

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6329484号
(P6329484)

(45) 発行日 平成30年5月23日 (2018. 5. 23)

(24) 登録日 平成30年4月27日 (2018. 4. 27)

(51) Int. Cl.

F I

G 0 6 F 9/50 (2006.01)

G 0 6 F 9/46 4 6 5 C

請求項の数 10 (全 34 頁)

(21) 出願番号	特願2014-535969 (P2014-535969)	(73) 特許権者	314015767
(86) (22) 出願日	平成24年10月15日 (2012. 10. 15)		マイクロソフト テクノロジー ライセン
(65) 公表番号	特表2014-530441 (P2014-530441A)		シング, エルエルシー
(43) 公表日	平成26年11月17日 (2014. 11. 17)		アメリカ合衆国 ワシントン州 9805
(86) 国際出願番号	PCT/US2012/060169		2 レッドモンド ワン マイクロソフト
(87) 国際公開番号	W02013/056204		ウェイ
(87) 国際公開日	平成25年4月18日 (2013. 4. 18)	(74) 代理人	100140109
審査請求日	平成27年10月15日 (2015. 10. 15)		弁理士 小野 新次郎
(31) 優先権主張番号	13/274, 041	(74) 代理人	100075270
(32) 優先日	平成23年10月14日 (2011. 10. 14)		弁理士 小林 泰
(33) 優先権主張国	米国 (US)	(74) 代理人	100101373
			弁理士 竹内 茂雄
		(74) 代理人	100118902
			弁理士 山本 修

最終頁に続く

(54) 【発明の名称】 多数のサーバーからクライアントへの単一エンド・ユーザー体験の配信

(57) 【特許請求の範囲】

【請求項 1】

データーを処理するためのシステムであって、
少なくとも1つの計算デバイスを備え、
前記計算デバイスは、

クライアント・デバイスとの第1の接続および少なくとも1つのグラフィクス・デバイスとの第2の接続を確立するステップであって、前記第1の接続は、前記クライアント・デバイスに割り当てられたリモート・セッションに前記クライアント・デバイスを接続する、ステップと、

前記クライアント・デバイスが前記少なくとも1つのグラフィクス・デバイスへの第3の接続を確立するのを可能にするために、前記少なくとも1つのグラフィクス・デバイスを示す情報を前記クライアント・デバイスに提供するステップであって、前記第3の接続は、前記第1の接続を介して前記クライアント・デバイスから受信されたデーターに回答して、前記クライアント・デバイスへエンコードされたグラフィクス・データーを伝送するように適合される、ステップと、

前記第1の接続を介して前記クライアント・デバイスからデーターを受信するステップと、

前記クライアント・デバイスから受信された前記データーを、前記リモート・セッションのランタイム環境内で実行されるアプリケーションによって処理することによって、表示コマンドを発行するステップと、

10

20

前記受信されたデーターに応答して前記アプリケーションによって発行された前記表示コマンドを、前記第2の接続を介して前記少なくとも1つのグラフィクス・デバイスへ送信するステップであって、前記表示コマンドは、前記少なくとも1つのグラフィクス・デバイスによって前記第3の接続を介して前記クライアント・デバイスへ送信されるグラフィクス・データーを生成するのに用いられる、ステップと、

を少なくとも行うように構成される、

システム。

【請求項2】

前記少なくとも1つの計算デバイスは、少なくとも1つの仮想機械内にインスタンス化される、請求項1に記載のシステム。

10

【請求項3】

前記少なくとも1つのグラフィクス・デバイスは、少なくとも1つの仮想化機械上の少なくとも1つのパーティション内にインスタンス化される、請求項1に記載のシステム。

【請求項4】

前記クライアント・デバイスと前記少なくとも1つのグラフィクス・デバイスとの前記第3の接続を確立するステップは、前記少なくとも1つの計算デバイスと前記少なくとも1つのグラフィクス・デバイスの負荷均衡を行うステップを含む、請求項1に記載のシステム。

【請求項5】

前記少なくとも1つのグラフィクス・デバイスのうちの1または複数のグラフィクス・デバイスは、多数のクライアント・デバイスに対応するグラフィクス・データーを1つのクライアント・デバイスへ送信するように構成される、請求項1に記載のシステム。

20

【請求項6】

グラフィクス・データーを処理するための方法であって、

複数の計算デバイスと複数のクライアント・デバイスとの間に第1の接続を確立し、前記複数の計算デバイスと複数のグラフィクス・デバイスとの間に第2の接続を確立するステップと、

前記複数の計算デバイスに、前記複数のグラフィクス・デバイスのうちの対応する1つとの第3の接続を確立するよう命令するステップであって、前記第3の接続は、前記第1の接続を介して前記複数のクライアント・デバイスから受信されたデーターに
30
前記複数のクライアント・デバイスへエンコードされたグラフィクス・データーを伝送するように適合される、ステップと、

前記複数の計算デバイスによって、前記第1の接続を介して前記複数のクライアント・デバイスからデーターを受信するステップと、

前記受信されたデーターを処理することによって、表示コマンドを発行するステップと、

前記第2の接続を介して前記複数のグラフィクス・デバイスへ前記表示コマンドを送信するステップであって、前記表示コマンドは、前記受信されたデーターに
40
アプリケーションによって発行され、前記アプリケーションは、前記複数のクライアント・デバイスのうちのそれぞれのクライアント・デバイスに対して割り当てられたリモート・セッションのランタイム環境内で実行される、ステップと、

前記表示コマンドを用いてグラフィクス・データーを生成し、前記生成されたグラフィクス・データーを前記第3の接続を介して前記複数のクライアント・デバイスへ送信するように、前記複数のグラフィクス・デバイスに命令するステップと、

を含む方法。

【請求項7】

前記複数の計算デバイスは、複数の仮想機械内にインスタンス化され、前記複数のグラフィクス・デバイスは、前記複数の計算デバイスと同じ複数の物理計算デバイス上の複数のパーティション内にインスタンス化される、請求項6に記載の方法。

【請求項8】

50

前記複数の計算デバイスは、前記複数のクライアント・デバイスとの複数のリモート・デスクトップ・セッションを実行しており、前記グラフィクス・データーは、ユーザーのリモート・デスクトップ・セッションの少なくとも一部のグラフィクス表現を含む、請求項 7 に記載の方法。

【請求項 9】

前記複数の計算デバイスと前記複数のクライアント・デバイスとの間に前記第 1 の接続を確立し、前記複数の計算デバイスと前記複数のグラフィクス・デバイスとの間に前記第 2 の接続を確立するステップは、複数のクライアント・デバイス・アドレスを取得して前記取得された複数のクライアント・デバイス・アドレスを前記複数のグラフィクス・デバイスに供給するステップと、複数のグラフィクス・デバイス・アドレスを取得して前記取得された複数のグラフィクス・デバイス・アドレスを前記複数のクライアント・デバイスに供給するステップと、前記複数のクライアント・デバイスおよび前記複数のグラフィクス・デバイスに、前記供給されたアドレスを用いて第 3 の接続を確立するよう命令するステップと、を含む、請求項 6 に記載の方法。

【請求項 10】

命令が格納されたコンピューター読み取り可能記憶デバイスであって、前記命令は、1 または複数のプロセッサによって実行されると、前記 1 または複数のプロセッサに、複数の計算デバイスと複数のリモート・クライアント・コンピューターとの間に第 1 の接続を確立し、前記複数の計算デバイスと複数のグラフィクス・デバイスとの間に第 2 の接続を確立するステップと、

前記複数のリモート・クライアント・コンピューターのそれぞれが前記複数のグラフィクス・デバイスのうちの対応する 1 つとの同時の第 3 の接続を確立することを可能にする情報を前記複数のリモート・クライアント・コンピューターへ送信するステップであって、前記同時の第 3 の接続のそれぞれは、前記第 1 の接続を介して前記複数のリモート・クライアント・コンピューターから受信されたデーターに応答して、前記複数のリモート・クライアント・コンピューターへエンコードされたグラフィクス・データーを伝送するように適合される、ステップと、

前記第 1 の接続を介して前記複数のリモート・クライアント・コンピューターによるデーターを受信するステップと、

前記受信されたデーターを処理することによって、表示コマンドを発行するステップと

、
前記第 2 の接続を介して前記複数のグラフィクス・デバイスへ前記表示コマンドを送信するステップであって、前記表示コマンドは、前記受信されたデーターに応答してアプリケーションによって発行され、前記アプリケーションは、前記複数のリモート・クライアント・コンピューターのうちのそれぞれのリモート・クライアント・コンピューターに対して割り当てられたリモート・セッションのランタイム環境内で実行される、ステップと

、
前記表示コマンドを用いてグラフィクス・データーを生成し、前記生成されたグラフィクス・データーを、前記複数のリモート・クライアント・コンピューターのそれぞれによって確立された前記同時の第 3 の接続を介して前記複数のリモート・クライアント・コンピューターへ送信するように、前記複数のグラフィクス・デバイスに命令するステップと

、
を実施させる、コンピューター読み取り可能記憶デバイス。

【発明の詳細な説明】

【背景技術】

【0001】

[0001] リモート計算システムでは、ユーザーが、このリモート計算システムによってホストされるリソースにアクセスすることができる場合がある。このリモート計算システム上のサーバーは、プログラムを実行し、ユーザー・インターフェースを示す信号をクライアントに送信することができる。クライアントは、TCP/IP、UDP、または他の

プロトコルというような通信プロトコルに準拠したネットワークを通じて、信号を送ることによって接続することができる。接続する各クライアントには、仮想デスクトップまたはセッション、即ち、1組のリソースを含む実行環境を提供されるのでもよい。各クライアントは、ユーザー入力を示す信号をサーバーに送信することができ、サーバーはユーザー入力をしかるべきセッションに適用することができる。クライアントは、サーバー・リソースに接続するためには、リモート・デスクトップ・プロトコル(RDP)のようなプロトコルを用いてもよい。

【0002】

[0002] ネットワーク帯域幅の可用性拡大、ならびに豊富な2Dおよび3Dクライアント・グラフィクス・アプリケーションに対する要望増大のため、リモート計算システム・アーキテクチャーにおいてシフトが生じつつある。純粋にローカルな計算パワーだけに頼る代わりに、サーバーがグラフィクス仮想化プラットフォームを組み込み、これらのプラットフォームが、グラフィクス処理インテリジェンスを、データ・センタに配備され(deploy)ホストされた仮想デスクトップ・インフラストラクチャ(VDI)にシフトする。クライアントは、共有グラフィクス処理ユニット(GPU)およびVDI上にインストールされたプロセッサのグラフィクス処理パワーを利用して、仮想デスクトップを最大の忠実度で体験する。グラフィクス仮想化プラットフォームの一例に、Microsoft(登録商標)RemoteFX(登録商標)がある。これは、Hyper-V(登録商標)VDIアーキテクチャー上に構築し、ホストされたVDIデスクトップのために設計された新たなペイロードを配信するためにRDPを統合する。

【0003】

[0003] 典型的なVDIアーキテクチャーは、ホスト・パーティションと、多数のゲスト・パーティションまたは仮想機械とを含むことができる。ホスト・パーティションは、GPU、中央処理ユニット(CPU)、およびメモリー空間というような、VDIの基礎的な物理リソースにアクセスし、仮想機械のこれらのリソースに対するアクセスを割り当てそして管理することができる。各仮想機械は、割り当てられた物理リソースの仮想化である、1組の仮想リソースを有する。

【0004】

[0004] このため、リモート計算システムでは、クライアントは、仮想機械またはその中で実行する仮想デスクトップ・セッションに接続することができ、このセッションにおいてこのクライアントの認証が管理される。ユーザー入力データまたはグラフィクス・データのような、クライアントから仮想機械に送信すべきデータは、最初にホスト・パーティション上のネットワーク・インターフェース・カード(NIC)に送信され、次いで仮想機械に導き直される。仮想機械は、その仮想リソースを用いてデータを処理することができる。仮想機械は、処理したデータを、基礎的な物理リソース上での更なる処理のために、ホスト・パーティションに送る。ホスト・パーティションは、更に、このデータを処理し、クライアントとの認証のために仮想機械に返送する。仮想機械は、このデータをホスト・パーティションNICを通じてクライアントに送信するために、パッケージ化し再度ホスト・パーティションに導き直す。ホスト・パーティションと仮想機械との間におけるデータの反復横断は、集中的な動作を必要とする可能性があり、大量のメモリーおよびCPUリソースを消費する可能性があり、クライアントに対するデータ配信レイテンシが増大する可能性がある。

【発明の概要】

【0005】

[0005] リモート計算システム環境においてクライアントから受信される、グラフィクス・データのような、データの処理を最適化するためのシステム、方法、およびコンピューター読み取り可能媒体について開示する。現在のアーキテクチャーと比較すると、このような最適化は、ホストされるメモリーおよびCPUリソースの使用削減、およびクライアントに対するデータ配信レイテンシの短縮を含む。

【0006】

[0006] 一実施形態では、クライアントは、計算サーバーのような他の計算デバイスとの第1接続を開始して、仮想デスクトップまたはその中におけるセッションにおいてクライアント作業負荷を実行することができる。計算サーバーは、第1接続を認証することができ、クライアント・アドレスを取得することができる。計算サーバーは、次に、グラフィクス・サーバーとの接続をネゴシエートすることができ、グラフィクス・サーバーを初期化して、クライアント・アドレスから始まる接続を予想する(expect)ようにグラフィクス・サーバーに命令する。また、計算サーバーはグラフィクス・サーバーからグラフィクス・サーバー・アドレスも取得することができる。計算サーバーは、このグラフィクス・サーバー・アドレスをクライアントに供給することができる。一方、クライアントは、このグラフィクス・サーバー・アドレスを用いて、グラフィクス・サーバーとの第3接続を確立することができる。一旦これらの接続が確立されたなら、クライアントは、キーボード入力、マウス入力等のようなユーザーの画面入力を計算サーバーに、第1接続を通じて、供給することができる。計算サーバーは、供給された入力を処理し、表示コマンドやコールのような、処理データを出力することができる。計算サーバーは、処理したデータを、処理のためにグラフィクス・サーバーに送ることができる。次いで、グラフィクス・サーバーは、受信したデータを処理し、グラフィクス出力データをクライアントに送ることができる。したがって、グラフィクス・サーバーがグラフィクス出力データを計算サーバーに返送する必要はない。同様に、計算サーバーがグラフィクス出力データをクライアントに送る必要もない。

10

【0007】

20

[0007] 計算サーバーは、仮想化計算システム上にホストされるゲスト・パーティションまたは仮想機械であることができ、一方グラフィクス・サーバーは仮想化計算システム上のホスト・パーティションであることができる。第1接続は、例えば、TCP/IP、UDP、または他のネットワーク・ベース通信のいずれでも可能であり、リモート・デスクトップ・セッション接続を含むことができる。第2接続は、例えば、VMBus、ハイパーバイザー（仮想機械モニターとしても知られる）を介した通信、TCP/IP、UDP、または他のネットワーク・ベース接続のいずれかというような、イントラ・パーティション通信チャネルを含むことができる。第3接続は、例えば、TCP/IP、UDP、FCOE、100GBイーサネット(登録商標)、またはネットワーク・ベース接続のいずれかとすることができる。

30

【0008】

[0008] 一実施形態では、複数のクライアントを同時に複数の計算サーバーにそして複数のグラフィクス・サーバーに接続することができる。複数のクライアントの内1つのクライアントとの接続を確立し、このクライアントを認証する最初のステップにおいて、クライアントを複数の計算サーバーの内第1計算サーバーに接続するために、リディレクターおよび/またはブローカーを用いることができる。リディレクターおよび/またはブローカーは、複数の計算サーバーの可用性を判断し、それに応じてクライアントと第1計算サーバーとの間の接続を割り当てることができる。したがって、計算リソースの負荷均衡の場合、仮想機械ライブ・マイグレーションのような技法は継ぎ目なく作業負荷を第1計算サーバーから第2計算サーバーに移転することができる。第2計算サーバーへのクライアント接続は、再確立することができ、一方クライアントと複数のグラフィクス・サーバーの内1つのグラフィクス・サーバーとの間の接続は、同じままであり続けることができる。同様に、グラフィクス・サーバー・マネージャーは、クライアントが接続された計算サーバーを、複数のグラフィクス・サーバーの内1つのグラフィクス・サーバーに接続するために用いることができる。グラフィクス・サーバー・マネージャーは、複数のグラフィクス・サーバーの可用性を判断し、これに応じてクライアントが接続された計算サーバーと第1グラフィクス・サーバーとの間の接続を割り当てることができる。したがって、グラフィクス・サーバーの負荷均衡の場合、クライアントが接続された計算サーバーは、第2グラフィクス・サーバーとの新たな接続を形成することができ、第2グラフィクス・サーバーとの新たな接続を確立するようにクライアントに要求することができる。次いで

