

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5776776号
(P5776776)

(45) 発行日 平成27年9月9日(2015.9.9)

(24) 登録日 平成27年7月17日(2015.7.17)

(51) Int.Cl.	F I
G 0 6 F 12/02 (2006.01)	G O 6 F 12/02 5 1 O M
G 0 6 F 9/50 (2006.01)	G O 6 F 9/46 4 6 2 A

請求項の数 12 (全 20 頁)

(21) 出願番号	特願2013-527764 (P2013-527764)	(73) 特許権者	000005223
(86) (22) 出願日	平成23年8月5日(2011.8.5)		富士通株式会社
(86) 国際出願番号	PCT/JP2011/067990		神奈川県川崎市中原区上小田中4丁目1番1号
(87) 国際公開番号	W02013/021441	(74) 代理人	100104190
(87) 国際公開日	平成25年2月14日(2013.2.14)		弁理士 酒井 昭徳
審査請求日	平成26年1月22日(2014.1.22)	(72) 発明者	山下 浩一郎
			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	山内 宏真
			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	鈴木 貴久
			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】 データ処理システム、およびデータ処理方法

(57) 【特許請求の範囲】

【請求項 1】

複数のデータ処理装置と、
 周辺装置と、
 前記複数のデータ処理装置および前記周辺装置に共有されるメモリと、
 前記周辺装置に対応して設けられる周辺メモリと、
 前記複数のデータ処理装置または前記周辺装置で実行されるスレッド情報をシーケンスに格納するヒープ領域から読み出される前記スレッド情報に基づいて前記スレッド情報に基づくスレッドに前記メモリまたは前記周辺メモリの領域を確保するメモリ管理ユニットと

を含むことを特徴とするデータ処理システム。

【請求項 2】

前記複数のデータ処理装置の負荷が均衡状態にないときは、前記メモリ管理ユニットの実行を前記複数のデータ処理装置のうちで負荷の最も小さいデータ処理装置に割り当てるスケジューラを含むこと

を特徴とする請求項 1 に記載のデータ処理システム。

【請求項 3】

前記複数のデータ処理装置の負荷が均衡状態にあるときは、前記スレッド情報に基づくスレッドを前記スレッドが過去に割り当てられたデータ処理装置に割り当てるスケジューラを含むこと

を特徴とする請求項 1 に記載のデータ処理システム。

【請求項 4】

前記メモリ管理ユニットは、前記スレッド情報に基づく優先度情報に基づいて前記スレッドに前記メモリまたは前記周辺メモリの領域を確保すること

を特徴とする請求項 1 乃至請求項 3 の何れかに記載のデータ処理システム。

【請求項 5】

前記メモリ管理ユニットは、前記優先度情報が高優先を示すとき、前記スレッドに前記メモリまたは前記周辺メモリのうちでメモリアクセス速度が速いメモリの領域を確保すること

を特徴とする請求項 4 に記載のデータ処理システム。

10

【請求項 6】

前記メモリ管理ユニットは、前記優先度情報が中優先または低優先を示すとき、前記周辺メモリに空きがあるときは前記スレッドに前記周辺メモリの領域を確保すること

を特徴とする請求項 4 または請求項 5 に記載のデータ処理システム。

【請求項 7】

前記メモリ管理ユニットは、前記スレッド情報に基づくスレッドが前記周辺装置を管理するスレッドであるとき、前記スレッドに前記周辺メモリの領域を確保すること

を特徴とする請求項 1 乃至請求項 6 の何れかに記載のデータ処理システム。

【請求項 8】

前記メモリおよび前記周辺メモリの使用状況を管理する上位メモリ管理ユニットを含み、

20

前記メモリ管理ユニットは、定期的に前記メモリおよび前記周辺メモリの使用状況を前記上位メモリ管理ユニットに通知すること

を特徴とする請求項 1 乃至請求項 7 の何れかに記載のデータ処理システム。

【請求項 9】

複数のデータ処理装置のうちの第 1 データ処理装置は、前記複数のデータ処理装置または周辺装置で実行されるスレッド情報をヒープ領域にシーケンスに格納し、

前記ヒープ領域から読み出される前記スレッド情報に基づいて前記スレッド情報に基づくスレッドに前記複数のデータ処理装置で共有されるメモリまたは前記周辺装置に対応する周辺メモリの領域を確保すること

30

を特徴とするデータ処理方法。

【請求項 10】

前記スレッド情報に含まれる優先度情報に基づいて、前記スレッドに前記メモリまたは前記周辺メモリの領域を確保すること

を特徴とする請求項 9 に記載のデータ処理方法。

【請求項 11】

前記優先度情報が高優先を示すとき、前記スレッドに前記メモリまたは前記周辺メモリのうちでメモリアクセス速度が速いメモリの領域を確保すること

を特徴とする請求項 10 に記載のデータ処理方法。

【請求項 12】

40

前記優先度情報が中優先または低優先を示すとき、前記周辺メモリに空きがあるときは前記スレッドに前記周辺メモリの領域を確保すること

を特徴とする請求項 10 または請求項 11 に記載のデータ処理方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、メモリを管理するデータ処理システム、およびデータ処理方法に関する。

【背景技術】

【0002】

従来から、携帯電話等の携帯端末のメモリアーキテクチャには、様々なメモリが搭載さ

50

れている。具体的なメモリの種類としては、CPU (Central Processing Unit) が主にアクセスするメインメモリとなるRAM (Random Access Memory) の他に、GPU (Graphics Processing Unit) が用いるGRAM (Graphics RAM)、DSP (Digital Signal Processor) が用いるバッファ用メモリ、LCD (Liquid Crystal Display) コントローラのVRAM (Video RAM) などが存在する。

【0003】

このような様々なメモリのアクセス方法に関して、たとえば、アプリケーションソフトウェア（以下、「アプリ」と称する）からのメモリアクセスに対してスレッドを生成し、
10 該当のスレッドがメモリアクセスに関する管理を行う技術が開示されている（たとえば、下記特許文献1を参照。）。なお、スレッドとはCPUで行う処理を管理するための基本単位である。

【先行技術文献】

【特許文献】

【0004】

【特許文献1】特開2008-108089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、上述した従来技術において、メモリアクセスの管理をするスレッドを、
任意のCPUで実行できるようにすると、同時アクセスによる競合を避けるため、排他制御
20 処理を追加せねばならず、オーバーヘッドが増加するという問題があった。

【0006】

本発明は、上述した従来技術による問題点を解消するため、複数アクセスによる競合を
発生させずに複数のメモリを管理できるデータ処理システム、およびデータ処理方法を提
供することを目的とする。

【課題を解決するための手段】

【0007】

上述した課題を解決し、目的を達成するため、本発明の一側面によれば、複数のデータ
30 処理装置と、周辺装置と、複数のデータ処理装置および周辺装置に共有されるメモリと、
周辺装置に対応して設けられる周辺メモリと、複数のデータ処理装置または周辺装置で
実行されるスレッド情報をシーケンスに格納するヒープ領域から読み出されるスレッド情報
に基づいてスレッド情報に基づくスレッドにメモリまたは周辺メモリの領域を割り当てる
メモリ管理ユニットとを含むデータ処理システム、およびデータ処理方法が提案される。

