MPIとマイクロタスクの混在による並列有限要素法の最適化

MPI とマイクロタスクの混在による 並列有限要素法の最適化

奥田洋司＊　工藤真吾＊＊

1. はじめに

近年、有限要素法による解析対象は大規模化、複雑化してきている。そのため高精度・短時間で解析を行うためには、並列処理技術の利用が必要不可欠であり、超並列計算機の最大性能を引き出す計算手法が必要とされている。超並列計算機には大きく分けて、分散メモリ型、共有メモリ型、分散・共有メモリ型と三方式ある[(1)]。

分散メモリ型での並列化は各プロセッサ間で通信しながら計算を進めて行くことになる。このプロセッサ間の通信には MPI(Message Passing Interface)と言った並列処理用ライブラリや、PVM(Parallel Virtual Machine)と言った並列化のためのパッケージを使用する。このため分散メモリ型の並列化はコンパイラ（処理系）に依存しないため、汎用性のあるコード開発ができる。最近発展が著しい PC クラスタなどはこの分散メモリ型のアーキテクチャに相当する。

一方共有メモリ型の並列化は、プログラムの局所的なループの繰り返しなどの並列性に着目し、ソースプログラム中にディレクティブ（並列化制御を指定したコンパイル指示行、Directive for SMP）を挿入することによって、並列化を指定していく。そのため共有メモリ型の並列化はコンパイラに依存したものとなるため、各種計算機ごとに独自の発展を遂げてきた。本研究で扱った[(2)]SX-4 のマイクロタスクなどはその一例である。しかし最近では HPF(High Performance Fortran)、OpenMP(A Standard Application Program Interface for Shared Memory Programming)[(3)]といったコンパイラに依存しない、汎用性のある並列化プログラミングインターフェースの開発が進められている。

ここで、本研究で扱う分散・共有メモリ型の計算機は、共有メモリ型の計算機をノードとし、それらをネットワークで接続した計算機である。そのためノード内とノード間でバンド幅が異なった階層的な構造となるため、並列化には通信や同期の面で工夫が必要となってくる。

そこで本研究では MPI とマイクロタスクによる 2 つの並列化を混在させたハイブリッド並列化手法を開発した。そして Element-by-Element 有限要素法[(4)]定常熱伝導解析を共有・分散メモリ型並列計算機 SX-4 上で行った。ただし計算は 1 ノード（最大 32PE）で実行した。まず入力データを領域分割して、領域間の通信は MPI で行い、各領域内の DO ループをマイクロタスクによってさらに並列に計算した。そして MPI とマイクロタスクによる並列分割数の組み合わせを変えていったときの並列化効率の検討を行った。

＊ 東京大学工学系研究科システム量子工学専攻
＊＊ 横浜国立大学大学院工学研究科

2. 並列処理の種類

(1) データパラレル

Fortran の DO ループなど、同じ作業を違うデータに対して行う部分を並列に処理するプログラミング技法。

・データがどのプロセッサにあるかを意識する必要がない
・複数のプロセッサで動作することを意識しなくて良い
・並列化したいループに対して指示文（Directive for SMP）を挿入
・既存のプログラムの並列化が容易

マイクロタスクや OpenMP、などはこの方法にあたる。

(2) SPMD(Single Program Multiple Data)

全てのプロセッサ上で動く複数のプログラムをユーザが用意し、プロセッサ間の通信を明示的に行うプログラミング法

・基本的に複数のプログラムを作成する
・データがどのプロセッサ上にあるかを意識する
・プログラムが動作しているプロセッサを意識する
・N 個のプロセッサがある場合最大 N 個のプログラムが動作する
（N 個すべてが異なるプログラムである必要はない）
・最も自由度が高く計算機性能を引き出しやすい

MPI を用いたプログラミング。構造－流体連成解析で構造解析と流体解析を別々のプロセッサに割り当てる計算など。

(3) MPMD(Multiple Program Multiple Data)

