

(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
2020年1月9日(09.01.2020)



(10) 国際公開番号

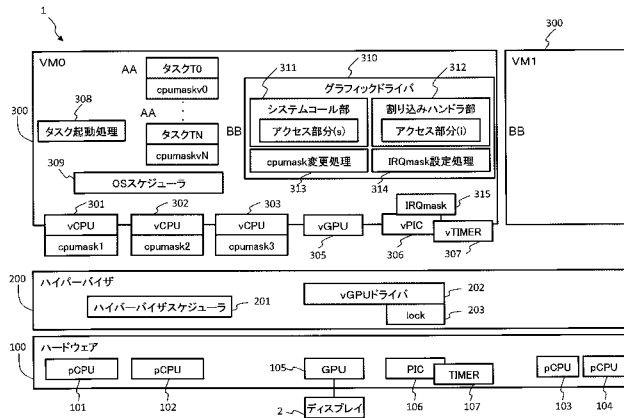
WO 2020/008757 A1

- (51) 国際特許分類: *G06F 13/10* (2006.01) *G06F 9/50* (2006.01)
G06F 9/455 (2018.01)
- (72) 発明者: 小川 修一 (OGAWA, Shuichi);
〒4488661 愛知県刈谷市昭和町1丁目1番地 株式会社デンソー内 Aichi (JP).
- (21) 国際出願番号: PCT/JP2019/021077
- (74) 代理人: 特許業務法人 サトー国際特許事務所 (SATO INTERNATIONAL PATENT FIRM);
〒4600008 愛知県名古屋市中区栄四丁目6番15号 フォーティーンヒルズセンタービル Aichi (JP).
- (22) 国際出願日: 2019年5月28日(28.05.2019)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2018-126799 2018年7月3日(03.07.2018) 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, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY,
- (71) 出願人: 株式会社デンソー (DENSO CORPORATION) [JP/JP]; 〒4488661 愛知県刈谷市昭和町1丁目1番地 Aichi (JP).

(54) Title: METHOD OF ACCESSING DEVICE IN VIRTUALIZED ENVIRONMENT

(54) 発明の名称: 仮想化環境におけるデバイスへのアクセス方法

[図1]



- 2 Display
- 100 Hardware
- 107 Timer
- 200 Hypervisor
- 201 Hypervisor scheduler
- 202 vGPU driver
- 203 Lock
- 307 timer
- 308 Task activation process
- 309 OS scheduler
- 310 Graphic driver
- 311 System call unit
- 312 Interrupt handler unit
- 313 cpumask change process
- 314 IRQmask setting process
- AA Task
- BB Access part

(57) Abstract: An access method of an embodiment causes a non-resident virtual CPU to deputize a process for accessing a virtual device (vGPU305), when resident virtual CPUs (vCPU301, vCPU302) to which physical CPUs (pCPU101, pCPU102) are usually allocated and a non-resident virtual CPU (vCPU303) to which a physical CPU is not usually allocated are embedded in a virtual machine 300 in a virtualized environment 1, and the resident virtual CPU accesses the virtual device (vGPU305) that corresponds to an occupying physical device (GPU105).



WO 2020/008757 A1

MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ,
NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT,
QA, RO, RS, RU, RW, SA, 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, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, 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), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類：

- 一 国際調査報告 (条約第21条(3))

(57) 要約：実施形態のアクセス方法は、仮想化環境1において、仮想マシン300上に、常には物理CPU (pCPU101、pCPU102) が割り当てられている常駐型の仮想CPU (vCPU301、vCPU302) と、常には物理CPUが割り当てられていない非常駐型の仮想CPU (vCPU303) とを実装し、常駐型の仮想CPUから占有型の物理デバイス (GPU105) に対応する仮想デバイス (vGPU305) にアクセスする際、当該仮想デバイスにアクセスする処理を非常駐型の仮想CPUに代行させる。

明 細 書

発明の名称： 仮想化環境におけるデバイスへのアクセス方法

関連出願の相互参照

[0001] 本出願は、2018年7月3日に出願された日本出願番号2018-126799号に基づくもので、ここにその記載内容を援用する。

技術分野

[0002] 本開示は、仮想的なコンピュータシステムを構築した仮想化環境においてデバイスにアクセスするためのアクセス方法に関する。

背景技術

[0003] 近年、ハードウェア上に仮想的なコンピュータシステムを構築した仮想化環境の採用が進んでいる。このような仮想化環境では、一般的に、1つのハードウェア上でハイパーバイザを動作させ、そのハイパーバイザ上で複数の仮想的なコンピュータシステムを動作させている。これにより、例えば複数の処理を並列して実行することが可能になる等、システムの高集積化や高性能化を実現することができる。以下、仮想的なコンピュータシステムを仮想マシンと称し、ハードウェアが備えるCPUを物理CPU、デバイスを物理デバイス、仮想マシン上に仮想化されたCPUを仮想CPU、デバイスを仮想デバイスと称する。

[0004] さて、仮想マシンは、ハードウェアが備えるストレージやネットワークインターフェース等の物理デバイスにアクセスしながら動作する。このとき、仮想マシン上では、仮想CPUが仮想デバイスにアクセスすることによって、物理デバイスにアクセスしている。そのため、複数の仮想マシンが構築されている場合であっても、各仮想マシンは、他の仮想マシンの存在を特段考慮することなく独自に動作することができる。

[0005] ただし、1つの物理デバイスに対して複数の仮想マシンから同時にアクセスすることはできないため、複数の仮想マシンが構築されている場合には、各仮想マシンは、物理デバイスを排他的に利用することになる。その場合、

アクセスするタイミングによっては他の仮想マシンからのアクセスの完了を待機する排他待ちが発生することから、排他待ちを効率的よく実現するために *virtio* と呼ばれる仮想化技術が存在している。

[0006] *virtio* は、ハイパーバイザ上のデバイスドライバが各仮想マシンからのアクセスの要求をキューイングし、物理デバイスへのアクセスが完了するとその要求に対して応答を返す構成となっている。このため、物理デバイスへのアクセスの要求とその要求に対する応答とが切り分けられ、アクセスを要求してから応答が返ってくるまでの期間がいわゆる I/O 待ちの状態となることから、アクセス元となった仮想 CPU では I/O 待ちの期間に他のタスクを実行することが可能になる。そして、このような *virtio* に対しては、例えば特許文献 1 のように割り込みのオーバーヘッドを改善する等、仮想化環境を効率化するための提案がなされている。

先行技術文献

特許文献

[0007] 特許文献 1：特表 2017-518589 号公報

発明の概要

[0008] ところで、物理デバイスの中には、アクセスの要求と応答とを切り分けることが困難なデバイスが存在する。例えば、Graphics Processing Unit(以下、GPU) は、独自のレジスタをアクセスして複雑な構造を持つ描画要求メモリを伝達する必要があることから、アクセスの要求と応答とを切り分けることが一般的に困難である。そのため、従来では、GPU にアクセスする場合には、描画要求メモリへの read/write 命令をトリガーとする描画の開始から描画が完了するまでの処理を一括で実行するようにしていた。

[0009] しかしながら、アクセスの終了までの処理を一括で実行する場合、アクセス元となる仮想 CPU は、描画が完了するまで GPU のレジスタへの read/write 命令を実行中になる。その場合、描画を処理するタスクを I/O 待ちの状態にすることができないことから、アクセス元の仮想 CPU は、GPU からの応答を待っているだけの状態であったとしても、その待ち時

間に他のタスクを実行することができなかった。

[0010] このように仮想CPUが占有された状態になると、他のタスクに切り替えることができなくなり、仮想マシンの処理能力が低下することになるとともに、割り当てた物理CPUの利用効率も低下してしまう等、仮想化環境全体のパフォーマンスの低下を招いていた。

[0011] 本開示の目的は、ハードウェア上に仮想的なコンピュータシステムを構築した仮想化環境において、アクセス元のCPUを占有するタイプのデバイスにアクセスする際のパフォーマンスの低下を抑制することができるアクセス方法を提供することにある。

