

近畿大学 生物理工学部様
SMP型計算システム

利用手引書

Version 1.3
2014年11月26日
富士通株式会社

改版履歴

版	発行年月日	改版理由
1.0	2012.06.12	初版作成
1.1	2012.07.11	正式登録版
1.2	2012.07.24	利用者説明会向けに資料を修正
1.3	2014.11.26	CPU 情報追加、ストレージ追加による修正

目次

はじめに	1
1. SMP 型計算システムの概要.....	2
1.1 システム構成.....	2
1.2 ソフトウェア構成	3
1.3 アカウントと認証方式	3
1.4 ネットワークアクセス	3
1.5 システムへのログイン(Windows 環境).....	4
1.6 システムへのログイン(UNIX 環境).....	4
1.7 ログイン環境.....	4
2. システム環境	6
2.1 利用ファイルシステム	6
2.2 コンパイル/リンクの概要.....	6
2.3 Intel コンパイラ	7
2.4 PGI コンパイラ.....	9
3. ジョブ実行.....	11
3.1 ジョブシステム概要.....	11
3.2 ジョブ実行リソース.....	11
3.3 ジョブ投入オプション	12
3.4 バッチジョブ投入(qsub コマンド).....	13
3.5 ジョブ状態表示(qstat コマンド)	15
3.6 ジョブキャンセル(qdel コマンド).....	17
3.7 ジョブ保留(qhold コマンド).....	17
3.8 ジョブ開放(qrls コマンド).....	17
4. MPI 実行	18
4.1 Intel コンパイラ による MPI プログラム実行	18
4.2 PGI コンパイラ による MPI プログラム実行.....	18
5. ご利用時に発生した問題について	19

はじめに

本利用者マニュアルは、近畿大学様が導入した SMP 型計算システムの利用方法について説明した資料です。システムを利用する方は、必ずお読みください。

本利用者マニュアルの内容は、不定期に更新いたします。

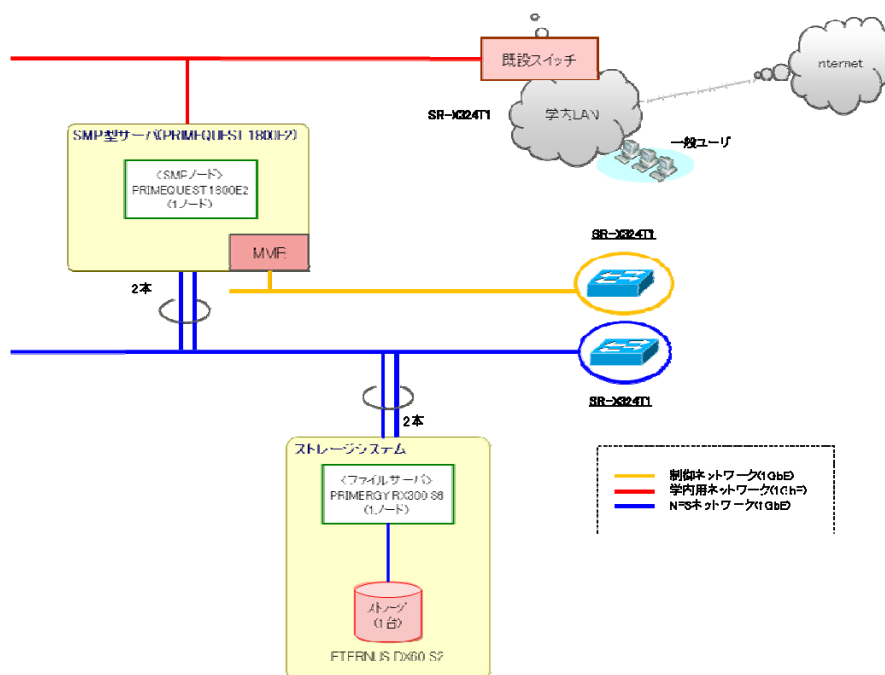
本書の一部、または全部を無断で複製、転載、再配布することを禁じます。

SMP 型計算システムの概要

1.1 システム構成およびハードウェア構成

SMP 型計算システムは、SMP 型サーバ、共有ファイルシステムから構成されるシステムです。

図 0-1 システム構成図



SMP 型計算サーバ(PRIMEQUEST 1400S2)は CPU として、Xeon プロセッサ E7-4870 (2.4GHz/10 コア/30MB) を 4 個搭載しており、ハイパースレッディングにより、システム全体として 80 論理コアを有しています。

