

# OpenMP プログラミングと処理系に関する考察 ～ 富士通 PRIMEPOWER2000 での評価 ～

岩下 英俊 山中 栄次 堀田 耕一郎

アドバンスト並列化コンパイラ研究体  
富士通(株) ソフトウェア事業本部 開発企画統括部 基盤テクノロジー開発部

SPEC OMP2001 は OpenMP 言語で記述されたベンチマークセットであり、SMP システムと言語処理系を評価する上で今後標準的な尺度となり得るものである。我々は富士通の SMP マシンである PRIMEPOWER2000 と Parallelnavi Fortran/C/C++ コンパイラを使い、このベンチマークを実施して良好な結果を得た。この評価を通して、OpenMP プログラムではスレッド毎のメモリ獲得時の競合と小粒度の並列計算がオーバーヘッドとなりやすいことが分かった。これらの知見は、OpenMP を出力インタフェースとする自動並列化コンパイラにも活かすことができる。

## A Study of OpenMP Programming and the Language Processor

-- An Evaluation on Fujitsu PRIMEPOWER2000 --

Hidetoshi Iwashita, Eiji Yamanaka, and Kohichiro Hotta

Advanced Parallelizing Compiler Project  
Strategy and Technology Division, Software Group, Fujitsu Limited

SPEC OMP2001 is a benchmark suite written in OpenMP language specification and can be used as a yardstick for measuring shared-memory systems, which support the OpenMP API. We have got a good result of this benchmark suite using Parallelnavi Fortran and C/C++ compilers on PRIMEPOWER 2000 system, a SMP machine of Fujitsu. According to analysis of the result, we found that conflict of memory allocation between threads and fine-grain parallelization will often cause high overhead. We give some recommendations for an OpenMP programming style.

### 1. はじめに

OpenMP API は SMP を対象とする並列プログラムの記述する言語としてデファクト標準と言えるものであり、OpenMP Fortran は V1.1[1]と

V2.0[2]、OpenMP C/C++ は V1.0[3]が OpenMP Architecture Review Board から公開されている。SPEC OMP2001 は、SPEC の High Performance Group が 2001 年 6 月に公開したベンチマークセットである[4]。OpenMP に関するベンチマークセ

ットとしては他に EPCC マイクロベンチ[5]があるが、これは同期とループスケジュールのオーバーヘッドの計測を目的に作られたものであった。SPEC OMP2001 は、実アプリケーションに近いプログラムに対しての OpenMP 処理系の評価を行うことができる、初めての標準的なベンチマークセットであると言える。また、SMP ハードウェアの HPC に関する性能評価という観点でも、これまでにこのようなベンチマークセットはなかった。

我々は、OpenMP Fortran API V1.1 と C/C++ API V1.0 をサポートしたコンパイラ Parallelnavi Fortran と Parallelnavi C/C++ を開発し提供している。我々は、富士通の SMP マシン PRIMEPOWER 2000 上でこれらのコンパイラを使って SPEC OMP2001 ベンチマークを実施した。その結果は SPEC HPG を通して公開されていて、トータルで高い絶対性能と台数効果を示している[4]。しかし一部のアプリケーションでは十分な台数効果が得られていない。これは必ずしも我々の実装に固有な問題ではなく、他社の結果でも同じような傾向が見られることが多いことから、OpenMP の実装に共通した問題、または OpenMP プログラミングの一般的な問題がかなり隠れているのではないかと考えられる。

本稿では、我々のベンチマークテストの結果を元に、OpenMP プログラミングで陥りやすい問題点と、実アプリケーション対応で必要となる OpenMP コンパイラの機能を考察する。続く 2 章で SPEC OMP2001 の概要を説明し、3 章で PRIMEPOWER2000 と Parallelnavi コンパイラを紹介する。4 章でベンチマーク結果を示し、各論の考察を加える。5 章でまとめとする。

## 2. SPEC OMP2001 ベンチマーク

### 2.1 構成・概要

SPEC OMP2001 ベンチマークセットは 11 の科学技術計算アプリケーションプログラムから成る。うち 8 本は OpenMP Fortran API V1.1、3 本は OpenMP C API V1.0 で記述されている。データセットには M と L の 2 つのサイズがある。M サイズは 10CPU 程度の並列実行に向いていて、1CPU 当り最大 1.6GB のメモリが必要である。L サイズ

