(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6064756号
(P6064756)
(45)発行日　平成29年1月25日(2017.1.25)　(24)登録日　平成29年1月6日(2017.1.6)

(51)Int.Cl.　F I
G06F 11/34 (2006.01)　G06F 11/34 109
G06F 11/34 133

請求項の数 6　(全 18 頁)

(21)出願番号　特願2013-82397(P2013-82397)
(22)出願日　平成25年4月10日(2013.4.10)
(65)公開番号　特開2014-206786(P2014-206786A)
(43)公開日　平成26年10月30日(2014.10.30)
審査請求日　平成28年1月13日(2016.1.13)

(73)特許権者　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100079049
弁理士　中島　淳
(74)代理人　100084995
弁理士　加藤　和詳
(74)代理人　100099025
弁理士　福田　浩志
(72)発明者　松尾　美由紀
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

最終頁に続く

(54)【発明の名称】性能データ収集プログラム、装置、及び方法

(57)【特許請求の範囲】
【請求項１】
演算処理装置と転送制御部とを有する情報処理装置に、
所定の関数で記述された解析対象のプログラムを実行する前記演算処理装置が呼び出した関数を示す情報を含む性能データを、所定のサンプリング間隔で採取して、主記憶部に格納させ、
採取した性能データに基づいて、前記演算処理装置の処理負荷を判定させ、
前記演算処理装置の処理負荷が所定値以下の場合に、前記主記憶部に格納した性能データの少なくとも一部を、前記転送制御部により前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送させる
ことを特徴とする性能データ収集プログラム。
【請求項２】
前記演算処理装置の処理負荷の判定は、
前記情報処理装置に、
採取した性能データに基づいて、前記演算処理装置がアイドル状態かを判定させることを特徴とする請求項１記載の性能データ収集プログラム。
【請求項３】
前記演算処理装置の処理負荷の判定は、
前記情報処理装置に、
所定期間に採取した複数の性能データの各々に含まれる情報が示す関数におけるアイド

ル関数の割合が所定割合以上の場合に、前記演算処理装置がアイドル状態であると判定させることを特徴とする請求項２記載の性能データ収集プログラム。
【請求項４】
前記主記憶部以外の記憶部への転送は、
前記情報処理装置に、
前記主記憶部に格納した性能データが所定量を超えた場合に、前記主記憶部に格納した性能データの少なくとも一部を、前記転送制御部により前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送させることを特徴とする請求項１～請求項３のいずれか１項記載の性能データ収集プログラム。
【請求項５】
所定の関数で記述された解析対象のプログラムを実行する演算処理装置と、
前記演算処理装置が呼び出した関数を示す情報を含む性能データを、所定のサンプリング間隔で採取して、主記憶部に格納する採取格納部と、
採取された性能データに基づいて、前記演算処理装置の処理負荷を判定する判定部と、
前記演算処理装置の処理負荷が所定値以下の場合に、前記主記憶部に格納された性能データの少なくとも一部を、前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送する転送制御部と、
を含む性能データ収集装置。
【請求項６】
演算処理装置と転送制御部とを有する情報処理装置が、
所定の関数で記述された解析対象のプログラムを実行する前記演算処理装置が呼び出した関数を示す情報を含む性能データを、所定のサンプリング間隔で採取して、主記憶部に格納し、
採取した性能データに基づいて、前記演算処理装置の処理負荷を判定し、
前記演算処理装置の処理負荷が所定値以下の場合に、前記主記憶部に格納した性能データの少なくとも一部を、前記転送制御部により前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送する
ことを特徴とする性能データ収集方法。
【発明の詳細な説明】
【技術分野】
【０００１】
開示の技術は、性能データ収集プログラム、装置、及び方法に関する。
【背景技術】
【０００２】
コンピュータで動作するプログラムにレスポンス低下などの性能劣化の問題が発生した際の原因解明や、プログラムの性能向上のためのチューニング箇所を特定するために、プログラムの性能を解析することが行われている。プログラムの性能解析は、プログラム実行時に収集した性能データなどの各種情報に基づいて行われる。性能データの収集の一手法として、解析対象のプログラムの実行中に、動作したプロセスや呼び出された関数等を示す性能データをサンプリングにより採取し、複数の性能データを収集する手法が存在する。
【０００３】
例えば、所定のサンプリング間隔でＣＰＵ（Central Processing Unit）が実行しているアドレスを取得し、取得したアドレスに取得時刻を関連づけ、アドレスデータとしてメインメモリ上のアドレスデータ記憶部に時系列に格納する技術が提案されている。この技術では、サンプリング期間終了後に、アドレスデータ記憶部に格納されたアドレスデータを、ＨＤＤ（Hard Disk Drive）上のファイルに書き出し、一括して解析処理を行っている。
【０００４】
また、サンプリングにより採取したデータを、カーネル空間の一時バッファ領域を介し

てユーザ空間にコピーし、コピーしたデータに対して、デーモン（バックグラウンド処理）で解析処理を行い、解析結果をＨＤＤ上のファイルに書き出す技術が存在する。

【先行技術文献】

【特許文献】

【０００５】

【特許文献１】特開２００７－２１３２０５号公報

【非特許文献】

【０００６】

【非特許文献１】“openSUSE 12.3　システム分析とチューニングガイド”、［online］、［平成２５年３月８日検索］、インターネット＜ＵＲＬ：http://opensuse-man-ja.berlios.de/opensuse-html/cha.tuning.oprofile.html＞

【発明の概要】

【発明が解決しようとする課題】

【０００７】

１台のコンピュータに複数のプロセッサを搭載しているコンピュータで実行されるプログラムを解析対象のプログラムとする場合には、コンピュータに搭載されているプロセッサ毎に性能データを収集する必要がある。また、近年、プロセッサのマルチコア化に伴い、１つのＬＳＩ（Large Scale Integrated circuit）パッケージ内に組み込まれるコア数が増加している。また、１つのコアが複数のスレッドをそれぞれ実行するマルチスレッド化も進んでいる。このようにマルチコア化されたプロセッサで実行されるプログラムや、マルチスレッドを利用したプログラムを解析対象とする場合には、コア毎またはスレッド毎に性能データを収集する必要がある。

【０００８】

また、ＨＰＣ（High Performance Computing）などの科学技術計算向けアプリケーションプログラムでは、プログラムの実行時間が長いため、１回に収集すべき性能データのデータ量が増大する。さらに、金融などのミッションクリティカルなシステムで使用されるアプリケーションプログラムでは、高速なレスポンスが要求される。例えば、株式売買システムにおける注文処理で、１／１０００秒（１ｍｓ）以下のレスポンスタイムを実現するシステムが存在する。なお、注文処理とは、例えば、ユーザ端末からの注文を受け付け、注文を受け付けたことを確認し、注文内容をチェックしてからサーバへ登録し、登録した旨をユーザ端末へ通知する一連の処理である。このような注文処理のシステムを実現するアプリケーションプログラムの性能解析を行う場合には、システムの挙動を捉えるために、より細かいサンプリング間隔で性能データを採取して、サンプリング期間分の性能データを収集する必要がある。

