ハイパフォーマンス
コンピューティング 52−16
（1994．7．23）

KU PVM3/AP1000の性能評価

岩下茂信　　村上和彰

九州大学 大学院総合理工学研究科 情報システム学専攻
〒816 春日市春日公園 6−1
E-mail: {iwashita, murakami}@is.kyushu-u.ac.jp

ワークステーション・クラスタ構成用並列プログラミング・ライブラリを実際の並列計算機に実装することにより，並列計算機をワークステーション・クラスタと同様に扱うことができるようになる．並列プログラミング・ライブラリの実装方法の違いにより，マシンの機能および性能が変わってくる．本稿では，並列プログラミング・ライブラリ PVM を並列計算機 AP1000 に実装し，ライブラリ，実装方法，通信媒体，等の違いによる性能の違いを，基本通信性能およびアプリケーション・プログラム性能について測定し，評価を行った．

Performance Evaluation of the KU PVM3/AP1000

Shigenobu Iwashita　Kazuaki Murakami

Department of Information Systems
Interdisciplinary Graduate School of Engineering Sciences
Kyushu University
6−1 Kasuga-koen, Kasuga-shi, Fukuoka 816 Japan
E-mail: {iwashita, murakami}@is.kyushu-u.ac.jp

By implementing a parallel programming library for workstation clusters on a parallel computer, we can treat the parallel computer as a virtual workstation cluster. Functions and performances of the parallel machine depend on the implementation. We implemented the PVM, one of parallel programming libraries on AP1000. In this paper, we compare the performances of different libraries , different implementations, and different network configurations through basic communications and realistic applications.

1 はじめに

並列プログラミング・ライブラリは，プロセッサ間通信や同期など，並列処理に必要な機能を関数として用意したものである．ユーザはプログラム中で陽にこれらの関数を呼び出すことにより，並列プログラムを作成する．一般的な並列計算機ではマシンごとに独自に並列プログラミング・ライブラリが用意されている．

並列プログラミング・ライブラリを用いて，ワークステーション・クラスタを実現することが可能である．ワークステーション・クラスタとは，ネットワークで接続された複数のワークステーションを見かけ上1台の並列計算機として使用するものである [9]．並列プログラミング・ライブラリにより，各ワークステーション上のプロセス間の通信を行う．

筆者らは，並列プログラミング・ライブラリ PVM3 (Parallel Virtual Machine Ver.3) をメッセージ交換型マルチコンピュータ AP1000 に実装した [4][3]．これを KU PVM3/AP1000 (Kyushu University PVM3 on AP1000) と呼ぶことにする．実装の目的の一つに，並列プログラミング・ライブラリによるインタフェースの統一が挙げられる．元々 AP1000 用のプログラムは，AP1000 独自の並列プログラミング・ライブラリを用いて作成する．しかし，そのプログラムは AP1000 上でしか実行することができず汎用性がない．そこで，PVM という一種の標準化された並列プログラミング・ライブラリを用いることにより，PVM が実装されている他の計算機とのプログラミングに関するインタフェースの統一が可能となる．

また，ANU (Australian National University) においても，PVM の AP1000 上への実装がなされている．これを ANU PVM3/AP1000 と呼ぶことにする．KU PVM3/AP1000 は，PVM によって書かれたプログラムを AP1000 で実行可能にするものであり，AP1000 を他のワークステーションと接続して分散処理を行うことはできないが，ANU PVM3/AP1000 は AP1000 をワークステーション・クラスタの1構成要素として他のワークステーションと一緒に分散処理を行うことができる．

筆者らは，文献 [4] において AP1000 独自の並列プログラミング・ライブラリ，KU PVM3/AP1000，および，ワークステーション上の PVM についてそれらの基本通信性能を評価した．本稿では，これに ANU PVM3/AP1000 を評価対象として加え，基本通信性能およびアプリケーション・プログラムを用い，ライブラリ，実装方法，通信媒体，等の違いによる性能の違いを測定した．以下，2章で，ワークステーション上の PVM，KU PVM3/AP1000，および，ANU PVM3/AP1000 について説明する．3章で，性能測定の方法について述べる．4章で測定結果とその評価および考察を行い，最後に5章で本稿のまとめとする．

