2000 年 11 月 15 日

サイエンティフィック・システム研究会
科学技術計算分科会　第 2 回会合資料

# Starfire の NASA Benchmark をベースにした性能評価

北陸先端科学技術大学院大学
井口　寧、 黒川　原佳、 松澤　照男
{inoguchi,kurokawa,matuzawa}@jaist.ac.jp

## 1. はじめに

北陸先端科学技術大学院大学では、汎用計算サーバとして 1998 年 3 月よりスカラ並列システム Starfire (S-7/7000U モデル 1000) を導入している。本システムは、UltraSPARCII 250MHz 32CPU と 4GB の共有メモリがクロスバスイッチで結合された SMP システムであり、自動並列化プログラミングが容易である特徴を持つ。学内のクライアントワークステーションとバイナリコンパチブルであり、各自のワークステーションで処理できない大規模な問題を容易に実行することが可能である。

本報告では、この Starfire 上で代表的な科学技術計算ベンチマークである NASA Benchmark を実行し、デスクトップ型ワークステーション Ultra5 (GP400S モデル 5) およびベクトル型システムの Fujitsu VX-E と比較することにより、プログラムの最適化( 並列化)の手間と、得られる並列化性能について議論する。

## 2. Starfire の運用方針

北陸先端科学技術大学院大学では、 1990 年の創立以来、 "FRONTIER" と呼ばれる次世代の情報環境を目指した計算機・ネットワークを設計し、実現してきた。そのために、教官及び学生はもちろん事務部門に至るまで、一人 1 台に近いワークステーションを配置し、これらを超高速のネットワークで接続している。また、各種サーバ類は一ヶ所に配置し、全学に対してサービスを行っている。

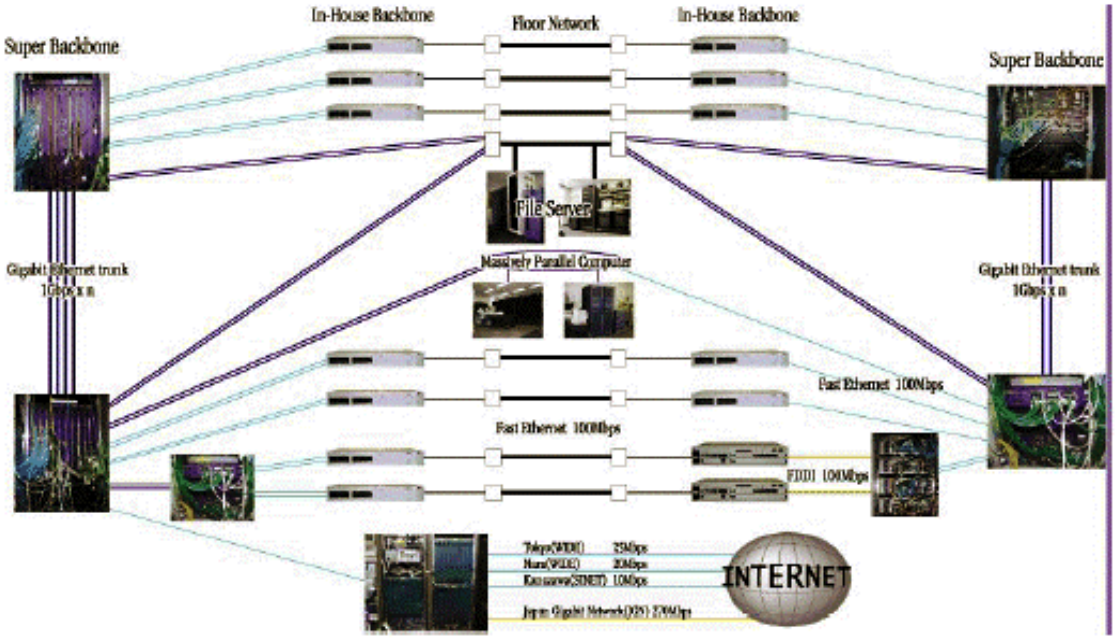


図 1: FRONTIER の構成

図 1 に FRONTIER の構成を示す。学内の教官や学生は、電子文房具として一人一台のワークステーション(Ultra5) が割り当てられており、通常の文書編集やプログラム作成は各自のワークステーション上で行なうことができる。各自のワークステーションで処理できない大規模な問題は、情報科学センター(以後