【０００９】

収集する性能データのデータ量は、例えば下記のように計算することができる。

・収集する性能データのデータ量＝１性能データのサイズ×１システム当たりのプロセッサ数×１プロセッサ当たりのコア数×１コア当たりのスレッド数×１スレッド当たりの収集する性能データの個数

・なお、収集する性能データの個数＝サンプリング期間÷サンプリング間隔

従って、マルチプロセッサ化、マルチコア化、マルチスレッド化、またはシステムの高速化により、収集する性能データのデータ量は増大する。このように、プログラムの性能解析では、大量の性能データを収集できることが要求される。

【００１０】

しかし、サンプリング期間終了後に一括して解析処理を行う従来技術では、収集可能なデータ量はメモリサイズに依存するが、メモリサイズを大きくするには限界がある、という問題がある。

【００１１】

また、性能データのサンプリング処理を行いながらデーモンで解析処理を行う従来技術では、カーネル空間の一時バッファ領域に保持した解析結果をＨＤＤ上のファイルに書き

出した後は、カーネル空間の一時バッファ領域を再利用することができる。しかし、同一のコンピュータが性能データのサンプリングを行いながら転送指示を行うことにより、コンピュータの処理負荷が高い場合に転送指示を行っているため、解析対象のプログラムの動作に影響を与えてしまい、収集した性能データの解析時において性能データに影響を与えた要因の切り分けが困難であった、という問題がある。

【００１２】

開示の技術は、一つの側面として、収集した性能データの解析時において、性能データに影響を与えた要因の切り分けを容易にすることが目的である。

【課題を解決するための手段】

【００１３】

開示の技術は、演算処理装置と転送制御部とを有する情報処理装置に、所定の関数で記述された解析対象のプログラムを実行する演算処理装置が呼び出した関数を示す情報を含む性能データを、所定のサンプリング間隔で採取して、主記憶部に格納させる。また、開示の技術は、前記情報処理装置に、採取した性能データに基づいて、前記演算処理装置の処理負荷を判定させる。また、開示の技術は、前記演算処理装置の処理負荷が所定値以下の場合に、前記主記憶部に格納した性能データの少なくとも一部を、前記転送制御部により前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送させる。

【発明の効果】

【００１４】

開示の技術は、一つの側面として、演算処理装置の処理負荷が所定値以下の場合に、転送指示を行うため、収集した性能データの解析時において性能データに影響を与えた要因の切り分けが容易になる、という効果を有する。

【図面の簡単な説明】

【００１５】

【図１】本実施形態に係る性能データ収集装置の構成の一例を示すブロック図である。

【図２】性能データの一例を示す図である。

【図３】シンボルテーブル（Ｓｙｓｔｅｍ．ｍａｐファイル）の一部の一例を示す図である。

【図４】性能データ格納部のアドレス管理を説明するための図である。

【図５】性能データ格納部のアドレス管理を説明するための図である。

【図６】性能データ格納部のアドレス管理を説明するための図である。

【図７】性能データ格納部のアドレス管理を説明するための図である。

【図８】性能データ収集装置として機能するコンピュータの一例を示す概略ブロック図である。

【図９】解析対象プログラムの実行を示すフローチャートである。

【図１０】性能データ収集処理を示すフローチャートである。

【図１１】アイドル状態判定処理を示すフローチャートである。

【図１２】転送制御処理を示すフローチャートである。

【図１３】性能データ収集装置の他の構成例を示すブロック図である。

【発明を実施するための形態】

【００１６】

以下、図面を参照して開示の技術の実施形態の一例を詳細に説明する。

【００１７】

図１に、本実施形態の概略を示す。図１に示すように、本実施形態に係る性能データ収集装置１０は、ＣＰＵ（Central Processing Unit）１２、メモリ１４、及びＨＣＡ（Host Channel Adapter）１６を含んでいる。性能データ収集装置１０は、コンピュータ等の情報処理装置である。ＣＰＵ１２は、解析対象のプログラム及び後述する性能データ収集プログラムを実行する演算処理装置である。メモリ１４は、例えばＲＡＭ（random access memory）などの主記憶部である。ＨＣＡ１６は、例えばＩｎｆｉｎｉＢａｎｄ（登録商標）等のインタコネクトを利用したＲＤＭＡ（Remote Direct Memory Access）を行うた

めの通信部としてのインターフェースカードである。
【００１８】
また、性能データ収集装置１０は、例えばＩｎｆｉｎｉＢａｎｄ（登録商標）等のインタコネクトを利用した通信を行うことができる通信ケーブル４８を介して、性能データ収集装置１０とは異なるコンピュータであるリモート装置４０と接続されている。リモート装置４０は、性能データ収集装置１０と同様の構成とすることができ、少なくとも主記憶部であるメモリ４４及び通信部であるＨＣＡ４６を備えている。性能データ収集装置１０は、メモリ１４に格納されたデータを、ＣＰＵ１２を介することなく、ＨＣＡ１６、通信ケーブル４８、及びリモート装置４０のＨＣＡ４６を介して、リモート装置４０のメモリ４４へ直接転送するＲＤＭＡ転送を行う。
【００１９】
また、図１には、性能データ収集装置１０の各機能を示す機能ブロックも合わせて示している。性能データ収集装置１０は、ＣＰＵ１２の各機能として、採取格納部３２、判定部３４、及び転送制御部３６を備えている。また、メモリ１４上に性能データを格納する領域である性能データ格納部３８を備えている。
【００２０】
採取格納部３２は、複数の関数で記述された解析対象のプログラムを実行中のＣＰＵ１２が呼び出した関数を示す情報を含む性能データを、設定されたサンプリング期間（サンプリングの開始から終了までの期間）において、設定されたサンプリング間隔で採取する。採取格納部３２は、例えば図２に示すような性能データを採取する。図２の例では、各性能データは、ＣＰＵＩＤ、ＰＩＤ、及び実行アドレスを含んでいる。ＣＰＵＩＤは、解析対象のプログラムを実行中のＣＰＵを識別するための識別番号である。ＰＩＤは、ＣＰＵ１２が実行中のプロセス（関数を含むプログラムの実行単位）を識別するための識別番号である。実行アドレスは、ＣＰＵ１２が実行中の関数が格納されたメモリ１４上のアドレスである。また、性能データに、採取時刻を示すタイムスタンプを付加してもよい。
【００２１】
例えばＬｉｎｕｘ（登録商標）の場合には、解析対象のプログラムに関するプロセスを表現するための情報を含む構造体がカーネル空間に生成される。従って、採取格納部３２は、カーネル空間に生成された構造体から性能データとして必要な情報を採取することができる。構造体に含まれる情報は、ＣＰＵ１２内のレジスタに格納された値などに基づいて、カーネルにより設定される。より具体的には、解析対象のプログラムを実行中のＣＰＵのＣＰＵＩＤを返す関数（例えば、smp_processor_id）を呼び出して実行することにより、ＣＰＵＩＤを採取することができる。また、ＣＰＵで実行中のプロセスを示すシンボル（current）を用いて、ＰＩＤを返す関数（例えば、current->pid）を呼び出して実行することにより、実行中のプロセスのＰＩＤを採取することができる。また、実行中のプロセスの状態に関する情報が格納された構造体pt_regsを参照して、実行アドレスを返す関数（例えば、pt_regs->ip）を呼び出して実行することにより、実行アドレスを採取することができる。
【００２２】
なお、マルチプロセッサを利用して解析対象のプログラムが実行される場合には、上記のＣＰＵＩＤによりそれぞれのＣＰＵを識別可能であるため、ＣＰＵ毎の性能データを採取することができる。前述のＣＰＵＩＤを返す関数で取得するＣＰＵＩＤは論理的なＣＰＵを識別する識別番号であり、マルチコア及びマルチスレッドの場合にも異なるＣＰＵＩＤとなる。このようにして、コア毎及びスレッド毎の性能データを採取することができる。
【００２３】
また、採取格納部３２は、採取した性能データを格納するための領域である性能データ格納部３８をメモリ１４上に確保し、性能データ格納部３８の先頭アドレスをａｄｄｒ＿ＲＳ、末尾アドレスをａｄｄｒ＿ＲＥで定められるアドレス範囲の記憶領域に記憶する。採取格納部３２は、採取した性能データを、性能データ格納部３８の空き領域の先頭から

