

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7297769号
(P7297769)

(45)発行日 令和5年6月26日(2023.6.26)

(24)登録日 令和5年6月16日(2023.6.16)

(51)国際特許分類		F I			
G 0 6 F	9/445(2018.01)	G 0 6 F	9/445		
G 0 6 T	15/00 (2011.01)	G 0 6 T	15/00	5 0 1	
A 6 3 F	13/77 (2014.01)	A 6 3 F	13/77		

請求項の数 14 (全28頁)

(21)出願番号	特願2020-537754(P2020-537754)	(73)特許権者	517160525 パルプ コーポレーション アメリカ合衆国、9 8 0 0 4 ワシントン州、ベルビュー、ノースイースト 4 ティーエイチ ストリート 1 0 4 0 0 スイート 1 4 0 0
(86)(22)出願日	平成31年1月25日(2019.1.25)	(74)代理人	110001737 弁理士法人スズ工国際特許事務所
(65)公表番号	特表2021-512391(P2021-512391 A)	(72)発明者	グリファイス、ピエール・ループ・ミグ エル アメリカ合衆国、9 8 0 0 4 ワシントン州、ベルビュー、ノースイースト 4 ティーエイチ ストリート 1 0 4 0 0
(43)公表日	令和3年5月13日(2021.5.13)	審査官	加藤 優一
(86)国際出願番号	PCT/US2019/015205		
(87)国際公開番号	WO2019/147974		
(87)国際公開日	令和1年8月1日(2019.8.1)		
審査請求日	令和4年1月18日(2022.1.18)		
(31)優先権主張番号	15/881,552		
(32)優先日	平成30年1月26日(2018.1.26)		
(33)優先権主張国・地域又は機関	米国(US)		

最終頁に続く

(54)【発明の名称】 プリキャッシュのためのクライアント・マシン間でのシェーダの配布

(57)【特許請求の範囲】

【請求項 1】

クライアント・マシンであって、
中央処理装置（CPU）と、
グラフィックス・プロセッシング・ユニット（GPU）と、
コンピュータが実行可能な命令を格納するメモリと、を具備し、前記命令は、前記CPUにより実行されると、前記クライアント・マシンに以下のことをさせる
前記クライアント・マシンに関連付けられたディスプレイ上でグラフィックスを描写する複数のインストールされたプログラムのうちの1つを実行するように構成されたクライアント・アプリケーションをロードし、
前記複数のインストールされたプログラムのうちの1つのプログラムのためにコンパイルされたシェーダ・コードを求めるため、前記クライアント・マシンのハードウェア構成を含んだ要求をリモート・コンピューティング・システムに送信し、
前記リモート・コンピューティング・システムから、前記コンパイルされたシェーダ・コードが利用できない旨の指示を受信し、
前記プログラムを実行し、
前記クライアント・マシンのグラフィックス・ドライバにより、前記GPUによって実行可能なマシン・コード内のコンパイルされたシェーダとしてシェーダのセットをコンパイルし、
前記グラフィックス・ドライバによって、前記コンパイルされたシェーダを、予め定義

されたメモリ・ロケーションに、前記プログラムのアプリケーション識別子（ID）と関連づけてキャッシュし、

前記コンパイルされたシェーダのためのシェーダIDのセットを生成し、
前記リモート・コンピューティング・システムに情報を送信し、前記情報は、
前記クライアント・マシンの前記ハードウェア構成と、
前記プログラムの前記アプリケーションIDと、
前記コンパイルされたシェーダの前記シェーダIDのセットと、
を含む、
クライアント・マシン。

【請求項2】

前記プログラムは、ビデオゲームである、請求項1のクライアント・マシン。

【請求項3】

前記ハードウェア構成は、前記グラフィックス・ドライバのバージョンと前記GPUのモデルを指定する、請求項1のクライアント・マシン。

【請求項4】

前記コンピュータが実行可能な命令は、前記CPUにより実行されると、前記クライアント・マシンに、前記コンパイルされたシェーダ・コードの前記マシン・コードをハッシュ化してハッシュのセットを生成させ、前記ハッシュのセットが前記リモート・コンピューティング・システムに送信される前記情報に含まれる、請求項1のクライアント・マシン。

【請求項5】

リモート・コンピューティング・システムにより、第1クライアント・マシンから第1情報を受信すること、前記第1情報は

前記第1クライアント・マシンの第1ハードウェア構成と、

クライアント・マシンでプログラムの実行中に、前記クライアント・マシンに関連付けられたディスプレイ上にグラフィックスを描写する前記プログラムのアプリケーション識別子（ID）と、

第1クライアント・マシンで前記プログラムの実行中に、前記第1クライアント・マシンのグラフィックス・ドライバによってコンパイルされたシェーダのシェーダIDのセットと、

を含み、

前記リモート・コンピューティング・システムにより、前記第1ハードウェア構成及び前記アプリケーションIDに従ってシェーダIDのセットをカタログ化すること、

前記リモート・コンピューティング・システムによって、前記第1クライアント・マシンから、前記シェーダの1つ以上のシェーダに対応するコンパイルされたシェーダ・コードを要求すること、

前記リモート・コンピューティング・システムによって、前記第1クライアント・マシンから前記コンパイルされたシェーダ・コードを受信すること、

コンパイルされたシェーダ・コードを再配布のため、アプリケーションID及び第1ハードウェア構成と関連付けること、

前記リモート・コンピューティング・システムにより、第2クライアント・マシンから第2情報を受信することと、前記第2情報は、

前記第2クライアント・マシンの第2ハードウェア構成と、

前記プログラムのアプリケーションIDと、

を含み、

前記リモート・コンピューティング・システムにより、前記第2ハードウェア構成が前記第1ハードウェア構成と一致することを決定すること、

前記リモート・コンピューティング・システムにより、前記第2クライアント・マシンで前記プログラムの実行中に、前記第2クライアント・マシンのグラフィックス・プロセッシング・ユニット（GPU）によって実行されるように、前記コンパイルされたシェー

10

20

30

40

50

ダ・コードを前記第 2 クライアント・マシンに送信すること、
を含む方法。

【請求項 6】

前記プログラムは、ビデオゲームである、請求項 5 の方法。

【請求項 7】

前記シェーダ ID のセットは、シェーダキーのセットを具備し、前記シェーダキーのセットは、タブルのセット内に前記第 1 クライアント・マシンから受信され、個々のタブルがシェーダキーと、前記シェーダキーに対応するシェーダのために前記コンパイルされたシェーダ・コードのハッシュとを含む、請求項 5 の方法。

【請求項 8】

前記リモート・コンピューティング・システムにより、コンパイルされたシェーダ・コードの検証用ハッシュを生成するため、前記第 1 クライアント・マシンから受信したコンパイルされたシェーダ・コードをハッシュ化すること、をさらに具備し、

前記コンパイルされたシェーダ・コードと再配布のための前記アプリケーション ID 及び前記第 1 ハードウェア構成との関連付けは、前記第 1 クライアント・マシンから受信した前記タブルのセットの 1 つのタブル内の前記コンパイルされたシェーダ・コードのハッシュと一致する検証ハッシュの少なくとも一部に基づいている、請求項 7 の方法。

【請求項 9】

前記第 2 ハードウェア構成が前記第 1 ハードウェア構成と一致することを決定することは、前記第 1 ハードウェア構成と前記第 2 ハードウェア構成が、共通のグラフィックス・ドライバのバージョンと共通の GPU モデルを指定することを決定することを具備する、請求項 5 の方法。

【請求項 10】

前記コンパイルされたシェーダ・コードの前記要求は、前記リモート・コンピューティング・システムが、少なくとも所定の数の異なるクライアント・マシンから、前記第 1 クライアント・マシン及び少なくとも 1 つの追加のクライアント・マシンを含む異なるクライアント・マシンから、前記 1 つ又は複数のシェーダの 1 つ又は複数のシェーダ ID を受信したことに応答して実行される、請求項 5 の方法。

【請求項 11】

前記コンパイルされたシェーダ・コードを再配布のための前記アプリケーション ID 及び第 1 ハードウェア構成に関連付ける前に、

前記リモート・コンピューティング・システムにより、異なるハードウェア構成を持ち、異なるアプリケーション ID を提供する複数のクライアント・マシンから、追加のシェーダ ID を受信すること、

前記リモート・コンピューティング・システムにより、ハードウェア構成とアプリケーション ID による追加のシェーダ ID をカタログ化すること、

前記リモート・コンピューティング・システムにより、前記 1 つ又は複数のシェーダの特定のシェーダ ID について、前記リモート・コンピューティング・システムに前記特定のシェーダ ID を送信した複数の異なるクライアント・マシンの数をカウントすること、

前記異なるクライアント・マシンの数が閾値を満たすか、又は超えていると判断すること、

をさらに具備し、

前記コンパイルされたシェーダ・コードと再配布のための前記アプリケーション ID 及び前記第 1 ハードウェア構成との関連付けは、前記異なるクライアント・マシンの数が閾値を満たすか、又は超えると判断することに少なくとも一部に基づいている、請求項 5 の方法。

【請求項 12】

前記コンパイルされたシェーダ・コードを再配布のための前記アプリケーション ID 及び前記第 1 ハードウェア構成に関連付ける前に、

前記第 1 クライアント・マシンに関連付けられたユーザアカウントが信頼されたアカウ

10

20

30

40

50

ントであるかどうかを判断すること、をさらに具備し、

前記コンパイルされたシェーダ・コードと再配布のための前記アプリケーションID及び前記第1ハードウェア構成との関連付けは、前記ユーザアカウントが信頼されたアカウントであるとの判断に少なくとも部分的に基づいている、請求項5の方法。

【請求項13】

前記コンパイルされたシェーダ・コードを再配布のための前記アプリケーションID及び前記第1ハードウェア構成と関連付ける前に、

前記リモート・コンピューティング・システムにより、前記第1ハードウェア構成と一致するハードウェア構成を有する複数のクライアント・マシンから追加のシェーダIDを受信すること、

前記リモート・コンピューティング・システムにより、前記第1クライアント・マシンから前記コンパイルされたシェーダ・コードの受信に加えて、前記複数のクライアント・マシンのうちの少なくとも1つのクライアント・マシンから前記コンパイルされたシェーダ・コードを受信すること、

前記少なくとも1つのクライアント・マシンから受信した前記コンパイルされたシェーダ・コードが、前記第1クライアント・マシンから受信した前記コンパイルされたシェーダ・コードと一致することを判断すること、

をさらに具備し、

前記コンパイルされたシェーダ・コードと再配布のための前記アプリケーションID及び前記第1ハードウェア構成との関連付けは、前記少なくとも1つのクライアント・マシンから受信した前記コンパイルされたシェーダ・コードが前記第1クライアント・マシンから受信した前記コンパイルされたシェーダ・コードと一致するとの判断に少なくとも部分的に基づいている、請求項5の方法。

【請求項14】

前記コンパイルされたシェーダ・コードを前記第2クライアント・マシンに送信した後、

前記リモート・コンピューティング・システムにより、前記第1ハードウェア構成と一致する第3ハードウェア構成を有する第3クライアント・マシンから1つ以上のシェーダIDのシェーダIDを受信すること、前記第3クライアント・マシンに関連付けられたユーザアカウントは、信頼されたユーザアカウントであり、

前記リモート・コンピューティング・システムにより、前記第1クライアント・マシンからの前記コンパイルされたシェーダ・コードの受信に加えて、前記第3クライアント・マシンから前記コンパイルされたシェーダ・コードを受信すること、

前記第3クライアント・マシンから受信した前記コンパイルされたシェーダ・コードが、前記第1クライアント・マシンから受信した前記コンパイルされたシェーダ・コードと一致しないと判断すること、

前記第3クライアント・マシンから受信した前記コンパイルされたシェーダ・コードが前記第1クライアント・マシンから受信した前記コンパイルされたシェーダ・コードと一致しないと判断に少なくとも部分的に基づいて、前記コンパイルされたシェーダ・コードを追加のクライアント・マシンに送信することを控えること、

をさらに具備する、請求項5の方法。

【発明の詳細な説明】

【関連出願との相互参照】

【0001】

本出願は、2018年1月26日に出願された米国特許出願第15/881,552号の優先権を主張するPCT出願であり、“DISTRIBUTING SHADERS BETWEEN CLIENT MACHINES FOR PRECACHING”と題され、これは参照によりその全体が本明細書に組み込まれる。

【技術分野】

【0002】

シェーダとは、ディスプレイ上に様々な視覚効果を作り出すためにグラフィックカード

10

20

30

40

50

上で実行されるプログラムである。最近のビデオゲームは、ゲームプレイ中に画面上に異なるマテリアルを描写(render)するためにシェーダを使用している(例えば、艶消しのマテリアル、光沢を有するマテリアルなど)。これらのプログラムがグラフィックスパイプラインで実行されると、シェーダはピクセル単位の値(例えば、色の値など)を出力し、それらが組み合わされてディスプレイ上に所望の視覚効果を持つ画像が生成される。

