

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6171658号
(P6171658)

(45) 発行日 平成29年8月2日(2017.8.2)

(24) 登録日 平成29年7月14日(2017.7.14)

(51) Int.Cl.

F I

G 0 6 F 9/50 (2006.01)

G 0 6 F 9/46 4 6 5 Z

請求項の数 7 (全 21 頁)

(21) 出願番号 特願2013-150945 (P2013-150945)
 (22) 出願日 平成25年7月19日(2013.7.19)
 (65) 公開番号 特開2015-22574 (P2015-22574A)
 (43) 公開日 平成27年2月2日(2015.2.2)
 審査請求日 平成28年4月5日(2016.4.5)

(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 100089118
 弁理士 酒井 宏明
 (72) 発明者 福本 尚人
 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
 (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 漆原 孝治

最終頁に続く

(54) 【発明の名称】 並列処理最適化プログラム、並列処理最適化方法および情報処理装置

(57) 【特許請求の範囲】

【請求項 1】

コンピュータに、

並列処理を実行可能なコア数の範囲内で、並列して実行する所定数の処理を、割り振る
最大処理数を変えて、それぞれ当該最大処理数以下の範囲で、かつ、処理が割り振られる
 コアの数 が最も少なくなるようにコアに割り振る場合に、前記最大処理数数毎に、処理が
 割り振られる コア数を特定し、

特定したコア数のなかから、処理性能の最も高いコア数を並列処理を実行するコア数と
 決定する

処理を実行させることを特徴とする並列処理最適化プログラム。

10

【請求項 2】

前記コアは、並列処理を実行可能な複数のスレッドを有し、

前記特定する処理は、コアの並列処理を実行可能なスレッド数および並列処理を実行可
 能なコア数の範囲内で、各コアで動作させるスレッド数および前記最大処理数を変えて、
それぞれ当該最大処理数以下の範囲で、かつ、処理が割り振られるコアの数 が最も少く
 なるようにスレッドに前記所定数の処理を割り振る場合に、処理が割り振られるスレッド
 数およびコア数を特定し、

前記決定する処理は、特定したスレッド数およびコア数のなかから、処理性能の最も高
 いスレッド数およびコア数を並列処理を実行するスレッド数およびコア数と決定する

ことを特徴とする請求項 1 に記載の並列処理最適化プログラム。

20

【請求項 3】

前記特定する処理は、特定したコア数のなかから、コア数の多い順に性能の評価を行って性能を順次比較し、少ないコア数の性能が多いコア数の性能を最初に下回った際の多い側のコア数を並列処理を実行するコア数と決定する

ことを特徴とする請求項 1 に記載の並列処理最適化プログラム。

【請求項 4】

前記特定する処理は、特定したスレッド数およびコア数をスレッド数が多いものから、スレッド数毎に、コア数の多い順に性能の評価を行って性能を順次比較し、少ないコア数の性能が多いコア数の性能を最初に下回った際の多い側のコア数を当該スレッド数の最も良い性能のコア数と特定し、少ないスレッド数の最も良い性能が多いスレッド数の最も良い性能を最初に下回った際の多い側のスレッド数およびコア数を並列処理を実行するスレッド数およびコア数と決定する

ことを特徴とする請求項 2 に記載の並列処理最適化プログラム。

【請求項 5】

前記決定する処理は、特定したスレッド数およびコア数のなかから、所定の予測モデルを用いて、並列処理を実行するスレッド数およびコア数を決定する

ことを特徴とする請求項 2 に記載の並列処理最適化プログラム。

【請求項 6】

コンピュータが、

並列処理を実行可能なコア数の範囲内で、並列して実行する所定数の処理を、割り振る最大処理数を変えて、それぞれ当該最大処理数以下の範囲で、かつ、処理が割り振られるコアの数が最も少なくなるようにコアに割り振る場合に、前記最大処理数数毎に、処理が割り振られるコア数を特定し、

特定したコア数のなかから、処理性能の最も高いコア数を並列処理を実行するコア数と決定する

処理を実行することを特徴とする並列処理最適化方法。

【請求項 7】

並列処理を実行可能な複数のコアと、

並列処理を実行可能なコア数の範囲内で、並列して実行する所定数の処理を、割り振る最大処理数を変えて、それぞれ当該最大処理数以下の範囲で、かつ、処理が割り振られるコアの数が最も少なくなるようにコアに割り振る場合に、前記最大処理数数毎に、処理が割り振られるコア数を特定する特定部と、

前記特定部により特定されたコア数のなかから、処理性能の最も高いコア数を並列処理を実行するコア数と決定する決定部と、

を有することを特徴とする情報処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、並列処理最適化プログラム、並列処理最適化方法および情報処理装置に関する。

【背景技術】

【0002】

近年、複数のコアを搭載したマルチコア・プロセッサが主流となっている。このようなマルチコア・プロセッサは、複数コアで並列処理を行うことで処理性能を高められる。マルチコア・プロセッサは、微細化技術の進歩とともに、チップに搭載されるコア数が増加する傾向にあり、数十コアが搭載されたものも販売されている。

【先行技術文献】

【特許文献】

【0003】

【特許文献 1】特開 2001 - 282553 号公報

10

20

30

40

50

【特許文献2】国際公開第2004/044745号

【特許文献3】特開2006-24088号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

ところで、並列処理を行うプログラムには、マルチコア・プロセッサの全てのコアで並列処理を行うことで高性能化を達成できるものがある。また、一部のコアを停止させる方が、高い性能が得られるプログラムもある。例えば、コア間の同期が多いプログラムや、メモリバンド幅ネックのプログラムでは、一部のコアを停止させる方が高い性能が得られる場合がある。

10

【0005】

一側面では、高効率で並列処理を実行可能なコア数を速やかに求めることができる並列処理最適化プログラム、並列処理最適化方法および情報処理装置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】

【0006】

本発明の一側面によれば、並列処理最適化プログラムは、コンピュータに、並列処理を実行可能なコア数の範囲内で、並列して実行する所定数の処理を割り振る処理数を変えて同程度ずつコアに割り振る場合に、処理が実行されるコア数を特定する処理を実行させる。また、並列処理最適化プログラムは、コンピュータに、並列処理を実行可能なコア数の範囲内で、特定したコア数のなかから、処理性能の最も高いコア数を並列処理を実行するコア数と決定する処理を実行させる。

20

【発明の効果】

【0007】

本発明の一側面によれば、高効率で並列処理を実行可能なコア数を速やかに求めることができる。

【図面の簡単な説明】

【0008】

【図1】図1は、情報処理装置の概略的な構成の一例を示した図である。

【図2】図2は、情報処理装置におけるハードウェアとソフトウェアとの関係を説明するための図である。

30

【図3】図3は、並列処理が可能な処理を示す図である。

【図4】図4は、並列処理プログラムと並列処理最適化プログラムをスタティックリンクさせる一例を示す図である。

【図5A】図5Aは、NPB EPのコア数毎の性能の評価結果の一例を示す図である。

【図5B】図5Bは、NPB FTのコア数毎の性能の評価結果の一例を示す図である。

【図6】図6は、実行された並列処理最適化プログラムの機能的な構成を示す図である。

【図7】図7は、NPB FTのコア数毎の性能の評価結果の一例を示す図である。

【図8】図8は、評価結果の一例を示す図である。

【図9】図9は、実施例1に係る並列処理最適化処理の手順の一例を示すフローチャートである。

40

【図10】図10は、実施例1に係る試行パターン特定処理の手順の一例を示すフローチャートである。

【図11】図11は、実施例2に係るプロセッサの構成を模式的に示した図である。

【図12A】図12Aは、NPB EPのコア数毎の性能の評価結果の一例を示す図である。

【図12B】図12Bは、NPB FTのコア数毎の性能の評価結果の一例を示す図である。

【図13】図13は、NPB FTのコア数毎の性能の評価結果の一例を示す図である。

【図14】図14は、評価結果の一例を示す図である。

50

【図 1 5】図 1 5 は、実施例 2 に係る並列処理最適化処理の手順の一例を示すフローチャートである。

【図 1 6】図 1 6 は、実施例 2 に係る試行パターン特定処理の手順の一例を示すフローチャートである。

【図 1 7】図 1 7 は、実施例 3 に係る並列処理最適化処理の手順の一例を示すフローチャートである。

【発明を実施するための形態】

【0009】

以下に、本発明にかかる並列処理最適化プログラム、並列処理最適化方法および情報処理装置の実施例を図面に基づいて詳細に説明する。なお、この実施例によりこの発明が限定されるものではない。そして、各実施例は、処理内容を矛盾させない範囲で適宜組み合わせることが可能である。

