

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5458998号
(P5458998)

(45) 発行日 平成26年4月2日 (2014.4.2)

(24) 登録日 平成26年1月24日 (2014.1.24)

(51) Int.Cl.

F I

G 0 6 F 9/48 (2006.01)

G 0 6 F 9/46 4 5 2 Z

G 0 6 F 9/46 (2006.01)

G 0 6 F 9/46 3 5 0

請求項の数 6 (全 27 頁)

(21) 出願番号 特願2010-64045 (P2010-64045)
 (22) 出願日 平成22年3月19日 (2010.3.19)
 (65) 公開番号 特開2011-198027 (P2011-198027A)
 (43) 公開日 平成23年10月6日 (2011.10.6)
 審査請求日 平成25年1月8日 (2013.1.8)

(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 100119161
 弁理士 重久 啓子
 (74) 代理人 100111822
 弁理士 渡部 章彦
 (72) 発明者 岩松 昇
 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 矢崎 昌朋
 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】 仮想マシンシステムおよび仮想マシン管理方法

(57) 【特許請求の範囲】

【請求項 1】

プロセスを実行する複数の仮想マシンを仮想マシン管理装置上に構築し、前記仮想マシン管理装置が、前記構築された仮想マシンにCPUを割り当てて、各仮想マシンのプロセスの実行を管理する仮想マシンシステムであって、

前記仮想マシン管理装置は、

プログラムの実行を管理する、プログラム管理部と、

前記プログラムを構成する複数のプロセスを各々で実行する複数の仮想マシンへのCPU割当を、スケジュールにもとづいて管理する、仮想マシンスケジュール管理部とを備えて、

前記プログラム管理部は、

前記プログラムを構成する複数のプロセスのうち当該プログラムの起動要求を受けて実行される第1プロセスを実行する第1仮想マシンから、前記第1プロセスの識別情報、前記プログラムを構成する前記第1プロセス以外の第2プロセス以降のプロセスを特定する情報を含むプログラム構成情報を取得して、前記プログラム構成情報および前記各プロセスを実行する仮想マシンの識別情報をプログラムテーブルに登録する、プログラムテーブル管理部と、

前記仮想マシン管理装置上に生成した仮想マシンで実行されているプロセスから通知を受信する、通知受信部と、

前記プログラムテーブルを参照して、前記通知受信部が受信した通知に対応するプロセ

スに関する処理の要求を前記仮想マシンスケジュール管理部へ送信する、通知処理部とを備えて、

前記仮想マシンスケジュール管理部は、

前記仮想マシンに対するCPU割当のスケジュールを決定する、スケジューリング部と

、
前記通知処理部から受信した前記要求が前記プログラムの起動要求である場合に、当該起動対象のプログラムを構成する前記第1プロセスを実行する第1仮想マシンを生成する処理と、前記通知処理部から受信した前記要求が前記第1プロセスに関連する第2プロセス以降のプロセスの実行要求である場合に、前記第2プロセス以降のプロセスを実行する仮想マシンを生成する処理とを行う、仮想マシン状態管理部と、

10

前記通知処理部からの通知にもとづいて、前記要求により実行が要求される要求先のプロセスを実行する仮想マシンを前記プログラムテーブルにより特定し、前記特定した仮想マシンへCPU割当を切り替える、仮想マシン切替部とを備えて、

前記第1仮想マシンは、

前記第1プロセスの実行時に、前記プログラム構成情報を前記プログラム管理部へ送信する処理と、要求元および要求先のプロセスを特定する情報を含むプロセス呼び出しの通知を前記プログラム管理部へ送信する処理とを行う、仮想マシンモニタ通知部を備え、

前記第2プロセス以降のプロセスを実行する各仮想マシンは、

要求元および前記処理したプロセスの返却先のプロセスを特定する情報を含むプロセス返却の通知を前記プログラム管理部へ送信する、仮想マシンモニタ通知部を備える

20

ことを特徴とする仮想マシンシステム。

【請求項2】

前記仮想マシン状態管理部は、前記通知受信部が、前記第2プロセス以降のプロセスの各属性情報を含むプログラム構成情報を受信した場合に、前記受信されたプログラム構成情報にもとづくプログラムテーブルを参照して、前記第2プロセス以降のプロセスとして前記属性情報に対応するプロセスを実行する仮想マシンの生成を行う

ことを特徴とする請求項1に記載の仮想マシンシステム。

【請求項3】

前記仮想マシン状態管理部は、前記プログラムの実行に関連するプロセスであって前記プログラムのプロセスと独立して実行される独立プロセスの情報を含むプログラム構成情報を受信した場合に、前記受信されたプログラム構成情報にもとづくプログラムテーブルを参照して、前記独立プロセスを実行する仮想マシンの生成を行い、

30

前記スケジューリング部は、前記生成された独立プロセスを実行する仮想マシンのスケジューリングを行う

ことを特徴とする請求項1または請求項2に記載の仮想マシンシステム。

【請求項4】

前記仮想マシンスケジュール管理部は、

前記仮想マシン毎のCPU消費時間を取得する処理と、前記プログラムテーブルをもとに、前記第2プロセス以降のプロセスを実行する各仮想マシンの前記CPU消費時間を、前記第1仮想マシンのCPU消費時間に加算する処理とを行う消費時間計算部を備えて、

40

前記スケジューリング部は、前記仮想マシンのCPU割当のスケジュールを決定する際に、前記第1仮想マシンのCPU消費時間として前記消費時間計算部により加算されたCPU消費時間を用いてスケジュールを決定する

ことを特徴とする請求項1ないし請求項3のいずれか一項に記載の仮想マシンシステム。

【請求項5】

前記仮想マシン状態管理部が、前記プログラムの第1プロセスを実行する第1仮想マシンを複数生成かつ前記複数の仮想マシンが実行するプロセスが前記第2プロセス以降のプロセスを共有している場合に、

前記消費時間計算部は、前記プログラムテーブルをもとに、前記複数の第1仮想マシン

50

によって共有されたプロセスを実行する仮想マシンのCPU消費時間を，前記共有する第1仮想マシンの各々のCPU消費時間に加算する

ことを特徴とする請求項4に記載の仮想マシンシステム。

【請求項6】

プロセスを実行する複数の仮想マシンを構築して，前記構築した仮想マシンにCPUを割り当てて，各仮想マシンのプロセスの実行を管理する仮想マシン管理装置が実行する仮想マシン管理方法であって，

前記仮想マシン管理装置が，プログラムの実行を管理するプログラム管理部と，前記プログラムを構成する複数のプロセスを各々で実行する複数の仮想マシンへのCPU割当を，スケジュールにもとづいて管理する仮想マシンスケジュール管理部と，プログラムテーブルとを備えて，

前記プログラム管理部が，前記プログラムの起動要求を受信する処理ステップと，

前記仮想マシンスケジュール管理部が，前記プログラムの起動要求が通知された場合に，前記プログラムを構成するプロセスであって前記起動要求を受けて実行する第1プロセスを実行する第1仮想マシンを生成する処理ステップと，

前記プログラム管理部が，前記第1仮想マシンから，前記第1プロセスの識別情報，前記プログラムを構成する前記第1プロセス以外の第2プロセス以降のプロセスを特定する情報を含むプログラム構成情報を取得する処理ステップと，

前記プログラム管理部が，前記プログラム構成情報および前記各プロセスを実行する仮想マシンの識別情報を前記プログラムテーブルに格納する処理ステップと，

前記プログラム管理部が，前記第1プロセスから，前記第1プロセスに関連する第2プロセス以降のプロセスの実行要求を受信する処理ステップと，

前記仮想マシンスケジュール管理部が，前記第2プロセス以降のプロセスを実行する仮想マシンを生成する処理ステップと，

前記プログラム管理部が，前記第1プロセスから前記第2プロセス以降のプロセスの呼び出し通知を受信する処理ステップと，

前記仮想マシンスケジュール管理部が，前記プログラムテーブルを参照して，前記呼び出し通知に対応するプロセスを実行する仮想マシンへCPU割当を切り替える処理ステップと，

前記プログラム管理部が，前記第2プロセスから前記呼び出し通知に関する処理の返却通知を受信する処理ステップと，

前記仮想マシンスケジュール管理部が，前記プログラムテーブルを参照して，前記返却通知に対応するプロセスを実行する仮想マシンへCPU割当を切り替える処理ステップとを備える

ことを特徴とする仮想マシン管理方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は，仮想マシンのスケジューリング処理技術に関する。

【背景技術】

【0002】

仮想マシンシステムは，実ハードウェアのCPU等の演算装置や，メモリ・入出力装置などのリソースを仮想化して，1つの物理的コンピュータ上で複数の仮想マシンを実行するものである。

【0003】

仮想マシンシステムの仮想マシン管理装置は，仮想マシンの起動・実行・シャットダウンを管理し，仮想マシンを時分割で実行させることによって，複数の仮想マシンが同時に実行されているかのような環境を提供する。

【0004】

仮想マシン管理装置内に備えられるスケジューラ機能が、どの仮想マシンにCPUを割り当てて実行させるかを決定して、仮想マシンの実行の切り替えの制御を行う。また、仮想マシン上のOS（ゲストOS）は、仮想マシン管理装置で行われる仮想マシンのスケジュールと同様に、自ゲストOS上で実行するプロセスのスケジューリングを行っている。

【0005】

図16は、各仮想マシン上のプロセス・スケジューリングの例を示す図である。

【0006】

図16に示すように、仮想マシン管理装置（VMM：仮想マシンモニタ）が、CPUに割り当てられる仮想マシンおよびその順序（VMプロセス・スケジューラ）を決定する。各仮想マシンVM_A、VM_BのゲストOSは、通常、仮想マシン管理装置によって自マシンにCPUが割り当てられている期間内に限り、自マシンでのプロセスをスケジューリングする。

10

【0007】

すなわち、仮想マシンシステムのOSによるプロセス・スケジューリングと、仮想マシン管理装置のゲストOSによる仮想マシン内プロセス・スケジューリングとは全く独立に行われるため、各仮想マシンVM_A、VM_BのゲストOSは、他の仮想マシン上のプロセスについて、プロセスの状態を知ること、制御することもできない。

【0008】

仮想マシン同士を関連させるスケジューリングの従来手法として、仮想マシン管理装置が、送信側仮想マシンから受信側仮想マシンへのOS間の通信要求を受けて、受信側仮想マシンのゲストOSに受信割り込みによる通知を行って、受信割り込み解除後にその受信処理に必要なCPU時間を受信側仮想マシンに追加割り当てするスケジューラ調整手法が知られている。

20

【0009】

また、仮想マシン管理装置が、相互に通信する仮想マシンの組を特定して、特定された組の各仮想マシンの実行スケジューリングにおいて、対応する仮想マシンのCPU割当時間を短縮するスケジューラ手法が知られている。

【先行技術文献】

【特許文献】

30

【0010】

【特許文献1】特開2009-075766号公報

【特許文献2】特開2008-165410号公報

【発明の概要】

【発明が解決しようとする課題】

【0011】

本願の発明者は、仮想化技術の応用として、プログラムを複数のモジュールやプロセスに分割して、分割したモジュールやプロセスの処理を、それぞれ別々の仮想マシン上の処理として独立実行させるような手法について検討をした。

