

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5820525号
(P5820525)

(45) 発行日 平成27年11月24日(2015.11.24)

(24) 登録日 平成27年10月9日(2015.10.9)

(51) Int.Cl. F I
G06F 9/48 (2006.01) G O 6 F 9/46 4 5 2 A
G06F 9/46 (2006.01) G O 6 F 9/46 3 5 0

請求項の数 8 (全 19 頁)

(21) 出願番号	特願2014-507165 (P2014-507165)	(73) 特許権者	000005108
(86) (22) 出願日	平成24年3月29日 (2012. 3. 29)		株式会社日立製作所
(86) 国際出願番号	PCT/JP2012/058295		東京都千代田区丸の内一丁目6番6号
(87) 国際公開番号	W02013/145199	(74) 代理人	110001689
(87) 国際公開日	平成25年10月3日 (2013. 10. 3)		青稜特許業務法人
審査請求日	平成26年5月21日 (2014. 5. 21)	(72) 発明者	金 成 昊
			日本国東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
		(72) 発明者	大平 崇博
			日本国東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
		審査官	田中 幸雄

最終頁に続く

(54) 【発明の名称】 仮想計算機のスケジュールシステム及びその方法

(57) 【特許請求の範囲】

【請求項 1】

プロセッサとメモリを備え、前記メモリに格納されたホストOSおよび仮想計算機を前記プロセッサで実行する計算機システムにおける仮想計算機のスケジュール方法であって、

前記仮想計算機上に設けた仮想計算機内スケジュール処理部が、

前記仮想計算機上で動作するアプリケーションの優先度に従って優先度グループを作成し、

通知手段が、

前記優先度グループの動作開始と終了の通知を受け、前記優先度グループの動作開始と終了を前記ホストOSへ通知し、

仮想計算機スケジューラ制御部が、

前記優先度グループの動作開始と終了から、前記優先度グループを実行する仮想計算機の処理優先度を決定することを特徴とする仮想計算機のスケジュール方法。

【請求項 2】

前記通知手段は、前記優先度グループの先頭アプリケーションの実行と最後尾アプリケーションの実行により前記優先度グループの動作開始と終了の通知を受けることを特徴とする請求項 1 に記載の仮想計算機のスケジュール方法。

【請求項 3】

前記仮想計算機スケジューラ制御部は、前記優先度グループの動作開始と終了から前記

10

20

優先度グループの処理時間を求め、前記優先度グループの処理時間を基に、前記優先度グループを実行する仮想計算機の処理優先度を決定することを特徴とする請求項 2 に記載の仮想計算機のスケジュール方法。

【請求項 4】

前記仮想計算機スケジューラ制御部に設けた割り込み管理部は、仮想計算機毎に外部装置からの割り込み数を制限することによって、仮想計算機で実行されるアプリケーションが予め定められたデッドラインまでに処理されるリアルタイム性を保証する制御を行うことを特徴とする請求項 2 に記載の仮想計算機のスケジュール方法。

【請求項 5】

プロセッサとメモリを備え、前記メモリに格納されたホスト OS および仮想計算機を前記プロセッサで実行する仮想計算機のスケジュールシステムであって、

前記仮想計算上で動作するアプリケーションの優先度に従って優先度グループを作成する仮想計算機内スケジュール処理部と、

前記優先度グループの動作開始と終了の通知を受け、前記優先度グループの動作開始と終了を前記ホスト OS へ通知する通知手段と、

前記優先度グループの動作開始と終了から前記優先度グループの処理時間を求め、前記優先度グループの処理時間を基に、前記優先度グループを実行する仮想計算機の処理優先度を決定する仮想計算機スケジューラ制御部を備えることを特徴とする仮想計算機のスケジュールシステム。

【請求項 6】

前記通知手段は、前記優先度グループの先頭アプリケーションの実行と最後尾アプリケーションの実行により前記優先度グループの動作開始と終了の通知を受けることを特徴とする請求項 5 に記載の仮想計算機のスケジュールシステム。

【請求項 7】

前記仮想計算機スケジューラ制御部は、前記優先度グループの動作開始と終了から前記優先度グループの処理時間を求め、前記優先度グループの処理時間を基に、前記優先度グループを実行する仮想計算機の処理優先度を決定することを特徴とする請求項 6 に記載の仮想計算機のスケジュールシステム。

【請求項 8】

前記仮想計算機スケジューラ制御部に、仮想計算機毎に外部装置からの割り込み数を制限する割り込み管理部を設け、仮想計算機で実行されるアプリケーションが予め定められたデッドラインまでに処理されるリアルタイム性を保証する制御を行うことを特徴とする請求項 6 に記載の仮想計算機のスケジュールシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、サーバ仮想化環境においてリアルタイム性が必要な仮想計算機の運用する計算機システムに関し、特に CPU リソースの制限がある計算機上の仮想計算機をスケジュールする方法である。

【背景技術】

【0002】

サーバ仮想化技術は、1 台の物理計算機上で複数の仮想計算機 (VM) を実行して、物理計算機の台数を削減することで、運用コスト、計算機の設置場所及び消費電力等を削減できる利点があり、注目を浴びている。その際、仮想計算機内のアプリケーションによってリアルタイム性が要求される。しかし、物理計算機に仮想計算機に割り当てる CPU リソースが十分に存在する場合は、CPU リソースを占有割当するなどの性能保証の手法が存在するが、CPU リソースが不十分な計算機環境では、十分なリアルタイム性保証は困難である。また、計算機と接続されているネットワークデバイス等の外部装置からの割り込みが頻繁に発生する場合、仮想計算機内のアプリケーションの処理遅延が発生し、その状況をさらに悪化させる。しかし、制御システムにおいてはこのような計算機状況でも、

10

20

30

40

50

仮想計算機上のアプリケーションのリアルタイム性保証が要求される。

【先行技術文献】

【特許文献】

【0003】

【特許文献1】特開2011-198027号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

特許文献1に示す従来の仮想計算機のスケジュール方式では、仮想計算機で処理するアプリケーションをスレッド単位に分離し、分離したスレッド単位を複数の仮想計算機で分
担して処理を行うこととし、アプリケーションの処理順に仮想計算機の実行順番をスケ
ジュールしていた。しかし、複数のアプリケーションから構成されるシステムの場合、リソ
ースの占有状態を考慮すると、アプリケーションの処理順序のみに基づいて、仮想計算機
の実行順番を決定するのは困難である。また、それぞれのCPUの負荷（さらには処理能
力）が異なるので複数CPU構成のマルチCPU環境での対応が困難であるため、物理計
算機のCPUリソースの有効な活用ができない。

10

【0005】

そこで、本発明は1つ以上のCPUリソースが利用できる物理計算機環境において、大
規模なアプリケーションを稼働する複数仮想計算機で構成される仮想化システムのスケ
ジュール方法において、仮想計算機で処理するアプリケーションのリアルタイム性を保証で
きる仮想計算機のスケジューリング技術を提供することを目的とする。

20

【課題を解決するための手段】

【0006】

物理計算機及び各仮想計算機は物理NICを共有し、それを介して計算機サーバやコン
トローラ等の外部装置と通信を行っている。また、ホストOSおよび複数の仮想計算機を
搭載する計算機システムにおいて、上記の計算機内部のホストOS上にVMスケジューラ
制御部を設ける。また、各仮想計算機上にはゲストOSが稼働して、仮想計算機上のア
プリケーションの実行処理を行う。ゲストOS上には、前記アプリケーションの実行優先度
別グループでアプリケーションの実行順を決定するVM内スケジュール制御部を設ける。
ただし、ホストOSには各仮想計算機にCPUリソースを割り当てるスケジュール機能を
備え、外部装置からの割り込みを該当仮想計算機へ伝達する割り込み処理機能を備えた仮
想化プログラムを備えていることを前提とする。また、ゲストOSには、ホストOSから
各仮想計算機に割り当てたCPUリソースを仮想計算機内アプリケーションに割り当てて
アプリケーションを実行できるスケジュール機能、及び、ホストOS上の仮想化プログラ
ムから通知された割り込みを処理する機能を備えていることを前提とする。

