(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5780296号
(P5780296)
(45)発行日　平成27年9月16日(2015.9.16)　(24)登録日　平成27年7月24日(2015.7.24)

(51)Int.Cl.　　F I
G06F 9/52 (2006.01)　　G06F 9/46 475A
G06F 9/54 (2006.01)　　G06F 9/46 480A

請求項の数 7　(全 19 頁)

(21)出願番号　特願2013-510763(P2013-510763)
(86)(22)出願日　平成23年4月18日(2011.4.18)
(86)国際出願番号　PCT/JP2011/059571
(87)国際公開番号　WO2012/144012
(87)国際公開日　平成24年10月26日(2012.10.26)
審査請求日　平成25年10月24日(2013.10.24)

(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　100104190
弁理士　酒井　昭徳
(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　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　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】スレッド処理方法、およびスレッド処理システム

(57)【特許請求の範囲】
【請求項１】
特定の装置が、
前記特定の装置および複数の装置で共有される共有メモリにアクセスする処理と同期処理とを行う複数のスレッドのうちの一のスレッドを、前記複数の装置の各々の装置に割り当て、
前記各々の装置に対応して前記各々の装置に割り当てられたスレッドの実行結果を受け付けるときの第１時刻情報を取得し、
前記各々の装置に対応する第１時刻情報が大きい順に、前記共有メモリにアクセスする処理と前記同期処理とを行う新たなスレッドを前記各々の装置が実行する際の前記各々の装置の前記共有メモリへのアクセス権の優先度を高く設定すること
を特徴とするスレッド処理方法。
【請求項２】
前記特定の装置が、
前記各々の装置の前記共有メモリへのアクセス権の優先度に関する情報を前記各々の装置に通知すること
を特徴とする請求項１に記載のスレッド処理方法。
【請求項３】
前記特定の装置が、
前記複数の装置のうちの実行結果を最後に受け付けた装置に割り当てられた前記共有メ

モリにアクセスする処理と前記同期処理とを行うスレッドの実行結果を受け付けるときの第２時刻情報と前記各々の装置に対応する第１時刻情報との差を、前記各々の装置が同期待ち状態となった同期待ち時間として算出し、
前記各々の装置に対応する同期待ち時間の短い順に、前記各々の装置の前記共有メモリへのアクセス権の優先度を高く設定すること
を特徴とする請求項１または請求項２に記載のスレッド処理方法。
【請求項４】
前記特定の装置が、
前記各々の装置に対応する同期待ち時間が短いとき前記各々の装置の前記共有メモリへのアクセス権の優先度を高く設定すること
を特徴とする請求項３に記載のスレッド処理方法。
【請求項５】
特定の装置と複数の装置とを含むスレッド処理システムであって、
前記特定の装置は、
前記特定の装置および前記複数の装置で共有される共有メモリにアクセスする処理と同期処理とを行う複数のスレッドのうちの一のスレッドを、前記複数の装置の各々の装置に割り当て、
前記各々の装置に対応して前記各々の装置に割り当てられたスレッドの実行結果を受け付けるときの第１時刻情報を取得し、
前記各々の装置に対応する第１時刻情報が大きい順に、前記共有メモリにアクセスする処理と前記同期処理とを行う新たなスレッドを前記各々の装置が実行する際の前記各々の装置の前記共有メモリへのアクセス権の優先度を高く設定すること
を特徴とするスレッド処理システム。
【請求項６】
前記特定の装置は、
前記各々の装置の前記共有メモリへのアクセス権の優先度を前記各々の装置に通知すること
を特徴とする請求項５に記載のスレッド処理システム。
【請求項７】
前記各々の装置は、
前記共有メモリへのアクセスを制御するメモリコントローラを前記各々の装置の前記共有メモリへのアクセス権の優先度に応じて制御すること
を特徴とする請求項５または請求項６に記載のスレッド処理システム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、分散処理を行うスレッド処理方法、およびスレッド処理システムに関する。
【背景技術】
【０００２】
従来から、複数の端末装置でデータを共有する技術として、各端末装置の実メモリに分散された分割されたデータを保持する仮想分散共有メモリという技術が開示されている。仮想分散共有メモリは、たとえば、インターネットなどの広域のネットワーク上にある装置を、一つの複合したコンピュータシステムとして提供するグリッド・コンピューティングにて使用されている。
【０００３】
仮想分散共有メモリの特徴として、アクセスの分散が図られ、ネットワークの負荷を軽減させることができる。また、端末装置で実行されるアプリケーションソフトウェア（以下、アプリ）が仮想分散共有メモリを意識することなく、一つの連続したメモリとしてアクセスすることが可能である。また、複数の端末装置で実行されるアプリは、各端末装置にスレッドを割り当てて分散処理、並列処理を実行する。なお、スレッドとは、プログラ

ムの実行単位である。
【０００４】
仮想分散共有メモリに関する技術として、たとえば、有線による広域ネットワークで接続された装置のメモリを、仮想分散共有メモリとして使用する技術が開示されている。または、無線ネットワークの例にて、送受信バッファを仮想化することで、データをリレーし、直接接続されていない端末装置でデータを送受信できる技術が開示されている（たとえば、下記特許文献１、２を参照。）。
【０００５】
また、仮想分散共有メモリを用いた情報ストレージシステムにおいて、利用者が生産する情報を意味付けした情報要素として蓄積する。続けて、複数の端末装置からの情報取得要求があった場合、動的に情報取得要求を処理するマスタ端末装置を決定して、情報取得要求に対する情報を効率的に交換、または複製する技術が開示されている（たとえば、下記特許文献３を参照。）。
【先行技術文献】
【特許文献】
【０００６】
【特許文献１】特開平６－１９７８５号公報
【特許文献２】特開２００５－９４７７６号公報
【特許文献３】特開２００５－４４７４号公報
【発明の概要】
【発明が解決しようとする課題】
【０００７】
上述した従来技術において、複数の分散処理、並列処理を実行する場合、複数の端末装置で同期処理を行いながら処理を実行することがある。同期処理とは、複数のスレッドの処理を特定のコード位置までで一旦停止させ、全てのスレッドが特定のコード位置に到達した際に、次の処理を継続する処理である。停止中となったスレッドを実行している端末装置は、同期待ちの状態となる。このとき、各スレッドが割り当てられた端末装置の処理性能や通信速度が異なる場合、処理性能、通信速度が高い端末装置での同期待ちの時間が長くなり、システム全体の処理性能が低下するという問題があった。
【０００８】
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、同期処理に伴うシステム全体の処理性能の低下を抑止するスレッド処理方法、およびスレッド処理システムを提供することを目的とする。
【課題を解決するための手段】
【０００９】
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、特定の装置は複数の装置のそれぞれに複数のスレッドのうちの一のスレッドを割り当て、複数の装置のそれぞれから対応するスレッドの実行結果を受け付けるときの第１時刻情報を取得し、複数の装置から複数のスレッドの実行結果を受け付け終わったときの第２時刻情報と第１時刻情報とに基づいて、複数の装置の特定の装置および複数の装置で共有される共有メモリへのアクセス権の優先度を設定するスレッド処理方法、およびスレッド処理システムが提案される。
【発明の効果】
【００１０】
本発明の一側面によれば、同期処理に伴うシステム全体の処理性能の低下を抑止できるという効果を奏する。
【図面の簡単な説明】
【００１１】
【図１】図１は、スレッド処理システム１００の動作例を示す説明図である。
【図２】図２は、スレッド処理システム１００内の接続例を示す説明図である。

