

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2013年10月3日(03.10.2013)



(10) 国際公開番号

WO 2013/145199 A1

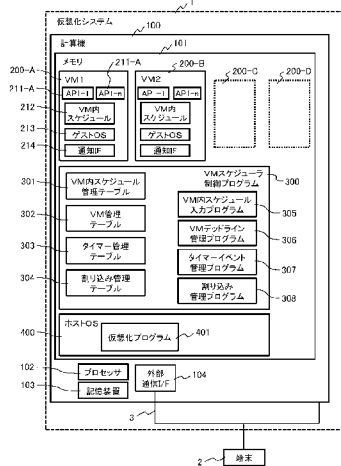
- (51) 国際特許分類:
G06F 9/46 (2006.01) **G06F 9/48** (2006.01)
- (21) 国際出願番号: PCT/JP2012/058295
- (22) 国際出願日: 2012年3月29日(29.03.2012)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (71) 出願人(米国を除く全ての指定国について): 株式会社日立製作所(HITACHI, LTD.) [JP/JP]; 〒1008280 東京都千代田区丸の内一丁目6番6号 Tokyo (JP).
- (72) 発明者; および
- (75) 発明者/出願人(米国についてのみ): 金 成昊(KIM Sungho) [KR/JP]; 〒2440817 神奈川県横浜市戸塚区吉田町292番地 株式会社日立製作所横浜研究所内 Kanagawa (JP). 大平 崇博(OHIRA Takahiro) [JP/JP]; 〒3191293 茨城県日立市大みか町五丁目2番1号 株式会社日立製作所 情報制御システム社内 Ibaraki (JP).
- (74) 代理人: ポーレル特許業務法人(POLAIRE I.P.C.); 〒1040032 東京都中央区八丁堀二丁目7番1号 Tokyo (JP).
- (81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), ユーロピア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR),

[続葉有]

(54) Title: VIRTUAL COMPUTER SCHEDULE METHOD

(54) 発明の名称: 仮想計算機のスケジュール方法

図 1



- 1 Virtualized system
- 2 Terminal
- 100 Computer
- 101 Memory
- 102 Processor
- 103 Storage device
- 104 External communication interface
- 211 VM-based scheduler
- 212 VM management table
- 213 Timer management table
- 214 Interrupt management table
- 300 VM scheduler control program
- 301 VM-based schedule management table
- 302 VM management table
- 303 Timer management table
- 304 Interrupt management table
- 305 VM-based schedule input program
- 306 VM deadline management program
- 307 Timer event management program
- 308 Interrupt management program
- 400 Host operating system
- 401 Virtualized program

(57) Abstract: In a virtualized system that runs multiple virtual computers, due to a restriction on a CPU core resource and due to a process delay in I/O control specific to the virtualized system, it has been difficult to operate a virtual computer demanding real-time performance. In the present invention, in a virtualized system that runs multiple virtual computers: applications processed on the virtual computers are grouped by priority level; the start of a first task of each priority level group is managed as a process deadline for the priority level group, and the end of a last task of each priority level group is managed as a process deadline for the priority level group; and a host operating system is notified of a process start of each priority level group, and is notified of a process end of each priority level group. The host operating system: receives the start notification for each priority level group, and receives the end notification for each priority level group; adjusts the priority level of a virtual computer; and in response to an interrupt process from an external device, restricts the number of interrupt notifications so as to meet the deadlines of the highest-priority-level group among the virtual computers.

(57) 要約: 複数の仮想計算機を稼働する仮想化システムにおいて、CPUコアのリソースの制限と仮想化システム特有のI/O制御における処理遅延によりリアルタイム性を要求する仮想計算機の運用は困難であった。複数の仮想計算機を稼働する仮想化システムにおいて、仮想計算機上で処理されるアプリケーションを優先度別にグループ化し、各優先度グループの先頭タスクの開始と後尾タスクの終了を該当優先度グループの処理デッドラインとして管理し、各優先度グループの処理開始と終了をホストOSに通知し、ホストOSでは、優先度グループの開始と終了通知を受け、仮想計算機の優先度を調整し、外部装置からの割り込み処理に対して、仮想計算機内の最高優先度グループのデッドラインを守るように割り込み通知の数を制限する。

WO 2013/145199 A1



OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告 (条約第 21 条(3))

明 細 書

発明の名称： 仮想計算機のスケジュール方法

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、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。

[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。

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内スケジューラ制御部を設ける。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ラ機能を備え、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ラ機能、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。

[0007] まず、ゲストOS上で稼働するVM内スケジューラ制御部は、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。

[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、「L 1」が最高優先度グループとし、「L n」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「L 1」から「L
n」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
ジュール状態を通知するために、各ゲストOSのVM内スケジュール制御部
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ル状態を
把握するために、ゲストOS上のVM内スケジュール制御部は、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「L n」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。「L n」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。

[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内スケジュール制御部からホスト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ル制御ではなく、後述するCPUリソースを割り当てるQuota時間でスケジュール制御を行う。

[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。

[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内部スケジュール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。

[0016] 以上のように、仮想計算機内のVM内スケジュール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。

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ル制御の全体構成図である。
[図2]本発明のVM内スケジュール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]本発明のタイマー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内スケジュールプログラムのフローチャートである。
[図8]本発明のVM内スケジュール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]本発明のタイマー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Q u o t a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

[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ル制御の全体構成図である。仮想マシンのスケジュール制御は、一つの仮想化システム1及び一端末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。

[0021] VMスケジューラ制御プログラム300は、VM内スケジュール管理テーブル301、VM管理テーブル302、タイマー管理テーブル303、割り込み管理テーブル304、VM内スケジュール入力プログラム305、VMデッドラン管理プログラム306、タイマー管理プログラム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内スケジュール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VM２００は「VM１」、あるいは「VM２」の識別子を保有する。

[0027] AP２１１は、VM２００内で動作アプリケーションプログラムである。AP２１１として、AP１－１（２１１－A）、AP１－n（２１１－B）が示されているが、VM２００内で動作するAP２１１の数を限定するものではない。ゲストOS２１３は、VM２００環境内でVM内スケジュール２１２を利用してAP２１１を制御する。通知IF２１４は、VM内スケジュール２１２の処理結果をゲストOS２１３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、VM２００で稼働するとき、VM内スケジュール２１２から、VM２００内のAP２１１のスケジュール情報を、通知IF２１４を介して受け取る。VMスケジューラ制御プログラム３００は、AP２１１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、AP２１１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] VM２００内で動作するVM内スケジュール２１２は、AP２１１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] 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

プのAP211のタイマー登録をタイマーイベント管理プログラム307に通知IF214を介して通知する。タイマー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ー管理テーブル303に登録し、仮想化プログラム401にタイマー制御を委ねる。仮想化プログラム401は、タイマーが発火（起動）するとタイマーイベント管理プログラム307に通知する。タイマーイベント管理プログラム307は、タイマー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にタイマー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からタイマーイベントを処理し、処理結果を、タイマー登録を行った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内スケジューラ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L 2、次にL 3、次にL nの順に優先度が高い。

[0037] また、グループに属するA P 2 1 1は、同グループ内で最初に実行するA P 2 1 1から最後に実行するA P 2 1 1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A P 2 1 1をリング状に構成するため、V M 2 0 0には、各優先度グループの最初と最後のA P 2 1 1を事前に登録しておく必要がある。最初と最後のA P 2 1 1は、ユーザが登録するアプリケーションとは異なり、V M内スケジュール2 1 3が登録されたA P 2 1 1である。

