

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号  
特許第7419843号  
(P7419843)

(45)発行日 令和6年1月23日(2024.1.23)

(24)登録日 令和6年1月15日(2024.1.15)

(51)国際特許分類 F I  
G 0 6 F 9/50 (2006.01) G 0 6 F 9/50 1 5 0 C

請求項の数 3 (全15頁)

(21)出願番号	特願2020-14221(P2020-14221)	(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
(22)出願日	令和2年1月30日(2020.1.30)	(74)代理人	110002147 弁理士法人酒井国際特許事務所
(65)公開番号	特開2021-120824(P2021-120824 A)	(72)発明者	未安 直樹 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
(43)公開日	令和3年8月19日(2021.8.19)	(72)発明者	一瀬 克己 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
審査請求日	令和4年10月6日(2022.10.6)	審査官	坂東 博司

最終頁に続く

(54)【発明の名称】 並列処理装置

## (57)【特許請求の範囲】

## 【請求項1】

所定の並列化パラメータを設定した、2重ループを有する最適化対象のプログラムを並列処理で1回だけ実行する実行部と、

前記実行部によって実行された結果から前記最適化対象のプログラムの特徴値を算出する算出部と、

2重ループを有するサンプルプログラムにおける特徴値と最適な並列化パラメータとの相関関係を示すデータベースを参照して、前記算出部によって算出された特徴値に対応する最適な並列化パラメータを抽出する抽出部と、

を有することを特徴とする並列処理装置。

10

## 【請求項2】

特徴値を持つサンプルプログラムに対して複数通りの並列化パラメータをそれぞれ設定して実行した際の結果を基に当該プログラムにおける特徴値と最適な並列化パラメータとの相関関係を示すデータベースを生成する生成部とをさらに有し、

前記抽出部は、前記生成部によって生成されたデータベースを参照して、前記算出部によって算出された前記最適化対象のプログラムの特徴値に対応する最適な並列化パラメータを抽出する

ことを特徴とする請求項1に記載の並列処理装置。

## 【請求項3】

前記相関関係は、ループごとの、実行命令数の平均、実行命令数の相対標準偏差および

20

キャッシュの再利用率の平均を示す複数の特徴値と、ループごとの、プロセッサコアの割り当て、並列化のスケジューリング方式および並列化のスケジューリング方式のチャンク数を示す複数の並列化パラメータとの相関関係である

ことを特徴とする請求項 1 または請求項 2 に記載の並列処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、並列処理装置に関する。

【背景技術】

【0002】

並列処理プログラムの並列処理パラメータを決定する技術が開示されている。例えば、特許文献 1 では、処理部が、評価対象の第 1 プログラムの推奨するノード数とプロセス数とを決定する。具体的には、処理部は、ノード数とプロセス数とを示す第 1 サンプル点に従って第 1 プログラムを実行させ、第 1 サンプル点それぞれの間の評価値の変動率に基づく第 1 統計量を算出する。そして、処理部は、第 1 統計量が第 1 閾値以下になるまで第 1 サンプル点 4 を生成する。また、処理部は、第 1 サンプル点から所定距離内に第 2 サンプル点を追加し、第 2 サンプル点に従って第 1 プログラムを実行したときの評価値を用いて第 1 サンプル点ごとの第 2 統計量を算出する。そして、処理部は、第 2 統計量が第 2 閾値以下になるまで第 2 サンプル点を生成する。そして、処理部は、第 1 サンプル点と第 2 サンプル点との評価値を補完して、推奨するノード数とプロセス数とを決定する。

【0003】

また、2 重ループを有するプログラムを並列処理で実行するとき、適切な並列処理パラメータを設定する必要がある。スレッド並列化規格 OpenMP の並列処理パラメータとして、スケジューリング方式、チャンクサイズおよびプロセッサコアの割り当てが挙げられる。従来では、並列処理パラメータの組み合わせごとに、組み合わせのパラメータをプログラムに設定し、プログラムを実行し、実行結果を参照して適切な並列化パラメータとなるように調整していた。あるいは、並列化パラメータを経験者による経験で調整していた。

【先行技術文献】

【特許文献】

【0004】

【文献】特開 2018 - 120387 号公報

【文献】特開 2016 - 9972 号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

しかしながら、ループを有するプログラムを並列処理で実行するとき、プログラムに設定する最適な並列処理パラメータを高速に推定することが難しいという問題がある。

【0006】

例えば、推奨するノード数とプロセス数とを決定する技術では、評価対象の第 1 プログラムをノード数とプロセス数とを示す第 1 サンプル点に従って評価対象の第 1 プログラムを何度も実行させて、推奨するノード数とプロセス数とを決定する。また、決定する並列処理パラメータは、ノード数とプロセス数だけである。したがって、スケジューリング方式、チャンクサイズおよびプロセッサコアの割り当てを含む最適な並列処理パラメータを高速に推定することが難しい。

【0007】

また、2 重ループを有するプログラムを並列処理で実行するときの並列化パラメータを決定する場合は、並列処理パラメータの組み合わせごとに、プログラムを実行して、最適な並列処理パラメータを求める。したがって、かかる場合でも、最適な並列処理パラメータを高速に推定することが難しい。

10

20

30

40

50

## 【 0 0 0 8 】

本発明は、1つの側面では、ループを有するプログラムを並列処理で実行するときの最適な並列処理パラメータを高速に推定することを目的とする。

## 【課題を解決するための手段】

## 【 0 0 0 9 】

1つの態様では、並列処理装置は、所定の並列化パラメータを設定した、2重ループを有する最適化対象のプログラムを並列処理で1回だけ実行する実行部と、前記実行部によって実行された結果から前記最適化対象のプログラムの特徴値を算出する算出部と、2重ループを有するサンプルプログラムにおける特徴値と最適な並列化パラメータとの相関関係を示すデータベースを参照して、前記算出部によって算出された特徴値に対応する最適な並列化パラメータを抽出する抽出部と、を有する。

