

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2020-520205
(P2020-520205A)

(43) 公表日 令和2年7月2日(2020.7.2)

(51) Int.Cl.	F I	テーマコード (参考)
HO4N 21/44 (2011.01)	HO4N 21/44	2FOO2
GO4G 5/00 (2013.01)	GO4G 5/00 J	5C164
HO4L 7/04 (2006.01)	HO4L 7/04	5K047
HO4L 7/00 (2006.01)	HO4L 7/00 990	

審査請求 未請求 予備審査請求 未請求 (全 31 頁)

(21) 出願番号 特願2020-507983 (P2020-507983)
 (86) (22) 出願日 平成30年4月9日 (2018.4.9)
 (85) 翻訳文提出日 令和1年12月10日 (2019.12.10)
 (86) 国際出願番号 PCT/US2018/026756
 (87) 国際公開番号 W02018/200184
 (87) 国際公開日 平成30年11月1日 (2018.11.1)
 (31) 優先権主張番号 62/489,264
 (32) 優先日 平成29年4月24日 (2017.4.24)
 (33) 優先権主張国・地域又は機関 米国 (US)

(71) 出願人 519379400
 フェニックス リアル タイム ソリュー
 ションズ, インコーポレイテッド
 アメリカ合衆国 60611 イリノイ州
 シカゴ, ノース ワバシ アベニュー
 330, フロア 23
 (74) 代理人 110000062
 特許業務法人第一国際特許事務所
 (72) 発明者 ビーラー, ステファン
 アメリカ合衆国 60611 イリノイ州
 シカゴ, ノース ワバシ アベニュー
 330, フロア 23

最終頁に続く

(54) 【発明の名称】 リモートデータのアプリケーションによる消費を同期させるための方法と装置

(57) 【要約】

本出願では、複数のコンシューマが、順序付けられたデータストリームを同時に視聴することを可能にするための方法とシステムが、提示されている。本発明の一実施形態は、データのソースである1つのサーバと、データのコンシューマである複数のクライアントが存在するクライアントサーバモデルに従う。さらに、本発明は、1) データソースとして機能する複数のサーバまたはサーバクラスタ、およびコンシューマとして機能する複数のクライアント、2) 各ピアが、データのソースおよび/またはコンシューマの両方になり得る純粋なピアツーピア、3) ピアが、データのソースおよび/またはコンシューマのセットの両方になることができ、かつ1つまたは複数のサーバが、ソースとして機能し、そしてどのデータがどのピアから利用可能かという様なピアに関するメタ情報をバックアップしかつ提供する、サーバにより支援されるピアツーピアモデル、を含むがこれらに限定されない他の多くのモデルによっても実施させることができる。

【選択図】 図1

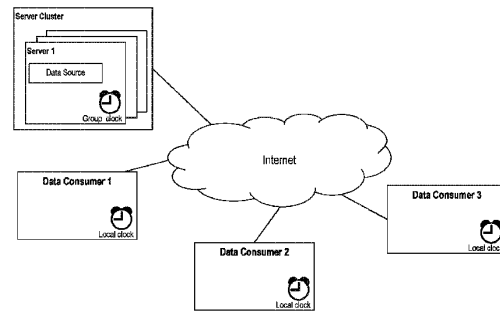


FIG. 1

【特許請求の範囲】**【請求項 1】**

順序付けられたデータストリームの複数のコンシューマによって順序付けられたデータストリームを受信するステップであって、前記順序付けられたデータストリームが、データチャンクの順序付けられたシーケンスを含む、ステップ、および

前記順序付けられたデータストリームの消費を開始する前に、前記ローカル時刻と前記グループ時刻の間のマッピングを取得するために、前記複数のコンシューマが前記データチャンクを消費すべきグループ時刻を示すグループクロックと、前記複数のコンシューマのローカル時刻を示すローカルクロックとを同期させることと；

前記順序付けられたデータストリームの前記データチャンクと、前記順序付けられたデータストリームの前記データチャンクに関連付けられているグループ時刻の表示を受信することと；

前記グループ時刻および前記ローカル時刻と前記グループ時刻の間の前記マッピングに基づいて、前記グループ時刻に対応する前記ローカル時刻、前記ローカル時計に関連付けられている前記ローカル時刻を計算することと；

前記ローカル時刻を前記ローカル時計に関連付けられている現在時刻と比較して、前記データチャンクに関連付けられている前記ローカル時刻が、前記現在時刻より前か、前記現在時刻と実質的に同一か、または前記現在時刻の後であることを示す比較を生成することと；

前記比較に基づいて、前記データチャンクを削除する、前記データチャンクの持続時間を調整する、または前記データチャンクを消費することにより、前記複数のコンシューマが、前記データチャンクを同期した状態で消費することを保証することと

により、前記複数のコンシューマ間で前記順序付けられたデータストリーム内のデータチャンクの消費を、同期させるステップ、

を備える方法。

【請求項 2】

前記現在時刻が、前記ローカル時刻より前であると決定するステップ、および

受信されたデータチャンクのシーケンス内の前記データチャンクを、前記データチャンクに関連付けられている前記ローカル時刻を示す場所に、配置するステップ、

を備える、請求項 1 に記載の方法。

【請求項 3】

前記現在の時刻が、前記ローカル時刻と前記データチャンクの前記持続時間で定義された時間範囲内にあると決定するステップ、

前記データチャンクの前記持続時間を短縮させるステップ、そして

前記データチャンクの短縮させた持続時間に対応する持続期間に対する、前記データチャンクを消費するステップ

を備える、請求項 1 に記載の方法。

【請求項 4】

前記現在の時刻が、前記ローカル時刻より後であると決定ステップ、そして

前記データチャンクを削除するステップ

を備える、請求項 1 に記載の方法。

【請求項 5】

以前に消費されたデータチャンクに続く第 2 のデータチャンクが、利用可能ではないと決定するステップ、そして

前記以前に消費されたデータチャンクの前記持続時間を延長するステップ、

を備える、請求項 1 に記載の方法。

【請求項 6】

前記データチャンクが、ビデオフレームまたはオーディオのセクションを備える、

請求項 1 に記載の方法。

【請求項 7】

10

20

30

40

50

前記グループロックと前記ローカルロックを同期させる当該サブステップが、
前記グループロックに関連付けられている複数のグループ時刻と、前記複数のグループ時刻に対応する複数のローカル時刻を取得することと、
前記複数のグループ時刻および前記対応する複数のローカル時刻に基づいて、グループロックとローカルロックとの間の線形対応を計算することと
を備える、請求項 1 に記載の方法。

【請求項 8】

前記順序付けられたデータストリームの消費中に、前記グループロックと前記ローカルロックを同期させる当該サブステップを実行し、これにより前記ローカルタイムと前記グループタイム間の前記マッピングを改善する、
請求項 1 に記載の方法。

10

【請求項 9】

順序付けられたデータストリーム内のデータチャンクが、複数のコンシューマによって消費されるべき時を示すグループ時刻を測定するグループロックを定義するステップ、
前記グループロックを、前記順序付けられたデータストリームの消費を要求する前記複数のコンシューマに関連付けられている複数のローカルロックに同期させるステップであって、当該同期させるステップが、複数のグループ時刻を前記複数のコンシューマに送信することを備える、ステップ、

前記データチャンクに関連付けられている前記グループ時刻および前記データチャンクを前記複数のコンシューマに送信するステップ、

20

前記複数のコンシューマの内のコンシューマから前記データチャンクの到着時刻を受信するステップ、

前記データチャンクの到着時刻と前記データチャンクに関連付けられている前記グループ時刻に基づいて、前記コンシューマが、かなりの割合のデータチャンクを遅れて既に受信しているか否かを決定するステップ、そして

前記コンシューマが、かなりの割合のデータチャンクを遅れて既に受信していると決定されると、前記データチャンクの品質または順序付けられたデータストリームのレイテンシを調整することにより、前記複数のコンシューマが、前記順序付けられたデータストリームを同期した状態で消費することを保証するステップ

を備える方法。

30

【請求項 10】

前記複数のコンシューマの内の前記コンシューマに送信される前記データチャンクに関連付けられている最小品質閾値を指定する設定を取得するステップ、

前記データチャンクの品質を最小品質閾値まで下げることにより、前記コンシューマが、データチャンクのかかなりの割合を遅く受信することが阻止されるか否かを決定するステップ、そして

前記データチャンクの品質を最小品質閾値まで下げることにより、前記コンシューマが、データチャンクのかかなりの割合を遅く受信することが阻止されると決定されると、前記データチャンクの品質を最大でも前記最小品質閾値まで下げ、そして前記データチャンクを、前記かなりの割合のデータチャンクを遅れて既に受信している前記コンシューマに送信するステップ

40

を備える、請求項 9 に記載の方法。

【請求項 11】

前記コンシューマが、前記かなりの割合のデータチャンクを遅れて既に受信していると決定されると、前記かなりの割合のデータチャンクが遅れて受信されなくなるまで、前記コンシューマに送信される前記データチャンクの品質を低下させる、

請求項 9 に記載の方法。

【請求項 12】

前記複数のコンシューマの内の前記コンシューマに送信される前記データチャンクに関連付けられている最小品質閾値を指定する設定を取得するステップ、

50

前記データチャンクの品質を最小品質閾値まで下げると、前記コンシューマが前記データチャンクのかなりの割合を遅く受信することが阻止されるか否かを決定するステップ、そして

前記データチャンクの品質を最小品質閾値まで下げると、前記コンシューマが前記データチャンクのかなりの割合を遅く受信することが阻止されないと決定されると、前記データチャンクに関連付けられていてかつ前記複数のコンシューマに送信されるグループ時刻を、前記かなりの割合のデータチャンクを遅れて受信する前記コンシューマでの前記データチャンクの前記到着時刻に対応する後の時刻にまで増大させ、これにより複数のコンシューマ間の消費の同期を保証するステップ

を備える、請求項 9 に記載の方法。

10

【請求項 13】

前記複数のコンシューマが、前記順序付けられたデータストリームを消費している間に、前記複数のコンシューマに加わる新しいコンシューマからの要求を受信するステップ、そして

前記第 2 のデータチャンクおよび前記第 2 のデータチャンクに関連する第 2 のグループ時刻を新しいコンシューマに送信するステップであって、前記第 2 のグループ時刻が、前記複数のコンシューマに現在送信されつつある前記グループ時刻より遅い、ステップ

を備える、請求項 9 に記載の方法。

【請求項 14】

データチャンクの順序付けられたシーケンスを備える順序付けられたデータストリームの複数のコンシューマと、

20

前記複数のコンシューマに関連付けられている複数のローカルクロックであって、前記複数のローカルクロック内のローカルクロックが、前記複数のコンシューマ内のコンシューマのローカル時刻を示す、ローカルクロックと、

前記複数のコンシューマが、前記順序付けられたデータストリーム内のデータチャンクを消費すべきグループ時刻を示すグループクロックと、

前記複数のコンシューマの間で前記順序付けられたデータストリーム内の前記データチャンクの消費を同期させる以下の命令を実行する、前記複数のコンシューマ内の前記コンシューマに関連付けられている 1 つまたは複数のプロセッサと、

