

細粒度最適化問題アプリケーションのグリッドテストベッド上への実装

合 田 憲 人^{†,††} 中 村 心 至[†]

Implementation of a Fine-grain Application Program for Optimization Problems on a Grid Testbed

KENTO AIDA^{†,††} and MOTOYUKI NAKAMURA[†]

1. はじめに

グリッド計算技術は、高性能計算に要するコストを大幅に削減し、大規模並列計算の新たな分野への応用を可能にするものとして注目されている。例えば制御工学やオペレーションズリサーチ分野では、問題解決の手段として数値最適化問題を求解する必要性が高いが、大規模問題求解には莫大な計算時間を要するため、グリッド計算技術への期待が大きい。

本稿では、最適化問題求解のために用いられる分枝限定法プログラムをグリッドテストベッド上へ実装し、実証実験を行った結果について述べる。本実験結果より、分枝限定法プログラムを階層的マスタ・ワーカ方式を用いて並列化し、マスタとワーカ間の通信をローカルサイトの PC クラスタ内に局所化することにより、タスク粒度の小さい最適化問題ベンチマークにおいてもグリッド上で実行時間を短縮できることが確認された。

2. グリッド上での分枝限定法の並列化

分枝限定法は、与えられた問題を複数の子問題に分割し、子問題毎に解の上界値、下界値を計算する処理を繰り返す手法であるが、各子問題の計算は独立に行うことができるため、マスタ・ワーカ方式による並列化が数多く行われている。しかし、一般に通信オーバーヘッドの大きいグリッド上では、単純なマスタ・ワーカ方式による並列化では、マスタとワーカ間の通信時間が大きくなり、十分な実行時間短縮を行えないという問題がある。特に計算対象とする最適化問題によ

ては、計算時間の短い子問題（細粒度タスク）を大量に処理する場合も多く、マスタとワーカ間の通信性能が全体の性能に大きく影響する。

本問題を解決するため、階層的マスタ・ワーカ方式を用いた並列分枝限定法が提案されている¹⁾。階層的マスタ・ワーカ方式では、単一のマスタプロセス（マスタ）と複数のワーカプロセス（ワーカ）からなるグループが複数生成され、スーパーバイザプロセス（スーパーバイザ）がこれら複数のグループを統括する。本方式では、同一グループに属するマスタとワーカをローカルサイト内の PC クラスタ上に配置することにより、マスタとワーカ間の通信を LAN または高性能ネットワーク内に局所化でき、マスタ・ワーカ間の通信性能低下を防ぐとともに、マスタの処理を複数の計算機に分散できるため、単一マスタの処理がボトルネックとなる問題も解決できる。

階層的マスタ・ワーカ方式による並列分枝限定法では、図 1 に示すように、個々のマスタとワーカからなるグループ内では、マスタ・ワーカ方式による並列分枝限定法を用いて対象とする最適化問題の部分問題が計算され、スーパーバイザは複数のグループ（マスタ）間で負荷分散を行うとともに、計算の途中結果である暫定値（Z）の更新を全マスタに対して行う。

3. グリッドテストベッド上での実証実験

本実証実験では、階層的マスタ・ワーカ方式による並列分枝限定法を用いて、数値最適化問題の一つである BMI 固有値問題を求解するアプリケーションプログラムを開発し、日本国内の 4 サイトに設置された PC クラスタからなるグリッドテストベッド上で性能評価を行った。

3.1 実験環境

本実験で使用したアプリケーションプログラムは、

[†] 東京工業大学
Tokyo Institute of Technology
^{††} 科学技術振興機構さきかけ
PRESTO, JST

表 1 グリッドテストベッドを構成する PC クラスタ

名前	計算ノード構成	グリッドソフトウェア	設置場所
blade	dual PIII 1.4GHz, 512MB mem.,	Globus 2.2.3, Ninf-G 1.1.1	東工大 (横浜市緑区)
prestoIII	dual Athlon 1.6GHz, 768MB mem.	Globus 2.4.2, Ninf-G 1.1.1	東工大 (東京都目黒区)
ume	dual PIII 1.4GHz, 1GB mem.	Globus 2.4.2, Ninf-G 1.1.1	産総研 (茨城県つくば市)
sdpa	dual Athlon 1.2GHz, 1GB mem.	Globus 2.4.3, Ninf-G 1.1.1	東京電機大 (埼玉県比企郡)