【図３】図３は、実施の形態にかかる端末装置１０３＃０のハードウェアを示すブロック図である。
【図４】図４は、スレッド処理システム１００の機能例を示すブロック図である。
【図５】図５は、同期待ち開始時刻テーブル４２１の記憶内容の一例を示す説明図である。
【図６】図６は、アクセス権優先度情報４２３の生成例を示す説明図である。
【図７】図７は、アクセス権優先度情報が設定された場合のスレッド処理システム１００の動作例を示す説明図である。
【図８】図８は、マスタ端末装置によるスレッド実行時の処理手順の一例を示すフローチャート（その１）である。
【図９】図９は、マスタ端末装置によるスレッド実行時の処理手順の一例を示すフローチャート（その２）である。
【図１０】図１０は、スレーブ端末装置によるスレッド実行時の処理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】
【００１２】
以下に添付図面を参照して、開示のスレッド処理方法、およびスレッド処理システムの実施の形態を詳細に説明する。
【００１３】
図１は、スレッド処理システム１００の動作例を示す説明図である。符号１０１で示す説明図は、共有メモリへのアクセス権の優先度が設定されていない状態でのスレッド処理システム１００の動作例を示している。また、符号１０２で示す説明図は、共有メモリへのアクセス権の優先度が設定されている状態でのスレッド処理システム１００の動作例を示している。
【００１４】
スレッド処理システム１００は、携帯電話や携帯端末等といった端末装置１０３を複数含む。たとえば、図１で示すスレッド処理システム１００は、端末装置１０３＃０～端末装置１０３＃２を含む。以下、接尾記号“＃ｎ”が付随された記号は、ｎ番目の端末装置１０３に対応する記号であることを示している。端末装置１０３＃０～端末装置１０３＃２は、無線ＬＡＮ（Ｌｏｃａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）、Ｂｌｕｅｔｏｏｔｈ（登録商標）等の無線通信によって接続されている。また、端末装置１０３＃０～端末装置１０３＃２は、仮想共有メモリ１０４にアクセス可能である。端末装置１０３＃０～端末装置１０３＃２は、仮想共有メモリ１０４にアクセスすることで複数のスレッドを同期しつつ実行する。なお、仮想共有メモリ１０４の格納箇所は、端末装置１０３＃０～端末装置１０３＃２内の記憶領域の一部となる。
【００１５】
また、スレッド処理システム１００全体を制御するマスタ端末装置が端末装置１０３＃０となり、マスタ端末装置によって制御されるスレーブ端末装置が端末装置１０３＃１、端末装置１０３＃２であることを想定する。また、端末装置１０３＃２は高処理性能であり、端末装置１０３＃１は低処理性能であり、端末装置１０３＃０は、端末装置１０３＃１と端末装置１０３＃２の中間の処理性能を有していることを想定する。
【００１６】
このような状態で、符号１０１で示す説明図では、端末装置１０３＃０が、時刻ｔ０以降にて、端末装置１０３＃１にスレッド１を割り当て、端末装置１０３＃２にスレッド２を割り当てる。端末装置１０３＃１、端末装置１０３＃２は、割り当てられたスレッドを実行する。また、端末装置１０３＃０は、自身にもスレッド０を割り当てて実行する。端末装置１０３＃０～端末装置１０３＃２は、端末装置間で同期を取るために、スレッド０～スレッド２が全て終了する時刻ｔ１まで待機する。
【００１７】
高性能である端末装置１０３＃２はスレッド２を早く終了し、実行結果を端末装置１０

３＃０に通知した後、同期待ち状態となる。端末装置１０３＃２による通知後、端末装置１０３＃０は、スレッド２の実行結果を受け付ける。続けて、中性能である端末装置１０３＃０がスレッド０を終了する。最後に、低性能である端末装置１０３＃１がスレッド１を終了し、実行結果を端末装置１０３＃０に通知する。端末装置１０３＃１による通知後、端末装置１０３＃０は、スレッド１の実行結果を受け付ける。

【００１８】

結果、符号１０１で示すスレッド処理システム１００は、端末装置１０３＃１がボトルネックとなったため、端末装置１０３＃２の同期待ち時間が長くなり、スレッド処理システム１００全体の処理性能が低下している状態である。また、端末装置１０３＃１は、同期待ち時間が短くなり、端末装置１０３＃０の同期待ち時間は、端末装置１０３＃１と端末装置１０３＃２の中間となる。

【００１９】

次に、符号１０２で示す説明図は、仮想共有メモリ１０４へのアクセス権の優先度を端末装置１０３＃０～端末装置１０３＃２に設定した状態である。具体的には、符号１０１で示すスレッド処理システム１００にて実行結果の通知が遅かった端末装置１０３＃１を、スレッド処理システム１００は、優先度：高に設定する。また、スレッド処理システム１００は、実行終了の時刻が中程度となった端末装置１０３＃０を優先度：中に設定し、実行結果の通知が早かった端末装置１０３＃０を優先度：低に設定する。

【００２０】

このようなアクセス権の優先度を設定することで、たとえば、端末装置１０３＃１と端末装置１０３＃３が同時に仮想共有メモリ１０４にアクセスした場合、端末装置１０３＃１が優先して仮想共有メモリ１０４にアクセスするようになる。これにより、符号１０２で示す端末装置１０３＃１は、符号１０１で示す端末装置１０３＃１より、スレッド１を早く終了することができる。スレッド１が早く終了することで、同期待ちが終了する時刻がｔ１からｔ１’となり、端末装置１０３＃０、端末装置１０３＃２の同期待ち時間が短縮されるため、スレッド処理システム１００の処理性能が向上することになる。

【００２１】

図２は、スレッド処理システム１００内の接続例を示す説明図である。図２で示すスレッド処理システム１００は、図１で示した端末装置１０３＃０～端末装置１０３＃２に加えて、端末装置１０３＃３を含む。端末装置１０３＃３は、端末装置１０３＃２と同等の処理性能となることを想定する。

【００２２】

また、端末装置１０３＃０～端末装置１０３＃３は、無線通信で接続されている。端末装置１０３＃０と端末装置１０３＃２間、端末装置１０３＃０と端末装置１０３＃３間、端末装置１０３＃２と端末装置１０３＃３間は、無線ＬＡＮといった高速通信で接続されている。また、端末装置１０３＃０と端末装置１０３＃１間、端末装置１０３＃１と端末装置１０３＃２間、端末装置１０３＃１と端末装置１０３＃３間は、Ｂｌｕｅｔｏｏｔｈ（登録商標）といった低速通信で接続されている。

【００２３】

図３は、実施の形態にかかる端末装置１０３＃０のハードウェアを示すブロック図である。なお、端末装置１０３＃１～端末装置１０３＃３は、端末装置１０３＃０と処理性能が異なるが同一のハードウェアを含むため、説明を省略する。また、図３の説明に登場するハードウェアは、全て端末装置１０３＃０に含まれているため、説明の簡略化のため接尾記号“＃０”を省略する。

【００２４】

図３において、端末装置１０３＃０は、ＣＰＵ３０１と、ＲＯＭ（Ｒｅａｄ‐Ｏｎｌｙ Ｍｅｍｏｒｙ）３０２と、ＲＡＭ（Ｒａｎｄｏｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）３０３と、を含む。また、端末装置１０３＃０は、フラッシュＲＯＭ３０４と、フラッシュＲＯＭコントローラ３０５と、フラッシュＲＯＭ３０６と、を含む。また、端末装置１０３＃０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３０７と、Ｉ／Ｆ（

Ｉｎｔｅｒｆａｃｅ）３０８と、キーボード３０９と、を含む。また、各部はバス３１０によってそれぞれ接続されている。

【００２５】

ここで、ＣＰＵ３０１は、端末装置１０３＃０の全体の制御を司る。また、ＣＰＵ３０１は、専用のキャッシュメモリを有してもよい。また、端末装置１０３は、複数のコアを含むマルチコアプロセッサシステムであってもよい。なお、マルチコアプロセッサシステムとは、コアが複数搭載されたプロセッサを含むコンピュータのシステムである。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。

【００２６】

ＲＯＭ３０２は、ブートプログラムなどのプログラムを記憶している。ＲＡＭ３０３は、ＣＰＵ３０１のワークエリアとして使用される。また、ＲＡＭ３０３とバス３１０は、メモリコントローラ３１１で接続されている。メモリコントローラ３１１は、ＣＰＵ３０１によるＲＡＭ３０３へのアクセスの制御をする。また、メモリコントローラ３１１は、ＲＡＭ３０３以外にも、ＲＯＭ３０２、フラッシュＲＯＭ３０４などのアクセスを制御してもよい。

【００２７】

フラッシュＲＯＭ３０４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。たとえば、フラッシュＲＯＭ３０４は、ＯＳ（Ｏｐｅｒａｔｉｎｇ　Ｓｙｓｔｅｍ）などのシステムソフトウェアやアプリケーションソフトウェアなどを記憶している。たとえば、ＯＳを更新する場合、端末装置１０３＃０は、Ｉ／Ｆ３０８によって新しいＯＳを受信し、フラッシュＲＯＭ３０４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。

【００２８】

フラッシュＲＯＭコントローラ３０５は、ＣＰＵ３０１の制御に従ってフラッシュＲＯＭ３０６に対するデータのリード／ライトを制御する。フラッシュＲＯＭ３０６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ＲＯＭ３０６は、フラッシュＲＯＭコントローラ３０５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、端末装置１０３＃０を使用するユーザがＩ／Ｆ３０８を通して取得した画像データ、映像データや、また本実施の形態にかかるスレッド処理方法を実行するプログラムなどである。フラッシュＲＯＭ３０６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。