前記順序付けられたデータストリームの前記消費を開始する前に、前記ローカル時刻と前記グループ時刻の間のマッピングを取得するために前記グループクロックと前記ローカルクロックを同期させる命令；

30

前記データチャンクおよび前記順序付けられたデータストリーム内の前記データチャンクに関連付けられているグループ時刻の指示を受信するための命令；

前記グループ時刻と、前記ローカル時刻と前記グループ時刻の間の前記マッピングに基づいて、前記グループ時刻に対応する、前記ローカル時計に関連付けられている前記ローカル時刻を計算するための命令；

前記ローカル時刻を前記ローカル時計に関連付けられている現在時刻と比較して、前記データチャンクに関連付けられている前記ローカル時刻が、前記現在時刻より前か、前記現在時刻と実質的に同じか、または前記現在時刻の後かを示す比較を生成する命令；そして

40

前記比較に基づいて、前記複数のコンシューマが、前記データチャンクを削除する、前記データチャンクの持続時間を調整する、または前記データチャンクを消費することにより、前記データチャンクを同期した状態で消費することを保証する命令

を備える、システム。

【請求項 15】

前記グループクロックと同期する第 2 のグループクロックであって、前記複数のコンシューマが前記順序付けられたデータストリーム内の前記データチャンクを消費すべき前記グループ時刻を示す、第 2 のグループクロックと、

前記第 2 グループクロックに関連付けられている 1 つ以上のプロセッサであって、以下

50

の命令：

前記グループクロックが動作していないと決定する命令；および

前記グループクロックが動作していないと判断すると、第2のグループクロックによって示される前記グループ時刻を前記複数のコンシューマに送信する命令、
 を実行する前記第2グループクロックに関連付けられている1つ以上のプロセッサと、
 を備える、請求項14に記載のシステム。

【請求項16】

前記命令が、

複数の順序付けられたデータストリームを受信する命令であって、前記複数の順序付けられたデータストリーム内の順序付けられたデータストリームの夫々が、残りの複数の順序付けられたデータストリームと同期している、命令、

前記複数の順序付けられたデータストリーム内の複数のデータチャンクに関連付けられている複数のグループ時刻を取得する命令、および

前記同じ複数のグループ時刻を有する前記複数のデータチャンクを同期した状態で消費する命令、

を備える、請求項14に記載のシステム。

【請求項17】

前記データチャンクが、前記データチャンクを消費する時刻を前記複数のコンシューマに示すグループクロックに関連付けられている消費時刻、フレーム識別、および消費されるデータを備える、請求項14に記載のシステム。

【請求項18】

複数の受信されたデータチャンクを記憶するように動作可能なメモリ、

前記現在時刻が前記ローカル時刻より前であると決定する命令、そして

前記複数の受信されたデータチャンクを、前記データチャンクに関連付けられている前記ローカル時刻を示す位置に記憶する前記メモリ内に、前記データチャンクを配置する命令

を備える、請求項14に記載のシステム。

【請求項19】

前記命令が、

現在時刻が、前記ローカル時刻と前記データチャンクの前記持続時間によって定義された時間範囲内にあると決定する命令、

前記データチャンクの持続時間を短縮させる命令、そして

前記短縮された持続時間に対応する持続期間に対して、前記データチャンクを消費する命令

を備える、請求項14に記載のシステム。

【請求項20】

前記命令が、

前記現在時刻が、前記ローカル時刻より後であると決定する命令、および

前記データチャンクを削除する命令

を備える、請求項14に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本出願は、データの配信に、より具体的には、リモートデータのアプリケーションによる消費を同期させるための方法およびシステムに関する。

【背景技術】

【0002】

アプリケーションのストリーミングは、ネットワーク遅延および他の遅延の影響を受けやすい。そのため、通常、アプリケーションの特定のストリーミング（例、ビデオのストリーミング）の複数のユーザーは、同時に同じフレームを見ることは無い。これらのユー

10

20

30

40

50

ザーが、アプリケーションのストリーミング中にチャットセッションまたは他のリアルタイム通信にも参加している場合、これらの参加者は、しばしば同じことについて話し合っていないため、このような場合には意味のある会話を維持することは困難である。つまり、一方の参加者は、他方の参加者がまだ見たことのない何かについてコメントしている可能性がある。

【発明の概要】

【発明を解決するための手段】

【0003】

本出願では、複数のコンシューマが、順序付けられたデータストリームを同時に視聴することが出来るようにするための方法およびシステムが提示される。本発明の一実施形態は、データのソースである1つのサーバと、データのコンシューマである複数のクライアントが存在するクライアントサーバモデルに従う。さらに、本発明は、1) データソースとして機能する複数のサーバまたはサーバクラスタ、およびコンシューマとして機能する複数のクライアント、2) 各ピアが、データのソースおよび/またはコンシューマの両方になり得る純粋なピアツーピア、3) ピアが、データのソースおよび/またはコンシューマのセットの両方になることができ、かつ1つまたは複数のサーバが、ソースとして機能し、そしてどのデータがどのピアから利用可能かという様なピアに関するメタ情報をバックアップしかつ提供する、サーバにより支援されるピアツーピアモデル、を含むがこれらに限定されない他の多くのモデルによっても実施させることが出来る。

10

【図面の簡単な説明】

20

【0004】

【図1】本発明による例示的な同期されているグループを示すブロック図である。

【図2】単一フレームのデータチャンクおよびマルチフレームのデータチャンクの例を示す。

【図3】本発明による2つのコンシューマのフレーム同期を示すタイミング図である。

【図4】本発明によるシステムアーキテクチャを示す。

【図5】本発明によるシステムアーキテクチャを示す。

【図6】本発明によるシステムアーキテクチャを示す。

【図7】同期ステップを実行するデータソースを示す。

【図8】セッションに参加し、クロック同期ステップを実行するコンシューマを示す。

30

【図9】パブリックインターネットを介してコンシューマにデータを送信するデータソースを示す。

【図10】データの一部がコンシューマに既に届いていることを示す。

【図11】3つのビデオフレームと1つのグループ消費タイムスタンプ(GCTS: Group Consumption Timestamps)を受信したコンシューマの詳細図を示している。

【図12】コンシューマが各ビデオフレームのローカル消費タイムスタンプ(LCTS: Local Consumption Timestamps)を計算したことを示している。

【図13】より多くのデータがコンシューマに届いたことを示す。

【図14】コンシューマがLCTS3とLCTS4を計算したことを示す。

【図15】LCTS_kが計算されたことを示す。LCTS_kは、データソースによって提供されたGCTS_kから直接計算することが出来る。

40

【図16】第2のデータコンシューマ2がセッションに参加したことを示す。

【図17】データコンシューマ2が、データコンシューマ1が受信しているのと同じデータの受信を開始することを示す。

【図18】コンシューマがクロック同期ステップを定期的に繰り返すことを示す。

【図19】各コンシューマがそれ自体のローカルクロックを有しているにもかかわらず、データが、同じグループ時刻に(例、同期した状態で)データコンシューマ1およびデータコンシューマ2によってどのように消費されるかを示す。

【図20】本発明によるキューおよび同期されたデータ消費を示すブロック概略図である。

50

【図 2 1 A】本発明によるデータの優先度に基づく順序付けを示す。

【図 2 1 B】本発明によるデータの優先度に基づく順序付けを示す。

【図 2 1 C】本発明によるデータの優先度に基づく順序付けを示す。

【図 2 1 D】本発明によるデータの優先度に基づく順序付けを示す。

【図 2 1 E】本発明によるデータの優先度に基づく順序付けを示す。

【図 2 1 F】本発明によるデータの優先度に基づく順序付けを示す。

【図 2 1 G】本発明によるデータの優先度に基づく順序付けを示す。

【図 2 1 H】本発明によるデータの優先度に基づく順序付けを示す。

【図 2 2】本発明による複数のデータストリームの消費の同期を示す。

【図 2 3】一実施形態による、複数のコンシューマの間でデータチャンクの消費を同期させる方法のフローチャートである。

【図 2 4】別の実施形態による、複数のコンシューマの間でデータチャンクの消費を同期させる方法のフローチャートである。

【図 2 5】本明細書で説明されている方法論の 1 つまたは複数を実行させるための命令のセットを実施させることが出来るコンピュータシステムの例示的な形態のマシンの構成図を示す。

【発明を実施するための形態】

【0005】

次に、様々な実施形態の例を記載する。以下の記載では、これらの例を完全に理解し、かつ説明を有効にするための特定の詳細が、提供される。しかし、関連技術の当業者は、開示された実施形態のいくつかは、これらの詳細の多く無しに実施することが出来ることを理解するであろう。

【0006】

同様に、関連技術の当業者は、実施形態のいくつかは、本明細書には詳細に記載されていない他の明らかな特徴を含むことが出来ることを理解するであろう。さらに、様々な例の関連する説明を不必要に曖昧にすることを避けるために、いくつかのよく知られている構造または機能は、以下では、詳細に示されないかまたは記載されない場合がある。

【0007】

以下で使用される用語は、実施形態の特定の例の詳細な説明と併せて使用されている場合でも、その最も合理的な方法で解釈されるものとする。実際、特定の用語が以下で強調されることさえある。しかしながら、制限された方法で解釈されることを意図した用語は、この発明の詳細な説明のセクションでそのように明確かつ具体的に定義されるであろう。

【0008】

定義

以下の段落には、本文書全体で使用される定義のセットが含まれる。

【0009】

データチャンクおよびフレーム

データチャンク (DataChunk) は、関連付けられている消費タイムスタンプを持つデータフレームのコレクション、おそらく 1 つのコレクションを意味するように使用される。本発明の例示的な実施形態では、各データフレームは、フレーム識別子、データタイムスタンプ、およびデータ自体から構成される。消費タイムスタンプ (ConsumptionTS) が、各データフレーム (DataFrame) とともにヘッダーフィールドとして送信されるような他の実施形態では、フレーム識別子 (FrameID) とデータタイムスタンプ (DataTS) は、オプションである。タイムスタンプは、優先度を示すために使用されるが、これは、例えば、フレームタイプまたはソースに基づいて、簡単に一般化させることが出来る。

DataChunk ::= ConsumptionTS {DataFrame} 1-n

DataFrame ::= (FrameID) (DataTS) Data

【0010】

図 2 は、このことを、単一のおよびマルチフレームのデータチャンクにより示す。

【0011】

消費のバッファリング

データチャンクはクライアントに到着し、そしてバッファ（図21A～H）に配置され、そこからデータチャンクが消費される。一例としてビデオを使用すると、データチャンクは、複数のビデオフレームで構成されていて、かつこれらのフレームは、レンダリングのために着信バッファから消費される。このモデルの別のインスタンス化では、消費されるデータチャンクとフレームは、サードパーティのアプリケーションに渡すことができる。再び、一例としてビデオを使用すると、これは、ほとんどのHTML5対応ブラウザに組み込まれているMedia Service Extensionになり得る。

【0012】

バッファ消費と消費時刻

消費タイムスタンプCT（consumption time）を有するデータチャンクを受信すると、ローカル時刻LCiを有するクライアントiは、チャンクのフレームをバッファリングする、それらを即座に再生する、またはLCiとCTの相違に応じて、それらのサブセット、すなわち、チャンクの一部を破棄する、の何れかを行う。

