AP1000互換通信ライブラリ: WSクラスタ向けの新しい計算環境 －実装と評価－*

2L－2

小林 健一　林 憲一　堀江 健志†
富士通株式会社 HPC本部 第二開発統括部‡

1 はじめに

近年ワークステーションの高速化とともにSymmetrical Multiprocessors (以下SMP)やWSクラスタなどが有力な計算資源として注目されている。しかし、現状ではこれらの異なったプラットフォームをまとまった計算資源として有効に活用するための、効率よいアプリケーションの実行環境は存在しない。

我々はこのような計算資源を活用するために、マルチスレッドを基盤とした仮想的な計算ノードをつくる、並列計算環境を提案する[1]。マルチスレッドによる仮想ノードを用いることにより、SMPが持つ複数のプロセッサ資源を最大限に活用できる。またWS内に複数の仮想ノードを生成することで、並列度を上げたり、ノード毎の負荷を制御することができる。

この計算環境の一例として、我々は高並列計算機AP1000の基本OSであるCellOSと互換の実行制御・通信ライブラリAPlibを開発した。本論文ではこのAPlibの特長と実装について述べ、APlibによってSMPやWSクラスタで得られる性能を評価する。

2 APlibの特長

APlibはマルチスレッドを実行基盤に採り入れた計算環境を持つメッセージパッシング型の通信ライブラリである。SMPやWSクラスタなどの異なる構成でも効率よく動作する汎用的な並列処理環境を目標としている。APlibは前章で述べたマルチスレッドを利用した機能に加え、以下の特長を持つ。

- AP1000(CellOS)との互換性

  AP1000の持つ通信インタフェース、およびノード毎のマルチタスク機能を備えた実行環境との互換性を持つ。
- アーキテクチャ依存機能のモジュール化
- プログラミングの簡易性

  ユーザに対してマルチスレッドプログラミングを意識させないようにしている。例えば、マルチスレッドでは共有されてしまう大域変数や静的変数を自動的に非共有変数に変換する機能を持つ。
- 移植性、運用性

  APlibは現在Solaris2上で動作しているが、標準的な機構を用いて実装されているので、高い移植性を持っている。また、マルチスレッドデバッガなどの多くの標準的なツールが利用可能である。

*AP1000 compatible communication library: A new computing environment for workstation clusters(2)

†K. Kobayashi, K. Hayashi, and T. Horie
‡High Performance Computing Group, Fujitsu Ltd.

3 実装

APlibはマルチスレッドを基盤にして実装されている。図1はAPlibを使って作成されたプログラムの動作時のタスク実行制御、通信制御などをモデル化したものである。

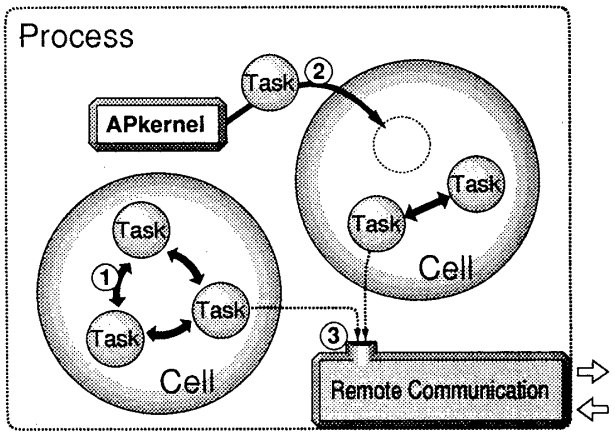

図 1: APlibの動作モデル

3.1 タスク実行制御機構

図1はあるWS内で複数のタスクが動作している様子である。このWSには二つのセル(AP1000におけるノードの別名)が仮想ノードとして生成されており、タスクがそれぞれ3個と2個動作している。これらは全て単一のプロセス内で動作している。通常、単一のWSまたはSMPにはAPlibのプロセスはだた一つ生成される。

セルはLight Weight Process(LWP)として実装されており、他のセルと並行/並列に動作する。SMPではプロセッサの数だけ同時にセルが並列動作することになる。タスクはスレッドとして実装されており、他のタスクとは受信待ちなどのイベントを契機としてコンテクストを切替える(図1①)。この切替えはタスクが呼び出すAPlibライブラリの内部で処理され、OSが介入しないため、高速なスケジューリングが実現できる。

3.2 タスクの生成/回収

図1中のAPkernelがタスクの生成および回収を担当するスレッドであり、プログラムが起動した時に最初に呼び出される。APkernelは起動時だけでなく、稼働時にもプロセス内に新たなタスクをロードすることができる(図1②)。加えて、このタスクローダは大域変数を非共有変数に変換することにより、2章で述べたように、マルチスレッド向けに書かれていないプログラムを正しく

実行することができる。これらの機能はOSの提供するdynamic link機構を利用して実装されている。

3.3 ローカル通信と排他/キャッシュ制御

WS内のタスクは全て同じプロセス空間を共有しており、タスク間の通信はメモリコピーによって高速に行なわれる。並列動作するタスク間のメモリアクセスは排他制御を行なう必要があるが、APlibはメッセージパッシング型であり、タスク間のメモリアクセスはライブラリの監視下にあるので、排他制御をユーザが行なう必要はない。

