

6C-2

配置間相互作用計算の並列処理：
ATOMCIにおける並列化と効果の検証本城信光[§], 大槻一雅^{*}, 関谷雅弘^{*}, 佐々木不可止^{*}[§] 日本アイ・ビー・エム株式会社・東京基礎研(現所属: 同社・東京NICセンター)^{*} 北海道大学 理学部

1. はじめに。 配置間相互作用(CI)計算を電子計算機で高速に実行できるようにすることは, 原子や分子の電子構造を理論的に解き明かす研究を進展させる上での重要な課題である。今回, 並列処理によるCI計算の高速化の可能性を検討する目的で, 原子の全エネルギーを求めるためのCI計算プログラムシステム・ATOMCI[1]の, FORTRAN言語を使ったユーザープログラムレベルでの並列処理化を試み, 並列処理の効果を検証した[2]。並列処理化と効果の検証には, 密結合多重プロセッサ構成計算機上の, 多重仮想記憶システム(MVS)/VS-FORTRAN多重タスク処理機能(MTF)[3]を用いた。

2. 並列化。 2-1) 並列処理の可能性。 ATOMCI原プログラムの実行時間分析を, 計算規模の異なる三つのサンプル(ホウ素B(²P), 炭素C(³P), 酸素O(³P), CI次元数はB(790次元), C(993次元), O(2427次元))に対して行った[4]。その結果, CPU時間の約70%以上(74%(B), 86%(C), 97%(O))がエネルギー表現部(EE部)で消費され, さらにEE部のCPU時間の約99%は, 図式1に示す二重にネストされたループの内側で消費されることが判明した。

```

DO 1 k = 1, N, 1
DO 2 l = k, N, 1
(エネルギー表現Eklの計算)
2 CONTINUE
1 CONTINUE

```

図式1

ループ制御変数のテスト値NはCSF(配置状態関数)タイプ(複数のCSFをタイプ別にまとめて, グループ化する)の数である。外側と内側ループの制御変数の各々でCSFタイプを一つずつ指定したうえで, ループの内側で二つのCSFタイプ(kとl)

の間のハミルトニアン行列要素のブロックHklのエネルギー表現Eklを計算する。各々のエネルギー表現Eklは互いに独立に計算できる。すなわち, あるタスクで読み取りまたは書き込みするデータが, 別のタスクにより変更されることはない。したがって, それらの並列処理は論理上可能である。

2-2) 多重タスク機能利用における効果的並列化法。MVS/VS-FORTRANの多重タスク機能(MTF)のfork-join機構による並列処理を“(複数タスクからなるジョブのポアソン到着/一つのタスクが一つのプロセッサに対して要求するサービス時間がパラメーターrの指数分布/多重度cのプロセッサ: $M_x/M/c$ の待ち行列システム[5])によりモデル化する。ここでは, 一つのジョブがプロセッサを含む全システムを排他的に使用する場合を考える。いまタスク一つ(主タスク)だけからなるジョブの負荷をn個のサブタスクに分割するとすれば, ひとつのサブタスクに対する平均サービス時間は $1/(nr)$ であり, 上述したシステムにおける平均ジョブ・サービス時間 $S(c, n)$ は, Nelsonら[5]が提出した式を用いて次式で与えられる。

Parallel Processing of Configuration Interaction Computing:

Nobumitsu HONJOU, Kazumasa OHTSUKI, Masahiro SEKIYA, Fukashi SASAKI

IBM Japan, Ltd., Hokkaido Univ.