は未公開であるが、30CPU 以上の並列向けで、1CPU 当り最大 6GB のメモリが必要である。

SPEC に登録できる測定方法には base と peak の 2 通りがある。base は全アプリケーションをそのまま共通のオプションを使って翻訳実行する方法である。peak では限定された範囲でプログラムを修正し、各アプリケーションに適したオプションを使用することができる。

GAFORT 以外の 10 本のアプリケーションは、SPEC CPU2000 にあるものを元にして作られている。SPEC OMP ではアプリケーションが並列化されただけでなく、データの大きさが M サイズでも数倍になっている。M サイズで 1.6GB の仮想メモリが必要であり、200MHz の単一プロセッサの実行で 4 時間程度かかると言われている。

### 2.2 アプリケーション

APPLU、APSI、GALGEL、MGRID、および SWIM は、計算流体動力学(CFD)のアプリケーションであり、いずれも Fortran で書かれたプログラムである。APPLU は SSOR を用いた陰解法の非線形偏微分方程式ソルバ (4000 行)、APSI は汚染物質の拡散を計算する湖の環境のモデル (7500 行)、GALGEL は対流の非定常性の数値解析 (15300 行)、MGRID はシンプルなマルチグリッドソルバ (500 行)、SWIM は shallow water 方程式のソルバ (400 行) である。

その他の Fortran コードは 3 本ある。FMA3D は三次元物体の衝突のシミュレーションであり、60000 行の巨大なコードである。GAFORT は遺伝子アルゴリズムを使ったコードである (1500 行)。WUPWISE は QCD に近い分野のプログラムである (2200 行)。

C のコードは 3 本ある。AMMP は分子力学・分子動力学のシミュレーション (13500 行)、ART は熱分布から物体を認識するニューロネットワーク (1300 行)、EQUAKE は地震モデル (1500 行) である。

各アプリケーションについてのより詳細な説明は、Aslot らの論文[7]を参照されたい。

### 3. 測定環境

#### 3.1 PRIMEPOWER2000 システム

富士通 PRIMEPOWER2000 は、最大 128CPU と 512GB のメモリが搭載可能な並列計算サーバである[8]。CPU には 64 ビットアーキテクチャ SPARC V9 に準拠した独自の SPARC64 GP プロセッサを搭載して[9]、使用した環境では動作クロック 593MHz であった。

図 1 にシステムの最大構成を示す。4 台のキャビネット(ノード)はそれぞれ 8 枚のシステムボードをもち、それぞれのシステムボードは、4CPU、16GB メモリ、PCI カード 6 枚をもつことができる。これらの資源はボード内とボード間の 2 階層のクロスバススイッチで結合されていて、すべての CPU がすべてのメモリにアクセス可能であり、かつメモリの一貫性はスヌープにより保証されている。すなわち、アプリケーションから見てメモリの階層がない、単一 SMP システムが構成されている。

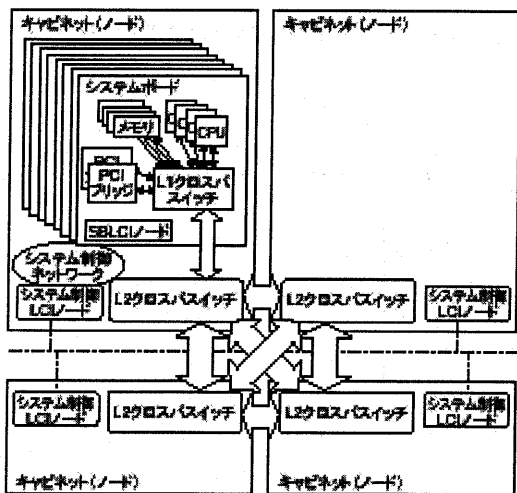


図 1 PRIMEPOWER2000 の最大構成

#### 3.2 Parallelnavi OpenMP コンパイラ

Parallelnavi は富士通の PRIMEPOWER シリーズサーバのソフトウェアパッケージであり、

Fortran と C/C++ のコンパイラ、サポートツール、数学ライブラリ、メッセージパッシングライブラリ (MPI) といった言語系製品と、多機能のジョブ管理環境を含んでいる。Parallelnavi Fortran V1.0.2 は OpenMP Fortran の V1.1 [1] をサポートしていて、次版では V2.0 [2] をサポートする予定である。Parallelnavi C/C++ V1.0.2 は OpenMP C/C++ の V1.0 [3] をサポートしていて、こちらも OpenMP C/C++ の次版が公開され次第、サポートしていく予定である。