総理論演算性能 768GFLOPS、総主記憶容量 1TByte を有しています。

ストレージ環境は、共有ファイルシステム(NFS)から構成されます。

共有ファイルシステムは、各ユーザーのホームディレクトリを格納するファイルシステムであり、全計算サーバおよびログインサーバから参照可能です。利用可能容量は約 68Tbyte(/home)と約 15Tbyte(/work)です。

1.2 ソフトウェア構成

システムのソフトウェア環境を以下に示します。

表 0-1 システムソフトウェア一覧

項目	SMP 型サーバ
OS	Red Hat Enterprise Linux 6.1
コンパイラ	【Intel コンパイラ】 Fortran コンパイラ C/C++ コンパイラ 【PGI コンパイラ】 Fortran コンパイラ C/C++ コンパイラ
ライブラリ	インテル・マス・カーネル・ライブラリ (IMKL) AMD Core Math Library (ACML)
アプリケーション	Gaussian09 Discovery Studio MATLAB Mathmatica
ジョブ管理システム	TORQUE

SMP 型サーバは、SSH によるログイン後、コマンドの対話的実行が可能であり、主にプログラムの作成・編集、実行モジュールのコンパイル/リンク、ジョブ投入を行います。

1.3 アカウントと認証方式

システムへのアクセスに使用するユーザ名は、申込み時に通知される利用者番号(ユーザ名)です。ログインサーバへのアクセスは SSH(version2)をご利用ください。

1.4 ネットワークアクセス

SMP 型サーバは、外部からのアクセスが可能です。ご利用端末から ssh で **smp01.waka.kindai.ac.jp** にアクセスします。接続情報は以下のとおりです。

表 0-2 アクセス先一覧

ホスト名(FQDN)	サービス	アクセス用途
smp01.waka.kindai.ac.jp	ssh	<ul style="list-style-type: none">プログラムの作成/編集/コンパイル/リンク/実行有償/無償アプリケーションの実行

1.5 システムへのログイン(Windows 環境)

Windows で使用できるターミナルソフトには PuTTY や Tera Term などがあります。
PuTTY、TeraTerm は、以下のサイトからダウンロードすることができます。

PuTTY: <http://www.chiark.greenend.org.uk/~sgtatham/putty/>

Tera Term: <http://sourceforge.jp/projects/ttssh2/>

ログアウトは、ターミナルソフト上で、"exit" もしくは "logout" と入力します。

1.6 システムへのログイン(UNIX 環境)

UNIX 系 PC や WS や Windows 環境で Cygwin を使って SMP 型計算システムへログインする場合は、ssh サービスを利用します。

```
% ssh -l username smp01.waka.kindai.ac.jp
username@smp01's password: ← パスワードを入力

[username@smp01 :~]
```

ログアウトは、ターミナルソフト上で、"exit" もしくは "logout" と入力します。

1.7 ログイン環境

システムは、ログインシェルとして bash が登録されています。ログインシェルを変更する場合は、chsh コマンドを実行してください。chsh の実行方法は「1.7.1 ログインシェル変更」を参照してください。

なお、ログイン時にシステムを利用するための環境設定が自動で設定されます。環境変数 PATH にパスを追加する際には、PATH の最後に追加してください。PATH の先頭に追加した場合、システムを正常に使用できなくなる恐れがあります。

1.7.1 ログインシェル変更

chsh コマンドを利用して、ログインシェルを変更します。変更したログインシェルは次回ログイン時から有効となります。

```
$ chsh [option] <new_shell>
```

表 0-3 chsh コマンドオプション一覧

オプション	説明
-l	利用可能なシェル一覧を表示する [/bin/bash, /bin/tcsh]
-s	利用可能なシェルを変更する

例) ログインシェルを tcsh に変更する。

```
[username@smp01:~]$ chsh -s /bin/tcsh
```

Enter Password:

← ユーザパスワードを入力します

```
modifying entry
```

1.7.2 メール転送設定

ジョブ終了時などメールにて通知を受けることができます。通知を受けるメールアドレスは、ユーザ名@smp01 に設定されています。希望するメールアドレスで受信するためには、メール転送の設定(.forward)が必要です。メール転送の設定は、以下の通りです。

例) foo@foo.com に転送する場合

```
[username@smp01:~]$ vi .forward
```

```
foo@foo.com
```


2. システム環境

2.1 利用ファイルシステム

システムが提供するファイルシステム領域は以下のとおりです。

表 2-1 利用可能ファイル領域一覧

領域	領域名	サイズ	備考
共有ファイルシステム	/home	68TB	ホーム領域
共有ファイルシステム	/work	15.7TB	ホーム領域

2.1.1 共有ファイルシステム

共有ファイルシステム(NFS)は ext3 で構成され、ユーザのホーム領域として提供されます。

ホーム領域の使用量は Quota にて 1 ユーザーあたり 500MB(softlimit は 450MB)に制限されています。

共有ファイルシステムは SMP 型サーバから参照可能であり、主な使用目的は以下のとおりです。

- ソースプログラム/オブジェクトファイル/実行モジュールファイルの格納
- データの格納

2.2 コンパイル/リンクの概要

コンパイル/リンクの書式とコマンド一覧は以下のとおりです。

コマンド [option] sourcefile [...]
--

表 2-4 コンパイル/リンクコマンド一覧

コンパイラ	言語処理系	コンパイル/リンクコマンド	備考
Intel コンパイラ	Fortran90	ifort	Intel コンパイラと PGI コンパイラを同時に使用することはできません。一方の環境を使用していて、もう一方の環境に切り替えたい場合はログアウト後、再度ログインして使用してください。
	C	icc	
	C++	icpc	
PGI コンパイラ	Fortran90	pgf77	
	C	pgcc	
	C++	pgCC	

2.3 Intel コンパイラ

Intel コンパイラの利用方法を示します。

2.3.1 環境変数の設定

Intel コンパイラ用に環境変数を整えます。

- bash の場合

```
[username@smp01~]# cd /usr/local/bin
[username@smp01~]# source intel_compilervars.sh intel64
[username@smp01~]# source intel_mpivars.sh 注1
```

注 1: MPI を実行する場合の環境設定

- cshell の場合

```
[username@smp01~]# cd /usr/local/bin
[username@smp01~]# source intel_compilervars.csh intel64
[username@smp01~]# source intel_mpivars.csh 注1
```

注 1: MPI を実行する場合の環境設定

2.3.2 Fortran プログラムのコンパイル

Fortran プログラムのコンパイルは下記コマンドにより行います。またコンパイラオプションについての詳細は下記 URL よりオンラインマニュアル「インテル Fortran コンパイラー XE 12.0 ユーザ・リファレンス・ガイド」の「コンパイラー・オプション」を参照してください。

<http://www.xlsoft.com/jp/products/intel/compilers/fcl/12/ug/index.htm>

- 逐次プログラムの場合

```
[username@smp01~]# ifort <options> sourcefile
```

- OpenMP プログラムの場合

```
[username@smp01~]# ifort -openmp <options> sourcefile
```

- MPI プログラムの場合

```
[username@smp01~]# mpifc <options> sourcefile
```

2.3.3 Cプログラムのコンパイル

Cプログラムのコンパイルは下記コマンドにより行います。またコンパイラオプションについての詳細は下記 URL よりオンラインマニュアル「インテル C++ コンパイラー XE 12.0 ユーザ・リファレンス・ガイド」の「コンパイラー・オプション」を参照してください。

<http://www.xlsoft.com/jp/products/intel/compilers/ccl/12/ug/index.htm>

- 逐次プログラムの場合

```
[username@smp01~]# icc <options> sourcefile
```

- OpenMP プログラムの場合

```
[username@smp01~]# icc -openmp <options> sourcefile
```

- MPI プログラムの場合

```
[username@smp01~]# mpiicc <options> sourcefile
```

2.3.4 C++プログラムのコンパイル

C++プログラムのコンパイルは下記コマンドにより行います。またコンパイラオプションについての詳細は下記 URL よりオンラインマニュアル「インテル C++ コンパイラー XE 12.0 ユーザ・リファレンス・ガイド」の「コンパイラー・オプション」を参照してください。