【００２９】

ディスプレイ３０７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３０７は、たとえば、ＴＦＴ（Ｔｈｉｎ　Ｆｉｌｍ　Ｔｒａｎｓｉｓｔｏｒ）液晶ディスプレイなどを採用することができる。

【００３０】

Ｉ／Ｆ３０８は、通信回線を通じてＬＡＮ、ＷＡＮ（Ｗｉｄｅ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）、インターネットなどのネットワーク３１２に接続され、ネットワーク３１２を介して他の装置に接続される。そして、Ｉ／Ｆ３０８は、ネットワーク３１２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／Ｆ３０８には、たとえばモデムやＬＡＮアダプタなどを採用することができる。

【００３１】

キーボード３０９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３０９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。

【００３２】

（スレッド処理システム１００の機能）

次に、スレッド処理システム１００の機能例について説明する。図４は、スレッド処理

システム１００の機能例を示すブロック図である。スレッド処理システム１００は、検出部４０１と、割当部４０２と、実行部４０３と、取得部４０４と、登録部４０５と、算出部４０６と、設定部４０７と、通知部４０８と、制御部４０９と、管理部４１０と、を含む。さらに、スレッド処理システム１００は、管理部４１１と、受付部４１２と、実行部４１３と、制御部４１４と、を含む。

【００３３】

この制御部となる機能（検出部４０１～制御部４１４）は、記憶装置に記憶されたプログラムをＣＰＵ３０１＃０、ＣＰＵ３０１＃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３に示したＲＯＭ３０２、ＲＡＭ３０３、フラッシュＲＯＭ３０４、フラッシュＲＯＭ３０６などである。

【００３４】

また、スレッド処理システム１００は、同期待ち開始時刻テーブル４２１、共有メモリ４２２にアクセス可能である。さらに、スレッド処理システム１００は、アクセス権優先度情報４２３を生成する。また、ＣＰＵ３０１＃０は、ＯＳ４３１＃０、スケジューラ４３２を実行し、ＣＰＵ３０１＃１は、ＯＳ４３１＃１を実行する。さらに、スレッド処理システム１００は、アプリ４３３を実行しており、アプリ４３３内のスレッド０がＣＰＵ３０１＃０に割り当てられており、スレッド１がＣＰＵ３０１＃１に割り当てられている。

【００３５】

また、検出部４０１～管理部４１０は、マスタ端末装置となる端末装置１０３＃０の機能であり、管理部４１１～制御部４１４は、スレーブ端末装置となる端末装置１０３＃１の機能である。なお、端末装置１０３＃１がマスタ端末装置となる場合、端末装置１０３＃１は検出部４０１～管理部４１０を含む。また、端末装置１０３＃０がスレーブ端末装置となる場合、端末装置１０３＃０は管理部４１１～制御部４１４を含む。また、検出部４０１～実行部４０３は、スケジューラ４３２の機能に含まれる。なお、取得部４０４～制御部４０９がスケジューラ４３２の機能に含まれてもよい。

【００３６】

ＯＳ４３１は、端末装置１０３を制御するプログラムである。具体的に、ＯＳ４３１＃０は端末装置１０３＃０を制御しており、ＯＳ４３１＃１は端末装置１０３＃１を制御している。ＯＳ４３１は、たとえば、アプリ４３３が使用するライブラリを提供する。また、ＯＳ４３１は、フラッシュＲＯＭコントローラ３０５、Ｉ／Ｆ３０８、キーボード３０９等を制御するデバイスドライバを有している。

【００３７】

スケジューラ４３２は、スレッド処理システム１００で実行されているアプリ内のスレッドをＣＰＵ３０１に割り当てる順番を決定するプログラムである。また、本実施の形態にかかるスケジューラ４３２は、次に割り当てられることが決定したスレッドをＣＰＵ３０１に割り当てるディスパッチ機能が含まれていることを想定している。たとえば、スケジューラ４３２は、スレッド０をＣＰＵ３０１＃０に割り当て、スレッド１をＣＰＵ３０１＃１に割り当てる。

【００３８】

同期待ち開始時刻テーブル４２１は、複数の装置のそれぞれから対応するスレッドの実行結果を受け付けるときの第１時刻情報を記憶するテーブルである。同期待ち開始時刻テーブル４２１の具体的な記憶内容については、図５にて後述する。なお、同期待ち開始時刻テーブル４２１は、マスタ端末装置である端末装置１０３＃０のＲＡＭ３０３＃０内に記憶される。

【００３９】

共有メモリ４２２は、仮想共有メモリ１０４に対する実メモリである。共有メモリ４２２＃０、共有メモリ４２２＃１、また、他の端末装置１０３内にある共有メモリ４２２が結合され仮想共有メモリ１０４を形成する。なお、共有メモリ４２２＃０はＲＡＭ３０３＃０内に存在し、共有メモリ４２２＃１はＲＡＭ３０３＃１内に存在する。

【００４０】
アクセス権優先度情報４２３は、マスタ端末装置となる特定の装置にて、スレーブ端末装置となる複数の装置との仮想共有メモリ１０４へのアクセス権の優先度を記憶する情報である。アクセス権優先度情報４２３の生成例については、図６にて後述する。
【００４１】
検出部４０１は、新たなスレッドの割当要求が発生したことを検出する機能を有する。たとえば、検出部４０１は、実行中のアプリ４３３にて、新たなスレッドの割当要求が発生したことを検出する。具体的な検出方法として、実行中のアプリ４３３の実行コードに、新規スレッドを実行するＡＰＩが呼び出された場合、検出部４０１は、新たなスレッドの割当要求が発生したことを検出する。なお、検出された割当要求は、ＣＰＵ３０１＃０のレジスタ、キャッシュメモリ、ＲＡＭ３０３などの記憶領域に記憶される。
【００４２】
割当部４０２は、マスタ端末装置となる特定の装置にてスレーブ端末装置となる複数の装置のそれぞれに複数のスレッドのうちの一のスレッドを割り当てる機能を有する。また、割当部４０２は、マスタ端末装置にスレッドを割り当ててもよい。たとえば、割当部４０２は、スレッド０をＣＰＵ３０１＃０に割り当て、スレッド１をＣＰＵ３０１＃１に割り当てる。割当部４０２は、自装置以外にスレッドを割り当てる場合、割当を行う装置に対して、割当要求を通知する。なお、スレッドを割り当てるという情報は、ＣＰＵ３０１＃０のレジスタ、キャッシュメモリ、ＲＡＭ３０３などの記憶領域に記憶されてもよい。
【００４３】
実行部４０３、実行部４１３は、割り当てられたスレッドを実行する機能を有する。たとえば、実行部４０３はスレッド０を実行し、実行部４１３はスレッド１を実行する。なお、スレッドの実行結果は、ＣＰＵ３０１＃０のレジスタ、キャッシュメモリ、ＲＡＭ３０３などの記憶領域に記憶される。
【００４４】
取得部４０４は、複数の装置のそれぞれから対応するスレッドの実行結果を受け付けるときの第１時刻情報を取得する機能を有する。たとえば、スレッド０が実行開始した時刻を０［ミリ秒］とし、スレッド０が終了した時刻を１１０［ミリ秒］とする場合を想定する。このとき、取得部４０４は、スレッド０の実行結果を受け付けるときの第１時刻情報となる同期待ち開始時刻を１１０［ミリ秒］として取得する。なお、取得された同期待ち開始時刻は、ＣＰＵ３０１＃０のレジスタ、キャッシュメモリ、ＲＡＭ３０３などの記憶領域に記憶される。
【００４５】
登録部４０５は、複数の装置のそれぞれからスレッドの実行結果を受け付けるときの第１時刻情報を登録する機能を有する。たとえば、登録部４０５は、スレッド０の同期待ち開始時刻となった１１０［ミリ秒］を同期待ち開始時刻テーブル４２１に登録する。
【００４６】
算出部４０６は、第１時刻情報と複数の装置から複数のスレッドの実行結果を受け付け終わったときの第２時刻情報とに基づいて複数の装置のそれぞれの所定時間を算出する。ここで、所定時間は、対象の装置からの実行結果を受け付けた時刻から、全てのスレッドからの実行結果を受け付けた時刻までとなる同期待ち時間である。
【００４７】
たとえば、スレッド０が同期待ち開始時刻となった時刻が１１０［ミリ秒］であり、スレッド１が同期待ち開始時刻となった時刻が１９０［ミリ秒］であると想定する。また、スレッド１の実行結果を受け付けたときが、全てのスレッドの実行結果を受け付け終わったときであることを想定する。このとき、算出部４０６は、端末装置１０３＃０の同期待ち時間を、１９０－１１０＝８０［ミリ秒］として算出し、端末装置１０３＃１の同期待ち時間を、１９０－１９０＝０［ミリ秒］として算出する。なお、算出された同期待ち時間は、ＣＰＵ３０１＃０のレジスタ、キャッシュメモリ、ＲＡＭ３０３などの記憶領域に記憶される。