【0013】

同期されているグループ

同期されているグループという用語は、全てのコンシューマが、同時にデータを消費することを望む、データソースとコンシューマから構成されるデバイスのグループを指すために使用される。

【0014】

リモートデータのアプリケーションによる消費の同期

本発明の実施形態は、クライアントのグループ間でデジタルデータ（例、ビデオ）、を同期した状態で消費するためのアプローチを提供する。消費されるデータは、1つまたはソースのコレクションによって提供され、クライアントとデータソースは、インターネット経由で接続され、かつメモリともグループクロックとも共有されていないと仮定されている。本発明によってサポートされる同期された消費により、クライアントは、消費された共有コンテンツに関して、あたかもそれらが物理的に並置されているかのように、互いに同時に同期した状態に対話することが出来る。

【0015】

本発明の実施形態は、クライアントのアプリケーションがグループクロックを共有しない場合に、2つ以上のクライアントのアプリケーションによるリモートデータの消費を同期させる方法および装置を提供する。本発明の実施形態では、参加者は、以下で説明するクロック同期アルゴリズムを使用する。

【0016】

文脈を提供するために、最初に、本発明の実施形態の意図された使用の一例が記載される。異なる都市に、インターネットからストリーミングされたビデオを同時に見たい2人の人がいる。ビデオを見ている間、彼らは、電話、テキスト、チャット、または他のリアルタイム通信方法を使用して、ビデオのイベントについて議論することを望む。彼らは、一方の人が見ているビデオの再生が、他方の人が見ているビデオの再生と同期し、その結果、彼らの議論が明確になり、そして一方の人が、他方の人が見ていないビデオ内のイベントに言及することが確実に起こらないことを望む。本発明の実施形態は、この例における二人のビデオの再生を同期させる。

【0017】

本発明の現在好ましい実施形態は、2つ以上のコンピュータアプリケーションによって実行される一連のステップから構成される。データの発信元とコンピュータアプリケーションを実行するデバイスの全てが、地理的に離れている場合であっても、これらのステップを実行すると、2つのコンピュータアプリケーションは、同期した状態でデータを消費する。

【0018】

本発明の一実施形態は、データのソースである1つのサーバと、データのコンシューマ

10

20

30

40

50

である複数のクライアントが存在するクライアントサーバモデルに従う。従って、「サーバ」と「データソース」という用語は、同じ意味で使用される。同様に、「クライアント」と「コンシューマ」という用語も、同じ意味で使用される。しかしながら、本発明は、以下の例を含むがこれらに限定されない多くの他のモデルによって具体化することも出来る：

データソースとして機能する複数のサーバまたはサーバクラスタ、およびコンシューマとして機能する複数のクライアント。

各ピアが、データのソースまたはコンシューマ、あるいはその両方になることが出来る純粋なピアツーピア。

ピアを、データのソースまたはコンシューマ、およびその両方とすることができ、かつ1つ以上のサーバのセットが、ソースとして機能し、かつどのピアからどのデータが利用可能という様なピアに関するメタ情報をバックアップしかつ提供する、サーバ支援ピアツーピアモデル。

10

【0019】

新しいメンバが、同期されているグループに参加するときは常に、それは、クロック同期アルゴリズムを実行しなければならない。初期化後、グループのメンバは、クロック同期アルゴリズム（後で詳しく説明する）を使用して、定期的にクロックを同期させることが出来る。同期時間の間、クライアントクロックは、ドリフトラット d_r に基づいてドリフトする可能性がある（クロック C の d_C / d_t 、ドリフトは正にも負にもなり得る）。

20

【0020】

このステップの結果として、新しいグループメンバは、任意のグループクロックタイムスタンプをローカルクロックタイムスタンプに、およびその逆に変換することが出来る。グループクロック同期の詳細は、以下で詳しく説明される。

【0021】

サーバによってそのクライアントに送信されるデータチャンクは、チャンクを構成するデータフレームがクライアントによって消費される（潜在的に）将来の時刻である、消費タイムスタンプ（CS: consumption timestamp）を含む。異なる実施形態では、消費タイムスタンプは、異なるチャンネルで送信され、かつ後のマッチングのためにデータチャンクIDまたはデータフレームIDを含むことが出来るであろう。

30

【0022】

クライアントは、データのチャンクを受信すると、クライアントのクロックドリフト率に基づいて、サーバのタグ付き消費タイムスタンプをローカル消費タイムスタンプに変換する（以下の式 AAを参照）。ローカル消費タイムスタンプがクライアントのクロックに対して未来にある場合、クライアントは、そのクロックが、消費タイムスタンプに等しいまたはそれ以上の時間に達するまで、データチャンクを消費することを待機する。

【0023】

クライアントは、サーバからの全てのデータチャンクの受信を確認する。確認応答は、サーバに、チャンクの到着時刻または消費時刻と到着時刻との差を通知する。わずかな変更例では、クライアントは、データチャンクごとに応答を送信するのではなく、消費時刻後にデータチャンクが到着したときにのみ、到着時刻、または消費タイムスタンプと到着時刻との差を示す応答を送信する。

40

【0024】

消費時刻が経過した後にデータチャンクが到着する場合、クライアントは、含まれているデータフレームの一部または全てを破棄する、またはクライアントのローカル時刻が消費時刻のそれと一致するまで、それらのデータフレームの一部または全てをより高速で消費することが出来る。以下に簡単な説明例を示す。データチャンクが、関連付けられている持続時間が40ミリ秒の単一のビデオフレームで構成されていて、かつデータが、消費時刻の10ミリ秒後に到着する場合、クライアントは、それがフレーム持続時間を30ミリ秒に短縮すると、依然としてデータを消費することが出来る。しかしながら、特定のアプリケーションでは、データの到着が遅れた場合、そのデータは破棄するという条件が適

50

用される。

【0025】

グループ時刻をローカル時刻に変換する別の方法は、ドリフト(drift)を計算することである：

$$\text{drift} = (\text{LCTS1} - \text{LCTS0}) / (\text{GCTS1} - \text{GCTS0}) \quad (\text{式 AA})$$

【0026】

ドリフトを使用して、GCTSをLCTSに変換する

$$\text{LCTSk} = \text{drift} \times (\text{GCTSk} - \text{GCTS1}) + \text{LCTS1} \quad (\text{式 BB})$$

LCTS：ローカルタイムスタンプ

GCTS：グループタイムスタンプ

GCTS1は、最後の同期時のグループタイムスタンプである。

LCTS1は、最後の同期時のローカルタイムスタンプである。

10

【0027】

取引データの品質、レイテンシ、同期

本発明の実施形態は、データが不完全であるまたは近似している場合でも、いくつかのタイプのデータは、依然として有用であるという事実を活用する。不完全データまたは近似データの意味を説明するために、いくつかの例を示す。

ビデオまたはサウンドをより低いビットレートでエンコードして、元のビデオまたはサウンドの近似値を生成することが出来る。

画像を低解像度でエンコードして、元の画像の近似値を作成することが出来る。

20

データポイントのランダムサンプリングを削除する、または各データポイントを低い精度に丸めることにより、一連の測定値を近似することが出来る。

【0028】

同期を維持するためのデータ品質 / 完全性の犠牲

一部のコンシューマは、ネットワーク接続が遅いので、消費時刻の後に一貫してデータチャンクを受信する場合がある。本発明の実施形態は、上述したデータ近似および圧縮を使用して、ネットワーク条件が、他のグループメンバに対してよりも一部のグループメンバに対してはるかに悪い場合でも、全てのグループメンバに対して同期されたデータ消費を実行する能力を保持する。クライアントが、消費時刻後にデータチャンクを一貫して受信すると、クライアントは、サーバに通知し、かつより小さいチャンクを要求することが出来る。サーバは、他のクライアントに完全なデータを提供する一方、そのクライアントには小さなデータチャンクを提供することが出来る。この小さいデータチャンクは、他のクライアントに完全なデータを提供する一方、そのクライアントの元のデータの近似値を表す。

30

【0029】

サーバは、複数レベルのデータ近似またはデータ圧縮を提供することが出来る。消費時刻の前にフルサイズのチャンクを受信することが出来るクライアントもあれば、消費時刻の前に(元のデータの近似値に近い)わずかに小さいチャンクを受信することが出来るクライアントもあれば、消費時刻前のデータの非常に大まかな近似値しか受信できないクライアントもある。

40

【0030】

近似または圧縮の程度に関係なく、同期されているグループ内の全てのクライアントには、サーバによって同じ消費タイムスタンプが提供される。これにより、例え、一部のクライアントがデータの近似値のみを消費し、かつ他のクライアントが完全な元のデータを消費している場合であっても、全てのクライアントが、同時にデータを消費することが保証される。

【0031】

データの品質 / 完全性を維持するためのレイテンシの犠牲

いくつかの用途では、近似データを消費することは望ましくないまたは実用的ではない場合がある。これらの場合、データソースが、常に一定の品質閾値を超えるデータ(例、

50

2.5 Mビット / 秒より大のビデオビットレート)を提供するように、同期されているグループを構成することが出来る。

【0032】

サーバは、クライアントから応答をデータチャンクの到着時刻に関する情報とともに受信するので、サーバは、どのクライアントが、到着時刻の後にデータチャンクを遅く受信しているかを把握している。グループが、データの品質 / 完全性を保持するように構成されている場合、サーバは、遅い消費タイムスタンプを将来の送信データに割り当てること出来る。これにより、グループのデータ消費のレイテンシが高くなる。つまり、データチャンクは、より長い時間が経過した後に消費される。しかしながら、これは、グループが同期とデータ品質を保持することを可能にする。

10

【0033】

消費時刻が明示的に提供されていない場合の消費時刻の推測

データチャンクに関連付けられている消費タイムスタンプは、データチャンクの第一フレームの消費時刻であることが出来る。後続の全てのデータフレームの消費時刻は、第一フレームの消費時刻と各フレームのタイムスタンプとに基づいて計算することが出来る。例えば、フレーム0からフレーム2までの一連の3つのビデオフレームを考えてみる。フレーム0のみが、150,000の明示的なグループ消費タイムスタンプ(GCTS)を持つとする。明示的な消費タイムスタンプを持たないフレーム1のビデオタイムスタンプは、フレーム0のビデオタイムスタンプの後3,750ティックを持つ。

20

【0034】

従って、フレーム1のGCTSは、 $150,000 + 3,750 = 153,750$ であると推測される。このタイムスタンプは、グループクロックで表されているので、コンシューマのローカルクロックに変換する必要がある。グループクロックに対してクライアントのローカルクロックにドリフトがある場合、ローカルタイムでのフレーム1の消費タイムスタンプは、ローカルクロックの3750ティックよりも大きくなる(ドリフト > 1の場合)、または小さくなる(ドリフト < 1の場合)であろう。これは、同期を維持するために、クライアントの再生速度が、グループの再生速度と実際異なることを意味する。