[0012] 上記目的を達成するために、本開示の一態様によるアクセス方法では、仮想マシン上に、常には物理CPUに割り当てられている常駐型の仮想CPUと、常には物理CPUが割り当てられていない非常駐型の仮想CPUとを実装し、常駐型の仮想CPUから占有型の物理デバイスに対応する仮想デバイスにアクセスする際、当該仮想デバイスにアクセスする処理を非常駐型の仮想CPUに代行させる。

[0013] この場合、当初のアクセス元である常駐型の仮想CPUは、仮想デバイスにアクセスする処理から開放されることから、他の処理を実行することが可能になる。一方、仮想デバイスにアクセスする処理を代行する非常駐型の仮想CPUは、仮想マシン上においては仮想デバイスに占有された状態になるものの、その状態は、ハイパーバイザから見れば物理デバイスからの応答を待ち合わせて何も動作していないものとして認識される。

[0014] そのため、ハイパーバイザ上では、何も動作していない非常駐型の仮想CPUへの物理CPUの割り当てを解除でき、割り当てが解除された物理CPUを、他の処理を実行することが可能な状態となっている常駐型の仮想CPUに割り当てることができる。

[0015] これにより、仮想化環境において占有型の物理デバイスにアクセスする際、仮想マシン上では、非常駐型の仮想CPUによってアクセスする処理を継続したまま、常駐型の仮想CPUによって他の処理を実行することができる

。したがって、アクセス元のCPUを占有する占有型の物理デバイスにアクセスする場合であっても、仮想化環境のパフォーマンスが低下することを抑制できる。

図面の簡単な説明

- [0016] 本開示についての上記目的及びその他の目的、特徴や利点は、添付の図面を参照しながら下記の詳細な記述により、より明確になる。その図面は、
[図1]図1は、第1実施形態による仮想化環境の一例を模式的に示す図であり、
、
[図2]図2は、アクセス時の作動と物理CPUの割り当て状態との一例を模式的に示す図であり、
[図3]図3は、第2実施形態による仮想化環境の一例を模式的に示す図である。
。

発明を実施するための形態

- [0017] 以下、複数の実施形態について図面を参照しながら説明する。また、各実施形態において実質的に共通する構成には同一の符号を付して説明する。

- [0018] (第1実施形態)

以下、第1実施形態について、図1および図2を参照しながら説明する。

図1に示すように、本実施形態の仮想化環境1は、ハードウェア100、ハードウェア100上で動作するハイパーバイザ200、ハイパーバイザ200上で動作する複数の仮想マシン300を含んで構成されている。以下、ハードウェア100が備えるCPUを物理CPU、デバイスを物理デバイス、仮想マシン300上に仮想化されたCPUを仮想CPU、デバイスを仮想デバイスと称して説明する。

- [0019] ハードウェア100は、物理CPUに相当する例えば4つのpCPU101～pCPU104、物理デバイスに相当するGPU105 (Graphics Processing Unit)、Programmable Interrupt Controller (以下PIC) 106、およびTIMER107等を備えている。本実施形態の場合、ハードウェア100上には4つのCPUコアを単一パッケージに収容したCPUが設け

られており、これら4つのCPUコアをそれぞれ個別の物理CPUとして扱っている。また、各物理CPUは、pCPU101およびpCPU102がVM0の仮想マシン300に割り当てられ、pCPU103およびpCPU104がVM1の仮想マシン300に割り当てられている。

[0020] GPU105は、周知のようにディスプレイ2への描画を処理する機能を有しており、本実施形態では、独自のレジスタに複雑な構造を持つ描画要求メモリを伝達することで描画を処理する。このGPU105は、詳細は後述するが、GPU105へのアクセスの要求と応答とを切り分けることが困難であり、レジスタへのアクセス中にはアクセス元のCPUを占有するタイプのインターフェースを備えた占有型の物理デバイスに相当する。また、このGPU105は、それぞれの仮想マシン300によって共有される共有デバイスでもある。

[0021] PIC106は、GPU105等の物理デバイスからの割り込みの設定や割り込みの有効化／無効化あるいは割り込みの通知先となるCPUの設定等、ハイパーバイザ200上で発生する割り込みを制御するための周知の機能を有している。TIMER107は、ハイパーバイザ200上で必要となる各種の時間を計時する機能を有しており、例えばタイマ割り込み等の周知の処理に利用される。なお、図示は省略するが、ハードウェア100上には、例えばembedded Multi Media Card（以下、eMMC）等のストレージデバイスやNetwork Interface Card（以下、NIC）のような通信デバイス等の物理デバイス等も設けられている。

[0022] ハイパーバイザ200は、周知のようにハードウェア100を仮想化することによって例えば複数の仮想マシン300を動作させるソフトウェアである。本実施形態では、ハードウェア100上でハイパーバイザ200が直接動作し、そのハイパーバイザ200上で仮想マシン300が動作するベアメタル型あるいはネイティブ型と称されるものを想定している。このハイパーバイザ200には、ハイパーバイザスケジューラ201、vGPUドライバ202等が設けられている。

- [0023] ハイパーバイザスケジューラ201は、各仮想マシン300のスケジューリング等を行うソフトウェアである。vGPUドライバ202は、仮想マシン300上に仮想デバイスを実現するためのドライバソフトウェアであり、本実施形態ではGPU105を仮想化した仮想デバイスであるvGPU305を仮想マシン300上に実現している。
- [0024] vGPUドライバ202は、各仮想マシン300からのアクセスを排他的に処理する排他ロック機能203を有している。なお、図1では、排他ロック機能203をlockとして示している。vGPUドライバ202は、各仮想マシン300から物理デバイスへのアクセスが要求された場合、排他ロック機能203により各仮想マシン300からのアクセスを排他的に処理することで、それぞれの仮想マシン300からのアクセスの要求に対して正しく物理デバイスを利用できるように制御する。なお、図1では図示を省略しているが、ハイパーバイザ200には他のアプリケーションや他の物理デバイス用のドライバソフトウェア等も実装されている。
- [0025] 仮想マシン300は、周知のようにハイパーバイザ200上で動作する仮想的なコンピュータシステムであり、仮想CPU上で仮想Operating System（以下、仮想OS）を動作させ、仮想デバイスにアクセスしながら各種の処理を実行するように構成されている。本実施形態では、ハイパーバイザ200上に構築されているVM0およびVM1の2つの仮想マシン300は同一構成となっている。そのため、以下では主としてVM0の仮想マシン300を例にして説明する。なお、異なる構成の仮想マシン300を構築することも可能である。
- [0026] 仮想マシン300は、仮想CPUに相当するvCPU301、vCPU302およびvCPU303、仮想デバイスに相当するvGPU305、vPIC306、vTIMER307等を備えている。このうち、vCPU301およびvCPU302は、常には物理CPUに割り当てられて動作可能になっている常駐型の仮想CPUに相当し、vCPU303は、常には物理CPUに割り当てられておらず、必要に応じて物理CPUが割り当てられて動

作可能になる非常駐型の仮想CPUに相当する。つまり、仮想マシン300は、常には動作可能な仮想CPUが、割り当てられる物理CPUの数だけ実装されている。

[0027] 各仮想CPUには、割り当て可能な物理CPUを示す`cpumask`がそれぞれ設定されている。この`cpumask`には、割り当てが可能な物理CPUの識別情報が例えばビット列として設定されている。例えばVM0の場合、`vCPU301`、`vCPU302`および`vCPU303`の3つの仮想CPUには、それぞれ`pCPU101`および`pCPU102`の2つの物理CPUが割り当て可能であることを示す値が設定されている。なお、VM1の場合、3つの仮想CPUの`cpumask`には、`pCPU103`と`pCPU104`の2つの物理CPUが割り当て可能であることを示す値が設定される。つまり、各仮想マシン300上の各仮想CPUには、それぞれ物理CPUが重複しないように割り当てられている。

[0028] これにより、各仮想マシン300の仮想CPUに対して物理CPUが個別に配分され、互いの仮想マシン300の動作が他方の仮想マシン300のパフォーマンスに影響を与えるおそれを無くすることができる。なお、`cpumask`は割り当て可能な物理CPUを識別する値であって、実際にどの物理CPUが割り当てられるかは、ハイパーバイザスケジューラ201によって管理される。

