

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6140190号
(P6140190)

(45) 発行日 平成29年5月31日 (2017.5.31)

(24) 登録日 平成29年5月12日 (2017.5.12)

(51) Int. Cl.		F I			
G06F	9/48	(2006.01)	G06F	9/46	457
G06F	9/46	(2006.01)	G06F	9/46	350
G06T	15/00	(2011.01)	G06T	15/00	501

請求項の数 12 (全 18 頁)

(21) 出願番号	特願2014-553324 (P2014-553324)	(73) 特許権者	500046438
(86) (22) 出願日	平成25年1月10日 (2013.1.10)		マイクロソフト コーポレーション
(65) 公表番号	特表2015-511346 (P2015-511346A)		アメリカ合衆国 ワシントン州 9805
(43) 公表日	平成27年4月16日 (2015.4.16)		2-6399 レッドモンド ワン マイ
(86) 国際出願番号	PCT/US2013/020916		クロソフト ウェイ
(87) 国際公開番号	W02013/109451	(74) 代理人	100140109
(87) 国際公開日	平成25年7月25日 (2013.7.25)		弁理士 小野 新次郎
審査請求日	平成28年1月5日 (2016.1.5)	(74) 代理人	100075270
(31) 優先権主張番号	13/352, 121		弁理士 小林 泰
(32) 優先日	平成24年1月17日 (2012.1.17)	(74) 代理人	100101373
(33) 優先権主張国	米国 (US)		弁理士 竹内 茂雄
		(74) 代理人	100118902
			弁理士 山本 修
		(74) 代理人	100153028
			弁理士 上田 忠

最終頁に続く

(54) 【発明の名称】 準仮想化された高性能コンピューティングおよびGDI高速化

(57) 【特許請求の範囲】

【請求項1】

物理グラフィックス処理装置（「GPU」）を含むコンピューターシステムにおいて、準仮想化実行環境の子パーティション内で実行されるゲストオペレーティングシステムにGPU高速化コンピューティング機能を提供するための、前記コンピューターシステムが実行する方法であって、

仮想マシンセッションをインスタンス化する動作であって、ハイパーバイザーをインスタンス化することを含み、該ハイパーバイザーは、（i）前記物理GPUへのアクセスを有するルートパーティションと、（ii）前記ゲストオペレーティングシステムを実行する前記子パーティションとを提供する、動作と、

グラフィックスデバイスインターフェイス（「GDI」）インターフェイスを前記ゲストオペレーティングシステムに供する動作であって、前記GDIインターフェイスは前記子パーティションのユーザーモード内で実行され、前記GDIインターフェイスは、1以上のGDIコマンドインターフェイスを前記ゲストオペレーティングシステムに関連するグラフィックスランタイムに公開し、前記グラフィックスランタイムからのGDI高速化コマンドを、前記子パーティション内で実行される仮想化グラフィックス処理装置（「vGPU」）のカーネルモードドライバ（「KMD」）に転送する、動作と、

vGPUを前記ゲストオペレーティングシステムに供する動作であって、

前記子パーティションの前記ユーザーモード内で実行され、複数のコンピュータシェーダーデバイスドライバインターフェイス（「DDI」）を前記ゲストオペレーティング

10

20

システムに供するユーザーモードドライバ(「UMD」)のアプリケーションプログラミングインターフェイスを供することであって、前記複数のコンピュータシェーダーDDIは、コンピュータシェーダーを用いて前記物理GPUにおいて汎用グラフィックス処理装置(「GPGPU」)計算を行うために、前記ゲストオペレーティングシステムがコンピュータシェーダーコマンドを前記vGPUに送ることを可能にする、供することと、

前記子パーティションのカーネルモードにおいて実行され、

1以上の高速化GDIオペレーションを前記GDIインターフェイスから前記物理GPUでの実行のために受けるための1以上のGDIコマンドインターフェイスを実装するGDIコンポーネントと、

コマンドバッファを前記UMDから受け、対応するダイレクトメモリアクセス(「DMA」)バッファを構築するコンピュータシェーダーコンポーネントとを含むカーネルモードドライバ(「KMD」)のアプリケーションプログラミングインターフェイスを供することと

を含む動作と、

前記ルートパーティション内で実行されるレンダーコンポーネントにおいて、物理GPU固有コンピュータシェーダーコマンドおよびGDI高速化レンダリングコマンドを前記vGPUの前記KMDから受ける動作と、

前記レンダーコンポーネントにより、前記物理GPUにおける実行のために前記物理GPU固有コンピュータシェーダーコマンドをスケジュールする動作と、

前記レンダーコンポーネントにより、前記GDI高速化レンダリングコマンドを前記ルートパーティション内のGDI合成デバイスにスケジュールする動作であって、前記GDI合成デバイスは、前記GDI高速化レンダリングコマンドを前記物理GPUにおいて実行するように構成された、動作と

を含む方法。

【請求項2】

請求項1に記載の方法であって、前記UMDは、前記ゲストオペレーティングシステムから受けた全てのコンピュータシェーダーコマンドを、対応する物理GPU固有コンピュータシェーダーコマンドに変換し、前記対応する物理GPU固有コンピュータシェーダーコマンドをコマンドバッファに記憶する、方法。

【請求項3】

請求項2に記載の方法であって、前記UMDは、物理GPU固有コンピュータシェーダーコマンドを含む前記コマンドバッファを前記KMDに送る、方法。

【請求項4】

請求項1に記載の方法であって、前記GDIコンポーネントは、前記GDI高速化レンダリングコマンドを実行するためのGDIサーフェスの生成を可能にする、方法。

【請求項5】

請求項1に記載の方法であって、前記レンダーコンポーネントは、前記物理GPU固有コンピュータシェーダーコマンドを実行するために、前記コンピュータシェーダーコンポーネントを用いる、方法。

【請求項6】

請求項1に記載の方法であって、前記UMDは、レガシーUMDと同時に実行することが可能な新しいUMDを含む、方法。

【請求項7】

準仮想化実行環境の子パーティション内で実行されるゲストオペレーティングシステムにGPU高速化コンピューティング機能を提供するための方法を実行するコンピューターシステムであって、前記方法は、

仮想マシンセッションをインスタンス化する動作であって、ハイパーバイザーをインスタンス化することを含み、該ハイパーバイザーは、(i)前記物理GPUへのアクセスを有するルートパーティションと、(ii)前記ゲストオペレーティングシステムを実行する前記子パーティションとを提供する、動作と、

10

20

30

40

50

グラフィックスデバイスインターフェイス(「GDI」)インターフェイスを前記ゲストオペレーティングシステムに供する動作であって、前記GDIインターフェイスは前記子パーティションのユーザーモード内で実行され、前記GDIインターフェイスは、1以上のGDIコマンドインターフェイスを前記ゲストオペレーティングシステムに関連するグラフィックスランタイムに公開し、前記グラフィックスランタイムからのGDI高速化コマンドを、前記子パーティション内で実行される仮想化グラフィックス処理装置(「VGPU」)のカーネルモードドライバ(「KMD」)に転送する、動作と、

