

【序】 Hartree-Fock 計算の殆どは二電子積分とそれに続く Fock 行列計算で占められており、理論上は大規模並列化が有効なはずである。だが実際は、Gaussian や GAMESS などの市販プログラムにおける並列計算の並列化効率はいくつもない。それは、現在の並列化手法が、1000 基底程度までの大きさの分子を 10 台規模までの並列計算機環境で計算する事を前提として作られているからである。そこで我々は、WS クラスターのような並列分散計算機環境に適した大規模 Fock 行列計算法として、RT 並列アルゴリズム[1]を開発してきた。

RT 並列アルゴリズムは、行列データ転送をできるだけ外側のループで行うように工夫すると同時に、蛋白質のような大規模系で重要となる積分カットオフの概念を行列データ転送にも拡張することで高いデータ再利用性およびデータ通信量の最小化を実現しており、大規模計算を分散メモリ計算機環境で実行するのに適した並列計算方法である。また、各並列ノードに必要なメモリ量は分子の大きさによらずほぼ一定 (数 MB) であり、どんな大規模分子にも適用できる。

今回、WS クラスターを用いて様々なテスト計算を行い、GAMESS と計算時間を比較した。その結果、**RT 並列計算法は、並列化効率がよいだけでなく、1cpu での計算でさえも大規模系ほど GAMESS に比べて顕著に高速になる**事が分かった[2]ので、あわせて報告する。

【方法】 並列化は MPI を用い、マスタースレーブ方式で行った。計算機は、AthlonMP 1.2GHz が 2cpu 搭載されたマシン 25 台 (合計 50cpu) を用いた。各マシン間は 2 本の 100Base-T で接続されている。OS は Linux RedHat 7.1 である。二電子積分計算は classical な小原法を用いた。

【結果】 1 台の WS で RT 並列の計算速度を検証したところ、大規模系になるほど GAMESS に比べて高速であり、(RT 並列であるがゆえに積分計算数が多いにも関わらず) 最大 4.5 倍以上の高速計算を実現した (Table 1)。次に、50cpu の並列 WS クラスターで並列計算を実施したところ、25 台程度までは極めて良好な並列化効率を示した。ただしこれ以上の台数での並列化効率は少し低下しており、48cpu で約 33 倍であった (Fig.1)。これらの結果、GAMESS での単体計算に比べて最大 150 倍以上の高速計算を実証した。

【考察】 1cpu での計算でさえ RT 並列プログラムが (特に大規模系で) 高速である理由として、カットオフを予め考慮したループ制御、nonredundant basis sets の概念の導入[2]、及びキャッシュヒットを意識したアルゴリズム設計などが挙げられる。実際、RT 並列においては、原始積分 1 個あたりの計算時間は系の大きさによらずほぼ一定であるが、系が大きくなるにつれて積分縮約度が低下するために縮約積分 1 個あたりの計算時間は短くなっている。逆に GAMESS では、系が大きくなるにつれて縮約積分 1 個あたりの計算時間が急速に増大している (Table 2)。また、並列化効率が低下している理由は、並列台数の増加によってホスト計算機での計算 (送受信する行列データの処理) や通信が律速になっているからであり、この部分を中心に現在プログラム改良中である。なお、計算時間の 60~80% は初期積分計算等で占められており、この部分の高速計算手法の開発が望まれる。

(データいずれも 2003 年 7 月現在)

Table 1 1 cpu による Fock 行列計算時間(sec): RT 並列と GAMESS の比較

分子	基底数	Fock 計算時間(sec)			RT/GMS 積分数比 ^(a)
		RT	GAMESS	速度比	
1) Asn	97	15.5	7.7	0.50	2.29
2) Val-Leu-Phe	305	604.3	546.0	0.90	2.95
3) PATGDFMNMSPVG(PDB:1GAQ)	997	9397.0	43717.4	4.65	3.62

基底関数は MIDI。Cutoff threshold=1.0e-15。Schwarz inequality cutoff, Differential Fock method は利用せず。 ^(a)GAMESS に比べて RT 並列で計算される二電子積分数の比。

