

共有メモリ型並列マシン (Symmetry S81) と
SIMD 型超並列マシン (MasPar MP-1) の
連続鋳造設備モールド部温度計算への適用並びにその評価

大城 卓¹ 喜連川 優

新日本製鉄君津製鉄所 東京大学生産技術研究所

製鉄プロセスの一部である連続鋳造設備のモールド部温度計算という実際の応用を対象に、バス結合共有メモリ型並列計算機 (Symmetry S81 (17CPU 並列)) と SIMD 型超並列計算機 (MasPar MP-1 (4096PE 並列)) と、
いうアーキテクチャの異なる商用機を用い、対象プログラムの並列化を行い、各マシンのプログラミング性、処理能力の比較を行った。

Applying Shared-Memory Parallel Computer Symmetry S81
and Massive Parallel SIMD Computer MasPar MP-1
to computing mold temperature distribution in a Continuous Casting ,
and their evaluations.

Takashi Oshiro Masaru Kitsuregawa

Nippon Steel, Kimitsu Works
University of Tokyo, Institute of Industrial Science

Kimitsu Kimitsu-City Chiba Japan
Roppongi Minato-ku Tokyo Japan

With an application program which computes the distribution of temperatures at the mold in a Continuous Casting in Steel-Making Plants, we evaluated the difference between two commercial parallel computers which are Shared-Memory Parallel Computer Symmetry S81 , and Massive Parallel SIMD Computer MasPar MP-1 in their programming manner and their ability .

¹現 新日本製鉄エレクトロニクス研究所 勤務

1 はじめに

種々の並列コンピュータの商用化が進んでいるが、どの様な分野にどの様なマシンが適しているのか明らかでない部分が多い。並列処理の研究に於ては、実際のアプリケーションがどの様な形態の並列度を有するかに関する情報の欠如が広く指摘されている。各種アプリケーションに関し1つ1つ並列化の有効性を検討してゆくことが不可欠である。ここでは連続鋳造設備モールド部温度計算という1つの応用を取り上げて、共有メモリ型並列マシン (Sequent 社製 Symmetry S81) と SIMD 型超並列マシン (MasPar 社製 MasPar MP-1) の評価を行ったので報告する。

2 連続鋳造設備におけるモールド部温度計算

2.1 計算モデル

製鉄プロセスの一部である連続鋳造設備は、溶鋼（溶けた鋼）を鋳型（モールド）上方より連続的に流し入れ、モールドを通過中に冷却しその下部から冷え固まった鋼片を連続的に引き出す設備である。この後鋼片は圧延され最終製品となる。連続鋳造設備の中でもモールド部は溶鋼が冷却によって周辺部から凝固して行く場所であり、高品質の鋼片を安定的に製造するためにはモールド部分の凝固状態を巧く制御する必要がある。（図1）

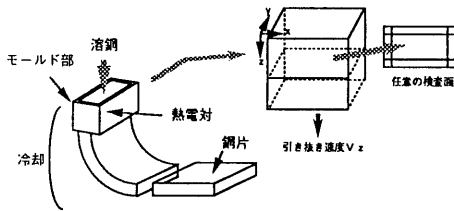


図1: モールド部と計算点

凝固状態を知るためにモールド部には数組の熱電対が埋め込まれており、そこからの温度データをもとにモールド内面の温度を計算し、さらにモールド内部を通過している鋼の温度計算を行う。モールド内部の鋼の温度分布を求めるためには、モールドをz方向に分割した各x、y平面（検査面、図1）内の2次元の温度分布を計算し、それをz方向に積み上げて全体の温度分布とする。検査面内の任意の計算点の温度については、各点の前の温度により固相状態、固液共存相状態、液相状態の3つに分け、表1の各状態に対応した式（固相状態(1)、固液共存相状態(2)、液相状態(3)）を用いて計算を行う。

