

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5029245号
(P5029245)

(45) 発行日 平成24年9月19日(2012.9.19)

(24) 登録日 平成24年7月6日(2012.7.6)

(51) Int. Cl. F I
G06F 11/28 (2006.01) G O 6 F 11/28 3 1 O A
G06F 11/34 (2006.01) G O 6 F 11/34 S

請求項の数 6 (全 16 頁)

<p>(21) 出願番号 特願2007-243478 (P2007-243478)</p> <p>(22) 出願日 平成19年9月20日 (2007.9.20)</p> <p>(65) 公開番号 特開2009-75812 (P2009-75812A)</p> <p>(43) 公開日 平成21年4月9日 (2009.4.9)</p> <p>審査請求日 平成22年4月30日 (2010.4.30)</p> <p>前置審査</p>	<p>(73) 特許権者 308014341 富士通セミコンダクター株式会社 神奈川県横浜市港北区新横浜二丁目10番 23</p> <p>(74) 代理人 100107766 弁理士 伊東 忠重</p> <p>(74) 代理人 100070150 弁理士 伊東 忠彦</p> <p>(74) 代理人 100146776 弁理士 山口 昭則</p> <p>(72) 発明者 木村 茂 神奈川県川崎市中原区上小田中4丁目1番 1号 富士通株式会社内</p> <p>審査官 石川 亮</p> <p style="text-align: right;">最終頁に続く</p>
--	---

(54) 【発明の名称】 プロファイリング方法及びプログラム

(57) 【特許請求の範囲】

【請求項1】

所定の条件が満たされた際に発生する割り込みにより起動される割り込みハンドラにより、情報収集の対象となる対象プログラムの割り込み発生時の実行アドレスに対応する情報を収集するコンピュータによるプロファイリング方法であって、

前記対象プログラムに、前記割り込みハンドラを含むプロファイリングプログラムと、コード範囲マーク用シンボルソース又はオブジェクトをリンクさせる際に、前記対象プログラムに対して前記割り込みハンドラにより情報収集を行うターゲット範囲を指定するターゲット範囲指定ステップと、

前記対象プログラムの割り込み発生時の実行アドレスが前記ターゲット範囲内であれば前記割り込みハンドラにより収集される当該実行アドレスに対応するイベントの値をメモリに設定する情報設定ステップと

を含むことを特徴とするプロファイリング方法。

【請求項2】

前記割り込みは、前記コンピュータ内部のイベント及び/又は前記コンピュータ外部とのやり取りのイベントをカウントするハードウェアカウンタのカウント値に基づき発生することを特徴とする請求項1記載のプロファイリング方法。

【請求項3】

前記対象プログラムの割り込み発生時の実行アドレスと割り込み回数と前記メモリに格納されているプログラム定義とに基づいて、プログラム位置と前記割り込み回数を、前記

10

20

プログラム位置と対応するソースプログラムの関数名を対応付けて解析プロセッサの表示部に表示するステップ

を更に含むことを特徴とする請求項 1 又は 2 記載のプロファイリング方法。

【請求項 4】

前記割り込みハンドラにより収集される情報に含まれる前記イベントの情報及びハードウェアイベントの情報をマージして経験値として前記メモリに保持するステップ

を更に含むことを特徴とする請求項 1 乃至 3 のいずれか 1 項記載のプロファイリング方法。

【請求項 5】

コンピュータに、所定の条件が満たされた際に発生する割り込みにより起動される割り込みハンドラにより、情報収集の対象となる対象プログラムの割り込み発生時の実行アドレスに対応する情報を収集するプロファイリングを行わせるプログラムであって、

前記対象プログラムに、前記割り込みハンドラを含むプロファイリングプログラムと、コード範囲マーク用シンボルソース又はオブジェクトをリンクさせる際に、前記対象プログラムに対して前記割り込みハンドラにより情報収集を行うターゲット範囲を指定させるターゲット範囲指定手順と、

前記対象プログラムの割り込み発生時の実行アドレスが前記ターゲット範囲内であれば前記割り込みハンドラにより収集される当該実行アドレスに対応するイベントの値をメモリに設定させる情報設定手順と

を前記コンピュータに実行させることを特徴とするプログラム。

【請求項 6】

命令毎に対応付けた前記メモリの領域に、前記割り込みハンドラにより収集された情報に含まれる前記イベントの実行回数又はイベント発生回数、及びハードウェアイベントの実行回数又はイベント発生回数を設定させる手順

を前記コンピュータに実行させることを特徴とする請求項 5 記載のプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、プロファイリング方法及びプログラムに係り、特に所定の条件が満たされた際に発生する割り込みにより起動される割り込みハンドラによりプログラムの実行状況に関する情報を収集するプロファイリング方法及びプログラムに関する。

【背景技術】

【0002】

コンピュータシステムにおいて、性能分析や最適化等を行う手段としてプロファイリング手法が広く用いられている。プロファイリングは、ターゲットとするプログラムコードの走行頻度、時間分布、プログラム内部の呼出し頻度等の分析に有効である。プログラムを動作マシンで動作させてプロファイリングする手法には、以下の 2 つ手法がある。

【0003】

第 1 の手法は、コンパイラにプロファイリング用コードを挿入させ、実行情報を収集するプロファイリングである。第 1 の手法は、最も一般的に使用されているプロファイリング手法であり、コンパイラ製品に標準機能として実装されている。又、第 1 の手法においてデータの収集を効率化する方法が、例えば、特許文献 1 及び特許文献 2 にて提案されている。

【0004】

第 2 の手法は、ハードウェアタイマ又は CPU の性能モニタリング機構を利用したサンプリングによるプロファイリングである。第 2 の手法では、一定の時間毎に、或いは、プロセッサや周辺回路で計測可能な実行命令数やキャッシュミス数のハードウェアカウンタ等が一定の回数に到達する毎に、サンプリング又はハードウェアイベントの割り込みを発生させ、割り込み処理として登録したプロファイリングプログラムで割り込み発生時の実行命令アドレス等を記録することで、統計的に最も時間を費やしたコード範囲や最も頻繁

10

20

30

40

50

に実行されたコード範囲更に最もハードウェアイベントの発生したコード範囲等を抽出する。第2の手法は、例えば、非特許文献1及び非特許文献2にて提案されている。

【0005】

尚、命令レベルシミュレータにより命令をトレースする手法も、例えば特許文献3にて提案されている。しかし、シミュレータで高アクセスコスト、即ち、実行サイクル数をシミュレートする場合、1命令ずつシミュレータでトレースするためシミュレーションに時間がかかり、シミュレータではアクセスレイテンシの遅延等の実機環境特有の問題に起因する正確な情報を取得できない等の問題があった。