[0038] レベルリング開始終了識別子3 1 0 4には、グループに属する最初のA P 2 1 1が開始することを知らせる「S」と最後に実行するA P 2 1 1の終了を示す「E」が格納される。A P 2 1 1がグループの最初と最後のA P 2 1 1ではない場合はレベルリング開始終了識別子3 1 0 4に「N」が格納される。

[0039] タスク識別子3 1 0 5には、グループ内のA P 2 1 1を一意に識別する情報を格納する。優先度3 1 0 6には、ゲストO S 2 1 3がA P 2 1 1を実行する順位を決めるA P 2 1 1の実行優先度情報を格納する。ランタイム3 1 0 7には、A P 2 1 1がゲストO S 2 1 3で実行された時間を「m s」単位で格納する。順序制御タスク識別子3 1 0 8には、A P 2 1 1の実行が終了した場合、次に実行するA P 2 1 1のタスク識別子3 1 0 5を格納する。順序制御タスク識別子3 1 0 8にタスク識別子3 1 0 5「E」以外の情報が格納されている場合は、該当するA P 2 1 1を実行する。タスク識別子3 1 0 5に「E」の情報が格納され、レベルリング開始終了識別子3 1 0 4に「N」が格納されている場合は同グループの他A P 2 1 1の実行を行う。

[0040] Q u o t a 3 1 0 9には、優先度レベル識別子3 0 1 3が「L n」に属しているA P 2 1 1の実行に割り当てる時間を格納する。

[0041] V M内スケジュール2 1 2の詳細は図7と共に後述する。

[0042] （V M管理テーブル）

図3は本発明の実施形態のV M管理テーブル3 0 2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。

[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。

[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によって実行していない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、後述するタイマー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。

[0047] タイマー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Q u o t a 3 2 0 8の情報は、タイマーイベントと割り込みがV M 2 0 0のゲストO S 2 1 3によって処理された後、ゲストO S 2 1 3によって実行されるA P 2 1 1の優先度グループが優先度識別子3 2 0 2と一致しないことがある。特に、「L n」優先度グループのA P 2 1 1が実行される場合、開始終了がV Mデッドライン管理プログラム3 0 6に通知されないため、V M管理テーブル3 0 4の優先度識別子3 2 0 2が「L n」より高い優先度グループのV M 2 0 0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、タイマーイベントと割り込み通知にはQ u o t a 3 2 0 8を格納する必要がある。

[0048] T i m e S t a m p 3 2 0 9にはランタイム3 2 0 7を格納する。後述するV Mデッドライン管理プログラム3 0 7がQ u o t a 3 2 0 8の時間が経過したかを判断するためである。元優先度3 2 1 0には、タイマーイベントと割り込み処理では優先度識別子3 2 0 2に優先度「L 1」を格納する際に優先度識別子3 2 0 2の情報を格納する。タイマー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V M 2 0 0が優先的に処理するようにする必要があるため、V M 2 0 0の実行優先度に関係なく最高優先度「L 1」にする必要がある。Q u o t a 3 2 0 8の経過後は、元優先度3 2 1 0の優先度を優先度識別子3 2 0 2に格納する。

[0049] (タイマー管理テーブル)

図4は本発明の実施形態のタイマー管理テーブル3 0 3の構成を示す図である。タイマー管理テーブル3 0 3は、V M 2 0 0のゲストO S 2 1 3が実行するA P 2 1 1の中で実行優先度グループ「L 1」に属しているA P 2 1 1のタイマー登録を仮想化プログラム4 0 1で行うために必要なテーブルである。「L 1」のグループのA P 2 1 1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、V M 2 0 0のゲストO S 2 1 3ではなく、ホストO S 4 0 0の仮想化プログラム4 0 1で処理を行う。「L 1」のグループに属しているA P 2 1 1のタイマー登録を、V M内スケジュール2 1 2が通知I F 2 1 4を介してタイマーイベント管理プログラム3 0 7に通知する。

[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にはタイマーイベントの発火時間を格納する。カウンタ3305にはタイマー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。

[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にはタイマー管理テーブル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。

[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する。

[0054] （通知IF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)、VM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OS 4 0 0の仮想化プログラム4 0 1がVM 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ーイベントは仮想化プログラム4 0 1が処理し、VMスケジューラ制御プログラム3 0 0に通知する。VMスケジューラ制御プログラム3 0 0はVM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VM 2 0 0に通知する(5 0 1)。

[0055] 図6 Bの通知フォーマット5 0 0における、VM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VM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。VM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。

[0056] (VM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VM内スケジューラ2 1 2のフローチャートである。ホストOS 4 0 0及びプロセッサ1 0 2は、VMスケジューラ制御プログラム3 0 0及びVM 2 0 0を計算機1 0 0のメモリ1 0 1内に展開すると、VM 2 0 0の処理を開始する。VM 2 0 0が処理を始めるとゲストOS 2 1 3の処理を始め、VM内スケジューラ2 1 2の処理を始める。

[0057] VM内スケジューラ2 1 2は、記憶装置1 0 3からVM内スケジューラ管

理テーブル301をメモリ101内に読み込む(S1001)。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3103の情報を格納している「Ln」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AP211と終了用AP211を登録する(S1002)。

[0058] VM内スケジュール管理テーブル301の優先度3106の情報から最高優先度のAP211を、タスク識別子3105から選択する(S1003)。優先度3106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3105が複数ある場合(S1004のY)、VM内スケジュール管理テーブル301のランタイム3107の情報からランタイムが最小のタスク識別子3105を選択する(S1011)。

[0059] 前記タスク識別子3105の属する優先度レベル識別子3103が「Ln」かどうかを判断する(S1012)。優先度レベルが「Ln」の場合、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が起動するアプリケーションとして登録する(S1021)。

[0060] 優先度レベルが「Ln」ではない場合、通知フォーマット500に、VM識別子5001と周期5002と優先度レベル5003とタスク識別子5004をVM内スケジュール管理テーブル301から取得して該当情報を格納する。開始終了5005には「S」を格納する(S1032)。通知IF214を介して通知フォーマット500をVMスケジュ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。

[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）。

[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。

[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、タイマー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、タイマー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)、タイマー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。

[0066] (VMデッドライン管理プログラムのフローチャート)

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(S

313)。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)。

[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に検索したタスク識別子3205に登録する(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)。

[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。

[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)。

[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、実行待ちであることを示す(S421)。ホストOS400がプロセッサ102で処理するVM200として、選択したVM識別子3201をホストOS400に登録する(S422)。

[0073] (タイマーイベント管理プログラムのフローチャート)

図12は本発明の実施形態のタイマーイベント管理プログラム(307)のフローチャートである。タイマーイベントの通知を待つ(S501)。VM200内のVM内スケジュール212からの通知の場合(S502のY)、通知フォーマット500の優先度レベル5003を読み込み(S503)。優先度レベルが「L1」であれば(S504)、タイマー管理テーブル303に優先度レベルを設定し、ホストOS400のタイマーイベントとして登録する(S505)。

[0074] ホストOS400の仮想化プログラム401からの通知でなければ(S502のN)、タイマーイベントを受信し(S511)、外部イベント処理を行う(S512)(図13参照)。処理505、512の後、次の通知を待つ。

[0075] (外部イベント処理のフローチャート)

図13は本発明の実施形態の外部イベント処理（図12のS512、図15のS821）のフローチャートである。外部イベントは、ホストOS400からVMスケジューラ制御プログラム300に通知される。外部イベントには通知先のVM200の情報が格納されている。

