5G−7　高並列計算機AP1000の性能評価ツール

井川 英子　堀江 健志
(株)富士通研究所

1　はじめに

スーパコンピュータをはじめ多くのシステムでプログラムの性能を解析するツールが開発されている[1][2]. 特に, 並列計算機で動作するプログラムの性能を解析するツールは, 並列プログラムを開発するうえでなくてはならないものである.

近年, 100台以上のプロセッサ構成の並列システムが広く使われるようになってきている. しかし, 並列プログラムの解析ツールはいずれも少数台数のシステムを対象にしており, そのまま大規模システムに適用するのは困難である.

100台以上のプロセッサから構成される並列システムでは, 従来の並列プログラム解析ツールにはなかった問題として以下の事柄が考えられる.

- 実行時にプロセッサの状態や稼働率を表示するためには, 各プロセッサから必要な情報を収集する必要がある. 情報収集のために大きな負荷がかかってしまい, 実行順序に大きな乱れが生じてしまう.
- トレースデータが巨大になり, その収集, 解析に非常に時間がかかる. トレースデータの効率の良い収集と解析の方法が必要である.
- プロセッサの台数が多くなると表示することさえ困難になり, そこから解析に必要なデータを検出するとこはさらに困難になる.

我々は, 高並列計算機AP1000用の性能解析ツールを開発した. 本論文では, AP1000上に実現した性能解析ツールと上記問題をどのように解決しているかについて述べる.

2　AP1000アーキテクチャ

AP1000は富士通研究所で開発された分散メモリ型のメッセージ通信を基本とした並列計算機である[3]. プロセッシングエレメント（セル）は, 16から1024台まで接続可能である.

2.1　ハードウェア

図1にAP1000の全体構成を示す.

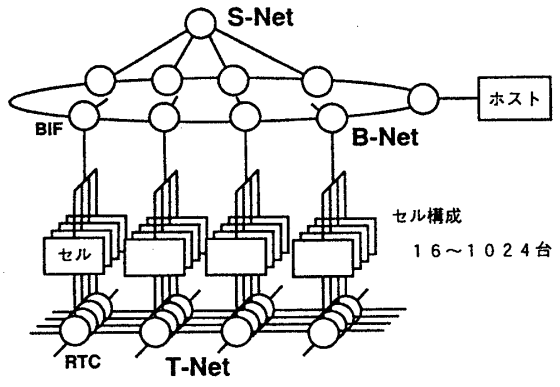

図 1: AP1000 アーキテクチャ

AP1000は, ホストと多数のセルが三種類のネットワークによって結合している.

T-Net　1対1通信を行なう.

B-Net　ホストとセル間, あるいはセル間の放送, 分散, 収集通信を行なう.

S-Net　バリア同期. ステータスの検出を高速に行なう.

さらに, メッセージの送信あるいは受信におけるメッセージハンドリングのオーバヘッドを削減するためにメッセージコントローラ (MSC) が搭載されている.

2.2　ソフトウェア

ユーザは, ホスト用プログラムとセル用プログラムを別々に記述する. 通信, 同期といった並列処理に必要な機能はライブラリを呼び出すことにより記述される.

*CAREN*と呼ばれるサーバは, セルの初期化, 実行環境設定, セルプログラムの起動（タスク生成）, メッセージの送信あるいは受信などの処理を行なう. ユーザのホストプログラム, セルプログラムのデバッガ, ランタイムモニタ, ユーザインタフェース用のプロセス *UIO* などのプロセスはサーバとパイプで通信を行なう.

各セルには, きわめて軽い, メッセージ通信を基本としたOS（セルOS）が載っている. セルOSは, 各セルで基本的には独立に動作する. そして解析ツールのログ情報は, セルOS内でとられる.

我々は, AP1000用の性能解析ツールとして, ランタイムモニタとパフォーマンスアナライザを開発した. ランタイムモニタは, 実行時に各セルのタスクの状態や稼働率を表示する. パフォーマンスアナライザは, 実行中にトレース情報を各セルとホストで蓄積し, 実行後, トレース情報を解析し表示する.

なお, 性能解析のための表示はすべて *X-Window* を使用している.