データパラレルと同様に、同じあるいは似た計算を行う部分を並列に処理する方法。ただし、データがどのプロセッサにあるかを意識し、プロセッサ間の通信を明示的に行う。

・基本的に一つのプログラムを作成する
・データがどのプロセッサ上にあるかを意識する
・複数のプロセッサで動作することを意識する

MPI を用いてプログラミングする。

3. ハイブリッド並列処理

本研究におけるハイブリッド並列処理とは、共有・分散メモリ型並列コンピュータ上のノード間（分散メモリ）を MPI で（MPMD)、ノード内（共有メモリ）を OpenMP で（データパラレル）並列処理し、どの計算機プラットホーム上でも実行することができる汎用性のある並列環境の開発を目指している。Fig.1 はハイブリッド並列処理の様子を表している。ノード内の並列処理を OpenMP で行い、ノード間の通信を MPI で行っている。

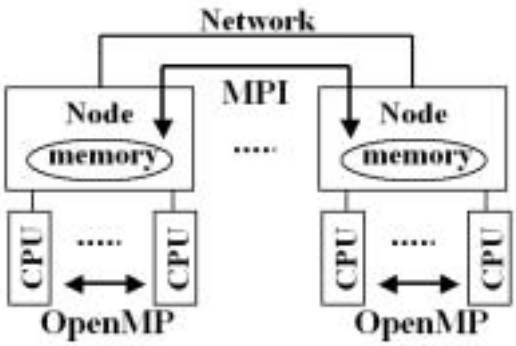

Fig.1 The hybrid image (1)

(1) MPI＋マイクロタスク

本研究では、上記ハイブリッド並列処理の第 1 ステップとしてノード内の並列処理には SX-4 のマイクロタスクを用い、1 ノード内で MPI とマイクロタスクによるハイブリッド並列処理の開発をした。Fig.2 は 1 つのノード内で仮想的に領域分割した並列計算を MPI で通信して、仮想領域内をマイクロタスクによって並列計算する様子を表している。

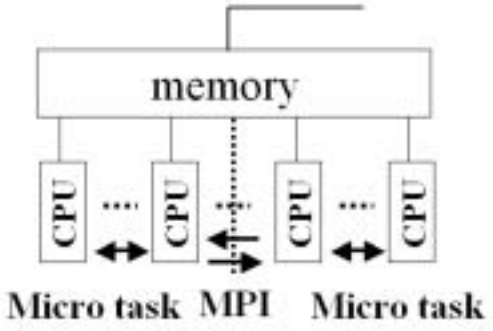

Fig.2 The hybrid image (2)

まず並列処理処理の流れは次のようになる。

① 入力データを領域分割。（パーティショニング）
② 袖領域など、他のプロセッサの値が必要なときは MPI で通信を行いながら並列計算。
③ 1 プロセッサ中の DO ループ計算をマイクロタスクでさらに分割し並列計算。

Fig.3 は 2 領域の問題を MPI で 2PE、各部分領域についてマイクロタスクで 4PE、トータル 8PE で並列処理を行う場合の処理の流れを表したものである。

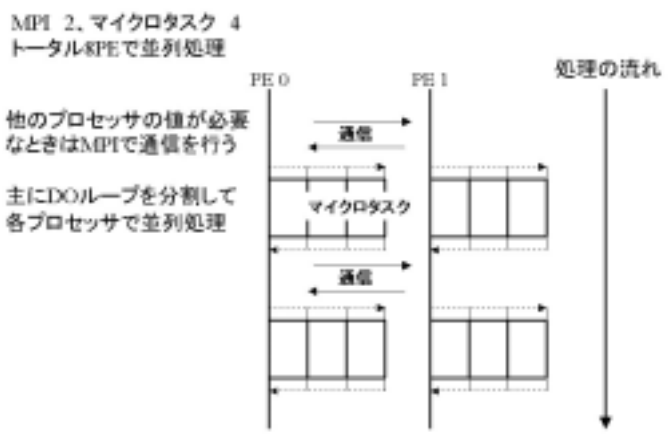

Fig.3 The flow chart of parallel processing

(2) MPI とマイクロタスクの混在における問題点

MPI とマイクロタスクを混在させるときに注意しなければならない点がいくつかある。

一つめの問題は、プログラムの中で MPI とマイクロタスクを同時に利用する場合、マイクロタスク指示行を含むサブルーチン（マイクロタスクサブルーチン）の中では MPI による通信は行えない[5]という問題である。それは本来 1 つしか存在しないあるランクの MPI のプロセスがマイクロタスクサブルーチン中では複数存在する状態になるため送信と受信の対応が取れなくなるためである。