[0076] 受信外部イベントからVM管理テーブル302のVM識別子3201と一致するVM200を選択する（S601）。VM識別子3201の列の優先度識別子3202を読み込む（S602）。優先度識別子3202が「L_n」であれば（S603のY）、Q_{uota}3208にその値「10」を格納する（S604）。

[0077] 優先度識別子3202が「L_n」でない場合（S603のN）、元優先度識別子3210に優先度識別子3202を格納し、優先度識別子3202に最高優先度「L₁」を格納する（S611）。実行状態3206に実行待ちであることを示す「R」を格納し、ホストOS時刻をTimestamp3209に格納する（S612）。通知IF214を介して前記外部イベントをVMスケジューラ制御プログラム300に通知する（S613）。ホストOS400が、プロセッサ102で処理するVM200として、ホストOS400にVM識別子3201を登録する（S614）。

[0078] （Q_{uota}管理処理のフローチャート）

図14は本発明の実施形態のQ_{uota}管理処理（図10のS341）のフローチャートである。VM管理テーブル302のQ_{uota}3208が「0」以外のVM識別子3201を選択する。VM識別子3201に関するQ_{uota}3208を読み込む（S701）。ホストOS400の時刻とTimestamp3209の差がQ_{uota}3208を超過した場合（S702のY）、前記VM識別子3201に関する優先度識別子3202に元優先度識別子3210、即ち「0」を格納する（S711）。

[0079] 優先度識別子3202が「L_n」であれば（S712のY）、ランタイム3207にQ_{uota}3208を加算する（S713）。優先度識別子3202が「L_n」でない場合（S712のN）、ランタイム3207にQ_{uota}

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）。

[0080] ホストOS 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OS 4 0 0の時刻とTimestamp 3 2 0 9の差を加算する（S 7 2 1）。ホストOS 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。

[0081] （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。VM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。

[0082] ホストOS 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VM 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VM 2 0 0の情報とVM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）。

[0083] 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）、割り込みがタイマーイベントであるかどうかを判断する（S 8 1 4）。タイマーイベントであれば（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）、タイマーイベント管理プログラム3 0 7に通知

する（S 8 1 6）。割り込みがタイマー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13参照）。

[0084] 割り込み累積3409が割り込み数臨界値3407を超過した場合（S 8 1 3のN）、ホストOS400の仮想化プログラム401の割り込みペディング処理（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

[0085] 1…仮想化システム、2…端末、3…外部ネットワーク、100…計算機ノード、101…メモリ、102…プロセッサ、103…記憶装置、104…外部通信I/F、200…VM、211…AP、212…VM内スケジューラプログラム、213…ゲストOS、214…通知I/F、300…VMスケジューラ制御プログラム、301…VM内スケジューラ管理テーブル、302…VM管理テーブル、303…タイマー管理テーブル、304…割り込み管理テーブル、306…VMデッドライン管理プログラム、307…タイマーイベント管理プログラム、308…割り込み管理プログラム、400…ホストOS、401…仮想化プログラム

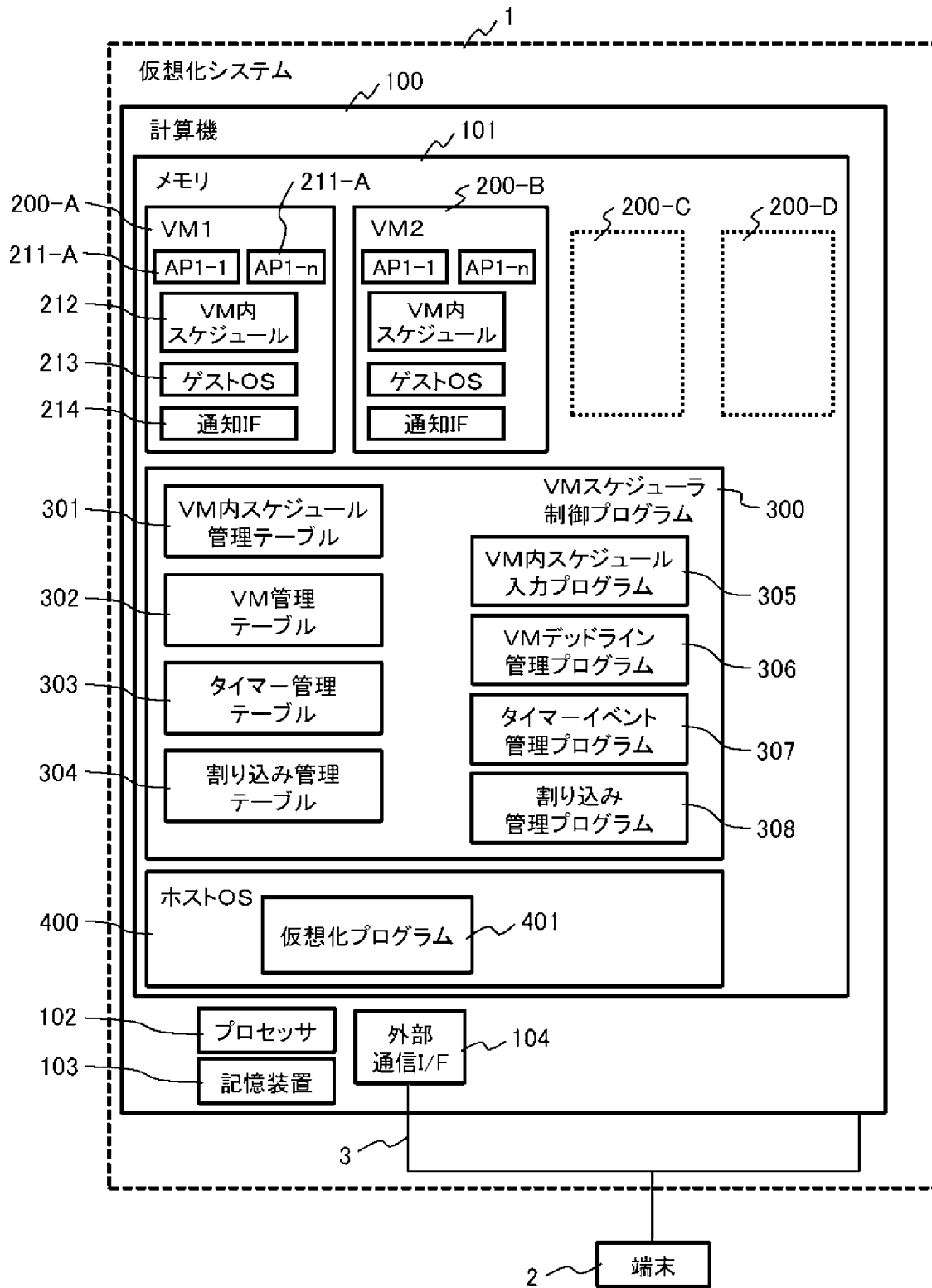
請求の範囲

- [請求項1] ホストOSおよび仮想計算機を搭載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VM内スケジュール処理部は、
- 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、
- リアルタイム処理が必要な高優先度グループの動作開始と終了を通知する前記優先度グループの先頭タスクと後尾タスクを導入することで前記優先度グループをリング状で管理し、
- 前記ホストOS上のVMスケジューラ制御部は、
- 前記優先度グループの先頭タスクと後尾タスクからの前記優先度グループの処理が開始と終了の通知をうけ、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VMスケジュール処理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、仮想計算機の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1に記載の仮想計算機のスケジュール方法。
- [請求項3] ホストOSおよび仮想計算機を搭載する計算機システムにおいて、
- 前記仮想計算機上に設けたVM内スケジュール処理部は、
- 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、リアルタイム処理が必要な高優先度グループの動作開始と終了を通知する前記優先度グループの先頭タスクと後尾タスクを導入することで前記優先度グループをリング状で管理する、
- VM内スケジュール処理部を前記仮想計算機上に設け、
- 前記優先度グループの先頭タスクと後尾タスクからの前記優先度グループの処理が開始と終了の通知をうけ、仮想計算機の処理優先度を決定するVMスケジューラ制御部を前記ホストOS上に設けたことを特徴とする仮想計算機システム。

