

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号

特許第7073494号

(P7073494)

(45)発行日 令和4年5月23日(2022.5.23)

(24)登録日 令和4年5月13日(2022.5.13)

(51)国際特許分類

F I

G 0 6 T 15/00 (2011.01)

G 0 6 T 15/00 5 0 1

G 0 6 F 12/00 (2006.01)

G 0 6 F 12/00 5 7 0

G 0 6 F 12/0875(2016.01)

G 0 6 F 12/0875 1 0 6

G 0 6 F 12/0806(2016.01)

G 0 6 F 12/0806 1 0 0

G 0 6 F 12/0842(2016.01)

G 0 6 F 12/0842

請求項の数 20 (全59頁) 最終頁に続く

(21)出願番号 特願2020-529148(P2020-529148)

(86)(22)出願日 平成31年4月9日(2019.4.9)

(65)公表番号 特表2021-517997(P2021-517997
A)

(43)公表日 令和3年7月29日(2021.7.29)

(86)国際出願番号 PCT/US2019/026611

(87)国際公開番号 WO2019/199848

(87)国際公開日 令和1年10月17日(2019.10.17)

審査請求日 令和2年12月8日(2020.12.8)

(31)優先権主張番号 62/655,688

(32)優先日 平成30年4月10日(2018.4.10)

(33)優先権主張国・地域又は機関
米国(US)

(73)特許権者 502208397

グーグル エルエルシー

G o o g l e L L C

アメリカ合衆国 カリフォルニア州 9 4

0 4 3 マウンテン ビュー アンフィシ

アター パークウェイ 1 6 0 0

1 6 0 0 Amphitheatre P

arkway 9 4 0 4 3 Mounta

in View, C A U . S . A .

(74)代理人 110001195

特許業務法人深見特許事務所

(72)発明者 ラロンド, ボール

アメリカ合衆国、9 4 0 4 3 カリフォ

ルニア州、マウンテン・ビュー、アンフ

ィシアター・パークウェイ、1 6 0 0

最終頁に続く

(54)【発明の名称】 ゲームレンダリングにおけるメモリ管理

(57)【特許請求の範囲】

【請求項1】

オンラインゲームセッションのために画像フレームをレンダリングする方法であって、
1つ以上のCPU上で、ゲームタイトルについて複数のオンラインゲームセッションを並
行して実行することと、

1つ以上のGPUが、複数のコンパイルされたシェーダの第1のシーケンスを用いて、前
記複数のオンラインゲームセッションの各々について複数の画像フレームをレンダリング
することと、

前記ゲームタイトルと関連付けられた前記複数のオンラインゲームセッションについて、
前記複数のコンパイルされたシェーダの第1のシーケンスの最適化において満たされる1
つ以上の性能基準を識別することと、

前記複数のオンラインゲームセッションの各々の実行を特徴付けるリアルタイム使用統計
を収集することと、

前記リアルタイム使用統計に基づいて、前記複数のオンラインゲームセッションについて
前記複数のコンパイルされたシェーダの第1のシーケンスの性能を前記1つ以上の性能基
準に関して最適化するために、前記複数のコンパイルされたシェーダの第1のシーケンス
から前記1つ以上の性能基準に従って選択された1つ以上のシェーダを修正することによ
って、前記複数のコンパイルされたシェーダの第1のシーケンスにおいて前記1つ以上の
シェーダが修正された、複数のコンパイルされたシェーダの修正された第1のシーケンス
を生成することを含む、方法。

【請求項 2】

メモリのシェーダキャッシュに格納された前記複数のコンパイルされたシェーダの第 1 のシーケンスを前記修正された第 1 のシーケンスと置換えることをさらに含む、請求項 1 に記載の方法。

【請求項 3】

前記複数のコンパイルされたシェーダの第 1 のシーケンスの第 1 のバージョンとして、前記複数のコンパイルされたシェーダの第 1 のシーケンスを前記メモリのシェーダキャッシュに格納することと、

前記複数のコンパイルされたシェーダの第 1 のシーケンスの第 2 のバージョンとして、前記修正された第 1 のシーケンスを前記シェーダキャッシュに格納することと、

前記複数のオンラインゲームセッションの各々の実行を特徴付ける次のリアルタイム使用統計を収集することと、

前記次のリアルタイム使用統計に従って、前記複数のコンパイルされたシェーダの第 1 のシーケンスの前記第 1 および第 2 のバージョンのうち 1 つを選択して、前記複数のオンラインゲームセッションの各々について前記複数の画像フレームをレンダリングすることとを含む、請求項 2 に記載の方法。

10

【請求項 4】

前記複数のコンパイルされたシェーダの第 1 のシーケンス内の前記 1 つ以上のシェーダを修正することはさらに、前記複数のオンラインゲームセッションの性能が前記 1 つ以上の性能基準を満たすように、代替的なシェーダオプションのグループから代替的なシェーダを選択して、前記複数のコンパイルされたシェーダの第 1 のシーケンス内の第 1 のシェーダを置換えることを含み、前記修正された第 1 のシーケンスは、前記代替的なシェーダを含む、請求項 1 ~ 3 のいずれか 1 項に記載の方法。

20

【請求項 5】

前記複数のコンパイルされたシェーダの第 1 のシーケンス内の第 1 のシェーダを修正するユーザ命令を受信することをさらに含み、前記第 1 のシェーダは、前記ユーザ命令にตอบสนองして、前記複数のコンパイルされたシェーダの第 1 のシーケンスで修正される、請求項 1 ~ 4 のいずれか 1 項に記載の方法。

【請求項 6】

前記複数のコンパイルされたシェーダの第 1 のシーケンスの性能を前記 1 つ以上の性能基準に関して最適化するために前記複数のコンパイルされたシェーダの第 1 のシーケンス内の第 1 のシェーダは自動的にユーザの介入なしで修正される、請求項 1 ~ 4 のいずれか 1 項に記載の方法。

30

【請求項 7】

前記修正された第 1 のシーケンスを生成することはさらに、前記複数のコンパイルされたシェーダの第 1 のシーケンスの性能を前記 1 つ以上の性能基準に関して最適化するために前記複数のコンパイルされたシェーダの第 1 のシーケンス内の第 2 のシェーダを修正することを含む、請求項 1 ~ 6 のいずれか 1 項に記載の方法。

【請求項 8】

前記複数のオンラインゲームセッションは前記 1 つ以上の性能基準に従ってレンダリングされる必要があると判断することと、

前記複数のオンラインゲームセッションの各々について、前記複数の画像フレームをレンダリングするために前記複数のコンパイルされたシェーダの第 1 のシーケンスを識別することとをさらに含む、請求項 1 ~ 7 のいずれか 1 項に記載の方法。

40

【請求項 9】

前記 1 つ以上の性能基準は第 1 の性能基準を含み、前記複数のオンラインゲームセッションは、前記第 1 の性能基準とは異なる第 2 の性能基準に従ってレンダリングおよび実行され、前記方法はさらに、

前記第 1 の性能基準を、前記修正された第 1 のシーケンスと関連付けることと、

前記第 2 の性能基準を、前記複数のコンパイルされたシェーダの第 1 のシーケンスと関連

50

付けることとを含む、請求項 1 ~ 8 のいずれか 1 項に記載の方法。

【請求項 10】

前記 1 つ以上の性能基準は第 1 の性能基準を含み、前記方法はさらに、
前記修正された第 1 のシーケンスを前記第 1 の性能基準と関連して前記メモリのシェーダ
キャッシュに格納することを含む、請求項 2 または 3 に記載の方法。

【請求項 11】

前記複数のオンラインゲームセッションを実行しつつ、
前記ゲームタイトルと関連付けられた前記複数のオンラインゲームセッションについて第
2 の性能基準を識別することと、
前記複数のコンパイルされたシェーダの第 1 のシーケンスの性能を前記第 2 の性能基準に
関して最適化するために複数のコンパイルされたシェーダの第 2 のシーケンスを判断する
ことと、
前記第 2 の性能基準と関連して、前記複数のコンパイルされたシェーダの第 2 のシーケ
ンスを前記シェーダキャッシュに格納することとをさらに含む、請求項 10 に記載の方法。

10

【請求項 12】

前記リアルタイム使用統計は前記複数のオンラインゲームセッションの全消費電力を含み
、前記第 1 の性能基準および前記第 2 の性能基準はそれぞれ、前記複数のオンラインゲー
ムセッションの応答時間および電力使用量を最適化し、前記方法はさらに、
前記複数のオンラインゲームセッションの前記全消費電力が事前に定義された電力閾値を
超えているという判断に従って、前記複数のコンパイルされたシェーダの第 2 のシーケ
ンスを選択し、前記複数のコンパイルされたシェーダの第 2 のシーケンスを用いて、前記複
数のオンラインゲームセッションの各々について前記複数の画像フレームをレンダリング
することを含む、請求項 11 に記載の方法。

20

【請求項 13】

複数の性能基準の各々を複数のコンパイルされたシェーダのそれぞれのシーケンスと関連
付けるシェーダハッシュテーブルを確立することとをさらに含み、前記確立することは、前
記シェーダハッシュテーブルにおいて前記 1 つ以上の性能基準と前記複数のコンパイルさ
れたシェーダの第 1 のシーケンスとを関連付けることとを含み、前記方法はさらに、
前記シェーダハッシュテーブルを前記メモリのシェーダキャッシュに格納することを含む
、請求項 2、3、または 10 ~ 12 のいずれか 1 項に記載の方法。

30

【請求項 14】

前記リアルタイム使用統計はユーザ入力情報を含み、前記方法はさらに、
前記シェーダハッシュテーブルにおいて、前記複数のコンパイルされたシェーダの第 1 の
シーケンスの複数のバージョンを前記ユーザ入力情報と関連付けることとを含む、請求項 1
3 に記載の方法。

【請求項 15】

前記リアルタイム使用統計に従って前記シェーダハッシュテーブルをダイナミックにアッ
プデートすることをさらに含み、前記アップデートすることは、前記シェーダキャッシュ
内の前記複数のコンパイルされたシェーダの第 1 のシーケンスを、前記修正された第 1 の
シーケンスと置換えることとを含む、請求項 13 に記載の方法。

40

【請求項 16】

前記リアルタイム使用統計は、前記複数のオンラインゲームセッションの各々に対応する
前記複数のコンパイルされたシェーダの第 1 のシーケンスのユーザ入力、消費電力、1 つ
以上のゲームレイテンシ、コードサイズ、メモリ使用量およびキャッシュ使用量のうち少
なくとも 1 つまたは全てを含む、請求項 1 ~ 15 のいずれか 1 項に記載の方法。

【請求項 17】

前記 1 つ以上の性能基準は、前記複数のオンラインゲームセッションを実行する過程にお
ける前記複数のコンパイルされたシェーダの第 1 のシーケンスの全消費電力、コードサイ
ズ、平均レイテンシ、全キャッシュ使用量、および全メモリ使用量のうち 1 つ以上に対応
する、請求項 1 ~ 16 のいずれか 1 項に記載の方法。

50

【請求項 18】

前記複数のオンラインゲームセッションは、前記ゲームタイトルについて、前記 1 つ以上の CPU および前記 1 つ以上の GPU によって処理されるオンラインゲームセッションの全てからサンプリングされる、請求項 1 ~ 17 のいずれか 1 項に記載の方法。

【請求項 19】

1 つ以上の CPU と、

複数の GPU と、

前記 1 つ以上の CPU および / または前記複数の GPU による実行のための 1 つ以上のプログラムを格納するメモリとを備え、前記 1 つ以上のプログラムは、請求項 1 ~ 18 のいずれか 1 項に記載の方法を実現するための命令を含む、サーバシステム。

10

【請求項 20】

サーバシステムの 1 つ以上のプロセッサに、請求項 1 ~ 18 のいずれか 1 項に記載の方法を実現するための命令を実行させる、プログラム。

【発明の詳細な説明】**【技術分野】****【0001】**

本出願は一般に、1 つ以上のリアルタイムユーザインタラクティブアプリケーションに対応する複数のオンラインインタラクティブセッションをサポートするようにサーバシステムを管理するための方法およびシステムを含むがこれらに限定されない、コンピュータ技術に関する。

20

【背景技術】**【0002】**

背景

インターネット接続された電子デバイスは、さまざまなクラウドベースのメディアおよびエンターテインメントアプリケーションをサポート可能である。これらのアプリケーションは、サーバがコンテンツをユーザデバイスにストリームするメディアストリーミングアプリケーション、ユーザがサーバで実行されるゲームとユーザデバイスからインタラクトするゲームアプリケーション、ならびに、多数のユーザに互いに同時に、およびにインターネット接続されたデバイスを介してクラウドホスト型コンテンツおよびアプリケーションとインタラクトさせることが可能な、さまざまなソーシャルメディアおよび通信アプリケーションを含む。クラウドベースのアプリケーションのうち、クラウドゲーミングは、ゲームタイトルのハードウェアデマンドが大きく異なること、（たとえば、一人のプレーヤによって、1 つの場所で複数のプレーヤによって、または複数の場所で複数のプレーヤによって）クラウドベースのゲームをプレー可能なトポロジが多岐にわたること、ゲームセッションを実行するゲームサーバへのユーザ入力、およびゲームサーバからプレーヤのデバイス / ディスプレーへのゲームセッション出力を、レイテンシなく確実に送信する必要があること、ゲームプレーの速度および応答性についてプレーヤの期待が大きく異なること、ならびに、状況によってはリアルタイムゲームコンテンツを観客に提供したいという要望があることによって、特有の難題を提起することがある。クラウドベースのゲームの他の難題は、プレーヤの位置（たとえば、サーバに対して近いか遠いか）、プレーヤのゲームサービスに対する接続態様（たとえば、早いインターネット接続を介するか遅いインターネット接続を介するか）、および、プレーヤがゲームをプレーし（たとえば、一般的なパーソナルデバイスまたは専用ゲームコントローラ）ゲームプレー出力を視聴するのに用いるデバイス（たとえば、メディアストリーミングデバイスに接続されたパーソナルデバイスまたはメディアデバイス）の種類にかかわらず、プレーヤにとって矛盾のないゲームプレー体験を提供することに関する。

30

40

【0003】

具体的には、許容可能なレイテンシおよび応答性でゲームを同時に実行可能な、複数のゲームタイトルについて、たとえば、同じまたは異なる場所から同じゲームタイトルをプレーしている複数のプレーヤについて、多種多様の入力デバイスおよび出力デバイスならび

50

にネットワーク接続によって、複数のゲームセッションをサポートするクラウドゲームシステムが必要とされている。さらに、ゲームセッションにおいてプレーヤの入力（たとえば、エンドユースゲームデバイス / コントローラ上で入力されるゲーム入力）を受信すると、ユーザ入力を即座に処理し、同時に許容可能なレイテンシで全てのゲームプレーヤのためにプレーヤ入力アクションの結果を反映した高解像度画像を出力するクラウドゲームシステムが必要とされている。また、状況によっては、観客がそれぞれの表示デバイスでリアルタイムにゲームプレーを追えるように、ゲームプレーアクティビティの高解像度ビデオストリームを提供するゲームシステムが必要とされている。したがって、同じ場所に集まっているユーザによる自発的なゲームプレーから異なる場所からの複数のユーザによるオンラインインタラクティブゲームプレーまでの広範囲のゲーム設定でゲーム体験を拡張するために、効率のよいゲーム処理および出力メカニズムを有するクラウドゲームシステムを提供することが有益である。

10

【発明の概要】**【課題を解決するための手段】****【0004】****概要**

本明細書で説明される実現例は、許容可能なレイテンシおよび応答性で複数のオンラインゲームセッションの同時実行を容易にするように、サーバシステムの計算およびストレージリソースを構成することに向けられている。たとえば、不揮発性メモリに格納されたスタティックメモリプールは、常にリフレッシュを必要とするメインダイナミックメモリに一般的に格納されるスタティックデータ項目を記憶するように提供され得る。このスタティックメモリプールは、サーバシステムの複数のプロセッサによって共有され、メインダイナミックメモリが必要とするのと同じだけの電力を要求することなく、合理的に早い速度でアクセス可能である。スタティックデータ項目を格納するようにメインダイナミックメモリの一部はスタティックメモリプールによって置換えられるため、メインダイナミックメモリのサイズを減らすことが可能である。さらに、シェーダキャッシュは、確実に複数のゲームセッションが望ましい電力とレイテンシ性能でレンダリングされるように、コンパイルされたシェーダの2つ以上のシーケンスを格納する。複数のオンラインゲームセッションはシェーダキャッシュに既に存在するコンパイルされたシェーダのシーケンスのうち1つを用いて処理される一方で、シェーダキャッシュに格納されたコンパイルされたシェーダの異なるシーケンスは、リアルタイム使用統計に基づいて、異なる性能基準を満たすようにダイナミックに生成される。コンパイルされたシェーダのこれらの異なるシーケンスは、複数のオンラインゲームセッションをレンダリングする次のゲームコンテンツのために使用可能である。

20

30

【0005】

本出願のある態様は、サーバシステムの記憶能力を管理する方法であって、サーバシステムは、1つ以上の中央処理装置（CPU）と、複数のグラフィックス処理ユニット（GPU）と、プログラムの実行中に1つ以上のCPUおよび/または複数のGPUによる使用のためのプログラムおよびデータを格納するメインダイナミックメモリと、不揮発性メモリに格納されたスタティックメモリプールと、スタティックメモリプールを管理するように構成されたメモリコントローラとを含む。複数のGPUの各々は、ローカルキャッシュを含み、メモリコントローラを介してスタティックメモリプールにアクセスするように構成される。方法は、1つ以上のCPU上でゲームタイトルについて複数のゲームセッションを並行して実行することを含む。複数のゲームセッションの各々は、スタティックデータ項目と関連付けられ、スタティックデータ項目を用いてそれぞれのGPUによって実行可能なグラフィックス処理を必要とする。スタティックデータ項目はスタティックメモリプールに格納される。方法はさらに、1つ以上のCPU上で実行される複数のゲームセッションのうち第1のゲームセッションについて、グラフィックス処理を第1のGPUに割当てることと、第1のGPUが、スタティックデータ項目の第1のコピーをメモリコントローラにリクエストすることと、スタティックメモリプールから読出されたスタティック

40

50

データ項目の第1のコピーをメモリコントローラから受信することと、第1のGPUが、スタティックデータ項目の第1のコピーを用いてグラフィックス処理を実行することと、複数のゲームセッションのうち第1のゲームセッションと関連して、グラフィックス処理の第1の結果をローカルに（たとえば、ローカルキャッシュに、特に第1のGPUのローカルキャッシュに）格納することとを含む。方法はさらに、1つ以上のCPU上で実行される複数のゲームセッションのうち第2のゲームセッションについて、グラフィックス処理を第2のGPUに割り当てることと、第1のGPUが、スタティックデータ項目の第2のコピーをメモリコントローラにリクエストすることと、スタティックメモリプールから読出されたスタティックデータ項目の第2のコピーをメモリコントローラから受信することと、第2のGPUが、スタティックデータ項目の第2のコピーを用いてグラフィックス処理を実行することと、複数のゲームセッションのうち第2のゲームセッションと関連して、グラフィックス処理の第2の結果をローカルに（たとえば、ローカルキャッシュに、特に第2のGPUのローカルキャッシュに）格納することとを含む。

10

【0006】

他の態様は、ゲームタイトルをサーバシステムにロードする方法であって、サーバシステムは、1つ以上のCPUと、複数のGPUと、プログラムの実行中に1つ以上のCPUおよび/または複数のGPUによる使用のためのプログラムおよびデータを格納するメインダイナミックメモリと、不揮発性メモリプールに格納されたスタティックメモリプールと、スタティックメモリプールを管理するように構成されたメモリコントローラとを含む。GPUの各々は、メモリコントローラを介してスタティックメモリプールにアクセスするように構成される。方法は、1つ以上のCPUが、ゲームタイトルに対応する実行可能なゲームプログラムをメインダイナミックメモリにロードすることを含む。1つ以上のCPUは、複数のオンラインゲームセッションを並行して実行するように構成される。複数のゲームセッションの各々は、実行可能なゲームプログラムに対応し、それぞれのオンラインゲームセッションをレンダリングするように割り当てられたそれぞれのGPUによって実行可能なグラフィックス処理を必要とする。方法はさらに、1つ以上のCPUが、複数のゲームセッションのグラフィックス処理において使用されるスタティックデータ項目を識別することと、メモリコントローラが、スタティックデータ項目をスタティックメモリプールに格納することと、メモリコントローラが、複数のオンラインゲームセッションを実行している複数のGPUがグラフィックス処理を実行しつつスタティックデータ項目にアクセスすることを可能にすることとを含む。

20

30

【0007】

さらに他の態様では、オンラインゲームセッションのために画像フレームをレンダリングする方法がサーバシステムで実現され、サーバシステムは、1つ以上のCPUと、1つ以上のGPUと、プログラムの実行中に1つ以上のCPUおよび/または1つ以上のGPUによる使用のためのプログラムおよびデータを格納するメインダイナミックメモリと、シェーダキャッシュとを含む。方法は、1つ以上のCPU上でゲームタイトルについて複数のオンラインゲームセッションを並行して実行することと、1つ以上のGPUが、コンパイルされたシェーダの第1のシーケンスを用いて、複数のオンラインゲームセッションの各々について複数の画像フレームをレンダリングすることとを含む。方法はさらに、複数のオンラインゲームセッションを実行しつつ、ゲームタイトルと関連付けられた複数のオンラインゲームセッションについて性能基準を識別することと、複数のオンラインゲームセッションの各々の実行を特徴付ける使用統計を収集することと、コンパイルされたシェーダの第1のシーケンスの性能を性能基準に関して改良するためにコンパイルされたシェーダの第1のシーケンス内の第1のシェーダを修正して、コンパイルされたシェーダの修正された第1のシーケンスを生成することとを含む。

40

【0008】

本出願のいくつかの態様によると、サーバシステムは、サーバシステムに上述の方法のいずれかを実行させるための命令を格納するメモリを含む。

【0009】

50

さらに、本出願のいくつかの態様によると、サーバシステムによる実行のために構成された1つ以上のアプリケーションを格納する非一時的なコンピュータ可読記憶媒体は、サーバシステムに上述の方法のいずれかを実行させるための命令を含む。

【0010】

他の実施形態および利点は、本明細書の説明および図面を考慮して、当業者によって明らかであろう。

【0011】

さまざまな説明される実現例をより十分理解するために、以下の図面に関連して以下の実現例の説明を参照する必要がある。図面中、同様の参照番号は、図を通して対応する部分を表す。

10

【0012】

図面を通じて、同様の参照番号は対応する部分を指す。

【図面の簡単な説明】

【0013】

【図1】いくつかの実現例に係る、サーバシステムおよび複数のクライアントデバイスを含むオンラインインタラクティブゲーム環境の例を示す図である。

【図2】いくつかの実現例に係る、複数のゲームサーバがサーバシステムのスタティックメモリプールを共有するオンラインインタラクティブゲーム環境の例を示す図である。

【図3A】いくつかの実現例に係る、データをサーバシステムに格納するように構成されたメモリシステムの例を示す図である。

20

【図3B】いくつかの実現例に係る、複数のレベルのキャッシュを有する中央処理装置(CPU)の例を示す図である。

【図3C】いくつかの実現例に係る、複数のレベルのキャッシュを有するグラフィックス処理ユニット(GPU)の例を示す図である。

【図3D】いくつかの実現例に係る、サーバシステム内の複数のプロセッサによって共有されるスタティックメモリプールに対する読み書きアクセスを制御するメモリコントローラの例を示す図である。

【図4A】いくつかの実現例に係る、サーバシステム上でゲームタイトルをロードするプロセス中のデータフローの例を示す図である。

【図4B】いくつかの実現例に係る、サーバシステム上でオンラインゲームセッションを実行するプロセス中のデータフローの例を示す図である。

30

【図4C】いくつかの実現例に係る、シェーダ最適化プロセス中のデータフローの例を示す図である。

【図4D】いくつかの実現例に係る、複数のオンラインゲームセッションが実行されているときのスタティックデータ項目のデータフローの例を示す図である。

【図5A】いくつかの実現例に係る、サーバシステムの例を示すブロック図である。

【図5B】いくつかの実現例に係る、サーバシステムに格納されたデータを示すブロック図である。

【図6】いくつかの実現例に係る、クライアントデバイス(たとえば、携帯電話、タブレットコンピュータ、ラップトップコンピュータまたはゲームコントローラ)の例を示すブロック図である。

40

【図7】いくつかの実現例に係る、メディアデバイスの例を示すブロック図である。

【図8A】いくつかの実現例に係る、1つ以上のゲームタイトルをホストするサーバシステムの記憶能力を管理するための方法の例を示すフローチャートである。

【図8B】いくつかの実現例に係る、1つ以上のゲームタイトルをホストするサーバシステムの記憶能力を管理するための方法の例を示すフローチャートである。

【図9】いくつかの実現例に係る、ゲームタイトルをサーバシステムにロードするための方法の例を示すフローチャートである。

【図10】いくつかの実現例に係る、サーバシステムでゲームレンダリングと同時にパイプラインされたシェーダを最適化するプロセスの例を示す図である。

50

【図 1 1 A】いくつかの実現例に係る、コンパイルされたシェーダのシーケンスを最適化するプロセスの例を示す図である。

【図 1 1 B】いくつかの実現例に係る、コンパイルされたシェーダの複数のシーケンスを複数の性能基準と関連付けるシェーダハッシュテーブルを示す図である。

【図 1 1 C】いくつかの実現例に係る、コンパイルされたシェーダのシーケンスの複数のバージョンを複数のユーザ入力と関連付けるシェーダハッシュテーブルを示す図である。

【図 1 2】いくつかの実現例に係る、複数のオンラインゲームセッションのために画像フレームをレンダリングする方法（たとえば、画像レンダリングのバックグラウンドでパイプラインされたシェーダの最適化を含む）の例を示すフローチャートである。

【発明を実施するための形態】

10

【0014】

実現例の説明

ここで、実現例が詳細に参照され、それらの例が添付の図面に示される。以下の詳細な説明では、記載されているさまざまな実現例の徹底した理解を提供するために、多数の具体的な詳細が説明されている。しかしながら、記載されたさまざまな実現例がこれらの具体的な詳細がなくても実施されてよいことは、当業者にとって明らかであろう。他の例では、これらの実現例の態様を不必要に曖昧にしないように、よく知られている方法、プロシージャ、コンポーネント、回路、およびネットワークは詳細には記載されていない。

【0015】

本明細書に記載される実現例は、効率的で携帯可能であり、レイテンシが低く、インタラクティブなゲーム体験を分散したゲームプレーヤのコミュニティのために提供することに向けられている。いくつかの実現例では、クラウドゲームハードウェアリソース（たとえば、GPUおよびエンコーダ）がダイナミックに割当てられ、個々のエンドユーザが利用可能なネットワーク帯域幅が監視および利用されて、多数のゲームプレーヤに同時に最適なオンラインゲーム体験が提供される。いくつかの実現例では、オンラインゲーム体験を効率的に提供するために、クラウドゲームストレージリソースが管理される。たとえば、ゲームサーバシステムは、ゲームコンテンツレンダリングで使用するスタティックデータ項目を格納するためのスタティックゲーム格納リソースのレベルを追加して、消費電力が比較的大きいダイナミックメモリの需要を低減可能である。いくつかの実現例では、コンパイルされたシーケンスがダイナミックに最適化されて複数のオンラインゲームセッションのリアルタイム使用統計量に従って1つ以上の性能基準が満たされる一方で、コンパイルされたシェーダはフォアグラウンドで使用されて、これらのオンラインゲームセッションのためにゲームコンテンツをレンダリングする。