2 KU PVM3/AP1000

2.1 PVM の概要

PVM は，ワークステーション・クラスタを実現するための並列プログラミング・ライブラリである．PVM により，ワークステーション・クラスタ上で並列プログラムを実行することが可能となる．並列プログラムにおいて並列に実行されるプログラムの単位をタスクと呼ぶ．始めに起動されるタスクをマスタ・タスク，マスタにより生成されるタスクをスレーブ・タスクと呼ぶ．ユーザは，逐次型言語で記述したアプリケーション・プログラム (各タスク) 中で PVM の通信用関数を呼び出すことにより，他のワークステーション上にある PVM のタスクと通信することができる．これにより，複数のワークステーションをあたかも一台のメッセージ交換型マルチコンピュータのように使用することができる．以下，PVM による仮想的並列計算機をバーチャル・マシン，バーチャル・マシンを構成する個々のワークステーションを PVM ホストと呼ぶ．また，ワークステーション上に実装された PVM を PVM/WS と呼ぶ．

PVM は，図1に示すように，PVM-Daemon と PVM ライブラリの2つで構成される．PVM-Daemon は，各 PVM ホスト上にあるタスク間の通信を司るプロセスである．各タスク間のデータ通信はすべて PVM-Daemon を介して行われる．PVM ライブラリは，PVM タスク間の通信用関数のライブラリである．タスク間のメッセージ通信，バリア同期，タスクの生成，PVM ホストをバーチャルマシンに加える，等の機能がそれぞれ関数として用意されている．PVM ライブラリの関数を使ってプログラミングを行うことで，ユーザは各ワークステーション間の複雑なプロトコルを直接触ることなく並列プログラムを記述することができる．

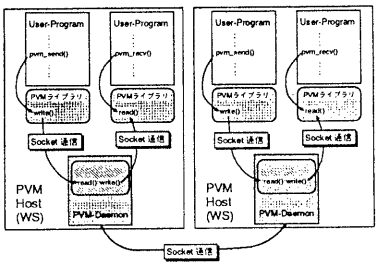

図 1: PVM の構成

2.2 KU PVM3/AP1000

KU PVM3/AP1000 とは，九州大学が AP1000 上に実装した PVM である [4][3]. KU PVM3/AP1000 の構造を図 2に示す. AP1000 単体で 1 個の PVM ホストという形になる. 現在のところ，PVM-Daemon は実装しておらず，PVM ライブラリのみを実装している. そのため，他の PVM ホストとは接続することが出来ず，AP1000 単体でのみバーチャルマシンを構成する. すなわち，KU PVM3/AP1000 は PVM を用いて作成されたプログラムを AP1000 単体上で実行可能にするための並列プログラミング・ライブラリという形になる.

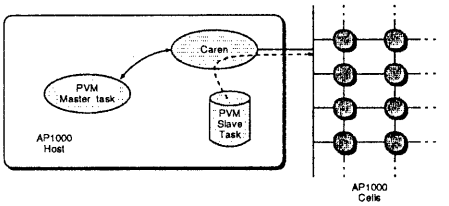

図 2: KU PVM3/AP1000 の構造

KU PVM3/AP1000 は PVM のほとんどの関数を実現している. ただし，AP1000 の機能の関係上，次に挙げる制限がある.

1. 各セル (AP1000 のプロセッシング・エレメント) 上で実行されるスレーブ・タスクの生成は，1 つのプログラムにつき 1 回のみ可能である. ただし，一度に複数のスレーブ・タスクの生成は可能である.

2. 一度に複数のスレーブ・タスクを生成可能であるが，それらはすべて同一のものに限る. つまり，1 つのプログラムにおいて，複数の異なるスレーブ・タスクを生成および実行することはできない.