【0012】

40

図1は、本願の発明者により検討された仮想マシンシステムの構成例を示す図である。

【0013】

図1に示す仮想マシンシステムにおいて、仮想マシン管理装置90は、1つのプログラムを、ユーザとのGUIを有する処理を行うアプリケーションプロセス側（フォアグラウンド側プロセス）と、アプリケーションプロセスから呼び出されるような処理を行うプロセス側（バックグラウンド側プロセス）とに分離して、それらのプロセスをそれぞれ別々の仮想マシン91x、91y、91zで実行する。

【0014】

ここで、他のアプリケーションによるハッキングで、プロセスXで動作をさせている仮想マシン91xが乗っ取られたと仮定する。この場合に、プロセスXから独立したプロセ

50

スY, プロセスZを実行している仮想マシン91y, 91zは, 仮想マシン91xと隔離されている。そのため, プロセスY, プロセスZが, センシティブなデータやデバイスへのアクセスを行うものであっても, これらをハッキングから保護することができ, セキュリティの向上を図ることができる。

【0015】

しかし, 元々1つのプログラムとして1つの仮想マシン上で動作するようなプログラム群(例えば, メインプログラムとサブルーチンプログラム)を, 複数のプロセスに分割して, 別々の仮想マシンで実行する場合に, 仮想マシン管理装置のスケジューリングの影響によって, 以下の問題が生じる。すなわち, メインプログラムのプロセスを実行するメイン仮想マシンと, サブルーチンプログラムのプロセスを実行するサブ仮想マシンとメインプログラムを動作するメイン仮想マシンとのスケジューリングについて, 元来1つのプログラムとして動作するサブルーチン呼び出しなどの順序や時間間隔を制御することができず, 一体化したプログラムとしての動作が保証されなくなるという問題がある。

10

【0016】

特に, 1つのプログラムを構成するプロセスを実行するメイン仮想マシンとサブ仮想マシン以外にも, 他のプログラムやプロセスを実行する複数の仮想マシンが動作している場合には, その数が増加するほど, メイン仮想マシンとサブ仮想マシンとの切り換えのタイミングや割り当て順序が合わないという問題がより多く生じて, 問題が深刻になる。

【0017】

図2は, メインプロセスと, メインプロセスから呼び出されるサブプロセスとのスケジューリングの理想例を示す図である。

20

【0018】

図2は, プログラムを, メイン仮想マシンVM__Aで実行されるプロセス1(以下, メインプロセスと呼ぶ)と, サブ仮想マシンVM__Bのプロセス1(以下, サブプロセスと呼ぶ)とに分離して実行している場合の, メインプロセスからサブプロセスの関数呼び出し時の望ましいスケジューリングを示している。

【0019】

メインプロセスからサブプロセス上の関数呼び出し(処理依頼)を行う際に, メインプロセスからの関数呼び出しのタイミングで, メインVM__AからサブVM__BへとCPUの割り当てが行われ, サブプロセスでの処理完了後に, 再びメインVM__Aに再割り当てされて, メインプロセスがサブプロセスの処理結果を直ちに取得して, プログラムの実行を継続できることが望ましい。

30

【0020】

図3は, メインプロセスと, メインプロセスと独立して実行するサブプロセス(独立サブプロセス)とのスケジューリングの理想例を示す図である。

【0021】

図3は, プログラムを, メイン仮想マシンVM__Aのプロセス1(以下, メインプロセスと呼ぶ)と, サブ仮想マシンVM__Bのプロセス1(以下, サブプロセスと呼ぶ)とに分離して実行していて, サブプロセスがデバイスへの定期的なデータ受信(ポーリング)などの動作を, メインプロセスと独立して行っている場合の望ましいスケジューリングを示している。

40

【0022】

サブVM__Bが, メインVM__Aとは全く独立に定期的にスケジューリングされることによって, サブプロセスは, ポーリング動作を行うことができる。メインプロセスは, メインVM__A上でスケジューリングされたタイミングでサブプロセスのデータを取得しようとする。この時, 常に, サブプロセスがデバイスからデータを受信済みであり, メインプロセスは, そのデータを直ちに取得して, 処理を滞りなく続けられることが望ましい。

【0023】

図2および図3に示すスケジューリング例は, プログラムが同一のプロセスであるかのように理想的な動作を行った場合の例である。

50

【 0 0 2 4 】

しかし、実際にプログラムを分離して異なる仮想マシンで実行する場合に、特に他のプログラムに関する複数の仮想マシンも存在しているときは、メインプロセスとサブプロセス間でのデータの受け渡しタイミングや順序が想定通りにならず、処理待ちなどが生じてプログラムの動作に支障がある場合がある。

【 0 0 2 5 】

図 4 は、メインの仮想マシンとサブ仮想マシン間に生じる不具合を説明するための図である。

【 0 0 2 6 】

図 4 に、プログラムを、メイン仮想マシン V M _ A のプロセス 1（以下、メインプロセスと呼ぶ）と、サブ仮想マシン V M _ B のプロセス 1（以下、サブプロセスと呼ぶ）とに分離して実行していて、他のプログラムのプロセスを実行する仮想マシン V M _ C がある場合に、実際に生じやすいスケジューリング動作の例を示している。

10

【 0 0 2 7 】

図 4 に示すスケジュール例において、サブプロセスの処理後にサブ V M _ B からメイン V M _ A へとスケジューリングが行われ、サブプロセスとメインプロセスが連続して実行されたため、メインプロセスがサブプロセスのデータをうまく取得できている（ケース 1）。

【 0 0 2 8 】

その後、メイン V M _ A 他 V M _ C メイン V M _ A とスケジューリングされたため、メイン V M _ A 上でメインプロセスがスケジューリングされるまでにサブプロセスが実行されなかったとする。この場合には、メインプロセスはサブプロセスからのデータ取得に失敗をしてしまう（ケース 2）。

20

【 0 0 2 9 】

さらにその後、メイン V M _ A 他 V M _ C サブ V M _ B 他 V M _ C メイン V M _ A とスケジューリングされ、メインプロセスはサブプロセスのデータを取得することができたが、サブプロセスの処理からメインプロセスの処理までに、他 V M _ C への割り当てが割り込まれていたとする。この場合には、メインプロセスは、サブプロセスのデータ取得に時間がかかってしまう（ケース 3）。

【 0 0 3 0 】

このように、1つのプログラムを複数の仮想マシン上で実行する場合に、関数の呼び出し、データの受け渡しなどが適切なタイミングや順序にならない状況が生じたり、他の仮想マシンに割り込まれて処理待ちが生じたりするなどの問題が発生する。

30

【 0 0 3 1 】

ところで、複数の仮想マシンへのスケジューリングを制御する従来手法として、以下の2つの方法がある。

【 0 0 3 2 】

[仮想マシンへの固定優先度付け]

仮想マシン管理装置から、仮想マシンに対して固定的な優先度を設定するスケジューリング手法がある。例えば、プログラムを、仮想マシン V M 1 , V M 2 で実行する場合に、仮想マシン V M 1 と仮想マシン V M 2 への C P U 割当時間を、C P U 全体の 4 0 % , 2 0 % に固定して設定する。

40

【 0 0 3 3 】

[グループスケジューリング]

複数の仮想マシンをグループ化し、C P U の割り当てをそのグループ単位で管理するスケジューリング手法がある。例えば、複数のプログラムを複数の仮想マシンに分割して実行する場合に、プログラム単位で仮想マシンをグループ化し、グループ単位での C P U 割当量の総計をもとにスケジューリングを行う。

【 0 0 3 4 】

しかし、固定優先度付けの手法では、仮想マシン V M 1 , V M 2 上に分割して実行され

50

るプロセス各々についてのCPU割当量を制御することができるが、仮想マシンVM1と仮想マシンVM2のCPU割当のタイミング(順序, 時間間隔など)は考慮されないため、上述の処理待ちなどの問題は解決できない。

【0035】

また、グループスケジューリング手法は、グループ間での公正な優先度付けやリソース配分が可能となるが、グループ内における仮想マシン間のCPU割当タイミングまでは考慮されないため、問題は解決できない。

【0036】

本願発明は、複数の仮想マシン上で実行される複数のプロセスが、1つのプログラムとして一体化した動作を効率的に行えるように、仮想マシン間のプロセッサ(CPU)の割り当ての順序や時間間隔等を考慮して仮想マシンのスケジュールを管理できる仮想マシンシステムを提供することを目的とする。

10

【0037】

また、本願発明は、複数の仮想マシン上で実行される複数のプロセスが、1つのプログラムとして一体化した動作を効率的に行える仮想マシン管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】

【0038】

本願発明の一態様として開示される仮想マシンシステムは、プロセスを実行する複数の仮想マシンを仮想マシン管理装置上に構築し、前記仮想マシン管理装置が、前記構築された仮想マシンにCPUを割り当てて、各仮想マシンのプロセスの実行を管理する仮想マシンシステムであって、前記仮想マシン管理装置は、プログラムの実行を管理する、プログラム管理部と、前記プログラムを構成する複数のプロセスを各々で実行する複数の仮想マシンへのCPU割当を、スケジュールにもとづいて管理する、仮想マシンスケジュール管理部とを備える。

20

【0039】

さらに、前記プログラム管理部は、1)前記プログラムを構成する複数のプロセスのうち当該プログラムの起動要求を受けて実行される第1プロセスを実行する第1仮想マシンから、前記第1プロセスの識別情報、前記プログラムを構成する前記第1プロセス以外の第2プロセス以降のプロセスを特定する情報を含むプログラム構成情報を取得して、前記プログラム構成情報および前記各プロセスを実行する仮想マシンの識別情報をプログラムテーブルに登録する、プログラムテーブル管理部と、2)前記仮想マシン管理装置上に生成した仮想マシンで実行されているプロセスから通知を受信する、通知受信部と、3)前記プログラムテーブルを参照して、前記通知受信部が受信した通知に対応するプロセスに関する処理の要求を前記仮想マシンスケジュール管理部へ送信する、通知処理部とを備える。

30

【0040】

さらに、前記仮想マシンスケジュール管理部は、1)前記仮想マシンに対するCPU割当のスケジュールを決定する、スケジューリング部と、2)前記通知処理部から受信した前記要求が前記プログラムの起動要求である場合に、当該起動対象のプログラムを構成する前記第1プロセスを実行する第1仮想マシンを生成する処理と、前記通知処理部から受信した前記要求が前記第1プロセスに関連する第2プロセス以降のプロセスの実行要求である場合に、前記第2プロセス以降のプロセスを実行する仮想マシンを生成する処理とを行う、仮想マシン状態管理部と、3)前記通知処理部からの通知にもとづいて、前記要求により実行が要求される要求先のプロセスを実行する仮想マシンを前記プログラムテーブルにより特定し、前記特定した仮想マシンへCPU割当を切り替える、仮想マシン切替部とを備える。