30

【0007】

まず、ゲストOS上で稼働するVM内スケジュール制御部は、仮想計算機内のアプリケ
ーションを優先度別にグループに分ける。また、各グループの先頭と最後尾にはVM内ス
ケジュール制御部が用意するグループの中で最初に実行するアプリケーションと最後に実
行するアプリケーションが登録される。

40

【0008】

前記優先度グループは複数存在する。各優先度グループは優先度を表す識別子を持つ。
本発明では、「L1」が最高優先度グループとし、「Ln」が最低優先度グループとする。
VM内スケジュール制御部は「L1」から「Ln」までのいずれかのグループに属する
全アプリケーションは、あるデッドラインで処理されることを保証するように各優先度グ
ループの処理時間が管理される。各グループの処理時間は、前記グループ先頭アプリケ
ーションの起動時間と、最後尾アプリケーションを終了時間で管理している。前記デッド
ライン設計は、本発明を提供するシステムの要件によって調整する必要がある。

【0009】

なお、ホストOS上のVMスケジューラ制御部へ各仮想計算機の内部スケジュール状態

50

を通知するために、各ゲストOSのVM内スケジューラ制御部は、ホストOS上のVMスケジューラ制御部との通知手段を備える。ホストOS上のVMスケジューラ制御部が各仮想計算機の内部スケジューラ状態を把握するために、ゲストOS上のVM内スケジューラ制御部は、仮想計算機内の優先度グループの先頭アプリケーションの開始と最後尾アプリケーションの終了をホストOS上のVMスケジューラ制御部に通知する。

【0010】

ただし、優先度が最も低い「Ln」優先度グループにおいては、前記の通知は行わない。仮想計算機からホストOSへの通知のオーバーヘッドは大きいため、低優先度のグループの開始・終了は管理しないことで、前記オーバーヘッドの低減を実施する。「Ln」以外の高優先度グループからのアプリケーションの開始と終了の時間の通知によって、ホストOS上のVMスケジューラ制御部は各仮想化計算機で実行されているアプリケーションの優先度が把握できる。

10

【0011】

仮想計算機内の実行優先度から、CPUリソースに割り当てるべき仮想計算機を選択する。そのため、ホストOS上のVMスケジューラ制御部は、各仮想計算機の実行状態として優先度グループの優先度を常に記録管理し、高優先度グループの終了通知をもって通知元の仮想計算機の優先度を下げていく。したがって、複数の仮想計算機の中で最優先にCPUリソースを割り当てるべき仮想計算機が把握できる。

【0012】

なお、各仮想計算機が守るべきデッドラインを監視し、現状の実行優先度が低い仮想計算機であってもデッドラインに近い仮想計算機は最高優先度に調整し、かならずデッドラインを守るように調整する。「Ln」のような優先度が低いグループは、仮想計算機内のVM内スケジューラ制御部からホストOSへの、先頭アプリケーションの開始及び最後尾アプリケーションの終了の通知によるデッドラインによるスケジューラ制御ではなく、後述するCPUリソースを割り当てるQuota時間でスケジューラ制御を行う。

20

【0013】

一方、割り込み処理においては、ホストOS上のVMスケジューラ制御部内に割り込み管理制御部を設け、仮想計算機のデッドライン内で許容される割り込み数を制限する。割り込み数制限は、各仮想計算機のデッドラインの許容値に対して1つの割り込み処理時間を割った数を割り込み数の制限値として管理する。ホストOS上の仮想化プログラムからの割り込み通知に対して割り込み数制限値内であれば、仮想計算機へ割り込みの通知を行う。

30

【0014】

割り込みの通知の際には、該当仮想計算機の現在の実行優先度にかかわらず最高優先度に変更し、ホストOSがCPUのリソースを該当仮想計算機に割り当てるようにする。該当仮想計算機内では、ゲストOSで割り込み処理が行われ、ゲストOSのスケジューラはVM内部スケジューラ制御部で設定している優先度グループのアプリケーションを選択し、CPUリソースを割り当てる。

【0015】

しかし、そのアプリケーションより高い実行優先度のアプリケーションを実行する仮想計算機にCPUリソースが割り当てられなくなるため、割り込み通知による最高優先度設定には、その優先度を維持するQuota時間を設ける。即ち、割り込み処理に割り当てられていた時間がQuota時間となると元の優先度に調整され、常にホストOSは最高優先度の仮想計算機にCPUリソースを割り当てるようにする。

40

【0016】

以上のように、仮想計算機内のVM内スケジューラ制御部による優先度グループでのアプリケーション管理方法と、ホストOS上のVMスケジューラ制御部によって、仮想計算機内で実行している優先度グループに従って、CPUリソースを仮想計算機へ割り当てることで、CPUリソースの制限があるシステムにおいても仮想計算機内のアプリケーションのリアルタイム性保証が可能である。また、ホストOS上のVMスケジューラ制御部に

50

割り込み管理制御部を設けることで、割り込み処理の高負荷の計算機環境においても仮想計算機内のアプリケーションのリアルタイム保証が可能である。

【発明の効果】

【0017】

本発明によれば、VMスケジューラ制御部が仮想計算機内の処理アプリケーションの優先度グループのデッドラインによって仮想計算機のCPUリソース割当を管理するため、CPUリソースが制限される計算機システムにおいてもリアルタイム性保証が可能となる。また、外部割り込みが多く発生する外部ネットワークを利用する仮想計算機の運用においても、仮想計算機内の処理アプリケーションの優先度グループのデッドラインによって割り込み通知・処理の数制限を設けるため、リアルタイム性の保証が可能となる。

10

【図面の簡単な説明】

【0018】

【図1】本発明の仮想計算機のスケジューラ制御の全体構成図である。

【図2】本発明のVM内スケジューラ管理テーブルの構成図である。

【図3】本発明のVM管理テーブルの構成図である。

【図4】本発明のタイマ・管理テーブルの構成図である

【図5】本発明の割り込み管理テーブルの構成図である。

【図6A】本発明の通知IF処理の概要図である。

【図6B】本発明の通知IF処理の概要図である。

【図7】本発明のVM内スケジューラプログラムのフローチャートである。

20

【図8】本発明のVM内スケジューラプログラムの終了処理のフローチャートである。

【図9】本発明の仮想化システムの全体フローチャートである。

【図10】本発明のVMデッドライン管理プログラムのフローチャートである。

【図11】本発明のVMデッドライン管理プログラムの実行VM選択処理のフローチャートである。

【図12】本発明のタイマ・イベント管理プログラムのフローチャートである。

【図13】本発明の外部イベント処理のフローチャートである。

【図14】本発明のQuota管理処理のフローチャートである。

【図15】本発明の割り込み管理プログラムのフローチャートである。

【発明を実施するための形態】

30

【0019】

以下、本発明の一実施の形態を、図面を用いて説明する。なお、本発明が実施の形態に制限されることは無く、本発明の思想に合致するあらゆる応用例が本発明に該当する。

【0020】

(全体構成図)

図1は本発明の実施形態の仮想マシンのスケジューラ制御の全体構成図である。仮想マシンのスケジューラ制御は、一つの仮想化システム1及び一つ端末2及び外部ネットワーク3から構成される。仮想化システム1は、1台の計算機で構成されている。計算機100は、メモリ101、プロセッサ102、記憶装置103、及び外部通信インタフェース104を備える。メモリ101内には、ホストオペレーティングシステム400(図中、ホストOSと略記)、VMスケジューラ制御プログラム300、及び、一つまたは複数の仮想マシン環境200(図中、VMと略記)が保持される。ホストOS400内には仮想化プログラム401が保持される。

40

【0021】

VMスケジューラ制御プログラム300は、VM内スケジューラ管理テーブル301、VM管理テーブル302、タイマ・管理テーブル303、割り込み管理テーブル304、VM内スケジューラ入力プログラム305、VMデッドライン管理プログラム306、タイマ・管理プログラム307、及び割り込み管理プログラム308から構成される。VM200は、アプリケーション211(図中、APと略記)、VM内スケジューラ212、ゲストオペレーティングシステム213(図中、ゲストOSと略記)、及び通知インタフェ