【００４８】
設定部４０７は、複数の装置の第１時刻情報と第２時刻情報に基づいて、特定の装置および複数の装置で共有される共有メモリ４２２へのアクセス権の優先度を設定する機能を有する。なお、共有メモリ４２２へのアクセス権の優先度と、共有メモリ４２２が結合された仮想共有メモリ１０４へのアクセス権の優先度は同一であるため、以降、仮想共有メモリ１０４へのアクセス権の優先度で統一して説明を行う。
【００４９】
たとえば、端末装置１０３＃０の第１時刻情報が１１０［ミリ秒］、端末装置１０３＃１の第１時刻情報が１９０［ミリ秒］であり、第２時刻情報が１９０［ミリ秒］である状態を想定する。たとえば、設定部４０７は、第１時刻情報が大きい順にアクセス権の優先度を設定する。この例では、設定部４０７は、端末装置１０３＃０の優先度を低く設定し、端末装置１０３＃１の優先度を高く設定する。
【００５０】
また、設定部４０７は、算出部４０６によって算出された所定時間に基づいてアクセス権の優先度を設定してもよい。たとえば、所定時間である同期待ち時間に関して、端末装置１０３＃０の同期待ち時間が８０［ミリ秒］、端末装置１０３＃１の同期待ち時間が０［ミリ秒］であることを想定する。このとき、設定部４０７は、同期待ち時間が短い順にアクセス権の優先度を設定する。この例でも、設定部４０７は、端末装置１０３＃０の優先度を低く設定し、端末装置１０３＃１の優先度を高く設定する。なお、設定された優先度は、アクセス権優先度情報４２３として生成される。アクセス権優先度情報４２３は、ＣＰＵ３０１＃０のレジスタ、キャッシュメモリ、ＲＡＭ３０３などの記憶領域に記憶される。
【００５１】
通知部４０８は、アクセス権の優先度に関する情報を複数の装置に通知する機能を有する。たとえば、通知部４０８は、設定部４０７によって生成されたアクセス権優先度情報４２３を端末装置１０３＃１に通知する。
【００５２】
制御部４０９、制御部４１４は、共有メモリ４２２へのアクセスを制御するメモリコントローラ３１１をアクセス権の優先度に応じて制御する機能を有する。たとえば、設定部４０７が端末装置１０３＃０の優先度を低く設定し、端末装置１０３＃１の優先度を高く設定した場合を想定する。
【００５３】
このとき、制御部４０９は、ＣＰＵ３０１＃０からの共有メモリ４２２＃０へのアクセス中に、Ｉ／Ｆ３０８＃０を経由したアクセス要求があった場合、ＣＰＵ３０１＃０からのアクセスを一旦停止させるようにメモリコントローラ３１１＃０を制御する。停止後、制御部４０９は、Ｉ／Ｆ３０８＃０を経由した共有メモリ４２２＃０へのアクセスを行う。具体的に、制御部４０９は、メモリコントローラ３１１＃０の設定レジスタに、各端末装置１０３の優先度を書き込むことで、メモリコントローラ３１１＃０の動作を制御する。同様に、制御部４１４は、受付部４１２によって受け付けたアクセス権の優先度に応じてメモリコントローラ３１１＃１を制御する。
【００５４】
管理部４１０、管理部４１１は、仮想共有メモリ１０４を管理する機能を有する。たとえば、管理部４１０は、スレッド０からの仮想共有メモリ１０４へのアクセスを、仮想共有メモリ１０４に対応する実メモリである共有メモリ４２２＃０、共有メモリ４２２＃１へのアクセスに変換する。共有メモリ４２２＃１へのアクセスに変換された場合、管理部４１０は、Ｉ／Ｆ３０８＃０を経由して端末装置１０３＃１にアクセス要求を通知する。
【００５５】
受付部４１２は、アクセス権の優先度を受け付ける機能を有する。たとえば、受付部４１２は、端末装置１０３＃０からアクセス権優先度情報４２３を受け付ける。さらに、受付部４１２は、スレッドの割当要求を受け付ける。なお、受け付けたアクセス権優先度情

報４２３、スレッドの割当要求の情報は、ＣＰＵ３０１＃１のレジスタ、キャッシュメモリ、ＲＡＭ３０３などの記憶領域に記憶される。

【００５６】

図５は、同期待ち開始時刻テーブル４２１の記憶内容の一例を示す説明図である。同期待ち開始時刻テーブル４２１は、端末装置ＩＤ（ＩＤｅｎｔｉｆｉｃａｔｉｏｎ）、同期待ち開始時刻という２つのフィールドを含む。端末装置ＩＤフィールドには、端末装置１０３＃０～端末装置１０３＃３の識別情報が格納される。同期待ち開始時刻フィールドには、端末装置１０３＃０～端末装置１０３＃３が同期待ちを開始した時刻が格納される。

【００５７】

たとえば、図５に示す同期待ち開始時刻テーブル４２１には、レコード４２１－０～レコード４２１－３が含まれる。また、図５では、同期処理を伴うスレッドを実行開始した時刻を０［ミリ秒］と想定している。このとき、レコード４２１－０は、端末装置１０３＃０の同期待ち開始時刻が１１０［ミリ秒］であり、レコード４２１－１は端末装置１０３＃１の同期待ち開始時刻が１９０［ミリ秒］であることを示している。同様に、レコード４２１－２は端末装置１０３＃２の同期待ち開始時刻が７０［ミリ秒］であり、レコード４２１－３は端末装置１０３＃３の同期待ち開始時刻が６０［ミリ秒］であることを示している。

【００５８】

図６は、アクセス権優先度情報４２３の生成例を示す説明図である。アクセス権優先度情報４２３を生成するマスタ端末装置は、同期処理を行うスレッドを実行する端末装置１０３の全てから実行結果を受け付けた場合、同期待ち時間を算出する。なお、マスタ端末装置も同期処理を行うスレッドを実行してもよい。この場合、同期待ち時間の算出を開始するタイミングは、マスタ端末装置がスレッドを終了しており、さらに、スレーブ端末装置もスレッドを終了している場合である。マスタ端末装置は、同期待ち時間を、下記（１）式で算出する。

【００５９】

同期待ち時間＝実行結果を最後に受け付けた端末装置の同期待ち開始時刻－端末装置の同期待ち開始時刻　…（１）

【００６０】

たとえば、マスタ端末装置は、レコード４２１－０～レコード４２１－３の同期待ち開始時刻フィールドから、端末装置１０３＃１を、実行結果を最後に受け付けた端末装置であると特定する。続けて、マスタ端末装置は、端末装置１０３＃０の同期待ち時間を（１）式から以下のように算出する。

【００６１】

端末装置１０３＃０の同期待ち時間＝１９０－１１０＝８０［ミリ秒］

【００６２】

マスタ端末装置は、端末装置１０３＃１～端末装置１０３＃３の同期待ち時間についても、それぞれ、０［ミリ秒］、１２０［ミリ秒］、１３０［ミリ秒］と算出する。算出後、マスタ端末装置は、同期待ち時間の短い順に仮想共有メモリ１０４へのアクセス権の優先度を設定する。設定されたアクセス権の優先度は、アクセス権優先度情報４２３として生成される。

【００６３】

アクセス権優先度情報４２３は、端末装置ＩＤ、優先度という２つのフィールドを含む。端末装置ＩＤフィールドには、端末装置１０３＃０～端末装置１０３＃３の識別情報が格納される。優先度フィールドには、仮想共有メモリ１０４へのアクセス権の優先度が格納される。なお、図６で示す優先度は、１が最も優先度が高く、数値が大きくなるほど優先度が低くなる状態であると定義する。