20

30

【0016】

具体的には、いくつかの実現例では、サーバシステムは、1つ以上のCPUと、複数のGPUと、プログラムの実行中にCPUおよび/またはGPUによる使用のためのプログラムおよびデータを格納するメインダイナミックメモリとを備える。複数のGPUの各々は、ローカルキャッシュを含む。不揮発性メモリに格納されたスタティックメモリプールは、サーバシステムに追加される。サーバシステムのメモリコントローラは、スタティックメモリプールを管理するように、たとえば、スタティックメモリプールに対する読出しアクセス/書込みアクセスを複数のGPUの各々に提供するように構成される。一例では、メインダイナミックメモリは、常にリフレッシュが必要なダイナミックランダムアクセスメモリ（DRAM）であり、スタティックメモリプールは、電源に接続されている限りリフレッシュを必要としない相変化ランダムアクセスメモリ（PRAM）である。

40

【0017】

サーバシステムは、ゲームタイトルについて、1つ以上のCPU上で並行して複数のゲームセッションを実行する。複数のゲームセッションの各々は、スタティックデータ項目と関連付けられ、スタティックデータ項目を用いてそれぞれのGPUによって実行可能なグラフィックス処理（たとえば、オープングラフィックスライブラリ（OpenGL）レンダリングパイプラインでの頂点処理、テッセレーションおよびプリミティブアセンブリ）

50

を必要とする。スタティックデータ項目は、スタティックメモリプールに格納される。複数のゲームセッションの各々を実行するために、CPUは、グラフィックス処理をそれぞれのGPUに割当て、それぞれのGPUは、グラフィックス処理を実行する目的でメモリコントローラにスタティックデータ項目のコピーを要求する。それぞれのGPUは、スタティックメモリプールから読出されたスタティックデータ項目のコピーをメモリコントローラから受信し、スタティックデータ項目のコピーを用いてグラフィックス処理を行う。グラフィックス処理の結果が、複数のゲームセッションのそれぞれ1つと関連して（たとえば、ローカルキャッシュまたはメインダイナミックメモリに）ローカルに格納される。

【0018】

いくつかの実現例では、CPUがゲームタイトルに対応する実行可能なゲームプログラム（たとえば、サーバ側ゲームモジュール）をメインダイナミックメモリにロードすると、スタティックメモリプールに格納されたスタティックデータ項目が、スタティックメモリプールにロードされる。具体的には、CPUは、スタティックデータ項目がグラフィックス処理で使用されると判断し、メモリコントローラを制御して、スタティックデータ項目をスタティックメモリプールに格納する。スタティックデータ項目は場合によっては、サーバシステムのローカルハードドライブから抽出される、またはリモートライブラリサーバから受信される。スタティックデータ項目がスタティックメモリプールに格納されると、スタティックデータ項目は、メインダイナミックメモリに格納される必要がなくなり、そのため、メインダイナミックメモリは、サーバシステム内のCPUおよびGPUの計算性能を損なうことなくサイズおよび消費電力を減らすことによって、効率を改善することができる。

【0019】

いくつかの実現例では、サーバシステムは、1つ以上のCPUと、1つ以上のGPUと、プログラムの実行中にCPUおよび/またはGPUによる使用のためのプログラムおよびデータを格納するメインダイナミックメモリと、シェーダキャッシュを備える。サーバシステムは、CPU上でゲームタイトルについて複数のオンラインゲームセッションを並行して実行し、1つ以上のGPUは、コンパイルされたシェーダの第1のシーケンスを用いて、複数のオンラインゲームセッションの各々について複数の画像フレームをレンダリングする。サーバシステムは、フォアグラウンドで複数のオンラインゲームセッションを実行しつつ、バックグラウンドでは、ゲームタイトルと関連付けられた複数のオンラインゲームセッションについて（たとえば、電力使用量およびゲームレイテンシに関する）性能基準を識別し、オンラインゲームセッションの各々の実行を特徴付ける使用統計（たとえば、ユーザ入力、電力使用量およびゲームレイテンシ）を収集する。サーバシステムは、コンパイルされたシェーダの第1のシーケンスの性能を性能基準に関して改良するためにコンパイルされたシェーダの第1のシーケンス内の第1のシェーダを修正して、コンパイルされたシェーダの修正された第1のシーケンスを生成する。必要に応じて、コンパイルされたシェーダの修正された第1のシーケンスは、コンパイルされたシェーダの別のシーケンスとして、コンパイルされたシェーダの第1のシーケンスの新しいバージョンとして、またはコンパイルされたシェーダの第1のシーケンスの代わりに格納される。コンパイルされたシェーダの修正された第1のシーケンスは、複数のオンラインゲームセッションについて画像フレームのレンダリングを行い、かつ、これらのオンラインゲームセッションの性能基準と関連付けられた性能を改良するために連続して使用可能である。これらの手段によって、コンパイルされたシェーダは、オンラインゲームセッションのリアルタイム使用統計に基づいて、オンラインゲームセッションの実行を中断することなく、ダイナミックに最適化される。

【0020】

ゲームタイトルの複数のゲームセッションをホストするゲーム環境

図1は、複数の実現例に係るオンラインインタラクティブゲーム環境100の例を示す図である。オンラインインタラクティブゲーム環境100は、1つ以上のクライアントデバイス（たとえば、クライアントデバイス102および104）を含む。クライアントデバ

10

20

30

40

50

イス102の各々は、1つ以上のゲームアプリケーションを実行する。オンラインゲームセッションは、起動され、特定のゲームアプリケーションで実行されて、クライアントデバイス102のユーザに、サーバシステム114によってホストされるオンラインインタラクティブゲーム（「ゲームタイトル」とも呼ばれる）をプレーさせることができる。いくつかの実現例では、クライアントデバイス102（たとえば、ホストクライアント）は、オンラインインタラクティブゲームのゲームシーンに加わるよう1つ以上の他のクライアントデバイス102を招待するように構成される。これらのクライアントデバイス102のゲームセッションは、オンラインインタラクティブゲームの同じゲームシーンを表示するように同期されており、必要に応じて、別個の視点がそれぞれのユーザに対応している。

10

【0021】

逆に、サーバシステム114は、オンラインインタラクティブゲームプラットフォームをホストして、クライアントデバイス102が特定のゲームアプリケーションを含む1つ以上のゲームアプリケーションをプレーするのをサポートする。具体的には、サーバシステム114は、クライアントデバイス102と関連付けられた複数のユーザアカウントを備え、1つ以上のゲームアプリケーションの各々と関連してクライアントデバイス102のユーザを認証する。サーバシステム114は、ゲームタイトルのシーンと関連付けられた対応するゲームセッションに参加するクライアントデバイス102上で、このシーンのレンダリングおよびリフレッシュを行う。いくつかの実現例では、サーバシステム114は、クライアントデバイス102の能力および/またはサーバシステム114とクライアントデバイス102の各々との間の通信接続の質を評価可能であり、適応的に、クライアントデバイス102と関連付けられたゲームセッションについて同時データストリームを生成する。これらの手段によって、サーバシステム114は、同時に実質的に低いレイテンシで、2つ以上のクライアントデバイス102上でオンラインインタラクティブゲームの同時ゲームセッションを容易にするように構成される。

20

【0022】

いくつかの実現例では、クライアントデバイス102は、それぞれのオンラインゲームセッションと関連付けられたメディアコンテンツを表示するための、内部に一体化された表示スクリーンを有する。いくつかの実現例では、クライアントデバイス102は、メディアデバイス106および出力デバイス108に接続される。具体的には、クライアントデバイス102は、メディアデバイス106に直接（たとえば、電線を介して、Bluetooth（登録商標）もしくは他の無線通信リンクを介して）、ローカルネットワーク110（たとえば、Wi-Fiネットワーク）を介して、または1つ以上の通信ネットワーク112を介して、通信可能に接続される。いくつかの実現例では、クライアントデバイス102とメディアデバイス106とは、（たとえば、同じ部屋で、同じ家で、など）互いにローカルである。メディアデバイス106はさらに、ビジュアルおよび/または音声コンテンツを出力可能な1つ以上の出力デバイス108（たとえば、テレビ、表示モニター、サウンドシステム、スピーカなど）に接続される。メディアデバイス106は、出力デバイス108（複数可）にコンテンツを出力するように構成される。いくつかの実現例では、メディアデバイス106は、キャストデバイス（たとえば、Google社のCHROMECAST（登録商標））、または他の態様ではキャスト機能を含むデバイスである。たとえば、ゲームコンテンツがサーバシステム116によってクライアントデバイスにおいてユーザ入力に回答してレンダリングされる場合、クライアントデバイス102は、メディアデバイス106を制御して、ゲームコンテンツを通信ネットワーク112を介してサーバシステム116から受信し、受信したゲームコンテンツを出力デバイス108にキャストする。

30

40

【0023】

いくつかの実現例では、サーバシステム114は、ゲームサーバ108およびメディアストリーミングサーバ120を備える。ゲームサーバ118は、第1のクライアントデバイス102A上で実行されているオンラインインタラクティブゲームセッションについて、

50

2つ以上のメディアストリーム130および132を同時に提供するように構成される。2つ以上のメディアストリームはそれぞれ、第1のクライアントデバイス102Aおよびレビュークライアントデバイス104に1つ以上の通信ネットワーク112を介して提供される低レイテンシストリーム130および標準レイテンシストリームを含む。必要であれば、標準レイテンシストリーム132は、教育目的で提供される。第1のクライアントデバイス102Aのユーザがゲームセッションを第1のクライアントデバイス102Aでプレーする一方で、ゲームセッションは、記録され、標準レイテンシストリーム132を介して一人以上の観客に対してブロードキャストされる、すなわち、観客は、レビュークライアントデバイス104上でゲームセッションのレビューが可能である。低レイテンシストリーム130は、オンラインインタラクティブゲームセッションのゲームプレーに対応し、関連付けられたレビューセッションに対応する標準レイテンシストリーム132と比較して早い反応速度および遅い送信レイテンシを有する。必要であれば、レビュークライアントデバイス104は、クライアントデバイス102と同じ会場に位置する。必要であれば、レビューデバイスは、クライアントデバイス102から離れている。いくつかの実現例では、レビュークライアントデバイス104は、必要であれば一体化されたディスプレイまたは外部ディスプレイを有するクライアントデバイス102Aのような、他のクライアントデバイスである。いくつかの実現例では、レビュークライアントデバイス104は、ビジュアルおよび/または音声コンテンツを出力可能な1つ以上の出力デバイス108に接続されたメディアデバイス106（たとえば、キャスティング機能を有するデバイス）である。いくつかの実現例では、低レイテンシストリームと標準レイテンシストリームとの両方は、同じ出力デバイスのスクリーン上に表示される。

【0024】

各クライアントデバイス102は、中央サーバまたはクラウドコンピューティングシステム（たとえば、サーバシステム114）、および/またはネットワーク接続された他のデバイス（たとえば、他のクライアントデバイス102、レビュークライアントデバイス104、メディアデバイス106および出力デバイス108）とのデータ通信及び情報共有が可能である。データ通信は、さまざまなカスタムまたは標準無線プロトコール（たとえば、IEEE 802.15.4、Wi-Fi、ZigBee、6LoWPAN、Thread、Z-Wave、Bluetooth Smart、ISA100.11a、WirelessHART、MiWiなど）のいずれか、および/またはさまざまなカスタムもしくは標準有線プロトコール（たとえば、Ethernet（登録商標）、HomePlugなど）のいずれか、または、本特許出願日においてまだ開発されていない通信プロトコールを含む他の好適な通信プロトコールを用いて実行され得る。いくつかの実施形態では、オンラインインタラクティブゲーム環境100は、クライアントデバイス102のセットおよびそれらの対応するメディアデバイスおよび出力デバイス（設けられている場合）がそれを介してローカルエリアネットワーク上で互いに通信可能に接続される従来のネットワークデバイス（たとえば、ルータ（図1には図示されていない））を含む。ローカルエリアネットワークは、通信ネットワーク112（たとえば、ワイドエリアネットワークおよびインターネット）の他の部分に通信可能に接続される。いくつかの実施形態では、クライアントデバイス102の各々は必要であれば、1つ以上の無線通信ネットワーク（たとえば、ZigBee、Z-Wave、Insteon、Bluetooth、Wi-Fi、および/または他の無線通信ネットワーク）を用いて、1つ以上の他のクライアントデバイス102もしくは104、それぞれのメディアデバイス106、またはそれぞれの出力デバイス108と通信する。

【0025】

いくつかの実現例では、クライアントデバイス102は互いに離れている、すなわち、同じ場所内にまたは構造物内にさえ位置していない。ゲームタイトルは、各クライアントデバイス102における実行のためにゲームアプリケーション（たとえば、ゲームアプリケーション629、図6）を起動することによって、クライアントデバイス102上で開始可能である。いくつかの実現例では、クライアントデバイス102毎に、ゲームアプリケ

10

20

30

40

50

ーションは、サーバシステム 114 とのオンラインゲームセッション 116 を個別に確立する。2 つ以上のクライアントデバイス 102 (たとえば、102A および 102B) のオンラインゲームセッション 116 は、(たとえば、ゲームタイトルの同じゲームドメインにおいてプレーされるため) 互いに関連しており、そのため、ゲームアプリケーション内のゲームシーンを共有する。関連するオンラインゲームセッション 116 は互いに同期されており、各オンラインゲームセッション 116 は必要であれば、それぞれのクライアントデバイス 102 に対応する唯一のプレーヤの視点を有する同じゲームシーンを示す。各クライアントデバイス 102 のユーザはそのため、それぞれのクライアントデバイス上でゲームをプレーすることができ、オンラインゲームセッション 116 からの出力の影響を他のクライアントデバイス 102 (複数可) に及ぼすことができる。たとえば、いくつかの実現例では、ゲームアプリケーション 528 は、同時に同じゲームをプレーしているユーザの間で矛盾のないゲーム状態を維持するように構成される。いくつかの実現例では、サーバシステム 114 およびゲームアプリケーション 528 は、ユーザの個々のゲームプレー状態およびビューに特有のシーンのレンダリングされたビューをもたらし得るゲームをプレーしている各ユーザのために、個々のセッションを維持するが、これらのセッションの各々は、同じゲームをプレーし、かつ、同じシーンにインタラクトしこれを眺めているユーザの他のセッションと一致している。

10

【0026】

代替的に、他の実現例では、第 1 のクライアントデバイス 102A のゲームアプリケーションがオンラインゲームセッション 116A を確立した後で、1 つ以上の第 2 のクライアントデバイス 102B が招待メッセージによってオンラインゲームセッション 116A に参加するように招待され、たとえば、オンラインゲームセッション 116A に参加するためのリンク (たとえば、URL アドレス) を有するメッセージが、第 2 のクライアントデバイス 102B の各々に送信される。適切なコントローラ構成が、オンラインゲームセッション 116A に参加するように招待された各第 2 のクライアントデバイス 102B に提供される。本出願では、第 2 のクライアント 102B がオンラインゲームセッション 116A に参加すると、サーバシステム 114 は、個別の第 2 のクライアントデバイス 102B ために別個のゲームセッション 116B を生成する。それぞれの第 2 のクライアントデバイス 102 の各別個のゲームセッション 116B は、第 1 のクライアントデバイス 102A のゲームセッション 116A と同期しており、これと同じシーンを共有するが、それぞれの第 2 のクライアントデバイス 102B に対応する唯一のプレーヤの視点を有し得る。各第 2 のクライアントデバイス 102B が適切なコントローラ構成を受信しオンラインゲームセッション 116A に参加すると (より正確には、その関連するオンラインゲームセッション 116B を開始すると)、ユーザは、それぞれの第 2 のクライアントデバイス 102B 上でゲームをプレーし、他のクライアントデバイス 102 (複数可) で実行されているオンラインゲームセッション 116 の出力に影響を及ぼすことができる。

20

30

【0027】

クライアントデバイス 102 は、ゲームアプリケーションを含む 1 つ以上の別個のユーザアプリケーションを含み、これを実行可能なデバイスである。いくつかの実現例では、クライアントデバイス 102 は、スマートフォン、タブレットデバイス、ラップトップコンピュータ、デスクトップコンピュータ、マルチメディアデバイスまたはリモートコントローラである。いくつかの実現例では、クライアントデバイス 102 は、専用ゲームコントローラデバイス (たとえば、ゲームコントローラ 102B) を含む。いくつかの実現例では、クライアントデバイス 102 は、メディアデバイス 106 と連動して動作するように構成された 1 つ以上のユーザアプリケーションを含む。いくつかの実現例では、アプリケーションは、クライアントデバイス 102 をメディアデバイス 106 とペアリングし、メディアデバイス 106 を構成するためのメディアデバイスアプリケーションを含む。また、アプリケーションは、関連付けられたコンテンツをメディアデバイス 106 にキャスト可能な 1 つ以上のアプリケーションを含む。いくつかの実現例では、アプリケーションは、データ / コンテンツを直接メディアデバイス 106 に (たとえば、ローカルネットワー

40

50

クを介して)送信することによって、および/または、メディアデバイス106がそれからデータ/コンテンツをストリーミング、もしくは他の態様では受信可能なリモートロケーション(たとえば、URLまたはサーバシステム116もしくはゲームコンテンツホストにおける記憶場所へのリンク)にメディアデバイス106を向けることによって、データおよび/またはコンテンツをメディアデバイス106にキャストする。メディアデバイス106は、アプリケーションおよび/またはリモートロケーションからデータ/コンテンツを受信し、受信したデータ/コンテンツに対応するビジュアルおよび/または音声コンテンツを、出力デバイス108に出力する。そのため、オンラインゲームセッション116は、クライアントデバイス102上で実行されているゲームアプリケーション、リモートサーバシステム114、およびメディアデバイス106間で確立される。

10

【0028】

いくつかの実現例では、関連するオンラインゲームセッション116をリンクさせるプロセスの一部として、サーバシステム114は、対応するクライアントデバイス102の各々の能力、および/またはサーバシステム114とクライアントデバイス102との間の通信接続の質を評価する。いくつかの実現例では、サーバシステム114は、クライアントデバイス102とサーバシステム114との間のネットワークレイテンシを計測する。計測されたレイテンシが閾値を上回り低レイテンシ接続が利用可能な場合、サーバシステム114は、クライアントデバイス102に低レイテンシ接続に変更するように提案可能である、または、クライアントデバイス102を低レイテンシ接続に変更するようにクライアントデバイス102のユーザを勧誘可能である。たとえば、クライアントデバイス102がセルラー無線接続されていて、ローカルエリアネットワークが利用可能である場合、サーバシステム114は、クライアントデバイス102が利用可能なローカルネットワークを通じて接続するように提案可能である。いくつかの実現例では、レイテンシ閾値要件はゲーム間で異なる。たとえば、低レイテンシ接続で最良に体験されるゲーム(たとえば、アクションゲーム)もあれば、レイテンシに関して要求がそれほど厳しくない他のゲーム(たとえば、オンラインボードゲームまたはカードゲーム)もある。サーバシステム114は、異なる種類のゲームと関連付けられたこれらの異なる要件を考慮して、接続を推奨し得る。

20

【0029】

ゲームセッション116を始める、またはこれに参加するクライアントデバイス102の一部として、サーバシステム114は、クライアントデバイス102と通信して、クライアントデバイス102上にコントローラをセットアップする。いくつかの実現例では、これは、クライアントデバイス102がコントローラに必要とされるリソースおよび通信能力を有しているかどうかをサーバシステム114が評価することを含む。クライアントデバイス102において利用可能なリソース、接続の質、およびゲーム要件によって、コントローラは、クライアントデバイス102において異なる態様で実現され得る。いくつかの実現例では、ゲームタイトルは、ウェブページベースのコントローラインターフェースでプレー可能である。たとえば、ゲームのためのコントローラインターフェースはウェブページ内に埋込まれてもよく、ウェブページは、クライアントデバイス102上のウェブブラウザでレンダリングされる。代替的にいくつかの実現例では、標準化されたコントローラが、ゲームに特有でないまたはゲームと直接関連付けられていない事前に定義されたアプリケーション(たとえば、CHROMECASTまたはGoogle社のGOOGLE CAST、もしくは他のメディアデバイスアプリケーション)において、またはクライアントデバイス102のオペレーティングシステムにおいて実現される。たとえば、クライアントデバイス102上のデバイスオペレーティングシステムまたは所定のアプリケーションは、コントローラサブモジュールを有し得る。コントローラサブモジュールは、1つ以上の標準化されたコントローラ構成、テンプレートなどを含む。標準化されたコントローラ構成の各々は、コントローラサブモジュールを、ある方法でクライアントデバイス102上の入力デバイスおよび/またはセンサを利用するように構成して、バーチャルコントローラを実現する。使用される標準化されたコントローラ構成は、ゲームによって

30

40

50

および／またはクライアントデバイスの種類に応じて異なり得る。

【 0 0 3 0 】

さらにいくつかの実現例では、ゲームは、コントローラサブモジュールで実現され得る特定のコントローラ構成を有する。そのような構成は、オンラインゲームセッション 1 1 6 に参加するまたはこれを始めるクライアントデバイス 1 0 2 のプロセスの一部として、サーバシステム 1 1 4 において格納されてもよい、および、クライアントデバイス 1 0 2 に送信されてもよい。いくつかの実現例では、特定のコントローラ構成は、完全なカスタムコントローラ、または標準化されたコントローラとカスタムコントローラとの組合せでもよい。さらにいくつかの実現例では、ゲームは、当該ゲームと関連付けられた特定のアプリケーションを必要とする。たとえば、ゲームは、当該ゲームと特別に関連付けられたコントローラアプリケーションを必要とし得る。いくつかの実現例では、クライアントデバイス 1 0 2 は、セッション 1 1 6 を始めるまたはこれに参加する一部として、特定のアプリケーションまたは所定のアプリケーションをダウンロードするよう向けられてもよい。たとえば、クライアントデバイス 1 0 2 が（コントローラサブモジュールを有する）事前に定義されたアプリケーションまたはゲームと関連付けられた特定のアプリケーションを既に有しておらず、そのようなアプリケーションがプレーのために必要とされる場合、サーバシステム 1 1 4 は、クライアントデバイス 1 0 2 に、ユーザにダウンロードが必要であると促し、かつ、ユーザに処理の許可を求めるように指示する。

10

【 0 0 3 1 】

サーバシステム 1 1 4 はさらに、データベース 1 6 0 を備える。データベース 1 6 は少なくとも、サーバシステム 1 1 4 上でホストされる 1 つ以上のゲームアプリケーション（たとえば、ゲームアプリケーション 6 2 9、図 6）の各々のユーザアカウントと関連付けられたユーザ情報を格納する。ユーザ情報の例は、ユーザアカウント情報（たとえば、本人確認およびパスワード）、会員の種類、プリファレンス、および活動履歴を含むが、これらに限定されない。いくつかの実現例では、データベース 1 6 0 は、クライアントデバイス 1 0 2 上でプレーされるオンラインゲームセッションと関連付けられたセッションデータを格納する。各オンラインゲームセッション 1 1 6 のためのセッションデータの例は、フレームレート、レンダリング仕様、標準レイテンシ要件、GPU 割当て情報、エンコード割当て情報、関連セッション識別、シェーダパイプライン識別、および最新状態情報を含むが、これらに限定されない。セッションデータは、ユーザ情報よりも頻繁にアクセスされる。いくつかの実現例では、ユーザ情報およびセッションデータを格納するメモリはそれぞれ、フラッシュメモリおよびランダムアクセスメモリ（RAM）で構成される。いくつかの実現例では、ユーザ情報およびセッションデータのサブセットが、それぞれのゲームタイトルに対応する各ゲームサーバ 1 1 8 に格納される。

20

30

【 0 0 3 2 】

サーバシステム 1 1 4 は、1 つ以上のゲームサーバ 1 1 8、メディアストリーミングサーバ 1 2 0、ゲームライブラリサーバ 1 2 2、シェーダライブラリサーバ 1 2 4 およびパイプラインオブティマイザサーバ 1 5 0 のうち 1 つ以上を備える。ゲームサーバ 1 1 8 または代替的なフロントエンドサーバ（図 1 には図示せず）は、クライアントデバイス 1 0 2 および 1 0 4 と関連付けられたユーザアカウント、たとえば、ユーザアカウントによる 1 つ以上のオンラインインタラクティブゲームの会員登録を管理する。クライアントデバイス 1 0 2 がそれぞれのユーザアカウントにログオンし、オンラインゲームセッション 1 1 6 を開始すると／オンラインゲームセッション 1 1 6 に参加すると、ゲームサーバ 1 1 8 は、ゲームセッション 1 1 6 をセットアップするだけでなく、コンテンツサーバ（図示せず）からゲームコンテンツを取得し、クライアントデバイス 1 0 2 または指定されたメディアデバイス 1 0 4 上で実行されるゲームアプリケーションにゲームコンテンツを送信し、ユーザリクエストまたはアクションを識別し、ユーザリクエストまたはアクションに回答してクライアントデバイス 1 0 2 またはメディアデバイス 1 0 4 のためにゲームプレー出力をレンダリングし、および、それぞれのゲームセッション 1 1 6 中にゲーム状態データを格納することによって、それぞれのクライアントデバイス 1 0 2 について各特定のゲ

40

50

ームセッションを管理する。

【 0 0 3 3 】

いくつかの実現例では、ゲームサーバ 1 1 8 は、クラウドゲームハードウェアリソース（たとえば、G P U 1 4 0 およびエンコーダ 1 4 2 ）をダイナミックに割当て、個別のエンドユーザが利用可能なネットワーク帯域幅を監視および利用して、最適なクラウドゲーム体験を提供する。いくつかの実現例では、ゲームサーバ 1 1 8 は複数の性能階層を提供し、これらの性能階層は、高解像度ビデオ／メディアストリームを有する高性能のリアルタイムゲームセッションをサポートする階層と、ほぼレイテンシがない状態で 1 つ以上の実際のゲームストリームに対応するレビューメディアストリームをサポートする他の階層とを含む。必要であれば、そのようなレビューメディアストリームは、Y o u T u b e（登録商標）などのメディアストリーミングサイトを介して一人以上のユーザに提供され、メディアストリーミングサイトは必要であれば、メディアストリーミングサーバ 1 2 0 によって管理される。

10

【 0 0 3 4 】

いくつかの実現例によると、ゲームサーバ 1 1 8 は、第三者のゲームコンテンツを提供するゲームコンテンツホスト（図 1 では図示せず）に接続される。第三者のゲームコンテンツの例は、スポーツゲーム、レースゲーム、ロールプレーゲーム（R P G）および一人称シューティング（F P S）ゲームを含むが、これらに限定されない。これらのゲームの異なる例は、異なる関連付けられたレイテンシ要件および予想、出力ビデオ解像度、ならびにゲームサーバ計算作業負荷およびビデオ符号化／ストリーミングリソース、ならびにネットワーク帯域幅に基づいて、（たとえば、いくつかの例では異なる登録性能階層と一致する最適なユーザゲーム体験を確実なものにするために）大きく異なるクラウドハードウェア要件およびネットワークを有し得る。

20

【 0 0 3 5 】

説明したように、サーバシステム 1 1 4 は、1 つ以上のゲームタイトルをホストし、ただ 1 つのゲームタイトルの複数のオンラインゲームセッションを実行して複数のクライアントデバイス上で同時ゲームプレーをサポートするように構成される。1 つ以上のゲームタイトルの各々は、いずれかのオンラインゲームセッションをサポートするために実行される前に、ゲームサーバ 1 1 8 にロードされる。ゲームライブラリサーバ 1 2 2 は、ゲームサーバ 1 1 8 に接続され、実行可能なゲームプログラム（たとえば、図 5 A のサーバ側ゲームモジュール 5 2 6 ）および実行可能なゲームプログラムの実行に必要なスタティックデータ項目を提供するように構成される。実行可能なゲームプログラムは、ゲームサーバ 1 1 8 にロードされると、C P U によって実行されて、G P U に、1 つ以上の画像レンダリングプログラム（たとえば、シェーダプログラム）を用いて複数のオンラインゲームセッションの各々について複数の画像フレームを生成するように指示する。シェーダライブラリサーバ 1 2 4 は、ゲームサーバ 1 1 8 に接続され、異なるゲームタイトルによって必要とされると標準シェーダプログラムを提供するように構成される。シェーダプログラムがゲームタイトルによる使用のために選択される場合、それは、各オンラインゲームセッションについて複数の画像フレームをレンダリングするための使用のために、シェーダライブラリサーバ 1 2 4 から取出され、ゲームサーバ 1 1 8 によってコンパイルされ、ゲームサーバ 1 1 8 にローカルに格納される。いくつかの実現例では、オンラインゲームセッションの各画像フレームは、（たとえば、頂点シェーダ、テッセレーションシェーダ、幾何学シェーダおよびフラグメントシェーダを含む）コンパイルされたシェーダの順序シーケンスによってレンダリングされ、これらのコンパイルされたシェーダの各々は、それぞれの画像フレームを作成する過程において 1 つ以上の特定の画像レンダリング処理を実行する。