40

50

、クライアントは第2グラフィクス・サーバーに継ぎ目無しに移行することができる。

【0009】

[0009] 複数のクライアントが同時に複数の計算サーバーおよび複数のグラフィクス・サーバーに接続される実施形態では、少なくとも1つのクライアントが、複数のグラフィクス・サーバーからの、レンダリングされ、キャプチャされ、圧縮されたデータを受信するように構成することができる。したがって、少なくとも1つのクライアントを通じてインターフェースするユーザーは、1つ以上のクライアントから発する、レンダリングされ、キャプチャされ、圧縮されたデータを見ることができる。同様に、少なくとも1つのグラフィクス・サーバーは、複数のクライアントから発する、レンダリングされ、キャプチャされ、圧縮された処理データをクライアントに送信するように構成することができる。

10

【0010】

[0010] この摘要は、本発明の態様の全体像を規定することを意図している。これは、本発明のあらゆる必要なステップまたはコンポーネントを特定することを意図しているのではない。以上のことに加えて、他の態様も特許請求の範囲、図面、および本開示の一部をなす本文において記載される。尚、本開示の1つ以上の種々の態様が、ここで引用した本開示の態様を実施するための回路および/またはプログラミングを含むのでもよいがこれらには限定されないこと、その回路および/またはプログラミングは、システム設計者の設計選択に応じて、ここで引用した態様を実施するように構成されるハードウェア、ソフトウェア、および/またはファームウェアの事実上あらゆる組み合わせであることができることは、当業者には認めることができよう。

20

【図面の簡単な説明】

【0011】

[0011] 本明細書にしたがってリモート計算環境において受信された、グラフィクス・データのようなデータの処理を最適化するためのシステム、方法、およびコンピューター媒体について、添付図面を参照しながら更に説明する。

【図1】図1は、本開示の態様を実現することができる計算環境の一例を示す。

【図2】図2は、本開示の態様を実施するためのリモート計算環境を示す。

【図3】図3は、本開示の態様を実施するために、複数のクライアントが複数のリモート・サーバーに接続することができるリモート環境を示す。

30

【図4】図4は、複数の仮想機械を有する仮想機械環境の一例を示す。

【図5】図5は、複数の仮想機械を有する仮想機械環境の他の例を示す。

【図6】図6は、本開示の態様を実施するために複数の仮想デスクトップ・セッションをホストするリモート・サーバーを示す。

【図7】図7は、本開示の態様を実施するための計算サーバーおよびグラフィック・サーバーのアーキテクチャー例を示す。

【図8】図8は、本開示の態様を実施するための計算サーバーおよびグラフィクス・サーバーの他のアーキテクチャー例を示す。

【図9】図9は、本開示の態様を実施するための計算サーバーおよび複数のグラフィクス・サーバーの例を示す。

40

【図10】図10は、本開示の態様を実施するための仮想環境において実装される(effectuate)計算サーバーおよびグラフィクス・サーバーの例を示す。

【図11】図11は、本開示の態様を実施するための、クライアント、計算サーバー、およびグラフィクス・サーバーを有する計算環境を示す。

【図12】図12は、本開示の態様を実施するための、複数のクライアント、計算サーバー、およびグラフィクス・サーバーを有する計算環境を示す。

【図13】図13は、本開示の態様を実施するための方法の一例を表すフロー・チャートを示す。

【図14】図14は、本開示の態様を実施するためのシステムおよびコンピューター読み取り可能記憶媒体の例を示す。

50

【発明を実施するための形態】

【0012】

【0026】 本開示の種々の実施形態の完全な理解が得られるようにするために、以下の説明および図においてある程度具体的な詳細を明記する。計算およびソフトウェア技術に関連することが多いある種の周知の詳細については、本開示の種々の実施形態を不要に曖昧にするのを避けるために、以下の開示では明記しないこととする。更に、当業者は、以下に説明する詳細の内1つ以上がなくても、本開示の他の実施形態を実施できることが理解されよう。最後に、以下の開示ではステップおよびシーケンスを参照して種々の方法について説明するが、このような説明は、本開示の実施形態の明確な実施態様を提示するためであり、これらのステップおよびステップのシーケンスが本開示を実施するためには必須であると解釈してはならない。

10

【0013】

【0027】 尚、本明細書において説明する種々の技法は、ハードウェアまたはソフトウェア、あるいははしかるべきときには、双方の組み合わせで実現できることは言うまでもない。つまり、本開示の方法および装置、またはその一部のある種の態様は、フロッピー(登録商標)・ディスク、CD-ROM、ハード・ドライブ、または他のあらゆる機械読み取り可能記憶媒体というような、有形媒体に具体化されたプログラム・コード(即ち、命令)の形態をなすことができ、プログラム・コードがコンピューターのような機械にロードされこの機械によって実行されると、この機械が本開示を実施する装置になる。プログラマブル・コンピューター上におけるプログラム・コード実行の場合、計算デバイスは、通常、プロセッサ、このプロセッサによって読み取り可能な記憶媒体(揮発性および不揮発性メモリーおよび/または記憶エレメントを含む)、少なくとも1つの入力デバイス、および少なくとも1つの出力デバイスを含む。例えば、アプリケーション・プログラミング・インターフェース(API)、再利用可能な制御手段等の使用によって、1つ以上のプログラムが、本開示に関して説明されるプロセスを実現するまたは利用することもできる。このようなプログラムは、コンピューター・システムと通信するために、高級手順プログラミング言語またはオブジェクト指向プログラミング言語で実現されることが好ましい。しかしながら、プログラム(1つまたは複数)は、所望であれば、アセンブリまたは機械語で実現することもできる。いずれの場合でも、言語は、コンパイル型言語でもインタプリタ型言語でもよく、ハードウェアの実施形態と組み合わせてもよい。

20

30

【0014】

【0028】 本開示全体を通じて用いられる回路(circuitry)という用語は、ハードウェア割り込みコントローラ、ハード・ドライブ、ネットワーク・アダプター、グラフィクス・プロセッサ、ハードウェア・ベース・ビデオ/オーディオ・コーデックというようなハードウェア・コンポーネント、およびこのようなハードウェアを動作させるために用いられるファームウェア/ソフトウェアを含むことができる。また、回路という用語は、ファームウェアによってまたはある種の方法で設定されるスイッチによって機能(1つまたは複数)を実行するように構成されたマイクロプロセッサ、または1つ以上の論理プロセッサ、例えば、マルチコア汎用処理ユニットの1つ以上のコアも含むことができる。この例における論理プロセッサ(1つまたは複数)は、機能(1つまたは複数)を実行するように動作可能なロジックを具体化するソフトウェア命令によって構成することができる。このソフトウェア命令は、メモリー、例えば、RAM、ROM、ファームウェア、および/または仮想メモリーからロードされる。回路がハードウェアおよびソフトウェアの組み合わせを含む実施形態例では、実施者(implementer)がロジックを具体化するソース・コードを書くのでもよく、続いて、このソース・コードが機械読み取り可能コードにコンパイルされ、論理プロセッサによって機械読み取り可能コードを実行することができる。技術的現状はハードウェア、ソフトウェア、またはハードウェア/ソフトウェアの組み合わせの間の差が殆どなくなるところまで発展したことは当業者は認めることができるはずであるので、機能を実装するためのハードウェアかソフトウェアかの選択は、単なる設計の選択事項に過ぎない。つまり、ソフトウェア・プロセスを等価のハードウェア構造に

40

50

変換でき、ハードウェア構造自体を等価なソフトウェア・プロセスに変換できることは当業者は認めることができるので、ハードウェアの実施態様かまたはソフトウェアの実施態様の選択は、実施者に委ねられる。

【 0 0 1 5 】

【0029】 本発明の実施形態は、1つ以上のコンピューター上で実行することができる。図1および以下に続く論述は、本開示を実現することができる適した計算環境について端的で総合的な説明を行うことを意図している。当業者は、コンピューター・システムが本明細書において以下で説明するコンポーネントの一部または全部を有することができることを認めることができよう。

【 0 0 1 6 】

【0030】 図1は、本開示の態様と共に作用するように構成された計算システムの一例を示す。この計算システムは、コンピューター100等を含むことができ、コンピューター100は、論理処理ユニット102、システム・メモリー22、およびシステム・バス23を含む。システム・バス23は、システム・メモリーから論理処理ユニット102までを含む種々のシステム・コンポーネントを結合する。システム・バス23は、メモリー・バスまたはメモリー・コントローラ、周辺バス、および種々のバス・アーキテクチャーの内いずれかを用いるローカル・バスを含む、様々なタイプのバス構造の内いずれでもよい。システム・メモリーは、リード・オンリー・メモリー(ROM)24およびランダム・アクセス・メモリー(RAM)104を含む。基本入力/出力システム26(BIOS)は、起動中のように、コンピューター100内部の要素間で情報を転送するのを助ける基本的なルーチンを収容し、ROM24に格納される。コンピューター100は、更に、図示しないハード・ディスクに対する読み取りおよび書き込みを行うハード・ディスク・ドライブ27、リムーバブル磁気ディスク118に対する読み取りおよび書き込みを行う磁気ディスク・ドライブ28、およびCD-ROMまたは他の光媒体のようなリムーバブル光ディスク31に対する読み取りおよび書き込みを行う光ディスク・ドライブ30も含むとよい。実施形態例では、本開示の態様を具体化するコンピューター実行可能命令は、ROM24、ハード・ディスク(図示せず)、RAM104、リムーバブル磁気ディスク118、光ディスク31、および/または論理処理ユニット102のキャッシュに格納してもよい。ハード・ディスク・ドライブ27、磁気ディスク・ドライブ28、および光ディスク・ドライブ30は、システム・バス23に、それぞれハード・ディスク・ドライブ・インターフェース32、磁気ディスク・ドライブ・インターフェース33、および光ドライブ・インターフェース34を介して接続される。ドライブおよびそれらに付随するコンピューター読み取り可能媒体は、コンピューター100のためのコンピューター読み取り可能命令、データ構造、プログラム・モジュール、およびその他のデータの揮発性ストレージを設ける。本明細書において説明する環境はハード・ディスク、リムーバブル磁気ディスク118、およびリムーバブル光ディスク31を採用するが、磁気カセット、フラッシュ・メモリー・カード、デジタル・ビデオ・ディスク、ベルヌーイ・カートリッジ、ランダム・アクセス・メモリー(RAM)、リード・オンリー・メモリー(ROM)等のような、コンピューターによってアクセス可能なデータを格納することができる他のタイプのコンピューター読み取り可能媒体もこの動作環境において使用してもよいことは、当業者には認められてしかるべきである。

【 0 0 1 7 】

【0031】 オペレーティング・システム35、1つ以上のアプリケーション・プログラム36、他のプログラム・モジュール37、およびプログラム・データ38を含む多数のプログラム・モジュールを、ハード・ディスク、磁気ディスク118、光ディスク31、ROM24、またはRAM104に格納することができる。ユーザーは、キーボード40およびポインティング・デバイス42のような入力デバイスを介して、コマンドおよび情報をコンピューター100に入力することができる。他の入力デバイス(図示せず)は、マイクロフォン、ジョイスティック、ゲーム・パッド、衛星ディスク、スキャナー等を含むことができる。これらおよびその他の入力デバイスは、多くの場合、システム・バスに

10

20

30

40

50

結合されたシリアル・ポート・インターフェース 46 を介して、論理処理ユニット 102 に接続されるが、パラレル・ポート、ゲーム・ポート、またはユニバーサル・シリアル・バス (USB) のような他のインターフェースによって接続されてもよい。ディスプレイ 47 または他のタイプのディスプレイ・デバイスも、GPU/ビデオ・アダプター 112 のようなインターフェースを介して、システム・バス 23 に接続することができる。ディスプレイ 47 に加えて、コンピューターは、通例、スピーカーおよびプリンターのような、他の周辺出力デバイス (図示せず) も含む。また、図 1 のシステムは、ホスト・アダプター 55、小型コンピューター・システム・インターフェース (SCSI) バス 56、および SCSI バス 56 に接続された外部記憶デバイス 62 も含む。

【0018】

[0032] コンピューター 100 は、リモート・コンピューター 49 のような、1 つ以上のリモート・コンピューターへの論理接続を用いて、ネットワーク接続環境において動作することもできる。リモート・コンピューター 49 は、他のコンピューター、サーバー、ルーター、ネットワーク PC、ピア・デバイスまたは他の一般的なネットワーク・ノード、仮想機械であってもよく、通例、コンピューター 100 に関して以上で説明したエレメントの多くまたは全部を含むことができるが、図 1 には、メモリー記憶デバイス 50 だけが示されている。図 1 に示す論理接続は、ローカル・エリア・ネットワーク (LAN) 51 およびネットワーク 52 を含むことができ、ネットワーク 52 は、一例として、ワイド・エリア・ネットワーク (WAN) である。このようなネットワーク接続環境は、事務所、企業規模のコンピューター・ネットワーク、イントラネット、およびインターネットでは極普通である。

【0019】

[0033] LAN ネットワーク接続環境において用いられる場合、コンピューター 100 は、ネットワーク・インターフェース・コントローラ (NIC) 114 またはアダプターを介して LAN 51 に接続することができる。WAN ネットワーク接続環境において用いられる場合、コンピューター 100 は、通例、モデム 54、またはインターネットのようなネットワーク 52 を通じて通信を確立する他の手段を含むことができる。モデム 54 は、内蔵型でも外付け型でもよく、シリアル・ポート・インターフェース 46 を介してシステム・バス 23 に接続することができる。ネットワーク接続環境では、コンピューター 100 に関して図示したプログラム・モジュールまたはその一部が、リモート・メモリー記憶デバイスに格納されてもよい。尚、図示するネットワーク接続は例であり、コンピューター間で通信リンクを確立する他の手段を用いてもよいことは認められよう。更に、本開示の多数の実施形態がコンピューター・システムには特に非常に適していると考えられるが、本文書において、本開示をこのような実施形態に限定することを意図するものは何もない。

【0020】

[0034] 場合によっては、ユーザーが計算アプリケーション、即ち、別の計算デバイス上で実行しているアプリケーションに、リモートにアクセスすることを望むこともある。一実施態様では、仮想デスクトップのようなリモート・デスクトップを介してこのようなアクセスをユーザーに付与する。リモート・デスクトップ・システムの実施形態は、1 つ以上のコンピューターを実行することができ、または図 1 のコンピューター 100 に関して説明したコンポーネントの一部または全部を有することもできる。リモート・デスクトップ・システムは、クライアント・コンピューター・システムによってリモートに実行し表示することができるアプリケーションを維持するコンピューター・システムである。図 2 は、リモート・デスクトップ・システム 200 のアーキテクチャー例を示す。リモート・デスクトップ・システム 200 は、リモート・クライアント・コンピューター 210 とリモート・サーバー・コンピューター 220 とを含むことができる。リモート・クライアント・コンピューター 210 およびリモート・サーバー・コンピューター 220 は、仮想デスクトップ・セッションのようなリモート・セッションを互いに実行するように構成される。