センターという) の超並列計算機やベクトル型計算サーバを用いて解くことができる環境となっている。このような利用法におけるユーザの利便性のため、各ユーザのホームディレクトリはセンターのファイルサーバに集約し、アカウントを一つの NIS ドメインとして運用することにより、各ユーザは学内のどのシステムでも透過的に利用することができる。

ユーザが利用できる計算サーバや超並列システムとして、表 1 のようなものがあり、ユーザは解く問題に応じて適切なシステムを自由に選ぶことができる。この中で Starfire はユーザのクライアントワークステーションとバイナリコンパチブルで同一の OS で稼働するため、システムに応じた最適化や再コンパイルの必要すらなく、ユーザからは極めて容易に大規模な処理を実行できるシステムとして位置付けられている。

| システム名 | CPU 数 | 特徴 |
|---|---|---|
| GP400S model 5 | 1 | ユーザのワークステーション |
| Starfire | 32 | Ultra5 と同一 OS が稼働 |
| VX-E | 2 | 疎結合型ベクトルシステム |
| Cray SV1 | 8 | 共有メモリ型ベクトル型システム |
| Cray T3E | 128 | 超並列処理研究用システム |
| IBM RS/6000 SP | 288 | 並列データベース研究システム |

表 1: FRONTIER 内の計算機システム

## 3. NASA Benchmark

NASA Benchmark について概略を述べる。NASA Benchmark は、NASA Ames Reseearch Center で開発された熱流体関連の科学技術計算(NAS:Numerical AerodynamicsSimulations) のベンチマークであり、本報告では NPB(NAS Parallel Benchmark) を用いる。NPB は、5 つの主要アルゴリズムのカーネルと 3 つの数値流体計算コード( アプリケーションコード：圧縮性流体を擬似的に計算) からなる。それぞれ、並列計算コードと逐次計算コードからなり、並列計算コードは、Fortran 77 と MPI 、逐次計算コードは、Fortran 77 で記述されている。また、計算サイズも 4 種類用意されており、用途に応じた計算サイズを用いることが出来る。計測結果として、Mop/s(Mega Operation per second) 値が得られる。この値は、ほぼ MFlop/s と同等の値である。以下にプログラムコードの概要を示す。

- カーネルベンチマーク
  - CG　正値対称大規模疎行列の最小固有値を共役勾配法により求めるプログラムである。ベクトル化や自動並列化が容易である。
  - EP　乗算合同法によって一様、正規乱数を生成するプログラムである。並列で解く際に通信がほとんど発生しない。ベクトル化や自動並列化が難しい。
  - FT　FFT を用いて三次元偏微分方程式を解くプログラムである。ベクトル化や自動並列化が難しいと考えられる。
  - IS　大規模な整数値ソートを行うプログラムである。ベクトル化や自動並列化が難しいと考えられる。
  - MG　三次元ポアッソン方程式をマルチグリッド法によって求めるプログラムである。ベクトル化は容易であるが、自動並列化は難しい。
- アプリケーションベンチマーク
  - BT　ブロック 3 重対角方程式を ADI 法を用いて解くプログラムである。
  - LU　上下三角行列を対称 SOR 法を用いて解くプログラムである。
  - SP　5 重対角方程式をスカラーADI 法を用いて解くプログラムである。

以下では、問題サイズを CLASS W と A を用い比較検討を行う。CLASS W は小規模サイズ、CLASS A は中規模サイズである。例えば MG で、CLASS W は格子点数 $64^3$、 CLASS A では格子点数 $256^3$ である。

## 4. 自動並列化コンパイラ

次に、Starfire でのコンパイラの種別によるスカラー性能を比較する。図 2 では、Starfire の 1CPU を用いて、問題サイズ CLASS A の NPB 逐次版により、次の 3 通りのコンパイラおよびコンパイルオプションで計測を行なった結果を示す。