<http://www.xlsoft.com/jp/products/intel/compilers/ccl/12/ug/index.htm>

- 逐次プログラムの場合

```
[username@smp01~]# icpc <options> sourcefile
```

- OpenMP プログラムの場合

```
[username@smp01~]# icpc -openmp <options> sourcefile
```

- MPI プログラムの場合

```
[username@smp01~]# mpiicpc <options> sourcefile
```

2.4 PGI コンパイラ

PGI コンパイラの利用方法を示します。

2.4.1 環境変数の設定

PGI コンパイラ用に環境変数を整えます。

- bash の場合

```
[username@smp01~]# cd /usr/local/bin  
[username@smp01~]# source pgi_compilervars.sh
```

- cshell の場合

```
[username@smp01~]# cd /usr/local/bin  
[username@smp01~]# source pgi_compilervars.csh
```

2.4.2 Fortran プログラムのコンパイル

Fortran プログラムのコンパイルは下記コマンドにより行います。またコンパイラオプションについての詳細は下記 URL よりオンラインマニュアル「PGI コンパイラ使用ガイド」の「2 コンパイル・オプションの使用例」を参照してください。

http://www.softek.co.jp/SPG/Pgi/public/doc/PGI_softek_ug.pdf

- 逐次プログラムの場合

```
[username@smp01~]# pgf77 <options> sourcefile
```

- OpenMP プログラムの場合

```
[username@smp01~]# pgf77 -openmp <options> sourcefile
```

- MPI プログラムの場合

```
[username@smp01~]# pgf77 -Mmpi=mpich2 <options> sourcefile
```

2.4.3 Cプログラムのコンパイル

Cプログラムのコンパイルは下記コマンドにより行います。またコンパイラオプションについての詳細は下記 URL よりオンラインマニュアル「PGI コンパイラ使用ガイド」の「2 コンパイル・オプションの使用例」を参照してください。

http://www.softek.co.jp/SPG/Pgi/public/doc/PGI_softek_ug.pdf

- 逐次プログラムの場合

```
[username@smp01~]# pgcc <options> sourcefile
```

- OpenMP プログラムの場合

```
[username@smp01~]# pgcc -openmp <options> sourcefile
```

- MPI プログラムの場合

```
[username@smp01~]# pgcc -Mmpi=mpich2 <options> sourcefile
```

2.4.4 C++プログラムのコンパイル

C++プログラムのコンパイルは下記コマンドにより行います。またコンパイラオプションについての詳細は下記 URL よりオンラインマニュアル「PGI コンパイラ使用ガイド」の「2 コンパイル・オプションの使用例」を参照してください。

http://www.softek.co.jp/SPG/Pgi/public/doc/PGI_softek_ug.pdf

- 逐次プログラムの場合

```
[username@smp01~]# pgCC <options> sourcefile
```

- OpenMP プログラムの場合

```
[username@smp01~]# pgCC -openmp <options> sourcefile
```

- MPI プログラムの場合

```
[username@smp01~]# pgCC -Mmpi=mpich2 <options> sourcefile
```

3. ジョブ実行

3.1 ジョブシステム概要

システムの全ジョブは、ジョブ管理システム(TORQUE)により実行が制御されます。ユーザはジョブ開始時に必要な CPU 数、経過時間などを指定し、ジョブ管理システム(TORQUE)に対してジョブ実行を指示します。システムで利用可能なジョブはバッチジョブです(表 3-1 ジョブの種類)。バッチジョブは、CPU やメモリなどの計算に必要なリソースが排他的に割り当てられます。

表 3-1 ジョブの種類

ジョブ形式	計算 CPU コア数	用途
バッチジョブ	64	バッチ形式でジョブを実行する。

ユーザがジョブ操作に用いるコマンドは、以下のとおりです。

表 3-2 バッチジョブ操作コマンド一覧

機能	コマンド名
ジョブ投入	qsub
ジョブ参照	qstat
ジョブ削除	qdel
ジョブ保留	qhold
ジョブ解除	qrls
ジョブ開始	qrun

3.2 ジョブ実行リソース

3.2.1 バッチジョブ用キュー

ジョブ管理システム(TORQUE)は、キューという単位でジョブの実行を管理します。バッチジョブを投入する場合、ユーザはジョブを実行するためのキューを指定します。指定可能なキューは以下のとおりです。