【発明の効果】

【0008】

本発明の一側面によれば、複数アクセスによる競合を発生させずに複数のメモリを管理
できるという効果を奏する。

【図面の簡単な説明】

【0009】

【図1】図1は、マルチコアプロセッサシステムの動作例を示す説明図である。

【図2】図2は、マルチコアプロセッサシステムのハードウェア例を示すブロック図であ
る。

【図3】図3は、マルチコアプロセッサシステムのソフトウェア例を示すブロック図であ
る。

【図4】図4は、マルチコアプロセッサシステムの機能例を示すブロック図である。

【図5】図5は、管理要求ディスパッチテーブルの記憶内容の一例を示す説明図である。

【図6】図6は、メモリ使用テーブルの記憶内容の一例を示す説明図である。

【図7】図7は、スレッド動作履歴テーブルの一例を示す説明図である。

10

20

30

40

50

【図 8】図 8 は、負荷が均衡状態でない場合のスレッド割当方法の一例を示す説明図である。

【図 9】図 9 は、負荷が均衡状態である場合のスレッド割当方法の一例を示す説明図である。

【図 10】図 10 は、メモリの確保方法の一例を示す説明図（その 1）である。

【図 11】図 11 は、メモリの確保方法の一例を示す説明図（その 2）である。

【図 12】図 12 は、メモリ管理要求処理手順の一例を示すフローチャートである。

【図 13】図 13 は、メモリ管理スレッドの処理手順の一例を示すフローチャートである。

【図 14】図 14 は、スケジューラによる割当対象スレッドの割当先 CPU 選択処理の一例を示すフローチャートである。 10

【図 15】図 15 は、メモリ確保先選択処理の一例を示すフローチャートである。

【図 16】図 16 は、本実施の形態にかかるコンピュータを用いたシステムの適用例を示す説明図である。

【発明を実施するための形態】

【0010】

以下に添付図面を参照して、開示のデータ処理システム、およびデータ処理方法の実施の形態を詳細に説明する。なお、本実施の形態にかかるデータ処理システムの例として、複数の CPU を有するマルチコアプロセッサシステムにて説明を行う。マルチコアプロセッサとは、コアが複数搭載されたプロセッサである。コアが複数搭載されていれば、複数のコアが搭載された単一のプロセッサでもよく、シングルコアのプロセッサが並列されているプロセッサ群でもよい。なお、本実施の形態では、説明を単純化するため、シングルコアのプロセッサが並列されているプロセッサ群を例に挙げて説明する。 20

【0011】

図 1 は、マルチコアプロセッサシステムの動作例を示す説明図である。マルチコアプロセッサシステム 100 は、CPU #0 と、CPU #1 と、GPU 101 と、GPU メモリ 102 と、メインメモリ 103 と、を含む。以下、接尾記号 “#n” が付随された記号は、n 番目の CPU に対応する記号であることを示している。CPU #0 と CPU #1 は、ローカルバス 112 で接続されており、GPU 101 と GPU メモリ 102 は、ローカルバス 113 で接続されている。ローカルバス 112 とローカルバス 113 とメインメモリ 103 は、バス 111 で接続されている。メインメモリ 103 は、アクセス速度が高速なメモリであり、GPU メモリ 102 は、アクセス速度が低速なメモリである。なお、CPU 以外の装置を周辺装置と称する。また、周辺装置に対応して設けられるメモリを周辺メモリと称する。図 1 の例では、GPU メモリ 102 が周辺メモリとなる。 30

【0012】

また、CPU #0 は、高優先度のスレッド 0 を実行し、CPU #1 は、低優先度のスレッド 1 とメモリ管理スレッド 122 を実行する。CPU #0、CPU #1 は、スレッド 0、スレッド 1 による malloc () 関数等といったメモリ確保要求を実行すると、ダミーデバイスドライバ 121 #0、ダミーデバイスドライバ 121 #1 を実行する。 40

【0013】

ダミーデバイスドライバ 121 は、個々の周辺機器における通常のデバイスドライバと同一のインターフェースを有する API (Application Programming Interface) である。スレッドは、ダミーデバイスドライバ 121 に対して通常のデバイスドライバと同一の操作を行うことができる。ダミーデバイスドライバ 121 は、メモリ確保要求を受け付けると、メインメモリ 103 に格納されている管理要求ディスパッチテーブル 123 にスレッド情報を付与したメモリ確保要求を格納する。スレッド情報とは、スレッドの名称、スレッドの優先度情報等が含まれた情報である。 50

【0014】

メモリ管理スレッド 122 は、管理要求ディスパッチテーブル 123 からメモリ確保要求を順に読み出し、メモリ確保要求のスレッド情報に基づいて、メインメモリ 103、ま

たは、GPUメモリ102の領域を確保する。

【0015】

図1の例では、メモリ管理スレッド122は、初めにスレッド0用領域131をメインメモリ103内に確保し、次にスレッド1用領域132をGPUメモリ102内に確保する。このように、マルチコアプロセッサシステム100は、複数のCPUで実行中のスレッドからのメモリ確保要求を管理要求ディスパッチテーブル123に格納し、確保要求を順に読み出し、メインメモリ103や周辺装置のメモリに割り当てる。これにより、マルチコアプロセッサシステム100は、複数アクセスによる競合を発生させずにメモリ確保できる。図2～図15にて、図1で示した動作を行うマルチコアプロセッサシステム100について説明を行う。

10

【0016】

図2は、マルチコアプロセッサシステムのハードウェア例を示すブロック図である。図2において、携帯電話などの携帯端末を想定するマルチコアプロセッサシステム100は、CPU#0と、CPU#1と、GPU101と、GPUメモリ102と、メインメモリ103と、DSP201と、DSPメモリ202と、を含む。DSP201とDSPメモリ202は、ローカルバス203で接続されている。また、マルチコアプロセッサシステム100は、ROM(Read-Only Memory)204と、RAM205と、フラッシュROM206と、フラッシュROMコントローラ207と、フラッシュROM208と、を含む。

20

【0017】

また、マルチコアプロセッサシステム100は、ユーザやその他の機器との入出力装置として、ディスプレイ209と、I/F(Interface)210と、キーボード211と、を含む。また、各部はバス111によってそれぞれ接続されている。なお、図1に示したメインメモリ103は、RAM205であってもよいし、RAM205の一部であってもよい。また、メインメモリ103は、CPU#0、CPU#1、GPU101、DSP201、等から共有してアクセスできるメモリである。

【0018】

ここで、CPU#0、CPU#1は、マルチコアプロセッサシステム100の全体の制御を司る。CPU#0、CPU#1は、シングルコアのプロセッサを並列して接続した全てのCPUを指している。なお、マルチコアプロセッサシステム100は、3つ以上のCPUを含んでいてもよい。CPU#0～CPU#nは、それぞれ専用のキャッシュメモリを有している。

30

【0019】

ROM204は、ブートプログラムなどのプログラムを記憶している。RAM205は、CPU#0、CPU#1のワークエリアとして使用される。フラッシュROM206は、読出し速度が高速なフラッシュROMであり、たとえば、NOR型フラッシュメモリである。フラッシュROM206は、OS(Operating System)などのシステムソフトウェアやアプリなどを記憶している。たとえば、OSを更新する場合、マルチコアプロセッサシステム100は、I/F210によって新しいOSを受信し、フラッシュROM206に格納されている古いOSを、受信した新しいOSに更新する。

40

【0020】

フラッシュROMコントローラ207は、CPU#0、CPU#1の制御にしたがってフラッシュROM208に対するデータのリード/ライトを制御する。フラッシュROM208は、データの保存、運搬を主に目的としたフラッシュROMであり、たとえば、NAND型フラッシュメモリである。フラッシュROM208は、フラッシュROMコントローラ207の制御で書き込まれたデータを記憶する。データの具体例としては、マルチコアプロセッサシステム100を使用するユーザがI/F210を通して取得した画像データ、映像データなどや、また本実施の形態にかかるデータ処理方法を実行するプログラムなどを記憶してもよい。フラッシュROM208は、たとえば、メモリカード、SDカードなどを採用することができる。

50

【0021】

ディスプレイ209は、カーソル、アイコンあるいはツールボックスをはじめ、文書、画像、機能情報などのデータを表示する。ディスプレイ209は、たとえば、TFT液晶ディスプレイなどを採用することができる。

【0022】

I/F210は、通信回線を通じてLAN(Local Area Network)、WAN(Wide Area Network)、インターネットなどのネットワーク212に接続され、ネットワーク212を介して他の装置に接続される。そして、I/F210は、ネットワーク212と内部のインターフェースを司り、外部装置からのデータの入出力を制御する。I/F210には、たとえばモデムやLANアダプタなどを採用す

10

【0023】

キーボード211は、数字、各種指示などの入力のためのキーを有し、データの入力を行う。また、キーボード211は、タッチパネル式の入力パッドやテンキーなどであってもよい。