10

## 【発明の効果】

## 【 0 0 1 0 】

1実施態様によれば、ループを有するプログラムを並列処理で実行するときの最適な並列処理パラメータを高速に推定することができる。

## 【図面の簡単な説明】

## 【 0 0 1 1 】

【図1】図1は、実施例に係る並列処理装置の機能構成を示すブロック図である。

【図2】図2は、プログラムの一例を示す図である。

【図3】図3は、実施例に係る関係データベースの一例を示す図である。

20

【図4】図4は、プログラムを実行するハードウェアの一例を示す図である。

【図5】図5は、プロセッサコアの割り当ての一例を示す図である。

【図6】図6は、DOLoopの実行例を示す図である。

【図7】図7は、最適化パラメータ抽出の一例を示す図である。

【図8】図8は、最適化されたプログラムの一例を示す図である。

【図9】図9は、実施例に係る生成処理のフローチャートの一例を示す図である。

【図10】図10は、実施例に係る最適化パラメータ抽出処理のフローチャートの一例を示す図である。

【図11】図11は、実施例に係るプログラム本実行処理のフローチャートの一例を示す図である。

30

【図12】図12は、並列処理プログラムを実行するコンピュータの一例を示す図である。

## 【発明を実施するための形態】

## 【 0 0 1 2 】

以下に、本願の開示する並列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、本発明は、実施例により限定されるものではない。

## 【実施例】

## 【 0 0 1 3 】

## [実施例に係る並列処理装置の機能構成]

図1は、実施例に係る並列処理装置の機能構成を示すブロック図である。図1に示す並列処理装置1は、複数レベルの並列化が行なえるアプリケーションプログラムに対する最適な並列化パラメータを選択して、実行性能をチューニングする。実施例では、複数レベルの並列化を行なえるアプリケーションプログラムとしてスレッド並列化規格であるOpenMPを適用するものとする。また、実施例では、例えば、2重ループを有するアプリケーションプログラムを並列処理で実行する際に適切な並列化パラメータを選択する場合について説明する。なお、OpenMP規格は、例えば、「OpenMP Application Program Interface Version 5.0」の記載に準じるものであり、詳細な説明を省略する。

40

## 【 0 0 1 4 】

並列処理装置1は、制御部10および記憶部20を有する。制御部10は、CPU(Central Processing Unit)などの電子回路に対応する。そして、制御部10は、各種の処理手順を規定したプログラムや制御データを格納するための内部メモリを有し、これら

50

によって種々の処理を実行する。制御部 10 は、生成部 11、プログラム仮実行部 12、特徴値算出部 13、最適化パラメータ抽出部 14 およびプログラム本実行部 15 を有する。なお、プログラム仮実行部 12 は、実行部の一例である。特徴値算出部 13 は、算出部の一例である。最適化パラメータ抽出部 14 は、抽出部の一例である。生成部 11 は、生成部の一例である。

#### 【0015】

記憶部 20 は、例えば、RAM (Random Access Memory)、フラッシュメモリ (Flash Memory) などの半導体メモリ素子、または、ハードディスク、光ディスクなどの記憶装置である。記憶部 20 は、基準プログラム 21 および関係データベース 22 を有する。

10

#### 【0016】

基準プログラム 21 は、並列化パラメータに影響を与える、プログラムの複数の実行特徴値の組み合わせごとに、それぞれの組み合わせの特徴値を持つプログラムのことをいう。基準プログラム 21 は、並列化パラメータを設定する変数を有するとともに、2重ループを有する。ここでは、プログラムの複数の実行特徴値の組み合わせを「アプリ実行特徴値セット」というものとする。アプリ実行特徴値セットごとに、基準プログラム 21 が作成される。

#### 【0017】

ここで、実施例で扱うプログラムの一例を、図 2 を参照して説明する。図 2 は、プログラムの一例を示す図である。図 2 では、Fortran 言語のプログラムの一例である。図 2 に示すように、プログラムは、2レベルのDOLープの並列化を行う。プログラム内の L1, U1 が、1次元目のループの変動範囲の定数である。L2, U2 が、2次元目のループの変動範囲の定数である。

20

#### 【0018】

かかるプログラムは、スレッド並列化規格である OpenMP の並列化パラメータを設定する変数を有する。並列化パラメータには、変数 P1, P2, D1, D2, C1 および C2 が挙げられる。変数 P1, P2 には、DOLープへのプロセッサコアの割り当てを示す値が設定される。変数 D1, D2 は、スケジューリング方式を示す値が設定される。変数 C1, C2 は、チャンク数が設定される。スケジューリング方式には、例えば、static、dynamic、guided が設定可能である。チャンクには、例えば、任意の整数が設定可能である。任意の整数は、例えば、1, 8 または 32 である。DOLープへのプロセッサコアの割り当てとして、例えば 48 コアを例とすると、(1, 48), (2, 24), (3, 16), (4, 12), (6, 8), (8, 6), (12, 4), (16, 3), (24, 2), (48, 1) が設定可能である。

30

#### 【0019】

かかるプログラムは、parallel do 構文を持ち、変動範囲の do ループを並列に実行する。ここでいう並列とは、プロセス並列ではなく、プログラム内のスレッド並列のことをいう。parallel do 構文は、プロセッサコアの割り当てに従ってスレッドに do ループの繰り返し処理の一部を割り当てる。

#### 【0020】

40

かかるプログラムに設定可能な並列化パラメータの組み合わせは、上記の例では、810 通り存在する。すなわち、スケジューリング方式に 3 種類、チャンクに 3 種類が存在するので、2レベルのDOLープに適用する場合には、81 (= 3 × 3 × 3 × 3) 通りが存在する。加えて、DOLープへのプロセッサコアの割り当てに 10 種類が存在するので、プログラムに設定可能な並列化パラメータの組み合わせは、810 (= 10 × 81) 通りとなる。

