MPI プログラムの簡易実行による実行時間予測

岩渕寿寛† 堀井洋‡ 山名早人†

早稲田大学理工学部情報学科† 早稲田大学大学院理工学研究科‡

1 はじめに

本稿では、MPI プログラムを複数の PU を用いて実行したときの実行時間を、短時間で精度良く予測する手法について論じる。MPI プログラムのような並列プログラムは、一般に同一プログラムを様々な PU 台数で実行することが可能である。しかしながら PU 台数を増やして実行しても、プログラム中の通信時間の増加やプログラムの逐次性部分の影響で、台数効果がそのまま実行時間に反映しない場合がある。

このような背景のもと、想定するプラットフォームでの実行に先立って、MPI プログラムの実行時間を予測することが重要になってきている。

従来、MPIプログラム実行時間の予測手法として、シミュレータを用いてアセンブラコードレベルのトレースを行い、計算時間の予測を行う手法[1]や、シミュレーションを行わずソースプログラムを静的に解析することで、プログラムの挙動を予測する手法[2] が提案されている。前者はキャッシュヒット率や、ネットワークレイテンシ、バンド幅の変化に対応した詳細な挙動の解析が可能であるが、予測にかかる処理時間として実際の実行時間に対し数倍の時間を要する。また後者はコンパイラ最適化や、キャッシュ効果、ネットワーク性能を考慮することが困難であり、一般に予測精度が悪い。

我々はこれまでに、プログラムの簡易実行による短時間での実行時間予測手法[3]を提案した。本稿では本手法の有効性を確認する。以下、2節では従来の予測手法に対する本手法の利点と手法の概要、3節は本手法の評価について述べ、4節でまとめを行う。

2 手法概要[3]

1節で述べた従来の手法の問題に対して、我々は[3]において、[1]の手法と比較し短時間で、かつ[2]の手法と比較し精度の高い予測手法を提案している。本手法では対象となるMPIプログラムを2台のPU上で簡易実行することで得られる、基礎パラメータを用いてプログラム全体の実行時間を予測する。

2.1 プログラムのモデル化

本手法ではプログラムを最内ループのループボディを構成する「計算ブロック」とMPI通信関数を含む文である「通信ブロック」とに分割し、ブロック単位で実行時間予測を行う。以下にプログラムを計算ブロックと通信ブロックに分割する方法を述べる。

(1) ループ構造を持たない部分は、繰り返し回数が1回のループとする。

(2) 最内ループのループボディを1つの計算ブロックとする。

(3) (2)で得られた計算ブロック内で、MPI 通信関数が宣言されている場合は、MPI 関数を含む文の前後で 2 つの計算ブロックに分割する。また、MPI 通信関数を含む文を通信関数とする。

2.2 基礎パラメータの取得

予測に必要な各ブロックの基礎パラメータ及び各ブロックの実行回数を取得する方法を述べる。尚、これら基礎パラメータは 2 台の PU を用いて取得する。

計算ブロックの基礎パラメータは各計算ブロックが1回実行されるときの実行時間である。

尚、コンパイラ最適化及びキャッシュ効果を考慮した計算ブロックの実行時間を得るため、計算ブロック 1 回分の実行時間は、当該計算ブロックのみをループボディとしてもつループ全体の実行時間を計測した後に求める。

通信ブロックの基礎パラメータは各通信ブロックの通信関数における引数である。具体的にはメッセージサイズ、メッセージタイプ、タグが基礎パラメータとなる。これらパラメータを取得した後2PU間での擬似通信を行い、各通信ブロック1回分の実行時間とする。

2.2.1 基礎パラメータの取得プログラムの実行

各ブロックの基礎パラメータ取得に必要な時間の短縮のため、対象プログラムに以下の修正を施す。

(1) 当該計算ブロックのみをループボディとして持つループを除いて、DO文をコメントアウトする。

(2) 他の通信のパラメータを変化させない限り通信ブロックをコメントアウトする。

以上により作成されたプログラムを2台のPUを用いて計算ブロック、通信ブロックの基礎パラメータを取得する。この実行方法を簡易実行と呼ぶ。

2.2.2 ブロック実行回数

次にPU台数がp台時のブロック実行回数を求める方法を述べる。ここでもブロック実行回数の取得時間削減のため、対象プログラムに以下の修正を施す。

(1) プログラム中の全てのループに対して、ブロック実行回数に依存する変数の計算のみを残し、他の実行文をコメントアウトする。

(2) MPI通信関数も、(1)のブロック実行回数に依存する通信以外はコメントアウトする。

以上の方法で作成したプログラムを実行し、計算ブロック、通信ブロックそれぞれの実行回数を求める。

最後に、求めた各ブロック 1 回分の実行時間にブロック実行回数を乗じ、総和を以って予測時間とする。

3 検証

NAS Parallel Benchmarks(NPB) ver2.3の8つのプログラム、EP,FT,MG,CG,IS,LU,SP,BTを用いて、2〜16台時の実行時間予測を行った結果を以下に示す。対象クラスはFTがCLASS A、他はCLASS Bを用いた。計算ブロックの基礎パラメータは2PUのものを使用するが、SP,BTに関しては実行PU台数がN^2台であり2PUでの実行ができないので、4PUのものを使用した。

A Execution-time Prediction Scheme of MPI Programs Based Simpe Execution

Toshihiro Iwabuchi†, Hiroshi Horii‡, Hayato Yamana†

†Department of Information and Computer Science, School of Science and Engineering, Waseda Univ.