50

ース 214 (図中、インタフェースを I / F と略記) から構成される。図 1 では、計算機 100 内に VM 200 として実線枠で示した VM 1 (200 - A)、VM 2 (200 - B) 及び点線枠で示したその他の VM 200 の例が示されているが、特に VM の台数は限定されるものではない。

【 0022 】

端末 2 と計算機 100 は外部ネットワーク 3 を介して接続される。計算機 100 では、外部ネットワーク 3 は外部通信 I / F 104 に接続される。

【 0023 】

端末 2 は外部ネットワーク 3 を介して仮想化システム 1 との通信を行うことができる。

【 0024 】

計算機ノード 100 の記憶装置 103 は、ホスト OS 400、VM スケジューラ制御プログラム 300 (プログラムと各種テーブル情報)、さらに、VM 200 を構成する AP 211、VM 内スケジュール 212、ゲスト OS 213、及び通知 I / F 214 を格納する。プロセッサ 102 は記憶装置 103 からホスト OS 400、VM スケジューラ制御プログラム 300、VM 200 の構成要素をメモリ 101 に展開して実行し、また、外部通信 I / F 104 からの割込みを処理する。外部通信 I / F 104 は、外部ネットワーク 3 を通じて端末 2 とのデータの送受信を行う。

【 0025 】

仮想化プログラム 401 は、VM 200 を作成して管理するためのプログラムであり、外部通信 I / F 104 から割り込みを、通知 I / F 214 を介して VM 200 に通知する。ホスト OS 400 は、仮想化プログラム 401 を用いて、VM スケジューラ制御プログラム 300 や VM 200 の実行を制御し、外部通信 I / F 104 や記憶装置 103 へのアクセス制御を行う。

【 0026 】

VM 200 は、固有のプログラムやオペレーティングシステムが動作することが可能な、仮想化プログラム 401 により作成された仮想的な実行環境である。複数の VM 200 が存在する場合に、異なる VM 200 内で動作するプログラムやオペレーティングシステムは同じでも良いし、異なっても良い。また、異なる VM 200 内で動作するプログラムやオペレーティングシステムが互いに直接影響を与え合うことはない。VM 200 は、仮想化システム 1 内で一意の識別子を持ち、例えば、図 1 中の VM 200 は「VM 1」、あるいは「VM 2」の識別子を保有する。

【 0027 】

AP 211 は、VM 200 内で動作アプリケーションプログラムである。AP 211 として、AP 1 - 1 (211 - A)、AP 1 - n (211 - B) が示されているが、VM 200 内で動作する AP 211 の数を限定するものではない。ゲスト OS 213 は、VM 200 環境内で VM 内スケジュール 212 を利用して AP 211 を制御する。通知 I / F 214 は、VM 内スケジュール 212 の処理結果をゲスト OS 213 から受け取り VM スケジューラ制御プログラム 300 へ出力する。

【 0028 】

VM スケジューラ制御プログラム 300 は、VM 200 で稼働するときに、VM 内スケジュール 212 から、VM 200 内の AP 211 のスケジュール情報を、通知 I / F 214 を介して受け取る。VM スケジューラ制御プログラム 300 は、AP 211 のスケジュール情報を、VM 内スケジュール入力プログラム 305 に入力する。VM 内スケジュール入力プログラム 305 は、AP 211 のスケジュール情報を VM 内スケジュール管理テーブル 301 に格納する。

【 0029 】

VM 200 内で動作する VM 内スケジュール 212 は、AP 211 の動作優先度に応じてグループ化してプロセッサ 102 での実行時間をスケジュールする。動作優先度によるスケジュール方法は後述する。

【 0030 】

10

20

30

40

50

VM200内で動作するVM内スケジュール212は、動作優先度グループのAP211の動作開始と終了を、通知IF214を介してVMスケジューラ制御プログラム300で動作するVMデッドライン管理プログラム306に通知する。VMデッドライン管理プログラムは、AP211の動作開始と終了を元にVM200の優先度の調整を行う。また、割り込みテーブル304に、動作開始と終了の時間を実行時間のデッドラインとして書き込む。VM200の優先度に応じて仮想化プログラム401はVM200を実行する。

【0031】

VM200内で動作するVM内スケジュール212は、最高優先度グループのAP211のタイマ-登録をタイマ-イベント管理プログラム307に通知IF214を介して通知する。タイマ-イベント管理プログラム307は、タイマー制御のための情報をタイマ-管理テーブル303に登録し、仮想化プログラム401にタイマ-制御を委ねる。仮想化プログラム401は、タイマ-が発火(起動)するとタイマ-イベント管理プログラム307に通知する。タイマ-イベント管理プログラム307は、タイマ-イベントを割り込み管理プログラム308に通知する。割り込み管理プログラム308は、通知IF214を介してVM200にタイマ-イベントを通知する。ゲストOS213が割り込み管理プログラム308からタイマ-イベントを処理し、処理結果を、タイマ-登録を行ったAP211に通知する。

【0032】

割り込み管理プログラム308は、外部IF104からの割り込みを、通知IF214を介してVM200に通知する。割り込みはゲストOS213で処理する。

【0033】

VMスケジューラ制御プログラム300の詳細は、図7と共に後述する。

【0034】

(VM内スケジューラ管理テーブル)

図2は本発明の実施形態のVM内スケジューラ管理テーブル301の構成を示す図である。VM内スケジューラ管理テーブル301は、VM内スケジュール212とVMスケジューラ制御プログラム300内に保持され、計算機100内に作成されているVM200-AやVM200-Bの各VM200のAP211のスケジュール情報を保持する。

【0035】

VM識別子3101には、計算機100内でVM200を一意に識別する情報を格納する。周期3102は、各VM200で動作する全AP211が実行される時間を格納する。図2では、VM200-AのVM1で動作するAP1-1からAP1-nまで実行するのに1秒(1000ms)かかることを意味している。

【0036】

優先度レベル識別子3103は、AP211をゲストOS213がAP211を実行する順位を決める実行優先度別にグループ化し、そのグループを一意に識別する情報を格納する。図2では、L1グループの優先度が最も高く、次にL2、次にL3、次にLnの順に優先度が高い。

【0037】

また、グループに属するAP211は、同グループ内で最初に実行するAP211から最後に実行するAP211がリング状(円環)状となるように構成されている。同一グループに属しているAP211をリング状に構成するため、VM200には、各優先度グループの最初と最後のAP211を事前に登録しておく必要がある。最初と最後のAP211は、ユーザが登録するアプリケーションとは異なり、VM内スケジュール213が登録されたAP211である。

【0038】

レベルリング開始終了識別子3104には、グループに属する最初のAP211が開始することを知らせる「S」と最後に実行するAP211の終了を示す「E」が格納される。AP211がグループの最初と最後のAP211ではない場合はレベルリング開始終了識別子3104に「N」が格納される。

10

20

30

40

50

【0039】

タスク識別子3105には、グループ内のAP211を一意に識別する情報を格納する。優先度3106には、ゲストOS213がAP211を実行する順位を決めるAP211の実行優先度情報を格納する。ランタイム3107には、AP211がゲストOS213で実行された時間を「ms」単位で格納する。順序制御タスク識別子3108には、AP211の実行が終了した場合、次に実行するAP211のタスク識別子3105を格納する。順序制御タスク識別子3108にタスク識別子3105「E」以外の情報が格納されている場合は、該当するAP211を実行する。タスク識別子3105に「E」の情報が格納され、レベルリング開始終了識別子3104に「N」が格納されている場合は同グループの他AP211の実行を行う。

10

【0040】

Quota3109には、優先度レベル識別子3013が「Ln」に属しているAP211の実行に割り当てた時間を格納する。

【0041】

VM内スケジュール212の詳細は図7と共に後述する。

【0042】

(VM管理テーブル)