#### 【0021】

実施例では、並列処理装置 1 は、プログラムに設定可能な並列化パラメータの組み合わせの中から実行時間が最も短い組み合わせを示す最適並列化パラメータを抽出する。

#### 【0022】

50

図 1 に戻って、ここで、アプリ実行特徴値セットについて説明する。アプリ実行特徴値セットは、並列化パラメータに影響を与える、プログラムの複数の実行特徴値の組み合わせのことをいう。並列化パラメータに影響を与える、プログラムの複数の実行特徴値には、例えば、以下の  $x$ 、 $y$ 、 $z$  が考えられる。

$x$ ) 各 D O ループの繰り返しごとの処理部分の実行命令数の平均値

$y$ ) 各 D O ループの繰り返しごとの処理部分の実行命令数の相対標準偏差値

$z$ ) 各 D O ループの繰り返しごとの処理部分のキャッシュ再利用率の平均値

【 0 0 2 3 】

$x$  が並列化パラメータに影響を与える、プログラムの実行特徴値として適用される理由は、以下の通りである。処理部分の実行命令数の大小により、例えば並列化パラメータの 1 つであるスケジューリング方式に影響を与えるからである。すなわち、処理コスト（オーバーヘッド）が大きい場合に採用される D y n a m i c スケジューリング方式を採用できるかどうかを判定するためである。

10

【 0 0 2 4 】

$y$  が並列化パラメータに影響を与える、プログラムの実行特徴値として適用される理由は、以下の通りである。処理部分の負荷バランスの度合いにより、例えば並列化パラメータの 1 つであるスケジューリング方式に影響を与えるからである。すなわち、処理部分の負荷バランスにバラツキが無い場合には s t a t i c スケジューリング方式を、負荷バランスにバラツキが有る場合には D y n a m i c スケジューリング方式を採用するためである。

20

【 0 0 2 5 】

$z$  が並列化パラメータに影響を与える、プログラムの実行特徴値として適用される理由は、以下の通りである。キャッシュの再利用率の高低により、例えば並列化パラメータの 1 つであるチャンク数に影響を与えるからである。すなわち、キャッシュの再利用率が高い場合には、チャンク数を大きくすることにより、キャッシュの再利用を促進すべく、大きいチャンク数を採用するためである。

【 0 0 2 6 】

実施例では、並列処理装置 1 は、この 3 種類の実行特徴値を、1 次元目の D O ループおよび 2 次元目の D O ループでそれぞれ算出し、6 個の実行特徴値の組み合わせを、アプリ実行特徴値セットとして定義する。

30

【 0 0 2 7 】

関係データベース 2 2 は、アプリ実行特徴値セットと、最適な並列化パラメータとの対応関係を保持する。関係データベース 2 2 は、後述する生成部 1 1 によって生成される。なお、関係データベース 2 2 の一例は、後述する。

【 0 0 2 8 】

生成部 1 1 は、関係データベース 2 2 を生成する。

【 0 0 2 9 】

例えば、生成部 1 1 は、アプリ実行特徴値セットの各要素を、次の刻みで変化させるような複数の基準プログラム 2 1 を生成する。各 D O ループの繰り返しごとの処理部分の実行命令数の平均値は、D y n a m i c スケジューリング方式の場合の実行時ライブラリの実行命令数の 1 ~ 1 0 倍とする。すなわち、各 D O ループの繰り返しごとの処理部分の実行命令数の平均値は、D O ループごとに、1, 2, ..., 10 の値を採る。各 D O ループの繰り返しごとの処理部分の実行命令数の相対標準偏差値は、0.1 ~ 1.0 を 0.1 刻みとする。すなわち、各 D O ループの繰り返しごとの処理部分の実行命令数の相対標準偏差値は、D O ループごとに、0.1, 0.2, ..., 1.0 の値を採る。各 D O ループの繰り返しごとの処理部分のキャッシュ再利用率の平均値は、0.1 ~ 1.0 を 0.1 刻みとする。すなわち、各 D O ループの繰り返しごとの処理部分のキャッシュ再利用率の平均値は、D O ループごとに、0.1, 0.2, ..., 1.0 の値を採る。この例では、アプリ実行特徴値セットは、最終的に、 $10^6$  (= 100 万) 個になる。生成部 1 1 は、100 万個のアプリ実行特徴値セットを持つそれぞれの基準プログラム 2 1 を生成する。

40

50

## 【 0 0 3 0 】

そして、生成部 1 1 は、1つのアプリ実行特徴値セットを持つ基準プログラム 2 1 に対して複数通りの並列化パラメータの組をそれぞれ設定して実行する。一例として、プログラムに設定可能な並列化パラメータの組み合わせが 8 1 0 通りあるとすると、生成部 1 1 は、1つの基準プログラム 2 1 に対して 8 1 0 通りの並列化パラメータをそれぞれ設定して実行する。

## 【 0 0 3 1 】

そして、生成部 1 1 は、複数通りの実行結果を基に、最も実行時間が短かった並列化パラメータの組を最適並列化パラメータとして決定する。そして、生成部 1 1 は、基準プログラム 2 1 におけるアプリ実行特徴値セットと最適並列化パラメータとを対応付けて関係データベース 2 2 に追加する。そして、生成部 1 1 は、他のアプリ実行特徴値セットについても同様に、アプリ実行特徴値セットを持つ基準プログラム 2 1 における最適並列化パラメータを決定して、関係データベース 2 2 に追加する。