【0035】

クロック同期の詳細な説明

本発明の実施形態は、クロック同期フェーズに依存する。以下に、1クロック同期アルゴリズムが、Cristianの時刻同期アルゴリズム(https://en.wikipedia.org/wiki/Cristian%27s_algorithm)に基づいて説明される。このアルゴリズムは、本発明を満たすために使用することが出来るであろう。クライアントCは、サーバの時刻を要求するためにタイムサーバSにコンタクトする。要求を受信した後、サーバは、応答を準備し、そしてその時刻T(UTCのような外部ソースに基づいて設定される場合とそうでない場合がある)を追加する。応答を受信した後、クライアントはその時刻を $T + RTT / 2$ に設定する(RTTは通信の往復時間)。これに代えて、クライアントは、ローカルクロックをサーバ時刻に設定するのではなく、単にグループクロックとローカルクロックとの間のオフセットを記録することも出来る。より緊密な同期を実現するために、クライアントは、この要求を複数回実行し、そして最短のRTTで応答を使用することが出来る。ここに記載されている方法は、RTTが、要求と応答の間で均等に分割されていることを前提としている。本発明の異なる実施形態では、任意の数の異なるクロック同期アルゴリズムを利用することが出来るであろう。唯一の要件は、アルゴリズムが完了した後、グループメンバが、グループタイムスタンプとローカルタイムスタンプの間で変換することが出来る必要があることである。この要件を満たす他の同期アルゴリズムには、BerkeleyとNTPが含まれるが、これらに限定されることはない。本発明は、アプリケーションによっては、外部クロックとの同期を必要としないことがあるので、これらの代わりにいくつかの他のアルゴリズムを使用することも出来る。

30

40

【0036】

50

クロック同期アルゴリズムを2回実行することにより、クライアントは、ローカルクロックとグループクロックとの間のドリフト率を計算することが出来る。

【0037】

グループクロックの重要性を考えると、この同期グループは、プライマリグループクロックとの同期を維持する1つまたは複数のバックアップグループクロックを維持することにより、その障害許容力を高めることが出来る。プライマリグループクロックに障害が発生した場合、このグループは、バックアップグループクロックの1つを即座に昇格させて、リーダー選択アルゴリズムを使用して選択される新しいプライマリグループクロックとして機能させることが出来る。

【0038】

図の説明

図1は、本発明による例示的な同期グループを示すブロック図である。本発明の例示的な実施形態では、サーバは、1つ以上のフレームから構成することが出来るであろうデータチャンクを、追加の消費時刻(C T: consumption time)とともに、送信する。時刻C Tのデータチャンクを受信すると、ローカル時刻L C iのクライアントiは、L C iとC Tの差に応じて、チャンクのフレームをパフアリングするか、すぐに再生するか、またはそれらのサブセット(つまり、チャンクの一部)を破棄する、の何れかを行う。

【0039】

図2は、単一フレームのデータチャンクおよびマルチフレームのデータチャンクの例を示す。本発明は、単一フレームのデータチャンクのみ、マルチフレームのデータチャンクのみ、または単一フレームのデータチャンクとマルチフレームのデータチャンクの組み合わせを使用して実施することが出来るであろう。

【0040】

図3は、本発明による2つのコンシューマのフレーム同期を示すタイミング図である。この図は、2つのコンシューマが、個別のローカルクロックを持っているが、特定のグループ時刻でデータを同期した状態で消費することが出来ることを示す。より詳細なタイミング図と説明は、図19に示されている。

【0041】

図4~6は、本発明によるシステムアーキテクチャを示す。

【0042】

図4は、プライベートクラスタ内に2つのサーバを有するアーキテクチャの例を示す。サーバ1は、データソースとして機能し、そこからデータが様々なコンシューマに送信される。サーバ1は、この例のアーキテクチャでグループクロックとして機能するクロックも有する。サーバ2は、プライベートクラスタの一部でもあり、かつバックアップグループクロックを維持する。バックアップグループクロックは、サーバ1のグループクロックと同期して維持されるため、サーバ1のグループクロックに障害が発生した場合、サーバ2のクロックは、グループクロックとして機能するよう即座に昇格させることが出来る。データコンシューマ1~3は、パブリックインターネットを介してプライベートクラスタに接続されている。各コンシューマは、それ自体のローカルクロックを有する。各コンシューマは、グループクロックを使用して定期的に同期アルゴリズムを実行する。このようなアルゴリズムの結果、これらのコンシューマは、グループ時刻を自分のローカル時刻に、そしてローカル時刻をグループ時刻に変換させることが出来る。サーバ1は、パブリックインターネット経由で、各コンシューマに、データを、このデータに関連付けられている消費タイムスタンプとともに送信する。これらのコンシューマは、データを受信し、そして消費タイムスタンプを使用してこのデータを同期した状態で消費する。

【0043】

図5は、次の詳細が削除されている点を除いて、図4と同じである。

どのサーバがデータソースとして機能するかは示されていない。これは、実際には、1台のサーバがデータソースとして機能する、または複数のサーバが共同でデータソースとして機能する可能性があるためである。

10

20

30

40

50

どのサーバもグループロックを保持することが出来るため、どのサーバがグループロックを保持しているかは表示されていない。

バックアップロックは必要でないため、バックアップロックは表示されていない。

【0044】

図6は、データソースとグループロックが、完全に独立したボックスとして表示され、かつパブリックインターネット経由で接続されている点を除けば、図5と同じである。これは、グループロックがデータソースに連結される必要がないことを示すためである。データソースは、グループロックと同期するであろうそれ自体のローカルロックを有することが出来るであろう。

【0045】

図7～19は、本発明による、クライアントが参加し、それらのロックと同期し、かつソースによって公開されたデータを消費する、複数のステップによるセッションを示す。

【0046】

図7は、同期ステップを実行するデータソースを示す。これは、グループロックが、データソースのローカルロックと異なる場合に、必要となる。これにより、データソースが、グループ時刻をローカル時刻に、またローカル時刻をグループ時刻に変換させることが可能になる。

【0047】

図8は、セッションに参加し、かつロック同期ステップを実行するコンシューマを示している。これにより、コンシューマが、グループ時刻をローカル時刻に、またローカル時刻をグループ時刻に変換させることが可能になる。

【0048】

図9は、パブリックインターネットを介してコンシューマにデータを送信するデータソースを示す。この例では、送信されるデータは、(図ではVFとラベル付けされている)ビデオフレーム (Video Frame) により構成されているビデオデータである。データソースは、特定のビデオフレームに関連付けられている (図ではGCTS (Group Consumption Timestamp) とラベル付けされている) グループ消費タイムスタンプも送信する。この例では、各ビデオフレームは、全てのビデオフレームがGCTSを必要としないように、ビデオタイムスタンプも有する。図9は、GCTS1を、それが関連するデータとは別に送信することが出来ることも示す。GCTSは、グループロックで表される。GCTSは、ローカル消費タイムスタンプ (LCTS) に変換されるであろう。LCTSは、コンシューマのローカル時計で表される。

【0049】

図10は、データの一部がコンシューマに既に届いていることを示す。コンシューマは、各ビデオフレームのローカル消費タイムスタンプ (LCTS) を計算しなければならないであろう。続く図は、これを詳細に示す。

【0050】

図11は、3つのビデオフレームと1つのGCTSを既に受信しているコンシューマの詳細図を示す。

【0051】

図12は、コンシューマが、各ビデオフレームについてローカル消費タイムスタンプを既に計算していることを示している。LCTS0は、GCTS0から直接計算される。LCTS1とLCTS2は、2つのステップで計算される。まず、各ビデオフレームに関連付けられている持続時間情報を使用して、そのフレームのGCTSが推測される。次に、GCTSは、通常どおりLCTSに変換される。例えば、ビデオフレームレートが24フレーム/秒の場合、クライアントは、 $GCTS1 = GCTS0 + 1 / 24$ 秒であると推測することが出来る。その後、クライアントは、通常どおりGCTS1をLCTS1に変換することが出来る。

【0052】

10

20

30

40

50

図13は、より多くのデータが、コンシューマに届いていることを示す。つまり、より多くのビデオフレームと別のGCTSとが、到着している。

【0053】

図14は、コンシューマが、LCTS3とLCTS4を計算したことを示す。これらのLCTSは、図12に記載された2段階法を使用して計算される。この図は、また、「...」というラベルの付いたボックスで示されるように、任意の数の追加LCTSが計算されたことを示す。

【0054】

図15は、LCTS_kが計算されたことを示す。LCTS_kは、データソースによって提供されたGCTS_kから直接計算することが出来る。

【0055】

図16は、第2のコンシューマである、データコンシューマ2が、セッションに参加したことを示す。データコンシューマ2は同期ステップを実行する。一方、データコンシューマ1は、データソースからデータを受信し続ける。

【0056】

図17は、データコンシューマ2が、データコンシューマ1が受信しつつあるデータと同じものを受信することを開始することを示している。この図は、データが、それがデータコンシューマ1に到着するよりも遅くまたは早くデータコンシューマ2に到着することがあることを示す。データコンシューマ2が、グループ消費タイムスタンプとそれに対応するデータ(例、GCTS₇₅およびVF₇₅)を受信するや否や、それは、グループの他のメンバと同期した状態でデータの消費を開始することが出来る。ストリーミングビデオのコンテキストでは、このステップは、映画の途中で映画館に入る人に似ている。

【0057】

図18は、コンシューマが、クロック同期ステップを定期的に繰り返すことを示している。クロック間のドリフトは、時間とともに変化する可能性があるため、消費のより緊密な同期は、ドリフトを定期的に再計算することにより、保証されるであろう。再同期は、全てのグループメンバに対して同時に実行する必要は無い。コンシューマは、再同期を実行している間にデータの消費を続けることが出来る。データソースがそれ自体のローカルクロックを有する場合、それは、同期ステップを定期的に繰り返す必要もある。

【0058】

図19は、各コンシューマがそれ自体のローカルクロックを有しているにもかかわらず、データが、同じグループ時刻に(例、同期した状態で)データコンシューマ1およびデータコンシューマ2によってどのように消費されるかを示す。この図では、消費されるデータは、ビデオである。フレームレートは24フレーム/秒で、かつビデオのタイムベースは90kHzである。従って、ビデオフレーム当たりのクロックティック数は、 $90,000 / 24 = 3,750$ である。この例では、グループメンバが、同期グループに参加するために正確なクロックを必要としないことを示すために、コンシューマのローカルクロックは、意図的に非現実的なものになっている。グループクロックのティックごとに、データコンシューマ1のクロックは、約0.0099回ティックする。グループ時刻 = 450,000で、データコンシューマ1のクロックとグループクロックとの間のオフセットは、449,851である。グループクロックのティックごとに、データコンシューマ2のクロックは、約0.0024回ティックする。グループ時刻 = 450,000で、データコンシューマ2のクロックとグループクロックとの間のオフセットは、449,131である。

【0059】

フレーム73は、グループ時刻 = 442,500で再生される。これは、データコンシューマ1のローカル時刻 = 75に相当する。データコンシューマ2は、まだ、参加していない。

【0060】

フレーム時刻74は、グループ時刻 = 446,250で再生される。これは、データコ

10

20

30

40

50

ンシューマ 1 のローカル時刻 = 1 1 2 に相当する。データコンシューマ 2 は、まだ、参加していない。

【 0 0 6 1 】

フレーム 7 5 は、グループ時刻 = 4 5 0 , 0 0 0 で再生される。これは、データコンシューマ 1 のローカル時刻 = 1 4 9 に相当し、かつ参加したばかりのデータコンシューマ 2 のローカル時刻 = 8 6 9 に相当する。