40

【0041】

そして、前記第1仮想マシンは、前記第1プロセスの実行時に、前記プログラム構成情報を前記プログラム管理部へ送信する処理と、要求元および要求先のプロセスを特定する

50

情報を含むプロセス呼び出しの通知を前記プログラム管理部へ送信する処理とを行う，仮想マシンモニタ通知部を備える。

【0042】

さらに，前記第2プロセス以降のプロセスを実行する各仮想マシンは，要求元および前記処理したプロセスの返却先のプロセスを特定する情報を含むプロセス返却の通知を前記プログラム管理部へ送信する，仮想マシンモニタ通知部を備える。

【発明の効果】

【0043】

開示した仮想マシンシステムによれば，プログラムを複数のプロセスに分割して，分割したプロセスを異なる仮想マシンでそれぞれ実行する場合に，プログラムの最初のプロセス起動時に，プログラム構成情報を取得して管理し，仮想マシンで実行されているプロセスから他の仮想マシンで実行されるプロセスの呼び出しや返却が通知されると，該当するプロセスを実行する仮想マシンを特定して，特定した仮想マシンへCPU割当を切り替えることができる。

【0044】

これにより，1つのプログラムを複数のプロセスで実行した場合に，複数のプロセス間の実行順序を考慮したスケジューリングを実現することができ，プログラムを分割せずに実行している場合と同等にCPU割当時間を制御することができる。よって，複数の仮想マシンで実行されているプログラムを1つの仮想マシンで実行している場合と同様の最小の応答時間で処理を実現することができる。

【図面の簡単な説明】

【0045】

【図1】本願の発明者により検討された仮想マシンシステムの構成例を示す図である。

【図2】メインプロセスと，メインプロセスから呼び出されるサブプロセスとのスケジューリングの理想例を示す図である。

【図3】メインプロセスと，メインプロセスと独立して実行するサブプロセス（独立サブプロセス）とのスケジューリングの理想例を示す図である。

【図4】メインの仮想マシンとサブ仮想マシン間に生じる不具合を説明するための図である。

【図5】本発明の一実施例における仮想マシンシステムの構成例を示す図である。

【図6】プログラムの起動時の処理例を示す図である。

【図7】サブルーチン呼び出し時の処理例を示す図である。

【図8】独立サブプロセスの実行時の処理例を示す図である。

【図9】仮想マシンの状態遷移を示す図である。

【図10】優先度がUNDERとOVERの場合の待ち状態例を示す図である。

【図11】優先度がUNDER，OVERおよびIDLEの場合の待ち状態例を示す図である。

【図12】スケジューリング部によるCPU割当の例を示す図である。

【図13】独立実行されるサブVMとして，リアルタイムVMが生成される場合の割り当て例を示す図である。

【図14】スケジューリング部によるCPU割当の例を示す図である。

【図15】仮想マシンシステムを実現するハードウェア構成例を示す図である。

【図16】各仮想マシン上のプロセス・スケジューリングの例を示す図である。

【発明を実施するための形態】

【0046】

図5は，本発明の一実施例における仮想マシンシステムの構成例を示す図である。

【0047】

図5に示す仮想マシンシステム1は，プロセスを実行する複数の仮想マシン（VM）2，3，4，...を仮想マシン管理装置（VMM）10上に構築して，仮想マシン管理装置10が，構築された仮想マシン2，3，4，...にコンピュータのプロセッサ（CPU）（図

10

20

30

40

50

示しない)を割り当てて、各仮想マシン2, 3, 4, ...のプロセスの実行を仮想マシン管理装置10が管理するシステムである。

【0048】

本形態において、プログラムは、例えば、OS上で動作するアプリケーションプログラムであるが、そのみを指すものではなく、アプリケーションプログラムから呼び出される特定のライブラリモジュール、OS中の一機能を実現するサブシステム、デバイスドライバなどの、コンピュータの実現する処理手順によって実現される機能を含むものである。

【0049】

プロセスは、OSが生成するプログラム動作単位を示すものであるが、そのみを指すものではなく、タスクやスレッド、ジョブなどと呼ばれる動作単位を含むものである。

10

【0050】

本形態において、プログラム実行中に実行されるプロセスのうち、仮想マシン上で最初に起動される第1プロセスをメインプロセスmpとする。メインプロセスmpとは別個に動作するプロセス群、すなわち第2プロセス以降のプロセス群を、サブプロセスspとして分離する。

【0051】

メインプロセスmpは、主に、ユーザが直接あるいは間接的に操作し、プログラムの起動停止などを行うための仮想マシン上で起動されるプロセスであり、ユーザの操作を受け付けるユーザインタフェースなどを備える。

20

【0052】

サブプロセスspは、主に、プログラムを構成するモジュールやオブジェクトの一部として、同期的または非同期的に呼び出されるサブルーチンとして実装されるプロセスである。サブプロセスspを実行する仮想マシンは、サブプロセスspのみを実行する特別な仮想マシンであってよい。

【0053】

メインプロセスmpとサブプロセスspとは異なる仮想マシン上で実行される。図5に示す構成例では、メインプロセスmpは、仮想マシン(以下、メインVMとする)2で、サブプロセスspは、仮想マシン(以下、サブVMとする)3でそれぞれ実行される。

【0054】

30

仮想マシンシステム1が備える仮想マシン管理装置(VMM)10は、プログラム管理部11、プログラムテーブル13、仮想マシンスケジュール管理部15、および仮想マシン間通信部17を備える仮想マシン管理装置である。

【0055】

プログラム管理部11は、プログラムのメインプロセスmpおよびサブプロセスspを管理する処理部である。

【0056】

プログラムテーブル13は、プログラムのメインプロセスmp、サブプロセスspの構成を示す情報、各プロセスを実行する仮想マシン(メインVM2、サブVM3)を識別する情報が格納される記憶部である。

40

【0057】

仮想マシンスケジュール管理部(VMスケジュール管理部)15は、複数の仮想マシン2, 3, 4, ...へのCPUの割り当てを、スケジュールにもとづいて管理する処理部である。

【0058】

仮想マシン間通信部(VM間通信部)17は、メインVM2、サブVM3の仮想マシン間通信部23, 33と、メッセージ通信を行う処理部である。

【0059】

プログラム管理部11は、通知受信部111、プログラムテーブル管理部113、および通知処理部115を備える。

50

【 0 0 6 0 】

通知受信部 1 1 1 は、メイン V M 2 のメインプロセス m p から、プログラム構成情報を受信する。また、通知受信部 1 1 1 は、メイン V M 2 またはサブ V M 3 で実行されているプロセスからの通知を受信する。

【 0 0 6 1 】

具体的には、通知受信部 1 1 1 は、仮想マシン（メイン V M 2 , サブ V M 3 ）上の O S やアプリケーションから送信された、仮想マシン管理装置 1 0 内の処理の呼び出し（ハイパーコール）を受信する。

【 0 0 6 2 】

プログラムテーブル管理部 1 1 3 は、プログラムのメインプロセス m p を実行するメイン V M 2 から、メインプロセス m p およびサブプロセス s p に関する識別情報と構成情報とを含むプログラム構成情報を取得して、プログラムを構成するメインプロセス m p およびサブプロセス s p の構成、ならびにこれらのプロセス（メインプロセス m p およびサブプロセス s p ）を実行する仮想マシンの識別情報とを対応付けた情報を、プログラムテーブル 1 3 として管理する。

10

【 0 0 6 3 】

一例として、プログラム構成情報は、プログラムのメインプロセス m p の識別情報、メインプロセス m p から呼び出されるサブプロセス s p の構成情報を含む。

【 0 0 6 4 】

メインプロセス m p の識別情報は、実行される仮想マシン上でメインプロセスを一意に特定可能なプロセス番号やプログラム名などである。

20

【 0 0 6 5 】

サブプロセス s p の構成情報は、例えば、メインプロセス m p からサブプロセス s p を識別するためのプログラム名、初期状態、使用するメモリ量などのパラメタである。

【 0 0 6 6 】

また、サブプロセス s p の構成情報は、サブプロセス s p の役割や機能を示す属性情報を含む情報であってもよい。

【 0 0 6 7 】

また、サブプロセス s p の構成情報は、メインプロセス m p と独立して実行されるサブプロセス（独立サブプロセス）の識別情報および制御情報を含む情報であってもよい。この場合に、独立サブプロセスの制御情報は、例えば、サブプロセス s p を実行する仮想マシン 3 のスケジューリングに関するパラメタである。具体的なパラメタは、仮想マシン管理装置 1 0 のスケジューリング部 1 5 5 が採用するスケジューリングアルゴリズムにより異なるが、例えば、スケジューリングの優先度、デッドライン時間などを示すものである。

30

【 0 0 6 8 】

独立サブプロセスを実行するサブ V M 3 により、例えば特定デバイスのポーリング動作などが実行可能となる。

【 0 0 6 9 】

仮想マシンの識別情報は、仮想マシン管理装置 1 0 が仮想マシン 2 , 3 , 4 , ...を一意に特定可能な仮想マシン番号、仮想マシン名などである。

40

【 0 0 7 0 】

なお、プログラム構成情報のサブプロセス s p の構成情報は、サブプロセス s p 自体の実行イメージ、またはサブプロセス s p の実行イメージを含む仮想マシン自体の実行イメージを含む情報であってもよい。

【 0 0 7 1 】

プログラムテーブル管理部 1 1 3 は、プログラムテーブル 1 3 に、メインプロセス m p の識別情報と、それを実行するメイン V M 2 の識別情報、サブプロセス s p の識別情報とサブ V M 3 の識別情報とを登録し、参照する。

【 0 0 7 2 】

50

通知処理部 115 は、プログラムテーブル 13 を参照して、通知受信部 111 が受信した通知に対応する依頼（要求）を仮想マシンスケジュール管理部 15 へ送信する。

【0073】

具体的には、通知処理部 115 は、メイン VM 2 およびサブ VM 3 から受信した通知を解釈して、メインプロセス mp とサブプロセス sp に関する処理の依頼として、「仮想マシン（VM）切替依頼」、「サブ VM 起動依頼」、「仮想マシン（VM）実行依頼」を仮想マシンスケジュール管理部 15 に通知する。

【0074】

「VM 切替依頼」は、メインプロセス mp またはサブプロセス sp からの通知が、プロセスの呼び出し通知である場合に、通知された呼び出し先のプロセスを実行する仮想マシンへの CPU の切り替えを要求するものである。

10

【0075】

「サブ VM 起動依頼」は、メインプロセス mp からの通知がサブプロセスの起動依頼である場合に、通知された起動依頼のサブプロセス sp を実行する仮想マシンの起動を要求し、メイン VM 2 と起動したサブプロセス sp を実行するサブ VM 3 との仮想マシン間の通信路確立の要求である。

【0076】

「VM 実行依頼」は、メインプロセス mp からの通知が独立サブプロセスの実行通知である場合に、通知された独立サブプロセス sp を実行するサブ VM 2 のスケジューリングを要求するものである。

20

【0077】

仮想マシンスケジュール管理部 15 は、仮想マシン 2, 3, 4, ... の実行状態、仮想マシン 2, 3, 4, ... が消費した CPU の時間（CPU 消費時間）を管理し、仮想マシン 2, 3, 4, ... への CPU 割当を時分割で切り替えるスケジュール管理処理を行う。