10

【実施例 1】

【0010】

〔全体構成〕

実施例 1 に係る情報処理装置 10 の構成について説明する。図 1 は、情報処理装置の概略的な構成の一例を示した図である。

【0011】

情報処理装置 10 は、例えば、パーソナルコンピュータやサーバコンピュータ、ワークステーションなどの情報処理を行う装置である。なお、情報処理装置 10 は、例えば、スマートフォン、タブレット端末、PDA (Personal Digital Assistants) などの携帯端末であってもよい。図 1 に示すように、情報処理装置 10 は、プロセッサ 20 と、DISK 21 と、RAM (Random Access Memory) 22 とを有する。これらプロセッサ 20、DISK 21、RAM 22 は、バス 23 を介して接続される。なお、情報処理装置 10 は、図 1 に示した機能部以外にも既知の情報処理装置 10 が有する各種の機能部を有することとしてもかまわない。例えば、情報処理装置 10 は、通信インタフェースや、キーボード、マウスなどの操作部、表示部などその他の機能部を有することとしてもかまわない。

20

【0012】

プロセッサ 20 は、複数のコア 24 を有し、演算処理を実行するマルチコア・プロセッサである。プロセッサ 20 は、各コア 24 により並列に演算処理を実行可能とされている。なお、図 1 の例では、プロセッサ 20 が 4 個のコア 24 を有する場合を示したが、コア 24 の数はこれに限定されるものではない。マルチコア・プロセッサは、微細化技術の進歩とともに、チップに搭載されるコア数が増加している。例えば、インテル社製の Xeon Phi Coprocessor 5110P は、コア 24 が 60 個の搭載されている。

30

【0013】

DISK 21 は、各種の情報を記憶するデバイスである。例えば、DISK 21 は、OS (Operating System) 30 と、並列処理プログラム 31 と、並列処理最適化プログラム 32 とを記憶する。この DISK 21 としては、例えば、HDD (Hard Disk Drive)、SSD (Solid State Drive) などが挙げられる。なお、DISK 21 に代えて、フラッシュメモリなどの不揮発性の半導体メモリを各種の情報を記憶するデバイスとして用いてもよい。

40

【0014】

OS 30 は、多くのアプリケーションソフトから共通して利用される基本的な機能を提供し、コンピュータ全体を管理するソフトウェアである。OS 30 は、情報処理装置 10 の起動時にプロセッサ 20 により実行される。

【0015】

並列処理プログラム 31 は、情報処理装置 10 において実行されるアプリケーションソフトである。並列処理プログラム 31 には、並列処理が可能な処理が含まれている。例えば、並列処理プログラム 31 には、ループ処理など並列処理が可能な処理が含まれている

50

。

【 0 0 1 6 】

並列処理最適化プログラム 3 2 は、並列処理の制御および最適化を行うプログラムである。例えば、並列処理最適化プログラム 3 2 は、並列処理を実行する際に処理を実行させるコア数を最適化する。

【 0 0 1 7 】

R A M 2 1 は、各種のデータを一時的に記憶するデバイスである。R A M 2 1 には、O S 3 0 やアプリケーションソフトが利用する各種のデータが展開されて記憶される。

【 0 0 1 8 】

次に、図 2 を用いて、情報処理装置 1 0 におけるハードウェアとソフトウェアとの関係について説明する。図 2 は、情報処理装置におけるハードウェアとソフトウェアとの関係を説明するための図である。なお、図 2 に示す例では、ソフトウェアとして、O S 3 0 と、並列処理プログラム 3 1 と、並列処理最適化プログラム 3 2 とが示されている。

10

【 0 0 1 9 】

O S 3 0 は、コンピュータ全体を管理しており、アプリケーションプログラムおよびハードウェアの監視や、割りこみ処理など、O S 3 0 としての基本機能を提供する。例えば、O S 3 0 は、プロセッサ 2 0 の各コア 2 4 の状況の監視やコア 2 4 に対する割りこみ処理などの機能を提供する。

【 0 0 2 0 】

並列処理プログラム 3 1 は、所定のプログラム記述言語で記述されたソースコード 4 0 をコンパイラ 4 1 によりコンパイルすることにより作成される。並列処理プログラム 3 1 は、ループ処理などの並列処理が可能な処理が含まれる。

20

【 0 0 2 1 】

図 3 は、並列処理が可能な処理の一例を示す図である。図 3 の例では、変数 i の初期値をゼロとし、処理を 1 回行う毎に変数 i の値を 1 つずつ加算し、変数 i の値が N よりも小さい間、処理を繰り返すループ処理が示されている。このような処理を N 回繰り返し行うループ処理は、N 回繰り返す処理をそれぞれ最大で N 等分に分割して並列に処理させることが可能である。以下では、このようなループ処理などで繰り返し行われる処理の繰り返し回数をイタレーション数とも言う。図 3 の例は、イタレーション数が N のループ処理である。

30

【 0 0 2 2 】

図 2 に戻り、並列処理最適化プログラム 3 2 は、例えば、ライブラリとして提供される。並列処理最適化プログラム 3 2 は、ソースコード 4 0 をコンパイルする際に並列処理プログラム 3 1 とリンクされる。並列処理最適化プログラム 3 2 は、並列処理プログラム 3 1 が並列処理を実行する際に呼び出され、O S 3 0 を介してプロセッサ 2 0 の各コア 2 4 への処理の割り振りを制御する。なお、並列処理プログラム 3 1 と並列処理最適化プログラム 3 2 のリンクは、スタティックリンクであってもよく、ダイナミックリンクであってもよい。

【 0 0 2 3 】

図 4 は、並列処理プログラムと並列処理最適化プログラムをスタティックリンクさせる一例を示す図である。例えば、既存の並列処理ライブラリ 4 2 に対して、修正を行って並列処理最適化プログラム 3 2 を作成する。この既存の並列処理ライブラリ 4 2 としては、例えば、O p e n M P が挙げられる。そして、コンパイラ 4 1 によりソースコード 4 0 をコンパイルする際のリンク先を並列処理最適化プログラム 3 2 に変更してコンパイルを行う。これにより、並列処理最適化プログラム 3 2 では、並列処理を行う際に並列処理最適化プログラム 3 2 が実行される。

40

【 0 0 2 4 】

ところで、並列処理の性能を評価するベンチマークとしては、N P B (N A S P a r a l l e l B e n c h m a r k s) がある。以下では、コア 2 4 が 6 0 個のプロセッサ 2 0 を用いて、N P B 3 . 3 O p e n M P 版の E P , F T により性能の評価を行った例を説明する。なお、入

50

力サイズは、C l a s s Bとする。

【 0 0 2 5 】

図 5 A は、N P B E P のコア数毎の性能の評価結果の一例を示す図である。図 5 A の横軸は、並列処理を実行させたコア数を示す。図 5 A の縦軸は、コア数が 1 での性能に対する相対性能を示す。相対性能は、例えば、各コア数での性能値を、コア数が 1 での性能値で除算した値である。図 5 A に示すように、N P B E P では、並列処理を行うコア数が多いほど高い性能が得られる。図 5 A の例では、コア数が 6 0 の場合に最も高い性能が得られている。

【 0 0 2 6 】

図 5 B は、N P B F T のコア数毎の性能の評価結果の一例を示す図である。図 5 B の横軸は、並列処理を実行させたコア数を示す。図 5 B の縦軸は、コア数が 1 での性能に対する相対性能を示す。図 5 B に示すように、N P B F T では、コア数に比例して一律に性能が高くなり、特定のコア数で性能が飽和して低下する。例えば、図 5 B の例では、コア数が 3 3 の場合に性能がピークとなる。このように特定のコア数で並列処理を行った方が高い性能が得られるプログラムもある。例えば、コア間の同期が多いプログラムや、メモリバンド幅ネックのプログラムでは、特定のコア数で並列処理を行った方が高い性能を得られる場合がある。

【 0 0 2 7 】

次に、図 6 を用いて、実行された並列処理最適化プログラム 3 2 の機能的な構成について説明する。図 6 は、実行された並列処理最適化プログラムの機能的な構成を示す図である。図 6 に示すように、並列処理最適化プログラム 3 2 は、O S 3 0 を介してプロセッサ 2 0 の各コア 2 4 への処理の割り振りを制御しており、特定部 5 0 と、決定部 5 1 と、プロセッサ制御部 5 2 とを有する。