【 0 0 2 1 】

[0035] 図示のように、リモート・サーバー・コンピューター 220 は、リモート・セッションをリモート・クライアント・コンピューター 210 に配給し(serve)、この場合、リモート・サーバー・コンピューター 220 が、リモート・クライアント・セッション 222 を実行するユーザーからのクライアント・グラフィック出力を送る。リモート・ユーザー入力、リモート・クライアント・コンピューター 210 において入力される。入力マネージャー 212 が、リモート・ユーザー入力を処理し、ネットワークを通じて(例えばリモート・デスクトップ・プロトコル(RDP)のような、国際電気通信連合(ITU)T.120 ファミリーのプロトコルに基づくプロトコルを用いる)リモート・サーバー・コンピューター 220 上のリモート・ユーザー・アプリケーション 224 に転送することができる。ネットワークは、ローカル・エリア・ネットワーク、ワイド・エリア・ネットワーク、ケーブル・ネットワーク、インターネット、ワールド・ワイド・ウェブ、または企業内ネットワーク(corporate enterprise network)のような、いずれのタイプの通信ネットワークでもよい。リモート・ユーザー・アプリケーション 224 は、リモート・サーバー・コンピューター 220 上にホストされたリモート・クライアント・セッション 222 において実行することができる。リモート・ユーザー・アプリケーション 224 は、入力がリモート・サーバー・コンピューター 220 において入力されたかのように、この入力を処理する。リモート・ユーザー・アプリケーション 224 は、受信した入力に回答して、リモート・サーバー出力を生成し、この出力はネットワークを通じてリモート・クライアント・コンピューター 210 に転送される。リモート・クライアント・コンピューター 210 は、この出力データをリモート・ユーザーに提示する。このように、リモート・クライアント・コンピューター 210 において入力が受信され出力が提示されるが、処理は実際にはリモート・サーバー・コンピューター 220 において行われる。

【 0 0 2 2 】

[0036] リモート・ユーザー・アプリケーション 224 に加えて、リモート・クライアント・セッション 222 は、シェルおよびデスクトップのようなユーザー・インターフェース、このデスクトップ内部におけるマウスの動きを追跡するサブシステム、アイコン上のマウス・クリックを、プログラムのインスタンスを実装するコマンドに変換するサブシステム、他のアプリケーション等を含むことができる。尚、以上の論述は例示であり、本明細書において開示する主題は、種々のクライアント/サーバー環境において実現することができ、特定のリモート・プレゼンテーション生産物(remote presentation product)には限定されないことは言うまでもない。

【 0 0 2 3 】

[0037] 全てではないにしても、殆どのリモート・デスクトップ環境において、リモート・ユーザーの入力データ(リモート・クライアント・コンピューター 210 において入力される)は、通例、アプリケーションに対するコマンドを表すマウスおよびキーボード・データを含む。出力データ(リモート・サーバー・コンピューター 220 においてリモート・ユーザー・アプリケーションによって生成される)は、通例、リモート・クライアント・コンピューター 210 における表示のためのグラフィクス・データを含む。また、多くのリモート・デスクトップ環境は、他のタイプのデータを転送するために拡張する機能も含む。一実施形態例では、ユーザー・アプリケーション 224 から出力されたグラフィック・データを、リモート・サーバー・コンピューター 220 上にホストされたグラフィクス・マネージャー 226 に送ることができる。グラフィクス・マネージャー 226 は、グラフィクス・データをレンダリングし、キャプチャし、圧縮し、ネットワークを通じてリモート・クライアント・コンピューター 210 上のリモート・ユーザー・ディスプレイ 214 に転送することができる。リモート・ユーザー・ディスプレイ 214 は、このグラフィクス出力データをリモート・ユーザーに表示することができる。

【 0 0 2 4 】

[0038] リモート・デスクトップ環境の一実施形態では、リモート・サーバー・コンピューターが複数のリモート・クライアント・コンピューターのために複数のリモート・セ

ッション（または仮想デスクトップ）を実行することもできる。したがって、複数のリモート・クライアント・コンピューターに対するセッションの割り当てを制御するために、ブローカーを用いることもできる。加えて、リモート・デスクトップ環境では、特定のリモート・クライアント・コンピューターを配給することができる、複数のリモート・サーバー・コンピューターがあることもある。したがって、特定のリモート・クライアント・コンピューターを配給するリモート・サーバー・コンピューターの割り当てを制御するために、リディレクターを用いることもできる。図3は、このようなリモート・デスクトップ・システム300の実施形態例を示す。

【0025】

[0039] 複数のリモート・クライアント・コンピューター310（A～N）は、図2のリモート・クライアント・コンピューター210のように、ネットワークを通じてリモート・サーバー・システム350と通信することができる計算デバイスであればいずれでもよい。リモート・サーバー・システム350は、リディレクター330、ブローカー340、および複数のリモート・サーバー・コンピューター320（A～N）を含むことができる。リディレクター330およびブローカー340は、本明細書において以下で説明するようなデバイスのそれぞれの機能を実現するように構成された、プロセッサおよびメモリを含む計算デバイスであってもよい。リモート・サーバー・コンピューター320（A～N）は、図2のコンピューター100および図2のリモート・サーバー・コンピューター220に関して説明したコンポーネントの一部を有してもまたは全部を有してもよい。リモート・サーバー・コンピューター320（A～N）は、仮想機械として実現されてもよい。仮想機械は、1つのハードウェア・インフラストラクチャ上または別々のハードウェア・インフラストラクチャ上で実行することができる。ブローカー340は、ゲートウェイ（図示せず）を用いてリディレクター330に接続された単体デバイスであってもよく、リディレクター330内部に配置されてもよく、またはリモート・サーバー・コンピューター320（A～N）内に配置されてもよい。また、リディレクター330もリモート・サーバー320（A～N）に配置されてもよい。

【0026】

[0040] ブローカー340は、当該ブローカー340に格納されているセッション状態情報に基づいて、セッションをリモート・クライアント・コンピューターに割り当てる。セッション状態情報は、例えば、セッションID、ユーザー名、セッションが存在しているリモート・サーバー・コンピューターの名前、各リモート・サーバー・コンピューター内にあるアクティブ・セッションの数等を含んでもよい。本明細書において用いる場合、セッションとは仮想デスクトップ・セッション（仮想機械セッションとしても知られる）であってもよい。

【0027】

[0041] リモート・クライアント・コンピューター310は、最初に、リモート・クライアント・コンピューター310（A～N）の負荷均衡に備えることができるリディレクター330に接続する。このような場合、リディレクター330は通例最初に接続要求を受ける。次いで、リディレクター330はこの接続要求を受け入れ、リモート・クライアント・コンピューター310をどこにリディレクトするか決定するように、ブローカー340に問い合わせる。ブローカー340は、その特定の環境のセッション状態情報を分析し、リモート・クライアント・コンピューター310をリディレクトすることができるリモート・サーバー・コンピューター320を特定する。特定されたリモート・サーバー・コンピューター320は、リモート・クライアント・コンピューター310によって以前にアクセスされたが後に切断され、リモート・クライアント・コンピューター310を再接続することができるセッションを保有するのでもよい。一実施形態では、リモート・クライアント・コンピューター310が他に既存のセッションを全く保有していないことを条件に、特定されたリモート・サーバー・コンピューター320が、リモート・クライアント・コンピューター310を接続することができる新たなセッションを提供することもできる。

10

20

30

40

50

【 0 0 2 8 】

[0042] ブローカー 340 は、要求されたリモート・サーバー・コンピューター 320 に情報を送り、このリモート・クライアント・コンピューター 320 が、特定されたリモート・サーバー・コンピューター 310 との接続を確立することを可能にする。例えば、この情報は、機械 ID、セッション ID、および特定されたリモート・サーバー・コンピューター 320 の位置を含んでもよい。一旦リモート・クライアント・コンピューター 310 が、特定されたリモート・サーバー・コンピューター 320 との接続を確立すると、リモート・クライアント・コンピューター 310 は、特定されたリモート・サーバー・コンピューター 320 内にあるアプリケーションにアクセスすることができる。これらのアプリケーションは、リモート・サーバー・システム 350 においてリモート・サーバー・コンピューター 320 を特定するときに用いられたブローカー 340 のロジックに対して適合する (compatible) とよい。

10

【 0 0 2 9 】

[0043] 一実施形態では、以上で説明したシステムは、例えば、リモート・クライアント・コンピューター 310 を、リモート・サーバー・コンピューター上で実行している複数の仮想デスクトップまたはセッションの内 1 つに接続するために用いることができる。リモート・クライアント・コンピューターは、リモート・デスクトップ・プロトコル (RDP) パケットにおけるリディレクター・トークンを調べる。リモート・クライアント・コンピューターは、このリディレクター・トークンに含まれる情報に基づいて、多くの仮想デスクトップの内 1 つに接続する。

20

【 0 0 3 0 】

[0044] 他の実施形態では、ブローカー 340 およびプール・マネージャー (図示せず) を用いて、リモート・クライアント・コンピューター 310 を仮想デスクトップの 1 つに接続することができる。プール・マネージャーは、ブローカー 340 内部に配置するのでもよい。ブローカー 340 は、リモート・クライアント・コンピューター 310 が仮想機械 (VM) 上にホストされた仮想デスクトップに接続されているときに、仮想デスクトップをリモート・クライアント・コンピューター 310 に割り当て、プール・マネージャーは、仮想デスクトップの内どれが、割り当てるために利用可能か示す。

【 0 0 3 1 】

[0045] 更に他の実施形態では、リモート・クライアント・コンピューター 310 を仮想デスクトップに接続することができる。リモート・クライアント・コンピューター 310 は、インターネット・プロトコル (IP) アドレスを生成し、リモート・クライアント・コンピューター 310 と仮想デスクトップとの間に接続を確立するためにブローカー 340 によって用いられるネットワーク名を示す。個々の仮想デスクトップ IP アドレスをリモート・クライアント・コンピューター 310 (A ~ N) から隠すことによって、最初はブローカー 340 の 1 つのネットワーク名のみが、外部に対してリモート・クライアント・コンピューター 310 (A ~ N) に露出されるだけで済む。

30

【 0 0 3 2 】

[0046] 図 4 は、複数の仮想機械を有する仮想機械環境の一例を示す。仮想化コンピューター・システム 400 は、図 2 のリモート・サーバー・コンピューター 220 および図 3 のリモート・サーバー・コンピューター 320 (A ~ N) を実現するために用いることができる。

40

【 0 0 3 3 】

[0047] 図 4 に示すように、コンピューター・システム 400 は、図 1 において説明したエレメントと、仮想機械を実装するように動作可能なコンポーネントとを含むことができる。このようなコンポーネントの 1 つに、ハイパーバイザー・マイクロカーネル (hypervisor microkernel) 402 がある。これは、当技術分野では、仮想機械モニターとも呼ばれることもある。ハイパーバイザー・マイクロカーネル 402 は、コンピューター・システム 400 のハードウェアに対するアクセスを制御し調停するように構成することができる。ハイパーバイザー・マイクロカーネル 402 は、ゲスト・パーティション 1 からゲスト

50

ト・パーティションNまで(Nは1よりも大きい正の整数)のような、パーティションと呼ばれる実行環境を生成することができる。ここで、ゲスト・パーティションは、ハイパーバイザー・マイクロカーネル402によってサポートされる分離の基本単位である。ゲスト・パーティションは、チャイルド・パーティションと言っても分かるであろう。ハイパーバイザー・マイクロカーネル402は、1つのパーティションにおけるプロセスを、他のパーティションのリソースにアクセスすることから分離することができる。ハイパーバイザー・マイクロカーネル402の制御下にある1組のハードウェア・リソース、例えば、メモリー、デバイス、プロセッサ・サイクル等に、各ゲスト・パーティションをマッピングすることができる。実施形態では、ハイパーバイザー・マイクロカーネル402は、単体のソフトウェア生産物であることができ、オペレーティング・システムの一部であることができ、マザーボードのファームウェア内に埋め込むことができ、特殊集積回路であることができ、またはこれらの組み合わせとすることができる。

10

【0034】

[0048] ハイパーバイザー・マイクロカーネル402は、物理コンピューター・システムにおいてメモリーのゲスト・オペレーティング・システムのビューを制限することによって、パーティション形成(partitioning)を施行することができる。ハイパーバイザー・マイクロカーネル402が仮想機械をインスタンス化するとき、ページ、例えば、システム物理メモリー(SPM)の、開始アドレスおよび終了アドレスを有する固定長のメモリー・ブロックを、仮想機械にゲスト物理メモリー(GPM)として割り当てることができる。ここで、ゲストの制限されたシステム・メモリーのビューは、ハイパーバイザー・マイクロカーネル402によって制御される。ゲスト物理メモリーという用語は、仮想機械の観点からメモリーのページを記述する省略表現(shorthand way)であり、システム物理メモリーという用語は、物理システムの観点からメモリーのページを記述する省略表現である。つまり、仮想機械に割り当てられるメモリーのページは、ゲスト物理アドレス(仮想機械によって用いられるアドレス)とシステム物理アドレス(そのページの実際のアドレス)とを有することになる。

20

【0035】

[0049] ゲスト・オペレーティング・システムが、ゲスト物理メモリーを仮想化することもできる。仮想メモリーとは、オペレーティング・システムにメモリーをオーバー・コミットさせ、連続する作業メモリーに対する1回のアクセスをアプリケーションに与えさせる管理技法である。仮想化環境では、ゲスト・オペレーティング・システムが1つ以上のページ・テーブルを用いて、仮想ゲスト・アドレスとして知られる仮想アドレスをゲスト物理アドレスに変換することができる。この例では、メモリー・アドレスがゲスト仮想アドレス、ゲスト物理アドレス、およびシステム物理アドレスを有してもよい。

30

【0036】

[0050] 図示する例では、コンピューター・システム400は、ホスト・パーティションを含む。ホスト・パーティションは、Xenのオープン・ソース・ハイパーバイザーのドメイン0と同様であると考えることができる。また、ホスト・パーティションは、ペアレント・パーティションまたはルート・パーティションと呼ぶこともできる。図示のように、ホスト・パーティションはホスト404を含むことができる。ホスト404は、アプリケーションのコンピューター・システム400の基礎物理ハードウェアとの相互作用を可能にするデバイス・ドライバ424を含むことができる。したがって、ホスト404は、論理処理ユニット102、GPU112、およびNIC114のような、コンピューター・システム400の物理ハードウェアにアクセスすることができる。ホスト404は、オペレーティング・システム(または1組の構成ユーティリティ)であることができる。

40

【0037】

[0051] ホスト404は、仮想化サービス・プロバイダー428(VSP)を用いることによって、ゲスト・パーティション1~Nにおいて実行するゲスト・オペレーティング・システムにリソースを供給するように構成することができる。VSP428は、通例、

50

オープン・ソース・コミュニティにおけるバックエンド・ドライバーと呼ばれ、仮想化サービス・クライアント(VSC)(通例、オープン・ソース・コミュニティまたは準仮想化デバイスにおいてフロントエンド・ドライバーと呼ばれる)によって、インターフェースをハードウェア・リソースに多重化するために用いることができる。図4に示すように、仮想化サービス・クライアントは、ゲスト・オペレーティング・システムのコンテキストの範囲内で実行する。しかしながら、これらのドライバーはゲストにおける残りのドライバーとは異なり、これらのドライバーにはゲストではなくハイパーバイザーが供給されてもよい。一実施形態例では、仮想化サービス・プロバイダー428がVSC416および418と通信するために用いられるパスは、仮想化パスであると考えることができる。

【0038】

[0052] ホスト・パーティションVSP428ならびにゲスト・パーティションVSC416および418は、仮想機械バス(VMBus)に接続することができる。VMBusは、パーティション間通信を可能にする論理チャネルである。仮想プロセッサ430および432のような仮想リソースに対するゲスト・パーティションの要求は、VMBusを通じて、要求を管理することができるホスト・パーティションにおけるデバイスにリダイレクトすることができる。ホスト・パーティションからゲスト・パーティションへの応答も、VMBusを通じてリダイレクトすることができる。このプロセス全体は、ゲストOS220および222に対して透過的であることができる。他の実施形態では、ホスト・パーティションVSP228ならびにゲスト・パーティションVSC216および218は、メッセージ・パケットを送信し受信することによって、TCP/IPネットワークのようなネットワークを通じて通信することができる。