30

40

【 0 0 3 6 】

図 2 は、いくつかの実現例に係る、複数のゲームサーバ 1 1 8 がサーバシステム 1 1 4 内のスタティックメモリプール 2 0 2 を共有するオンラインインタラクティブゲーム環境 2 0 0 の例を示す図である。サーバシステム 1 1 4 は、複数のゲームサーバ 1 1 8、メディ

50

アストリーミングサーバ 120、ゲームライブラリサーバ 122 およびシェーダライブラリサーバ 124 のうち 1 つ以上を備え、1 つ以上のゲームタイトルをホストするように構成される。そうは言っても、サーバシステム 114 は、1 つ以上の通信ネットワーク 112 によって複数のクライアントデバイス 102 に通信可能に接続され、ゲームタイトルに対応する複数のオンラインゲームセッションを並行して実行して、複数のクライアントデバイス 102 の各々を、それぞれのオンラインゲームセッションを介してそのゲームタイトルでゲームプレーに参加させる。ゲームサーバ 118 は、複数のクライアントデバイス 102 からセッションリクエストを受信し、それに応じて、オンラインゲームセッションを開始し、これらに参加し、またはこれらを終了する。各セッションリクエストに回答して、ゲームサーバ 118 のうち 1 つが、対応するオンラインゲームセッションを実行するように割当てられる。各オンラインゲームセッションの実行の過程において、それぞれのオンラインゲームセッションを実行するように割当てられたゲームサーバ 118 は、オンラインゲームセッションをリクエストしたクライアントデバイスのために複数の画像フレーム（すなわち、セッションデータストリーム）をレンダリングする。状況によっては、複数の画像フレームは、ゲームタイトルのシーンを共有するオンラインゲームセッションのサブセットのうちいずれかからのユーザ入力（すなわち、ジャンプ、シューティング、回転するコマンド）の受信に回答してレンダリングされる。必要であれば、ゲームサーバ 118 はまた、レビューメディアストリームを生成し、これを直接またはメディアアストリーミングサーバ 120 を介して間接的に、リアルタイムでゲームプレーをレビューしたい観客に提供する。

10

20

【0037】

各ゲームサーバ 118 は、1 つ以上の処理ユニット（たとえば、CPU 138、GPU 140 およびエンコーダ 142）、メインダイナミックメモリ 144、および 1 つ以上のローカルキャッシュ 146 を含む。メインダイナミックメモリ 144 は、プログラムの実行中に CPU 138 および GPU による使用のためのプログラムおよびデータを格納するように構成される。必要であれば、各ゲームサーバ 118 は、それ自体のメインダイナミックメモリ 144 を有する。必要であれば、ゲームサーバ 118 のサブセットまたは全ては、メインダイナミックメモリ 144 を共有する。ローカルキャッシュ 146 は、データ、たとえば、頂点後処理が終了し GPU 140 でのプリミティブアセンブリを待つメディアコンテンツ、GPU 140 によって生成されさらに符号化されるためにエンコーダ 142 に提供されるメディアコンテンツを一時的に格納するように構成される。

30

【0038】

いくつかの実現例では、サーバシステム 114 はさらに、メインメモリ 144 およびキャッシュ 146 に加えて、スタティックメモリプール 202 を備える。スタティックメモリプール 202 は、ゲームサーバ 118（たとえば、118-1、118-2～118-N）のサブセットまたは全ての間でシェアされ、同じゲームタイトルの複数のオンラインゲームセッションにおいて使用される 1 つ以上のスタティックデータ項目を格納するように構成される。サーバシステム 114 はさらに、スタティックメモリプール 202 を管理するように構成されたメモリコントローラ 204 を備える。いくつかの実現例では、複数のゲームセッションの各々は、特定のスタティックデータ項目と関連付けられ、特定のスタティックデータ項目を用いてそれぞれの GPU 140 によって実行可能な特定のグラフィックス処理を必要とし、特定のスタティックデータ項目は、スタティックメモリプール 202 に格納される。それぞれの GPU 140 は、特定のスタティックデータ項目を抽出するために、メモリコントローラ 204 を介してスタティックメモリプール 202 にアクセスするように構成される。

40

【0039】

なお、いくつかの実現例では、スタティックメモリプール 202 に格納されたスタティックデータ項目は、メインダイナミックメモリ 144 に格納可能である。メインダイナミックメモリ 144 は、スタティックメモリプール 202 と比較して早いレートで GPU によってアクセスされる。しかしながら、メインダイナミックメモリ 144 は、メモリからの

50

電荷漏洩を補償するために数ミリ秒毎にリフレッシュを必要とする、または新しい電子電荷を必要とするため、電力性能においては効率的ではない。スタティックデータ項目がスタティックメモリプール202に移動される場合、ゲームサーバ118内のメインダイナミックメモリ144の一部が解放され、より多くのプログラムおよびダイナミックデータを格納するために使用可能になる、または、消費電力を低減するためにより小さなメインダイナミックメモリ144を使用可能になる。

【0040】

サーバシステム114は、ローカルキャッシュ146、メインダイナミックメモリ144およびスタティックメモリプール202を含むマルチレベルストレージ方式を有する。ローカルキャッシュ146は、少なくとも1つのキャッシュレベル（たとえば、L1、L2）を含む。メインダイナミックメモリ144およびスタティックメモリプール202と比べて、ローカルキャッシュ146は、より小さなサイズを有し、CPUおよびGPUに物理的により近接し、より早いレートでアクセス可能である。ローカルキャッシュ146は、CPU138およびGPU140による直接使用のための、メインダイナミックメモリ144またはスタティックメモリプール202から取出された命令およびデータのコピーを格納する。メインダイナミックメモリ144は、主メモリとも呼ばれる。CPU138およびGPU140によって実行される全てのプログラムは、メインダイナミックメモリ144にロードされ、実行の前にブロックまたは命令単位でキャッシュに移動される。メインダイナミックメモリ144のサイズは、一度に実行可能なプログラムの数およびプログラムが容易に利用可能なデータの量を定めるため、重要である。スタティックメモリプール202は、キャッシュ146およびメインダイナミックメモリ144に付加的なものである。いくつかの実現例では、スタティックメモリプール202は、CPU138およびGPU140の両方によってアクセス可能なレベル4キャッシュとも呼ばれるが、複数のゲームサーバ118の間で共有される。スタティックメモリプール202は、ローカルキャッシュ146およびメインダイナミックメモリ144と比較して大きなレイテンシを有しており、そのため、オンラインゲームセッションであまり使用されないスタティックデータ項目を格納するために使用される。サーバシステム114では、このマルチレベルストレージ方式によって、ゲームタイトルの複数のオンラインゲームセッションが、少なくとも消費電力に関して効率的に実行可能になる。

【0041】

ゲーム環境におけるデータ格納

いくつかの実現例では、ゲームタイトルに対応するゲームコンテンツは、グラフィックスパイプラインプロセス（たとえば、OpenGLレンダリングパイプライン）によるオンラインゲームセッションについて、GPU140によってレンダリングされる。グラフィックスパイプラインプロセスは、グラフィックス処理の順序シーケンスを含み、各グラフィックス処理は、1つ以上のユーザが定義したシェードプログラムに基づいて実現される。各グラフィックス処理において、GPU140は、前のグラフィックス処理からユーザ入力または出力を受信し、それ自体の出力を、それぞれのグラフィックス処理の中間出力またはグラフィックスパイプラインプロセス全体の最終的な出力として生成する。このプロセスの間に、それぞれのグラフィックス処理のシェードプログラムは、前のグラフィックス処理から受信されたユーザ入力および出力に加えて、1つ以上のスタティックデータ項目を用いる。たとえば、OpenGLレンダリングパイプラインのグラフィックス処理で使用されるスタティックデータ項目は、1つ以上のテクスチャオブジェクト、頂点データバッファオブジェクト、および定数データオブジェクトを含む。グラフィックスパイプラインプロセスによって、各オンラインゲームセッションについて画像フレームのシーケンスがレンダリングされ、各画像フレームは、ある解像度を有するピクセルアレイ（ページとも呼ばれる）を含む。いくつかの実現例では、グラフィックスパイプラインプロセスに対応するスタティックデータ項目（たとえば、テクスチャオブジェクト）は、オンラインゲームセッションの画像フレームと同じ解像度を有する1つ以上の画像を含む。

【0042】

スタティックデータ項目は、ゲームタイトルに対応する複数のオンラインゲームセッションのグラフィックスパイプラインプロセスにおいて共有される、すなわち、異なるオンラインゲームセッション間で変わらない。状況によっては、スタティックデータ項目は、個々のGPU140と関連付けられたランダムアクセスメモリ(GPU RAM)の一部に格納される。GPU RAMのこの一部は、「リードオンリー」としてアプリケーションプログラミングインターフェース(API)によって明示的にラベルをつけられる。状況によっては、GPU RAMは、その電力予算のかなりの部分をメモリリフレッシュに費やすDRAM(たとえば、メインダイナミックメモリ144)である。リフレッシュサイクルを必要としないメモリ技術が存在すると仮定すると、スタティックデータ項目を格納するためにGPU RAMを使用することはあまり望ましい解決策ではない。

10

【0043】

いくつかの実現例では、GPUのDRAMのほんの一部(たとえば、メインダイナミックメモリの一部)は、リフレッシュを必要とせずDRAMと比較して使用電力の少ない不揮発性メモリで置換えられる。この不揮発性メモリは、グラフィックスパイプラインプロセスで使用されるリードオンリーアセット(すなわち、スタティックデータ項目のサブセットまたは全て)を格納するために使用される。このような不揮発性メモリの例には、PRAM、フラッシュメモリおよびソリッドステートドライブが含まれる。いくつかの実現例では、PRAMが使用されると、DRAMと比較して桁違いに少ない書込みサイクルを処理可能である。したがっていくつかの実現例では、PRAMは、ウェアレベリングおよびハッシュ割当てを用いる。PRAM内の各ページは、格納されたコンテンツを示す関連付けられたハッシュを有し、PRAMに対する読出しおよび書込みは、(ページテーブル変換を用いて)このハッシュにおいてインデックスが付けられる。ゲームタイトルに対応するゲームアプリケーションがその「リードオンリー」アセット(たとえば、スタティックデータ項目)を解放すると、メモリコントローラ204は、リードオンリーアセットに対応するページを即座に解放しないが、PRAMが一杯の場合、リードオンリーアセットに対応するページの解放のみを行う。リードオンリーアセットに対応するページは解放されると、最長時間未使用(LRU)法に従って廃棄される。そのため、PRAM内のメモリユニットは、近似的に同じ量の作業負荷を実行し(たとえば、もう一度同じゲームタイトルについて同じ量のアクセスを提供し)、永久にメモリユニットにダメージを与えることがある過剰な書込みから保護される。

20

30

【0044】

クラウドベースゲームアプリケーションのいくつかの実現例では(たとえば、サーバシステム114では)、GPUのDRAMのほんの一部(たとえば、メインダイナミックメモリ144の一部)が不揮発性メモリに置換えられた後で、各GPUのDRAMは、RAMの読出し/書込みに必要なものまで低減される。各GPU140の不揮発性メモリは、他のGPUの不揮発性メモリと統合されて集中型のスタティックメモリプール202(たとえば、PRAMストレージデバイス)を形成し、これらのGPU140の間で共有される。いくつかの実現例では、アクセスは早いインラック相互接続によってスタティックメモリプール202に送られ、早いインラック相互接続は、各サーバラックにまたは同様の規模で配置されたバスを含む。また、メモリルックアップテーブルまたはハッシュが、スタティックメモリプール202に格納される。

40

【0045】

GPUアセットのリードオンリー部分は、複数の実行にわたって再使用可能であり、集中型のリードオンリー記憶装置に付随する作業負荷のうち多くが同様である場合、シェア可能である。そうは言っても、いくつかの実現例では、スタティックメモリプール202に格納されるスタティックデータ項目は、抽出される場合、2つ以上のオンラインゲームセッションの間で再使用される。このスタティックメモリプール202へ書込まれると、スタティックデータ項目は、繰り返し同じスタティックデータ項目が格納されることを避けるために、メモリルックアップテーブルまたはハッシュに従ってハッシュされる。いくつかの実現例では、スタティックデータ項目がスタティックメモリプール202に既に存在

50

する場合、スタティックメモリプール 202 が完全に一杯になるのを避けるために、拡張 LRU ポリシーが適用される。いくつかの実現例では、オフセットがスタティックメモリプール 202 内に存在するデータについて検索され、このスタティックデータ項目への今後のメモリアクセスのために保管される。ゲームアプリケーションがスタティックデータ項目を解放すると、スタティックメモリプール 202 のスペースがなくなるまで、それを消去する必要はない。

【0046】

いくつかの実現例では、レイテンシが各 GPU 140 からスタティックメモリプール 202 までの往復について計測され、1 つ以上の GPU レイテンシ隠蔽方法を用いて補償される。1 つ以上の GPU レイテンシ隠蔽方法は、スタティックメモリプール 202 への書込みアクセス、タイルされた / 圧縮された / 空間的にコヒーレントなテクスチャ、およびスタティックメモリプール 202 からのページサイズの転送を組み合わせる書込み統合を含むが、これに限定されない。ある状況下では、スタティックデータ項目を格納するために（ローカルな DRAM ではなく）スタティックメモリプール 202 が用いられるため、レイテンシを隠蔽するためにワープの数が増やされる。

【0047】

いくつかの実現例では、集中型のスタティックメモリプール 202 は、PRAM および書込みサイクルが限定されていることが多い PRAM のようなメモリを含む。スタティックメモリプール 202 はスワップされる、または置換えられる一方で、GPU 140 はいずれもラックに詰め込まれない。

【0048】

いくつかの実現例では、サーバシステム 114 は、2 つ以上のゲームタイトルについてオンラインゲームセッションをサポートする。作業負荷が、スタティックメモリプール 202 にアップロードされた「リードオンリー」アセットの以前の作業負荷に従って、複数のサーバを含むラックに割当てられる。ゲームタイトルはレベルによって分割され、同じレベルの人気のあるゲームタイトルは、より小さなスタティックメモリプールを可能にするために、同じラック上のゲームサーバ 118 によって実行される。

【0049】

いくつかの実現例では、多数の GPU が同じ作業負荷を実行していて（たとえば、ゲームタイトルに対応するオンラインゲームセッション）DRAM リフレッシュおよびネットワーク電力コストを償却するのに十分な再利用を達成可能であれば、スタティックメモリプール 202 は、共有 DRAM プール（PRAM プールまたはその等価物に代えて）に置換え可能である。

【0050】

図 3A は、いくつかの実現例に係る、データをサーバシステム 114 に格納するように構成されたマルチレベルストレージ方式を有するメモリシステム 300 の例を示す図である。マルチレベルストレージ方式に従って、サーバシステム 114 のメモリシステム 300 は、各個別のサーバ（たとえば、ゲームサーバ 118）内に直接配設された複数のキャッシュ 302 を含む。より具体的には、複数のキャッシュ 302 の各々は、個別のサーバのプロセッサ、たとえば、ゲームサーバ 118 の CPU 138 および GPU 140 に直接関連付けられ、個別のサーバのプロセッサによって実行される命令と命令の実行中に使用されるデータとの両方を提供するように構成される。サーバシステム 114 のメモリシステム 300 はさらに、サーバ（たとえば、ゲームサーバ 118）の 1 つ以上のプロセッサによって実行されるプログラムおよびプログラムの実行中に使用されるデータを格納するように構成されたメインダイナミックメモリ 144 を含む。各プログラムは複数の命令を含み、実行中、これらの命令は、対応するプロセッサのキャッシュ 302 に順番にロードされ、それぞれのプロセッサによる実行を待つ。いくつかの実現例では、サーバシステム 114 のサーバのサブセットまたは全ては、メインダイナミックメモリ 144 を共有する。データバスは、サーバシステム 114 のサーバのサブセットまたは全てをメインダイナミックメモリ 144 に接続して、メインダイナミックメモリ 144 に対する書込みおよび読

10

20

30

40

50

出しアクセスを実行可能にする。必要であれば、データバスは、それを介してメインダイナミックメモリ 144 が閾値バスレイテンシよりもかなり小さなレイテンシでアクセスされる高速データバスを含む。いくつかの実現例では、サーバシステム 114 の各サーバは、それぞれのサーバの 1 つ以上のプロセッサのみがアクセス可能なそれぞれのメインダイナミックメモリ 144 を有する。たとえば、ゲームサーバ 118 は、ゲームサーバ 118 の CPU 138 と GPU 140 との両方がアクセス可能なメインダイナミックメモリ 144 を含む。ゲームサーバ 118 のメインダイナミックメモリ 144 は、1 つ以上のゲームタイトルを実行するように構成された命令 304、たとえば、図 5 A のサーバ側ゲームモジュール 526 の命令と、各ゲームタイトルのオンラインゲームセッションと関連付けられたダイナミックデータ 306、たとえば、第 1 のゲームセッションデータ 308 および第 2 のゲームセッションデータ 310 を格納する。具体的には、第 1 のゲームセッションデータ 308 は必要であれば、それぞれのゲームタイトルの対応するオンラインゲームセッションについて、GPU 140 によって生成される 1 つ以上の画像フレームのテクスチャページ 308 A、頂点ページ 308 B、定数ページ 308 C およびフレームデータ 308 D を含む。サーバシステム 114 内のメインダイナミックメモリ 144 に格納されるプログラムおよびデータについては、図 5 A および図 5 B を参照して以下でより詳細に説明される。

【0051】

マルチレベルストレージ方式に従って、サーバシステム 114 のメモリシステム 300 はさらに、サーバシステム 114 内の複数のゲームサーバ 118 のために集中化されこれらによって共有されるスタティックメモリプール 202 を含む。スタティックメモリプール 202 は、ゲームタイトルのオンラインゲームセッションの画像フレームをレンダリングするために使用される 1 つ以上のスタティックデータ項目を格納するように構成される。1 つ以上のスタティックデータ項目の例は、テクスチャオブジェクト 312、頂点データバッファオブジェクト 314、定数データオブジェクト 316 を含むが、これらに限定されない。複数のゲームサーバ 118 が複数のオンラインゲームセッションを実行するために割当てられると、各ゲームサーバ 118 は、スタティックメモリプール 202 から 1 つ以上のスタティックデータ項目を取得し、対応するメインダイナミックメモリ 144 から 1 つ以上のダイナミックデータ項目 306 を取得する。

【0052】

さらにいくつかの実現例では、ゲームサーバ 118 はさらに、GPU 140 に接続されたシェーダキャッシュ 318 を含む。シェーダキャッシュ 318 は、コンパイルされたシェーダの命令が GPU 140 に直接ロードされてグラフィックスパイプラインプロセスを介してゲームコンテンツのレンダリングが可能になるように、コンパイルされたシェーダの少なくとも 1 つのシーケンスを格納するように構成される。いくつかの実現例では、サーバシステム 114 は、各々が性能基準に従って構成された複数のコンパイルされたシェーダシーケンス 320 を含み、各々は必要であれば、異なるユーザ入力状況に対応する複数のバージョンを含む。必要であれば、複数のコンパイルされたシェーダシーケンス 320 は、シェーダキャッシュ 318 に格納される。必要であれば、複数のコンパイルされたシェーダシーケンス 320 は、スタティックメモリプール 202 に格納され、シェーダキャッシュ 318 に格納される、または、ゲームコンテンツレンダリングでの使用のために選択される場合は GPU 140 のキャッシュに直接移動される。

【0053】

さらに、マルチレベルストレージ方式に従って、サーバシステム 114 のメモリシステム 300 は、1 つ以上のゲームタイトルに対応する 1 つ以上の実行可能なゲームプログラムを格納するハードドライブ 324 を備える。ゲームタイトルをホストするようにサーバシステム 114 を構成する過程において、サーバシステムの CPU 138 は、ゲームタイトルに対応する実行可能なゲームプログラム 328 のプログラムリクエスト 326 を送信する。プログラムリクエスト 326 に応答して、ハードドライブ 324 は、ゲームタイトルに対応する実行可能なゲームプログラム 328 を CPU 138 に戻す。いくつかの実現例

10

20

30

40

50

では、ハードドライブ 3 2 4 は、リクエストされた実行可能なゲームプログラム 3 2 8 を格納せず、CPU 1 3 8 は、より離れたサーバ（たとえば、ゲームライブラリサーバ 1 2 2 または他の第三者のゲームプロバイダ）に実行可能なゲームプログラム 3 2 8 をリクエストするプログラムリクエスト 3 2 6 を、送信する。CPU 1 3 8 は、ハードドライブ 3 2 4 またはリモートサーバから戻された実行可能なゲームプログラム 3 2 8 を、サーバ側ゲームモジュール 5 2 6 としてインダイナミックメモリ 1 4 4 に格納する。

【 0 0 5 4 】

さらにいくつかの実現例では、ハードドライブ 3 2 4 は、1 つ以上のゲームタイトルのスタティックデータ項目も格納する。ゲームタイトルをロードしている過程において、サーバシステムの CPU 1 3 8 は、ゲームタイトルのスタティックデータ項目のデータリクエスト 3 3 0 を送信する。データリクエスト 3 2 6 に応答して、ハードドライブ 3 2 4 は、ゲームタイトルに対応するスタティックデータ項目 3 3 2 のサブセットまたは全てを CPU 1 3 8 に戻す。いくつかの実現例では、ハードドライブ 3 2 4 は特定のスタティックデータ項目を格納せず、CPU 1 3 8 は、より離れたサーバ（たとえば、ゲームライブラリサーバ 1 2 2 またはシェーダライブラリサーバ 1 2 4 ）に特定のスタティックデータ項目をリクエストするデータリクエスト 3 3 0 を送信する。ハードドライブ 3 2 4 またはリモートサーバから戻された 1 つ以上のスタティックデータ項目 3 3 2 A は、スタティックメモリプール 2 0 2 に格納される。代替的にいくつかの実現例では、1 つ以上のスタティックデータ項目 3 3 2 B が、ゲームサーバ 1 1 8 の CPU および GPU による直接アクセスのために、メインダイナミックメモリ 1 4 4 に格納される。

【 0 0 5 5 】

さらに CPU 1 3 8 は、シェーダライブラリサーバにシェーダリクエストを送信して、ゲームタイトルに関連して用いられる 1 つ以上の標準シェーダプログラム 3 3 6 を取得し、戻されたシェーダプログラム 3 3 6 は、シェーダキャッシュ 3 1 8 にローカルに格納される。CPU 1 3 8 および GPU 1 4 0 は、ゲームタイトルに対応するオンラインゲームセッションを実行しつつ、複数のダイナミックデータ項目 3 3 8 を生成する。これらの複数のダイナミックデータ項目 3 3 8 は、サーバシステム 1 1 4 のメインダイナミックメモリ 1 4 4 に格納される。

【 0 0 5 6 】

いくつかの実現例では、サーバシステム 1 1 4 は、それぞれメインダイナミックメモリ 1 4 4 およびスタティックメモリプール 2 0 2 に対する読み書きアクセスを管理するように構成されたダイナミックメモリコントローラ 2 0 4 A およびスタティックメモリコントローラ 2 0 4 B を含む。いくつかの実現例では、サーバシステム 1 1 4 は、メインダイナミックメモリ 1 4 4 およびスタティックメモリプール 2 0 2 に対する読み書きアクセスを管理するように構成されたただ 1 つのメモリコントローラ 2 0 4 を含む。メインダイナミックメモリ 1 4 4 は、メインダイナミックメモリ 1 4 4 内の命令 3 0 4、ダイナミックデータ項目 3 0 6、および（設けられている場合）スタティックデータ項目 3 3 2 B の位置を識別するディレクトリ 3 0 4 を含む。いくつかの実現例では、スタティックメモリプール 2 0 2 は、スタティックメモリプール 2 0 2 内のスタティックデータ項目 3 3 2 A の位置を識別するハッシュテーブル 3 4 2 を含む。必要であれば、スタティックデータ項目 3 3 2 は、コンテキスト情報に基づいて識別される。一例では、スタティックメモリプール 2 0 2 は、複数のコンパイルされたシェーダシーケンス 3 2 0 を格納し、スタティックメモリプール 2 0 2 内のそれらのアドレスは、ハッシュテーブル内の対応するグラフィックスパイプラインプロセスの性能基準、ユーザ入力および / または使用統計量を用いて識別される。必要であれば、ハッシュテーブル 3 4 2 内のマッピングに基づくそのようなコンテキストも、他のスタティックデータ項目に適用される。ハッシュテーブルは、異なる状況に関連して繰り返しスタティックデータ項目を格納することからスタティックメモリプール 2 0 2 を保護し、これによって、メモリリソースの無駄が生じる。

【 0 0 5 7 】

図 3 B は、いくつかの実現例に係る、複数のレベルのキャッシュを有する CPU 1 3 8 の

10

20

30

40

50

例を示す図である。いくつかの実現例では、各CPUは、複数のCPUコア350を含む。各CPUコア350は、それぞれのCPUコアによって実行される命令と、命令の実行中に使用されるデータとの両方を格納するキャッシュ352を有する。キャッシュ352は、それぞれのCPUコア350と直接接続され、命令及びデータをそれぞれのCPUコア350に即座に直接提供可能である。いくつかの実現例では、CPUコア350はさらに、CPU138の複数のCPUコア350によって共有されるCPUキャッシュ354を含む。CPUキャッシュ354は、CPUコア350に格納された命令の実行中に使用されるデータを格納する。CPUキャッシュ354は、CPUコア350と直接接続されたキャッシュ352にデータを供給する。

【0058】

図3Cは、いくつかの実現例に係る、複数のレベルのキャッシュを有するGPU140の例を示す図である。いくつかの実現例では、各GPUは、複数のGPUコア360を含む。各GPUコア360は、それぞれのGPUによって実行される命令と、命令の実行中に使用されるデータとの両方を格納するキャッシュ362を有する。キャッシュ362は、それぞれのGPUコア360と直接接続され、命令およびデータをそれぞれのGPUコア362に即座に直接提供可能である。いくつかの実現例では、GPUコア360はさらに、GPU138の複数のGPUコア360によって共有されるGPUキャッシュ364を含む。GPUキャッシュ364は、CPUコア360に格納された命令の実行中に使用されるデータを格納する。GPUキャッシュ364は、GPUコア360と直接接続されているキャッシュ362にデータを供給する。ゲームタイトルのゲームコンテンツをレンダリングするために、メモリシステム300内の読み書きがページによって（すなわち、ゲームコンテンツの画像フレームに対応する画像ピクセルのアレイによって）管理される。ページの大きさの読出しは、ラストレベルキャッシュとして使用されるDRAMの小さなプールを含む場合があるローカルGPUメモリ階層（L3-L1キャッシュ）においてキャッシュされる、すなわち、キャッシュ362およびGPUキャッシュ364においてキャッシュされる。