ここで θ_{ij}^k は、任意の検査面のx、y方向に各々i、j番目の計算点の、時刻kにおける温度であり、 Δx 、

表1: モデル式

$$\rho \gamma \frac{\theta_{ij}^{k+1} - \theta_{ij}^k}{\Delta t} = \lambda \left\{ \frac{\theta_{i-1j}^k - 2\theta_{ij}^k + \theta_{i+1j}^k}{\Delta x^2} + \frac{\theta_{ij-1}^k - 2\theta_{ij}^k + \theta_{ij+1}^k}{\Delta y^2} \right\} \quad (1)$$

$$\rho \left\{ \gamma + \frac{\kappa}{\varphi_1 - \varphi_3} \right\} \frac{\theta_{ij}^{k+1} - \theta_{ij}^k}{\Delta t} = \lambda \left\{ \frac{\theta_{i-1j}^k - 2\theta_{ij}^k + \theta_{i+1j}^k}{\Delta x^2} + \frac{\theta_{ij-1}^k - 2\theta_{ij}^k + \theta_{ij+1}^k}{\Delta y^2} \right\} \quad (2)$$

$$\theta_{ij}^{k+1} = t_{in} \quad (3)$$

Δy は計算点のx、y方向の計算点の間隔である。

モールド内の鋼はz方向に速度 V_z で、引き抜かれているので、z方向の間隔 Δz と時間間隔 Δt の間には $\Delta z = V_z \cdot \Delta t$ なる関係が成り立つ。

また φ_1 、 φ_3 は各々液相～固液共存相、固液共存相～固相との界面温度、 ρ 、 γ 、 κ 、 λ は各々密度、比熱、潜熱、熱伝達率である。 t_{in} は溶鋼温度である。

2.2 プログラム構造

図2にモールド部温度計算プログラムの全体構造を示した。

前述した温度計算を行うために、x方向に75、y方向に64そしてz方向に153分割し、計734400点の計算点を割り当てている。

このプログラムはFortranで記述されており、ICPUマシンでの動作確認がなされている。

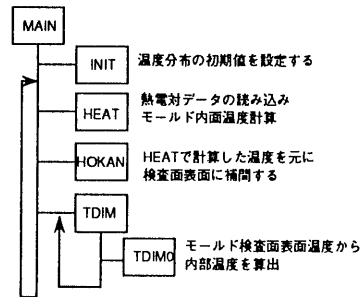


図2: モールド部温度計算プログラム全体構造

モールド部温度計算プログラムの中でも温度計算の主要部であるTDIM0を対象に並列化を行い、その処理時間を測定することで処理能力の評価を行う。TDIM0のプログラムを図3に示す、ここで $tmp1(ix, iy, iz)$ は今回計算する温度、 $tmp0(ix, iy, iz)$ は前回計算した温度であり、使用している計算式は各々表1の固相状態(1)

、固液共存相状態 (2)、そして液相状態 (3) の各計算式に対応している。

```

do 10 iz = 1, 152
  do 10 iy = 1, 62
    do 10 ix = 1, 73
      if tmp0(ix, iy, iz) < 凝固点温度 then
        % 固相状態の計算 %
        tmp1(ix, iy, iz)
          = tmp0(ix, iy, iz - 1)
            + ((tmp0(ix - 1, iy, iz - 1)
              - 2.0 * tmp0(ix, iy, iz - 1)
              + tmp0(ix + 1, iy, iz - 1)) / dx ** 2.0
              + (tmp0(ix, iy - 1, iz - 1)
              - 2.0 * tmp0(ix, iy, iz - 1)
              + tmp0(ix, iy + 1, iz - 1)) / dy ** 2.0) * dt * aa
      else if tmp0(ix, iy, iz) < 溶解温度 then
        % 固液共存相状態の計算 %
        tmp1(ix, iy, iz) = 略
        ( 固相状態と定数が異なっているのみ )
      else
        % 液相状態の計算 %
        tmp1(ix, iy, iz) = tin
    10 continue
  
```

図 3: TDIM0 のプログラム

3 共有メモリ型並列マシンと SIMD 型超並列マシン

3.1 Symmetry S81

Symmetry S81 は、米国 Sequent 社が 1987 年に発表・発売した商用バス結合共有メモリ型並列計算機である。

図 4 に Symmetry S81 のシステム構成を示した。

CPU には 16MHz で動作するインテル 80386² を使用し、浮動小数点演算プロセッサ Weitek 1167 を付加できる。CPU の最大実装可能数は 30 であるが、評価に用いたマシンは浮動小数点演算プロセッサを付加した CPU を 18 個実装している。

CPU 毎に Copy-Back 方式の cache が 64KB 用意されている。

システムバスはデータ 64bit、アドレス 32bit のマルチブックス方式 80MB/sec の同期式バスであり、パイプライン処理によって 53.4MB/sec の転送速度を有している。加えて SLIC(System Link and Interrupt Controller) バスと呼ぶビットシリアルバスによって割り込み制御やプロセッサ間通信を行う。

