OUKA
The University of Osaka
Institutional Knowledge Archive

Title	MPIを使った並列化について : SX-4およびExemplarでの演算・通信性能評価
Author(s)	日置, 慎治
Citation	大阪大学大型計算機センターニュース. 1999, 112, p. 110-114
Version Type	VoR
URL	https://hdl.handle.net/11094/66337
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

MPIを使った並列化について

～SX-4 および Exemplar での演算・通信性能評価～

帝塚山大学　日置慎治

1　はじめに

最近の大規模数値計算を考えるとき、並列計算は必須になってきている。その理由はさまざまであろうが、大別すると、

- 計算機単体では計算速度が不足
- 計算機単体ではメモリが不足
- コストパフォーマンスがよい

などとなると思われる。

第1の計算速度の問題はもっとも単純な理由と言えるだろう。現実としては、これが並列計算の大きな原動力となっているわけである。よく笑い話にするのだが、これには人間の寿命が限られてる、という避けては通れない経験事実があるからである。もし、寿命が無限にあるならば、第1の理由の存在意義はかなり薄れるに違いない。結果がでるまで釣りでもしながらのんびり待てばいいのである。とはいえ、他の研究者との論文競争ではやはり結果を相手よりも早く出さないと意味が無くなる場合があり、重要性は残る。

第2のメモリ容量の問題は、遭遇している研究者にとっては致命的なものであるのだが、そうでない研究者にとってはどうでもいい事も事実である。つまり、ある問題を解くときに必要となるメモリサイズが使用する計算機で利用可能な最大メモリサイズを越えている場合がこれにあたる。この場合には、その計算機ではどうあがいても結果を得ることはできない。(仮想記憶を使えばいいのでは？と思われるかもしれないが、ここではこれを含めたメモリサイズを越えた場合の話と考えることにする。) 結果を得るためには、以下のような事が考えられる。

a) アルゴリズムを変更するなどして、必要メモリを減らす
b) メモリが大きい他の計算機を探す、またはしばらく待つ
c) 研究対象を変える

最後の c) は冗談であるが、b) が普通だろう。例えば、メモリの大きな機種を持つ大型センターをさがしたり、または、次期機種更新までは現在できる範囲で細々と計算をいていればよい。一方計算科学的側面からは、a) は非常に面白い。同じ計算をするにしても、アルゴリズムによってメモリの必要量は違う。メモリはあまり必要でないし、計算量も少ないアルゴリズムがあればいいのであるが、大抵の場合、計算量を少なくして高速に実行可能なアルゴリズムは大きなメモリ容量を必要とするものが多いし、その逆もありえる。つまり、利用者は計算機の性格にあったアルゴリズムを選択する必要がある。

第3のコストパフォーマンスに関してであるが、何を言っているのかちんぷんかんぷんという方もおられるだろう。簡単に説明するとこういう事である。現在利用している計算機では、速度やメモリ容量に不足がでたとする。しかしながら、より高速な計算機は存在するが高価で手がでないし、メモリも購入するにはとてもお金がかかったり、増設できない機種だったりするような場合である。この場合に、現在ポピュラーになりつつある方向は、「コモディティハードウエアを用いた計算機クラスタによる並列計算」である。簡単に言えば、よく出回っている安いパソコンを複数台ネットワークで繋いで並列計算機にしてしま

おう、ということだ。パソコン用CPUおよびネットワークアダプタの高性能化と低価格化のために市販の並列計算機を購入する場合に比べて、格段によいコストパフォーマンスを実現することができるのである。

もし興味がおありならば、私が自作したパソコンクラスタを是非見て頂きたい。そして、(見ればわかるのだが) 笑って頂きたい。

`http://tupc3472.tezukayama-u.ac.jp/parabsd/index.html`

このような様々な理由で並列計算が行われるわけであるが、実際にやってみて予想した通りの結果がでるのであろうか？というのが素朴な疑問として湧き上がって来る。つまり、例えば計算速度を2倍にしようと思って、単純に2台の計算機を使ったら期待通り半分の時間で計算が終るのだろうか？

残念ながらここには落し穴があり、答えは「No」であろう。ここでの落し穴は

- 単体で利用していたプログラムはそのままの形では並列計算に向いていない。
  計算機が自動で並列化してくれればいいのであるが、この「自動並列化」が現在の技術ではまだまだ未熟であるためである。
- 仮に並列化をうまくプログラミングできたとしても、並列に分割した事に起因するあらたな仕事（通信や並列化のための準備など）が出て来る可能性がある。