前記VGPUを前記ゲストオペレーティングシステムに供する動作であって、

前記子パーティションの前記ユーザーモード内で実行され、複数のコンピュータシェーダーデバイスドライバインターフェイス(「DDI」)を前記ゲストオペレーティングシステムに供するユーザーモードドライバ(「UMD」)のアプリケーションプログラミングインターフェイスを供することであって、前記複数のコンピュータシェーダーDDIは、コンピュータシェーダーを用いて前記物理GPUにおいて汎用グラフィックス処理装置(「GPGPU」)計算を行うために、前記ゲストオペレーティングシステムがコンピュータシェーダーコマンドを前記VGPUに送ることを可能にする、供すること、

前記子パーティションのカーネルモードにおいて実行され、

1以上の高速化GDIオペレーションを前記GDIインターフェイスから前記物理GPUでの実行のために受けるための1以上のGDIコマンドインターフェイスを実装するGDIコンポーネントと、

コマンドバッファを前記UMDから受け、対応するダイレクトメモリアクセス(「DMA」)バッファを構築するコンピュータシェーダーコンポーネントとを含むカーネルモードドライバ(「KMD」)のアプリケーションプログラミングインターフェイスを供することを含む動作と、

前記ルートパーティション内で実行されるレンダーコンポーネントにおいて、物理GPU固有コンピュータシェーダーコマンドおよびGDI高速化レンダリングコマンドを前記VGPUの前記KMDから受ける動作と、

前記レンダーコンポーネントにより、前記物理GPUにおける実行のために前記物理GPU固有コンピュータシェーダーコマンドをスケジュールする動作と、

前記レンダーコンポーネントにより、前記GDI高速化レンダリングコマンドを前記ルートパーティション内のGDI合成デバイスにスケジュールする動作であって、前記GDI合成デバイスは、前記GDI高速化レンダリングコマンドを前記物理GPUにおいて実行するように構成され、前記GDI合成デバイスはまた、前記GDI高速化レンダリングコマンドに対応するGDIサーフェスを、デスクトップによる合成のために共有可能とマークするように構成された、動作とを含む、コンピュータシステム。

【請求項8】

請求項7に記載のコンピュータシステムであって、前記GDI高速化レンダリングコマンドは、前記物理GPUにおける前記コンピュータシェーダーの一部として使用可能である、コンピュータシステム。

【請求項9】

請求項7に記載のコンピュータシステムであって、前記UMDは、前記ゲストオペレーティングシステムから受けたグラフィックスコマンドを、対応するハードウェア固有コンピュータシェーダーコマンドに変換し、前記対応するハードウェア固有コンピュータシェーダーコマンドをコマンドバッファに記憶する、コンピュータシステム。

【請求項10】

請求項7に記載のコンピュータシステムであって、前記VGPUは、前記物理GPUへのアクセスを提供するためのソフトウェアインターフェイスを含む、コンピュータシステム。

【請求項11】

請求項 7 に記載のコンピューターシステムであって、前記 UMD は、コンピュータシェーダーを含むレンダリングフレームワークに対応するすべての DDI を供することが可能な新しい UMD を含み、前記レンダリングフレームワークは、ドメインシェーダーと、ハルシェーダーと、ジオメトリシェーダーとのうちの 1 つまたは複数も含むレンダリングフレームワークを含む、コンピューターシステム。

【請求項 12】

準仮想化実行環境の子パーティション内で実行されるゲストオペレーティングシステムに GPU 高速化コンピューティング機能を提供するための方法をコンピューターシステムに実行させるコンピュータープログラムであって、前記方法は、

仮想マシンセッションをインスタンス化する動作であって、ハイパーバイザーをインスタンス化することを含み、該ハイパーバイザーは、(i) 物理 GPU へのアクセスを有するルートパーティションと、(ii) 前記ゲストオペレーティングシステムを実行する前記子パーティションとを提供する、動作と、

グラフィックスデバイスインターフェイス(「GDI」)インターフェイスを前記ゲストオペレーティングシステムに供する動作であって、前記 GDI インターフェイスは前記子パーティションのユーザーモード内で実行され、前記 GDI インターフェイスは、1 以上の GDI コマンドインターフェイスを前記ゲストオペレーティングシステムに関連するグラフィックスランタイムに公開し、前記グラフィックスランタイムからの GDI 高速化コマンドを、前記子パーティション内で実行される仮想化グラフィックス処理装置(「vGPU」)のカーネルモードドライバ(「KMD」)に転送する、動作と、

前記 vGPU を前記ゲストオペレーティングシステムに供する動作であって、

前記子パーティションの前記ユーザーモード内で実行され、複数のコンピュータシェーダーデバイスドライバインターフェイス(「DDI」)を前記ゲストオペレーティングシステムに供するユーザーモードドライバ(「UMD」)のアプリケーションプログラミングインターフェイスを供することであって、前記複数のコンピュータシェーダー DDI は、コンピュータシェーダーを用いて前記物理 GPU において汎用グラフィックス処理装置(「GPGPU」)計算を行うために、前記ゲストオペレーティングシステムがコンピュータシェーダーコマンドを前記 vGPU に送ることを可能にする、供することと、

前記子パーティションのカーネルモードにおいて実行され、

1 以上の高速化 GDI オペレーションを前記 GDI インターフェイスから前記物理 GPU での実行のために受けるための 1 以上の GDI コマンドインターフェイスを実装する GDI コンポーネントと、

コマンドバッファを前記 UMD から受け、対応するダイレクトメモリアクセス(「DMA」)バッファを構築するコンピュータシェーダーコンポーネントとを含むカーネルモードドライバ(「KMD」)のアプリケーションプログラミングインターフェイスを供することと

を含む動作と、

前記ルートパーティション内で実行されるレンダーコンポーネントにおいて、少なくとも 1 つの物理 GPU 固有コンピュータシェーダーコマンドおよび少なくとも 1 つの GDI 高速化レンダリングコマンドを前記 vGPU の前記 KMD から受ける動作と、

前記物理 GPU における実行のために前記少なくとも 1 つの物理 GPU 固有コンピュータシェーダーコマンドをスケジュールする動作と、

前記少なくとも 1 つの GDI 高速化レンダリングコマンドを前記ルートパーティション内の GDI 合成デバイスにスケジュールする動作であって、前記 GDI 合成デバイスは、前記 GDI 高速化レンダリングコマンドを前記物理 GPU において実行するように構成され、前記 GDI 合成デバイスはまた、前記 GDI 高速化レンダリングコマンドに対応する GDI サーフェスを、デスクトップによる合成のために共有可能とマークするように構成された、動作と

を含む、コンピュータープログラム。

【発明の詳細な説明】

【技術分野】

【0001】

[0001] コンピューターシステムおよび関連する技術は、社会の多くの側面に影響を与える。

【背景技術】

【0002】

実際、情報を処理するコンピューターシステムの能力は、我々の生活の仕方および働き方を変化させた。今ではコンピューターシステムは一般に、コンピューターシステムが出現する前までは手で行われていた多くのタスク（例えばワードプロセッシング、スケジュール設定、会計など）を行う。より最近ではコンピューターシステムは互いに、また他の電子デバイスに結合されて、コンピューターシステムおよび他の電子デバイスがそれを通して電子データを伝送することができる、有線および無線コンピューターネットワークの両方を形成する。それに応じて、多くのコンピューティングタスクの実行は、いくつかの異なるコンピューターシステム、および/またはいくつかのコンピューティング環境にわたって分散される。

