［大規模科学計算システムの利用法］

並列コンピュータ Express5800の利用法

情報部情報基盤課　共同研究支援係　共同利用支援係
サイバーサイエンスセンター　スーパーコンピューティング研究部

はじめに

並列コンピュータ Express5800 は、最大 32 並列による並列処理やベクトル化に不向きなプログラムの高速な演算が可能であり、また Gaussian 等アプリケーションプログラムも、より高速化されています。バッチ処理においては、ジョブ管理システム NQSⅡを使用しています。この資料では、Express5800 の利用法について説明します。

システム構成

システム構成は、既設のシステムを含め図1のようになっています。

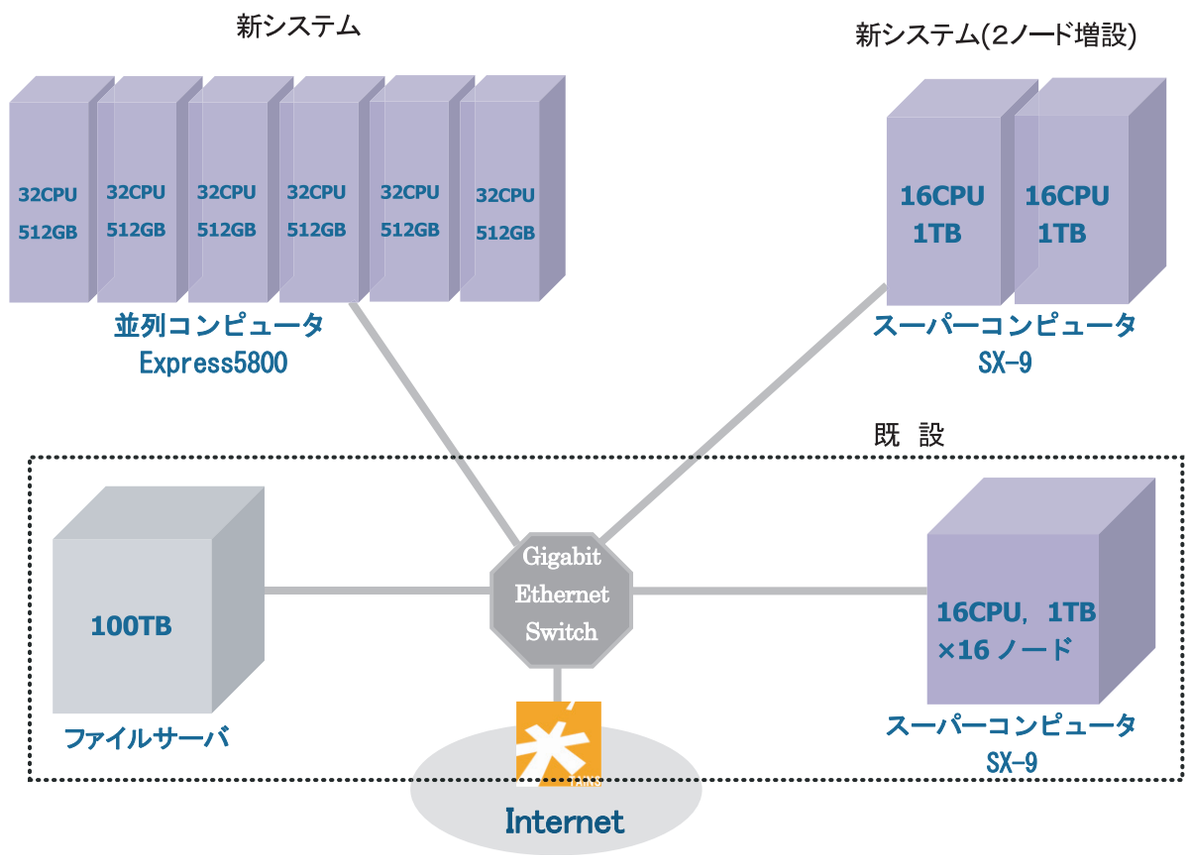

図1　システム構成

並列コンピュータ Express5800

並列コンピュータ Express5800 の1つのノードは、Intel 最新の 64 ビットプロセッサを 32 台と 512GB の主記憶装置を搭載しています。1CPU あたり 9.06GFLOPS の計算能力を有しますので、ノードあたりの最大演算性能は 289.92GFLOPS となります。ベクトル演算に不向きなプログラムの高速な実行が可能です。また、スーパーコンピュータ SX-9 のフロントエンドサーバとしての役割も担っています。