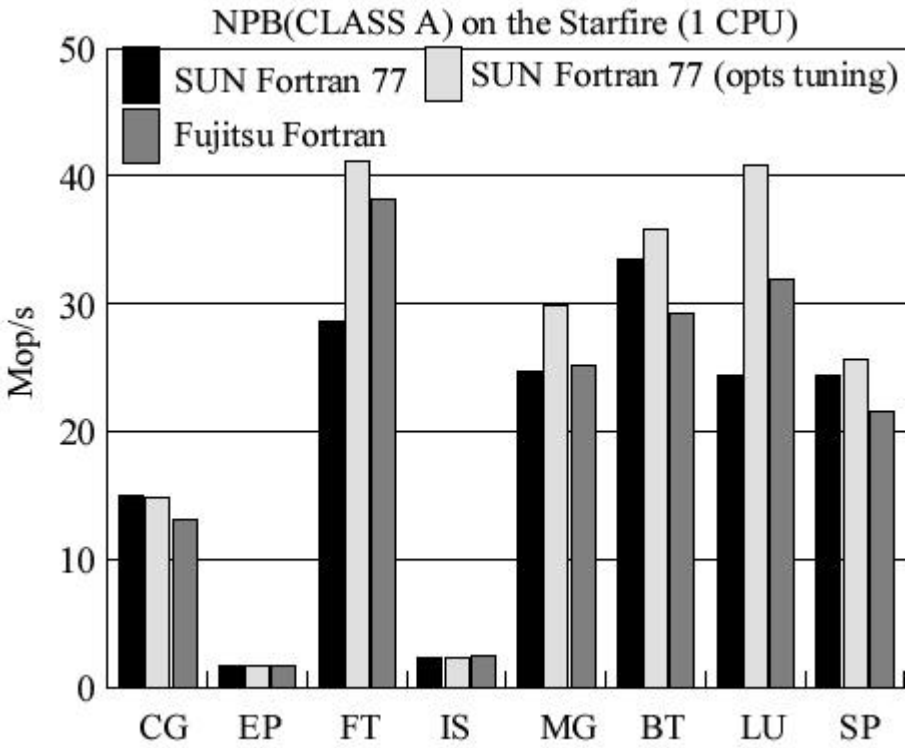

図 2: 自動並列化コンパイラ

1. Sun Fortran 77 Ver 5.0,
   option: -fast
2. Sun Fortran 77 Ver 5.0,
   option: -fast -xchip=ultra2 -xcache=16/32/1:1024/32/1 -xarch=v8plus -depend -lmopt -lsunmath
3. Fujitsu Fortran Ver 4.0,
   option:-Kfast

3. の Fujitsu Fortran の場合、コンパイルオプションのチューニングを行なわないのであれば、Sun Fortran 77 よりも良い性能が得られることが多い。また、Sun Fortran 77 を用いてコンパイルオプションのチューニングを行うことにより、CG、EP、IS を除き最も良い結果が得られた。このため、以後の Starfire 及び Ultra5 での計測には、2. の Sun Fortran77 コンパイラおよびコンパイルオプションを用いる。

## 5．自動並列化演算性能

自動並列化演算性能について検討するため、Starfire 、Ultra5 および VX-E を用いて、問題サイズ Class W の NPB により計測を行なった。測定に用いたシステムの仕様は次の通りである。