【0003】

ゲーム開発者は、シェーダのソースコードを、高レベルシェーディング言語(HLSL)、OpenGLシェーディング言語(GLSL)、又は類似のプログラミング言語などの高レベルプログラミング言語で記述し、このソースコードはシェーダのバイトコード表現にコンパイルされる。しかし、シェーダをグラフィックスカード上で実行する前に、シェーダはさらに、クライアント・マシンのグラフィックス・プロセッシング・ユニット(GPU)によって実行可能なマシン・コードにコンパイルされる。換言すると、顧客がクライアント・マシンで初めてゲームを実行すると、グラフィックス・ドライバは、シェーダに関連付けられたバイトコードをロードし、そのバイトコードをクライアント・マシンの特定のGPUで実行可能なマシン・コードにコンパイルする。この最終的なシェーダコンパイルプロセスの結果、クライアント・マシンのハードウェア構成に固有の(例えば、GPUモデル及び/又はグラフィックス・ドライバのバージョンに固有の)コンパイルされたシェーダ・コードが得られる。殆どのゲーム開発者は、異なるハードウェア構成のパーソナルコンピュータ(PC)プラットフォーム上でビデオゲームを実行したいと考えているため、ゲーム開発者は多くの場合、シェーダをバイトコード形式で顧客に配布し、シェーダのコンパイルプロセスをクライアント・マシン上で実行時に行うようにしておき、特定のグラフィックス・ドライバがシェーダを問題のハードウェア構成に合った正しいマシン・コードにコンパイルできるようにしている。対照的に、コンソール・ビデオゲームは、シェーダ用のマシン・コードがバイトコードから既にコンパイルされており、コンソールのGPUで実行できる状態出荷される。これは、コンソールのハードウェア構成がゲーム開発者に知られているためである。

【0004】

PCプラットフォームでは、ビデオゲームがクライアント・マシン上でロードされているときにシェーダがコンパイルされると、ゲームの初期ロード画面が望ましくないほど長い時間(例えば、数分)に亘って提示されることがある。これは、ゲームがロードされているときに、グラフィックス・ドライバが、ゲームプレイ中にGPUによって実行可能なマシン・コードを作成するため、ゲームのデータセットの全てのシェーダをコンパイルするからである。あるいは、ゲーム開発者がゲームプレイ中にシェーダをコンパイルするようにゲームを設計した場合、初めて新しいマテリアルに遭遇すると、グラフィックス・ドライバはそのマテリアルのシェーダをマシン・コードにコンパイルするが、これには時間がかかり、ゲーム内のスタッタリング(stuttering)やヒッチング(hitching)(グラフィックス・ドライバがシェーダをマシン・コードにコンパイルするためにとても多くの時間を費やし、十分な速度でフレームを描写することができない状態)を引き起こす可能性がある。このように、クライアント・マシン上でシェーダをローカルにコンパイルすると、ロード時間が長くなり、ゲーム内のスタッタリングが発生する可能性がある。ここでは、これらのシステムやその他のシステムを改善し、強化するための技術的なソリューションを提供する。

【図面の簡単な説明】

【0005】

添付の図を参照して詳細な説明が記載される。図において、参照番号の左端の桁は、参照番号が最初に現れる図を識別する。異なる図で同じ参照番号を使用することは、類似又は同一の構成要素又は特徴を示す。

【0006】

【図1】クライアント・マシン間でシェーダを配布してプリキャッシングするシステムを含む環境の一例を示す図である。

10

20

30

40

50

【 0 0 0 7 】

【図 2】クライアント・マシンの例示的な構成要素を示すブロック図であり、コンパイルされたシェーダをリモート・コンピューティング・システムにアップロード及び/又はダウンロードするための例示的なプロセスのフロー図である。

【 0 0 0 8 】

【図 3】リモート・コンピューティング・システムの例示的な構成要素を示すブロック図であり、クライアント・マシンから情報を受信し、受信した情報に含まれるシェーダ識別子 (ID) をカタログ化し、コンパイルされたシェーダ・コードを選択的に要求し、再配布のためにコンパイルされたシェーダ・コードを選択的に準備するための例示的なプロセスのフロー図である。

10

【 0 0 0 9 】

【図 4】クライアント・マシンのハードウェア構成に基づいて、要求するクライアント・マシンにコンパイルされたシェーダ・コードを送信するための例示的なプロセスのフロー図である。

【 0 0 1 0 】

【図 5】コンパイルされたシェーダ・コードを選択的に要求するための例示的なプロセスのフロー図である。

【 0 0 1 1 】

【図 6】再配布用にコンパイルされたシェーダ・コードを選択的に準備するための例示的なプロセスのフロー図である。

20

【 0 0 1 2 】

【図 7】ダウンロードしたシェーダ・コードの一部を破棄し、破棄したシェーダ・コードに対応するシェーダを再コンパイルし、再コンパイルしたシェーダの情報をアップロードすることによって完全性チェックを実施するための例示的なプロセスのフロー図である。

【 0 0 1 3 】

【図 8】図 7 のプロセスの完全性チェックを使用して、既に配布されているシェーダ・コードの更なる配布を停止するための例示的なプロセスのフロー図である。

【 発明を実施するための形態 】

【 0 0 1 4 】

ここに記載されているのは、特に、クライアント・マシン上でコンパイルされたシェーダをプリキャッシングする目的で、クライアント・マシン間でシェーダを配布するための技術、装置、及びシステムである。プログラムは、クライアント・マシンにインストールされ、クライアント・アプリケーションによって実行されてもよい。そのようなプログラムは、プログラムの実行の一部として、クライアント・マシンに関連付けられたディスプレイ上にグラフィックスを与えてもよい。このようなプログラムの例は、ビデオゲームである。

30

【 0 0 1 5 】

新しいビデオゲームが開発されると、ユーザはそれぞれのクライアント・マシンでゲームのプレイ/実行を開始することができ、それらのクライアント・マシンはハードウェア構成により異なることがある。「ハードウェア構成」は、ここで使用されるクライアント・マシンの GPU モデル、及び/又はグラフィック・ドライバのバージョンにより定義される。最初に、新しいビデオゲームがクライアント・マシン上で実行されたことがなく、複数のクライアント・マシンがゲームの実行を開始すると、ゲームのためのシェーダ (バイトコード) は、各クライアント・マシン上のグラフィックス・ドライバによって、各クライアント・マシン上の特定の GPU によって実行可能なマシン・コードにコンパイルされる。ここに記載の技術、装置、及びシステムは、プログラムがクライアント・マシン上で実行される際に、コンパイルされたシェーダに関する情報をリモート・コンピューティング・システムに報告することに関連し、リモート・システムは、それらのマシンのハードウェア構成に基づいて、コンパイルされたシェーダ・コードを他のクライアント・マシンに選択的に再配布することができる。これは、特定のプログラム (例えば、ビデオゲー

40

50

ム)のシェーダを未だコンパイルしていないクライアント・マシンが、リモート・コンピューティング・システムからコンパイルされたシェーダ・コードを、要求してダウンロードすることを可能にし、そのシェーダ・コードに対応するシェーダをローカルにコンパイルする必要があるクライアント・マシンに代わり、それが実行されるように、このコンパイルされたシェーダ・コードが、クライアント・マシンにプリキャッシュされる。

【0016】

いくつかの実施形態において、クライアント・マシンは、プログラム(例えば、ビデオゲーム)に関連するコンパイルされたシェーダ・コードを求める要求をリモート・コンピューティング・システムに送信するように構成されている。このような要求は、クライアント・マシンがプログラムのためのシェーダをまだコンパイルしていないことに基づいて送信されてもよい(例えば、クライアント・マシンがプログラムをインストールしたが、まだプログラムを実行していない場合)。クライアント・マシンが、コンパイルされたシェーダ・コードが利用できない旨を示す応答をリモート・コンピューティング・システムから受信した場合、クライアント・マシンのグラフィックス・ドライバは、プログラムの実行時に、シェーダをコンパイルして、所定のメモリ・ロケーションに、プログラムのアプリケーション識別子(ID)と関連付けてキャッシュしてもよい。クライアント・アプリケーションは、コンパイルされたシェーダのためのシェーダIDのセットを生成してもよく、リモート・コンピューティング・システムに情報を送信してもよく、その情報は、クライアント・マシンのハードウェア構成、プログラムのアプリケーションID、及びコンパイルされたシェーダのためのシェーダIDのセットを含む。これにより、リモート・システムは、クライアント・マシンから、それらのマシン上でコンパイルされているシェーダに関する情報を収集することができる。

【0017】

一方、クライアント・マシンが、要求したシェーダ・コードが利用可能であることを示す応答をリモート・システムから受信した場合、クライアント・マシンは、リモート・コンピューティング・システムから要求したコンパイルされたシェーダ・コードを受信してもよい。コンパイルされたシェーダ・コードを受信すると、クライアント・マシンは、コンパイルされたシェーダ・コードの少なくとも一部をプリキャッシュするか、又はその後の使用のために、コンパイルされたシェーダ・コードの少なくとも一部を所定のメモリ・ロケーションに格納してもよい。プログラム(例えば、ビデオゲーム)の実行時に、クライアント・マシンのGPUは、グラフィックス・ドライバがコンパイルされたマシン・コードを独自に作成しなければならない代わりに、所定のメモリ・ロケーションから取得されたコンパイルされたシェーダ・コードの少なくとも一部を実行してもよい。

【0018】

いくつかの実施形態において、リモート・コンピューティング・システムは、クライアント・マシンがプログラム(例えば、ビデオゲーム)を実行する際に、クライアント・マシンから情報を受信するように構成されている。所与のクライアント・マシンについてリモート・コンピューティング・システムで受信した情報は、クライアント・マシンのハードウェア構成、クライアント・マシンで実行されたプログラムのアプリケーションID、及びプログラムの実行中にクライアント・マシンのグラフィックス・ドライバによってコンパイルされたシェーダのシェーダIDのセットを含んでもよい。リモート・コンピューティング・システムは、ハードウェア構成及びアプリケーションIDに従ってシェーダIDをカタログ化してもよく、受信したシェーダIDに対応するシェーダのうちの一つ又は複数のシェーダのためにコンパイルされたシェーダ・コードを選択的に要求してもよい。コンパイルされたシェーダ・コードは、対応するアプリケーションIDとハードウェア構成とをコンパイルされたシェーダ・コードに関連付けることにより、再配布のために準備されてもよい。このようにして、クライアント・マシンのハードウェア構成と特定のプログラムのアプリケーションIDとを含むクライアント・マシンからの要求を受信すると、リモート・コンピューティング・システムは、そのアプリケーションIDと一致するハードウェア構成に関連付けられたコンパイルされたシェーダ・コードが利用可能であるかど

10

20

30

40

50

うかを判断することができる。利用可能であれば、コンパイルされたシェーダ・コードをクライアント・マシンのGPUによる後続の実行のため（すなわち、プリキャッシングのために）、要求するクライアント・マシンに送信することができる。

【0019】

ここに記載された技術及びシステムは、1つ以上の装置が少なくとも処理リソースに関するリソースを節約することが可能である。例えば、コンパイルされたシェーダ・コードのプリキャッシングは、プログラムがクライアント・マシン上にロードされたとき（例えば、初めてロードされたプログラム）に、クライアント・マシンがシェーダ・コードをコンパイルすることを不要にすることによって、ロード時間を短縮することができる。ここに記載された技術及びシステムは、ビデオゲームの実行中のゲーム内スタッタリング又はヒッチングを減少させることによって、コンピュータ自体の機能をさらに向上させることができる。これは、既にコンパイルされているプリキャッシュされたシェーダ・コードをプログラムの実行中にグラフィックス（例えば、マテリアル）を描写するために使用することができるからであり、リアルタイムでそれらのマテリアルのためのシェーダ・コードをコンパイルするのとは対照的である。さらに、ユーザの大規模なコミュニティ間でコンパイルされたシェーダ・コードを収集し、再配布する中央ハブとしてのリモート・システムを使用することにより、単一のクライアント・マシンは、そのクライアント・マシンがそうでなければ独自にキャッシュしていたであろうよりも多くのコンパイルされたマシン・コードを提供される可能性がある。このように、ユーザのコミュニティからコンパイルされたシェーダ・コードを収集することで、ディスプレイ上でグラフィックスを描写するプログラムの実行効率を最適化することが可能になる。