【0024】

このように、マルチコアプロセッサシステム100は、CPU群だけを見ると、共有メモリアーキテクチャとなるが、GPU101、DSP201等といった、メモリに対して行うマスタを複数有する分散メモリアーキテクチャとなる。さらに、メインメモリ103、GPUメモリ102、DSPメモリ202は、各マスタから互いにアクセス可能な共有空間として扱われており、複雑な入れ子型階層となる。

20

【0025】

図3は、マルチコアプロセッサシステムのソフトウェア例を示すブロック図である。マルチコアプロセッサシステム100は、OSが提供するソフトウェアとして、カーネル301、スケジューラ302、メモリ管理スレッド122、DSPダミーデバイスドライバ303、GPUダミーデバイスドライバ304を実行する。また、メモリ管理スレッド122は、メインメモリ管理部311、DSPメモリ管理部312、GPUメモリ管理部313を含む。

【0026】

なお、カーネル301、DSPダミーデバイスドライバ303、GPUダミーデバイスドライバ304は、CPU#0、CPU#1それぞれで実行される。具体的に、CPU#0は、カーネル301#0、DSPダミーデバイスドライバ303#0、GPUダミーデバイスドライバ304#0を実行する。また、CPU#1は、カーネル301#1、DSPダミーデバイスドライバ303#1、GPUダミーデバイスドライバ304#1を実行する。スケジューラ302は、CPU#0、CPU#1のいずれが動作してもよいが、本実施の形態ではマルチコアプロセッサシステム100でのマスタCPUとなるCPU#0が実行する。メモリ管理スレッド122は、CPU#0、CPU#1のいずれかが実行する。

30

【0027】

カーネル301は、OSの中核となる機能を有する。たとえば、カーネル301は、アプリが起動された場合、プログラムコードをメインメモリ103内に展開する。スケジューラ302は、マルチコアプロセッサシステム100内で実行されるスレッドを各CPUに割り当て、次に実行するスレッドを選択する機能を有する。たとえば、スケジューラ302は、スレッド0をCPU#0に割り当て、スレッド1をCPU#1に割り当てる。

40

【0028】

DSPダミーデバイスドライバ303、GPUダミーデバイスドライバ304は、それぞれ、GPU101、DSP201のデバイスドライバに対するダミーデバイスドライバ121である。

【0029】

メインメモリ管理部311、DSPメモリ管理部312、GPUメモリ管理部313は

50

、それぞれ、メインメモリ103、DSPメモリ202、GPUメモリ102を管理する機能を有する。具体的に、メインメモリ管理部311、DSPメモリ管理部312、GPUメモリ管理部313は、各メモリの物理アドレス、確保可能な物理アドレス範囲を記憶している。たとえば、メインメモリ管理部311は、メインメモリ103への確保要求、解放要求に応じて、使用状況を更新する。DSPメモリ管理部312、GPUメモリ管理部313も、それぞれ、DSPメモリ202、GPUメモリ102の使用状況を更新する。

【0030】

(マルチコアプロセッサシステム100の機能)

次に、マルチコアプロセッサシステム100の機能について説明する。図4は、マルチ
コアプロセッサシステムの機能例を示すブロック図である。マルチコアプロセッサシステム100は、格納部401と、判断部402と、割当部403と、上位メモリ管理部404と、メモリ管理部405と、更新部411と、読出部412と、選択部413と、確保部414と、通知部415と、を含む。この制御部となる機能(格納部401～通知部415)は、記憶装置に記憶されたプログラムをCPU#0、CPU#1が実行することにより、その機能を実現する。記憶装置とは、具体的には、たとえば、図2に示したROM204、RAM205、フラッシュROM206、フラッシュROM208などである。

【0031】

なお、格納部401は、ダミーデバイスドライバ121の機能である。また判断部402と割当部403は、スケジューラ302の機能であり、上位メモリ管理部404はメモリ管理スレッド122の機能であり、メモリ管理部405はメインメモリ管理部311～GPUメモリ管理部313の機能である。さらに、更新部411と読出部412と選択部413は、上位メモリ管理部404に含まれ、確保部414と通知部415は、メモリ管理部405に含まれる。また、図4では、判断部402と割当部403がCPU#0の機能であり、上位メモリ管理部404がCPU#1の機能として図示されているが、判断部402と割当部403がCPU#1の機能であり、上位メモリ管理部404がCPU#0の機能であってもよい。

【0032】

また、マルチコアプロセッサシステム100は、管理要求ディスパッチテーブル123と、メモリ使用テーブル421と、スレッド動作履歴テーブル422と、にアクセス可能である。

【0033】

管理要求ディスパッチテーブル123は、動的に確保可能なメモリであるヒープ領域に存在し、CPU#0、CPU#1、GPU101、DSP201で実行されるスレッド情報とメモリ管理要求とを格納している。なお、管理要求ディスパッチテーブル123の詳細については、図5にて後述する。メモリ使用テーブル421は、メインメモリ103、および周辺メモリの空き状況を格納している。なお、メモリ使用テーブル421の詳細については、図6にて後述する。スレッド動作履歴テーブル422は、各スレッドがどのCPUで動作したかという情報を記憶している。なお、スレッド動作履歴テーブル422の詳細については、図7にて後述する。

【0034】

格納部401は、複数のデータ処理装置または周辺装置で実行されるスレッド情報をヒープ領域にシーケンスに格納する機能を有する。たとえば、格納部401は、CPU#0、CPU#1、GPU101、DSP201で実行されるスレッド情報を管理要求ディスパッチテーブル123に格納する。スレッド情報をシーケンスに格納するとは、スレッド情報を順番に格納していくことである。

【0035】

判断部402は、複数のデータ処理装置の負荷が均衡状態にあるか否かを判断する機能を有する。たとえば、判断部402は、CPU#0とCPU#1の負荷が均衡状態にあるか否かを判断する。判断方法の一例として、たとえば、判断部402は、OSが有してい

10

20

30

40

50

る負荷量判定機能を利用して、スレッドがCPUを占有している時間をCPUごとに算出し、各CPUの負荷量を基に判断する。各CPUの負荷量が均等、または、閾値を元に均等と近似できる場合、判断部402は、均衡状態であると判断し、各CPUの負荷量が閾値以上離れた値となった場合、均衡状態にないと判断する。なお、判断結果は、RAM205、フラッシュROM206といった記憶領域に記憶される。

【0036】

割当部403は、判断部402によって負荷均衡状態にない場合、上位メモリ管理部404、メモリ管理部405の実行を、複数のデータ処理装置のうちで負荷の最も小さいデータ処理装置に割り当てる機能を有する。たとえば、割当部403は、CPU#0とCPU#1が負荷均衡状態にない場合、メモリ管理スレッド122を負荷の小さいCPUに割り当てる。

10

【0037】

また、割当部403は、判断部402によって負荷均衡状態にある場合、上位メモリ管理部404、メモリ管理部405の実行を、過去に割り当てられたデータ処理装置または周辺装置に割り当ててもよい。たとえば、割当部403は、CPU#0とCPU#1が負荷均衡状態にある場合、スレッド動作履歴テーブル422を参照して、過去に割り当てられたCPUに割り当てる。

【0038】

上位メモリ管理部404は、メインメモリ103および周辺メモリの使用状況を管理する機能を有する。たとえば、上位メモリ管理部404は、メモリ管理部405にメモリの使用状況を問い合わせ、メモリ管理部405から使用状況の通知を受け付ける。

20

【0039】

メモリ管理部405は、メモリを管理する機能を有する。メモリ管理部405は、メインメモリ103、GPUメモリ102、DSPメモリ202、といったメモリごとに存在する。メモリ管理部405は、各メモリに対して、論物変換を行って、領域の確保、再確保、解放、リード、ライト等を行う。

【0040】

更新部411は、通知された使用状況から、メモリ使用テーブル421を更新する機能を有する。たとえば、更新部411は、メモリ管理部405から通知された使用状況をメモリ使用テーブル421に格納する。

