

大規模 SMP クラスタにおける OpenMP/MPI ハイブリッド NPB, RSDFT の評価

辻 美和子^{†,††} 佐藤 三久^{††}

本稿では、RSDFT コードの OpenMP/MPI ハイブリッド並列化、および RSDFT コードと NPB コードを用いた大規模マルチコア SMP クラスタ上における OpenMP/MPI ハイブリッド・プログラミングの性能評価を行った。実験から以下が明らかになった：ハイブリッド・プログラミングは、適切なデータ分散が困難な場合でも、動的な負荷分散を行うことができた。ハイブリッドによるプロセス数の減少は、MPI プロセス間の同期のための待ち時間を軽減した。ただし、OpenMP スレッドがローカルメモリを利用できない場合、大幅な性能低下が見られた。

Performance Evaluations of OpenMP/MPI hybrid NPB and RSDFT codes on a large-scale SMP cluster

MIWAKO TSUJI ^{†,††} and MITSUHIKA SATO^{††}

We evaluated the performance of hybrid NPB and RSDFT codes, which are parallelized by OpenMP/MPI hybrid programming. In our experiments, it is shown that : The OpenMP/MPI hybrid programming model takes advantage of dynamic load balancing in problems for which appropriate data distribution to each MPI process is difficult. Smaller number of MPI processes in hybrid programming reduces the waiting time for MPI synchronization. The advantage of OpenMP/MPI hybrid programming can be realized if OpenMP is used within a multicore processor and each thread can utilize its local memory through the numactl utility.

1. はじめに

本論文では、そのようなマルチコア SMP クラスタにおいて、共有メモリ、マルチスレッド・プログラミングと分散メモリ、メッセージパッシングとのハイブリッドの可能性を検討する。プラットフォームとしては T2K Tsukuba を使い、NAS Parallel Benchmark (NPB) に対して OpenMP/MPI ハイブリッド並列とフラット MPI 並列の比較する。また、第一原理計算を行う Real-Space Density Functional Theory (RSDFT) コードを OpenMP を用いてハイブリッド並列化し、性能評価およびハードウェア・イベント・カウンターの収集を行う。

既存の SMP クラスタに対する性能評価においては、しばしばハイブリッド・モデルはフラットな MPI モデルに劣ると報告されてきたが、実験の結果、マルチコア型のプログラミングの台頭により、ハイブリッド・モデルはしばしばフラット MPI と同等もしくはより高速であることが明らかになった：OpenMP/MPI ハイブリッドプログラミングモデルは、分散メモリ型の並列処理では適切なデータ分散が困難な場合でも、動的な負荷分散が可能

である。また、MPI プロセスの数を抑えることができるため、プロセス同期時の待ち時間が減少する。

2. 関連研究

本章では、OpenMP と MPI のハイブリッド・プログラミングに関する既存の研究について述べる。初期の研究においては、フラットな MPI モデルが、OpenMP/MPI ハイブリッド・モデルよりも優れた性能を示した。例えば、Cappello ら⁴⁾は、ハイブリッドプログラミングの性能はアプリケーション依存であるとしながらも、多くの場合でフラット MPI のそれに劣るとした。SPAM コード³⁾では、ハイブリッドモデルは L2 キャッシュミス OpenMP 並列の不規則なデータアクセスに起因するキャッシュミスにより、ハイブリッドプログラミングは高い性能を示さなかった。他のベンチマークや CFD, CG などのアプリケーションにおいても^{7),10)}ハイブリッド化による顕著な性能の改善は見られなかった。

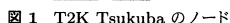
近年では台頭するマルチコアクラスタに対して、いくつかの性能評価が行われている^{5),9)}。本論文では、大規模マルチコアシステムの性質に焦点をあて、ベンチマークおよび実際の問題での性能評価を行う。