【0020】

ここに記載された例の多くは、コンパイルされたマシン・コードでクライアント・マシンに配布され、プリキャッシュされる「シェーダ・コード」に言及しているが、シェーダ・コードとは異なる他のタイプのデータ及び/又はコードが、ここに記載された技術及びシステムで使用されてもよいことが認識される。実際には、任意のタイプのプログラムを実行することによる任意の副産物は、（プログラムがグラフィックスを描写するか否かにかかわらず）ユーザコミュニティの間で配布されてもよく、特に、これら副産物（データ及び/又はコード）は、リソースを節約し、及び/又はクライアント・マシン上でのプログラムの実行を最適化する。実際には、計算の最後で小さなペイロードをもたらす計算（複数可）を実行する任意のタイプのプログラムは、本開示の恩恵を受けることができる。このように、「ビデオゲーム」は、ここに記載された様々な実施形態を例示するためのプログラムとして使用されているが、技術、装置、及びシステムは、ビデオゲーム以外の任意のタイプのプログラムで利用することができることが理解されるであろう。このように、本開示は、ビデオゲームで使用するためのシェーダのプリキャッシングに限定されるものではなく、他のプログラムもまた、プログラム実行の任意の副産物（例えば、コンパイルされたシェーダ・コード、コンパイルされたJavaScript（登録商標）コードなど）のプリキャッシングから利益を得ることができる。

【0021】

図1は、クライアント・マシン間でシェーダを配布し、プリキャッシングするためのシステムを含む例示的な環境100を示す図である。コミュニティのユーザ102（「顧客」と呼ばれることもある）は、1つ以上のクライアント・マシン104に関連付けられていてもよい。したがって、図1に示されるクライアント・マシン104（1）～（N）は、ユーザコミュニティ（又は顧客ベース）によって利用され、その上でプログラムを実行することができるコンピューティング・デバイスを表している。クライアント・マシン104は、関連するディスプレイ上でグラフィックスを処理し、描写するように構成された任意の適切なタイプのコンピューティング・デバイスとして実装することができ、PC、デスクトップコンピュータ、ラップトップコンピュータ、携帯電話（例えば、スマートフォン）、タブレットコンピュータ、ポータブル・デジタル・アシスタント（PDA）、ウェアラブルコンピュータ（例えば、仮想現実（VR）ヘッドセット、拡張現実（AR）へ

10

20

30

40

50

ッドセット、スマートグラスなど)、車載(例えば、車内)コンピュータ、テレビ(スマートテレビ)、セットトップボックス(STB)、ゲームコンソール、及び/又は任意の類似のコンピューティング・デバイスを含むが、これらに限定されない。

【0022】

クライアント・マシン104のプラットフォーム(例えば、ハードウェア及びソフトウェア)は、クライアント・マシン104のサブセットが共通のハードウェア構成を共有し、クライアント・マシン104のこれらのサブセットの間でハードウェア構成が異なってもよい。例えば、クライアント・マシン104のサブセットは、それぞれ特定のGPUモデル及び特定のバージョンのグラフィックス・ドライバを使用してもよく、一方、クライアント・マシン104の別のサブセットは、それぞれ異なるGPUモデル及び/又は異なるバージョンのグラフィックス・ドライバを使用してもよい。グラフィックス・ドライバがソフトウェア(例えば、バイナリ)で実施されているとしても、ここで使用されるように、一对のクライアント・マシン104が同じ(又は共通の)「ハードウェア構成」を有すると言われる場合、これは、一对のクライアント・マシン104が同じGPUモデル及びグラフィックス・ドライバのバージョンを使用していることを意味する。換言すると、ここで使用されるような「ハードウェア構成」は、ハードウェアに加えてソフトウェアの要素を包含してもよい。

10

【0023】

クライアント・マシン104は、コンピュータネットワーク108を介してリモート・コンピューティング・システム106(「リモート・システム106」と略称されることもある)と通信してもよい。コンピュータネットワーク108は、インターネット、他のタイプのデータ及び/又は音声ネットワーク、有線インフラストラクチャ(例えば、同軸ケーブル、光ファイバケーブルなど)、無線インフラストラクチャ(例えば、無線周波数(RF)、セルラー、衛星など)、及び/又は他の接続技術を表し、及び/又は含むことができるが、これらに限定されない。リモート・システム106は、いくつかの実施形態において、コンピュータネットワーク108を介して維持され、アクセス可能なネットワーク・アクセス可能コンピューティング・プラットフォームの一部であってもよい。このようなネットワーク・アクセス可能コンピューティング・プラットフォームは、「オンデマンド・コンピューティング」、「サービスとしてのソフトウェア(SaaS)」、「プラットフォーム・コンピューティング」、「ネットワーク・アクセス可能なプラットフォーム」、「クラウド・サービス」、「データセンタ」などの用語を使用して言及されてもよい。一般に、リモート・システム106は、クライアント・マシン104から情報とコンパイルされたシェーダ・コードの両方を収集するように構成され、受け取った情報とコンパイルされたシェーダ・コードをカタログ化(例えば、整理、分類(categorize)、類別(classify)など)し、コンパイルされたシェーダ・コードを要求するクライアント・マシン104に配布して、それらの要求するクライアント・マシン104上でコンパイルされたシェーダ・コードをプリキャッシングするように構成される。

20

30

【0024】

いくつかの実施形態において、リモート・システム106は、クライアント・マシン104にプログラム(及びコンテンツ)を配布(例えば、ダウンロード)するためのプラットフォームとして機能するか、又はアクセス権を有する。一例において、クライアント・マシン104は、クライアント・アプリケーションをそこにインストールしてもよい。クライアント・アプリケーションは、クライアント・マシン104に関連付けられたディスプレイ上でグラフィックスを描写するプログラム(例えば、リモート・システム106からダウンロードされたプログラム)を実行するように構成されてもよい。ビデオゲーム・プログラムの文脈において、インストールされたクライアント・アプリケーションは、ビデオゲーム・クライアント(例えば、ビデオゲームをプレイするためのゲーミングソフトウェア)であってもよい。クライアント・アプリケーションがインストールされた状態で、クライアント・マシン104は、その後、コンピュータネットワーク108を介してリモート・システム106からプログラム(例えば、ビデオゲーム)をダウンロードする能

40

50

力を有してもよい。任意のタイプのコンテンツ配布モデルがこの目的のために利用することができ、プログラム（例えば、ビデオゲーム）は、クライアント・マシン 104 上でダウンロード及び実行するために個別に購入可能な直接購入モデル、サブスクリプションベースのモデル、プログラムが一定期間レンタル又はリースされるコンテンツ配布モデルなどのようなものである。したがって、個々のクライアント・マシン 104 は、クライアント・アプリケーションをロードすることによって実行可能な 1 つ以上のインストールされたプログラムを含んでもよく、これらのプログラムは、実行中にディスプレイ上にグラフィックスを与えてもよい。一例において、ユーザは、クライアント・アプリケーションをロードし、所望のビデオゲームを選択してビデオゲームの実行を開始することにより、リモート・システム 106 から購入してダウンロードした複数のビデオゲームのうちの 1 つをプレイすることを選択してもよい。

10

【0025】

ユーザ 102 の一部が、リモート・システム 106 によって利用可能にされた新しいビデオゲームをプレイする非常に最初のユーザ 102 である図 1 の例を考える。このユーザ 102 のグループは、発売日に未プレイのゲームを購入する一般ユーザ 102 の最初のユーザ 102 であってもよい。あるいは、このユーザグループ 102 は、「テストユーザ」のグループ、又はゲームのプレリリースコピーへの独占的なアクセスを得ることができるゲームの出版社又は開発者に関連するユーザであってもよい。

【0026】

図 1 に示すように、ステップ 1 において、この第 1 ユーザグループ 102 がそれぞれのクライアント・マシン 104 上で新しいビデオゲーム（すなわち、ディスプレイ上でグラフィックスを描写するプログラム）の実行を開始すると、クライアント・マシン 104 は（それぞれのグラフィックス・ドライバを使用して）シェーダのセットをコンパイルし、コンパイルされたシェーダ・コードは、プログラムのためのグラフィックスを描写するためにクライアント・マシン 104 のそれぞれの GPU によって使用される。その後、コンパイルされたシェーダ・コードは、プログラム（例えば、ビデオゲーム）を実行するクライアント・アプリケーションにアクセス可能な所定のメモリ・ロケーションにキャッシュされ得る。また、プログラム（例えば、ビデオゲーム）のアプリケーション ID は、コンパイルされたシェーダ・コードに関連して、予め定義されたメモリ・ロケーションにキャッシュされてもよい。

20

30

【0027】

ステップ 2 において、クライアント・マシン 104 は、プログラム（例えば、ビデオゲーム）の実行を停止した後などのある時点で、情報 110 をリモート・システム 106 に送信してもよい。所定のクライアント・マシン 104 について、リモート・システム 106 に送信される情報 110 は、クライアント・マシン 104 のハードウェア構成、クライアント・マシン 104 上で実行されたプログラム（例えば、ビデオゲーム）のアプリケーション ID、及びプログラムの実行中にコンパイルされたシェーダに対応するシェーダ ID のセットを含んでもよい。クライアント・マシン 104 は、コンパイルされたシェーダ・コードを予め定義されたメモリ・ロケーションから読み出すと、シェーダ ID のセットを生成してもよい。図 1 は、クライアント・マシン 104 がプログラム（例えば、ビデオゲーム）の実行を停止した後に、情報 110 がリモート・システム 106 に送信されるシナリオを描いているが、情報 110 は、それらのマシン 104 のグラフィックス・ドライバがシェーダ・コードをコンパイルし、所定のメモリ・ロケーションにシェーダ・コードをキャッシュする際に、クライアント・マシン 104 からリアルタイムで、又は実質的にリアルタイムで、情報 110 をストリーミング又はその他の方法で送信することができることを理解されたい。

40

【0028】

情報 110 はリモート・システム 106 によって受信され、ステップ 3 において、リモート・システム 106 は、情報 110 から抽出されたハードウェア構成及びアプリケーション ID に従って、シェーダ ID のセットをカタログ化してもよい。図 1 は、カタログ 1

50

12を示し、これはリモート・システム106がクライアント・マシン104から受信したシェーダIDをカタログ化するために、リモート・システム106によって維持されるデータリポジトリ(又はデータストア)を表している。したがって、カタログ112は、シェーダIDを、ハードウェア構成及びアプリケーションIDのユニークな組み合わせにそれぞれ関連付けられたグループ(又はバケット)に整理してもよい。換言すると、シェーダIDの各バケットは、特定のプログラム(例えば、ビデオゲーム)及び特定のハードウェア構成(例えば、GPUモデル及びグラフィックス・ドライバのバージョン)に関連付けられていてもよい。

【0029】

ステップ4において、リモート・システム106は、情報110をリモート・システム106に報告したクライアント・マシン104にコンパイルされたシェーダ・コード116を選択的に要求し、クライアント・マシン104から受信してもよい。任意の適切なアルゴリズムが、以下でより詳細に説明するように、コンパイルされたシェーダ・コード116を選択的に要求114するために利用され得る。例示において、リモート・システム106は、シェーダIDの第1セットを有する第1クライアント・マシン104(1)から第1情報110を受信してもよい。例示的な実施形態において、リモート・システム106が、第1クライアント・マシン104(1)と同じハードウェア構成を有する第2クライアント・マシン104(2)から、第1クライアント・マシン104(1)から以前に受信したシェーダIDと一致する少なくとも1つのシェーダIDを含む第2情報110を受信すると直ぐに、リモート・システム106は、一致するシェーダIDに対応するコンパイルされたシェーダ・コード116の要求114を第2クライアント・マシン104(2)に送信してもよい。換言すると、リモート・システム106は、いくつかの実施形態において、同じハードウェア構成を有する所定の数(例えば、2つの)異なるクライアント・マシン104から同じシェーダIDを受信すると直ぐに、コンパイルされたシェーダ・コードを要求してもよい。

【0030】

コンパイルされたシェーダ・コード116を受信した後、リモート・システム106は、ステップ5で、コンパイルされたシェーダ・コード116を、その対応するシェーダIDと関連づけて(及び対応するハードウェア構成及びアプリケーションIDと関連づけて)格納してもよい。さらに、ステップ5において、リモート・システム106は、同じハードウェア構成を有する他のクライアント・マシン104に再配布するために、コンパイルされたシェーダ・コード116をいつ準備するかを決定するアルゴリズムを使用してもよい。詳細なアルゴリズム(例えば、シェーダIDの人気度(popularity)に基づくアルゴリズム、ユーザアカウントが信頼されているかどうかの判断など)は、この目的のために以下でより詳細に説明される。リモート・システム106がコンパイルされたシェーダ・コード116を再配布することを決定すると(例えば、コンパイルされたシェーダ・コード116を要求するクライアント・マシン104に利用可能にする)、リモート・システム106は、再配布のためにコンパイルされたシェーダ・コード116を準備してもよい(例えば、コンパイルされたシェーダ・コード116を含む再配布可能なデータ構造を構築すること、コンパイルされたシェーダ・コード116を圧縮することなどによって)。したがって、ステップ5の後、リモート・システム106は、特定のハードウェア構成を有するクライアント・マシン104に対して、特定のアプリケーションIDに対応するコンパイル済みシェーダ・コード116を利用可能にすることを決定してもよい。