10

## 【 0 0 3 2 】

ここで、関係データベース 2 2 の一例を、図 3 を参照して説明する。図 3 は、実施例に係る関係データベースの一例を示す図である。図 3 に示す関係データベース 2 2 は、アプリ実行特徴値セット ( X 1 , Y 1 , Z 1 , X 2 , Y 2 , Z 2 ) と、最適並列化パラメータ ( P 1 , P 2 , D 1 , D 2 , C 1 , C 2 ) とを対応付けて記憶する。

## 【 0 0 3 3 】

アプリ実行特徴値セット ( X 1 , Y 1 , Z 1 , X 2 , Y 2 , Z 2 ) は、1次元目の D O ループの上記 x、y、z および 2次元目の D O ループの上記 x、y、z に対応する。つまり、X 1 は、1次元目の D O ループの x、すなわち、1次元目の D O ループの繰り返しごとの処理部分の実行命令数の平均値を示す。Y 1 は、1次元目の D O ループの y、すなわち、1次元目の D O ループの繰り返しごとの処理部分の実行命令数の相対標準偏差値を示す。Z 1 は、1次元目の D O ループの z、すなわち、1次元目の D O ループの繰り返しごとの処理部分のキャッシュ再利用率の平均値を示す。X 2 は、2次元目の D O ループの x、すなわち、2次元目の D O ループの繰り返しごとの処理部分の実行命令数の平均値を示す。Y 2 は、2次元目の D O ループの y、すなわち、2次元目の D O ループの繰り返しごとの処理部分の実行命令数の相対標準偏差値を示す。Z 2 は、2次元目の D O ループの z、すなわち、2次元目の D O ループの繰り返しごとの処理部分のキャッシュ再利用率の平均値を示す。ここでは、X 1 や X 2 は、1, 2, …, 10 の値を採る。Y 1 や Y 2 は、0.1, 0.2, …, 1.0 の値を採る。Z 1 や Z 2 は、0.1, 0.2, …, 1.0 の値を採る。

20

30

## 【 0 0 3 4 】

最適並列化パラメータ ( P 1 , P 2 , D 1 , D 2 , C 1 , C 2 ) は、アプリ実行特徴値セット ( X 1 , Y 1 , Z 1 , X 2 , Y 2 , Z 2 ) を持つ基準プログラム 2 1 における最適な並列化パラメータに対応する。P 1 , P 2 は、D O ループへのプロセッサコアの割り当てを示す値である。D 1 は、1次元目の D O ループのスケジューリング方式を示す値である。D 2 は、2次元目の D O ループのスケジューリング方式を示す値である。C 1 は、1次元目の D O ループのチャンク数である。C 2 は、2次元目の D O ループのチャンク数である。

40

## 【 0 0 3 5 】

一例として、アプリ実行特徴値セットが ( 1 , 0 . 1 , 0 . 1 , 1 , 0 . 1 , 0 . 1 ) である場合に、最適並列化パラメータとして ( 6 , 8 , s t a t i c , s t a t i c , 1 , 1 ) が記憶されている。また、アプリ実行特徴値セットが ( 1 0 , 0 . 1 , 0 . 1 , 1 , 0 . 1 , 0 . 1 ) である場合に、最適並列化パラメータとして ( 6 , 8 , d y n a m i c , s t a t i c , 1 , 1 ) が記憶されている。

## 【 0 0 3 6 】

プログラム仮実行部 1 2 は、最適化パラメータを抽出したい最適化対象プログラムを仮実行する。例えば、プログラム仮実行部 1 2 は、最適化対象プログラムの特徴を求めるた

50

めに、所定の並列化パラメータを設定した最適化対象プログラムを1回だけ仮実行する。最適化対象プログラムに設定する所定の並列化パラメータ(P1, P2, D1, D2, C1, C2)は、例えば(6, 8, static, static, 1, 1)である。しかしながら、所定の並列化パラメータ(P1, P2, D1, D2, C1, C2)は、これに限定されるものではない。

#### 【0037】

特徴値算出部13は、プログラム仮実行部12によって実行された際のアプリ実行特徴値セットのそれぞれの特徴値を算出する。実行時のアプリ実行特徴値セットのそれぞれの特定値は、特定のプロファイル等の性能情報取得ツールを用いて求められれば良い。特定のプロファイル等の性能情報取得ツールは、例えば、スーパーコンピュータ「京」に向けたアプリケーション開発支援ツール(“スーパーコンピュータ「京」の性能プロファイル”を参照)を利用しても良い。

10

#### 【0038】

最適化パラメータ抽出部14は、関係データベース22を参照して、特徴値算出部13によって算出されたアプリ実行特徴値セットに対応する最適並列化パラメータを抽出する。例えば、最適化パラメータ抽出部14は、関係データベース22から、特徴値算出部13によって算出されたアプリ実行特徴値セットと最も距離が近いアプリ実行特徴値セットを取得する。これは、関係データベース22に記憶されたアプリ実行特徴値セットは、離散的であるからである。そして、最適化パラメータ抽出部14は、関係データベース22から、取得したアプリ実行特徴値セットに対応する最適並列化パラメータを抽出する。これにより、最適化パラメータ抽出部14は、ループを有する最適化対象プログラムを並列処理で実行するときの最適な並列化パラメータを高速に推定することができる。

20

#### 【0039】

プログラム本実行部15は、最適化対象プログラムを本実行する。例えば、プログラム本実行部15は、最適化パラメータ抽出部14によって抽出された最適並列化パラメータを設定した最適化対象プログラムを本実行する。

#### 【0040】

##### [プログラムを実行するハードウェアの一例]

図4は、プログラムを実行するハードウェアの一例を示す図である、図4に示すように、計算機が、1つのメモリを共有する48個のプロセッサコア(計算コア)を有する場合とする。それぞれの計算コアは、メモリへのアクセスのコストを軽減するために、キャッシュを有する。そして、12個の計算コアごとに、UMAグループを生成する。そして、4組のUMAグループが、NUMA(Non-Uniform Memory Access)結合している。なお、図4の例では、計算機に48個の計算コアがある場合を説明したが、これに限定されるものではない。