【0003】

[0002] いくつかのコンピューターシステムは準仮想化実行環境をもたらすように構成され、これはゲストソフトウェアが、単一のコンピューターシステムのハードウェアデバイスを分離された形で共有することを可能にする。一般に準仮想化実行環境は、ハイパーバイザーによってサポートされる複数のパーティションを形成する。パーティションは、異なるゲストソフトウェアの間の分離をもたらす。パーティションは一般に、ルートパーティションと、1つまたは複数の子パーティションを含む。ルートパーティションは、ホストオペレーティングシステムを稼働させ、仮想化スタックを管理する。ルートパーティションは、物理デバイスへのアクセスを取得することができる。各子パーティションは、ゲストソフトウェア（例えばゲストオペレーティングシステム、およびゲストアプリケーション）をホストする。子パーティションには、仮想デバイス、およびハイパーバイザーのソフトウェアインターフェイスを通して、物理デバイスへのアクセスがもたらされる。

【発明の概要】

【発明が解決しようとする課題】

【0004】

[0003] いくつかの準仮想化実行環境は、1つまたは複数の物理グラフィックス処理装置（「GPU」）への準仮想化アクセスを有する、子パーティション（およびその中で実行するゲストソフトウェア）をもたらす。一般に物理GPUへの準仮想化アクセスの各実装形態は、（1つまたは複数の）特定の3次元レンダリングフレームワークをサポートする。したがってゲストソフトウェアが物理GPUの機能にアクセスすることは、それらの機能をサポートしない準仮想化実行環境内でそのゲストソフトウェアが実行している場合、できないことがあり得る。いくつかの場合にはゲストソフトウェアは、潜在的に大幅な性能の犠牲を被りながら、物理GPUへの準仮想化アクセスがサポートされないタスクを行うために、仮想化CPUの使用に依存する場合がある。

【課題を解決するための手段】

【0005】

[0004] 本発明は、準仮想化環境において高性能コンピューティング、およびグラフィックスデバイスインターフェイス（「GDI」）高速化をもたらすための方法、システム、およびコンピュータープログラム製品に及ぶ。

【0006】

[0005] いくつかの実施形態は、準仮想化実行環境の子パーティション内で実行するゲストアプリケーションに、グラフィックス処理装置（「GPU」）高速化コンピューティング機能をもたらすための方法を含む。仮想マシンセッションがインスタンス化される。仮想マシンセッションにおいてハイパーバイザーは、（i）ルートパーティション（物理GPUへのアクセスを有する）と、（ii）子パーティション（ゲストアプリケーションを

実行する)とを形成する。

【0007】

[0006]子パーティション内で実行する仮想化グラフィックス処理装置(「vGPU」)は、ゲストアプリケーションに提供される。vGPUのユーザーモードドライバ(「UMD」)は、ゲストアプリケーションにコンピュータシェーダー(compute shader)デバイスドライバインターフェイス(「DDI」)を提供する。コンピュータシェーダーDDIは、ゲストアプリケーションがvGPUにコンピュータシェーダーコマンドを送ることを可能にする、アプリケーションプログラミングインターフェイス(「API」)をもたらす。コンピュータシェーダーコマンドは、コンピュータシェーダーを用いて物理GPUにおいて汎用グラフィックス処理装置(「GPGPU」)計算を行うために用いられる。ルートパーティション内で実行するレンダーコンポーネントは、vGPUから物理GPU固有コンピュータシェーダーコマンドを受け取り、物理GPUにおける実行のためにコマンドをスケジュールする。

10

【0008】

[0007]さらなる実施形態は、準仮想化実行環境の子パーティション内で実行するゲストアプリケーションに、GPU高速化GDI機能をもたらすための方法を含む。仮想マシンセッションがインスタンス化される。仮想マシンセッションにおいてハイパーバイザーは、(i)物理GPUへのアクセスを有するルートパーティションと、(ii)ゲストアプリケーションを実行する子パーティションとを形成する。

20

【0009】

[0008]子パーティション内で実行するvGPUは、ゲストアプリケーションに提供される。vGPUのカーネルモードドライバ(「KMD」)のAPIは、ゲストオペレーティングシステムが、ゲストアプリケーションによって提出されたGDIレンダリングコマンドを高速化することを可能にする。次いでこれらのコマンドは、vGPUのKMDによって処理される。

【0010】

[0009]ルートパーティション内で実行するレンダーコンポーネントは、vGPUからGDI高速化レンダリングコマンドを受け取る。それに応答してレンダーコンポーネントは、ルートパーティション内のGDI合成デバイス上に、GDI高速化レンダリングコマンドをスケジュールする。GDI合成デバイスは、物理GPUにおいてGDI高速化レンダリングコマンドを実行するように構成される。GDI合成デバイスは、デスクトップによる合成のために共有可能として、GDIコマンドに対応するGDIサーフェスにマークする。

30

【0011】

[0010]この概要は、以下の「発明を実施するための形態」でさらに詳しく述べられる、選ばれた概念を簡略化した形で紹介するために示される。この概要は、特許請求される主題の主要な特徴または本質的な特徴を特定するものではなく、特許請求される主題の範囲を決定するための補助として用いられることを意図するものでもない。

【0012】

[0011]本発明のさらなる特徴および利点については、以下に続く説明で述べられ、一部は説明から明らかとなり、または本発明の実践によって理解することができる。本発明の特徴および利点は、添付の「特許請求の範囲」に具体的に指摘された手段および組み合わせを用いて実現することができ、得ることができる。本発明の上記その他の特徴は、以下の説明および添付の「特許請求の範囲」からより十分に明らかとなり、または本明細書において以下で述べられる本発明の実践によって理解することができる。

40

【0013】

[0012]本発明の上記その他の利点および特徴を得ることができるやり方で説明するために、上記で簡潔に述べられた本発明のより具体的な説明は、添付の図面に示されるそれらの特定の実施形態を参照して述べられる。これらの図面は本発明の典型的な実施形態のみを示し、したがってその範囲を限定すると見なされるべきでないとの理解の下に、添付の

50

図面の使用を通して、本発明についてさらなる具体性および詳細を有して述べられ、説明される。

【図面の簡単な説明】

【 0 0 1 4 】

【図 1 A】[0013]物理グラフィックス処理装置（「GPU」）ハードウェアのコンピュータシェーダー機能への準仮想化アクセスを可能にする、例示のコンピュータアーキテクチャを示す図である。

【図 1 B】[0014]物理GPUハードウェアによる、準仮想化グラフィックスデバイスインターフェイス（「GDI」）高速化を可能にする、例示のコンピュータアーキテクチャを示す図である。

10

【図 1 C】[0015]物理GPUハードウェアによる、コンピュータシェーダー機能および準仮想化GDI高速化への準仮想化アクセスを可能にする、例示のコンピュータアーキテクチャを示す図である。

【図 2 A】[0016]準仮想化実行環境の子パーティション内で実行するゲストアプリケーションに、GPU高速化コンピューティング機能をもたらすための例示の方法のフローチャートである。

【図 2 B】[0017]準仮想化実行環境の子パーティション内で実行するゲストアプリケーションに、GPU高速化GDI機能をもたらすための例示の方法のフローチャートである。

