バッチリクエストによるジョブの実行方法（AOBA-A / AOBA-B）

情報部デジタルサービス支援課 スーパーコンピューティングサポートユニット

2020 年 10 月 1 日 初版　2026 年 2 月 1 日 第 4 版

1. はじめに

本センターの計算機は、バッチリクエストの方法で使用します。ジョブ管理システムは、NEC Network Queuing System V（以下、NQSV）を採用しています。

本書では、AOBA-A および AOBA-B を対象として、バッチリクエストによるジョブの実行方法について説明します。AOBA-S については、システム構成および利用方法が異なるため、本書とは別途用意しているマニュアルにてご案内しています。

2. バッチリクエストの概要

バッチリクエストは、バッチ処理で計算機にジョブの実行を依頼するリクエストのことです。ジョブとは、実行可能ファイル、プログラム、またはコマンドのことで、リクエストは 1 つまたは複数のジョブから構成されます。実行すべきジョブをシェルスクリプト（ジョブスクリプト）に記述し、NQSV に投入することにより実現されます。

図 1 に、バッチリクエストの実行の流れを示します。①は 4 章、②は 5 章、③は 6～8 章、④は 9 章で詳しく解説します。

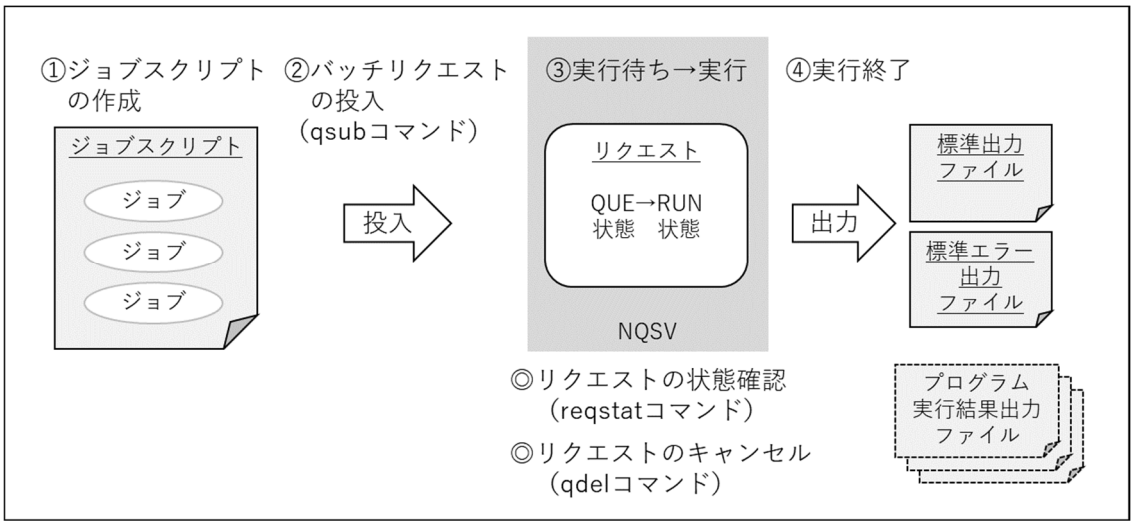

図 1　バッチリクエストの実行の流れ

①ジョブスクリプトの作成

プログラムの実行手続きを書いたテキストファイルを作成します。利用する計算資源（投入キュー名、利用 VE 数／ノード数、最大経過時間）等も記述します。

②バッチリクエストの投入、③実行待ち→実行

以下のコマンドでバッチリクエストを操作します。

qsub コマンド　　NQSV にバッチリクエストを投入する

reqstat コマンド　投入したリクエストの状態を表示する（状態確認）

qdel コマンド　　投入したリクエストをキャンセルする

④実行終了

リクエストの実行が終了すると標準出力ファイルと標準エラー出力ファイルが出力されます。終了したリクエストは reqstat コマンドの表示から消えます。

3. キュー構成（AOBA-A / AOBA-B）

リクエストの投入先（キュー）について説明します。

表 1、表 2 に、新システムのキュー構成を示します。ジョブスクリプトで、投入キュー名、利用 VE 数／ノード数、最大経過時間を指定します。

表 1　サブシステム AOBA-A のキュー構成

利用形態	キュー名	VE 数	実行形態（※）	最大経過時間 既定値／最大値	メモリサイズ
無料	sxf	1	1VE	1 時間／1 時間	48GB×VE 数
共有	sx	1	1VE	72 時間／720 時間	
		2～256	8VE 単位で確保 （VH を共用しない）		
占有	個別設定				