30

#### 【0041】

##### [プロセッサコアの割り当ての一例]

図5は、プロセッサコアの割り当ての一例を示す図である。すなわち、上記並列化パラメータの変数P1, P2で表わしたD0ループへのプロセッサコアの割り当ての一例を示す。図5の一例では、48個のプロセッサコア(計算コア)の場合である。

40

#### 【0042】

図5上図は、48個のプロセッサコアの割り当てが(12, 4)の場合である。かかる場合には、プログラムは、コアごとに、1次元目(1重目)のD0ループを12分割で実行され、2次元目(2重目)のD0ループを4分割で実行される。

#### 【0043】

図5下図は、48個のプロセッサコアの割り当てが(8, 6)の場合である。かかる場合には、プログラムは、コアごとに、1次元目(1重目)のD0ループを8分割で実行され、2次元目(2重目)のD0ループを6分割で実行される。

#### 【0044】

##### [D0ループの実行例]

50

図 6 は、D O ループの実行例を示す図である。図 6 の一例では、図 2 で示したプログラムの L 1 , L 2 が「 1」、U 1 , U 2 が「 1 0 0」の場合である。すなわち、1次元目(1重目)のループの変動範囲 L 1 , U 1 は、1 , 1 0 0 である。2次元目(2重目)のループの変動範囲 L 2 , U 2 は、1 , 1 0 0 である。また、最適化パラメータのプロセッサコアの割り当て(P 1 , P 2)が( 8 , 6)の場合である。

【 0 0 4 5】

ここでは、C O R E # 0 が示すプロセッサコアは、1重目のD O ループの変動範囲( 1 , 1 3 )を担い、2重目のD O ループの変動範囲( 1 , 1 7 )を担う。C O R E # 1 が示すプロセッサコアは、1重目のD O ループの変動範囲( 1 4 , 2 6 )を担い、2重目のD O ループの変動範囲( 1 , 1 7 )を担う。C O R E # 2 が示すプロセッサコアは、1重目のD O ループの変動範囲( 2 7 , 3 9 )を担い、2重目のD O ループの変動範囲( 1 , 1 7 )を担う。・・・C O R E # 7 が示すプロセッサコアは、1重目のD O ループの変動範囲( 9 2 , 1 0 0 )を担い、2重目のD O ループの変動範囲( 1 , 1 7 )を担う。また、C O R E # 8 が示すプロセッサコアは、1重目のD O ループの変動範囲( 1 , 1 3 )を担い、2重目のD O ループの変動範囲( 1 8 , 3 4 )を担う。C O R E # 1 6 が示すプロセッサコアは、1重目のD O ループの変動範囲( 1 , 1 3 )を担い、2重目のD O ループの変動範囲( 3 5 , 5 1 )を担う。C O R E # 2 4 が示すプロセッサコアは、1重目のD O ループの変動範囲( 1 , 1 3 )を担い、2重目のD O ループの変動範囲( 5 2 , 6 8 )を担う。C O R E # 3 2 が示すプロセッサコアは、1重目のD O ループの変動範囲( 1 , 1 3 )を担い、2重目のD O ループの変動範囲( 6 9 , 8 5 )を担う。C O R E # 4 0 が示すプロセッサコアは、1重目のD O ループの変動範囲( 1 , 1 3 )を担い、2重目のD O ループの変動範囲( 8 6 , 1 0 0 )を担う。このように、プロセッサコアの割り当てに従って、スレッドにd o ループの繰り返し処理の一部を割り当てる。

【 0 0 4 6】

[最適化パラメータ抽出の一例]

図 7 は、最適化パラメータ抽出の一例を示す図である。図 7 に示すように、生成部 1 1 によって生成された関係データベース 2 2 が表わされている。なお、最適化対象プログラムに対応するアプリ実行特徴値セットが特徴値算出部 1 3 によって算出されたものとする。

【 0 0 4 7】

最適化パラメータ抽出部 1 4 は、関係データベース 2 2 を参照して、特徴値算出部 1 3 によって算出されたアプリ実行特徴値セットと最も距離が近いアプリ実行特徴値セットを取得する。ここでは、アプリ実行特徴値セットとして( 5 , 0 . 8 , 0 . 2 , 1 , 0 . 2 , 0 . 1 )が取得される。すると、最適化パラメータ抽出部 1 4 は、関係データベース 2 2 を参照して、取得したアプリ実行特徴値セットに対応する最適並列化パラメータを抽出する。ここでは、最適並列化パラメータとして( 1 2 , 4 , g u i d e d , s t a t i c , 3 2 , 1 )が抽出される。

【 0 0 4 8】

[最適化されたプログラムの一例]

図 8 は、最適化されたプログラムの一例を示す図である。ここでは、最適化対象プログラムの最適並列化パラメータとして( 1 2 , 4 , g u i d e d , s t a t i c , 3 2 , 1 )が抽出されたものとする。符号 p 1 に示すように、D O ループへのプロセッサコアの割り当てを示す値として、1 2 , 4 が設定されている。符号 p 2 に示すように、1次元目のD O ループのスケジューリング方式として「 g u i d e d 」が設定されている。符号 p 3 に示すように、2次元目のD O ループのスケジューリング方式として「 s t a t i c 」が設定されている。符号 p 4 に示すように、1次元目のD O ループのチャンク数として「 3 2 」が設定されている。符号 p 5 に示すように、2次元目のD O ループのチャンク数として「 1 」が設定されている。

【 0 0 4 9】

そして、最適並列化パラメータを設定した最適化対象プログラムを、プログラム本実行部 1 5 は、本実行する。

10

20

30

40

50

## 【 0 0 5 0 】

[ 生成処理のフローチャートの一例 ]