スーパーコンピュータ SX-9

スーパーコンピュータ SX-9 は、既設のシステムに 2 ノード増設し、合計 18 ノードで構成されます。並列処理はノード内 16 並列まで可能で、自動並列化、OpenMP、MPI を使った並列処理が実行できます。複数のノードを使用した並列処理は、MPI プログラムの利用により最大 64 並列まで実行可能です。

ログイン

並列コンピュータ Express5800 にログインします。ログインは、SSH(Secure SHell)を利用します(リスト1)。はじめて接続する場合、コマンド入力後

”Are you sure you want to continue connecting (yes/no)?”

と問い合わせがありますので yes を入力してからパスワードを入力します。

リスト1　並列コンピュータへのログイン

```
yourhost$ ssh  gen.isc.tohoku.ac.jp -l 利用者番号
Password: パスワード

[利用者番号@gen ~]$
```

Windows からログインする場合は、Putty 等 SSH 対応のリモート接続ソフトをご利用ください。

センターに利用登録すると、図１すべてのサーバが利用可能となります。

ログイン名とパスワードは各サーバで共通です。ログイン名は利用者番号を用い、パスワードは初期パスワードが設定されていますので yppasswd コマンドで速やかに変更してください(リスト2)。また、パスワードはセキュリティ保護のためときどき変更することをお奨めします。利用登録時のログインシェルは csh を設定しています。tcsh 等に変更したい場合は ypchsh コマンドをご利用後、再ログインしてください(リスト3)。

ホームディレクトリはファイルサーバの「/uhome/利用者番号」となります。NFS(Network File System)によるファイル管理を行っていますので、スーパーコンピュータ、並列コンピュータから共通に利用できます。

リスト2　パスワードの変更

```
gen00$ yppasswd
Changing password for x2xxx9.
Changing password x2xxx9
(current)UNIX Password: 現在のパスワード
New UNIX password: 新しいパスワード
Retype new UNIX password: 新しいパスワード（再度）
Password: all authentication tokens updated successfully.
```

リスト3　ログインシェルの変更（tcsh への変更例）

```
gen00$ ypchsh
Changing NIS account information for x2xxx9 on gen00.
Please enter Password: パスワード

Changing login shell for x2xxx9 on gen00.
To accept the default, simply press return.
Login Shell [/bin/csh]: /bin/tcsh
The login shell has benn changed on kanri-ux.
```

並列コンピュータの日本語環境は、UTF-8 となっております。日本語を表示させる場合には、接続ソフトウェアの文字コードを UTF-8 としてください。

プログラミング言語、ライブラリ

プログラミング言語および科学技術計算用ライブラリとして表1に示すものが利用できます。なお、ASL はコンパイル時のライブラリ指定(-l オプション)は不要です。

表1　プログラミング言語およびライブラリ

Fortran95	ISO/IEC 1539-1:1997,自動並列化, OpenMP
C++	ISO/IEC 14882:1998,自動並列化, OpenMP
MPI	並列処理ライブラリ
ASL	Fortran95 用科学技術計算ライブラリ
Math Kernel Library	数値演算ライブラリ

ファイルエディット

ソースファイルは、並列コンピュータの emacs エディタまたは vi エディタで作成します。研究室等のパソコンにあるソースファイルを利用するには、gen.isc.tohoku.ac.jp にファイル転送してください。送り元のホストが Windows の場合、転送モードの設定を”ASCII”にすることで適切な改行コードで転送できます。転送手順につきましては、Web ページ（http://www.ss.isc.tohoku.ac.jp/service/USE/FTP/index.html）をご参考ください。