【0031】

ステップ6において、クライアント・マシン104(N)は、特定のプログラム(例えば、ビデオゲーム)のためにコンパイルされたシェーダ・コードを受信するために、要求118をリモート・システム106に送信してもよい。要求118が送信された時点で、クライアント・マシン104(N)は、プログラムをインストールするプロセス中であってもよいし、クライアント・マシン104(N)は、既にプログラムをインストールしており、クライアント・アプリケーションをロードするプロセス中であってもよい。要求1

10

20

30

40

50

18は、要求するクライアント・マシン104(N)のハードウェア構成とプログラムのアプリケーションIDを含んでもよく、これは、特定のハードウェア構成と特定のプログラム(例えば、ビデオゲーム)のために利用可能であるかもしれないコンパイルされたシェーダ・コード116を探すようにリモート・システム106に指示する。実行例において、リモート・システム106は、クライアント・マシン104(N)からの要求118で指定されたアプリケーションID及びハードウェア構成のためにコンパイルされたシェーダ・コード116を予め用意していてもよく、したがって、要求するクライアント・マシン104(N)は、そのハードウェア構成に関連し、かつ問題のアプリケーションIDに関連するコンパイルされたシェーダ・コード116を受信する。

【0032】

ステップ7において、クライアント・マシン104(N)は、プログラム(例えば、ビデオゲーム)の実行中に使用するため、コンパイルされたシェーダ・コード116を所定のメモリ・ロケーションにプリキャッシュする。このようにすることにより、クライアント・マシン104(N)が、問題となっている特定のプログラム(例えば、ビデオゲーム)のためのコンパイルされたシェーダを以前に有していなかったとしても、クライアント・マシン104(N)は、他のクライアント・マシン104上でのプログラムの実行中に、同じハードウェア構成を有する他のクライアント・マシン104によって以前にコンパイルされた、リモート・システム106から受信したコンパイルされたシェーダ・コード116を活用することができるようになる。これにより、クライアント・マシン104(N)上の計算集約的な処理ステップ(すなわち、シェーダをコンパイルするステップ(複数可))が除去され、これにより、クライアント・マシン104(N)上でのプログラム実行中の処理リソースが節約される(例えば、ロード時間の高速化、ゲーム内スタッタリング又はヒッチングの緩和/解消など)。リモート・システム106によって使用されるアルゴリズム及び/又は特定のビデオゲームの人気に応じて、新しいビデオゲームのためのコンパイルされたシェーダ・コードは、ゲームのリリースから数時間以内、場合によっては数分以内に、要求するクライアント・マシン104に利用可能にされてもよい。これは、多くのユーザがビデオゲームのリリース直後にそのビデオゲームをプレイ始めると、シェーダ・コードの配布がすぐに急増する可能性があるからである。さらに、テストユーザのグループは、ビデオゲームのリリース日の前であっても、コンパイルされたシェーダ・コード116の生成を開始してもよく、これにより、少なくともいくつかのコンパイルされたシェーダ・コード116は、一般ユーザがゲームのプレイを開始したときに、最初のユーザに利用可能になる可能性がある。

【0033】

図2は、クライアント・マシン104の例示的な構成要素を示すブロック図と、図1のリモート・コンピューティング・システム106との間でコンパイルされたシェーダをアップロード及び/又はダウンロードするための例示的なプロセス200のフロー図を示す。図示された実施形態において、クライアント・マシン104は、他の構成要素の中で、中央処理装置(CPU)202及びグラフィックス・プロセッシング・ユニット(GPU)204などのプロセッサ、ディスプレイ206、メモリ208(又は非一時的なコンピュータ読み取り可能媒体208)、及び通信インターフェース210を含む。図2の例示的なクライアント・マシン104は、クライアント・マシン104が埋め込まれたディスプレイ206を含むことを示唆しているが、クライアント・マシン104は、実際にはディスプレイを省略してもよく、その代わりに周辺ディスプレイ206に結合されてもよい。したがって、ディスプレイ206は、クライアント・マシン104に埋め込まれているか、又は(有線プロトコル又は無線プロトコルを介して)そこに接続されているかにかかわらず、関連するディスプレイ206を表すことを意味する。

【0034】

メモリ208(又は非一時的なコンピュータ読み取り可能媒体208)は、揮発性及び不揮発性メモリ、コンピュータ読み取り可能命令、データ構造、プログラムモジュール、又は他のデータなどの情報を記憶するための任意の方法又は技術で実装されたリムーバブル

10

20

30

40

50

ル及び非リムーバブル媒体を含むことができる。このようなメモリは、RAM、ROM、EEPROM、フラッシュメモリ又は他のメモリ技術、CD-ROM、デジタル多目的ディスク(DVD)又は他の光学ストレージ、磁気カセット、磁気テープ、磁気ディスクストレージ又は他の磁気ストレージ装置、RAIDストレージシステム、又は所望の情報を格納するために使用することができ、コンピューティング装置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が、これらに限定されるものではない。コンピュータ読み取り可能媒体208は、コンピュータ読み取り可能記憶媒体(「CRSM」)として実装されてもよく、これは、メモリ208に格納された命令を実行するためにプロセッサ(複数可)202/204によってアクセス可能な任意の利用可能な物理媒体であってもよい。一基本的な実装において、CRSMは、ランダム・アクセス・メモリ(「RAM」)及びフラッシュメモリを含んでもよい。他の実装において、CRSMは、リード・オンリー・メモリ(「ROM」)、電氣的に消去可能なプログラマブル・リード・オンリー・メモリ(「EEPROM」)、又は所望の情報を格納するために使用することができ、プロセッサ(複数可)202/204によってアクセス可能な任意の他の有形媒体を含んでもよいが、これらに限定されない。

10

【0035】

ここで、図2に示されたプロセス200を参照する。ここに記載されたプロセスは、論理フローグラフ内のブロックの集合として図示されており、これは、ハードウェア、ソフトウェア、又はそれらの組み合わせで実装され得る一連の操作を表している。ソフトウェアの文脈において、ブロックは、1つ又は複数のプロセッサによって実行されたときに、記載された操作を実行するコンピュータ実行可能命令を表している。一般に、コンピュータ実行可能命令には、特定の機能を実行したり、特定の抽象的なデータ型を実装したりするルーチン、プログラム、オブジェクト、コンポーネント、データ構造などが含まれる。操作が記載されている順序は、制限として解釈されることを意図しておらず、記載されているブロックの任意の数は、任意の順序及び/又は並列に組み合わせて処理を実施することができる。

20

【0036】

図2は、図1のリモート・コンピューティング・システム106にコンパイルされたシェーダをアップロード及び/又はダウンロードするための例示的なプロセス200のフロー図である。議論の目的のために、プロセス200を実行するクライアント・マシン104は、事前にクライアント・アプリケーション(例えば、ビデオゲーム・クライアント212)をインストールしていることが想定され、クライアント・アプリケーションは関連するディスプレイ206上でグラフィックスを描写するプログラムを実行するように構成される。そのようなプログラムは、インストールされ(例えば、リモート・システム106からダウンロードされ)、メモリ208に保持され得る。図2の例において、プログラムはビデオゲームであり、ビデオゲームがクライアント・マシン104に最初にインストールされるとき、ビデオゲームは、コンパイルされていないシェーダ214(又はゲームプレイ中にシェーダをコンパイルする前のバイトコード表現のシェーダ)を有するビデオゲームとしてメモリ208内に保持される。

30

【0037】

222において、クライアント・マシン104のユーザ102は、ビデオゲーム・クライアント212(すなわち、ディスプレイ206上でグラフィックスを描写する複数のインストールされたプログラムのうちの1つを実行するように構成されたクライアント・アプリケーション)をロードしてもよい。ロードされたビデオゲーム・クライアント212は、コンパイルされていないシェーダ214を有する以前にダウンロードされたビデオゲームを実行する能力、及び/又はコンパイルされていないシェーダ214を有するビデオゲームとしての新しいビデオゲームをダウンロードする能力をユーザ102に提供する。

40

【0038】

ブロック224において、クライアント・マシン104は、ビデオゲーム(ここでは、ビデオゲームは、プログラム実行中にディスプレイ206上にグラフィックスを描写する

50

プログラムを表す)のためのコンパイルされたシェーダ・コードを求める要求 1 1 8 をリモート・コンピューティング・システム 1 0 6 に送信してもよい。ブロック 2 2 4 で送信される要求 1 1 8 は、ブロック 2 2 2 でユーザ 1 0 2 がビデオゲーム・クライアント 2 1 2 をロードすること、及び/又はユーザ 1 0 2 が新しいビデオゲームをダウンロードすることなど、様々な方法でトリガされてもよい。後者の場合、要求 1 1 8 は、新しいビデオゲームのダウンロードの前、途中、又は後に送信されてもよい。ブロック 2 2 4 で送信される要求 1 1 8 は、クライアント・マシン 1 0 4 のハードウェア構成、例えば、GPU 2 0 4 のモデル、及び/又はグラフィックス・ドライバ 2 1 6 のバージョンを含んでもよい(又は指定してもよい)。ブロック 2 2 4 で送信される要求 1 1 8 は、コンパイルされていないシェーダ 2 1 4 を有する 1 つ以上のビデオゲームに対応する 1 つ以上のアプリケーション ID をさらに含んでもよい(又は指定してもよい)。例えば、ビデオゲーム・クライアント 2 1 2 は、ブロック 2 2 4 において、コンパイルされていないシェーダ 2 1 4 が存在するインストールされたビデオゲームの全てを列挙し、それらのゲームの対応するアプリケーション ID をクライアント・マシン 1 0 4 のハードウェア構成と共にリモート・システム 1 0 6 に送信してもよい。

10

【0039】

ブロック 2 2 6 において、クライアント・マシン 1 0 4 は、リモート・システム 1 0 6 からの応答に基づいて、コンパイルされたシェーダ・コードがプログラム(複数可)に対して利用可能であるかどうかを決定してもよい。例えば、リモート・システム 1 0 6 は、ブロック 2 2 4 で送信された要求 1 1 8 に対して、要求されたシェーダ・コードがダウンロード可能ではないという表示を提供することによって応答してもよい。これは、以下でより詳細に説明するように、様々な理由であってもよい。少なくともいくつかの実例において、リモート・システム 1 0 6 は、クライアント・マシン 1 0 4 によって要求されたコンパイルされたシェーダ・コードを所有していてもよいが、再配布のために利用可能にしないことを決定していてもよい。クライアント・マシン 1 0 4 が、要求されたシェーダ・コードが利用可能ではないという指示を受信した場合、プロセス 2 0 0 は、ブロック 2 2 6 からブロック 2 2 8 への「NO」ルートをたどるが、ここで、ユーザの入力に応答して、クライアント・マシン 1 0 4 は、コンパイルされていないシェーダ 2 1 4 でビデオゲームを開始してもよい(すなわち、クライアント・マシン 1 0 4 上でビデオゲーム・プログラムを実行してもよい)。

20

30

【0040】

ブロック 2 3 0 において、クライアント・マシン 1 0 4 のグラフィックス・ドライバ 2 1 6 は、シェーダのセット(コンパイルされたシェーダとして)を、GPU 2 0 4 によって実行可能なマシン・コードにコンパイルしてもよい。ブロック 2 3 0 におけるコンパイルは、シェーダのバイトコード表現を、GPU 2 0 4 によって実行可能であり、且つ GPU 2 0 4 の特定のモデルに固有のシェーダの低レベルのマシン・コード表現にコンパイルすることを含んでもよい。ブロック 2 3 0 におけるコンパイルは、ゲームがロードされる時、及び/又はゲームプレイ中に実行されてもよい。

【0041】

2 3 2 において、グラフィックス・ドライバ 2 1 6 は、コンパイルされたシェーダ(コンパイルされたシェーダ・コード 1 1 6)を、所定のメモリ・ロケーション 2 1 8 に、ビデオゲームのアプリケーション ID と関連付けてキャッシュしてもよい。これは、「アプリケーション ID 2 2 0 を有するコンパイルされたシェーダ・コード」として図 2 に示されている。いくつかの実施形態において、ビデオゲーム・クライアント 2 1 2 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)を使用して)グラフィックス・ドライバ 2 1 6 に、コンパイルされたシェーダ・コード 1 1 6 を事前に定義されたメモリ・ロケーション 2 1 8 に格納するように指示してもよいが、これは、一般的なシステムロケーションがディスク上のキャッシュに格納するために一般的に使用するのとは異なるロケーションであってもよい。コンパイルされたシェーダ・コード 1 1 6 を定義済みメモリ・ロケーション 2 1 8 に格納するためのこの命令は、ブロック 2 3 2 でのキャッシュ