【特許文献1】特開平11-212837号公報

【特許文献2】特開2003-140928号公報

【特許文献3】特開平7-191882号公報

【非特許文献1】「OProfile - A System Profiler for Linux」、[平成18年3月18日検索]、インターネット<URL: <http://sourceforge.net/projects/oprofile/>>

【非特許文献2】「インテル VTune パフォーマンス・アナライザ」、[平成18年3月18日検索]、インターネット<URL: <http://www.intel.com/cd/software/products/ijkk/jpn/vtune/index.htm>>

【発明の開示】

【発明が解決しようとする課題】

【0006】

シミュレータではなくプログラムを動作マシンで動作させてプログラムの実行状況に関する情報を取得するプロファイリング手法の場合、以下の如き問題があった。

【0007】

上記第1の手法の場合、対象とするコード全般にプロファイリング用処理が付加されるために時間的オーバーヘッドが生じるか、或いは、コード挿入のために本来のプログラムバイナリと動的、メモリ配置的に差異が生じてしまった。特に、通信等のタイミングが重要なプログラムでは本来の動作と異なる場合が多く、必要精度により使用が制限される場合があった。

【0008】

又、上記第2の手法の場合、対象とするコードに対する改変は必要ないためオーバーヘッドの問題やメモリ配置の問題は抑制可能である。しかし、この第2の手法はプロファイリングプログラムで割り込み発生時の実行命令アドレス等を記録する領域の確保が必要であり、組み込み装置等の場合のようにメモリ資源が乏しい環境では実現できていなかった。

【0009】

そこで、本発明は、少ないオーバーヘッドでプログラムの実行状況に関する詳細な情報の収集を可能とするプロファイリング方法及びプログラムを提供することを目的とする。

【課題を解決するための手段】

【0010】

上記の課題は、所定の条件が満たされた際に発生する割り込みにより起動される割り込みハンドラにより情報収集の対象となる対象プログラムの実行状況に関する情報を収集するコンピュータによるプロファイリング方法であって、該対象プログラムに対して該割り込みハンドラにより情報収集を行うターゲット範囲を指定するターゲット範囲指定ステップと、該ターゲット範囲で該割り込みが発生した際に該割り込みハンドラにより収集される情報をメモリに設定する情報設定ステップとを含むことを特徴とするプロファイリング方法により達成できる。

【0011】

上記の課題は、コンピュータに、所定の条件が満たされた際に発生する割り込みにより起動される割り込みハンドラにより情報収集の対象となる対象プログラムの実行状況に関する情報を収集するプロファイリングを行わせるプログラムであって、該対象プログラムに対して該割り込みハンドラにより情報収集を行うターゲット範囲を指定させるターゲット範囲指定手順と、該ターゲット範囲で該割り込みが発生した際に該割り込みハンドラに

10

20

30

40

50

より収集される情報をメモリに設定させる情報設定手順とを該コンピュータに実行させることを特徴とするプログラムにより達成できる。

【発明の効果】

【0012】

本発明によれば、組み込み装置等のメモリ資源が乏しいメモリ制約が大きい装置においても少ないオーバーヘッドでプログラムの実行状況に関する詳細な情報の収集を可能とするプロファイリング方法及びプログラムを実現することができる。

【発明を実施するための最良の形態】

【0013】

本発明のプロファイリングは、所定の条件が満たされた際に発生する割り込みにより起動される割り込みハンドラにより情報収集の対象となるプログラムの実行状況に関する情報を収集する。具体的には、プロファイリングは、割り込みハンドラにより情報収集を行うターゲット範囲を指定し、指定されたターゲット範囲で割り込みが発生した際に割り込みハンドラにより収集される情報を設定する。

【0014】

例えば、ハードウェアカウンタの、データキャッシュミス発生ハードウェアイベントを使用すれば、キャッシュミスの発生した命令で割り込みが発生するため、キャッシュミスの命令を解析してキャッシュミスのアクセス先を解析することができる。

【0015】

本発明によれば、組み込み装置等のメモリ資源が乏しい等のメモリ制約が大きい装置においても、全ての実行プログラムについて、若しくは、一部の実行プログラムについてプロファイリング対象を絞り込むことで実行頻度情報を取得でき、従来は取得できなかったチューニング情報の取得が可能となる。又、実行プログラムの少なくとも一部について、ハードウェアタイマやハードウェアイベントに着目したプロファイリング対象を絞り込むことで、イベント情報を取得でき、従来は取得できなかったチューニング情報の取得が可能となる。更に情報収集の対象となるプログラムにプロファイリングプログラムをリンクする時に測定対象を絞り込めるため、プログラムの再翻訳が不要となり、プロファイリングを容易に行なうことができると共に、利用者の負担も軽減できる。

【0016】

以下に、本発明のプロファイリング方法及びプログラムの各実施例を図面と共に説明する。

【実施例】

【0017】

先ず、本発明が適用可能な電子装置の構成を、図1乃至図3と共に説明する。

【0018】

図1は、本発明が適用可能な電子装置の構成を示す図である。図1に示すように、電子装置1は、コンピュータハードウェア(又は、プロセッサ)10と、オペレーティングシステム20と、アプリケーションプログラム(又は、実行形式ファイル)30と、プロファイリングプログラム100とが連携して情報を収集する構成を有する。尚、オペレーティングシステムのない環境でもオペレーティングシステムの代替機構を用意することで本発明は実現できる。そのため以降の説明では、オペレーティングシステム20のある例を示す。電子装置1の基本構成は、周知の汎用コンピュータシステムと同様の基本構成を採用可能である。

【0019】

コンピュータハードウェア10は、プロファイリングプログラム100を含めてプログラムを実行する装置であり、オペレーティングシステム20により設定された条件が満たされた場合、即ち、オペレーティングシステム20により指定されたイベントが発生した場合に割り込みを発生して予め登録された割り込み処理プログラムを実行する。コンピュータハードウェア10は、CPU等のプロセッサで構成されたプロセッサコアと、ハードウェアカウンタと、内蔵タイマと、キャッシュメモリ等の記憶部を有する。

【 0 0 2 0 】

オペレーティングシステム 20 は、コンピュータハードウェア 10 のリソースを管理し、アプリケーションプログラム 30 の実行制御等を行う制御プログラムであり、コンピュータハードウェア 10 に対して割り込み発生イベントの設定等を行う。