2.3 ANU PVM3/AP1000

ANU PVM3/AP1000 とは，ANU が AP1000 に実装した PVM である [5][6]. AP1000 と他の PVM ホストを接続し，AP1000 全体が 1PVM ホストとなって，より大きなバーチャルマシンを構成することができる. ANU PVM3/AP1000 の構造を図 3に示す. 他の PVM ホスト上のタスクと AP1000 の各セル上のタスクとの通信は，PVM-Daemon, ApPvmDriver (PVM-Daemon と Caren との通信を司るプロセス)，および，Caren (AP1000 ホストと AP1000 セルとの通信を司るプロセス) を通じて行われる.

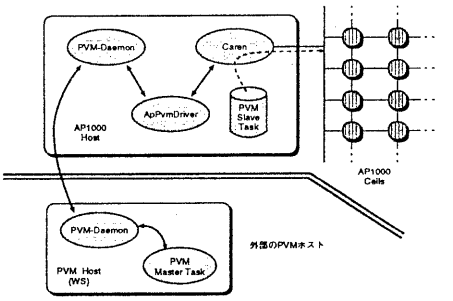

図 3: ANU PVM3/AP1000 の構成

ANU PVM3/AP1000 においても，AP1000 の機能の関係上，2.2節で述べたの制限 と同じ制限がある. ただし，AP1000 とともにバーチャルマシンを構成している他の PVM ホストにおいては，この限りではない. また，PVM にはいくつかのタスクがグループを作り，グループ内でのメッセージ放送やバリア同期をとるグルーピングの機能があるが，AP1000 単体のみをバーチャルマシンとして使用する場合，このグルーピングが使用できない.

3 性能測定

3.1 評価目的

KU PVM3/AP1000, ANU PVM3/AP1000, AP1000 独自の並列プログラミング・ライブラリ (以下，Native AP1000 と記す)，および，PVM/WS において，

1. 最も基本的な通信である 1 対 1 通信

2. アプリケーション・プログラム

の性能を測定した. その目的は，ライブラリ，実装方法，通信媒体，等の違いにより，基本通信性能およびアプリケーション性能にどの程度の差が生じるかを調べることである.

3.2 測定に用いたマシン

今回測定に使用したマシンの諸元は，次の通り.

AP1000

- セル: SPARC (25MHz)
- セル数: 最大 64 セル
- ネットワーク: 2 次元トーラス (1 リンク当り 25MBytes/sec.)

ワークステーション・クラスタ

- 機種構成
  - S-4/10 30 (36MHz SuperSPARC) × 1
  - S-4/10 40 (40MHz SuperSPARC) × 1
  - S-4/2 (40MHz SPARC) × 2
- ネットワーク: Ethernet(10Mbytes/sec.)

3.3 基本通信性能の測定

AP1000 上ではホスト-セル間および隣接するセル-セル間の 1 対 1 通信，また，PVM/WS では異なる PVM ホスト (S-4/10 30 と S-4/10 40) 上にあるタスク間の 1 対 1 通信について，通信量を 0〜4000 Bytes の範囲で変えて所要時間を測定した．2 タスク間でメッセージを 1000 回往復させるのに要した時間を測定し，その時間を 2000 で割って 1 回の通信時間を求めた．

3.4 アプリケーション・プログラム性能の測定

文献 [2] 中の例題プログラムである「差分法による 2 次元の熱伝導方程式解法プログラム」を用いて性能測定を行った．これは，2 次元の領域の一部に熱を加えた時，一定時間後の最高温度を求めるものである．領域を 320 × 320 の正方サブ領域に分割し，各サブ領域ごとに各々独立に温度を計算する．各サブ領域は，それぞれ隣接するサブ領域の温度により，次タイム・ステップでの自サブ領域の温度を求める．プログラムの流れを図 4に，また，各タスクへの領域の割当てを図 5に示す．AP1000 の場合は，マスタ・タスクは AP1000 ホスト上で，スレーブ・タスクは AP1000 セル上で実行される．