30

【0041】

読出部412は、ヒープ領域からスレッド情報を読み出す機能を有する。たとえば、読出部412は、管理要求ディスパッチテーブル123からスレッド情報を読み出す。なお、読出結果は、RAM205、フラッシュROM206といった記憶領域に記憶される。

【0042】

選択部413は、ヒープ領域から読み出されるスレッド情報に基づいてスレッド情報に基づくスレッドからのメモリ確保要求に対する確保先を、メモリまたは周辺メモリから選択する機能を有する。また、選択部413は、スレッドからのメモリ確保要求に対する確保先を、スレッド情報に基づく優先度情報に基づいて選択してもよい。たとえば、選択部413は、優先度が高優先度であれば、高速なメインメモリ103を確保先として選択する。また、選択部413は、低優先度であれば、メインメモリ103より低速なDSPメモリ202を確保先に選択する。

40

【0043】

また、選択部413は、スレッド情報に基づくスレッドが周辺装置で実行されるスレッドであるとき、メモリ確保要求に対する確保先として、周辺メモリを選択してもよい。なお、選択結果は、RAM205、フラッシュROM206といった記憶領域に記憶される。

【0044】

確保部414は、選択部413によって選択された確保先のメモリ内の領域を確保する機能を有する。たとえば、確保部414は、メインメモリ103が確保先に選択された場

50

合、メインメモリ103内の領域を確保する。なお、確保された領域は、要求元のスレッドに通知される。

【0045】

通知部415は、メモリの使用状況を上位メモリ管理部404に通知する機能を有する。たとえば、メインメモリ管理部311の機能となる通知部415は、メインメモリ103の使用状況を上位メモリ管理部404に通知する。同様に、DSPメモリ管理部312の機能となる通知部415は、DSPメモリ202の使用状況を上位メモリ管理部404に通知する。また、GPUメモリ管理部313の機能となる通知部415は、GPUメモリ102の使用状況を上位メモリ管理部404に通知する。

【0046】

図5は、管理要求ディスパッチテーブルの記憶内容の一例を示す説明図である。図5では、管理要求ディスパッチテーブル123の記憶内容について説明を行う。管理要求ディスパッチテーブル123はメモリごとに存在する。

【0047】

本実施の形態では、管理要求ディスパッチテーブル123は、メインメモリ103、GPUメモリ102、DSPメモリ202、という3つのメモリに対する管理要求を記憶する。メインメモリ103の管理要求は管理要求ディスパッチテーブル123__Mが記憶し、GPUメモリ102の管理要求は管理要求ディスパッチテーブル123__Gが記憶し、DSPメモリ202の管理要求は管理要求ディスパッチテーブル123__Dが記憶する。以下、管理要求ディスパッチテーブル123__Mについて説明を行うが、管理要求ディスパッチテーブル123__G、管理要求ディスパッチテーブル123__Dも同様の記憶内容となるため、説明を省略する。

【0048】

図5で示す管理要求ディスパッチテーブル123__Mは、レコード123__M-1～レコード123__M-nを格納している。nは自然数である。管理要求ディスパッチテーブル123__Mは、要求元スレッドID、要求メモリサイズという2つのフィールドを含む。要求元スレッドIDフィールドには、メモリ確保要求を行ったスレッドが格納される。要求メモリサイズフィールドには、メモリ確保要求が行われたバイト数が格納される。たとえば、レコード123__M-1は、スレッド0から32[バイト]のメモリ確保要求を格納している。

【0049】

また、管理要求ディスパッチテーブル123__Mは、動的に確保可能なメモリであるヒープ領域に存在し、各レコードがポインタで接続された構造となっている。具体的に、レコード123__M-1が、レコード123__M-2へのポインタを有している。同様に、レコード123__M-2が、レコード123__M-3へのポインタを有しており、レコード123__M-3が、レコード123__M-4へのポインタを有している。また、レコード123__M-nが、レコード123__M-1へのポインタを有している。

【0050】

なお、図5で示した管理要求ディスパッチテーブル123は、メモリの管理要求として、メモリ確保要求を図示したが、他のメモリの管理要求として、メモリの再確保要求、メモリの解放要求が含まれてもよい。さらに、メモリの管理要求として、メモリへのリード、ライト要求が含まれてもよい。この場合、管理要求ディスパッチテーブル123の1レコードが保持するフィールドとしては、要求元スレッドIDフィールドと、各要求の引数と、である。各要求の引数としては、たとえば、メモリの再確保要求であれば、`realloc()`関数の引数である、割当済みのアドレスと要求メモリサイズであり、メモリの解放要求であれば、`free()`関数の引数である、解放対象のアドレスである。

【0051】

図6は、メモリ使用テーブルの記憶内容の一例を示す説明図である。メモリ使用テーブル421は、メモリ種別ごとの空き容量を記憶するテーブルである。図6で示すメモリ使用テーブル421には、レコード421-1～レコード421-3が登録されている。ま

10

20

30

40

50

た、レコード421-1～レコード421-3は、CPU#0、CPU#1で実行されるスレッドからのアクセス速度の順序で登録されている。たとえば、図6で示すレコード421-1～レコード421-3は、降順にて登録されているが、昇順で登録されていてもよい。

【0052】

メモリ使用テーブル421は、メモリ種別、空き容量という2つのフィールドを含む。メモリ種別フィールドには、各メモリを識別する識別情報が格納される。空き容量フィールドには、メモリの空き容量が格納される。

【0053】

たとえば、レコード421-1は、メモリ群のうちアクセス速度が最も速いメインメモリ103の空き容量が50[Mバイト]であることを示している。次に、レコード421-2は、アクセス速度が次に速いGPUメモリ102の空き容量が10[Mバイト]であることを示している。最後に、レコード421-3は、アクセス速度が遅いDSPメモリ202の空き容量が20[Mバイト]であることを示している。

【0054】

図7は、スレッド動作履歴テーブルの一例を示す説明図である。スレッド動作履歴テーブル422は、スレッドが過去に動作した履歴を記憶するテーブルである。図7で示すスレッド動作履歴テーブル422は、レコード422-1～422-5を格納している。スレッド動作履歴テーブル422は、スレッドID、動作CPUIDという2つのフィールドを含む。スレッドIDフィールドは、動作したスレッドのIDが格納される。動作CPUIDフィールドには、スレッドが動作したCPUIDが格納される。

【0055】

たとえば、図7で示すスレッド動作履歴テーブル422は、メモリ管理スレッド122とスレッド1がCPU#1で2回動作しており、スレッド0がCPU#0で1回動作していることを示している。

【0056】

マルチコアプロセッサシステム100は、図4で示した機能、図5～図7で示した記憶内容を使用して、スレッドの割当、メモリの確保を行う。図8、図9にて、スレッドの割当方法の一例を示し、図10、図11にて、メモリの確保方法の一例を示す。

【0057】

図8は、負荷が均衡状態でない場合のスレッド割当方法の一例を示す説明図である。図8で示すマルチコアプロセッサシステム100は、CPU#0がスレッド0を実行し、CPU#1がスレッド1とスレッド2を実行している。この状態で、メモリ管理スレッド122を割り当てる場合、図8のマルチコアプロセッサシステム100の状態は、各CPUの負荷が均衡状態にないため、スケジューラ302は、負荷の最も小さいCPUにメモリ管理スレッド122を割り当てる。

【0058】

図9は、負荷が均衡状態である場合のスレッド割当方法の一例を示す説明図である。図9で示すマルチコアプロセッサシステム100は、CPU#0がスレッド0とスレッド2を実行し、CPU#1がスレッド1とスレッド3を実行している。この状態で、メモリ管理スレッド122を割り当てる場合、図9のマルチコアプロセッサシステム100の状態は、各CPUの負荷が均衡状態にあるため、スケジューラ302は、メモリ管理スレッド122が過去に割り当てられたCPUに割り当てる。