【発明を実施するための形態】

【 0 0 1 5 】

20

[0018]本発明は、準仮想化環境において高性能コンピューティング、およびグラフィックスデバイスインターフェイス（「GDI」）高速化をもたらすための方法、システム、およびコンピュータプログラム製品に及ぶ。

【 0 0 1 6 】

[0019]いくつかの実施形態は、準仮想化実行環境の子パーティション内で実行するゲストアプリケーションに、グラフィックス処理装置（「GPU」）高速化コンピューティング機能をもたらすための方法を含む。仮想マシンセッションがインスタンス化される。仮想マシンセッションにおいてハイパーバイザーは、（i）ルートパーティション（物理GPUへのアクセスを有する）と、（ii）子パーティション（ゲストアプリケーションを実行する）とを形成する。

30

【 0 0 1 7 】

[0020]子パーティション内で実行する仮想化グラフィックス処理装置（「vGPU」）は、ゲストアプリケーションに提供される。vGPUのユーザーモードドライバ（「UMD」）は、ゲストアプリケーションにコンピュータシェーダーデバイスドライバインターフェイス（「DDI」）を提供する。コンピュータシェーダーDDIは、ゲストアプリケーションがvGPUにコンピュータシェーダーコマンドを送ることを可能にする、アプリケーションプログラミングインターフェイス（「API」）をもたらす。コンピュータシェーダーコマンドは、コンピュータシェーダーを用いて物理GPUにおいて汎用グラフィックス処理装置（「GPGPU」）計算を行うために用いられる。ルートパーティション内で実行するレンダーコンポーネントは、vGPUから物理GPU固有コンピュータシェーダーコマンドを受け取り、物理GPUにおける実行のためにコマンドをスケジュールする。

40

【 0 0 1 8 】

[0021]さらなる実施形態は、準仮想化実行環境の子パーティション内で実行するゲストアプリケーションに、GPU高速化GDI機能をもたらすための方法を含む。仮想マシンセッションがインスタンス化される。仮想マシンセッションにおいてハイパーバイザーは、（i）物理GPUへのアクセスを有するルートパーティションと、（ii）ゲストアプリケーションを実行する子パーティションとを形成する。

【 0 0 1 9 】

[0022]子パーティション内で実行するvGPUは、ゲストアプリケーションに提供され

50

る。vGPUのカーネルモードドライバ(「KMD」)のAPIは、ゲストオペレーティングシステムが、ゲストアプリケーションによって提出されたGDIレンダリングコマンドを高速化することを可能にする。次いでこれらのコマンドは、vGPUのKMDによって処理される。

【0020】

[0023]ルートパーティション内で実行するレンダーコンポーネントは、vGPUからGDI高速化レンダリングコマンドを受け取る。それに応答してレンダーコンポーネントは、ルートパーティション内のGDI合成デバイス上に、GDI高速化レンダリングコマンドをスケジュールする。GDI合成デバイスは、物理GPUにおいてGDI高速化レンダリングコマンドを実行するように構成される。GDI合成デバイスは、デスクトップによる合成のために共有可能として、GDIコマンドに対応するGDIサーフェスにマークする。

10

【0021】

[0024]本発明の実施形態は、以下でより詳しく述べられるように、例えば1つまたは複数のプロセッサおよびシステムメモリなどのコンピューターハードウェアを含む、専用または汎用コンピューターを備えるまたは利用することができる。本発明の範囲内の実施形態はまた、コンピューター実行可能命令および/またはデータ構造を担持するまたは記憶するための、物理的およびその他のコンピューター可読媒体を含む。このようなコンピューター可読媒体は、汎用または専用コンピューターシステムによってアクセスすることができる任意の利用可能な媒体とすることができる。コンピューター実行可能命令を記憶するコンピューター可読媒体は、コンピューター記憶媒体(装置)である。コンピューター実行可能命令を担持するコンピューター可読媒体は、伝送媒体である。したがって、例としてかつ非限定的に本発明の実施形態は、少なくとも2つの明瞭に異なる種類のコンピューター可読媒体、すなわちコンピューター記憶媒体(装置)と伝送媒体とを備えることができる。

20

【0022】

[0025]コンピューター記憶媒体(装置)は、RAM、ROM、EEPROM、CD-ROM、ソリッドステートドライブ(「SSD」)(例えばRAMに基づく)、フラッシュメモリ、相変化メモリ(「PCM」)、他のタイプのメモリ、他の光ディスク記憶装置、磁気ディスク記憶装置または他の磁気記憶装置、またはコンピューター実行可能命令またはデータ構造の形での所望のプログラムコード手段を記憶するために用いることができ、汎用または専用コンピューターによってアクセスすることができる任意の他の媒体を含む。

30

【0023】

[0026]「ネットワーク」は、コンピューターシステムおよび/またはモジュールおよび/または他の電子デバイスの間での電子データの移送を可能にする、1つまたは複数のデータリンクとして定義される。情報がネットワークまたは他の通信接続(有線、無線、または有線または無線の組み合わせのいずれか)を通してコンピューターに転送または供給されるときは、コンピューターは接続を適切に伝送媒体と見なす。伝送媒体は、コンピューター実行可能命令またはデータ構造の形での所望のプログラムコード手段を担持するために用いることができ、汎用または専用コンピューターによってアクセスすることができるネットワークおよび/またはデータリンクを含むことができる。上記の組み合わせも、コンピューター可読媒体の範囲に含められるべきである。

40

【0024】

[0027]さらに、様々なコンピューターシステムコンポーネントに到達するとすぐに、コンピューター実行可能命令またはデータ構造の形でのプログラムコード手段は、自動的に伝送媒体からコンピューター記憶媒体(装置)に転送される(またはその逆に)。例えば、ネットワークまたはデータリンクを通して受け取ったコンピューター実行可能命令またはデータ構造は、ネットワークインターフェイスモジュール(例えば「NIC」)内のRAMにバッファすることができ、次いで最終的にコンピューターシステムRAM、および

50

／またはコンピューターシステムにおける揮発性の低いコンピューター記憶媒体（装置）に転送することができる。したがってコンピューター記憶媒体（装置）は、伝送媒体も（さらには主として）利用するコンピューターシステムコンポーネントに含み得ることが理解されるべきである。

【 0 0 2 5 】

[0028]コンピューター実行可能命令は例えば、プロセッサで実行されたときに汎用コンピューター、専用コンピューター、または専用処理デバイスに、特定の機能または一群の機能を行わせる命令およびデータを備える。コンピューター実行可能命令は、例えば2進値、アセンブリ言語などの中間フォーマット命令、さらにはソースコードとすることができる。本主題については構造的特徴および／または方法論的動作に固有の用語で述べてきたが、添付の「特許請求の範囲」において定義される本主題は、必ずしも述べられる特徴または上述の動作に限定されないことが理解されるべきである。むしろ述べられる特徴および動作は、諸請求項を実施する例示の形として開示される。