【 0 0 2 8 】

特定部 5 0 は、各種の特定を行う。例えば、特定部 5 0 は、性能の評価対象とするコア数を特定する。

【 0 0 2 9 】

ここで、例えば、イタレーション数 N を 2 5 6 とし、プロセッサ 2 0 のコア数を 6 0 とした場合を例にして説明する。並列処理では、複数のコアに処理を割り振った場合、並列処理を行う全てのコア 2 4 で処理が完了するまでの時間により処理性能が定まる。すなわち、並列処理では、最も遅く処理が完了するコア 2 4 での実行時間により処理性能が定まる。

【 0 0 3 0 】

また、並列処理では、並列して実行可能な処理を各コア 2 4 に同程度ずつ割り振ったとしてもコア 2 4 あたりの処理数が不均一になることが多い。特に、イタレーション数 N がコア数に対してあまり大きくない場合、コア 2 4 あたりの処理数の違いより処理性能が階段状になる。例えば、イタレーション数 N が 2 5 6 のループ処理を同程度ずつコアに割り振るものとして、例えば、5 2 個のコア 2 4 で並列処理を行うものとした場合、コアあたりに割り振られる最大のイタレーション数は、5 個である。また、イタレーション数 N が 2 5 6 のループ処理を同程度ずつコアに割り振るものとして、例えば、6 0 個のコア 2 4 で並列処理を行うものとした場合も、コアあたりに割り振られる最大のイタレーション数は、5 個である。この場合、割り振られる最大のイタレーション数が共に 5 個であるため、5 2 個のコア 2 4 で並列処理した場合と 6 0 個のコア 2 4 で並列処理した場合の処理性能は、略同等となる。すなわち、並列処理で処理を行うコア数が変わっても、最大のイタレーション数が同じ場合、処理性能は、略同等となる。

【 0 0 3 1 】

一方、並列処理で処理を行うコア数が多いほど、消費電力は多くなる。

【 0 0 3 2 】

そこで、特定部 5 0 は、並列処理を実行可能なコア数の範囲内で、イタレーション数の処理を、割り振る処理数を変えて同程度ずつコア 2 4 に割り振る場合に、処理が実行され

10

20

30

40

50

るコア数を特定する。例えば、特定部 5 0 は、イタレーション数 N を 2 5 6 とし、プロセッサ 2 0 のコア数を 6 0 とし、割り振る処理数を a とした場合、処理が実行されるコア数を次のように特定する。

【 0 0 3 3 】

特定部 5 0 は、割り振る処理数 a を 1 から順に 1 つずつ変化させ、イタレーション数 N を割り振る処理数 a で除算し、除算した値の少数点以下を切り上げた値 X を、値 X が 1 以上である間それぞれ求める。これにより、例えば、 $a = 1$ の場合は、 $X = 2 5 6$ 、 $a = 2$ の場合は、 $X = 1 2 8$ 、 $a = 3$ の場合は、 $X = 8 6$ 、 $a = 4$ の場合は、 $X = 6 4$ 、 $a = 5$ の場合は、 $X = 5 2$ 、 $a = 6$ の場合は、 $X = 4 3 \cdots$ とそれぞれ求まる。この値 X は、 a 個程度ずつ処理をコア 2 4 に割り振る場合に、並列処理が実行されるコア数である。

10

【 0 0 3 4 】

しかし、プロセッサ 2 0 のコア数は、6 0 個である。よって、値 X が並列処理を実行可能なコア数である 6 0 を超えている場合、並列処理できない。そこで、特定部 5 0 は、値 X が 6 0 以下の範囲内で、値 X を特定する。例えば、上述の例では、値 X は、5 2、4 3、3 7、3 2、2 9、2 6、2 4、2 2、2 0、1 9、 \cdots 、1 と特定される。この特定された値 X が、それぞれ処理が実行されるコア数である。以下、特定部 5 0 により特定された各値 X を試行パターンと呼ぶ。

【 0 0 3 5 】

図 7 は、N P B F T のコア数毎の性能の評価結果の一例を示す図である。図 7 の例は、並列処理を実行するコア数を 1 ~ 6 0 の範囲で変えて同程度ずつコア 2 4 に処理を割り振った場合の並列処理を実行したコア数毎の性能の一例である。図 7 の横軸は、並列処理を実行させたコア数を示す。図 7 の縦軸は、コア数が 1 での性能に対する相対性能を示す。図 7 に示すように、イタレーション数 N がコア数に対してあまり大きくない場合、コア 2 4 あたりの処理数の違いより処理性能が階段状になる。例えば、図 7 では、コア数が 4 3 ~ 5 1 の範囲は、性能が同程度である。また、コア数が 5 2 ~ 6 0 の範囲も、性能が同程度である。特定部 5 0 は、この階段状に性能が変わる各コア数を試行パターンとして求めている。例えば、特定部 5 0 は、図 7 において丸を付した 5 2、4 3、3 7、3 2、2 9 のコア数を特定している。

20

【 0 0 3 6 】

図 6 に戻り、決定部 5 1 は、各種の決定を行う。例えば、決定部 5 1 は、特定部 5 0 により特定されたコア数のなかから、処理性能の最も高いコア数を並列処理を実行するコア数と決定する。一例として、決定部 5 1 は、試行パターンの各コア数のなかから、ヒルクライム法により、並列処理を実行するコア数と決定する。例えば、決定部 5 1 は、試行パターンの各コア数のなかから、コア数の多い順に、コア数のコア 2 4 を用いてイタレーション数 N の処理を並列に実行して性能の評価を行う。そして、決定部 5 1 は、コア数の多い順に性能を順次比較する。そして、決定部 5 1 は、少ないコア数の性能が多いコア数の性能を最初に下回った際の多い側のコア数を並列処理を実行するコア数と決定する。

30

【 0 0 3 7 】

例えば、図 7 の例では、決定部 5 1 は、5 2 個のコア 2 4 によりイタレーション数 N の処理を並列に実行して性能の評価を行う。次に、決定部 5 1 は、4 3 個のコア 2 4 によりイタレーション数 N の処理を並列に実行して性能の評価を行う。そして、決定部 5 1 は、コア数が 5 2 個の場合の性能と、コア数が 4 3 個の場合の性能を比較する。図 7 の例では、コア数が 4 3 個の場合の性能は、コア数が 5 2 個の場合の性能よりも下回っている。この場合、決定部 5 1 は、並列処理を実行するコア数を 5 2 個と決定する。

40

【 0 0 3 8 】

なお、決定部 5 1 は、特定部 5 0 により特定された試行パターンの各コア数でイタレーション数 N の処理を並列に実行して性能を求め、処理性能の最も高いコア数を並列処理を実行するコア数と決定してもよい。

【 0 0 3 9 】

プロセッサ制御部 5 2 は、プロセッサ 2 0 を制御して決定部 5 1 で決定されたコア数で

50

並列処理を実行させる。

【 0 0 4 0 】

ここで、決定部 5 1 により決定されるコア数の評価結果の一例を説明する。図 8 は、評価結果の一例を示す図である。なお、図 8 は、コア数に対する性能が図 7 に示す関係である場合の評価結果である。試行回数は、決定部 5 1 が並列処理を実行するコア数を決定するまでに性能の評価を行った回数である。精度は、処理性能の最も高い性能を決定できたかの予測精度を示し、決定されたコア数での処理性能を最大性能で除算した値である。

【 0 0 4 1 】

図 8 の「ヒルクライム」は、決定部 5 1 が試行パターンの各コア数のなかから、ヒルクライム法により、並列処理を実行するコア数と決定した場合の評価結果であり、試行回数が 2、精度が 0.98 である。図 8 の「全探索」は、決定部 5 1 が試行パターンの各コア数の性能を求め、処理性能の最も高いコア数を並列処理を実行するコア数と決定した場合の評価結果であり、試行回数が 28、精度が 0.98 である。

【 0 0 4 2 】

また、図 8 には、試行パターンを特定しないで、並列処理を実行するコア数を従来のように決定した場合の評価結果も示している。図 8 の「従来のヒルクライム」は、図 7 に示す 1 ~ 60 の各コア数のなかから、ヒルクライム法により、並列処理を実行するコア数と決定した場合の評価結果であり、試行回数が 2、精度が 0.95 である。また、図 8 の「従来の全探索」は、図 7 に示す 1 ~ 60 の全てのコア数の性能を求め、処理性能の最も高いコア数を並列処理を実行するコア数と決定した場合の評価結果であり、試行回数が 60、精度が 1.00 である。