表 3-3 キュー

キュー名	最大割当て CPU 数	最大割当てメモリ数	実行時間制限	備考
normal	32	0.5TB	無制限	通常利用。デフォルト割当て CPU 数 16。
large	64	1.4TB	無制限	ユーザグループ「large」に所属するユーザのみ利用可。

3.3 ジョブ投入オプション

ジョブ投入時は、ジョブの実行に応じて、オプションを指定します。

下記は qsub の主なオプションです。"-q"オプションでキューを指定しない場合は「表 3-3 キュー」の「normal」で実行されます。

表 3-4 ジョブ投入オプション

オプション名	引数	説明
-a	[YYYYMMDDhhmm]	ジョブの開始時刻を指定
-q	文字列	投入するキューを指定
-N	文字列	ジョブ名を指定
-u	ユーザ名	ジョブを実行するユーザを指定
-l	リソースリスト	-
-A	文字列	アカウント名
-P	数値	-1024 から+1023 までの数値。ジョブのプライオリティ。指定しない場合には0。
-e	ファイル名	標準エラー出力用ファイル名
-o	ファイル名	標準出力用ファイル名
-j	なし	標準エラー出力を標準出力にマージして出力
-m	A	ジョブがアボートした場合にメールで通知【デフォルト】
	B	ジョブが開始した場合にメールで通知
	E	ジョブが終了した場合にメールで通知
-M	user1,user2,user3,...	メールの通知先
-t	数値[,数値,...]	投入するジョブアレイの数を指定
-f	なし	mom に異常があった場合にもジョブを継続実行
-h	なし	ジョブ投入後、HOLD 状態に移行
-I	なし	インタラクティブジョブとしてサブミット。ジョブが実行されると標準入出力は qsub を実行したターミナルセッションに接続される。
-d	Path	ワーキングディレクトリ(PBS_O_INITDIR)を指定。デフォルトはホームディレクトリ
-w	Path	ワーキングディレクトリ(PBS_O_WORKDIR)を指定。デフォルトは qsub 時のディレクトリ
-T	ファイル名	開始スクリプト(prologue.xxxxx)もしくは終了スクリプト(epilogue.xxxxx)を指定
-S	シェル	スクリプトを実行するシェルを指定
-r	[y/n]	ジョブが再実行可能かどうかを指定
-v	環境変数	ジョブに引き継ぐ環境変数のリスト
-V	なし	すべての環境変数をジョブに継続
-z	なし	qsub 実行後にジョブ ID 等の文字出力しない

3.3.1 ジョブ資源オプション

ジョブが利用する資源に関する主要オプションは以下のとおりです。"-l"オプションに続けて利用資源を指定します。その他の"-l" オプションについての詳細は下記 URL よりオンラインマニュアル「TORQUE Administrator Guide」の「2.1.2 Requesting resources」を参照してください。

http://www.adaptivecomputing.com/download/resources/docs/torque/3-0-5/pdf/TORQUE_Administrator's_Guide.pdf

表 3-5 ジョブ資源オプション

オプション名		説明
-l	nodes=<n>:ppn=<m>	使用するノード数 n と、そのノードで使う CPU(コア) 数 m を指定する。注1
	walltime=<h/m/s>	このジョブの実行時間を指定する。指定時間は、h 時:m 分:s 秒である。
	mem=<n>mb	使用するメモリ量 n を指定する。

注1: SMP 型計算システムは 1 ノードで構成されているため、n=1 を指定する。

3.4 バッチジョブ投入(qsub コマンド)

バッチジョブを実行するためには、実行するプログラムとは別に「ジョブスクリプト」を作成し、利用するジョブクラス、実行時間、CPU 数などの資源や実行形式を記載したオプションを記述した上で、実行するプログラムを記載します。ユーザはジョブスクリプトを qsub コマンドで投入し、実行を待ちます。投入されたジョブはスケジューラにより自動で実行開始、完了が制御されます。