【0059】

図9の例にて、スケジューラ302は、スレッド動作履歴テーブル422を参照した結果、メモリ管理スレッド122が過去にCPU#1で2回動作しているため、メモリ管理スレッド122をCPU#1に割り当てる。なお、過去の動作履歴に基づくことで、各スレッドは同じCPUに割り当てられることが多くなる。

【0060】

次に、メモリ確保方法の一例について図10、図11にて説明する。図10、図11で

10

20

30

40

50

示すスレッド0は、メニュープログラムを想定しており、GPU101を利用し、優先度が中優先度である。スレッド1は、マルチメディアプログラムを想定しており、DSP201を利用し、優先度が高優先度である。スレッド3は、通信プログラムを想定しており、優先度が低優先度である。

【0061】

図10は、メモリの確保方法の一例を示す説明図（その1）である。初めに、スレッド0を実行するCPU#0は、メモリ管理スレッド122によってメインメモリ103内に割り当てられたスレッド0用領域1001にアクセスして処理を行っている。なお、スレッド0用領域1001がメインメモリ103内に割り当てられた要因としては、たとえば、割当が行われたときに、GPUメモリ102、DSPメモリ202の空きがなかった場合である。

10

【0062】

次に、CPU#0がスレッド0を実行し、CPU#1が高優先度であるスレッド1を実行する場合、メモリ管理スレッド122は、優先度の低いスレッドからメモリ確保要求を行う。図10の場合では、メモリ管理スレッド122は、スレッド0のメモリ確保要求を行った後、スレッド1のメモリ確保要求を行う。

【0063】

初めに、メモリ管理スレッド122は、GPUメモリ管理部313を実行して、アクセス速度が低速なGPUメモリ102内にスレッド0用領域1002を確保する。次に、メモリ管理スレッド122は、メインメモリ管理部311を実行して、アクセス速度が高速なメインメモリ103内にスレッド1用領域1003を確保する。

20

【0064】

図11は、メモリの確保方法の一例を示す説明図（その2）である。図11で示すマルチコアプロセッサシステム100は、CPU#0にスレッド0とスレッド2が割り当てられ、CPU#1にスレッド1が割り当てられている。

【0065】

このとき、メモリ管理スレッド122は、初めにスレッド2のメモリ確保要求を行い、次にスレッド0のメモリ確保要求を行い、最後にスレッド1のメモリ確保要求を行う。初めに、メモリ管理スレッド122は、DSPメモリ管理部312を実行して、アクセス速度が低速なDSPメモリ202内にスレッド2用領域1101を確保する。次に、メモリ管理スレッド122は、GPUメモリ管理部313を実行して、アクセス速度が中速なGPUメモリ102内にスレッド0用領域1102を確保する。最後に、メモリ管理スレッド122は、メインメモリ管理部311を実行して、アクセス速度が高速なメインメモリ103内にスレッド1用領域1103を確保する。

30

【0066】

なお、図11の状態、GPU101を利用する新たなスレッドが動作した結果、マルチコアプロセッサシステム100は、スレッド0用領域1102を解放する場合もある。この場合、GPUを用いるスレッドがGPU101およびGPUメモリ102を初期化する。

【0067】

40

図12は、メモリ管理要求処理手順の一例を示すフローチャートである。メモリ管理要求処理は、各CPUのダミーデバイスドライバ121によって実行されるが、図12では、CPU#0のダミーデバイスドライバ121#0によって実行されることを想定する。CPU#0は、ユーザスレッドから周辺装置へのアクセス要求を受け付ける（ステップS1201）。次に、CPU#0は、管理要求ディスパッチテーブル123にメモリ管理要求を格納する（ステップS1202）。続けて、CPU#0は、メモリ管理スレッド122に実行要求を通知し（ステップS1203）、メモリ管理要求処理を終了する。

【0068】

図13は、メモリ管理スレッドの処理手順の一例を示すフローチャートである。メモリ管理スレッド122は、後述する図14で示すスケジューラ302の割当先CPU選択処

50

理によって、CPU#0、CPU#1のいずれかで実行する。図13では、メモリ管理スレッド122はCPU#0に実行されることを想定する。

【0069】

CPU#0は、メモリ管理スレッド122が定期実行されたか否かを判断する（ステップS1301）。定期実行された場合（ステップS1301：Yes）、CPU#0は、各デバイスのメモリ管理部に使用状況、空き状況を問い合わせる（ステップS1302）。CPU#0は、各デバイスのメモリ管理部として問い合わせを受け付けた後、メモリ管理スレッド122に使用状況、空き状況を通知する（ステップS1303）。通知を受け付けたCPU#0は、通知結果から、メモリ使用テーブル421を更新し（ステップS1304）、メモリ管理スレッド122の処理を終了する。

10

【0070】

定期実行されていない場合（ステップS1301：No）、CPU#0は、管理要求ディスパッチテーブル123の読み出す（ステップS1305）。なお、管理要求ディスパッチテーブル123には、複数のレコードが格納されている可能性がある。メモリ確保要求が複数存在する場合、CPU#0は、対応するスレッドの優先度が低いメモリ確保要求から順に読み出してもよい。

【0071】

優先度が低いスレッドのメモリを確保した後、優先度が高いスレッドのメモリを確保することで、CPU#0は、高いスレッドのメモリが確保できない場合、低いスレッドのメモリをフラッシュROM206、フラッシュROM208にスワップアウトする。このように、確保領域を再配置することで、マルチコアプロセッサシステム100は、記憶領域が断片化を防ぐことができる。もし、優先度が高いスレッドのメモリから順に確保した場合、優先度の低いスレッドからは、優先度の高いスレッドに対して確保された領域を移動することはできず、記憶領域が断片化するおそれがある。

20

【0072】

レコードの読出後、CPU#0は、読み出したレコードにメモリ確保要求が存在するかどうかを判断する（ステップS1306）。メモリ確保要求が存在する場合（ステップS1306：Yes）、CPU#0は、メモリ確保先選択処理を実行する（ステップS1307）。なお、メモリ確保先選択処理は、図15にて後述する。

【0073】

メモリ確保先選択処理実行後、または、メモリ確保要求が存在しない場合（ステップS1306：No）、CPU#0は、対象となるデバイスのメモリ管理部を実行する（ステップS1308）。対象となるデバイスのメモリ管理部の実行処理として、たとえば、メモリ管理要求がメモリ確保要求であれば、CPU#0は対象となるメモリに対してメモリ確保を行う。また、対象となるメモリとは、後述するステップS1504、またはステップS1505で選択されたメモリとなる。たとえば、メインメモリ103が確保先として選択された場合、メインメモリ103が対象となるメモリとなる。

30

【0074】

続けて、CPU#0は、読み出した管理要求を処理完了テーブルに移行し（ステップS1309）、スレッド動作履歴テーブル422、負荷情報、メモリ使用テーブル421を更新し（ステップS1310）、メモリ管理スレッド122の処理を終了する。なお、処理完了テーブルとは、処理が完了したメモリ管理要求を戻り値と併せて格納するテーブルである。ダミーデバイスドライバ121は、処理完了テーブル内のメモリ管理要求に対する戻り値を参照して、呼び元のアプリにメモリ管理要求に対する応答を行う。

40

【0075】

図14は、スケジューラによる割当対象スレッドの割当先CPU選択処理の一例を示すフローチャートである。スケジューラ302は、本実施の形態ではCPU#0で実行されることを想定している。なお、割当対象スレッドとなり得るスレッドは、ユーザスレッド、メモリ管理スレッド122などである。

【0076】

50

CPU#0は、負荷均衡状態が均衡か否かを判断する(ステップS1401)。負荷が均衡である場合(ステップS1401:Yes)、CPU#0は、割当対象スレッドが過去に動作したCPUに割当対象スレッドを割り当てる(ステップS1402)。負荷が均衡でない場合(ステップS1401:No)、CPU#0は、低負荷CPUに割当対象スレッドを割り当てる(ステップS1402)。ステップS1402、ステップS1403の終了後、CPU#0は、スレッドの割当先CPU選択処理を終了する。

