Ad-Hoc greedy と SA による ScaLapack のコスト見積もり関数を用いたスケジューリングの実装

斉藤 宏樹

1　今月の課題

GrADS プロジェクトが開発した ScaLapack のコスト見積もり関数を用いて，グリッド上でのスケジューリングを行った論文がある [1]．今月は，この論文で用いられていた Ad-Hoc greedy 法と SA によるスケジューリングのプログラムを実装し，論文をもとにした実験を行い，それぞれの性能評価を行った．

2　数値実験

2.1　SA のパラメータ

SA のパラメータを以下に示す．論文にはパラメータが示されていないため，予備実験によって得られた良好なパラメータに設定した．

Table 1　SA parameter

parameter	value
Max.temperature	100
Min.temperature	10
Number of Step	1000
Cooling Function	$a(T) = aT$
Cooling Rate	0.9977
Cooling Period	10

2.2　計算資源の設定

論文で示されていた計算資源の情報は限られていたため，予想したいくつかのパターンの中から論文に近い実験結果が得られる計算資源の設定を行った．Fig. 1 にその値を示す．Fig. 1 は利用できるノードの数と CPU 速度，ネットワーク速度を示している．

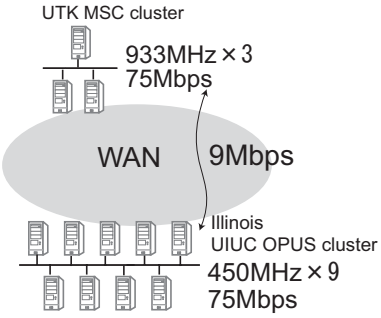

Fig. 1　各計算資源の静的情報

2.3　実験結果

Fig. 1 の情報をもとに，自作した SA と Ad-Hoc で同様の実験を行った結果，Fig. 2 に示す結果が得られた．

Fig. 2 は，問題サイズを 1000〜14000 まで変更したときに得られた最適な見積もりコスト（実行時間）を示している．

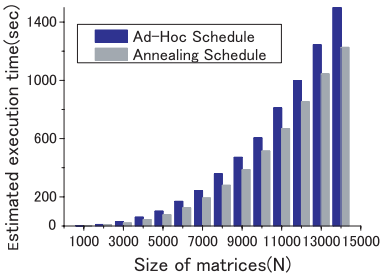

Fig. 2　実装した Ad-Hoc greedy と SA の見積もりコストの比較結果

Fig. 2 より，SA が Ad-Hoc よりも小さい見積もりコストを求めていることが確認できる．SA が良好であるのは，ノードの使用順序も考慮した探索が行えるからである．この傾向は Fig. 3 に示す論文の結果と類似している．ただし自作した手法の結果は，全問題サイズで SA が良いが，論文の結果では問題サイズ 9000 以下では Ad-Hoc が良い値となっている．これは，計算機の設定と ScaLapack のパラメータの設定の関係にあると予想され，今後調査する必要がある．

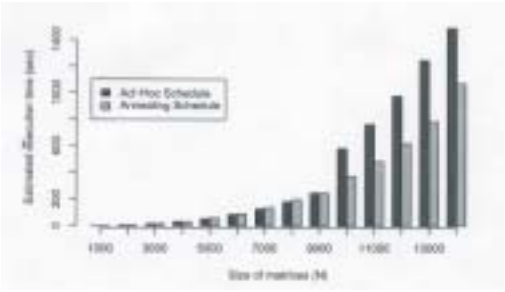

Fig. 3　論文での Ad-Hoc greedy と SA の見積もりコストの比較結果

3　翌月への課題

- GA による ScaLapack のスケジューリング実験
- iSIGHT7.0 における Tech/Gen による実装

参考文献

1) Asim YarKhan, Jack J. Dongarra. Experiments with scheduling using simulated annealing in grid environment. *Workshop on Grid Computing (submitted)*, June 7, 2002