図3は本発明の実施形態のVM管理テーブル302の構成を示す図である。VM管理テーブル302は、仮想化プログラム401がVM200を実行するために用いる。

【0043】

VM識別子3201には、計算機100内でVM200を一意に識別する情報を格納する。優先度識別子3202には、ゲストOS213が実行しているAP211が属している優先度レベル識別子3102を格納する。優先度識別子3202は、VM内スケジュール212が後述する通知フォーマット500に格納する情報から取得する。レベルリング開始終了識別子3203には、優先度レベル識別子3103の優先度グループの最初AP211の開始「S」または、最後のAP211の終了「E」の情報を格納する。「I」は優先度IDの変更を表わす。

20

【0044】

デッドライン3204は、優先度レベル識別子3103の優先度グループの最初のAP211の開始から最後のAP211の終了までの時間を「ms」単位で格納する。タスク識別子3205は、通知フォーマット500に格納する情報から取得する。

30

【0045】

実行状態3206には、仮想化プログラム401が実行するVM200の実行状態を格納する。仮想化プログラム401がVM200を実行する場合、実行状態3206には「R」と格納する。仮想化プログラム401によって実行してないVM200は「0」を格納する。プロセッサ102がマルチコア構成の場合は、仮想化プログラム401によって実行されるVM200は複数個となる。

【0046】

ランタイム3207には、仮想化プログラム401によって実行されたVM200の実行時間が「ms」単位で格納する。Quota3208には、ゲストOS213が実行しているAP211の優先度グループが「Ln」の場合、仮想化プログラム401によって該当VM200を実行する時間が格納される。また、後述するタイマ-イベント管理プログラム307と割り込み管理プログラム308の処理では、優先度グループが「Ln」以外の優先度であってもQuota3208には実行時間を格納する。

40

【0047】

タイマ-イベント管理プログラム307と割り込み管理プログラム308によって格納するQuota3208の情報は、タイマ-イベントと割り込みがVM200のゲストOS213によって処理された後、ゲストOS213によって実行されるAP211の優先度グループが優先度識別子3202と一致しないことがある。特に、「Ln」優先度グループのAP211が実行される場合、開始終了がVMデッドライン管理プログラム306

50

に通知されないため、VM管理テーブル304の優先度識別子3202が「Ln」より高い優先度グループのVM200がある場合、優先度による実行順序が守れなくなる。優先度による実行順序を守るため、タイマ-イベントと割り込み通知にはQuota3208を格納する必要がある。

【0048】

TimeStamp3209にはランタイム3207を格納する。後述するVMデッドライン管理プログラム307がQuota3208の時間が経過したかを判断するためである。元優先度3210には、タイマ-イベントと割り込み処理では優先度識別子3202に優先度「L1」を格納する際に優先度識別子3202の情報を格納する。タイマ-イベントと割り込み処理は、制御システムのリアルタイム性保証のためにVM200が優先的に処理するようにする必要があるため、VM200の実行優先度に関係なく最高優先度「L1」にする必要がある。Quota3208の経過後は、元優先度3210の優先度を優先度識別子3202に格納する。

10

【0049】

(タイマ-管理テーブル)

図4は本発明の実施形態のタイマ-管理テーブル303の構成を示す図である。タイマ-管理テーブル303は、VM200のゲストOS213が実行するAP211の中で実行優先度グループ「L1」に属しているAP211のタイマ-登録を仮想化プログラム401で行うために必要なテーブルである。「L1」のグループのAP211はリアルタイム性が必要なアプリケーションである。そのため、VM200のゲストOS213ではなく、ホストOS400の仮想化プログラム401で処理を行う。「L1」のグループに属しているAP211のタイマ-登録を、VM内スケジュール212が通知IF214を介してタイマ-イベント管理プログラム307に通知する。

20

【0050】

VM識別子3301には、計算機100内でVM200を一意に識別する情報を格納する。優先度レベル3302には優先度レベル識別子3103を格納する。タスク識別子3303にはAP211の識別する情報を格納する。インタバール3304にはタイマ-イベントの発火時間を格納する。カウンタ3305にはタイマ-イベントの単位時間(例えば1秒間)当たりの発火頻度を格納する。

【0051】

(割り込み管理テーブル)

図5は本発明の実施形態の割り込み管理テーブル304の構成を示す図である。仮想化システムにおいては前記AP211のリアルタイム性を保証するにはVM200が処理する割り込み数を制限する必要がある。後述する割り込み管理プログラム308によって管理される。

30

【0052】

VM識別子3401には計算機100内でVM200を一意に識別する情報を格納する。周期3402にはVM内スケジュール管理テーブル301の周期3102の情報を格納する。優先度レベル3403にはタイマ-管理テーブル303で管理しているVM識別子3401を格納する。デッドライン3404にはVM管理テーブル302のデッドライン3204を格納する。デッドライン許容値3405には優先度レベル3404に属するAP211のグループの実行デッドライン時間からリアルタイム性を保証できる許容時間を「ms」で格納する。

40

【0053】

1つ割り込み処理時間3406にはVM200のゲストOS213が外部IF102からの1つの割り込みを処理する時間を「ms」で格納する。割り込み数臨界値3407にはデッドライン許容値3405に対して1つ割り込み処理時間3406を割った数の90%の値を格納する。90%を格納することで、10%の余裕値を設ける。周期デッドライン3408にはVM内スケジュール管理テーブル301の周期3102を格納する。割り込み累積3409には、周期デッドライン3408の時間内で発生した割り込み数を格納

50

する。

【 0 0 5 4 】

(通知 I F 処理概要)

図 6 A は本発明の実施形態の通知 I F 2 1 4 の処理概要である。図 6 B は、通知フォーマット 5 0 0 の構成を示す。優先度グループの最初の A P 2 1 1 の開始と最後の A P 2 1 1 の終了の際に通知フォーマット 5 0 0 に A P 2 1 1 の情報を格納し (5 0 3)、V M スケジューラ制御プログラム 3 0 0 に通知する (5 0 2)。通知によってホスト O S 4 0 0 の仮想化プログラム 4 0 1 が V M 2 0 0 の実行順序を決定する。外部通信 I F 1 0 4 からの割り込みとタイマ - イベントは仮想化プログラム 4 0 1 が処理し、V M スケジューラ制御プログラム 3 0 0 に通知する。V M スケジューラ制御プログラム 3 0 0 は V M 管理テーブル 3 0 2 の優先度識別子 3 2 0 2 に「 L 1 」を設定し、Q u o t a 3 2 0 8 を設定し、通知 I F 2 1 4 で V M 2 0 0 に通知する (5 0 1)。

10

【 0 0 5 5 】

図 6 B の通知フォーマット 5 0 0 における、V M 識別子 5 0 0 1、周期 5 0 0 2、優先度レベル 5 0 0 3、及び開始時のタスク識別子 5 0 0 4 は、図 2 の V M 内スケジューラ管理テーブル 3 0 1 のそれぞれの項目に対応して、優先度グループの最初の A P 2 1 1 の開始時に、最初の A P 2 1 1 によって設定され、開始終了 5 0 0 5 に「 S 」が設定される。V M スケジューラ制御プログラム 3 0 0 によって、図 5 の割り込み管理テーブル 3 0 4 の項目 3 4 0 6 及び 3 4 0 7 に基づいて、1 つ割り込み処理時間 5 0 0 6、及び割り込み数臨界値 5 0 0 7 が設定される。最後の A P 2 1 1 の終了時に、終了時のタスク識別子 5 0 0 4 が設定され、開始終了 5 0 0 5 に「 E 」が設定される。

20

【 0 0 5 6 】

(V M 内スケジューラのフローチャート)

図 7 は本発明の実施形態の V M 内スケジューラ 2 1 2 のフローチャートである。ホスト O S 4 0 0 及びプロセッサ 1 0 2 は、V M スケジューラ制御プログラム 3 0 0 及び V M 2 0 0 を計算機 1 0 0 のメモリ 1 0 1 内に展開すると、V M 2 0 0 の処理を開始する。V M 2 0 0 が処理を始めるとゲスト O S 2 1 3 の処理を始め、V M 内スケジューラ 2 1 2 の処理を始める。

