◇◇◇◇◇
特　集
◇◇◇◇◇

負荷分散機構 LSF とバッチの投入について

山之上　卓[1]

1　概要

本年度から研究システムにプラットフォームコンピューティング 社 の LSF (Load Sharing Facility, 負荷分散機構) を導入した．LSF はコンピュータネットワーク上でプログラムが起動されるとき，ネットワーク上の最も負荷の軽いコンピュータで自動的にそのプログラムを起動することによって，負荷分散を行なうものである．

新研究システムは，大きな計算を行なうための専用 CPU サーバ群，バックエンドプロセッサを備えている．バックエンドプロセッサを利用するためには，以下で述べるバッチシステムを使用する．このバッチシステムは LSF を利用しており，システム全体で CPU の有効利用を計っている．

CPU の利用効率を高めるため，バックエンドプロセッサはバッチジョブ専用になっている。login はできない．

2　バッチシステムの概要

バッチとはコンピュータを操作するための一連のコマンドを記述して，まとめて実行できるようにしたものである．直接人間とやりとりを行なう必要がない・定型的な処理を行なうプログラムの場合は，バッチを利用した方が便利で実行効率があがる．バッチジョブとは，バッチで記述された処理のことである．以下にバッチの例を示す．

```
#!/bin/sh
make
rm output
myprog < input > output
```

[1]情報科学センター戸畑キャンパス

バッチの投入 (submit) とは，バッチをバッチキューと呼ばれる待ち行列に，実行したいバッチを並べることである．バッチキューに並んだバッチは，先着順などの順番でバックエンドプロセッサで実行される．