【 0 0 2 1 】

アプリケーションプログラム 30 は、情報収集の対象（又は、チューニング対象）となるプログラムであり、プロファイリングプログラム 100 が提供する API を用いてライブラリ関数（又は、プロファイリング開始関数）を呼び出すことにより情報収集に必要な処理を実行する。本実施例では、アプリケーションプログラム 30 に、割り込みハンドラ 140 を含むプロファイリングプログラム 100 と、コード範囲マーク用シンボルソース又はオブジェクトを連携（即ち、リンク）させる。このとき、プロファイリングしたいプログラムや関数のみを、ターゲット範囲（又は、測定範囲）として任意に抽出して指定する。ターゲット範囲は、コード範囲マーク用シンボルで対象のコード部を挟むことで指定する。プロファイリングでは、対象のコード部と同じサイズのメモリがコンピュータハードウェア 10 内に必要となるため、このようにターゲット範囲を指定することにより、たとえ電子装置 1 がメモリ資源が乏しい等のメモリ制約が大きい組み込み装置であっても、ターゲット範囲を絞ったプロファイリングが可能となる。

【 0 0 2 2 】

尚、以下の説明では、プロファイリング値を格納するメモリ内の領域がコードサイズと同じサイズであり、プロファイリング値を格納する領域サイズと 1 命令コードサイズが同じである場合を例に取るが、本発明の適用はそのような場合に限られるものではない。つまり、命令の長さにかかわらず、命令毎に対応するプロファイリング値を格納する領域を用意すれば良い。

【 0 0 2 3 】

プロファイリングプログラム 100 は、アプリケーションプログラム 30 の情報収集を行うプログラムであり、初期処理ルーチン 110 と、ターゲット範囲指定インタフェース 120 と、イベント設定インタフェース 130 と、割り込みハンドラ 140 と、イベント管理テーブル 150 と、割り込みハンドラ記録テーブル 160 とを有する。本発明のプログラムは、本実施例ではこのプロファイリングプログラム 100 に相当する。又、本発明のプロファイリング方法は、本実施例ではこのプロファイリングプログラム 100 を実行するコンピュータによるものである。

【 0 0 2 4 】

初期処理ルーチン 110 は、オペレーティングシステム 20 に対して、サンプリング割り込みを発生するタイマの設定や、サンプリング割り込みが発生した際に起動される割り込みハンドラ 140 の登録等を依頼する。尚、サンプリング割り込み発生イベントとしては、タイマによる一定時間の経過の他に、一定回数の命令実行や一定回数の特定アドレスへのアクセス、更にキャッシュミス発生等のプロセッサ性能モニタリング機構が割り込みを発生するイベントを設定することができる。

【 0 0 2 5 】

ターゲット範囲指定インタフェース 120 は、アプリケーションプログラム 30 が測定開始を指定する開始指定 API にて実行される。ターゲット範囲指定インタフェース 120 は、情報収集のターゲット範囲として関数名或いは実行コードに付されたラベルが指定された場合、該当関数（又は、ラベルを含む関数）の開始アドレスと終了アドレスを、スタートアップルーチンを含むオペレーティングシステム 20 及びアプリケーションプログラム 30 より取得してイベント管理テーブル 150 に登録する。本実施例では、上記の如く、アプリケーションプログラム 30 に、割り込みハンドラ 140 を含むプロファイリングプログラムと、コード範囲マーク用シンボルソース又はオブジェクトをリンクさせているので、プロファイリングしたいプログラムや関数のみを任意に抽出して指定することでターゲット範囲を指定でき、たとえ電子装置 1 がメモリ制約の厳しい組み込み装置であってもターゲット範囲を絞ったプロファイリングが可能となる。

【 0 0 2 6 】

イベント設定インタフェース 1 3 0 は、アプリケーションプログラム 3 0 からイベント値に関する引数を受け取り、情報収集のターゲット範囲に対するイベント値を設定するライブラリ関数であり、アプリケーションプログラム 3 0 がイベント値の指定に使用する A P I を提供する。

【 0 0 2 7 】

ここで、イベントとは、情報収集のターゲット範囲内で割り込みハンドラ 1 4 0 が起動された場合に割り込みハンドラ 1 4 0 により収集される値に関連する属性(変数)である。又、イベント値としては複数の値を設定することができる。このイベント設定インタフェース 1 3 0 が、アプリケーションプログラム 3 0 からイベント値に関する引数として受け取り、情報収集のターゲット範囲に対するイベント値を設定することにより、アプリケーションプログラム 3 0 の実行状況に関する様々な情報を収集することができる。たとえば一定時間ごとの割り込みを発生させる場合、イベントの属性としては割り込み間隔を指定する。アプリケーションプログラム 3 0 から、開始指定 A P I のパラメタとして割り込み間隔を指定する。

10

【 0 0 2 8 】

割り込みハンドラ 1 4 0 は、タイマによるサンプリング割り込みやプロセッサ性能モニタリング機構によるサンプリング割り込みにより起動され、割り込み発生時の実行アドレスを割り込みハンドラ記録テーブル 1 6 0 に記録する割り込み処理プログラムであり、イベント管理テーブル 1 5 0 に登録された情報収集のターゲット範囲内で起動されると、割り込み発生時の実行アドレスに対応するイベント値を割り込みハンドラ記録テーブル 1 6 0 に記録する。たとえば一定時間に割り込みを発生させる場合では、割り込み発生プログラムカウンタに対応させた領域に、割り込み回数を割り込み記録テーブル 1 6 0 に格納する。

20

【 0 0 2 9 】

イベント管理テーブル 1 5 0 は、アプリケーションプログラム 3 0 の情報収集に関する情報を格納する。図 2 は、イベント管理テーブル 1 5 0 の一例を示す図である。図 2 に示すように、このイベント管理テーブル 1 5 0 は、アプリケーションプログラム 3 0 の情報収集のターゲット範囲毎に、ターゲット範囲の開始アドレスと、ターゲット範囲の終了アドレスと、イベント値を対応させて格納する。

30

【 0 0 3 0 】

例えば、情報収集のターゲット範囲は、開始アドレスが「 1 0 0 0 」であり、終了アドレスが「 2 0 0 0 」であり、ターゲット範囲内で割り込みハンドラ 1 4 0 が起動されると割り込みハンドラ 1 4 0 によりイベントとして割り込み回数が情報として収集される。