[0029] 仮想マシン300上には、周知のように、各種のタスクT0~TNを起動するタスク起動処理308、起動されたタスクのスケジューリング等を行うOSスケジューラ309が実装されている。また、仮想マシン300上には、仮想デバイスにアクセスするためのドライバソフトウェアである仮想ドライバが実装されている。本実施形態では、`vGPU305`にアクセスするためのグラフィックドライバ310を想定している。なお、図1では図示を省略しているが、他の仮想デバイス用のドライバソフトウェアも実装されている。

[0030] 各タスクには、タスク起動処理308によって`cpumaskv`が設定さ

れる。この `cpumaskv` は、そのタスクを実行可能な仮想CPUの番号等の識別情報が設定されている。本実施形態の場合、各タスクの `cpumaskv` には、常駐型の仮想CPUである `vCPU301` および `vCPU302` での実行を許可する一方、非常駐型の仮想CPUである `vCPU303` での実行を許可しない値が設定される。そのため、起動したタスクは、`cpumaskv` を参照したOSスケジューラ309によって、`vCPU301` または `vCPU302` のいずれかに割り振られる。このとき、OSスケジューラ309は、各仮想CPUが均等に使用されるようにタスクを割り振る。

[0031] グラフィックドライバ310は、占有型の仮想デバイスである `vGPU305` にアクセスするためのものであり、仮想デバイスにアクセスするアクセス部分 (s) を有するシステムコール部311と、アクセス部分 (i) を有する割り込みハンドラ部312とを含んでいる。なお、システムコール部311および割り込みハンドラ部312の動作は周知であるので、説明を省略する。

[0032] また、グラフィックドライバ310は、`cpumask` 変更処理313、および、`IRQmask` 設定処理314を有している。詳細は後述するが、`cpumask` 変更処理313は、詳細は後述するがタスクに設定されている `cpumask` を変更する処理であり、`IRQmask` 設定処理314は、割り込みハンドラ部312を動作させる仮想CPUを示す `IRQmask` 315を設定する処理である。`vGPU305` の場合、仮想マシン300の起動時に、`IRQmask` 設定処理314によって割り込みハンドラ部312が `vCPU303` でのみ動作するように設定されている。

[0033] 次に、上記した構成の作用について説明する。

まず、ハイパーバイザ200を採用した仮想化環境1の動作について簡単に説明する。ハイパーバイザ200は、ハイパーバイザスケジューラ201、物理CPU、`PIC106` や `TIMER107` 等を利用しつつ仮想マシン300を並列動作させている。このとき、各仮想マシン300のリアルタイム性能を最大限に活かすため、`cpumask` を設定して物理CPUをそれ

それぞれの仮想マシン300に対して個別に配分し、他の仮想マシン300の影響を受けないように、また、他の仮想マシン300に影響を与えないようにしている。

[0034] 具体的には、例えばvCPU301とvCPU302が稼動状態となっており、vCPU303が待機状態となっている場合には、ハイパーバイザスケジューラ201は、vCPU301とvCPU302のcpumaskを参照して、例えばpCPU101をvCPU301に割り当て、pCPU102をvCPU302に割り当てる。なお、ここで言う稼動状態とは、仮想CPUに何らかの処理が割り振られた状態を意味しており、稼動状態の仮想CPUに物理CPUが割り当てられると、その仮想CPUは動作可能な状態になる。

[0035] 一方、ハイパーバイザ200は、vCPU303が稼動状態になると、pCPU101またはpCPU102をvCPU303に割り当て、その期間はvCPU301かvCPU302のいずれかは物理CPU待ちの状態になる。このように、ハイパーバイザ200は、仮想CPUに対して極力均等に物理CPUが割り当てることで、仮想化環境1が効率的に動作するようにしている。これにより、各仮想マシン300は、2つの物理CPUを備えたOSと同等の性能で動作することができ、複数のタスクの同時実行がハイパーバイザ200を使わない通常のOSと同様に実現されている。

[0036] さて、このような仮想化環境1では、各仮想マシン300は、ハードウェア100上のGPU105や上記したeMMCあるいはNIC等の物理デバイスを共有してアクセスすることがある。そのため、仮想化環境1では、例えばvGPUドライバ202をハイパーバイザ200上で動作させ、仮想マシン300上に仮想デバイスを実現することにより、各仮想マシン300では独立したデバイスのようにアクセスすることを可能としている。

[0037] ただし、ハードウェア100が備える物理デバイスが1つの場合には、複数の仮想マシン300からのアクセスの要求は時分割で排他的に処理されることになる。その場合、他の仮想マシン300がアクセスしている場合には

排他待ち状態となることから、前述した `virtualio` のような仮想化技術が存在している。

[0038] ただし、物理デバイスの中には、本実施形態の GPU 105 のようにアクセスの要求と応答とを切り分けることが困難であり、また、アクセス時にレジスタ命令の実行中となってタスクを I/O 待ちの状態にすることができないような占有型のデバイスが存在する。このような占有型のデバイスにアクセスする場合には、例えば GPU 105 からの応答を待っているだけの状態であったとしても、その待ち時間に他のタスクを実行することができないことから、仮想化環境 1 のパフォーマンスが低下することになる。

[0039] 具体的には、GPU 105 で言えば描画実行中には描画以外のタスクを実行できず、CPU の有効活用ができないことから、他処理の遅延を招いてしまうことになる。特に、描画処理では大量の仮想マシン 300 上のメモリにアクセスすることから、その待ち時間が例えば数十ミリ秒かかってしまうようなケースも想定される。その場合、他のタスクが実行できなくなると、画像とともに音声再生のタスクを同時実行している場合には、音切れを発生させてしまう。また、リアルタイム性を求める処理が実行されない場合や遅延する場合も想定される。

[0040] そこで、本実施形態では、占有型のデバイスへのアクセス中には仮想 CPU が占有されて他のタスクへの切り替えはできないものの、物理 CPU は何も処理をしていない状態になっていることに着目し、仮想化環境 1 のパフォーマンスが低下することを抑制している。

[0041] 以下、仮想マシン 300 上で起動されたタスク T0 から GPU 105 に対するアクセスの要求が発行される場合を例にして、本実施形態のアクセス方法について説明する。なお、一般的な OS の場合にはアプリケーションはプロセス空間を有するものでありスレッドと称されることもあるが、本明細書では、アプリケーションであっても OS 内部で実行されるものであっても、同時に実行可能なプログラム単位をタスクと称している。

[0042] さて、仮想マシン 300 上でタスク T0 が起動されると、タスク T0 は、

cpumaskvの設定によりvCPU301またはvCPU302に割り当てられる。ここでは、図2に示すように、例えば時刻t0においてタスクT0が起動され、タスクT0がvCPU301に割り当てられたとする。なお、図2の物理CPUの割り当て状態として示すように、タスクT0を実行するvCPU301には物理CPUとしてpCPU101が割り当てられているものとする。また、図2において各タスクに対応して設けた太線は、物理CPUが割り当てられていることを示している。

[0043] タスクT0が実行されてグラフィックドライバ310を呼び出されると、タスクT0ではシステムコール部311が動作する。システムコール部311では、まず、タスクT0のcpumaskv0をvCPU303でのみ実行可能となるように変更するcpumask変更処理313が実行される。また、このcpumask変更処理313では、cpumaskv0がvCPU301およびvCPU302を含まない値に変更されたことを確認すると、タスクT0を実行可能になったvCPU303にタスクT0を再割り当てするようにOSスケジューラ309に指示が出される。

[0044] OSスケジューラ309に指示が出されると、vCPU303が他のタスクに使用されている場合には一旦待ち状態となるが、vCPU303が使われていなかった場合あるいは待ち状態が終了した場合には、vCPU303にタスクT0が割り当られてvCPU303が稼働状態になる。仮想CPUの稼働状態はハイパーバイザ200上のハイパーバイザスケジューラ201によってチェックされていることから、vCPU303が稼働状態になったこともハイパーバイザスケジューラ201に検知されることになる。