【 0 0 2 6 】

[0029]当業者には、本発明はパーソナルコンピューター、デスクトップコンピューター、ラップトップコンピューター、メッセージプロセッサ、ハンドヘルドデバイス、マルチプロセッサシステム、マイクロプロセッサベースまたはプログラム可能な民生用電子機器、ネットワークPC、ミニコンピューター、メインフレームコンピューター、携帯電話、PDA、タブレット、ポケットベル、ルータ、スイッチなどを含む、多くのタイプのコンピューターシステム構成を有するネットワークコンピューティング環境において実施できることが理解されるであろう。本発明はまた、ネットワークを通してリンクされた（有線データリンク、無線データリンクのいずれかにより、または有線および無線データリンクの組み合わせにより）、ローカルおよびリモートコンピューターシステムの両方がタスクを実行する、分散システム環境において実施することができる。分散システム環境ではプログラムモジュールは、ローカルおよびリモートメモリ記憶装置の両方に配置することができる。

【 0 0 2 7 】

[0030]準仮想化コンピュートシェーダー（GPGPU）機能

図1Aは、物理GPUハードウェアのコンピュートシェーダー機能への準仮想化アクセスを可能にする、例示のコンピューターアーキテクチャ100aを示す。図1Aを参照するとコンピューターアーキテクチャ100aは、物理ハードウェア101を含む。物理ハードウェア101は、1つまたは複数の汎用プロセッサ、システムメモリなどの適切なハードウェアデバイスを含むことができる。図示のように物理ハードウェアは、少なくとも1つの物理GPU101aを含む。

【 0 0 2 8 】

[0031]物理GPU101aは、グラフィックスレンダリングなどの並列処理タスクを行うように構成された処理デバイスである。物理GPU101aは、物理GPU101aが汎用（すなわち非グラフィックスレンダリング）計算を行うことを可能にする、コンピュートシェーダーを実行するためのサポートを含む。言い換えれば物理GPU101aは、コンピュートシェーダーデバイス上のGPGPU計算をサポートする。

【 0 0 2 9 】

[0032]コンピューターアーキテクチャ100aはまた、ハイパーバイザー102を含む。ハイパーバイザー102は、物理ハードウェア101の上で実行し、仮想化プラットフォームをサポートする。仮想化プラットフォームは、複数のパーティションを形成する。各パーティションは、ゲストソフトウェアをその中で実行することができる、分離の論理ユニットをもたらす。例えばコンピューターアーキテクチャ100aは、ルートパーティション103と、子パーティション104とを含む。

【 0 0 3 0 】

[0033]ルートパーティション103は、ホストオペレーティングシステムを実行し、物理ハードウェア101への直接アクセスを有する（物理ハードウェア101の上に表され

10

20

30

40

50

たルートパーティション 103 によって示されるように)。各子パーティションは、ゲストソフトウェア (例えばオペレーティングシステムおよび/またはアプリケーション) を実行するための実行環境をもたらし、準仮想化のやり方で間接的に物理ハードウェア 101 にアクセスすることができる。すなわちハイパーバイザー 102 を通して各子パーティションは、1 つまたは複数のソフトウェアインターフェイス (例えば仮想化ハードウェア) をゲストソフトウェアにもたらし、一方、ゲストソフトウェアは、物理ハードウェア 101 にアクセスするために、(1 つまたは複数の) ソフトウェアインターフェイスを用いる。ハイパーバイザー 102 は、複数の子パーティションのためのサポートをもたらしことができる。

【0031】

10

[0034] 図示のようにゲストソフトウェア 105 は、子パーティション 104 内で実行する。ゲストソフトウェア 105 は、オペレーティングシステム、および/またはオペレーティングシステム内で実行するアプリケーションプログラムなどの任意の適切なゲストソフトウェアを備える。ゲストソフトウェア 105 は、グラフィックスランタイム 105a を含むまたは使用する。グラフィックスランタイム 105a は、グラフィックスをレンダリングするおよび/または GPGPU 計算を行うための、フレームワーク (例えば API) をもたらし。

【0032】

[0035] 子パーティション 104 は、vGPU 106 へのアクセスをゲストソフトウェア 105 にもたらし、vGPU 105 は物理 GPU 101a を仮想化し、ゲストソフトウェア 105 が物理 GPU 101a に間接的にアクセスすることを可能にする。したがって vGPU 106 は、物理 GPU 101a の任意の対応する機能と共に、少なくとも 1 つのレンダリングフレームワーク (グラフィックスランタイム 105a に対応する) の機能のすべてまたはサブセットを、ゲストソフトウェア 105 に公開するように構成される。

20

【0033】

[0036] 具体的には vGPU 106 は、物理 GPU 101a において GPGPU 計算を行うために物理 GPU 101a のコンピュータシェーダー機能にアクセスするために、ゲストソフトウェア 105 が vGPU 106 を呼び出すことを可能にする、1 つまたは複数のソフトウェアインターフェイスを公開するように構成される。一方、vGPU 106 は、ルートパーティション 103 でのレンダーコンポーネントと一緒に働いて、物理 GPU 101a 上でコンピュータシェーダー機能を行う。図示のように例えばルートパーティション 103 は、レンダーコンポーネント 112 を含む。一方、レンダーコンポーネント 112 は、コンピュータシェーダーコマンドおよびデータを取り扱うためのコンピュータシェーダーコンポーネント 113 を含む。vGPU 106 は、物理 GPU 101a 上でレンダリングを行うために、コンピュータシェーダーコマンドおよびデータ 117 をレンダーコンポーネント 112 にリモート接続する。

30

【0034】

[0037] レンダーコンポーネント 112 は、vGPU 106 から受け取ったグラフィックスコマンドを、物理 GPU 101a 上での実行のためにスケジュールする。レンダーコンポーネント 112 はまた、これらのコマンドを実行するための適切なコンテキストを生成する。したがってレンダーコンポーネント 112 は、子パーティション 104 内の vGPU 106 から受け取ったコンピュータシェーダー関連コマンドの物理 GPU 101a 上での実行を、コンピュータシェーダーコンポーネント 113 を用いてスケジュールするように構成される。

40

【0035】

[0038] 図示のように vGPU 106 は、子パーティション 104 のユーザーモードにおいて実行するユーザーモードドライバ 106a と、子パーティション 104 のカーネルモードにおいて実行するカーネルモードドライバ 109 とを含む。ユーザーモードドライバ 106a は、コンピュータシェーダー DDI 107 として示されるコンピュータシェーダー (GPGPU) 機能に関連する DDI を含む、少なくとも 1 つのレンダリングフレーム

50

ワークのデバイスドライバインターフェイス(「DDI」)を公開する。コンピュータシェーダーDDI 107は、ゲストソフトウェア105が、GP GPU計算を行うためにv GPU 106に呼び出しを行うことを可能にする。

【0036】

[0039]いくつかの実施形態ではユーザーモードドライバ106bは、コンピュータシェーダー機能(例えばMicrosoft(登録商標)CorporationからのDirectX(登録商標)バージョン10および/または11)をサポートするレンダリングフレームワークのDDIを公開する。例えば実施形態では、ユーザーモードドライバ106aがDirectX(登録商標)バージョン10および11のコンピュータシェーダー機能を公開するときは、ユーザーモードドライバ106aは、コンピュータシェーダーDDI 107の一部として以下のDDIの1つまたは複数を公開することができる:

```
PFND3D11DDI__SETSHADER__WITH__IFACES    pfnCs
SetShaderWithIfaces
```

```
PFND3D10DDI__SETSHADER    pfnCsSetShader
```

```
PFND3D10DDI__SETSHADERRESOURCES    pfnCsSet
ShaderResources
```

```
PFND3D10DDI__SETSAMPLERS    pfnCsSetSampler
s
```

```
PFND3D10DDI__SETCONSTANTBUFFERS    pfnCsSet
ConstantBuffers
```

```
PFND3D11DDI__SETUNORDEREDACCESSVIEWS    pfn
CsSetUnorderedAccessViews
```