【 0 0 4 3 】

このように、並列処理最適化プログラム 3 2 は、試行パターンを特定して試行パターンの各コア数のなかから、並列処理を実行するコア数と決定することにより、ヒルクライム法を用いた場合でも、処理性能の高いコア数を高い精度で求めることができる。また、並列処理最適化プログラム 3 2 は、試行パターンを特定して試行パターンの各コア数のなかから、並列処理を実行するコア数と決定することにより、全探索を用いた場合でも、より短時間で処理性能の高いコア数を求めることができる。また、並列処理最適化プログラム 3 2 は、各コア 24 に同程度の処理を割り振る場合、少ない数で割り振れるコア数を特定するため、消費電力を少なく抑えることができる。すなわち、並列処理最適化プログラム 3 2 は、高効率で並列処理を実行可能なコア数を速やかに求めることができる。

【 0 0 4 4 】

[処理の流れ]

次に、本実施例に係る情報処理装置 10 が並列処理最適化プログラム 3 2 を実行した際に並列処理に適したコア数を決定する並列処理最適化処理の流れについて説明する。図 9 は、実施例 1 に係る並列処理最適化処理の手順の一例を示すフローチャートである。この並列処理最適化処理は、例えば、並列処理プログラム 3 1 が並列処理を実行するタイミングで実行される。

【 0 0 4 5 】

図 9 に示すように、特定部 50 は、試行パターンを特定する試行パターン特定処理を実行して、試行パターンを特定する (S 10)。この試行パターン特定処理の詳細は、後述する。

【 0 0 4 6 】

決定部 51 は、特定された試行パターンの各コア数のなかから、性能が未評価である最も多いコア数を特定する (S 11)。決定部 51 は、特定したコア数での並列処理の性能を計測する (S 12)。例えば、決定部 51 は、イタレーション数 N の処理を並列に実行して 1 イタレーション当たりの処理時間を計測する。

【 0 0 4 7 】

決定部 51 は、計測した性能が前回計測された性能よりも悪化しているか判定する (S 13)。例えば、決定部 51 は、計測した処理時間が前回計測された処理時間よりも長い

10

20

30

40

50

場合、性能が悪化していると判定する。性能が悪化していない場合（Ｓ１３否定）、上述のＳ１１の処理へ移行する。

【００４８】

一方、性能が悪化している場合（Ｓ１３肯定）、決定部５１は、前回計測したコア数を処理性能の最も高いコア数と決定する（Ｓ１４）。プロセッサ制御部５２は、決定されたコア数で並列処理が実行されるように制御し（Ｓ１５）、処理を終了する。

【００４９】

次に、試行パターン特定処理の流れについて説明する。図１０は、実施例１に係る試行パターン特定処理の手順の一例を示すフローチャートである。この試行パターン特定処理は、例えば、図９に示す並列処理最適化処理のＳ１０から実行される。

10

【００５０】

図１０に示すように、特定部５０は、並列処理プログラム３１から並列処理のイタレーション数 N を受け取る（Ｓ２０）。また、特定部５０は、変数 a を１に初期化する（Ｓ２１）。

【００５１】

特定部５０は、イタレーション数 N を変数 a で除算し、除算結果の値の少数点以下を切り上げた値 X を算出する（Ｓ２３）。特定部５０は、値 X が１以上であるか判定する（Ｓ２４）。値 X が１以上ではない場合（Ｓ２４否定）、処理を終了して呼び出し元の処理に戻る。この試行パターン特定処理で特定された値 X が、試行パターンのコア数となる。

【００５２】

20

一方、値 X が１以上である場合（Ｓ２４肯定）、特定部５０は、値 X がプロセッサ２０のコア数以下であるか否かを判定する（Ｓ２５）。値 X がプロセッサ２０のコア数以下ではない場合（Ｓ２５否定）、後述するＳ２８の処理へ移行する。

【００５３】

一方、値 X がプロセッサ２０のコア数以下である場合（Ｓ２５肯定）、特定部５０は、試行パターンに値 X が既にあるか否かを判定する（Ｓ２６）。試行パターンに値 X が既にある場合（Ｓ２６肯定）、後述するＳ２８の処理へ移行する。

【００５４】

一方、試行パターンに値 X がない場合（Ｓ２６否定）、特定部５０は、試行パターンに値 X を試行するコア数として加える（Ｓ２７）。そして、特定部５０は、変数 a に１加算し、Ｓ２３の処理へ移行する。

30

【００５５】

[効果]

上述してきたように、本実施例に係る情報処理装置１０は、並列処理を実行可能なコア２４の数の範囲内で、並列して実行する所定数の処理を、割り振る処理数を変えて同程度ずつコアに割り振る場合に、処理が実行されるコア数を特定する。そして、情報処理装置１０は、特定したコア数のなかから、処理性能の最も高いコア数を並列処理を実行するコア数と決定する。これにより、情報処理装置１０は、高効率で並列処理を実行可能なコア数を速やかに求めることができる。

【００５６】

40

また、情報処理装置１０は、特定したコア数のなかから、コア数の多い順に性能の評価を行って性能を順次比較し、少ないコア数の性能が多いコア数の性能を最初に下回った際の多い側のコア数を並列処理を実行するコア数と決定する。これにより、情報処理装置１０は、並列処理を実行するコア数をより速やかに求めることができる。

【実施例２】

【００５７】

次に、実施例２について説明する。実施例２に係る情報処理装置１０の構成は、実施例１と略同一であるため、同一部分については同一の符号を付し、主に異なる部分について説明する。

【００５８】

50

図 1 1 は、実施例 2 に係るプロセッサの構成を模式的に示した図である。実施例 2 に係るプロセッサ 2 0 は、例えば、SMT (Simultaneous Multithreading) 搭載しており、各コア 2 4 に並列に処理を実行可能な複数のスレッド 2 5 を有する。このスレッド 2 5 は、それぞれプログラムの処理を実行可能な機構であり、ハードウェアスレッドなどとも呼ばれる。各コア 2 4 内のスレッド 2 5 は、コア内のメモリなどの資源を共有して処理を実行する。なお、図 1 1 の例では、コア 2 4 内に 2 つのスレッド 2 5 を有する場合を示したが、スレッド 2 5 の数はこれに限定されるものではない。例えば、インテル社製の Xeon Phi Coprocessor 5110P は、コア 2 4 毎に 4 つのスレッド 2 5 が搭載されている。

【 0 0 5 9 】

ところで、並列処理プログラム 3 1 には、並列して処理を実行するスレッド 2 5 の数が多いほど高性能化を達成できるものがある。また、一部のコア 2 4 や一部のスレッド 2 5 を停止させる方が、高い性能が得られるプログラムもある。例えば、スレッド 2 5 は、コア内の資源を共有するため、動作させるスレッド数を制限した方が高い性能が得られる場合がある。

【 0 0 6 0 】

図 1 2 A は、NPB EP のコア数毎の性能の評価結果の一例を示す図である。図 1 2 A の横軸は、並列処理を実行させたコア数を示す。図 1 2 A の縦軸は、動作させるスレッド数が 2、コア数が 1 での性能に対する相対性能を示す。図 1 2 A には、各コア 2 4 で 2 つのスレッド 2 5 を動作させた場合と、各コア 2 4 で 4 つのスレッド 2 5 を動作させた場合の性能が示されている。図 1 2 A に示すように、NPB EP では、各コア 2 4 で 4 つのスレッド 2 5 を動作させた場合の方が性能が高く、また、並列処理を行うコア数が多いほど高い性能が得られる。図 1 2 A の例では、スレッド数が 4 でコア数が 6 0 の場合に最も高い性能が得られている。

【 0 0 6 1 】

図 1 2 B は、NPB FT のコア数毎の性能の評価結果の一例を示す図である。図 1 2 B の横軸は、並列処理を実行させたコア数を示す。図 1 2 B の縦軸は、動作させるスレッド数が 2、コア数が 1 での性能に対する相対性能を示す。図 1 2 B には、各コア 2 4 で 2 つのスレッド 2 5 を動作させた場合と、各コア 2 4 で 4 つのスレッド 2 5 を動作させた場合の性能が示されている。図 1 2 B に示すように、NPB FT では、各コア 2 4 で動作させるスレッド数が 2、4 の場合で共に、コア数に比例して一律に性能が高くなり、特定のコア数で性能が飽和して低下する。また、図 1 2 B の例では、各コア 2 4 で動作させるスレッド数が 2 の場合の方が高い性能が得られている。動作させるスレッド数を制限した方が高い性能が得られるプログラムもある。