【0077】

図15は、メモリ確保先選択処理の一例を示すフローチャートである。メモリ確保先選択処理は、メモリ管理スレッド122と同一のCPUで実行される。ここでは、CPU#0が実行することを想定する。CPU#0は、メモリ管理スレッド122の呼び元がユーザスレッドか否かを判断する(ステップS1501)。

10

【0078】

ユーザスレッドである場合(ステップS1501:ユーザスレッド)、CPU#0は、続けて、ユーザスレッドが高優先度か否かを判断する(ステップS1502)。低優先度、中優先度である場合(ステップS1502:低優先度、中優先度)、CPU#0は、スレッド動作履歴テーブル422を参照して、周辺メモリに空きがあるか否かを判断する(ステップS1503)。

【0079】

空きがない場合(ステップS1503:No)、または、高優先度である場合(ステップS1502:高優先度)、CPU#0は、メインメモリ103と周辺メモリのうち、メインメモリ103を確保先として選択する(ステップS1504)。確保先を選択後、CPU#0は、メモリ確保先選択処理を終了する。空きがある場合(ステップS1503:Yes)、または、呼び元がデバイスドライバである場合(ステップS1501:デバイスドライバ)、CPU#0は、メインメモリ103と周辺メモリのうち、周辺メモリを確保先として選択する(ステップS1505)。確保先を選択後、CPU#0は、メモリ確保先選択処理を終了する。

20

【0080】

図16は、本実施の形態にかかるコンピュータを用いたシステムの適用例を示す説明図である。図16において、ネットワークNWは、サーバ1601、サーバ1602とクライアント1631～クライアント1634とが通信可能なネットワークであり、たとえば、LAN、WAN、インターネット、携帯電話網などを含む。

30

【0081】

サーバ1602は、クラウド1620を有するサーバ群(サーバ1621～サーバ1625)の管理サーバである。クライアント1631はノート型PC(Personal Computer)である。クライアント1632はデスクトップ型PC、クライアント1633は携帯電話機である。携帯電話機として、クライアント1633は、スマートフォンであってもよいし、PHS(Personal Handyphone System)であってもよい。クライアント1634はタブレット型端末である。

【0082】

図16のサーバ1601、サーバ1602、サーバ1621～サーバ1625、クライアント1631～クライアント1634は、たとえば、実施の形態で説明したデータ処理装置として、本実施の形態にかかるデータ処理システムを実行する。たとえば、サーバ1621が最も高速なメモリを有し、サーバ1622～サーバ1625が低速なメモリを有している場合を想定する。このとき、データ処理システムは、サーバ1621のメモリを本実施の形態におけるメインメモリ103とみなし、サーバ1622～サーバ1625のメモリを周辺装置のメモリとみなすことで、本実施の形態にかかるデータ処理方法を実行することができる。

40

【0083】

以上説明したように、データ処理システム、およびデータ処理方法によれば、複数のデータ処理装置で実行中のスレッドからのメモリ管理要求を共有メモリに格納し、管理要求

50

を順に読み出し、メインメモリや周辺装置のメモリを確保する。これにより、データ処理システム、複数のデータ処理装置からの、複数アクセスによる競合を発生させずにメモリを管理することができる。具体的には、データ処理装置は、共有メモリから順に読み出すため、メモリ管理要求が同じタイミングで呼び出されることがなく、アクセス競合が発生しない。

【0084】

また、データ処理システムは、データ処理装置の負荷が均衡状態にない場合、メモリ管理を行うメモリ管理スレッドを、データ処理装置のうちで負荷の最も小さいデータ処理装置に割り当ててもよい。これにより、データ処理システムは、アクセス競合を起こさずにメモリ管理処理を負荷分散することができる。従来例におけるデータ処理システムでは、任意のデータ処理装置で複数のメモリを管理しようとする排他制御処理をせねばならず、オーバーヘッドが増加していた。また、従来例におけるデータ処理システムは、1つのデータ処理装置で複数のメモリの管理を行うとすると、負荷が偏ってしまっていた。

10

【0085】

また、データ処理システムは、データ処理装置の負荷が均衡状態にある場合、メモリ管理スレッドを、該当のスレッドが過去に割り当てられたデータ処理装置に割り当ててもよい。これにより、メモリ管理スレッドが割り当てられたデータ処理装置は、キャッシュメモリに残ったスレッドのキャッシュを用いて、高速に処理を行うことができる。

【0086】

具体的に、メモリ管理スレッドは、プログラムとしてのメモリ管理ルーチンと、データとして確保したメモリ領域のアロケートテーブルを有する。メモリ管理ルーチンがインストラクションキャッシュに残っていた場合、メモリ管理スレッドは、即座に処理を実行することができる。また、アロケートテーブルがデータキャッシュに残っていた場合、メモリ管理スレッドは、テーブル管理を即座に実行することができる。

20

【0087】

また、データ処理システムは、メモリ確保要求を行ったスレッドの優先度情報に基づいて、メインメモリまたは周辺メモリのいずれかから領域を確保してもよい。また、データ処理システムは、メモリ確保要求を行ったスレッドが高優先度である場合、メモリアクセス速度が速いメモリの領域を確保してもよい。これにより、データ処理システムは、高速処理を求められるスレッドに対しては、高速アクセスが行えるメモリを使用して、高速処理を行うことができる。

30

【0088】

また、データ処理システムは、メモリ確保要求を行ったスレッドが中優先度または低優先度である場合、周辺メモリに空きがあれば、周辺メモリの領域を確保してもよい。周辺メモリは、メインメモリより低速である分、消費電力が小さい。したがって、データ処理システムは、高速処理を行わなくてよい場合、消費電力の小さい周辺メモリを確保することで、消費電力を削減し、電力効率を向上することができる。

【0089】

また、データ処理システムは、メモリ確保要求を行ったスレッドが周辺装置を管理するスレッドであれば、周辺メモリの領域を確保してもよい。これにより、データ処理システムは、周辺装置を用いるスレッドであっても、周辺装置に対応するスレッドを実行することができる。

40

【0090】

また、データ処理システムは、メモリおよび周辺メモリの使用状況を収集してもよい。これにより、データ処理システムは、各周辺メモリの空き容量を確認できるため、メモリに空きがあり、低速なメモリから確保していくことができる。

【0091】

なお、本実施の形態で説明したデータ処理方法は、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等のコンピュータで実行することにより実現することができる。本データ処理方法を実行するプログラムは、ハードディスク、フレキシ

50

ブルディスク、CD-ROM、MO、DVD等のコンピュータで読み取り可能な記録媒体に記録され、コンピュータによって記録媒体から読み出されることによって実行される。また本データ処理方法を実行するプログラムは、インターネット等のネットワークを介して配布してもよい。