3　ランタイムモニタ

ランタイムモニタは, 実行時にセル内のタスク状態と負荷を一定間隔ごとにホストプロセッサに収集する. タスク状態とは, 実行中, レディー, メッセージ待ち, バリア同期待ちという状態である.

ランタイムモニタにおけるデータの収集には, AP1000のネットワークの一つであるB-Netのデータの収集の機能（ギャザ）を用いる. 通常であれば, 各セルからの別々のデータがホストにあるサーバに送信される. ホスト側では受信処理をセル毎に行なうことになるので, そのオーバヘッドはセル台数に比例してしまう. ギャザは, ネットワーク内ですべてのセルのデータを結合する. ホスト側が受信するときは, 一つのメッセージとして受信することができるので, ホスト側の受信のオーバヘッドは, セル台数に依存しないでほぼ一定のままになる. このようなすべてのセルからのデータ収集機能は並列システムの大規模化に伴いなくてはならない機能と考えられる.

収集されたデータは, サーバプロセスを介してランタイムモニタのプロセスに渡される. ランタイムモニタは, セルの負荷を表示するロードプロセス, ホストとセルの状態を表示するステータスプロセスから構成される. ステータスプロセスは, セルにロードされたタスクごとに生成される.

なお, ランタイムモニタは, セルのタスクの状態を表示するだけではなく, 各タスクの標準出力の表示, ソースデバッガの起動が可能である.

Performance Analyzer for a Highly Parallel Computer AP1000
Hideko IKAWA, Takeshi HORIE
FUJITSU LABORATORIES Ltd.

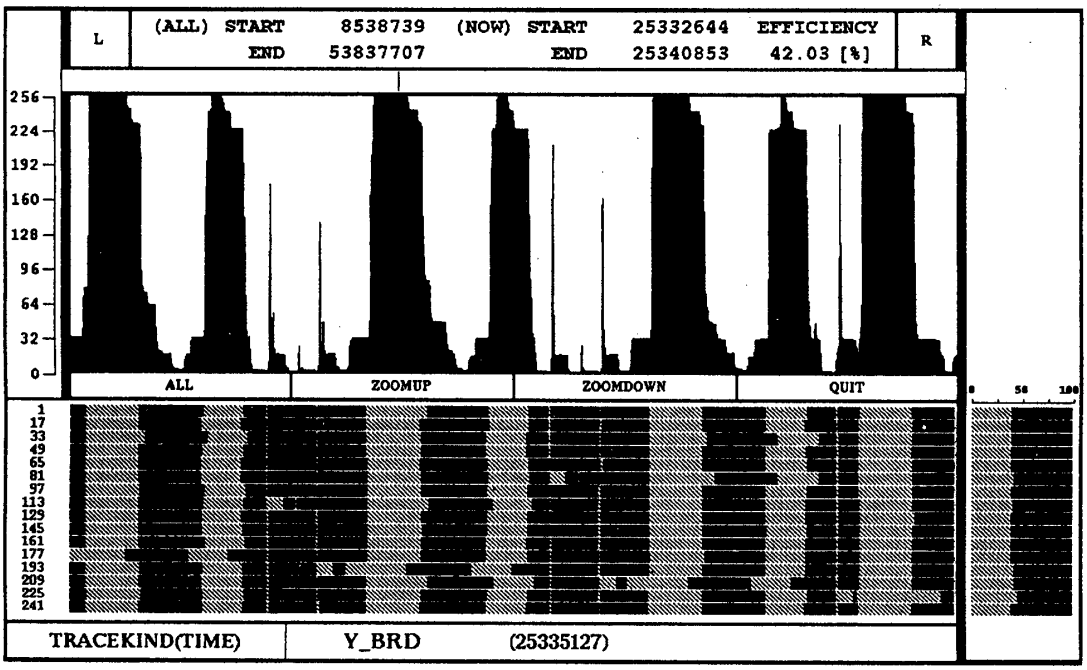

図 2: ロードモニタとタイミングトレース

4 パフォーマンスアナライザ

パフォーマンスアナライザは, 実行中にトレース情報を各セルとホストで蓄積し, 実行後, トレース情報を解析し表示するツールである.