コンパイル

●Fortran （自動並列・OpenMP）

【形式】 **f95 オプション ソースファイル名**

主なオプション

-parallel	自動並列化機能を利用する。
-openmp	OpenMP を利用する。
-O0	最適化を無効にする。
-O1	最適化を行うが、コードサイズが増える最適化は行わない。
-O2 または -O	一般的な最適化を行う。（規定値）

-O3	高度の最適化(プリフェッチ、スカラリプレスメント、ループ変換等)を行う。
-ip	インライン展開を行う。
-c	コンパイルのみ行う。(リンクはしない)
-o	実行可能形式のオブジェクトファイルの名前を指定する。省略時は a.out になる。
-w90	非標準 Fortran 機能に関する警告メッセージを抑止する。
-r8	精度の自動拡張を行う。(倍精度化)
-help	オプションの種類と説明を表示する。

ソースファイル名

Fortran のソースプログラムファイル名を指定します。複数のファイルを指定するときは、空白で区切ります。
ソースファイル名には、サフィックス.f90 か.F90(自由形式)、または.fか.F(固定形式)が必要です。

●Fortran (MPI)

【形式】 **mpif95 オプション MPI ソースファイル名**

主なオプション

-O0	最適化を無効にする。
-O1	最適化を行うが、コードサイズが増える最適化は行わない。
-O2 または -O	一般的な最適化を行う。(規定値)
-O3	高度の最適化(プリフェッチ、スカラリプレスメント、ループ変換等)を行う。
-ip	インライン展開を行う。
-c	コンパイルのみ行う。(リンクはしない)
-o	実行可能形式のオブジェクトファイルの名前を指定する。省略時は a.out になる。
-w90	非標準 Fortran 機能に関する警告メッセージを抑止する。
-r8	精度の自動拡張を行う。(倍精度化)
-help	オプションの種類と説明を表示する。

ソースファイル名

Fortran のソースプログラムファイル名を指定します。複数のファイルを指定するときは、空白で区切ります。
ソースファイル名には、サフィックス.f90 か.F90(自由形式)、または.fか.F(固定形式)が必要です。

●C/C++ (自動並列・OpenMP)

【形式】 **cc オプション ソースファイル名**

主なオプション

-parallel	自動並列化機能を利用する。
-openmp	OpenMP を利用する。
-O0	最適化を無効にする。

-O1	最適化を行うが、コードサイズが増える最適化は行わない。
-O2 または -O	一般的な最適化を行う。(規定値)
-O3	高度の最適化(プリフェッチ、スカラリプレスメント、ループ変換等)を行う。
-ip	インライン展開を行う。
-c	コンパイルのみ行う。(リンクはしない)
-o	実行可能形式のオブジェクトファイルの名前を指定する。省略時は a.out になる。
-help	オプションの種類と説明を表示する。

ソースファイル名

C/C++のソースプログラムファイル名を指定します。複数のファイルを指定するときは、空白で区切ります。
ソースファイル名にはサフィックス .c 、C++プログラムのソースファイル名にはサフィックス .cc または .C が必要です。

●C/C++ (MPI)

【形式】 **mpicc オプション ソースファイル名**

主なオプション

-O0	最適化を無効にする。
-O1	最適化を行うが、コードサイズが増える最適化は行わない。
-O2 または -O	一般的な最適化を行う。(規定値)
-O3	高度の最適化(プリフェッチ、スカラリプレスメント、ループ変換等)を行う。
-ip	インライン展開を行う。
-c	コンパイルのみ行う。(リンクはしない)
-o	実行可能形式のオブジェクトファイルの名前を指定する。省略時は a.out になる。
-help	オプションの種類と説明を表示する。

ソースファイル名

C/C++のソースプログラムファイル名を指定します。複数のファイルを指定するときは、空白で区切ります。
ソースファイル名にはサフィックス .c 、C++プログラムのソースファイル名にはサフィックス .cc または .C が必要です。