その解決策として本研究ではマイクロタスクによって並列処理をする部分をループ単位でサブルーチンコールしている。その部分を Fig.4 に示す。

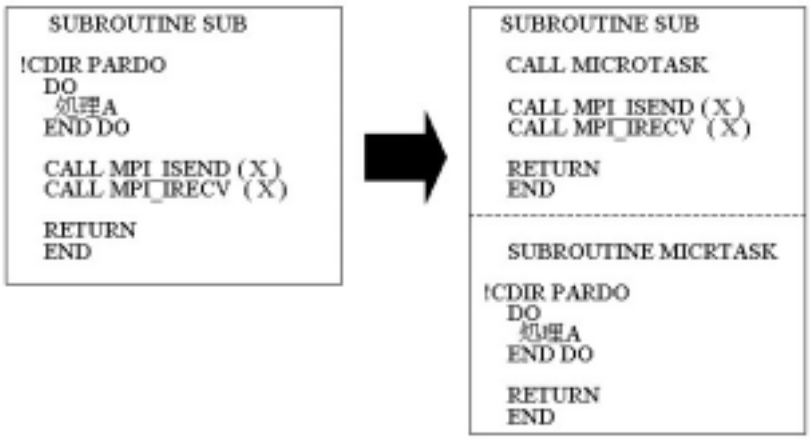

Fig.4 Decomposition to MPI and Microtasking subroutines

もう一つの問題はマイクロタスクサブルーチンの生成のたびにオーバーヘッドが生じる問題である。マイクロタスクサブルーチンを呼び出すたびにオーバーヘッドが生じると、ソルバなどの反復計算を並列処理したときに計算時間に占めるオーバーヘッドの割合が大きくなり、高い並列化効率が得られなくなる。Fig.5 は 1 つの領域で（ここでは PE0、PE1 それぞれ）マイクロタスク手続きを呼び出すたびにオーバーヘッドが生じる様子を表している。

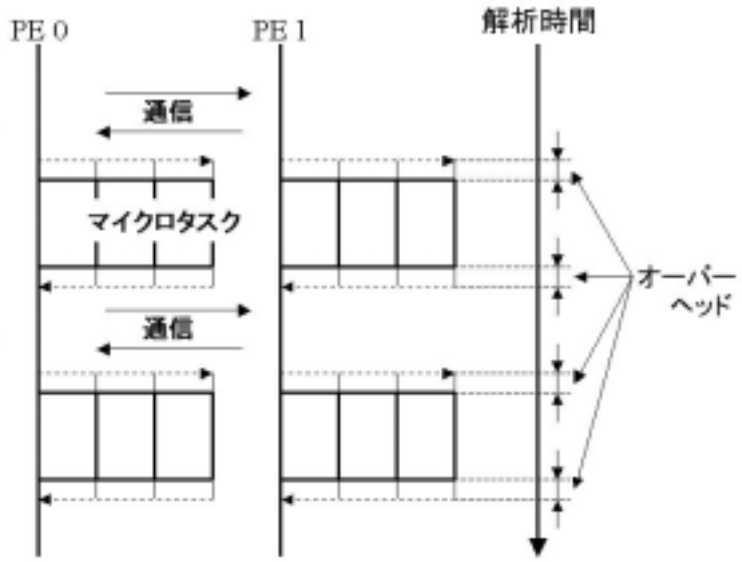

Fig.5 Heavy overhead

そこでその解決策として反復計算するサブルーチン（本研究ではソルバ）の中でマイクロタスク手続きが呼び出されるたびに子タスクを生成するのではなく、反復計算するサブルーチンに入る前に子タスクの生成を行う。