[請求項4] 前記VMスケジュール処理部に、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、仮想計算機の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3に記載の仮想計算機システム。

[図1]

図 1



[図2]

図 2

301

3101 VM 識別子	3102 周期 [ms]	3103 優先度 レベル 識別子	3104 レベル リング 開始 終了 識別子	3105 タスク 識別子	3106 優先度	3107 ランタ イム [ms]	3108 順序制御 タスク 識別子	3109 Quota [ms]
VM1	1000	L1	S	P1	50	100	P2	0
		L1	N	P2	50	100	P3	0
		...	N					0
		L1	N	P32	50	100	E	0
		L1	N	P33	50	100	E	0
		L1	E	P34	50	100	E	0
		L2	S	P101	40	100	P102	0
		L2	E	P102	40	100	E	0
		Ln	N	N	N	N	N	10
VM2	1000	L1	S	P1	50	100	P2	0

[図3]

図 3

302

3201 VM 識別子	3202 優先度 識別子	3203 レベル リング 開始 終了 識別子	3204 デッド ライン [ms] E-S	3205 タスク 識別子	3206 実行 状態	3207 ランタ イム [ms]	3208 Quota [ms]	3209 Time Stamp [ms]	3210 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4]

図 4

303

3301	3302	3303	3304	3305		3306	
VM 識別子	優先度 レベル	タスク 識別子	インタ バール [ms]	カウンタ			
VM1	L1	P1	10	100			
VM1	L1	P3	20	10			
VM3	L2	P10	100	0			

[図5]

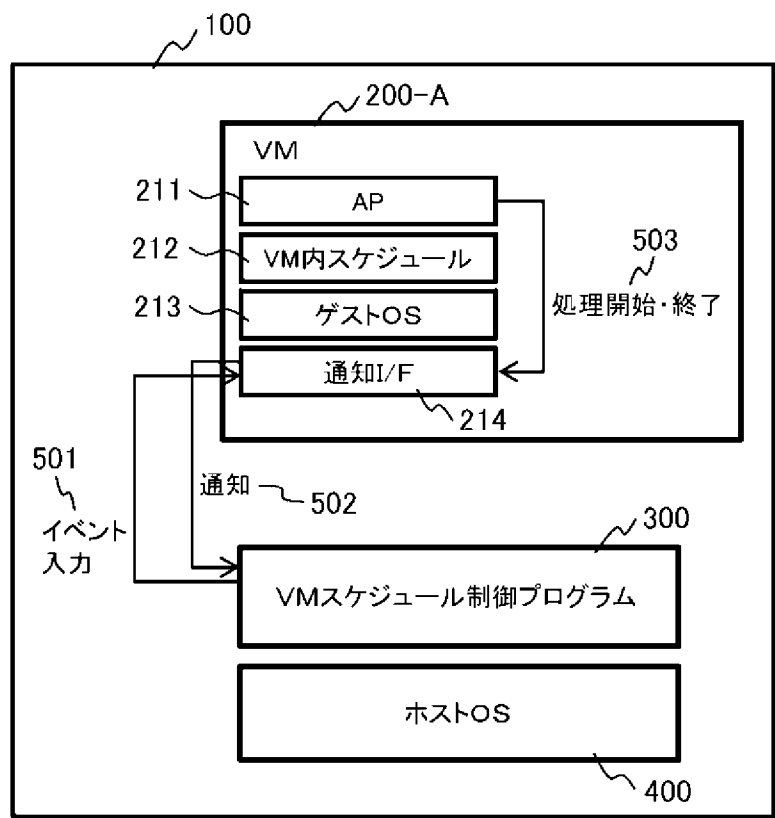
図 5

304

3401	3402	3403	3404	3405	3406	3407	3408	3409
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6A]

図 6A



[図6B]

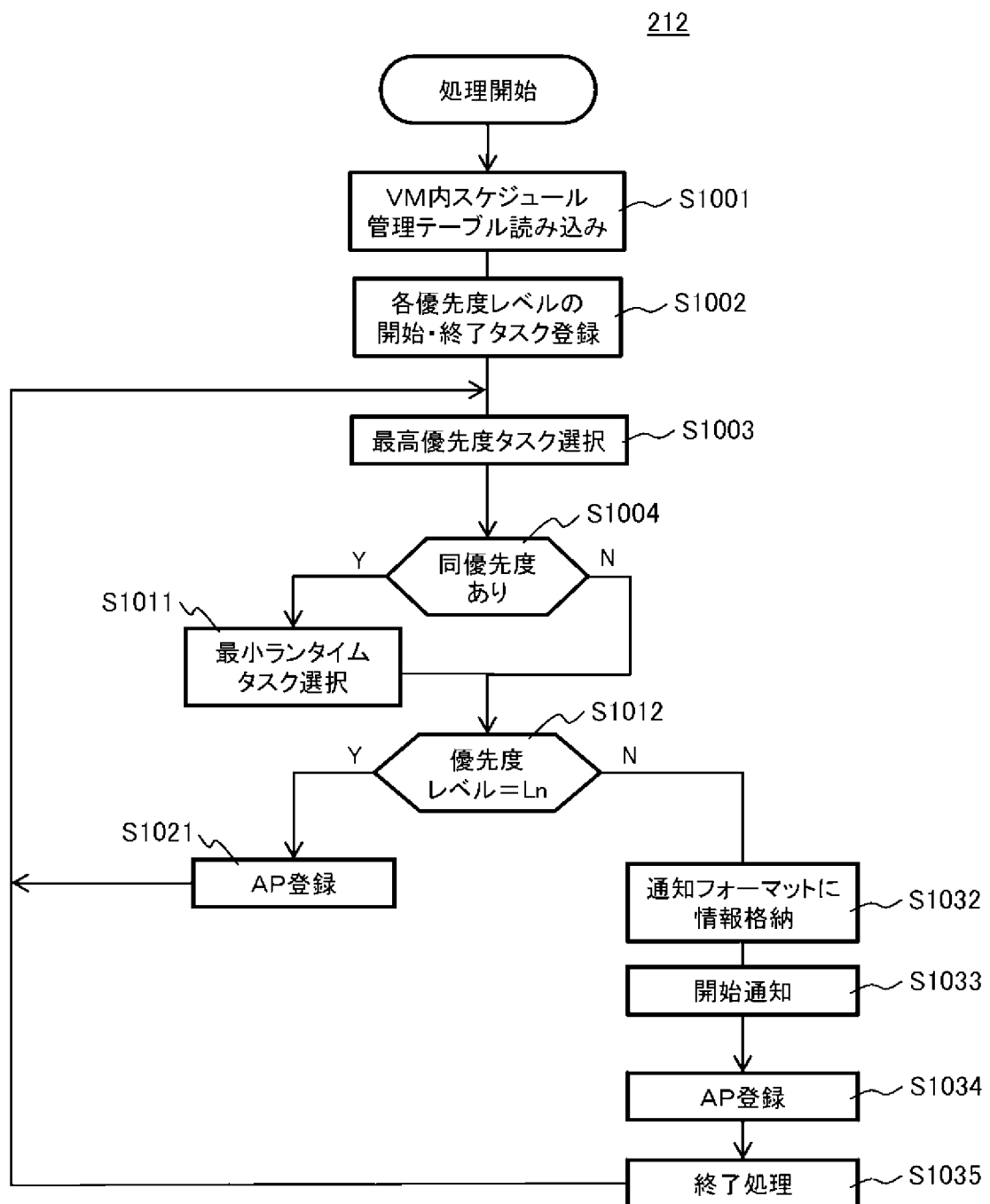
図 6B

500

5001	5002	5003	5004	5005	5006	5007	
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	