40

50

と同時に発生してもよいし、ブロック 2 3 2 でのキャッシュに先立つある時点で発生してもよい。グラフィックス・ドライバ 2 1 6 がコンパイルされたシェーダ・コード 2 1 6 を格納するための所定のメモリ・ロケーション 2 1 8 は、問題となっている特定のビデオゲームに関連付けられたディレクトリであってもよい。このようにして、コンパイルされたシェーダ・コード 2 1 6 は、異なるビデオゲームに対応するディレクトリ（アプリケーション ID でタグ付けされたディレクトリ）に格納されてもよい。

【 0 0 4 2 】

ブロック 2 3 4 において、ビデオゲームは、停止又は終了されてもよい（例えば、ユーザ 1 0 2 がゲームを終了することに応答して）。このように、ブロック 2 3 4 において、クライアント・マシン 1 0 4 は、ビデオゲームの実行を停止してもよい。この時点で、ビデオゲームの実行中にコンパイルされた任意のシェーダについては、ビデオゲームのアプリケーション ID と関連して、所定のメモリ・ロケーション 2 1 8 にキャッシュされた対応するコンパイルされたシェーダ・コード 1 1 6 が存在するであろう。

10

【 0 0 4 3 】

2 3 6 において、ビデオゲーム・クライアント 2 1 2 は、ビデオゲームの実行を停止することに応答して、ビデオゲームの実行中にコンパイルされたシェーダのためのシェーダ ID を生成する（又はそうでなければ取得する）ために、キャッシュされたシェーダ・コード 1 1 6 を開いて読み出し／解析してもよい（例えば、所定のメモリ・ロケーション 2 1 8 内のフォルダを開いてその内容を読み出し／解析することによって）。いくつかの実施形態において、シェーダ ID は、シェーダキーの形式であり、それは、グラフィックス・ドライバ 2 1 6 のキャッシュによって内部的に生成され、その後、ブロック 2 3 6 のキャッシュエントリからビデオゲーム・クライアント 2 1 2 によって読み出される。いくつかの実施形態において、シェーダ ID は、ビデオゲームがダウンロードされたときに、コンパイルされていないシェーダ・コードと共に提供され、これらのシェーダ ID は、その後、ブロック 2 3 6 のキャッシュエントリから読み出される。いくつかの実施形態において、シェーダ ID は、同じ方法でシェーダをコンパイルする同じハードウェア構成を有する全てのクライアント・マシン 1 0 4 が同じシェーダ ID（又はシェーダキー）を生成するような決定論的な方法で、コンパイルされたシェーダ・コード 1 1 6 に基づいてビデオゲーム・クライアント 2 1 2 によって生成される。

20

【 0 0 4 4 】

サブブロック 2 3 8 に示すように、ビデオゲーム・クライアント 2 1 2 は、また、キャッシュエントリから読み出されたコンパイルされたシェーダ用のマシン・コードをハッシュ化してもよく、このハッシュは、タプル(tuple)のセット（各タプルは、シェーダ ID / キー及びシェーダ用のコンパイルされたマシン・コードのハッシュを具備する）内のシェーダ ID（又はシェーダキー）とともにリモート・システム 1 0 6 に送信されてもよい。

30

【 0 0 4 5 】

ブロック 2 4 0 において、クライアント・マシン 1 0 4 は、情報 1 1 0 をリモート・システム 1 0 6 に送信（又はアップロード）してもよく、その情報は、クライアント・マシン 1 0 4 によってコンパイルされたシェーダのシェーダ ID の少なくともセットを含む。ブロック 2 4 0 で送信される情報 1 1 0 は、クライアント・マシン 1 0 4 のハードウェア構成、ビデオゲームのアプリケーション ID、及び／又はコンパイルされたシェーダのマシン・コードのハッシュをさらに含んでもよい。この情報 1 1 0 のいずれか又は全てが、受信時にリモート・システム 1 0 6 によって消費される「シェーダ使用レポート」として送信されてもよい。一例において、単一のクライアント・マシン 1 0 4 によって送信される単一のシェーダ使用レポートは、特にビデオゲームがクライアント・マシン 1 0 4 上で初めてプレイされる場合、単一のビデオゲームのための数万個を超すシェーダに関する情報（例えば、シェーダ ID など）を含んでもよい。ブロック 2 2 6 に続く「NO」ルートに由来するプロセス 2 0 0 の分岐は、リモート・システム 1 0 6 でシェーダ関連情報を収集するための「テレメトリ」アプローチを表す。多数のクライアント・マシン 1 0 4 がプロセス 2 0 0 を実行している可能性があることを考慮すると、リモート・システム 1 0 6

40

50

は、異なるハードウェア構成を有する膨大な数のクライアント・マシン 104 からブロック 240 でアップロードされる説明された情報 110 を収集してもよい。

【0046】

図 2 は、クライアント・マシン 104 がブロック 234 でプログラム（例えば、ビデオゲーム）の実行を停止した後、情報 110 がブロック 240 でリモート・システム 106 に送信される（又はアップロードされる）シナリオを描いているが、情報 110 は、リアルタイムで、又は実質的にリアルタイムで、リモート・システム 106 に送信され得る（例えば、ストリーミングによって）ことが理解されるであろう。例えば、ブロック 240 で情報 110 をアップロードする前にブロック 234 でゲームを終了するのを待つ代わりに、クライアント・マシン 104 は、グラフィックス・ドライバがシェーダ・コードをコンパイルしてキャッシュするように、リアルタイムで情報 110 をストリーミング又はその他の方法で送信してもよい。

10

【0047】

ブロック 226 に戻り、リモート・システム 106 が、要求するクライアント・マシン 104 がプリキャッシュするために利用可能なコンパイルされたシェーダ・コードを持っている場合、プロセス 200 は、ブロック 226 からブロック 242 までの「YES」ルートを進んでもよく、ここで、クライアント・マシン 104 は、当該ビデオゲームのハードウェア構成及びアプリケーション ID に対応するコンパイルされたシェーダ・コード 116 を受信（又はダウンロード）し、クライアント・マシン 104 は、コンパイルされたシェーダ・コード 116 の少なくとも一部を所定のメモリ・ロケーション 218 にプリキャッシュ（又はその後の使用のために保存）する。コンパイルされたシェーダ・コード 116 をダウンロードするため、リモート・システム 106 は、クライアント・マシン 104 に、コンパイルされたシェーダ・コード 116 をダウンロードするためにどこに行けばよいかをクライアント・マシン 104 に伝えるための情報（例えば、識別子、ユニフォーム・リソース・ロケータ（URL）など）を提供してもよい。クライアント・マシン 104 は、コンパイルされたシェーダ・コード 116 を圧縮された形式（例えば、zip ファイル）でダウンロードしてもよい。

20

【0048】

244 において、クライアント・マシン 104 は、ビデオゲームを実行してもよく、ここで、ビデオゲームを実行することは、クライアント・マシン 104 のグラフィックス・ドライバ 216 がコンパイルされたシェーダ・コード自体を作成する代わりに、GPU 204 が、所定のメモリ・ロケーション 218 から検索されたコンパイルされたシェーダ・コード 116 の少なくとも一部を実行することを含む。すなわち、ブロック 242 におけるプリキャッシングは、クライアント・マシン 104 がシェーダ自体をコンパイルしなければならないことから救済する。その代わりとして、ダウンロードされたシェーダ・コード 116 は、プログラム実行のパフォーマンスを最適化するために、ゲーム実行中に利用することができる。

30

【0049】

図 3 は、リモート・コンピューティング・システム 106 の例示的な構成要素を示すブロック図であり、クライアント・マシン 104 から情報 110 を受信し、受信した情報 110 に含まれるシェーダ ID をカタログ化し、コンパイルされたシェーダ・コード 116 を選択的に要求し、再配布のためにコンパイルされたシェーダ・コード 116 を選択的に準備するための例示的なプロセス 300 のフロー図である。図示された実装において、リモート・システム 106 は、他の構成要素の中で、1 つ又は複数のプロセッサ 302、メモリ 304（又は非一時的なコンピュータ可読媒体 304）、及び通信インターフェース（複数可）306 を含む。メモリ 304（又は非一時的なコンピュータ読み取り可能媒体 304）は、コンピュータ読み取り可能命令、データ構造、プログラムモジュール、又は他のデータなどの情報を記憶するための任意の方法又は技術で実装された揮発性及び非揮発性メモリ、取り外し可能及び非取り外し可能な媒体を含むことができる。このようなメモリは、RAM、ROM、EEPROM、フラッシュメモリ又は他のメモリ技術、CD-

40

50

R O M、デジタル多目的ディスク (D V D) 又は他の光学ストレージ、磁気カセット、磁気テープ、磁気ディスクストレージ又は他の磁気ストレージ装置、R A I Dストレージシステム、又は所望の情報を格納するために使用することができ、コンピューティング装置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が、これらに限定されるものではない。コンピュータ読み取り可能媒体 3 0 4 は、コンピュータ読み取り可能記憶媒体 (「 C R S M 」) として実装されてもよく、これは、メモリ 3 0 4 に格納された命令を実行するためにプロセッサ (複数可) 3 0 2 によってアクセス可能な任意の利用可能な物理媒体であってもよい。一基本的な実装において、C R S M は、ランダム・アクセス・メモリ (「 R A M 」) 及びフラッシュメモリを含んでもよい。他の実装において、C R S M は、リード・オンリー・メモリ (「 R O M 」) 、電氣的に消去可能なプログラマブル・リード・オンリー・メモリ (「 E E P R O M 」) 、又は所望の情報を格納するために使用することができ、プロセッサ (複数可) 3 0 2 によってアクセス可能な任意の他の有形媒体を含んでもよいが、これらに限定されない。シェーダ配布モジュール 3 0 8 は、プロセッサ (複数可) 3 0 2 によって実行されると、リモート・システム 1 0 6 がここに記載された技術及び操作を実行させ、メモリ 3 0 4 に格納された命令を表してもよい。

10

【 0 0 5 0 】

ここで、図 3 に示されたプロセス 3 0 0 を参照する。3 1 4 において、リモート・システム 1 0 6 は、クライアント・マシン 1 1 4 から情報 1 1 0 を受信してもよい (例えば、上述したプロセス 2 0 0 のブロック 2 2 6 からの「 N O 」ルートに続く「テレメトリ」プランチの一部として)。特定のクライアント・マシン 1 0 4 から送信される受信情報 1 1 0 は、クライアント・マシン 1 0 4 上でのビデオゲームの実行中にクライアント・マシン 1 0 4 のグラフィックス・ドライバ 2 1 6 によってコンパイルされたシェーダのシェーダ I D のセットを少なくとも含んでもよい。さらに、ブロック 3 1 4 で受信した情報 1 1 0 は、クライアント・マシン 1 0 4 のハードウェア構成、実行されたビデオゲームのアプリケーション I D、コンパイルされたシェーダのためのマシン・コードのハッシュ、クライアント・マシン 1 0 4 のデバイス I D などの「シェーダ使用レポート」の追加情報を含んでもよい。

20

【 0 0 5 1 】

3 1 6 において、リモート・システム 1 0 6 は、ブロック 3 1 4 で受信したシェーダ I D のセットを、ハードウェア構成及びアプリケーション I D によって (又はそれに応じて) カタログ化してもよい。例えば、カタログ 1 1 2 は、アプリケーション I D とハードウェア構成のユニークな組み合わせによって分類された複数のグループ又はバケット 3 1 0 (1) ~ (M) を含んでもよい。例えば、ハードウェア構成「 1 」を有する第 1 クライアント・マシン 1 0 4 (1) を例にとると、ブロック 3 1 4 で第 1 クライアント・マシン 1 0 4 (1) からシェーダ I D の第 1 セットを受信した場合、ブロック 3 1 4 でシェーダ I D の第 1 セットが第 1 バケット 3 1 0 (1) (又はグループ) にカタログ化され、このバケット 3 1 0 (1) は、ブロック 3 1 4 で第 1 クライアント・マシン 1 0 4 (1) から受信した情報 1 1 0 より抽出されたアプリケーション I D と第 1 クライアント・マシン 1 0 4 (1) のハードウェア構成とでタグ付けされてもよい。同様に、第 2 クライアント・マシン 1 0 4 (2) は、シェーダ I D の第 2 セットを送信してもよく、このシェーダ I D の第 2 セットは、ブロック 3 1 4 で第 2 バケット 3 1 0 (2) にカタログ化されてもよく、これはブロック 3 1 4 で第 2 クライアント・マシン 1 0 4 (2) から受信した情報 1 1 0 から抽出されたアプリケーション I D と第 2 クライアント・マシン 1 0 4 (2) の第 2 ハードウェア構成「 2 」とでタグ付けされてもよい。これは、テレメトリ情報を報告するクライアント・マシン 1 0 4 の数に応じて、それらのクライアント・マシン 1 0 4 が異なるハードウェア構成を有するかどうかに応じて、及び / 又はそれらのクライアント・マシン 1 0 4 上で実行されているゲームの数に応じて、任意の数の「 M 」バケットについて継続することができる。