SMPのプログラムの性能を落す原因の一つであるキャッシュの競合の発生も、タスク間メモリアクセスに伴って発生する。APlibはキャッシュ競合を低減させるメモリ配置を行なっている。

3.4 リモート通信とモジュラリティ

プロセス外(つまりWS外)への通信はリモート通信用のモジュールを用いて実現される(図1③)。TCPによる通信機能を提供するモジュールや、ATMなどの専用高速ネットワークを用いた通信機能を提供するモジュールがあり、実行する計算機の構成や、利用できる通信デバイスなどによって、起動時に適切なモジュールが選択され、リンクされる。

4 評価

4.1 SMPとWSクラスタ

APlib上のアプリケーションの台数効果を、SMPとWSクラスタのそれぞれについて測定する。

アプリケーションはscg(共役勾配法による二次元Poisson方程式ソルバ)で、問題サイズは90000×90000の疎行列である。このアプリケーションは一定のパターン(数KBと数ワードのメッセージおよびグローバル演算)の通信を繰り返す。

SMPは8CPUのSPARC Server1000 (SuperSPARC+, 50MHz) であり、WSクラスタはSPARC Station20 (HyperSPARC, 125MHz)8台をイーサネット上でTCPを用いて通信させたものである。どちらの測定にも同一のプログラム(バイナリ)を使用した。測定はネットワークおよびプロセッサの閑静時に3回行ない、中間の値を採った。

得られた結果を図2に示す。横軸が構成台数(SMPの場合は使用プロセッサ数、WSクラスタの場合はWS台数)であり、縦軸がMFlops値である。

グラフより、SMPは線形を超える台数効果が得られていることが判る。これはCPU台数が増えるに従い、各CPUの担当するデータ量が減り、キャッシュのヒット率が向上するからである。

WSクラスタにおいても、台数に応じた性能向上が得られることが判る。こちらは台数が増えるに従い、効率が落ちるが、これはTCPによる通信遅延が大きいことが原因である。

4.2 不均質な性能のWSクラスタ

計算資源を有効に利用するために異なる性能のWSを結合させて処理を行なう場合がある。しかし、データパラレルな並列処理ではノードの性能はできるだけ均一な

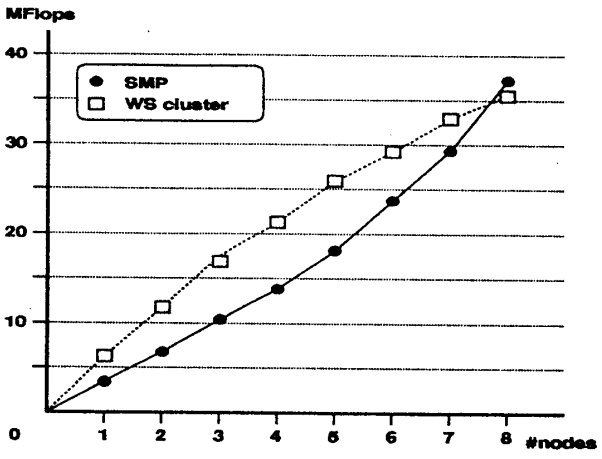

図 2: SMP、WSクラスタの性能

方がよい。ここでは以下のような性能の異なるWSの組合せに対し、仮想ノードを生成して性能の均一化を行ない、その効果を調べる。

測定は以下の2台のWSを結合して行なった。単体性能は前述のscgを$2^{16} \times 2^{16}$の疎行列について解かせた時のMFlops値である。

機種	プロセッサ	単体性能
SS/LX	microSPARC 50MHz	2.44MFlops
SS10	SuperSPARC 40MHz	4.30MFlops

これをSS/LXとSS10を結合したもの(1)と、SS/LXと2つの仮想ノードを生成したSS10を結合したもの(2)を同じscgについて実行した結果を示す。

	構成	結合性能
1	SS/LX + SS10	4.52MFlops
2	SS/LX + SS10(2仮想ノード)	5.78MFlops

scgの処理はデータパラレルであるため、(1)はSS10の性能に関わらず、SS/LXの元々の性能の約2倍しか得られていない。しかし、(2)ではそれぞれのノードの性能はかなり近くなるため、結果としてSS/LXとSS10の性能和に近い(85%)性能が得られている。このように性能の異なるWSの計算能力を仮想ノードを用いて活かすことができる。

5 まとめ

APlibはマルチスレッドを基盤にした実行モデルを持ち、SMPからWSクラスタまでの種々のプラットフォームを統合した計算環境を提供する。本論文ではこのAPlibの実装を示し、SMPとWSクラスタの上でAPlibのアプリケーションの性能評価を行なった。また、不均一なWS環境も有効に活用できることを示した。これにより、APlibおよび提案されたマルチスレッド実行モデルがSMPやWSクラスタにおいて有効であることが示された。今後はWSやSMPの高速なネットワークによる結合や、大規模な結合に関して評価を行なっていく。

参考文献

[1] 林 憲一, 小林 健一, 堀江 健志. AP1000互換通信ライブラリ: WSクラスタ向けの新しい計算 環境 – 基本コンセンプト –. 情報処理学会第52回全国大会. 情報処理学会, 1996.