【００６４】

マスタ端末装置は、同期待ち時間が最も短かった端末装置１０３＃１を優先度１に設定する。続けて、マスタ端末装置は、同期待ち時間の短い順に、端末装置１０３＃０を優先

度２に設定し、端末装置１０３＃２を優先度３に設定し、端末装置１０３＃３を優先度４に設定する。

【００６５】

図７は、アクセス権優先度情報が設定された場合のスレッド処理システム１００の動作例を示す説明図である。図７で示すスレッド処理システム１００は、０［ミリ秒］の時点からスレッド０～スレッド３を実行し、２００［ミリ秒］にて、スレッド０’～スレッド３’を実行する。ここで、スレッド０～スレッド３とスレッド０’～スレッド３’の処理量は、同一であることを想定している。

【００６６】

また、０［ミリ秒］～１９０［ミリ秒］のスレッド処理システム１００は、仮想共有メモリ１０４へのアクセス権の優先度が設定されていない状態である。続けて、１９０［ミリ秒］～２００［ミリ秒］のスレッド処理システム１００は、仮想共有メモリ１０４へのアクセス権の優先度を設定する処理を実行している。さらに、２００［ミリ秒］～３７０［ミリ秒］のスレッド処理システム１００は、仮想共有メモリ１０４へのアクセス権の優先度が設定された状態となる。

【００６７】

また、マスタ端末装置である端末装置１０３＃０が、スレーブ端末装置となる端末装置１０３＃１～端末装置１０３＃３と通信し、端末装置１０３＃１～端末装置１０３＃３間では通信しないことを想定する。したがって、端末装置１０３＃０は、端末装置１０３＃２、端末装置１０３＃３とは高速通信し、端末装置１０３＃１とは低速通信する。

【００６８】

なお、スレッド０～スレッド３の実行時間、およびスレッドの実行によって発生する通信時間について、図７では次のように想定する。端末装置１０３＃０のスレッド０の処理時間が６０［ミリ秒］であり、端末装置１０３＃１のスレッド１の処理時間が１１０［ミリ秒］であり、端末装置１０３＃２のスレッド２の処理時間と端末装置１０３＃３のスレッド３の処理時間が４０［ミリ秒］である。

【００６９】

また、低速通信である端末装置１０３＃０－端末装置１０３＃１間の通信時間は、３０［ミリ秒］となることを想定する。高速通信である端末装置１０３＃０－端末装置１０３＃２、端末装置１０３＃０－端末装置１０３＃３間の通信時間は１０［ミリ］秒であることを想定する。

【００７０】

このような状態で、端末装置１０３＃０は、０［ミリ秒］の時点にて、スレッド０～スレッド３を端末装置１０３＃０～端末装置１０３＃３に割り当てる。スレッドを割り当てられた端末装置１０３＃０～端末装置１０３＃３は、スレッドを実行するために仮想共有メモリ１０４にアクセスする。なお、このときアクセスする仮想共有メモリ１０４に対応する共有メモリ４２２が、共有メモリ４２２＃０である場合を想定する。したがって、端末装置１０３＃１～端末装置１０３＃３は、端末装置１０３＃０と通信を行う。このとき、高速通信であり最も早く通信を開始できた端末装置１０３＃３が端末装置１０３＃０と通信を行う。

【００７１】

時刻１０［ミリ秒］にて、端末装置１０３＃０と端末装置１０３＃３の通信が終了すると、次に端末装置１０３＃０と端末装置１０３＃２が通信を開始する。また、通信を終了した端末装置１０３＃３は、スレッド３を実行する。また、時刻２０［ミリ秒］にて、端末装置１０３＃０と端末装置１０３＃２の通信が終了すると、続けて端末装置１０３＃０と端末装置１０３＃１が通信を開始する。また、通信を終了した端末装置１０３＃３は、スレッド２を実行する。続けて、時刻５０［ミリ秒］にて、端末装置１０３＃０と端末装置１０３＃１の通信が終了すると、端末装置１０３＃０がスレッド０を実行し、端末装置１０３＃１がスレッド１を実行する。

【００７２】

また、端末装置１０３＃３は、１０［ミリ秒］の時点から４０［ミリ秒］経過した５０［ミリ秒］の時点でスレッド３を終了し、時刻６０［ミリ秒］にて実行結果を端末装置１０３＃０に通知する。端末装置１０３＃０は、実行結果を受け付け、端末装置１０３＃３の同期待ち開始時刻を６０［ミリ秒］として、同期待ち開始時刻テーブル４２１に登録する。時刻７０［ミリ秒］でも同様に、端末装置１０３＃２がスレッド２を終了し、実行結果を端末装置１０３＃０に通知する。通知を受けた端末装置１０３＃０は、実行結果を受け付け、端末装置１０３＃２の同期待ち開始時刻を７０［ミリ秒］として、同期待ち開始時刻テーブル４２１に登録する。

【００７３】

また、時刻１１０［ミリ秒］にて、端末装置１０３＃０がスレッド０を終了すると、端末装置１０３＃０は、端末装置１０３＃０の同期待ち開始時刻を１１０［ミリ秒］として、同期待ち開始時刻テーブル４２１に登録する。さらに、時刻１６０［ミリ秒］にて、端末装置１０３＃１がスレッド１を終了し、時刻１９０［ミリ秒］にて実行結果を端末装置１０３＃０に通知する。端末装置１０３＃０は、端末装置１０３＃１の同期待ち開始時刻を１９０［ミリ秒］として、同期待ち開始時刻テーブル４２１に登録する。マスタ端末装置は、全ての端末装置１０３がスレッドを終了したため、次の処理を実行開始する。

【００７４】

また、時刻１９０［ミリ秒］にて、端末装置１０３＃０は、各端末装置１０３の同期待ち時間を算出する。端末装置１０３＃０～端末装置１０３＃３の同期待ち時間は、それぞれ、８０［ミリ秒］、０［ミリ秒］、１２０［ミリ秒］、１３０［ミリ秒］、と算出される。算出結果から、端末装置１０３＃０は、端末装置１０３＃０～端末装置１０３＃３の仮想共有メモリ１０４へのアクセス権を、それぞれ、優先度：２、優先度：１、優先度：３、優先度：４に設定する。

【００７５】

優先度が設定された状態で、端末装置１０３＃０は、２００［ミリ秒］の時点にて、スレッド０’～スレッド３’を端末装置１０３＃１～端末装置１０３＃３に割り当てる。スレッドを割り当てられた端末装置１０３＃１～端末装置１０３＃３は、スレッドを実行するために仮想共有メモリ１０４にアクセスする。端末装置１０３＃１～端末装置１０３＃３は、端末装置１０３＃０と通信を行う。

【００７６】

このとき、初めに、高速通信であり最も早く通信を開始できた端末装置１０３＃３が端末装置１０３＃０と通信を行う。しかし、その後、優先度が高い端末装置１０３＃１からのアクセス要求を受け付けると、端末装置１０３＃０は、端末装置１０３＃３による共有メモリ４２２＃０へのアクセスを一旦中断し、端末装置１０３＃１からの共有メモリ４２２＃０へのアクセスを行う。時刻２３０［ミリ秒］にて、端末装置１０３＃０と端末装置１０３＃１の通信が終了すると、次に優先度が高い端末装置１０３＃２が端末装置１０３＃０と通信を開始する。また、時刻２４０［ミリ秒］にて、端末装置１０３＃０と端末装置１０３＃２の通信が終了すると、次に優先度が高い端末装置１０３＃３が端末装置１０３＃０と通信を開始する。

【００７７】

通信が終了した端末装置１０３から順に、割り当てられたスレッド０’～スレッド３’を実行する。端末装置１０３＃２は、時刻２９０［ミリ秒］にて、スレッド２’の実行結果を通知し、端末装置１０３＃０は、スレッド２’の実行結果を受け付ける。同様に、端末装置１０３＃３は、時刻３００［ミリ秒］にて、スレッド３’の実行結果を通知し、端末装置１０３＃０は、スレッド３’の実行結果を受信する。端末装置１０３＃１は、時刻３７０［ミリ秒］にて、スレッド１’の実行結果を通知し、端末装置１０３＃０は、スレッド１’の実行結果を受け付ける。

【００７８】

このように、スレッド処理システム１００は、仮想共有メモリ１０４へのアクセス権の優先度が設定されることで、ボトルネックとなっていた端末装置１０３＃１の処理を早く

終了でき、スレッド処理システム１００の処理性能を向上できる。具体的に、図７では、仮想共有メモリ１０４へのアクセス権の優先度が設定されていないスレッド処理システム１００は、全てのスレッドが終了するまで１９０［ミリ秒］経過していた。仮想共有メモリ１０４へのアクセス権の優先度が設定されたスレッド処理システム１００では、全てのスレッドが終了するまで１７０［ミリ秒］となり、処理性能を向上させることができる。