主記憶は最大 240MB まで実装可能であるが評価に用いたマシンは 40MB 実装している。

²現在は 80486 を使用している

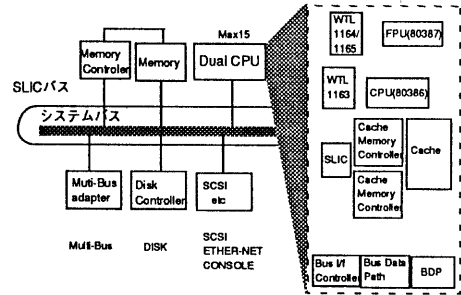


図 4: Symmetry S81 システム構成

並列処理プログラミング環境としては、C、Fortran、Pascal プログラムで利用可能なパラレル・プログラミング・ライブラリと、パラレル ADA と Fortran の 2 つの並列化コンパイラが用意されている。さらに dbx を拡張した pdbx というパラレルデバッガ等がプログラム開発のために提供されている。

3.2 MasPar MP-1

MasPar は、米国 MasPar 社が 1990 年に発表・発売した商用 SIMD 型超並列マシンである。

SIMD 型超並列計算機の動作等については参考文献 [9][10][11][12] に詳しい。ここでは MasPar に特徴的な事項について述べる。

図 5 に示したように MasPar は、データ並列処理を行う DPU(Data Parallel Unit) と逐次処理とプログラム環境等を提供するフロントエンド (VAX Station 3520³) の 2 つから構成される。さらに DPU は、データを並列に処理する PE(Processor Element) Array と、PE Array を制御しかつ逐次処理部分を司る ACU(Array Control Unit) からなる。

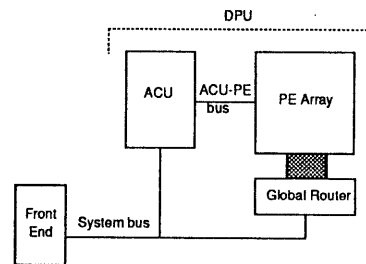


図 5: MasPar のシステム構成

ACU は 32bit のオリジナル RISC プロセッサ 1 個からなり、128KB のデータメモリと実装 1MB で仮想

³フロントエンドの処理能力向上のため現在では Dec Station 5000 が採用されている

記憶方式を用いることで最大 4GB まで利用可能な命令メモリを持つ。PE Array は 4bit のオリジナル RISC プロセッサからなりメモリ 16KB⁴を持つ。PE に使用されている RISC プロセッサは 4bit 加算を 1 マシンサイクル (80ns) で 32bit 加算を 8 サイクルで実行する。

PE と ACU は図 5 のように ACU-PE bus で接続されており、PE 同士は X-Net と Global Router と呼ばれる 2 つのネットワークで接続されている。

図 6 に示したように、PE-Array は、各 PE が 2 次元 8 方向が結合された構造になっている。これを X-Net と呼ぶ。X-Net を用いた X-Net 通信は、1 つの PE と隣接する 8 つの PE との通信を行い、各 PE に対して独立動作が可能である。そのデータ転送速度は 1 マシンサイクル (80ns) で 1bit である。さらに X-Net はトロイダル・ラップ構造になっており、例えば最右端にある PE がその右側にある PE に対して通信するとき、その相手先は最左端の PE となる。同様なことは上下間でも言える。

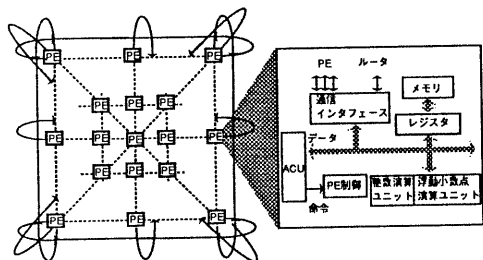


図 6: PE Array 構成

Global Router は 3 段のクロスバースイッチで構成され、任意の PE 間の通信を行うことができる。しかしながら同時に 1024 本のリンクしか動作できず、データ転送速度も最大 1.3GB/sec と X-Net の通信速度と比較して 10 分の 1 以下である。