順に格納していく。

【００２４】

判定部３４は、性能データ格納部３８に格納された性能データに基づいて、ＣＰＵ１２の処理負荷を判定する。例えば、判定部３４は、ＣＰＵ１２がアイドル状態の場合には、処理負荷が所定値以下であり、ＣＰＵが性能に影響を与えるような処理を実行していないと判定することができる。ＣＰＵ１２がアイドル状態か否かの判定は、例えば、所定期間に採取された性能データに含まれる実行アドレスが示す関数におけるアイドル関数の割合に基づいて判定することができる。

【００２５】

より具体的には、所定期間に採取された性能データに含まれる実行アドレスが示す関数におけるアイドル関数の割合が所定値以上の場合に、ＣＰＵ１２がアイドル状態であると判定することができる。例えば、サンプリング期間または解析対象のプログラムの動作時間に基づいて、アイドル状態を判定するための所定期間を定めておく。例えば、下記のように所定期間を定めることができる。

例１：サンプリング期間が、例えば１００秒の場合に、サンプリング期間の１％である１秒

例２：サンプリング期間内に動作時間が例えば１ｍｓの間、解析対象のプログラムが複数回実行される場合に、例えば１０回分のプログラムの動作時間１０ｍｓ

上記のように定めた所定期間をサンプリング間隔で割り、所定期間内に採取される性能データの数、すなわち所定期間内の関数の数を計算しておく。例えばサンプリング間隔を１００μｓとすると、所定期間内の関数の数は、上記例１の場合は１００００個、上記例２の場合は１００個である。

【００２６】

この所定期間内の関数の数に対するアイドル関数の数の割合が所定値以上か否かを判定する。例えば所定値を９５％とすると、上記例１の場合は９５００個以上、上記例２の場合は９５個以上がアイドル関数であれば、ＣＰＵ１２がアイドル状態であると判定する。また、所定期間に採取された性能データに含まれる実行アドレスが示す関数のうち、アイドル関数以外の関数の割合が所定値以下の場合に、ＣＰＵ１２がアイドル状態であると判定してもよい。例えば所定期間のうち５％は他の関数が含まれていてもよいとすると、所定期間内で他の関数が許容される許容数は、上記例１の場合は５００個以下、上記例２の場合は５個以下である。

【００２７】

また、所定期間内の関数に対するアイドル関数の数の割合が、例えば１００％か否かを判定するようにしてもよい。アドレス関数の数の割合が１００％の場合は、実行アドレスがアイドル関数であることを示す性能データが、所定期間連続して採取されたことを示すこととなる。

【００２８】

なお、実行アドレスが示す関数がアイドル関数か否は、関数名を含むシンボルの名前とメモリ１４上のアドレスとの対応関係を示すシンボルテーブルに基づいて判定することができる。例えばＬｉｎｕｘ（登録商標）の場合、アイドル関数はカーネル関数であり、カーネルが使用するシンボルテーブルＳｙｓｔｅｍ．ｍａｐファイルを参照することにより、カーネル関数の名前とアドレスとの対応関係が得られる。図３に、Ｓｙｓｔｅｍ．ｍａｐファイルの一部の一例を示す。図３の例では、先頭の項目は各関数のアドレス範囲の先頭アドレス、次の項目はシンボルの型、及び最後の項目は関数名である。ＣＰＵ１２がアイドル状態か否かを判定するためのアイドル関数は、例えば「ｐｏｌｌ＿ｉｄｌｅ」など、予め関数名で識別することができる。そこで、Ｓｙｓｔｅｍ．ｍａｐを参照して、対象のアイドル関数のアドレス範囲を取得することができる。上記のｐｏｌｌ＿ｉｄｌｅ関数を対象のアイドル関数とする場合には、ffffffff810148d0≦アイドル関数のアドレス範囲＜ffffffff810149480となる。従って、実行アドレスが対象のアイドル関数のアドレス範囲に含まれる場合には、実行アドレスが示す関数はアイドル関数であると判定することが

できる。

【００２９】

転送制御部３６は、性能データ格納部３８の空き領域のアドレス及び格納された性能データの数を管理する。また、転送制御部３６は、判定部３４によりＣＰＵ１２がアイドル状態であると判定され、かつ性能データ格納部３８に格納された性能データが所定量を超えた場合に、性能データの転送制御を行う。以下、詳細に説明する。

【００３０】

例えば図４に示すように、転送制御部３６は、性能データ格納部３８内の空き領域の先頭アドレスａｄｄｒ＿１、末尾アドレスａｄｄｒ＿２のアドレス範囲の記憶領域に性能データを記憶する。図４に示すように、性能データ格納部３８に性能データが格納されていない状態では、性能データ格納部３８の先頭アドレスａｄｄｒ＿ＲＳ＝ａｄｄｒ＿１、性能データ格納部３８の末尾アドレスａｄｄｒ＿ＲＥ＝ａｄｄｒ＿２である。また、転送制御部３６は、空き領域の先頭アドレスａｄｄｒ＿１と末尾アドレスａｄｄｒ＿２とを比較して、空き領域のサイズを判定する。ａｄｄｒ＿１＞ａｄｄｒ＿２の場合には、アドレスａｄｄｒ＿１からアドレスａｄｄｒ＿ＲＥまでの領域と、アドレスａｄｄｒ＿ＲＳからアドレスａｄｄｒ＿２までの領域が空き領域となる。すなわち、性能データ格納部３８を、先頭アドレスａｄｄｒ＿ＲＳと末尾アドレスａｄｄｒ＿ＲＥとを繋いだ循環したメモリ領域とみなす。