【 0 0 3 1 】

割り込みハンドラ記録テーブル 1 6 0 には、サンプリング割り込みにより割り込みハンドラ 1 4 0 が起動された場合に割り込みハンドラ 1 4 0 により収集された情報が記録される。図 3 は、割り込みハンドラ記録テーブル 1 6 0 の一例を示す図である。図 3 に示すように、この割り込みハンドラ記録テーブル 1 6 0 には、サンプリング割り込みが発生した時点で、割り込み時実行アドレスが情報収集のターゲット範囲内である場合にアプリケーションプログラム 3 0 の実行アドレスである割り込み時実行アドレスに対応付けられたイベント値(例えば割り込み回数)が記録される。収集した情報の詳細な方法は、図 6 での 5 1 B で示す方法で格納する。

40

【 0 0 3 2 】

例えば、サンプリング割り込みが発生した時点でのアプリケーションプログラム 3 0 の実行アドレスが「 1 2 0 0 」である場合、図 2 に示した情報収集のターゲット範囲がイベント管理テーブル 1 5 0 の測定範囲であると、割り込み時実行アドレスが情報収集のターゲット範囲となるので、実行アドレス「 1 2 0 0 」に対応付けられたとイベント値(割り込み回数)として「 + 1 」した値が記録される。

【 0 0 3 3 】

50

次に、本実施例の処理手順を、図4と共に説明する。図4は、本実施例の処理手順を説明するフローチャートである。図4に示すように、このプロファイリングシステムにおいて、ステップS101では、プロファイリングプログラム100の初期処理ルーチン110がタイマ設定等のサンプリング割り込み発生イベントに関する設定及び割り込みハンドラ140の登録をオペレーティングシステム20に対して依頼する（オペレーティングシステムのない環境では、代わりに登録する）。

【0034】

ステップS102では、オペレーティングシステム20が、タイマ設定等のサンプリング割り込み発生に必要な設定をコンピュータハードウェア10に対して行う（オペレーティングシステムのない環境では、代わりに設定する）。これにより一定の時間毎に、或いは、プロセスイベントの一定回数毎等にサンプリング割り込みが発生し、割り込みハンドラ140が呼ばれる。

10

【0035】

ステップS103では、アプリケーションプログラム30がユーザにより埋め込まれたターゲット範囲指定インタフェース120の呼出しを行うと、ターゲット範囲指定インタフェース120は、情報収集のターゲット範囲に関する情報をイベント管理テーブル150に登録する。

【0036】

ステップS104では、アプリケーションプログラム30がユーザにより経路情報、変数状態等のイベントへの設定のためにイベント設定インタフェース130の呼出しを行うと、イベント設定インタフェース130はイベント値をイベント管理テーブル150に記録する。尚、ユーザは、イベント値の対象となる情報収集の範囲を指定する。

20

【0037】

ステップS105では、タイマ割り込み等により割り込みハンドラ140が起動されると、割り込みハンドラ140は、割り込み時実行アドレスを割り込みハンドラ記録テーブル160に記録し、そのアドレスが情報収集のターゲット範囲内であればアドレスに対応するイベント値を記録する。

【0038】

このように、イベント設定インタフェース130がイベント値をイベント管理テーブル150に記録し、割り込みハンドラ140が割り込み時実行アドレスが情報収集のターゲット範囲内であればイベント値を割り込みハンドラ記録テーブル160に記録することにより、イベント値に設定された経路情報や変数状態等の詳細な情報を収集することができる。イベント管理テーブル150や割り込みハンドラ記録テーブル160内のデータは、コンピュータハードウェア10内のメモリに格納される。

30

【0039】

メモリに格納してあるサンプリング値を含むデータは、電子装置1に接続可能な外部装置である解析プロセッサ（図示せず）のデバッグ等に取り込まれる。デバッグ等により取得されたサンプリング値を格納したデータとアプリケーションプログラム30のシンボル情報を照らし合わせるプログラムにより、高コスト関数、高コスト処理や高コスト命令毎にサンプリング値が解析プロセッサの表示部に表示される。これにより、アプリケーションプログラム30でのボトルネック箇所を特定することができる。

40

【0040】

次に、プロファイリングのターゲット範囲の指定方法とサンプリング値のメモリ内の格納領域について、より詳細に説明する。

【0041】

上記の如く、本実施例では、プロファイリングはサンプリングしたアドレスではなく、対応するアドレスのカウント値を設定する。又、アプリケーションプログラム30にプロファイリングプログラム100をリンクする際に、指定したコード範囲マーク用シンボルを設定する。このコード範囲マーク用シンボルのシンボル名は任意であり、例えばTOP_LABEL、BOTTOM_LABELをコード部配置領域の開始位置及び終端位置のアドレスとする。TOP_LABEL

50

BEL, BOTTOM_LABELを定義したソース又はオブジェクトを、情報収集の対象となるアプリケーションプログラム30のプロファイリングプログラム100へのリンク時に結合する。

【0042】

情報を収集したいアプリケーションプログラム30の箇所は、モジュール又は関数単位でセクションに出力され、コンパイラで自動的に出力されるか、或いは、ユーザが明示的にセクションの切り分けを行なう。尚、全てのコード部を1つのターゲット範囲に設定することもできる。

【0043】

上記リンク時のターゲット範囲の指定(リンクコマンド文字列)は、例えば次の通りである。

(a) -sc TOP_LABEL,*/code, BOTTOM_LABEL,...,*,WORK_AREA

(b) -sc TOP_LABEL = 0x00000100, BOTTOM_LABEL = 0x00001100, WORK_AREA = 0x01000000

リンク時のターゲット範囲の指定(GUI指定方法)は、例えば図5に示すようになる。図5は、セクション名を表示したリスト一覧から、TOP_LABEL(S)とBOTTOM_LABEL(E)を指定する場合を示す。

【0044】

上記の方法では、ユーザがリンク時に情報収集する対象となるアプリケーションプログラム30を設定する。ユーザは、アプリケーションプログラム30で使用するサンプリング割り込み回数の格納領域のサイズと、このチューニング対象となるアプリケーションプログラム30を見極める必要がある。通常、チューニングすべきプログラムは判明しており、又、不明な場合でも範囲を順次絞ることでチューニング対象を絞ることができる。

【0045】

上記の如く、プロファイリングプログラム100をリンクしたアプリケーションプログラム30は、電子装置1の実装環境において実行される。プロファイリングプログラム100の割り込みハンドラ140では、内蔵タイマの割り込み毎にプログラムカウンタやプロセッサ状態を示すハードウェアカウンタの値をメモリに格納する。割り込み処理とサンプリング値の格納方法と、サンプリング値が格納されるメモリの領域の関係は、例えば図6に示すようになっている。図6は、プロファイリングのターゲット範囲の指定方法とサンプリング値のメモリ内の格納領域を説明する図である。図6中、51Aはコンピュータハードウェア10のメモリ内のコード部配置領域、51Bはコンピュータハードウェア10のメモリ内の書き込み可能領域(即ち、ワークエリア)である。