以下のパラメータを変えながら，以下の項目について測定を行った．

測定パラメータ

- スレーブ・タスク数: 4, 64
  (PVM/WS は 4 のみ)
- タイム・ステップ数: 10, 100, 200, 400, 600, 800, 1000
- バリア同期: 有/無
  (ANU PVM3/AP1000 は無のみ)

測定項目

テストプログラムの全実行時間，タスク間通信に要した時間，および，バリア同期に要した時間 (ANU PVM3/AP1000 では，バリア同期を行っていない)．タスク間通信時間およびバリア同期時間は，それぞれ通信用関数の実行に要した時間を測定した．

このプログラムの各通信パターンにおける通信量および通信回数は以下の通り．

1. マスタ-スレーブ間通信
   - スレーブ数 4: 36Bytes × 1 回 + 20Bytes × 1 回
   - スレーブ数 64: 276Bytes × 1 回 + 20Bytes × 1 回
2. スレーブ-スレーブ間通信
   - スレーブ数 4: 640Bytes ×タイム・ステップ数× 4 回
   - スレーブ数 64: 160Bytes ×タイム・ステップ数× 4 回

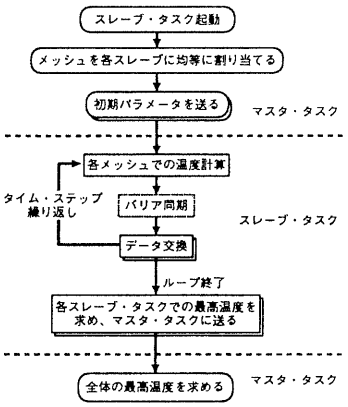

図 4: 熱伝導プログラムの流れ

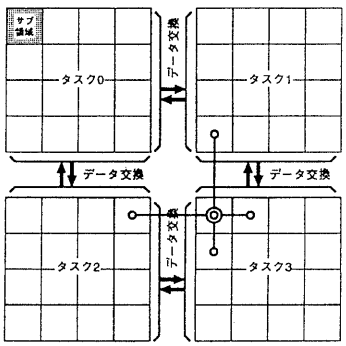

図 5: 各タスクへの割付け

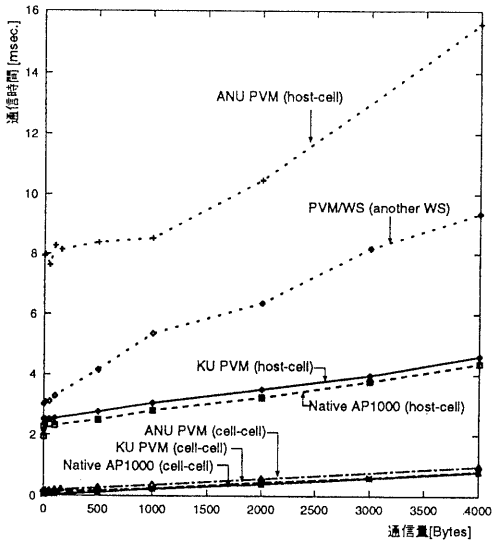

図 6: 1 対 1 通信時間

4 結果と考察

基本通信性能の測定結果を図 6に示す．また，アプリケーション・プログラムの時間測定結果を表 1に示す．

表 1: プログラム実行時間

	タイム・ステップ数	実行時間 [sec]	通信時間 [sec] (%)	同期時間 [sec] (%)
KU-PVM (4 セル)	10	8.200	0.034 (0.41)	0.043 (0.52)
	100	74.088	0.307 (0.41)	0.434 (0.58)
	400	293.663	1.215 (0.41)	1.736 (0.59)
	1000	732.808	3.031 (0.41)	4.341 (0.59)
ANU-PVM (4 セル)	10	11.076	0.034 (0.31)	–
	100	77.031	0.345 (0.45)	–
	400	296.674	1.381 (0.47)	–
	1000	735.862	3.451 (0.47)	–
Native AP1000 (4 セル)	10	7.804	0.013 (0.17)	0.001 (0.02)
	100	73.497	0.131 (0.18)	0.011 (0.02)
	400	292.479	0.523 (0.18)	0.043 (0.02)
	1000	730.451	1.308 (0.18)	0.108 (0.02)
WS-PVM (4WS)	10	5.437	0.585 (10.75)	1.205 (20.68)
	100	47.966	4.933 (10.28)	11.029 (21.50)
	400	189.537	12.270 (6.47)	52.390 (25.69)
	1000	471.762	31.706 (6.72)	128.279 (25.35)
KU-PVM (64 セル)	10	1.498	0.091 (6.10)	0.338 (19.01)
	100	5.845	0.292 (5.00)	3.138 (35.16)
	400	20.333	0.959 (4.72)	12.457 (38.03)
	1000	49.327	2.294 (4.65)	31.126 (38.73)