【 0 0 5 7 】

V M 内スケジューラ 2 1 2 は、記憶装置 1 0 3 から V M 内スケジューラ管理テーブル 3 0 1 をメモリ 1 0 1 内に読み込む (S 1 0 0 1)。V M 内スケジューラ 2 1 2 は V M 内スケジューラ管理テーブル 3 0 1 の優先度レベル識別子 3 1 0 3 の情報に格納している「 L n 」レベルグループを除く全優先度レベルグループに対して開始用 A P 2 1 1 と終了用 A P 2 1 1 を登録する (S 1 0 0 2)。

30

【 0 0 5 8 】

V M 内スケジューラ管理テーブル 3 0 1 の優先度 3 1 0 6 の情報から最高優先度の A P 2 1 1 を、タスク識別子 3 1 0 5 から選択する (S 1 0 0 3)。優先度 3 1 0 6 の情報の数字が大きいほど優先度は高い。優先度が同じであるタスク識別子 3 1 0 5 が複数ある場合 (S 1 0 0 4 の Y)、V M 内スケジューラ管理テーブル 3 0 1 のランタイム 3 1 0 7 の情報からランタイムが最小のタスク識別子 3 1 0 5 を選択する (S 1 0 1 1)。

40

【 0 0 5 9 】

前記タスク識別子 3 1 0 5 の属する優先度レベル識別子 3 1 0 3 が「 L n 」かどうかを判断する (S 1 0 1 2)。優先度レベルが「 L n 」の場合、V M 内スケジューラ 2 1 2 は、前記タスク識別子 3 1 0 5 の A P 2 1 1 を、ゲスト O S 2 1 3 が起動するアプリケーションとして登録する (S 1 0 2 1)。

【 0 0 6 0 】

優先度レベルが「 L n 」ではない場合、通知フォーマット 5 0 0 に、V M 識別子 5 0 0 1 と周期 5 0 0 2 と優先度レベル 5 0 0 3 とタスク識別子 5 0 0 4 を V M 内スケジューラ管理テーブル 3 0 1 から取得して該当情報を格納する。開始終了 5 0 0 5 には「 S 」を格納する (S 1 0 3 2)。通知 I F 2 1 4 を介して通知フォーマット 5 0 0 を V M スケジュー

50

ーラ制御プログラム300にAP211の開始を送信する(S1033)。VM内スケジュール212は、前記タスク識別子3105のAP211を、ゲストOS213が処理するアプリケーションとして登録する(S1034)。終了処理を行う(S1035)(図8参照)。その後、VM内スケジュール212は、優先度レベル識別子3103のグループに属しているAP211をすべて起動する。

【0061】

(終了処理のフローチャート)

図8は本発明の実施形態の図7に示した終了処理(1035)のフローチャートである。VM内スケジュール212は、VM内スケジュール管理テーブル301の優先度レベル識別子3104が優先度グループ内にあるAP211すべてを、ゲストOS213が起動するアプリケーションとしてVM内スケジュール212に登録する。

10

【0062】

優先度レベル識別子3104の優先度グループ内の最初のAP211がゲストOS213によって処理された際のランタイムを、図2のVM内スケジュール管理テーブル301のランタイム3107に加算する(S2002)。AP211が、優先度レベル識別子3104の優先度グループ内の最後AP211かどうかを、VM内スケジュール管理テーブル301のレベルリング開始終了識別子3104から判断する(S2003)。開始終了識別子3104の情報が「E」であれば、これを通知フォーマット500に設定して、優先度グループ内AP211のアプリケーションの処理が終了したことを、VMスケジューラ制御プログラム300に通知し、処理を終了する(S2011、S2012)。

20

【0063】

開始終了識別子3104の情報が「E」ではない場合、VM内スケジュール管理テーブル301の順序制御タスク識別子3108に、順序制御タスク識別子3108から順序に処理するAP211が存在するかどうかを判断する(S2021)。順序制御タスク識別子3108にAP211のタスク識別子3105の情報が格納されていれば、該当AP211を、ゲストOS213が起動するアプリケーションとしてVM内スケジュール212に登録する(S2031)。順序制御タスク識別子3108にAP211のタスク識別子3105の情報が「0」であれば、同グループ内のAP211から、ランタイム3107が最小のタスク識別子3105を選択して、該当AP211を、ゲストOS213が起動するアプリケーションとしてVM内スケジュール212に登録する(S2022、S2023)。その後、次のタスクについて上記の処理を行う。

30

【0064】

(仮想化システムの全体フローチャート)

図9は本発明の実施形態の仮想化システム1の全体フローチャートである。仮想化システム1の全体処理は、VMスケジューラ制御プログラム300の起動処理及び、VMデッドライン管理プログラム306、タイマ-イベント管理プログラム307、割り込み管理プログラム308より成り立つ。

【0065】

ホストOS400及びプロセッサ102はVMスケジューラ制御プログラム300及びVM200を計算機100のメモリ101内に展開すると、VMスケジューラ制御プログラム300の処理を開始する。VMスケジューラ制御プログラム300の処理は処理が開始されると、記憶装置103からVM内管理テーブル301、VM管理テーブル302、タイマ-イベント管理テーブル303、及び割り込み管理テーブル304をメモリ101内に読み込み(S101)、VM200を起動する(S102)。VMスケジューラ制御プログラム300はテーブルの読み込みS102の後、VMデッドライン管理プログラム306(図10参照)、タイマ-イベント管理プログラム307(図12参照)、及び割り込み管理プログラム308(図15参照)のそれぞれのプログラムを立ち上げ、通知又は割り込みがあるまで待機する。

40

【0066】

(VMデッドライン管理プログラムのフローチャート)

50

図10は本発明の実施形態のVMデッドライン管理プログラム306のフローチャートである。VM200のVM内スケジュール212から通知があれば(S301のY)、通知IF214からの通信フォーマット500の開始終了情報5005を読み取る(S302)。

【0067】

通信フォーマット500の開始終了情報5005が終了の「E」であれば(S303のY)、通信フォーマット500のタスク識別子5004と一致するVM管理テーブル302のタスク識別子3205を検索し、検索したタスク識別子3205に関するレベルリング開始終了識別子3203に「E」を書き込む(S311)。実行状態3206には「0」セットし、前記VM200が実行状態ではないことを示す(S312)。デッドライン3204に、ホスト400の時刻とTimestamp3209との差をデッドラインとして格納する。ラインタイム3207には、デッドライン3204を加算する(S313)。前記VM200の優先度識別子3202のレベルを1つ下げる(S314)。例えば、優先度3202のレベルが「L1」であれば「L2」を格納する。優先度3202のレベルが「Ln」であれば(S315のY)、Timestamp3209にホストOS400の時刻を格納する(S316)。レベルリング開始終了識別子3203には、優先度識別子3202の変更があることを示す「I」を格納する(S317)。ホストOS400がプロセッサ102で処理するVM200を選択する(S318)(図11参照)。

10

【0068】

通信フォーマット500の開始終了情報5005が終了の「E」でない場合(S303のN)、通信フォーマット500のタスク識別子5004と一致するVM管理テーブル302のタスク識別子3205を検索する(S321)。検索したタスク識別子3205がVM管理テーブル302に登録されていないVM200であれば(S321のY)、VM管理テーブル302に登録されたタスク識別子3205を登録する(S331)。検索したタスク識別子3205がVM管理テーブル302に登録されているVM200であれば(S321のN)、レベルリング開始終了識別子3203に開始の「S」を格納し(S322)、Timestamp3209にはホストOS400の時刻を格納する(S323)。

20

【0069】

VM200のVM内スケジュール212から通知がない場合(S301のN)、VM内スケジュール管理テーブル301のQuota3208の処理を行う(S341)(図14参照)。処理S318、331、323、341の後、次の通知を待つ。

30

【0070】

(実行VM選択のフローチャート)