これらの通信は後述する MPL (MasPar Parallel Application Language) という言語を使用することで直接制御できる。

評価に用いた MP-1 では、PE を 1K (1024)、2K (2048)、4K (4096)、8K (8192)、16K (16384) の順に増設可能であるが、評価にあたっては PE を 4K (4096 個) 結合したものを使用した。

並列処理プログラミング環境として F77 に準拠し配列や並列処理用の拡張機能を取り込んだ Fortran である MPF (MasPar Fortran)、同様に拡張が施されている ANSI-C 準拠の C 言語である MPC (MasPar C)、そして MPL (MasPar Parallel Application Language) と

⁴現在データメモリについても 64KB 利用可能なものが提供されている

いう、C 言語にデータ並列の記述ならびに処理を付加しさらにデータ並列処理のライブラリをサポートした言語が提供される。MPL は、MPF、MPC と異なり直接変数を PE に割当て、PE 間の通信を制御する必要がある。さらにコネクションマシンに見られるような、実装 PE よりも大きな配列を扱うことのできるバーチャル PE の機能は有していない [12]。

また MasPar は、1 つの処理をフロントエンドと DPU に分割して実行できる。この場合フロントエンドの処理は既存のプログラミング言語で、DPU の処理は MPL で各々記述する。フロントエンド部と DPU 部とのデータ送受信や処理の呼出しが行え、2 つのプログラムをリンクして実行する。これは逐次処理に関してフロントエンドマシンの方が DPU 内の ACU より処理能力が高いことによる。

さらに開発支援環境として MPPE (MasPar Programming Environment) と呼ばれるシンボリックデバックやデータ可視化機能をもった環境が提供されている。

4 並列化アプローチ

2 で述べたように、各計算機上で TDIM0 部分を並列実行するようにプログラムを記述する。

4.1 Symmetry S81 に於ける並列化

評価プログラムは、Sequent 社から提供されている並列処理対応の Fortran を用いて作成した。この Fortran はコンパイル時にオプションをつけることで、並列処理用のディレクティブを受け付ける。そのようなディレクティブのなかに \$DO-ACROSS があり、この指定を DO ループの前につけることによって、当該 DO ループを複数のプロセッサにより並列的に実行することが可能となる。

さらに \$DO-ACROSS はループ中の変数の性質 (local, shared, reduction, shared ordered, shared locked variables) も指定でき、別のディレクティブ (\$ORDER と \$ENDORDER, \$LOCK と \$UNLOCK) と併せて使用することでループ内のシリアル実行を制御する等様々な処理に対応できる。

\$DO-ACROSS 展開を行うと DO ループ内に実行が移る直前に複数のスレッドが生成され各スレッドが DO ループの内部を実行する。この時各スレッドは DO ループの繰り返し毎にループ変数の値を動的に取得する。変数の動的取得によって、評価プログラムのような各繰り返し処理の重みが異なっているプログラム (液相状態は、固相や固液共存相状態に比較して極めて軽い処理である) でも、各 CPU に均等に処理を割当てることが可能である。

今回は最外 DO ループ (ループ変数 iz) について \$DO-ACROSS 展開し、その内部 DO ループのループ変数 (ix , iy) 等を local 変数に設定、それ以外の変数は shared 変数に設定した。評価プログラム (図 3) から分かるよ

らに $tmp1(ix, iy, iz)$ を計算するのに計算済の $tmp0(ix, iy, iz)$ 以外参照しておらずループ内で並列実行の障害はなく、これ以外は TDIM0 以外の部分も含めオリジナルの Fortran で記述されたプログラムに手を加えることなく並列化できた。

処理時間の測定にはこのシステムに特に用意されている micro second clock を用いた、これはカウンタの値のあるアドレスのメモリの値として読みだしているので数マイクロ秒程度の誤差で時間を測定可能である。

4.2 MasPar MP-1 に於ける並列化

評価の時点で MPF⁵、MPC は入手できず、評価プログラムは TDIM0 以外の部分も含めて全て MPL で記述した。

(1) 変数の PE Array への割当

MasPar のようなデータ並列型の計算機では計算点を各 PE に割当てて各点の計算を並列に実行する。MPL では plural 宣言によって変数を PE Array に割当てることができる。評価プログラムにおいても温度計算点 (x 方向 75 点 y 方向 64 点 z 方向 153 点計 734400 点) を各 PE に割当てる。しかしながら評価に使用したマシンは 64×64 の 4096 個の PE しか実装しておらず、また MPL はパーチャル PE の機能 (コネクションマシンではサポートしている) をサポートしていないため、不足している x 方向の要素を配列を用い図 7 のように割当てた。