【００３１】

採取格納部３２により採取された性能データは、性能データ格納部３８の空き領域の先頭から順次格納される。転送制御部３６は、性能データ格納部３８に格納された性能データの個数ｋをカウントする。また、図５に示すように、性能データ格納部３８にｋ番目に格納された性能データの末尾アドレスをａｄｄｒ＿ｋＥとする。転送制御部３６は、性能データ格納部３８にｋ番目の性能データが格納された際に、空き領域の先頭アドレスａｄｄｒ＿１を、アドレスａｄｄｒ＿ｋＥに更新する。

【００３２】

また、転送制御部３６は、性能データ格納部３８に格納された性能データが所定量を超えたか否かを判定する。この判定は、性能データ格納部３８に格納された性能データの個数ｋが閾値Ｋ（１≦Ｋ≦ｋ）を超えたか否かにより判定することができる。例えば、Ｋは一度に転送する性能データの個数とすることができる。転送制御部３６は、性能データ格納部３８に格納された性能データが所定量を超えたと判定すると、例えば図６に示すように、空き領域の末尾アドレスａｄｄｒ＿２を、転送する性能データの先頭アドレスａｄｄｒ＿ＫＳに指定する。また、Ｋ番目に格納された性能データの末尾アドレスを、転送する性能データの末尾アドレスａｄｄｒ＿ＫＥに指定する。

【００３３】

ここで、一度に転送する性能データの個数は、性能データ格納部３８に格納された性能データの全てであってもよいし、一部であってもよい。一度に全ての性能データを転送する場合には、上記のＫ番目に格納された性能データの末尾アドレスａｄｄｒ＿ＫＥはａｄｄｒ＿１となる。また、一度に転送する性能データのデータ量を、判定部３４でＣＰＵ１２のアイドル状態を判定する際のアイドル関数の連続時間やデータ転送速度などを考慮して決定してもよい。例えばデータ転送速度が２ＧＢｐｓ（１秒間に２ＧＢ）で、アイドル関数の連続時間を１秒としている場合には、一度に転送する性能データのデータ量を２ＧＢとすることができる。さらに、性能データ格納部３８の性能データの格納以外の利用を考慮して、その半分の１ＧＢを一度に転送するデータ量としてもよい。このように定めたデータ量を１性能データ当たりのデータ量で割って、一度に転送する性能データの個数Ｋを定めておく。

【００３４】

転送制御部３６は、転送する性能データの先頭アドレスａｄｄｒ＿ＫＳ及び末尾アドレスａｄｄｒ＿ＫＥと、転送先のリモート装置４０のメモリ４４を指定する情報を含む転送情報を通信部であるＨＣＡ１６に設定することにより、ＨＣＡ１６に性能データの転送を

指示する。性能データの転送を指示されたＨＣＡ１６は、設定された転送情報に基づいて、ＣＰＵ１２を介することなく、性能データ格納部３８内の指定されたアドレスに格納された性能データを、指定された転送先のリモート装置４０のメモリ４４へＲＤＭＡ転送する。

【００３５】

なお、上記では、性能データ格納部３８に格納された性能データが所定量を超えたか否かを、性能データ格納部３８に格納された性能データの数ｋが、一度に転送する性能データの数Ｋを超えたか否かにより判定する場合について説明したが、これに限定されない。一度に転送する性能データの個数Ｋと、性能データ格納部３８に格納された性能データが所定量を超えたか否かを判定するための閾値とは、それぞれ異なる値を設定してもよい。

【００３６】

また、図７に示すように、転送制御部３６は、空き領域の末尾アドレスａｄｄｒ＿２を、転送する性能データの末尾アドレスａｄｄｒ＿ＫＥに更新する。これにより、格納されていた性能データが転送された領域を、再び空き領域として利用して、新たな性能データを格納することができる。

【００３７】

性能データ収集装置１０は、例えば図８に示すように、ＣＰＵ１２、メモリ１４、及びＨＣＡ１６に加え、不揮発性の記憶部１８、及び入出力インターフェース（Ｉ／Ｆ）２０を備えたコンピュータ８０で実現することができる。ＣＰＵ１２、メモリ１４、ＨＣＡ１６、記憶部１８、及び入出力Ｉ／Ｆ２０は、バス２２を介して互いに接続されている。入出力Ｉ／Ｆ２０には、マウス、キーボード、ディスプレイ等の入出力装置が接続されている。

【００３８】

記憶部１８はＨＤＤ、ＳＳＤ（Solid State Drive）、フラッシュメモリ等によって実現できる。記録媒体としての記憶部１８には、コンピュータ８０を性能データ収集装置１０として機能させるための性能データ収集プログラム５０が記憶されている。また、記憶部１８には、解析対象プログラム６０も記憶されている。ＣＰＵ１２は、解析対象プログラム６０を記憶部１８から読み出してメモリ１４に展開し、解析対象プログラム６０が有するプロセスを順次実行する。また、ＣＰＵ１２は、解析対象プログラムを実行中に、性能データ収集プログラム５０を記憶部１８から読み出してメモリ１４に展開し、性能データ収集プログラム５０が有するプロセスを順次実行する。解析対象プログラムは特に限定されないため、ここでは詳細な説明を省略する。

【００３９】

性能データ収集プログラム５０は、採取格納プロセス５２、判定プロセス５４、及び転送制御プロセス５６を有する。ＣＰＵ１２は、採取格納プロセス５２を実行することで、図１に示す採取格納部３２として動作する。また、ＣＰＵ１２は、判定プロセス５４を実行することで、図１に示す判定部３４として動作する。また、ＣＰＵ１２は、転送制御プロセス５６を実行することで、図１に示す転送制御部３６として動作する。これにより、性能データ収集プログラム５０を実行したコンピュータ８０が、性能データ収集装置１０として機能することになる。

【００４０】

なお、ＣＰＵ１２により実現される各機能は、例えば半導体集積回路、より詳しくはＡＳＩＣ（Application Specific Integrated Circuit）等で実現することも可能である。

【００４１】

次に、本実施形態に係る性能データ収集装置１０の作用について説明する。まず、ＣＰＵ１２が解析対象プログラム６０の実行を開始する。具体的には、図９に示すように、ステップ１２０で、ＣＰＵ１２が、解析対象プログラム６０を記憶部１８から読み出してメモリ１４に展開する。この際、ＣＰＵ１２内のレジスタに格納された値などに基づいて、解析対象のプログラムに関するプロセスを表現するための情報を含む構造体がカーネル空間に生成される。