【0046】

ステップS1では、サンプリング割り込みを発生するタイマを起動する。ステップS2では、割り込み回数カウント領域を、ターゲット範囲のサイズL分だけゼロクリアする。ここで、Lは次式に示すようにシンボルアドレスの差から求める。

【0047】

$$L = (\text{BOTTOM_LABELのアドレス}) - (\text{TOP_LABELのアドレス})$$

ステップS3では、起動されたタイマにより一定間隔で割り込み発生させる。割り込み間隔は、開始指定APIで指定する。ステップS4では、コード部配置領域51Aの開始位置TOP_LABELのアドレスからの相対アドレスを求め、書き込み可能領域51Bの先頭からこの相対アドレスに対応する位置に割り込み回数データを「+1」だけインクリメントしたデータを格納する。ステップS5では、タイマを終了し、メモリにある割り込み回数データがデバッガ等に取り込まれる。

【0048】

尚、本発明は、プロセッサ内部のイベント又はプロセッサ外部とのやり取りのイベントをカウントすることによりプロセッサ性能をモニタリングするハードウェアカウンタを有するプロセッサにも適用することができる。この場合、上記の如き一定時間毎に発生する割り込みを情報収集のトリガとしても良く、更に、ハードウェアカウンタの任意のイベント発生時の状態をトリガとしても良い。具体的には、プロセッサの実行サイクル数を指標として情報を取得するケースを、ハードウェアカウンタの例えばデータキャッシュミス発

10

20

30

40

50

生イベントに置き換える。この場合、キャッシュミスの発生した命令で割り込みが発生するため、キャッシュミスの命令を解析してキャッシュミスのアクセス先を解析することができる。このように、一定時間で発生するハードウェアカウンタやハードウェアカウンタのハードウェアイベントの発生をトリガにしても良い。

【 0 0 4 9 】

ハードウェアイベントの発生は、キャッシュミス数、TLB (Translation Lookaside Buffer) ミス数、実行命令数、実行命令並列度、分岐命令数、パイプラインストール要因、レジスタ干渉サイクル数、バスアクセス情報等のプロセッサのハードウェア情報を取得するためのイベントカウンタをプロセッサ側に設けることで検出可能である。

【 0 0 5 0 】

図 7 は、タイマ割り込み発生時の処理を説明するフローチャートである。図 7 において、タイマ割り込みが発生すると、ステップ S 1 1 は、今回のタイマ割り込み時のプログラムカウンタ (PC) の値を求める。ステップ S 1 2 は、プログラムカウンタの値がターゲット範囲内であるか否か、即ち、(開始位置TOP_LABEL) < PC < (終了位置BOTTOM_LABEL) であるか否かを判定する。ステップ S 1 2 の判定結果が YES であると、ステップ S 1 3 は、開始位置TOP_LABELのアドレスからの相対アドレス を求め、書き込み可能領域の先頭からその相対アドレス に対応する位置に割り込み回数データを「+1」だけインクリメントしたデータを格納し、処理は呼び出し元に復帰する。

【 0 0 5 1 】

図 8 は、解析プロセッサの処理を説明するフローチャートである。解析プロセッサ (図示せず) が電子装置 1 に接続されて解析処理が開始されると、図 8 において、ステップ S 2 1 は、アプリケーションプログラム 3 0 の割り込みアドレスと割り込み回数を電子装置 1 内のメモリから取り込む。ステップ S 2 2 は、電子装置 1 内のメモリに格納されているプログラム定義や割り込み回数等を読み取って、プログラム位置と割り込み回数をグラフ等の形式で解析プロセッサの表示部に表示する。プログラム位置は、対応するソースプログラムの関数名と照合して表示される。プログラム位置とソースプログラムの関数名の対応付けには、プログラムファイルのシンボル情報やデバッグ情報等を利用可能である。このようにして、例えば図 9 に示す如きプロファイリングの結果が表示される。

【 0 0 5 2 】

図 9 は、プロファイリングの結果の表示例を示す図である。図 9 中、ウィンドウ W 1 内の「38422」等の値は割り込み回数を示し、「FuncE」等は関数、「184」はメモリ領域のサイズを示す。例えば、ウィンドウ W 1 内の関数をダブルクリックすることで、対応する C ソースやアセンブラソースウィンドウ W 2 に分岐する。アセンブラソースウィンドウ W 2 内には、アドレスと対応する割り込み検出回数、関数FuncE内での情報収集対象プログラムの実行頻度 (即ち、実行状況) が表示されている。図 9 に示す表示例の場合、割り込みが高頻度の関数順に表示がなされ、関数内でのアセンブラ命令単位にアクセスコスト、即ち、割り込み回数が表示され、対応する関数、命令位置をクリック等で照合することで対応する C ソースやアセンブラソースウィンドウ (デバッガ、エディタ、ツール等) に分岐することができる。

【 0 0 5 3 】

図 8 の説明に戻るに、ステップ S 2 3 は、アクセスコスト、即ち、割り込み回数を、データ定義位置に対して例えば図 8 に示す如きグラフ形式で表示する。データ定義位置は、ソース名 S、関数名 F、変数 V、アドレス A 等で特定可能である。この場合、XXXXXX で示すアドレス近傍でアクセスコストが大きいことがわかる。

【 0 0 5 4 】

このようにして、割り込みハンドラ 1 4 0 により収集されるサンプリング値及び任意のハードウェアイベント情報を、命令毎に対応付けた領域に実行回数として設定することができる。又、割り込みハンドラ 1 4 0 により、収集されたサンプリング値を実行形式プログラムと照合することにより、実行頻度の高い関数、処理及び命令を対比させ、又、対応する C ソースや機械語命令と照合して表示することもできる。更に、割り込みハンドラ 1

10

20

30

40

50

40により収集されるサンプリング値及び任意のハードウェアイベント情報をマージすることで、これらの情報を取得する毎にマージされた値を経験値としてメモリに保持することで、プロファイリングの精度を向上させることもできる。

【0055】

図10は、コンピュータハードウェアと解析プロセッサを示すブロック図である。

【0056】

図10は、測定用プロセッサ201は、図1に示すコンピュータハードウェア10に相当する。測定用プロセッサ201は、各種命令を処理するプロセッサコア211、データを保持するキャッシュメモリ212、測定用プロセッサ201内部のイベント及び/又は測定用プロセッサ201外部とのやり取りのイベントをカウントすることにより性能をモニタリングするハードウェアカウンタ213、一定間隔で割り込み信号を出力する内蔵タイマ214を有する。キャッシュメモリ212は、測定用プロセッサ201に対して外部接続されていても良い。