[†] 東京大学物性研究所

Institute for Solid State Physics, University of Tokyo

^{††} 筑波大学計算科学研究センター

Center for Computational Sciences, University of Tsukuba



648 node (quad-core x 4socket / node)
Optoner "Barcelona" B8000 CPU
2.3GHz x 4FLOP/c x 4core x 4socket
= 147.2 GFLOPS / node
95.3 TFLOPS / system
20.8 TB memory / system
800 TB (physical 1PB) RAID-6, Luster cluster file system



```
NSLOTS=4
SLOT=$(( $MPIRUN_RANK % $NSLOTS ))
numactl --cpunodebind=$SLOT --membind=$SLOT ./a.out
```

— 164 —

表 2 ターゲット問題とそれぞれの問題で用いられたプロセス数およびスレッド数. OpenMP/MPI ハイブリッドに関しては、総スレッド数に続いて MPI プロセス数と OpenMP スレッド数を [(# of MPI processes) x (# of OpenMP threads)] の形で示す.

CLASS B			CLASS D		
MPI	4 OMP/MPI	16 OMP/MPI	MPI	4 OMP/MPI	16 OMP/MPI
1	1 [1x1]	1 [1x1]	64	64 [16x4]	-
16	16 [4x4]	16 [1x16]	256	256 [64x4]	256 [16x16]
64	64 [16x4]	64 [4x16]	1024	1024 [256x4]	1024 [64x16]

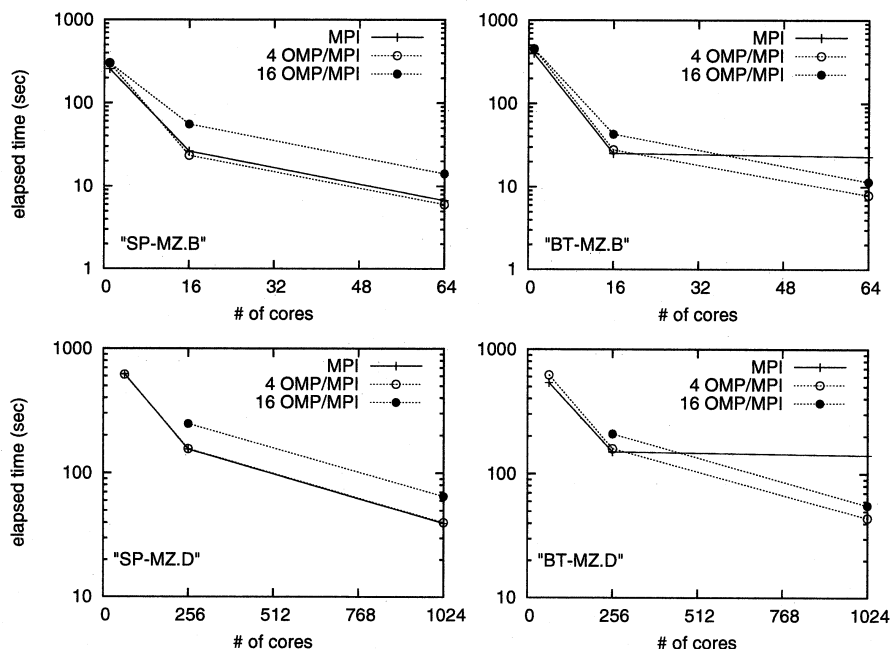


図 4 SP-MZ.B (右上), SP-MZ.D (右下), BT-MZ.B (左上), BT-MZ.D (左下) の実行時間

表 3 SP-MZ.B に対する実行時間. 1 ソケットが 4 プロセスもしくは 4 スレッドを実行する場合と 1 ソケットが 1 プロセスを実行する場合

# of nodes	MPI(4 process in 1 socket)	MPI(1 process in 1 socket)	4 OpenMP/MPI	16 OpenMP/MPI
1	26.29	-	23.44	55.33
4	6.73	14.96	6.04	14.16
16	-	3.63	1.45	3.98

うち 3 つはアイドル状態になる. 表 3 から, 1 ソケットが 1 プロセスを実行する場合, 高いバンド幅を利用できるため, 同じ数のコアを使用する他の場合と比較して高速だった. しかし, 確保されるノード数で比較すれば, 他のモデルのほうが高速だった.