【0039】

[0053] 図4に示すように、エミュレーター434、例えば、仮想化IDEデバイス、仮想化ビデオ・アダプター、仮想化NIC等は、ホスト404内部で実行するように構成することができ、ゲスト・オペレーティング・システム420および422に利用可能なリソースに添付される(attach)。例えば、ゲストOSが、デバイスのレジスタがあるところにマッピングされたメモリー位置、またはメモリー・マッピングされたデバイスにタッチする(touch)とき、ハイパーバイザー・マイクロカーネル402は要求を横取りし(intercept)、ゲストが書き込もうとする値を、関連するエミュレーターに渡すことができる。ここで、この例におけるリソースは、仮想デバイスが位置するところであると考えることができる。このようなエミュレーターの使用は、エミュレーション・パス(emulation path)と見なすことができる。エミュレーション・パスは、仮想化パスと比較すると非効率的である。何故なら、VSPとVSCとの間でメッセージを渡すよりも、デバイスをエミュレートする方が多くのCPUリソースを必要とするからである。例えば、エミュレーション・パスを経由して値をディスクに書き込むために必要とされるレジスタにマッピングされたメモリーにおける数百回の動作は、仮想化パスにおいてVSCからVSPに渡される1つのメッセージに減らすことができる。

【0040】

[0054] 各ゲスト・パーティションは、1つ以上の仮想プロセッサ(430および432)を含むことができ、これらの仮想プロセッサをゲスト・オペレーティング・システム(420および422)が管理して、そこで実行するスレッドのスケジューリングを行うことができる。一般に、仮想プロセッサは、実行可能命令および関連する状態情報である。状態情報は、物理プロセッサの表現に特定のアーキテクチャーを付与する。例えば、1つの仮想機械が、Intel社のx86プロセッサの特性を有する仮想プロセッサを有するのであってもよく、一方他の仮想プロセッサがPowerPCプロセッサの特性を有するのでもよい。この例における仮想プロセッサは、これらの仮想プロセッサを実装する命令がプロセッサによって支援されるように、コンピューター・システムのプロセッサにマッピングすることができる。つまり、多数のプロセッサを含む実施形態では、例えば、他のプロセッサがハイパーバイザー命令を実行する間に、仮想プロセッサを同時にプロセッサによって実行することができる。パーティションにおける仮想プロ

10

20

30

40

50

セッサとメモリーとの組み合わせは、仮想機械と見なすことができる。

【 0 0 4 1 】

[0055] ゲスト・オペレーティング・システム (4 2 0 および 4 2 2) は、例えば、Microsoft (登録商標)、Apple (登録商標)、オープン・ソース・コミュニティ等からのオペレーティング・システムのような、いずれのオペレーティング・システムでも可能である。ゲスト・オペレーティング・システムは、ユーザー / カーネル動作モードを含むことができ、スケジューラ、メモリー・マネージャー等を含むことができるカーネルを有することができる。一般的に言うと、カーネル・モードは、少なくとも特権プロセッサ命令にアクセスを付与するプロセッサにおける実行モードを含むことができる。各ゲスト・オペレーティング・システムは、関連するファイル・システムを有することができ、このファイル・システムには、リモート・サービスまたは仮想デスクトップ・セッション、端末サーバー、電子商取引サーバー、電子メール・サーバー等のようなアプリケーション、およびゲスト・オペレーティング・システム自体を格納することができる。ゲスト・オペレーティング・システムは、仮想プロセッサ上で実行するスレッドのスケジューリングを行うことができ、このようなアプリケーションのインスタンスを実装することができる。

10

【 0 0 4 2 】

[0056] 図 5 は、図 4 のコンポーネントに類似するコンポーネントを示す。しかしながら、この実施形態例では、ハイパーバイザー 5 4 2 が、マイクロカーネル・コンポーネントと、仮想化サービス・プロバイダー 4 2 8 およびデバイス・ドライバ 4 2 4 のような図 4 のホスト 4 0 4 におけるコンポーネントと同様のコンポーネントを含むことができ、一方管理オペレーティング・システム 5 4 0 は、例えば、ハイパーバイザー 5 4 2 を構成するために用いられる構成ユーティリティ (configuration utilities) を収容してもよい。このアーキテクチャでは、ハイパーバイザー 5 4 2 は、図 4 のハイパーバイザー・マイクロカーネル 4 0 2 およびホスト 4 0 4 と同じ機能または同様の機能を実行することができる。ハイパーバイザー 5 4 2 は、単体ソフトウェア生産物であることができ、オペレーティング・システムの一部であることができ、マザーボードのファームウェア内部に埋め込むことができ、および / またはハイパーバイザー 5 4 2 の一部は特殊集積回路によって実装することもできる。

20

【 0 0 4 3 】

[0057] 種々の実施形態では、図 2 のリモート・サーバー・コンピューター 2 2 0 のような、リモート・サーバー・コンピューターは、多数のリモート・クライアント・セッションまたは仮想デスクトップを実行することができる。リモート・クライアント・セッション 2 2 2 のような各リモート・クライアント・セッションは、接続クライアントに対するアプリケーション環境を表すことができる。リモート・サーバー・コンピューターは、図 6 に示すように、接続するリモート・クライアント・コンピューター毎に、少なくとも 1 つのリモート・クライアント・セッションを生成することができる。更に、先に注記したように、リモート・サーバー・コンピューター 2 2 0 は、図 4 および図 5 のコンピューター・システム 4 0 0 のコンポーネントの一部または全部を実行する仮想機械であってもよく、一方、コンピューター・システム 4 0 0 は多数のリモート・クライアント・セッションを実行する。

30

40

【 0 0 4 4 】

[0058] 図 6 に示すのは、コンピューター・システム 6 0 0 であり、リモート・サーバー・コンピューターを実装するように構成された回路を含んでもよく、また他の実施形態では、コンピューター・システム 6 0 0 は、リモート・デスクトップ接続をサポートするように構成された回路を含むことができる。図示する例では、コンピューター・システム 6 0 0 は、セッション 1 から N まで (N は 2 よりも大きい正の整数) というような、接続クライアントに対して 1 つ以上のリモート・クライアント・セッションを生成するように構成することができる。端的に言うと、本発明の実施形態例におけるセッションは、一般に、複数のサブシステム、例えば、ソフトウェア・コードによって実装される動作環境を

50

含むことができ、これらのサブシステムは、コンピューター・システム 600 のカーネル 614 と相互作用するように構成される。例えば、セッションは、デスクトップ・ウィンドウのようなユーザー・インターフェース、ウィンドウ内におけるマウスの動きを追跡するサブシステム、アイコン上でのマウス・クリックをプログラムのインスタンスを実装するコマンドに変換するサブシステム等をインスタンス化するプロセスを含むことができる。セッションは、例えば、コンピューター・システム 600 が図 2 のリモート・クライアント・コンピューター 210 のようなクライアントからの接続要求をネットワーク接続を通じて受信したときに、コンピューター・システム 600 によってユーザー毎に生成することができる。一般に、接続要求は、最初にトランスポート・ロジック 610 によって扱うことができ、例えば、トランスポート・ロジック 610 は、コンピューター・システム 600 の回路によって実装することができる。トランスポート・ロジック 610 は、実施形態では、ネットワーク・アダプター、ファームウェア、および接続メッセージを受信しこれらをエンジン 612 に転送するように構成することができるソフトウェアを含むことができる。図 6 に示すように、トランスポート・ロジック 610 は、実施形態では、セッション毎にプロトコル・スタック・インスタンスを含むことができる。一般に、各プロトコル・スタック・インスタンスは、ユーザー・インターフェースの出力をクライアントに導き、クライアントから受信したユーザー入力を、そのセッションに関連するセッション・コア 644 に導くように構成することができる。

【0045】

[0059] 図 6 の全体的な説明を続けると、本発明の実施形態例におけるエンジン 612 は、セッション要求を処理し、セッション毎に機能を決定し、1組の物理リソースをセッションに割り当てることによってセッションを生成し、そのセッションに対するプロトコル・スタック・インスタンスをインスタンス化するように構成することができる。実施形態では、以上で述べた動作手順の一部を実現する(implement)ことができる特殊回路コンポーネントによってエンジン 612 を実装することもできる。例えば、実施形態例における回路は、メモリと、エンジン 612 を実装するコードを実行するように構成されたプロセッサとを含むことができる。図 6 に示すように、場合によっては、エンジン 612 が接続要求を受けて、例えば、ライセンスが入手可能であり、その要求に対してセッションを生成できると判断することができる。コンピューター・システム 600 がリモート・デスクトップ能力を含むリモート・コンピューターであるという状況では、エンジン 612 が接続要求に応答して、ライセンスをチェックすることなく、セッションを生成するように構成することができる。図 6 に示すように、セッション・マネージャー 616 は、エンジン 612 からのメッセージを受信するように構成することができ、このメッセージに

【0046】

[0060] 図 6 に示すように、セッション・マネージャー 616 は、セッション・コア 644 のようなカーネル・モード部を含むことができるランタイム・サブシステム 640 のような、環境サブシステムをインスタンス化することができる。例えば、一実施形態における環境サブシステムは、サービスのある一部(subset)をアプリケーション・プログラムに露出させ、アクセス・ポイントをコンピューター・オペレーティング・システム 602 のカーネルに提供するように構成される。実施形態例では、ランタイム・サブシステム 640 がプロセスおよびスレッドの実行を制御することができ、セッション・コア 644 は、スレッドにメモリを割り当てこれらを実行する時間をスケジューリングする要求を、カーネル 614 のエグゼキューティブ(executive)に送ることができる。一実施形態では、セッション・コア 644 は、グラフィクス・ディスプレイ・インターフェース 646 (GDI)、セキュリティ・サブシステム 650、および入力サブシステム 652 を含むことができる。入力サブシステム 652 は、これらの実施形態では、セッションに関連するブ

ロトコル・スタック・インスタンスを通じてクライアントからユーザー入力を受信し、この入力をしかるべきセッションのためにセッション・コア 6 4 4 に送信するように構成することができる。ユーザー入力は、実施形態では、絶対および/または相対マウス移動コマンド、マウス座標、マウス・クリックを示す信号、キーボード信号、ジョイスティック移動信号等を含むことができる。ユーザー入力、例えば、アイコン上におけるマウスのダブル・クリックは、セッション・コア 6 4 4 によって受けることができ、入力サブシステム 6 5 2 は、アイコンがダブル・クリックに関連する座標に位置すると判断するように構成することができる。次いで、入力サブシステム 6 5 2 は、そのアイコンに関連するアプリケーションに対してプロセスを実行することができるという通知を、ランタイム・サブシステム 6 4 0 に送るように構成することができる。

10

【 0 0 4 7 】

[0061] クライアントから入力を受信することに加えて、アプリケーションおよび/またはデスクトップからドロワー・コマンドを受け取ることでもでき、このドロワー・コマンドを G D I 6 4 6 によって処理することができる。G D I 6 4 6 は、一般に、グラフィカル・オブジェクト・ドロワー・コマンドを生成することができるプロセスを含むことができる。この実施形態例における G D I 6 4 6 は、G D I 6 4 6 出力をリモート・ディスプレイ・サブシステム 6 5 4 に渡すように構成することができ、リモート・ディスプレイ・サブシステム 6 5 4 において、コマンドは、そのセッションに添付されたディスプレイ・ドライバに合わせてフォーマットされる。ある種の実施形態例では、1つ以上の物理ディスプレイをコンピューター・システム 6 0 0 に、例えば、リモート・デスクトップの状況において添付することができる。これらの実施形態例では、リモート・ディスプレイ・サブシステム 6 5 4 は、リモート・コンピューター・システムのディスプレイ・ドライバー(1つまたは複数)によってレンダリングされるドロワー・コマンドをミラーリングし(mirror)、このミラーリングした情報をクライアントに、セッションに関連するスタック・インスタンスを通じて送信するように構成することができる。他の実施形態例では、リモート・ディスプレイ・サブシステム 6 5 4 は、仮想ディスプレイ・ドライバー(1つまたは複数)を含むように構成することができる。仮想ディスプレイ・ドライバーは、コンピューター・システム 6 0 0 に物理的に添付されたディスプレイと関連付けられていなくてもよく、例えば、コンピューター・システム 6 0 0 がヘッドレスで(headless)で実行していることも可能である。この実施形態におけるリモート・ディスプレイ・サブシステム 6 5 4 は、1つ以上の仮想ディスプレイに対するドロワー・コマンドを受け、これらをクライアントに、セッションに関連するスタック・インスタンスを通じて送信するように構成することができる。本発明の一実施形態では、リモート・ディスプレイ・サブシステム 6 5 4 は、ディスプレイ・ドライバー毎に表示解像度を決定し、例えば、仮想ディスプレイに関連する仮想ディスプレイ・ドライバー(1つまたは複数)の表示解像度、または物理ディスプレイに関連するディスプレイ・ドライバーの表示解像度を決定し、関連するプロトコル・スタック・インスタンスを通じてパケットをクライアントに導くように構成することができる。

20

30

【 0 0 4 8 】

[0062] 実施形態例では、セッション・マネージャー 6 1 6 が加えてセッションのセッション識別子に関連するログオン・プロセスのインスタンスをインスタンス化することもできる。ログオン・プロセスは、セッションのログオンおよびログオフを扱うように構成することができる。これらの実施形態例では、ログオン・プロセスに関連するグラフィカル・ユーザー・インターフェースを示すドロワー・コマンドをクライアントに送信することができ、このクライアントのユーザーがアカウント識別子、例えば、ユーザー名/パスワードの組み合わせ、スマート・カード識別子、および/または生物計量情報をログイン画面に入力することができる。この情報は、コンピューター・システム 6 0 0 に送信され、エンジン 6 1 2 およびセッション・コア 6 4 4 のセキュリティ・サブシステム 6 5 0 に導くことができる。例えば、ある種の実施形態例では、エンジン 6 1 2 は、ユーザー・アカウントがライセンスと関連付けられているか否かを判断するように構成することができ、セ

40

50

セキュリティ・サブシステム 659 は、セッションに対してセキュリティ・トークンを生成するように構成することができる。

【0049】

【0063】 図 6 によって説明したように、リモート・サーバー・コンピューターは、多数のリモート・デスクトップ・セッションを接続リモート・クライアント・コンピューターに提供することができる。リモート・デスクトップ・セッションは、リモート・クライアント・コンピューターによって要求される 1 つ以上のアプリケーションと関連付けられてもよい。加えて、そして図 6 によって説明したように、リモート・サーバー・コンピューターは、ユーザー・インターフェース画面、ユーザー入力コマンド等のような、クライアント・デスクトップを表すグラフィックス・データーを処理することができる。更に、リモート・サーバー・コンピューターは、グラフィックス・データーをレンダリングし、キャプチャし、圧縮し、クライアント・リモート・コンピューターに送信することもできる。レンダリングとは、リモート・デスクトップ・セッション内において実行するアプリケーションによって行われる、回転する、ひっくり返す、および描画するというような、生の表示コール(raw display call)を変換するプロセスを言う。キャプチャとは、画面上ビットマップまたはフレーム変更というような、レンダリングされたアプリケーション・コンテンツを受け入れ(take)、このアプリケーション・コンテンツの以前のレンダリングに対する変更をインテリジェントにキャプチャすることを言う。圧縮は、エンコーディングとも呼ばれ、グラフィックス・リソースを、接続されているリモート・クライアント・コンピューターの各々に最適にそして公正に配信するプロセスを言う。ネットワーク状態の品質、およびターゲット・リモート・クライアント・コンピューターが、キャプチャされたコンテンツを最適に配信するために用いられる圧縮/エンコーディングのタイプを決定する。

【0050】

【0064】 本明細書において以下で更に説明するように、種々の実施形態では、リモート・サーバー・コンピューターが、計算サーバーおよびグラフィックス・サーバーを備えるのもよい。計算サーバーは、グラフィックス・データーをリモート・クライアント・コンピューターから受信し、このグラフィックス・データーを処理し、処理したグラフィックス・データーをグラフィックス・サーバーに送るように構成することができる。グラフィックス・サーバーは、計算サーバーから受信したデーターをレンダリングし、キャプチャし、そして圧縮してグラフィックス出力データーを得るように構成することができる。また、グラフィックス出力データーを計算サーバーに送り、計算サーバーを用いてグラフィックス出力データーをリモート・クライアント・コンピューターに送信する代わりに、グラフィックス・サーバーが、グラフィックス出力データーをリモート・クライアント・コンピューターに直接送信するように構成されてもよい。