ANU-PVM (64 セル)	10	5.276	0.021 (0.40)	–
	100	9.610	0.211 (2.20)	–
	400	24.147	0.852 (3.53)	–
	1000	53.088	2.140 (4.03)	–
Native AP1000 (64 セル)	10	1.228	0.006 (0.46)	0.001 (0.07)
	100	5.396	0.055 (1.03)	0.008 (0.14)
	400	19.307	0.220 (1.14)	0.032 (0.16)
	1000	47.137	0.549 (1.17)	0.080 (0.17)

注:実行時間および通信時間はバリア同期なしでの測定結果

4.1 ライブラリの実現方法に関する比較

図6より，KU PVM3/AP1000 と ANU PVM3/AP1000 を比較してみると，セル-セル間通信時間においては大きな差はない．ホスト- セル間通信においては ANU の方が KU に比べておよそ 4 倍程度通信時間が長くなっている．この原因として，次のことが考えられる．

- セル-セル間通信: KU, ANU ともに Native AP1000 のセル間通信用関数を直接用い，同じような実現方法をとっている．
- ホスト-セル間通信: ANU PVM3/AP1000 は，ホスト上のタスクから PVM-Daemon および ApPvmDriver を通してセル上のタスクと通信を行うため，オーバーヘッドが大きい．これに対して，KU PVM3/AP1000 はホスト上のタスクとセル上のタスクが直接通信を行っている．

アプリケーション・プログラムの全実行時間は ANU PVM3/AP1000 が KU PVM3/AP1000 より 3〜4 秒大きい．今回用いたアプリケーション・プログラムではホスト-セル間通信がループ回数にかかわらず一定回数であったため，全実行時間は常に 3〜4 秒の差となったが，ホスト-セル間通信がループ回数に比例するようなプログラムの場合には，実行時間の差はループ回数に比例して大きくなると考えられる．

KU PVM3/AP1000 と Native AP1000 のバリア同期を比較してみる．PVM ではバリア同期はグループ単位で行う．一方，AP1000 のハードウェアが提供するバリア同期には全セルが参加する必要がある．よって，KU PVM3/AP1000 は AP1000 ハードウェアが持つバリア同期機能を使用していない．そのため，KU PVM3/AP1000 上でのバリア同期時間はタスク数に比例して大きくなる．このように，ハードウェアでせっかくバリア同期機構を備えていても，その上の並列プログラミング・ライブラリとの間で機能が一致しなければ使うことができない．なお，グループ単位でのバリア同期をハードウェアでサポートするには，任意参加バリア同期 [8] の機能を備えている必要がある．一般に，任意参加バリア同期機能は，AP1000 の備える強制参加バリア同期 [8] に比べ，ハードウェア・コストが大き

くなる.

4.2 マシンおよびネットワークに関する比較

PVM/WSとPVM/AP1000を比較してみると, PVM/WSの方がAP1000より全体の実行時間は短くなっている (表1参照). これは, ワークステーション単位の処理性能がAP1000のセルの処理性能よりも優れているためである. 通信時間については, PVM/WSはPVM/AP1000上のPVMよりおよそ10倍程度遅く, 全実行時間に占める通信時間の割合も6~10%程度とかなり大きくなっている. これは, 通信路の通信容量がAP1000の方が大きく, また, PVM/WSはスレーブ・タスク間通信時にもPVM-Daemonを通すため, その分通信にかかる時間が大きくなるためである.