プログラムの実行

コンパイルして作成された実行形式ファイルを実行するには、以下の2つの処理方法があります。通常はバッチ処理を利用します。

バッチ処理

バッチ処理は、実行の手続きをジョブという単位でジョブ管理システムに登録し、一括に処理します。ジョブ管理システムはNQSⅡ(Network Queuing SystemⅡ)を用意しており、ジョブの操作はNQSⅡのコマンドで行います。通常のプログラム(長時間実行するプログラム、並列実行するプログラム等)はバッチ処理で実行します。

会話型処理

会話型処理は、コマンドラインでプログラムを実行する形式です。CPU 時間や使用できるメモリサイズに制限がありますので、短時間の演算やデバッグ作業にお使いください。

バッチ処理

プログラムの実行は、NQSⅡのコマンドを用いて操作します。図2は作業の流れを示しています。まずNQSⅡにプログラムの実行を依頼するため、実行の手続きを書いたシェルスクリプトファイルを作成します。作成したファイルはバッチリクエストとして NQSⅡに投入することで、実行の依頼をします。NQSⅡではキューと呼ばれる実行クラスがあり、並列数やメモリサイズの違いにより複数のキューを設定しています。プログラムに合わせ適切なキューを1つ選択し投入します。リクエストの順番が来ましたら、NQSⅡは自動的に実行します。

リクエスト投入後は、リクエスト状態の確認やキューの込み具合等の確認、また投入済みのリクエストをキャンセルすることも可能です。プログラムが終了するとリクエストはキューの情報から消え、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。

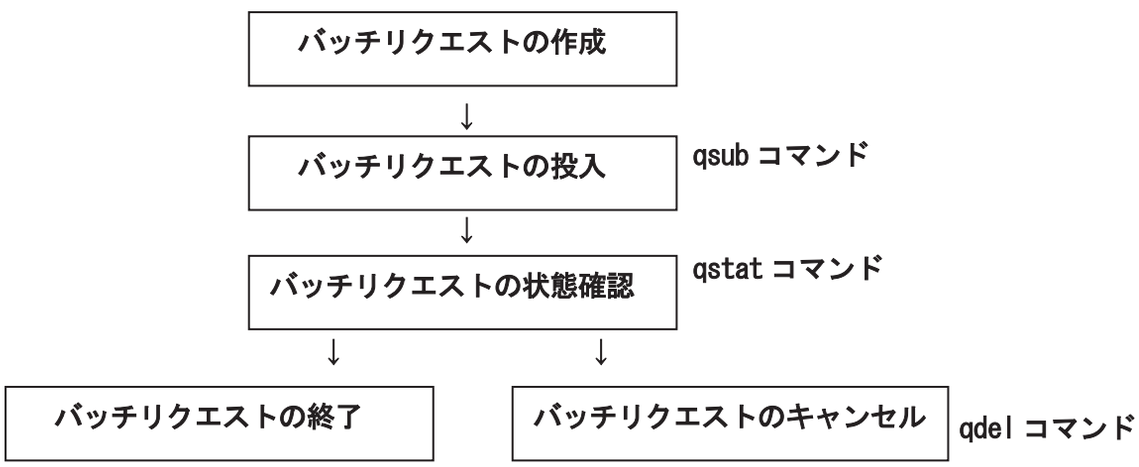

図2　NQSⅡによるリクエストの流れ

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を作成します。プログラムの実行手続きを、通常のシェルスクリプトと同じ形式で記述します。csh スクリプトと sh スクリプト、どちらでも記述できます（以降、解説は csh スクリプトとします）。リクエストファイル名は任意です。

リスト4はバッチリクエストファイルの一例です。実行形式ファイル a.out を実行する手続きを記述しています。

リスト4　バッチリクエストファイル例

```
# test job-a              コメント行
cd work                   作業ディレクトリへ移動
a.out                     実行形式ファイル名
```