【 0 0 6 2 】

そこで、実施例 2 にかかる特定部 5 0 は、コア 2 4 の並列処理を実行可能なスレッド数および並列処理を実行可能なコア数の範囲内で、処理が実行されるスレッド数およびコア数を特定する。例えば、特定部 5 0 は、各コア 2 4 で動作させるスレッド数を変えて、イタレーション数の処理を、動作させるスレッド 2 5 に同程度ずつ割り振る場合に処理が実行されるスレッド数およびコア数を特定する。例えば、特定部 5 0 は、イタレーション数 N を 256 とし、プロセッサ 2 0 のコア数を 60 とし、各コア 2 4 で動作させるスレッド数を b とし、スレッド 2 5 に割り振る処理数を a とした場合、処理が実行されるスレッド数およびコア数を次のように特定する。

【 0 0 6 3 】

特定部 5 0 は、動作させるスレッド数 b を 1 から順に 1 つずつ変化させ、スレッド数 b 毎に、割り振る処理数 a を 1 から順に 1 つずつ変化させる。そして、特定部 5 0 は、割り振る処理数 a とスレッド数 b を乗算した値でイタレーション数 N を除算し、除算した値の少数点以下を切り上げた値 X を、値 X が 1 以上である間それぞれ求める。これにより、例えば、スレッド数 $b = 4$ とすると、 $a = 1$ の場合は、 $X = 64$ 、 $a = 2$ の場合は、 $X = 32 \cdots$ とそれぞれ求まる。また、スレッド数 $b = 3$ とすると、 $a = 1$ の場合は、 $X = 86$ 、 $a = 2$ の場合は、 $X = 43 \cdots$ とそれぞれ求まる。この値 X は、コア 2 4 でスレ

10

20

30

40

50

ド数 b のスレッド 25 を動作させ、各スレッド 25 に a 個程度ずつ処理を割り振る場合に、並列処理が実行されるコア数である。

【0064】

そして、特定部 50 は、値 X がプロセッサ 20 のコア数以下の範囲内で、値 X を特定する。例えば、プロセッサ 20 のコア数が 60 個である場合、スレッド数 b 毎のコア数は、以下のように求まる。以下、特定部 50 により特定されたスレッド数毎の各値 X を試行パターンと呼ぶ。

【0065】

スレッド数 $b = 1$: $X = 52, 43, 37, 32, 29, \dots, 1$

スレッド数 $b = 2$: $X = 43, 32, 26, 22, 19, \dots, 1$

スレッド数 $b = 3$: $X = 43, 29, 22, 18, 15, \dots, 1$

スレッド数 $b = 4$: $X = 32, 22, 16, 13, 11, \dots, 1$

【0066】

図 13 は、NPB FT のコア数毎の性能の評価結果の一例を示す図である。図 13 の例は、各コア 24 で動作させるスレッド数を 1 ~ 4 として、それぞれ並列処理を実行するコア数を 1 ~ 60 の範囲で変えて同程度ずつコア 24 に処理を割り振った場合の並列処理を実行したコア数毎の性能の一例である。図 13 の横軸は、並列処理を実行させたコア数を示す。図 13 の縦軸は、動作させるスレッド数が 1、コア数が 1 での性能に対する相対性能を示す。図 13 に示すように、イタレーション数 N がコア数に対してあまり大きくない場合、コア 24 あたりの処理数の違いにより処理性能が階段状になる。

【0067】

決定部 51 は、特定部 50 により特定されたスレッド数およびコア数のなかから、処理性能の最も高いスレッド数およびコア数を並列処理を実行するスレッド数およびコア数と決定する。一例として、決定部 51 は、試行パターンのスレッド数およびコア数のなかから、スレッド数毎に、ヒルクライム法により、最も性能の高いコア数を求める。そして、決定部 51 は、スレッド数毎の最も性能の高いコア数での性能を比較し、性能が最も高いコア数を並列処理を実行するコア数と決定する。例えば、決定部 51 は、試行パターンのスレッド数およびコア数をスレッド数が多いものから、スレッド数毎に、コア数の多い順に、コア数のコア 24 を用いてイタレーション数 N の処理を並列に実行して性能の評価を行う。そして、決定部 51 は、スレッド数毎に、コア数の多い順に性能を順次比較し、少ないコア数の性能が多いコア数の性能を最初に下回った際の多い側のコア数を当該スレッド数の最も良い性能のコア数と特定する。そして、決定部 51 は、少ないスレッド数の最も良い性能が多いスレッド数の最も良い性能を最初に下回った際の多い側のスレッド数およびコア数を、並列処理を実行するスレッド数およびコア数と決定する。

【0068】

例えば、図 13 の例では、決定部 51 は、スレッド数が 4 の場合について、32 個のコア 24 によりイタレーション数 N の処理を並列に実行して性能の評価を行う。次に、決定部 51 は、22 個のコア 24 によりイタレーション数 N の処理を並列に実行して性能の評価を行う。そして、決定部 51 は、コア数が 32 個の場合の性能と、コア数が 22 個の場合の性能を比較する。図 13 の例では、コア数が 22 個の場合の性能は、コア数が 32 個の場合の性能よりも下回っている。この場合、決定部 51 は、スレッド数が 4 の場合について、最も良い性能のコア数を 32 個と決定する。

【0069】

決定部 51 は、スレッド数が 3 の場合について、43 個のコア 24 によりイタレーション数 N の処理を並列に実行して性能の評価を行う。次に、決定部 51 は、29 個のコア 24 によりイタレーション数 N の処理を並列に実行して性能の評価を行う。そして、決定部 51 は、コア数が 43 個の場合の性能と、コア数が 29 個の場合の性能を比較する。図 13 の例では、コア数が 29 個の場合の性能は、コア数が 43 個の場合の性能よりも下回っている。この場合、決定部 51 は、スレッド数が 3 の場合について、最も良い性能のコア数を 43 個と決定する。

【 0 0 7 0 】

決定部 5 1 は、スレッド数が 4、コア数を 5 2 個の場合の性能と、スレッド数が 3、コア数を 4 3 個の場合の性能と比較する。図 1 3 の例では、スレッド数が 3、コア数を 4 3 個の場合の性能は、スレッド数が 4、コア数を 5 2 個の性能を下回っていない。

【 0 0 7 1 】

この場合、スレッド数が少ないとより性能が高くなる可能性がある。そこで、決定部 5 1 は、スレッド数が 2 の場合について、4 3 個のコア 2 4 によりイタレーション数 N の処理を並列に実行して性能の評価を行う。次に、決定部 5 1 は、3 2 個のコア 2 4 によりイタレーション数 N の処理を並列に実行して性能の評価を行う。そして、決定部 5 1 は、コア数が 4 3 個の場合の性能と、コア数が 3 2 個の場合の性能と比較する。図 1 3 の例では、コア数が 3 2 個の場合の性能は、コア数が 4 3 個の場合の性能よりも下回っている。この場合、決定部 5 1 は、スレッド数が 2 の場合について、最も良い性能のコア数を 4 3 個と決定する。

10

【 0 0 7 2 】

決定部 5 1 は、スレッド数が 3、コア数を 4 3 個の場合の性能と、スレッド数が 2、コア数を 4 3 個の場合の性能と比較する。図 1 3 の例では、スレッド数が 2、コア数を 4 3 個の場合の性能は、スレッド数が 3、コア数を 4 3 個の性能を下回っている。

【 0 0 7 3 】

この場合、決定部 5 1 は、スレッド数が 3、コア数が 4 3 を、並列処理を実行するスレッド数およびコア数と決定する。

20

【 0 0 7 4 】

なお、決定部 5 1 は、特定部 5 0 により特定された試行パターンの各スレッド数およびコア数でイタレーション数 N の処理を並列に実行して性能を求め、処理性能の最も高いコア数を並列処理を実行するコア数と決定してもよい。

【 0 0 7 5 】

ここで、決定部 5 1 により決定されるコア数の評価結果の一例を説明する。図 1 4 は、評価結果の一例を示す図である。なお、図 1 4 は、スレッド数およびコア数に対する性能が図 1 3 に示す関係である場合の評価結果である。試行回数は、決定部 5 1 が並列処理を実行するスレッド数およびコア数を決定するまでに性能の評価を行った回数である。精度は、処理性能の最も高い性能を決定できたかの予測精度を示し、決定されたスレッド数、コア数での処理性能を最大性能で除算した値である。