（マイクロタスク指示行 ”RESERVE” ）

ここで、マイクロタスク指示行 ”RESERVE” によって生成された子タスクはマイクロタスク手続きが呼び出されるまで CPU をつかんだまま待ち状態（スピンウェイト状態）になる[5]、という特性がある。

今ソルバの中は MPI サブルーチン（並列化の要素が MPI のみであるサブルーチン）の中にマイクロタスクサブルーチンがあるという入れ子状になっているため Fig.6 のように MPI 手続きの時に CPU はスピンウェイト状態になるため、MPI による通信を行うことができ、また、マイクロタスクサブルーチンでは手続きを呼び出すたびに生じるオーバーヘッドもなくなり、計算時間に占めるオーバーヘッドの割合は少なくなる。

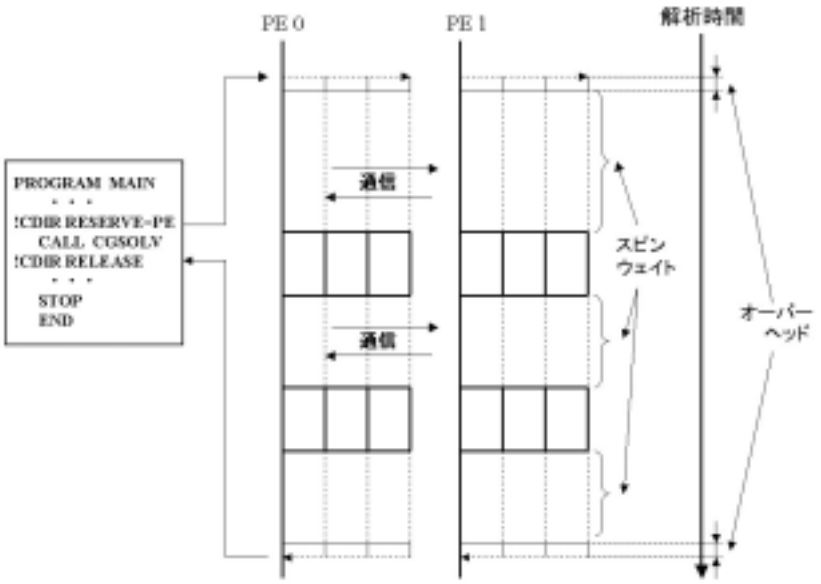

Fig.6 Overhead and spin weight condition

4. 解析例

(1) 解析条件

本研究では Element-by-Element 有限要素法定常熱伝導解析を MPI とマイクロタスクによるハイブリッド並列化手法により並列計算して並列性能評価を行う。
以下解析条件と Fig.7 に解析モデルを示す。

・解析モデル　地下鉄駅構内モデル（Fig.7）
・解析規模　　small －総節点数 29,526
　　総要素数 22,080
　　large －総節点数 204,424
　　総要素数 176,640
・境界条件　　各出口部分（A~F）温度拘束

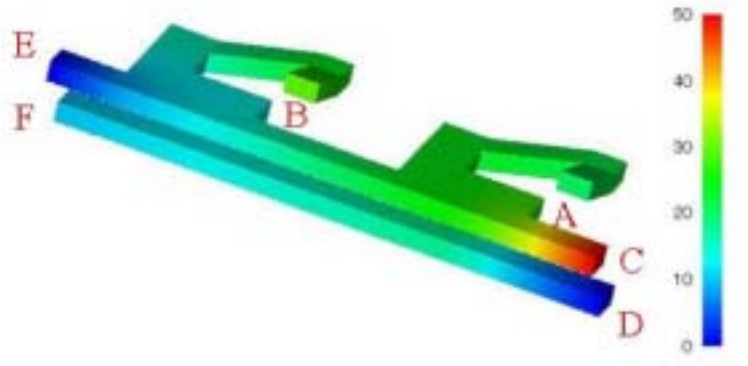

Fig.7 Subway station model

・使用計算機

SX-4 (32CPU/1Node)