（VH：ベクトルホスト、VE：ベクトルエンジン）

（※）実行形態について

8VE 単位で確保 （VH を共用しない）	8VE 単位（1VH+8VE）で計算資源が確保され、投入したリクエストが他のリクエストと VH を共用しないで実行されます。他のリクエストと VH を共用しないため、演算時間のバラツキが少なくなります。
1VE 単位で確保 （VH を共用する）	1VE 単位で計算資源が確保され、投入したリクエストは他のリクエストと VH を共用して実行されます。指定した数の VE を確保しやすく、混雑時にも待ち時間が短くなります。
1VE	「VH を共用する」実行形態となります。

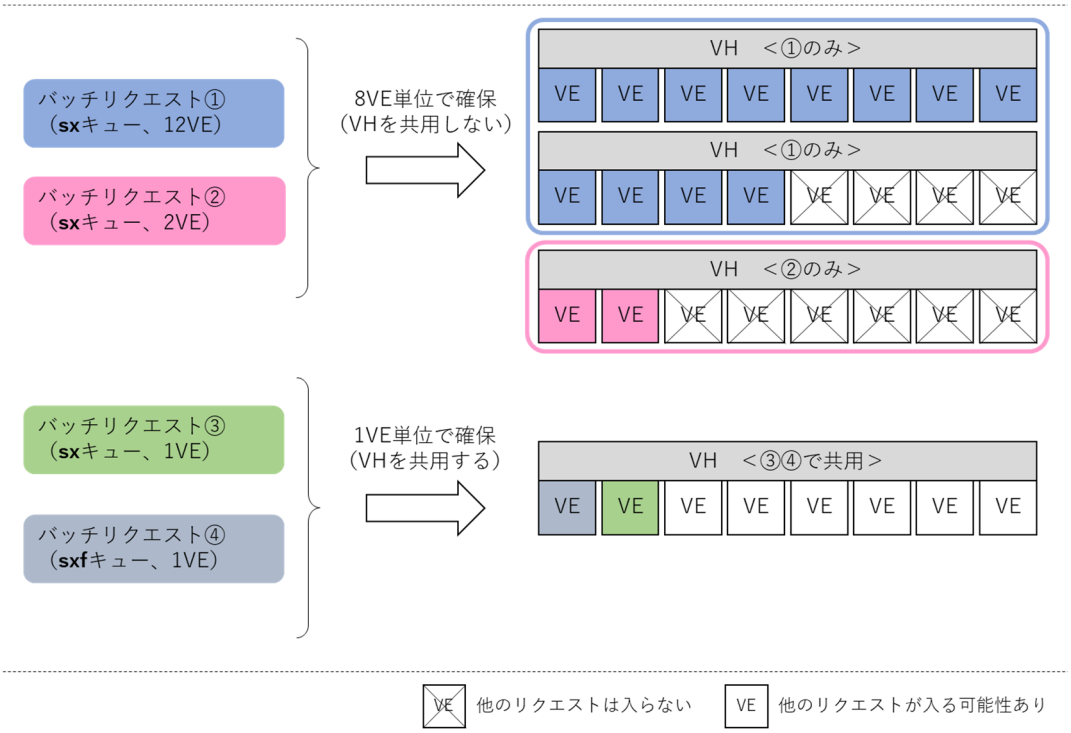

表 2　サブシステム AOBA-B のキュー構成

利用形態	キュー名	ノード数	最大経過時間 既定値／最大値	メモリサイズ
共有	lx	1～16	72 時間／720 時間	256GB×ノード数
占有	個別設定			

【重要】フロントエンドサーバでの実行について

フロントエンドサーバ上で、計算負荷の高いプログラムを実行しないでください。フロントエンドサーバに過度な負荷がかかることで、他の利用者への影響だけでなく、システム全体が正常に動作しなくなるおそれがあります。

なお、負荷の高いプロセスについては、管理者の判断により強制終了する場合があります。

4. バッチリクエストの作成

バッチリクエスト用のシェルスクリプトファイル（ジョブスクリプト）を作成します。通常のシェルスクリプトと同じように、テキストファイルに、任意のコマンドを組み合わせてプログラムの実行手続きを記述します。ジョブスクリプトを実行するシェルは sh スクリプト、csh スクリプト、どちらも使用できます。ファイル名は任意の名前をつけることができます。（以降の解説は sh スクリプトで記述し、ファイル名は run.sh とします）

ジョブスクリプトに記述する基本項目は、以下のとおりです。リスト 1 に、ジョブスクリプトの例を示します。

- 投入キュー名
- 利用 VE 数／ノード数
- 最大経過時間
- 作業ディレクトリへの移動コマンド
- プログラムの実行コマンド

他に、環境変数の指定やファイルの操作コマンド等があれば、適切な箇所に記述します。

リスト 1　AOBA-A 用ジョブスクリプトの例

```
（run.sh の記述内容）
#!/bin/sh
#PBS -q sx                      # 投入キュー名
#PBS --venode 1                 # 利用 VE 数
#PBS -l elapstim_req=2:00:00    # 最大経過時間
cd $PBS_O_WORKDIR               # 作業ディレクトリへの移動コマンド
./a.out                         # プログラムの実行コマンド
```