【0078】

仮想マシンスケジュール管理部 15 は、仮想マシン切替部（VM 切替部）151、仮想マシン状態管理部（VM 状態管理部）153、スケジューリング部 155、および消費時間計算部 157 を備える。

【0079】

仮想マシン切替部 151 は、通知処理部 115 からの要求（VM 切替依頼）にもとづいて、要求先の仮想マシンへ CPU の割り当てを切り替える。

30

【0080】

仮想マシン状態管理部 153 は、プログラムの起動要求、サブプロセスの起動要求などの通知によるサブ VM 起動依頼、VM 実行依頼にもとづいて、該当するプロセスに対応する仮想マシンを生成する。なお、本実施例における仮想マシンの生成は、仮想マシンを新規に起動することにより実現しても良いし、既に起動済みの仮想マシンを該当するプロセスに割当てることにより実現しても良い。

【0081】

スケジューリング部 155 は、仮想マシン 2, 3, 4, ... の CPU の割り当てのスケジュールを決定する。スケジューリング部 155 は、仮想マシンの CPU 消費時間、優先度などから、所定のスケジューリングアルゴリズムにもとづいて、次にどの仮想マシンに CPU をどれだけ割り当ててかを決定する。

40

【0082】

消費時間計算部 157 は、仮想マシン 2, 3, 4, ... 毎に、使用した CPU の消費時間を計算して、プログラムテーブル 13 をもとに、サブプロセス sp を実行するサブ VM 3 での CPU 消費時間を、メイン VM 2 の CPU 消費時間に加算する。

【0083】

この場合に、スケジューリング部 155 は、仮想マシンの CPU 割当のスケジュールを決定する際に、メイン VM のメインプロセス mp の消費時間として、消費時間計算部 157 により加算された消費時間を用いて、スケジュールを決定する。

50

【 0 0 8 4 】

メインVM 2は、ユーザがアプリケーションを利用し実際に操作を行う仮想マシンである。メインVM 2に割り当てられたCPUは、さらにメインVM 2上で動作するOSのプロセススケジューラにより、時分割でアプリケーションに割り当てられる。

【 0 0 8 5 】

メインVM 2が実行するメインプロセスmpは、メインVM 2とサブVM 3に分割して実行されるプログラムのうち、メインVM 2で実行されるプロセスである。

【 0 0 8 6 】

メインVM 2は、仮想マシンモニタ通知部（VMM通知部）21を備える。仮想マシンモニタ通知部21は、メインプロセスmpの実行時に、プログラム構成情報、メインプロセスmpで発行される要求や制御の通知（例えば、サブVM 3の生成、サブVM 3へのCPU割当の切り替えなどを要求するためのハイパーコール）をプログラム管理部11へ送信する。

10

【 0 0 8 7 】

サブVM 3は、メインVM 2からの要求で生成される、サブプロセスspを実行するための仮想マシンである。

【 0 0 8 8 】

サブVM 3が実行するサブプロセスspは、メインVM 2とサブVM 3に分割して実行されるプログラムのうち、サブVM 3で実行され、主にメインVM 2からのサブルーチンとして実装されたプロセスである。

20

【 0 0 8 9 】

サブVM 3は、仮想マシンモニタ通知部（VMM通知部）31を備える。仮想マシン仮想マシンモニタ通知部31は、サブプロセスspで発行される要求や制御の通知（例えば、メインVM 2へのCPU割当の切り替えなどを要求するためのハイパーコール）をプログラム管理部11へ送信する。

【 0 0 9 0 】

仮想マシンモニタ通知部21、31は、例えば、プログラムから仮想マシン管理装置10を呼び出すためのハイパーコールの他、プログラムからOSを経由して仮想マシン管理装置10を呼び出すためのシステムコール、あるいは、CPU命令における特定の特権命令や例外命令を発行し、仮想マシン管理装置10にその命令をトラップさせて、仮想マシン管理装置10に制御を遷移させることにより実現されてもよい。

30

【 0 0 9 1 】

また、メインVM 2およびサブVM 3は、それぞれ、仮想マシン間通信部（VM間通信部）23、33を備えてもよい。仮想マシン間通信部23、33は、メインVM 2とサブVM 3との間のメッセージ通信を行う。

【 0 0 9 2 】

また、メインVM 2およびサブVM 3は、仮想マシン間の共有メモリなどにより、サブルーチン呼び出し後の返却値、処理データの授受を行うことができる。

【 0 0 9 3 】

以下に、仮想マシンシステムの処理動作例を示す。

40

【 0 0 9 4 】

〔プログラムの生成時の処理例〕

図6は、プログラムの生成時の処理例を示す図である。

【 0 0 9 5 】

ステップS1：ユーザによるアプリケーションプログラムの起動があると、メインVM 2が生成される。メインVM 2が、メインプロセスmpを生成する。

【 0 0 9 6 】

ステップS2：メインプロセスmpは、サブプロセスspの実行イメージをメモリ上に展開する。

【 0 0 9 7 】

50

ステップS3：仮想マシンモニタ通知部21は、プログラム構成情報として、仮想マシン管理装置10に対して、メインプロセスmpの識別情報（プロセスID）、サブプロセスspの実行イメージ、サブプロセスの識別情報（プロセスID）を展開したメモリ上のアドレスを、ハイパーコールにより通知する。

【0098】

ステップS4：通知受信部111がプログラム構成情報を受けると、通知処理部115は、対応するサブプロセスspを実行するためのサブVM起動依頼を仮想マシンスケジュール管理部15へ通知する。そして、VM状態管理部153が、サブVM3を生成する。

【0099】

ステップS5：さらに、VM状態管理部153が、生成したサブVM3上にサブプロセスspの実行イメージをコピーして、サブプロセスspを起動させる。

10

【0100】

ステップS6：プログラムテーブル管理部113が、メインVM2の識別情報（VM番号）とメインプロセスmpのID、生成されたサブVM3の識別情報（VM番号）とサブプロセスspのIDとを対応づけてプログラムテーブル13に登録して、メインプロセスmpのハイパーコールに応答する。

【0101】

これにより、仮想マシン管理装置10では、プログラム毎に、メインプロセスmpを実行する仮想マシン（メインVM2）と、サブプロセスspを実行する仮想マシン（サブVM3）とを紐付けて管理することが可能となる。

20

【0102】

図6に示す処理例において、ステップS4において、VM状態管理部153は、メインVM2からの通知を受けてサブVM3を生成する。

【0103】

しかし、VM状態管理部153は、サブプロセスspを実行するサブVM3を、メインプロセスmpの実行開始前にあらかじめ生成しておき、プログラムテーブル管理部113は、メインプロセスmpからの通知により、メインプロセスmpとサブプロセスspとを関連付けてプログラムテーブル13に登録してもよい。

【0104】

これにより、特定の動作や機能を実現するサブプロセスspを実行するサブVM3が既に生成されている場合に、メインプロセスmpからの通知にもとづいて、該当するサブVM3をメインプロセスmpのメインVM2と対応付けて利用することが可能となる。

30

【0105】

また、この場合に、プログラムテーブル管理部113は、サブプロセスspを実行するサブVM3が、複数のメインプロセスmpから共有されるように、プログラムテーブル13に登録して管理しても良い。

【0106】

すなわち、サブVM3が事前に生成されている場合に、後からメインプロセスmpとサブプロセスspを対応付けて、1対1で使用されるようにしてよく、または、複数のメインプロセス間でサブプロセスspを共有して使用されるようにしてもよい。また、既に生成されているサブVM3を使用する場合であっても、メインプロセスmp間でサブプロセスspを共有しないようにすることもできる。

40

【0107】

さらに、消費時間計算部157は、共有されたサブVM3のCPU消費時間を、共有したメインプロセスmpのメインVM2各々のCPU消費時間に、それぞれ加算する。

【0108】

これにより、サブVM3を、あるサブルーチンを複数のメインプロセスから呼び出す共通ライブラリとして利用することが可能となる。また、プログラム間のスケジューリングをCPUの消費時間に則して適切に行うことが可能となる。

【0109】

50

〔サブルーチン呼び出し時の処理例〕

図7は、サブルーチン呼び出し時の処理例を示す図である。

【0110】

ステップS11：仮想マシン管理装置10のスケジューリング部155が決定したスケジュールにもとづいて、仮想マシン切替部151が、メインVM2にCPUを割り当てる。

【0111】

ステップS12：メインVM2において、メインプロセスmpにCPUを割り当てられる。

【0112】

ステップS13：メインプロセスmpが、サブプロセスspのサブルーチン呼び出しのために、仮想マシン管理装置10に、サブプロセスspのIDを指定したVM切替のハイパーコール呼び出しを行う。

【0113】

ステップS14：通知受信部111が、ハイパーコール呼び出しを受信すると、通知処理部115は、サブプロセスspのIDを指定したVM切替依頼を仮想マシンスケジュール管理部15へ通知する。VM切替部151は、プログラムテーブル13を参照して、サブプロセスspIDから、切替先のサブVM3のVM番号を特定する。

【0114】

ステップS15：VM切替部151は、特定したサブVM3にCPUの割り当てを切り替える。

【0115】

ステップS16：サブVM3において、サブプロセスspにCPUが割り当てられて、サブルーチンが実行される。

【0116】

ステップS17：サブルーチンの処理終了後、仮想マシンモニタ通知部31は、仮想マシン管理装置10に、VM切替通知のハイパーコール呼び出しを行い、メインVM2へのCPU割当を通知する。

【0117】

ステップS18：通知受信部111が、ハイパーコール呼び出しを受けると、通知処理部115は、サブVM3からのVM切替依頼を仮想マシンスケジュール管理部15へ通知する。VM切替部151は、プログラムテーブル13を参照して、サブプロセスspIDから、処理の戻り先のメインVM2のVM番号を確認する。

【0118】

ステップS19：消費時間計算部157は、サブVM3のCPU消費時間を取得して、取得した消費時間をメインVM2の消費時間に加算する。

【0119】

ステップS110：VM切替部151は、メインVM2にCPUの割り当てを切り替える。

【0120】

ステップS111：メインVM2において、メインプロセスmpにCPUが割り当てられる。VM間通信部23、33の通信によって、メインプロセスmpは、サブプロセスspのサブルーチン呼び出しの出力結果を取得して、処理を継続する。

【0121】

以上のように、仮想マシン管理装置10は、従来のサブルーチン呼び出しと同様に、メインプロセスmpからサブプロセスspへとCPU割当を変更し、サブプロセスspでの処理終了後に、再びメインプロセスmpへとCPU割当を戻すことができる。

【0122】

また、仮想マシン管理装置10は、サブVM3のCPU消費時間をメインVM2のCPU消費時間として扱う。

10

20

30

40

50

【 0 1 2 3 】

これにより、メインプロセス $m p$ とサブプロセス $s p$ とが同一の仮想マシン上で実行されているように、一体化してスケジューリングすることが可能となる。