【0059】

図3Dは、いくつかの実現例に係る、サーバシステム114において複数のプロセッサによって共有されるスタティックメモリプール202に対する読み書きアクセスを制御するメモリコントローラ204の例を示す図である。メモリコントローラ204は、アクセス制御モジュール372、ページ解放モジュール374、ウェアレベリングモジュール376およびLRUモジュール378のうち1つ以上を含む。アクセス制御モジュール372は、そこに格納されるハッシュテーブル342に基づいてスタティックメモリプール202にアクセスするように構成される。ページ解放モジュール374は、スタティックメモリプール202からスタティックデータ項目を削除するように構成される。必要であれば、特定のデータ項目を削除するリクエストにตอบสนองして、ページ解放モジュール374は、スタティックメモリプール202が一杯かどうかを判断する。スタティックメモリプール202が一杯ではないという判断に従って、ページ解放モジュール374は、ハッシュテーブル342を修正して、削除されるべき特定のデータ項目に対するアクセスを無効にする。スタティックメモリプール202が一杯であるという判断に従って、ページ解放モジュール374は、スタティックメモリプール202から特定のデータ項目を削除して、対応するメモリ空間を解放する。

【0060】

ウェアレベリングモジュール376は、スタティックメモリプール202におけるメモリユニットの再書込みを監視し、スタティックメモリプール202の耐用年数を延ばす目的でスタティックメモリプール202において均等に再書込みを分散するように構成される。LRUモジュール378は、最も長い間使われていないスタティックデータ項目をスタティックメモリプール202から削除するように構成される。具体的には、いくつかの実現例では、LRUモジュール378は、スタティックメモリプール202に格納された各スタティックデータ項目と共に世代ビットを格納する。メモリ制限に達すると、最も長い

10

20

30

40

50

間アクセスされていないスタティックデータ項目が、世代ビットに従って識別される最も古いデータ項目から削除される。LRUモジュールがスタティックデータ項目を削除すると、世代ビットは、スタティックメモリプール202に残っているスタティックデータ項目について変わる。

【0061】

図4Aは、いくつかの実現例に係る、サーバシステム114上でゲームタイトルをロードするプロセス400の間のデータフローの例を示す図である。サーバシステム114は、互いに離れたゲームサーバ118、ゲームライブラリサーバ122、およびシェーダライブラリサーバ124を含む。サーバシステム114はまた、ゲームサーバ118によってホストされるゲームタイトルのゲームプレーに關与する複数のクライアントデバイス102から離れている。サーバシステム114（具体的には、ゲームサーバ118）は、1つ以上のCPU138と複数のGPU140とを含み、マルチレベルストレージ方式を採用する。マルチレベルストレージ方式に従って、サーバシステム114は、プログラムの実行中にCPUおよび/またはGPUによる使用のためのプログラムおよびデータを格納するメインダイナミックメモリ144と、不揮発性メモリに格納されるスタティックメモリプール202と、スタティックメモリプール202を管理するように構成されたメモリコントローラ204とを備える。ゲームタイトルをロードするユーザリクエストの受信（401）に回答して、CPU138は、ゲームサーバ118のハードドライブ324またはゲームライブラリサーバ122にプログラムリクエストを送信して（402）、ゲームタイトルに対応する実行可能なプログラム（たとえば、サーバ側ゲームモジュール526）をリクエストする。いくつかの実現例では、CPU138はまず、プログラムリクエストをハードドライブ324に送信する。ハードドライブ324が実行可能なゲームプログラムのコピーを提供できない場合、CPU138は、プログラムリクエストをゲームライブラリサーバ122に送信する。プログラムリクエストに回答して、ハードドライブ324またはゲームライブラリサーバ122は、メインダイナミックメモリ144に格納されている（404）実行可能なゲームプログラムのコピーを戻す。

【0062】

1つ以上のスタティックデータ項目（たとえば、テクスチャオブジェクト）は、実行可能なゲームプログラムの実行の過程において必要とされ、そのため、CPU138は、データリクエストを、ハードドライブ324、ゲームライブラリサーバ122、およびシェーダライブラリサーバ124のうち少なくとも1つに送信して（406）、スタティックデータ項目の所定のセットをリクエストする。ハードドライブ324がそれぞれのデータ項目を有していない場合、CPU138は、順序にしたがって、たとえば、まずハードドライブ324に、次にゲームライブラリサーバ122および/またはシェーダライブラリサーバ124に、データリクエストを送信する。この順序は必要に応じて、スタティックデータ項目のセットについて同じである、または、各スタティックデータ項目に従ってカスタマイズされる。リクエストされたスタティックデータ項目の第1のサブセットの各々について、ハードドライブ324、ゲームライブラリサーバ122およびシェーダライブラリサーバ124のうちの1つは、スタティックメモリプール202に格納されている（408）それぞれのデータ項目のコピーを戻す。いくつかの実現例では、リクエストされたスタティックデータ項目の第2のサブセットが戻され、メインダイナミックメモリ144に格納される（410）。いくつかの実現例では、リクエストされたスタティックデータ項目は、リクエストされたデータ項目の第3のサブセット（たとえば、1つ以上のシェーダプログラム）を含む。リクエストされたスタティックデータ項目の第3のサブセットは、ハードドライブ324とシェーダライブラリサーバ124とのうち1つによって戻され、シェーダキャッシュ318に格納される（412）。

【0063】

図4Bは、いくつかの実現例に係る、サーバシステム114上で（具体的には、ゲームサーバ118内で）オンラインゲームセッションを実行するプロセスの間のデータフロー420の例を示す図である。CPUは、複数の分散されたクライアントデバイス102から

10

20

30

40

50

、ゲームタイトルで複数のオンラインゲームセッションを開始する複数のセッションリクエストを受信する(422)。各クライアントデバイス102は、ゲームタイトルに対応するリアルタイムユーザインタラクティブゲームアプリケーションを実行し、それぞれのセッションリクエストがそれぞれのクライアントデバイス102のゲームアプリケーションから送信される。それぞれのセッションリクエストは、リクエストしているクライアントデバイス102のユーザ情報を含む。それぞれのクライアントデバイス102のユーザ情報を受信すると、CPU138は、ユーザ情報に基づいて対応するオンラインゲームセッションを認証し(424)、GPU140のうち1つを割当てて(426)、対応するオンラインゲームセッションを実行する。いくつかの実現例では、複数のオンラインゲームセッションのうち2つ以上が、GPU140のうち1つに割当てられる。つまり、2つ以上のオンラインゲームセッションは、時分割多重化(TDM)、空間多重化、またはそれらの組合せに基づいて、GPUのうち1つの計算リソースおよびストレージリソースを共有する必要がある。ゲームコンテンツレンダリングにおけるGPU割当てについては、2018年3月22日に出願され、「Methods and Systems for Rendering and Encoding Content for Online Interactive Gaming Sessions(オンラインインタラクティブゲームセッションのためにコンテンツをレンダリングおよび符号化するための方法およびシステム)」と題された米国仮特許出願第62/646,824号において説明され、その全体が本明細書において引用により援用される。

10

【0064】

各セッションリクエストに応答して、CPU138は、ゲームタイトルの実行可能なゲームプログラムの命令をメインダイナミックメモリ144からCPU138のローカルキャッシュにコピーし(428)、命令を1つずつ実行する。同様に、CPU138によって命令されると、それぞれのオンラインゲームセッションを実行するように割当てられたそれぞれのGPUもまた、実行可能なゲームプログラム内の関連するゲームレンダリング命令をそのローカルキャッシュにロードし、ゲームレンダリング命令を1つずつ実行する。

20

【0065】

いくつかの実現例では、複数のオンラインゲームセッションの各々は、たとえば每秒60フレーム(fps)のレートで常にリフレッシュされる。いくつかの実現例では、複数のオンラインゲームセッションの各々は、それぞれのセッション自体または関連するセッションによって入力されたユーザアクションに응答してリフレッシュされる。たとえば、オンラインゲームセッションのサブセットは、ゲームタイトルの同じゲームシーンにあるときに関連付けられ、オンラインゲームセッションのサブセットの各々に配信されたゲームコンテンツは、少なくともオンラインゲームセッションのサブセットのうちいずれか1つがユーザアクションを入力すると、アップデートされる。

30

【0066】

具体的には、オンラインゲームセッションについて、CPU138は、オンラインゲームセッションまたはオンラインゲームセッションと同じゲームシーンを共有している他の別のゲームセッションに参与している第1のクライアントデバイスのゲームアプリケーションを介して入力されたユーザ入力を受信する(430)。これに응答して、CPU138は、ゲームタイトルに対応する実行可能なゲームプログラムの命令を実行して(432)、オンラインゲームセッションを実行するように割当てられたGPU140に、ユーザ入力に従って画像フレームのシーケンスをレンダリングするように命令する(438)。このオンラインゲームセッションの実行中、このオンラインゲームセッションと関連付けられたダイナミックデータ項目は、メインダイナミックメモリ144に格納される、またはこれから抽出される(434)。このオンラインゲームセッションと関連付けられたスタティックデータ項目が、スタティックメモリプール202から抽出される(436)。いくつかの実現例では(本明細書では図示せず)、スタティックデータ項目のセットがメインダイナミックメモリ144から抽出される。

40

【0067】

その後、割当てられたGPU140は、ユーザ入力に基づいてオンラインゲームセッショ

50

ンの画像フレームをレンダリングするゲームレンダリング命令を実行する(439)。オンラインゲームセッションの実行中、GPU140は、シェーダキャッシュ318からコンパイルされたシェーダを取得し(440)、スタティックデータ項目をスタティックメモリプール202から抽出し(442)、ダイナミックデータ項目をメインダイナミックメモリ144に格納する、またはこれから抽出する(444)。いくつかの実現例では(本明細書では図示せず)、GPUはまた、メインダイナミックメモリ144から1つ以上のスタティックデータ項目を抽出する。GPUによってレンダリングされた画像フレームは、エンコーダ142によって符号化され、オンラインゲームセッションを実行しているクライアントデバイス102にストリーミングされる(446)。いくつかの実現例では、GPU(たとえば、観客レビュー向けの標準レイテンシストリーム)によってレンダリングされた画像フレームは、メディアストリーミングサーバ120を介してクライアントデバイス102にストリーミングされる(448)。

10

【0068】

図4Cは、いくつかの実現例に係る、シェーダ最適化プロセス450におけるデータフローの例を示す図である。いくつかの実現例では、実行可能なゲームプログラム(たとえば、図5Aのサーバ側ゲームモジュール526)は、ゲームアプリケーション、およびオンラインゲームセッションとシェーダパイプライン最適化プロセスとを同時に実行するように構成されたシェーダパイプラインオプティマイザを含む。たとえば、ゲームアプリケーションがフォアグラウンドでグラフィックスパイプラインプロセスにおけるオンラインゲームセッションを実行する一方で(動作438~448)、シェーダパイプラインオプティマイザは、バックグラウンドでこのグラフィックスパイプラインプロセスにおいて用いられるコンパイルされたシェーダを最適化する。つまり、グラフィックスパイプラインプロセスが実行される一方で(動作438~448)、ゲームサーバ118は、1つ以上の性能基準を満たすために、バックグラウンドでコンパイルされたシェーダの1つ以上のシーケンスを生成する(450)。

20

【0069】

具体的には、いくつかの実現例では、CPU138は、必要であればゲームタイトルに関連する全てのアクティブなオンラインゲームセッションからサンプリングされたオンラインゲームセッションのセットの使用統計を収集する(452)。使用統計は、CPU138およびGPU140のローカルキャッシュまたはメインダイナミックメモリ144から取得され得る。CPU138は、最適化の間に満たされる必要がある1つ以上の性能基準を識別し、取得する(454)。いくつかの実現例では、1つ以上の性能基準は、スタティックメモリプール202またはメインダイナミックメモリ144(任意)に格納された1つ以上のスタティックデータ項目を含む。CPU138はその後、使用統計および性能基準に基づいてアクティブなオンラインゲームセッションを実行するために使用されているコンパイルされたシェーダのシーケンスを最適化するリクエストを、GPUに送信する(456)。1つ以上の性能基準の各々について、GPU140は、コンパイルされたシェーダのシーケンスの性能をそれぞれの性能基準に関して改良するためにコンパイルされたシェーダのシーケンス内の1つ以上のシェーダを修正して(458)、コンパイルされたシェーダの修正されたシーケンスを生成する。状況によっては、コンパイルされたシェーダの修正されたシーケンスは、必要であれば、次のオンラインゲームセッションによって使用されるコンパイルされたシェーダのシーケンスの代わりに格納される。状況によっては、コンパイルされたシェーダの修正されたシーケンスは、使用統計に対応するコンパイルされたシェーダのシーケンスの新しいバージョンとして格納され、オンラインゲームセッションの次のゲーム状態が使用統計に一致するときに使用される。いくつかの実現例では、コンパイルされたシェーダの修正されたシーケンスの各々は、対応する性能基準と関連してシェーダキャッシュ318に格納される(460)。代替的に、図4Cに図示されていないいくつかの実現例では、関連付けられた、コンパイルされたシェーダの修正されたシーケンスおよび性能基準が、スタティックメモリプール202に格納される。

30

40

【0070】

50

図 4 D は、いくつかの実現例に係る、複数のオンラインゲームセッションが実行されているときにスタティックデータ項目を取得するデータフロー 480 の例を示す図である。説明したように、複数のオンラインゲームセッションがゲームサーバ 118 によって並行して実行されるとき (482)、1 つ以上の CPU は、各オンラインゲームセッションをそれぞれの GPU に割当てて。各オンラインゲームセッションの実行中、それぞれの GPU 140 は、コンパイルされたシェーダをシェーダキャッシュ 318 から取得し (440)、スタティックデータ項目をスタティックメモリプール 202 から抽出し (442)、ダイナミックデータ項目をメインダイナミックメモリに格納する、またはこれから抽出する (444)。図 4 D を参照すると、いくつかの実現例では、2 つのクライアントデバイス 102 A および 102 B はそれぞれ、複数のオンラインゲームセッションのうち第 1 のオンラインゲームセッションおよび第 2 のオンラインゲームセッションを実行する。クライアントデバイス 102 A と 102 B との両方は、たとえば、クライアントデバイス 102 A によるゲームタイトルに対応するゲームアプリケーション内のユーザ入力を受信 (484) に応答して、ゲームサーバ 118 によってレンダリングされた画像フレームを受信する。それぞれの画像フレームのレンダリングの過程において、(第 1 および第 2 のゲームセッションを含む) 複数のゲームセッションの各々が、スタティックデータ項目と関連付けられ、スタティックデータ項目を用いてそれぞれの GPU によって実行可能なグラフィックス処理を必要とし、スタティックデータ項目はスタティックメモリプール 202 に格納される。

10

【0071】

20

複数のゲームセッションのうち第 1 のゲームセッションについて、CPU は、グラフィックス処理を第 1 の GPU に割当てて。第 1 の GPU は、第 1 の GPU によって、メモリコントローラ 204 にスタティックデータ項目の第 1 のコピーをリクエストし (486)、スタティックメモリプール 202 から読出されたスタティックデータ項目の第 1 のコピーを、メモリコントローラ 204 から受信する (488)。第 1 の GPU はその後、スタティックデータ項目の第 1 のコピーを用いてグラフィックス処理を行い、複数のゲームセッションのうち第 1 のゲームセッションと関連して (たとえば、識別子を用いることによって) グラフィックス処理の第 1 の結果をローカルに格納する。同様に、複数のゲームセッションのうち第 2 のゲームセッションについて、CPU は、グラフィックス処理を第 2 の GPU に割当てて。第 2 の GPU は、メモリコントローラ 204 に、スタティックデータ項目の第 2 のコピーをリクエストし (490)、スタティックメモリプール 202 から読出されたスタティックデータ項目の第 2 のコピーを、メモリコントローラ 204 から受信する (492)。第 2 の GPU は、スタティックデータ項目の第 2 のコピーを用いてグラフィックス処理を行い、複数のゲームセッションのうち第 2 のゲームセッションと関連して、グラフィックス処理の第 2 の結果をローカルに格納する。

30

【0072】

いくつかの実現例では、複数のオンラインゲームセッションのうち第 3 のオンラインゲームセッションについて、CPU は、グラフィックス処理を第 3 の GPU に割当てて。第 3 の GPU は、第 1 の GPU が所定の期間内にスタティックデータ項目の第 1 のコピーをリクエストしたと判断する。第 3 の GPU は、第 1 の GPU からスタティックデータ項目の第 1 のコピーを直接取得する。いくつかの実現例では、第 1 の GPU 自体がメモリコントローラからまだ第 1 のコピーを受信しておらず、第 3 の GPU は待機して、第 1 の GPU が第 1 のコピーを受信した後で第 1 の GPU から第 1 のコピーを取得する。つまり、第 1 および第 3 の GPU は、スタティックデータ項目のコピーのリクエストを統合する。

40

【0073】

いくつかの実現例では、複数のゲームセッションの各々は優先順位と関連付けられ、第 1 および第 2 の GPU は、第 1 および第 2 のゲームセッションの優先順位に従って、スタティックメモリプール 202 から読出されたスタティックデータ項目のコピーをメモリコントローラから受信する。たとえば、第 1 のゲームセッションは、第 2 のゲームセッションより高い優先順位を有する。第 2 の GPU が第 2 のコピーをリクエストした後で閾値時間

50

と同時に、またはこの閾値時間内に第1のGPUが第1のコピーをリクエストすると、第2のGPUは、第1のGPUがメモリコントローラ204から第1のコピーを受信するまで第2のコピーの受信を中断される。

【0074】

サーバシステム、クライアントデバイス、ゲームコントローラおよびメディアデバイス
図5Aは、いくつかの実現例に係る、サーバシステム114を示すブロック図である。図5Bは、いくつかの実現例に係る、サーバ500に格納されたデータを示すブロック図である。サーバシステム114は通常、1つ以上の処理ユニット（たとえば、CPU18（複数可）、GPU140（複数可）およびエンコーダ142）、1つ以上のネットワークインターフェース504、メモリ506、およびこれらのコンポーネント（チップセットとも呼ばれる）を相互接続するための1つ以上の通信バス508を備える。サーバシステム114は必要であれば、キーボード、マウス、音声コマンド入力ユニットもしくはマイクロホン、タッチスクリーンディスプレイ、タッチ感知入力パッド、ジェスチャ撮像カメラ、または他の入力ボタンもしくは制御装置などの、ユーザ入力を容易にする1つ以上の入力デバイス510を備え得る。さらに、サーバシステム114は、マイクロホンおよび音声認識またはカメラおよびジェスチャ認識を用いてキーボードを補足する、またはこれに置換わる。いくつかの実現例では、サーバシステム114は必要であれば、たとえば、電子デバイスに印刷されたグラフィックシリーズコードを撮像するための、1つ以上のカメラ、スキャナ、または光検出ユニットを備える。サーバシステム114はまた、1つ以上のスピーカおよび/または1つ以上のビジュアルディスプレイを含む、ユーザインターフェースおよびディスプレイコンテンツの提示を可能にする1つ以上の出力デバイス512を備え得る。

10

20

【0075】

メモリ506は、DRAM、SRAM、DDR RAMなどの高速ランダムアクセスメモリ、またはその他のランダムアクセス固体記憶装置を含み、必要であれば、1つ以上の磁気ディスク記憶装置、1つ以上の光学ディスク記憶装置、1つ以上のフラッシュメモリデバイスなどの不揮発性メモリ、または1つ以上のその他の不揮発性固体記憶装置を含む。メモリ506は必要であれば、1つ以上の処理ユニットから離れて位置する1つ以上のストレージデバイスを含む。メモリ506、または代替的にメモリ506内の不揮発性メモリは、非一時的なコンピュータ可読記憶媒体を含む。いくつかの実現例では、メモリ506は、メインダイナミックメモリ144、スタティックメモリプール202およびシェーダキャッシュ318のサブセットまたは全てを含む。いくつかの実現例では、メモリ506またはメモリ506内の非一時的なコンピュータ可読記憶媒体は、以下のプログラム、モジュール、およびデータ構造、またはそれらのサブセットもしくは上位セットを格納する。

30

【0076】

●さまざまな基本システムサービス进行处理するための、およびハードウェア依存タスクを実行するためのプロシージャを含むオペレーティングシステム516。

【0077】

●1つ以上のネットワークインターフェース504（有線または無線）およびインターネット、他のワイドエリアネットワーク、ローカルエリアネットワーク、メトロポリタンエリアネットワークなどの1つ以上の通信ネットワーク112を介して、サーバ500（たとえば、ゲームサーバ118）を他のデバイス（たとえば、サーバシステム114内の他のサーバ、クライアントデバイス102および/または104、ならびにメディアデバイス106）に接続するためのネットワーク通信モジュール518。

40

【0078】

●クライアントデバイス102および/または104において、情報の提示を可能にするためのユーザインターフェースモジュール520（たとえば、アプリケーション（複数可）、ウィジェット、ウェブサイトおよびそのウェブページ、ならびに/またはゲーム、音声および/もしくは映像コンテンツ、テキストなどを提示するするためのグラフィカルユ

50

ーザインターフェース)。

【0079】

●デバイスの提供、デバイス制御、およびメディアデバイス106(複数可)と関連付けられたユーザアカウント管理のためのデバイスサーバ側機能を提供するように実行されるメディアデバイスモジュール522(任意)。

【0080】

●クライアントデバイス(102および/または104)ならびにメディアデバイス106のいずれかの位置情報に基づいて、クライアントデバイス102および/または104の位置を判断するための位置判断モジュール524。

【0081】

●ゲームセッションの設定、セッション状態および他のゲーム関連データの格納、クライアントデバイス102からのゲームプレー入力の処理、ならびにゲームプレー入力に応じたゲームプレー出力(たとえば、画像フレームのシーケンス)のレンダリングを含むがこれらに限定されない、1つ以上のゲームタイトルと関連付けられたサーバ側機能を提供するためのサーバ側ゲームモジュール526。

【0082】

●メディアストリーミングサイトのホスト、オンラインゲームセッションと関連付けられた同時付帯(concurrent ancillary)または補足メディアストリームの受信、および同じクライアントデバイス104または別個のクライアントデバイス102のゲームアプリケーション629上で実行中のオンラインゲームセッションと同時に表示するための同時メディアストリームのクライアントデバイス104への提供のためのメディアストリーミングサーバモジュール538。

【0083】

●クライアントデバイス102/104への接続のネットワーク帯域幅の評価、およびクライアントデバイス102/104がゲームをプレーするために必要なモジュールまたはアプリケーションを有しているかどうかの評価を含むがこれらに限定されない、クライアントデバイス102/104のデバイスおよびネットワーク性能を評価するためのデバイス/ネットワーク評価モジュール540。

【0084】

●データ(たとえば、ゲームコントローラ構成556、ソフトウェアアップデートなど)をクライアントデバイス102/104に提供するためのデータ送信モジュール542。

【0085】

●以下を含むサーバシステムデータ544。

○共通デバイス設定(たとえば、サーバ階層、デバイスモデル、記憶容量、処理能力、通信能力など)を含む、クライアントデバイス102および104と関連付けられた情報を格納するためのクライアントデバイス設定552。

【0086】

○アカウントアクセス情報およびデバイス設定のための情報(たとえば、サーバ階層、デバイスモデル、記憶容量、処理能力、通信能力など)のうち1つ以上を含む、メディアデバイスモジュール522のユーザアカウントと関連付けられた情報を格納するためのメディアデバイス設定554(任意)。

【0087】

○クライアントデバイス102、レビュークライアントデバイス104およびメディアデバイス106のうちいずれかの存在、近接度または位置と関連付けられた情報を含む位置/近接度データ556。

【0088】

○さまざまなゲームタイトルについてコントローラ構成を格納するためのゲームコントローラ構成558。

【0089】

○たとえばユーザアカウント情報(たとえば、本人確認およびパスワード)、会員の種類

10

20

30

40

50

、プリファレンス、および活動履歴を含む、サーバシステム 114 上でホストされる 1 つ以上のゲームタイトル（たとえば、ゲームアプリケーション 629、図 6）の各々のユーザアカウントと関連付けられた情報を格納するためのユーザ情報 558。

【0090】

○たとえば第 1 のゲームセッションのためのデータ 562 および第 2 のゲームセッションのためのデータ 564 を含む、ゲームセッションと関連付けられたデータを格納するためのゲームセッションデータ 560 であり、各ゲームセッションのためのセッションデータ 560 は、それぞれのゲームセッションと関連付けられた 1 つ以上の画像フレームのフレームレート、レンダリング仕様、標準レイテンシ要件、GPU 割当て情報、エンコード割当て情報、関連セッションの識別、最新状態情報、シェーダパイプライン識別、テクスチャページ 308A、頂点ページ 308B、定数ページ 308C、およびフレームデータ 308D を含むが、これらに限定されない。

10

【0091】

○複数のオンラインゲームセッションによって共有されるデータを格納するためのスタティックデータ項目 568 であり、スタティックデータ項目 562 は、テクスチャオブジェクト、頂点データバッファオブジェクト、定数データオブジェクト、および異なる性能基準を満たすように仕様統計に基づいて最適化される、コンパイルされたシェーダのシーケンスのうち 1 つ以上を含むが、これらに限定されない。

【0092】

○コンパイルされグラフィックスパイプラインプロセスにおける実行の準備が整っているシェーダを格納するためのコンパイルされたシェーダのシーケンス 570 であり、コンパイルされたシェーダのシーケンスは、デフォルト設定でゲームコンテンツをレンダリングするために用いられるコンパイルされたシェーダのデフォルトシーケンス、および異なる性能基準を満たすために仕様統計に基づいて最適化されるコンパイルされたシェーダの複数のシーケンスを含むが、これらに限定されない。

20

【0093】

○最適化されたコンパイルされたシェーダのシーケンスと関連して格納される性能基準 572 および仕様統計 574 であり、性能基準 572 の各々は、全消費電量、コードサイズ、平均レイテンシ、全キャッシュ使用量、および複数のオンラインゲームセッションを実行している過程におけるコンパイルされたシェーダのシーケンスの全メモリ使用量のうち 1 つ以上に対応し、仕様統計 574 は、複数のオンラインゲームセッションの各々の実行を特徴付ける。

30

【0094】

いくつかの実現例では、サーバ側ゲームモジュール 526 は、以下のプログラム、モジュール、またはこれらのサブセットもしくは上位セットを含む。

【0095】

●オンラインゲームセッションの認証、セッションのサブセットを実行するための GPU の割当て、割当てられた GPU に対応するセッションを実行させる命令、および割当てられた GPU によってレンダリングされた画像フレームの符号化を含む、1 つ以上のゲームタイトルに対応するオンラインゲームセッションを実行するためのゲームアプリケーション 528（複数可）。

40

【0096】

●専門のゲームコントローラデバイス 102B にインタラクトしこれを制御するための、たとえば、ゲームコントローラデバイス 102B からユーザリクエストおよび入力を受信するためのコントローラデバイスモジュール 530。

【0097】

●ゲームタイトルに対応するオンラインゲームセッションについて、画像フレームをレンダリングするグラフィックスパイプラインプロセスのグラフィックス処理の順序シーケンスで用いられるシェーダをコンパイルするためのシェーダコンパイラ 532。

【0098】

50

●コンパイルされたシェーダのシーケンスの性能を性能基準に関して改良するために、オンラインゲームを実行するために用いられているコンパイルされたシェーダのシーケンスにおける１つ以上のシェーダを修正するためのシェーダパイプライン最適化５３４。

【００９９】

●オンラインゲームセッションの実行中のGPU１４０を制御して、デフォルトのコンパイルされたシェーダのシーケンスまたは最適化されたコンパイルされたシェーダのシーケンスを用いて画像フレームをレンダリングするためのゲームレンダリングモジュール５３６。