‡Graduate School of Science and Engineering, Waseda Univ.

まず表1に測定対象システムの構成を示し、表2にプログラム各々の実実行時間と予測時間を示す。また、表3に実測と予測にかかる時間の比較を示す。これはPU台数すべての実行にかかる時間の総和を以って結果とする。

表1：測定対象システム

CPU	Pentium4
Clock[1]	1.4GHz × 8Node 1.6GHz × 8Node
L2cache	256Kbyte
Memory	512Mbyte
Network	100BaseTX Ethernet

4 おわりに

本稿では[3]で提案された、2PUを利用して、MPIプログラムの実行時間を短時間で予測する手法の評価を行った。NPB2.3の実行時間を予測したところ、全てのプログラムで予測に要した時間は実実行時間の$O(10^{-1})$～$O(10^{-3})$であり、本手法の有効性が確認できた。またIS,CGを除き、予測誤差はほぼ10%以内で予測可能であった。

本手法では通信には同期がとれていることが前提で擬似通信を行い、通信ブロックの実行時間を予測する。ISのように通信時間が総実効時間に対して支配的であり、かつ通信に同期が取れていない[2]プログラムに対しては本手法を適用しても精度が落ちる。通信ギャップを考慮した通信時間の予測が今後の課題となる。

CGは他のプログラムと比べ計算時間の誤差が大きい。この要因は、計算時間の大部分を占める計算ブロックにおいて、実行文が配列へのランダムな参照を行っているためである。2～16PU台数それぞれでキャッシュ効果が一定でなく、計算ブロック1回分の実行時間に差が出る。本手法では2PUの基礎パラメータを用いて、他のPU台数における計算ブロック実行時間を予測するため、こういった場合は誤差が大きくる。故にCGの場合は、PU自身の持つ計算ブロックパラメータを使用しての予測が必要である。計算ブロックの基礎パラメータは、各々のPU台数で$O(10^{-1})$～$O(10^{-3})$の時間で取得が可能であり、この場合でも十分実用的な処理時間での予測が可能であると考えられる。

参考文献

[1] Kazuto Kubota, Ken'ichi Itakura, Mitsuhisa Sato, Taisuke Boku:"Practical Simulation of Large-Scale Parallel Programs and Its Performance Analysis of the NAS Parallel Benchmarks", Proc. ofEuro-Par, pp.244-254, 1998.

[2] Maurice Yarrow and Rob Va der Wijngaart:"Communication Improvement for the LU NAS Parallel Benchmark:A Model for E cient Parallel Relaxation Schemes ",NAS Technical Report NAS-97-032,1997.

[3] 堀井洋,山名早人:"実測に基づいたMPIプログラムの実行時間予測",情報処理学会研究報告(HPC),No 88,pp61-66,2001.

1 6台の Clock が一定でないので、8 台までの実行には 1.4GHz のノードを使用した。計算ブロックのパラメータ取得についても同様である。

2 同期関数 MPI_Barrier()を使用して通信ギャップの時間を測定したところ合計 10 秒程度であった。

表2：各プログラムにおける計算時間・通信時間の実測と予測時間

	PU	計算時間		通信時間		全体の予測誤差
		実測	予測	実測	予測	
EP	2	202.508	203.501	0.002	0.001	0.49%
	4	101.329	101.135	0.031	0.001	0.22%
	8	50.626	51.191	0.084	0.002	0.95%
	16	25.12	25.603	0.26	0.003	0.89%
FT	2	18.433	22.995	40.648	40.611	8.58%
	4	9.443	11.569	45.29	40.585	4.90%
	8	5.283	6.247	33.225	30.451	4.70%
	16	3.065	3.15	22.967	20.304	10.04%
MG	2	33.784	32.886	14.286	15.218	0.07%
	4	17.838	17.881	14.215	14.277	0.33%
	8	8.375	8.54	11.63	9.84	8.12%
	16	4.554	4.604	11.465	9.834	9.84%
CG	2	391.791	394.695	99.4367	188.085	18.64%
	4	199.723	200.595	196.668	195.563	0.06%
	8	41.762	101.531	160.821	143.05	20.73%
	16	22.8	52.92	166.799	143.717	3.71%
IS	2	9.176	8.857	58.25	59.142	0.85%
	4	4.254	4.414	70.436	57.775	16.74%
	8	2.187	2.245	53.152	42.409	19.31%
	16	1.775	1.652	40.157	30.837	22.51%
LU	2	768.552	775.551	46.848	38.809	0.13%
	4	372.206	385.081	93.014	77.726	0.14%
	8	196.001	199.828	90.269	71.429	1.34%
	16	106.547	103.486	65.833	65.27	2.10%
SP	4	528.286	534.543	267.479	303.477	5.31%
	9	234.756	242.248	389.977	370.483	1.92%
	16	133.106	138.945	338.3	367.228	7.38%
BT	4	557.093	557.467	141.959	163.442	3.13%
	9	250.59	250.613	221.754	193.188	6.04%
	16	142.399	143.399	179.826	189.421	3.37%

単位(s)

表3：NPB2.3プログラム各々の実行時における予測に必要な処理の総時間と実際の総実行時間

	EP	FT	MG	CG
予測に必要な総時間	0.12	19.76	7.46	2.85
実際の総実行時間	379.96	192.22	239.39	1279.10
	IS	LU	SP	BT
予測に必要な総時間	8.28	8.27	13.73	25.03
実際の総実行時間	239.39	1729.27	1891.91	1493.62

単位(s)