・#が先頭の行は、コメント行です。動作には影響しません。

・cd work で作業ディレクトリ（実行形式ファイルのあるディレクトリ）へ移動します。省略するとホームディレクトリを指定したことになります。

・a.out はコンパイルして作成した実行形式ファイルです。あらかじめ会話型処理で作成しておきます。自動並列、OpenMP 用オブジェクトも、同じ形式で指定します。

●作業ディレクトリの指定

NQSⅡ用の環境変数のひとつとしてPBS_O_WORKDIR 変数があります。この変数には、qsubコマンドを実行した時点のカレントディレクトリが設定されます。

NQSⅡの作業ディレクトリは規定値でホームディレクトリとなりますので、通常 cd コマンドで実行ファイルのある作業ディレクトリに移動する必要があります。 PBS_O_WORKDIR 変数を設定することで、ディレクトリの具体名を記述する必要がなくなります。

リスト5　バッチリクエストファイル（環境変数 PBS_O_WORKDIR の指定）

```
# test job-a1
cd $PBS_O_WORKDIR                    作業ディレクトリを環境変数で指定
a.out
```

●実行時のデータファイル指定

Fortranで、入出力ファイルを割り当てる環境変数はFORT*n*です。(nが1～9の場合には0をつけず1桁で指定します)

正しい指定方法:　setenv FORT2 datafile
間違った指定方法:　setenv FORT02 datafile

リスト6　バッチリクエストファイル（入出力ファイルの指定例）

```
# test job-b
setenv FORT1 datafile          装置番号1に、ファイル datafile を割り当てる
cd $PBS_O_WORKDIR
a.out < infile > outfile       標準入出力ファイルはリダイレクションでも可能
```

バッチリクエストの投入

プログラムの実行は、作成したバッチリクエストを NQSⅡに投入して行います。投入されたリクエストは、順番が来ると自動的に実行されます。

【形式】　**qsub　オプション　バッチリクエストファイル名**

システムからのメッセージがリスト7の形式であれば、リクエストは正常に受け付けられています。リクエストID(1234)は一意なもので、リクエストの状況確認やキャンセル等、ジョブの操作の際に必要となります。

リスト7　qsub コマンドの実行例

```
gen$ qsub -q a8 job-a
Request 1234.job submitted to queue: a8.
```

主な qsub コマンドオプション

-q	リクエストを投入するキュー名（クラス名）を指定します。（必須）
-N	リクエスト名（ジョブ名）を指定します。指定がなければ、バッチリクエストファイル名がリクエスト名になります。
-o	標準出力のファイル名を指定します。指定がなければ、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されます。
-e	標準エラー出力のファイル名を指定します。指定がなければ、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されます。
-j o	標準エラー出力を標準出力と同じファイルへ出力します。
-l cputim_job=hh:mm:ss	実行打ち切りの CPU 時間を指定します。設定時間は、時:分:秒を hh:mm:ss の形式で指定します。この指定がなければ無制限となります。並列処理で実行するときは、各プロセスの合計 CPU 時間を指定します。
-m b	リクエストの実行が開始したときにメールが送られます。
-m e	リクエストの実行が終了したときにメールが送られます。
-M メールアドレス	リクエストに関するメールの送信先を指定します。指定がなければ、「利用者番号@gen.isc.tohoku.ac.jp」宛に送られます。

●キュー

-q オプションで指定するキュー名の一覧です。単一 CPU で実行するプログラムは as、並列プログラムは a8,a16,a32 のいずれかに投入します。並列数に応じて適切なキューにリクエストを投入してください。並列用のキュー(a8,a16,a32)を利用するには、並列用のオブジェクトを作成している必要があります。am、am2 は、アプリケーションプログラム Marc 専用キューです。

表2 キュー名

キュー名（ジョブクラス）	利用可能 CPU 数（並列数）	CPU 時間	メモリサイズ制限（GBytes）
as	1	無制限	16
am（Marc 専用）	1	無制限	16
am2（Marc 専用）	1	無制限	128
a8	8	無制限	128
a16	16	無制限	256
a32	32	無制限	512