5. RSDFT

実空間密度汎関数法 (Real-Space Density Functional Theory, RSDFT) は, 電子系のエネルギーなどの物性を電子密度から計算することが可能であるとする密度汎関数理論に基づき, 実空間の格子において第一原理計算を行うものである. DFT に基づく第一原理計算は数学的に

は, 以下の Kohn-Sham 方程式の固有値問題の解を固有値の小さいものから順に N 個求めるという問題である:

$$\left[-\frac{1}{2} \nabla^2 + v[\rho](\mathbf{r}) + \hat{v}_{NL} \right] \psi_n(\mathbf{r}) = \varepsilon_n \psi_n(\mathbf{r}), \quad (1)$$

ただし ψ_n は波動関数, N は求める波動関数の数 $\rho(\mathbf{r}) = \sum_{n=1}^N |\psi_n(\mathbf{r})|^2$ は電荷密度である. 式 (1) の左辺の演算子 (行列) はハミルトニアンと呼ばれる. RSDFT では, (1) とハミルトニアンが自己無撞着に解かなければならない.

オリジナルのフラット MPI の RSDFT コードは, 岩田らによって Fortran を用いて開発された⁸⁾. このコードは, 格子実空間を $np_1 \times np_2 \times np_3$ にブロック分割する. ブロック数と MPI プロセス数は等しく, 1 プロセスが 1 ブロックを受け持つ. RSDFT のメインループは自

表 5 Si525H276						
	1x16	4x4	16	4x16	16x4	64
DIAG	94.07	33.17	28.53	31.84	10.93	9.14
(MatE)	11.07	6.85	7.84	2.83	2.03	2.10
(hpsi)	13.32	4.93	5.90	2.70	1.52	1.49
(pzheevd)	16.08	8.09	2.95	8.52	2.37	1.53
(RotV)	53.60	13.30	11.86	17.78	4.99	4.01
CG	122.19	53.62	62.59	30.24	17.69	17.06
(HPSI)	57.98	23.91	30.31	14.37	8.38	9.04
(PC)	51.15	22.17	25.23	11.26	7.20	6.18
GS	26.74	14.51	14.35	6.34	3.64	3.94
(total)	286.53	116.86	123.65	76.71	36.78	34.73

表 6 Si1978H642 (16x16 は実行不可能)				
64x4	256	64x16	256x4	1024
149.26	127.78	152.96	53.01	64.41
27.74	38.21	10.81	10.75	14.56
9.93	8.58	7.18	3.40	3.90
42.04	12.68	48.53	17.08	15.92
69.58	68.32	86.24	21.57	29.78
129.45	116.72	107.82	54.52	71.50
64.52	57.59	39.87	19.69	26.17
52.38	45.10	56.52	26.07	36.72
52.64	63.94	19.31	17.29	22.75
355.82	331.23	298.59	133.42	170.82

2種類の行列計算に関して、各プロセスごとに行列計算要した時間を合計し、それらの最大、最小、平均を示したものである。フラット MPI では 256 のプロセスが、OpenMP/MPI ハイブリッドでは 64 プロセス・4 スレッドが用いられている。ハイブリッドモデルでの行列計算の実行時間は、スレッドの立ち上げや同期が含まれているために、その平均はフラット MPI よりも大きくなっている。その一方で、実行時間の最大値はプロセス数が多いフラット MPI モデルのほうが大きくなっている。最大値の大きさ、および最大値と平均値との差は、フラット MPI モデルが MPI プロセスの同期のためにより多くの待ち時間を費やす必要があることを示唆する。

DIAG サブルーチン内の (pzheevd) はプロセス数に対してスケールしていない。例えば、フラット MPI で Si1978H642 を解く例を見ると、256 プロセスを用いたときには 12.68 秒を (pzheevd) に費しているのに対して、1024 プロセスを用いたときには 15.92 秒が必要とされている。(pzheevd) は現時点では陽に OpenMP によるスレッド並列がなされていないことから、フラット MPI のほうが多くの場合で高速である。将来の研究として、OpenMP/MPI の固有値問題ソルバーを実装する必要がある。