10

【0057】

記憶部202は、解析対象プログラム221、プロファイリングプログラム222、及び検出命令アドレス又はハードウェアカウンタ213のカウント値に基づく割り込み回数を格納するテーブル223を保持する。解析対象プログラム221はプロファイリングされるソフトウェアであり、図1に示すアプリケーションプログラム30に相当する。プロファイリングプログラム222は、解析対象プログラム221のプロファイル情報を取得するソフトウェアであり、図1に示すプロファイリングプログラム100に相当する。測定用プロセッサ201、記憶部202及び性能解析プロセッサ203はバス300で夫々接続されている。

20

【0058】

タイマ214は、解析対象プログラム221の実行が開始されてから所定時間間隔で割り込み信号を出力する。この割り込み信号の所定時間間隔は、プロファイリングプログラム222で設定可能である。割り込み発生を検出した命令アドレス又はハードウェアカウンタ213のイベント発生をトリガとして、ハードウェアカウンタ213の割り込み発生を検出した命令アドレスがテーブル223に格納される。ハードウェアカウンタ213は、タイマ214と同様に解析対象プログラム221の実行が開始されると同時に計数を開始する。テーブル223は、解析対象プログラム221の性能解析中に作成される。解析プロセッサ203は、測定用プロセッサ201における実行結果を表示する、測定用プロセッサ201と別体のプロセッサである。解析プロセッサ203は、テーブル223の保持された情報を読み出して、ソース名、関数、変数名、変数内相対アドレスに対応させて図8及び図9の如き表示を行う。

30

【0059】

更に、情報収集の対象となるアプリケーションプログラム30（解析対象プログラム221）を自動的に絞るようにしても良い。この場合、情報収集の対象となるアプリケーションプログラム30を、シミュレータで取得したプロファイル情報を基に、実行コストの大きいプログラムからプロファイル対象候補として抽出すれば良い。又、ソースコードを関数の複雑度や、関数間の呼び出し関係等の静的解析情報の大小によりプロファイル対象を抽出しても良い。

40

【0060】

図11は、プロファイリングのターゲット範囲を自動抽出する処理を説明するフローチャートである。図11において、ステップS31では、ユーザがプロファイリングの対象を判断できているか否かを判定する。ステップS31の判定結果がYESであると、ステップS32では、ユーザがプロファイリングの対象となるアプリケーションプログラム30をプロファイリングプログラム100にリンクする時に指定し、処理は終了する。

【0061】

他方、ステップS31の判定結果がNOであると、ステップS33では、コンピュータハードウェア10（即ち、プロセッサ）がプロファイリングの対象となるアプリケーショ

50

ンプログラム30を自動的に抽出する処理を開始する。ステップS34では、コンピュータハードウェア10のメモリ内の使用可能なメモリサイズUMSを算出する。ステップS35では、コンピュータハードウェア10が使用可能なメモリサイズUMSに納まる範囲内でアプリケーションプログラム30のターゲット範囲を抽出し、プロファイリングの対象となるアプリケーションプログラム30をプロファイリングプログラム100にリンクする時に自動的にこのターゲット範囲を指定する。

【0062】

ステップS35は、プロファイリングの対象となるアプリケーションプログラム30をシミュレータで取得したプロファイル情報を元に、実行コストの大きいプログラムからプロファイリングの対象候補として抽出する第1の方法を用いても、或いは、ソースコードを関数の複雑度や関数間の呼び出し関数等の静的解析情報の大小によりプロファイリングの対象候補を抽出する第2の方法を用いても良い。

10

【0063】

尚、本発明は、以下に付記する発明をも包含するものである。

(付記1)

所定の条件が満たされた際に発生する割り込みにより起動される割り込みハンドラにより情報収集の対象となる対象プログラムの実行状況に関する情報を収集するコンピュータによるプロファイリング方法であって、

該対象プログラムに対して該割り込みハンドラにより情報収集を行うターゲット範囲を指定するターゲット範囲指定ステップと、

20

該ターゲット範囲で該割り込みが発生した際に該割り込みハンドラにより収集される情報をメモリに設定する情報設定ステップとを含むことを特徴とするプロファイリング方法。

(付記2)

該ターゲット範囲指定ステップは、該対象プログラムに、該割り込みハンドラを含むプロファイリングプログラムと、コード範囲マーク用シンボルソース又はオブジェクトをリンクさせる際に、該ターゲット範囲を指定することを特徴とする付記1記載のプロファイリング方法。

(付記3)

該割り込みは、該コンピュータ内部のイベント及び/又は該コンピュータ外部とのやり取りのイベントをカウントするハードウェアカウンタのカウント値に基づき発生することを特徴とする付記1又は2記載のプロファイリング方法。

30

(付記4)

該割り込みハンドラにより収集された情報に含まれるサンプリング値及びハードウェアイベント情報を、命令毎に対応付けた該メモリの領域に実行回数又はイベント発生回数として設定するステップを含むことを特徴とする付記1乃至3のいずれか1項記載のプロファイリング方法。

(付記5)

該割り込みハンドラにより収集された情報に含まれるサンプリング値を実行形式プログラムと照合し、実行頻度の高い関数、処理及び命令を対比し、対応するCソース又は機械語命令と照合して、該コンピュータに接続される解析プロセッサの表示部に表示するステップを含むことを特徴とする付記1乃至3のいずれか1項記載のプロファイリング方法。

40

(付記6)

該割り込みハンドラにより収集される情報に含まれるサンプリング値及びハードウェアイベント情報をマージして経験値として該メモリに保持するステップを含むことを特徴とする付記1乃至5のいずれか1項記載のプロファイリング方法。

(付記7)

コンピュータに、所定の条件が満たされた際に発生する割り込みにより起動される割り込みハンドラにより情報収集の対象となる対象プログラムの実行状況に関する情報を収集するプロファイリングを行わせるプログラムであって、

50

該対象プログラムに対して該割り込みハンドラにより情報収集を行うターゲット範囲を指定させるターゲット範囲指定手順と、

該ターゲット範囲で該割り込みが発生した際に該割り込みハンドラにより収集される情報をメモリに設定させる情報設定手順とを該コンピュータに実行させることを特徴とするプログラム。

(付記 8)