図11は本発明の実施形態の実行VM選択(図10のS318)のフローチャートである。ホストOS400がプロセッサ102で処理するVM200を選択する。VM管理テーブル302から、優先度3202が「L1」のVM識別子3201を検索する(S401)。「L1」優先度を格納しているVM識別子3201がない場合(S401のY)、ラインタイム3207が最小であるVM識別子3201と一致するVM内スケジュール管理テーブル301内のVM識別子3101の周期3102を読み込む(S402)。ホストOS400の時刻とTimestamp3209との差が周期3102の長さ以上、即ち、周期デッドライン以上であれば(S403のY)、VM識別子3201に対応する優先度識別子3202に最高優先度「L1」を格納する(S404)。優先度識別子3202から最優先度を持つVM識別子3201を検索する(S405)。

40

【0071】

「L1」優先度を格納しているVM識別子3201がある場合(S401のN)、優先度識別子3202から最優先度を持つVM識別子3201を検索する(S405)。

【0072】

優先度識別子3202が同じVM識別子3201が複数ある場合(S406のY)、VM識別子3201の同列のラインタイム3207が最小のVM識別子3201を選択する(S411)。選択したVM識別子3201の同列の実行状態3206に「R」を格納し、

50

実行待ちであることを示す (S 4 2 1)。ホスト O S 4 0 0 がプロセッサ 1 0 2 で処理する V M 2 0 0 として、選択した V M 識別子 3 2 0 1 をホスト O S 4 0 0 に登録する (S 4 2 2)。

【 0 0 7 3 】

(タイマ - イベント管理プログラムのフローチャート)

図 1 2 は本発明の実施形態のタイマ - イベント管理プログラム (3 0 7) のフローチャートである。タイマ - イベントの通知を待つ (S 5 0 1)。V M 2 0 0 内の V M 内スケジュール 2 1 2 からの通知の場合 (S 5 0 2 の Y)、通知フォーマット 5 0 0 の優先度レベル 5 0 0 3 を読み込み (S 5 0 3)。優先度レベルが「 L 1 」であれば (S 5 0 4)、タイマ - 管理テーブル 3 0 3 に優先度レベルを設定し、ホスト O S 4 0 0 のタイマ - イベントとして登録する (S 5 0 5)。

10

【 0 0 7 4 】

ホスト O S 4 0 0 の仮想化プログラム 4 0 1 からの通知でなければ (S 5 0 2 の N)、タイマ - イベントを受信し (S 5 1 1)、外部イベント処理を行う (S 5 1 2) (図 1 3 参照)。処理 5 0 5、5 1 2 の後、次の通知を待つ。

【 0 0 7 5 】

(外部イベント処理のフローチャート)

図 1 3 は本発明の実施形態の外部イベント処理 (図 1 2 の S 5 1 2、図 1 5 の S 8 2 1) のフローチャートである。外部イベントは、ホスト O S 4 0 0 から V M スケジューラ制御プログラム 3 0 0 に通知される。外部イベントには通知先の V M 2 0 0 の情報が格納されている。

20

【 0 0 7 6 】

受信外部イベントから V M 管理テーブル 3 0 2 の V M 識別子 3 2 0 1 と一致する V M 2 0 0 を選択する (S 6 0 1)。V M 識別子 3 2 0 1 の列の優先度識別子 3 2 0 2 を読み込む (S 6 0 2)。優先度識別子 3 2 0 2 が「 L n 」であれば (S 6 0 3 の Y)、Q u o t a 3 2 0 8 にその値「 1 0 」を格納する (S 6 0 4)。

【 0 0 7 7 】

優先度識別子 3 2 0 2 が「 L n 」でない場合 (S 6 0 3 の N)、元優先度識別子 3 2 1 0 に優先度識別子 3 2 0 2 を格納し、優先度識別子 3 2 0 2 に最高優先度「 L 1 」を格納する (S 6 1 1)。実行状態 3 2 0 6 に実行待ちであることを示す「 R 」を格納し、ホスト O S 時刻を Timestamp 3 2 0 9 に格納する (S 6 1 2)。通知 I F 2 1 4 を介して前記外部イベントを V M スケジューラ制御プログラム 3 0 0 に通知する (S 6 1 3)。ホスト O S 4 0 0 が、プロセッサ 1 0 2 で処理する V M 2 0 0 として、ホスト O S 4 0 0 に V M 識別子 3 2 0 1 を登録する (S 6 1 4)。

30

【 0 0 7 8 】

(Q u o t a 管理処理のフローチャート)

図 1 4 は本発明の実施形態の Q u o t a 管理処理 (図 1 0 の S 3 4 1) のフローチャートである。V M 管理テーブル 3 0 2 の Q u o t a 3 2 0 8 が「 0 」以外の V M 識別子 3 2 0 1 を選択する。V M 識別子 3 2 0 1 に関する Q u o t a 3 2 0 8 を読み込む (S 7 0 1)。ホスト O S 4 0 0 の時刻と Timestamp 3 2 0 9 の差が Q u o t a 3 2 0 8 を超過した場合 (S 7 0 2 の Y)、前記 V M 識別子 3 2 0 1 に関する優先度識別子 3 2 0 2 に元優先度識別子 3 2 1 0、即ち「 0 」を格納する (S 7 1 1)。

40

【 0 0 7 9 】

優先度識別子 3 2 0 2 が「 L n 」であれば (S 7 1 2 の Y)、ランタイム 3 2 0 7 に Q u o t a 3 2 0 8 を加算する (S 7 1 3)。優先度識別子 3 2 0 2 が「 L n 」でない場合 (S 7 1 2 の N)、ランタイム 3 2 0 7 に Q u o t a 3 2 0 8 を加算し、Q u o t a 3 2 0 8 に「 0 」を格納する (S 7 1 4)。

【 0 0 8 0 】

ホスト O S 4 0 0 の時刻と Timestamp 3 2 0 9 の差が Q u o t a 3 2 0 8 の未満の場合 (S 7 0 2 の N)、ランタイム 3 2 0 7 にホスト O S 4 0 0 の時刻と Timestamp 3 2 0 9

50

の差を加算する (S 7 2 1)。ホスト O S 4 0 0 の時刻を Timestamp 3 2 0 9 に格納する (S 7 2 2)。処理 S 7 1 3、7 1 4、7 2 2 の後、処理 S 3 4 1 は終了する。

【 0 0 8 1 】

(割り込み管理プログラムのフローチャート)

図 1 5 は本発明の実施形態の割り込み管理プログラム (3 0 8) のフローチャートである。 V M 内スケジュール管理テーブル 3 0 1 を読み込み、割り込み管理テーブル 3 0 4 の項目 3 4 0 1 から 3 4 0 4 に情報を設定し、デッドライン許容値 3 4 0 5 には default 値の「 1 0 」を格納し、 1 つ割り込み処理時間 3 4 0 6 には「 0 . 1 」を格納する (S 8 0 1)。割り込み数臨界値 3 4 0 7 には、デッドライン許容値 3 4 0 5 を 1 つ割り込み処理時間 3 4 0 6 で割った数に「 1 0 」を引いた数を格納する (S 8 0 2)。 1 つ割り込み処理時間 3 4 0 6 割った数に「 1 0 」を引くことで、 1 つ割り込み処理時間 3 4 0 6 の処理時間の誤差によって発生する臨界値誤差に対する余裕値が設定される。

10

【 0 0 8 2 】

ホスト O S 4 0 0 から割り込みがあるか判断する (S 8 0 3)。割り込み情報には通常割り込み通知先の V M 2 0 0 の情報が格納されている。割り込み情報に含まれている V M 2 0 0 の情報と V M 識別子 3 4 0 1 が一致する、割り込み管理テーブル 3 0 4 の行を選択する (S 8 1 1)。割り込み累積 3 4 0 9 に「 1 」を加算する (S 8 1 2)。割り込み累積 3 4 0 9 が割り込み数臨界値 3 4 0 7 未満であるかどうかを判断する (S 8 1 3)。

【 0 0 8 3 】

割り込み累積 3 4 0 9 が割り込み数臨界値 3 4 0 7 内であれば (S 8 1 3 の Y)、割り込みがタイマ - イベントであるかどうかを判断する (S 8 1 4)。タイマ - イベントであれば (S 8 1 4 の Y)、割り込み累積 3 4 0 9 を「 1 」減算し (S 8 1 5)、タイマ - イベント管理プログラム 3 0 7 に通知する (S 8 1 6)。割り込みがタイマ - イベントでない場合 (S 8 1 4 の N)、外部イベント処理を呼び出す (S 8 2 1) (図 1 3 参照)。

