

A GPGPUを用いたシミュレーション

近年、GPGPU(General-purpose computing on graphics processing units)を用いて各種の計算が行われるようになった。重力多体問題専用の計算加速器として作られた GRAPE に対して、GPU は本来画像処理専用のプロセッサである。画像処理では多くのピクセルに対して同じ演算を行う必要があるため、GPU には数千コアという非常に多くの演算機が搭載されており、画像処理以外の並列度の高い計算も高速に処理することができる。GPU 向けの GRAPE パッケージを用いると、GRAPE 用のコードを GPU で実行することができる。CfCA でも GPU が搭載された計算機を運用しているので使ってみよう。

A.1 ジョブスクリプトの使用

研究に使う計算機は、多人数でシェアしていることが多い。実際に研究として計算を始めると何日もかかるのが普通なので、何も制限をかけないと誰かがずっと使っていてほかの人はいつまでも使えないというようなことになってしまう。それでは都合が悪いので、CfCA の計算機でも各大学の計算機センターの計算機でも、ジョブ管理システムというもので交通整理している。CfCA では PBS というジョブ管理システムを使っている。

ジョブスクリプトという、どの資源を何時間使用したいという情報が書かれたシェルスクリプトを PBS に渡してやると、優先順位をつけて順に実行してくれる。実習では計算を行う計算機に直接ログインして計算を実行していたが、本来は以下のようにユーザが計算ノードにログインすることはできず、ログインノードから PBS を通じて計算機を使用する。

- muv01 (GRAPE のログイン・管理ノード)
- g9-NUM (GRAPE-9 の計算ノード・ログインできない)
- gdr-NUM (GRAPE-DR の計算ノード・ログインできない)
- muv02 (GPU クラスタのログイン・管理ノード)
- gkNUM (GPU の計算ノード・ログインできない)

GPU を使う前に GRAPE でジョブスクリプトを使う練習をしてみよう。まず muv01 にログインして以下のようなスクリプトを作る。ここでは仮に hello.sh というファイル名にする。

```
#!/bin/bash
#PBS -l nodes=1:ppn=1
#PBS -q g9
cd /home/<USERNAME>/<DIR>
sleep 10
echo "hello world!" > hello.txt
hostname
```

次のようにするとジョブスクリプトがPBSに渡され、実行される、もしくは実行待ち状態になる。

```
$ qsub hello.sh
```

以下のコマンドを打つと、自分のジョブが実行中かどうか確認できる。

```
$ qstat
```

終了したら標準エラー出力と標準出力が

```
hello.sh.eXXXX
hello.sh.oXXXX
```

のようにファイルに保存される。今書いたスクリプトでは標準出力をリダイレクトしているので hello.txt に hello world と書かれているはずである。hello.txt へのリダイレクトを省くと標準出力は hello.sh.oXXXX に書き込まれる。今回の場合 hostname にはリダイレクトを書かなかったので hello.sh.oXXXX に計算ノード名が書き込まれているはずである。リダイレクトをまったくしないでプログラムは正常動作するが、バグなどで大量の標準出力が吐き出されたときに計算機を巻き込んでプログラムが止まることがあるので標準出力はリダイレクトしておいたほうが安全である。

基本的な使い方はこれだけで、講習会のプログラムを投入したければ次のようにすればよい。リアルタイムに図を描くことはできないので、ベリアル比の時間変化などをテキストで出力するタイプのプログラムを投入すること。ここで DIR は collapse の置いてあるディレクトリである。

```
#!/bin/bash
#PBS -l nodes=1:ppn=1
#PBS -q g9
cd /home/<USERNAME>/<DIR>
./collapse > collapse.dat
```

collapse.dat に結果が書き込まれるはずである。N 体学校終了後も試用期間として 2 週間程度は GRAPE を利用することができるが、講習期間が終了すると g9-NUM にはログインできなくなるので、帰ってから遊んでみたい人はこのようなジョブスクリプトを書いて muv01 から投入する必要がある。

そのほか主な PBS のコマンドとしては、

- qsub 投入
- qstat -a 状態の確認
- qdel ジョブ ID) ジョブの削除

などがある。その他の使い方については使用法のページを参照すること。

A.2 GPUを使ったN体シミュレーション

GRAPE ライブラリを使用すると多くの GRAPE のコードをほぼそのまま GPU 上で実行できる。以下のように include するヘッダファイルを変更すればよい。

```
//#include "g5nbutil.h"  
#include "g5util.h"
```

コンパイルオプションを変更して再コンパイルする。ファイル名が collapse-gpu.c であれば以下のようにする。

```
$ gcc collapse-gpu.c -o collapse-gpu -O2 -pthread -lm -lcuda5 -lcudart -lstdc++
```

ジョブスクリプトは以下のようにする。プログラムのあるディレクトリに cd する必要がある。

```
#!/bin/bash  
#PBS -q gk20-1-middle  
#PBS -l walltime=00:00:10  
cd /home/YOUR_ACCOUNT/DIR  
./collapse > collapse.log
```

投入すれば実行される。

```
muv02 $ qsub gpu.sh
```