【００７９】

次に、図７で示した動作を行うスレッド実行時の処理手順の一例を示すフローチャートを図８～図１０で示す。図８、図９では、マスタ端末装置によるスレッド実行時の処理手順を示しており、図１０では、マスタ端末装置からの通知によって動作するスレーブ端末装置によるスレッド実行時の処理手順を示している。

【００８０】

図８は、マスタ端末装置によるスレッド実行時の処理手順の一例を示すフローチャート（その１）である。マスタ端末装置は、新たなスレッドの割当要求が発生したか否かを判断する（ステップＳ８０１）。新たなスレッドの割当要求が発生していない場合（ステップＳ８０１：Ｎｏ）、マスタ端末装置は、一定時間経過後、ステップＳ８０１の処理を再び実行する。

【００８１】

新たなスレッドの割当要求が発生した場合（ステップＳ８０１：Ｙｅｓ）、マスタ端末装置は、新たなスレッドの割当要求をマスタ端末装置、またはスレーブ端末装置に通知する（ステップＳ８０２）。なお、新たなスレッドが複数発生した場合、マスタ端末装置は、ステップＳ８０２の処理を複数実行する。また、新たなスレッドをマスタ端末装置とスレーブ端末装置のうち、どの端末装置に割り当てるかを決定する処理は、スケジューラの機能に含まれる。スケジューラは、たとえば、マスタ端末装置とスレーブ端末装置のうち、最も負荷が最小となる端末装置に新たなスレッドを割り当ててもよい。スレッド割り当て後、マスタ端末装置は、マスタ端末装置に割り当てられたスレッドを実行する（ステップＳ８０３）。

【００８２】

続けて、マスタ端末装置は、割り当てられたスレッドの処理が終了したか否かを判断する（ステップＳ８０４）。スレッドの処理が終了した場合（ステップＳ８０４：Ｙｅｓ）、マスタ端末装置は、マスタ端末装置の実行結果を受け付けた時刻を同期待ち開始時刻として取得する（ステップＳ８０５）。取得後、マスタ端末装置は、取得された同期待ち開始時刻を同期待ち開始時刻テーブル４２１に登録する（ステップＳ８０６）。格納後、または、スレッドの処理が終了していない場合（ステップＳ８０４：Ｎｏ）、マスタ端末装置は、スレーブ端末装置から実行結果を受け付けたか否かを判断する（ステップＳ８０７）。なお、スレーブ端末装置の実行結果は、後述するステップＳ１００４の処理によって通知されてくる。

【００８３】

実行結果を受け付けた場合（ステップＳ８０７：Ｙｅｓ）、マスタ端末装置は、スレーブ端末装置から実行結果を受け付けた時刻を同期待ち開始時刻として取得する（ステップＳ８０８）。取得後、マスタ端末装置は、取得された同期待ち開始時刻を同期待ち開始時刻テーブル４２１に登録する（ステップＳ８０９）。格納後、または実行結果を受け付けていない場合（ステップＳ８０７：Ｎｏ）、マスタ端末装置は、割り当てを行ったスレッドの実行結果を全て受け付けたか否かを判断する（ステップＳ８１０）。実行結果を全て受け付けていない場合（ステップＳ８１０：Ｎｏ）、マスタ端末装置は、ステップＳ８０４の処理に移行する。実行結果を全て受け付けた場合（ステップＳ８１０：Ｙｅｓ）、マスタ端末装置は、ステップＳ９０１の処理に移行する。

【００８４】

図９は、マスタ端末装置によるスレッド実行時の処理手順の一例を示すフローチャート（その２）である。ステップＳ８１０の処理実行後、マスタ端末装置は、実行結果を最後に受け付けた端末装置１０３の同期待ち開始時刻から、各端末装置１０３の同期待ち時間

を算出する（ステップＳ９０１）。算出後、マスタ端末装置は、各端末装置１０３の同期待ち時間に応じて、仮想共有メモリ１０４へのアクセス権の優先度を設定する（ステップＳ９０２）。

【００８５】

続けて、マスタ端末装置は、設定された優先度から、アクセス権優先度情報４２３を生成し（ステップＳ９０３）、アクセス権優先度情報４２３をスレーブ端末装置に通知する（ステップＳ９０４）。通知後、マスタ端末装置は、マスタ端末装置の共有メモリ４２２へのアクセス権の優先度をマスタ端末装置のメモリコントローラ３１１に設定し（ステップＳ９０５）、ステップＳ８０１の処理に移行する。

【００８６】

図１０は、スレーブ端末装置によるスレッド実行時の処理手順の一例を示すフローチャートである。スレーブ端末装置は、マスタ端末からの通知を受け付けたか否かを確認する（ステップＳ１００１）。なお、マスタ端末装置からの通知は、ステップＳ８０２、ステップＳ９０５の処理によって通知されてくる。マスタ端末装置から通知を受け付けていない場合（ステップＳ１００１：通知なし）、スレーブ端末装置は、一定時間経過後、ステップＳ１００１の処理に移行する。

【００８７】

仮想共有メモリ１０４へのアクセス権優先度情報４２３の通知を受け付けた場合（ステップＳ１００１：アクセス権優先度情報４２３）、スレーブ端末装置は、スレーブ端末装置の共有メモリ４２２へのアクセス権の優先度をスレーブ端末装置のメモリコントローラ３１１に設定する（ステップＳ１００２）。設定後、スレーブ端末装置は、ステップＳ１００１の処理に移行する。

【００８８】

スレッド割当要求の通知を受け付けた場合（ステップＳ１００２：スレッド割当要求）、スレーブ端末装置は、割当要求のあったスレッドを実行する（ステップＳ１００３）。実行終了後、スレーブ端末装置は、スレッドの実行結果をマスタ端末装置に通知し、ステップＳ１００１の処理に移行する。

【００８９】

以上説明したように、スレッド処理方法、およびスレッド処理システムによれば、複数の端末装置で同期処理を行う場合、他端末装置を長時間待たせる低性能端末装置の共有メモリへのアクセス権の優先度を高く設定する。これにより、低性能端末装置の実行時間が短縮され他端末装置の同期待ち時間が削減されるため、スレッド処理システムは、全体の処理性能を向上できる。

【００９０】

また、全体の処理性能を向上させる他の技術として、データを冗長配置する方法が存在する。これにより、他の端末装置にアクセスする頻度が低下するため、処理性能を向上することができる。しかしながら、データを冗長配置しても、同期処理が頻繁に発生する場合、全体の処理性能が低下してしまっていた。本実施の形態におけるスレッド処理システムは、同期待ち時間を可能な限り削減するため、全体の処理性能を向上することができる。

【００９１】

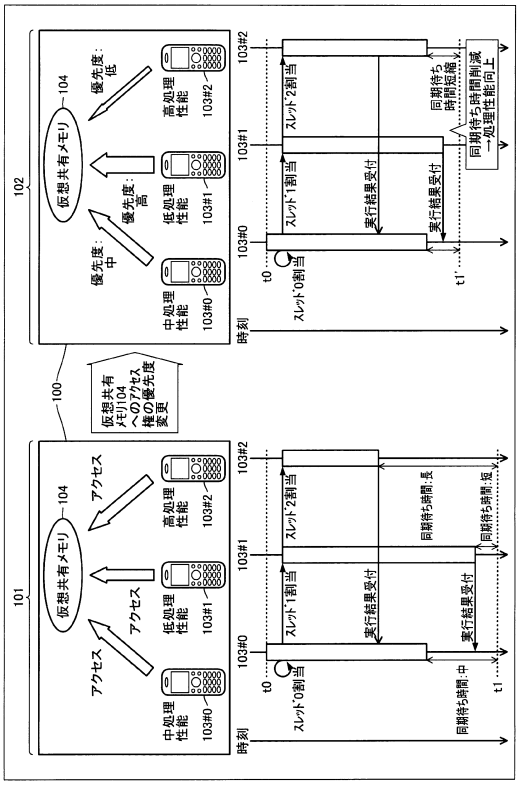
また、データを冗長配置する方法は、メモリの使用容量が大きくなってしまう問題があった。本実施の形態の適用の対象となる携帯電話などの端末装置は、メモリ容量に限界があるため、メモリの使用容量が大きいデータの冗長配置を適用するのが困難である。本実施の形態のスレッド処理システムは、データを冗長配置しなくてよいため、メモリ容量の削減が可能であり、端末装置にも適用することができる。