【００４２】
次に、ステップ１２２で、ＣＰＵ１２が、プログラムカウンタに格納されたアドレスから関数を取り出して命令レジスタに取り込み、命令レジスタに取り込んだ関数を実行する。この際、ＣＰＵ１２は、プログラムカウンタの値を、次に実行すべき関数が格納されたメモリ１４上のアドレスに更新するなど、ＣＰＵ１２内の各レジスタの値を適宜更新する。これにより、カーネル空間に生成された構造体に設定される情報も更新される。
【００４３】
次に、ステップ１２４で、ＣＰＵ１２４は、プログラムを終了するか否かを判定し、終了しない場合には、ステップ１２２へ戻って、次の関数を実行し、終了すると判定した場合には、解析対象プログラムの処理を終了する。
【００４４】
図９に示す解析プログラムの実行中に、ＣＰＵ１２が、図１０に示す性能データ収集処理を実行する。
【００４５】
図１０に示す性能データ収集処理のステップ１００で、採取格納部３２が、メモリ１４上に性能データ格納部３８を確保し、性能データ格納部３８の先頭アドレスａｄｄｒ＿ＲＳ、及び末尾アドレスａｄｄｒ＿ＲＥを所定の記憶領域に記憶する。また、転送制御部３６が、性能データ格納部３８内の空き領域の先頭アドレスａｄｄｒ＿１を性能データ格納部３８の先頭アドレスａｄｄｒ＿ＲＳに設定する。また、性能データ格納部３８内の空き領域の末尾アドレスａｄｄｒ＿２を性能データ格納部３８の末尾アドレスａｄｄｒ＿ＲＥに設定する。
【００４６】
次に、ステップ１０２で、採取格納部３２が、カーネル空間に生成された構造体から必要な情報を取得することにより、性能データを採取する。ここでは、例えば図２に示すように、ＣＰＵ１２の識別番号であるＣＰＵＩＤ、実行中のプロセスの識別番号であるＰＩＤ、及び実行アドレスを含む性能データを採取するものとする。採取格納部３２は、採取した性能データを、性能データ格納部３８の空き領域の先頭アドレスから格納する。
【００４７】
次に、ステップ１０４で、判定部３４が図１１に示すアイドル状態判定処理を実行する。
【００４８】
図１１に示すアイドル状態判定処理のステップ１０４０で、判定部３４が、ＣＰＵ１２が実行中のプロセスが、カーネルプロセスかユーザプロセスかを判定する。後段の処理で判定されるアイドル関数はカーネル関数であるため、実行中のプロセスがユーザプロセスの場合には、実行アドレスを判定するまでもなく、ＣＰＵ１２がアイドル状態ではないと判定することができる。例えばＬｉｎｕｘ（登録商標）の場合、性能データに含まれるＰＩＤが０であれば、実行中のプロセスはカーネルプロセス、ＰＩＤが０以外であればユーザプロセスであると判定することができる。ＣＰＵ１２により実行中のプロセスがカーネルプロセスの場合には、ステップ１０４２へ移行し、ユーザプロセスの場合には、ステップ１０５２へ移行する。
【００４９】
ステップ１０４２では、判定部３４が、カーネル関数の名前とアドレスとの対応関係を定めたシンボルテーブルを参照して、性能データに含まれる実行アドレスが、アイドル関数のアドレス範囲に含まれるか否かを判定する。実行アドレスがアイドル関数のアドレス範囲に含まれる場合には、実行アドレスがアイドル関数であると判定し、ステップ１０４４へ移行する。一方、実行アドレスがアイドル関数のアドレス範囲に含まれない場合には、実行アドレスがアイドル関数ではないと判定し、ステップ１０５２へ移行する。
【００５０】
次に、１０４４では、判定部３４が、アイドル関数の連続数を示す変数ｍを１インクリメントする。次に、ステップ１０４６で、判定部３４が、変数ｍが、アイドル状態を判定

するためのアイドル関数の連続数の閾値Ｍを超えたか否かを判定する。ｍ＞Ｍの場合には、ステップ１０４８へ移行し、判定部３４が、ＣＰＵ１２がアイドル状態であると判定する。次に、ステップ１０５０で、判定部３４が、変数ｍ及び変数ｎ（後述）を０に設定（リセット）し、性能データ収集処理へリターンする。一方、上記ステップ１０４６で、判定部３４が、ｍ≦Ｍと判定した場合には、ＣＰＵ１２がアイドル状態であると判定することなく、アイドル関数の連続数のカウントを継続するため、そのまま性能データ収集処理へリターンする。

【００５１】

また、上記ステップ１０４０またはステップ１０４２が否定判定の場合、すなわち、実行アドレスが示す関数がアイドル関数ではない場合には、ステップ１０５２で、判定部３４が、変数ｍが０を超えているか否かを判定する。ｍ＞０の場合には、判定部３４は、アイドル関数の連続数のカウントを継続中であると判定して、ステップ１０５４へ移行する。

【００５２】

ステップ１０５４では、判定部３４が、アイドル関数以外の他の関数の出現回数を示す変数ｎを１インクリメントする。次に、ステップ１０５６で、判定部３４が、変数ｎが、変数ｍが閾値Ｍに達するまでの期間内で他の関数が許容される許容数Ｎを超えたか否かを判定する。ｎ＞Ｎの場合には、アイドル関数の連続数のカウントをリセットするため、ステップ１０５０へ移行して、変数ｍ及び変数ｎを０に設定する。一方、判定部３４が、上記ステップ１０５２で、ｍ＝０と判定した場合、または上記ステップ１０５６で、ｎ≦Ｎと判定した場合には、アイドル関数の連続数のカウントを継続するため、そのまま性能データ収集処理へリターンする。

【００５３】

図１０に示す性能データ収集処理に戻って、ステップ１０６で、判定部３４が、上記ステップ１０４において、ＣＰＵ１２がアイドル状態であると判定したか否かを判定する。ＣＰＵ１２がアイドル状態であると判定した場合には、ステップ１０８へ移行し、転送制御部３６が、図１２に示す転送制御処理を実行する。ＣＰＵ１２がアイドル状態であると判定しなかった場合には、性能データ収集処理を終了する。

【００５４】

図１２に示す転送制御処理のステップ１０８０で、転送制御部３６が、性能データ格納部３８に格納された性能データの個数を示す変数ｋを１インクリメントする。次に、ステップ１０８２で、変数ｋが、一度に転送する性能データの個数として定めた閾値Ｋを超えたか否かを判定する。ｋ＞Ｋの場合には、ステップ１０８４へ移行し、転送制御部３６が、例えば図６に示すように、空き領域の末尾アドレスａｄｄｒ＿２を、転送する性能データの先頭アドレスａｄｄｒ＿ＫＳに指定する。また、Ｋ番目に格納された性能データの末尾アドレスを、転送する性能データの末尾アドレスａｄｄｒ＿ＫＥに指定する。転送制御部３６は、アドレスａｄｄｒ＿ＫＳ及びａｄｄｒ＿ＫＥと、転送先のリモート装置４０のメモリ４４を指定する情報とを含む転送情報をＨＣＡ１６に設定することにより、ＨＣＡ１６に性能データの転送を指示する。