バッチの投入は submit コマンドを使う事によって行なう．

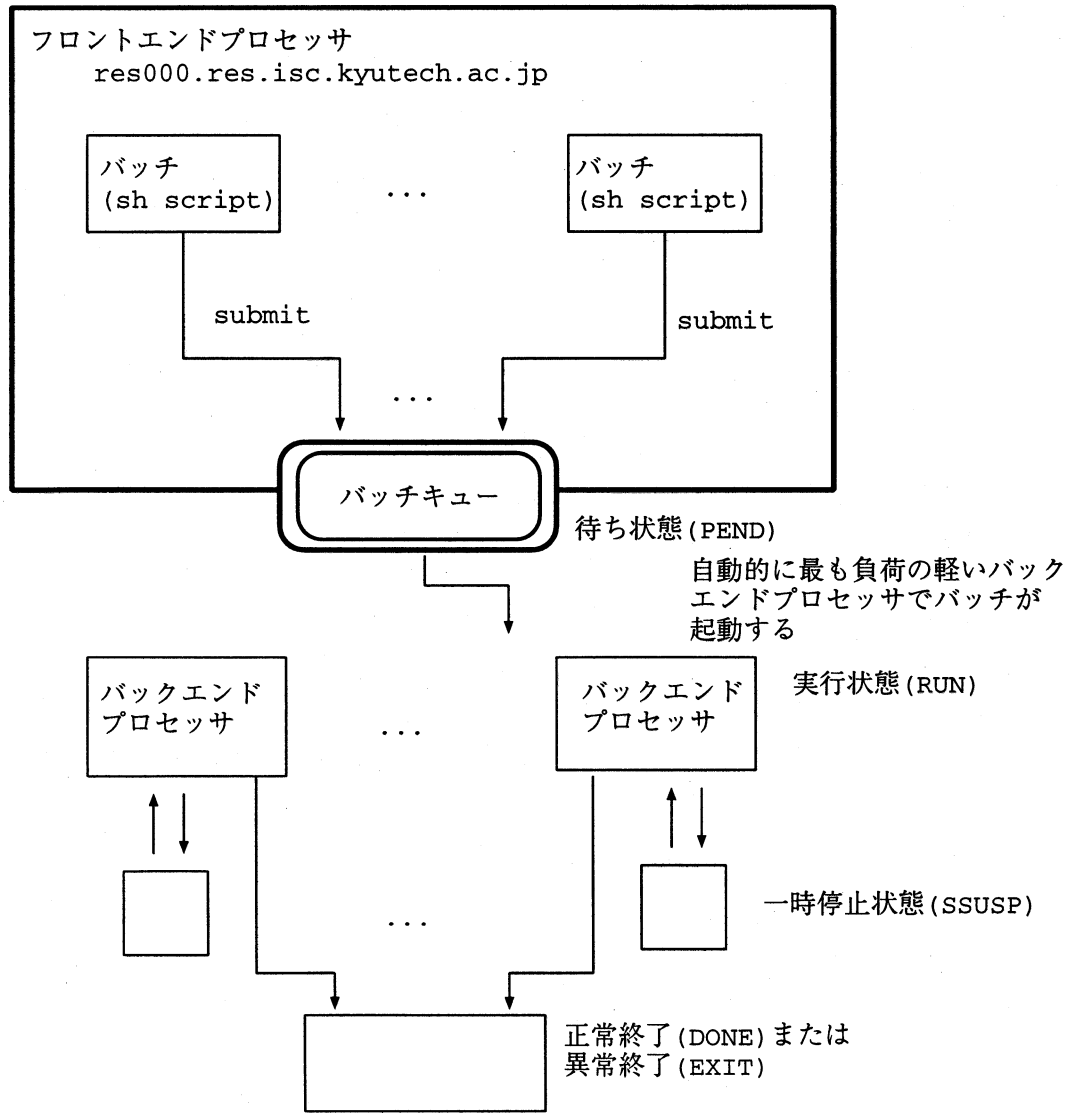

図 1: 研究システムにおけるバッチの流れ

submit コマンドで投入されたバッチには，ジョブ ID(JOBID) と呼ばれる識別子が自動的に割り当てられる．また，バッチシステムに投入されたバッチは以下ような状態を持っている．ジョブ ID とバッチの状態は bjobs コマンドで知る事ができる．

- PEND ... 実行開始待ち (pending) 状態．バッチキューに並んでいて，まだ実行されていない状態である．
- RUN ... 実行状態．
- SSUSP ... 一時停止 (suspending) 状態．CPU で動作する平均プログラムの数が一定値を越えた場合などに，CPU 利用効率を上げるため，1 部のバッチの実行を一時停止させる場合がある．CPU が空いたりするとまた実行を開始する．
- EXIT ... 異常終了状態．バッチそのものが異常終了したり，bkill コマンドでユーザが直接そのバッチを強制終了させた場合にこの状態になる．
- DONE ... 正常終了状態．

3 submit コマンドの使い方

バッチを投入するにはフロントエンドプロセッサ res000.res.isc.kyutech.ac.jp で

```
submit ジョブクラス シェルスクリプトファイル名
```

を実行する．ジョブクラスとは，計算時間の上限や，必要メモリの上限を定めたもので省略可能である．省略した場合は最も制限の強い，ジョブクラス -SA が仮定される．

ジョブクラスには以下のものがある

逐次処理プログラムの場合

ジョブクラス	クラス名
-SA	クラス SA
-SB	クラス SB
-SC	クラス SC
-SD	クラス SD

並列処理プログラム (PVM など) の場合

ジョブクラス	クラス名
-PA n	クラス PA のジョブを投入する
-PB n	クラス PB のジョブを投入する
-PC n	クラス PC のジョブを投入する
-PD n	クラス PD のジョブを投入する

n は使用するプロセッサ数で 4 から 8 まで.

各ジョブクラスは以下の制限を行なう.

逐次処理プログラムの場合

クラス名	CPU 制限時間 (分)	メモリ量
SA 逐次	10 分	16MB
SB 逐次	60 分	32MB
SC 逐次	360 分 (6 時間)	64MB
SD 逐次	1440 分 (24 時間)	96MB

並列処理プログラムの場合

クラス名	CPU 制限時間 (分)	メモリ量
PA 並列 (PVM)	10 分	16MB/CPU
PB 並列 (PVM)	60 分	32MB/CPU
PC 並列 (PVM)	360 分 (6 時間)	64MB/CPU
PD 並列 (PVM)	1440 分 (24 時間)	96MB/CPU
E 要登録	登録値	登録値

クラス PA, PB, PC, PD は PVM を使った並列プログラムのためのクラスである. PVM は, コンピュータネットワークで並列計算を行なうためのソフトウェアで, C や FORTRAN のプログラムにメッセージパッシングを行なう関数などを埋め込んで利用する. PVM については

`http://www.tobata.isc.kyutech.ac.jp/res-tebiki/pvm/`

や本広報の記事などを参照されたい.

なお, 軽い計算のためのクラス SA と PA は他のクラスに比べて優先的に実行される. また並列プログラムの利用を推進するため, 並列プログラムは逐次処理プログラムより優先的に実行させるようにしている.