東北大学大型計算機センター（現　東北大学情報シナジーセンター）

・並列 PE（Processor Element）数

最大 32PE（1Node×32CPU）

(2) 解析結果

解析した有限要素法ハイブリッド並列定常熱伝導解析の解析時間の大半は、ソルバが占める。ゆえに、並列性能評価はソルバの解析時間で行う。

また、今回はソルバに CG 法を利用した。CG ソルバは反復計算によって行われるため、計測時間は CG ソルバの 100 回の反復にかかる時間‘100 Iter Time’として与えた。

並列性能評価としては並列化効率 P_n を用いた。

$$P_n = \frac{T_1/T_n}{n} \qquad (1)$$

T_1:Elapsed time by 1PE

T_n:Elapsed time by nPE

n:Number of PEs

Fig.8、Fig.9 はそれぞれ small、large モデルの解析において、並列 PE 数を変化させたときの解析時間の推移をグラフにしたものである。PE 数を 2 倍にすると解析時間が約 1/2 になるのがわかる。

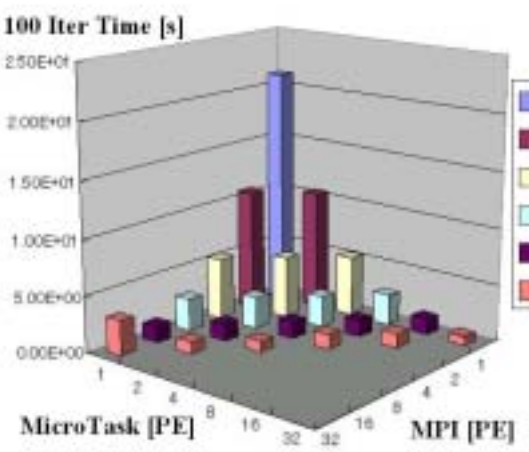

Fig.8 1ITER time of small model

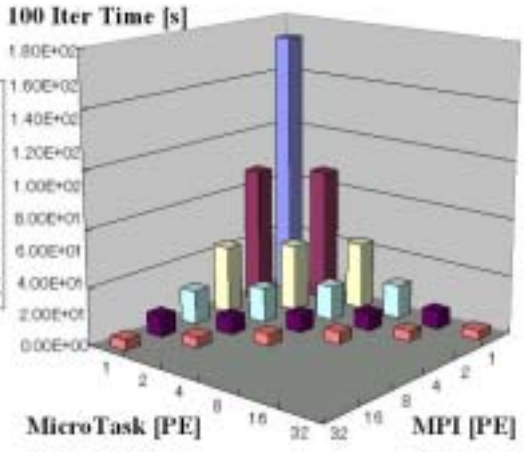

Fig.9 1ITER time of large model

Fig.10、Fig.11 はそれぞれ small、large モデルの解析において、用いる PE 数を変化させたときの並列化効率の推移をグラフにしたものである。