[0045] vCPU303が稼働状態になったことを検知したハイパーバイザスケジューラ201は、vCPU301およびvCPU302に加えて、vCPU303を物理CPUの割り当て対象とする。この状態では一時的には3つの仮想CPUが2つの物理CPU上で動作することになるため、例えば図2の時刻t1においてvCPU303にpCPU101が割り当てられたとすると、それまでpCPU101が割り当てられていたvCPU301は一時的

に物理CPU待ちの状態になる。つまり、時刻 t_1 において、タスク T_0 を実行する仮想CPUが、 $vCPU301$ から $vCPU303$ に切り替えられる。

[0046] この時点でシステムコール部 311 が $vCPU303$ で動作可能な状態になることから、システムコール部 311 の処理を $vCPU303$ 上で継続することができる。そのため、システムコール部 311 は、アクセス部分 (s) を呼び出し、アクセス部分 (s) は $vGPU305$ のレジスタにアクセスする。 $vGPU305$ へのアクセスはハイパーバイザ 200 上の $vGPU$ ドライバ 202 に伝達されるものの、他の仮想マシン 300 の実行の待ち合わせや $GPU105$ へアクセスする準備のために排他ロック機能 203 によって一旦待機する。このとき、 $vCPU303$ は、ロック解除待ちで待機状態になる。

[0047] 例えば図2の時刻 t_2 において $vCPU303$ が待機状態になったとすると、処理が仮想マシン 300 からハイパーバイザ 200 に移管されるVMホスト切り替えが行われる。このとき、ハイパーバイザスケジューラ 201 は、待機状態になって何も処理を行っていない $vCPU303$ への $pCPU101$ の割り当てを解除するとともに、物理CPU待ちの状態となっている $vCPU301$ に $pCPU101$ を再び割り当てて $vCPU301$ を動作可能な状態とする。 $vCPU301$ が動作可能な状態になると、仮想マシン 300 上では例えばタスク T_1 を $vCPU301$ で実行することが可能になる。

[0048] そのため、3つの仮想CPUに2つの物理CPUを割り当てている状態であっても、 $vCPU303$ は待機状態となっていることから物理CPUの割り当てを解除でき、その結果、実質的に $vCPU301$ と $vCPU302$ の2つの仮想CPUで2つの物理CPUを占有的に使用することが可能になる。このため、仮にアクセス部分 (s) のレジスタへのアクセスが遅延したりアクセスに時間を要したりする場合であっても、仮想化環境1全体としての性能は実質的に2つの仮想CPUに2つの物理CPUを割り当てた状態と同等となり、仮想化環境1のパフォーマンスを維持することが可能になる。

- [0049] さて、vGPUドライバ202にて排他待ちのロックが解除されてGPU105へのアクセスが終了すると、その結果がアクセス部分(s)に返されることから、タスクT0が動作可能な状態、より具体的に言えば、中断していたアクセス部分(s)が動作を再開可能な状態になる。この場合、vCPU303に物理CPUを割り当てる必要があるが、タスクT0が動作可能な状態になったことを検知したハイパーバイザスケジューラ201により所定のスケジューリング時間後に物理CPUの割り当てが見直され、vCPU303に例えばpCPU101が割り当てられる。これにより、アクセス部分(s)は動作を再開する。
- [0050] そして、アクセス部分(s)の処理が終わると、cpumask変更処理313によって、cpumaskv0を、vCPU301およびvCPU302での実行を許可する一方、vCPU303での実行を許可しない値に変更する。つまり、cpumaskv0は元の値に戻される。その結果、タスクT0がvCPU301またはvCPU302の動作待ちになるとともに、vCPU303が何も処理をしていない状態になる。
- [0051] そのため、ハイパーバイザスケジューラ201は、何も処理をしていないvCPU303を物理CPUの割り当てを解除し、vCPU301とvCPU302に物理CPUを割り当てるように動作する。これにより、vCPU301とvCPU302は、それぞれpCPU101とpCPU102をフルタイムで利用可能な状態に復帰する。
- [0052] ここで注目すべきは、レジスタへのアクセス中は、タスクT0としては待機状態ではなくvCPU303上で動作状態であったことである。OSスケジューラ309は、動作状態にあるタスクから他のタスクに切り替えることができないため、従来であれば、タスクT0を実行していたvCPU301に他のタスクを実行させることができなかった。
- [0053] これに対して、本実施形態では、cpumask変更処理313によってタスクT0の実行主体をvCPU301からvCPU303に切り替えている。つまり、vGPU305にアクセスする処理をvCPU303に代行さ

せている。この状態では3つの仮想CPUが動作することになるものの、実質的にはvGPUドライバ202は排他待ちの状態になっていることから、ハイパーバイザスケジューラ201は、vCPU303には物理CPUを割り当てず、vCPU301とvCPU302に物理CPUが割り当てている。

[0054] つまり、非常駐型の仮想CPUであるvCPU303を設けて、vCPU303がvGPU305の動作完了を待ち合わせて待機状態となるように構成したことにより、vCPU301をタスクT0から開放することが可能となり、vCPU301に他のタスクを実行させることができ、仮想化環境1全体として効率的な動作が可能になったのである。

[0055] ところで、グラフィックドライバ310のシステムコール部311は、vGPU305のレジスタへのアクセスだけでシステムコールの目的を果たすものもあれば、vGPU305の状態の変化を待ち合わせてからレジスタアクセスをする場合もある。この場合、システムコール部311は要求内容をメモリ上に記録し、vGPU305の状態変化があったときに動作するように設定して終了する。

[0056] vGPU305の状態変化は、割り込みによって通知される。このとき、PIC106によってハイパーバイザスケジューラ201に割り込みが通知される。通知された割り込みは、vPIC306のIRQmask315によってvCPU303が設定されていることから、vCPU303で処理されることになる。そのため、ハイパーバイザスケジューラ201は、vCPU303の割り込みハンドラ部312を動作させるために、vCPU303に物理CPUを割り当てる。

[0057] 割り込みハンドラ部312が動作してアクセス部分(i)がvGPU305のレジスタにアクセスすると、vGPU305へのアクセスはハイパーバイザ200上のvGPUドライバ202に伝達されるものの、他の仮想マシン300の実行の待ち合わせやGPU105へアクセスする準備のために排他ロック機能203によって一旦待機する。このとき、vCPU303は、

ロック解除待ちで待機状態になる。このとき、ハイパーバイザスケジューラ 201 は、アクセス部分 (s) と同様に、排他待ち状態となった vCPU 303 への物理 CPU の割り当てを解除するとともに、vCPU 301 と vCPU 302 に物理 CPU を割り当てて動作させる。これにより、アクセス部分 (i) のレジスタへのアクセス待ちの間も仮想マシン 300 上の他のタスクを動作可能にすることができるようになる。

[0058] そして、vGPU ドライバ 202 の排他待ちが完了すると、アクセス部分 (i) が動作を再開し、vGPU ドライバ 202 を経由して GPU 105 のアクセスが可能となる。また、割り込みハンドラ部 312 の動作が終了すると、vCPU 303 が空きとなり、ハイパーバイザスケジューラ 201 は、vCPU 301 と vCPU 302 の処理だけを行うようになる。

[0059] このように、本実施形態では、グラフィックドライバ 310 において、アクセス部分 (s) およびアクセス部分 (i) の処理を非常駐型の vCPU 303 で行うことにより、レジスタへのアクセスを待ち合わせてアクセス元の CPU が占有されるような場合であっても、他のタスクが常駐型の仮想 CPU で動作できるアクセス方法を採用している。

[0060] 以上説明したアクセス方法によれば、次のような効果を得ることができる。

アクセス方法は、仮想化環境 1 において、仮想マシン 300 上に、常には物理 CPU が割り当てられている常駐型の仮想 CPU (vCPU 301、vCPU 302) と、常には物理 CPU が割り当てられていない非常駐型の仮想 CPU (vCPU 303) とを実装し、常駐型の仮想 CPU から占有型の物理デバイスに対応する仮想デバイスにアクセスする際、当該仮想デバイスにアクセスする処理を非常駐型の仮想 CPU に代行させる。