30

【 0 0 7 6 】

図 1 4 の「ヒルクライム」は、決定部 5 1 が試行パターンの各スレッド数およびコア数のなかから、ヒルクライム法により、並列処理を実行するコア数と決定した場合の評価結果であり、試行回数が 6、精度が 1 . 0 0 である。図 1 4 の「全探索」は、決定部 5 1 が試行パターンの各スレッド数およびコア数の性能を求め、処理性能の最も高いものを並列処理を実行するスレッド数およびコア数と決定した場合の評価結果であり、試行回数が 7 9、精度が 1 . 0 0 である。

【 0 0 7 7 】

また、図 1 4 には、試行パターンを特定しないで、並列処理を実行するコア数を従来のように決定した場合の評価結果も示している。図 1 4 の「従来のヒルクライム」は、スレッド数が 1 ~ 4 について、図 1 3 に示す 1 ~ 6 0 の各コア数のなかから、ヒルクライム法により、並列処理を実行するコア数と決定した場合の評価結果であり、試行回数が 4 4、精度が 0 . 9 9 である。また、図 1 3 の「従来の全探索」は、スレッド数が 1 ~ 4 について、図 1 3 に示す 1 ~ 6 0 の全てのコア数の性能を求め、処理性能の最も高いコア数を並列処理を実行するコア数と決定した場合の評価結果であり、試行回数が 2 4 0、精度が 1 . 0 0 である。

40

【 0 0 7 8 】

このように、並列処理最適化プログラム 3 2 は、試行パターンの各スレッド数およびコア数のなかから、並列処理を実行するスレッド数およびコア数を決定することにより、ヒ

50

ルクライム法を用いた場合でも、処理性能の高いコア数を高い精度で求めることができる。また、並列処理最適化プログラム 32 は、試行パターンの各スレッド数およびコア数のなかから、並列処理を実行するコア数と決定することにより、全探索を用いた場合でも、より短時間で処理性能の高いコア数を求めることができる。また、並列処理最適化プログラム 32 は、各コア 24 に同程度の処理を割り振る場合、少ない数で割り振れるコア数を特定するため、消費電力を少なく抑えることができる。すなわち、並列処理最適化プログラム 32 は、高効率で並列処理を実行可能なコア数を速やかに求めることができる。

【0079】

[処理の流れ]

次に、実施例 2 に係る情報処理装置 10 が並列処理最適化プログラム 32 を実行した際に並列処理に適したコア数を決定する並列処理最適化処理の流れについて説明する。図 15 は、実施例 2 に係る並列処理最適化処理の手順の一例を示すフローチャートである。実施例 2 に係る並列処理最適化処理の一部は、図 9 に示した実施例 1 に係る並列処理最適化処理と同一であるため、同一部分については同一の符号を付し、主に異なる処理について説明する。

10

【0080】

決定部 51 は、性能が未評価の最も大きいスレッド数を性能の評価対象のスレッド数とする (S15)。例えば、決定部 51 は、スレッド数が 1 ~ 4 の全てで性能が未評価の場合、最も大きい 4 を評価対象のスレッド数とする。決定部 51 は、この評価対象のスレッド数のコア数について S11 ~ S13 の処理を行う。この S11 ~ S13 の処理により、評価対象のスレッド数での最も良い性能のコア数が特定される。

20

【0081】

そして、決定部 51 は、評価対象のスレッド数での最も良い性能が、評価対象のスレッド数よりも 1 つ多いスレッド数での最も良い性能よりも悪化しているか判定する (S16)。性能が悪化していない場合 (S16 否定)、上述の S15 の処理へ移行する。

【0082】

一方、性能が悪化している場合 (S16 肯定)、決定部 51 は、評価対象のスレッド数よりも 1 つ多いスレッド数での最も良い性能のコア数を、並列処理を実行するスレッド数およびコア数と決定する (S17)。

【0083】

次に、実施例 2 に係る試行パターン特定処理の流れについて説明する。図 16 は、実施例 2 に係る試行パターン特定処理の手順の一例を示すフローチャートである。実施例 2 に係る試行パターン特定処理の一部は、図 10 に示した実施例 1 に係る試行パターン特定処理と同一であるため、同一部分については同一の符号を付し、主に異なる処理について説明する。

30

【0084】

特定部 50 は、変数 a、変数 b をそれぞれ 1 に初期化する (S30)。特定部 50 は、変数 a と変数 b を乗算した値でイタレーション数 N を除算し、除算結果の値の少数点以下を切り上げた値 X を算出する (S31)。

【0085】

値 X が 1 以上ではない場合 (S24 否定)、特定部 50 は、変数 b がコア 24 で動作可能な最大のスレッド数 S であるか否かを判定する (S32)。変数 b が最大のスレッド数 S である場合 (S32 肯定)、処理を終了する。

40

【0086】

一方、変数 b が最大のスレッド数 S ではない場合 (S32 否定)、特定部 50 は、変数 a を 1 に初期化し、変数 b に 1 を加算して (S33)、S31 の処理へ移行する。

【0087】

また、試行パターンに値 X がない場合 (S26 否定)、特定部 50 は、試行パターンに、スレッド数が変数 b の値の場合に試行するコア数として値 X を加える (S34)。すなわち、スレッド数が変数 b、コア数が値 X を試行パターンに加える。

50

【 0 0 8 8 】

[効果]

上述してきたように、本実施例に係る情報処理装置 10 は、コア 24 の並列処理を実行可能なスレッド数および並列処理を実行可能なコア数の範囲内で、処理が実行されるスレッド数およびコア数を特定する。そして、情報処理装置 10 は、特定したスレッド数およびコア数のなかから、処理性能の最も高いスレッド数およびコア数を並列処理を実行するスレッド数およびコア数と決定する。これにより、情報処理装置 10 は、高効率で並列処理を実行可能なスレッド数およびコア数を速やかに求めることができる。

【 0 0 8 9 】

また、情報処理装置 10 は、特定したスレッド数およびコア数をスレッド数が多いものから、スレッド数毎に、コア数の多い順に性能の評価を行って性能を順次比較する。情報処理装置 10 は、少ないコア数の性能が多いコア数の性能を最初に下回った際の多い側のコア数を当該スレッド数の最も良い性能のコア数と特定する。そして、情報処理装置 10 は、少ないスレッド数の最も良い性能が多いスレッド数の最も良い性能を最初に下回った際の多い側のスレッド数およびコア数を並列処理を実行するスレッド数およびコア数と決定する。これにより、情報処理装置 10 は、並列処理を実行するスレッド数およびコア数をより速やかに求めることができる。

【 実施例 3 】

【 0 0 9 0 】

次に、実施例 3 について説明する。実施例 3 に係る情報処理装置 10 の構成は、実施例 2 と略同一であるため、同一部分については同一の符号を付し、主に異なる部分について説明する。

【 0 0 9 1 】

実施例 3 に係る決定部 51 は、特定部 50 により特定されたスレッド数およびコア数のなかから、所定の予測モデルを用いて、並列処理を実行するスレッド数およびコア数と決定する。例えば、決定部 51 は、実行するスレッド数を変えて 1 つのコア 24 の各スレッドに 1 イタレーションを割り振って処理を実行して、各スレッド数での処理性能を計測する。例えば、決定部 51 は、実行中のメモリスループットを測定する。そして、決定部 51 は、測定されたメモリスループットをスレッド数で除算して、スレッド数毎に、1 スレッドあたりのメモリスループットを求める。決定部 51 は、1 スレッドあたりのメモリスループットが最も少ないスレッド数を並列処理を実行するスレッド数と決定する。

【 0 0 9 2 】

また、決定部 51 は、決定したスレッド数での 1 スレッドあたりのメモリスループットに決定したスレッド数を乗算してコアあたりのメモリスループットを求める。このメモリスループットは、コア数に比例して増加し、メモリスループットがピークに達するとこれ以上実行コア数を増やしても性能が向上しない。そこで、決定部 51 は、情報処理装置 10 において可能なピークのメモリスループットをコアあたりのメモリスループットで割り、ピークのメモリスループットに達するコア数を算出する。例えば、コアあたりのメモリスループットが 20 であり、ピークのメモリスループットが 100 である場合、決定部 51 は、 $100 / 20 = 5$ をピークのメモリスループットに達するコア数と算出する。