【 0 1 2 4 】

〔独立サブプロセスの実行時の処理例〕

図 8 は、独立サブプロセスの実行時の処理例を示す図である。

【 0 1 2 5 】

独立サブプロセスの実行は、サブプロセス $s p$ を独立実行させて、ポーリングなどのタスクをメインプロセス $m p$ からの指示で行う処理で利用される。

【 0 1 2 6 】

ステップ $S 2 1$: スケジューリング部 1 5 5 が決定したスケジュールにもとづいて、仮想マシン切替部 1 5 1 が、メイン $V M 2$ に $C P U$ を割り当てる。

【 0 1 2 7 】

ステップ $S 2 2$: メイン $V M 2$ において、メインプロセス $m p$ に $C P U$ が割り当てられる。

【 0 1 2 8 】

ステップ $S 2 3$: メインプロセス $m p$ が、サブプロセス $s p$ にデバイスのポーリングをさせるため、サブプロセス $s p$ の $I D$ を指定して、仮想マシン管理装置 1 0 に $V M$ 実行依頼の通知をハイパーコール呼び出しにより行う。

【 0 1 2 9 】

ステップ $S 2 4$: 通知受信部 1 1 1 が、ハイパーコール呼び出しを受けると、通知処理部 1 1 5 は、サブプロセス $s p$ の $I D$ を指定した $V M$ 実行依頼を仮想マシンスケジュール管理部 1 5 へ通知する。 $V M$ 状態管理部 1 5 3、プログラムテーブル 1 3 を参照して、サブプロセス $s p$ の $I D$ から独立実行を行うサブ $V M$ の $V M$ 番号を特定し、サブ $V M$ 番号とスケジューリングのパラメタ（例えば優先度など）をスケジューリング部 1 5 5 に渡す。スケジューリング部 1 5 5 は、特定されたサブ $V M 3$ のスケジューリングを登録する。

【 0 1 3 0 】

ステップ $S 2 5$: $V M$ 切替部 1 5 1 は、メインプロセス $m p$ からの $V M$ 切替通知を受け付けることなく、サブ $V M 3$ に $C P U$ を割り当て、サブ $V M 3$ でサブプロセス $s p$ に $C P U$ が割り当てられる。メインプロセス $m p$ は、メインプロセス $m p$ に $C P U$ が割り当てられたタイミングで、 $V M$ 間通信部 2 3、3 3 により、サブプロセス $s p$ がポーリングにより取得したデータを受信して、処理を行う。

【 0 1 3 1 】

以上のように、仮想マシン管理装置 1 0 は、メインプロセス $s p$ から得たサブプロセス $s p$ の制御情報（スケジューリングパラメタ）にもとづいて、該当するサブ $V M 3$ を、メイン $V M 2$ とは独立にスケジューリングすることができる。

【 0 1 3 2 】

一方で、メインプロセス $m p$ から取得したサブプロセス $s p$ の制御情報（例えば、スケジューリングパラメタ）にもとづいて独立実行するサブプロセス $s p$ のスケジューリングが行われるため、メイン $V M 2$ と関連付けたスケジューリングも可能となる。

【 0 1 3 3 】

〔スケジューリング処理〕

仮想マシン管理装置 1 0 のスケジューリング部 1 5 5 の処理を説明する前に、スケジューリング部 1 5 5 が採用するクレジットスケジュール手法を説明する。

【 0 1 3 4 】

クレジットスケジュールリングでは、実行するプロセスに応じた重み w が各仮想マシンに設定され、一定時間（例えば、30 ミリ秒 [$m s$] ）毎に、その重み w に応じた $C P U$ の持ち時間 ($C r e d i t$) が仮想マシンに付与される。

【 0 1 3 5 】

例えば、 $C r e d i t + = 30 \times (\text{重み } w / \text{Active な仮想マシン } V M \text{ の総重み } W$

10

20

30

40

50

）とし，ここで，重み $w = 1$ から 10 までの値，デフォルト値は 1 とする。

【0136】

さらに，クレジットスケジューラは， $Credit$ により仮想マシンの状態を更新して，仮想マシンの実行優先度を再設定する。

【0137】

図9は，仮想マシンの状態遷移を示す図である。

【0138】

仮想マシンは， CPU を割り当てられて実行されると，実行された時間分だけ $Credit$ が消費（減算）される。付与された $Credit$ が残っている（正の値）の場合には，仮想マシンの状態は「 $UNDER$ 」となる。 $Credit$ を使いきっている（負の値）場合には，仮想マシンの状態は「 $OVER$ 」である。なお，一定時間（例えば， $30ms$ ）経過後に，クレジットスケジューラから $Credit$ を再付与されて，値が正になると「 $UNDER$ 」となる。

10

【0139】

また， $Credit$ が一定値（ここでは 30 ）以上である場合には，その仮想マシンは活発に動作していないと判断され， $Credit$ の加算対象から除外され， $Credit$ の値が 0 となり，状態が「 $UNDER (INACTIVE)$ 」となる。これは，実行を行わない待ち状態にある仮想マシンが延々と $Credit$ 値を加算され続けるのを防止するための処置である。

20

【0140】

CPU 割当により仮想マシンの実行が再開されると，状態は「 $UNDER$ 」となる。

【0141】

同様に $OVER$ の場合も， $Credit$ の下限値（ここでは -30 とする）を設定しておき，実行を続けている仮想マシンの $Credit$ が延々と減算され続けるのを防止する。

【0142】

クレジットスケジューラは， $UNDER$ ， $OVER$ の順に高い優先度として扱い， $UNDER$ ， $OVER$ の状態にある仮想マシンをランキューで管理し，ランキューの先頭の仮想マシンから順番に CPU を割り当て，また，実行後には，仮想マシンはランキューの末尾に移動させる。

30

【0143】

図10は，優先度が $UNDER$ と $OVER$ の場合の待ち状態例を示す図である。

【0144】

図10に示すように，仮想マシンの状態が $OVER$ から $UNDER$ に戻る場合には，その仮想マシンは， $UNDER$ のランキューの先頭へ移動される。また， $UNDER$ から $OVER$ に落ちる場合には，仮想マシンは， $OVER$ のランキューの末尾に移動される。

【0145】

クレジットスケジューラでは，仮想マシンの状態更新と仮想マシン間の優先度が再設定された次の $30ms$ の間で，図10に示すような $UNDER$ のランキューの先頭から順に実仮想マシンに CPU が割り当てられる。そして，仮想マシン自体が処理待ちあるいは I/O 命令発行などで待ち状態になると， CPU 割当がランキューの次の仮想マシンに切り替えられる。 $UNDER$ のランキューの仮想マシンの割り当てが一巡した後は， $OVER$ のランキューの先頭の仮想マシンから割り当てが行われる。

40

【0146】

クレジットスケジューラは， CPU 割当の切り替え時に，仮想マシンの実行された時間（ CPU 消費時間）を計算して，仮想マシンに付与している $Credit$ を減算する。例えば，仮想マシンが $10ms$ 実行された場合には， $Credit - 10ms$ （下限値を -30 ）として，その仮想マシンをランキューの末尾に移動する。

【0147】

上記の動作により， $Credit$ を使いきった仮想マシンは，まだ $Credit$ が残っ

50

ている仮想マシンが動作していない時間に限って動作可能になる。また、活発に動作していないと判断された仮想マシンには、新たなCreditが与えられず、活発に動作する仮想マシンがCreditを分けあうことになる。

【0148】

このようにして、各仮想マシンに設定された重みwに合わせて仮想マシン間の割り当てのバランスをとりながら、効率的なCPU割当が行われる。

【0149】

仮想マシン管理装置10のスケジューリング部155は、上記のクレジットスケジューリング手法をもとに、仮想マシンの状態として「UNDER」、「OVER」に加え、「IDLE」を設定する。

10

【0150】

図11は、優先度がUNDER、OVERおよびIDLEの場合の待ち状態例を示す図である。

【0151】

スケジューリング部155は、UNDERまたはOVERのランキューには、メインVM2のみを移動させるようにし、通常のランキュー順のCPU割当を、UNDERおよびOVERのランキューの仮想マシン(メインVM2)のみに対して行う。

【0152】

スケジューリング部155は、サブVM3を、生成時にはIDLE状態とし、UNDERまたはOVERのキューにあるメインVM2からの呼び出しによってのみ、CPUの割り当てを行う。すなわち、スケジューリング部155は、IDLE状態にあるサブVM3をCreditの加算対象および減算対象から除外する。

20

【0153】

図12は、スケジューリング部155によるCPU割当の例を示す図である。

【0154】

図12に、ある時点でのランキューの状態を示す。UNDERのランキューに、仮想マシンVM__A、VM__B、VM__C、VM__Dが、OVERのランキューに、VM__E、VM__F、VM__G、VM__Hが、IDLEのランキューに、VM__I、VM__J、VM__K、VM__Lが、それぞれあるとする。

【0155】

ここで、VM__Kが、VM__BのサブVMであり、VM__KからVM__Bを呼び出す処理があるとする。この場合に、仮想マシンがスケジューリングされる手順とCPUを割り当てられる仮想マシンは以下ようになる。

30

【0156】

ステップS41:スケジューリング部155は、UNDERのランキューの先頭のメインVM(VM__A)を割当対象とする。

【0157】

ステップS42:スケジューリング部155が、UNDERのランキューの次のメインVM(VM__B)に割り当て対象を移すと、VM切替部151は、メインVM(VM__B)にCPU割当を切り替える。

40

【0158】

ステップS43:仮想マシンVM__BからサブVM(VM__K)への切替依頼が通知されると、スケジューリング部155が、IDLEのサブVM(VM__K)に割り当て対象を移し、VM切替部151は、サブVM(VM__K)にCPU割当を切り替え、実行する。

【0159】

ステップS44:サブVM(VM__K)から、メインVM(VM__B)への割り当て切替が通知されると、スケジューリング部155は、UNDERのランキューのメインVM(VM__B)へ割り当て対象を移し、VM切替部151は、メインVM(VM__B)へCPU割当を切り替えて復帰する。

50

【 0 1 6 0 】

ステップ S 4 5 : スケジューリング部 1 5 5 は、U N D E R のランキューのメイン V M (V M _ C) へ割り当て対象を移し、V M 切替部 1 5 1 は、メイン V M (V M _ C) へ C P U 割当を切り替えて、次のプログラムのメイン V B _ C が実行される。

【 0 1 6 1 】

既に説明したように、I D L E 状態にあるサブ V M (V M _ K) は、C r e d i t の加算および減算対象ではないため、消費時間計算部 1 5 7 は、サブ V M (V M _ K) が消費した C P U 時間を、メイン V M (V M _ B) が消費した C P U 時間として C r e d i t から減算する。

【 0 1 6 2 】

これにより、スケジューリング部 1 5 5 は、メイン V M 2 の C r e d i t のみを通常の仮想マシンと同様に管理するだけで、サブ V M 3 の C r e d i t 消費量をメイン V M の消費量として計上することができる。

【 0 1 6 3 】

図 1 3 は、独立実行されるサブ V M として、リアルタイム V M が生成される場合の割り当て例を示す図である。

【 0 1 6 4 】

リアルタイム V M は、所定のスケジューリング期間中で常に一定の時間を割り当てられることが保証されている仮想マシンであり、他の仮想マシンとは異なる優先度でスケジューリングされる。