【００５５】

次に、ステップ１０８６で、転送制御部３６が、例えば図７に示すように、空き領域の末尾アドレスａｄｄｒ＿２を、転送する性能データの末尾アドレスａｄｄｒ＿ＫＥに更新する。次に、ステップ１０８８で、転送制御部３６が、変数ｋを、ｋから転送した性能データの個数Ｋを差し引いた値に更新して、図１０に示す性能データ収集処理へリターンする。

【００５６】

一方、上記ステップ１０８２で、転送制御部３６が、ｋ≦Ｋと判定した場合には、まだ、性能データの転送タイミングではないため、ステップ１０９０へ移行する。ステップ１０９０では、転送制御部３６が、例えば図５に示すように、空き領域の先頭アドレスａｄｄｒ＿１を、性能データ格納部３８にｋ番目に格納された性能データの末尾アドレスａｄ

ｄｒ＿ｋＥに更新して、図１０に示す性能データ収集処理へリターンする。

【００５７】

図１０に示す性能データ収集処理に戻って、ステップ１１０で、転送制御部３６が、空き領域の先頭アドレスａｄｄｒ＿１と末尾アドレスａｄｄｒ＿２とを比較して、空き領域のサイズを求める。転送制御部３６は、求めた空き領域のサイズが所定サイズ以上か否かを判定する。空き領域が所定サイズ以上の場合には、性能データの採取及び格納を継続可能であると判定して、ステップ１１２へ移行する。

【００５８】

ステップ１１２では、採取格納部３２が、予め設定されたサンプリング期間を終了したか、または解析対象プログラム６０の実行が終了したか否かを判定することにより、性能データの採取を終了するか否かを判定する。性能データの採取を継続する場合には、ステップ１０２へ戻って、サンプリング間隔毎に性能データの採取及び格納を繰り返す。

【００５９】

一方、上記ステップ１１０で、転送制御部３６が、空き領域が所定サイズに満たないと判定した場合、または上記ステップ１１２で、採取格納部３２が、性能データの採取を終了すると判定した場合には、性能データ収集処理を終了する。これにより、メモリ１４及びリモート装置４０のメモリ４４に、複数の性能データが収集される。

【００６０】

以上説明したように、本実施形態に係る性能データ収集装置１０によれば、ＣＰＵがアイドル状態の場合、すなわちＣＰＵが性能に影響を与えるような処理を実行していない場合に、メモリ上の性能データ格納部に格納された性能データをリモート装置へＲＤＭＡ転送する。これにより、ＣＰＵの処理負荷が所定値以下の場合に転送指示を行うため、収集した性能データの解析時において、性能データに影響を与えた要因の切り分けが容易になる。そして、解析対象のプログラムの動作に与える影響を抑制して、採取した性能データを転送しない場合に比べて、大量の性能データを収集することができる。

【００６１】

また、ＣＰＵがアイドル状態であり、かつメモリに格納された性能データが所定量を超えた場合に、性能データの転送を行うことで、転送回数を低減させることができるため、より解析対象のプログラムの動作に与える影響を抑制することができる。

【００６２】

収集した性能データは、解析対象プログラムの解析に用いることができる。例えば、収集した性能データを集計し、性能ボトルネックとなる呼び出し回数の多い関数を見つけることができる。また、性能データに採取時刻を示すタイムスタンプを付加した場合には、サンプリング期間全体の統計だけでなく、時系列分析も可能になる。

【００６３】

なお、上記実施形態では、ＣＰＵを介さないデータ転送としてＲＤＭＡ転送を行う場合について説明したが、ＣＰＵを介さない、またはＣＰＵの負荷が小さい転送方法であれば、例えばＤＭＡ（Direct Memory Access）転送などの他の転送方法を用いてもよい。

【００６４】

図１３に、ＤＭＡ転送を行う場合の性能データ収集装置２１０の一例を示す。性能データ収集装置２１０は、ＣＰＵ１２、メモリ１４、記憶部１８、及びＤＭＡＣ（Direct Memory Access controller）２４を含んでいる。なお、上記実施形態に係る性能データ収集装置１０と同一の部分については同一符号を付している。性能データ収集装置２１０では、主記憶部であるメモリ１４から、リモート装置４０のメモリ４４ではなく、自装置内の補助記憶部である記憶部１８へデータが転送される。転送制御部３６は、上記と同様に、転送する性能データのアドレス範囲及び転送先を指定する転送情報をＤＭＡＣ２４に設定することにより、ＤＭＡＣ２４に性能データの転送を指示する。性能データの転送を指示されたＤＭＡＣ２４は、設定された転送情報に基づいて、ＣＰＵ１２を介することなく、性能データ格納部３８内の指定されたアドレスに格納された性能データを、記憶部１８へＤＭＡ転送する。

【００６５】
　また、上記では開示の技術における性能データ収集プログラムの一例である性能データ収集プログラム５０が記憶部１８に予め記憶（インストール）されている態様を説明した。しかし、開示の技術における性能データ収集プログラムは、ＣＤ－ＲＯＭやＤＶＤ－ＲＯＭ等の記録媒体に記録されている形態で提供することも可能である。
【００６６】
　以上の実施形態に関し、更に以下の付記を開示する。
【００６７】
　（付記１）
　演算処理装置と転送制御部とを有する情報処理装置に、
　所定の関数で記述された解析対象のプログラムを実行する前記演算処理装置が呼び出した関数を示す情報を含む性能データを、所定のサンプリング間隔で採取して、主記憶部に格納させ、
　採取した性能データに基づいて、前記演算処理装置の処理負荷を判定させ、
　前記演算処理装置の処理負荷が所定値以下の場合に、前記主記憶部に格納した性能データの少なくとも一部を、前記転送制御部により前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送させる
　ことを特徴とする性能データ収集プログラム。
【００６８】
　（付記２）
　前記演算処理装置の処理負荷の判定は、
　前記情報処理装置に、
　採取した性能データに基づいて、前記演算処理装置がアイドル状態かを判定させることを特徴とする付記１記載の性能データ収集プログラム。
【００６９】
　（付記３）
　前記演算処理装置の処理負荷の判定は、
　前記情報処理装置に、
　所定期間に採取した複数の性能データの各々に含まれる情報が示す関数におけるアイドル関数の割合が所定割合以上の場合に、前記演算処理装置がアイドル状態であると判定させることを特徴とする付記２記載の性能データ収集プログラム。
【００７０】
　（付記４）
　前記主記憶部以外の記憶部への転送は、
　前記情報処理装置に、
　前記主記憶部に格納した性能データが所定量を超えた場合に、前記主記憶部に格納した性能データの少なくとも一部を、前記転送制御部により前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送させることを特徴とする付記１～付記３のいずれか１つに記載の性能データ収集プログラム。
【００７１】
　（付記５）
　所定の関数で記述された解析対象のプログラムを実行する演算処理装置と、
　前記演算処理装置が呼び出した関数を示す情報を含む性能データを、所定のサンプリング間隔で採取して、主記憶部に格納する採取格納部と、
　採取された性能データに基づいて、前記演算処理装置の処理負荷を判定する判定部と、
　前記演算処理装置の処理負荷が所定値以下の場合に、前記主記憶部に格納された性能データの少なくとも一部を、前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送する転送制御部と、
　を含む性能データ収集装置。
【００７２】