5.4 解 析

RSDFT コードに関して、PAPI (Performance API)¹⁾を用いて、それぞれのサブルーチンにおけるハードウェア・イベント・カウンタの収集を行った。問題は Si525H276、1 ノード・16 コアを使用し、フラット MPI のときは 16MPI プロセス、OpenMP/MPI ハイブリッドのときは 4MPI プロセス 4OpenMP スレッドとした。収集したイベントを下表に示す。ただし、フラット MPI のときはすべてのプロセスのカウンタの平均、ハイブリッドのときはすべてのプロセスのスレッド 0 の平均である。

項目	概要
PAPIL1/L2.DCM	L1/L2 data cache misses
PAPIL1/L2.DCA	L1/L2 data cache accesses
PAPIL1/L2.ICM	L1/L2 instruction cache misses
PAPIL1/L2.ICA	L1/L2 instruction cache accesses
PAPITOT_CYC	Total cycles
PAPITOT_INS	Instructions completed
PAPIFPU.IDL	Cycles integer units are idle
PAPISTLJCY	Cycles with no instruction issue
PAPIBR_MSP	Conditional branch instructions mispredicted
PAPIBR_INS	Branch instructions
PAPITLB.DM	Data translation lookaside buffer misses
PAPITLB.IM	Instruction translation lookaside buffer misses

結果を図 10 に示す。図からキャッシュ利用に関しては、L2.ICM において、ハイブリッドのほうが値が大きくなっている。また、インストラクション・トランスレーション・ルックアサイド・バッファ・ミスを示す PAPITLB.IM がハイブリッドで大きく、特に DTCG ではフラット MPI の 150 倍程度だった。命令発行のないサイクルを示す PAPISTLJCY や条件分岐命令の予測失敗を示す PAPIBR_MSP では、フラット MPI のほうが大きな値を示した。

6. 結 論

大規模マルチコア SMP クラスタ上での OpenMP/MPI ハイブリッド・プログラミング・モデルについて NPB および RSDFT コードが用いて性能評価を行い、以下を明らかにした：OpenMP/MPI ハイブリッド・プログラミング・モデルはデータ分散が難しくフラットな MPI ではスケーラビリティがない場合でも動的な負荷分散が期待できる。またハイブリッド化による MPI プロセス数の減少は、MPI 通信の同期のための待ち時間を軽減する。しかしながら、異なるソケットにスレッドが存在する場合には、ローカルメモリを効率的に利用することができず、性能が大きく低下する。OpenMP/MPI ハイブリッド・プログラミング・モデルの効果は、OpenMP がマルチコア・プロセッサ内部で利用された時に、もっとも発揮される。RSDFT コードにおいては、MPI による並列固有値問題ソルバのスケラビリティが充分でなく、将

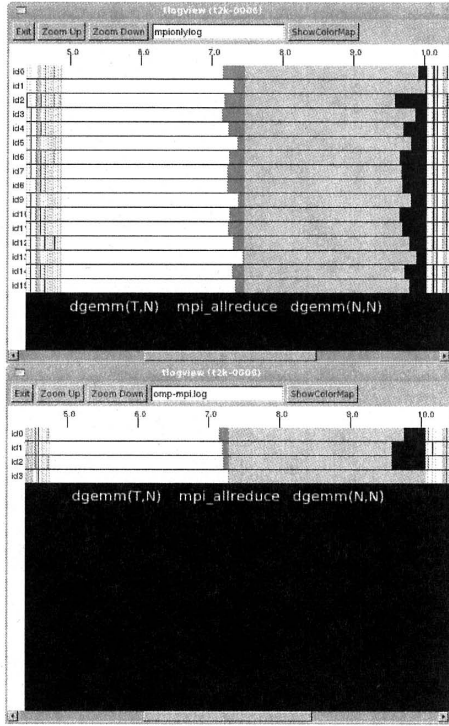


図 8 Si525H275 に対する図 6 の行列計算と allreduce のスナップショット。上が MPI のみ 16 プロセスの場合、下が 4 スレッド 4 プロセスの OpenMP/MPI ハイブリッドの場合。