【０１００】

いくつかの実現例では、サーバシステム１１４はさらに、１つ以上の通信バス５０８を用いてCPU、GPU、メインダイナミックメモリ１４４およびスタティックメモリプール２０２に相互接続された１つ以上のメモリコントローラ２０４を備える。１つ以上のメモリコントローラ２０４は、メインダイナミックメモリ１４４を管理するように構成された第１のメモリコントローラ２０４Ａと、スタティックメモリプール２０２を管理するように構成された第２のメモリコントローラ２０４Ｂとを備える。既に示したように、構成要素５１６～５４２および５５２～５６６はメインダイナミックメモリ１４４に格納され、第１のメモリコントローラ２０４Ａを介してアクセス可能である。スタティックデータ項目５６８はスタティックメモリプール２０２に格納されており、第２のメモリコントローラ２０４Ｂを介してアクセス可能である。コンパイルされたシェーダ５７０、性能基準５７２および使用統計５７４は、必要であれば、シェーダキャッシュ３１８またはスタティックメモリプール２０２と一緒に格納される。必要であれば、オンラインゲームセッションの特定のセットについて収集された使用統計５７４は、ゲームセッションデータ５６２としてメインダイナミックメモリ１４４に格納される。

【０１０１】

上記のように識別された構成要素の各々は、既に記載されたメモリデバイスの１つ以上に格納されてもよく、上述の機能を実行するための命令のセットに対応する。上記のように識別されたモジュールまたはプログラム（すなわち、命令のセット）は、別のソフトウェアプログラム、プロシージャ、モジュールまたはデータ構造として実現される必要はなく、そのため、これらのモジュールのさまざまなサブセットは、さまざまな実現例で組合されてもよい、または他の態様では再配置されてもよい。いくつかの実現例では、メモリ５０６は必要であれば、上記のように識別されたモジュールおよびデータ構造のサブセットを格納する。さらに、メモリ５０６は必要であれば、記載されていない付加的なモジュールおよびデータ構造を格納する。

【０１０２】

図６は、いくつかの実現例に係る、ゲーム環境１００のクライアントデバイス６００の例（たとえば、クライアントデバイス１０２または１０４）を示すブロック図である。クライアントデバイス６００の例は、携帯電話、タブレットコンピュータ、ラップトップコンピュータ、デスクトップコンピュータ、およびウェアラブルパーソナルデバイスを含むが、これらに限定されない。クライアントデバイス６００は、１つ以上の処理ユニット（CPU）６０２、１つ以上のネットワークインターフェース６０４、メモリ６０６、およびこれらのコンポーネント（チップセットとも呼ばれる）を相互接続するための１つ以上の通信バス６０８を含む。クライアントデバイス６００は、キーボード、マウス、音声コマンド入力ユニットもしくはマイクロホン、タッチスクリーンディスプレイ、タッチ感知入力パッド、ジェスチャ撮像カメラ、または他の入力ボタンもしくは制御装置などの、ユーザ入力を容易にする１つ以上の入力デバイス６１０を含み得る。さらに、クライアントデバイス６００のうち一部は、マイクロホンおよび音声認識またはカメラおよびジェスチャ認識を用いてキーボードを補足する、またはこれに置換わる。いくつかの実現例では、クライアントデバイス６００は、たとえば電子デバイスに印刷されたグラフィックシリーズコードを撮像するための１つ以上のカメラ、スキャナ、または光検出ユニットを備える。

10

20

30

40

50

クライアントデバイス 600 はまた、1 つ以上のスピーカおよび / または 1 つ以上のビジュアルディスプレイを含む、ユーザインターフェースおよびディスプレイコンテンツの提示を可能にする 1 つ以上の表示デバイス 612 を含む。必要であれば、クライアントデバイス 600 は、クライアントデバイス 600 の位置を判断するための、GPS (全地球測位衛星) または他の地理位置受信機などの、位置検出デバイス 614 を含む。クライアントデバイス 600 はまた、たとえば IR センサなどの、メディアデバイス 106 および / またはプレーヤクライアント 104 の近接度を判断するための、近接度判断デバイス 615 を含み得る。クライアントデバイス 600 は、入力として使用可能なクライアントデバイス 600 の動き、方位、および他のパラメータを検知するための 1 つ以上のセンサ 613 (たとえば、加速度計、ジャイロスコープなど) を含み得る。

10

【 0103 】

メモリ 606 は、DRAM、SRAM、DDR RAM などの高速ランダムアクセスメモリ、またはその他のランダムアクセス固体記憶装置を含み、必要であれば、1 つ以上の磁気ディスク記憶装置、1 つ以上の光学ディスク記憶装置、1 つ以上のフラッシュメモリデバイスなどの不揮発性メモリ、または 1 つ以上のその他の不揮発性固体記憶装置を含む。メモリ 606 は必要であれば、1 つ以上の処理ユニット 602 から離れて位置する 1 つ以上のストレージデバイスを含む。メモリ 606、または代替的にメモリ 606 内の不揮発性メモリは、非一時的なコンピュータ可読記憶媒体を含む。いくつかの実現例では、メモリ 606 またはメモリ 606 の非一時的なコンピュータ可読記憶媒体は、以下のプログラム、モジュール、およびデータ構造、またはこれらのサブセットもしくは上位セットを含む。

20

【 0104 】

• さまざまな基本システムサービス进行处理するための、およびハードウェア依存タスクを実行するためのプロシージャを含むオペレーティングシステム 616。

【 0105 】

• 1 つ以上のネットワークインターフェース 604 (有線または無線) およびローカルネットワーク 110、ならびにインターネット、他のワイドエリアネットワーク、ローカルエリアネットワーク、メトロポリタンエリアネットワークなどの 1 つ以上の通信ネットワーク 112 を介して、クライアントデバイス 600 を他のデバイス (たとえば、サーバシステム 114、メディアデバイス 106、および他のクライアントデバイス 102 または 104) に接続するためのネットワーク通信モジュール 618。

30

【 0106 】

• 1 つ以上の出力デバイス 612 (たとえば、ディスプレイ、スピーカなど) を介してクライアントデバイス 600 において、情報の提示を可能にするためのユーザインターフェースモジュール 620 (たとえば、アプリケーション、ウィジェット、ウェブサイトおよびそのウェブページ、ならびに / またはゲーム、音声および / もしくは映像コンテンツ、テキストなどを提示するためのグラフィカルユーザインターフェース)。

【 0107 】

• 1 つ以上の入力デバイス 610 のうち 1 つからの 1 つ以上のユーザ入力またはインタラクションを検出し、検出された入力またはインタラクションを解釈するための入力処理モジュール 622。

40

【 0108 】

• セッション 116 に参加するためのウェブインターフェースを含む、ウェブサイトおよびそのウェブページのナビゲーション、リクエスト (たとえば HTTP を介して)、ならびに表示を行うためのウェブブラウザモジュール 624。

【 0109 】

• メディアデバイス 106 と関連付けられたユーザアカウントへのログイン、ユーザアカウントと関連付けられている場合はメディアデバイス 106 の制御、ならびにメディアデバイス 106 と関連付けられた設定およびデータの編集およびレビューを含む、メディアデバイス 106 とインタラクトするためのメディアデバイスアプリケーション 626。

50

【 0 1 1 0 】

●ゲームセッションのセットアップ、セッション状態データおよび他のゲーム関連データのローカルな格納、入力デバイス 6 1 0 からのゲームプレー入力の受信、ならびにゲームプレー入力に応じたゲームプレー出力（たとえば、画像フレームのシーケンス）の提供を含むがこれらに限定されない、1つ以上のゲームタイトルと関連付けられたクライアント側機能を提供するためのクライアント側ゲームモジュール 6 2 8 であって、いくつかの実現例では、クライアント側ゲームモジュール 6 2 8 はさらに、1つ以上のゲームタイトルに対応する1つ以上のゲームアプリケーション 6 2 9 を含む。

【 0 1 1 1 】

●データ（たとえば、ゲームコントローラ構成 6 4 0、クライアント側ゲームモジュール 6 2 8 および他のアプリケーション、モジュール、アプリケーション、およびメモリ 6 0 6 内のデータに対するアップデート）をサーバシステム 1 1 4 ならびに他のコンテンツホストおよびプロバイダからダウンロードするためのデータダウンロードモジュール 6 3 0。

10

【 0 1 1 2 】

●少なくともゲームアプリケーション 6 2 9 および他のアプリケーション/モジュールと関連付けられたデータを格納するクライアントデータ 6 3 2 であって、以下を含む。

【 0 1 1 3 】

○共通のデバイス設定（たとえば、サーバ階層、デバイスモデル、記憶容量、処理能力、通信能力など）を含む、クライアントデバイス 6 0 0 自体と関連付けられた情報を格納するためのクライアントデバイス設定 6 3 4。

20

【 0 1 1 4 】

○アカウントアクセス情報およびデバイス設定のための情報（たとえば、サーバ階層、デバイスモデル、記憶容量、処理能力、通信能力など）のうち1つ以上を含むがこれらに限定されない、メディアデバイスアプリケーション 6 2 6 のユーザアカウントと関連付けられた情報を格納するためのメディアデバイス設定 6 3 6。

【 0 1 1 5 】

○アカウントアクセス情報、ゲーム内のユーザのプリファレンス、ゲームプレー履歴データ、および他のプレーヤについての情報のうち1つ以上を含む、ゲームアプリケーション（複数可）のユーザアカウントと関連付けられた情報を格納するためのゲームアプリケーション 6 3 8（複数可）。

30

【 0 1 1 6 】

○ゲームアプリケーション 6 2 8（複数可）についてのクライアント側ゲームモジュール 6 2 8 の構成（たとえば、ゲームコントローラ構成 4 5 6 から受信された構成、図 4）と関連付けられた情報を格納するためのゲームコントローラ構成 6 4 0（複数可）。

【 0 1 1 7 】

○クライアントデバイス 1 0 2 および 1 0 4 ならびにメディアデバイス 1 0 6 のうちいずれかの存在、近接度または位置と関連付けられた情報を含む位置データ 6 4 2。

【 0 1 1 8 】

○異なるゲームタイトルと関連付けられたクライアントデバイス 6 0 0 上で実行されるオンラインゲームセッションと関連付けられた情報、たとえば、最新の状態情報、頂点ページ、定数ページ、およびオンラインゲームセッションの各々と関連付けられた1つ以上の画像フレームのフレームデータを含むローカルゲームセッションデータ 6 4 4。

40

【 0 1 1 9 】

図 6 に示されないいくつかの実現例では、クライアント側ゲームモジュール 6 2 8 は、メニューモードでクライアントデバイス 1 0 2 とメディアデバイス 1 0 6 との間でメッセージ（たとえば、セットアップコマンド）を送信するためのコマンドモジュール、ゲームプレーモードでクライアントデバイス 1 0 2 とサーバシステム 1 1 4 との間（たとえば、メディアストリーミングサーバ 1 2 0 に対する）ゲームプレーコマンドを送信するためのストリーミングモジュール、およびゲームアプリケーションにゲームプレー入力インターフェースを提供するためのコントローラモジュールを含む。いくつかの実現例では、クライ

50

アント側ゲームモジュール628は、メディアデバイスアプリケーション626またはメモリ606内の他のアプリケーションの一部（たとえば、サブモジュール）である。いくつかの実現例では、クライアント側ゲームモジュール628は、オペレーティングシステム616の一部である。いくつかの実現例では、クライアント側ゲームモジュール628は、別個のモジュールまたはアプリケーションである。

【0120】

クライアントデバイス102のいくつかの実現例では、メディアデバイスアプリケーション626（および、それに応じてメディアデバイス設定636）ならびにゲームアプリケーション629（および、それに応じてゲームアプリケーション設定638）は任意である。クライアントデバイス102が参加するように招待される特定のゲームにตอบสนองして、メディアデバイスアプリケーション626およびゲームアプリケーション629は、プレーするように要求されない。これらのアプリケーションのいずれかがゲームをプレーするために必要であり（たとえば、ゲームは、メディアデバイスアプリケーション626内のクライアント側ゲームモジュール628を使用し）、このアプリケーションがメモリ606内にない場合、クライアントデバイス102は、このアプリケーションをダウンロードするように促される。

10

【0121】

いくつかの実現例では、クライアントデバイス102は、ゲームコントローラデバイスを含む。ゲームコントローラデバイスは、マイクロホン（または音声コマンド入力ユニット）、ボタン、方向パッド（複数可）、およびジョイスティック（複数可）などの、ユーザ入力を容易にする1つ以上の入力デバイス610を含む。いくつかの実現例では、ゲームコントローラ102はさらに、ジェスチャ認識特徴を含む。ゲームコントローラ102はまた、スピーカ320（複数可）、LED（複数可）、および/または必要であればディスプレイなどの、音声出力および/またはビジュアル出力を容易にする1つ以上の出力デバイスを含む。

20

【0122】

上述のように識別された構成要素の各々は、既に記載したメモリデバイスのうち1つ以上に格納されてもよく、上述の機能を実行するための命令のセットに対応する。上述のように識別されたモジュールまたはプログラム（すなわち、命令のセット）は、別のソフトウェアプログラム、プロシージャ、モジュールまたはデータ構造として実現される必要はなく、そのため、これらのモジュールのさまざまなサブセットはさまざまな実現例で組み合わせられてもよい、または他の態様では再配置されてもよい。いくつかの実現例では、メモリ606は必要であれば、上述のように識別されたモジュールおよびデータ構造のサブセットを格納する。さらに、メモリ606は必要であれば、上述されていない付加的なモジュールおよびデータ構造を格納する。

30

【0123】

図7は、いくつかの実現例に係るゲーム環境100のメディアデバイス106の例を示すブロック図である。メディアデバイス106は通常、1つ以上の処理ユニット（CPU）702、1つ以上のネットワークインターフェース704、メモリ706、およびこれらのコンポーネント（チップセットとも呼ばれる）を相互接続するための1つ以上の通信バス708を含む。必要であれば、メディアデバイス106は、IRセンサなどの、クライアントデバイス102または104の近接度を判断するための近接度/位置検出ユニット710を含む。

40

【0124】

さらにいくつかの実現例では、メディアデバイス106は、1つ以上の通信ネットワークを使用可能にし、かつ、メディアデバイス106に他のデバイスと（たとえば、クライアントデバイス102と）通信させる無線機を含む。いくつかの実現例では、無線機は、さまざまなカスタムまたは標準無線プロトコル（たとえば、IEEE 802.15.4、Wi-Fi、ZigBee、6LoWPAN、Thread、Z-Wave、Bluetooth Smart、ISA100.5A、WirelessHART、MiWi、

50

Whispernet、Bluetooth low energy など)、カスタムまたは標準有線プロトコル(たとえば、Ethernet、HomePlug など)、および/または本特許出願日においてまだ開発されていない通信プロトコルを含む、他の好適な通信プロトコルのうちいずれかを用いてデータ通信が可能である。

【0125】

メモリ706は、DRAM、SRAM、DDR RAMなどの高速ランダムアクセスメモリ、またはその他のランダムアクセス固体記憶装置を含み、必要であれば、1つ以上の磁気ディスク記憶装置、1つ以上の光学ディスク記憶装置、1つ以上のフラッシュメモリデバイスなどの不揮発性メモリ、または1つ以上のその他の不揮発性固体記憶装置を含む。メモリ706は必要であれば、1つ以上の処理ユニット702から離れて位置する1つ以上のストレージデバイスを含む。メモリ706、または代替的にメモリ706内の不揮発性メモリは、非一時的なコンピュータ可読記憶媒体を含む。いくつかの実現例では、メモリ706またはメモリ706の非一時的なコンピュータ可読記憶媒体は、以下のプログラム、モジュール、およびデータ構造、またはこれらのサブセットもしくは上位セットを含む。

10

【0126】

●さまざまな基本システムサービス进行处理するための、およびハードウェア依存タスクを実行するためのプロシージャを含むオペレーティングシステム716。

【0127】

●1つ以上のネットワークインターフェース704(有線または無線)およびインターネット、他のワイドエリアネットワーク、ローカルエリアネットワーク、メトロポリタンエリアネットワーク、ケーブルテレビシステム、衛星テレビシステム、IPTVシステムなどの1つ以上の通信ネットワーク112を介して、メディアデバイス106を他のコンピュータまたはシステム(たとえば、サーバシステム114およびクライアントデバイス102)に接続するためのネットワーク通信モジュール718。

20

【0128】

●1つ以上のコンテンツソース(たとえば、ゲームセッション116からの出力のためのサーバシステム114)から受信されたコンテンツ信号を復号し、復号された信号内のコンテンツをメディアデバイス106に接続された出力デバイス108に出力するためのコンテンツ復号モジュール720。

30

【0129】

●近接度検出ユニット710によって検出されたまたはサーバシステム114によって提供された近接度関連情報に基づいて、クライアントデバイス102および/または104の近接度を判断するための近接度判断モジュール722。

【0130】

●メディアディスプレイを制御するためのメディア表示モジュール724。
●メディアデバイス106上でゲームコンポーネントの全てのライフサイクルを制御し、ゲームコントローラ102からメッセージ(たとえば、コマンド)を受信し、ゲームコントローラ入力をサーバシステム114にリレーするためのゲームアプリケーション726。

【0131】

●ユーザコンテンツ(たとえば、プロフィール、アバター、購入されたゲーム、ゲームカタログ、友達、メッセージ)および表示のために最適化された画像にアクセスし、ゲームプレー中にクライアントデバイス102から入力を受信するためのゲームユーザインターフェースモジュール728。

40

【0132】

●ゲームアプリケーション726とゲームユーザインターフェースモジュール728との間の通信をインターフェース接続し、ゲーム起動中にサーバシステム114とのセッションペ어링リクエストを開始するためのインターフェース制御モジュール730。

【0133】

●サーバシステム114および認証システム(設けられている場合)と通信するためのサ

50

ーバイインターフェースモジュール 7 3 2 であり、以下を含む。

【 0 1 3 4 】

○認証システムと通信してゲームコントローラのリンクされたユーザを認証するための認証サブモジュール 7 3 4。

【 0 1 3 5 】

○サーバシステム 1 1 4 からゲームセッション情報（たとえば、セッション ID、ゲームサーバ提供 IP アドレス、ポートなど）を受信し、この情報を提供するインターフェース制御モジュール 7 3 0 へのコールバックを呼び出すためのセッションサブモジュール 7 3 6。

【 0 1 3 6 】

○サーバシステム 1 1 4 のメディアストリーミングサーバ 1 2 0 から送信されたゲームストリームを、ゲームプレー中にリアルタイムで受信し表示するためのストリーミングサブモジュール 7 3 8。

【 0 1 3 7 】

●少なくとも以下を含むデータを格納するメディアデバイスデータ 7 4 0。
○アカウントアクセス情報およびデバイス設定（たとえば、サーバ階層、デバイスモデル、記憶容量、処理能力、通信能力など）のための情報を含む、メディアデバイスアプリケーションのユーザアカウントと関連付けられた情報を格納するためのメディアデバイス設定 7 4 2。

【 0 1 3 8 】

○クライアントデバイス 1 0 2 および / または 1 0 4、ならびにメディアデバイス 1 0 6 のいずれかの存在、近接度または位置と関連付けられた情報を含む位置データ 7 4 4。

【 0 1 3 9 】

○異なるゲームタイトルと関連してクライアントデバイス 6 0 0 上で実行されるオンラインゲームセッションと関連付けられた情報、たとえば、オンラインゲームセッションの各々と関連付けられた 1 つ以上の画像フレームの最新の状態情報、テクスチャページ、頂点ページ、定数ページ、およびフレームデータと関連してを含むローカルゲームセッションデータ 7 4 6。

【 0 1 4 0 】

上述のように識別された構成要素の各々は、既に説明されたメモリデバイスのうち 1 つ以上に格納されてもよく、上述の機能を実行するための命令のセットに対応する。上述のように識別されたモジュールまたはプログラム（すなわち、命令のセット）は、別のソフトウェアプログラム、プロシージャ、モジュールまたはデータ構造として実現される必要はなく、そのため、これらのモジュールのさまざまなサブセットがさまざまな実現例で組み合わせられてもよい、または他の態様では再配置されてもよい。いくつかの実現例では、メモリ 7 0 6 は必要であれば、上述のように識別されたモジュールおよびデータ構造のサブセットを格納する。さらに、メモリ 7 0 6 は必要であれば、記載されていないさらに別のモジュールおよびデータ構造を格納する。

【 0 1 4 1 】

スタティックメモリプール

図 8 A および図 8 B は、いくつかの実現例に係る、1 つ以上のゲームタイトルをホストするサーバシステムの格納能力を管理するための方法の例を示すフローチャート 8 0 0 である。方法 8 0 0 は必要であれば、コンピュータメモリまたは非一時的なコンピュータ可読記憶媒体（たとえば、図 5 A のサーバ側ゲームモジュール 5 2 6）に格納され、かつ、サーバシステム 1 1 4（たとえば、ゲームサーバ 1 1 8 の CPU および GPU）の 1 つ以上のプロセッサによって実行される命令によって制御される。コンピュータ可読記憶媒体は、磁気もしくは光学ディスク記憶装置、フラッシュメモリなどの固体記憶装置、または他の不揮発性メモリデバイスもしくはデバイスを含み得る。コンピュータ可読記憶媒体に格納される命令は、ソースコード、アセンブリ言語コード、オブジェクトコード、または 1 つ以上のプロセッサによって解釈される他の命令フォーマットのうち 1 つ以上を含み得る。

10

20

30

40

50

方法 800 におけるいくつかの動作は組合わされてもよい、および / または、いくつかの動作の順番は変更されてもよい。

【0142】

サーバシステム 114 は、1つ以上の CPU 138 と、複数の GPU 140 と、プログラムの実行中に CPU および / または GPU によって使用されるためのプログラムおよびデータを格納するメインダイナミックメモリ 144 と、不揮発性メモリに格納されたスタティックメモリプール 202 と、スタティックメモリプールを管理するように構成されたメモリコントローラ 204 とを備える。GPU 140 の各々は、ローカルキャッシュを含み、メモリコントローラ 204 を介してスタティックメモリプール 202 にアクセスするように構成される。いくつかの実現例では、スタティックメモリプール 202 は、サーバシステム 114 の1つ以上の CPU 138 および複数の GPU 140 に近接して配設され、これらによって共有される。CPU 138 および GPU 140 は、データバスを介してスタティックメモリプール 202 にアクセスする。さらにいくつかの実現例では、メインダイナミックメモリ 144 は、サーバシステムの1つ以上の CPU および複数の GPU に近接して配設され、これらによって共有される。CPU 138 および GPU 140 は、データバスを介してメインダイナミックメモリ 144 にアクセスする。いくつかの実現例では、スタティックメモリプール 202 は、サーバシステム 114 の1つ以上の CPU および複数の GPU の各々から距離を有し、この距離は、所定の距離閾値よりも小さい（たとえば、< 3メートル）。そうは言っても、スタティックメモリプール 202 は、サーバシステム 114 の1つ以上の CPU および複数の GPU がその上に配設されるサーバラックに物理的に位置する。

10

20

【0143】

いくつかの実現例では、メモリコントローラ 204 は、メインダイナミックメモリとスタティックメモリプールとの両方を管理するように構成される。代替的にいくつかの実現例では、メモリコントローラ 204 は第1のメモリコントローラを含み、サーバシステムはさらに、第1のメモリコントローラから離れた、メインダイナミックメモリを管理するように構成された第2のメモリコントローラを備える。

【0144】

サーバシステム 114 は、ゲームタイトルについて1つ以上の CPU 138 上で並行に複数のゲームセッションを実行する（802）。複数のゲームセッションの各々は、スタティックデータ項目と関連付けられており、スタティックデータ項目を用いてそれぞれの GPU 140 によって実行可能なグラフィックス処理を必要とし、スタティックデータ項目は、スタティックメモリプール 202 に格納される。いくつかの実現例では、グラフィックス処理は、OpenGL グラフィックスレンダリングパイプラインの一部であり、頂点処理、テッセレーション、幾何学処理、プリミティブアセンブリ、ラスタ化、フラグメンテーションおよびプリサンプリングのうち1つを含む。いくつかの実現例では、スタティックデータ項目は、ゲーム開発段階中にゲームタイトルの開発者によって事前に定義される。ゲームタイトルがサーバシステムに最初にロードされると、事前に定義されたスタティックデータ項目はスタティックメモリプール 144 にロードされる。いくつかの実現例では、スタティックデータ項目は、OpenGL レンダリングパイプラインで使用される1つ以上の標準パラメータを含む。スタティックデータ項目の例は、OpenGL ライブラリから初めからロードされていた標準テクスチャ、頂点、幾何学、および定数を含むが、これらに限定されない。

30

40

【0145】

1つ以上の CPU 138 上で実行される複数のゲームセッションのうち第1のゲームセッションについて（804）、CPU 138 は、グラフィックス処理を第1の GPU に割当てる（806）。第1の GPU は、メモリコントローラ 204 にスタティックデータ項目の第1のコピーをリクエストし（808）、スタティックメモリプール 204 から読出されたスタティックデータ項目の第1のコピーをメモリコントローラから受信する（810）。いくつかの実現例では、スタティックメモリプールから読出されたスタティックデー

50

タ項目の第1のコピーをメモリコントローラから受信すると、第1のGPUは、一時的にローカルキャッシュまたはメインダイナミックメモリに第1のコピーを格納する。第1のGPUはその後、スタティックデータ項目の第1のコピーを用いてグラフィックス処理を実行し(812)、複数のゲームセッションのうち第1のゲームセッションと関連して、グラフィックス処理の第1の結果をローカルに格納する(814)。いくつかの実現例では、第1のGPUは、複数のGPUコアおよび複数のGPUコアによってシェアされる共通キャッシュを有し、各GPUコアは、それぞれのGPUコアによって使用される命令およびデータを格納するそれぞれのキャッシュメモリを含む。

【0146】

いくつかの実現例では、サーバシステム114は、グラフィックス処理を行うために複数のGPUによって使用される1つ以上のコンパイルされたシェーダを格納するシェーダキャッシュを備える。さらにいくつかの実現例では、サーバシステム114は、複数のゲームセッションのうち第1のゲームセッションについて性能基準を識別し、この性能基準に従って、複数のコンパイルされたシェーダのうち1つのシェーダを選択する。複数のコンパイルされたシェーダのうち選択された1つのシェーダは、グラフィックス処理を行うために第1のGPUによって使用される。

【0147】

同様に、1つ以上のCPU上で実行される複数のゲームセッションのうち第2のゲームセッションについて(816)、CPU138は、グラフィックス処理を第2のGPUに割当てる(818)。第2のGPUは、メモリコントローラにスタティックデータ項目の第2のコピーをリクエストし(820)、スタティックメモリプール202から読出されたスタティックデータ項目の第2のコピーをメモリコントローラ204から受信する(822)。第2のGPUはその後、スタティックデータ項目の第2のコピーを用いてグラフィックス処理を実行し(824)、複数のゲームセッションのうち第2のゲームセッションと関連して、グラフィックス処理の第2の結果をローカルに格納する。

【0148】

いくつかの実現例では、メインダイナミックメモリ144はダイナミックランダムアクセスメモリを含み、GPUのローカルキャッシュおよび1つ以上のCPUのローカルキャッシュの各々は、スタティックランダムアクセスメモリを含む。スタティックメモリプールは、ソリッドステートドライブ、フラッシュメモリおよび相変化ランダムアクセスメモリのうち1つを含む。ローカルキャッシュへのアクセスレートは、スタティックメモリプールへのアクセスレートよりも速いメインダイナミックメモリ144へのアクセスレートよりも速い。いくつかの実現例では、スタティックメモリプールへのアクセスによって生じるレイテンシが、それぞれのGPUにおけるゲームコンテンツレンダリングの間に補償される。