（付記６）
前記判定部は、採取された性能データに基づいて、前記演算処理装置がアイドル状態かを判定することを特徴とする付記５記載の性能データ収集装置。
【００７３】
（付記７）
前記判定部は、所定期間に採取された複数の性能データの各々に含まれる情報が示す関数におけるアイドル関数の割合が所定割合以上の場合に、前記演算処理装置がアイドル状態であると判定することを特徴とする付記６記載の性能データ収集装置。
【００７４】
（付記８）
前記転送制御部は、前記主記憶部に格納された性能データが所定量を超えた場合に、前記主記憶部に格納された性能データの少なくとも一部を、前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送させることを特徴とする付記５～付記７のいずれか１つに記載の性能データ収集プログラム。
【００７５】
（付記９）
演算処理装置と転送制御部とを有する情報処理装置が、
所定の関数で記述された解析対象のプログラムを実行する前記演算処理装置が呼び出した関数を示す情報を含む性能データを、所定のサンプリング間隔で採取して、主記憶部に格納し、
採取した性能データに基づいて、前記演算処理装置の処理負荷を判定し、
前記演算処理装置の処理負荷が所定値以下の場合に、前記主記憶部に格納した性能データの少なくとも一部を、前記転送制御部により前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送する
ことを特徴とする性能データ収集方法。
【００７６】
（付記１０）
前記演算処理装置の処理負荷の判定は、
前記情報処理装置が、
採取した性能データに基づいて、前記演算処理装置がアイドル状態かを判定することを特徴とする付記９記載の性能データ収集方法。
【００７７】
（付記１１）
前記演算処理装置の処理負荷の判定は、
前記情報処理装置が、
所定期間に採取した複数の性能データの各々に含まれる情報が示す関数におけるアイドル関数の割合が所定割合以上の場合に、前記演算処理装置がアイドル状態であると判定させることを特徴とする付記１０記載の性能データ収集方法。
【００７８】
（付記１２）
前記主記憶部以外の記憶部への転送は、
前記情報処理装置が、
前記主記憶部に格納した性能データが所定量を超えた場合に、前記主記憶部に格納した性能データの少なくとも一部を、前記転送制御部により前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送することを特徴とする付記９～付記１１のいずれか１つに記載の性能データ収集方法。
【００７９】
（付記１３）
演算処理装置と転送制御部とを有する情報処理装置に、
所定の関数で記述された解析対象のプログラムを実行する前記演算処理装置が呼び出し

た関数を示す情報を含む性能データを、所定のサンプリング間隔で採取して、主記憶部に格納させ、
採取した性能データに基づいて、前記演算処理装置の処理負荷を判定させ、
前記演算処理装置の処理負荷が所定値以下の場合に、前記主記憶部に格納した性能データの少なくとも一部を、前記転送制御部により前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送させ、
前記主記憶部及び前記主記憶部以外の記憶部に収集した複数の性能データを集計して、前記解析対象のプログラムを解析させる
ことを特徴とする解析プログラム。

【００８０】

（付記１４）
所定の関数で記述された解析対象のプログラムを実行する演算処理装置と、
前記演算処理装置が呼び出した関数を示す情報を含む性能データを、所定のサンプリング間隔で採取して、主記憶部に格納する採取格納部と、
採取された性能データに基づいて、前記演算処理装置の処理負荷を判定する判定部と、
前記演算処理装置の処理負荷が所定値以下の場合に、前記主記憶部に格納された性能データの少なくとも一部を、前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送する転送制御部と、
前記主記憶部及び前記主記憶部以外の記憶部に収集された複数の性能データを集計して、前記解析対象のプログラムを解析する解析部と、
を含む解析装置。

【００８１】

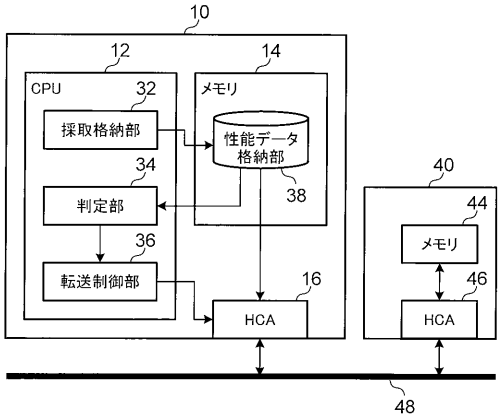
（付記１５）
演算処理装置と転送制御部とを有する情報処理装置が、
所定の関数で記述された解析対象のプログラムを実行する前記演算処理装置が呼び出した関数を示す情報を含む性能データを、所定のサンプリング間隔で採取して、主記憶部に格納し、
採取した性能データに基づいて、前記演算処理装置の処理負荷を判定し、
前記演算処理装置の処理負荷が所定値以下の場合に、前記主記憶部に格納した性能データの少なくとも一部を、前記転送制御部により前記演算処理装置を介さずに、前記主記憶部以外の記憶部へ転送し、
前記主記憶部及び前記主記憶部以外の記憶部に収集した複数の性能データを集計して、前記解析対象のプログラムを解析する
ことを特徴とする解析方法。

【符号の説明】

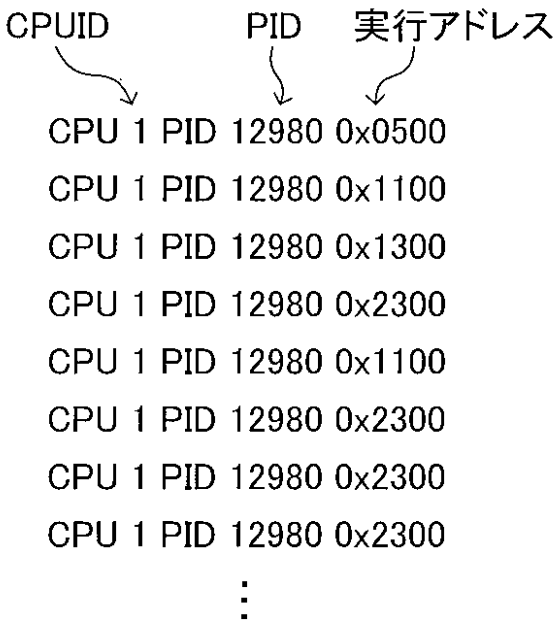
【００８２】

１０　性能データ収集装置
１４　メモリ
１６　ＨＣＡ
１８　記憶部
２４　ＤＭＡＣ
３２　採取格納部
３４　判定部
３６　転送制御部
３８　性能データ格納部
４０　リモート装置
４４　リモート装置のメモリ
８０　コンピュータ