リスト 2　AOBA-B 用ジョブスクリプトの例

```
（run.sh の記述内容）
#!/bin/sh
#PBS -q lx                      # 投入キュー名
#PBS -b 1                       # 利用ノード数
#PBS -l elapstim_req=2:00:00    # 最大経過時間
cd $PBS_O_WORKDIR               # 作業ディレクトリへの移動コマンド
./a.out                         # プログラムの実行コマンド
```

投入時オプション（バッチリクエスト実行のための設定情報。qsub コマンドのオプションや環境変数の指定）は、最初のシェルコマンドが現れる前のコメント部分に、行の先頭に#PBS という文字列をつけて記述します。表 3 に、qsub コマンドの主なオプションを示します。

表 3　qsub コマンドの主なオプション（○印：指定必須）

AOBA-A	AOBA-B	オプション	機能
○	○	-q 投入キュー名	投入キュー名を指定(※1)
○	-	--venode 利用 VE 数	AOBA-A で実行する場合、利用 VE 数を指定(※1) (注意:このオプションは先頭のハイフンが 2 つ)
	○	-b 利用ノード数	AOBA-B で実行する場合、利用ノード数を指定(※1)
○	○	-l elapstim_req=時間	最大経過時間を指定 (秒数または時:分:秒 (hh:mm:ss 形式)) (省略時：既定値) (※1)(※2)
		-A プロジェクトコード	課金先のプロジェクトコードを指定 (省略時：デフォルトのプロジェクトコード)
		-N リクエスト名	リクエスト名を指定 (省略時：ジョブスクリプトファイル名)
		-o ファイル名	標準出力のファイル名を指定 (省略時：リクエスト投入時のディレクトリに「リクエスト名.o リクエスト ID」のファイル名で出力)
		-e ファイル名	標準エラー出力のファイル名を指定 (省略時：リクエスト投入時のディレクトリに「リクエスト名.e リクエスト ID」のファイル名で出力)
		-jo	標準エラー出力を標準出力と同じファイルに出力 (省略時：別々のファイルに出力)
		-m b	リクエストが実行開始したときにメール送信 (省略時：メール送信しない)
		-m e	リクエストが実行終了したときにメール送信 (省略時：メール送信しない)
		-M メールアドレス	メールの送信先を指定 (省略時：利用者番号@front.cc.tohoku.ac.jp)
		-r {y | n}	リクエストのリラン可否を指定 (※3) (y：許可、n：許可しない、省略時：y)

(※1) キュー名、利用可能な VE 数／ノード数、最大経過時間の既定値／最大値は、3 章の表 1、表 2 を参照してください。

（※2）経過時間とは、バッチリクエストが実行開始してから終了するまでの時間です。最大でどのぐらいの経過時間の確保が必要かを指定します。指定した最大経過時間が短いリクエストほど、計算資源が確保されやすく、実行待ちの時間が短くなります。ただし、指定した最大経過時間を超えると、実行は打ち切りとなり強制終了します。I/Oが高負荷の場合、経過時間が想定より長くなることがあります。必要十分な最大経過時間を指定してください。

（※3）リランとは、リクエストをはじめから実行しなおすことです。計算機のメンテナンスや障害発生時に、管理者にてリクエストをリランする場合があります。リランによりプログラムの実行に不都合が生じる場合（たとえば、実時間で時間を計測している場合）は、あらかじめ該当オプションを指定して投入してください。

5. バッチリクエストの投入

バッチリクエストの投入は qsub コマンドで行います。リスト 2 に、qsub コマンドの実行例を示します。run.sh というジョブスクリプトを投入する例です。

リスト 2　qsub コマンドの実行例

```
front$ qsub run.sh
```

バッチリクエストが正常に投入されると、システムからリスト 3 のようなメッセージが返ってきます。この例は、リクエスト ID は 1234.job が採番され、投入先は sx キュー、課金先はプロジェクトコード un0000 が指定されたことを示しています。リクエスト ID は、このあと、バッチリクエストの状態確認やキャンセルの際に必要になります。

バッチリクエストの投入に失敗した場合は、エラーメッセージが返ってきます。ジョブスクリプトの記述などに誤りがないか確認してください。

リスト 3　qsub コマンド実行時のシステムメッセージ例

```
front$ qsub run.sh
プロジェクトコード：un0000 にリクエストを投入します
Request 1234.job submitted to queue：sx
```

6. バッチリクエストの実行

投入されたリクエストは、実行待ちの列に並びます。リクエストの実行順序はバックフィルスケジューリングで制御されます。バックフィルスケジューリングでは、計算資源が確保できたリクエストから実行開始します。実行に必要な計算資源量（利用 VE 数またはノード数×最大経過時間）が少ないリクエストは、投入順序によらず、早く実行開始する場