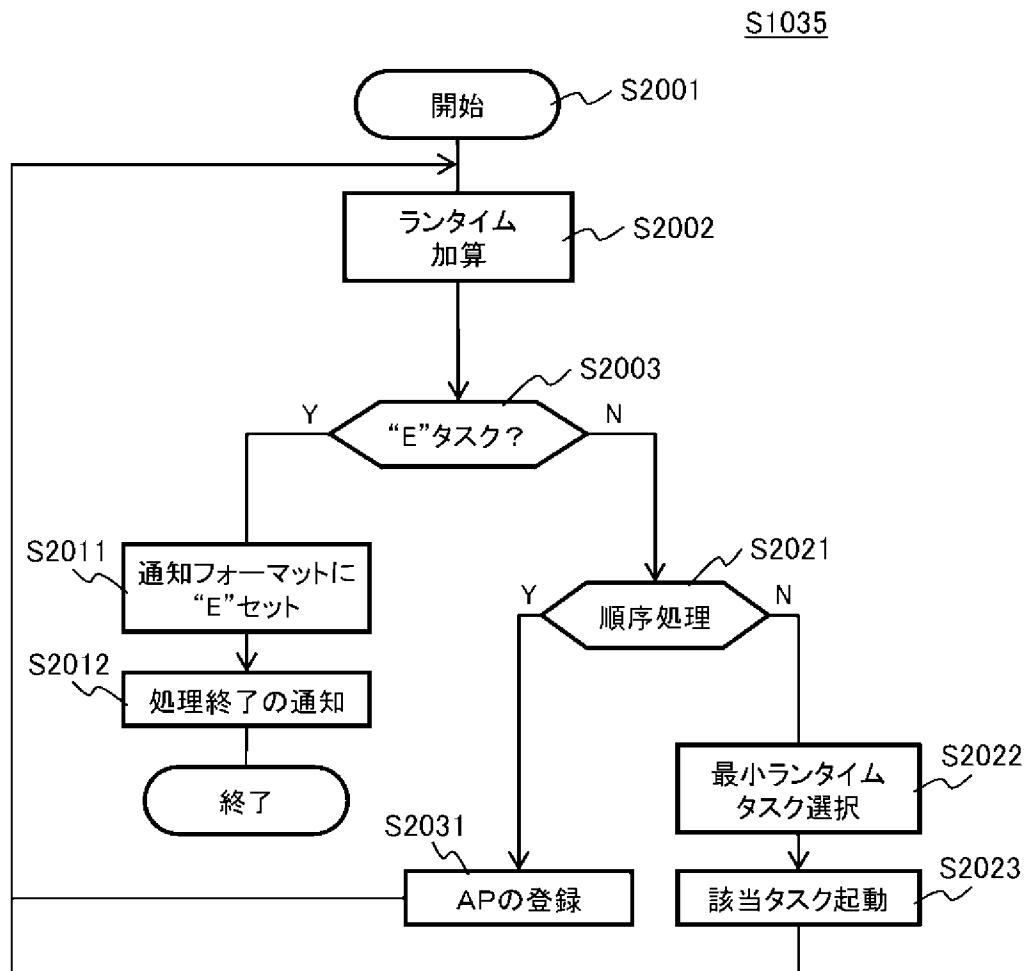
[図7]

図 7



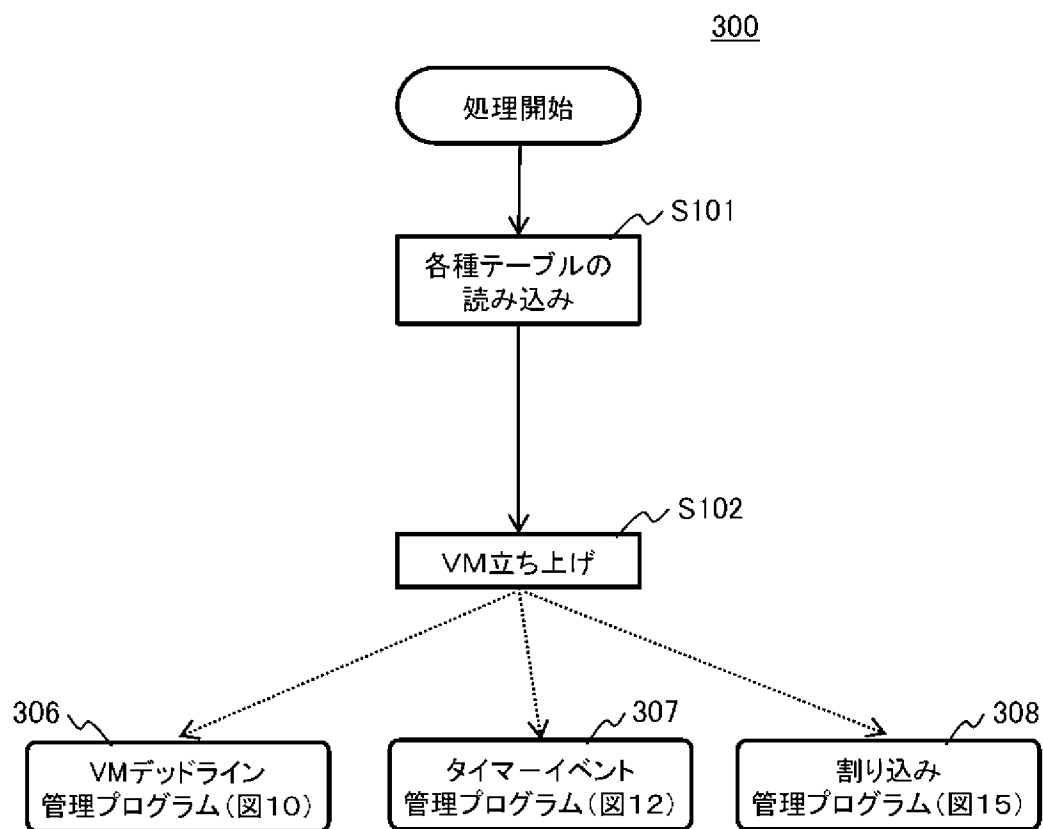
[図8]