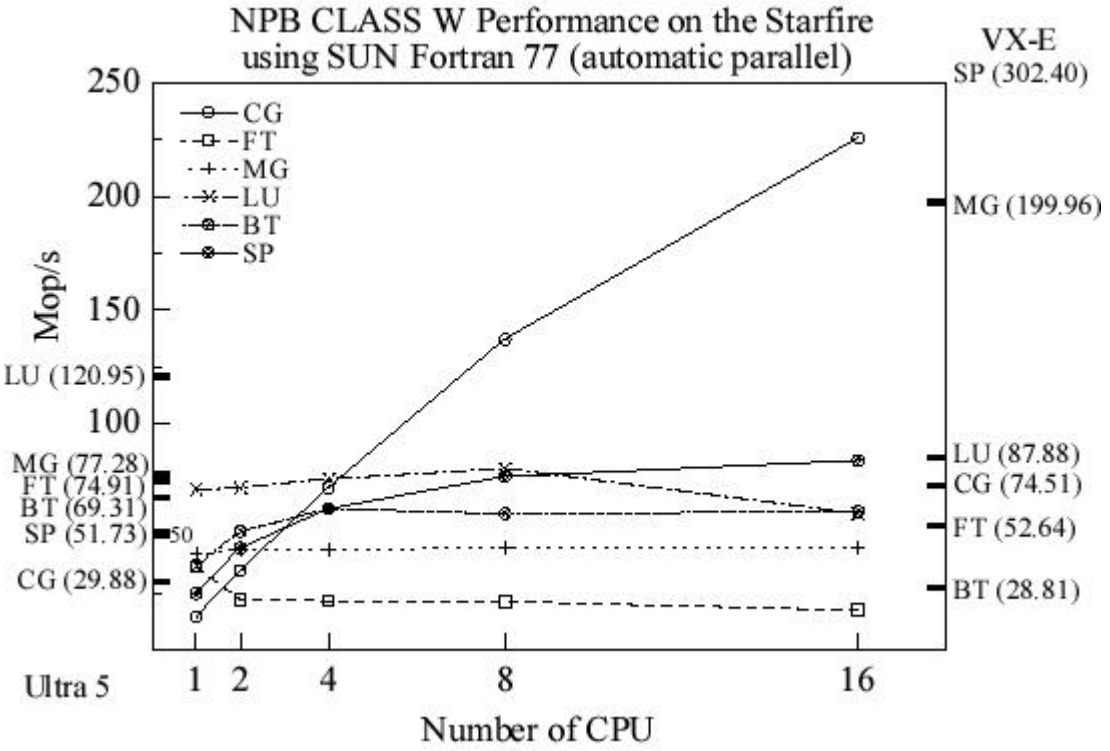


図 3: NPB 結果(Class W)

| | |
|---|---|
| Starfire | 250MHz Ultra SPARC II × 32CPU、<br>1 次キャッシュ 16+16KB(I+D) 、2 次キャッシュ 1MB 、<br>4GB Memory via X-Bar SW. |
| Ultra5 | 333MHz UltraSPARC IIi<br>1 次キャッシュ 16+16KB(I+D) 、2 次キャッシュ 2MB 、<br>128MB Memory |
| VX-E | 2.4GFLOPS、2GB Memory |

図 3 の左縦軸に Ultra5 の測定結果、右縦軸に VX-E の測定結果を示す。中央は自動並列化コンパイラ(Sun Fortran77) を用いた Starfire による測定結果であり、横軸が使用 CPU 数、縦軸が演算速度となっている。Ultra5 、VX-E は、1CPU での Mop/s 値を、Starfire では、自動並列を行い、それぞれの CPU 数における Mop/s 値を示す。Ultra5 では、Starfire と同一の Sun Fortran 77 とコンパイルオプションを用いているが、キャッシュ値だけはUltra5 の仕様に合致させた。VX-E では、コンパイルオプションに -Kfast、VX -ilfunc を用いた。

図 3 より、CG は自動並列化の効果が非常に高いことが分かる。CG は、ループ内に依存関係が存在しないことがコンパイラにより容易に( 指示行なしで) 解釈され、自動並列化が行いやすいプログラムであるため、自動並列化によって高い効率が得られている。一方、SP 、MG は、ベクトル計算機(VX-E) で非常に高い性能が得られたが、スカラ計算機(Starfire、Ultar5) では、それほど高い性能は得られなかった。SP 、MG は、最内ループのベクトル長が長く、ベクトル計算に適したプログラムコードであるため、ベクトル計算機での実行時に高い効率が得られたものと推測される。