いくつかの実施形態ではユーザーモードドライバ106bは、ドメインシェーダー、ハルシェーダー、またはジオメトリシェーダーの1つまたは複数に関連するDDIなどの、コンピュータシェーダー機能(例えばDirectX(登録商標)バージョン11)をサポートするレンダリングフレームワークのすべてのDDIを提供する。しかしユーザーモードドライバ106bは、コンピュータシェーダー機能をサポートする任意のレンダリングフレームワークから任意のDDIの組を公開することができる。

【0037】

[0040]いくつかの実施形態ではv GPU 106はまた、子パーティション104のユーザーモードにおいて実行するレガシユーザーモードドライバ106bを含むことができる。レガシユーザーモードドライバ106bは、1つまたは複数のレンダリングフレームワークのレガシバージョンのDDIを公開することができる。例えばレガシユーザーモードドライバ106bは、DirectX(登録商標)のレガシバージョン(例えばDirectX(登録商標)バージョン9)、または任意の他のレンダリングフレームワーク(例えばSilicon Graphics, Inc.からのOpenGL(登録商標))のレガシバージョンをサポートすることができる。

【0038】

[0041]一般にユーザーモードドライバ106aは、ハードウェアコンテキストおよびコマンドバッファを構築するように構成される。具体的にはユーザーモードドライバ106aは、ゲストソフトウェア105(またはゲストソフトウェア105のグラフィックスランタイム105a)によって発行されたグラフィックスコマンドを、ハードウェア固有コマンドに変換する。例えばユーザーモードドライバ106aは、ゲストソフトウェア105からコンピュータシェーダーを用いたGP GPU計算に関連するグラフィックスコマンド115を受け取ることができる。ユーザーモードドライバ106aは、グラフィックスコマンド115をハードウェア固有コマンド(すなわち物理GPU 101aに固有のコマンド)に変換するように構成される。変換の一部としてユーザーモードドライバ106aは、物理GPU 101aのための適切なハードウェアコンテキストを維持する。例えばユーザーモードドライバ106aは、グラフィックスパイプラインに影響する設定のための論理値を、値および対応する物理設定に変換する。ユーザーモードドライバ106aはま

10

20

30

40

50

た、変換されたハードウェア固有コマンドをコマンドバッファ 115 に記憶し、コマンドバッファ 116 をカーネルモードドライバ 109 に送るように構成される。

【0039】

[0042]さらに vGPU 106 は、コンピュータシェーダーコンポーネント 110 を含む、子パーティション 104 のカーネルモードにおいて実行するカーネルモードドライバ 109 を含む。カーネルモードドライバ 109 は、コマンドバッファ（例えばコマンドバッファ 116）を受け取り、対応するダイレクトメモリアクセス（「DMA」）バッファを構築するように構成される。DMA バッファが処理されるべき時点では、GPU スケジューラはカーネルモードドライバ 109 を呼び出す。次いでカーネルモードドライバ 109 は、DMA バッファの物理 GPU 101a への実際の提出の詳細を取り扱う。

10

【0040】

[0043]図 2A は、準仮想化実行環境の子パーティション内で実行するゲストアプリケーションに、GPU 高速化コンピューティング機能をもたらすための例示の方法 200a のフローチャートを示す。方法 200a については、コンピュータアーキテクチャ 100a のコンポーネントおよびデータに関連して述べられる。

【0041】

[0044]方法 200a は、(i) 物理 GPU へのアクセスを有するルートパーティションと、(ii) ゲストアプリケーションを実行する子パーティションとを形成する、ハイパーバイザーをインスタンス化するステップを含む、仮想マシンセッションをインスタンス化する動作を含む（動作 201）。例えばコンピューティング環境 100a は、仮想マシンセッションをインスタンス化するステップの一部としてハイパーバイザー 102 をインスタンス化することができる。ハイパーバイザー 102 は、ルートパーティション 103 と、子パーティション 104 とを形成する。ルートパーティション 103 は、物理 GPU 101a へのアクセスを有する。子パーティション 104 は、ゲストアプリケーション 105 を含む 1 つまたは複数のゲストアプリケーションを実行し、物理 GPU 101a への間接アクセスを有する。

20

【0042】

[0045]方法 200a はまた、vGPU をゲストアプリケーションに提供する動作であって、vGPU は子パーティション内で実行し、vGPU の UMD の一部として複数のコンピュータシェーダー DDI をゲストアプリケーションに提供するステップを含み、複数のコンピュータシェーダー DDI は、コンピュータシェーダーを用いて物理 GPU において GPGPU 計算を行うために、ゲストアプリケーションがコンピュータシェーダーコマンドを vGPU に送ることを可能にする API をもたらす、動作を含む（動作 202）。例えば子パーティション 104 は、vGPU 106 をゲストソフトウェア 105 に提供することができる。vGPU 106 はユーザーモードドライバ 106a を含む。ユーザーモードドライバ 106a はコンピュータシェーダー DDI 107 を提供し、これはゲストソフトウェア 105 が、物理 GPU 101a においてコンピュータシェーダー GPGPU 計算を行うために vGPU 106 を呼び出すことを可能にする。例えばゲストソフトウェア 105（またはグラフィックスランタイム 105a）は、物理 GPU 101a において GPGPU 計算を行うために、グラフィックスコマンド 115 を vGPU 106 に送ることができる。

30

40

【0043】

[0046]ユーザーモードドライバ 106a は、受け取ったグラフィックスコマンド 115 を、物理ハードウェア固有コマンドに変換する（例えばコマンドバッファ 116 の一部として）。次いで vGPU 106 は、コンピュータシェーダーコマンドおよびデータ 117（物理ハードウェア固有コマンドを含む）をレンダーコンポーネント 112 にリモート接続する。

【0044】

[0047]方法 200a はまた、ルートパーティション内で実行するレンダーコンポーネントが、vGPU から物理 GPU 固有コンピュータシェーダーコマンドを受け取る動作を含

50

む（動作203）。例えばルートパーティション103内で実行するレンダーコンポーネント112は、vGPU106から、物理GPU固有コンピュートシェーダーコマンドを含むコンピュートシェーダーコマンドおよびデータ117を受け取ることができる。

【0045】

[0048]方法200aはまた、レンダーコンポーネントが、物理GPUにおける実行のために物理GPU固有コンピュートシェーダーコマンドをスケジュールする動作を含む（動作204）。例えばコンピュートシェーダーコンポーネント113は、物理GPU101aにおけるコンピュートシェーダーデバイス上での実行のために、受け取った物理GPU固有コンピュートシェーダーコマンドをスケジュールすることができる。そうすることでコンピュートシェーダーコンポーネント113は、物理GPU固有コンピュートシェーダーコマンドの実行のための適切なコンテキストを構成し、維持することができる。