30

40

【 0 0 5 2 】

ブロック 3 1 8 において、リモート・システム 1 0 6 は、受信したシェーダ I D のいず

50

れかについてコンパイルされたシェーダ・コードを要求するかどうかを決定してもよい。ブロック 318 でこの決定を行うためのアルゴリズムは、以下でより詳細に説明されるように、様々であってもよい。

【0053】

リモート・システム 106 が、ブロック 318 において、当面の間、コンパイルされたシェーダ・コードを要求しないと決定した場合、プロセス 300 は、ブロック 318 からブロック 314 に戻る「NO」ルートを進んでもよく、ここでリモート・システム 106 はクライアント・マシン 104 から情報 110（例えば、シェーダ使用レポート）を収集／受信し続ける。リモート・システム 106 が受信したシェーダ ID（複数可）に対してコンパイルされたシェーダ・コードを要求することを決定する場合、プロセス 300 は、

10

【0054】

322 において、リモート・システム 106 は、特定のクライアント・マシン 104 から要求したコンパイルされたシェーダ・コード 116 を受信してもよく、リモート・システム 106 は、対応するシェーダ ID（複数可）に関連して（例えば、カタログ 112 に）コンパイルされたシェーダ・コードを格納してもよい。説明のために、リモート・システム 106 が、第 1 バケット 310（1）内のシェーダ ID の 1 つ以上についてコンパイルされたシェーダ・コードを要求し、この要求が、第 1 バケット 310（1）がタグ付けされたハードウェア構成を有するクライアント・マシン 104 に送信された場合、リモート・システム 106 は、受信したシェーダ・コードを第 1 バケット 310（1）内に、対応するシェーダ ID（複数可）と関連づけて格納してもよい。

20

【0055】

サブブロック 324 によって示されるように、リモート・システム 106 は、受信したシェーダ・コードが有効なコードであることを確認するために、受信したシェーダ・コードを格納する前に検証ハッシュを計算してもよい。すなわち、リモート・システム 106 は、ブロック 322 でクライアント・マシン 104 から受信したコンパイルされたシェーダ・コードをハッシュ化して、コンパイルされたシェーダ・コードの検証ハッシュを生成してもよい。ブロック 314 において、コンパイルされたシェーダ・コードのハッシュが、情報 110 で受信されることを想起されたい。したがって、ブロック 324 で計算された検証ハッシュは、カタログ 112 内の対応するシェーダ ID を持つタプルに格納されている以前に受信されたハッシュと比較されてもよい。検証ハッシュが、クライアント・マシン 104 から以前に受信されてカタログ化されたコンパイル済みシェーダ・コードのハッシュと一致する場合、そのコードは有効／認証されたものとみなすことができ、コンパイルされたコードは、ブロック 322 においてリモート・システム 106 によって格納されてもよい。

30

【0056】

ブロック 326 において、リモート・システム 106 は、コンパイルされたシェーダ・コードを「再配布 312 用に準備されたコンパイルされたシェーダ・コード」として再配布のために準備するかどうかを決定してもよい。ブロック 326 において決定を行うためのアルゴリズムは、以下でより詳細に説明されるように、変形されてもよい。

40

【0057】

リモート・システム 106 が、ブロック 326 において、当面の間、再配布のためにコンパイルされたシェーダ・コードを準備しないと決定した場合、プロセス 300 は、ブロック 326 からブロック 314 に戻る「NO」ルートを進んでもよく、ここでリモート・システム 106 がクライアント・マシン 104 からの情報 110（例えば、シェーダ使用レポート）の収集／受信を継続する。リモート・システム 106 が再配布のためにコンパイルされたシェーダ・コードを準備することを決定すると、プロセス 300 は、ブロック

50

3 2 6 からブロック 3 2 8 への「YES」ルートを辿ってもよく、ここで、リモート・システム 1 0 6 は、コンパイルされたシェーダ・コードを対応するアプリケーション ID 及びハードウェア構成に関連付けることによって、「再配布 3 1 2 のために準備されたコンパイルされたシェーダ・コード」を作成してもよい。例において、リモート・システム 1 0 6 は、所定のアプリケーション ID 及びハードウェア構成（すなわち、所定のバケット 3 1 0 に対して）に対してコンパイルされたシェーダ・コードの再分配可能な表現（又はデータ構造）を構築してもよく、このコードは、要求に応じてクライアント・マシン 1 0 4 に配布する準備ができていない場所にアーカイブされてもよい。いくつかの実施形態において、コンパイルされたシェーダ・コードは、ZIP ファイルを作成するなど、コンパイルされたシェーダ・コードを圧縮することによって、再配布 3 1 2 のために準備されてもよい。

10

【0058】

図 4 は、クライアント・マシン 1 0 4 のハードウェア構成に基づいて、要求するクライアント・マシン 1 0 4 にコンパイルされたシェーダ・コードを送信するための例示的なプロセス 4 0 0 のフロー図である。図 3 及び図 4 のページ外参照「A」によって示されるように、プロセス 4 0 0 は、図 3 のブロック 3 2 8 からの継続であり、コンパイルされたシェーダ・コードがリモート・システム 1 0 6 によって準備され、それらの要求クライアント・マシン 1 0 4 上でのプリキャッシングのために要求クライアント・マシン 1 0 4 に利用可能になった後などである。

【0059】

20

ブロック 4 0 2 において、リモート・システム 1 0 6 は、クライアント・マシン 1 0 4 の特定のハードウェア構成に係り、特定のプログラム（例えば、ビデオゲーム）のアプリケーション ID に係るコンパイルされたシェーダ・コードに対する要求 1 1 8 の一部として、クライアント・マシン 1 0 4 から情報を受信してもよい。したがって、ブロック 4 0 2 で受信した情報は、要求するクライアント・マシン 1 0 4 のハードウェア構成と、クライアント・マシン 1 0 4 にインストールされているか、又は現在インストールされている可能性のあるプログラム（例えば、ビデオゲーム）に対応するアプリケーション ID とを含んでもよい。

【0060】

4 0 4 において、リモート・システム 1 0 6 は、クライアント・マシン 1 0 4 のハードウェア構成に対応するコンパイルされたシェーダ・コードが利用可能であるかどうかと、要求に関連付けられたアプリケーション ID とを決定してもよい。一例において、リモート・システム 1 0 6 は、準備されたシェーダ・コード 3 1 2 を分析し、再配布 3 1 2 のために準備されたコンパイルされたシェーダ・コードのいずれかが、一致するハードウェア構成及び一致するアプリケーション ID に関連付けられているかどうかを判断してもよい。一致するハードウェア構成と一致するアプリケーション ID が見つかった場合、これは、一致するハードウェア構成を有する別のクライアント・マシン 1 0 4 が以前に同じプログラム（例えば、ビデオゲーム）を実行し、そのプログラムの実行中にシェーダ・コードをコンパイルしたことを示し、また、リモート・システム 1 0 6 がその後、コンパイルされたシェーダ・コードを受信し、再配布のために準備したことを示している。このような場合、プロセス 4 0 0 は、ブロック 4 0 4 からブロック 4 0 6 への「YES」ルートを辿ってもよく、ここで、リモート・システム 1 0 6 は、そのクライアント・マシン 1 0 4 上でアプリケーション ID を有するビデオゲームを実行しながら、クライアント・マシン 1 0 4 の GPU 2 0 4 によって実行されるように、コンパイルされたシェーダ・コードを要求元のクライアント・マシン 1 0 4 に送信してもよい。

30

40

【0061】

一方、ブロック 4 0 4 において、リモート・システム 1 0 6 が、要求するクライアント・マシン 1 0 4 の特定のハードウェア構成及びアプリケーション ID に対して利用可能なコンパイルされたシェーダ・コードがないと判断した場合、プロセス 4 0 0 は、ブロック 4 0 4 からブロック 4 0 8 への「NO」ルートを辿ってもよく、ここで、リモート・シス

50

テム106は、要求されたシェーダ・コードが利用できないことをクライアント・マシンに通知してもよい。

【0062】

図5は、コンパイルされたシェーダ・コードを選択的に要求するための例示的なプロセス500のフロー図である。示されているように、プロセス500は、図3のプロセス300のブロック318における操作（複数可）のサブプロセスを表してもよく、プロセス500は、リモート・システム106が少なくとも1つのシェーダID（又はシェーダキー）を受信した後に実行されてもよい。

【0063】

502において、クライアント・マシン104上でビデオゲームを実行した結果、クライアント・マシン104からシェーダID（又はシェーダキー）を含むクライアント情報110を受信した後、リモート・システム106は、クライアント・マシン104から受信した所定のシェーダIDについて、リモート・システム106が、少なくとも所定の数の異なるクライアント・マシン104からそのシェーダIDを受信したかどうかを判断してもよい。予め定義された数は、設定変更可能である。リモート・システム106が、少なくとも予め定義された数の異なるクライアント・マシン104から同じシェーダIDを受信した場合、プロセス500は、ブロック502からブロック504への「YES」ルートを進んでもよく、ここで、リモート・システム106は、シェーダIDをリモート・システム106に送信したクライアント・マシン104に対して、シェーダIDに対応するコンパイルされたシェーダ・コード116を要求する要求114を送信してもよい。

【0064】

一方、リモート・システム106が、少なくとも所定の数の異なるクライアント・マシン104から同じシェーダIDを受信していない場合、プロセス500は、ブロック502からブロック506への「NO」ルートを進んでもよく、ここで、リモート・システム106は、追加のクライアント・マシン104がシェーダIDを送信するのを待ち、新たに受信したシェーダIDに続いてプロセス500を反復することができる。

【0065】

ブロック502における予め定義された数は設定変更可能である。予め定義された数が2つである場合、リモート・システム106は、2つの異なるクライアント・マシン104から同じシェーダIDを受信すると直ぐに、コンパイルされたシェーダ・コード116を要求する。所定の数が2より大きい場合、リモート・システム106は、コンパイルされたシェーダ・コード116を要求する前に、より多くのクライアント・マシン104が同じシェーダIDを送信するまで待機する。いくつかの実施形態において、ブロック504でコンパイルされたシェーダ・コードを要求する前に満たされる追加の基準は、異なるクライアント・マシンが同じハードウェア構成を有することであってもよい。すなわち、「2」に設定された所定の数について、コンパイルされたシェーダ・コード504の要求は、リモート・システム106が、同じハードウェア構成（例えば、同じGPUモデル及び同じバージョンのグラフィックス・ドライバ）を有する2つの異なるクライアント・マシン104から同じシェーダID（又はシェーダキー）を受信したときにトリガされてもよい。このように、プロセス500は、対応するシェーダIDの受信時に直ちにコンパイルされたシェーダ・コードを要求するのではなく、選択的にコンパイルされたシェーダ・コードを要求する例である。

【0066】

図6は、再配布のためにコンパイルされたシェーダ・コードを選択的に準備するための例示的なプロセス600のフロー図である。示されているように、プロセス600は、図3のプロセス300のブロック326における操作のサブプロセスを表してもよく、プロセス600は、リモート・システム106が1つ以上のクライアント・マシン104からコンパイルされたシェーダ・コードを受信した後に実行されてもよい。

【0067】

602において、クライアント・マシン104からコンパイルされたシェーダ・コード

10

20

30

40

50

116を要求して受信した後、リモート・システム106は、コンパイルされたシェーダ・コード116を送信したクライアント・マシン104に関連付けられたユーザアカウントが信頼されたアカウントであるかどうかを判断してもよい。例えば、リモート・システム106は、ブロック602において照合可能な信頼されたユーザアカウントのリストを保持してもよい。これらの信頼されたユーザアカウントは、問題のビデオゲームのライセンスを合法的に取得したユーザ/クライアント・マシンに関連付けられていてもよい。さらに、又は代替的に、信頼されたユーザアカウントは、ゲーム出版社又は開発者（例えば、リリース日に先立ってビデオゲームを実行するユーザ）、及び/又は、信頼されたユーザが一般にリリースされる前にビデオゲームのバグ及び他の問題についてテストを行うことを任務としているテストラボを運営するユーザのアカウントに関連付けられていてもよい。リモート・システム106は、そのようなユーザを自動的に信頼するように構成されており、したがって、そこから受信したコンパイルされたシェーダ・コード116は、そのように構成されている。