クラス E はクラス SD や PD の制限値を越えるようなジョブのためのクラスである．このクラスを利用するためには

```
http://www.tobata.isc.kyutech.ac.jp/res-tebiki/how2batch/E-class.ps
```

の申請書に記入して両キャンパス情報科学センター事務室のどちらかに提出していただきたい．

投入されたバッチジョブは，複数台あるバックエンドプロセッサのうち，実行開始時に最も負荷の軽いプロセッサで実行される．

本システムのバッチは UNIX のシェルスクリプトで記述する．例えばユーザが

```
% rm ouput
% jikken-1 < input > output &
```

のようにプログラムを実行していた場合，これをバッチにするためには，

```
#!/bin/sh
rm output
jikken-1 < input > output
```

のようなシェルスクリプトを記述する．シェルスクリプトを記述したファイルは

```
chmod 755 ファイル名
```

などを行なって，実行可能な状態にしておく必要がある．

4 その他の関連したコマンド

submit の他に，バッチジョブの状態を表示する bjobs，バッチジョブを強制終了する bkill，今までに投入したバッチの記録を見る bhist コマンド等を使う事ができる．これらのコマンドの詳しい利用方法については man コマンドで参照する事ができる．

5 バッチシステム利用例

以下にバッチシステムの利用例を示す．

5.1 逐次処理プログラムの場合のバッチ投入

以下は，cat コマンドでバッチを表示し，そのバッチを逐次クラス SB で投入する例である。

```
% cat batch.sh

#!/bin/sh
rm output
myobj <input >output

% submit -SB batch.sh
```

5.2 並列処理プログラムの場合のバッチ投入

以下は，cat コマンドでバッチを表示し，そのバッチを 5 つの CPU を使って並列クラス PA で投入する例である。

```
% cat pvmpai

#!/bin/sh
rm output
pai_master 4 100000000 >output

% submit -PA 5 pvmpai
```

5.3 bjobs コマンド

以下は，バッチを投入後，bjobs コマンドを実行した例である。

```
yamanoue% submit -sa sh2-news

Job <11074> is submitted to queue <class-sa>.

yamanoue% bjobs
```

```
JOBID USER     STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
11074 yamanoue RUN   class-sa   res000      res005.res. sh2-news   Sep 30 17:23
```

5.4 bkill コマンド

以下は，5.3 を実行後，bkill コマンドで投入したジョブを強制終了させた例である。

```
yamanoue% bkill 11074

Job <11074> is being terminated

yamanoue% bjobs

No unfinished job found

yamanoue% bjobs -a

JOBID USER     STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
11074 yamanoue EXIT  class-sa   res000      res005.res. sh2-news   Sep 30 17:23
```

5.5 bhist による過去のバッチ投入記録の表示

以下は，bhist コマンドによって，ユーザが最近投入したバッチの履歴を表示させた例である。

```
yamanoue% bhist -al|more

Job Id <101>, Job Name <yama1>, User <yamanoue>, Project <default>, Command <ec
                     ho 10000000|~/fortran/a.out>
Thu May  9 17:02:45: Submitted from host <res000> to Queue <priority>, CWD <$HO
                     ME>;
Thu May  9 17:02:45: Started on <res001.res.isc.kyutech.ac.jp>, Pid <2656>;
```

```
Thu May  9 17:02:45: Running with execution home </home/hiko/yamanoue>, Executi
                     on CWD </home/hiko/yamanoue>;
Thu May  9 17:02:46: Exited with exit code 1. The CPU time used is 0.2 seconds.


Summary of time in seconds spent in various states by Thu May  9 17:02:46 1996
  PEND     PSUSP    RUN      USUSP    SSUSP    UNKWN    TOTAL
  0        0        1        0        0        0        1
- ------------------------------------------------------------------------------

Job Id <202>, User <yamanoue>, Project <default>, Command <echo 100000000|~/for
                     tran/a.out >~/fortran/output>
Thu May  9 17:21:54: Submitted from host <res000> to Queue <priority>, CWD <$HO
                     ME>;
Thu May  9 17:21:59: Started on <res000>, Pid <585>;
Thu May  9 17:21:59: Running with execution home </home/hiko/yamanoue>, Executi
- --More--
```

6 おわりに

新研究システムの主要な利用法である，バッチ投入について述べた。大型計算機の時代に戻ったような使い方ではあるが，バックエンドプロセッサを計算専用に利用し，研究システムフロントエンドプロセッサなどをプログラム開発に利用するようして，コンピュータを使い分ける事によって，快適に研究システムを利用できることを目指している。

参考文献

[1] ダイキン工業株式会社電子システム事業部，“LSF2.2 ユーザガイド”