【0149】

いくつかの実現例では、グラフィックス処理の第1の結果は、複数のゲームセッションのうち第1のゲームセッションと関連して、それぞれのGPUのローカルキャッシュ(たとえば、図3CのGPUキャッシュ364)に格納される。グラフィックス処理は第1のグラフィックス処理であり、第2のグラフィックス処理が続く。第2のグラフィックス処理の実行中、第1のGPU140は、ローカルキャッシュから第1のグラフィックス処理の第1の結果を取得し、第1のグラフィックス処理の第1の結果を用いて第2のグラフィックス処理を行う。必要であれば、第2のグラフィックス処理は、直接または間接的に第1のグラフィックス処理に続く。いくつかの実現例では、第1および第2のグラフィックス処理の両方は、複数のゲームセッションのうち第1のゲームセッションに参与するクライアントデバイスから受信されたユーザリクエストまたは入力に応答してゲームコンテンツストリームを生成するグラフィックスパイプラインプロセス(たとえば、OpenGLレンダリングパイプライン)の一部である。一例では、第1および第2のグラフィックス処理の両方は、OpenGLレンダリングパイプライン内の2つの連続する処理である。第1の結果は、OpenGLレンダリングパイプラインにおいて生成された中間結果で

10

20

30

40

50

あり、第2のグラフィックス処理によって使用された後にローカルキャッシュからページ可能である。

【0150】

代替的にいくつかの実現例では、グラフィックス処理の第1の結果は、複数のゲームセッションのうち第1のゲームセッションと関連してメインダイナミックメモリ144に格納される。グラフィックス処理は第1のグラフィックス処理を含み、第2のグラフィックス処理がその後に続く。第2のグラフィックス処理の実行中、第1のGPU140は、メインダイナミックメモリ144からグラフィックス処理の第1の結果を取得し、第1のグラフィックス処理の第1の結果を用いて、第1のグラフィックス処理に続く第2のグラフィックス処理を実行する。必要であれば、第1の結果はダイナミックデータ306の一部（たとえば、ゲームセッションデータ562）として格納される。

10

【0151】

いくつかの実現例では、サーバシステム114はさらにエンコーダ142を備える。エンコーダ142は、第1の結果を、複数のゲームセッションのうち第1のゲームセッションと関連付けられた画像フレームに符号化する。画像フレームは、複数のゲームセッションのうち第1のゲームセッションに關与するクライアントデバイスに提供される。

【0152】

いくつかの実現例では、スタティックデータ項目は、グラフィックス処理を行うために複数のGPU140によって使用される1つ以上のコンパイルされたシェーダを含む。さらにいくつかの実現例では、スタティックデータ項目が複数のコンパイルされたシェーダを含む場合、第1のGPUは、複数のゲームセッションのうち第1のゲームセッションについて性能基準を識別し、性能基準に従って、複数のコンパイルされたシェーダのうち1つのシェーダを選択する。複数のコンパイルされたシェーダのうち選択された1つのシェーダは、グラフィックス処理を行うために第1のGPUによって使用される。

20

【0153】

なお、スタティックデータ項目は、複数のゲームセッションのうち第1および第2のゲームセッションを含む複数のゲームセッションの間で共有される。いくつかの実現例では、複数のゲームセッションの各々は優先順位と関連付けられ、第1および第2のGPUの各々は、第1および第2のゲームセッションの優先順位に従ってスタティックメモリプールから読出されたスタティックデータ項目のそれぞれのコピーを、メモリコントローラから受信する。たとえば、第1のゲームセッションは第2のゲームセッションより高い優先順位を有し、第1のGPUは、第2のGPUが第2のコピーをリクエストした後で、閾値時間と同時にまたはこの時間内に第1のコピーをリクエストする。第2のGPUは、第1のGPUがメモリコントローラから第1のコピーを受信するまで、第2のコピーの受信を中断される。

30

【0154】

いくつかの実現例では、スタティックメモリプール202はハッシュテーブルを含み、複数のスタティックデータ項目が、ハッシュテーブルに従ってゲームタイトルと関連してスタティックメモリプール202に格納される。いくつかの実現例では、スタティックデータ項目は、スタティックデータ項目がリードオンリーであると示すタグと関連付けられる。

40

【0155】

いくつかの実現例では、メモリコントローラ204は、スタティックデータ項目を解放する解放リクエストを受信する。解放リクエストに回答して、メモリコントローラ204は、スタティック（たとえば、コールド）メモリプールが一杯かどうかを判断する。スタティックメモリプールが一杯であることに従って、メモリコントローラ204は、LRUページ置換方法に基づいてスタティックデータ項目を解放する。スタティックメモリプールが一杯でないことに従って、メモリコントローラ204は、スタティックデータ項目の解放を打ち切る。メモリコントローラ204は、スタティックメモリプール202に格納されたデータを解放するように構成されたページ解放モジュール374を含む。

【0156】

50

いくつかの実現例では、スタティックメモリプール 202 は複数のページに分割され、複数のページの各々は、それぞれのページに格納されたデータについて位置を識別するページハッシュを有する。メモリコントローラ 204 は、各ページのページハッシュに従って、スタティックメモリプール 202 に対する読み書きを行う。一例では、このスタティックメモリプール 202 は P R A M を含む。

【0157】

なお、図 8 A および図 8 B の処理が説明された特定の順序は例示に過ぎず、記載された順序が処理を実行可能な唯一の順序を示すと意図されていない。当業者であれば、本明細書で説明されたようなスタティックメモリプールに格納されたデータを用いてオンラインゲームセッションを実行するさまざまな方法が分かるであろう。さらに、なお、方法 900 (たとえば、図 9) に関して本明細書で説明された他のプロセスの詳細は、図 8 A ~ 図 8 B に関して説明された方法 800 に類似の態様でも適用可能である。簡潔に説明するために、これらの詳細はここでは繰り返さない。

【0158】

図 9 は、いくつかの実現例に係る、ゲームタイトルをサーバシステム 114 にロードするための方法 900 の例を示すフローチャートである。方法 900 は必要であれば、コンピュータメモリまたは非一時的なコンピュータ可読記憶媒体(たとえば、図 5 A のメモリ 506)に格納され、かつ、サーバシステム 114 の 1 つ以上のプロセッサ(たとえば、ゲームサーバ 118 の C P U および G P U)によって実行される命令によって制御される。コンピュータ可読記憶媒体は、磁気もしくは光学ディスク記憶装置、フラッシュメモリなどの固体記憶装置、または他の不揮発性メモリデバイスもしくはデバイスを含み得る。コンピュータ可読記憶媒体に格納された命令は、ソースコード、アセンブリ言語コード、オブジェクトコード、または 1 つ以上のプロセッサによって解釈される他の命令フォーマットのうち 1 つ以上を含み得る。方法 900 におけるいくつかの動作は組合わされてもよい、および/または、いくつかの動作の順番は変更されてもよい。

【0159】

サーバシステム 114 は、1 つ以上の C P U 138 と、複数の G P U 140 と、プログラムの実行中に C P U および/または G P U によって使用されるためのプログラムおよびデータを格納するメインダイナミックメモリ 144 と、不揮発性メモリに格納されたスタティックメモリプール 202 と、スタティックメモリプールを管理するように構成されたメモリコントローラ 204 とを備える。G P U 140 の各々は、メモリコントローラ 204 を介してスタティックメモリプール 202 にアクセスするように構成される。

【0160】

C P U 138 は、ゲームタイトルに対応する実行可能なゲームプログラム(たとえば、サーバ側ゲームモジュール 526)を、メインダイナミックメモリ 144 にロードする(902)。C P U 138 は、並行して複数のゲームセッションを実行する(904)ように構成され、ゲームセッションの各々は、実行可能なゲームプログラムに対応する。複数のオンラインゲームセッションの各々は、それぞれのオンラインゲームセッションをレンダリングするように割当てられたそれぞれの G P U によって実行可能なグラフィックス処理を必要とする(906)。

【0161】

C P U 138 は、複数のゲームセッションのグラフィックス処理で用いられるスタティックデータ項目を識別する(908)。いくつかの実現例では、スタティックデータ項目は第 1 のスタティックデータ項目を含み、第 1 のスタティックデータ項目は、所定のスタティックデータ項目のリストにおいて C P U 138 によって識別される。所定のスタティックデータ項目のリストは、スタティックデータ項目の第 1 のサブセットとスタティックデータ項目の第 2 のサブセットとを含む。メモリコントローラは、第 1 のスタティックデータ項目を含むスタティックデータ項目の第 1 のサブセットをスタティックメモリプール 202 に格納し、スタティックデータ項目の第 2 のサブセットをメインダイナミックメモリ 144 に格納する。

10

20

30

40

50

【 0 1 6 2 】

メモリコントローラ 2 0 4 は、スタティックデータ項目をスタティックメモリプールに格納し (9 1 0)、オンラインゲームセッションを実行している G P U に、グラフィックス処理を実行しつつスタティックデータ項目へアクセスすることを許可する (9 1 2)。いくつかの実現例では、スタティックデータ項目はゲームタイトルのレンダリングで使用される標準スタティックデータ項目である。スタティックデータ項目のスタティックメモリプールへの格納の前に、C P U 1 3 8 は、サーバシステム 1 1 4 のハードドライブ 3 2 4 またはリモートライブラリサーバからスタティックデータ項目を取出す。たとえば、スタティックデータ項目は、シェーダライブラリサーバ 1 2 4 に格納されたシェーダプログラムを含む。C P U 1 3 8 は、シェーダライブラリサーバ 1 2 4 からシェーダプログラムを取得し、ゲームタイトルをロードする過程において、スタティックメモリプール 2 0 2 にこれを格納する。代替的にいくつかの実現例では、スタティックデータ項目は、ゲームタイトルの開発者によって事前に定義されサーバシステム 1 1 4 のハードドライブ 3 2 4 に格納されたカスタムスタティックデータ項目である。スタティックデータ項目のスタティックメモリプールへの格納の前に、C P U 1 3 8 は、サーバシステム 1 1 4 のハードドライブ 3 2 4 からスタティックデータ項目を取出す。ゲームタイトルのサーバシステム 1 1 4 へのロードについては、図 4 A を参照して詳細に説明されている。

10

【 0 1 6 3 】

いくつかの実現例では、実行可能なゲームプログラムに従って、C P U 1 3 8 は、複数の G P U の各々を、その上で実行されるべき複数のオンラインゲームセッションのサブセットに割当てて。複数のゲームセッションの各々について、それぞれの G P U によるグラフィックス処理の実行中、それぞれの G P U 1 4 0 は、メモリコントローラを介してスタティックメモリプールからスタティックデータ項目のコピーを取得し、スタティックデータ項目のコピーを用いてグラフィックス処理を行う。いくつかの実現例では、複数のゲームセッションの各々について、それぞれの G P U によるグラフィックス処理の実行中、それぞれの G P U 1 4 0 は、スタティックデータ項目のコピーをそれぞれの G P U のそれぞれのローカルキャッシュにロードする。さらにいくつかの実現例では、複数のゲームセッションの各々について、それぞれの G P U によるグラフィックス処理の実行中、それぞれの G P U 1 4 0 は、それぞれのオンラインゲームセッションと関連して、ダイナミックデータ項目およびユーザ情報をメインダイナミックメモリ 1 4 4 に格納する。いくつかの実現例では、複数のゲームセッションのサブセットについて、G P U 1 4 0 は、スタティックデータ項目のコピーを G P U のローカルキャッシュに一度ロードし、複数のゲームセッションのサブセットに対応する G P U のサブセットによってスタティックデータ項目のコピーを共有する。スタティックデータ項目を用いたオンラインゲームセッションの画像フレームのレンダリングは、図 4 B および図 8 A ~ 図 8 B を参照して、より詳細に説明された。

20

30

【 0 1 6 4 】

なお、図 9 の動作が説明された特定の順序は例示に過ぎず、説明された順序は動作を実行可能な唯一の順序を示すと意図されていない。当業者であれば、本明細書で説明されたように、ゲームタイトルをサーバシステム 1 1 4 にロードするさまざまな方法が分かるであろう。さらに、なお、方法 8 0 0 (たとえば、図 8 A ~ 図 8 B) に関して本明細書で説明された他のプロセスの詳細も、図 9 に関して説明された方法 9 0 0 と類似の態様で適用可能である。簡潔に説明するために、これらの詳細はここでは繰り返さない。

40

【 0 1 6 5 】

シェーダ最適化

オンラインゲームセッションのために画像フレームをレンダリングする過程において、G P U 1 4 0 は、他のパイプライン状態データと共にソース形式 (G L S L、H L S L) または中間形式 (D X I L、S P I R - V) でシェーダを取得し、G P U が使用するためのシェーダをコンパイルして、ゲームコンテンツをレンダリングする (たとえば、オンラインゲームセッションと関連付けられた画像フレームのシーケンスを出力する)。シェーダコンパイルは費用がかかる場合がある。たとえば、ゲームアプリケーションのなかには、

50

各々が数ミリ秒かかるシェーダを100,000以上コンパイルすることを必要とするものもある。シェーダコンパイルがユーザ入力にตอบสนองしてリアルタイムに行われる場合、長い時間がかかり、ユーザ体験が損なわれてしまう。状況によっては、シェーダコンパイルは、コンパイル時間を削減する目的で簡潔にされるが、結果として生じるコンパイルされたシェーダの質は、シェーダコンパイルの時間が制約された性質のために損なわれる。このパラグラフから次の8番目のパラグラフまでの説明は、開示されたもののコピーである。

【0166】

本出願のいくつかの実現例では、GPU140は、サーバシステム114内の計算リソースへの高速パスによって接続されている。コンパイルされたパイプラインのフリーワイドキャッシングが事前に用意され、コンパイラ入力のハッシュによってインデックスを付けられ、ローカルに格納されて、シェーダコンパイル時間を削除または削減する。さらにいくつかの実現例では、手動でまたは遅いがより効果的なシェーダパイプラインコンパイラ（たとえば、図5Aのシェーダパイプラインオプティマイザ534）によってフリーワイド性能分析を行って、より広範囲にわたる最適化のために候補シェーダを識別可能である。いくつかの実現例では、コンパイルされたシェーダはフリーワイドキャッシュ（たとえば、シェーダキャッシュ318）において、より高性能のシェーダ、すなわち他の特徴を有するシェーダにダイナミックに置換えられる。いくつかの実現例では、性能に基づく最適化ステージのために特化されたシェーダと含む異なるコンパイルされたシェーダのセットが、特定の状況での展開のために格納される。いくつかの実現例では、ゲームコンテンツレンダリングで用いられるシェーダは、シェーダパイプラインコンパイラ（たとえば、コンパイラ532）が改良されていると再コンパイルされ、ゲームアプリケーション自体は、これらのシェーダ最適化動作によって影響されない。いくつかの実現例では、ゲームドライバ（図5Aではサーバ側ゲームモジュール526とも呼ばれる）を混乱させることなく、詳細な分析が特徴使用について実現される。いくつかの実現例では、GPU140上で実行されているゲームドライバ（すなわち、サーバ側ゲームモジュール526）を混乱させることなく、特定のゲームタイトルについて、特化されたシェーダがコンパイルされ、検査され、選択される。

【0167】

いくつかの実現例では、ゲームドライバ（具体的には、サーバ側ゲームモジュール526のシェーダパイプラインオプティマイザ534）は、グラフィックスを供給する全てのGPUで実行されるゲームドライバ内のシェーダパイプライン生成/コンパイル呼出しを透過的にインターセプトする。ゲームドライバは、シェーダコンパイラに対する全ての入力のハッシュを評価し、一致するコンパイルされたシェーダについてシェーダパイプラインサーバ（またはシェーダキャッシュ）にリクエストを送信する。シェーダパイプラインサーバまたはキャッシュが一致するシェーダがないと返す場合（またはタイムアウト）、ゲームドライバは、従来のコンパイルパイプラインを使用し、結果をシェーダパイプラインサーバまたはシェーダキャッシュに格納し、さらに、それをオンラインゲームセッションのために用いる。シェーダパイプラインサーバは、多くの態様で実現され得る。シェーダパイプラインサーバは必要であれば、中央の1つのサーバ、または標準分散データベース技術（遅いレイテンシアアクセスについてのレプリケーションを含む）を採用するサーバの分散/連合セットである。

【0168】

いくつかの実現例では、ゲームドライバ（具体的には、サーバ側ゲームモジュール526のシェーダパイプラインオプティマイザ534）は、フリーワイド分析および性能に基づく最適化を行う。フリーワイド分析の例は、時間分析、シェーダ性能分析および消費電力分析を含む。具体的には、いくつかの実現例では、セッションのサンプルセットが、ゲームタイトルと関連付けられた全てのオンラインゲームセッションから選択される。これらのロード（すなわち、サンプリングされたセッション）は、各シェーダパイプラインを実行するために必要な時間の収集および中央データベースへの報告に特化したゲームドライバによって実行される。必要であれば、サンプリングされたセッションは、フォアグ

10

20

30

40

50

ラウンドで同時にオンラインゲームセッションを実行するゲームサーバ118を用いてバックグラウンドで分析される。状況によっては、この計測はゲームアプリケーション（およびそれゆえユーザ体験）を混乱させることがあり、そのため、サンプリングされたセッションは、別のパイプラインオブティマイザサーバ150上で分析される。これによって、使用統計を集めつつ、ユーザ体験は高品質を保つことができる。いくつかの実現例では、シェーダパイプラインオブティマイザ534は、オンラインゲームセッション内で選択的に、たとえば、特定のシェーダまたは画像フレームのサブセットのサンプリングを行って、フォアグラウンドでのゲームコンテンツレンダリングへの影響を低減する。次に、これらのタイミングを用いて、さらなる最適化のための好ましい候補シェーダを選択し、手動レビューまたは自動システムを呼出して、シェーダを精緻化する。代替的にいくつかの実現例では、シェーダパイプライン（すなわち、コンパイルされたシェーダのシーケンス）を最適化するために、消費電力が監視され使用される。具体的には、シェーダパイプラインオブティマイザ534は、タイミング性能ではなくパワードローを計測するように取付けられている。

10

【0169】

詳細なシェーダ分析を含むいくつかの実現例では、シェーダパイプラインオブティマイザ534は、修正されたシェーダおよび粒子の細かいシェーダ実行情報を収集するドライバを選択して、たとえば、各基本ブロックを、補助GPUバッファ内にタグを出すように取付ける。補助GPUバッファはその後、ローカルに分析される、または、我々の（たとえば、データベース160に含まれる）分析データベースに全体がアップロードされる。取付けられたシェーダは、製造ドライバを混乱させることなく、我々のフリートワイドシェーダパイプラインキャッシュ内のアセットからオフラインで生成される。

20

【0170】

シェーダ特殊化のいくつかの実現例では、（多くの場合、高度な訓練を受けた技術者によって手動で）人気のあるゲームタイトル/アプリケーションについてシェーダを特殊化するためにフリークエントオブティマイゼーション（frequent optimization）がシェーダパイプラインオブティマイザ534で適用されて、マシンコンパイルされたコードを手動で調整されたコードで置換える。フリートワイドキャッシュは、フリートワイドキャッシュにおいてコンパイルされたシェーダパイプラインを置換えることによってドライバを再展開することなく、この置換を行うために使用される。いくつかの実現例では、コンパイルされたシェーダの複数のバージョンがフリートワイドキャッシュに格納され、異なる統計分析を行うために展開される。いくつかの実現例では、電力最適化シェーダおよび性能最適化シェーダの両方がシェーダキャッシュに格納されるが、たとえばゲームのピーク消費電力を管理するために、性能基準に従ってダイナミックに展開される。なお、これは多数の性能管理軸、たとえば、電力、時間、メモリ使用量およびコードサイズに及ぶ。さらにいくつかの実現例では、

30

コンパイラアップデートおよびプロファイリングを含むいくつかの実現例では、シェーダパイプラインオブティマイザ534は、シェーダコンパイラの新しいバージョンをシェーダキャッシュ内の全てのシェーダに対して実行し、シェーダコンパイラの前のバージョンと一緒にシェーダのこのシェーダコンパイラの新しいバージョンを格納する。オブティマイザ534はその後、オンラインゲームセッションのサンプルを用いて新しいシェーダコンパイラを実行し、性能影響を計測し、新しいシェーダコンパイラの適合性を判断する。これらの手段によって、コンパイルされたシェーダとシェーダコンパイラとの両方が、ゲームモジュールのゲームコンテンツレンダリングを混乱させることなく、バックグラウンドでダイナミックに改良される。

40

【0171】

いくつかの実現例では、サーバシステム114は、内部ユーザおよび外部ユーザについて分散シェーダパイプラインキャッシュに問い合わせるためにアプリケーションプログラミングインターフェース（API）を備える。分散シェーダパイプラインキャッシュは、所与の入力のセットについて問い合わせられて、さらなる検査および分析が可能になる。たとえば

50

、マッシュリデューススタイルクエリが、個々のシェーダおよびシェーダパイプラインのフリートワイド分析について分散シェーダパイプラインキャッシュ上で実行される。このAPIは、デブスオンリーパスについて最も使用される命令のブレンドであり、D r e m e l などに対するエクスポートパイプラインとして実現され得る。さらに、APIは、インフラストラクチャとしてのパイプライン - シェーダ - コンパイラを活用することによって、コンパイラエクスポートインターフェースを提供するように構成される。

【0172】

図10は、いくつかの実現例に係る、サーバシステム114においてゲームコンテンツレンダリングと同時にパイプラインされたシェーダを最適化するプロセス1000の例を示す図である。いくつかの実現例において図5Aに示すように、サーバ側ゲームモジュール526は、同時ゲームコンテンツレンダリングおよびシェーダパイプライン最適化を可能にするように協働して機能するゲームアプリケーション528と、シェーダパイプライン最適化534と、ゲームレンダリングモジュール538と、シェーダキャッシュ318とを含む。具体的には、ゲームアプリケーション528がフォアグラウンドでグラフィックスパイプラインプロセスにおいてオンラインゲームセッションを実行する一方で、シェーダパイプライン最適化534は、バックグラウンドでこのグラフィックスパイプラインプロセスにおいて使用されるコンパイルされたシェーダを最適化する。

【0173】

サーバシステム114のCPU138は、ゲームタイトルに対応するゲームアプリケーション528を実行して、複数のオンラインゲームセッションを並行して実行する。CPU138は、GPU140のサブセットの各々を割当てて、複数のオンラインゲームセッション（たとえば、1つ以上のセッション）のサブセットを実行する。GPU140はその後、コンパイルされたシェーダの第1のシーケンスを識別し、コンパイルされたシェーダの識別された第1のシーケンスを用いて、オンラインゲームセッションの各々について複数の画像フレームをレンダリングする。いくつかの実現例では、コンパイルされたシェーダの第1のシーケンスがシェーダキャッシュ318に格納されており、一度識別されており、直接シェーダキャッシュからGPU140にロードされて、ゲームアプリケーション528の命令に従って、オンラインゲームセッションのために画像フレームをレンダリングする。いくつかの実現例では、複数のオンラインゲームセッションの実行は特定の性能基準に従う必要があり、コンパイルされたシェーダの第1のシーケンスは、オンラインゲームセッションの各々について画像フレームをレンダリングするために使用される前に、特定の性能基準に従って識別される。

【0174】

GPUが複数のオンラインゲームセッションを実行する一方で、シェーダパイプライン最適化534は、バックグラウンドでコンパイルされたシェーダの第1のシーケンスを最適化する。別の言い方をすれば、CPU138は、サーバシステム114内のCPUおよびGPUの未使用のサブセットを割当てて、1つ以上の性能基準に関して、コンパイルされたシェーダの代替的なシーケンスまたはコンパイルされたシェーダの第1のシーケンスの他のバージョンを識別する。必要であれば、ゲームサーバ118はまた、シェーダパイプライン最適化について割当てられたCPUおよびGPUの未使用のサブセットと、オンラインゲームセッションを実行するアクティブなCPUおよびGPUとの両方を含む。必要であれば、ゲームサーバ118とは異なるパイプライン最適化サーバ150は、シェーダパイプライン最適化について割当てられたCPUとGPUとのサブセットを含む。

【0175】

具体的には、ゲームアプリケーション528が複数のオンラインゲームセッションを実行する一方で、シェーダパイプライン最適化534は、ゲームタイトルと関連付けられた複数のオンラインゲームセッションについて性能基準を識別し、コンパイルされたシェーダの第1のシーケンスを用いて、オンラインゲームセッションの各々の実行を特徴付ける使用統計を収集する。シェーダパイプライン最適化534は、使用統計を

10

20

30

40

50

分析し、コンパイルされたシェーダの第1のシーケンスの性能が性能基準に関してさらに改良可能であると判断する。たとえば、シェーダパイプラインオブティマイザ534は、コンパイルされたシェーダの第1のシーケンスの第1のシェーダの修正がコンパイルされたシェーダの第1のシーケンスの性能を性能基準に関して改良すると判断し、そのため、第1のシェーダを修正して、コンパイルされたシェーダの修正された第1のシーケンスを生成する。いくつかの実現例では、シェーダパイプラインオブティマイザ534はさらに、コンパイルされたシェーダの第1のシーケンス内の1つ以上の第2のシェーダを修正して、性能基準に関してコンパイルされたシェーダの第1のシーケンスの性能を改良する。

【0176】

いくつかの実現例では、シェーダパイプラインオブティマイザ534は、コンパイルされたシェーダの第1のシーケンスを性能基準に関して最適化するように構成された自動オブティマイザ1002を含む。そうは言っても、自動オブティマイザ1002は、シェーダキャッシュからコンパイルされたシェーダの第1のシーケンスを取得し、コンパイルされたシェーダの第1のシーケンスの性能を性能基準に関して改良するために、自動的にユーザの介入なしで、コンパイルされたシェーダの第1のシーケンス内の第1のシェーダを修正する。代替的にいくつかの実現例では、シェーダパイプラインオブティマイザ534は、ユーザにコンパイルされたシェーダの第1のシーケンスを性能基準に関して手動で最適化させるユーザインターフェースを提供するように構成された手動オブティマイザ1004を含む。管理者ユーザは、管理者クライアントデバイス102上で実行されているゲーム開発プラットフォーム上で管理者アカウントにログオンする。ユーザインターフェースは、性能基準、使用統計および他の関連情報項目を提示して、手動最適化を容易にする。サーバシステム114は、コンパイルされたシェーダの第1のシーケンス内の第1のシェーダを修正する命令を受信し、そのため、第1のシェーダは、ユーザ命令に応答して、コンパイルされたシェーダの第1のシーケンス内で修正される。さらにいくつかの実現例では、パイプラインシェーダオブティマイザ534は、自動オブティマイザ1002および手動オブティマイザ1004の両方を含み、自動最適化、手動最適化、またはこれらの組合せをサポートする。

【0177】

さらにいくつかの実現例では、コンパイルされたシェーダの第1のシーケンス内の第1のシェーダを置換えるために、代替的なシェーダオプションのグループから代替的なシェーダが選択される。必要であれば、自動オブティマイザ1002は、コンパイルされたシェーダの第1のシーケンスの性能を性能基準に関して改良するために、自動的に代替的なシェーダオプションを識別し、代替的なシェーダを選択する。必要であれば、自動オブティマイザ1002は、代替的なシェーダオプションを自動的に識別するが、管理者ユーザは、手動オブティマイザ1004によって代替的なシェーダを選択するように促される。必要であれば、手動オブティマイザ1004は、関連情報項目を提供して、管理者ユーザに代替的なシェーダオプションの識別および代替的なシェーダの選択を促す最適化プロセスを容易にする。

【0178】

図10を参照すると、いくつかの実現例では、コンパイルされたシェーダの修正された第1のシーケンスは、シェーダキャッシュ内のコンパイルされたシェーダの第1のシーケンスが生成された後にそれに置換わる。代替的にいくつかの実現例では、コンパイルされたシェーダの第1のシーケンスおよびコンパイルされたシェーダの修正された第1のシーケンスは、コンパイルされたシェーダの第1のシーケンスの2つのバージョンとして格納される。続いて、サーバシステム114は、オンラインゲームセッションの各々の実行を特徴付ける次の使用統計を収集し、次の使用統計に従って、GPUは、コンパイルされたシェーダの第1のシーケンスの2つのバージョンのうち1つを選択して、複数のオンラインゲームセッションの各々について複数の画像フレームのレンダリングを行い得る。