【0051】

【0065】 図 7 は、計算サーバー 710 およびグラフィックス・サーバー 720 を備えるリモート・サーバー・コンピューター 700 の一実施形態例を示す。リモート・サーバー・コンピューター 700 の実施形態は、図 1 のコンピューター 100、図 2 のリモート・サーバー・コンピューター 220、図 3 のリモート・サーバー・コンピューター 320、図 4 および図 5 のコンピューター・システム 400、ならびに図 6 のコンピューター・システム 600 に関して説明したコンポーネントの一部または全部を実行することができる。

【0052】

【0066】 計算サーバー 710 の実施形態は、図 1 のコンピューター 100、図 2 のリモート・サーバー・コンピューター 220、図 4 および図 5 のコンピューター・システム 400 または仮想機械 440、ならびに図 6 のコンピューター・システム 600 に関して説明したコンポーネントの一部または全部を実行することができる。また、計算サーバー 710 は、GPU リソースが十分でなくても、または GPU リソースを有していなくてもよい。更に他の実施形態では、計算サーバー 710 が、以下で説明する計算リソースを設けるためにしかるべく構成された標準的なサーバー・コンピューターであってもよい。他の

実施形態では、計算サーバー 710 が特定機能に合わせて構成された計算デバイスであってもよい。例えば、計算サーバーが 1 つのタイプの処理ユニットと、少量のキャッシュ・メモリーのみを有するのでもよい。

【0053】

[0067] グラフィクス・サーバー 720 は、レンダリング、キャプチャし、および圧縮動作というような、グラフィクス動作のためにリソースを提供するように構成されるのでもよい。また、グラフィクス・サーバーに複数の GPU リソースが構成されてもよい。一実施形態では、グラフィクス・サーバー 720 が、図 1 のコンピューターに関して説明したコンポーネントの一部または全部を実行することができる。更に他の実施形態では、グラフィクス・サーバーが、図 4 のホスト 404 のような、ホスト・パーティション上にホストされるのでもよい。計算サーバー 710 およびグラフィクス・サーバー 720 が、ネットワーク（例えば、ファイバー・チャネル、LAN、ワイヤレス、イーサネット（登録商標）等）を通じて接続されてもよい。図 4 の環境のような仮想化環境では、グラフィクス・サーバー 720 および計算サーバー 710 が、VMBus を用いて接続されてもよい。

【0054】

[0068] 計算サーバー 710 は、1 つ以上のアプリケーション 712 を実行することもできる。一態様では、アプリケーションがグラフィクス・デバイス・ドライバ 714 と関連付けられてもよい。グラフィクス・デバイス・ドライバ 714、アプリケーション 712、および/または計算サーバー 710 は、グラフィクス・サーバー 720 上のグラフィクス・サーバー・マネージャー 740 と関連付けられてもよい。グラフィクス・デバイス・ドライバ 714、アプリケーション 712、および/または計算サーバー 710 は、命令およびデータをグラフィクス・サーバー・マネージャー 740 に送ること、ならびに命令およびデータをグラフィクス・サーバー・マネージャー 740 から受信することができる。一例として、グラフィクス・デバイス・ドライバ 714、アプリケーション 712、および/または計算サーバー 710 が、第 1 データをグラフィクス・サーバー・マネージャー 740 に送ることができるとよく、この第 1 データは GPU リソースの要求を示す。グラフィクス・サーバー・マネージャー 740 が、第 2 データをグラフィクス・デバイス・ドライバ 714、アプリケーション 712、および/または計算サーバー 710 に送るのでもよく、第 2 データは、グラフィクス・サーバー 720 からの GPU 命令に対するルーティング(routing)を示す。

【0055】

[0069] グラフィクス・サーバー・マネージャー 740 が、グラフィクス・サーバー 720 を管理することもできる。グラフィクス・サーバー・マネージャー 740 は、命令およびデータをグラフィクス・サーバー 720 のコンポーネントに送ることができるとよく、グラフィクス・サーバー 720 のコンポーネントからの応答として、情報およびデータを受信すればよい。グラフィクス・サーバー 720 は、GPU ホスティングおよび処理に特殊化されてもよい。グラフィクス・サーバー 720 は、グラフィクス・サーバー・マネージャー 740、プロキシー・グラフィクス・アプリケーション 722、カーネル 726、および GPU ハードウェア 730 を備えることができる。プロキシー・グラフィクス・アプリケーション 722 が、第 1 グラフィクス・デバイス・ドライバ 724 と関連付けられてもよく、カーネル 726 が、第 2 グラフィクス・デバイス・ドライバ 728 と関連付けられてもよい。グラフィクス・デバイス・ドライバ 724 および 728 は、グラフィクス処理タスクに関連するデータおよび情報を変換、受信、そして送信することができる。一実施形態では、グラフィクス・デバイス・ドライバ 724 および 728 は、特定の GPU ハードウェア 730 と、グラフィクス・サーバー・マネージャー 740、計算サーバー 710、および/またはリモート・クライアント・コンピューター上のアプリケーション、ハードウェア、およびオペレーティング・システムとの間において変換を行うために選択される。

【0056】

【0070】 図7の実施形態では、グラフィクス処理タスクに関連する命令は、アプリケーション712からグラフィクス・サーバー・マネージャー740までの一連のレイヤーを通して、プロキシー・グラフィクス・アプリケーション722に、カーネル726に、そしてハードウェア730まで流れることができる。処理された情報は、同じパスを逆に辿ればよい。図8は、情報パスについての代替実施形態を示す。グラフィクス・サーバー・マネージャー850は、計算サーバー710、アプリケーション712、および/またはグラフィクス・デバイス・ドライバ714からGPUリソース要求を受け、ルーティング命令、状態命令等を計算サーバー710およびグラフィクス・サーバー720に送る。その後、GPUタスク、処理された情報、および命令は、直接グラフィクス・サーバー720と計算サーバー710との間で送ることができる。グラフィクス・サーバー・マネージャー850は、これらの相互作用を監視することができ、GPUハードウェア730のようなGPU上におけるリソースの割り当てに関係する他のタスクも実行することができる。

10

【0057】

【0071】 図9は、複数のグラフィクス・サーバー720(A~N)を示す。これらは、1組のGPUハードウェア730に関連するリソースがGPU処理タスクを実行するには不十分であるときに用いることができる。また、この実施形態は、グラフィクス・サーバー・マネージャー850がGPU処理タスクの一部を第1グラフィクス・サーバー720Aから第2グラフィクス・サーバー720Bに移すときにも用いることができる。このような実施形態では、グラフィクス・サーバー・マネージャー850が、第1グラフィクス・サーバー720Aの状態を第2グラフィクス・サーバー720Bにコピーするように動作することもできる。

20

【0058】

【0072】 図10は、計算サーバー1010およびグラフィクス・サーバー1020が仮想化環境または仮想デスクトップ・インフラストラクチャ(VDI)において実現される(implement)、リモート・サーバー・コンピューター1000の一実施形態例を示す。計算サーバー1010およびグラフィクス・サーバー1020は、図7~図9の計算サーバー710およびグラフィクス・サーバー720を実装するように構成することができる。リモート・サーバー・コンピューター1000の実施形態は、図1のコンピューター100、図2のリモート・サーバー・コンピューター220、図3のリモート・サーバー・システム350、図4および図5のコンピューター・システム400、ならびに図6のコンピューター・システム600に関して説明したコンポーネントの一部または全部を実行することができる。

30

【0059】

【0073】 図10は、VDIに対するホスト・ベースのグラフィクス仮想化を示す。このようなアーキテクチャーは、例えば、Microsoft(登録商標)RemoteFX(登録商標)プラットフォームを用いて実現する(implement)ことができる。仮想機械1011(A~N)は、仮想デスクトップと呼ぶことができる。このアーキテクチャーは、仮想グラフィクス処理ユニット(vGPU)1016(A~N)を用い、ゲスト・オペレーティング・システム(OS)1014(A~N)と物理GPU112との間の関係を抽象化し、ホストされたマルチ・ユーザー環境においてGPUリソースを最適に共有する。

40

【0060】

【0074】 ハイパーバイザー・マイクロカーネル1002は、複数のホストおよびゲスト・パーティションを実現するように構成することができる。他の実施形態では、ハイパーバイザー・マイクロカーネル1002は、リモート・デスクトップ・セッション・コンポーネント(図示せず)を統合することができる。

【0061】

【0075】 各ホスト・パーティションは、リモート・コンピューター・サーバー1000の物理GPUリソースにアクセスするグラフィクス・サーバー1020として構成することができる。また、各ホスト・パーティションは、グラフィクスのレンダリング、キャプ

50

チャシ、およびエンコーディングのために管理コンポーネントも含むことができる。また、各ホスト・パーティションは、物理GPU112および、ASICS（図示せず）のようなホスト・ベース・エンコーダーにインターフェースを設けるデバイス・ドライバー1026も含むことができる。デバイス・ドライバー1026は、GPU、CPU、およびエンコーダー特定ドライバーを含むことができる。

【0062】

[0076] 各ゲスト・パーティションは、vGPUリソースにアクセスする計算サーバー1010として構成することができる。各ゲスト・パーティションは、複数の接続されたリモート・クライアント・コンピューター（図示せず）に対して1つ以上の仮想デスクトップまたはセッションを実装することができる。各ゲスト・パーティションとリモート・クライアント・コンピューターとの間の接続は、図2および図3に示したものと同様である、RDP7.1のようなリモート・デスクトップ・セッションを含むことができる。

10

【0063】

[0077] vGPU1016は、各仮想機械1011にインストールされる仮想グラフィクス・アダプターを設けることができる。vGPU1016は、1つ以上のGPU112を利用する多数の仮想機械に対するグラフィック処理を抽象化することができる。仮想機械1011において実行するアプリケーションが、DirectX（登録商標）またはGDI動作のようなグラフィクス動作を呼び出すとき、vGPU1016は、ゲスト・パーティション1010とホスト・パーティション1020との間の通信チャネルを用いて、GPU112からリソースを得ることができる。この通信チャネルは、VMBusまたはTCP/IPチャネルを含むことができる。VMBusは、メモリー共有およびvGPUに特定の他の機能のために、ハイパーバイザー・マイクロカーネル1002内部に構成することができる。このようなVMBus構成は、統合メカニズムを直接ハイパーバイザー・マイクロカーネル1002内に設けることができ、グラフィクス関連デバイスを求める全てのリソース要求を転送することができる。

20

【0064】

[0078] また、vGPU1016は、ある品質のサービス・メカニズムを仮想機械1011に提供することもできる。サービス・メカニズムの品質は、GPU112を最も効率的に利用する負荷均衡方針に基づいて、公平にGPU112のリソースを仮想機械1011に配分する(deliver)ことができる。

30

【0065】

[0079] ホスト・パーティション1020は、リモート・デスクトップ仮想グラフィクス管理(RDVGM)機能を仮想機械1011に設けるように構成することができる。RDVGMは、リモート・コンピューター・サーバー1000の物理リソース間におけるリソース割り当ておよびプロセス制御と、各仮想機械のゲスト・オペレーティング・システム(OS)1014へのvGPU1016のリソース割り当てを管理することができる。RDVGM機能は、レンダリング、キャプチャし、および圧縮(RCC)プロセスを管理すること、GPU112リソースを仮想機械1011にvGPU1016を介して割り当てること、仮想機械1011に対するリソース方針を指定すること、および多数の仮想機械1011(A~N)に跨がってGPU112リソースの負荷均衡を行うことを含むことができる。また、RDVGMは、ブートの時点においてしかるべきGPU112リソースを仮想機械1011(A~N)に割り当てることもできる。

40

【0066】

[0080] RDVGMは、RCCエンジン1022を統合することができる。RCCエンジン1022は、グラフィクス・データーのレンダリング、キャプチャし、および圧縮を扱う。RCCは、グラフィクス要求を各仮想機械1011からの出力として受け、これらの要求を、例えば、ホスト・パーティション1020上のDirectX（登録商標）に準拠したコマンドに変換することができる。VMBusは、仮想機械1011(A~N)において実行する、ホストされたアプリケーション1012(A~N)からの物理GPU112のリソースに対するグラフィクス要求のために、高速通信バックプレーンを提供することがで

50

きる。DirectX(登録商標)準拠コマンドに対して、アプリケーション1012(A~N)は、DirectX(登録商標)9以降をサポートする必要がある、一方GPU112はDirectX(登録商標)10以降をサポートする必要がある。

【0067】

[0081] 既に説明したように、レンダリングとは、vGPU1016(A~N)を介してアプリケーション1012(A~N)によって行われる、回転する、ひっくり返す、および描画するというような生の表示コールを変換し、これらの要求をGPU112に与え(honoring)、このようにしてアプリケーション・コンテンツをレンダリングするプロセスを指す。レンダリングは、標準的なDirectX(登録商標)のシンタックスに基づくことができる。キャプチャは、画面上ビットマップまたはフレーム変更というような、レンダリングされたアプリケーション・コンテンツを受け入れ(take)、このアプリケーション・コンテンツの以前のレンダリングに対する変更をインテリジェントにキャプチャすることを言う。キャプチャの二次的機能は、キャプチャ品質およびエンコーディング・レベルに対してサービス方針の品質を指定することである。圧縮またはエンコーディングは、GPU112のリソースをリモート・クライアント・コンピュータに、例えば、リモート・デスクトップ・セッション・プロトコルを含む通信チャンネルを通じて、vGPU1016によって最適にそして公正に配信するプロセスを言う。通信チャンネルの品質および状態、ならびに目標とされるリモート・クライアント・コンピュータのタイプが、キャプチャされたコンテンツを最適に配信するために用いられる圧縮/エンコーディングのタイプを決定する。

【0068】

[0082] VM1011内において実行するアプリケーション1012が、描画、サイズ変更、および回転というような表示コールを発行するとき、vGPU1016は全てのレンダリング要求を仲介することができる。仮想化パスは、ゲストOS1014に対して透過的であることができる。既に説明したように、グラフィクス処理コマンドは、ホスト・パーティション1020が横取りすることができる。この横取りは、ソフトウェア・スタックにおける低いレベルで行うことができる。次いで、GPU112上においてグラフィクスを1つのフレーム・バッファにレンダリングすることができる。この1つのフレーム・バッファは、グラフィック更新のための一時的保存場所(holding station)として役割を果たす。フレーム・バッファは、エンド・ユーザーの仮想化表示を表すことができ、エンド・ユーザーとは、リモート・クライアント・コンピュータを用いて仮想機械1011に接続されたユーザーである。豊富なグラフィクス・アプリケーション、3Dプラグイン、ならびにその他のグラフィクス・コールおよびコマンドは、GPUを内蔵した専用ワークステーション上でアプリケーションが実行しているかのように、正確に実行することができる。

【0069】

[0083] ホスト・パーティション1020は、素早くそして効率的に、レンダリングされたコンテンツをキャプチャすることができる。コンテンツ内にある各フレームは、管理可能な単位に分割することができる。フレーム内における変化領域は、RCCエンジン1022によって設けられる最適化能力によって処理することができる。このキャプチャ・メカニズムによって、個々のフレームが表示変化のために横取りされる。フレーム内において変化した領域のみが、エンコーディングのためにキャプチャされる。

【0070】

[0084] また、ホスト・パーティション1020は、キャプチャされたコンテンツを圧縮することもできる。圧縮プロセスは、ある種の仮想機械に与える優先順位を高くまたは低くして、あるいはキャプチャされたフレーム内における変化領域のサイズによって動的に、管理ツールによって制御することができる。加えて、RDP7.1のようなリモート・デスクトップ・セッションを用いる実施形態では、このリモート・デスクトップ・セッションが、ネットワークの使用および公正さに基づいて、フレーム・レートの最適化に備

えることができる。R C Cエンジン1022は、リモート・デスクトップ・セッション・リスナー・プロセス(listener process)に接触し(reach out to)、リモート・クライアント・コンピューターのデコーディング能力を含む、リモート・クライアント・コンピューターの状態を評価することができる。フレーム・バッファに対する変更は、フレーム・レートでリモート・クライアント・コンピューターに送ることができ、フレーム・レートはネットワーク状態およびリモート・クライアント・コンピューターの変更を消費する能力に動的に適合する。エンコードされた出力は、リモート・デスクトップ・セッション内をくぐり抜けることができ、リモート・クライアント・コンピューターに送出することができる。