【 0 0 9 3 】

なお、決定部 51 は、メモリスループットに代えて、メモリのバンド幅を求めてもよい。例えば、コアあたりのバンド幅が 3 % であり、最大利用できるバンド幅が 100 % である場合、決定部 51 は、 $100 / 3 = 33.3$ を最大利用できるバンド幅に達するコア数と算出する。

【 0 0 9 4 】

そして、決定部 51 は、試行パターンの、並列処理を実行するスレッド数のコア数のなかから、ピークの性能となるコア数以内で、最大のコア数を並列処理を実行するコア数と特定する。例えば、並列処理を実行するスレッド数が 1 であり、試行パターンのスレッド数が 1 のコア数が 52、43、37、32、29、・・・1 であり、ピークの性能となる

10

20

30

40

50

コア数が 3 3 . 3 であるものとする。この場合、決定部 5 1 は、コア数が 3 3 . 3 以内で最大の 3 2 を並列処理を実行するコア数と特定する。

【 0 0 9 5 】

[処理の流れ]

次に、実施例 3 に係る情報処理装置 1 0 が並列処理最適化プログラム 3 2 を実行した際に並列処理に適したコア数を決定する並列処理最適化処理の流れについて説明する。図 1 7 は、実施例 3 に係る並列処理最適化処理の手順の一例を示すフローチャートである。実施例 3 に係る並列処理最適化処理の一部は、図 1 5 に示した実施例 2 に係る並列処理最適化処理と同一であるため、同一部分については同一の符号を付し、主に異なる処理について説明する。

10

【 0 0 9 6 】

決定部 5 1 は、実行するスレッド数を変えて 1 コア 2 4 のスレッドに 1 イタレーションを割り振って処理を実行して、各スレッド数での処理性能を計測する (S 5 0)。そして、決定部 5 1 は、各スレッド数での処理性能に基づき、並列処理を実行するスレッド数 s と、ピークの性能となるコア数 m を算出する (S 5 1)。

【 0 0 9 7 】

決定部 5 1 は、試行パターン of スレッド数 s の各コア数を少ない順に並べ、コア数の少ない順に T_1 、 T_2 、 $T_3 \cdots T_p$ とする (S 5 2)。決定部 5 1 は、変数 n を 0 に初期化する (S 5 3)。そして、決定部 5 1 は、変数 n に 1 を加算する (S 5 4)。

【 0 0 9 8 】

20

決定部 5 1 は、 $T_n < \text{コア数 } m < T_{n+1}$ を満たすか否かを判定する (S 5 5)。 $T_n < \text{コア数 } m < T_{n+1}$ を満たさない場合 (S 5 5 否定)、S 5 4 の処理へ移行する。

【 0 0 9 9 】

一方、 $T_n < \text{コア数 } m < T_{n+1}$ を満たす場合 (S 5 5 肯定)、決定部 5 1 は、 T_n のコア数を、並列処理を実行するコア数と決定する (S 5 6)。プロセッサ制御部 5 2 は、スレッド数 s および決定したコア数で、残りのイタレーションの処理が並列実行されるように制御し (S 5 7)、処理を終了する。

【 0 1 0 0 】

[効果]

上述してきたように、本実施例に係る情報処理装置 1 0 は、特定したスレッド数およびコア数のなかから、所定の予測モデルを用いて、並列処理を実行するスレッド数およびコア数を決定する。これにより、情報処理装置 1 0 は、並列処理を実行するスレッド数およびコア数をより速やかに求めることができる。

30

【 実施例 4 】

【 0 1 0 1 】

さて、これまで開示の装置に関する実施例について説明したが、開示の技術は上述した実施例以外にも、種々の異なる形態にて実施されてよいものである。そこで、以下では、本発明に含まれる他の実施例を説明する。

【 0 1 0 2 】

例えば、上記の実施例では、並列処理を実行する際に並列処理を実行するスレッド数、コア数を決定する場合について説明したが、開示のシステムはこれに限定されない。例えば、並列処理が定期的に行われるなど、実行される並列処理が事前にわかっている場合、並列処理の実行前にスレッド数、コア数を決定してもよい。

40

【 0 1 0 3 】

また、図示した各装置の各構成要素は機能概念的なものであり、必ずしも物理的に図示の如く構成されていることを要しない。すなわち、各装置の分散・統合の具体的状態は図示のものに限られず、その全部または一部を、各種の負荷や使用状況などに応じて、任意の単位で機能的または物理的に分散・統合して構成することができる。例えば、並列処理最適化プログラム 3 2 の特定部 5 0 と、決定部 5 1 と、プロセッサ制御部 5 2 の各処理部が適宜統合されてもよい。また、各処理部の処理が適宜複数の処理部の処理に分離されて

50

もよい。さらに、並列処理最適化プログラム 3 2 は複数のプログラムにより構成され、複数の装置に分散配置されてもよい。

【 0 1 0 4 】

なお、上記した並列処理最適化プログラム 3 2 については、必ずしも最初から D I S K 2 1 に記憶させることを要しない。

【 0 1 0 5 】

例えば、情報処理装置 1 0 に挿入されるフレキシブルディスク (F D)、C D - R O M、D V D ディスク、光磁気ディスク、I C カードなどの「可搬用の物理媒体」にプログラムを記憶させておく。そして、情報処理装置 1 0 がこれらからプログラムを読み出して実行するようにしてもよい。

10

【 0 1 0 6 】

さらには、公衆回線、インターネット、L A N、W A Nなどを介して情報処理装置 1 0 に接続される「他のコンピュータ (またはサーバ)」などにプログラムを記憶させておく。そして、情報処理装置 1 0 がこれらからプログラムを読み出して実行するようにしてもよい。

【 符号の説明 】

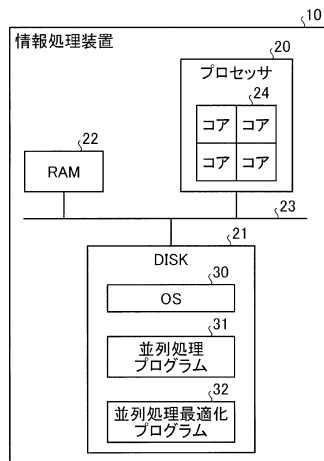
【 0 1 0 7 】

- 1 0 情報処理装置
- 2 0 プロセッサ
- 2 4 コア
- 2 5 スレッド
- 3 1 並列処理プログラム
- 3 2 並列処理最適化プログラム
- 5 0 特定部
- 5 1 決定部
- 5 2 プロセッサ制御部