【0179】

代替的にいくつかの実現例では、性能基準は第1の性能基準を含み、複数のオンラインゲ

10

20

30

40

50

ームセッションは、第1の基準とは別の第2の性能基準に従ってレンダリングおよび実行される。パイプラインオブティマイザ534は、シェーダキャッシュ318が、第1の性能基準をコンパイルされたシェーダの修正された第1のシーケンスと、および第2の性能基準をコンパイルされたシェーダの第1のシーケンスと関連付けることを可能にする。

【0180】

いくつかの実現例では、シェーダパイプラインオブティマイザ534は、バックグラウンドで2つ以上の性能基準について最適化されたシェーダパイプラインを取得する。性能基準は第1の性能基準を含み、コンパイルされたシェーダの修正された第1のシーケンスは、シェーダキャッシュ内の第1の性能基準と関連して記憶される。コンパイルされたシェーダの修正された第1のシーケンスを識別した後で、シェーダパイプラインオブティマイザ534は、コンパイルされたシェーダの第1のシーケンスの性能を第2の性能基準に関連して改良するために、ゲームタイトルと関連付けられた複数のオンラインゲームセッションについて第2の性能基準を識別し、コンパイルされたシェーダの第2のシーケンスを判断する。コンパイルされたシェーダの第2のシーケンスは、第2の性能基準に関連してシェーダキャッシュ318に格納される。

10

【0181】

シェーダキャッシュ318がコンパイルされたシェーダの複数のシーケンスを含む場合、複数の性能基準の各々をコンパイルされたシェーダのそれぞれのシーケンスと関連付けるシェーダハッシュテーブル1006を確立し、これを格納する(図11B)。性能基準およびコンパイルされたシェーダの第1のシーケンスは、シェーダハッシュテーブル内で関連付けられている。必要であれば、コンパイルされたシェーダの第1のシーケンスは、複数のバージョンを有し得る。使用統計はユーザ入力情報を含み、第1のシーケンスの複数のバージョンが、ユーザ入力の異なる情報と関連付けられる(図11C)。たとえば、最も人気のあるユーザ入力第1のユーザ入力(たとえば、ジャンプ)である場合、コンパイルされたシェーダの第1のシーケンスの第1のバージョンが、対応する性能基準を満たすために使用され、最も人気のあるユーザ入力第2のユーザ入力(たとえば、シューティング)である場合、コンパイルされたシェーダの第1のシーケンスの第2のバージョンが、対応する性能基準を満たすために使用される。さらにいくつかの実現例では、シェーダハッシュテーブルは、使用統計に従ってダイナミックにアップデートされる。

20

【0182】

なお、性能基準は、複数のオンラインゲームセッションを実行する過程において、コンパイルされたシェーダの第1のシーケンスの全消費電力、コードサイズ、平均レイテンシ、全キャッシュ使用量、および全メモリ使用量のうち1つ以上に対応する。たとえば、性能基準は、コードサイズが閾値サイズよりも低いことを必要とする。他の例では、性能基準は、使用統計を考慮して、消費電力が閾値電力レベルまで低減されることを必要とする。

30

【0183】

いくつかの実現例では、使用統計は、複数のオンラインゲームセッションの各々に対応するコンパイルされたシェーダの第1のシーケンスのユーザ入力、消費電力、1つ以上のゲームレイテンシ、コードサイズ、メモリ使用量およびキャッシュ使用量を含む。特定の例では、使用統計は、複数のオンラインゲームセッションの全消費電力を含む。第1の性能基準および第2の性能基準はそれぞれ、複数のオンラインゲームセッションの応答時間および電力使用量に対応する。複数のオンラインゲームセッションの全消費電力が事前に定義された電力閾値を超えており、低減する必要があるという判断に従って、ゲームアプリケーション528は、コンパイルされたシェーダの第2のシーケンスを選択し、コンパイルされたシェーダの第2のシーケンスを用いて複数のゲームセッションの各々について複数の画像フレームのレンダリングを行って、第2の性能基準に従って複数のオンラインゲームセッションの全消費電力を改良する。

40

【0184】

図11Aは、いくつかの実現例に係る、パイプラインされたシェーダを最適化するプロセスの例を示す図である。ゲームタイトルに関連する複数のオンラインゲームセッションは

50

、OpenGL レンダリングパイプラインを用いて実行される。OpenGL レンダリングパイプラインは少なくとも、頂点処理（たとえば、スクリーン空間内への頂点の変形）、プリミティブ処理（たとえば、クリップされ間引かれたプリミティブへの頂点の編成）、ラスタ化（たとえば、プリミティブのピクセルフラグメントへのラスタ化）、フラグメント処理（たとえば、各ピクセルにおいて色を計算するためのフラグメントのシェーディング）、およびピクセル処理（たとえば、フラグメントのそれらのピクセル位置におけるフレームバッファへのブレンディング）を含む処理のサブセットを含む。特定の例では、OpenGL レンダリングパイプラインは、オンラインゲームセッションの各々について画像フレームをレンダリングするために9つの連続したパイプライン処理またはステージを含む。これらの9つの連続したパイプラインステージは、以下のサブセットまたは全てを含む。

10

【0185】

- 頂点アレイデータを準備するための頂点仕様1102。
- 各頂点に作用するための頂点シェーダ1104。

【0186】

- 必要に応じて頂点データのパッチをより小さなプリミティブに細分するためのテッセレーション1106。

【0187】

- 必要に応じてより小さなプリミティブを出力プリミティブに変換するための幾何学シェーダ1108。

20

【0188】

- 変形フィードバック、プリミティブクリッピング、透視除算、ビューポート変換を含むがこれらに限定されない、最新のステージの出力プリミティブの調整および異なる位置への発送のための頂点後処理1110。

【0189】

- 出力プリミティブを個々の基本プリミティブのシーケンスに分割するためのプリミティブアセンブリ1112。

【0190】

- プリミティブのサンプルカバレッジに基づいて個々のプリミティブを個別の要素（フラグメントとも呼ばれる）に分解するためのラスタ化1114。

30

【0191】

- ラスタ化によって生成された各フラグメントを色のセットおよび1つの深度値に処理するためのフラグメントシェーダ1116。

【0192】

- エンコーダ142によって処理可能なゲームブレイクマルチメディアコンテンツに色のセットおよび1つの深度値を処理するための、たとえば、ピクセルオーナーシップテスト、シザーテスト、マルチサンプル処理、ステンシルテスト、深度テスト、オクルージョンクエリアップデート、ブレンディング、sRGB変換、ディザリング、論理演算、および書込みマスク論理演算を含むがこれらに限定されない、標準化、速度および/または圧縮のためのプリサンプル処理1118。

40

【0193】

オンラインゲームセッションのサブセットを実行するように割当てられた各GPU140は、オンラインゲームセッションのサブセットの各セッションについて、9つの連続したパイプラインステージを実行する。いくつかの実現例では、頂点シェーダ1104、テッセレーション1106、幾何学シェーダ1108およびフラグメントシェーダ1116の各々は、プログラム可能なシェーダのそれぞれのセットを含む。プログラム可能なシェーダのそれぞれのセット内のシェーダについて、シェーダパイプラインオブティマイザ534は、それぞれのシェーダを修正するオプションを有しており、これによって、シェーダパイプラインの修正されたシーケンスを生成する。

【0194】

50

より具体的には、一例では、ゲームアプリケーション 5 2 8 は複数のゲームセッションを実行し、コンパイルされたシェーダの第 1 のシーケンスを用いて複数のオンラインゲームセッションの各々について、複数の画像フレームをレンダリングする。コンパイルされたシェーダの第 1 のシーケンスは、プログラム可能なシェーダである頂点シェーダ 1 1 0 4 A、テッセレーションシェーダ 1 1 0 6 A、幾何学シェーダ 1 1 0 8 およびフラグメントシェーダ 1 1 1 6 A を含む。複数のオンラインゲームセッションがフォアグラウンドで実行される一方で、シェーダパイプラインオブティマイザ 5 3 4 は、ゲームタイトルと関連付けられた複数のオンラインゲームセッションについて性能基準を識別し、オンラインゲームセッションの各々を特徴付ける使用統計を収集する。その結果、シェーダパイプラインオブティマイザ 5 3 4 は、コンパイルされたシェーダの第 1 のシーケンスの性能を性能基準に関して改良するために、コンパイルされたシェーダの第 1 のシーケンス内の頂点シェーダ 1 1 0 4 A およびテッセレーションシェーダ 1 1 0 6 A を修正すると判断して、コンパイルされたシェーダの修正された第 1 のシーケンスを生成する。

10

【 0 1 9 5 】

具体的には、シェーダパイプラインオブティマイザ 5 3 4 は、コンパイルされたシェーダの修正された第 1 のシーケンス内の幾何学シェーダ 1 1 0 8 A およびフラグメントシェーダ 1 1 1 6 A を維持しつつ、頂点シェーダ 1 1 0 4 A およびテッセレーションシェーダ 1 1 0 6 A を頂点シェーダ 1 1 0 4 N およびテッセレーションシェーダ 1 1 0 6 B と置換える。コンパイルされたシェーダの修正された第 1 のシーケンスはその後、性能基準に関連してシェーダキャッシュに格納される。

20

【 0 1 9 6 】

図 1 1 B は、いくつかの実現例に係る、コンパイルされたシェーダの複数のシーケンスを複数の性能基準と関連付けるシェーダハッシュテーブル 1 0 0 6 を示す図である。図 1 1 C は、いくつかの実現例に係る、コンパイルされたシェーダのシーケンスの複数のバージョンを複数のユーザ入力と関連付けるシェーダハッシュテーブル 1 1 8 0 を示す図である。シェーダキャッシュ 3 1 8 がコンパイルされたシェーダの複数のシーケンスを含む場合、複数の性能基準の各々をコンパイルされたシェーダのそれぞれのシーケンスと関連付けるシェーダハッシュテーブル 1 0 0 6 を確立し、これを格納する。たとえば、性能基準（性能基準 1）とコンパイルされたシェーダの第 1 のシーケンスとは、シェーダハッシュテーブルにおいて関連付けられる。必要であれば、コンパイルされたシェーダの第 1 のシーケンスは、異なる仕様統計に対応する複数のバージョンを有し得る。いくつかの実現例では、使用統計はユーザ入力情報を含み、第 1 のシーケンスの複数のバージョンがユーザ入力の異なる情報と関連付けられる。たとえば、最も人気のあるユーザ入力（たとえば、ジャンプ）である場合、コンパイルされたシェーダの第 1 のシーケンスの第 1 のバージョンが、対応する性能基準を満たすために使用され、最も人気のあるユーザ入力（たとえば、シューティング）である場合、コンパイルされたシェーダの第 1 のシーケンスの第 2 のバージョンが、対応する性能基準を満たすために使用される。さらにいくつかの実現例では、シェーダハッシュテーブル 1 0 5 0 は、使用統計に従ってダイナミックにアップデートされる。

30

【 0 1 9 7 】

図 1 2 は、いくつかの実現例に係る、（たとえば、画像レンダリングのバックグラウンドでのパイプラインされたシェーダの最適化を含む）複数のオンラインゲームセッションについて画像フレームをレンダリングする方法 1 2 0 0 の例を示すフローチャートである。方法 1 3 0 0 は必要であれば、コンピュータメモリまたは非一時的なコンピュータ可読記憶媒体（たとえば、図 5 A のシェーダパイプラインオブティマイザ 5 3 4）に格納され、かつ、サーバシステム 1 1 4 の 1 つ以上のプロセッサによって実行される命令によって制御される。コンピュータ可読記憶媒体は、磁気もしくは光学ディスク記憶装置、フラッシュメモリなどの固体記憶装置、または他の不揮発性メモリデバイスもしくはデバイスを有し得る。コンピュータ可読記憶媒体に格納される命令は、ソースコード、アセンブリ言語コード、オブジェクトコード、または 1 つ以上のプロセッサによって解釈される他の命令

40

50

フォーマットのうち1つ以上を含み得る。方法1200におけるいくつかの動作は組合わされてもよい、および/または、いくつかの動作の順番は変更されてもよい。

【0198】

方法1200は、サーバシステム114において実現される。サーバシステム114は、1つ以上のCPU138と、1つ以上のGPU140と、プログラムの実行中にCPUおよび/またはGPUによる使用のためのプログラムおよびデータを格納するメインダイナミックメモリ144と、シェーダキャッシュ318とを備える。1つ以上のCPUは、ゲームタイトルについて複数のオンラインゲームセッションを並行して実行する(1202)。1つ以上のCPUは、GPU140に、ユーザリクエストまたは入力に応答して複数のオンラインゲームセッションの各々について連続する画像フレームをレンダリングするように命令する。1つ以上のGPUはその後、コンパイルされたシェーダの第1のシーケンスを用いて、複数のオンラインゲームセッションの各々について複数の画像をレンダリングする(1204)。いくつかの実現例では、複数のオンラインゲームセッションは、ゲームタイトルについてCPUおよびGPUによって処理される全てのオンラインゲームセッションからサンプリングされる。

10

【0199】

サーバシステム114は、複数のオンラインゲームセッションを実行しつつ、ゲームタイトルと関連付けられた複数のオンラインゲームセッションについて性能基準を識別し(1206)、オンラインゲームセッションの各々の実行を特徴付ける使用統計を収集する(1208)。いくつかの実現例では、性能基準は、複数のオンラインゲームセッションの実行の過程におけるコンパイルされたシェーダの第1のシーケンスの全消費電力、コードサイズ、平均レイテンシ、全キャッシュ使用量、および全メモリ使用量のうち1つ以上に対応する。いくつかの実現例では、使用統計は、複数のオンラインゲームセッションの各々に対応するコンパイルされたシェーダの第1のシーケンスのユーザ入力、消費電力、1つ以上のゲームレイテンシ、コードサイズ、メモリ使用量およびキャッシュ使用量を含む。

20

【0200】

サーバシステム114は、コンパイルされたシェーダの第1のシーケンスの性能を性能基準に関して改良するためにコンパイルされたシェーダの第1のシーケンス内の第1のシェーダを修正して(1210)、コンパイルされたシェーダの修正された第1のシーケンスを生成する。いくつかの実現例では、コンパイルされたシェーダの第1のシーケンスの性能を性能基準に関して改良するために、コンパイルされたシェーダの第1のシーケンス内の第2のシェーダが、第1のシェーダに加えて修正される。

30

【0201】

いくつかの実現例では、コンパイルされたシェーダの修正された第1のシーケンスは、シェーダキャッシュ内のコンパイルされたシェーダの第1のシーケンスに置換わる。代替的にいくつかの実現例では、オンラインゲームセッションの実行で使用されるコンパイルされたシェーダの第1のシーケンスは、コンパイルされたシェーダの第1のシーケンスの第1のバージョンとしてシェーダキャッシュに格納され、コンパイルされたシェーダの修正された第1のシーケンスは、コンパイルされたシェーダの第1のシーケンスの第2のバージョンとしてシェーダキャッシュに格納される。次のゲームコンテンツレンダリング中に、たとえば次のユーザ入力に応答して、サーバシステム114は、オンラインゲームセッションの各々の次の実行について特定の性能基準を識別し、特定の性能基準に従って、コンパイルされたシェーダの第1のシーケンスの第1および第2のバージョンを選択して、複数のオンラインゲームセッションの各々について複数の画像フレームをレンダリングする。

40

【0202】

いくつかの実現例では、コンパイルされたシェーダの第1のシーケンス内の第1のシェーダを修正するために、サーバシステム114は、複数のオンラインゲームセッションの性能が性能基準を満たすように代替的なシェーダオプションのグループから代替的なシェーダを選択して、コンパイルされたシェーダの第1のシーケンス内の第1のシェーダに置換

50

える。その結果、コンパイルされたシェーダの修正された第 1 のシーケンスは、代替的なシェーダを含む。なお、いくつかの実現例では、代替的なシェーダオプションのグループは、シェーダライブラリサーバ 1 2 4 から提供される。状況によっては、代替的なシェーダオプションのグループは、サーバシステム 1 1 4 上で実行されるゲームタイトルを管理する管理者ユーザに提示され、サーバシステム 1 1 4 は、管理者ユーザがログオンしたクライアントデバイス 1 0 2 を介して、管理者ユーザから代替的なシェーダのユーザ選択を受信する。

【 0 2 0 3 】

いくつかの実現例では、管理者ユーザは、管理者クライアントデバイス 1 0 2 上で実行されるゲームアプリケーション上の管理者アカウントにログオンし、性能基準および使用統計に基づいて、コンパイルされたシェーダの第 1 のシーケンスを手動で最適化する。この最適化プロセスの間、サーバシステム 1 1 4 は、コンパイルされたシェーダの第 1 のシーケンス内の第 1 のシェーダを修正するユーザ命令を受信し、そのため、第 1 のシェーダは、ユーザ命令に応答して、コンパイルされたシェーダの第 1 のシーケンスで修正される。逆に、いくつかの実現例では、コンパイルされたシェーダの第 1 のシーケンス内のシェーダは、コンパイルされたシェーダの第 1 のシーケンスの性能を性能基準に関して改良するために、自動的にユーザの介入なしで修正される。

【 0 2 0 4 】

いくつかの実現例では、GPU がコンパイルされたシェーダの第 1 のシーケンスを用いて複数のオンラインゲームセッションのために画像フレームをレンダリングする前に、サーバシステム 1 1 4 が、複数のオンラインゲームセッションが性能基準に従ってレンダリングされる必要があると判断する。性能基準を考慮して、コンパイルされたシェーダの第 1 のシーケンスは、複数のオンラインゲームセッションの各々について複数の画像フレームをレンダリングするために判断される。

【 0 2 0 5 】

いくつかの実現例では、性能基準は第 1 の性能基準を含み、複数のオンラインゲームセッションは、第 1 の基準とは別の第 2 の性能基準に従ってレンダリングおよび実行される。サーバシステム 1 1 4 は、第 1 の性能基準をコンパイルされたシェーダの修正された第 1 のシーケンスと関連付け、第 2 の性能基準をコンパイルされたシェーダの第 1 のシーケンスと関連付ける。

【 0 2 0 6 】

いくつかの実現例では、性能基準は第 1 の性能基準を含み、コンパイルされたシェーダの修正された第 1 のシーケンスは、シェーダキャッシュに第 1 の性能基準と関連して格納される。さらにいくつかの実現例では、サーバシステム 1 1 4 は複数のオンラインゲームセッションを実行しつつ、コンパイルされたシェーダの第 1 のシーケンスの性能を第 2 の性能基準に関して改良するために、ゲームタイトルと関連付けられた複数のオンラインゲームセッションについて第 2 の性能基準を識別し、コンパイルされたシェーダの第 2 のシーケンスを判断する。コンパイルされたシェーダの第 2 のシーケンスは、第 2 の性能基準に関連してシェーダキャッシュに格納される。さらにいくつかの実現例では、使用統計は、複数のオンラインゲームセッションの全消費電量を含み、第 1 の性能基準および第 2 の性能基準はそれぞれ、複数のオンラインゲームセッションの応答時間および電力使用量を最適化する。複数のオンラインゲームセッションの全消費電力が事前に定義された電力閾値を超えているという判断に従って、サーバシステム 1 1 4 は、コンパイルされたシェーダの第 2 のシーケンスを選択し、コンパイルされたシェーダの第 2 のシーケンスを用いて、複数のオンラインゲームセッションの各々について複数の画像フレームをレンダリングする。

【 0 2 0 7 】

いくつかの実現例では、サーバシステム 1 1 4 は、複数の性能基準の各々をコンパイルされたシェーダのそれぞれのシーケンスと関連付けるシェーダハッシュテーブルを確立する。シェーダハッシュに従って、性能基準とコンパイルされたシェーダの第 1 のシーケンス

10

20

30

40

50

とがシェーダハッシュテーブルで関連付けられる。シェーダハッシュテーブルは、シェーダキャッシュに格納される。さらにいくつかの実現例では、使用統計はユーザ入力情報を含む。コンパイルされたシェーダの第1のシーケンスの複数のバージョンが、シェーダハッシュテーブルにおいてユーザ入力情報と関連付けられる。さらにいくつかの実現例では、サーバシステム114は、たとえばシェーダキャッシュ内のコンパイルされたシェーダの第1のシーケンスをコンパイルされたシェーダの修正された第1のシーケンスと置換えることによって、使用統計に従ってシェーダハッシュテーブルをダイナミックにアップデートする。

【0208】

なお、シェーダ最適化は、CPUおよびGPUを有する1つのサーバだけでなく、複数のサーバ（すなわち、複数のCPUおよびGPUを有する）を含むサーバシステムにも適用可能である。ゲームコンテンツレンダリングで使用されるコンパイルされたシェーダのシーケンスの最適化については、図10および図11A～図11Cを参照してより詳細に説明されている。

【0209】

なお、図12の動作が説明された特定の順序は例示に過ぎず、説明された順序は動作を実行可能な唯一の順序を示すと意図されたものではない。当業者であれば、本明細書で説明されたコンパイルされたシェーダのシーケンスを最適化するさまざまな方法が分かるであろう。さらに、方法800および900（たとえば、図8A～図8Bおよび900）に関して本明細書で説明された他のプロセスの詳細も、図12に関して説明された方法1200に類似の態様で適用可能である。簡潔に説明するために、これらの詳細はここでは繰り返さない。

【0210】

本出願のいくつかの実現例は、1つ以上のCPUと、複数のGPUと、プログラムの実行中にCPUおよび/またはGPUによる使用のためのプログラムおよびデータを格納するメインダイナミックメモリと、不揮発性メモリに格納されたスタティックメモリプールと、スタティックメモリプールを管理するように構成されたメモリコントローラとを備えるサーバシステムに向けられている。GPUの各々はローカルキャッシュを含み、メモリコントローラを介してスタティックメモリプールにアクセスするように構成される。サーバシステムは、1つ以上のCPU上でゲームタイトルについて複数のゲームセッションを並行して実行する。複数のゲームセッションの各々は、スタティックデータ項目と関連付けられ、スタティックデータ項目を用いてそれぞれのGPUによって実行可能なグラフィックス処理を必要とし、スタティックデータ項目は、スタティックメモリプールに格納される。1つ以上のCPU上で実行されるゲームセッションのサブセットの各々について、サーバシステムは、グラフィックス処理をそれぞれのGPUに割当て、それぞれのGPUによって、スタティックデータ項目のコピーをメモリコントローラにリクエストし、スタティックメモリプールから読出されたスタティックデータ項目の第1をメモリコントローラから受信し、それぞれのGPUによって、スタティックデータ項目のコピーを用いてグラフィックス処理を実行し、それぞれのゲームセッションに関連してグラフィックス処理の結果をローカルに格納する。

【0211】

要約すると、本出願のいくつかの実現例は、1つ以上のCPUと、複数のGPUと、プログラムの実行中にCPUおよび/またはGPUによる使用のためのプログラムおよびデータを格納するメインダイナミックメモリと、不揮発性メモリに格納されたスタティックメモリプールと、スタティックメモリプールとを管理するように構成されたメモリコントローラとを備えるサーバシステム上でゲームタイトルをロードすることに向けられている。CPUは、ゲームタイトルに対応する実行可能なゲームプログラムをメインダイナミックメモリにロードする。CPUは、複数のオンラインゲームセッションを並行して実行するように構成される。ゲームセッションの各々は実行可能なゲームプログラムに対応し、それぞれのオンラインゲームセッションをレンダリングするように割当てられたそれぞれの

10

20

30

40

50

G P Uによって実行可能なグラフィックス処理を必要とする。C P Uは、複数のゲームセッションのグラフィックス処理で使用されるスタティックデータ項目を識別する。メモリコントローラは、スタティックデータ項目をスタティックメモリプールに格納し、オンラインゲームセッションを実行しているG P Uがグラフィックス処理を実行しつつスタティックデータ項目にアクセスすることを可能にする。

【 0 2 1 2 】

本出願のいくつかの実現例は、ゲームタイトルのための同時画像レンダリングおよびシェーダ最適化に向けられている。サーバシステムは、1つ以上のC P Uと、複数のG P Uと、プログラムの実行中にC P Uおよび/またはG P Uによる使用のためのプログラムおよびデータを格納するメインダイナミックメモリと、シェーダキャッシュとを備える。C P Uは、ゲームタイトルについて複数のオンラインゲームセッションを並行して実行する。G P Uは、コンパイルされたシェーダの第1のシーケンスを用いて、複数のオンラインゲームセッションの各々について複数の画像フレームをレンダリングする。サーバシステムは、複数のオンラインゲームセッションを実行しつつ、コンパイルされたシェーダの第1のシーケンスの性能を性能基準に関して改良するために、ゲームタイトルと関連付けられた複数のオンラインゲームセッションについて性能基準を識別し、オンラインゲームセッションの各々の実行を特徴付ける使用統計を収集し、コンパイルされたシェーダの第1のシーケンス内の第1のシェーダを修正して、コンパイルされたシェーダの修正された第1のシーケンスを生成する。

【 0 2 1 3 】

さまざまな実現例を詳細に参照し、それらの例を添付の図面に示した。上記詳細な説明において、本発明および記載の実現例の十分な理解のため、多数の具体的な詳細を説明した。しかしながら、これらの具体的な詳細がなくても、本発明を実行し得る。その他の場合、周知の方法、プロシージャ、コンポーネント、および回路は、実現例の態様を不必要にあいまいにしないように、詳細に説明しなかった。

【 0 2 1 4 】

さまざまな構成要素を説明するために、「第1の」、「第2の」といった用語が本明細書において使用される場合があるが、これらの構成要素は、これらの用語によって限定されるべきではないと理解されるだろう。これらの用語は、ある構成要素を別の構成要素と区別するために使用されるにすぎない。たとえば、第1のデバイスの名称がすべて矛盾なく変更され、第2のデバイスの名称がすべて矛盾なく変更される場合に限り、説明の意味を変更することなく、第1のデバイスを第2のデバイスと称することができ、同様に、第2のデバイスを第1のデバイスと称することができる。第1のデバイスおよび第2のデバイスは両方ともデバイスであるが、同じデバイスではない。

【 0 2 1 5 】

本明細書で使用される用語は、特定の實現例を説明する目的のためだけのものであり、特許請求の範囲を制限すると意図されたものではない。實現例および添付の請求の範囲の説明において使用されるように、単数形「a」、「an」、および「the」は、文脈が明らかに他のものを示さない限り、複数形も含むことを意図する。本明細書において使用される用語「および/または (and/or)」は、関連する記載された項目のうちの1つ以上のいずれか、およびすべての考えられる組み合わせを指し、それらを包含すると理解されるだろう。用語「備える/含む (comprises)」および/または「備える/含む (comprising)」は、本明細書において使用されるとき、記載の特徴、整数、ステップ、動作、構成要素、および/またはコンポーネントの存在を具体的に挙げるが、1つ以上のその他の特徴、整数、ステップ、動作、構成要素、コンポーネント、および/またはそれらのグループの存在もしくは追加を排除しないと理解されるだろう。

【 0 2 1 6 】

本明細書で使用するとき、「～である場合 (if)」という用語は、文脈に応答して、記載の先行条件が真である「ときに (when)」、「すると (upon)」、「と判定することに応答して (in response to determination)」、「という判定に従って (in accordan

10

20

30

40

50

ce with a determination)」、または「と検出することに対応して (in response to detecting)」を意味すると解釈され得る。同様に、「記載の先行条件が真であると判定された場合 (if it is determined [that a stated condition precedent is true])」、「記載の先行条件が真である場合 (if [a stated condition precedent is true])」、「記載の先行条件が真であるとき (when [a stated condition precedent is true])」という句は、文脈に応じて、記載の先行条件が真である「と判定すると (upon determining)」、「と判定することに対応して (in response to determining)」、「という判定に対応して (in accordance with a determination)」、「と検出すると (upon detecting)」、または「と検出することに対応して (in response to detecting)」を意味すると解釈され得る。

