

## 結合ソフトウェア ppOpen-MATH/MP の性能評価と最適化

Performance evaluation and optimization of the coupling software ppOpen-MATH/MP

荒川隆†

井上孝洋†

松本正晴§

佐藤正樹‡

Takashi Arakawa Takahiro Inoue Masaharu Matsumoto Masaki Satoh

### 1. はじめに

現代の計算科学における各分野のシミュレーションプログラムは大規模化・複雑化の一途を辿っており、気候シミュレーションのような複雑系のシミュレーションでは、個々の現象を表現する複数のコンポーネントモデルを結合して計算を実行することも一般的になっている。このようなコンポーネントモデルは、各分野を専門とするサブコミュニティで個別に開発されることが多く、並列化手法や物理量の表現形式は必ずしも一定していない。また、表現すべき現象に適用した独自の格子系や時空間スケールを持っている場合も多い。従って、これらのコンポーネント間で情報を交換するには、格子系や時空間スケール、領域分割手法等の相違を吸収し、適切な変換や通信を行うためのソフトウェア（カップラ）が重要になってくる。このような背景に基づき、結合ソフトウェア ppOpen-MATH/MP の開発計画が進捗中である。先の大会では、ppOpen-MATH/MP の構造や特徴、および適用事例として正二十面体大気モデル NICAM に用いた事例について報告した(1)。本大会では ppOpen-MATH/MP の性能評価と最適化について、特に初期化部分に焦点を絞り報告する。

### 2. ppOpen-MATH/MP の初期化プロセス

ppOpen-MATH/MP の初期化プロセスは、担当するコンポーネントの設定、交換データ情報の設定など複数の手順があるが、性能上問題となるのは、補間テーブル計算に関わるプロセスである。補間テーブルとは、あるコンポーネントの個々の格子点の値が送信側コンポーネントの何番の格子点値（複数あり得る）から計算され、その格子点は送信側コンポーネントの何番の計算ノードに所属するか、を記述したテーブルである。補間テーブルを計算するためにカップラに与えられる情報は 1)各コンポーネントの個々の計算ノードが担当する格子番号、2)受信側コンポーネントの全格子点番号に対する送信側コンポーネントの格子点番号の対応表（これをマッピングテーブルと呼ぶ）、の2種類である。

† 海洋研究開発機構

§ 東京大学情報基盤センター

‡ 東京大学大気海洋研究所

現行の ppOpen-MATH/MP は、これらの情報から補間テーブルを以下のように計算している。

Step1. : 全領域分の受信側格子番号と送信側格子番号の対応表（マッピングテーブル）と各計算ノードが担当する格子番号の情報から、個々の計算ノードが担当するローカルなマッピングテーブルを生成する。

Step2. : 相手格子の個々の格子番号が相手コンポーネントの何番の計算ノードに所属するかを求めため、相手コンポーネントの全ノードから格子点番号の配列を受信し、全領域分の格子番号と計算ノードの対応表から、所属ノード番号を求め。

この一連の手続きで最も深刻な問題は、Step2 において受信側コンポーネントの各計算ノードが送信側コンポーネントの全領域分の格子番号と計算ノード番号のテーブルを持つことである。この方法はメモリ要求が大きく、計算規模が大きくなると計算不能になる状況が容易に発生する。実際、昨年度実施した地震モデルと構造物モデルの結合計算では、当初予定していた計算規模でメモリ不足が発生し、規模を縮小せざるをえなかった(2)。Step2 におけるメモリ問題以外に実行時間に関わる問題も存在する。ここで、受信側コンポーネントの格子数を、全領域、ローカル領域それぞれ NRG, NRL、送信側コンポーネントの格子数を NSG, NSL とすると、Step1 の演算量は  $O(NRL*NRG)$ 、Step2 の演算量は  $O(NRL*NSG)$  になり、どちらもローカルな格子数  $\times$  グローバルな格子数の計算が要求される。そこで、以下に述べるような手法で、メモリ要求量および演算量の削減を図った。

### 3. 初期化プロセスの最適化

#### 3.1 構造格子の場合

対象とするコンポーネントモデルの格子が一定の構造を持っている場合、例えば図1に示すような構造を持っている場合、ある番号の格子がある計算ノードに所属するかどうかの判定は高々繰り返し回数回の計算で終了する。繰り返し回数は分割領域の形状に依存するが、正方形の場合

合、繰り返し回数はほぼ  $(NRL)^{1/2}$  になるため、演算量もそれに比例して少なくなる。また、ある格子番号がどの計算ノードに所属するかの計算については、4(要素)×4(byte)×計算ノード数の情報量で計算できるため、使用メモリ量は大幅に削減可能である。