20

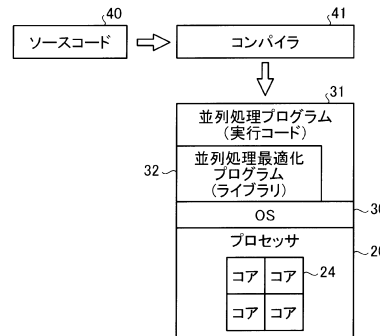
【図 1】

情報処理装置の概略的な構成の一例を示した図



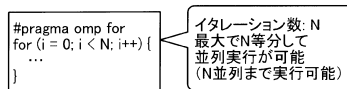
【図 2】

情報処理装置におけるハードウェアとソフトウェアとの関係を説明するための図



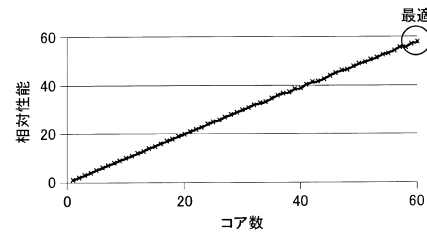
【図 3】

並列処理が可能な処理を示す図



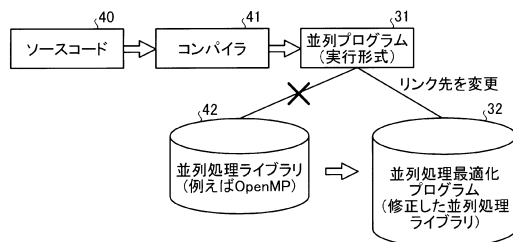
【図 5 A】

NPB EP のコア数毎の性能の評価結果の一例を示す図



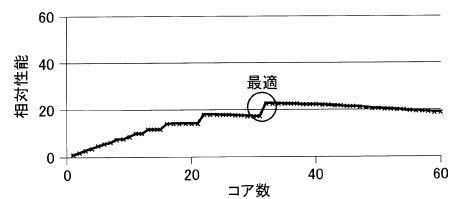
【図 4】

並列処理プログラムと並列処理最適化プログラムをスタティックリンクさせる一例を示す図



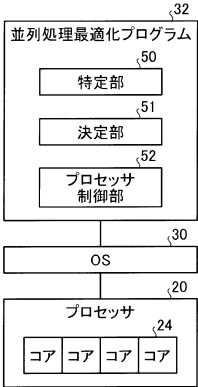
【図 5 B】

NPB FT のコア数毎の性能の評価結果の一例を示す図



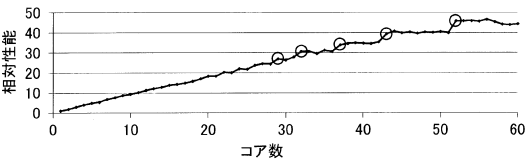
【図 6】

実行された並列処理最適化プログラムの機能的な構成を示す図



【図 7】

NPB FTのコア数毎の性能の評価結果の一例を示す図



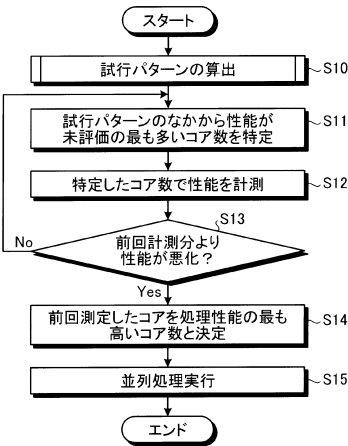
【図 8】

評価結果の一例を示す図

	全探索	ヒルクライム	従来の全探索	従来のヒルクライム
試行回数	28	2	60	2
精度	0.98	0.98	1.00	0.95

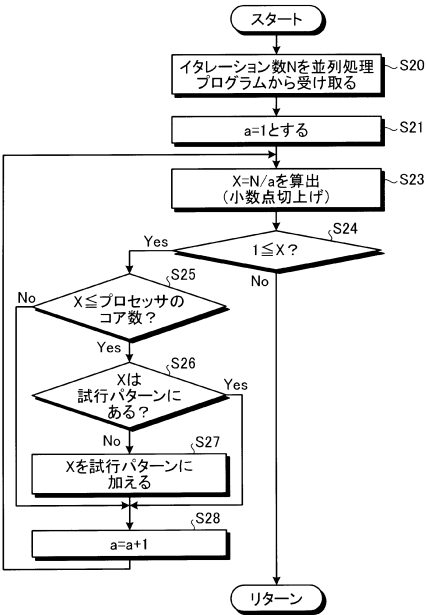
【図 9】

実施例1に係る並列処理最適化処理の手順の一例を示すフローチャート

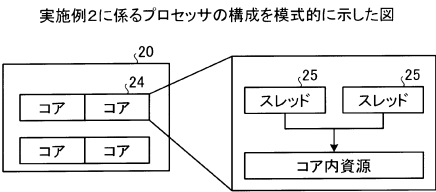


【図 10】

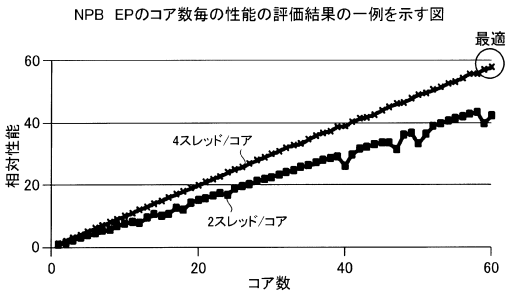
実施例1に係る試行パターン特定処理の手順の一例を示すフローチャート



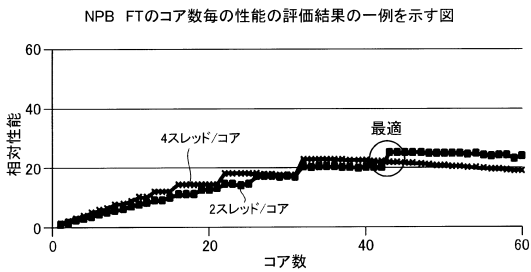
【図 1 1】



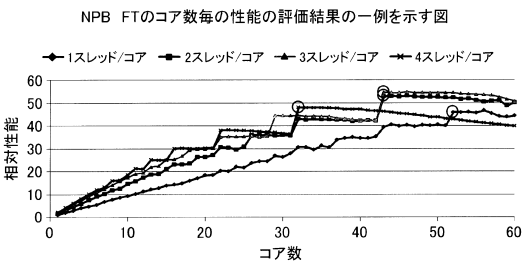
【図 1 2 A】



【図 1 2 B】



【図 1 3】



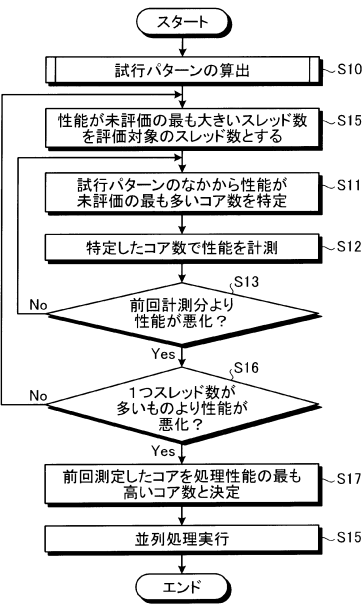
【図 1 4】

評価結果の一例を示す図

	全探索	ヒルクライム	従来の全探索	従来のヒルクライム
試行回数	79	6	240	44
精度	1.00	1.00	1.00	0.99

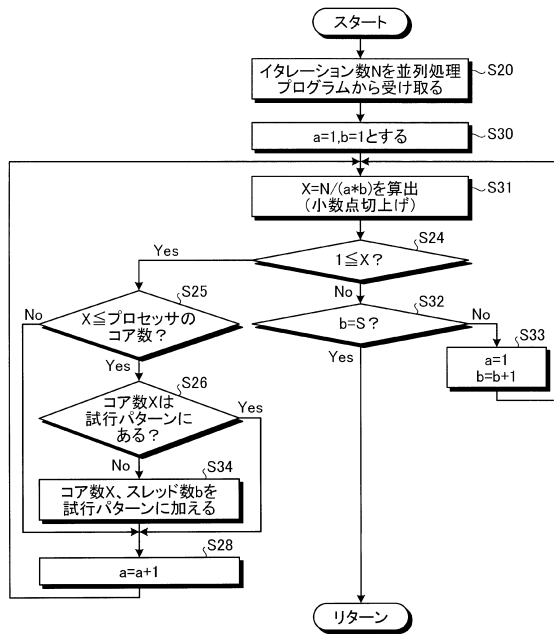
【図 1 5】

実施例2に係る並列処理最適化処理の手順の一例を示すフローチャート



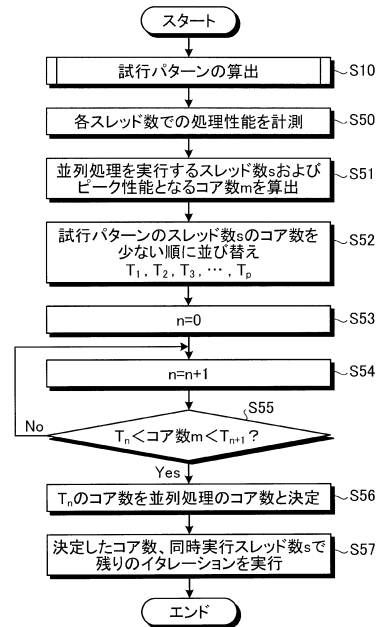
【図 16】

実施例2に係る試行パターン特定処理の手順の一例を示すフローチャート



【図 17】

実施例3に係る並列処理最適化処理の手順の一例を示すフローチャート



フロントページの続き

(56)参考文献 国際公開第2009/037731(WO,A1)

今村 智史, コア数と動作周波数の動的変更によるメニーコア・プロセッサ性能向上手法の提案, 情報処理学会論文誌 コンピューティングシステム, 社団法人情報処理学会, 2012年10月15日, Vol.5, No.4, pp.24-35

高畠 志泰, 細粒度並列アーキテクチャ用SISALコンパイラにおける並列粒度調整方式, 情報処理学会論文誌, 日本, 社団法人情報処理学会, 1998年 6月15日, 第39巻, 第6号, pp.1709-1717

(58)調査した分野(Int.Cl., DB名)

G06F 9/50