Figure 1 WS クラスターにおける RT 並列の並列化効率 (上記テスト 3) の場合)

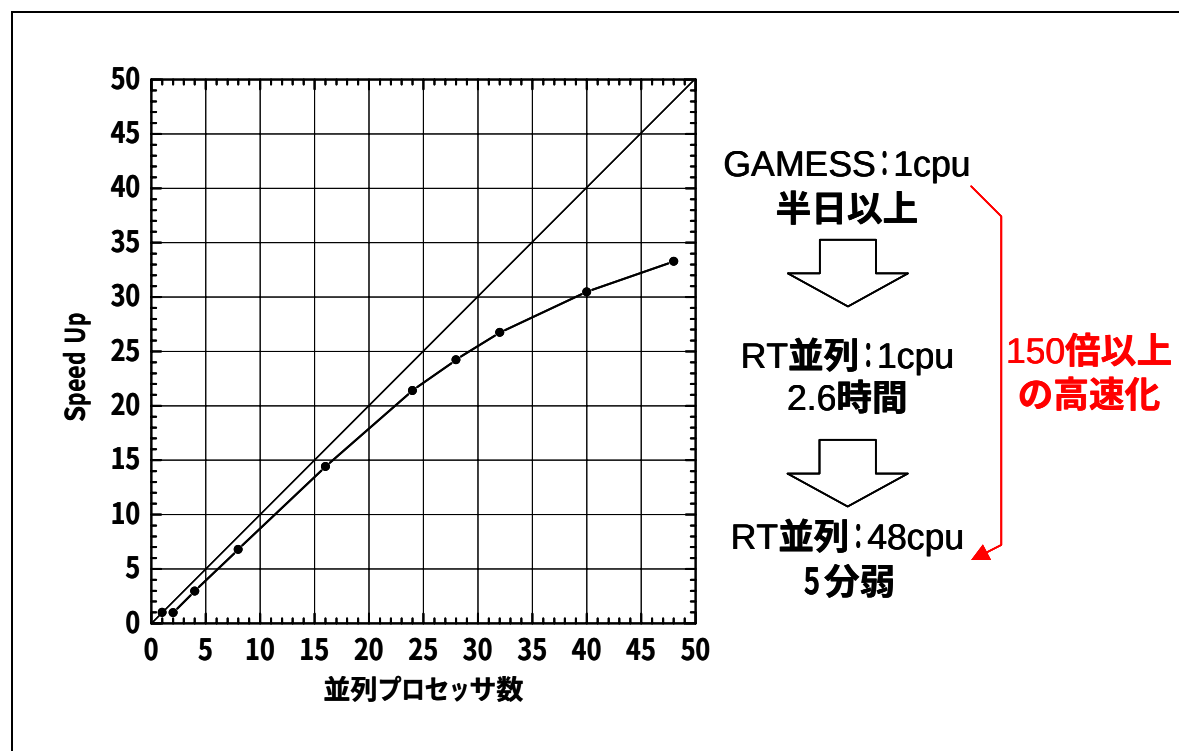


Table 2 積分一個あたりの平均計算時間(ns): RT 並列と GAMESS の比較

	基底 数	原始積分 1 個の平均計算時間(ns)			縮約積分 1 個の平均計算時間(ns)			平均縮 約度 ^(b)
		RT	GAMESS	速度比	RT	GAMESS	速度比	
1)	97	189	n/a ^(a)	n/a	774	881	1.14	4.10
2)	305	202	n/a	n/a	490	1306	2.66	2.43
3)	997	199	n/a	n/a	395	6662	16.85	1.99

^(a)n/a: not available. ^(b)平均縮約度=全原始積分数/全縮約積分数。

- [1] H.Takashima et al., "A Novel Parallel Algorithm for Large-Scale Fock Matrix Construction with Small Locally Distributed Memory Architectures: RT Parallel Algorithm", *J.Comp.Chem.*, **23**, 1337 (2002)
- [2] H.Takashima and K.Kitamura, "Rapid Computation of all sets of electron-repulsion integrals for large-scale molecules", *Chem.Phys.Lett.*, **377**, 43 (2003).