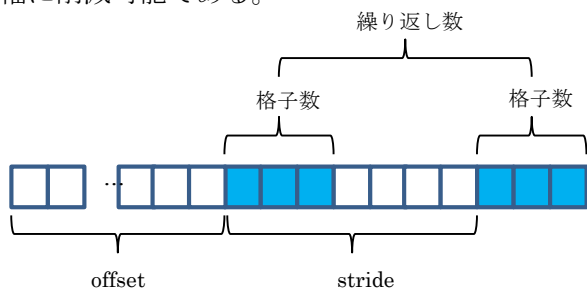


図1 構造格子の格子情報

### 3. 2一般的な格子の場合

構造格子を対象とした場合には、計算量、メモリ使用量とも削減可能である。しかし ppOpen-MATH/MP は様々な格子系に対応できることが大きな利点の一つであるため、特定の格子形状を前提とした最適化は現実的ではない。そこで任意の格子系での最適化を考える。Step1 は、自格子の格子番号を多分木(二分木)で表現することにより、高速な判定が可能となる。Step2 はマッピングテーブルに記述された送信側コンポーネントのそれぞれの格子番号が、送信側コンポーネントの何番の計算ノードに所属するかを求める手続きである。これを以下に述べる2段階の処理で実行するようにした。第一段階は、送信側コンポーネント側で、格子番号と計算ノード番号の対応表を作成するステップである。今、図2に示すように送信側コンポーネントが0から3番まで4つの計算ノードで構成されており、1番から20番までの格子番号が図のように割り当てられているとする。格子番号の最小と最大値および計算ノードの数から、各計算ノードがノード番号を保持すべき格子番を求め、そのアドレスに対して、ノード番号を送る。この手続きにはMPI2のRMA関数であるMPI\_Putを用いている。

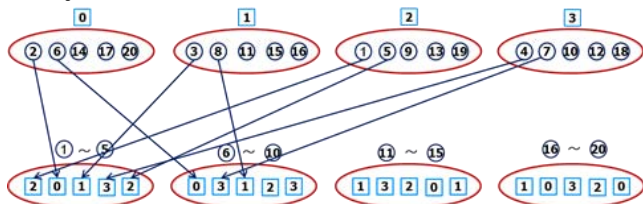


図2 送信側コンポーネントにおける計算ノード番号情報の生成

第二段階は、送信側コンポーネントで生成された計算ノード番号情報を受信側コンポーネントが取得するステップである。はじめに、送信側コンポーネントの格子番号の最小値と最大値および計算ノードの数が受信側コンポーネントに送受信される。これらの情報から、受信側コンポーネントは、ある送信側格子番号の所属ノード番号が保持されている計算ノード番号と配列の位置を知ることができる。従って、求められた計算ノード番号と配列位置にアクセスし、計算ノード番号情報取得する。図3はその過程を示したものである。ここで図上段は受信側コンポーネントを、下段は送信側コンポーネントを表す。図3下段は図2下段と同じ情報である。第一段階と同様、第二段階でも片側通信が要求されるためMPI\_Getを用いて情報を取得している。

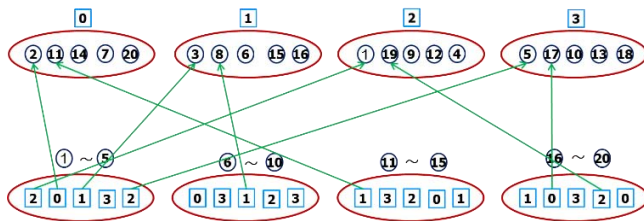


図3 受信側コンポーネントにおける計算ノード番号情報の取得。上段は受信側コンポーネント、下段は送信側コンポーネントを表す。

### 4. まとめ

結合ソフトウェア ppOpen-MATH/MP の初期化プロセスにおいてメモリ・演算量のボトルネックとなっていた格子対応関係の計算コードの最適化について、そのアルゴリズムを示した。現在、2つのコンポーネントが別プロセスで動作する場合のテストが終了している。しかし ppOpen-MATH/MP は2つ以上のコンポーネントが同一プロセス内で直列に結合される条件にも対応している。従って、今後はこれら、より複雑な条件に対応したコードを実装してゆく予定である。

(1) 荒川隆, 井上孝洋, 八代尚, 久保川陽呂鎮, 佐藤正樹: 全球大気モデル NICAM への結合ソフトウェアの適用と性能評価, 情報処理学会全国大会, 京都大学, 2015年3月17日

(2) Masaharu Matsumoto, Takashi Arakawa, Takeshi Kitayama, Futoshi Mori, Hiroshi Okuda, Takashi Furumura and Kengo Nakajima: Multi-Scale Coupling Simulation of Seismic Waves and Building Vibrations using ppOpen-HPC, Procedia Computer Science, 2015, vol.51, 1514-1523