10

【 0 2 1 7 】

具体的な実現例を参照して、説明する目的で以上の説明を記載した。しかしながら、例示的な上記の議論は、網羅的であることまたは開示される正確な形態に請求項の範囲を限定することを意図しない。上記教示に鑑みて多数の修正例および変更例が可能である。実現例は、請求項の基礎をなす原則およびそれらの実践的な適用例を最良に説明し、それにより、当業者が、考察された特定の使用に適するようなさまざまな修正例とともに本発明およびさまざまな実現例を最良に用いることができるようにするために、選択され記載された。

20

30

40

50

【図面】
【図 1】

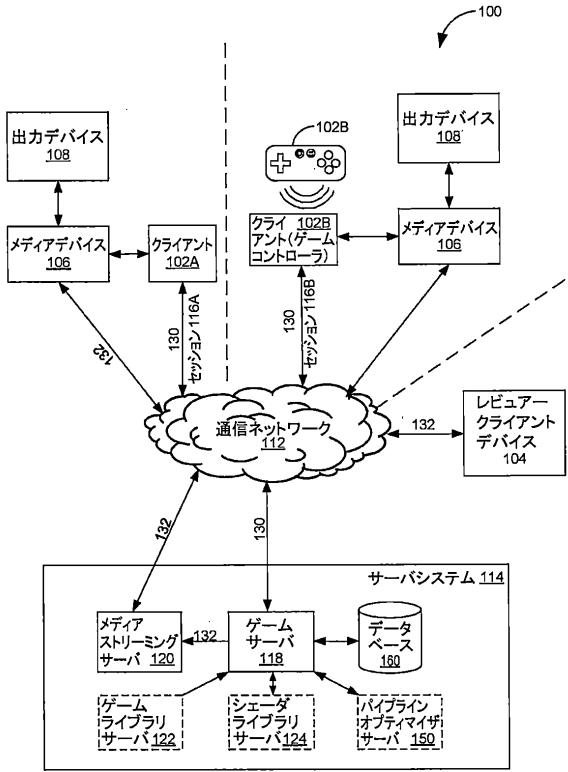


Figure 1

【図 2】

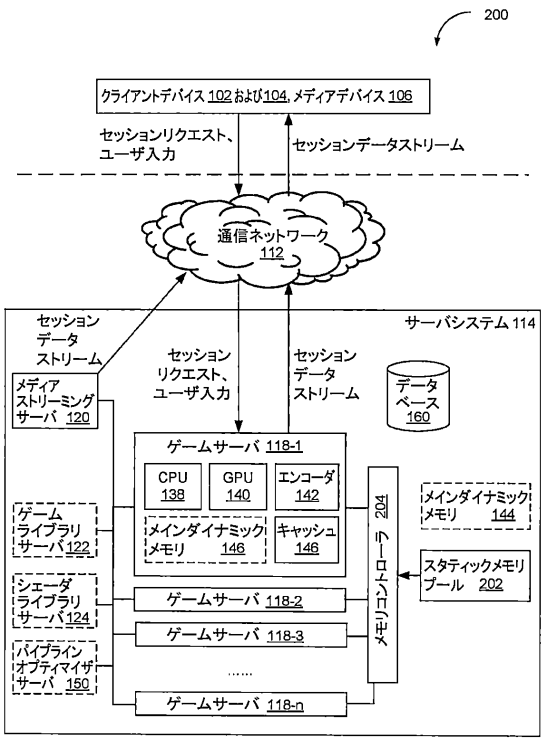
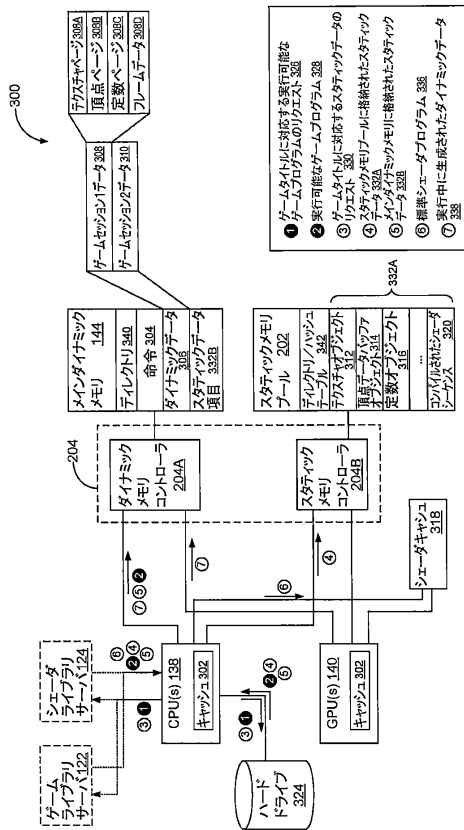


Figure 2

【 図 3 A 】



【 図 3 B 】

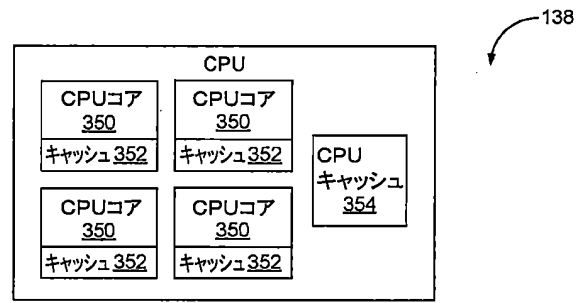


Figure 3B

Figure 3A

【 図 3 C 】

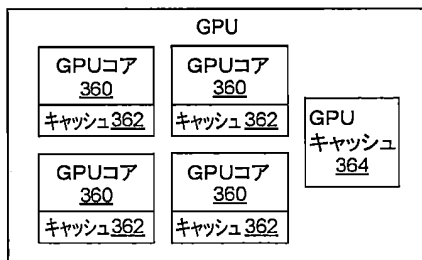


Figure 3C

【 図 3 D 】

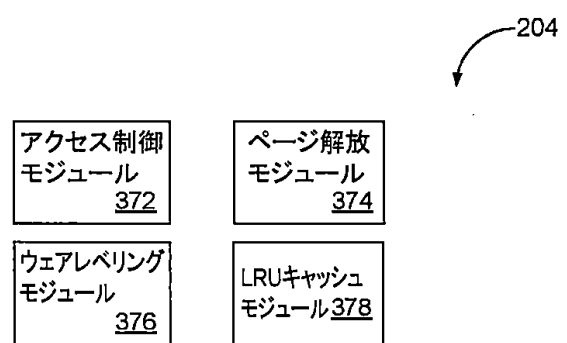


Figure 3D

【図 4 A】

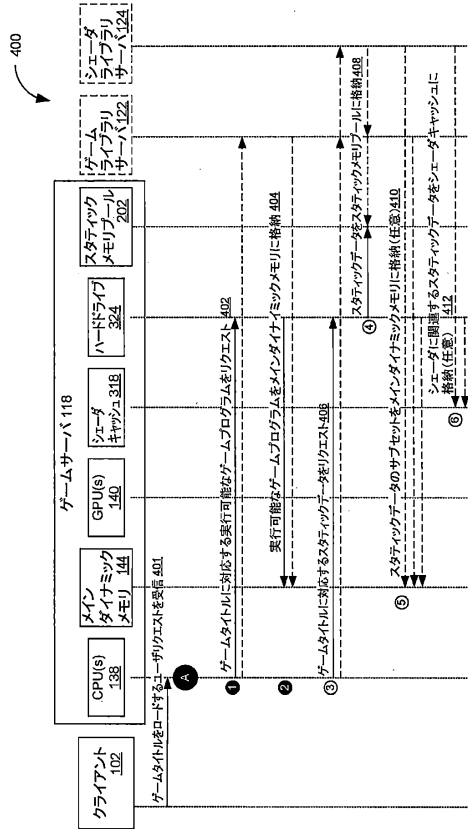


Figure 4A

【図 4 B】

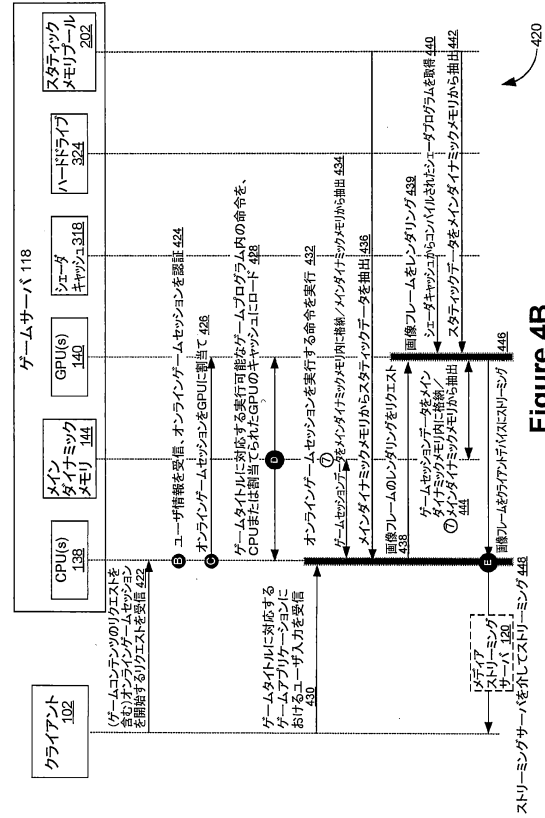


Figure 4B

【図 4 C】

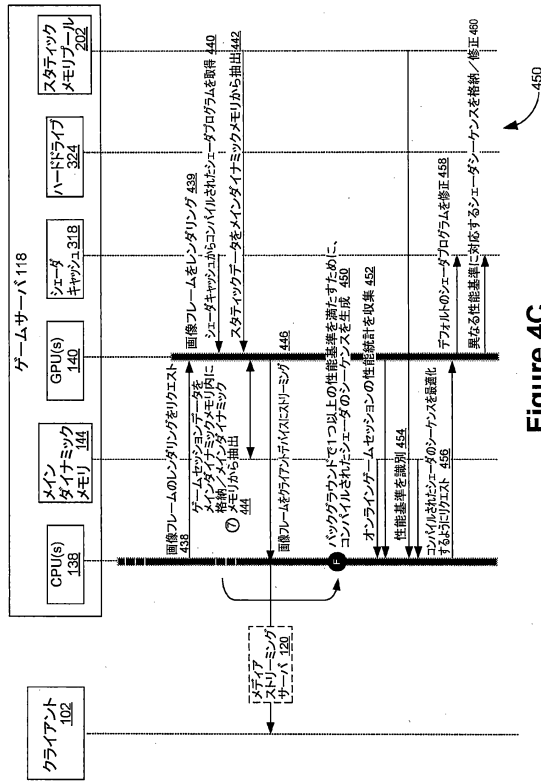


Figure 4C

【図 4 D】

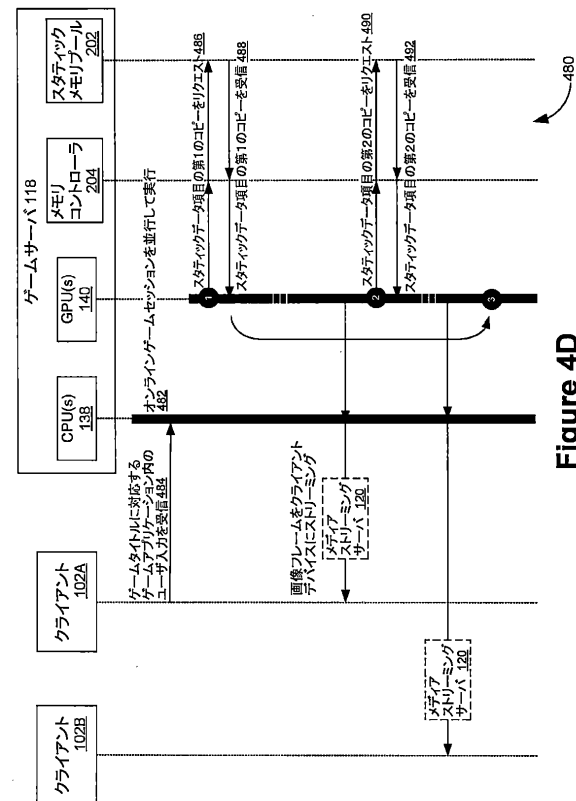


Figure 4D

【図 5 A】

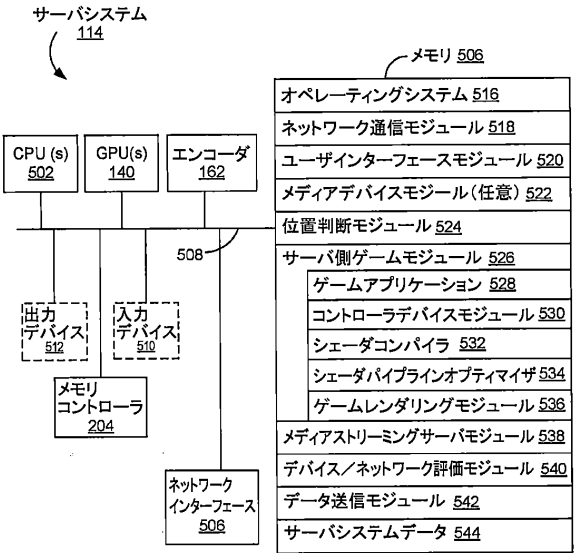


Figure 5A

【図 5 B】

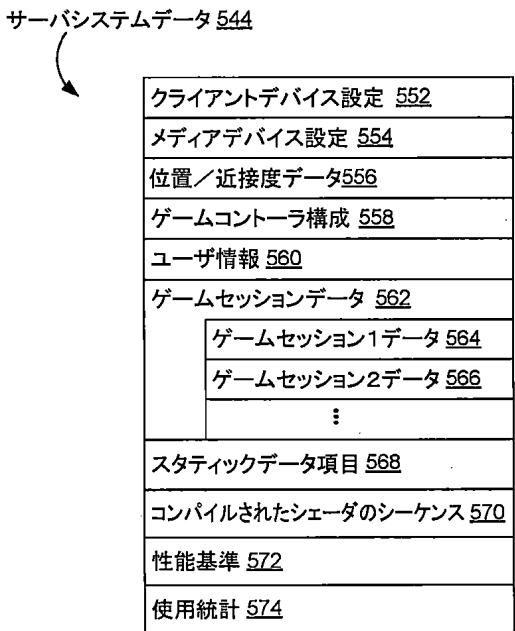


Figure 5B

【図 6】

クライアントデバイス 600 (たとえば、携帯電話またはゲームコントローラ)

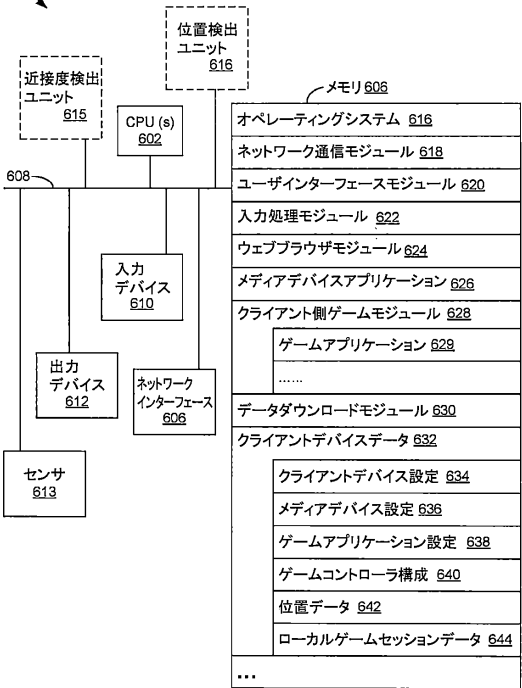


Figure 6

【図 7】

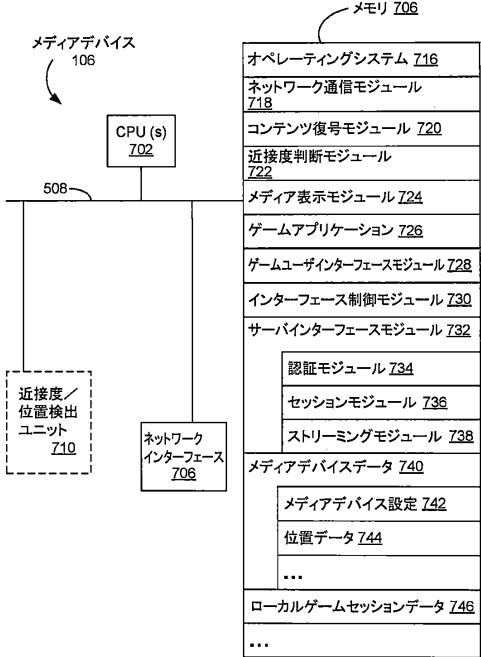


Figure 7

10

20

30

40

50

【図 8 A】

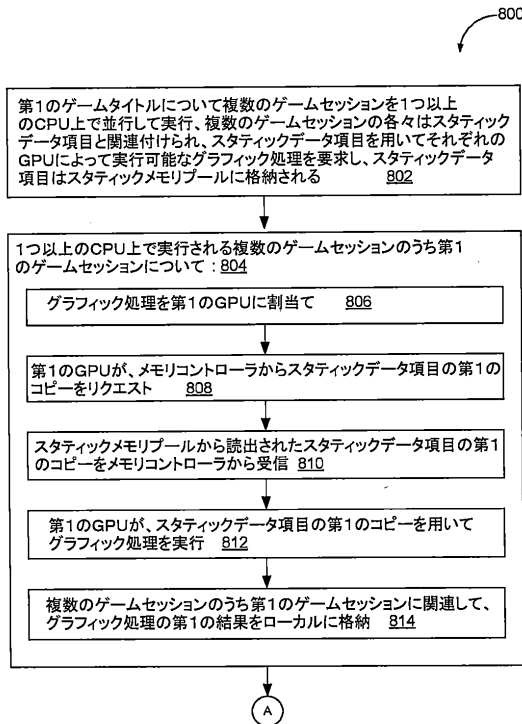


Figure 8A

【図 8 B】

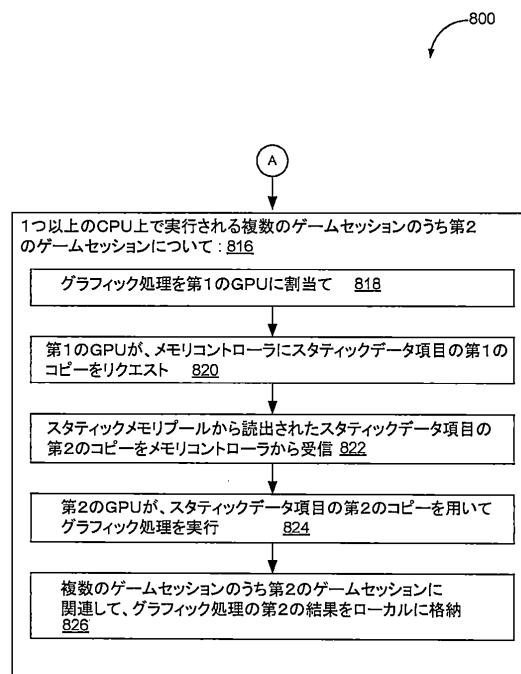


Figure 8B

【図 9】

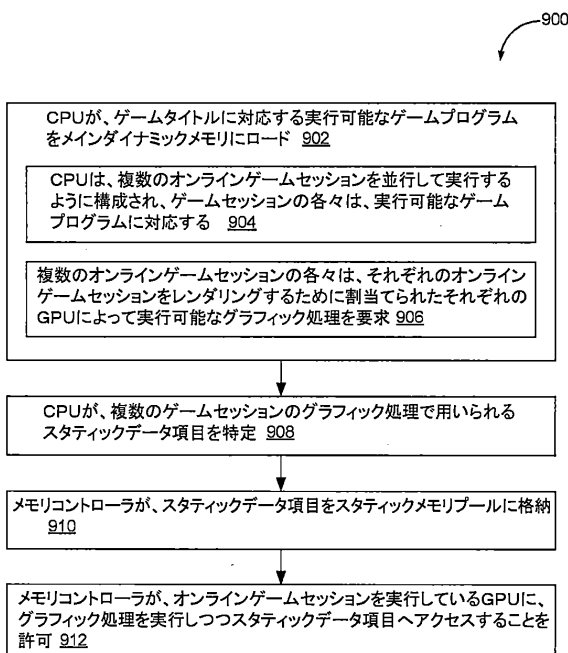


Figure 9

【図 10】

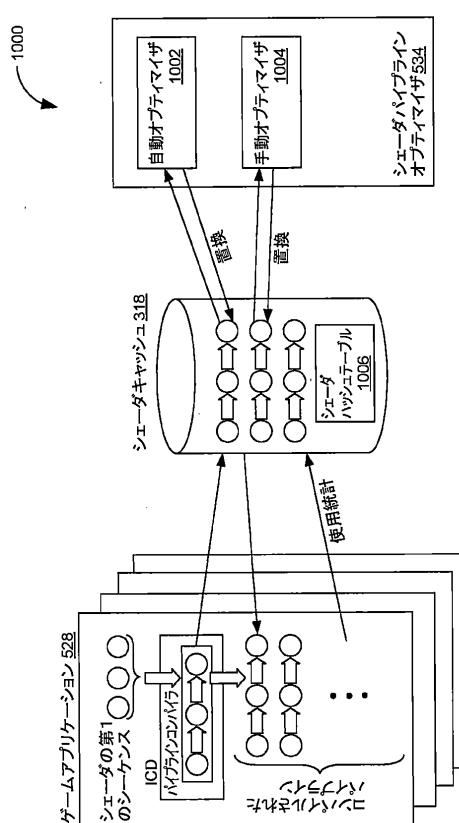


Figure 10

【図 1 1 A】

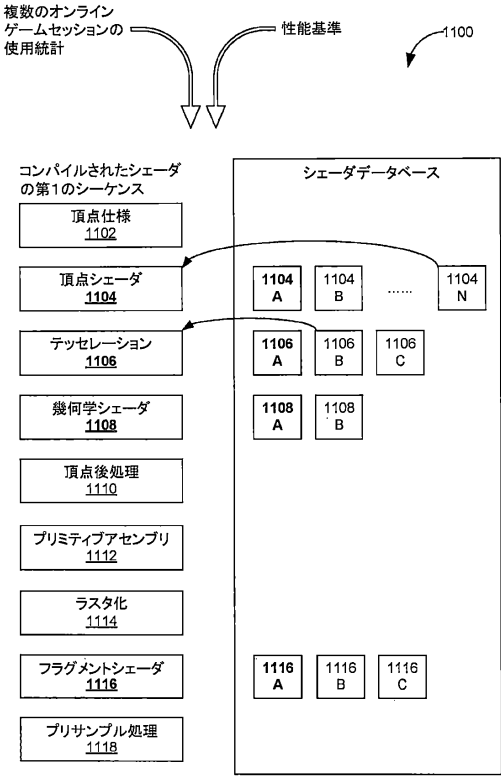


Figure 11A

【図 1 1 B】

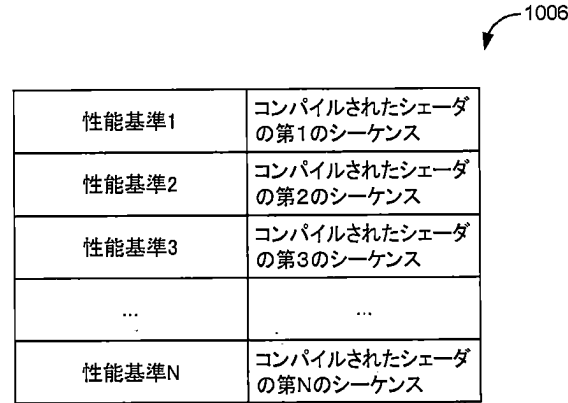


Figure 11B

【図 1 1 C】

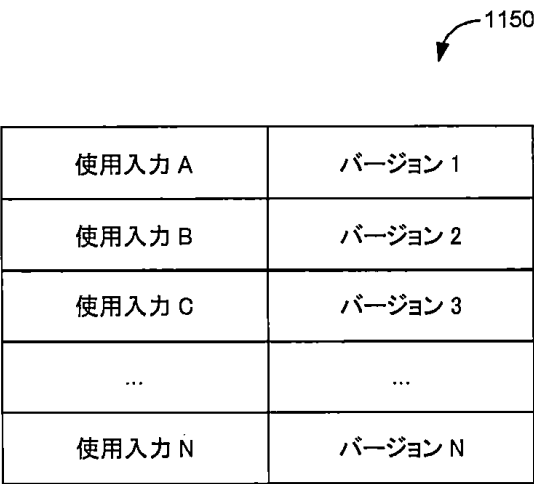


Figure 11C

【図 1 2】

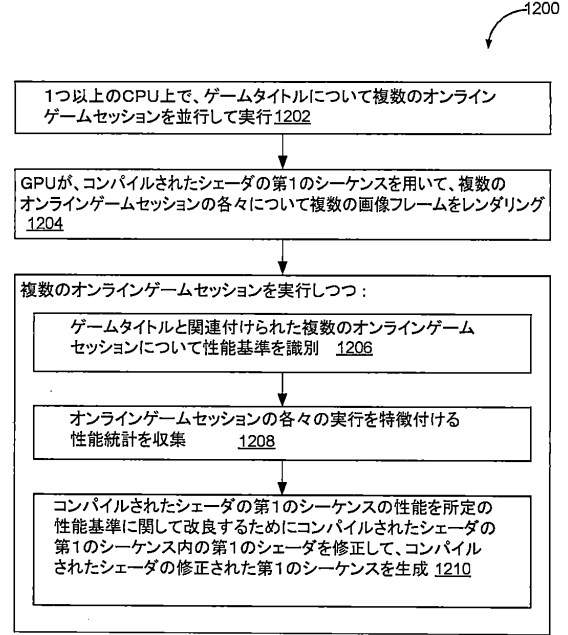


Figure 12

フロントページの続き

(51)国際特許分類

F I

A 6 3 F	13/52 (2014.01)	A 6 3 F	13/52
A 6 3 F	13/35 (2014.01)	A 6 3 F	13/35
A 6 3 F	13/70 (2014.01)	A 6 3 F	13/70

(72)発明者 レベンティス, ポール

アメリカ合衆国、9 4 0 4 3 カリフォルニア州、マウンテン・ビュー、アンフィシアター・パークウェイ、1 6 0 0

(72)発明者 ロイ, ジャン - フランソワ

アメリカ合衆国、9 4 0 4 3 カリフォルニア州、マウンテン・ビュー、アンフィシアター・パークウェイ、1 6 0 0

審査官 松永 隆志

(56)参考文献 米国特許出願公開第 2 0 1 6 / 0 1 7 1 7 5 7 (U S , A 1)

特表 2 0 1 7 - 5 1 6 2 0 7 (J P , A)

特表 2 0 1 5 - 5 2 4 0 9 2 (J P , A)

特表 2 0 1 7 - 5 1 0 8 6 2 (J P , A)

特開 2 0 1 7 - 2 1 5 9 9 7 (J P , A)

池田 孝利 他 2 名, G P Uアプリケーションを高速化するための命令割当て方式, 情報処理学会論文誌, 2005年08月15日, 第46巻 第SIG12(ACS11)号, p.140-149

(58)調査した分野 (Int.Cl., D B 名)

G 0 6 T 1 5 / 0 0

G 0 6 F 1 2 / 0 0

G 0 6 F 1 2 / 0 8 7 5

G 0 6 F 1 2 / 0 8 0 6

G 0 6 F 1 2 / 0 8 4 2

A 6 3 F 1 3 / 5 2

A 6 3 F 1 3 / 3 5

A 6 3 F 1 3 / 7 0