[0061] これにより、アクセス元となる常駐型の仮想 CPU から非常駐型の仮想 CPU に処理が移管され、非常駐型の仮想 CPU がアクセス中に占有状態になったとしても、常駐型の仮想 CPU において他のタスクを実行することが可能になる。この場合、アクセス中に占有状態になった非常駐型の仮想 CPU

が実質的に何も処理をしていない待機状態になれば物理CPUの割り当てを解除できるため、占有型のデバイスへのタスクを継続したまま、常駐型の仮想CPUに物理CPUを割り当てて他のタスクを実行させることができる。

[0062] したがって、占有型の物理デバイスにアクセスする場合であっても物理CPUを有効的に利用することができ、ハードウェア100上に仮想的なコンピュータシステムを構築した仮想化環境1において、占有型のデバイスにアクセスする際のパフォーマンスの低下を抑制することができる。

また、システムコール要求におけるレジスタアクセスは、システムコール呼び出しの延長時に行うものと、割り込みの延長時に行うものとの2つが想定されるものの、いずれの場合にも上記したアクセス方法を適用でき、占有型のデバイスへのアクセス時に物理CPUを有効活用することができる。

[0063] この場合、非常駐型の仮想CPUは稼動状態でなければ物理CPUの割り当て対象とはならないため、単純に3つの常駐型の仮想CPUを設けて時系列で3つの仮想CPUをスケジューリングする構成とは異なり、仮想化環境1のパフォーマンスに影響を与えることもない。したがって、仮想化環境1にリアルタイム性が求められる場合において、特に有意となる。

[0064] この場合、本実施形態のように、占有型の仮想デバイスにアクセスするためのドライバソフトウェアであって、システムコール部311と割り込みハンドラ部312とを含み、前記システムコール部311と前記割り込みハンドラ部312のそれぞれに仮想デバイスにアクセスするアクセス部分が設けられている仮想ドライバを実装し、常駐型の仮想CPUから占有型の仮想デバイスにアクセスする際、少なくともアクセス部分(s、i)を非常駐型の仮想CPUで代行するアクセス方法を採用することができる。

[0065] このような構成によっても、アクセス部分(s)を切り分けて処理することが難しいデバイスであっても占有型のデバイスへのアクセスを非常駐型の仮想CPUに容易に代行させることが可能となり、上記したように占有型のデバイスにアクセスする際のパフォーマンスの低下を抑制することができる。

[0066] また、本実施形態のように、ハイパーバイザ200上で2つ以上の仮想マシン300を動作させる場合であっても、それぞれの仮想マシン300において上記したアクセス方法を採用することにより、各仮想マシン300の性能を維持することができる。つまり、ハイパーバイザ200上で2つ以上の仮想マシン300を動作させ、複数の仮想マシン300に物理CPUを重複しないように割り当てる構成とすることで、より厳密には、各仮想マシン300上の各仮想CPUにそれぞれ物理CPUを重複しないように割り当てることで、仮想化環境1の性能が最も高くなると期待できる。

[0067] また、本実施形態ではハードウェア100が1つの占有型の物理デバイスを備える構成を例示したが、ハードウェア100が複数の占有型の物理デバイスを備えている場合には、ハードウェア100が備える占有型の物理デバイスと同じ数の非常駐型の仮想CPUを実装することにより、上記したアクセス方法を採用することができる。

[0068] 換言すると、仮想マシン300に割り当てられる物理CPUの数に等しい常駐型の仮想CPUと、仮想マシン300においてアクセス対象となる占有型の物理デバイスの数に等しい非常駐型の仮想CPUとを設けることにより、占有型の物理デバイスにアクセスするとき以外の期間においては従来と同等の性能を確保できるとともに、占有型の物理デバイスにアクセスする期間においては従来では不可能であった別のタスクの実行が可能となり、仮想化環境1の性能を向上させることができる。

[0069] ただし、常駐型の仮想CPUは、必要とされる場合には、仮想マシン300に割り当てられる物理CPUの数よりも多くすることもできる。そのような構成であっても、非常駐型の仮想CPUを別途設けることにより、占有型のデバイスにアクセスする際のパフォーマンスの低下を抑制することができる。

[0070] (第2実施形態)

以下、第2実施形態について、図3を参照しながら説明する。第2実施形態では、システムコール部311で動作するアクセス部分(s)を専用タス

クとしている点において第1実施形態と異なっている。

- [0071] 図3に示すように、本実施形態の仮想化環境1は、ハードウェア100、ハードウェア100上で動作するハイパーバイザ200、ハイパーバイザ200上で動作する複数の仮想マシン300を含んで構成されており、そのハードウェア100上でハイパーバイザ200が動作するとともに、そのハイパーバイザ200上でVM0およびVM1の2つの仮想マシン300が動作している。
- [0072] ハードウェア100は、第1実施形態と同様に、物理CPUに相当する4つのpCPU101~pCPU104、物理デバイスに相当するGPU105、PIC106、およびTIMER107等を備えている。このうち、pCPU101およびpCPU102は、VM0の仮想マシン300に割り当てられ、pCPU103およびpCPU104は、VM1の仮想マシン300に割り当てられている。
- [0073] GPU105は、第1実施形態と同様に、GPU105へのアクセスの要求と応答とを切り分けることが困難であり、レジスタへのアクセス中にはアクセス元のCPUを占有するタイプのインターフェースを備えた占有型の物理デバイスを想定している。
- [0074] ハイパーバイザ200は、第1実施形態と同様に、ハイパーバイザスケジューラ201、vGPUドライバ202等が設けられている。仮想マシン300は、第1実施形態と同様に、常駐型の仮想CPUに相当するvCPU301およびvCPU302、非常駐型の仮想CPUに相当するvCPU303、仮想デバイスに相当するvGPU305、vPIC306、vTIMER307等を備えているとともに、各種のタスクを起動するタスク起動処理308、起動されたタスクのスケジューリング等を行うOSスケジューラ309が実装されている。
- [0075] また、仮想マシン300上には、仮想ドライバとしてのグラフィックドライバ410が実装されている。本実施形態のグラフィックドライバ410は、専用タスク要求処理411により、システムコール部311から行われる

vGPU305へのアクセスを処理するアクセス部分(s)を専用タスクTSとして呼び出し可能に構成されている。この専用タスクTSは、vCPU303でのみ動作が可能となるようにcpumaskvSが設定されている。

[0076] さて、仮想マシン300上で起動したタスクT0からvGPU305へのアクセスが発生すると、グラフィックドライバ410が呼び出されてシステムコール部311が動作する。すると、システムコール部311は、専用タスク要求処理411によって専用タスクTSのアクセス部分(s)に処理の開始を要求するとともに、アクセス部分(s)の処理の完了を待ち合わせる。この時点でvCPU301は待ち状態となることから、他のタスクをvCPU301上で動作させることが可能になる。つまり、専用タスクTSの実行中にvCPU303が待ち状態になれば、vCPU301を動作させることができる。

[0077] この専用タスクTSは、cpumaskvSの設定によってvCPU303で処理を開始する。この状態では物理CPUを割り当てる仮想CPUが3つになるため、ハイパーバイザスケジューラ201は、pCPU101とpCPU102を3つの仮想CPUに均等に配分しようとする。そして、vCPU303に物理CPUが割り当てられると、専用タスクTSのアクセス部分(s)は、処理を開始してvGPUドライバ202をアクセスし、他の仮想マシン300の排他待ちであれば一時的に処理を待機し、排他待ちが解除されると処理を再開する。

[0078] アクセス部分(s)の処理が完了すると、vCPU303は、処理がなくなって待機状態となるとともに、完了が通知された専用タスク要求処理411が動作を再開し、システムコール部311は、第1実施形態と同様に割り込みハンドラ部312が動作するまでvGPU305の動作完了待ちの状態となる。