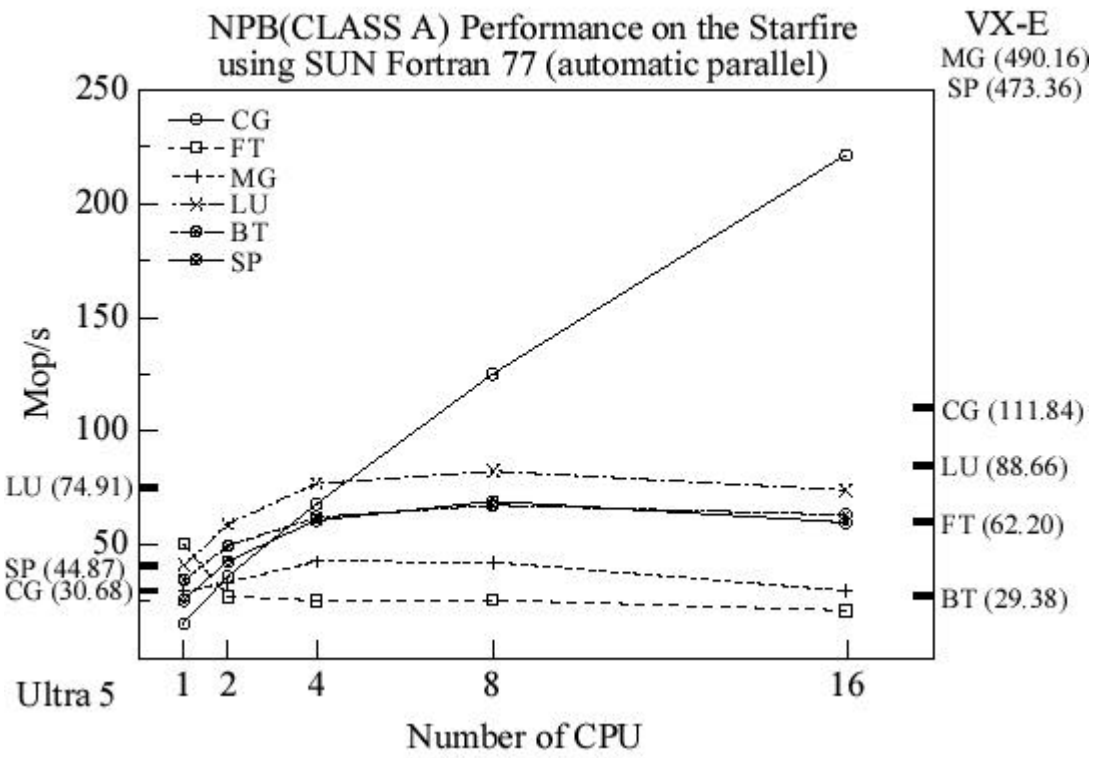

図 4: NPB 結果(Class A)

LU、BT、SP、MG は、CG よりも並列化の効率は低いが、8CPU 程度まででは並列化により実行速度が向上する。4CPU までの場合は、実行速度の向上は顕著である。これらのアプリケーションでは、本質的には並列化可能であるが、CPU 数が大きくなると、計算量よりも同期処理の時間が大きくなり、並列化効率が低下する。

BT については、ベクトル計算機よりスカラ計算機の方が良い結果が得られているが、これは最内ループのベクトル長が非常に短いものが多く、問題量によってもそのベクトル長に変化がない。このためベクトル化の効率が低く、スカラ計算機での実行の方が良い結果が得られたものと考えられる。

Starifre における FT では、CPU 数を増やすとむしろ演算性能が低下する現象が見られる。これは、FT コード自体が自動並列されにくいことと、自動並列されるとしても、多重ループでは変数の共有があり並列化できず、変数の共有が無い最内ループによる並列化が行なわれるため、実行時の同期遅延が多数入ることにより、CPU 数が増加した時の並列化効率が極端に低下するためと推測される。

Starfire の 1CPU と Ultra5 を比較すると、CPU クロックが Ultra5 の方が速く(250MHz と 333MHz)、また、CPU 当りの 2 次キャッシュ容量が Starfire の 1MB に対して Ultra5 は 2MB あるので、効率良く並列化できないのであれば Ultra5 の方が高速に実行できる結果となっている。

図 4 は、図 3 よりも計算規模が大きい( クラス A)。そのため、FT、MG および BT は Ultra5 の主記憶容量では計算出来ない。自動並列化の効果は、クラス W( 図 3) とほとんど変わらないが、並列化効率については、LU、BT、SP、MG は、CPU 数が大きい場合でも、より高い並列化効率を得ることができた。また、VX-E では、BT 、FT を除いて、平均ベクトル長が長くなったため、計算効率が向上した。