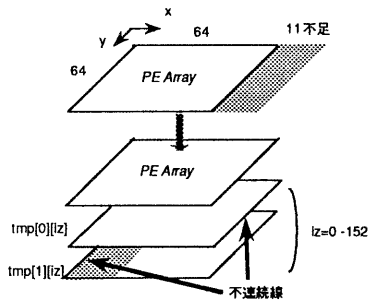


図 7: PE への割当

表 2 に Fortran と MPL の変数の対応関係をまとめた。ここで $izproc$ 、 $iyproc$ は、各々 PE の x 、 y 方向の位置を指す、システムが用意している plural 変数である。また $nxproc$ は、PE の x 方向の実装数 (今回のシステムでは 64) を保持している定数である。(y 方向にもこれに対応した $nyproc$ が用意されている。)

(2) X-Net 通信

⁵ 現在 MPF については利用可能

評価プログラムの様に、各 PE が同じように隣接する PE の変数を参照する場合には、同時に通信処理が行える X-Net 通信を使用する方が効率的である。X-Net 通信は $xnetdirection[n].variable$ のように通信相手の PE を方向 ($direction$ 、8 方向を N、NE、E、SE、S、SW、W、NW で指定) と距離 (n) で指定し、さらに使用する変数名 ($variable$ 、plural 型であることを) を記入する形で使用する。表 3 に Fortran と MPL での温度計算処理の記述の対比を示した。

(3) plural-if 文

plural-if 文は、条件を満足する PE (PE 群) に対してのみ処理を施すことができる。

図 8 に MPL で記述した TDIM0 部分のプログラムを示した。

```
for (iz = 1; iz < 153; iz++){
  if ((iyproc > 0)&&(iyproc < 63)){
    for (iix = 0; iix < 2; iix++){
      if ((ix[iix] > 0)&&(ix[iix] < 74)){
        if (ixproc == 0){ 処理 A' }
        elseif (ixproc == (nxproc - 1)){ 処理 A'' }
        else { 処理 A } } } } }
```

(処理 A は温度計算の主要部分である)

図 8: TDIM0 の MPL での記述

ここで $if ((iyproc > 0)&&(iyproc < 63))\{ 処理\}$ なる plural-if 文は y 方向 0 より大きく 63 より小さい PE に対してのみ以下の処理を施すことを意味しており、Fortran の $do 10 iy = 1, 62$ の繰り返し部分に対応している。

(4) パーチャル PE

前述のように各変数は、配列を用いて PE に割り当てたが、これによって本来連続である変数の並びに、図 7 に示すような不連続部分が生じている。

この不連続部分では、先に示した X-Net 通信を用いた隣接変数の参照 (表 3) に問題が生じる。なせならば X-Net はトロイダル・ラップ構造をしており、X-Net 通信によって同じ iix の不連続部分同士がリンクされるからである。

従って不連続部分 (PE-Array の両端に位置する PE) の処理は plural-if 文によって別途各々指定する必要がある。実際には $if (ixproc == 0)\{ 処理 A'\}$ という plural-if 文で $ixproc=0$ である PE、つまり最左端に位置する PE に対してのみ処理 A' を施す。

処理 A' では、PE がさらに左側の PE の変数を参照する $xnetW[1].tmp0[iix][iz]$ を、配列インデックス iix を 1 つ前に戻して $xnetW[1].tmp0[iix-1][iz]$ として 1 つ前の配列を参照するようにし、2 つの配列の不連続部分を

表 2: Fortran と MPL の変数対応

	Fortran	MPL
宣言	real tmp(0:74,0:63,0:152) integer ix,iy,iz	plural float tmp[2][153] int iix, iz (plural int ixproc,iyproc)
対応	tmp(ix,iy,iz) $0 \leq ix < nxproc$ $nxporc \leq ix \leq 74$	tmp[iix][iz] iyproc = iy iix = 0 ixproc = ix iix = 1 ixproc = ix mod nxproc