10

【0068】

ブロック602における決定が、コンパイルされたシェーダ・コード116が信頼されたユーザアカウントから受信されたものである場合、プロセス600は、ブロック602からブロック604への「YES」ルートを辿ってもよく、ここで、ポピュラリティ閾値が1に設定されてもよい。そうでなければ、リモート・システム602が、コンパイルされたシェーダ・コードを送信したユーザ102及び/又はマシン104を信頼できない場合、プロセス600は、ブロック602からブロック606への「NO」ルートを辿ってもよく、ここで、ポピュラリティ閾値が1よりも大きい数字（例えば、ポピュラリティ閾値10）に設定されてもよい。

20

【0069】

608において、リモート・システム106は、特定のハードウェア構成及び特定のアプリケーションIDを有するグループ（又はバケット310）にカタログ化された特定のシェーダID（又はシェーダキー）について、その特定のシェーダIDをリモート・システム106に送信した異なるクライアント・マシン104の数をカウントしてもよい。換言すると、リモート・システム106は、同じシェーダID（又はシェーダキー）及び同じアプリケーションIDを報告している同じハードウェア構成を有する異なるクライアント・マシン104の数を監視してもよい。

30

【0070】

610において、リモート・システム106は、その数（又は同じシェーダIDを報告する異なるクライアント・マシン104の数）がポピュラリティ閾値を満たすか、又は超えるかを判断してもよい。その数がブロック610においてポピュラリティ閾値を満たさない場合、プロセス600は、ブロック610からブロック608に戻る「NO」ルートを辿ってもよく、ここでリモート・システム106は、シェーダIDを報告するクライアント・マシン104の数をカウントし続ける（又は他の方法で監視し続ける）。その数（すなわち、同じシェーダIDを報告する異なるクライアント・マシン104の数）がブロック610でポピュラリティ閾値を満たすか、又は超える場合、プロセス600は、ブロック610からブロック612への「YES」ルートを辿ってもよく、ここで、受信されたシェーダ・コードが、1つ以上の追加のクライアント・マシン104から受信されたコンパイルされたシェーダ・コードと一致するかどうかを決定するために、オプションの検証ステップを実行することができる。換言すると、リモート・システム106は、同じシェーダIDを報告した複数の異なるクライアント・マシン104から同じコンパイルされたシェーダ・コード116であると思われるものを要求して受信し、ブロック612で異なるクライアント・マシン104から受信したコンパイルされたシェーダ・コード116を比較してもよい。異なるクライアント・マシン104からのコンパイルされたシェーダ・コードの比較に基づいて、ブロック612で検出された競合がある場合、プロセス600は、ブロック612からブロック614への「NO」ルートを辿ってもよく、ここで検出された競合に基づき受信されたシェーダ・コードが破棄されてもよい。同じハードウ

40

50

ア構成を有するクライアント・マシン104からのシェーダ・コードの不一致は、クライアント・マシン104のうちの1つが「フェイク」コードを提供していることを示している可能性があり、このコードは、破損しているか、又はマルウェアを含んでいる可能性がある。

【0071】

受信されたシェーダ・コードが異なるクライアント・マシンからのものと同じ（一致する）であると判断することによって、ブロック612での検証ステップが合格である場合、プロセス600は、ブロック612からブロック616への「YES」ルートを辿り、ここでリモート・システム106は、再配布可能なデータ構造を構築し、（再配布可能なデータ構造内の）コンパイルされたシェーダ・コードを対応するアプリケーションID及びハードウェア構成に関連付けることによって、再配布のためにコンパイルされたシェーダ・コードを準備する（図3の312参照）。

10

【0072】

図6のポピュラリティ閾値は設定変更可能であるため、リモート・システム106は、コミュニティの知覚された信頼及び/又は、意図せずに破損したコード又は悪意のあるコードを配布するリスク及び/又は感度に応じて調整することができる。また、ブロック604（ユーザアカウントが信頼されたアカウントであると決定されたときに生じる）において、ポピュラリティ閾値を1に設定することは、信頼されたユーザから受信したコンパイルされたシェーダ・コードが自動的に又は直ちに再配布のために準備されるプロセス600の抜粋版において効果的に結果をもたらすことが理解されよう、なぜならリモート・システム106は、そのシェーダ・コードに対応する同じシェーダIDを他のクライアント・マシンが送信するのを待たず、リモート・システム106は、信頼されたユーザアカウントから受信したシェーダ・コードを検証するために、他のシェーダ・コードを持たない（必要性がない）ためである。

20

【0073】

図7は、ダウンロードされたシェーダ・コードの一部を破棄し、破棄されたシェーダ・コードに対応するシェーダを再コンパイルし、再コンパイルされたシェーダのための情報をアップロードすることによって完全性チェックを実施するための例示的なプロセス700のフロー図である。

【0074】

702において、クライアント・マシン104は、ここに記載されるように、リモート・システム106からコンパイルされたシェーダ・コードを要求した後に、クライアント・マシン104のハードウェア構成のための、及び問題のプログラム（例えば、ビデオゲーム）のアプリケーションIDのためのコンパイルされたシェーダ・コード116を受信してもよい。

30

【0075】

704において、クライアント・マシン104は、コンパイルされたシェーダ・コード116の少なくとも一部を破棄されたシェーダ・コードとして破棄してもよい。例えば、クライアント・マシン104は、クライアント・マシン104のランダムなサブセットの一部であってもよく、（例えば、ビデオゲーム・クライアント212がリモート・システム106から命令を受信することを介して、及び/又はハードコード化された命令などを介して）リモート・システム106から受信したコンパイルされたシェーダの一部（例えば、10%）を、「良好な」シェーダ・コード（すなわち、破損していない、及び/又はマルウェアなどを含まない）であるという前提にかかわらず、破棄するように構成されている。このようにシェーダ・コードを廃棄するクライアント・マシン104の数は、リモート・システム106によって設定変更可能であってもよく、ビデオゲームを実行しているユーザ人口の小さな部分（例えば、10%）を表してもよい。

40

【0076】

706において、クライアント・マシン104は、破棄されたシェーダ・コード以外のコンパイルされたシェーダ・コードの残りの部分をプリキャッシュしてもよい。コンパイ

50

ルされたシェーダ・コードの残りの部分は、所定のメモリ・ロケーション 2 1 8 に格納されてもよい。

【 0 0 7 7 】

7 0 8 において、クライアント・マシン 1 0 4 は、受信したシェーダ・コードに対応するシェーダを使用するビデオゲームの実行を開始してもよく、そのうちの少なくとも一部は、所定のメモリ・ロケーション 2 1 8 にキャッシュされる。

【 0 0 7 8 】

7 1 0 において、クライアント・マシン 1 0 4 は、破棄されたシェーダ・コードに対応するシェーダをコンパイルしてキャッシュしてもよい。これは、ローディング画面の間、又はリアルタイムでのゲームの実行中に行うことができる。ブロック 7 0 6 においてプリ
10
キャッシュされた受信されたシェーダ・コードに対応するシェーダを利用する時が来ると、クライアント・マシン 1 0 4 の GPU 2 0 4 は、グラフィックス・ドライバ 2 1 6 が対応するシェーダを独自にコンパイルする代わりに、コンパイルされたシェーダ・コードを実行し、これによりクライアント・マシン 1 0 4 のリソースが節約される。

【 0 0 7 9 】

7 1 2 において、ビデオゲーム・クライアント 2 1 2 は、ユーザ 1 0 2 がビデオゲームを終了することに応答して、ゲームの実行を停止してもよい。

【 0 0 8 0 】

7 1 4 において、ビデオゲーム・クライアント 2 1 2 は、ビデオゲームの実行の停止に
20
応答して、新たにコンパイルされたシェーダ（ブロック 7 1 0 でコンパイルされたシェーダ）のためのキャッシュされたシェーダ・コード 1 1 6 を開いて読み出し / 解析して、それらの新たにコンパイルされたシェーダのためのシェーダ ID（又はシェーダキー）を生成してもよい。

【 0 0 8 1 】

サブブロック 7 1 6 で示されるように、ビデオゲーム・クライアント 2 1 2 は、新たに
コンパイルされたシェーダ用のマシン・コードをハッシュ化してもよく、このハッシュは、シェーダ ID（又はシェーダキー）とともに、タプル（シェーダ ID / キー及びシェー
ダ用にコンパイルされたマシン・コードのハッシュを具備する各タプル）のセットでリモ
ート・システム 1 0 6 に送信されてもよい。

【 0 0 8 2 】

ブロック 7 1 8 において、クライアント・マシン 1 0 4 は、情報 1 1 0 をリモート・シ
30
ステム 1 0 6 に送信（又はアップロード）してもよく、情報 1 1 0 は、少なくともクライ
アント・マシン 1 0 4 によってコンパイルされた新たにコンパイルされたシェーダのシェ
ーダ ID のセットを含む。ブロック 7 1 8 で送信される情報 1 1 0 は、クライアント・マ
シン 1 0 4 のハードウェア構成、ブロック 7 0 8 で実行されたビデオゲームのアプリケー
ション ID、及び / 又はコンパイルされたシェーダのマシン・コードのハッシュをさらに
含んでもよい。この情報 1 1 0 のいずれか又は全ては、受信時にリモート・システム 1 0
6 によって消費される「シェーダ使用レポート」で送信されてもよい。

【 0 0 8 3 】

図 8 は、図 7 のプロセス 7 0 0 の完全性チェックを使用して、既に配布されたシェーダ
40
・コードのさらなる配布を停止するための例示的なプロセス 8 0 0 のフロー図である。図
7 及び図 8 のページ外参照「B」は、プロセス 8 0 0 が、図 7 のプロセス 7 0 0 のブロッ
ク 7 1 8 から継続することができることを示している。

【 0 0 8 4 】

8 0 2 において、リモート・システム 1 0 6 は、クライアント・マシン 1 1 4 から情報
1 1 0 を受信してもよい（例えば、上述したプロセス 2 0 0 の「テレメトリ」分岐の一部
として）。プロセス 7 0 0 から続けて、ブロック 8 0 2 で受信したこの情報 1 1 0 は、送
信しているクライアント・マシン 1 0 4 によって（プロセス 7 0 0 のブロック 7 0 4 で）
破棄されたシェーダ・コードに対応する新たにコンパイルされたシェーダのシェーダ ID
のセットを少なくとも含んでもよく、このクライアント・マシン 1 0 4 は、信頼されたコ
50

ーザアカウント（例えば、システム上の完全性チェックの一部としてコンパイルされたシェーダを意図的に破棄するように指定された者）に関連付けられてもよい。さらに、ブロック 802 で受信した情報 110 は、クライアント・マシン 104 のハードウェア構成、実行されたビデオゲームのアプリケーション ID、コンパイルされたシェーダの全てのマシン・コードのハッシュ、クライアント・マシン 104 のデバイス ID、及び / 又はそのようなものなど、「シェーダ使用レポート」で提供された追加の情報を含んでもよい。

【0085】

804 において、リモート・システム 106 は、受信したシェーダ ID のうちの 1 つ以上に対応するコンパイルされたシェーダ・コードが、ここに記載されているようなプロセス 400 を実行し、送信されたシェーダ・コードに関する情報を記録することによって、1 つ以上のクライアント・マシン 104 に既に配布されていることを決定してもよい。

10

【0086】

806 において、リモート・システム 106 は、クライアント・マシン 104（すなわち、図 7 のプロセス 700 において、他のシェーダ・コードを破棄し、その破棄されたシェーダ・コードに対応するシェーダを再コンパイルしたのと同じクライアント・マシン 104）からコンパイルされたシェーダ・コードを要求してもよい。

【0087】

808 において、リモート・システム 106 は、クライアント・マシン 104 から要求したコンパイルされたシェーダ・コード 116 を受信してもよい。

【0088】

810 において、リモート・システム 106 は、受信したシェーダ・コード 116 を、既に配布されたシェーダ・コードと比較してもよい。

20

【0089】

812 において、リモート・システム 106 は、ブロック 810 における比較に基づいて、ブロック 808 においてクライアント・マシン 104 から受信したコンパイルされたシェーダ・コード 116 が、既に配布されたシェーダ・コードと一致するかどうかを決定してもよい。