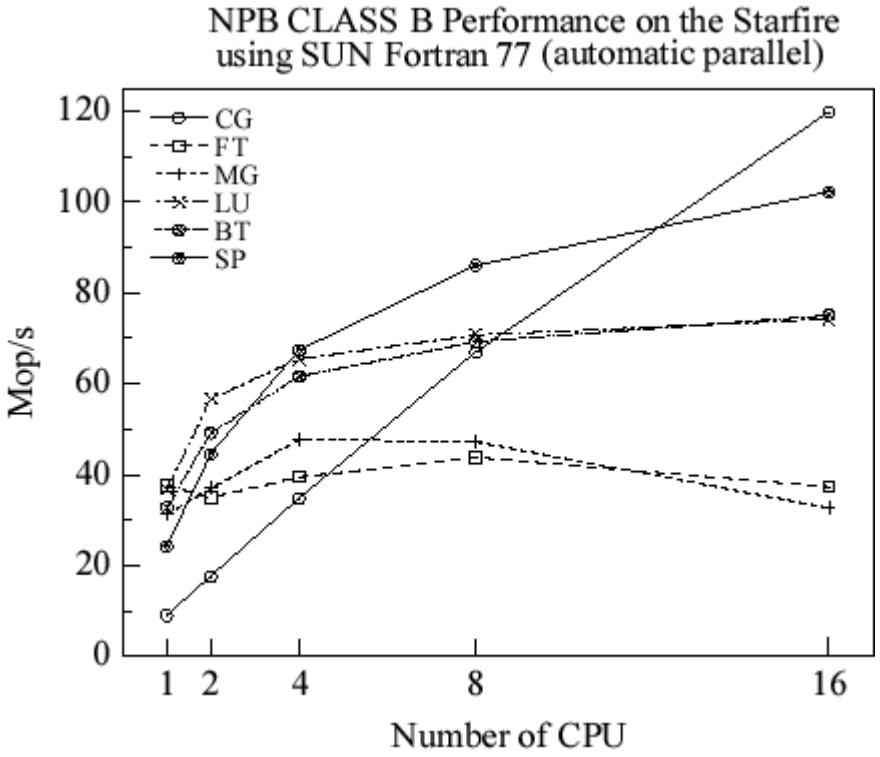

図 5: NPB 結果(Class B)

図 5 に Starfire 上でさらに大きな問題(クラス B) を測定した結果を示す。問題サイズが大きすぎると 1CPU での性能および全体のピーク性能が低くなる。これはデータが CPU のキャッシュに乗りきらず、キャッシュあふれが発生しやすくなるためだと考えられる。一方、CPU 数を増やした時の並列化効率は、問題サイズが小さい場合よりも良い傾向が見られる。1CPU 当りの処理性能が低下した分、並列化した場合の同期処理などのオーバーヘッドの割合が、見かけ上少なくなるからである。SP については、データをブロック化せずに計算を進めるため、ブロック化よりもデータの先読みが有効に機能するものと考えられ、問題サイズが大きい場合には高い並列化効率が得られている。

## 6．自動並列化と MPI

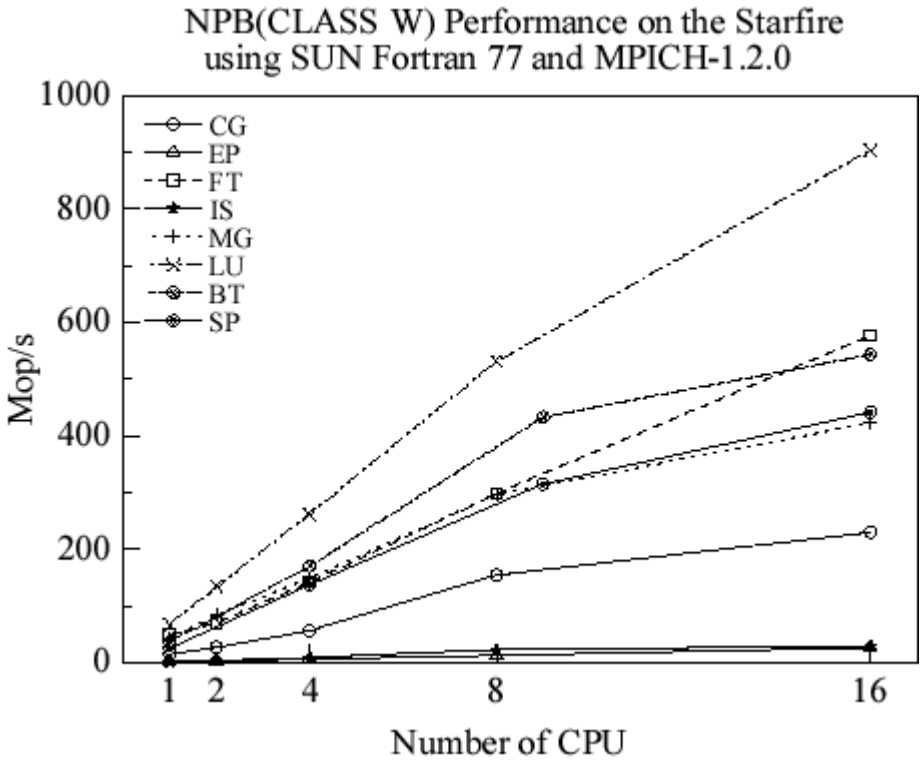


図 6: MPI による性能(Class W)