パフォーマンスアナライザは二つの動作モードを持つ. 実行後, トレース情報をホストのファイルに収集して解析するモード (オフラインモード), 実行後, トレース情報はセルのメモリに置いたままにしておきトレース解析プログラムをあたらにユーザプロセスとして生成し, そのユーザプロセスがセル側にセルのトレース解析プログラムをロードするモード (オンラインモード) である. オフラインモードは, トレース情報をホストのファイルとして収集するので, AP1000 を使用しないで解析することができる. オンラインモードは, トレースデータをセルに残したまま並列に解析するので, セル台数が多くても, また, トレースの情報が大きくても効率良く解析できる.

パフォーマンスアナライザの機能には, 統計解析, タイミングトレース, ロードモニタ, メッセージモニタ, クリティカルパス解析がある. 統計解析は, 個々のイベントから集計したイベントの種類と時刻, ライブラリコールの種類とそれぞれの回数, 各タスクの実行時間, アイドル時間, 割り込みの時間, メッセージの転送量, 送信先と送信元を数値として表示する. タイミングトレースは, イベント情報を映像化する. ロードモニタは, すべてのセルの稼働率を表示する. メッセージモニタは, メッセージの転送量を表示する. クリティカルパス解析は, クリティカルパスを発見し, それを映像化する.

4.1 ロードモニタとタイミングトレース

統計情報や個々のセルの稼働状態だけを解析しても実行全体のボトルネックとなっている箇所を発見することは困難である. ロードモニタは, すべてのセルの稼働率を表示することにより実行のボトルネックとなっている箇所を調べることが可能である.

ロードモニタは, トレースの種類, タスク番号, 時刻を元にすべてのプロセッサの稼働率を表示する. セルが稼働状態になっているときは, 並列処理のライブラリ, 割り込み, ナルタスクの実行を除いた時間を意味する.

次に, 表示のアルゴリズムについて述べる. 表示において, ロードの変化が表示の画素よりも細かい可能性がある. そこで, 1 画素に対応する開始時間と終了時間を定めて, その時間での各セルの稼働率を計算する. オンラインモードでは, 各セルが並列に計算する. その結果をネットワークを介して合計してホストに送信する. この処理を表示する画素分繰り返し, すべてのセルの稼働率を求める.

図 2 にロードモニタとタイミングトレースの表示例を示す. この例では, 256 台のプロセッサを使って LU 分解により連立一次方程式を解いている.

指定した範囲の平均の稼働率 (EFFICIENCY) が図の右上に表示されている. 図の上半分にロードモニタ, 下半分にタイミングトレースが表示されている. タイミングトレースにより, トレースの種類が時間軸に発生イベントごとに色分けして表示され, また, トレースの種類の割合が時間軸の右に表示されている. 指定されたライブラリの種類, 割り込みの種類などのイベントの詳細な種類も TRACEKIND の欄に表示することができる.

なお, 時間軸は, ロードモニタもタイミングトレースも同じである. ロードモニタで, 並列性の低い箇所を見つけ, その原因をタイミングトレースで探すことができる.

5 おわりに

本稿では, 実行時に並列計算機の稼働状況を表示するランタイムモニタと, 実行終了後, 稼働状況の情報を各プロセッサが並列に解析し表示するパフォーマンスアナライザについて述べた.

これらの性能解析ツールを用いることにより, 非常に多数のデータから並列プログラムの開発に必要な情報だけを効率良く得ることができる.

今後は, メッセージのスケジューリング等, 通信状況を詳細に解析できるように, セル間やホストとセル間の通信をわかりやすく表示する予定である.

6 謝辞

本プロジェクトに御協力賜わる (株) 富士通ソーシアルサイエンスラボラトリの諸兄に深謝します.

参考文献

[1] A.D.Malony, J.L.Larson, D.A.Reed, *Tracing Application Program Execution on the Cray X-MP and Cray 2*, Supercomputing, 1991, pp. 60-73.

[2] A.D.Malony, D.H.Hammerslag, D.J.Jablonowski, *TraceView: A Trace Visualization Tool*, IEEE Software, September, 1991.

[3] 石畑他, 高並列計算機 *CAP-II* の構成とメモリシステム, 情報処理学会研究会資料, 1990, 83-37, pp. 217-222.