【 0 0 6 2 】

フレーム 7 6 は、グループ時刻 = 4 5 3 , 7 5 0 で再生される。これは、データコンシューマ 1 のローカル時刻 = 1 8 6 に相当し、かつデータコンシューマ 2 のローカル時刻 = 8 7 8 に相当する。

【 0 0 6 3 】

フレーム 7 7 は、グループ時刻 = 4 5 7 , 5 0 0 で再生される。これは、データコンシューマ 1 のローカル時刻 = 2 2 3 に相当し、データコンシューマ 2 のローカル時刻 = 8 8 7 に相当する。

【 0 0 6 4 】

図 2 0 は、本発明による、キューおよび同期されたデータ消費を示すブロック概略図である。この図の部分 A は、クライアントのアプリケーションによって直接消費されるデータを示す。この場合、クライアントのアプリケーションは、消費のタイミングと方法を完全に制御する。ビデオ再生のコンテキストでは、これは、所望の時刻にビデオフレームをレンダリングすることのみを必要とする。

【 0 0 6 5 】

図 2 0 の部分 B では、データの最終的なコンシューマは、サードパーティのアプリケーションである。この場合、クライアントのアプリケーションは、依然として、次のようにサードパーティのアプリケーションの消費のタイミングに影響を与える可能性がある。

【 0 0 6 6 】

1) クライアントのアプリケーションは、サードパーティのアプリケーションにデータを供給するタイミングを制御することが出来る。クライアントのアプリケーションは、また、データフレームを削除したり、重複したデータフレームを送信したりすることもできるが、これは、後続のフレームが、サードパーティのアプリケーションによって消費される時刻に影響を与える。

【 0 0 6 7 】

2) サードパーティのアプリケーションは、その状態に関する情報を提供し、かつ制御入力を受け入れることができる。制御入力の例には、サードパーティのアプリケーションの消費率 (例、ビデオの再生率) を調整する機能、およびデータをスキップする、またはデータストリームの特定のポイントをシークする機能がある。クライアントのアプリケーションは、サードパーティのアプリケーションによって提供される情報を使用して、適切な制御入力を計算することが出来る。これは、消費が所望の消費タイムスタンプで発生すると言う結果をもたらし、これによりデータ消費の同期が維持されるであろう。

【 0 0 6 8 】

0 0 9 1

図 2 1 A ~ 2 1 H は、本発明によるデータの優先度に基づく順序付けを示す。

【 0 0 6 9 】

図 2 1 A は、コンシューマの優先度キューを示す。コンシューマは、ローカル消費タイムスタンプ (および提供されている場合には、シーケンス番号) に基づいて、データフレームの優先度キューを維持する。遅れて到着するフレームは、破棄される、またはより高速で消費されるので、コンシューマは、グループの残りの部分に追いつくことが出来る。早く到着したチャンクは、優先度キューに配置される。この図では、優先度キューは 1 つのデータフレームを有する。

【 0 0 7 0 】

図 2 1 B は、ネットワークを介して到着する新しいデータ (ビデオフレーム 8 0) を示

10

20

30

40

50

す。データには、グループ消費タイムスタンプ (G C T S) が関連付けられている。

【 0 0 7 1 】

図 2 1 C は、コンシューマが、新しいデータフレーム (ビデオフレーム 8 0) のローカル消費タイムスタンプ (L C T S) を計算したことを示す。

【 0 0 7 2 】

図 2 1 D は、ビデオフレーム 8 0 がビデオフレーム 7 5 よりも遅い L C T S を有するので、新しいデータフレーム (ビデオフレーム 8 0) が、既存のデータフレーム (ビデオフレーム 7 5) より低い優先度で優先度キューに置かれていることを示す。

【 0 0 7 3 】

図 2 1 E は、新しいデータフレーム (ビデオフレーム 7 6) が、ネットワークから到着したことを示す。このデータフレームには、明示的な G C T S は無い。

【 0 0 7 4 】

図 2 1 F は、コンシューマが、ビデオフレーム 7 6 の L C T S を計算することを示す。これは、新しいフレームのビデオタイムスタンプを、L C T S が既知であるフレームのビデオタイムスタンプと比較し、かつクロックドリフトを考慮することにより行うことが出来る。

【 0 0 7 5 】

図 2 1 G は、ビデオフレーム 7 6 が、優先度キューに置かれようとしている状態を示す。

【 0 0 7 6 】

図 2 1 H は、ビデオフレーム 7 6 が優先度キューに置かれていることを示す。これは、L C T S が遅いビデオフレーム 8 0 よりも優先度が高い。これは、L C T S が早いビデオフレーム 7 5 よりも優先順位が低い。

【 0 0 7 7 】

図 2 2 は、本発明による複数のデータストリームの消費の同期を示す。本発明を使用して、グループ間で同期した状態で複数のデータストリームを消費し、そして個々のコンシューマごとに複数のストリームの同期を維持することが出来る。例えば、コンシューマが、ビデオデータのストリームとオーディオデータのストリームを受信しつつ、かつメディアプレーヤで両方のストリームをレンダリングしつつある場合がある。2つのデータストリームがデータソースで既に同期されている場合 (例えば、オーディオとビデオが同期した状態にある場合)、データソースによって割り当てられた消費タイムスタンプは、ビデオとオーディオがコンシューマで同期したままであることを保証するであろう。従って、全てのビデオおよび全てのオーディオデータは、全てのグループメンバー間で同期した状態で消費されるであろう。

【 0 0 7 8 】

図 2 2 は、データソースが、複数のデータストリームの同期を支援する追加情報を含む制御パケットを送信することが出来ることを示す。複数のデータストリームは、各ストリームのデータに関連付けられている消費タイムスタンプを使用して、同期させることが出来るので、この情報は、厳密には必要ではない。しかしながら、データソースが、1つのストリーム (例、ビデオ) のみに対して消費タイムスタンプを送信する方が便利な場合がある。この場合、これらの制御パケットにより、コンシューマが、他のストリーム (オーディオなど) を前述のストリーム (ビデオ) に同期させることが可能となるであろう。特定の G C T S パケットが失われたり破損したりした場合、これらの制御パケットを使用して2つのストリームを同期させることも出来る。

【 0 0 7 9 】

図 2 3 は、一実施形態による、複数のコンシューマの間でデータチャンクの消費を同期させる方法のフローチャートである。ステップ 2 3 0 0 において、1つ以上のコンシューマは、データチャンクの順序付けられたシーケンスを備える順序付けられたデータストリームを受信することが出来る。このデータチャンクは、ビデオフレーム、ビデオフレームのセクション、および / またはオーディオのセクションとすることが出来、他方この順序

10

20

30

40

50

付けられたデータストリームは、ビデオファイルまたはオーディオファイルとすることが出来る。ステップ2310では、以下のステップを使用することにより、この順序付けられたデータストリーム内のデータチャンクの消費を、コンシューマ間で同期させることが出来る。

【0080】

ステップ2320において、この順序付けられたデータストリームの消費を開始する前に、プロセッサは、これらのコンシューマがこのデータチャンクを消費すべきグループ時刻を示すグループクロックと、コンシューマのローカル時刻を示すローカルクロックとを、同期させることが出来る。このプロセッサは、このローカル時刻とこのグループ時刻の間のマッピングを取得する。このプロセッサは、図1~22に示されるように、コンシューマに関連付けることが出来る。

10

【0081】

例えば、グループクロックとローカルクロックを同期させるために、プロセッサは、このグループクロックに関連付けられている複数のグループ時刻と、このグループ時刻に対応する複数のローカル時刻とを取得することが出来る。段落0035の「クロック同期の詳細な説明」のセクションで説明されているように、ローカル時刻は、グループ時刻を受信する際のネットワークレイテンシを考慮することが出来る。例えば、ネットワークレイテンシがない場合、ローカル時刻は、このグループ時刻を受信した時刻になる。しかしながら、プロセッサは、(このグループクロックによって送信されたこのグループ時刻が、コンシューマのプロセッサに到達する移動時間を意味する)このネットワークレイテンシを測定することが出来る。この場合、ローカル時刻は、(グループ時刻を受信したローカル時刻 - グループ時刻の移動時間)として表すことが出来る。このグループ時刻とこれに対応するローカル時刻とに基づいて、このプロセッサは、式AAおよびBBに示されるこのグループクロックとこのローカルクロックとの間の線形対応を計算することが出来る。

20

【0082】

ステップ2330において、このプロセッサは、順序付けられたデータストリーム内のデータチャンクと、この順序付けられたデータストリーム内のこのデータチャンクに関連付けられているグループ時刻の表示とを、受信することが出来る。このグループ時刻の表示は、(本出願に記載されているように)データチャンクに関連付けられている実際のグループ時刻GCTSとすることが出来る。さらに、このグループ時刻の表示は、(本出願に記載されているように、それからGCTSを計算することが出来る)ビデオタイムスタンプとすることが出来る。ステップ2340では、このグループ時刻、およびローカル時刻とグループ時刻との間のこのマッピングに基づいて、このプロセッサは、このローカルクロックに関連付けられている、このグループ時刻に対応するローカル時刻を計算することが出来る。

30

【0083】

ステップ2350において、このプロセッサは、このローカル時刻とこのローカルクロックに関連付けられている現在時刻とを比較して、このデータチャンクに関連付けられているこのローカル時刻が、この現在時刻より前であるか、この現在時刻と実質的に同一であるか、またはこの現在時刻より後である否かを示す比較を得ることが出来る。ステップ2360では、この比較に基づいて、このプロセッサは、このデータチャンクを削除する、このデータチャンクの持続時間を調整する、またはこのデータチャンクを消費することにより、これらのコンシューマがこのデータチャンクを同期した状態で消費することを保証することが出来る。このデータチャンクの持続時間は、この順序付けられたデータストリームの消費率、または後続のフレームの消費時刻に基づいて、計算することが出来る。例えば、この順序付けられたデータストリームの消費率は、1秒当たりのフレーム数の様な、1秒当たりのデータチャンクの数として表すことが出来る。従って、単一のデータチャンクの持続期間は、1秒/(1秒当たりのデータチャンクの数)に対応する。別の例では、データチャンクの持続時間は、(データチャンクマイニングに関連付けられているグループ時刻が、後続のデータチャンクに関連付けられているグループ時刻である)として

40

50

計算することが出来る。

【0084】

プロセッサは、現在時刻がローカル時刻より前である（これは、後でこのデータチャンクを表示する必要があることを意味する）と決定することが出来る。プロセッサは、このデータチャンクに関連付けられているローカル時刻を示す場所に、このデータチャンクをデータチャンクの受信されたシーケンスで配置することが出来る。

【0085】

プロセッサは、現在時刻が、このローカル時刻およびこのデータチャンクの持続時間によって定義される時間範囲内にある（これは、このデータチャンクが、消費されつつあるプロセス内にあるべきであることを意味する）と決定することが出来る。プロセッサは、このデータチャンクの持続時間が、このデータチャンクの予定持続時間を超えないようにこのデータチャンクの持続時間を短縮させることが出来る。最後に、このデータチャンクを受信したこのコンシューマは、この短縮された持続時間に対応する時間の範囲で、このデータチャンクを消費することが出来る。