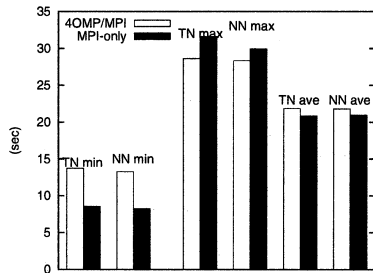


図 9 Si1978H642 に対して 256 コアが使われた場合の図 6 の dgemm(T,N) (TN min, TN max, TN ave) dgemm(N,N) (NN min, NN max, NN ave) の最小, 最大, 平均実行時間

来的にはハイブリッドモデルなどによるソルバの開発が必要とされる。

参考文献

- 1) : <http://icl.cs.utk.edu/papi/>.
- 2) : <http://www.hpcs.cs.tsukuba.ac.jp/omni-openmp/>.
- 3) Boku, T., Yoshikawa, S., Sato, M., Hoover, C. G. and Hoover, W. G.: Implementation and performance evaluation of SPAM particle code with OpenMP-MPI hybrid programming, *Proceedings of European Workshop on OpenMP 2001* (2001).
- 4) Cappello, F. and Etienne, D.: MPI versus

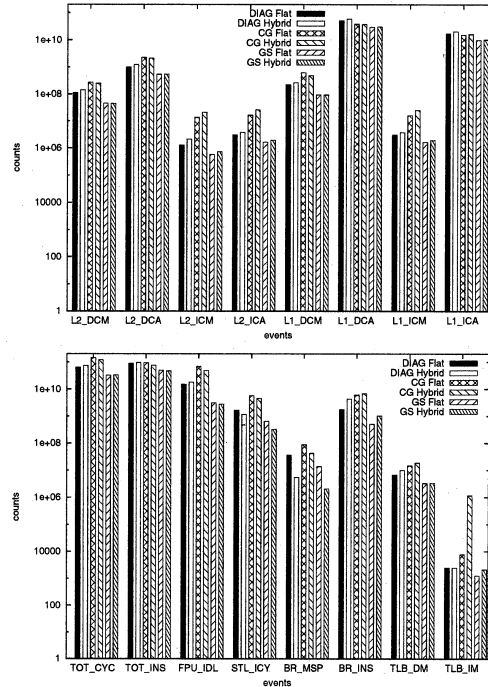


図 10 PAPI によるカウント

MPI+OpenMP on IBM SP for the NAS benchmarks, *Proceedings of the 2000 ACM/IEEE conference on Supercomputing* (2000).

- 5) Chai, L., Gao, Q. and Panda, D. K.: Understanding the Impact of Multi-Core Architecture in Cluster Computing: A Case Study with Intel Dual-Core System, *IEEE International Symposium on Cluster Computing and the Grid*, pp. 471–478 (2007).
- 6) der Wijngaart, R. F. V. and Jin, H.: NAS Parallel Benchmarks, Multi-Zone Versions, Technical Report NAS-03-010, NASA Advanced Supercomputing Division (2003).
- 7) Dong, S. and Karniadakis, G.E.: Dual-level parallelism for deterministic and stochastic CFD problems, *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pp. 1–17 (2002).
- 8) Iwata, J., Takahashi, D., Oshiyama, A., Shiraishi, K. and Okada, S.: A program code for massively parallel first-principles calculations based on the real-space density-functional theory, (*to be submitted*), pp. –.
- 9) Lusk, E. and Chan, A.: Early Experiments with the OpenMP/MPI Hybrid Programming Model, *Proceedings of IWOMP2008*, pp. 36–47 (2008).
- 10) MJahedDjomehri, H. H.J.: Hybrid MPI+OpenMP programming of an overset CFD solver and performance investigations, Technical Report NAS-02-002, NASA Ames Research Center (2002).