表 3: Fortran と MPL の処理記述

Fortran	MPL
tmp1(ix,iy,iz) =tmp0(ix,iy,iz-1) +((tmp0(ix-1,iy,iz-1)-2.0*tmp0(ix,iy,iz-1) + tmp0(ix+1,iy,iz-1))/dx**2.0 + (tmp0(ix,iy-1,iz-1)-2.0*tmp0(ix,iy,iz-1) + tmp0(ix,iy+1,iz-1))/dy**2.0)*dt*aa	tmp1[iix][iz] =tmp0[iix][iz-1] +((xnetW[1].tmp0[iix][iz-1]-2.0*tmp0[iix][iz-1] + xnetE[1].tmp0[iix][iz-1])/dx*dx) + (xnetN[1].tmp0[iix][iz-1]-2.0*tmp0[iix][iz-1] + xnetS[1].tmp0[iix][iz-1])/dy*dy)*dt*aa

正しく接続する。同様に処理 A” では最右端にある PE に対する処理を行う。(xnetE[1].tmp0[iix][iz] を xnetE[1].tmp0[iix+1][iz] とする。)

以上のように配列と plural-if 文を用いてバーチャル PE の機能を実現している。

処理時間の測定には gettimeofday というライブラリ関数を使用した。この関数は MPL 下でも動作が保証されている。測定誤差は約 0.1 秒程度であり、誤差を吸収するために実際の処理時間の測定では被測定サブルーチンを連続的に呼びその測定結果が 10 秒前後になるようにして測定を行った。

5 評価

プログラミング性と処理能力について評価する。

(1) プログラミング性

Symmetry の Fortran コンパイラは極めて記述が容易であり、従来プログラミングからの移行も容易であった。評価プログラムには必要無かったが、並列実行ループ内のデータ逐次実行部分も比較的容易に記述可能であり、このコンパイラは比較的広範に適用可能と考えられる。

MasPar は発表後間もないこともあり並列コンパイラ等言語処理系が整備されているとは言いがたい。評価プログラムも MPL というハードウェアを直接制御するプリミティブな言語にて記述したが、プログラミングに際し、言語仕様の相違点よりも計算機アーキテクチャの

相違点(データ並列処理)の方が影響が大であった。また評価プログラム中の TDIM0 以外の部分で、データ並列部分の無い処理を PE 上で行うには記述量も多く処理時間も必要とした。従って、先ず従来型言語でプログラム全体を記述し、それをフロントエンドで実行し、順次データ並列実行部分を DPU に移して行くのが効率のよいプログラム開発方法であるように思われる。またバーチャル PE の機能が早急に整備されることが必要である。

(2) 処理能力

TDIM0 の処理時間を表 4 に示した。

表 4: TDIM0 の処理時間

	MasPar (4096PEs)	Symmetry (17CPUs)	Symmetry (1CPU)
処理時間 (sec)	0.85	0.93	15.0

比較のため Symmetry の 1CPU 時の処理時間を併記した。また Symmetry の場合スレッドを生成するのに必要な時間(並列 CPU 数にほぼ比例し、生成時のみ発生)は処理時間から除いている。

Symmetry のようなバス結合共有メモリ型並列計算機の場合、バス競合が処理能力上の大きな問題である。バス競合の度合を測定するため CPU 数と処理能力の関係についても併せて測定した。測定は CPU 数を 1 から

17まで順次増加させ、その処理時間を測定する。測定結果は、1CPU時の処理能力を1とした時、n-CPU時では何倍の処理能力を示すかで表し、図9にまとめた。これから明らかなように処理能力向上はほぼリニアであり今回の評価プログラムでは17CPUまでバス競合はほとんど見られなかった。CPUの最大並列数(最大実装数は30であるが最大並列数は最大実装数 - 1 = 29である)までにバス競合が発生するか否かによってどの程度処理能力が向上するかが決まる。

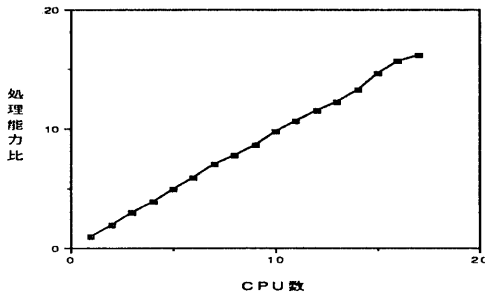


図9: CPU数対処理能力 (Symmetry S81)

一方MasParについては、実装PE数が計算点数より少ない場合に、バーチャルPEの機能を今回のようにMPLで配列とplural-if文を用いて実現する時の処理時間について考察する。