【 0 1 6 5 】

通常の仮想マシンの V M _ A , V M _ B , V M _ C , V M _ D と、リアルタイム V M の R T V M _ A , R T V M _ B が生成されている場合に、スケジューリング部 1 5 5 は、クレジットスケジューリング手法により処理を行う場合に、クレジットスケジューリングによってスケジューリングする時間と、リアルタイムでスケジューリングする時間とに分割して管理し、システムの中で一定時間をリアルタイム V M に割り当てるようにする。

【 0 1 6 6 】

スケジューリング部 1 5 5 は、一定のタイムスライス（ここでは 3 0 m s ）とする中で、タイムスライスの 9 0 % (2 7 m s) を、クレジットスケジューリングにもとづいて通常の V M (V M _ A ~ V M _ D) の割り当てをスケジューリングする。そして、スケジューリング部 1 5 5 は、タイムスライスの 1 0 % (3 m s) を、リアルタイム V M (R T V M _ A , R T V M _ B) に割り当てる。

【 0 1 6 7 】

リアルタイム V M は、優先度 p (ここでは 1 - 1 0 の値とし、デフォルト値は 1 とする) を持ち、各々のリアルタイム V M の C P U 持ち時間は、(3 m s × 優先度 p / リアルタイム V M の総優先度 P) とする。例えば、R T V M _ A の優先度 p = 1 , R T V M _ B の優先度 p = 2 の場合に、R T V M _ A に 1 m s , R T V M _ B に 2 m s がそれぞれ割り当てられるようにする。

【 0 1 6 8 】

図 1 4 は、スケジューリング部 1 5 5 による C P U 割当の例を示す図である。

【 0 1 6 9 】

図 1 4 に、ある時点でのランキューの状態を示す。U N D E R のランキューに、仮想マシン V M _ A , V M _ B , V M _ C , V M _ D が、O V E R のランキューに、V M _ E , V M _ F , V M _ G , V M _ H が、I D L E のランキューに、V M _ I , V M _ J , V M _ K , V M _ L が、リアルタイム V M 用 (R T) のランキューに、V M _ M が、それぞれあるとする。

【 0 1 7 0 】

サブ V M (V M _ L) は、生成時には通常のサブ V M として優先度が I D L E と設定される。その後、メイン V M からの独立実行のサブ V M として V M 実行通知がなされると、スケジューリング部 1 5 5 は、サブ V M (V M _ L) を I D L E のランキューから R T の

10

20

30

40

50

ランキューに移動させる。さらに、RTランキューに移動されたサブVM (VM__L) は、VM実行通知において付与された優先度pに従って、一定のタイムスライス内で必ずCPU時間が割り当てられることが保証される。

【0171】

これにより、メインVM2のリアルタイム処理などをリアルタイムRTすなわちRTランキューのサブVMが肩代わりして実行することが可能となる。

【0172】

図15は、仮想マシンシステム1を実現するハードウェア構成例を示す図である。

【0173】

図15に示すように、仮想マシンシステム1は、CPU101、主記憶部(メモリ)103、入出力インターフェイス105、外部記憶装置110、入力装置(キーボード等)120を備えるコンピュータ100によって実施することができる。

【0174】

仮想マシンシステム1は、コンピュータ100が実行可能なプログラムによって実施することができる。この場合に、仮想マシンシステム1が有すべき機能の処理内容を記述したプログラムが提供される。提供されたプログラムをコンピュータ100が実行することによって、上記説明した仮想マシンシステム1の各処理部がコンピュータ100上で実現される。

【0175】

なお、コンピュータ100は、可搬型記録媒体から直接プログラムを読み取り、そのプログラムに従った処理を実行することもできる。また、コンピュータ100は、サーバコンピュータからプログラムが転送されるごとに、逐次、受け取ったプログラムに従った処理を実行することもできる。

【0176】

さらに、このプログラムは、コンピュータ100で読み取り可能な記録媒体に記録しておくことができる。

【0177】

上記した仮想マシンシステム1によれば、以下の制御を実現するスケジューリングが可能となる。すなわち、1つのプログラムを複数のプロセスに分割して異なる仮想マシンで実行する場合に、ある仮想マシンのメインプロセスから他仮想マシンのプロセスのサブルーチン呼び出しを行い、サブルーチンの処理終了後に呼び出し元の仮想マシンのプロセスに制御が戻ることが可能となる。そのため、メインプロセスとサブルーチンを1つの仮想マシンで実行するように一体化して実行させることができる。

【0178】

また、ある仮想マシンのプロセスから他仮想マシン上のプロセスの動作について、一定時間ごとにCPUが割り当てられるなどの制御を行うことが可能となる。

【0179】

仮想化技術を用いて、プログラムを複数のプロセスに分割して、別々の仮想マシン上で実行させるモデルにおいて、主にユーザが操作するメインプロセスが、サブプロセスのサブルーチン呼び出しを必要とするタイミングに合わせて、サブプロセスを実行する仮想マシンへのCPU割当が行われ、プログラムを分割せずに実行している場合と同等にCPU割当時間を制御することができる。

また、サブプロセスを独立に実行させ、そのCPU割当についても制御することで、一定時間内のポーリングなどを必要とするリアルタイムプログラムの制御も行うことができる。

【0180】

本願発明の実施態様における特徴を列記すると以下のようなになる。

【0181】

(付記1) プロセスを実行する複数の仮想マシンを仮想マシン管理装置上に構築し、前記仮想マシン管理装置が、前記構築された仮想マシンにCPUを割り当てて、各仮想マ

10

20

30

40

50

シンのプロセスの実行を管理する仮想マシンシステムであって、

前記仮想マシン管理装置は、

プログラムの実行を管理する、プログラム管理部と、

前記プログラムを構成する複数のプロセスを各々で実行する複数の仮想マシンへのCPU割当を、スケジュールにもとづいて管理する、仮想マシンスケジュール管理部とを備えて、

前記プログラム管理部は、

前記プログラムを構成する複数のプロセスのうち当該プログラムの起動要求を受けて実行される第1プロセスを実行する第1仮想マシンから、前記第1プロセスの識別情報、前記プログラムを構成する前記第1プロセス以外の第2プロセス以降のプロセスを特定する情報を含むプログラム構成情報を取得して、前記プログラム構成情報および前記各プロセスを実行する仮想マシンの識別情報をプログラムテーブルに登録する、プログラムテーブル管理部と、

前記仮想マシン管理装置上に生成した仮想マシンで実行されているプロセスから通知を受信する、通知受信部と、

前記プログラムテーブルを参照して、前記通知受信部が受信した通知に対応するプロセスに関する処理の要求を前記仮想マシンスケジュール管理部へ送信する、通知処理部とを備えて、

前記仮想マシンスケジュール管理部は、

前記仮想マシンに対するCPU割当のスケジュールを決定する、スケジューリング部と

前記通知処理部から受信した前記要求が前記プログラムの起動要求である場合に、当該起動対象のプログラムを構成する前記第1プロセスを実行する第1仮想マシンを生成する処理と、前記通知処理部から受信した前記要求が前記第1プロセスに関連する第2プロセス以降のプロセスの実行要求である場合に、前記第2プロセス以降のプロセスを実行する仮想マシンを生成する処理とを行う、仮想マシン状態管理部と、

前記通知処理部からの通知にもとづいて、前記要求により実行が要求される要求先のプロセスを実行する仮想マシンを前記プログラムテーブルにより特定し、前記特定した仮想マシンへCPU割当を切り替える、仮想マシン切替部とを備えて、

前記第1仮想マシンは、

前記第1プロセスの実行時に、前記プログラム構成情報を前記プログラム管理部へ送信する処理と、要求元および要求先のプロセスを特定する情報を含むプロセス呼び出しの通知を前記プログラム管理部へ送信する処理とを行う、仮想マシンモニタ通知部を備え、

前記第2プロセス以降のプロセスを実行する各仮想マシンは、

要求元および前記処理したプロセスの返却先のプロセスを特定する情報を含むプロセス返却の通知を前記プログラム管理部へ送信する、仮想マシンモニタ通知部を備える

ことを特徴とする仮想マシンシステム。

【0182】

(付記2) 前記仮想マシン状態管理部は、前記通知受信部が前記プロセス呼び出しの通知を受信した場合に、前記受信されたプログラム構成情報にもとづくプログラムテーブルを参照して、前記プロセス呼び出しの呼び出し先である前記第2プロセス以降のプロセスを特定して、前記特定したプロセスを実行する仮想マシンの生成を行う

ことを特徴とする前記付記1に記載の仮想マシンシステム。

【0183】

(付記3) 前記仮想マシン状態管理部は、前記通知受信部が、前記第2プロセス以降のプロセスの各属性情報を含むプログラム構成情報を受信した場合に、前記受信されたプログラム構成情報にもとづくプログラムテーブルを参照して、前記第2プロセス以降のプロセスとして前記属性情報に対応するプロセスを実行する仮想マシンの生成を行う

ことを特徴とする前記付記1または2に記載の仮想マシンシステム。

【0184】

(付記４) 前記仮想マシン状態管理部は、前記プログラムの実行に関連するプロセスであって前記プログラムのプロセスと独立して実行される独立プロセスの情報を含むプログラム構成情報を受信した場合に、前記受信されたプログラム構成情報にもとづくプログラムテーブルを参照して、前記独立プロセスを実行する仮想マシンの生成を行い、

前記スケジューリング部は、前記生成された独立プロセスを実行する仮想マシンのスケジューリングを行う

ことを特徴とする前記付記１～３のいずれか一項に記載の仮想マシンシステム。

【０１８５】

(付記５) 前記仮想マシンスケジュール管理部は、

前記仮想マシン毎のＣＰＵ消費時間を取得する処理と、前記プログラムテーブルをもとに、前記第２プロセス以降のプロセスを実行する各仮想マシンの前記ＣＰＵ消費時間を、前記第１仮想マシンのＣＰＵ消費時間に加算する処理とを行う消費時間計算部を備えて、

前記スケジューリング部は、前記仮想マシンのＣＰＵ割当のスケジュールを決定する際に、前記第１仮想マシンのＣＰＵ消費時間として前記消費時間計算部により加算されたＣＰＵ消費時間を用いてスケジュールを決定する

ことを特徴とする前記付記１～４のいずれか一項に記載の仮想マシンシステム。

【０１８６】

(付記６) 前記仮想マシン状態管理部は、前記生成された第１仮想マシンおよび前記第２プロセス以降のプロセスを実行する各仮想マシンに、仮想マシン間のメッセージ通信を行う仮想マシン間通信部を生成する

ことを特徴とする前記付記１～５のいずれか一項に記載の仮想マシンシステム。

【０１８７】

(付記７) 前記スケジューリング部は、前記独立プロセスを実行する仮想マシンが生成されている場合に、所定のスケジュール期間内に前記独立プロセスを実行する仮想マシンのスケジュールを設定し、前記設定後に前記第１仮想マシンおよび他の仮想マシンのスケジュールを決定する