10

【0086】

このプロセッサは、この現在時刻がこのローカル時刻より後である、すなわち、このデータチャンクを消費する時刻が既に過ぎていると決定することが出来る。このプロセッサは、このデータチャンクを削除することが出来る。

【0087】

このプロセッサは、例えば、ネットワーク遅延またはネットワークエラーのために、以前に消費されたデータチャンクに続く第2のデータチャンクが、利用可能でないと判断することが出来る。このプロセッサは、この以前に消費されたデータチャンクの持続時間を延長して、順序付けられたデータストリームの消費が中断されることを防ぐことが出来る。

20

【0088】

このプロセッサは、ラッシュ同期以降に発生する2つのクロック間のドリフトの原因を除くために、この順序付けられたデータストリームの消費中に、このグループクロックとこのローカルクロックを同期させることが出来る。データストリームの消費中に時々同期を取ることににより、このプロセッサは、ローカル時刻とグループ時刻の間のマッピングを改善する。

30

【0089】

図24は、別の実施形態による、複数のコンシューマの間でデータチャンクの消費を同期させる方法のフローチャートである。ステップ2400では、グループクロックを、ハードウェアおよび/またはソフトウェアで定義することが出来る。例えば、このグループクロックは、物理クロックにする、またはソフトウェアアプリケーションとして実装することも出来る。このグループクロックは、順序付けられたデータストリーム内のデータチャンクが、複数のコンシューマによって消費されるタイミングを示すグループ時刻を測定することが出来る。このグループクロックは、図4～6に記載した方法で実装させることが出来る。

【0090】

ステップ2410において、プロセッサは、このグループクロックを、順序付けられたデータストリームの消費を要求するコンシューマに関連付けられている複数のローカルクロックに同期させることが出来る。同期させるために、このプロセッサは、これらのコンシューマに複数のグループ時刻を送信することが出来る。このプロセッサは、グループクロックで実装されたソフトウェアアプリケーションの命令を実行することが出来、または、このプロセッサは、ハードウェアおよび/またはソフトウェアで実装されたグループクロックと通信させることが出来る。このプロセッサは、図1、4、5に示されるサーバに関連付けることが出来る。

40

【0091】

ステップ2420において、このプロセッサは、データチャンクおよびこのデータチャ

50

ンクに関連付けられているグループ時刻を、これらのコンシューマに送信することが出来る。このプロセッサは、グループ時刻とデータチャンクを、全てのコンシューマに同期した状態で送信する、またはこのプロセッサは、データチャンクのグループ時刻を非同期に、つまり、本出願、例えば、図17に記載されているように、異なった時刻に、2つの異なるコンシューマに送信することが出来る。

【0092】

ステップ2430において、このプロセッサは、コンシューマからこのデータチャンクの到着時刻を受信することが出来る。このデータチャンクの到着時刻は、コンシューマが、ローカル時刻とグループ時刻の間のマッピングを確立することが出来るので、グループ時刻で表すことが出来る。換言すれば、このコンシューマは、このデータチャンクが到着したローカル時刻（すなわち、到着時刻）を知っているため、この到着時刻に対応するグループ時刻を計算することが出来る。このコンシューマは、この到着時刻に対応するグループ時刻をプロセッサに送信することが出来る。

10

【0093】

ステップ2440において、このデータチャンクの到着時刻およびこのデータチャンクに関連付けられているグループ時刻に基づいて、このプロセッサは、コンシューマがかなりの割合のデータチャンクを遅れて既に受信しているか否かを決定することが出来る。実質的な割合は10%以上とすることが出来る。このデータチャンクの到着時刻がこのグループ時刻より後の場合、このプロセッサは、コンシューマがこのデータチャンクを遅れて既に受信していると注記することが出来る。

20

【0094】

例えば、データチャンクの10%以上が遅れて受信されている場合、ステップ2450において、このプロセッサは、本出願に記載されているように、順序付けられたデータストリームのデータチャンクの品質またはレイテンシを調整することにより、これらのコンシューマがこの順序付けられたデータストリームを同期した状態で消費することを保証することが出来る。

【0095】

このデータチャンクの品質を調整するために、このプロセッサは、このデータチャンクに関連付けられている最小品質閾値を指定する設定を取得することが出来る。例えば、この品質閾値は、ビデオのビットレートが、毎秒2.5メガビットより大でなければならぬと指定することが出来る。

30

【0096】

このプロセッサは、データチャンクの品質を最小品質閾値まで低下させることにより、コンシューマが、実質的な割合のデータチャンクを遅れて受信することが、防止されるか否かを、決定することが出来る。例えば、このプロセッサは、ビデオのビットレートを最低品質に下げることが出来、そしてコンシューマから収集した到着時刻に基づいて、かなりの割合のデータチャンクをまだ受信していないコンシューマが、存在するか否かを決定することが出来る。別の例では、このプロセッサは、例えば、

ネットワークスループット = (コンシューマに配信されるデータ) / (到着時刻 - 対応するグループ時刻)

40

を計算することにより、到着時刻に対応するグループ時刻に基づいて、最低のネットワークスループットを決定することが出来る。

【0097】

このネットワークスループットが、最小ビットレート以上である場合、このプロセッサは、入力品質を低下させることが出来る。具体的には、このプロセッサは、入力品質を、ネットワークスループット以下のビットレートに低下させ、そしてこのデータチャンクを、かなりの割合のデータチャンクを遅れて受信しているコンシューマに送信することが出来る。残りのコンシューマは、より高品質のデータチャンクを受信することが出来る。

【0098】

50

このネットワークスループットが、最小ビットレートよりも低い場合、このプロセッサは入力品質を低下させることは出来ない。代わりに、このプロセッサは、このデータチャンクに関連付けられるレイテンシを増加させることが出来る。具体的には、このプロセッサは、このデータチャンクに関連付けられているグループ時刻を増大させることが出来るので、このグループ時刻は、このデータチャンクを最も遅くに受信するコンシューマにおけるこのデータチャンクの到着時刻と同じまたはそれよりわずかに遅れる時刻となる。このようにして、このプロセッサは、各コンシューマがデータチャンクを消費すると同時に、このコンシューマが最新の時刻にこのデータチャンクを受信することを保障することが出来る。この結果、このプロセッサは、これらのコンシューマ間の消費の同期を保障することが出来る。

10

【0099】

このプロセッサは、これらのコンシューマがこのデータストリームを消費している間に、これらのコンシューマに加わる新しいコンシューマから要求を受信することが出来る。このプロセッサは、新しいコンシューマに、第2のデータチャンクと、この第2のデータチャンクに関連付けられている第2のグループ時刻を送信することが出来る。この第2のグループ時刻は、現在これらのコンシューマに送信されつつあるグループ時刻よりも遅くすることが出来る。

【0100】

コンピュータシステム

図25は、いくつかの実施形態の特定の特徴を実装するために使用することができるコンピュータシステムのブロック図である。このコンピュータシステムは、サーバコンピュータ、クライアントコンピュータ、パーソナルコンピュータ(PC)、ユーザーデバイス、タブレットPC、ラップトップコンピュータ、携帯情報端末(PDA)、携帯電話、iPhone、iPad、BlackBerry、プロセッサ、電話、Webアプリケーション、ネットワークルーター、スイッチまたはブリッジ、コンソール、ハンドヘルドコンソール、(ハンドヘルド)ゲームデバイス、音楽プレーヤ、任意のポータブルでモバイルのハンド-ハンドヘルドデバイス、ウェアラブルデバイス、またはそのマシンが実行するアクションを指定するシーケンシャルまたはそれ以外の一連の命令を実行することが可能な任意のマシンとすることができる。

20

【0101】

このコンピューティングシステム2500は、1つまたは複数の中央処理装置(「プロセッサ」)2505、メモリ2510、入/出力デバイス2525(例、キーボードおよびポインティングデバイス、タッチデバイス、ディスプレイデバイス)、ストレージデバイス2520(例、ディスクドライブ)およびネットワークアダプタ2530(例、相互接続2515に接続されるネットワークインターフェース)を含むことができる。この相互接続2515は、1つ以上の個別の物理バス、ポイントツーポイント接続、または適切なブリッジ、アダプタ、またはコントローラーによって接続された両方、を表す抽象概念として示されている。従って、相互接続2515は、例えば、システムバス、周辺機器相互接続(PCI)バスまたはPCI-Expressバス、HyperTransportまたは業界標準アーキテクチャ(ISA)バス、小型コンピュータシステムインターフェース(SCSI)バスを含むことができる。ユニバーサルシリアルバス(USB)、IIC(I2C)バス、または米国電気電子学会(IEEE)標準の1394バス(FireWireとも呼ばれる)を含むことができる。

30

40

【0102】

メモリ2510および記憶装置2520は、様々な実施形態の少なくとも一部を実施する命令を記憶することができるコンピュータ可読記憶媒体である。加えて、このデータ構造およびメッセージ構造は、データ送信媒体(例、通信リンク上の信号)を介して記憶させまたは転送させることができる。様々な通信リンク(例、インターネット、ローカルエリアネットワーク、ワイドエリアネットワーク、またはポイントツーポイントダイヤルアップ接続)を使用することができる。従って、コンピュータ可読媒体は、コンピュータ可

50

読記憶媒体、例えば、を含むことが出来る。非一時的なメディア、およびコンピュータで読み取り可能な伝送メディアを含むことが出来る。

【0103】

メモリ2510に記憶された命令は、ソフトウェアおよび/またはファームウェアとして実装させて、プロセッサ2505が上述の動作を実行するようにプログラムすることが出来る。いくつかの実施形態では、このようなソフトウェアまたはファームウェアは、コンピューティングシステム2500を介して(例、ネットワークアダプタ2530経由で)リモートシステムからそれをダウンロードすることにより、先ず、コンピューティングシステム2500に提供させることができる。

【0104】

本明細書で紹介された様々な実施形態は、例えば、プログラマブル回路、例えば、ソフトウェアおよび/またはファームウェアでプログラムされた1つ以上のマイクロプロセッサ、または完全専用ハードワイヤード(プログラム不可)回路で、またはそのような形式の組み合わせで実装させることができる。専用ハードワイヤード回路は、例えば、1つまたは複数のASIC、PLD、FPGAなどの形式とすることができる。

10

【0105】

備考

本明細書で使用される言語は、主に読み易さおよび説明目的のために選択されており、そして本発明の主題を描写または制限するために選択されていない場合がある。従って、本発明の範囲は、この詳細な説明によって限定されるのではなく、本明細書に基づく出願に基づいて発行される請求項によって限定されることが、意図されている。従って、様々な実施形態の開示は、添付の特許請求の範囲に記載されている実施形態の範囲を例示することを意図したものであり、限定することを意図したものではない。