10

【0046】

[0049]準仮想化GDI高速化

ゲストソフトウェア準仮想化コンピュートシェーダーに、物理GPU101aへのアクセス（すなわちGPGPU計算能力）をもたらすことに加えて、実施形態は、物理GPUにおいてゲストソフトウェア準仮想化GDI高速化をもたらすことに及ぶ。図1Bは、物理GPUハードウェアによる準仮想化GDI高速化を可能にする代替のコンピューターアーキテクチャ100bを示す。したがってコンピューターアーキテクチャ100bは、物理または仮想中央処理装置を用いてGDIコマンドを取り扱うのとは対照的に、ゲストソフトウェアが物理GPUを用いたGDIコマンド高速化を要求することを可能にする。

20

【0047】

[0050]子パーティション104'内でコンピューターアーキテクチャ100bは、物理GPU101a'上でのGDIコマンドの高速化されたレンダリングを、これらのコマンドがゲストソフトウェア105'によって発行されたときに可能にするコンポーネントをもたらす。図示のように、例えば子パーティション104'はGDIインターフェイス108を含み、これはグラフィックスランタイム105a'（例えばゲストオペレーティングシステムの一部であるグラフィックスランタイム）と通信するように構成される。さらにvGPU106'は、カーネルモードドライバ109'のGDIコンポーネント111においてGDIインターフェイス108から受け取ったGDI高速化コマンド120を処理するように構成される。

30

【0048】

[0051]GDIインターフェイス108は、1つまたは複数のGDIコマンドインターフェイスをグラフィックスランタイム105a'に公開し、グラフィックスランタイム105a'から受け取ったGDI高速化コマンドをvGPUのカーネルモードドライバ109'に転送するように構成される。したがってゲストソフトウェア105'の要求により、物理GPU101a'においてGDIコマンド実行を高速化するために、グラフィックスランタイム105a'がGDI高速化グラフィックスコマンド120をGDIインターフェイス108を通じてvGPU106'のカーネルモードドライバ109'に送ることが可能になる。

【0049】

40

[0052]カーネルモードドライバ109'のGDIコンポーネント111は、対応する（1つまたは複数の）ハードウェアレンダリング動作と共に、1つまたは複数のGDIコマンドインターフェイスを実装するように構成される。いくつかの実施形態ではvGPU106'は、DirectX（登録商標）の特定のバージョン（例えばDirectX（登録商標）バージョン10および/または11）に対応するGDIコマンドを実装し、公開する。例えばGDIコンポーネント111は、対応するハードウェアレンダリング動作と共に、「DxgkDdiRenderKm」インターフェイスを公開および実装することができる。GDIコンポーネント111は、任意の適切なGDI動作を実装することができるが、いくつかの実施形態ではGDIコンポーネント111は以下の動作を実装する。

【0050】

50

```

typedef struct __DXGK__RENDERKM__COMMAND
{
    DXGK__RENDERKM__OPERATION Opcode;
    UINT CommandSize;
    union
    {
        DXGK__GDIARG__BITBLT BitBlit;
        DXGK__GDIARG__COLORFILL ColorFill;
        DXGK__GDIARG__ALPHABLEND AlphaBlend;
        DXGK__GDIARG__STRETCHBLT StretchBlit;
        DXGK__GDIARG__TRANSPARENTBLT TransparentBlit;
        DXGK__GDIARG__CLEARTYPEBLEND ClearTypeBlend;
    } Command;
} DXGK__RENDERKM__COMMAND;

```

[0053]カーネルモードドライバ109'は、GDIコマンドおよびデータ118をルートパーティション103'内のレンダーコンポーネント112'に送るように構成される。GDIコマンドおよびデータ118は、GDI高速化コマンドを実行するための、3Dレンダリングデバイスおよびコンテキストに関する情報を含む。レンダーコンポーネント112'はGDIコンポーネント114を含み、これはGDIコマンドおよびデータ118を受け取り、受け取ったGDI高速化コマンドを物理GPU101a'において実行するように構成される。

【0051】

[0054]具体的にはGDIコンポーネント114は、GDIサーフェス、および対応する3Dレンダリング（または合成）デバイス（例えばD3Dデバイス）を生成するように構成される。3Dレンダリングデバイスは、GDI高速化コマンドを実行するための適切なコンテキストをもたらすように構成される。GDIサーフェスは、以下のGDIサーフェスタイプのいずれかなどの、任意の適切なGDIサーフェスを含むことができる：

```

D3DKMDT__GDISURFACE__TEXTURE
D3DKMDT__GDISURFACE__STAGING__CPUVISIBLE
D3DKMDT__GDISURFACE__STAGING
D3DKMDT__GDISURFACE__LOOKUPTABLE
D3DKMDT__GDISURFACE__EXISTINGSYSTEMMEM
D3DKMDT__GDISURFACE__TEXTURE__CPUVISIBLE

```

GDIコンポーネント114は、生成されたGDIサーフェスを共有可能サーフェスとしてマークする。これによりデスクトップへの合成に成功できるようになる。

【0052】

[0055]図2Bは、準仮想化実行環境の子パーティション内で実行するゲストアプリケーションに、GPU高速化GDI機能をもたらすための例示の方法200bのフローチャートを示す。方法200bについては、コンピューターアーキテクチャ100bのコンポーネントおよびデータに関連して述べられる。

【0053】

[0056]方法200bは、(i)物理GPUへのアクセスを有するルートパーティションと、(ii)ゲストアプリケーションを実行する子パーティションとを形成する、ハイパーバイザーをインスタンス化するステップを含む、仮想マシンセッションをインスタンス化する動作を含む（動作205）。例えばコンピューティング環境100bは、仮想マシンセッションをインスタンス化するステップの一部としてハイパーバイザー102'をインスタンス化することができる。ハイパーバイザー102'は、ルートパーティション103'と、子パーティション104'とを形成する。ルートパーティション103'は、

物理GPU101a'へのアクセスを有する。子パーティション104'は、ゲストアプリケーション105'を含む1つまたは複数のゲストアプリケーションを実行し、物理GPU101a'への間接アクセスを有する。

【0054】

[0057]方法200bはまた、vGPUをゲストアプリケーションに提供する動作であって、vGPUは子パーティション内で実行し、vGPUのKMDによる処理のために、ゲストオペレーティングシステムが、ゲストアプリケーションによって用いられるGDIレンダリングコマンドを高速化することを可能にするvGPUのKMDのAPIを、vGPUに提供するステップを含む、動作を含む(動作206)。例えば子パーティション104'は、vGPU106'およびGDIインターフェイス108をゲストソフトウェア105'に提供する。GDIインターフェイス108は、カーネルモードドライバ109'による処理のために、および物理GPU101a'による高速化のために、vGPU106'に対して、グラフィックスランタイム105a'(例えばオペレーティングシステムのグラフィックスランタイム)がゲストソフトウェア105'によって用いられるGDIレンダリングコマンドを高速化することを可能にする。例えばグラフィックスランタイム105a'は、GDI高速化コマンドの実行のために、GDIインターフェイス108を通して、グラフィックスコマンド120をカーネルモードドライバ109'に送ることができる。vGPU106'は、受け取ったGDIコマンドを処理し、GDIコマンドおよびデータ118をルートパーティション103'内のレンダーコンポーネント112'に送るために、カーネルモードドライバ109'のGDIコンポーネント111を用いる。