次に、同じプログラム(NPB) を MPI により実装した場合の並列化効率について議論する。図 6 に、並列(MPI) 版 NPB(CLASS W) を、Starfire 上で MPICH-1.2.0 と Sun Fortran77 により実行した結果を示す。これら全てのプログラムは、MPI 化する( 並列アルゴリズムを用いる) ことによって、かなりの性能向上が達成できていることが分かる。自動並列で性能が得られないプログラムコードも、並列化を行えば Starfire で十分に高速演算が可能である。特に MG、FT の場合、自動並列化では並列化の効果が殆んど見られなかったが、MPI によるプログラムではかなり効率の良い並列化が行なわれている。また、MPI での並列化の効果が低い場合でも、自動並列化で見られたような、CPU 数を増やすと逆に性能が低下する現象は起っていない。

図 3 のベクトル型計算機と比べると、ベクトル型計算機で高い性能が得られていた MG と SP が 16CPU の Starfire とほぼ同じ性能、FT、LU および BT では 10 倍以上の性能が得られていることが分かる。並列化が効率良く行なえれば、ベクトル型計算機よりも非常に有効であることが言える。

図 7 に NPB 並列版(CLASS A) の結果を示す。図 6 に較べ問題サイズが大きい場合、計算時間に対する通信によるオーバーヘッドの割合が小さくなり、さらに大きな性能向上が得られていることが分かる。

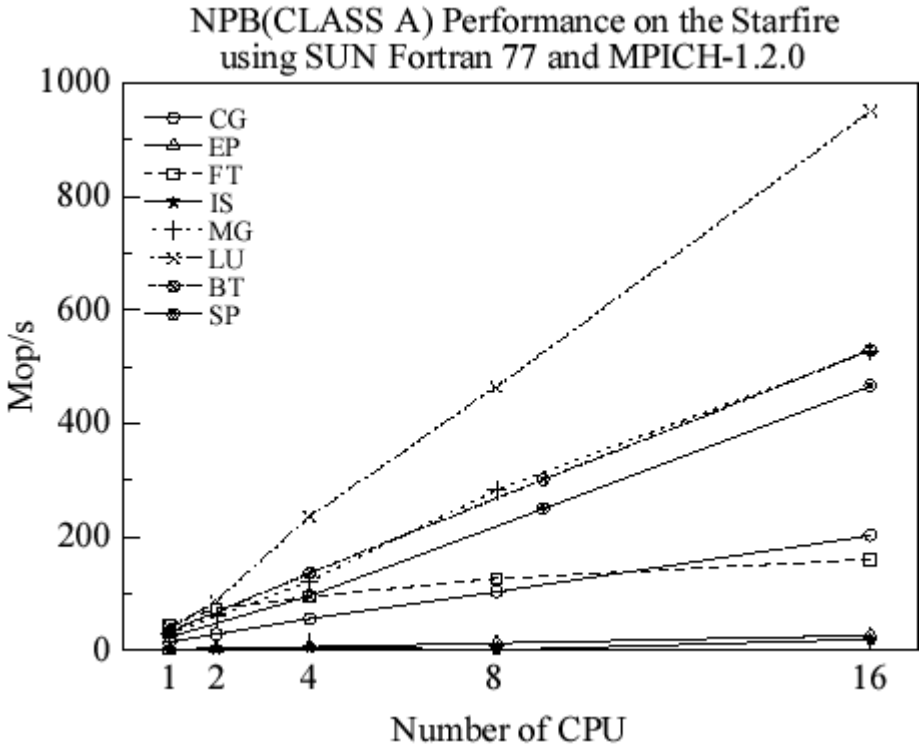


図 7: 自動並列化と MPI の性能(Class A)