ことを特徴とする前記付記４に記載の仮想マシンシステム。

【０１８８】

(付記８) 前記仮想マシン状態管理部が、前記プログラムの第１プロセスを実行する第１仮想マシンを複数生成かつ前記複数の仮想マシンが実行するプロセスが前記第２プロセス以降のプロセスを共有している場合に、

前記消費時間計算部は、前記プログラムテーブルをもとに、前記複数の第１仮想マシンによって共有されたプロセスを実行する仮想マシンのＣＰＵ消費時間を、前記共有する第１仮想マシンの各々のＣＰＵ消費時間に加算する

ことを特徴とする前記付記５に記載の仮想マシンシステム。

【０１８９】

(付記９) 前記プログラム構成情報は、前記プログラムを構成する前記第２プロセス以降のプロセスの実行イメージを含む情報である

ことを特徴とする前記付記１～８のいずれか一項に記載の仮想マシンシステム。

【０１９０】

(付記１０) 前記プログラム構成情報は、前記プログラムを構成する前記第２プロセス以降のプロセスを実行する仮想マシンの実行イメージを含む情報である

ことを特徴とする前記付記１～８のいずれか一項に記載の仮想マシンシステム。

【０１９１】

(付記１１) プロセスを実行する複数の仮想マシンを構築して、前記構築した仮想マシンにＣＰＵを割り当てて、各仮想マシンのプロセスの実行を管理する仮想マシン管理装置が実行する仮想マシン管理方法であって、

前記仮想マシン管理装置が、プログラムの実行を管理するプログラム管理部と、前記プログラムを構成する複数のプロセスを各々で実行する複数の仮想マシンへのＣＰＵ割当を、スケジュールにもとづいて管理する仮想マシンスケジュール管理部と、プログラムテー

10

20

30

40

50

ブルとを備えて、

前記プログラム管理部が、前記プログラムの起動要求を受信する処理ステップと、
前記仮想マシンスケジュール管理部が、前記プログラムの起動要求が通知された場合に、前記プログラムを構成するプロセスであって前記起動要求を受けて実行する第 1 プロセスを実行する第 1 仮想マシンを生成する処理ステップと、

前記プログラム管理部が、前記第 1 仮想マシンから、前記第 1 プロセスの識別情報、前記プログラムを構成する前記第 1 プロセス以外の第 2 プロセス以降のプロセスを特定する情報を含むプログラム構成情報を取得する処理ステップと、

前記プログラム管理部が、前記プログラム構成情報および前記各プロセスを実行する仮想マシンの識別情報を前記プログラムテーブルに格納する処理ステップと、

10

前記プログラム管理部が、前記第 1 プロセスから、前記第 1 プロセスに関連する第 2 プロセス以降のプロセスの実行要求を受信する処理ステップと、

前記仮想マシンスケジュール管理部が、前記第 2 プロセス以降のプロセスを実行する仮想マシンを生成する処理ステップと、

前記プログラム管理部が、前記第 1 プロセスから前記第 2 プロセス以降のプロセスの呼び出し通知を受信する処理ステップと、

前記仮想マシンスケジュール管理部が、前記プログラムテーブルを参照して、前記呼び出し通知に対応するプロセスを実行する仮想マシンへ CPU 割当を切り替える処理ステップと、

前記プログラム管理部が、前記第 2 プロセスから前記呼び出し通知に関する処理の返却通知を受信する処理ステップと、

20

前記仮想マシンスケジュール管理部が、前記プログラムテーブルを参照して、前記返却通知に対応するプロセスを実行する仮想マシンへ CPU 割当を切り替える処理ステップとを備える

ことを特徴とする仮想マシン管理方法。

【符号の説明】

【 0 1 9 2 】

1 仮想マシンシステム

1 0 仮想マシン管理装置

1 1 プログラム管理部

30

1 1 1 通知受信部

1 1 3 プログラムテーブル管理部

1 1 5 通知処理部

1 3 プログラムテーブル

1 5 仮想マシンスケジュール管理部

1 5 1 仮想マシン切替部

1 5 3 仮想マシン状態管理部

1 5 5 スケジューリング部

1 5 7 消費時間計算部

1 7 仮想マシン間通信部

40

2 第 1 仮想マシン (メイン VM)

2 1 仮想マシンモニタ通知部

2 3 仮想マシン間通信部

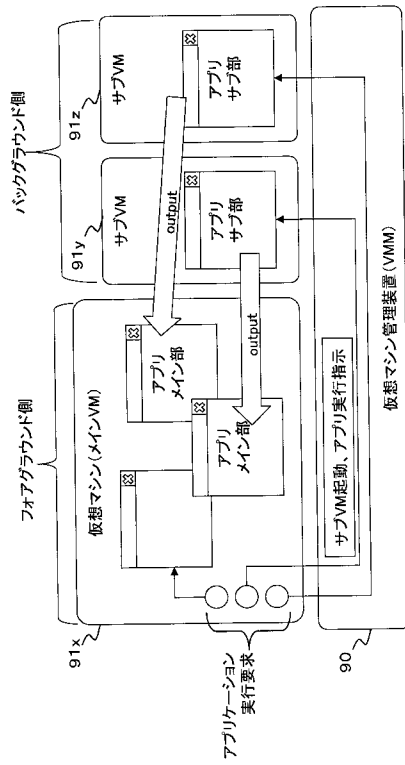
3 仮想マシン (サブ VM)

3 1 仮想マシンモニタ通知部

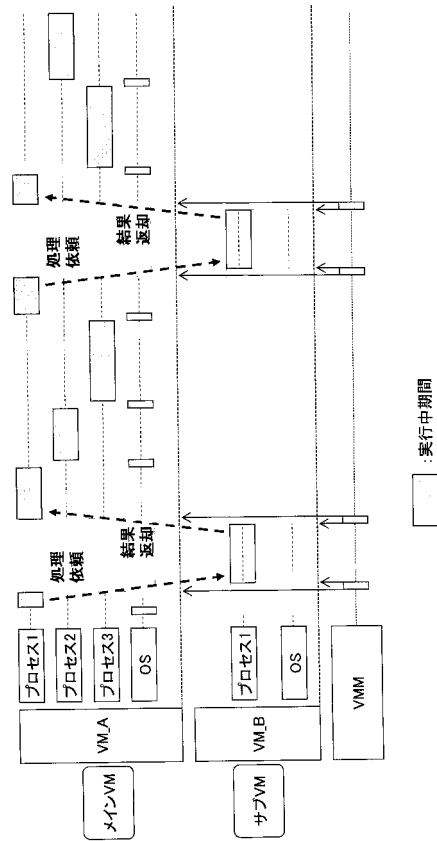
3 3 仮想マシン間通信部

4 仮想マシン (他 VM)