【0055】

[0058]方法200bはまた、ルートパーティション内で実行レンダーコンポーネントが、vGPUからGDI高速化レンダリングコマンドを受け取る動作を含む(動作207)。例えばルートパーティション103'内で実行するレンダーコンポーネント112'は、vGPU106'からGDIコマンドおよびデータ118を受け取ることができる。

【0056】

[0059]方法200bはまた、レンダーコンポーネントが、ルートパーティション内のGDI合成デバイス上にGDI高速化レンダリングコマンドをスケジュールする動作を含み、GDI合成デバイスは、物理GPUにおいて少なくとも1つのGDI高速化レンダリングコマンドを実行するように構成され、GDI合成デバイスはまた、デスクトップによる合成のために、少なくとも1つのGDI高速化レンダリングコマンドに対応するGDIサーフェスを共有可能としてマークするように構成される(動作208)。例えばGDIコンポーネント114は、3Dレンダリング(合成)デバイス、およびGDI高速化コマンドを実行するためのGDIサーフェスを生成することができ、3Dレンダリング(合成)デバイス上の実行のためにGDI高速化コマンドをスケジュールすることができる。GDIサーフェスは、デスクトップにおける合成のために共有可能としてマークすることができる。

【0057】

[0060]準仮想化コンピュータシェーダー(GPGPU)機能および準仮想化GDI高速化

いくつかの実施形態では単一のコンピュータアーキテクチャが、コンピュータシェーダー(GPGPU)機能および準仮想化GDI高速化の両方をもたらす。図1Cは、コンピュータシェーダー機能への準仮想化アクセス、および物理GPUハードウェアによる準仮想化GDI高速化を可能にする例示のコンピュータアーキテクチャ100cを示す。例えばコンピュータアーキテクチャ100cは、GDIインターフェイス108と、vGPU106"とを含む。vGPU106"は、コンピュータシェーダーDDI107を含むユーザーモードドライバ106a"を含んでいる。vGPU106"はまた、コンピュータシェーダーコンポーネント110aおよびGDIコンポーネント111の両方を含む、カーネルモードドライバ109"を含んでいる。

【0058】

10

20

30

40

50

【 0 0 5 9 】

〔0062〕本発明は、その趣旨または本質的な特徴から逸脱せずに、他の特定の形で実施することができる。述べられた実施形態は、すべての点で例示のみであり、限定的ではないと見なされるべきである。したがって本発明の範囲は上記の説明によってではなく、添付の「特許請求の範囲」によって示される。特許請求の範囲と等価な意味および範囲内に含まれるすべての変更は、それらの範囲に包含されるものである。

[illegible][illegible]

Figure 1B

【 ㄨ 1 C 】

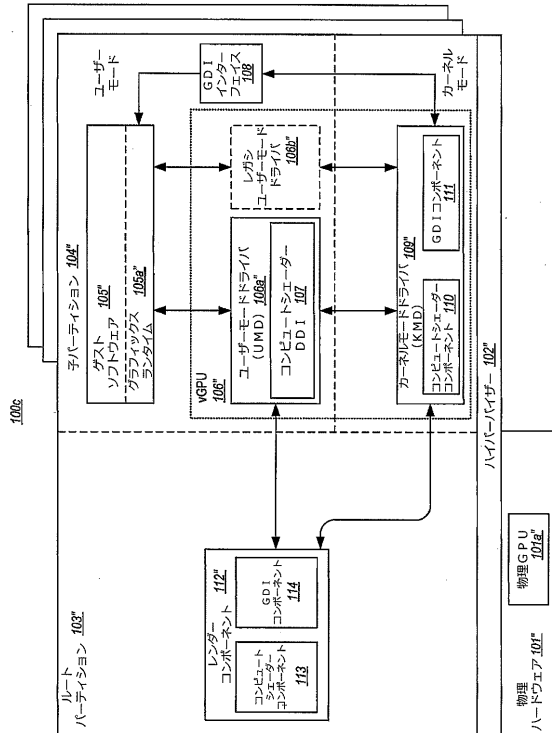


Figure 1C

【 図 2 A 】

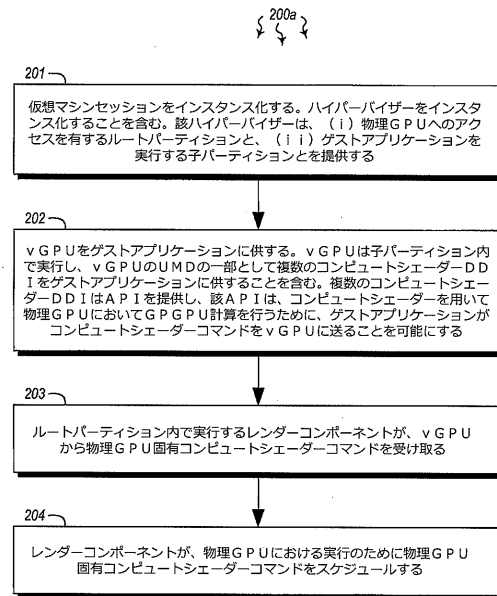


Figure 2A

【 ㄨ 2 B 】

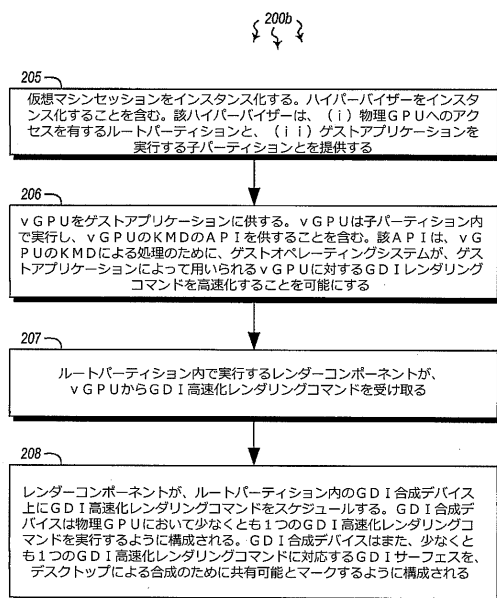


Figure 2B

フロントページの続き

- (72)発明者 マラカパッリ, メハー・プラサード
アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, レッドモンド, ワン・マイクロソフト・ウェイ, マイクロソフト コーポレーション, エルシーエイ - インターナショナル・パテンツ
- (72)発明者 ジャーン, ハオ
アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, レッドモンド, ワン・マイクロソフト・ウェイ, マイクロソフト コーポレーション, エルシーエイ - インターナショナル・パテンツ
- (72)発明者 タン, リン
アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, レッドモンド, ワン・マイクロソフト・ウェイ, マイクロソフト コーポレーション, エルシーエイ - インターナショナル・パテンツ

審査官 大桃 由紀雄

- (56)参考文献 米国特許出願公開第 2 0 1 1 / 0 1 0 2 4 4 3 (U S , A 1)
特開 2 0 1 0 - 0 2 0 7 5 5 (J P , A)

- (58)調査した分野(Int.Cl. , D B 名)
- | | |
|---------|-----------|
| G 0 6 F | 9 / 4 8 |
| G 0 6 F | 9 / 4 6 |
| G 0 6 T | 1 5 / 0 0 |