【0071】

[0085] 1つ以上の計算サーバーおよび1つ以上のグラフィクス・サーバーを備えたりリモート・サーバー・コンピューターを実現するための、図7～図10に示した全てのアーキテクチャの変形は、実施態様の例である。ここでは、いかなることであっても、本開示をいずれの特定の実施態様(implementation aspect)に限定するように解釈してはならない。

【0072】

[0086] リモート計算環境では、リモート・クライアント・コンピューターは通例コンピューター・サーバーのみと通信し、グラフィクス・サーバーとの直接的な接続を有さない。コンピューター・サーバーとの通信は、図3に示したものと同様のリディレクターおよび/またはブローカーの使用を含むことができる。計算サーバーは、通例、リモート・クライアント・コンピューターとの接続を管理する。例えば、計算サーバーのゲスト・オペレーティング・システムは、クライアントを認証するように構成することができる。一旦認証が完了したなら、リモート・クライアント・コンピューターと計算サーバーとの間における転送を開始することができる。したがって、リモート・クライアント・コンピューターからのグラフィクスの要求は、計算サーバーによって受けることができる。計算サーバーはこのグラフィクス要求を処理してグラフィクス・コールおよびコマンドを得て、レンダリング、キャプチャ、および圧縮のために、これらのコールおよびコマンドをグラフィクス・サーバーに送信することができる。グラフィクス・サーバーはリモート・クライアント・コンピューターとの直接的な通信パスを有さないため、グラフィクス・サーバーからのグラフィクス出力は、計算サーバーに送ることができる。計算サーバーは、このグラフィクス出力をパッケージ化し、エンド・ユーザーに対する表示のために、リモート・クライアント・コンピューターに送信することができる。

【0073】

[0087] 以上の説明を図10の仮想環境において適用すると、リモート・クライアント・コンピューターからのグラフィクス要求は、リモート・サーバー・コンピューター1000の物理NIC114上を、ゲスト・パーティション1010の仮想NIC(図示せず)を通してアプリケーション1012に送信される。次いで、グラフィクス要求は処理されて、レンダリング、キャプチャ、および圧縮のためにホスト・パーティション1012に導くことができる。一旦レンダリングされ、キャプチャされ、圧縮されたなら、出力データをホスト・パーティション1020から計算サーバー1010に、vGPU1016を用いて逆に導くことができる。出力データをゲスト・パーティション1010からリモート・クライアント・コンピューターに送信するために、ゲスト・パーティション1010は、データをパッケージ化し仮想NICを用いて送信することができる。仮想NICは、出力データを物理NIC114にリディレクトし、物理NIC114はこのデータをネットワークを通じてリモート・クライアント・コンピューターに送信することができる。既に説明したように、ゲスト・パーティション1010、ホスト・パーティション1020、およびリモート・サーバー・コンピューター1000の基礎リソース間において繰り返されるデータの横断は、集中的動作を必要とする可能性があり、大量のメモリおよびCPUリソースを消費する可能性があり、リモート・クライアント・コンピューターに対するデータ配信レイテンシが増大する可能性がある。

10

20

30

40

50

【 0 0 7 4 】

[0088] 図 1 1 は、リモート・サーバー・コンピューター 1 1 0 0 のリソース間における繰り返しデータ横断を解消する代替アーキテクチャーを示す。図 1 1 は、リモート・クライアント・コンピューター 1 1 3 0 との接続の管理を計算サーバー 1 1 1 0 内に保持し、グラフィクス・サーバー 1 1 2 0 が、グラフィクス出力データを直接リモート・クライアント・コンピューター 1 1 3 0 にストリーミングすることを可能にするアーキテクチャーについて説明する。リモート・サーバー・コンピューター 1 1 0 0 の実施形態は、図 7 ~ 図 9 のリモート・サーバー・コンピューター 7 0 0 および図 1 0 のリモート・サーバー・コンピューター 1 0 0 0 に関して説明したコンポーネントの一部または全部を実行することができる。リモート・クライアント・コンピューター 1 1 3 0 の実施形態は、図 1 のコンピューター 1 0 0、図 2 のリモート・クライアント・コンピューター 2 1 0、および図 3 のリモート・クライアント・コンピューター 3 1 0 に関して説明したコンポーネントの一部または全部を実行することができる。

10

【 0 0 7 5 】

[0089] リモート・クライアント・コンピューター 1 1 3 0 は、ネットワーク（図示せず）を通じた計算サーバー 1 1 1 0 との接続 1 1 0 5 を開始することができる。接続 1 1 0 5 は、TCP/IP に基づくことができ、RDP 7.1 のようなリモート・デスクトップ・セッションを含むことができる。計算サーバー 1 1 1 0 は、リモート・クライアント・コンピューター 1 1 3 0 との接続 1 1 0 5 を認証することができる。認証方法は、リモート・クライアント・コンピューター 1 1 3 0 の仮想デスクトップまたはセッションが計算サーバー 1 1 1 0 において設定される前に、リモート・クライアント・コンピューター 1 1 3 0 が認証されることを要求することができる。例えば、認証方法は、RDP 7.1 において利用可能なネットワーク・レベルの認証とすることができる。加えて、計算サーバー 1 1 1 0 は、リモート・クライアント・コンピューター 1 1 3 0 の IP アドレスのようなアドレスを取得することができる。

20

【 0 0 7 6 】

[0090] 一旦認証が完了し、リモート・クライアント・コンピューター 1 1 3 0 と計算サーバー 1 1 1 0 との間に接続 1 1 0 5 が確立されたなら、計算サーバー 1 1 1 0 は、仮想機械内においてリモート・デスクトップ・セッション、仮想機械、デスクトップ・セッション、またはその組み合わせを開始することができる。これらはリモート・クライアント・コンピューター 1 1 3 0 のために計算サーバー 1 1 1 0 内部で実行する。このような仮想機械またはデスクトップ・セッションは、図 2 ~ 図 1 0 の技法を具体化することができる。加えて、計算サーバー 1 1 1 0 は、グラフィクス・サーバー 1 1 2 0 を初期化することができる。この初期化は、例えば、グラフィクス・サーバーがインアクティブ状態にある場合にはグラフィクス・サーバー 1 1 2 0 を復元または起動し、計算サーバー 1 1 1 0 とグラフィクス・サーバー 1 1 2 0 との間に接続 1 1 1 5 を確立することを含むことができる。接続 1 1 1 5 は、TCP/IP に基づくことができ、または仮想環境における VM Bus を用いることができる。また、計算サーバー 1 1 1 0 は、リモート・クライアント・コンピューター 1 1 3 0 のアドレスをグラフィクス・サーバー 1 1 2 0 に供給し、グラフィクス・サーバー 1 1 2 0 に、リモート・クライアント・コンピューター 1 1 3 0 のアドレスから発する接続要求に対する準備をするように命令することができる。また、計算サーバーは、グラフィクス・サーバー 1 1 2 0 の、IP アドレスのような、アドレスを取得することができ、グラフィクス・サーバー 1 1 2 0 のアドレスをリモート・クライアント・コンピューター 1 1 3 0 に供給することができる。

30

40

【 0 0 7 7 】

[0091] 一旦グラフィクス・サーバー 1 1 2 0 が初期化され、リモート・クライアント・コンピューター 1 1 3 0 がグラフィクス・サーバー 1 1 2 0 のアドレスを取得したなら、クライアント・コンピューター 1 1 3 0 はネットワーク（図示せず）を通じてグラフィクス・サーバー 1 1 2 0 との接続 1 1 2 5 を開始することができる。接続 1 1 2 5 は、TCP/IP に基づくことができる。また、接続 1 1 2 5 は、RDP 7.1 のような、リモ

50

ート・デスクトップ・セッションを含むこともできる。他の実施形態では、接続 1 1 0 5 および接続 1 1 2 5 がリモート・デスクトップ・セッションを含む場合、リモート・デスクトップ・セッションは別個であることができ、または接続 1 1 0 5 および接続 1 1 2 5 に跨がって同一であることもできる。

【 0 0 7 8 】

[0092] 一旦接続 1 1 0 5、1 1 1 5、および 1 1 2 5 が確立されたなら、リモート・クライアント・コンピューター 1 1 3 0 は、キーボードまたはマウス入力のような、エンドユーザー・グラフィクス入力をコンピューター・サーバー 1 1 1 0 に送ることができる。コンピューター・サーバーは、図 2、図 6 ~ 図 1 0 の技法を用いて、グラフィクス入力を処理することができる。一実施形態では、計算サーバー 1 1 1 0 はエンド・ユーザーのキーボード押鍵またはアイコン上のマウス・クリックを、アプリケーションのインスタンスを実装する表示コマンドに、そして回転する、ひっくり返す、サイズ変更する、および描画するというような表示コールに変換することができる。計算サーバー 1 1 1 0 は、表示コマンドおよびコール・データーを、レンダリング、キャプチャ、および圧縮のために、グラフィクス・サーバー 1 1 2 0 に送ることができる。グラフィクス・サーバー 1 1 2 0 は、表示コマンドおよびコール・データーをレンダリングし、キャプチャし、圧縮することができる、そして出力を出力グラフィクス・データーとしてエンコードすることができる。出力グラフィクス・データーを計算サーバー 1 1 1 0 に送る代わりに、グラフィクス・サーバー 1 1 2 0 は出力グラフィクス・データーを直接リモート・クライアント・コンピューター 1 1 3 0 に、接続 1 1 2 5 を通して送信することができる。リモート・クライアント・コンピューター 1 1 3 0 は、エンド・ユーザーへの表示のために、出力グラフィクス・データーをデコードすることができる。

【 0 0 7 9 】

[0093] 一実施形態では、図 1 0 のリモート・サーバー・コンピューター 1 0 0 0 のように、リモート・サーバー・コンピューター 1 1 0 0 が仮想化される場合、計算サーバー 1 1 1 0 をゲスト・パーティション上に実装することができ、グラフィクス・サーバー 1 1 2 0 をホスト・パーティション上に実装することができる。各パーティションは、IP アドレスのようなアドレスを有するように構成することができる。ゲスト・パーティションまたは計算サーバー 1 1 1 0 のアドレスは、仮想 NIC と関連付けることができる。ホスト・パーティションまたはグラフィクス・サーバー 1 1 2 0 のアドレスは、仮想 NIC またはリモート・サーバー・コンピューター 1 1 0 0 の基礎となるハードウェアの物理 NIC 1 1 4 と関連付けることができる。計算サーバー 1 1 1 0 は、その中で実行する仮想機械またはセッションをリモート・クライアント・コンピューター 1 1 3 0 に提示することができる。計算サーバー 1 1 1 0 とグラフィクス・サーバー 1 1 2 0 との間の接続 1 1 1 5 は、V M B u s のようなパーティション内通信チャネルを構成することができる。

【 0 0 8 0 】

[0094] リモート・クライアント・コンピューター 1 1 3 0 におけるエンド・ユーザー入力のようなグラフィクス・データーは、リモート・クライアント・コンピューター 1 1 3 0 から接続 1 1 0 5 を通って、コンピューター・サーバー 1 1 1 0 の仮想 NIC と関連する計算サーバー 1 1 1 0 のアドレスに送ることができる。

【 0 0 8 1 】

[0095] 計算サーバー 1 1 1 0 は、受信したグラフィクス・データーを処理して、先に説明したように、表示コマンドおよびコール・データーを得ることができる。計算サーバー 1 1 1 0 は、処理したグラフィクス・データーを、レンダリング、キャプチャ、および圧縮のためにグラフィクス・サーバー 1 1 2 0 に送ることができる。一実施形態では、処理したデーターを送るとき、処理したデーターを、計算サーバー 1 1 1 0 に割り当てられたメモリー空間から、グラフィクス・サーバー 1 1 2 0 に割り当てられたメモリー空間に送信することを含むことができる。他の実施形態では、リモート・サーバー・コンピューター 1 1 0 0 内部のパーティションがメモリー空間を共有することができる。このような実施形態では、計算サーバー 1 1 1 0 からグラフィクス・サーバー 1 1 2 0 にデーターを

送るとき、2つのメモリー空間の間で処理したデータをコピーするのではなく、処理したデータを収容するメモリー空間のアドレスを送出することを含むことができる。例えば、計算サーバー1110およびグラフィクス・サーバー1120は、処理したデータを導くために、VMBus内部の共有メモリー空間を利用することができる。

【0082】

[0096] グラフィクス・サーバー1120は、先に説明したように、処理したデータをレンダリングし、キャプチャし、圧縮することができる。グラフィクス・サーバー1120は、レンダリングし、キャプチャし、圧縮したデータをエンコードして出力データを得ることができ、グラフィクス・サーバーNICを用いて、この出力データを、リモート・クライアント・コンピューター1130に関連するアドレスに送信することができる。リモート・クライアント・コンピューター1130は、受信した出力データを、エンド・ユーザーへの表示のためにデコードすることができる。

10

【0083】

[0097] 図12は、複数のリモート・クライアント・コンピューター1130(A~N)、複数の計算サーバー1110(A~N)、および複数のグラフィクス・サーバー1120(A~N)を備える環境における、図11のアーキテクチャーの一実施形態を示す。尚、リモート・クライアント・コンピューター、計算サーバー、およびグラフィクス・サーバーの数は、等しい必要はないことを注記するのは重要である。言い換えると、リモート・クライアント・コンピューター1130(A~N)と計算サーバー1110(A~N)との間の関係、計算サーバー1110(A~N)とグラフィクス・サーバー1120(A~N)との間の関係、およびグラフィクス・サーバー1120(A~N)とリモート・クライアント・コンピューター1130(A~N)との間の関係は、1対1、1対多数、またはその組み合わせとすることができる。

20

【0084】

[0098] リモート・サーバー・コンピューター1200の実施形態は、図2のリモート・サーバー・コンピューター220、図3のリモート・サーバー・コンピューター320、図7~図10のリモート・サーバー・コンピューター700、図10のリモート・サーバー・コンピューター1000、および図11のリモート・サーバー・コンピューター1100に関して説明したコンポーネントの一部または全部を実行することができる。一実施形態では、リモート・サーバー・コンピューター1200は複数の物理計算システムを備えることができる。リディレクターおよびブローカー1240は、図3のリディレクター330およびブローカー340に関して説明したコンポーネントの一部または全部を実行することができる。グラフィクス・サーバー・マネージャー1250は、図7のグラフィクス・サーバー・マネージャー740、および図8~図9のグラフィクス・サーバー・マネージャー850に関して説明したコンポーネントの一部または全部を実行することができる。加えて、リディレクターおよびブローカー1240は、計算サーバー111(A~N)と統合することができる。同様に、グラフィクス・サーバー・マネージャー1250もグラフィクス・サーバー1120(A~N)と統合することができる。他の実施形態では、リモート・サーバー・コンピューターが仮想化されてもよく、図4~図5の計算システム400、図10のリモート・サーバー・コンピューター1000、および図11のリモート・サーバー・コンピューター1100に関して説明したコンポーネントの一部または全部を実行することができる。このような実施形態では、計算サーバー1110(A~N)は、1つ以上のゲスト・パーティション上に実装することができ、グラフィクス・サーバー1120(A~B)は、1つ以上のホスト・パーティション上に実装することができる。リディレクターおよびブローカー1240ならびにグラフィクス・サーバー・マネージャー1250も、1つ以上のゲスト・パーティションおよび/または1つ以上のホスト・パーティションにおいて、既に説明した技法を用いて仮想化することもできる。更に他の実施形態では、リモート・サーバー・コンピューター1200は、仮想機械および物理機械の組み合わせとすることもできる。例えば、計算サーバー1110(A~N)およびグラフィクス・サーバー1120(A~N)が仮想化されてもよく、一方リディレク