図 9 は、実施例に係る生成処理のフローチャートの一例を示す図である。なお、アプリ実行特徴値セットは、図 3 で示す ( X 1 , Y 1 , Z 1 , X 2 , Y 2 , Z 2 ) であるとする。

## 【 0 0 5 1 】

図 9 に示すように、生成部 1 1 は、アプリ実行特徴値セットの 6 次元空間でグリッド状に配置される値を取り得る複数の基準プログラム 2 1 を生成する ( ステップ S 1 1 ) 。

## 【 0 0 5 2 】

そして、生成部 1 1 は、それぞれの基準プログラム 2 1 において、並列化パラメータの全組み合わせの実行を行う。生成部 1 1 は、最も実行時間が短い並列化パラメータを、最適並列化パラメータとして、基準プログラム 2 1 に対応するアプリ実行特徴値セットと対応付けて関係データベース 2 2 に格納する ( ステップ S 1 2 ) 。

10

## 【 0 0 5 3 】

[ 最適化パラメータ抽出処理のフローチャートの一例 ]

図 1 0 は、実施例に係る最適化パラメータ抽出処理のフローチャートの一例を示す図である。

## 【 0 0 5 4 】

図 1 0 に示すように、最適化対象プログラムを受け付けたプログラム仮実行部 1 2 は、受け付けた最適化対象プログラムを 1 回実行する。そして、特徴値算出部 1 3 は、アプリ実行特徴値セットを取得する ( ステップ S 2 1 ) 。実行時のアプリ実行特徴値セットのそれぞれの特定値は、特定のプロファイル等の性能情報取得ツールを用いて求められれば良い。

20

## 【 0 0 5 5 】

そして、最適化パラメータ抽出部 1 4 は、取得されたアプリ実行特徴値セットをまとめる ( ステップ S 2 2 ) 。

## 【 0 0 5 6 】

そして、最適化パラメータ抽出部 1 4 は、まとめた後のアプリ実行特徴値セットをキーに、関係データベース 2 2 を参照して、最適並列化パラメータを抽出する ( ステップ S 2 3 ) 。

[ プログラム本実行処理のフローチャートの一例 ]

30

図 1 1 は、実施例に係るプログラム本実行処理のフローチャートの一例を示す図である。

## 【 0 0 5 7 】

プログラム本実行部 1 5 は、最適化対象プログラムに最適並列化パラメータを設定して実行する ( ステップ S 3 1 ) 。

## 【 0 0 5 8 】

[ 実施例の効果 ]

上記実施例によれば、並列処理装置 1 は、所定の並列化パラメータを設定した、2 重ループを有する最適化対象のプログラムを並列処理で 1 回だけ実行する。並列処理装置 1 は、実行された結果から最適化対象のプログラムの特徴値を算出する。並列処理装置 1 は、2 重ループを有するサンプルプログラムにおける特徴値と最適な並列化パラメータとの相関関係を示すデータベースを参照して、算出された特徴値に対応する最適な並列化パラメータを抽出する。かかる構成によれば、並列処理装置 1 は、2 重ループを有する最適化対象のプログラムを並列処理で実行する場合に、1 回だけ実行するだけで、最適な並列化パラメータを高速に求めることができる。

40

## 【 0 0 5 9 】

また、上記実施例によれば、並列処理装置 1 は、特徴値を持つサンプルプログラムに対して複数通りの並列化パラメータをそれぞれ設定して実行した際の結果を基に当該プログラムにおける特徴値と最適な並列化パラメータとの相関関係を示すデータベースを生成する。並列処理装置 1 は、生成されたデータベースを参照して、算出された前記最適化対象のプログラムの特徴値に対応する最適な並列化パラメータを抽出する。かかる構成によれ

50

ば、並列処理装置 1 は、2 重ループを有するプログラムにおける特徴値と並列化パラメータとの相関関係を生成することで、最適な並列化パラメータを高速に求めることができる。

【 0 0 6 0 】

また、上記実施例によれば、並列処理装置 1 は、以下の相関関係を示すデータベースを生成する。相関関係は、ループごとの、実行命令数の平均、実行命令数の相対標準偏差およびキャッシュの再利用率の平均を示す複数の特徴値と、ループごとの、プロセッサコアの割り当て、並列化のスケジューリング方式および並列化のスケジューリング方式のチャンク数を示す複数の並列化パラメータとの相関関係である。かかる構成によれば、並列処理装置 1 は、2 重ループを有するプログラムにおける特徴値と並列化パラメータとの相関関係を生成することで、最適な並列化パラメータを高速に求めることができる。

10

【 0 0 6 1 】

[ その他 ]

なお、上記実施例では、プログラムとして Fortran 言語を一例として説明した。しかしながら、プログラムは、C 言語や C++ 言語であっても良く、OpenMP 規格で扱える言語であれば良い。

【 0 0 6 2 】

また、上記実施例では、複数レベルの並列化を行なえるアプリケーションプログラムとしてスレッド並列化規格である OpenMP 規格を一例として説明した。しかしながら、OpenMP 規格に限定されず、複数レベルの並列化を行なえるアプリケーションプログラムとしてスレッド並列化規格である所定の規格であれば良い。

20

【 0 0 6 3 】

また、図示した並列処理装置 1 の各構成要素は、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、並列処理装置 1 の分散・統合の具体的態様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、特徴値算出部 1 3 および最適化パラメータ抽出部 1 4 を一つの部として統合しても良い。また、生成部 1 1 を、基準プログラム 2 1 を生成する第 1 生成部と、関係データベース 2 2 を生成する第 2 生成部とに分散しても良い。また、記憶部 2 0 を並列処理装置 1 の外部装置としてネットワーク経由で接続するようにしても良い。