20

【0106】

優先権の主張

本出願は、2017年4月24日に出願された米国仮特許出願第62 / 489, 264号の優先権を主張し、その全体が参照により本明細書に組み込まれる。

【 図 1 】

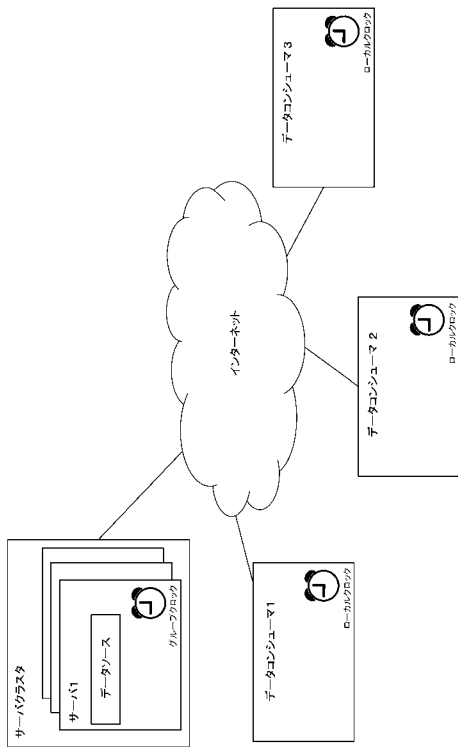


図1

【 図 3 】

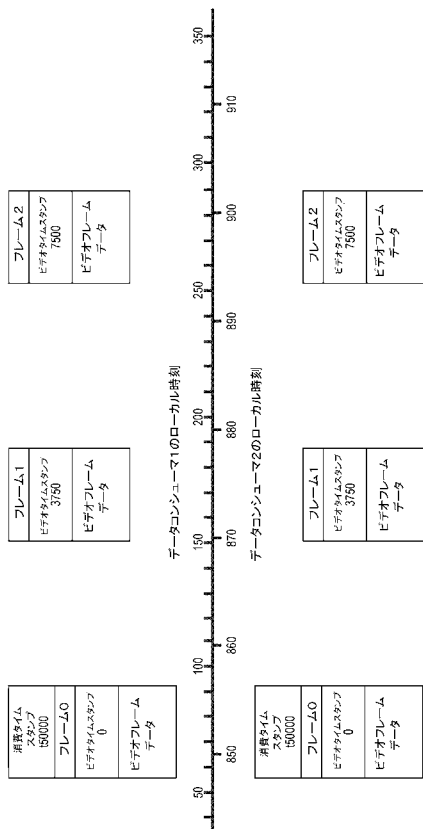


図3

【 図 2 】



図2

【 図 4 】

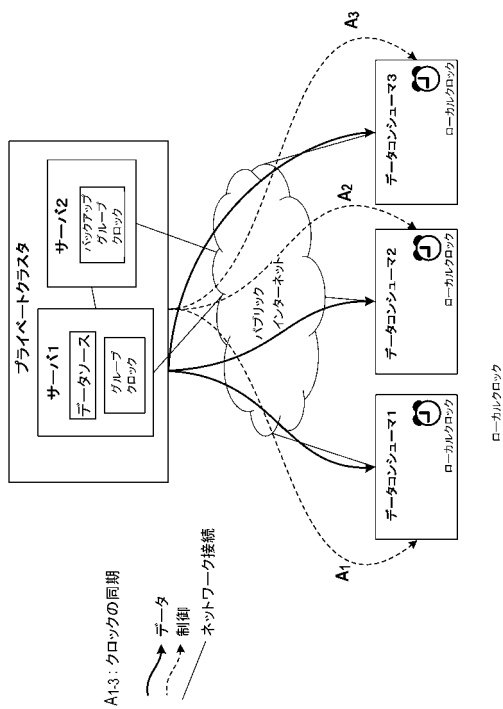
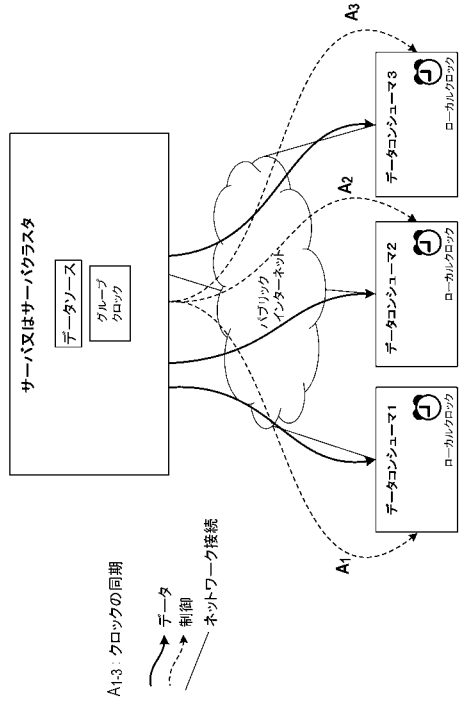


図4

【 図 5 】



【 図 6 】

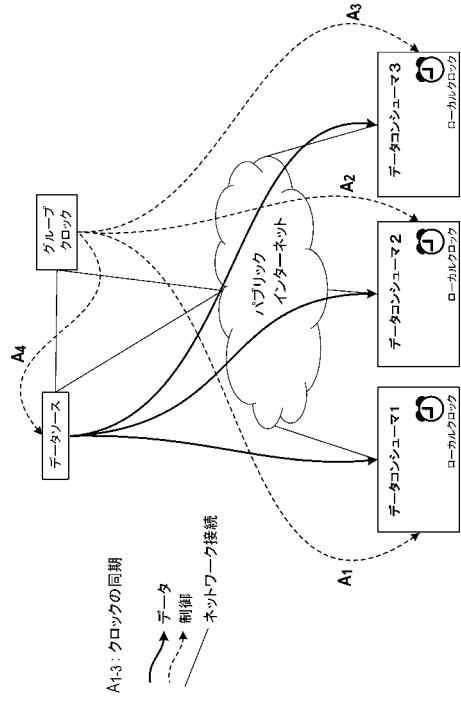
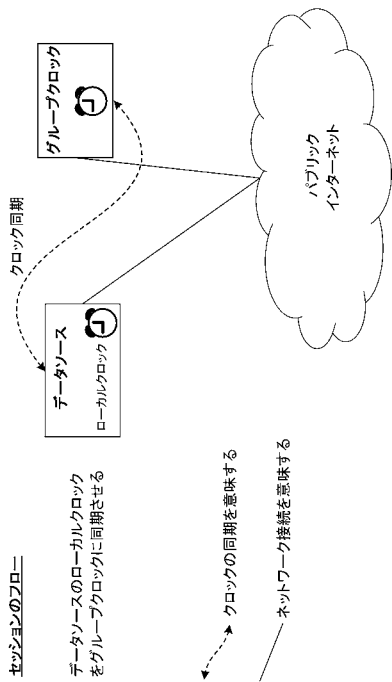


図5

図6

【 図 7 】



【 図 8 】

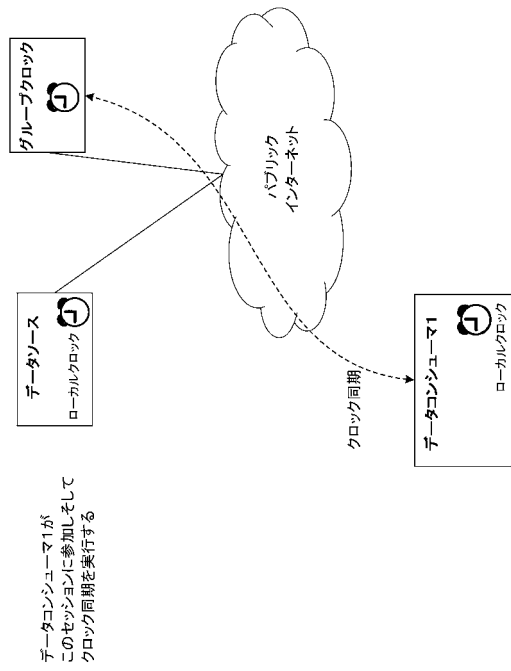
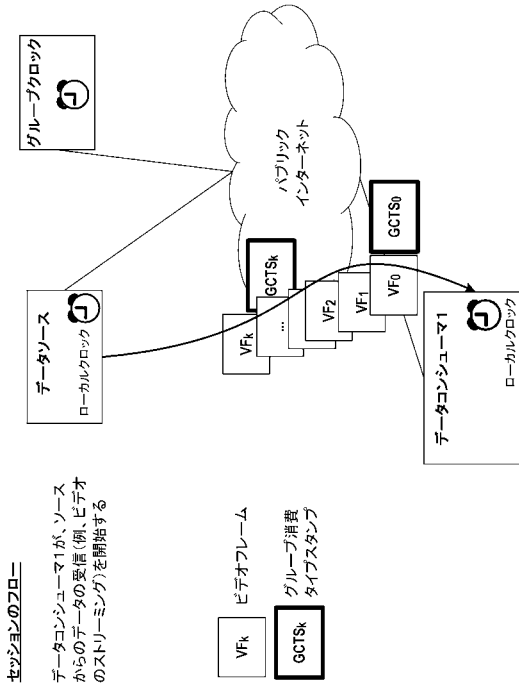