[0079] このように、本実施形態では、アクセス部分(s)を専用タスクとしてグラフィックドライバ410から呼び出す構成としたことにより、アクセスの

要求つまりはアクセス部分（s）への処理の開始と、その要求に対する応答つまりはアクセス部分（s）の処理の完了とを切り離すことが可能になる。これにより、vGPU305へのアクセスを言わば擬似的なI/Oとして行うことが可能となり、そのI/O待ちの期間に他のタスクを実行することが可能になる。

[0080] したがって、占有型の物理デバイスにアクセスする場合であっても物理CPUを有効的に利用することができ、ハードウェア100上に仮想的なコンピュータシステムを構築した仮想化環境1において、占有型のデバイスにアクセスする際のパフォーマンスの低下を抑制することができる。

[0081] この場合、第1実施形態と同様に、仮想化環境1のリソースを無駄に減らすことがなく、また、仮想化環境1のパフォーマンスに影響を与えることもないことから、仮想化環境1にリアルタイム性が求められる場合において特に有意となる。また、タスク起動処理308やOSスケジューラ309等の基本的な部分に手を加える必要もない。

[0082] また、本実施形態のように、占有型の仮想デバイスにアクセスするためのドライバソフトウェアであって、システムコール部311と割り込みハンドラ部312とを含み、システムコール部311から行われる仮想デバイスへのアクセスを処理するアクセス部分を、専用タスクTSとして呼び出し可能に構成した仮想ドライバを実装し、常駐型の仮想CPUから占有型の仮想デバイスにアクセスする際、専用タスクTSを非常駐型の仮想CPUで実行することにより、仮想デバイスへのアクセスを擬似的なI/Oとして行うことが可能となる。

[0083] これにより、仮想CPUの動作が第1実施形態よりもシンプルなものになるため、アクセス部分（s）を明確に分離できるデバイスを対象とする場合に容易に上記したアクセス方法を実現することができる。

[0084] ただし、アクセス部分（s）が明確に分離できないデバイスを対象とする場合には第1実施形態のアクセス方法の方がよい場合もあるため、例えば対象となる占有型の物理デバイスが複数存在する場合には、物理デバイスによ

って第1実施形態のアクセス方法と第2実施形態のアクセス方法とを使い分けるように、占有型の物理デバイスと同じ数の非常駐型の仮想CPUを実装するとともに、それぞれに適した複数の仮想ドライバを実装する構成とすることもできる。

[0085] (その他の実施形態)

第2実施形態では、仮想マシン300上に専用タスクTSを設ける構成を例示したが、ハイパーバイザ200上で類似の処理を行う構成とすることもできる。具体的には、ハイパーバイザ200上のハイパーバイザスケジューラ201に、仮想マシン300に加えて、占有型の物理デバイスに直接的にアクセス可能に構成したタスクをスケジューリング対象として配置し、非常駐型の仮想CPUからアクセスの要求があった際、ハイパーバイザ200上に配置したタスクを用いて占有型の物理デバイスにアクセスする構成とすることができる。

[0086] このような構成の場合、ハイパーバイザ200上のvGPUドライバ202は、タスクが物理デバイスへのアクセスを処理しているとき、第2実施形態のように、物理デバイスの動作が完了するまで仮想マシン300側からの処理を待ち合わせることが可能になる。そのため、非常駐型の仮想CPUからのアクセスの要求を擬似的なI/Oとして扱うことができ、I/O待ちの期間には非常駐型の仮想CPUへの物理CPUの割り当てを解除できることから、常駐型の仮想CPUにおいて他のタスクを実行することができる。

[0087] これにより、デバイスにアクセスする際のパフォーマンスの低下を抑制することができる、ハイパーバイザ200上で2つ以上の仮想マシン300を動作させる場合であっても各仮想マシン300の性能を維持することができる等、各実施形態と同様の効果を得ることができる。また、タスクをハイパーバイザスケジューラ201上に配置していることから、仮想マシン300上のOSスケジューラ309を介さずにその動作をスケジューリングすることができるため、例えばリアルタイム性が求められるアプリケーションに適用する等、優先度が高い処理に好適なものとなる。

- [0088] 各実施形態では占有型の物理デバイスへのアクセスを非常駐型の仮想CPUに代行させる例を示したが、アクセスに要する時間に基づいて非常駐型の仮想CPUに代行させるか否かを選択する構成とすることができる。具体的には、事前テスト等により計測した占有型の物理デバイスへのアクセスに要する最大待ち時間を予め設定し、最大待ち時間が予め設定した規定時間よりも長い場合にはアクセスする処理を非常駐型の仮想CPUに代行させる一方、最大待ち時間が規定時間よりも短い場合にはアクセスする処理を常駐型の仮想CPUで実行する構成とすることができる。
- [0089] これにより、例えばアクセスを代行させる処理に要する時間がアクセスを代行させることで得られる物理CPUの利用時間よりも長くなる場合等にはアクセスを代行させないことが可能となり、仮想化環境のパフォーマンスを意図せずに低下させてしまうことを抑制できる。
- [0090] 各実施形態では占有型の物理デバイスとしてGPU105を例示したが、本実施形態のアクセス方法は、GPU105に限定されず、例えばGPU105を画像処理以外の処理に応用するGeneral-purpose computing on graphics processing units（以下、GPGPU）や画像処理専用のImage processing unit（以下、IPU）、あるいはベンダー独自のインターフェースを有するようなデバイスへのアクセスにも適用することができる。
- [0091] 各実施形態では、4つの物理CPUを2つの仮想マシン300に割り当てた例を示したが、仮想マシン300の数は、これに限定されない。構築する仮想マシン300の数をMとすると、ハードウェア100が備える物理CPUの数がM以上であれば、各仮想マシン300に重複無く物理CPUを割り当てることができる。つまり、ハイパーバイザ200上で2つ以上の仮想マシン300を動作させ、複数の仮想マシン300に物理CPUを重複しないように割り当てる構成とすることができる。このとき、各仮想マシン300には割り当てられた物理CPUの数と同じ常駐型の仮想CPUを設け、対象となる占有型の物理デバイスの数と同じ非常駐型の仮想CPUを設け、上記したアクセス方法を適用することで、仮想化環境1のパフォーマンスが低下

することを抑制することができる。

[0092] 本開示は、実施例に準拠して記述されたが、本開示は当該実施例や構造に限定されるものではないと理解される。本開示は、様々な変形例や均等範囲内の変形をも包含する。加えて、様々な組み合わせや形態、さらには、それらに一要素のみ、それ以上、あるいはそれ以下、を含む他の組み合わせや形態をも、本開示の範疇や思想範囲に含まれるものである。

請求の範囲

[請求項1] 仮想化環境（1）においてデバイスにアクセスするためのアクセス方法であって、

前記仮想化環境（1）は、物理CPU（101、102、103、104）および物理デバイス（105）を備えたハードウェア（100）と、前記ハードウェア上で動作するハイパーバイザ（200）と、前記ハイパーバイザ上で動作し、前記物理CPUを仮想化した仮想CPU（301、302、303）および前記物理デバイスを仮想化した仮想デバイス（305）を備える仮想マシン（300）とを含み、

前記ハードウェアは、アクセスが要求されてからその要求に対する応答を返すまでアクセス元のCPUを占有する占有型の前記物理デバイス（105）を備えており、

前記仮想マシン上に、常には前記物理CPUが割り当てられている常駐型の前記仮想CPU（301、302）と、常には前記物理CPUが割り当てられていない非常駐型の前記仮想CPU（303）とを実装し、

常駐型の前記仮想CPU（301、302）から占有型の前記物理デバイス（105）に対応する前記仮想デバイス（305）にアクセスする際、アクセスする処理を非常駐型の前記仮想CPU（303）に代行させるアクセス方法。

[請求項2] 前記仮想マシン上に、占有型の前記仮想デバイスにアクセスするためのドライバソフトウェアであって、システムコール部（311）と割り込みハンドラ部（312）とを含み、前記システムコール部と前記割り込みハンドラ部とのそれぞれに前記仮想デバイスにアクセスするアクセス部分（s、i）が設けられている仮想ドライバ（310）を実装し、

前記アクセス部分を非常駐型の前記仮想CPUで実行する請求項1

に記載のアクセス方法。

[請求項3]