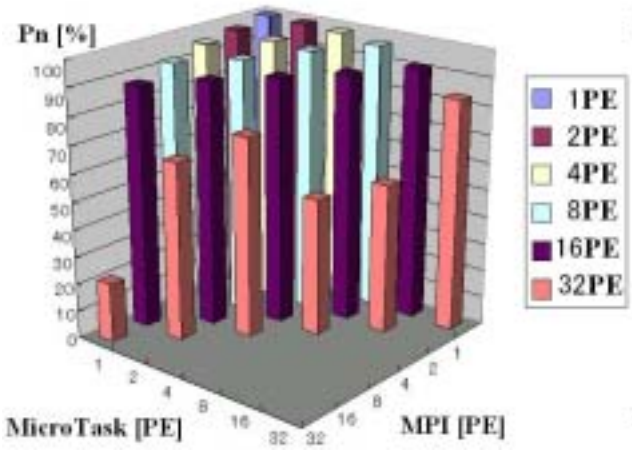

Fig.10 Parallel efficiency of small model

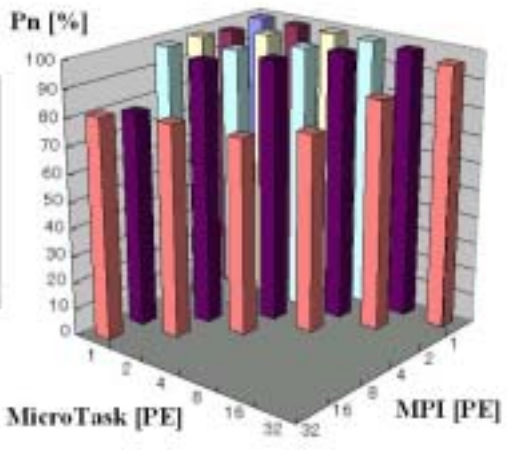

Fig.11 Parallel efficiency of large model

(3) **考察**

まず Fig.8、Fig.9 を見ると MPI でもマイクロタスクでも用いる PE 数が増えると計算時間が小さくなっているのがわかる。また、トータル PE 数が 32 のときに計算時間が大きくなる所があるがこれは 1CPU あたりの計算粒度が小さくなったためと考えられる。
また Fig.10、Fig.11 を見ると small モデルの解析では並列 PE 数が同じでも並列化効率にばらつきが現れるが、large モデルの解析ではそのばらつきは小さくなることがわかる。このことから大規模な解析の方がよりハイブリッド並列計算に向いていることがいえる。

また今回は共有メモリ上で解析を行っているためマイクロタスクよりも MPI の方が並列化効率は高い値となっている。これは本解析では、マイクロタスクによる総バリア同期時間が、MPI による総バリア同期時間よりも多かったため、マイクロタスクによるバリア同期時間が解析時間に占める割合が MPI のそれよりも大きくなったことが原因と考えられる。

今後ノード間にまたがる計算を行うようになったとき MPI は負荷の増大から PE が増えれば並列化効率が低下することが予想される。この並列化効率の、MPI の通信時間による低下と、マイクロタスクの上記理由による低下の兼ね合いが今後、ノード間計算におけるハイブリッド並列最適化の一つの指標になると思われる。

5. 結論

今回の解析でハイブリッド並列化手法による解析の並列化効率は、MPI のみによる並列計算の並列化効率や、マイクロタスクのみによる並列計算の並列化効率と同等の効率を出すことができた。たとえば large モデルの解析においての並列化効率が、32PE を用いた並列計算で、32 領域の問題を MPI のみを使用して並列計算した場合は 46%、1 領域の問題をマイクロタスクのみで 32PE 使用して並列計算をした場合は 49%、8 領域の問題を MPI で 4PE、各部分領域についてマイクロタスクで 8PE 使用して並列計算したハイブリッド並列の場合が 64%となった。

謝辞

本研究は東北大学大型計算機センター平成 12 年度共同研究「MPI とマイクロタスクの混在による並列有限要素法の最適化」として行われた。特に演算負担金の面で多大な便宜を取りはかって頂いたことに感謝の意を表する。

参考文献

(1) 日本数値流体力学会有限要素法研究委員会編,有限要素法による流れのシミュレーション,シュプリンガー・フェアラーク東京,(1988)

(2) 奥田洋司,工藤真吾,阿南統久:有限要素法における MPI とマイクロタスキングのハイブリッド並列処理,計算工学講演会論文集,Vol.5,No1,pp.345-348,(2000).

(3) http://www.openmp.org

(4) Yagawa, G., Nakabayashi, Y. and Okuda, H., Large-Scale Finite Element Fluid Analysis by Massively Parallel Processors, *Parallel Compuut*., 23,pp.1365-1377,(1997)

(5) SUPER-UX FORTRAN90/SX 並列処理機能利用の手引 G1AF08-1,NEC,(1996)