

ハードウェアトレーサを用いた性能解析ツールの概要

7P-1 片山 透* 米田 潔* 中本 幸一* 小川 正吾** 岩永 雅洋**

*NEC マイコンソフトウェア開発研究所

**NECマイコンテクノロジー 共通ソフトウェア部

1. はじめに

リアルタイムシステムの開発では、正しく動作しているかという「論理検証」に加えて、そのシステムが要求性能を満たしているかどうかという「性能検証」を行なうことが重要である。ここでいう「性能検証」とは、システムを実時間で動作させたときの処理時間やタイミングを測定し、設計目標を達成しているかどうかを確認することである。

しかし、リアルタイムシステムの実時間の性能解析を行なうツールは現状ではまだ少ない。あるとしても、それは特定のOSに依存したものである^[1]。

そこで我々はリアルタイムシステムの実時間の性能解析を行なえ、かつ、OSやCPUに依存しない性能解析ツールを開発した。ここでは、そのツールの概念を報告する。

2. 開発目標

リアルタイムシステムの性能解析を行なうためのツールとして、リアルタイムシステムの動作が分かり、異常動作の原因が究明できることが必要であると感じ、それを達成するために、以下の項目の実現を目標とした。

- ◎ タスクの振舞いを観測できる
→ 異常動作の原因究明のため
- ◎ ターゲットのソフトウェアに手を加えない
→ リアルタイムな動作を測定するため
- ◎ ツールを汎用とする
→ CPU, OSに依存しないようにするため

3. システム構成

我々は、上記の開発目標を達成するために、CPUのバスデータをトレースし、これを使って性能解析することを考え、図1のような構成の性能解析ツールを開発した。

このツールを使った解析手順を述べる。データ採取に際しては、ワークステーション上からトレーサをコントロールし、CPUプローブを介して採取されたCPU

のバスデータをトレーサのメモリに記憶する。そのデータに対し、ワークステーション上でソフトウェア解析ツールを用いて解析を行なう。採取したデータはトレーサのメモリに記憶されている。そのメモリから直接ワークステーション上の解析ツールで解析を行なうこともできる。また、ワークステーションのディスクにデータを移して解析することもできる。あるいは、DATを用いて大量のデータを連続して解析するような手法も考えられる。

4. システムの特徴

上記システム構成において、開発目標を達成するため、以下の項目について検討した。

4.1 トレース方式

一般にトレース方式には、ソフトウェアのみを使用したトレース、ハードウェアを使用したトレースの二つの方式が考えられる。各々の特徴は以下の通りである。

- ◎ ソフトウェアトレース: 測定対象システムで実行されるプログラムの中に、測定命令を埋め込みトレースする
→ レジスタ、スタックの内容をトレース可能
- ◎ ハードウェアトレース: ソフトウェアにトレースのための仕掛を入れず、ハードウェアで直接トレースする
→ 実機の性能を落さずにトレース可能

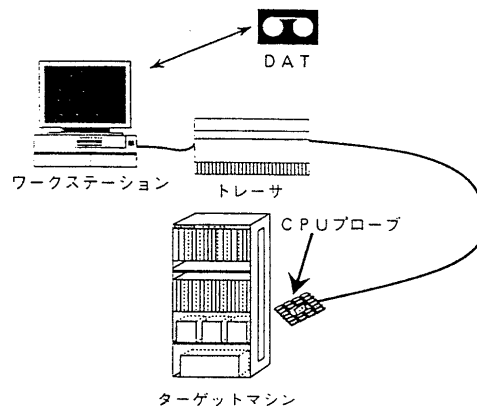


図1 システム構成

Overview of Performance Analyzing Tool with Hardware Tracer
Toru KATAYAMA*, Kiyoshi YONEDA*,
Yukikazu NAKAMOTO**, Shougo OGAWA**,
Masahiro IWANAGA**
*NEC Corporation
**NEC Micro Computer Technology Ltd.