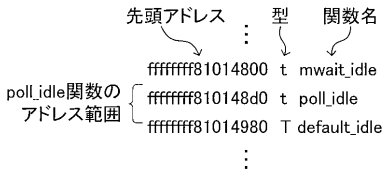
【図１】



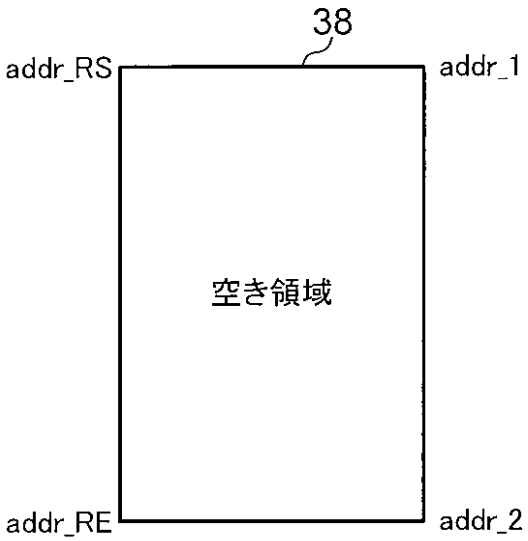
【図２】



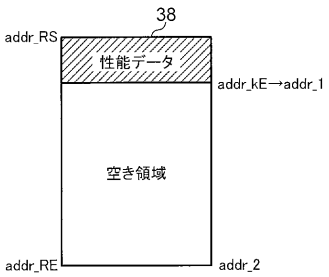
【図３】



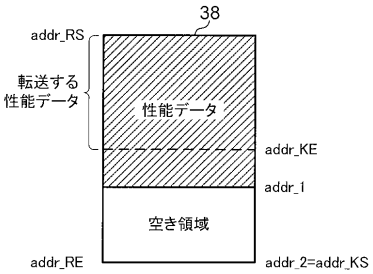
【図４】



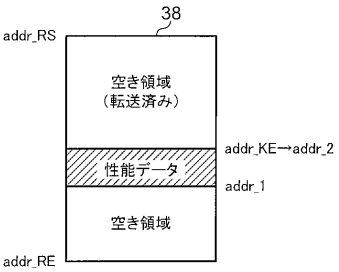
【図５】



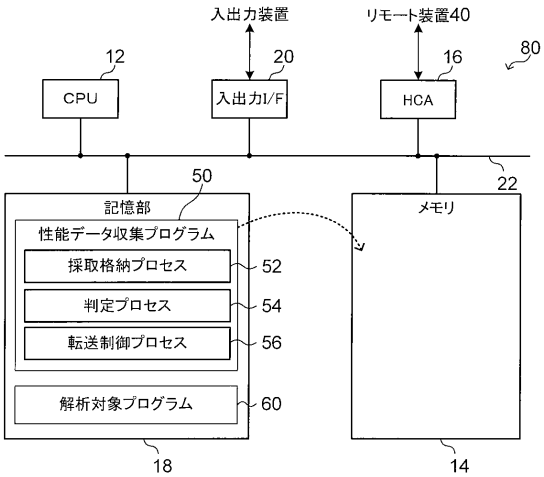
【図６】



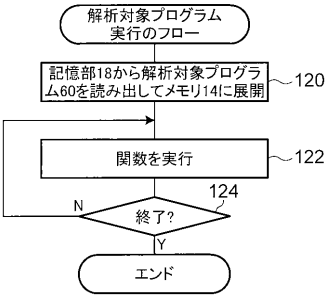
【図７】



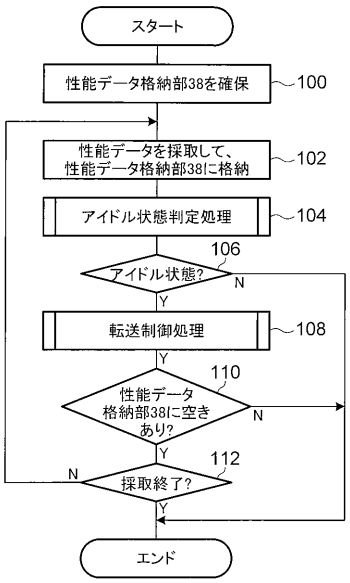
【図８】



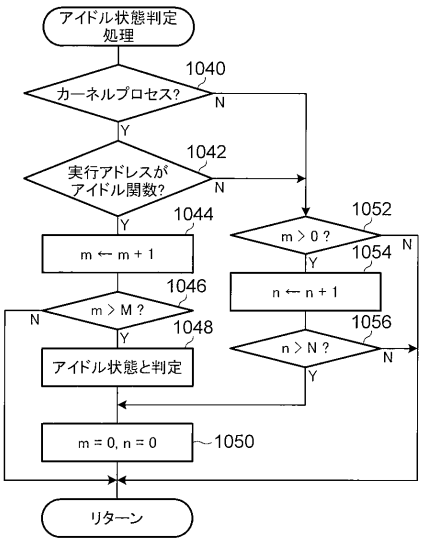
【図９】



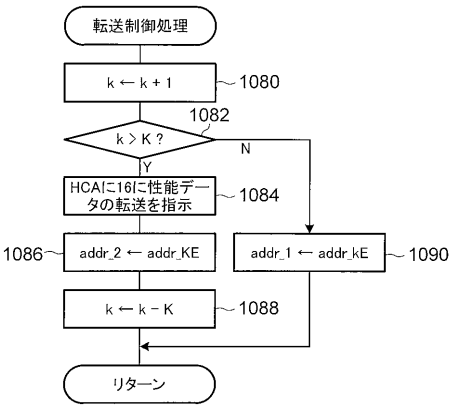
【図１０】



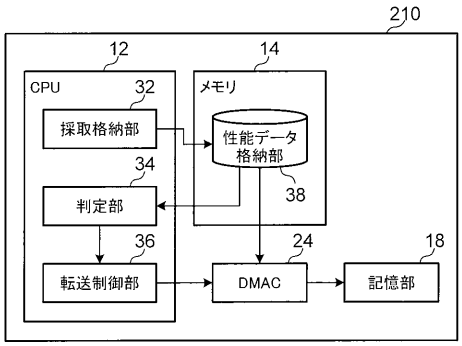
【図１１】



【図１２】


転送制御処理
k ← k + 1
1080
1082
k > K ?
N
Y
HCAに16に性能データの転送を指示
1084
1086
addr_2 ← addr_KE
addr_1 ← addr_kE
1090
k ← k − K
1088
リターン

【図１３】


210
12
14
CPU
32
メモリ
採取格納部
性能データ格納部
34
38
判定部
36
24
18
転送制御部
DMAC
記憶部

フロントページの続き

(72)発明者　中島　耕太
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

審査官　多胡　滋

(56)参考文献　特開２００５－３３９１０７（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１１／３４