【 0 0 6 4 】

また、上記実施例で説明した各種の処理は、予め用意されたプログラムをパーソナルコンピュータやワークステーションなどのコンピュータで実行することによって実現することができる。そこで、以下では、図 1 に示した並列処理装置 1 と同様の機能を実現する分析プログラムを実行するコンピュータの一例を説明する。図 1 2 は、並列処理プログラムを実行するコンピュータの一例を示す図である。

30

【 0 0 6 5 】

図 1 2 に示すように、コンピュータ 2 0 0 は、各種演算処理を実行する CPU 2 0 3 と、ユーザからのデータの受け付ける入力装置 2 1 5 と、表示装置 2 0 9 を制御する表示制御部 2 0 7 とを有する。また、コンピュータ 2 0 0 は、記憶媒体からプログラムなどを読取るドライブ装置 2 1 3 と、ネットワークを介して他のコンピュータとの間でデータの授受を行う通信制御部 2 1 7 とを有する。また、コンピュータ 2 0 0 は、各種情報を一時記憶するメモリ 2 0 1 と、HDD 2 0 5 を有する。そして、メモリ 2 0 1、CPU 2 0 3、HDD 2 0 5、表示制御部 2 0 7、ドライブ装置 2 1 3、入力装置 2 1 5、通信制御部 2 1 7 は、バス 2 1 9 で接続されている。

40

【 0 0 6 6 】

ドライブ装置 2 1 3 は、例えばリムーバブルディスク 2 1 1 用の装置である。HDD 2 0 5 は、並列処理プログラム 2 0 5 a および並列処理関連情報 2 0 5 b を記憶する。

【 0 0 6 7 】

CPU 2 0 3 は、並列処理プログラム 2 0 5 a を読み出して、メモリ 2 0 1 に展開し、プロセスとして実行する。かかるプロセスは、並列処理装置 1 の各機能部に対応する。並

50

列処理関連情報 2 0 5 b は、基準プログラム 2 1 および関係データベース 2 2 に対応する。そして、例えばリムーバブルディスク 2 1 1 が、並列処理プログラム 2 0 5 a などの各情報を記憶する。

【 0 0 6 8 】

なお、並列処理プログラム 2 0 5 a については、必ずしも最初から HDD 2 0 5 に記憶させておかなくても良い。例えば、コンピュータ 2 0 0 に挿入されるフレキシブルディスク (FD)、CD-ROM、DVD ディスク、光磁気ディスク、IC カードなどの「可搬用の物理媒体」に当該プログラムを記憶させておく。そして、コンピュータ 2 0 0 がこれらから並列処理プログラム 2 0 5 a を読み出して実行するようにしても良い。

【 符号の説明 】

10

【 0 0 6 9 】

1 並列処理装置

1 0 制御部

1 1 生成部

1 2 プログラム仮実行部

1 3 特徴値算出部

1 4 最適化パラメータ抽出部

1 5 プログラム本実行部

2 0 記憶部

2 1 基準プログラム

20

2 2 関係データベース

30

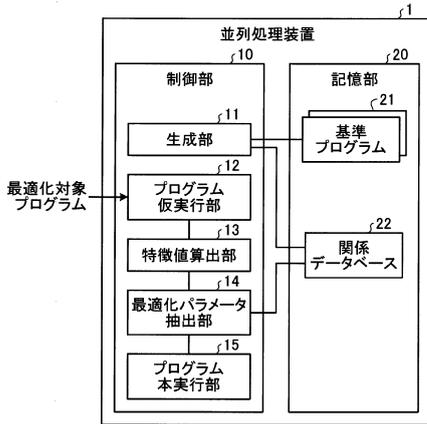
40

50

【 図 面 】

【 図 1 】

実施例に係る並列処理装置の機能構成を示すブロック図



【 図 2 】

プログラムの一例を示す図

```

ソースプログラムの例: sample.f
OMP_PARALLEL=P1,P2

subroutine sample
!$omp parallel do schedule(D1, C1)
do I1 = L1, U1
!$omp parallel do schedule(D2, C2)
do I2 = L2, U2
  <body>
end do
end do
end subroutine
  
```

10

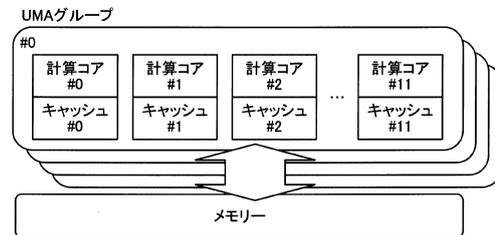
【 図 3 】

実施例に係る関係データベースの一例を示す図

アプリ実行特徴値セット (X1, Y1, Z1, X2, Y2, Z2)	最適並列化パラメータ (P1, P2, D1, D2, C1, C2)
(1, 0.1, 0.1, 1, 0.1, 0.1)	(6, 8, static, static, 1, 1)
(2, 0.1, 0.1, 1, 0.1, 0.1)	(6, 8, static, static, 1, 1)
...	...
(10, 0.1, 0.1, 1, 0.1, 0.1)	(6, 8, dynamic, static, 1, 1)
(1, 0.2, 0.1, 1, 0.1, 0.1)	(6, 8, static, static, 1, 1)
(2, 0.2, 0.1, 1, 0.1, 0.1)	(6, 8, static, static, 1, 1)
...	...
(10, 0.2, 0.1, 1, 0.1, 0.1)	(6, 8, dynamic, static, 1, 1)
(1, 0.3, 0.1, 1, 0.1, 0.1)から (10, 0.8, 1.0, 10, 1.0, 1.0)までの項目	.....
(1, 0.9, 1.0, 10, 1.0, 1.0)	(6, 8, static, guided, 32, 32)
(2, 0.9, 1.0, 10, 1.0, 1.0)	(6, 8, dynamic, guided, 32, 32)
...	...
(10, 0.9, 1.0, 10, 1.0, 1.0)	(6, 8, guided, guided, 32, 32)
(1, 1.0, 1.0, 10, 1.0, 1.0)	(6, 8, static, guided, 32, 32)
(2, 1.0, 1.0, 10, 1.0, 1.0)	(6, 8, dynamic, guided, 32, 32)
...	...
(10, 1.0, 1.0, 10, 1.0, 1.0)	(6, 8, guided, guided, 32, 32)

