

並列処理数値シミュレーションのための
スケーラビリティ検討方法

3P-8

折居茂夫

日本原子力研究所 計算科学技術推進センター

1. はじめに

科学技術分野の高速シミュレーションを並列処理で行う場合、シミュレーションの対象となる問題の性質、規模、計算スキーム、アルゴリズム、コーディング方法が性能に影響を与え、スケーラビリティの検討を困難なものにしている。

この原因は、並列処理が問題の中にある並列性を利用して性能向上を図ることにある。このため、並列化率を知ることがまず重要となる。

また並列処理のオーバーヘッドは、並列計算機のアーキテクチャと各種性能仕様に大きく依存する。

これらのことを考慮すると、数値シミュレーションコードのスケーラビリティを検討するためには、並列化率と、並列処理のオーバーヘッドを統一して定量的に評価することが必要と考える。そこで本論文では、任意のターゲット数値シミュレーションに対し、プログラムの処理時間のモデルを作り、実効並列化率を定義して複数の並列計算機のスケーラビリティを検討する方法について述べる。このモデルのパラメータを測定すれば、ターゲットプログラムに対する既存の並列計算機のスケーラビリティを検討することができる。このモデルとパラメータを手掛かりにして、ターゲット数値シミュレーションに対しスケーラビリティを持つ並列計算機を設計することも、可能と考える。

2. スケーラビリティ決定要因とモデルパラメータ

数値シミュレーションコードの処理時間モデルは、スケーラビリティ決定要因を含み、現実には起こる現象を近似できなければならない。また誰もがモデルの作成を容易にできる必要がある。これらを考慮して、スケーラビリティを決定する要因とそれに対応したパラメータを右上に示す項目a. ~d. にまとめた。原則的に測定可能な量を選んだため、1パラメータが複数の要因を含む。またリアリティだけ

〔並列処理に関するもの〕

- a. プロセッサの使用効率 (Ep)
(ロードバンス等)
- b. 並列オーバーヘッド
 - ・並列立ち上がり時間 (tp0)
(含む、共有メモリコンテンション)
 - ・通信時間 (tc)
 - ・同期時間 (tcyn)
 - ・タスク生成、消滅時間 (tf, tj)

〔1プロセッサの処理に関するもの〕

- c. 演算器の使用効率 (Es)
(複数演算器の稼働率等)
- d. 1プロセッサオーバーヘッド
 - ・演算立ち上がり時間 (ts0)
(メモリキャッシュのデタラックススピード等)

でなく処理時間を議論するため、項目c., d. を加えた。尚、括弧内の記号がモデルのパラメータである。

3. モデル化方法

数値シミュレーションコードに基づいた処理時間のモデル化は、四則演算の計算回数 $f(n)$ 、通信回数 $g(n, N)$ 、2. 節で示したパラメータ、プロセッサ台数 N 、1プロセッサの最大性能 Ra (flops)、通信最大性能 Rc (B/秒) を用いて行う。ここに、 n はメッシュ数等の問題の規模を表す。

〔あるカラムの1プロセッサ処理時間 t_{lpe} 〕

$$t_{lpe} \equiv f(n) \cdot Ra^{-1} \cdot Es^{-1} + ts0 \quad (1)$$

〔あるカラムの並列処理時間 t_{npe} 〕

$$t_{npe} \equiv t_{lpe} \cdot N^{-1} \cdot Ep^{-1} + tpovh \quad (2)$$

ここに、 $tpovh$ は並列オーバーヘッドで次式で表わされる。

$$tpovh = tp0 + tc + tcyn + tf + tj \quad (3)$$

また、通信時間は次の様にモデル化する。

$$tc = tc0 + g(n, N) \cdot Rc^{-1} \cdot Ec^{-1} \quad (4)$$

ここに、 $tc0$ は通信立ち上がり時間、 Ec は通信効率

A Method of Analyzing Scalability for Parallel Processing Numerical Simulation

Shigeo ORII, on leave from FUJITSU Ltd.

Japan Atomic Energy Research Institute, Center for Promotion of Computational Sci. & Eng.