図 8



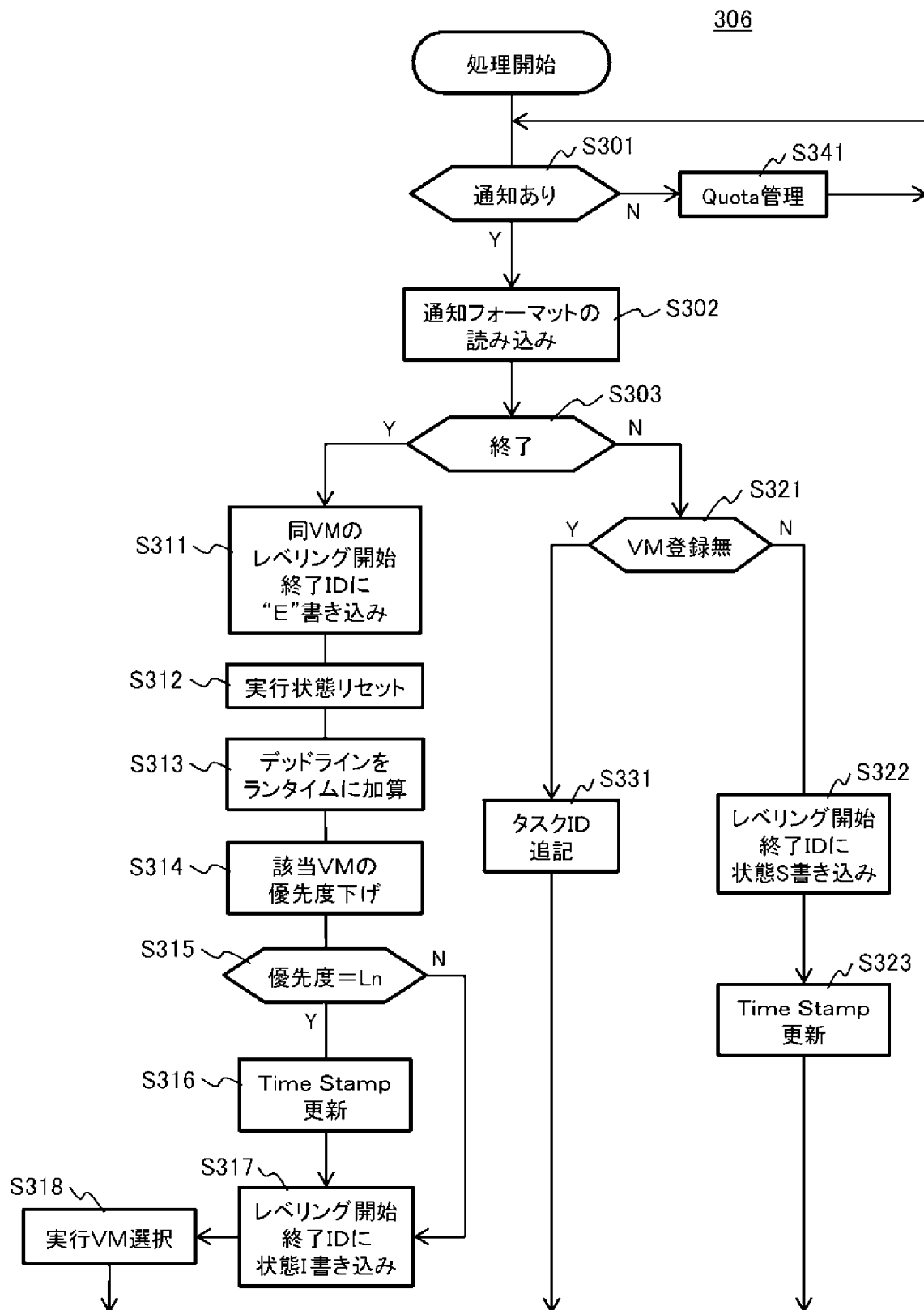
[図9]

図 9



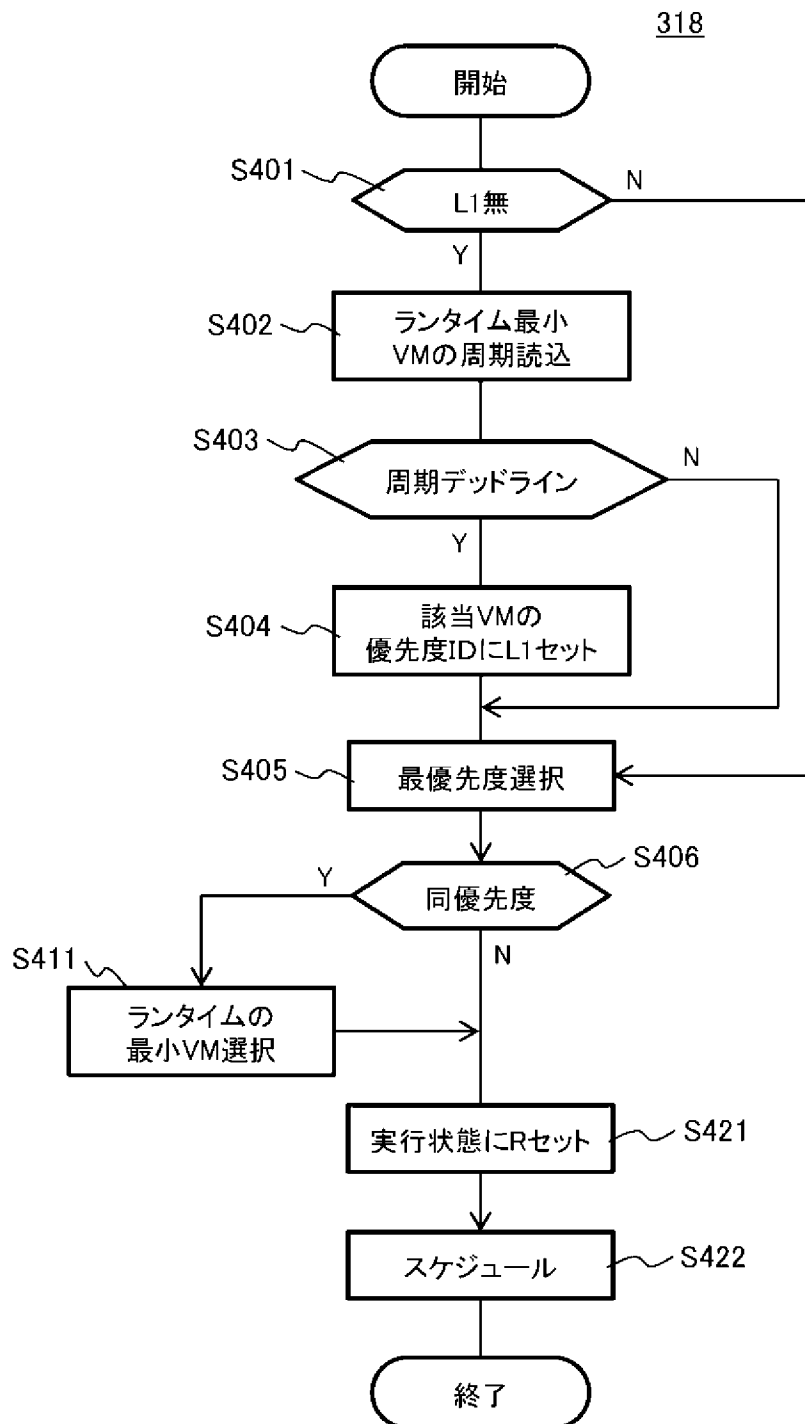
[図10]