【 図 4 】

プログラムを実行するハードウェアの一例を示す図



20

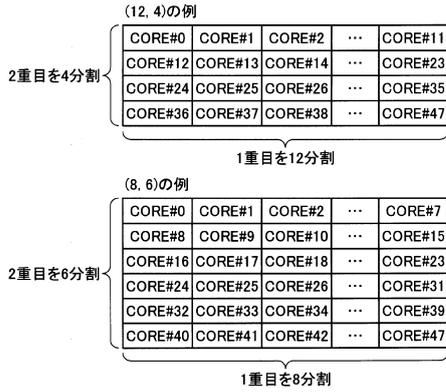
30

40

50

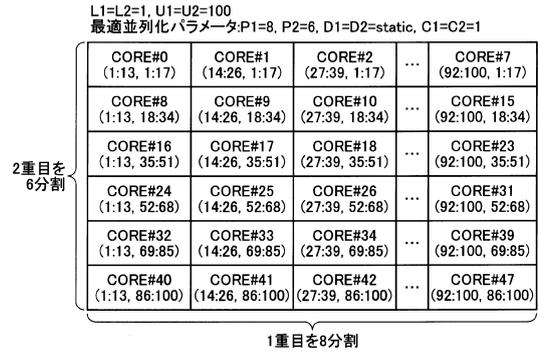
【 図 5 】

プロセッサコアの割り当ての一例を示す図



【 図 6 】

DOループの実行例を示す図



10

【 図 7 】

最適化パラメータ抽出の一例を示す図

アプリ実行特徴値セット (X1, Y1, Z1, X2, Y2, Z2)	最適並列化パラメータ (P1, P2, D1, D2, C1, C2)
(1, 0.1, 0.1, 1, 0.1, 0.1)	(6, 8, static, static, 1, 1)
...	...
(5, 0.8, 0.2, 1, 0.2, 0.1)	(12, 4, guided, static, 32, 1)
...	...
(10, 1.0, 1.0, 10, 1.0, 1.0)	(6, 8, guided, guided, 32, 32)

【 図 8 】

最適化されたプログラムの一例を示す図

```

最適化されたソースプログラムの例: sample-opt.f
OMP_PARALLEL=12,4
subroutine sample
!$omp parallel do schedule(guided, 32)
do I1 = L1, U1
!$omp parallel do schedule(static, 1)
<body>
end do
end do
end subroutine

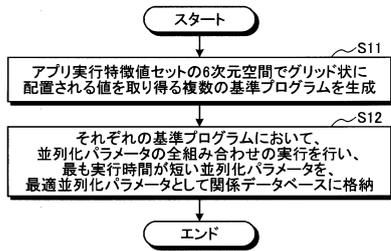
```

Diagram showing process flow: p1 (main), p2, p4 (parallel), p3, p5 (subroutine).

20

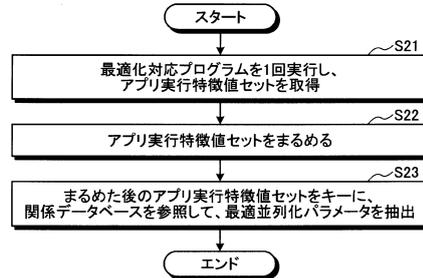
【 図 9 】

実施例に係る生成処理のフローチャートの一例を示す図



【 図 10 】

実施例に係る最適化パラメータ抽出処理のフローチャートの一例を示す図



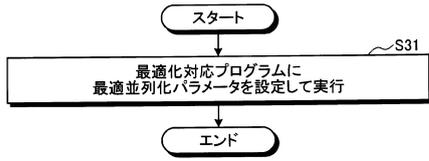
30

40

50

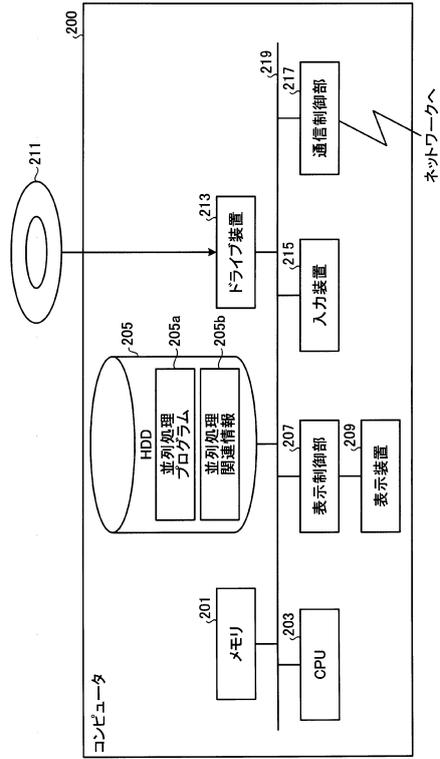
【図 1 1】

実施例に係るプログラム本実行処理のフローチャートの一例を示す図



【図 1 2】

並列処理プログラムを実行するコンピュータの一例を示す図



10

20

30

40

50

---

フロントページの続き

- (56)参考文献 特開 2 0 1 8 - 1 2 0 3 8 7 ( J P , A )  
米国特許出願公開第 2 0 1 9 / 0 0 1 2 1 5 5 ( U S , A 1 )
- (58)調査した分野 (Int.Cl. , D B 名)  
G 0 6 F 9 / 5 0