3.4.1 バッチジョブ投入

バッチジョブを投入する場合、qsub コマンドの引数にバッチジョブとして実行するスクリプトファイルを指定します。

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

- スクリプトファイルを指定しない場合、標準入力から実行命令を読み込みます。
- ジョブ投入オプションは、スクリプトファイル内にディレクティブを用いて記載可能です。
- ジョブ投入が完了後、ジョブに対して識別用 ID(ジョブ ID)が割り当てられます。

例)バッチジョブ投入例

```
[username@smp01:~] qsub go.sh          バッチジョブの投入
[username@smp01:~] qstat -a          バッチジョブの確認
smp01:
                                     Req'd Req'd Elap
Job ID  Username  Queue  Jobname  SessID  NDS  TSK Memory Time S Time
-----
47.smp01 username  normal  STDIN    50196   --   16   --   --   R 00:00
```


3.4.2 バッチジョブの終了確認

バッチジョブ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ジョブスケジューラの終了処理としてジョブ投入ディレクトリに出力されます。

標準出力ファイルにはジョブ実行中の標準出力、標準エラー出力ファイルにはジョブ実行中のエラーメッセージが出力されます。

ジョブ名.oxxxxx ... 標準出力ファイル

ジョブ名.exxxxx ... 標準エラー出力ファイル

(xxxxxx はジョブ投入時に表示されるジョブのジョブ ID)

3.4.3 バッチジョブスクリプト記述

バッチジョブを投入するためには、vi コマンドや emacs コマンドにてスクリプトを作成します。

- (1) 先頭行は "#!" に続けて、ジョブで利用するシェル名を指定してください。

[記述例]

<code>#!/bin/sh</code>	Bourne Shell を利用
------------------------	------------------

- (2) ジョブ投入オプションは qsub コマンドのオプションまたはスクリプト中に"#PBS"を用いて指定します。

[記述例]

<code>#PBS -l nodes=1:ppn=8</code>	CPU コア数を指定
<code>#PBS -l walltime=0/5/0</code>	経過時間を指定
<code>#PBS -j oe</code>	エラー出力と標準出力を 1つのファイルにまとめる

- (3) ジョブ投入オプションに続けて、実行時のオプションと、実行するプログラムを指定します。

[記述例]

<code>mpirun -np 4 ./a.out</code>	プログラムを実行
-----------------------------------	----------

3.5 ジョブ状態表示(qstat コマンド)

投入したジョブ状態やリソース情報を確認する場合、pjstat コマンドを使用します。

```
qstat [option] [JOBID[JOBID...]]
```

表 3-6 qstat コマンドオプション一覧

オプション説明	内容
-q	システムの全てのキューの状況を表示
-a	システムの全てのジョブの状況を表示
-s	全てのジョブをステータスコメント付きで表示
-r	実行中の全てのジョブを表示
-B	PBS Server のサマリ情報を表示する
-Q	全てのキューのリミット値を表示する
-au {userid}	指定したユーザのジョブを表示する
-f {JobID}	指定したジョブの詳細を表示する
-Qf {queue}	指定したキューの詳細を表示する

3.5.1 ジョブ状態表示

qstat コマンドを実行すると、現在実行中もしくは実行待ちのジョブ状態を表示します。

```
[username@l01:~] qstat -a
smp01:
                                     Req'd  Req'd  Elap
Job ID  Username Queue   Jobname  SessID NDS  TSK Memory Time  S Time
-----
7.smp01  fjse01  large   STDIN    309224 --   32  --   --  R  00:00
8.smp01  fjse01  normal  STDIN    309348 --   16  --   --  R  00:00
9.smp01  fjse01  normal  STDIN    309358 --   16  --   --  R  00:00
10.smp01 fjse01  normal  STDIN    309378 --   16  --   --  Q  --
```

表 3-7 ジョブ情報の表示項目

項目	説明
Job_ID	ジョブ ID と実行 Server 名
Username	ユーザ名
Queue	キューネーム
jobname	スクリプト名もしくは指定のジョブネーム
SessID	セッション ID
NDS	要求されたノードの数
TSK	同時タスク(または CPU)の数