【符号の説明】

【0092】

#0、#1 CPU

100 マルチコアプロセッサシステム

101 GPU

102 GPUメモリ

10

103 メインメモリ

122 メモリ管理スレッド

123 管理要求ディスパッチテーブル

201 DSP

202 DSPメモリ

401 格納部

402 判断部

403 割当部

404 上位メモリ管理部

405 メモリ管理部

20

411 更新部

412 読出部

413 選択部

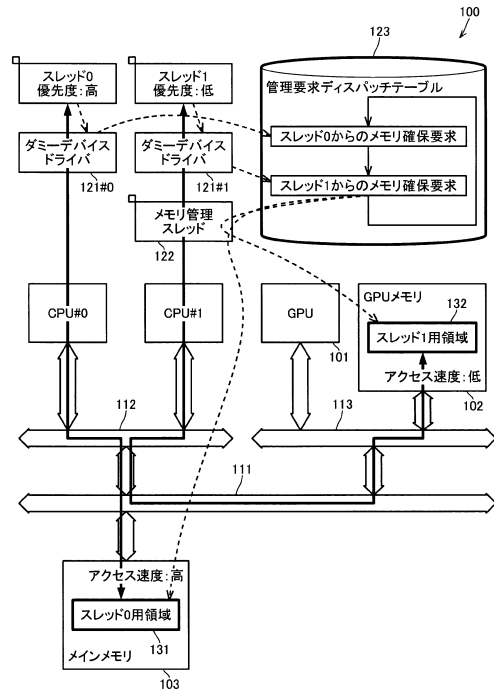
414 確保部

415 通知部

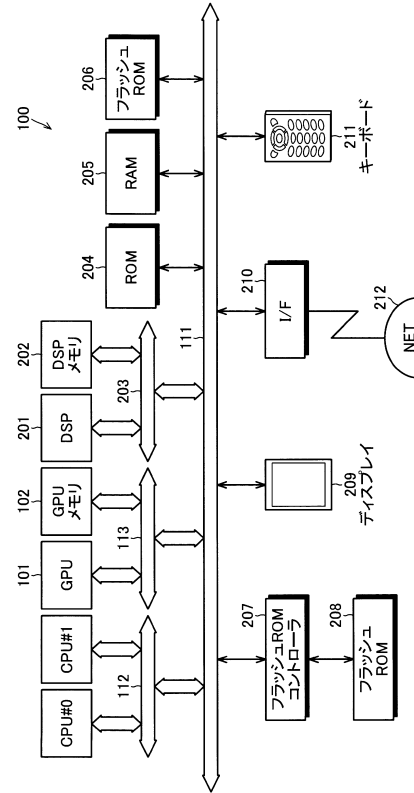
421 メモリ使用テーブル

422 スレッド動作履歴テーブル

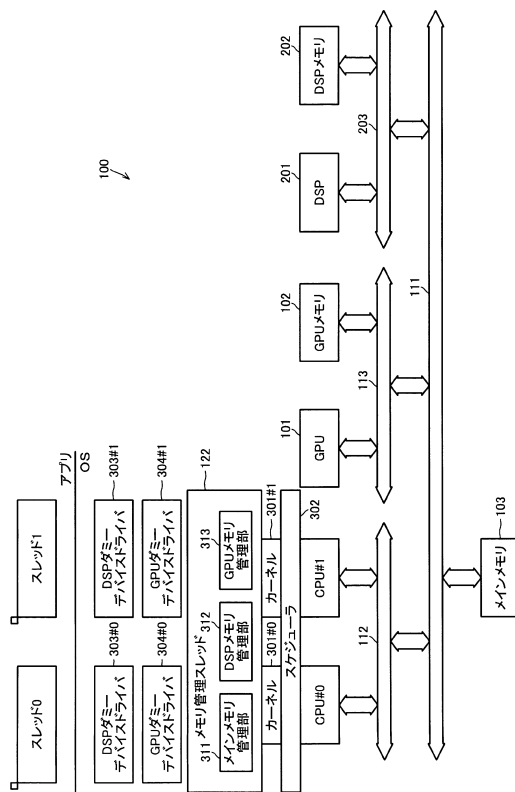
【図 1】



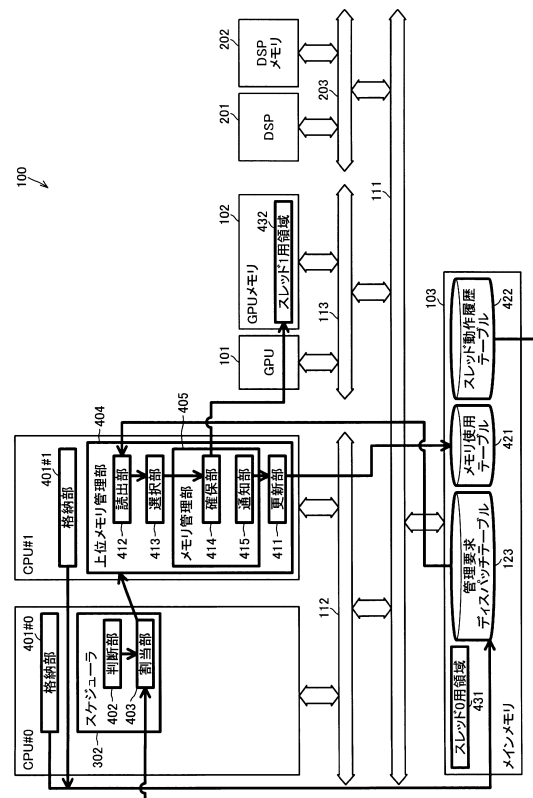
【図 2】



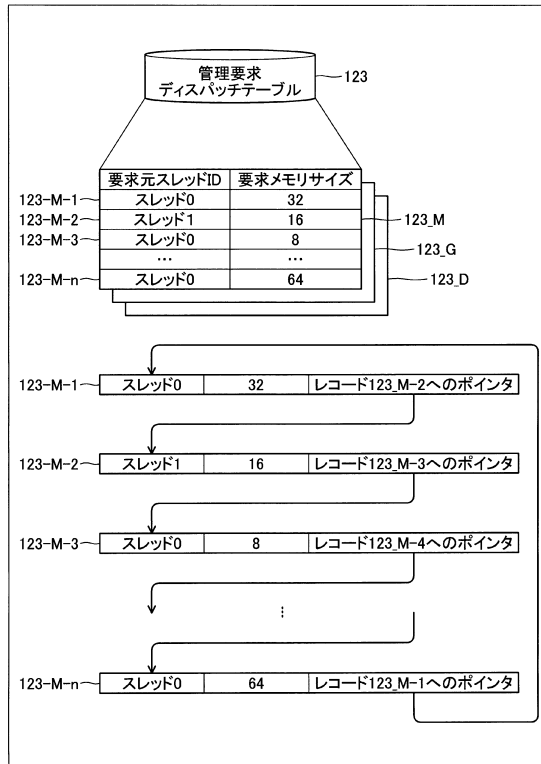
【図 3】



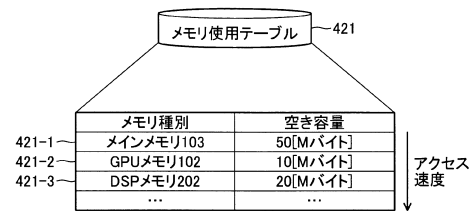
【図 4】



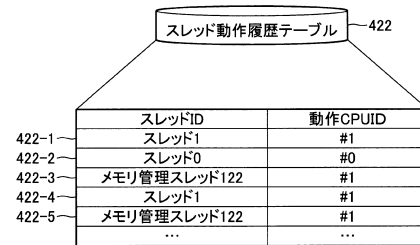
【図 5】



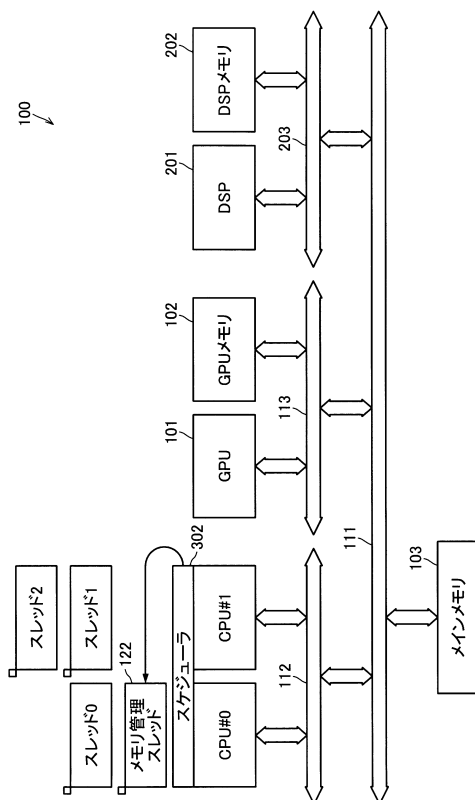
【図 6】



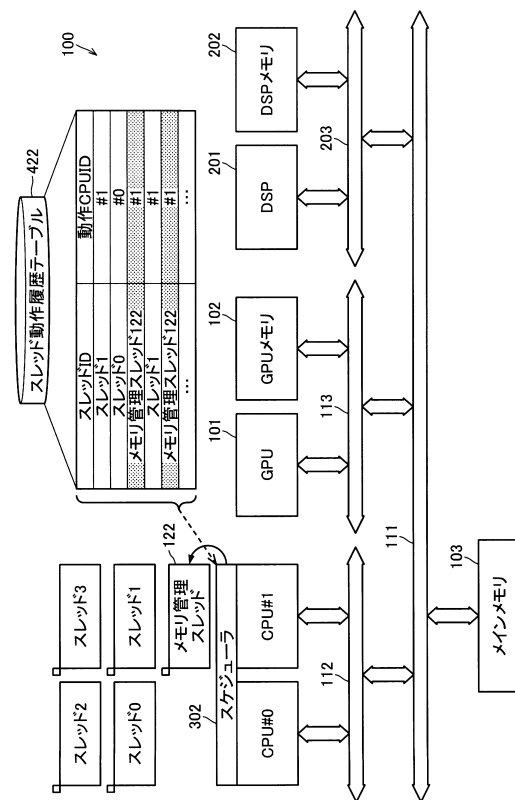