5 おわりに

KU PVM3/AP1000, ANU PVM3/AP1000, AP1000独自の並列プログラミング・ライブラリ, および, ワークステーション上のPVMについて, 基本通信性能およびアプリケーション・プログラム性能を測定し, ライブラリの実装方法の違い等による性能の違いについて考察した.

標準的並列プログラミング・ライブラリの実装により, 異なる並列計算機間でのプログラムの移植性が高まる. また, 複数の計算機を接続し, より大きなマシンとして使用することも可能となるが, この場合各マシン間の通信のオーバーヘッドは大きくなる.

また, 標準的並列プログラミング・ライブラリを並列計算機に実装する場合, たとえばバリア同期機能などについて, ハードウェアの持つ機能がライブラリの機能に柔軟に対応できない場合には, その機能をソフトウェアで実現する必要があり, その分性能が低下することになる.

今後の研究課題としては, 標準的並列プログラミング・ライブラリの実装によるプログラム実行環境の柔軟性と, 性能とのトレード・オフに関する検討, 並列プログラミング・ライブラリの持つ機能を十分に活かすことができるハードウェアの機能の検討, 等がある.

謝辞

日頃よりご討論頂き, 多くの有用なご意見を頂く九州大学 大学院総合理工学研究科 安浦寛人 教授, 岩井原瑞穂 助手, ならびに, 國貞勝弘 氏をはじめとする安浦研究室の諸氏に深く感謝します. 本稿の執筆に当たり, 有用なご意見を頂いた電子技術総合研究所 関口智嗣 氏, および, お茶の水女子大学 長嶋雲兵 助教授に深く感謝します. AP1000 および ANU PVM3/AP1000を提供して頂いた富士通研究所 並列処理研究センターに深く感謝します.

本研究は一部, 文部省科学研究費補助金 重点領域研究「超並列原理に基づく情報処理体系」, および, EAGL事業推進機構 育成研究助成金による.

参考文献

[1] Geist, A., Beguelin, A., Dongarra, j., Jiang, W., Manchek, R., and Sunderam, V., *PVM 3.0 user's guide and reference manual*, ORNL/TM-12187, Feb. 1993.

[2] 富士通研究所, AP1000プログラム開発手引書 (I) C言語インタフェース, 第2版, 1992年5月.

[3] 岩下茂信, 並列プログラミング・ライブラリPVMのAP1000への実装, および, その性能評価, 九州大学工学部情報工学科卒業論文, 1994年2月.

[4] 岩下茂信, 國貞勝弘, 村上和彰, "PVM/AP1000の実現および通信性能評価," 情処研報, HPC-50-13, 1994年3月.

[5] Johnson, C. W. and Walsh, D., "Porting the PVM Distributed ComputingEnvironment to the Fujitsu AP1000," *Proceedings of 2nd Parallel Computing Workshop*, P1-D-1-P1-D-12 ,Nov. 1993.

[6] Australian National University, *ANU/Fujitsu CAP Research Program AP1000 Software Release 1994* PVM Overview,

[7] Iwashita, S., Kunisada, K., and Murakami, K., "Implementing the PVM (Parallel Virtual Machine) on the AP1000," *Proceedings of 2nd Parallel Computing Workshop*, P2-E-1-P2-E-6, Nov. 1993.

[8] 山家陽, 村上和彰, "バリア同期モデル -Taxonomyと新モデルの提案, および, モデル間性能比較-," JSPP'93 pp.119-126, 1993年5月.

[9] 関口智嗣, 長嶋雲兵, 日向寺祥子, "ワークステーションクラスタとメッセージパッシングライブラリ," 情処研報, HPC-47-3, 1993年6月.