20

【 0 0 8 4 】

割り込み累積 3 4 0 9 が割り込み数臨界値 3 4 0 7 を超過した場合 (S 8 1 3 の N)、ホスト O S 4 0 0 の仮想化プログラム 4 0 1 の割り込みペンディング処理 (図示せず) を呼び出し、割り込み通知を遅延する (S 8 3 1)。

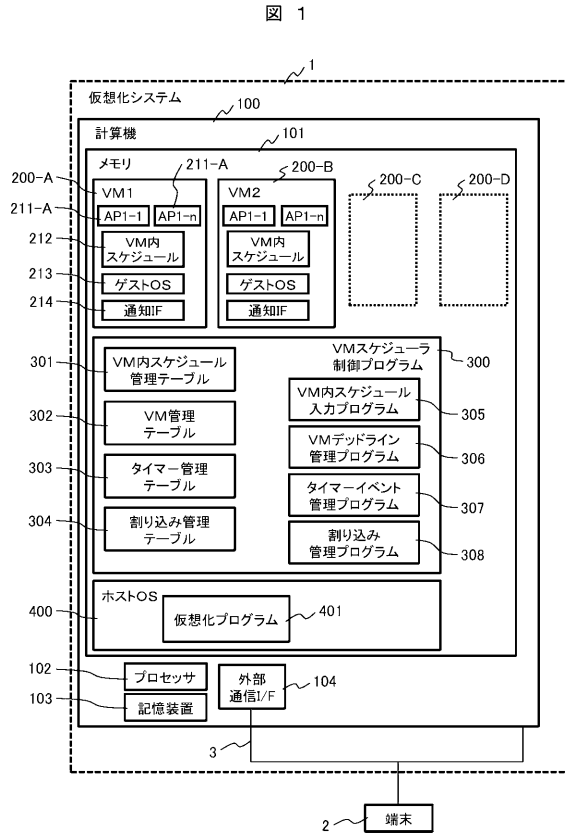
【 符号の説明 】

【 0 0 8 5 】

1 ... 仮想化システム、 2 ... 端末、 3 ... 外部ネットワーク、 1 0 0 ... 計算機ノード、 1 0 1 ... メモリ、 1 0 2 ... プロセッサ、 1 0 3 ... 記憶装置、 1 0 4 ... 外部通信 I / F、 2 0 0 ... V M、 2 1 1 ... A P、 2 1 2 ... V M 内スケジュールプログラム、 2 1 3 ... ゲスト O S、 2 1 4 ... 通知 I / F、 3 0 0 ... V M スケジューラ制御プログラム、 3 0 1 ... V M 内スケジュール管理テーブル、 3 0 2 ... V M 管理テーブル、 3 0 3 ... タイマ - 管理テーブル、 3 0 4 ... 割り込み管理テーブル、 3 0 6 ... V M デッドライン管理プログラム、 3 0 7 ... タイマ - イベント管理プログラム、 3 0 8 ... 割り込み管理プログラム、 4 0 0 ... ホスト O S、 4 0 1 ... 仮想化プログラム

30

【図1】



【図2】

図2

301

VM 識別子	周期 [ms]	優先度 レベル 識別子	レベル リング 開始 終了 識別子	タスク 識別子	優先度	ランタイム [ms]	順序制御 タスク 識別子	Quota [ms]
VM1	1000	L1	S	P1	50	100	P2	0
			N	P2	50	100	P3	0
			...	N				0
			N	P32	50	100	E	0
VM2	1000	L1	N	P33	50	100	E	0
			E	P34	50	100	E	0
			S	P101	40	100	P102	0
			E	P102	40	100	E	0
VM2	1000	Ln	N	N	N	N	N	10
			L1	S	P1	50	100	P2

【図3】

図3

302

VM 識別子	優先度 識別子	レベル リング 開始 終了 識別子	デッド ライン [ms] E-S	タスク 識別子	実行 状態	ランタイム [ms]	Quota [ms]	Time Stamp [ms]	元 優先度
VM1	L1	S	256	P1	R	100	0	100	0
VM2	L1	E	256	P32	0	120	0	120	0
VM3	L2	I	0	0	0	130	0	130	0
VM4	Ln	N	0	0	0	10	10	10	0

【図5】

図5

304

VM 識別子	周期 [ms]	優先度 レベル	デッド ライン [ms]	デッド ライン 許容値 [ms]	1つ割り込み処理時間 [ms]	割り込み数 臨界値	周期 デッド ライン [ms]	割り込み 累積
VM1	1000	L1	256	10	0.1	90	1000	80
VM2	1000	L1	256	10	0.1	90	1000	70
VM3	1000	L1	256	10	0.1	90	1000	60
VM4	1000	L1	256	10	0.1	90	1000	50

【図4】

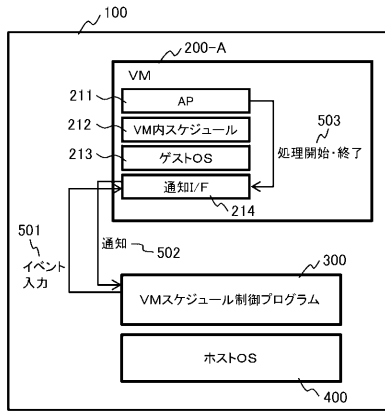
図4

303

VM 識別子	優先度 レベル	タスク 識別子	インタ パーナル [ms]	カウンタ		
VM1	L1	P1	10	100		
VM1	L1	P3	20	10		
VM3	L2	P10	100	0		

【図6A】

図 6A



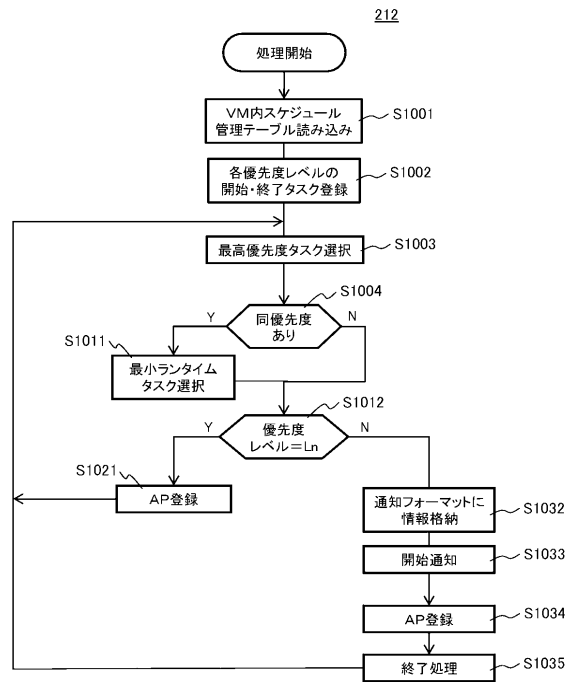
【図6B】

図 6B

5001	5002	5003	5004	5005	5006	5007	500
VM 識別子	周期 [ms]	優先度 レベル	タスク 識別子	開始 終了	1つ 割り込み 処理 時間 [ms]	割り込み数 臨界値	
VM1	1000	L1	P1	S	0.1	90	