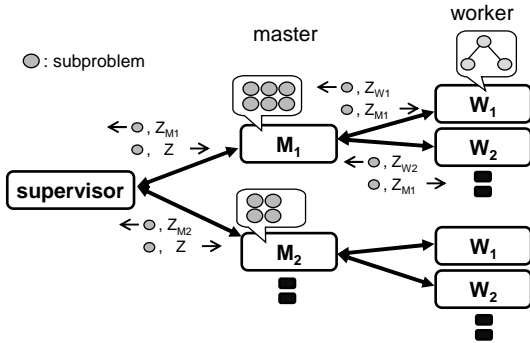


図 1 階層的マスタ・ワーカ方式による分枝限定法の並列化

グリッド上での効率良い階層的マスタ・ワーカ方式を実現するために、Ninf-G および Ninf⁽²⁾ を用いて実装されている。Ninf-G は、Globus Toolkit⁽³⁾ 上に実装された Grid RPC ミドルウェアであり、Globus Toolkit の機能を用いてグリッド上でのユーザ認証や通信データ暗号化といったセキュリティを確保しながら、クライアントマシンが遠隔サーバ上の計算ルーチン呼び出すことを可能にしている。また Ninf では、グリッド上のセキュリティを確保する機能はないが、クライアントマシンは、より少ないオーバーヘッドで遠隔サーバ上の計算ルーチン呼び出すことができる。本実験では、スーパーバイザはインターネット上の異なる PC クラスタ上で実行されるマスタと通信を行う必要があるため、スーパーバイザとマスタ間の処理は Ninf-G を用いて実装することによりグリッド上でのセキュリティを確保し、マスタとワーカ間の処理は Ninf を用いて実装することにより高速なマスタ・ワーカ間の通信を実現した。

表 1 は、本実験で使用したグリッドテストベッドを構成する PC クラスタ群を示す。本実験では、東工大 (横浜市緑区) に設置した PC (PIII 1.0GHz, 256MB mem.) 上でスーパーバイザプロセスを起動し、表 1 上の各々の PC クラスタ上でマスタプロセス 1 個とワーカプロセス 32 個を起動した。

3.2 実験結果

表 2 は、グリッドテストベッド上で BMI 固有値問題ベンチマーク (係数行列サイズ = 24×24 , 解ベクトルサイズ = 6) を求解した場合の実行時間を示す。表中、seq は逐次実行時間を意味し、mw は blade クラスタ上での Ninf のみを用いたマスタ・ワーカ方式による実行時間、hwm $m = N$ は、グリッド上で N 台の PC クラスタを用いて階層的マスタ・ワーカ方

表 2 グリッドテストベッド上での実行時間 [sec]

環境	初期化	計算	終了	総実行時間
seq	-	12042	-	12042
mw	-	427	-	427
hwm $m=1$	8	360	25	393
hwm $m=2$	17	256	55	328
hwm $m=3$	36	172	67	275
hwm $m=4$	49	151	118	318

式を適用した場合の実行時間を示す。なお、 $m = 1$, $m = 2$ および $m = 3$ の場合は、使用する PC クラスタ群に複数の組み合わせがあり得るが、このうち実行時間が最短だったものを示している。また表中の初期化および終了は、Ninf-G における初期化処理と終了処理のオーバーヘッドをそれぞれ示しており、総実行時間 = 初期化 + 計算 + 終了の関係がある。

表 2 より、グリッド上の複数の PC クラスタを用いることにより、問題求解に要するプログラムの実行時間を短縮できることが確認できる。しかしながら、 $m = 4$ の場合は、Ninf-G における終了処理オーバーヘッドが増大し、 $m = 3$ の場合に比べて実行時間がやや増大している。ただし本問題では、求める解が終了処理開始前に得られるため、終了処理を除いた時間のみで比較すると、 $m = 4$ の場合が最短となっている。また、本ベンチマーク問題では、ワーカに割り当てられるタスク 1 個当たりの平均計算時間が 0.5 [sec] 程度と非常に小さい。従って本実験結果から、本実験で用いたような細粒度アプリケーションにおいても、階層的マスタ・ワーカ方式を適用することにより、グリッド上で十分に実行時間短縮が行えることが確認された。

謝辞 性能評価にご協力いただいた東京工業大学 松岡聡氏、産業技術総合研究所 田中良夫氏、東京電機大学 藤沢克樹氏を初めとした Ninf Project のメンバーに感謝いたします。

参考文献

- 1) K. Aida, et al. Distributed computing with hierarchical master-worker paradigm for parallel branch and bound algorithm. In *Proc. CCGrid 2003*, 2003.
- 2) Ninf: A Global Computing Infrastructure. <http://ninf.apgrid.org/>.
- 3) I. Foster, et al. Globus: A Metacomputing Infrastructure Toolkit. *Int. J. of Supercomputing Applications*, 11(2):115-128, 1997.