ここでは、自動並列化は上の理由からあきらめ、並列化をうまくプログラミングできたと想定したときの並列化のメリットである速度向上率について報告したい。並列化を行うに当たり、計算自体の種類はもちろんのこと、単体性能と通信性能の比が重要である。このことから、阪大センターで利用できる2つの並列機、SX-4とExemplarの性能を比較しながら進めて行くことにする。

なお、利用するプログラムはMPIで完全に並列化を行っているQCDMPI (http://insam.sci.hiroshima-u.ac.jp/QCDMPI/) を用いる。

自動並列化を使用した性能評価は必要であると思うが、私には経験が浅いためにできないので、是非どなたか行って、結果を報告して頂きたい。MPIを利用した今回の結果と、自動並列化の結果は相補的なものとなり、よりいっそう総合的な性能評価ができると思われるからである。

2 単体性能評価

並列化に入る前に、単体での性能を押えておくことは重要である。昨年のモニター報告会ではSX-4の性能を評価したが、それにExemplarを加え今回新たに全て測定しなおした。

2.1 行列積演算による基本性能評価

まずは、3行3列複素行列の配列のかけ算の性能を評価する。これは、LINPACk同様性能がでやすい特徴を持つが、単純なため基本性能を測定するには最適だと思われる。

表1: 行列積の演算性能 (MFlops)

ループ長	100	250	1000	10000	100000
SX-4	1371	1639	1731	1731	1731
Exemplar	517	523	506	359	109
Ex-ratio(*)	2.65	3.13	3.42	4.82	15.9

(*)Ex-ratioはExemplar比、つまり性能がExemplarの何倍かを表す。

2.2. QCDMPI を使った実アプリでの性能評価

結果について私見を述べさせて頂くと、

- SX-4 については去年と同様の結果、というか、高性能ベクトル機としての標準的な結果となっている。つまり、ループ長が大きくなるにつれ演算性能はピーク性能に近付いて行く。
- Exemplar について、予想外に性能がよい。私自身びっくりしている。
- しかしながら、ループ長が大きくなるとキャッスミスの影響からか性能が低下するという RISC 特有の性格を表している。
- 両方の性格上あたりまえのことであるが、性能比 Ex-ratio はループ長の増加とともに増大している。

測定環境は以下の通りである。
SX-4: f77 -float1 でコンパイル
Exemplar: f77 +O4 でコンパイル、時間測定には SECNDS 利用（+E1 オプション）

2.2 QCDMPI を使った実アプリでの性能評価

ここでは、去年の報告でも利用した QCDMPI を利用して、性能評価を行う。MPI を使っているのにどうして並列でなく単体性能？と疑問に思われるかもしれないが、QCDMPI では、PE 数を指定するパラメータを 1 に変更するだけで単体で走るのだ。

単体性能比較であるので、参考のため、SX-4 と Exemplar の他にも通常我々が利用しているワークステーションとの比較をしてみると面白いだろう。ここでは Alpha マシン（センターにあるのと同じ種類、ただしクロックは異なる）を使って比較をした。結果は以下の通りである。

表 2: QCDMPI による単体演算性能 (MFlops)

マシン	SX-4	Exemplar	Alpha(*)
μ sec/link	3.9	16.9	15.2
MFlops	1465	337	375
Ex-ratio	4.35	1	1.11

(*)Alpha21164A(600MHz)

(μ sec/link とは QCD 計算において、1 自由度 (link) 当たりの演算に要した時間をマイクロ秒単位で表したもの、性能評価によく使われる。)

計算は 8^4サイズで行った。この時の主要ループ長は２０００程度である。したがって、行列積で得られていた Ex-ratio が大体今回も再現できたといえるだろう。

3 QCDMPI を使った並列性能評価

ここでは MPI による並列計算の性能評価を行う。ただし、ここでの結果を評価する際には、上の結果によって得られた情報、つまり、計算途中のループ長依存性、を十分考慮する必要がある。この点から、Exemplar に対しては、8^4および 16^4という 2 つのサイズで計算を行った。なお、SX-4 の場合はこれらにおいてあまり顕著な違いはない。

- 同じサイズ (16^4など) で PE 数を増やした場合にまず注目する
  全体として言えることは、PE 当たりの性能が徐々にではあるが低下していることである。これは並列

表 3: QCDMPI による並列演算性能

マシン	サイズ	PE 数	1	2	4	8
SX-4	16^4	全体性能 (MFlops)	1465	2879	5534	10364
		PE 当たり (MFlops)	1465	1440	1384	1288
		通信性能 (MB/sec)	—	1158	1176	515
Exemplar	8^4	全体性能 (MFlops)	337	633	1163	2036
		PE 当たり (MFlops)	337	317	291	255
		通信性能 (MB/sec)	—	121	105	70
	16^4	全体性能 (MFlops)	114	211	491	921
		PE 当たり (MFlops)	114	106	123	115
		通信性能 (MB/sec)	—	150	115	26