### 4. SPEC OMP2001 測定結果と考察

我々は、Parallelnavi ソフトウェアパッケージを用い、128CPU 構成の PRIMEPOWER2000 上で、SPEC OMP2001 の M サイズの base を測定した。結果は各社の最新の測定値と共に公開されている[4]。この章では、この結果を元にいくつかのアプリケーションについて評価と考察を行う。

図 2 にすべてのアプリケーションの 18CPU から 126CPU までの測定結果を示す。縦軸の単位は SPEC 値と呼ばれ、SGI 2100 4X 350MHz R12k (CPU は 350MHz の R12000 が 4 個) での値がそれぞれのアプリケーションで概ね 1000 となるように SPEC が定めた性能の相対値である。少数 CPU での計測は、膨大なマシン時間が必要なので同一条件では行っていないが、多少測定条件の異なる事前評価では台数に対して不連続・不規則な振る舞いを示す傾向は見られなかった。

#### 4.1 WUPWISE, SWIM, APPLU, GAFORT

図 2 に示した通り、これら 4 つのアプリケーション (と後述の APSI) では、高い絶対性能とスケラビリティが得られることを確認した。これらは PARALLEL DO 指示文または PARALLEL 指示文と DO 指示文の組による記述だけでほとんどの並列性が記述されているので、コンパイラは性能が出しやすいと言える。SWIM と APPLU はもともと並列性が出しやすい性質のアプリのようである。WUPWISE では並列版の LAPACK が使われている。GAFORT は並列化のために SPEC CPU から大きな変換が施されているが[7]、我々の処理系においてもこれが有効であった。

