; 0 hold 0 arrive

  REQUEST NAME      REQUEST ID      USER PRI STATE RGRP
  <1 request RUNNING>
  2:      batchfile      83.owani      userid 31  RUNNING nnn

acos@owani: type=PIPE; [ENABLED, INACTIVE]; pri=30
  0 depart; 0 route; 0 queued; 0 wait; 0 hold 0 arrive;
```

owani@owani owaniのローカルバッチキュー

acos@owani owaniからACOSをネットワークで利用するパイプキュー

STATEの意味

RUNNING	実行中状態
QUEUED	実行待ち状態 (スケジュール対象)
WAITNIG	実行遅延状態 (スケジュール対象外)
HOLDING	ホールド状態 (スケジュール対象外)

③バッチリクエストの取り消し

qdelコマンドは、キューにたまったバッチリクエストや実行中のバッチリクエストを削除するときに使います。削除するときに必要なリクエストIDはバッチリクエストを投入するときに表示されます。

```
%qdel [-k] [-h hostname] requestID ...
```

主要オプション	-k	実行中のリクエストを中止
	-h hostname	実行中のホスト名の指定

例1. owani上のキューowaniの実行中のバッチリクエストをキャンセルする

```
%qdel -k 82 (リクエストID 82番のジョブをキャンセルする)  
Request 82 is running, and has been signalled.
```

例2. ACOS上のキューowaniの実行中のバッチリクエストをキャンセルする

```
SYSTEM ?qdel -k -h owani 85 (ACOS からowani 上のリクエストID 85番  
のジョブをキャンセルする)  
Request 85 is running, and has been signalled.
```

4. バッチリクエストの作成、投入及び結果の取り出し

具体的にバッチリクエストの作成、投入及び結果の取り出しについて、説明します。

①ホストowaniからバッチキューowaniへのバッチリクエストファイル例

ファイル名 nchl

投入

```
#!/bin/csh (cシェル使用)  
cd $HOME/f77  
f77 prog.f  
a.out
```

→

```
%qsub -eo nchl  
Request 0070.owani submi...
```

→

出力結果

ファイル名「nchl.o70」
に出力結果及びエラー
メッセージ 格納

②ホスト owani からパイプキュー - ACOS へのバッチリクエストファイル例

ファイル名 acosb

投入

1 8 16 (カラム)

```
$ JOB      userid$password
$ FRT77    V
$ PRMFL    S*,R,S,userid/prog
$ LINKER
$ RUN
$ ENDJOB
```

→

```
%qsub -eo -q acos acosb
Request 0075.owani submiti..
```

→

出力結果

ファイル名「acosb.o75」
に出力結果及びエラー
メッセージ 格納

③ホスト LANPO (ACOS) からパイプキュー - owani のバッチリクエストファイル例

ファイル名 ojob

投入・終了確認 (短時間ジョブの場合)

```
# IDENT=TEST,userid1,userid2
# OPTION=C(S/1),FORMSET=ACOSA4,h
#! /bin/csh
cd $HOME/f77
f77 -ZLP prog.f
a.out
cat prog.L
```

→

```
SYSTEM ?qsub -eo ojob
Request 0079.LANPO submiti..
SYSTEM ?JMON * (終了確認)
CRJE399 E Q0079 は見つかりま
CRJE447 E デマンドファイル内のQ0079
(正常終了)
```

→

出力結果の印刷

```
SYSTEM ?JOUT *  
FUNCTION ?DIREUONL  
SYSTEM ?
```

JOUTコマンドはACOSのバッチジョブ終了時によく使われるコマンドで出力結果をファイルに格納したり(COPYコマンド)、通信ソフト ETOS で利用している場合は端末へSPRINTコマンドを利用して画面で出力結果をみることができます。

④ バッチリクエストを端末キーから直接投入する方法

①～③のバッチリクエストはファイルから投入しましたが、owaniでは端末キーから直接投入することが可能です。次の例では出力結果はファイル「STDIN.o90」に格納されます。

```
%qsub -eo      (バッチリクエスト投入)  
# sample      (コメント行の入力)  
f77 test.f    (FORTRANの翻訳)  
~d           (完了 コントロールを押しながらDを押す)  
Request 0090.owani submitted to queue: owani.
```

⑤ バッチリクエストファイルにオプションを含ませる方法

qsub コマンドに沢山のオプションがある場合は、ファイルの先頭にオプションを埋め込む方が便利です。

```
#!/bin/csh  
#@$-a"23:30" (23:30に開始)  
#@$-q acos  (キュー-acosに投入)  
#@$         (埋め込みオプションの終了)  
echo "nqs test"
```

⑥ ACOSからのバッチリクエストファイルの作成方法

ACOSバッチの結果出力は、JOUT コマンドを利用する方が便利です。そのためにスクリプトの前に以下の2行を入れます。

```
# IDENT=TEST,userid1,userid2  
# OPTION=C(S/1),FORMSET=ACOSA4,h  
#!/bin/csh  
:
```

通常のスキプトの前に上記2行が必要です。userid1はowaniの利用者ID,userid2はACOSのIDです。OPTION=以下は出力用紙や複写数、DEMAND出力ファイルに格納するか等の指定です。上記の例はA4サイズのレーザープリンタ, DEMAND出力の指定です。B4サイズでDEMAND出力指定は「# OPTION=h」で可能です。

5. 注意事項

NQSシステムは快適なバッチ環境を提供しますが、マルチユーザを対象としているシステムであるため利用者一人々平等に利用できることが必要です。一人の利用者があまりにも多くの計算機の資源を使ってシステムが動かなくなったり、他の利用者が使えなくなってしまう。そのためNQSでは1プロセス等に多くの制限（CPU時間やメモリ容量等）を設けることができます。しかし、現在は一切の制限を付けていません。どの程度利用者がいてどのようなジョブを実行するのかわからないためです。将来的にはある程度の制限は必要になってくるかもしれません。その場合は予告無く制限することがありますのでご了承願います。