$$S(c, n) = (n - c + 1) / (cnr) + H(c) / (nr) \quad \dots (1)$$

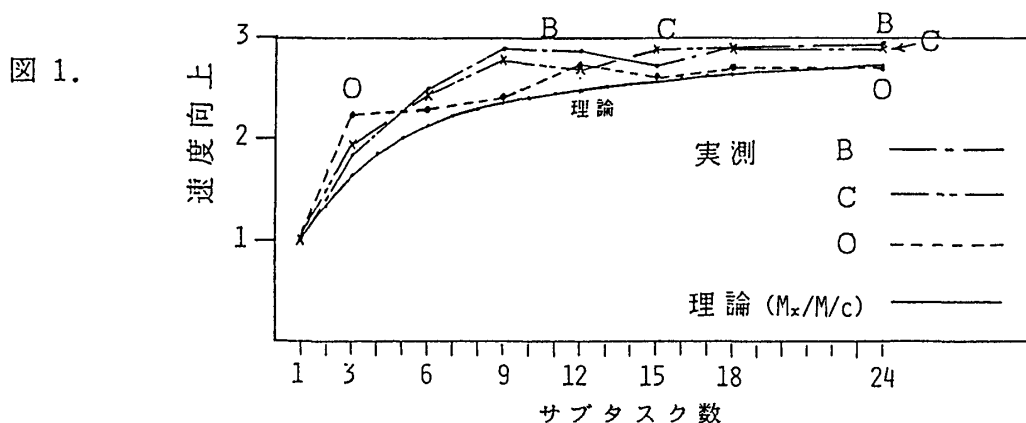
$$\text{ただし, } H(c) = \sum_{k=1}^{c-1} (1/k) \quad (c \geq 2), \quad = 0 \quad (c=1)$$

この式から、 n の増加にともない一つのジョブの経過時間の期待値が単調減少するので、ジョブの負荷をなるべく多くのサブタスクに分割できるような並列化により、高速化が図れることになる。今回の並列化は、なるべく多数のサブタスクへの分割を可能にする方針で実施した。

2-3) C I 計算の並列化法。 2-2の方針に合致した具体的方法として、図式1に示した二重ネストループをより一般化した、 m 重ネストループ($n H_m$ の“重複組合せループ”)の均一繰り返し数分割の方法[6]を用いた。

3. 効果の検証と考察。

図1に、並列処理による並列化プログラム部の計算の速度向上(=サブタスク数1の経過時間/サブタスク数 n の経過時間)を示す。効果を(速度向上/プロセッサ数 c)で定義する。今回の検証でのプロセッサ数は3である。



これより、C I 計算のCPU律速段階に対して、一定(今回は12)以上のサブタスク数では、ほぼ90%かそれ以上の高い並列処理の効果が得られた。C I 計算全体としての並列処理の効果は、C I 次元数が増えるほど高くなることも実測結果から示された。C I 次元数が2427で効果は82%と高い。以上のことから大規模C I 計算において並列処理が実際に有効になることが経験的に証明された。より大規模なC I 計算において、並列処理が一層効果的になると期待される。今後、プロセッサ多重度の増加による速度向上も見込まれる。図1に式(1)による理論値(=S(1,1)/S(c,n))も示した。理論値は、サービス時間が指数分布に従うとする仮定のために、実測値の下界を与える。

謝辞 並列化プログラムの時間測定に協力をいただいた東京NICセンターと同センターの長谷川 明氏、並列処理システムのモデル化に関する議論をしていただいた高木英明氏、奥田 晃氏に感謝します。

文献) [1] F.Sasaki, Int.J.Quantum Chem. Vol.8,605 (1974).

[2] 本城信光,大槻一雅,関谷雅弘,佐々木不可止, IBM Tokyo Research Lab. Research Report,RT0018(1989).

[3] "VS FORTRAN Version 2 Language and Library Reference Release 3", "VS FORTRAN Version 2 Programming Guide Release 3". [4] 使用計算機システム: ハードウェア=IBM3090-300S,オペレーティング・システム(OS)=MVS/ESAバージョン3.1.1,コンパイラ=VS-FORTRAN バージョン2.3. 時間測定は、計算機システムの排他的使用により実施した。

[5] R.Nelson,D.Towsley,A.N.Tantawi,IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, Vol.14,532(1988).

[6] N.Honjou, IBM Technical Disclosure Bulletin 31,No.8, 271 (1989).