【0090】

比較されたシェーダ・コードが一致する場合、プロセス 800 は、ブロック 812 からブロック 814 への「YES」ルートを進んでもよく、ここでリモート・システム 106 は、要求に応じてコンパイルされたシェーダ・コード 116 の配布を継続する。比較されたシェーダ・コードが一致しない場合、プロセス 800 は、ブロック 812 からブロック 816 への「NO」ルートを進んでもよく、ここで、リモート・システム 106 は、既に配布されたシェーダ・コードの更なる配布を停止してもよく、例えば、既に配布されたシェーダ・コードを追加のクライアント・マシン 104 に送信することを控えることによって、そのようなシェーダ・コードの配布を停止してもよい。

30

【0091】

図 7 及び図 8 の複合プロセスで説明されているように、そうでなければ「良い」（すなわち、破損していない、マルウェアを含まないなど）と考えられるシェーダ・コードの一部を意図的に破棄することは、システムにより多くの信頼を付加する。例えば、悪質な行為者は、リモート・システム 106 を介して再配布させるために、シェーダ・コード（「良い」シェーダ・コードのシェーダ ID と一致するシェーダ ID を有する）として偽装された悪質なコードを生成するクライアント・マシン 104 上のユーザアカウントのセットを作成しようとすることができる。しかしながら、悪意のあるコードは、実際の「良い」シェーダ・コードとは異なるので、リモート・システム 106 は、最終的にブロック 808 でクライアント・マシン 104 から「良い」シェーダ・コードを受け取り、ブロック 812 で不一致をキャッチして、悪意のあるコードのさらなる配布を停止させる。

40

【0092】

シェーダ・コードの一部を意図的に破棄することは、ビデオゲームのクラッシュの原因となり得る、又はゲームプレイ中の描画問題の原因となり得るコードの偶発的な破損（例

50

えば、ビットが反転されること)からもガードする。

【 0 0 9 3 】

主題は、構造的特徴に特有の言語で記載されてきたが、添付の特許請求の範囲に定義された主題は、必ずしも記載された特定の特徴に限定されるものではないことが理解されるべきである。むしろ、特定の特徴は、特許請求の範囲の実施形態の例示的な形態として開示されている。

10

20

30

40

50

【図面】

【図 1】

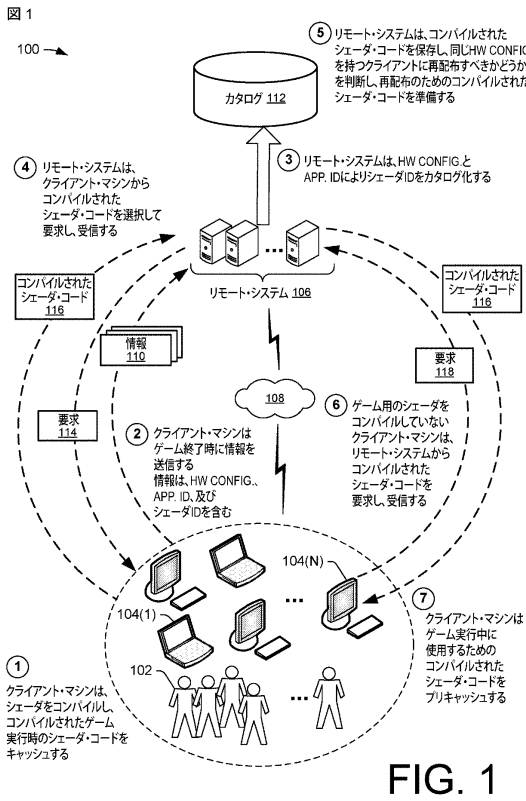


FIG. 1

【図 2】

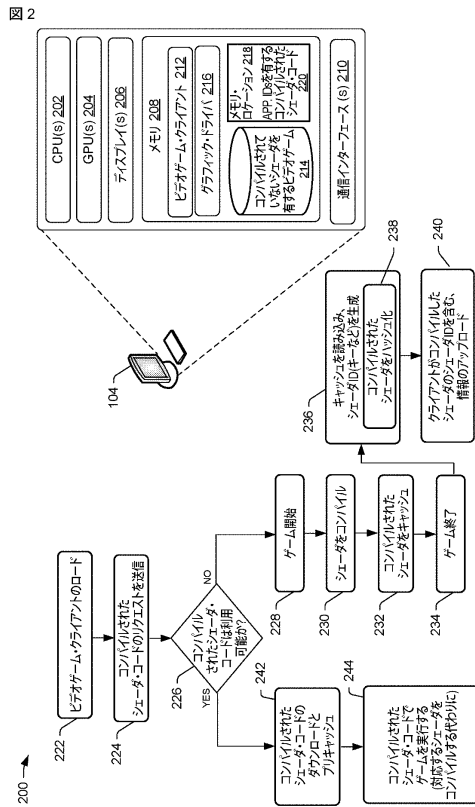


FIG. 2

【図 3】

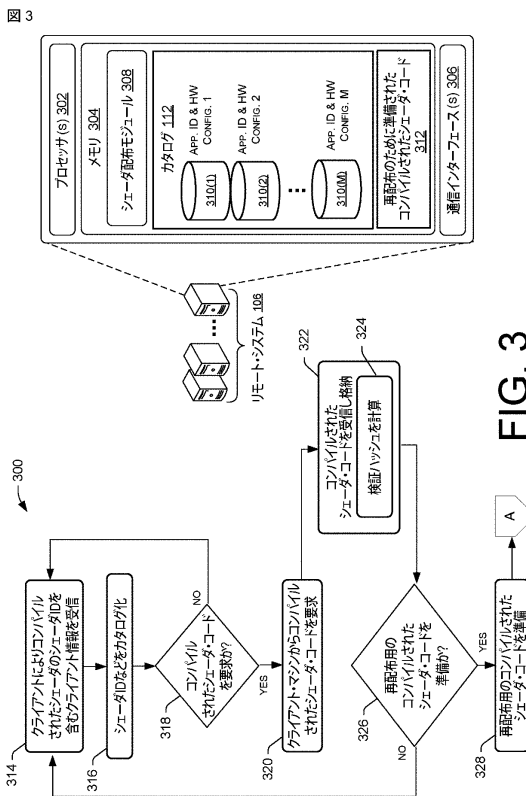


FIG. 3

【図 4】

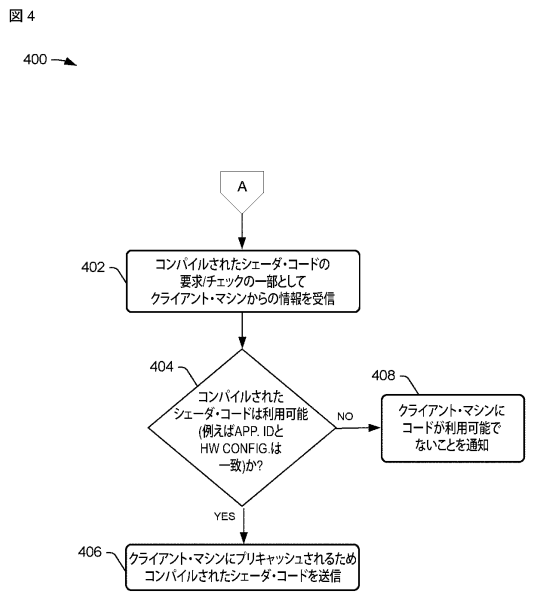


FIG. 4

10

20

30

40

50

【 図 5 】

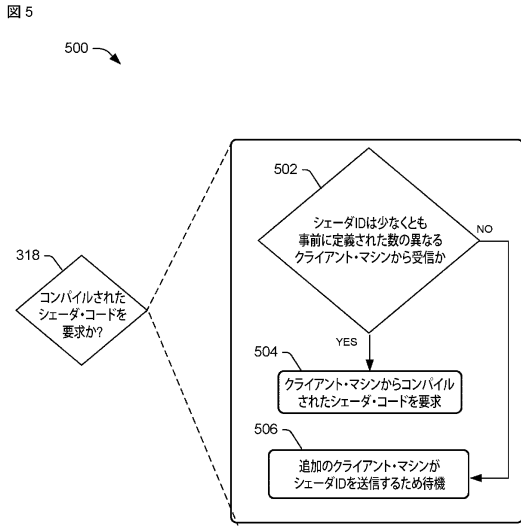


FIG. 5

【 図 6 】

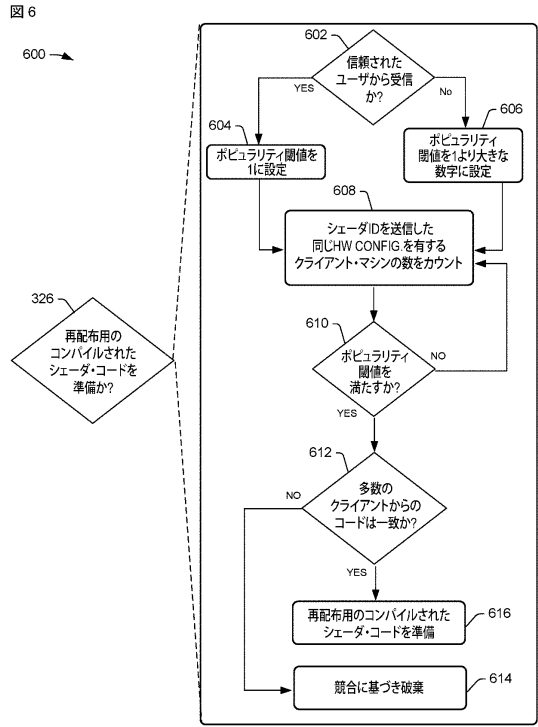


FIG. 6

【 図 7 】

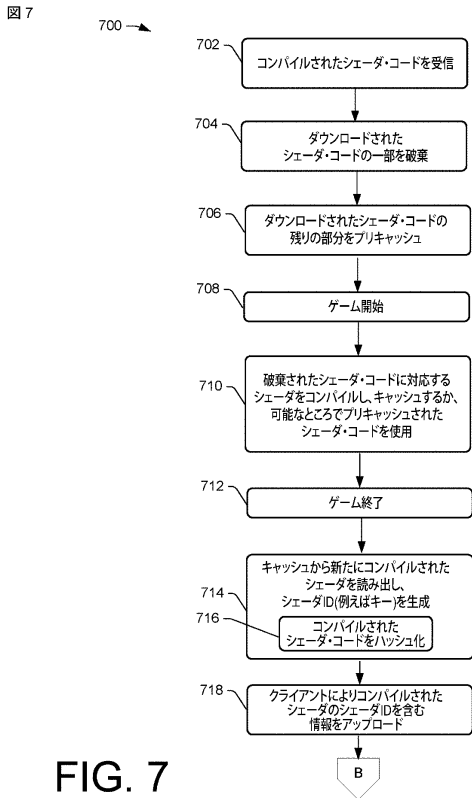


FIG. 7

【 図 8 】

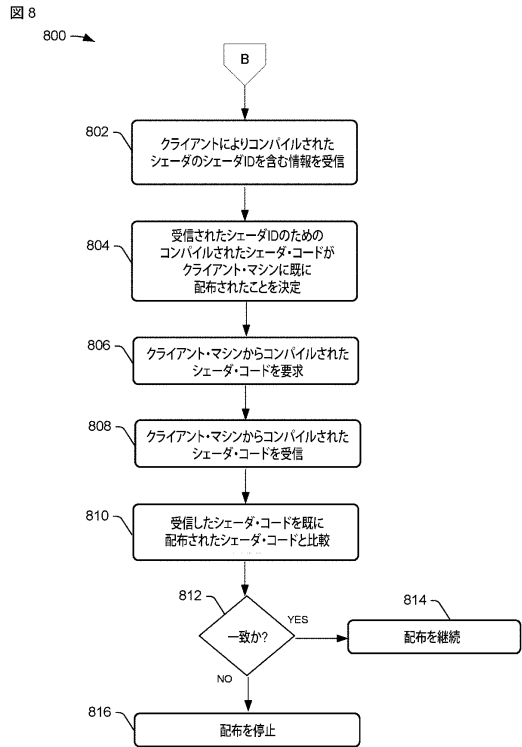


FIG. 8

10

20

30

40

50

フロントページの続き

- (56)参考文献 特表2013-514570(JP,A)
米国特許出願公開第2016/0070549(US,A1)
特開2009-098963(JP,A)
米国特許出願公開第2015/0022541(US,A1)
特開平11-328138(JP,A)

- (58)調査した分野 (Int.Cl., DB名)
- | | | | |
|------|-------|---|-------|
| G06F | 8/00 | - | 8/77 |
| G06F | 9/44 | - | 9/445 |
| G06F | 9/451 | | |
| G06T | 11/00 | - | 11/40 |
| G06T | 15/00 | - | 17/00 |
| G06T | 17/10 | - | 17/30 |
| A63F | 9/24 | - | 9/24 |
| A63F | 13/00 | - | 13/98 |
| H04L | 51/00 | - | 51/58 |
| H04L | 67/00 | - | 67/75 |