●qsub コマンドオプションの埋め込み

qsub コマンドに毎回オプションを入力することもできますが、手間を省くためバッチリクエストファイルに指定しておくこともできます。

指定方法は、最初のシェルコマンドより前の行に、#PBS という文字列を先頭に指定します。#PBS の後に空白を一文字以上入れ、指定したいオプションを続けます。一行に複数のオプション指定も可能です。リスト5は、2行目で a8 クラスのキューを指定(-q)、3行目で標準エラー出力を標準出力ファイルにひとまとめにする(-jo)、さらにリクエスト名を reqname とする(-N)を、それぞれ指定しています。

またコマンド列と埋め込みオプションに、同じオプションを指定した場合にはコマンド列の方を有効とします。

リスト8　バッチリクエストファイル（オプションの埋め込み）

```
# test job-a2
#PBS -q a8                          埋め込みオプション
#PBS -jo -N reqname                 埋め込みオプション（複数）
cd $PBS_O_WORKDIR
a.out
```

バッチリクエストの状態確認（1）

投入したリクエストの状態を表示します。実行待ち状態のときは、待ち順も表示します。

【形式】　**qstat**

リスト9　qstat コマンド表示例

```
gen$ qstat
  RequestID      ReqName  UserName Queue    Pri STT S   Memory  ACCPU Elapse R H M Jobs
  -------------- -------- -------- -------  --- --- - --------  ----  ------ - - - ----
  370.job        reqname    x2***9    a8        0 RUN  -  732.1B 42350 43012   Y Y Y    1
1:371.job        jobA       x2***9   a16        0 QUE  -  0.0B      0     0    Y Y Y    1
1:373.job        jobB       x2***9   a32        0 QUE  -  0.0B      0     0    Y Y Y    1
```

主な表示項目の解説

RequestID	リクエスト ID 待ち状態(QUE)のリクエストについては、先頭に待ち順の番号がつきます。番号がないのは、実行中（RUN）です。
ReqName	リクエスト名
UserName	ジョブの所有者
Queue	キュー名
STT	ステータス （QUE:待ち、RUN:実行中）
Memory	使用メモリサイズ （Byte）
ACCPU	演算時間(sec)／並列演算の場合、使用 CPU の総演算時間 （sec）
Elapse	経過時間 （sec）

システム内に自分のジョブが存在しない場合は、ヘッダのみ表示されます。

リスト10　ジョブの終了

```
gen$ qstat
RequestID      ReqName  UserName Queue    Pri STT S   Memory  ACCPU Elapse R H M Jobs
-------------- -------- -------- -------  --- --- - --------  ----  ------ - - - ----
                                                                  ヘッダのみ表示されます。
```

バッチリクエストの状態確認（2）

キューの情報を表示します。各ジョブクラスの件数が表示されますので、サーバの混雑度がわかり

ます。

【形式】　**qstat -Q**

リスト11 -Q オプションの表示例

```
gen$ qstat -Q
[EXECUTION QUEUE] Batch Server Host: job
=====================================
QueueName       SCH JSVs ENA STS  PRI  TOT ARR WAI QUE PRR RUN POR EXT HLD HOL RST SUS MIG STG CHK
--------------- --- ---- ------ ---- ---------------------------------------------------------------
a16               0    0 ENA ACT   32    1   0   0   0   0   1   0   0   0   0   0   0   0   0   0
a32               0    0 ENA ACT   32    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
a8                0    0 ENA ACT   32    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
am2               0    0 ENA ACT   32    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
am                0    0 ENA ACT   32    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
as                1    4 ENA ACT   32    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
p16               1    4 ENA ACT   32    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
p32               1    2 ENA ACT   32    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
p8                1    0 ENA ACT   32    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
s                 0    1 ENA ACT   32    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
ss                0    1 ENA ACT   32    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
--------------- --- ---- ------ ---- ---------------------------------------------------------------
 <TOTAL>                             5   0   0   2   0   3   0   0   0   0   0   0   0   0   0
--------------- --- ---- ------ ---- ---------------------------------------------------------------
```