【００９２】

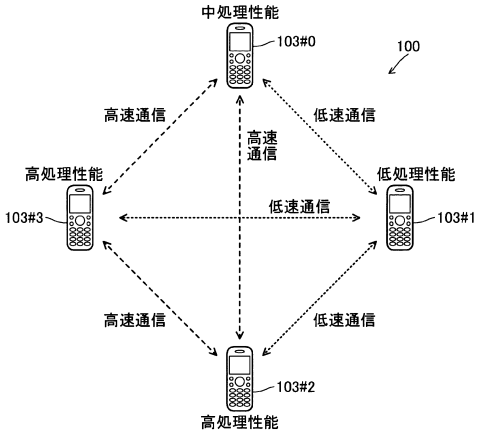
また、スレッド処理システムは、共有メモリへのアクセス権の優先度をマスタ端末装置からスレーブ端末装置に通知してもよい。これにより、全端末装置で優先度制御を行った仮想分散共有メモリを運用することができる。

【００９３】
　また、スレッド処理システムは、複数のスレッドの実行結果を受け付けるときの第１時刻情報と、複数のスレッドの実行結果を受け付け終わったときの第２時刻情報から、各端末装置の同期待ち時間を算出し、同期待ち時間に基づいて、優先度を設定してもよい。同期待ち時間を算出することで、スレッド処理システムは、ボトルネックとなっている端末装置を検出することができる。
【００９４】
　なお、ボトルネックとなる端末装置は、低処理性能であったり、低速通信であったりする。しかし、本実施の形態にかかるスレッド処理システムは、端末の処理性能の取得、通信速度の取得を行わずに、実行結果を受け付けた時刻によってボトルネックとなる端末装置を検出することができる。したがって、本実施の形態におけるスレッド処理システムは、共有メモリの優先度情報以外の新たな情報を端末間で通信しなくてよい。
【００９５】
　また、スレッド処理システムは、同期待ち時間が小さいとき共有メモリへのアクセス権の優先度を高く設定してもよい。これにより、スレッド処理システムは、ボトルネックとなり、同期待ち時間が小さくなった端末装置の優先度を高く設定することができ、全体の処理性能を向上させることができる。
【００９６】
　なお、本実施の形態で説明したスレッド処理方法は、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。本スレッド処理方法を実行するスレッド処理プログラムは、ハードディスク、フレキシブルディスク、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。また本スレッド処理プログラムは、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】
【００９７】
　１０３　端末装置
　３０１　ＣＰＵ
　３０３　ＲＡＭ
　３０８　Ｉ／Ｆ
　３１０　バス
　３１１　メモリコントローラ
　４０１　検出部
　４０２　割当部
　４０３、４１３　実行部
　４０４　取得部
　４０５　登録部
　４０６　算出部
　４０７　設定部
　４０８　通知部
　４０９、４１４　制御部
　４１０、４１１　管理部
　４１２　受付部
　４２１　同期待ち開始時刻テーブル
　４２２　共有メモリ
　４２３　アクセス権優先度情報
　４３１　ＯＳ
　４３２　スケジューラ
　４３３　アプリ

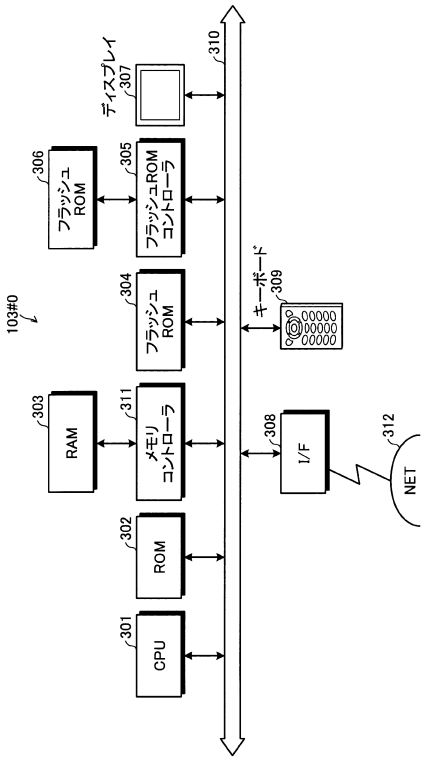
【図１】


102
仮想共有メモリ
104
優先度：低
優先度：高
優先度：中
高処理性能
103#2
低処理性能
103#1
中処理性能
103#0
100
仮想共有メモリ104へのアクセス権の優先度変更
101
アクセス
時刻
t0
t1
スレッド0割当
スレッド1割当
スレッド2割当
実行結果受付
同期待ち時間：長
同期待ち時間：短
同期待ち時間：中
同期待ち時間短縮
同期待ち時間削減
処理性能向上

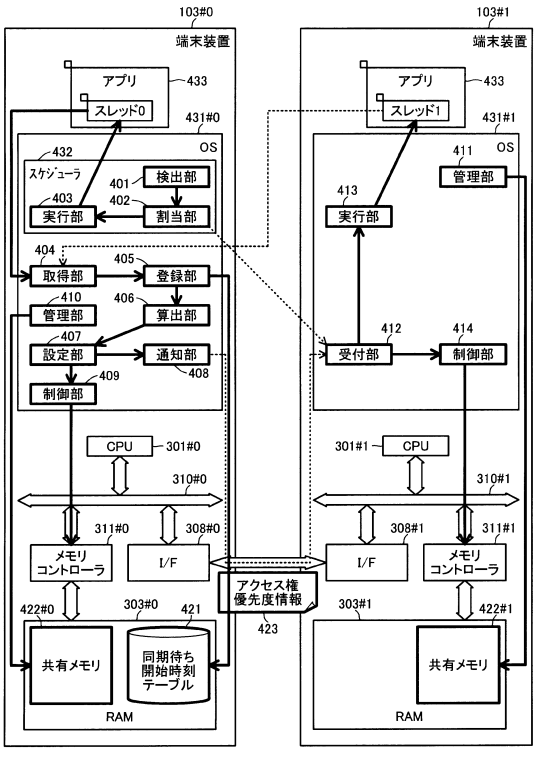
【図２】


中処理性能
103#0
100
高速通信
低速通信
高速通信
高処理性能
103#3
低処理性能
103#1
高速通信
低速通信
高速通信
103#2
高処理性能

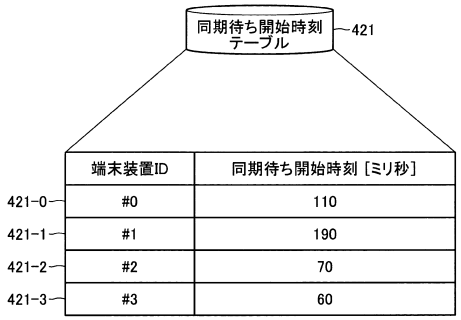
【図３】


103#0
ディスプレイ
307
310
フラッシュROM
306
フラッシュROMコントローラ
305
フラッシュROM
304
キーボード
309
RAM
303
メモリコントローラ
311
I/F
308
312
NET
ROM
302
CPU
301

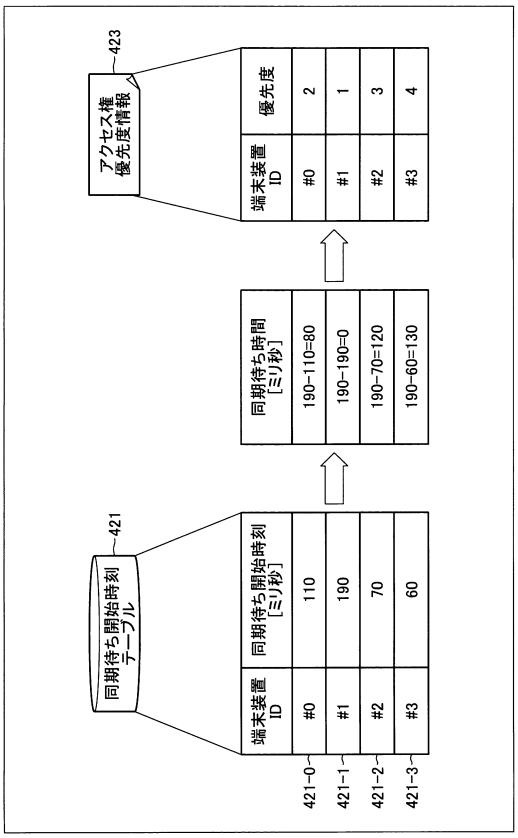
【図４】


103#0
端末装置
103#1
端末装置
アプリ
433
スレッド0
スレッド1
431#0
431#1
OS
411
管理部
432
スケジューラ
401
検出部
402
割当部
403
実行部
413
実行部
404
取得部
405
登録部
410
管理部
406
算出部
412
414
受付部
制御部
407
設定部
通知部
408
409
制御部
CPU
301#0
301#1
310#0
310#1
311#0
308#0
308#1
311#1
メモリコントローラ
I/F
I/F
メモリコントローラ
アクセス権優先度情報
423
422#0
303#0
421
303#1
422#1
共有メモリ
同期待ち開始時刻テーブル
共有メモリ
RAM
RAM