化によって通信が生じたために仕方がないと簡単に考えることもできる。この速度低下がない場合には、ある意味で理想的な並列マシンと言えるからだ。

しかし、通信性能に注目してみると、SX-4 と Exemplar どちらも PE 数が 2 から 8 ではかなり性能が低下していることがわかる。通信が入ったために逐次計算 (PE 数が 1) よりも効率が悪くなるのはいいとして、通信性能が PE 数を大きくした時に低下する事はどういう理由からであろうか？

この理由として 2 つ考えられる。

- 1 回あたりの通信量

  全体のサイズが同じで、PE 数が異なっている場合には、1 PE 当たりの負荷が PE 数の逆数に比例して小さくなっている事を意味する。QCDMPI ではまさにそうなっていて、しかもそれは 1 回当たりの通信量にもあてはまる。通信は一般には 1 回当たり大量に送る程、バンド幅が大きくなる傾向がある。したがって、PE 数が増えて 1 回当たりの通信量が減った分、通信性能が低下したと考えることができる。

- メモリ競合

  QCDMPI では全 PE がある時点で一斉に通信する形態をとっており、共有メモリ型計算機の場合、メモリへの競合が起こっているのではと思われる。これは分散メモリ型の場合にも、ネットワークの衝突という形で現れる現象と同じである。

- Exemplar の 2 つのサイズでの結果の違い

  一見して明らかであるが、全体のサイズが 8^4 の場合には PE 当たり 255-337MFlops なのに対して、16^4 の場合にはこれが 106-123 に半分以下の性能になっている。これはどうしたことだろう？

  答えは簡単だ、先に述べたように、PE 数が同じで全体のサイズが異なる場合には、当然ループ長が変わって来る。16^4 の場合は、8^4 の場合の 1 6 倍のループ長となり、これが RISC 特有のキャッシュミスを引き起こして性能低下を招いている。

- PE 当たりの性能低下再考

  最初に、「PE の増加に伴い、PE 当たりの性能が徐々にではあるが低下している」と述べた、そして、通信が存在するためこれは自然な結論であるかのように説明した。が、ちょっとまてよ、PE の増加に伴い、1 回当たりの通信量が小さくなるので通信性能は低下するが、一方、ループ長は短くなるはずだから、Exemplar の場合には演算性能は逆によくなるはずだ。すると、一概にはどうなるかわからないはずである。しかしながら、結果としては全体性能が低下しているので、ここから出て来る結論はこの場合には、通信性能の低下の方が全体に与える影響が大きい、という事であろう。

現実に所要時間を調べてみると、8^4を8PEで実行した場合には、通信時間が全体に占める　割合は16%になり、無視できない事が分かる。
一方のSXの場合には、ループ長が短くなる性能低下と通信性能低下が重なるため、必ず全体性能は低下するといえる。

コンパイル／リンク環境は
SX-4: f77 -P multi -G local -float1 -I/usr/include -lmpi
Exemplar: mpif77 +O4
実行は
SX-4: mpisx -p 8 -e qcd
Exemplar: mpirun -np 8 qcd
などとして行なった。

4　おわりに

現在センターで利用できる2種類の並列計算機に対する性能評価を行った。メモリモデルとしてはどちらも共有メモリ型と言われるタイプなので2つは同じように思う方もおられるかもしれないが、演算システムとして見た場合には、SX-4はベクトル計算機、ExemplarはRISC型スカラー計算機という違いがあるために、今回報告したような様々な性格の違いが見えて来る。

並列化の場合には演算システムとしての問題の他に、通信というあらたな性格が加わるために性能評価は一筋縄ではいかない場合が多い。これは何も難しいといっているのではなく、それぞれのアプリケーションに依存する部分が大きいという事を意味しているだけである。つまり、どちらの計算機が絶対的によい、という事は判断するべきではなく、自分のやりたい計算に適した方を選択すればいいだけである。

今回の結果を見れば、小規模な計算の場合には違いがあまり見えないが、大規模になるにつれSX-4の優位性が顕著になって来る。PE当たりの性能でいって、1桁違って来ている。これで勝負あった！　と性急に判断してはいけないのだ！　実はここにもう一つの評価軸が存在する。それは、価格である。つまりコストパフォーマンスを考えたときには答えはまたまた難しくなりそうだ。

この答えは、読者の方に任せることにしよう。