該ターゲット範囲指定手順は、該対象プログラムに、該割り込みハンドラを含むプロファイリングプログラムと、コード範囲マーク用シンボルソース又はオブジェクトをリンクさせる際に、該ターゲット範囲を指定させることを特徴とする付記 7 記載のプログラム。

(付記 9)

該割り込みは、該コンピュータ内部のイベント及び/又は該コンピュータ外部とのやり取りのイベントをカウントするハードウェアカウンタのカウント値に基づき発生することを特徴とする付記 7 又は 8 記載のプログラム。

(付記 10)

該割り込みハンドラにより収集された情報に含まれるサンプリング値及びハードウェアイベント情報を、命令毎に対応付けた該メモリの領域に実行回数として設定させる手順を該コンピュータに実行させることを特徴とする付記 7 乃至 9 のいずれか 1 項記載のプログラム。

(付記 11)

該割り込みハンドラにより収集された情報に含まれるサンプリング値を実行形式プログラムと照合させ、実行頻度の高い関数、処理及び命令を対比させ、対応する C ソース又は機械語命令と照合させ、該コンピュータに接続される解析プロセッサの表示部に表示させる手順を該コンピュータに実行させることを特徴とする付記 7 乃至 9 のいずれか 1 項記載のプログラム。

(付記 12)

該割り込みハンドラにより収集される情報に含まれるサンプリング値及びハードウェアイベント情報をマージして経験値として該メモリに保持させる手順を該コンピュータに実行させることを特徴とする付記 7 乃至 11 のいずれか 1 項記載のプログラム。

【0064】

以上、本発明を実施例により説明したが、本発明は上記実施例に限定されるものではなく、本発明の範囲内で種々の変形及び改良が可能であることは言うまでもない。

【図面の簡単な説明】

【0065】

【図 1】本発明が適用可能な電子装置の構成を示す図である。

【図 2】イベント管理テーブルの一例を示す図である。

【図 3】割り込みハンドラ記録テーブルの一例を示す図である。

【図 4】実施例の処理手順を説明するフローチャートである。

【図 5】リンク時のターゲット範囲の指定を説明する図である。

【図 6】プロファイリングのターゲット範囲の指定方法とサンプリング値のメモリ内の格納領域を説明する図である。

【図 7】タイマ割り込み発生時の処理を説明するフローチャートである。

【図 8】解析プロセッサの処理を説明するフローチャートである。

【図 9】プロファイリングの結果の表示例を示す図である。

【図 10】コンピュータハードウェアと解析プロセッサを示すブロック図である。

【図 11】プロファイリングのターゲット範囲を自動抽出する処理を説明するフローチャートである。

【符号の説明】

【0066】

1 電子装置

10 コンピュータハードウェア

10

20

30

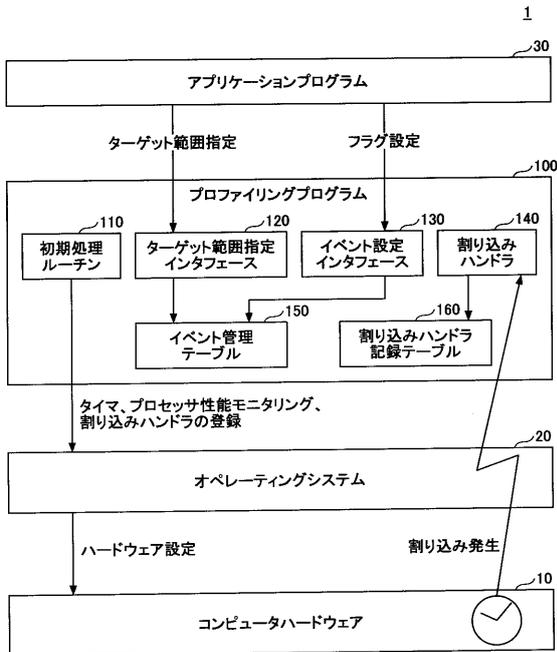
40

50

- 2 0 オペレーティングシステム
- 3 0 アプリケーションプログラム
- 1 0 0 プロファイリングプログラム
- 1 4 0 割り込みハンドラ
- 2 0 1 測定用プロセッサ
- 2 0 2 メモリ
- 2 0 3 解析プロセッサ
- 2 1 1 プロセッサコア
- 2 1 2 キャッシュメモリ
- 2 1 3 ハードウェアカウンタ
- 2 1 4 内蔵タイマ
- 2 2 1 解析対象プログラム
- 2 2 2 プロファイリングプログラム
- 2 2 3 テーブル

【図1】

本発明が適用可能な電子装置の構成を示す図



【図2】

イベント管理テーブルの一例を示す図

開始アドレス	終了アドレス	イベントの属性情報
1000	2000	10 μm(割り込み間隔)

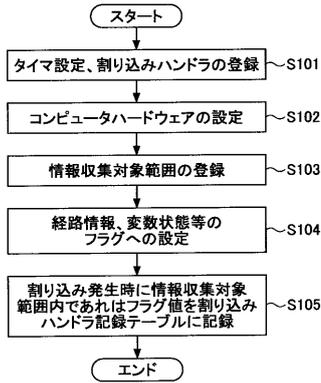
【図3】

割り込みハンドラ記録テーブルの一例を示す図

イベントの情報 (例えば割り込み回数)
0
1
0
2
⋮

【図4】

実施例の処理手順を説明するフローチャート



【図5】

リンク時のターゲット範囲の指定を説明する図

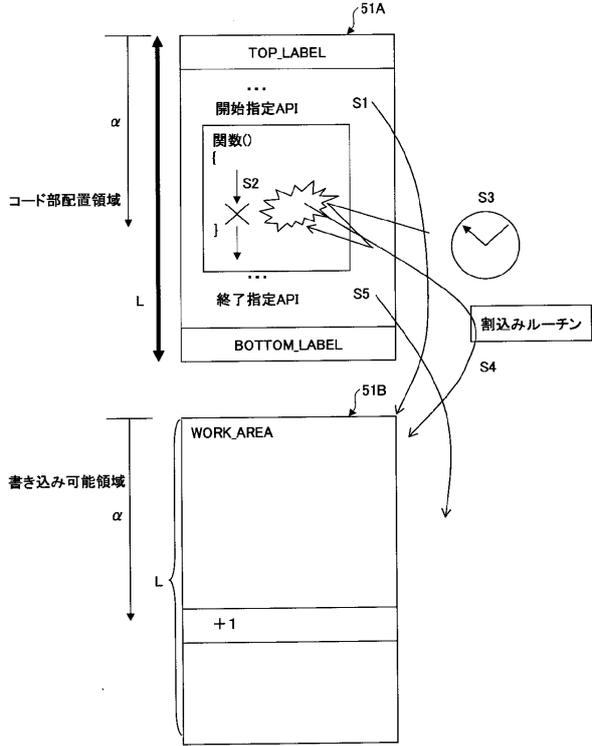
S	CODE_PRG1	xxxxx
	CODE_PRG2	yyyyy
E	CODE_PRG3	zzzzz

セクション名 セクションサイズ

...