枠で囲った箇所が、並列コンピュータのキューです。

主な表示項目の解説

QueueName	キュー名
TOT	リクエストの総数
QUE	待ちの件数
RUN	実行中の件数

バッチリクエストのキャンセル

投入済みのリクエストを取り消すこともできます。

【形式】　**qdel** リクエスト ID

リスト12 qdel コマンドの表示例

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

会話型処理

会話型処理は、短時間の演算やデバッグ作業に使用します。一般的なUNIXを利用する手順と同様で、コマンドラインから実行形式ファイル名を入力し実行する形式です（リスト13）。表3は会話型処理の制限値です。

リスト13 会話型処理の例（a.out を実行する）

```
yourhost$ ssh gen.isc.tohoku.ac.jp -l 利用者番号      gen にログインする
    :
gen$ a.out
  （実行中）
gen$
```

表３　会話型処理の制限値

CPU 時間（時間）（並列処理は総 CPU 時間）	メモリサイズ（GBytes）	同時使用 CPU 台数（並列数）
1	8	4

MPI プログラムの実行

MPI プログラムは、mpirun コマンドを使用して実行します。

リスト14 MPI プログラム用バッチリクエストファイル例

```
# test job-a              コメント行
cd work                   作業ディレクトリへ移動
mpirun -n 32 a.out        実行形式ファイル名
```

・-n または-np は起動させるプロセス数を指定します。（省略時は 1 となり並列されません）

ハイブリッド並列プログラムの実行（MPI と自動並列/OpenMP を組み合わせた並列）

ハイブリッド並列プログラム実行時は「プログラム並列数＝MPI 並列数×SMP 並列（自動並列/OpenMP）」になります。MPI プログラムの並列数は mpirun コマンドの-nオプションで制御し、自動並列/OpenMP 並列数は OMP_NUM_THREADS 環境変数で制御します。

リスト15 ハイブリッド並列プログラム用バッチリクエストファイル例
（MPI 並列数 4、自動並列数 8 の 32 並列で実行する場合）

```
# test job-a                コメント行
setenv OMP_NUM_THREADS 8    自動並列/Open MP での並列数
cd work                     作業ディレクトリへ移動
mpirun -n 4 a.out           実行形式ファイル名
```

その他

プログラムの使用メモリサイズ

プログラムを実行した際、使用するメモリサイズをバイト単位で表示します。あらかじめ、必要とするメモリサイズが判断できます。なお、allocate 等で動的に確保するメモリサイズは含まれません。

【形式】　**size** 実行形式ファイル名

```
gen$ size a.out
1046912 + 140272 + 418928 = 1606112        1,606,112 バイト使用します
```

バイナリファイルの扱い(Fortran の場合)

他のサーバで作成したバイナリファイルを扱う場合、注意が必要です。センターでは、並列コンピュータ Express5800 のエンディアン仕様は Big-Endian に設定しております。もし Little-Endian 仕様のバイナリファイルを扱う場合は、環境変数 F_UFMTENDIAN の設定をクリアします。

Little-Endian 仕様のファイルを扱う設定(csh 形式)

【形式】　**unsetenv F_UFMTENDIAN**

マニュアル

現在準備中です。

アプリケーションプログラム

表4は、アプリケーションプログラム一覧です。それぞれの詳しい利用方法は、センターの Web ページをご覧ください。

Web ページ「アプリケーションプログラム」 http://www.ss.isc.tohoku.ac.jp/service/AP/

●表4　アプリケーションプログラムとサービスホスト

アプリケーションソフトウェア		サービスホスト
分子軌道計算ソフトウェア	Gaussian	gen.isc.tohoku.ac.jp
統合型数値計算ソフトウェア	Mathematica	
汎用構造解析プログラム	MARC/Mentat	
対話型解析ソフトウェア	MATLAB	
統計解析システム	SAS	

おわりに

ジョブ管理システム NQSⅡを中心に利用方法の解説と、アプリケーションプログラムの紹介をしました。研究の強力なツールとしてセンターのシステムを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、お気軽にセンターまでお問い合わせください。