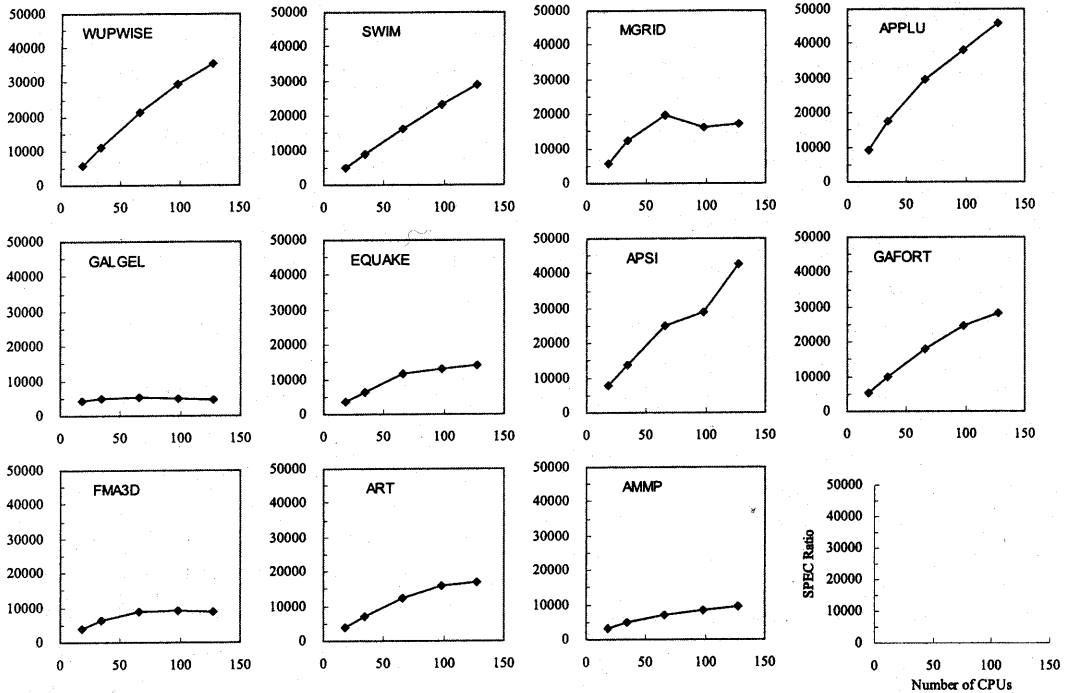


図2 Parallelnavi V1.0.2 / PRIMEPOWER2000 (563MHz) による SPEC OMP2001 の結果

#### 4.2 APSI : スレッドスタックの効果

APSI もまた、結果的に高い性能を得たアプリであるが、事前評価ではスケーラビリティが非常に低い値を示していた。調査の結果、APSI では並列実行中に巨大な配列の割り付け・開放を非常に頻りに繰り返している、その処理がスレッド間で激しいロック競合を起していることが分かった。これはメモリ割り付け・開放の資源の一部がスレッド間で共有されているためであった。幸いこの割り付け・開放が行われるのは Fortran90 で言う自動割り付け配列だけであったので、スタックで実現することができ、また、我々のコンパイラはそのような実装を選ぶオプションを以前の版から持っていたので、翻訳時オプションの選択だけで問題を回避することができた。このオプションの効果は非常に大きく、それまで並列効果の出なかった APSI が、非常に高いスケーラビリティを示すようになった (図3)。

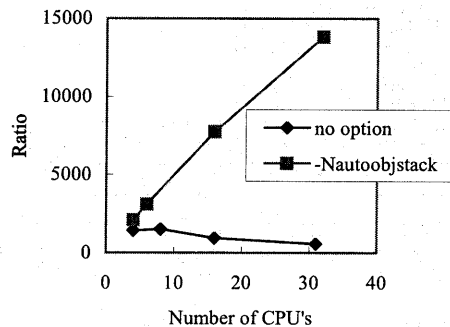


図3 割り付け・開放の実装による性能差 (APSI)

#### 4.3 GALGEL : 逐次実行による最悪値保証

GALGEL は SPEC OMP2001 の中で最もスケーラビリティの悪いアプリケーションだった。プログラム中に約 90 箇所の PARALLEL DO 構文があ

るが、その中で性能上問題となるのは、特にファイル `lapak.f90` の中に含まれるループである。典型的な2つの例を図4に示す。これらは数行の代入文しか含まず、DO文のネストもない。このようにループ内の実行コストが軽いループでは、並列スレッドの生成・消滅のコストが相対的に非常に大きくなるため、ナイーブな実装では並列化するとかえって遅くなるという現象が起きる。

```

-----
!$OMP PARALLEL DO
  DO i = 1, NUMNE
    NNPSETS(i) = 0
  ENDDO
-----
(a)
-----
!$OMP PARALLEL DO PRIVATE(SUM)
  DO 140 J = 1, N
    SUM = V1*C(1, J) + V2*C(2, J) +
$      V3*C(3, J) +
$      V4*C(4, J) + V5*C(5, J) +
$      V6*C(6, J) +
$      V7*C(7, J)
    C(1, J) = C(1, J) - SUM*T1
    C(2, J) = C(2, J) - SUM*T2
    C(3, J) = C(3, J) - SUM*T3
    C(4, J) = C(4, J) - SUM*T4
    C(5, J) = C(5, J) - SUM*T5
    C(6, J) = C(6, J) - SUM*T6
    C(7, J) = C(7, J) - SUM*T7
  140 CONTINUE
-----
(b)

```

図4 PARALLEL DO の例 (GALGEL)

このような問題を回避するため、我々のコンパイラはループ内のコストを静的または動的に判断し、逐次実行を選択する (PARALLEL DO 指示に忠実に従うオプションも持っている)。逐次実行を選択する場合には、そのループで並列効果を期待することはできないが、最悪でも逐次実行と同程度の性能を保証することができる。この効果は図2に見ることができる。18CPUのスペック値4349 (4CPU基準値の4.3倍) は良好であるが、CPU数をそれ以上増加させてもほとんど性能は変化せず、66CPUあたりでピークとなっている。しかしさらにCPU数を増やしても極端に性能が

悪化しないのは、逐次実行を選択して最悪値を保証している効果である。

#### 4.4 EQUAKE : ループ交換の効果

GALGELと同様、EQUAKE (OpenMP Cで書かれている) のPARALLEL FOR 指示には計算量の少ないループがあった (図5)。しかしこのケースではコンパイラがループ交換に相当する技術を使い、ループjも並列実行部内に取り込むことができたので、128CPUに至るまで高いスケラビリティを獲得している。

```

for (j = 0; j < numthreads; j++) {
  #pragma omp parallel for private(i)
  for (i = 0; i < nodes; i++) {
    w2[j][i] = 0;
  }
}

```

図5 Parallel for の例 (EQUAKE)