ELAPSE	ジョブの経過時間(実行中でないジョブは"---:---:---)
TOKEN	使用トークン時間
Req'd Memory	要求されたメモリの量
Req'd Time	要求された経過時間
S	ジョブの現在のステータス
Elap Time	現在のジョブ状態の経過時間

3.5.2 キューの詳細情報の表示(-B -f オプション)

現在設定されている詳細な Queue の状態またはサーバーの状態を確認するためには qstat コマンドのオプションに「-B -f」をつけます。

```
[username@l01:~] qstat -B -f
Server: smp01

  server_state = Active
  scheduling = True
  total_jobs = 0
  state_count = Transit:0 Queued:0 Held:0 Waiting:0 Running:0 Exiting:0
  acl_hosts = smp01
  managers = root@smp01
  operators = root@smp01
  default_queue = normal
  log_events = 511
  mail_from = adm
  resources_assigned.nodect = 0
  scheduler_iteration = 600
  node_check_rate = 150
  tcp_timeout = 6
  mom_job_sync = True
  pbs_version = 3.0.4
  keep_completed = 0
  next_job_number = 498
  net_counter = 1 0 0
```

3.6 ジョブキャンセル(qdel コマンド)

投入済みのジョブをキャンセルする場合、qdel コマンドを実行します。

```
qdel [JOBID [JOBID...]]
```

ジョブのジョブ ID を pjqdel の引数に指定します。qstat でジョブがキャンセルされたことを確認してください。

```
[username@smp01:~] qdel 1234  
[username@smp01:~] qstat -a  
[username@smp01:~]
```

3.7 ジョブ保留(qhold コマンド)

ジョブを保留状態(実行不可)にする場合は、qhold コマンドを実行します。
行を保留する場合、pjhold コマンドを指定します。

```
qhold [JOBID [JOBID...]]
```

ジョブのジョブ ID を qhold の引数に指定します。

```
[username@smp01:~] qhold 1234
```

3.8 ジョブ開放(qrls コマンド)

保留されたジョブを解除する場合、pjrls コマンドを指定します。

```
qrls [JOBID [JOBID...]]
```

ジョブのジョブ ID を pjrls の引数に指定します。

```
[username@smp01:~] qrls 1234
```

4. MPI 実行

4.1 Intel コンパイラ による MPI プログラム実行

4.1.1 mpirun

MPI ライブラリを付与した実行モジュールを実行するために、mpirun コマンドを利用します。

```
mpirun [options] 実行モジュール
```

オプションについての詳細は下記 URL よりオンラインマニュアル「Intel MPI Library for Linux OS」の「2.2 Simplified Job Startup Command」を参照してください。

http://software.intel.com/sites/products/documentation/hpc/mpi/linux/reference_manual.pdf

4.2 PGI コンパイラ による MPI プログラム実行

4.2.1 mpirun

MPICH-1 ライブラリを付与した実行モジュールを実行するために、mpirun コマンドを利用します。

```
mpirun [options] 実行モジュール
```

表 4-3 オプション一覧

オプション名	説明
-np <proc_num>	MPI プログラムのプロセス数を指定

その他オプションについての詳細は下記 URL よりオンラインマニュアル「PGI コンパイラ使用ガイド」の「2.8 MPI プログラムを開発時に使用するオプション」を参照してください。

http://www.softtek.co.jp/SPG/Pgi/public/doc/PGI_softek_ug.pdf

4.2.2 mpiexec

MPICH-2 ライブラリを付与した実行モジュールを実行するために、mpiexec コマンドを利用します。

mpiexec [options] 実行モジュール

表 4-4 オプション一覧

オプション名	説明
-n <proc_num>	MPI プログラムのプロセス数を指定

オプションについての詳細は下記 URL よりオンラインマニュアル「PGI コンパイラ使用ガイド」の「2.8 MPI プログラムを開発時に使用するオプション」を参照してください。

http://www.softek.co.jp/SPG/Pgi/public/doc/PGI_softek_ug.pdf

5. ご利用時に発生した問題について

ご利用時のプログラムの実行異常・翻訳異常、システムの異常が発生した場合は、下記までご連絡いただきますよう、よろしくお願いいたします。

hpc@ml.waka.kindai.ac.jp (生物理工学部 HPC 管理者グループ 宛)

以上