30

40

50

ターおよびブローカー 1 2 4 0 ならびにグラフィクス・サーバー・マネージャー 1 2 5 0 が物理計算デバイスであってもよい。

【 0 0 8 5 】

[0099] リディレクターおよびブローカー 1 2 4 0 は、リモート・クライアント・コンピューター 1 1 3 0 と計算サーバー 1 1 1 0 との間に第 1 接続を確立するために用いられてもよい。例えば、リディレクターおよびブローカー 1 2 4 0 は、複数の計算サーバー 1 1 1 0 (A) の中から計算サーバー 1 1 1 0、および/またはそこにおけるセッションをリモート・クライアント・コンピューター 1 1 3 0 に割り当てることができる。したがって、リディレクターおよびブローカー 1 2 4 0 は、計算サーバー 1 1 1 0 (A) の可用性に関して負荷均衡技法を設けることができる。一旦その割り当てが完了したなら、割り当てられた計算サーバー 1 1 1 0 は、リモート・クライアント・コンピューター 1 1 3 0 を認証することができる。リモート・クライアント・コンピューター 1 1 3 0 および割り当てられた計算サーバー 1 1 1 0 は、既に説明したように第 1 接続を確立することができる。例えば、第 1 接続は、図 1 1 の接続 1 1 0 5 を構成することができる。また、割り当てられた計算サーバー 1 1 1 0 は、リモート・クライアント・コンピューター 1 1 3 0 のアドレスを取得することもできる。

10

【 0 0 8 6 】

[0100] グラフィクス・サーバー・マネージャー 1 2 5 0 は、割り当てられた計算サーバー 1 1 1 0 と、複数のグラフィクス・サーバー 1 1 2 0 (A ~ N) からのグラフィクス・サーバー 1 1 2 0 との間に第 2 接続を確立するために用いられてもよい。グラフィクス・サーバー・マネージャー 1 2 5 0 は、負荷均衡技法を実行して、グラフィクス・サーバー 1 1 2 0 (A ~ N) の可用性を判断することができる。これに応じて、グラフィクス・サーバー・マネージャー 1 2 5 0 は、次に、グラフィクス・サーバー 1 1 2 0 を、割り当てられたコンピューター・サーバー 1 1 1 0 との接続に割り当ててもよい。一旦この割り当てが完了すると、第 2 接続を確立することができる。例えば、第 2 接続は図 1 1 の接続 1 1 1 5 を構成することができる。

20

【 0 0 8 7 】

[0101] 割り当てられた計算サーバー 1 1 1 0 は、リモート・クライアント・コンピューター 1 1 3 0 のアドレスをグラフィクス・サーバー 1 1 2 0 に供給することができ、割り当てられたグラフィクス・サーバー 1 1 2 0 のアドレスを取得して、リモート・クライアント・コンピューター 1 1 3 0 に供給することができる。リモート・クライアント・コンピューター 1 1 3 0 および割り当てられたグラフィクス・サーバー 1 1 1 2 0 は、リモート・クライアント・コンピューター 1 1 3 0 および割り当てられたグラフィクス・サーバー 1 1 2 0 の取得したアドレスを用いて、第 3 接続を確立することができる。第 3 接続は、図 1 1 の接続 1 1 2 5 を含むことができる。

30

【 0 0 8 8 】

[0102] 更に、図 1 2 のアーキテクチャー上で負荷均衡技法を実行することができる。例えば、グラフィクス・サーバー 1 1 2 0 (A ~ N) の負荷均衡の場合、割り当てられたグラフィクス・サーバー 1 1 2 0 がもはやリモート・クライアント・コンピューター 1 1 3 0 に適正に配給する(serve)ことができないとき、計算サーバー 1 1 1 0 は利用可能なグラフィクス・サーバー 1 1 2 0 との新たな接続を確立し、リモート・クライアント・コンピューター 1 1 3 0 に、利用可能なグラフィクス・サーバー 1 1 2 0 との対応する新たな接続を確立するように命令することができる。リモート・クライアント・コンピューター 1 1 3 0 は、次いで、利用可能なグラフィクス・サーバー 1 1 2 0 に継ぎ目なく移行することができる。加えて、計算サーバー 1 1 1 0 (A ~ N) の負荷均衡の場合、割り当てられた計算サーバー 1 1 1 0 がもはやリモート・クライアント・コンピューター 1 1 3 0 に適正に配給することができないとき、リモート・クライアント・コンピューター 1 1 3 0 と利用可能な計算サーバー 1 1 1 0 との間、および利用可能な計算サーバー 1 1 1 0 と既に割り当てられているグラフィクス・サーバー 1 1 2 0 との間に新たな接続を確立することができ、一方リモート・クライアント・コンピューター 1 1 3 0 と既に割り当てられ

40

50

ているグラフィクス・サーバー 1 1 2 0 との間の接続は同じままで有り続けることができる。仮想機械ライブ・マイグレーションのような技法は、作業負荷を 1 つの計算サーバー 1 1 1 0 から他の計算サーバーに継ぎ目なく転移させることができ、一方グラフィクス・サーバー 1 1 2 0 の作業負荷は同じままで有り続けることができる。

【 0 0 8 9 】

[0103] 一実施形態では、1 つ以上のリモート・クライアント・コンピューター 1 1 3 0 が種々のグラフィクス・サーバー 1 1 2 0 から表示ストリームまたはグラフィクス出力データを受け入れることを可能にする技法を用いることができる。例えば、リモート・クライアント・コンピューター 1 1 3 0 A は他のリモート・クライアント・コンピューター 1 1 3 0 (B ~ N) から、他のリモート・クライアント・コンピューター 1 1 3 0 (B ~ N) に関連するエンド・ユーザー表示を見る許可を得ることができる。この許可は、例えば、他のリモート・クライアント・コンピューター 1 1 3 0 (B ~ N) の各々に割り当てられた種々のグラフィクス・サーバー 1 1 2 0 のアドレスを含むことができる。加えて、この許可は、他のリモート・クライアント・コンピューター 1 1 3 0 (B ~ N) のアドレス、他のリモート・クライアント・コンピューター 1 1 3 0 (B ~ N) の各々に割り当てられた計算サーバー 1 1 1 0 のアドレス、対応するセッション ID、ユーザー名、計算サーバー 1 1 1 0 およびグラフィクス・サーバー 1 1 2 0 の名称、各計算サーバー 1 1 1 0 におけるアクティブなセッションの数等も含むことができる。リモート・クライアント・コンピューター 1 1 3 0 A は、この許可を用いて、他のリモート・クライアント・コンピューター 1 1 3 0 (B ~ N) の各々に割り当てられたグラフィクス・サーバー 1 1 2 0 との接続を確立することができる。これらの接続は、図 1 1 の接続 1 1 2 5 を含むことができる。一旦接続が確立されると、種々のグラフィクス・サーバー 1 1 2 0 は、他のリモート・クライアント・コンピューター 1 1 3 0 (B ~ N) のグラフィクス・データ出力をリモート・クライアント・コンピューター 1 1 3 0 A に送信することができる。リモート・クライアント・コンピューター 1 1 3 0 A は、種々のグラフィクス・サーバー 1 1 2 0 から受信したグラフィクス出力データをデコードすることができ、デコードしたデータをエンド・ユーザーに表示することができる。例えば、リモート・クライアント・コンピューター 1 1 3 0 A は、デコードしたデータをタイル状に並べて表示することができ、または他のリモート・クライアント・コンピューター 1 1 3 0 (B ~ N) と関連する画面の各々をユーザーにばらばらとめくらせることができる。更に、リモート・クライアント・コンピューター 1 1 3 0 A は、ユーザー名、対応する計算サーバー 1 1 1 0 およびグラフィクス・サーバー 1 1 2 0 の名称等というような、許可において入手可能な他の情報を表示することもできる。他の実施形態では、リモート・クライアント・コンピューター 1 1 3 0 (B ~ N) に対応する種々のグラフィクス出力データを送信するグラフィクス・サーバー 1 1 2 0 が、種々のグラフィクス出力データおよび対応するリモート・クライアント・コンピューター 1 1 3 0 (B ~ N) についての他の情報を他のリモート・クライアント・コンピューター 1 1 3 0 A に送信するように構成することもできる。したがって、リモート・クライアント・コンピューター 1 1 3 0 A は、リモート・クライアント・コンピューター 1 1 3 0 (B ~ N) の画面およびこれらについての情報を表示することができる。

【 0 0 9 0 】

[0104] 図 1 3 は、クライアントに送信するためにグラフィクス・データを処理する動作手順の一例を示す。この動作手順は、動作 1 3 0 0、1 3 1 0、1 3 2 0、1 3 3 0、1 3 4 0、1 3 5 0、および 1 3 6 0 を含む。動作 1 3 0 0 は、この動作手順を開始し、動作 1 3 1 0 は、リモート・クライアントと計算サーバーとの間に第 1 接続を確立することを示す。第 1 接続は、図 1 1 の接続 1 1 0 5 を含むことができる。動作 1 3 2 0 は、計算サーバーとグラフィクス・サーバーとの間に第 2 接続を確立することを示す。第 2 接続は、図 1 1 の接続 1 1 1 5 を含むことができる。動作 1 3 3 0 は、リモート・クライアントとグラフィクス・サーバーとの間に第 3 接続を確立することを示す。第 3 接続は、図 1 1 の接続 1 1 2 5 を含むことができる。動作 1 3 4 0 は、第 1 接続を通じてリモート・

クライアントからグラフィクス・データーを計算サーバーが受信することを示す。受信されたグラフィクス・データーは、例えば、リモート・クライアントに関連するキーボード打鍵またはマウス・クリックというような、ユーザー・グラフィクス入力を含むことができる。動作 1 3 5 0 は、受信したグラフィクス・データーを処理し、処理したグラフィクス・データーをグラフィクス・サーバーに第 2 接続を通じて送ることを示す。受信したグラフィクス・データーの処理は、計算サーバーが受信したグラフィクス・データーを表示コマンドおよびコールに変換することを含むことができる。動作 1 3 6 0 は、処理したグラフィクス・データーをレンダリングし、キャプチャし、リモート・クライアントに第 3 接続を通じて送信することを示す。

【 0 0 9 1 】

[0105] 図 1 4 は、前述のように、クライアント・コンピューターに送信するためにグラフィクス・データーを処理するシステム例を示す。システム 1 4 0 0 は、プロセッサ 1 4 1 0 およびメモリー 1 4 2 0 を備えている。一実施形態では、プロセッサ 1 4 1 0 は、図 1 における論理処理ユニット 1 0 2 として実現することができ、一方メモリー 1 4 2 0 は図 1 におけるシステム・メモリー 2 2 のコンポーネントの一部または全部を有するものとして実現することができる。更に、メモリー 1 4 2 0 は、リモート・クライアントへの送信のためにグラフィクス・データーをシステムに処理させるように構成されたコンピューター命令も備えている。ブロック 1 4 2 2 は、リモート・クライアントと計算サーバーとの間に第 1 接続を確立することを示す。ブロック 1 4 2 4 は、計算サーバーとグラフィクス・サーバーとの間に第 2 接続を確立することを示す。ブロック 1 4 2 6 は、リモート・クライアントとグラフィクス・サーバーとの間に第 3 接続を確立することを示す。ブロック 1 4 2 8 は、リモート・クライアントからグラフィクス・データーを第 1 接続を通じて計算サーバーが受信することを示す。ブロック 1 4 3 0 は、受信したグラフィクス・データーを処理し、処理したグラフィクス・データーをグラフィクス・サーバーに第 2 接続を通じて送ることを示す。ブロック 1 4 3 2 は、処理したグラフィクス・データーをレンダリングし、キャプチャし、圧縮し、リモート・クライアントに第 3 接続を通じて送信することを示す。

【 0 0 9 2 】

[0106] 以上で述べた態様は、いずれも、方法、システム、コンピューター読み取り可能媒体、またはあらゆるタイプの製造物(manufacture)で実現することができる。

[0107] 以上の詳細な説明では、例および/または動作図によって、本システムおよび/またはプロセスの種々の実施形態を明記した。このようなブロック図、および/または例が 1 つ以上の機能および/または動作を含む限りにおいて、このようなブロック図または例の中にある各機能および/または動作は、広い範囲のハードウェア、ソフトウェア、ファームウェア、または事実上そのあらゆる組み合わせによって、個々におよび/または集合的に実現できることは、当業者には理解されよう。

【 0 0 9 3 】

[0108] 尚、本命書において説明した種々の技法は、ハードウェアまたはソフトウェア、あるいははしかるべきときには双方の組み合わせに関して実現できることは言うまでもない。つまり、本開示の方法および装置、あるいはその一部のある種の態様は、フロッピー(登録商標)・ディスクット、CD-ROM、ハード・ドライブ、または他のいずれかの機械読み取り可能記憶媒体というような、有形媒体に具体化されたプログラム・コード(即ち、命令)の形態をなすことができ、このプログラム・コードが、コンピューターのような機械にロードされこの機械によって実行されると、この機械が本開示を実施する装置となる。プログラマブル・コンピューター上におけるプログラム・コードの実行の場合、計算デバイスは通常プロセッサ、このプロセッサによって読み取り可能な記憶媒体(揮発性および不揮発性メモリーおよび/または記憶エレメントを含む)、少なくとも 1 つの入力デバイス、および少なくとも 1 つの出力デバイスを含む。1 つ以上のプログラムは、アプリケーション・プログラミング・インターフェース(API)、再利用可能な制御手段等の使用によって、本開示に関連して説明したプロセスを実現するまたは利用すること

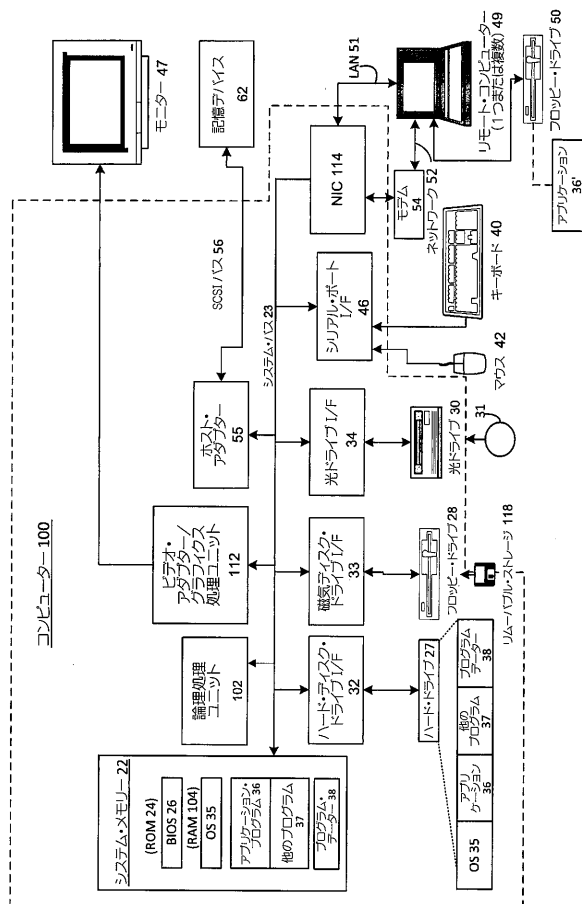
ができる。このようなプログラムは、好ましくは、コンピューター・システムと通信するために、高級手続きプログラミング言語またはオブジェクト指向プログラミング言語で実現される。しかしながら、プログラム（１つまたは複数）は、望ましければ、アセンブリまたは機械語で実現することもできる。いずれの場合でも、言語は、コンパイル型であってもインタプリタ型であってもよく、ハードウェアの実施態様と組み合わせられてもよい。

【 0 0 9 4 】

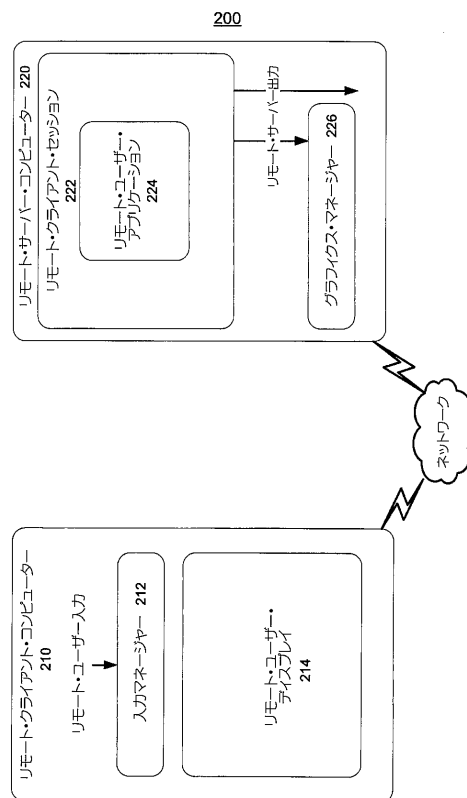
[0109] 以上、本発明について、その好ましい実施形態を参照して特定的に示し説明したが、以下の特許請求の範囲に明記される本発明の範囲から逸脱することなく、形式および詳細には種々の変更がなされてもよいことは当業者には理解されよう。更に、本発明の