## 7. MultiThread による実行性能

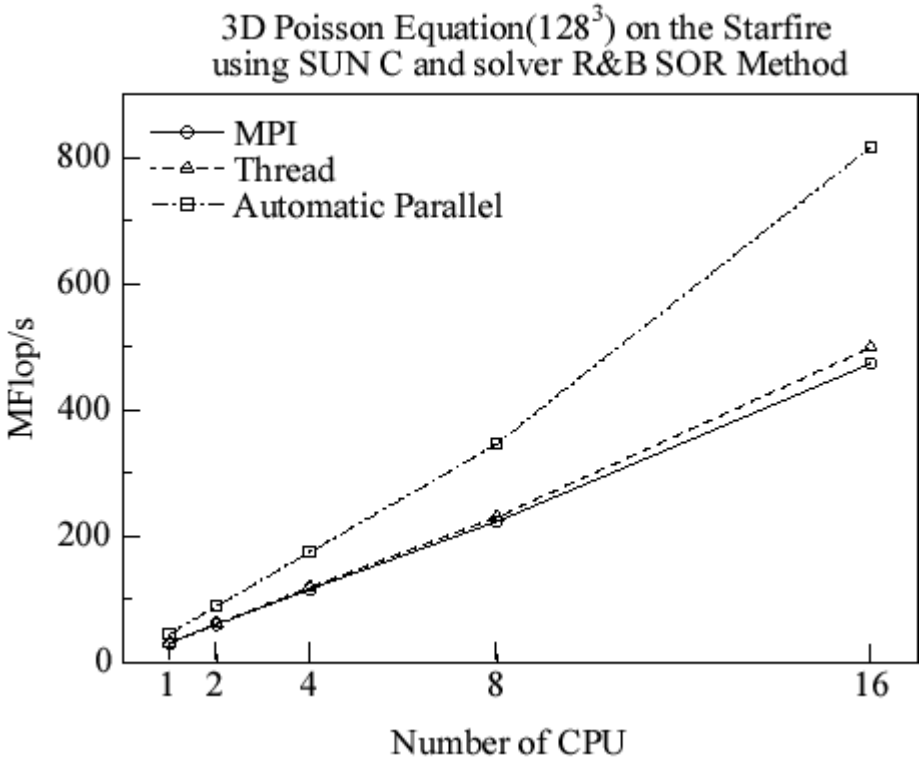


図 8: MPI と MultiThread の比較(3 次元ポアッソン方程式)

MultiThread による並列化効率を議論するため、図8 に、三次元ポアッソン方程式を R&BSOR 法で解いた場合の演算速度を示す。格子点数は $128^3$、格子はカルテシアンとし、離散化は差分法( 二次精度中心差分) を用いた。また、並列化の手法として自動並列化、MPI 化、MultiThread 化を用いた。

R&B SOR 法は、並列化指示行を挿入することで自動並列化が可能となる。また、MultiThread 化には、バリア同期(Dissemination) を用い、1 反復中に三カ所用いた。そして、誤差の足し込みのため、mutex lock()、mutex unlock() を用いた。MPI 化は、同期通信を用い、1 反復中に隣接領域と 4 回のデータ交換を行い、誤差の足し込みに MPI All reduce()を用いた。

並列化手段として、MultiThread を行った場合、わずかではあるが、MPI による並列化よりも良い結果が得られた。MultiThread における同期待ちのコストは、MPI の場合の通信コストよりも低く押さえることが可能である。そのため、SMP マシンでは、MultiThread 化することで、MPI での並列性能よりも良い性能が得られる。

自動並列化の場合、高い演算性能が得られているが、本ベンチマークでは自動並列化が極めて容易であり、MultiThread や MPI のサブルーチンコールが無い分、1 CPU あたりの性能が高い。また、CPU 数が増えた場合でもその効果が現われている。また、8 CPU から 16 CPU でスーパーリニアとなっているが、16 CPU の場合、計算すべきデータ量がキャッシュサイズに収まるために現れたと考えられる。

## 8．まとめ

本報告で用いたアプリケーションは、並列化の難易度について次の 2 つに大別できる。

1. 自動並列化により高い並列化効率が得られるもの
2. MPI を用いた手動並列化により高い並列化効率が得られるもの

自動並列化で高い並列化効率を得るためには、ループ内に相互依存が全く無い場合のように、自動並列化が容易なプログラム構造である必要がある。その場合、手動による並列化よりも、MPI などの通信サブルーチンを呼ばなくて済むため、自動並列化の方が高い並列化効率が得られる。一方、自動並列化により処理速度が向上しない場合でも、MPI や MultiThread などの通信ライブラリを用いて並列化することにより、高い並列化効率を得ることができる。このとき、ベクトル向きのプログラムでも、ベクトル型計算機による実行とほぼ同等の性能が得られ、ベクトル型計算機に向かないプログラムの場合では、16CPU のシステムで 10 倍程度の速度が得られることが分かった。

SMP 型の並列計算機は、従来のエンドユーザ向けのワークステーションと極めて高い親和性を保ちつつ、高い処理性能が得られるシステムとして、非常に有意義であると言える。