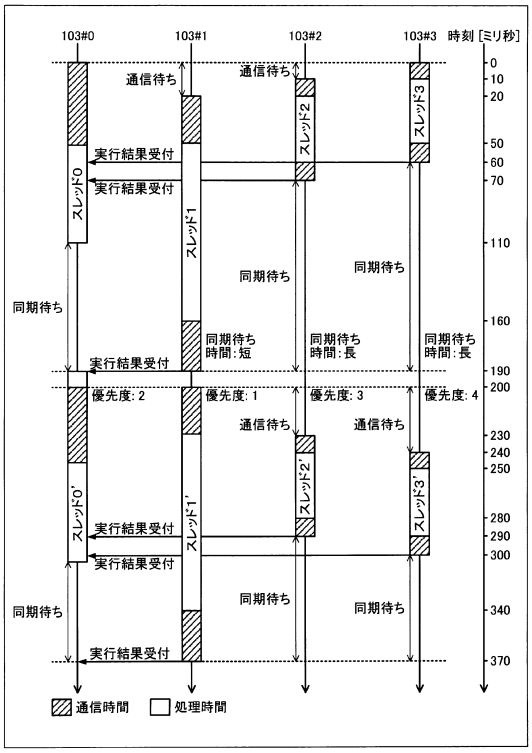
【図５】


同期待ち開始時刻テーブル
421
端末装置ID
同期待ち開始時刻［ミリ秒］
421-0
#0
110
421-1
#1
190
421-2
#2
70
421-3
#3
60

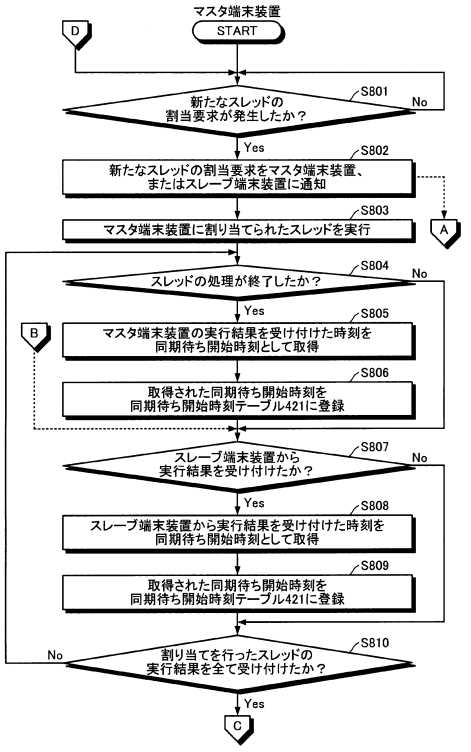
【図６】


423
アクセス権優先度情報
端末装置ID
優先度
#0
2
#1
1
#2
3
#3
4
同期待ち時間［ミリ秒］
190−110=80
190−190=0
190−70=120
190−60=130
421
同期待ち開始時刻テーブル
端末装置ID
同期待ち開始時刻［ミリ秒］
#0
110
#1
190
#2
70
#3
60
421-0
421-1
421-2
421-3

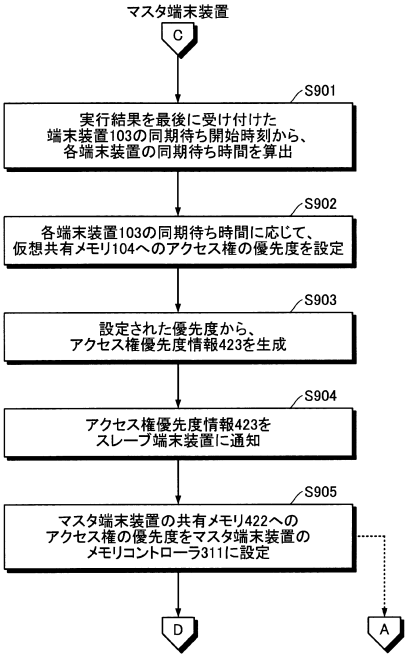
【図７】


103#0
103#1
103#2
103#3
時刻［ミリ秒］
通信待ち
実行結果受付
スレッド0
スレッド1
スレッド2
スレッド3
同期待ち
同期待ち時間：短
同期待ち時間：長
優先度：2
優先度：1
優先度：3
優先度：4
スレッド0'
スレッド1'
スレッド2'
スレッド3'
通信時間
処理時間

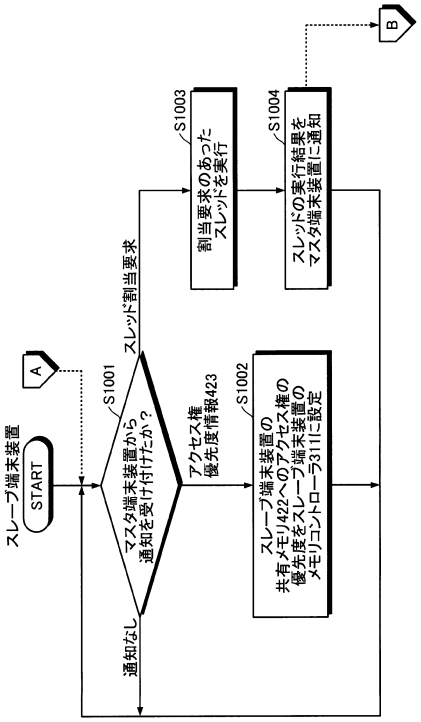
【図８】


マスタ端末装置
START
D
S801
新たなスレッドの割当要求が発生したか？
No
Yes
S802
新たなスレッドの割当要求をマスタ端末装置、またはスレーブ端末装置に通知
S803
マスタ端末装置に割り当てられたスレッドを実行
A
S804
スレッドの処理が終了したか？
No
Yes
B
S805
マスタ端末装置の実行結果を受け付けた時刻を同期待ち開始時刻として取得
S806
取得された同期待ち開始時刻を同期待ち開始時刻テーブル421に登録
S807
スレーブ端末装置から実行結果を受け付けたか？
No
Yes
S808
スレーブ端末装置から実行結果を受け付けた時刻を同期待ち開始時刻として取得
S809
取得された同期待ち開始時刻を同期待ち開始時刻テーブル421に登録
S810
割り当てを行ったスレッドの実行結果を全て受け付けたか？
No
Yes
C

【図９】


マスタ端末装置
C
S901
実行結果を最後に受け付けた
端末装置103の同期待ち開始時刻から、
各端末装置の同期待ち時間を算出
S902
各端末装置103の同期待ち時間に応じて、
仮想共有メモリ104へのアクセス権の優先度を設定
S903
設定された優先度から、
アクセス権優先度情報423を生成
S904
アクセス権優先度情報423を
スレーブ端末装置に通知
S905
マスタ端末装置の共有メモリ422への
アクセス権の優先度をマスタ端末装置の
メモリコントローラ311に設定
D
A

【図１０】


スレーブ端末装置
START
A
S1001
マスタ端末装置から
通知を受け付けたか？
スレッド割当要求
通知なし
アクセス権
優先度情報423
S1002
スレーブ端末装置の
共有メモリ422へのアクセス権の
優先度をスレーブ端末装置の
メモリコントローラ311に設定
S1003
割当要求のあった
スレッドを実行
S1004
スレッドの実行結果を
マスタ端末装置に通知
B

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
(72)発明者　大友　俊也
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　大舘　尚記
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　井上　宏一

(56)参考文献　特開平　４－２６０９６２（ＪＰ，Ａ）
特開平　９－　４４３６６（ＪＰ，Ａ）
特開２００６－　６５４５３（ＪＰ，Ａ）
特開２０１０－　７９５０４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　９／４６　－９／５４