10

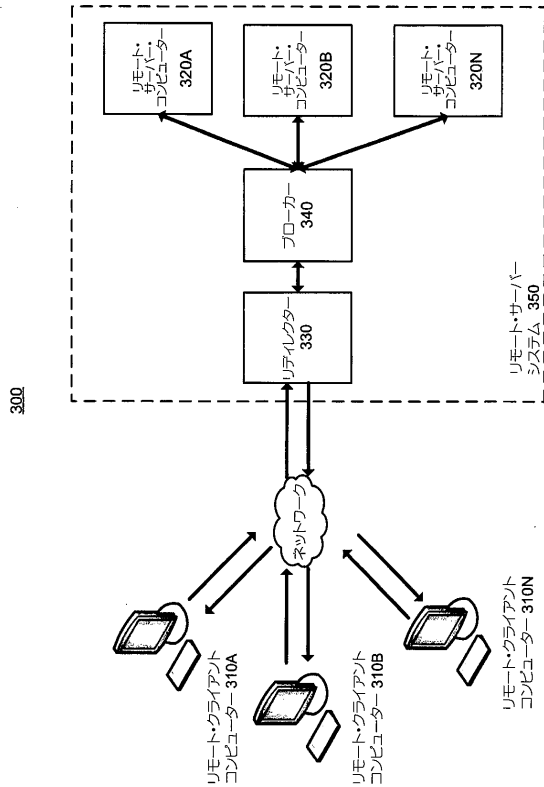
【 図 1 】



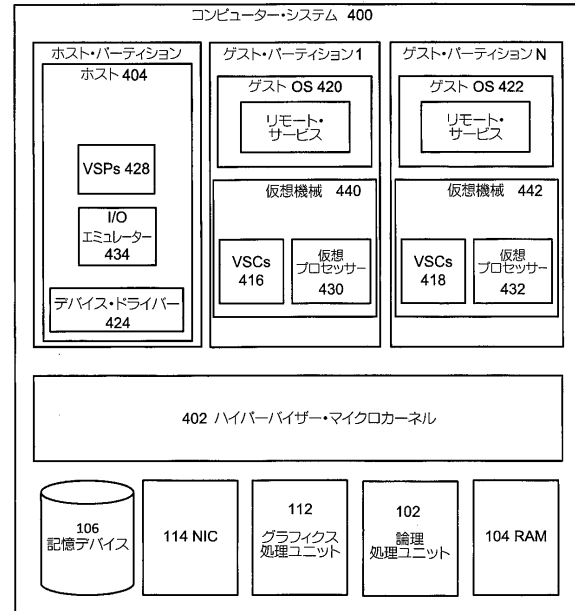
【 図 2 】



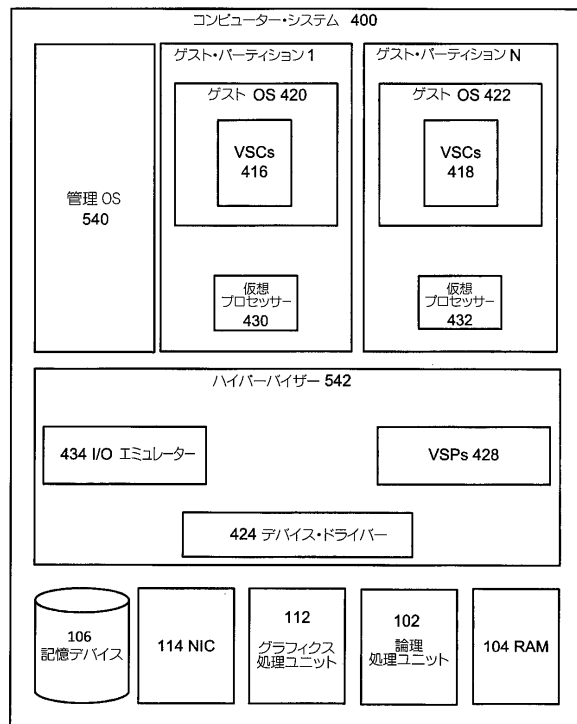
【図 3】



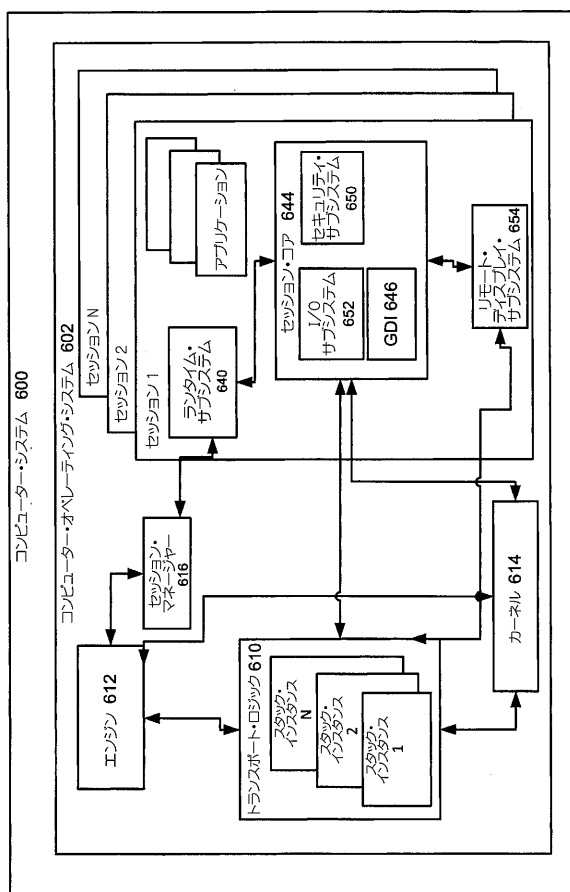
【図 4】



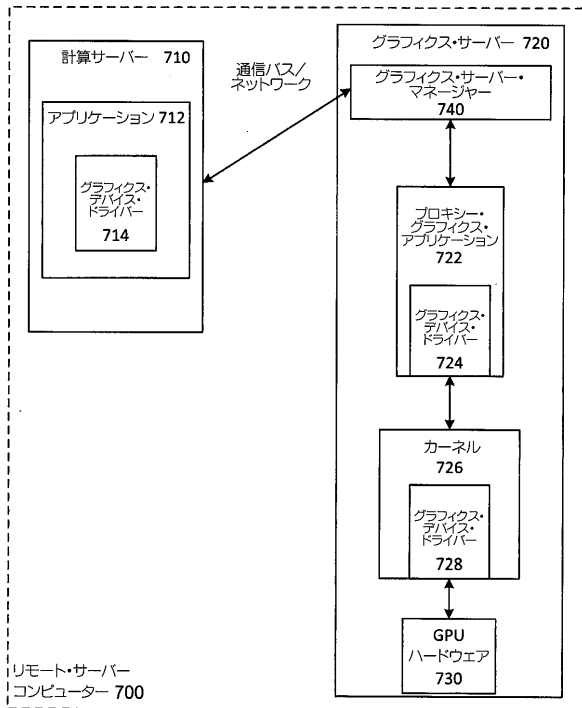
【図 5】



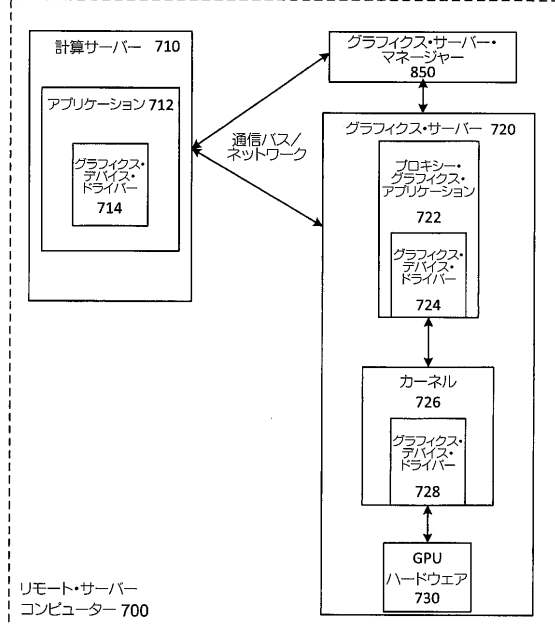
【図 6】



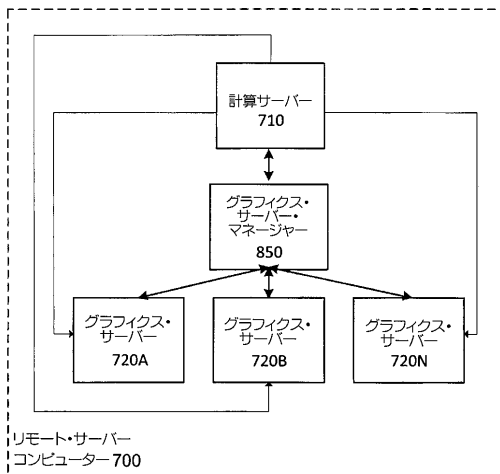
【図 7】



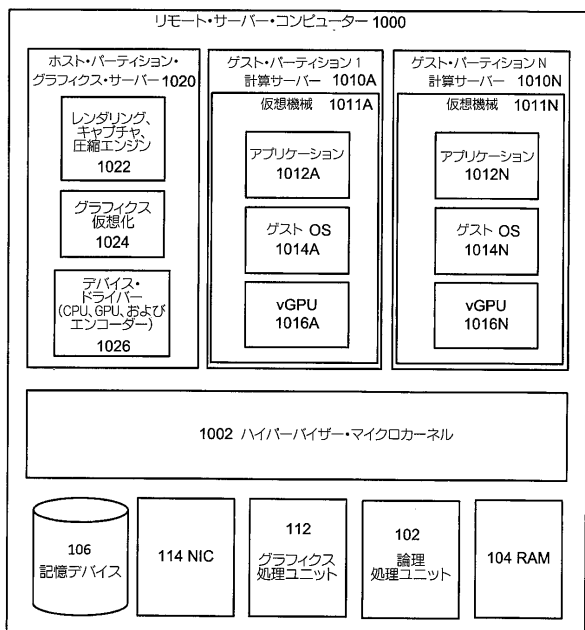
【図 8】



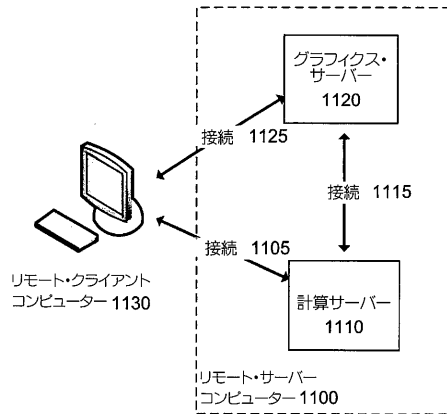
【図 9】



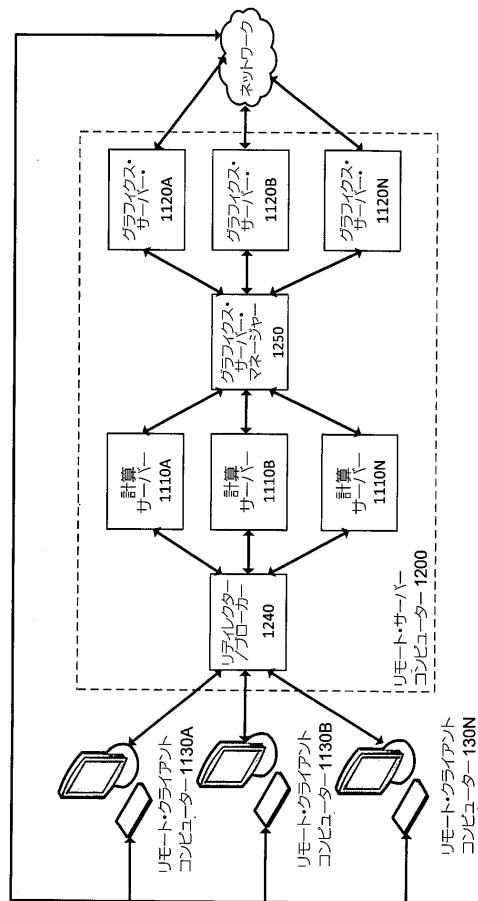
【図 10】



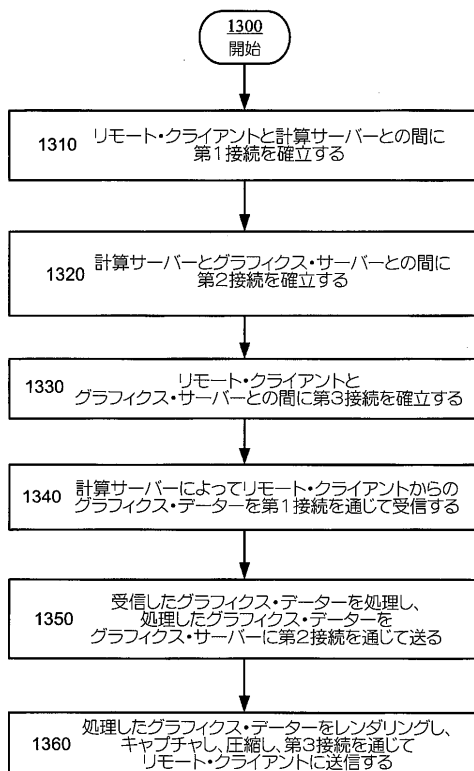
【図 1 1】



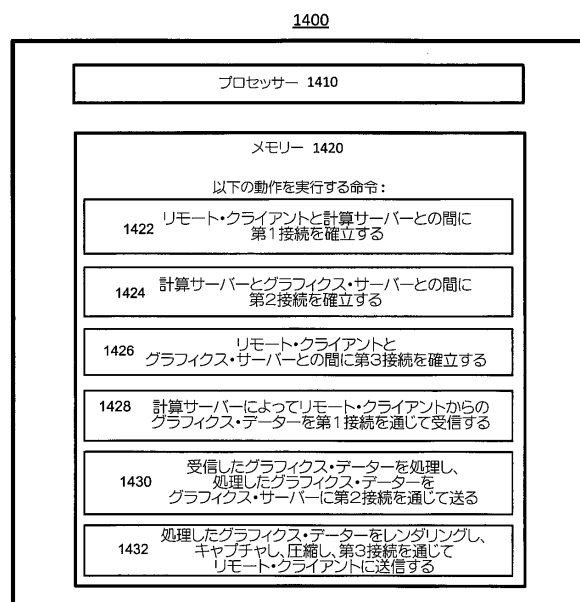
【図 1 2】



【図 1 3】



【図 1 4】



フロントページの続き

- (74)代理人 100153028
弁理士 上田 忠
- (74)代理人 100120112
弁理士 中西 基晴
- (74)代理人 100196508
弁理士 松尾 淳一
- (74)代理人 100147991
弁理士 鳥居 健一
- (74)代理人 100119781
弁理士 中村 彰吾
- (74)代理人 100162846
弁理士 大牧 綾子
- (74)代理人 100173565
弁理士 末松 亮太
- (74)代理人 100138759
弁理士 大房 直樹
- (72)発明者 チャクラボルティー, パラグ
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 ポスト, ブラッドレイ
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 小林 哲雄

- (56)参考文献 米国特許出願公開第 2 0 1 1 / 0 1 8 5 0 6 8 (U S , A 1)
特表 2 0 0 8 - 5 2 1 0 7 6 (J P , A)
特表 2 0 0 8 - 5 2 6 1 0 7 (J P , A)
国際公開第 2 0 1 0 / 1 0 4 6 8 5 (W O , A 1)
米国特許出願公開第 2 0 0 6 / 0 0 9 8 6 4 5 (U S , A 1)

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