図 10



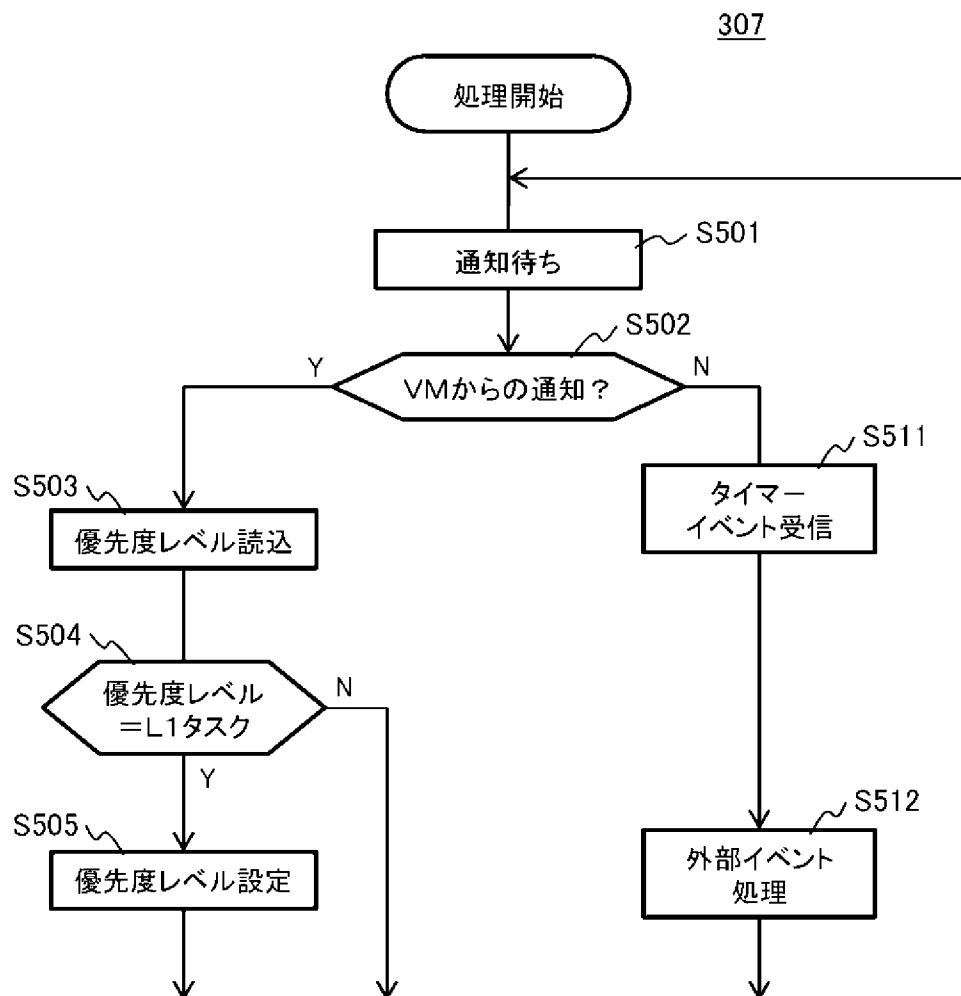
[図11]

図 11



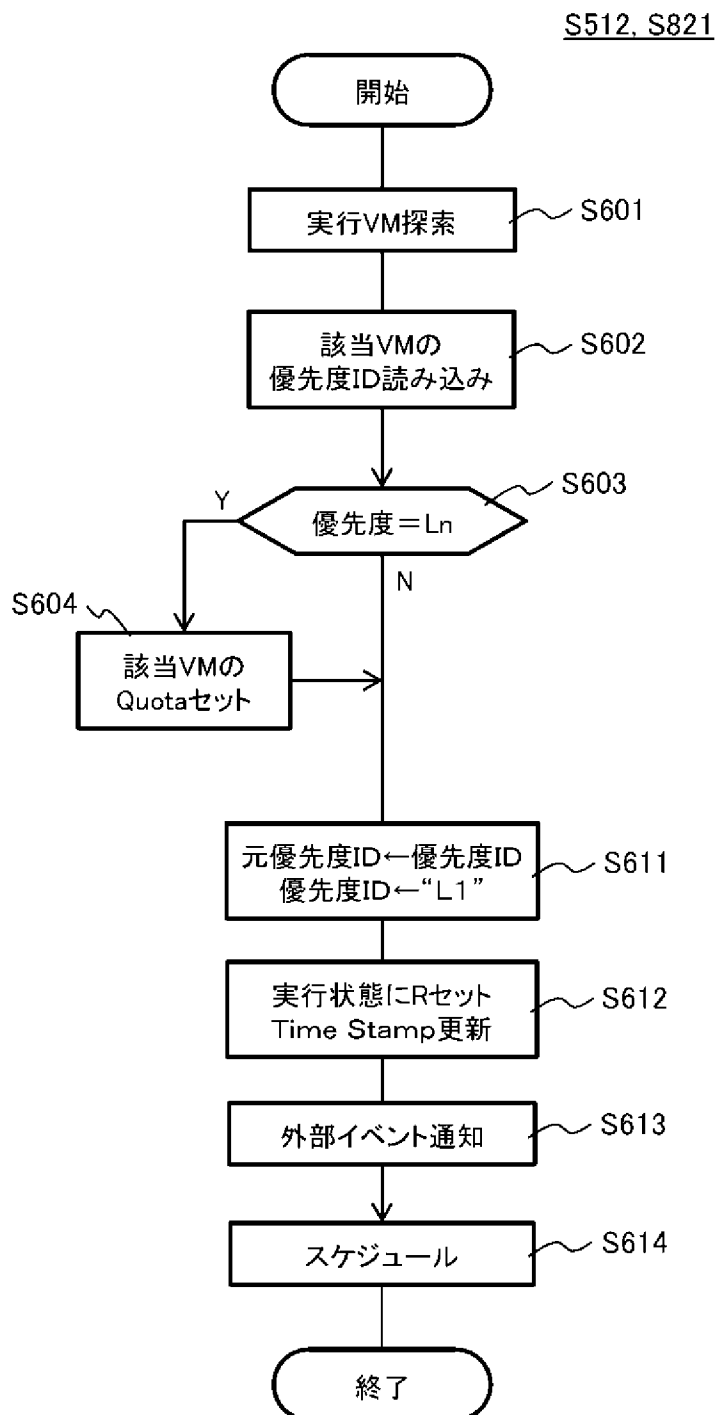
[図12]