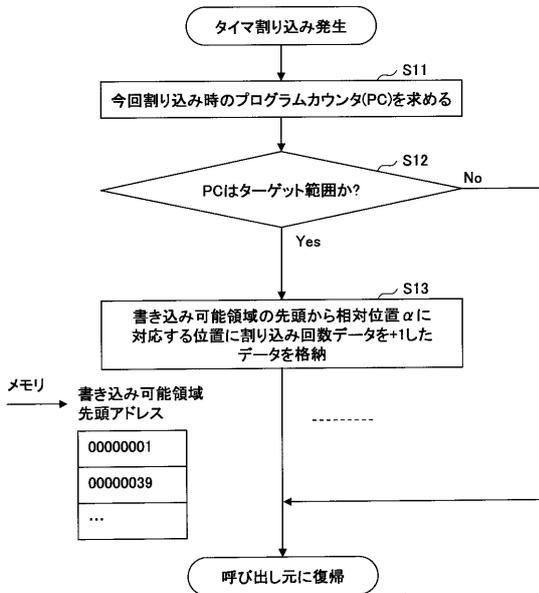
【図6】

プロファイリングのターゲット範囲の指定方法とサンプリング値のメモリ内の格納領域を説明する図



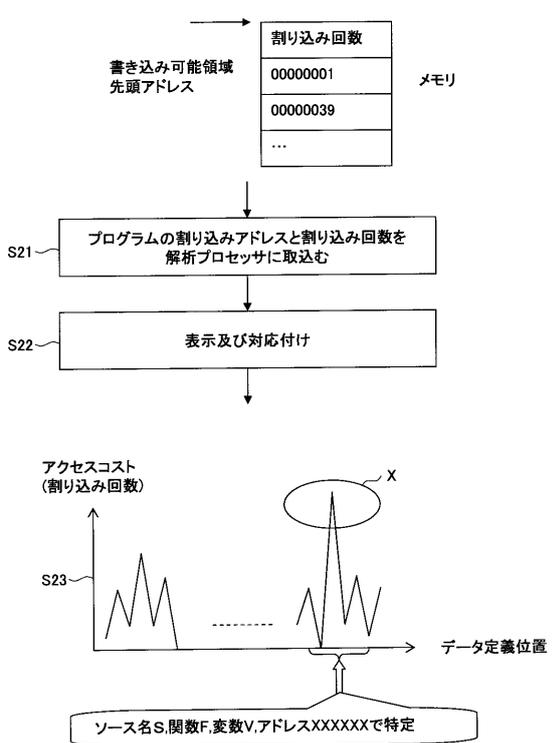
【図7】

タイマ割り込み発生時の処理を説明するフローチャート



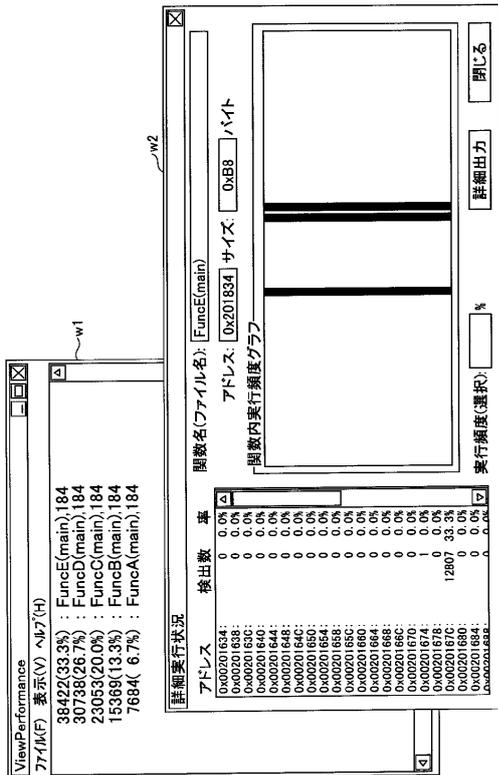
【図8】

解析プロセッサの処理を説明するフローチャート



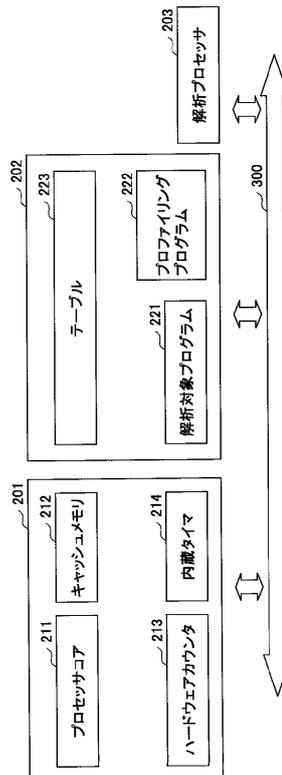
【図 9】

プロファイリングの結果の表示例を示す図



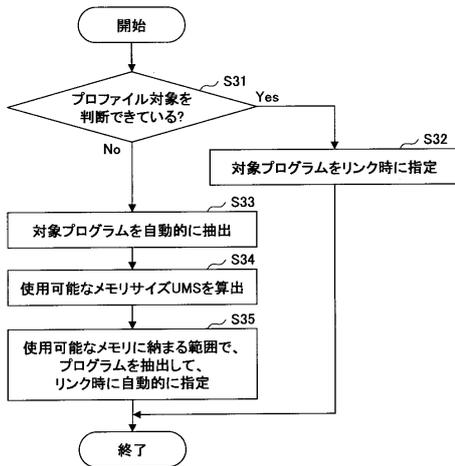
【図 10】

コンピュータハードウェアと解析プロセッサを示すブロック図



【図 11】

プロファイリングのターゲット範囲を自動抽出する処理を説明するフローチャート



フロントページの続き

- (56)参考文献 特開2005-215816(JP,A)
特開平04-320548(JP,A)
特開昭59-003651(JP,A)
特開平09-006644(JP,A)
特開平07-219755(JP,A)
特開平11-212837(JP,A)
特開2003-140928(JP,A)
特開平07-191882(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 11/28 - 11/34