【図 7】



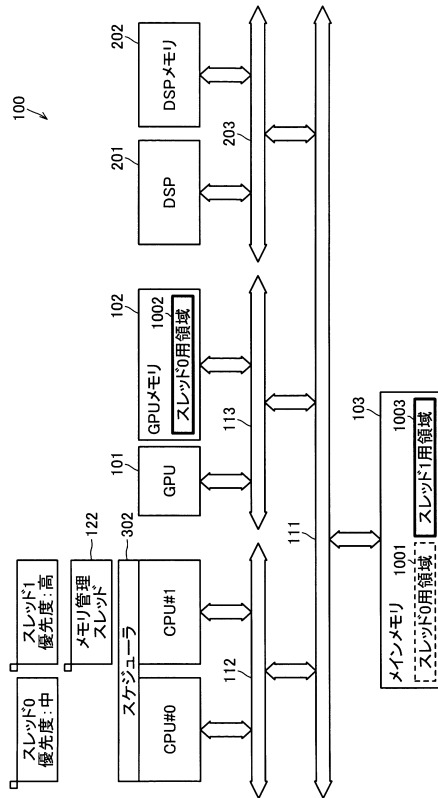
【図 8】



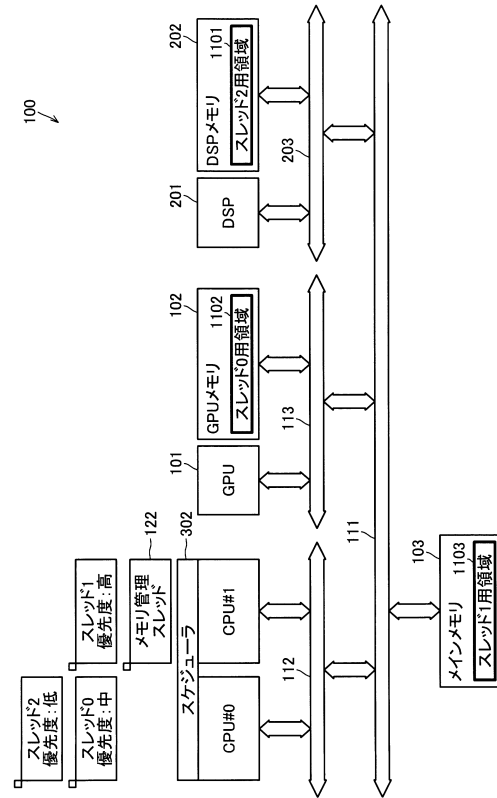
【図 9】



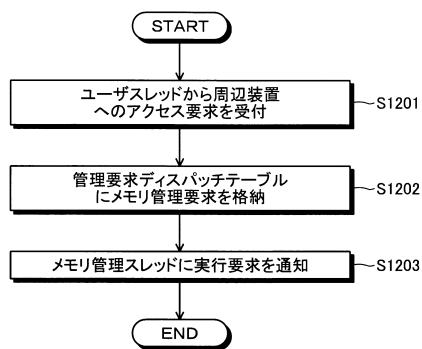
【図 10】



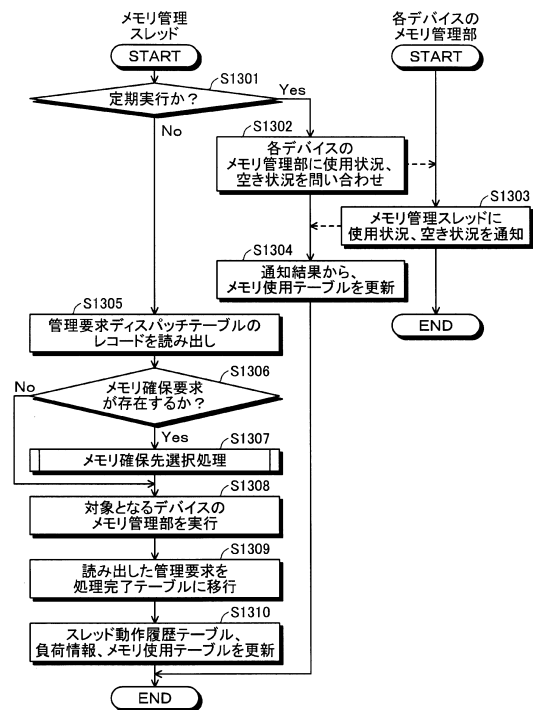
【図 11】



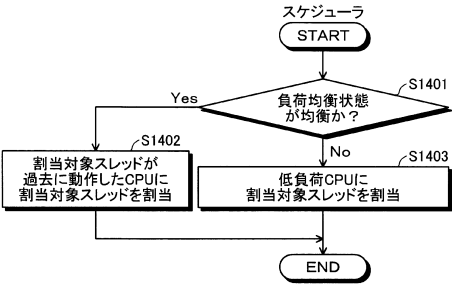
【図 12】



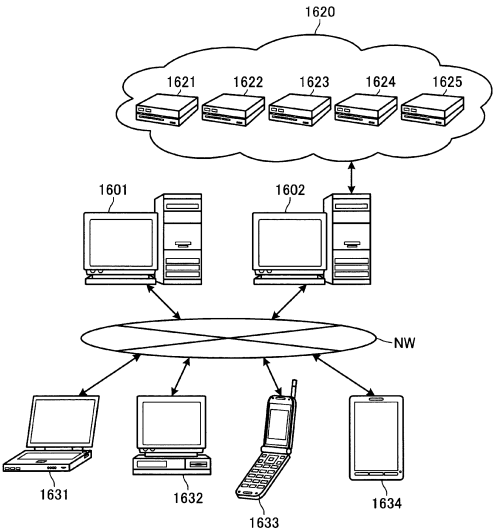
【図 13】



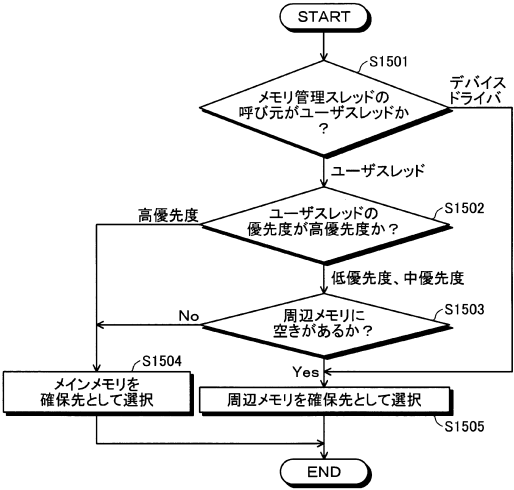
【図 1 4】



【図 1 6】



【図 1 5】



フロントページの続き

(72)発明者 栗原 康志
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 滝谷 亮一

(56)参考文献 特開2007-323393 (JP, A)
特開2004-054933 (JP, A)
特開昭63-184601 (JP, A)
国際公開第2007/017932 (WO, A1)
特開平10-187527 (JP, A)
特表2008-515838 (JP, A)
特開2000-235502 (JP, A)

(58)調査した分野(Int.Cl., DB名)
G06F 12/02
G06F 9/50