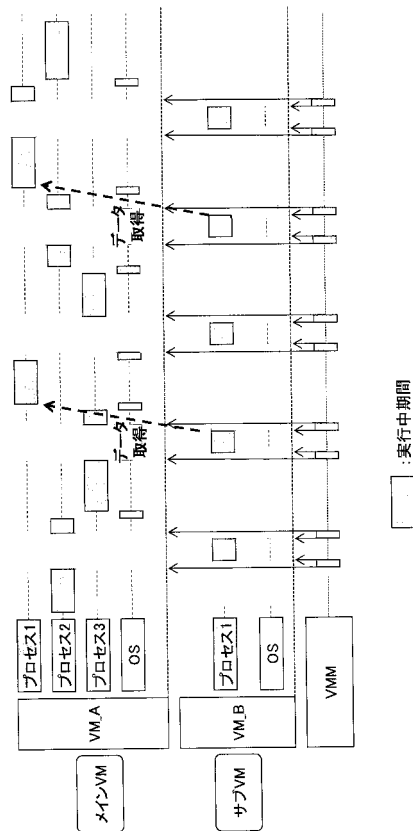
【図 1】



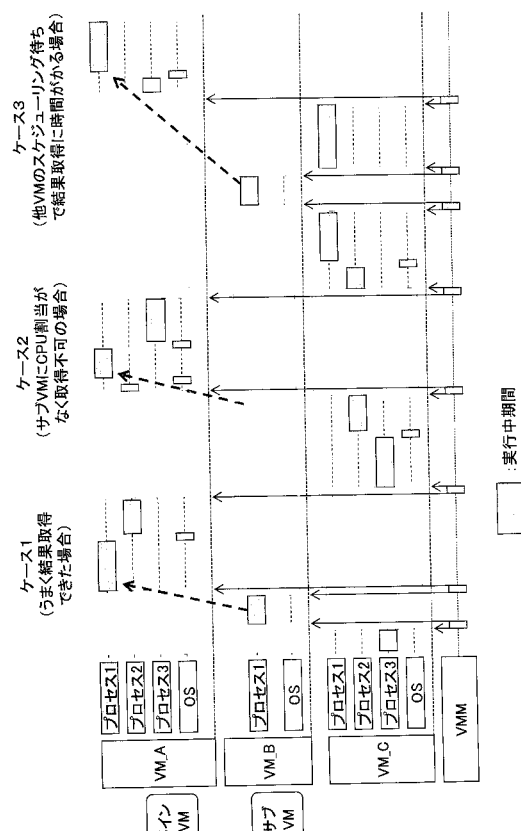
【図 2】



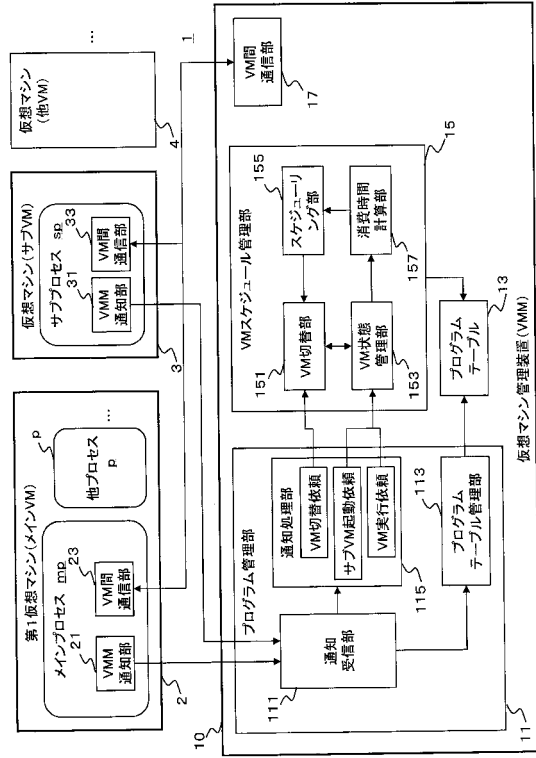
【図 3】



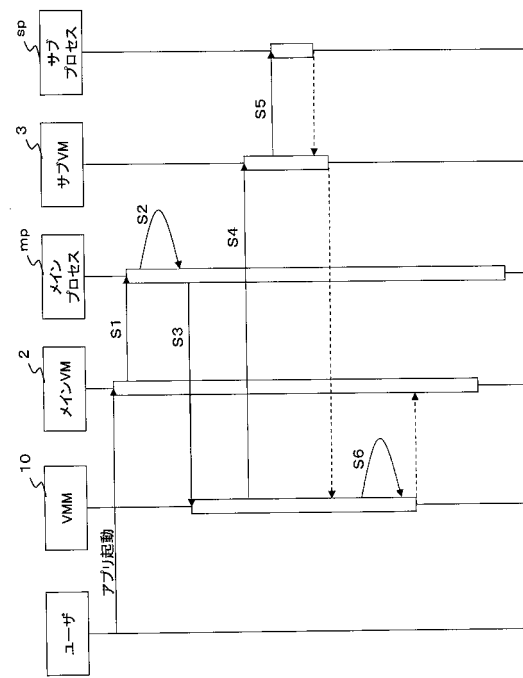
【図 4】



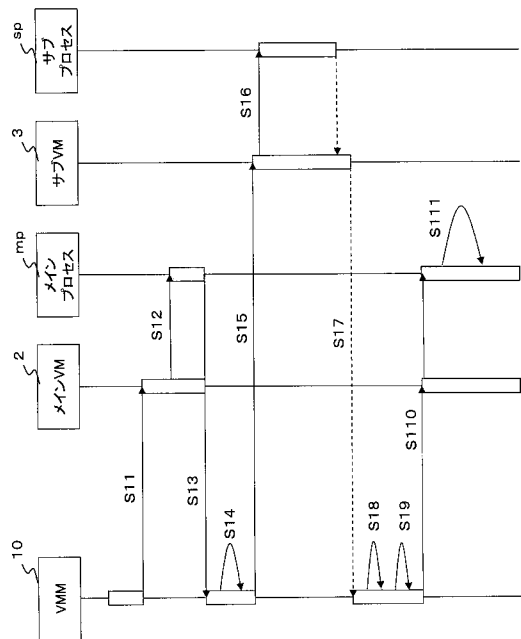
【図 5】



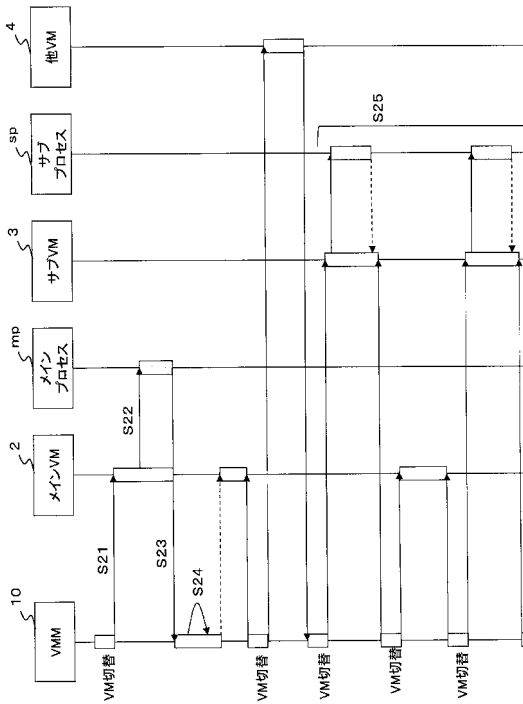
【図 6】



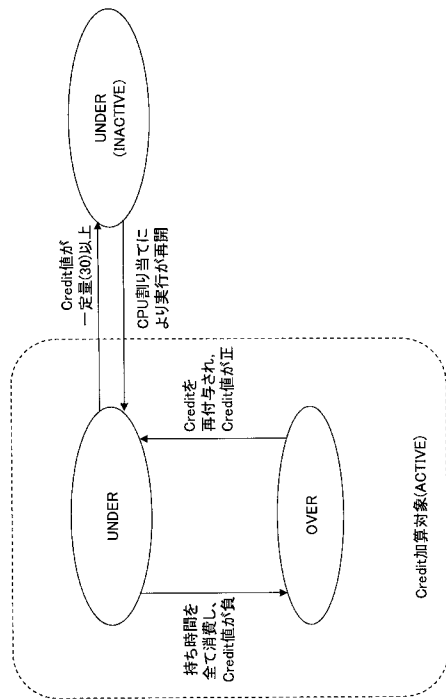
【図 7】



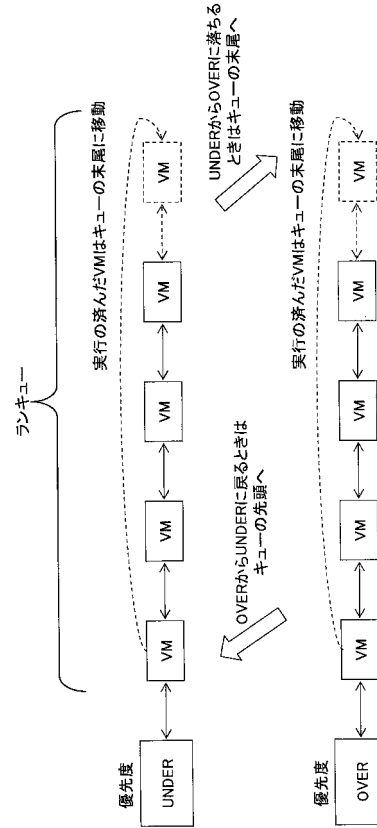
【図 8】



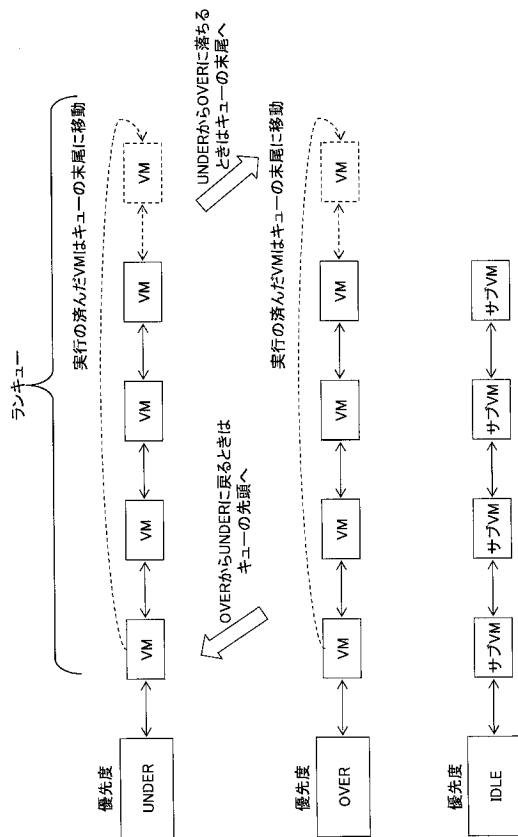
【図 9】



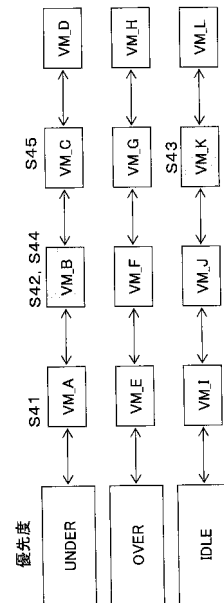
【図 10】



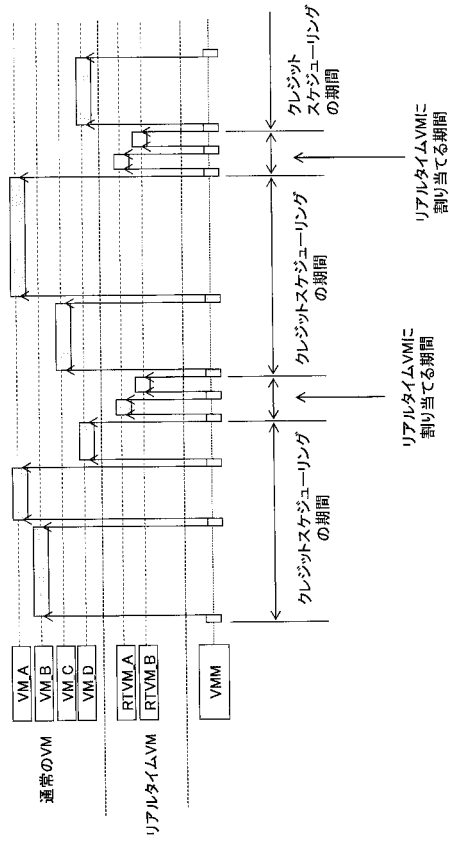
【図 11】



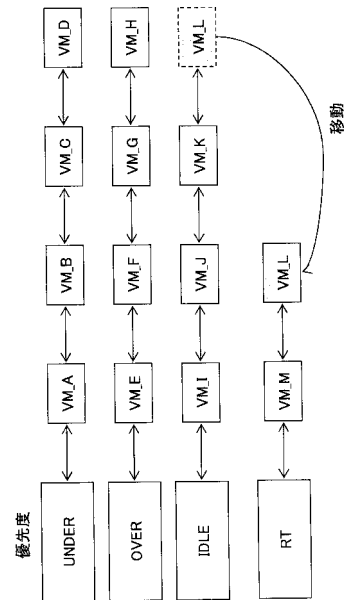
【図 12】



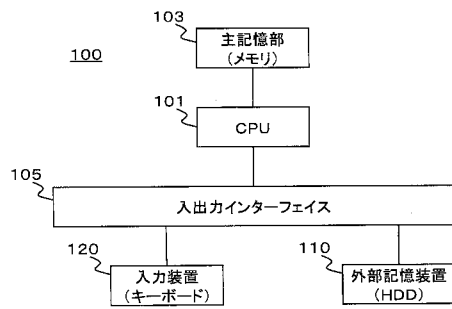
【図 13】



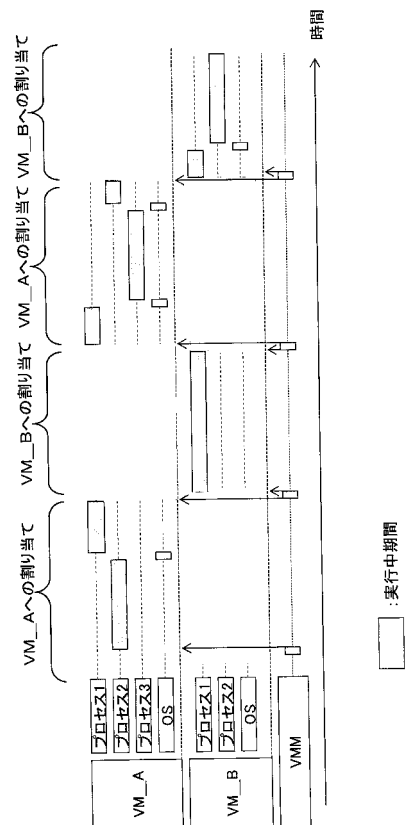
【図 14】



【図 15】



【図 16】



フロントページの続き

(72)発明者 西口 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

審査官 漆原 孝治

(56)参考文献 特開昭61-062939(JP,A)

特開2010-039626(JP,A)

(58)調査した分野(Int.Cl., DB名)

G 0 6 F 9 / 4 8

G 0 6 F 9 / 4 6