合があります。

7. バッチリクエストの状態確認

バッチリクエストの状態確認は reqstat コマンドで行います。

reqstat コマンドを実行すると、実行待ち／実行中のリクエストの状態が表示されます。利用者自身のリクエストが全て表示されます。該当するリクエストがない場合は何も表示されません。

表 4 に reqstat コマンドの表示項目、表 5 にリクエストの主な状態を示します。

リクエストの実行開始予定時刻は、StartTime に表示されます。実行開始予定時刻は、他のリクエストの終了状況により随時更新されます。通常、更新前の時刻よりも遅くなることはありませんが、障害発生時などはその限りではありません。

表 4　reqstat コマンドの表示項目

表示項目	内容
RequestID	リクエスト ID
RequestName	リクエスト名
User	利用者番号
PJCode	課金先のプロジェクトコード
Que	実行キュー名
Node	投入時に指定した VE 数またはノード数
ElapseLimit	投入時に指定した最大経過時間
STT	リクエストの状態
StartTime	実行開始予定時刻（実行開始したら、実際に実行開始した時刻）
Memory	現在の使用メモリ量
ElapseTime	現在までの経過時間
NodeTime	現在までのノード時間（ElapseTime×Node）

表 5　リクエストの主な状態

表記	リクエストの状態
RUN	実行中
QUE	実行待ち
EXT	標準出力／標準エラー出力(リダイレクトも含む)の出力中（※）

（※） 以下のような場合、EXT 状態で長く停滞することがあります。他のリクエスト(他の利用者のリクエストも含む)の実行に影響がでる場合がありますので、十分ご留意ください。

- ファイル容量がクォータにかかり、標準出力／標準エラー出力を出力できない。
  空き容量が確保できるまで、EXT で停滞したままになります。ただちに、ファイルを整理して

容量を空けてください。容量が確保できれば、正常に出力され、リクエストは終了します。なお、クォータにかからないよう、リクエスト投入前に、十分な空き容量があることを確認してください。なお、ファイル容量を追加することも可能です。詳しくは以下をご参照ください。

ストレージシステムについて（ストレージ容量追加申請）：
https://www.ss.cc.tohoku.ac.jp/storage/

・ システム上で問題が発生しており、管理者にて対応中のため、そのままお待ちください。

8. バッチリクエストのキャンセル

投入したリクエストをキャンセルしたい場合は、qdel コマンドで行います。qdel コマンドを実行すると、実行中のプログラムは強制終了し、リクエストは削除されます。キャンセルは利用者自身で行うことができます。キャンセルできるリクエストは、利用者自身のリクエストのみです。

リスト 4 に、qdel コマンドの実行例を示します。1234.job というリクエストをキャンセルする例です。qdel に続けてキャンセルするリクエストのリクエスト ID を指定します。

リスト 4　qdel コマンドの実行例

```
front$ qdel 1234.job
```

システムがキャンセルを受け付けると、リスト 5 のようなメッセージを返します。

リスト 5　qdel コマンドの実行時のシステムメッセージ例

```
front$ qdel 1234.job
Request 1234.job was deleted.
```

9. バッチリクエストの終了

リクエストが終了すると、reqstat コマンドで表示されなくなります。そのリクエストの実行結果ファイルとして、リクエスト実行中に標準出力に出力された内容を格納した＜標準出力ファイル＞と、標準エラー出力に出力された内容を格納した＜標準エラー出力ファイル＞が作成されます。

投入時にオプション-o、-e を指定した場合はそのファイル名になります。指定しなかった場合は、以下のように命名されます。

<標準出力ファイル>　　リクエスト名.o リクエスト ID

<標準エラー出力ファイル>　リクエスト名.e リクエスト ID

なお、標準出力／標準エラー出力のファイルサイズの上限は 100MB です。ファイルサイズが上限値を超えた場合はプログラムが強制終了しますので、実行結果は、標準出力／標準エラー出力ではなく、ファイル名を指定して書き出すようにしてください。

10. マニュアル

各コマンドの詳しいオプションなどは、以下に公開されているマニュアル「NQSV 利用の手引 操作編」をご参照ください。なお、当センター独自の運用方法により、マニュアルに記載のとおりに動作しない場合もあります。

マニュアル：https://www.hpc.nec/documentation

11. おわりに

本稿では、バッチリクエストの投入と確認方法についてご紹介しました。センターの計算資源を効率的にご活用いただければ幸いです。ご不明な点、ご質問等ございましたら、お気軽にセンター（利用相談）までお問い合わせください。

利用相談：https://www.ss.cc.tohoku.ac.jp/consultation/