#### 4.5 考察

全体として、満足のいく絶対性能とスケラビリティを得ることができた。10CPU程度までの並列処理に向くといわれるMサイズの評価で、128CPUに至るまでスケラブルであるという結果が得られたのは、PRIMEPOWER2000が実効的にメモリ階層のないSMPであることを示している。

メモリの割付け・開放に関わるロックの競合の問題は、APSIではスレッド毎のスタックに領域を確保することでコンパイラによって回避できたが、一般には避けることはできない。Fortran90の実装では、配列式、配列代入文、WHERE文、部分配列を引数にする手続呼出しなど、様々な部分でプログラマが意識しない領域割付けが発生するので、OpenMPのプログラミングでは十分に注意する必要がある。一方でコンパイラと実行時の処理系とオペレーティングシステムに求められることは、並列実行中に競合の起るメモリ割付けをできるだけ回避する技術と、並列実行中のメモリ割付けを高速化する技術である。

GALGEL と EQUAKE で見られたような、計算量の少ない PARALLEL DO/FOR 構文は、できるだけ避けるようプログラマは工夫するべきである。一つの手段として、PARALLEL DO 構文を  $n$  個連続で実行させる代わりに、大きな一つの PARALLEL 構文の中で  $n$  個の DO 構文を実行させる方法がある。この記述により、コストの大きいスレッドの生成・消滅を  $n$  分の 1 に削減することができる。可能なら DO 構文に NOBARRIER 指示を加えることにより、さらに実行時ライブラリのオーバヘッドを削減することができる。

## 5. まとめ

富士通の SMP サーバ PRIMEPOWER2000 上で Parallelnavi Fortran/C/C++コンパイラを用い、Mサイズの SPEC OMP2001 ベンチマークを実施した。ベンチマークコードに修正を加えることなく (base モデル)、高い絶対性能とスケーラビリティを得ることができた。この結果は、PRIMEPOWER2000 が OpenMP の実装に向けたフラットな SMP であることを示している。

ベンチマークの評価を通して、OpenMP によるプログラミングで注意すべき事項と、コンパイラ・実行時システムの課題が明らかになった。OpenMP プログラミングの注意事項は、そのままスレッド並列性を抽出する並列化コンパイラの課題でもある。

## 参考文献

- [1] OpenMP Architecture Review Board. *OpenMP Fortran Application Program Interface Version 1.1*, November 1999.  
(<http://www.openmp.org/specs/mp-documents/fspec11.pdf>)
- [2] OpenMP Architecture Review Board. *OpenMP Fortran Application Program Interface Version 2.0*, November 2000.  
(<http://www.openmp.org/specs/mp-documents/fspec20.pdf>)
- [3] OpenMP Architecture Review Board. *OpenMP C/C++ Application Program Interface Version 1.0*, October 1998.  
(<http://www.openmp.org/specs/mp-documents/cs-spec10.pdf>)
- [4] SPEC HPG : <http://www.spec.org/hpg/omp2001/>
- [5] EPCC Microbenchmarks :  
<http://www.epcc.ed.ac.uk/research/openmpbench/>
- [6] J.M. Bull. Measuring Synchronisation and Scheduling Overheads in OpenMP. In *Proc. Of EWOMP99, First European Workshop on OpenMP*.  
(<http://www.epcc.ed.ac.uk/research/openmpbench/ewomp.pdf>)
- [7] V. Aslot, M. Domeika, R. Eigenmann, G. Gaertner, W.B. Jones, and B. Parady. SPECComp: A New Benchmark Suite for Measuring Parallel Computer Performance. In *Proc. Of WOMPAT2001, Workshop on OpenMP Applications and Tools, Lecture Notes in Computer Science*, 2104, pages 1-10, July 2001.
- [8] 泉田直樹、渡部徹、清水俊幸、市橋哲彦 : PRIMEPOWER2000/1000/800 のハードウェア、雑誌 FUJITSU, Vol.51, No.4, pp.208-213 (2000)  
(<http://magazine.fujitsu.com/vol51-4/paper03.pdf>)
- [9] 引地徹、加藤哲、大田秀信、嘉喜村靖 : 64 ビット RISC プロセッサ : SPARC64 GP、雑誌 FUJITSU, Vol.51, No.4, pp.226-231 (2000)  
(<http://magazine.fujitsu.com/vol51-4/paper06.pdf>)