図 12



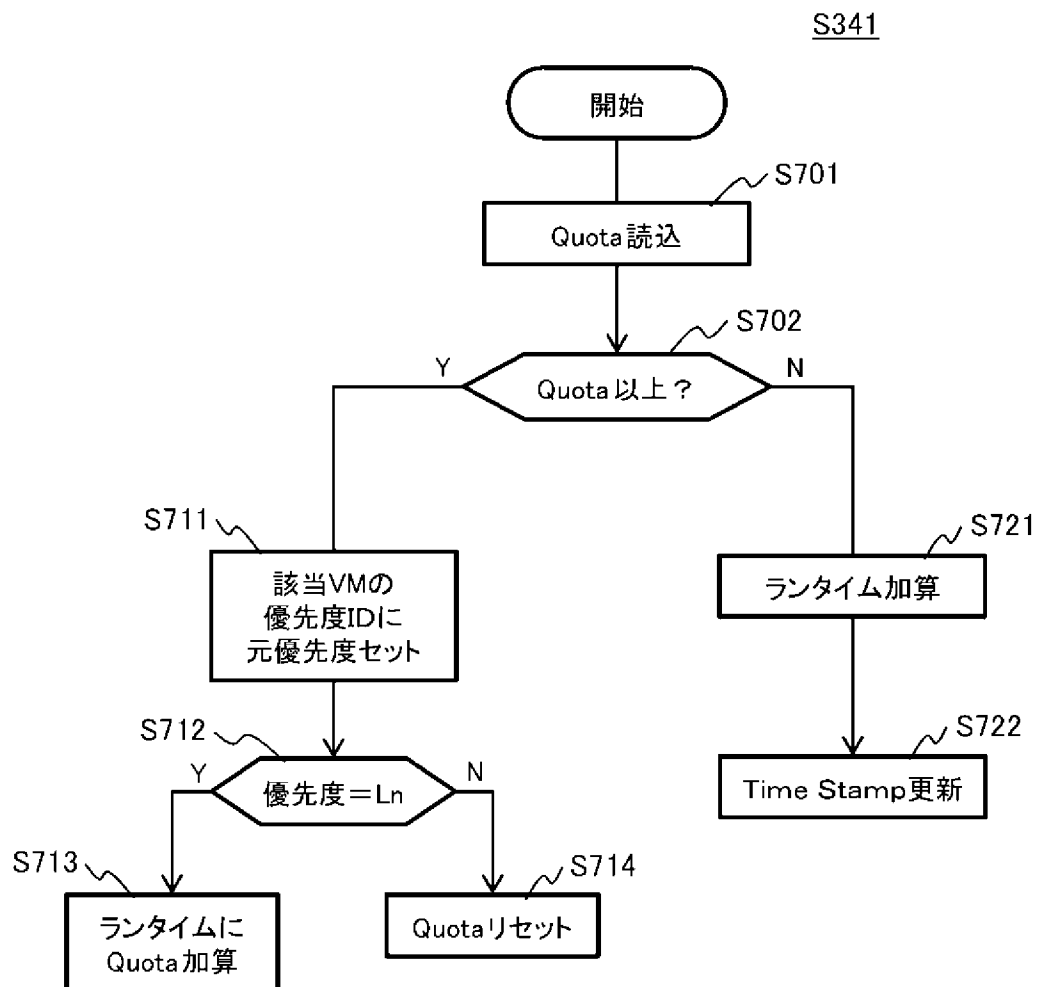
[図13]

図 13



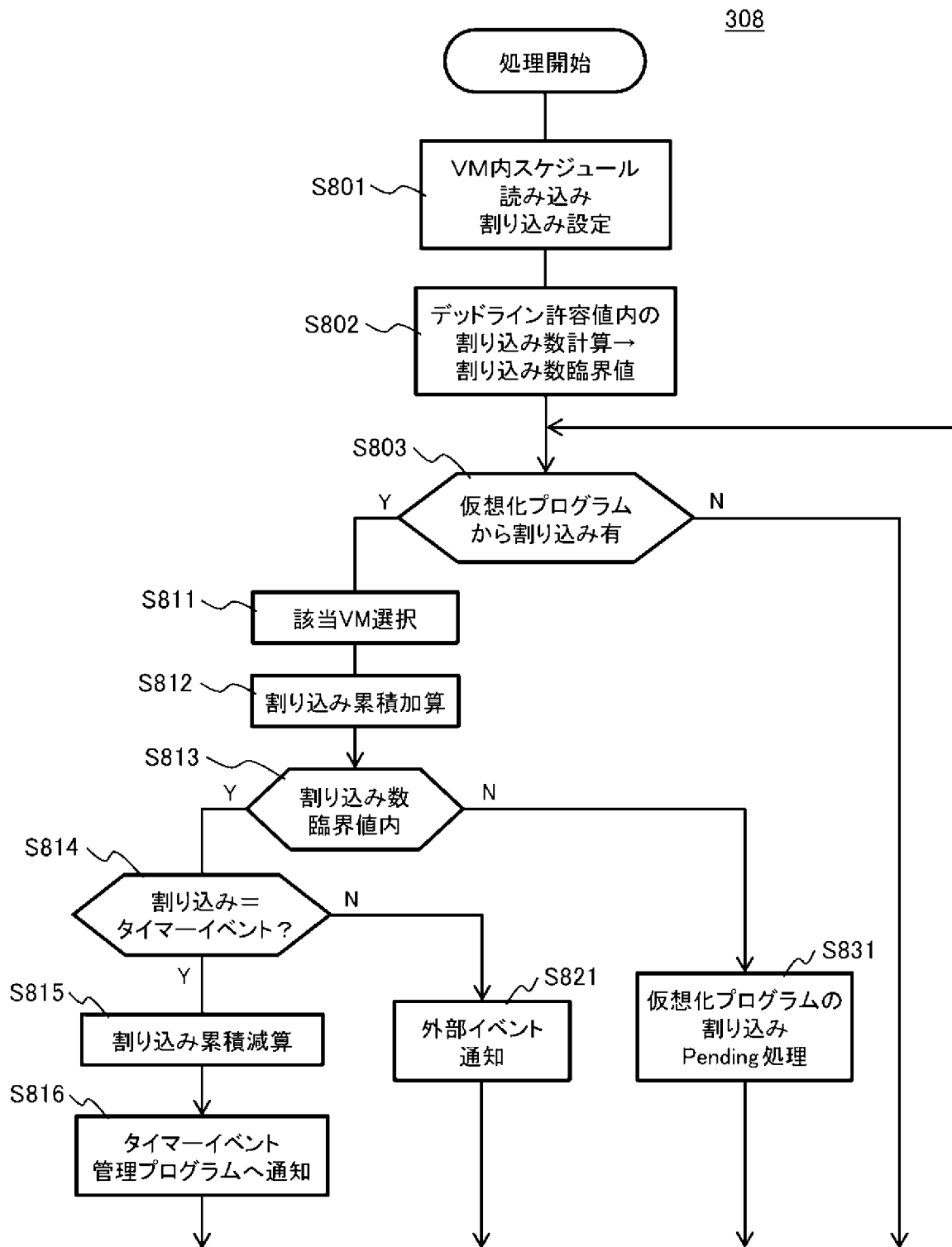
[図14]

図 14



[図15]

図 15



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2012/058295

A. CLASSIFICATION OF SUBJECT MATTER

G06F9/46(2006.01) i, G06F9/48(2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F9/46, G06F9/48

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2012
Kokai Jitsuyo Shinan Koho	1971-2012	Toroku Jitsuyo Shinan Koho	1994-2012

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2005-43959 A (Toshiba Corp.), 17 February 2005 (17.02.2005), abstract; paragraphs [0034], [0045] to [0049]; fig. 13, 53 & US 2005/0060709 A1 & EP 1501013 A2	1-4

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search
13 June, 2012 (13.06.12)

Date of mailing of the international search report
26 June, 2012 (26.06.12)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06F9/46(2006.01)i, G06F9/48(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（I P C））

Int.Cl. G06F9/46, G06F9/48

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 1 2 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 2 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 2 年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 2005-43959 A (株式会社東芝) 2005.02.17, 【要約】, 【0 0 3 4】, 【0 0 4 5】～【0 0 4 9】, 【図 1 3】, 【図 5 3】 & US 2005/0060709 A1 & EP 1501013 A2	1 - 4

C欄の続きにも文献が列挙されている。

パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日

1 3 . 0 6 . 2 0 1 2

国際調査報告の発送日

2 6 . 0 6 . 2 0 1 2

国際調査機関の名称及びあて先

日本国特許庁（I S A / J P）

郵便番号 1 0 0 - 8 9 1 5

東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

田中 幸雄

5 B

9 1 9 1

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 4 5