2-28-8, Honkomagome, Bunkyo-ku, Tokyo, Japan

である。

ノード全体の並列実行時間 $(t_{npe})_T$ は、並列処理のノードの個数を K_p 、逐次処理のそれを K_s とする時、次式で表せる。

$$(t_{npe})_T = \sum_{k=1}^{K_p} (t_{npe})_k + \sum_{k=1}^{K_s} (t_{lpe})_k \quad (5)$$

問題の規模、計算スキーム、アルゴリズムの並列化率への寄与は、式(1)、式(2)を作る時、包含される。

4. スケーラビリティ検討方法

スケーラビリティの検討は、実効並列化率 P_{eff} を次式(6)のように定義して行う。もし、 $K_s=0$ 、 $K_p=1$ で式(6)の t_{povh} が分母の10%を占めれば、 P_{eff} は約90%となり、最大性能は10倍にしかならない。

$$P_{eff} \equiv \frac{\sum_{k=1}^{K_p} (t_{lpe} \cdot E_p^{-1})_k}{\sum_{k=1}^{K_s} (t_{lpe})_k + \sum_{k=1}^{K_p} (t_{lpe} \cdot E_p^{-1} + t_{povh})_k} \quad (6)$$

スケーラビリティの検討手順は次のように行う。

- ・ t_{lpe} , t_c をモデル化
- ・ t_{npe} , t_{lpe} , t_c , t_{cyn} , t_f , t_j をターゲットマシンで測定
- ・ E_s , E_p , E_c , t_{s0} , t_{p0} , t_{c0} を測定結果から推定
- ・ 式(5), 式(6) よりスケーラビリティを検討

5. モデル化方法の妥当性

図-1のLU分解のモデル化と時間測定を行い、モデル化方法の妥当性を検討した。

do20の外側が並列処理ループであり、モデル化すると次の式となる。

$$t_{lpe} = \{ 1/3(n^2-1)Ra^{-1} \cdot E_s^{-1} + (1/2n + 1/3) \cdot t_{s0} \} \cdot n$$

通信は転送用いた総和 $sum(s)$ に含まれており[1]、次のようにモデル化した。

$$t_c = \{ 2 \cdot [t_{c0} + (N-1)/N \cdot (n \cdot 8Byte)/N \cdot Rc^{-1}] \} N \cdot n$$

次に、 $n=5000$ 、 $N=1\sim 16$ 台、並列計算機にVPP500を用い、do20の外側ループについて、計算時間と通信

```
parameter(N$=16, n=5000)
real*8 rL(n, n), s(n)
!xocl processor pe(N$)
do 10 j=1, n
!xocl spread do/(pe, index=1:n, part=cyclic)
do 20 k=1, j-1
do 20 i=j, n
s(i)=s(i)+rL(j, k)*rL(i, k)
20 continue
!xocl end spread sum(s)
10 continue
```

図-1 カラムコレスキー法のLU分解の一部

時間を個別に測定した。その結果、 $E_s=0.45$ 、 $t_{s0}=0.0$ 、 $E_p=0.89$ 、 $t_{p0}=0.00$ 、 $t_{c0}=1.4\mu$ 秒、 $E_c=0.70$ を得た。計算には $R_a=1.6Gflops$ 、 $R_c=400MB/秒$ を用いた。ここに、 $K_p=1$ 、 $K_s=0$ である。図-2にノード数 N に対する並列処理時間の測定値を実線、式(5)の計算値を破線で示す。両者は良く一致し、このモデル化方法が、ノードに対して使用できることがわかる。また、式(6)より、 $P_{eff}(N=16)=98.3\%$ であった。

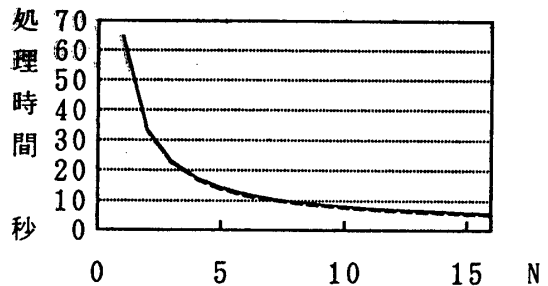


図-2 ノード数Nに対する処理時間

6. 議論

処理時間をモデル化し、パラメータを測定によって求めて評価することにより、スケーラビリティが客観的に議論できる。典型的な数値シミュレーションコードに対しこれを適用することにより、その計算分野が必要とする並列計算機のアーキテクチャと性能仕様が明確になろう。

[文献]

[1]岡田、坂本、浅井：VPP500システムのソフトウェア、電子情報通信学会論文誌 D-1 Vol. J78-D-1, No. 2, pp.149-161 (1995)