計算点が全て実PEに割当て可能であった場合の処理時間を T_a とする。

x 方向のPEが不足しこれを配列を m 回用いて処理した場合は、plural-if文の使用によって $3Ta$ (両端部の処理と中央部の処理が必要)の処理時間が必要であり、さらに配列数 m だけ繰り返しが必要なので全部で $3mTa$ ($m > 1$)だけ必要である。 y 方向についても同様に配列を n 回用いた時、同様に $3nTa$ ($n > 1$)だけ必要である。 x 、 y 両方向に配列を用いた場合の全処理時間は $9mnTa$ ($m, n > 1$)である。

前述の通りMasParの場合、PEは1K(32×32)、2K(64×32)、4K(64×64)、8K(96×64)、16K(96×96)の順に増設可能である。ここでカッコ内は x 、 y 方向のPEの実装数である。従って、今回の評価プログラムについては表5のようになることが予測される。

表5: バーチャルPEの処理時間

PE数	1K	2K	4K	8K	16K
m	3	2	2	1	1
n	3	2	1	1	1
処理時間	$54Ta$	$36Ta$	$6Ta$	Ta	Ta

Symmetryと同様に処理能力比と実装PE数の関係で表したのが図10である。

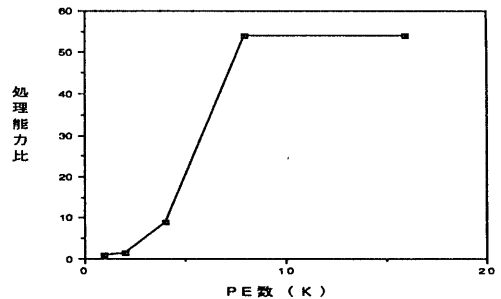


図10: PE数対処理能力 (MasPar MP-1)

このことから、MasParの場合、実PEで実行できるか否かが処理能力に大きく影響することがわかる。

評価では4K(4096)のPEが実装された計算機を使用したのが、これが8K(8192)のPEを実装する計算機で実行すれば6倍程度の処理能力の向上が期待される。

Symmetryが30CPU実装しても、現状の高々1.7倍の処理能力しか出せず、またバス競合の発生を考慮に入れると、MasParのようなSIMD型並列計算機の方が今回のような応用には適していると考えられる。

6 おわりに

様々な商用並列計算機が出現しているが、ある特定の応用という側面から並列計算機の評価を行った。SIMD型超並列マシンならびに共有メモリ型並列マシンのいずれを選択するかは応用に依存するが、今回の検討では同程度のコストで同程度の性能が偶然得られた点が興味深い。様々な応用例によって並列計算機が評価されることがユーザーにとっては重要なことで、本報告がそれに寄与できれば幸いである。

謝辞

MasParの利用に当たっては理経技術センター、および同センターの徳田氏にお世話になった。紙面を借り感謝する。

参考文献

- [1] Sequent Computer Systems, Inc.: Sequent Guide To Parallel Programming, P/N:1003-44459 Rev.A.
- [2] Sequent Computer Systems, Inc.: Symmetry Technical Summary, P/N:1003-44447 Rev.A.
- [3] Sequent Computer Systems, Inc.: Sequent FORTRAN compiler Users Manual, Rev.1.18.
- [4] 平尾 延夫: 汎用並列計算機 Symmetry/Balance, Bit Vol.21、No.4、pp597-605
- [5] MasPar Computer Corporation : MasPar System Overview , PN:9300-0100-2790, July 1990
- [6] MasPar Computer Corporation : MasPar Parallel Application Language (MPL) User Guide, PN:9302-0100-2790, July 1990
- [7] MasPar Computer Corporation : MasPar Parallel Application Language (MPL) Reference Manual, PN:9302-0100-2790, July 1990
- [8] 日経エレクトロニクス、No.503、pp.102-103 (June 1989)
- [9] 馬場 敬信: 超並列マシンへの道、情報処理、Vol.32、No.4、pp.348-364、1991
- [10] 小柳 滋、田辺 昇: 超並列マシンの実現技術、情報処理、Vol.32、No.4、pp.365-376、1991
- [11] 喜連川、湯浅 太一: SIMD 型商用超並列コンピュータとその応用、情報処理、Vol.32、No.4、pp.401-413、1991
- [12] 喜連川 訳著: コネクションマシン、パーソナルメディア,1990