

姫野ベンチマークの計測
中尾 昌広

1 前回からの課題

- ディスクレスクラスタ構築
- 姫野ベンチマークの測定

2 進捗状況と実験結果

2.1 ディスクレスクラスタの構築

釘井が作成したディスクレスクラスタのマニュアルを参考にして、ディスクレスクラスタを構築した。

2.1.1 ディスクレスクラスタ構築の手順

今回構築したディスクレスクラスタでは Grub のインストールされた FD により、ネットワークを通してクライアントがマスターからのカーネルの読み込みを行った。

2.2 姫野ベンチマーク

2.2.1 姫野ベンチマークとは

姫野ベンチマークとは理化学研究所の姫野龍太郎氏が非圧縮流体解析コードの性能評価のために考案したベンチマークで、ポアソン方程式解法をヤコビの反復法で解く場合に主要なループの処理速度を計るものである。

コードは非常に短く簡単にコンパイル・実行できるので、即座に実測速度を求めることが出来るのが特徴である。また Windows や Machintosh でも利用可能であり、Fortran と C どちらのプログラムも公開されている。

2.2.2 パラメータの設定

姫野ベンチマークのパラメータの種類は問題サイズ (Table 1 参照) とその問題をどのように分割するかの種類だけである¹。

Table 1 問題サイズの大きさ

問題サイズ	配列の数
S	128 x 64 x 64
M	256 x 128 x 128
L	512 x 256 x 256
XL	1024 x 512 x 512

今回の実験では問題サイズは S で、問題の分割はすべて 3 次元目の配列で行うことにする。

¹姫野ベンチマークのアルゴリズムは 3 次元配列を用いて計算を行っており、それぞれの次元において分割を行うことができる

2.2.3 姫野ベンチマークの実験結果

クラスター班で作成したクラスター (magi) と自分のマシン (my machine), そして cambria をを用いて姫野ベンチマークテストを行った。それぞれのマシンスペックは Table 2 の通りである。

Table 2 マシンスペック

name	my machine	magi	cambria
CPU	pentium4	pentium	pentium
クロック	1.7GHz	233MHz	800MHz
memory	128MB	64MB	128MB
Network	-	Ethernet	FastEthernet

問題サイズを固定させて、それぞれのマシンで計測を行った。その結果 Fig. 1 を得た。

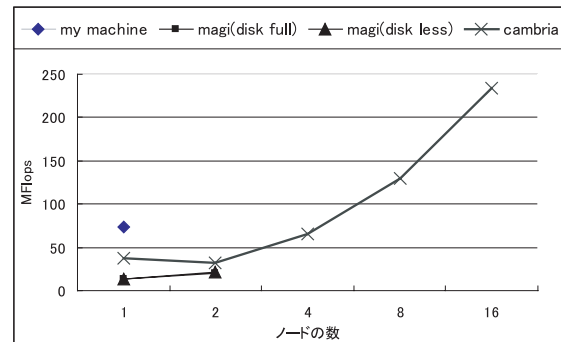


Fig. 1 姫野ベンチマークの結果

cambria で姫野ベンチマークテストを行ったところ、なぜか 1 ノードの方が 2 ノードを用いた結果よりも良かったが、それ以上ノード数を増やすと、そのノード数に比例して増えていくことがわかる。

クラスター班で作成したディスクフルクラスターとディスクレスクラスターとの性能結果はほぼ同じであった。

3 今後の課題

- Linpack ベンチマークのアルゴリズム解析
- 新規導入される PC クラスターの Linpack 計測

4 参考文献

姫野ベンチマーク <http://w3cic.riken.go.jp/HPC/HimenoBMT/result.htm>