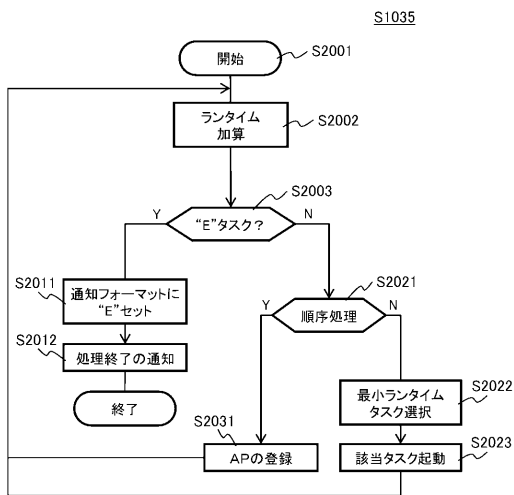
【図7】

図 7



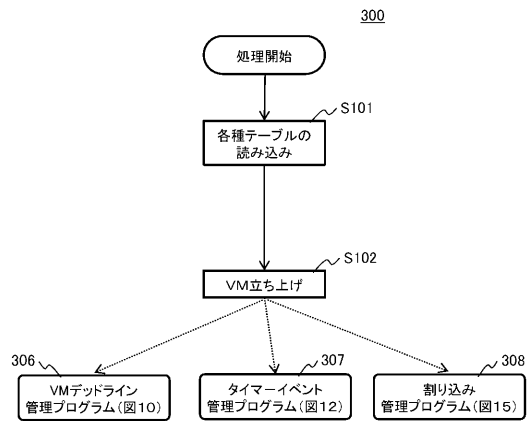
【図8】

図 8

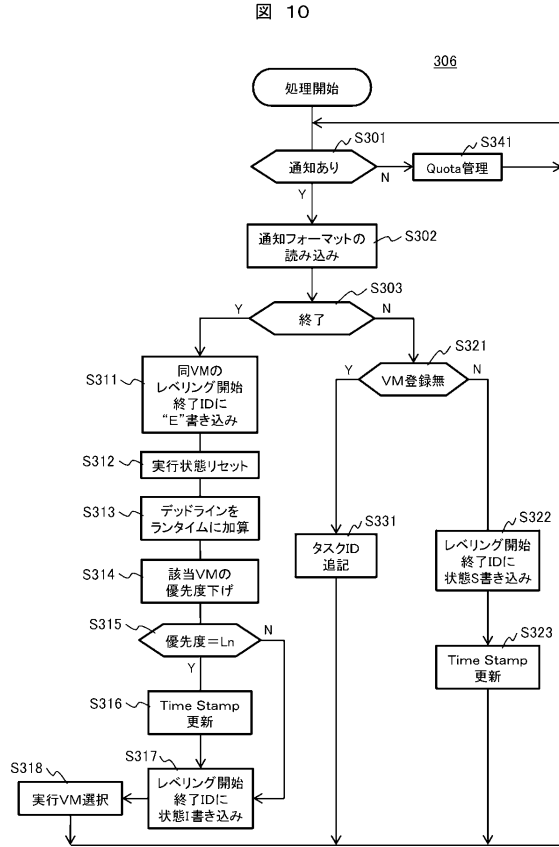


【図9】

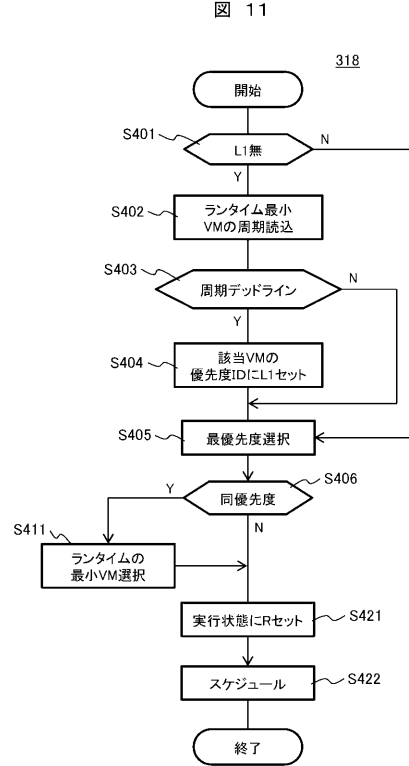
図 9



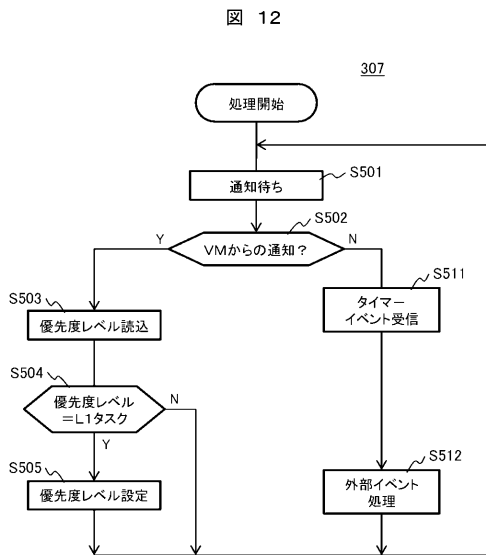
【図10】



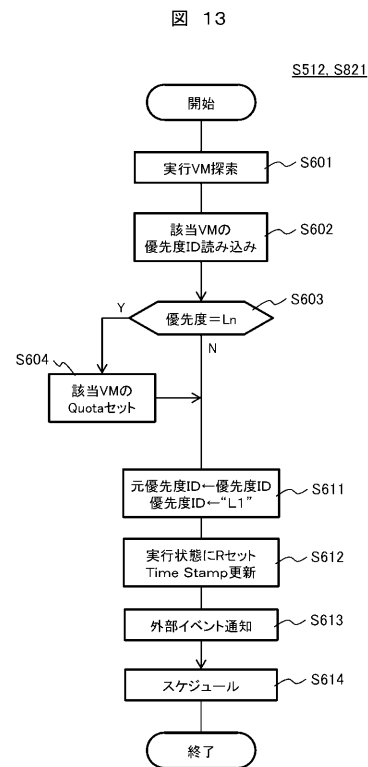
【図11】



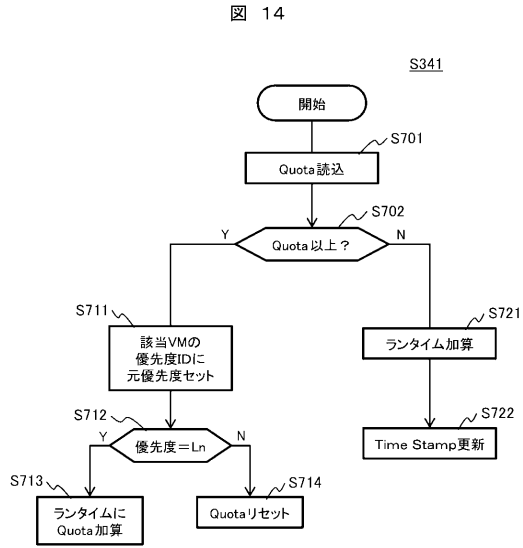
【図12】



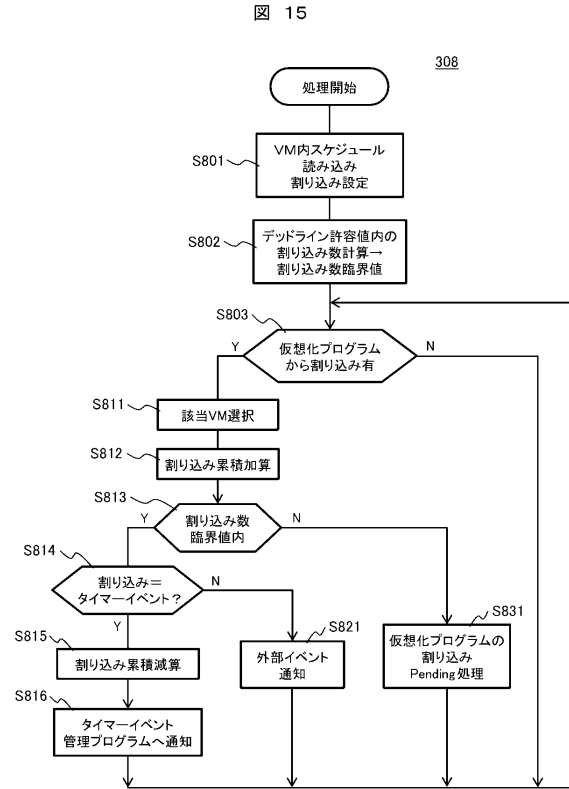
【図13】



【 図 1 4 】



【 図 1 5 】



フロントページの続き

(56)参考文献 特開2005 - 43959 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/48

G06F 9/46