前記仮想マシン上に、占有型の前記仮想デバイスにアクセスするためのドライバソフトウェアであって、システムコール部と割り込みハンドラ部とを含み、前記システムコール部から行われる前記仮想デバイスへのアクセスを処理するアクセス部分（s）を、専用タスク（TS）として呼び出し可能に構成した仮想ドライバ（410）を実装し、

前記専用タスクを非常駐型の前記仮想CPUで実行する請求項1に記載のアクセス方法。

[請求項4]

前記ハイパーバイザ上のハイパーバイザスケジューラ（201）に、前記仮想マシンに加えて、占有型の前記物理デバイスに直接的にアクセス可能に構成したタスクをスケジューリング対象として配置し、

常駐型の前記仮想CPUから占有型の前記物理デバイスに対応する前記仮想デバイスにアクセスする際、前記専用タスクを非常駐型の前記仮想CPUで実行し、

非常駐型の前記仮想CPUからアクセスの要求があった際、前記ハイパーバイザ上に配置したタスクを用いて占有型の前記物理デバイスにアクセスする請求項1に記載のアクセス方法。

[請求項5]

占有型の前記物理デバイスへのアクセスに要する最大待ち時間を予め設定し、

前記最大待ち時間が予め設定した規定時間よりも長い場合にはアクセスする処理を非常駐型の前記仮想CPUに代行させる一方、前記最大待ち時間が前記規定時間よりも短い場合にはアクセスする処理を常駐型の前記仮想CPUで実行する請求項1から4のいずれか一項に記載のアクセス方法。

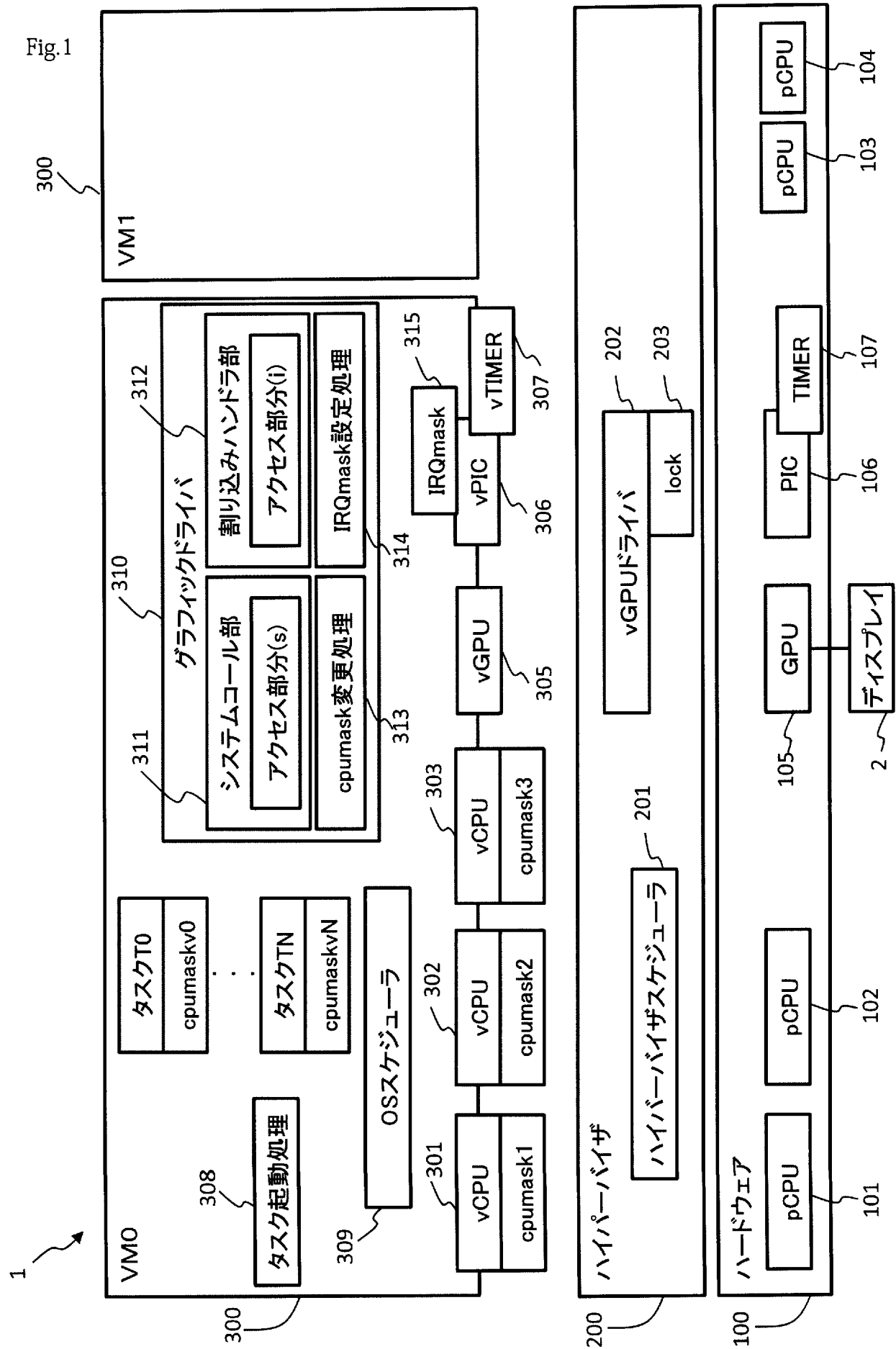
[請求項6]

前記ハイパーバイザ上で2つ以上の仮想マシンを動作させ、

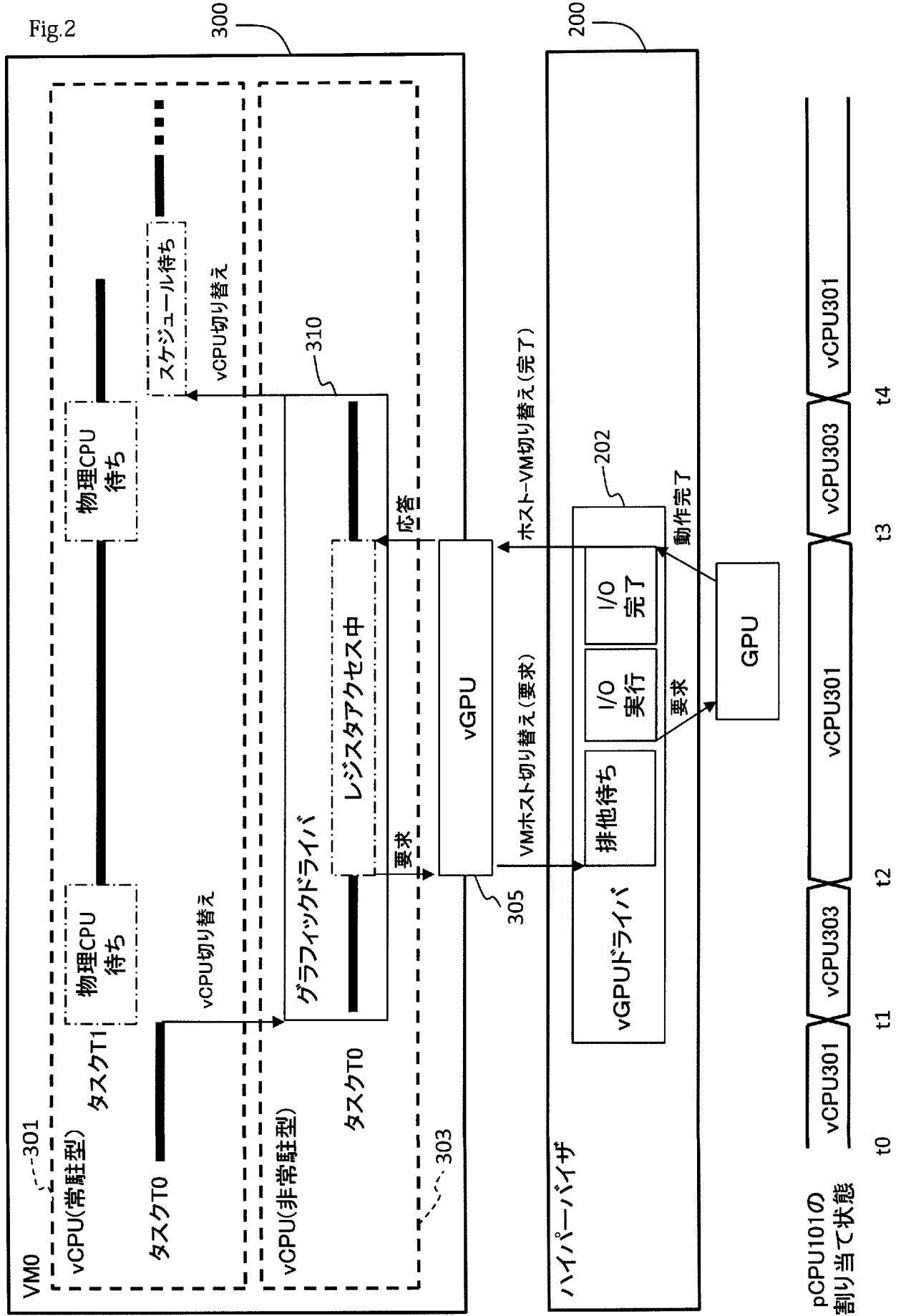
複数の前記仮想マシンに前記物理CPUを重複しないように割り当てて請求項1から5のいずれか一項に記載のアクセス方法。