リアルタイムシステムでは実時間でデータをトレースできなければ、正確に性能を解析することは困難である。そこで、本性能解析ツールでは、トレースのためにオーバーヘッドのかかるソフトウェアトレースより、ターゲットの性能を落さないハードウェアトレース方式を採用することとした。

4.2 トレースデータの種類

データ採取時にハードウェアによる特定イベントのフィルタリングを行ない、解析に必要なデータのみを採取することによって、メモリの節約をすることができる。これによって通常のロジックアナライザ等の4~8Kワード程度のメモリ容量でも、解析を行なうことは可能である。

しかし、性能解析において、データ採取の段階でボトルネックとなる要因が明白であることは少ない。また、その要因を解析するためにはどのようなデータが必要であるかは定かではない。もし、データ採取時にフィルタリングを行なうと、ボトルネックの要因を解析するのに必要な情報が欠落して、それを得るために新しくデータを採取し直す必要がある。

従って、本解析ツールでは、上記の問題を解決するために、図2のバスサイクル中の全CPUバスデータをもれなく採取することとした。そうすることによって、データ採取のやり直しなどの必要なく解析を行なうことができる。また、さらに詳細な解析を行なうことが可能となっている。

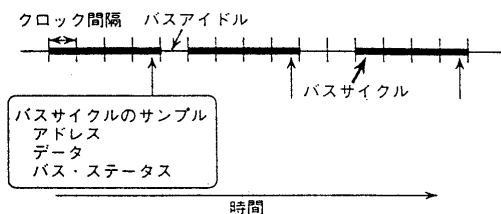


図2 データサンプリングイメージ

また、データ採取時に、フィルタリングを行なわないため、トレースしたい処理を採取するためには大きなメモリ容量を必要とする。そこで、本性能解析ツールでは128bit*8Mワードという通常の1,000倍以上の容量を持たせている。

5. 性能解析機能

リアルタイムシステムの性能解析を行なうために、以下のような解析ツールの機能が必要であると考えた。

5.1 統計処理

性能解析において、従来から考えられていた統計処理データは、CPUの性能を把握する、メモリのアクセス時間を把握する等ハードウェアの性能検証については有効である。本ツールでは解析結果として、これらハードウェアの統計情報を出力する。

更に本ツールではタスクや関数レベルでの統計処理データを解析結果として出力する。関数やタスクの走行時間の最大値等を知ることにより、デッドラインの検出等が可能となり、ソフトウェアの性能検証が容易になる。

5.2 タイミングチャート

上記のような統計処理データを用いることにより、ターゲットのソフトウェアのボトルネック等を発見することは可能である。しかし、統計処理データでは、ソフトウェアのタイミング上の動作を表現できず、システムの振舞いを理解することは困難であり、ボトルネックの原因解明にまでは至らない。

そこで、縦軸にタスクあるいは関数、横軸に時間をとったタイミングチャートが重要になる^{[2][3]}。このタイミングチャートでは、ソフトウェアのリアルタイムな動きを視覚的に表示し、処理の流れを明確に表示することによって、詳細な振舞いを解析することが可能である。これにより、ターゲットシステムのソフトウェアのアルゴリズムやシステム設計についての検証を行なうことが可能となる。

6. おわりに

以上のように、本性能解析ツールには様々な新しい試みがある。まず、ターゲットとなるリアルタイムシステムの特徴を考慮して実時間のデータを採取することを試みた。

また、解析結果については、システムの性能をソフトウェアの側面から解析するために、タスクや関数レベルに着目した統計処理データを解析するツールを開発した。さらに、タスクタイミングチャートによって、システムのソフトウェアの振舞いを視覚的に解析することができるようになった。

【参考文献】

- [1] Tokuda, H. et al.: A Real-Time Monitor for a Distributed Real-Time Operating System, In Proceedings of ACM SIFOPS and SIGPLAN workshop on parallel and distributed debugging, May, 1988.
- [2] 上野 他: 性能解析ツールのユーザインタフェース, 情報処理学会第45回全国大会, 7P-2, 1992.
- [3] 寺峯 他: ハードウェアトレーサを用いた性能解析ツールの適用, 情報処理学会第45回全国大会, 7P-3, 1992.