[請求項7] 前記仮想マシンに、前記ハードウェアが備える前記物理デバイスと同じ数の非常駐型の前記仮想CPUを実装する請求項1から6のいずれか一項に記載のアクセス方法。

[図1]

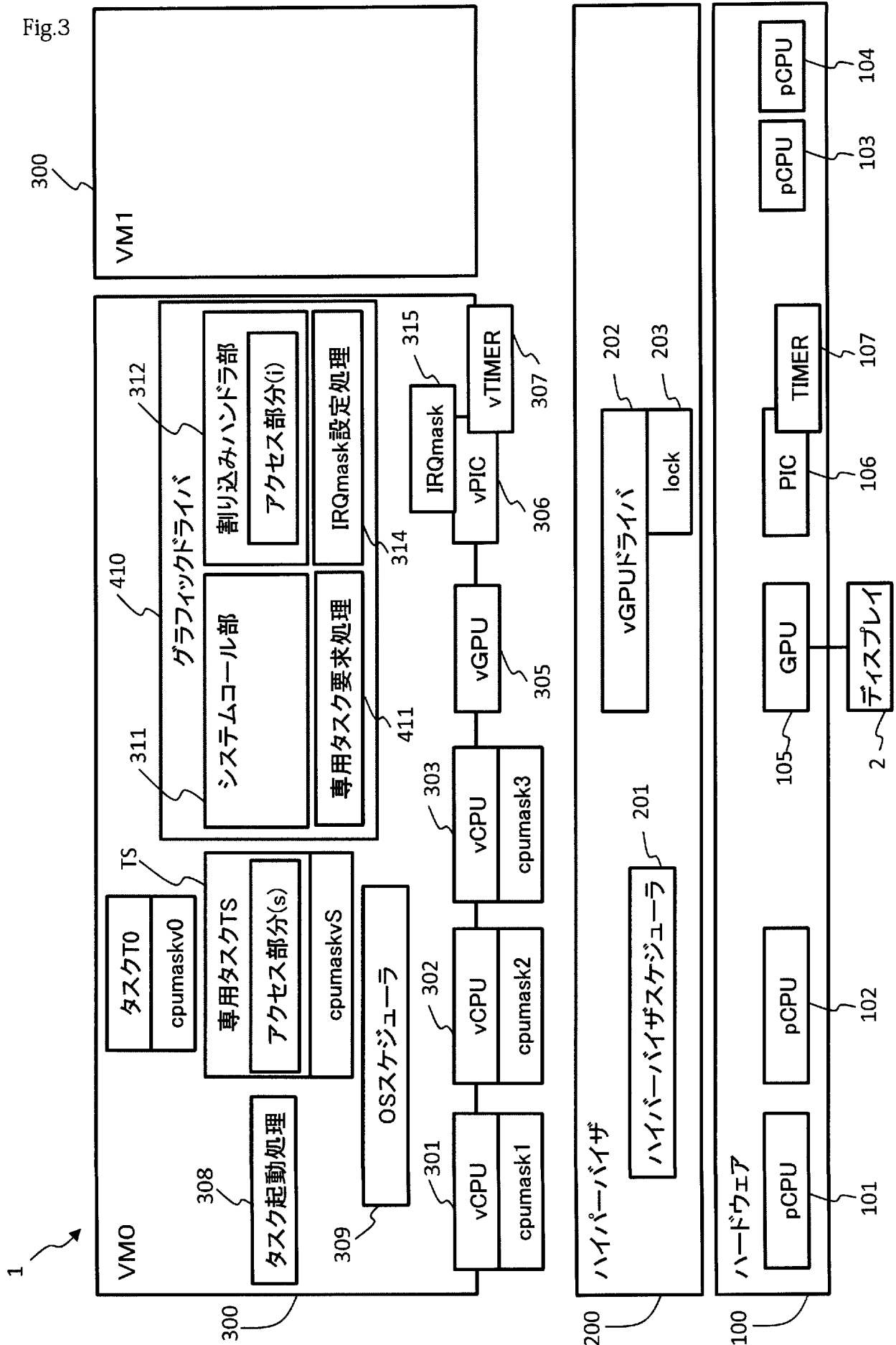


[図2]



[図3]

Fig.3



INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2019/021077

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl. G06F13/10 (2006.01) i, G06F9/455 (2006.01) i, G06F9/50 (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)
Int.Cl. G06F13/10, G06F9/455, G06F9/50

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

Published examined utility model applications of Japan	1922-1996
Published unexamined utility model applications of Japan	1971-2019
Registered utility model specifications of Japan	1996-2019
Published registered utility model applications of Japan	1994-2019

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 5-250263 A (HITACHI, LTD.) 28 September 1993, paragraphs [0059]-[0070], fig. 2 (Family: none)	1-7
A	WO 2017/094132 A1 (HITACHI, LTD.) 08 June 2017, paragraph [0039] (Family: none)	1-7
A	JP 2006-209386 A (HITACHI, LTD.) 10 August 2006, paragraph [0017] (Family: none)	1-7

Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents:	"I" 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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier application or patent but published on or after the international filing date	"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
"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)	"&" document member of the same patent family
"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	

Date of the actual completion of the international search 02 August 2019 (02.08.2019)	Date of mailing of the international search report 20 August 2019 (20.08.2019)
--	---

Name and mailing address of the ISA/ Japan Patent Office 3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan	Authorized officer Telephone No.
--	---

A. 発明の属する分野の分類（国際特許分類（IPC））
 Int.Cl. G06F13/10(2006.01)i, G06F9/455(2006.01)i, G06F9/50(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（IPC））
 Int.Cl. G06F13/10, G06F9/455, G06F9/50

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	1922-1996年
日本国公開実用新案公報	1971-2019年
日本国実用新案登録公報	1996-2019年
日本国登録実用新案公報	1994-2019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 5-250263 A（株式会社日立製作所）1993.09.28, 段落[0059]-[0070], 図2（ファミリーなし）	1-7
A	WO 2017/094132 A1（株式会社日立製作所）2017.06.08, 段落[0039]（ファミリーなし）	1-7
A	JP 2006-209386 A（株式会社日立製作所）2006.08.10, 段落[0017]（ファミリーなし）	1-7

☐ 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	の日の後に公表された文献
「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	「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
「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	「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
「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）	「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
「O」口頭による開示、使用、展示等に言及する文献	「&」同一パテントファミリー文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願	

国際調査を完了した日 02.08.2019	国際調査報告の発送日 20.08.2019
--------------------------	--------------------------

国際調査機関の名称及びあて先 日本国特許庁（ISA/J P） 郵便番号100-8915 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） 田名網 忠雄	5 S	6306
	電話番号 03-3581-1101 内線 3546		