図7

図8

【 図 9 】



【 図 10 】

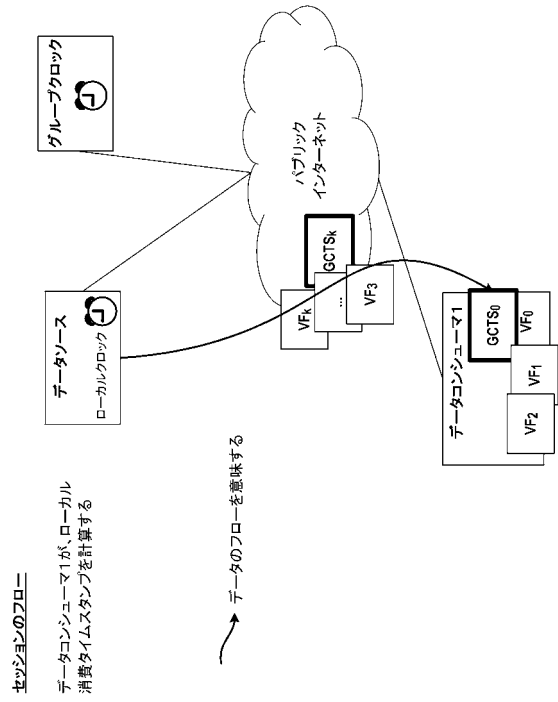


図9

【 図 11 】

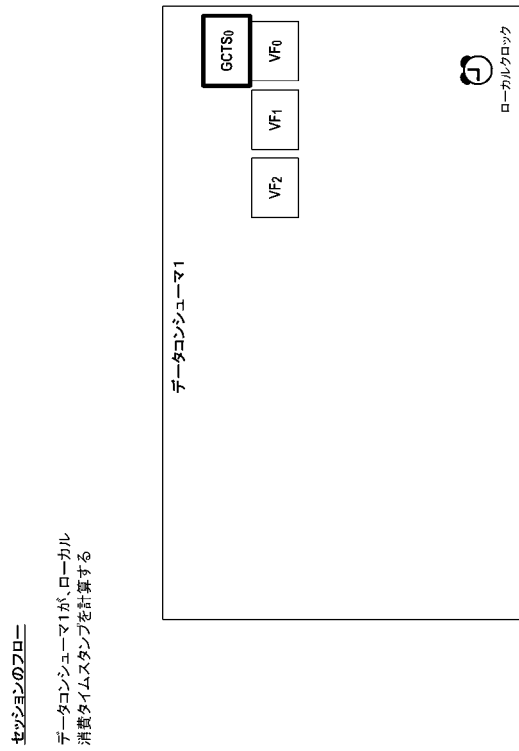


図11

図10

【 図 12 】

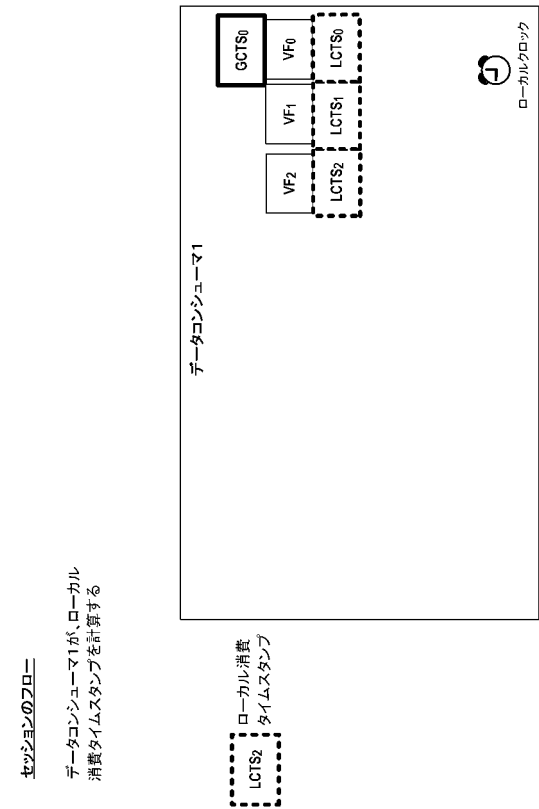


図12

図13

【 図 1 3 】

セッションのプロセ

データコンシューマ1が、ローカル消費タイムスタンプを計算する

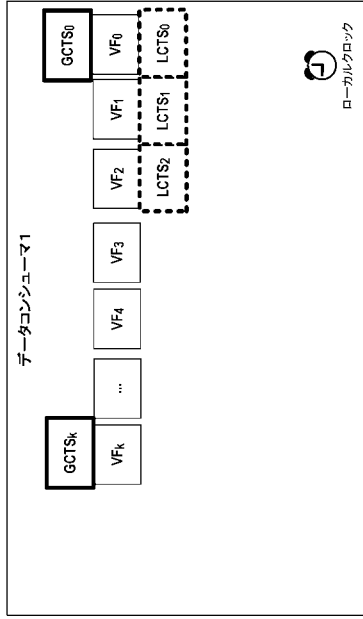


図15

【 図 1 5 】

セッションのプロセ

データコンシューマ1が、ローカル消費タイムスタンプを計算する

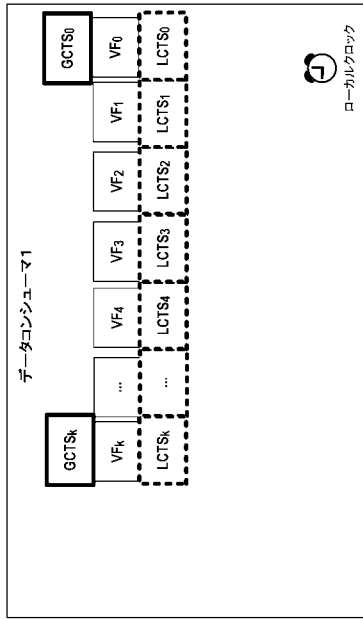


図14

【 図 1 4 】

セッションのプロセ

データコンシューマ1が、ローカル消費タイムスタンプを計算する

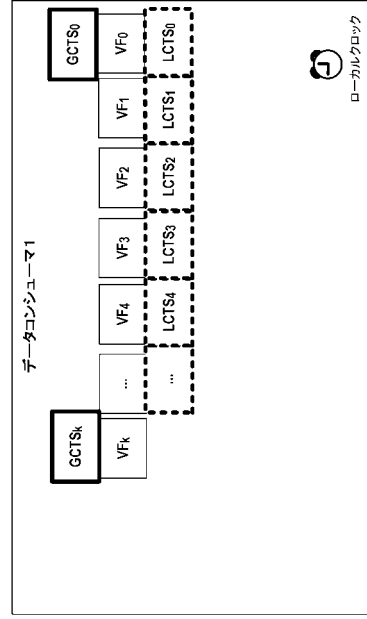
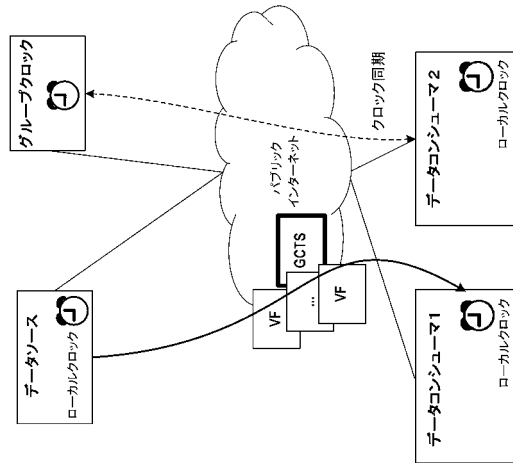


図16

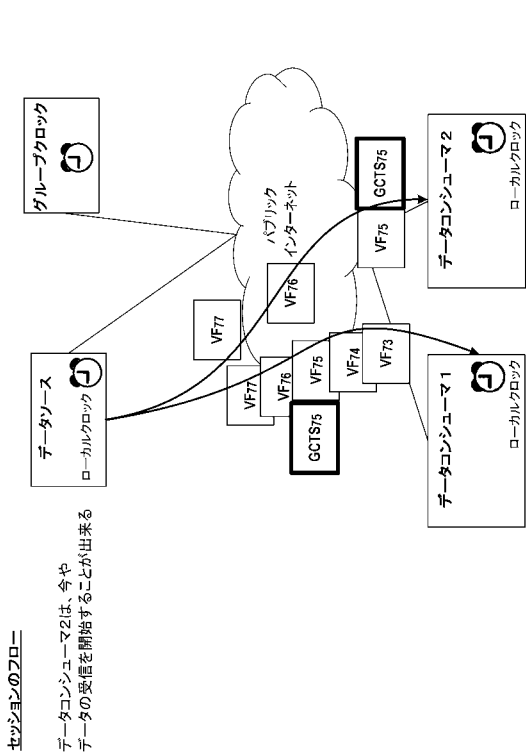
【 図 1 6 】

セッションのプロセ

データコンシューマ2がこのセッションに参加し、そのクロック同期を実行する



【 図 17 】



【 図 18 】

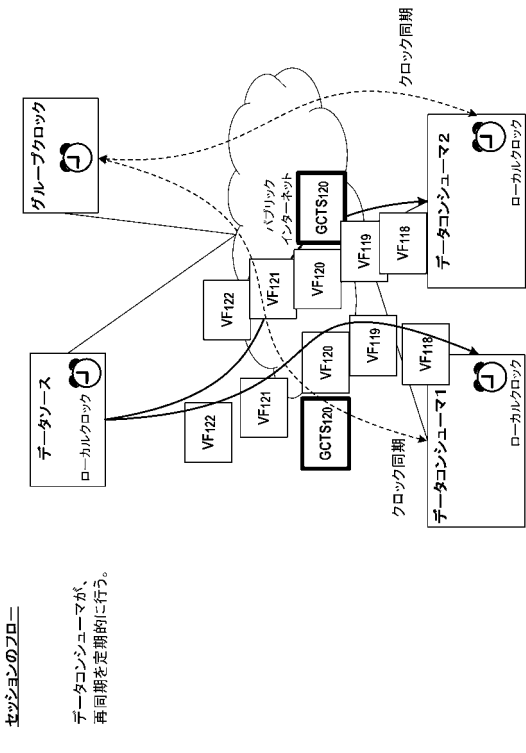


図17

【 図 19 】

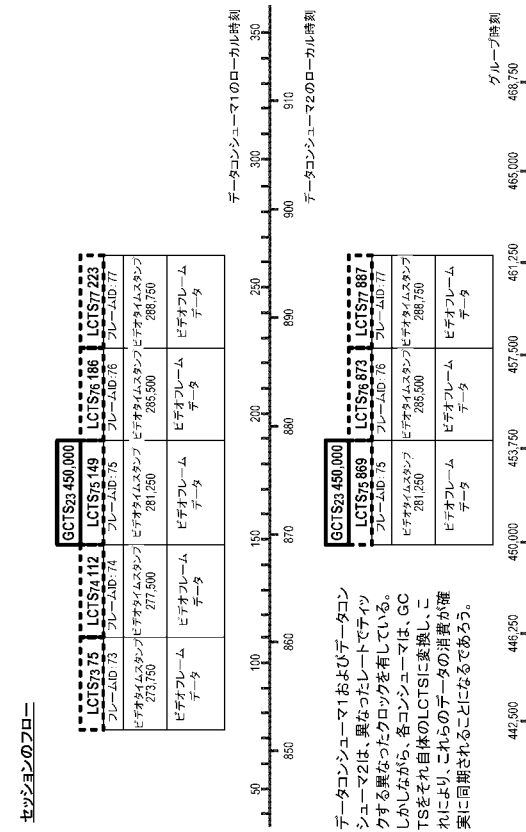


図19

図18

【 図 20 】

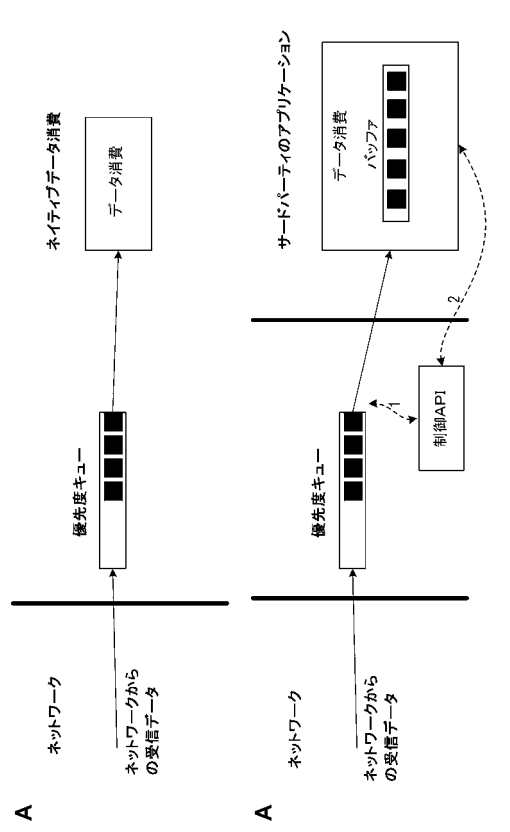
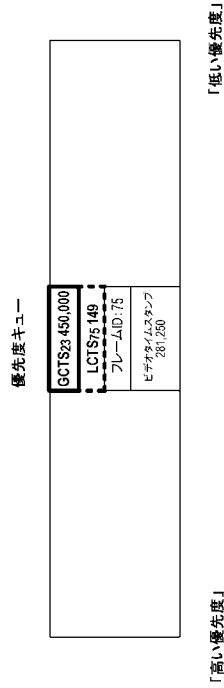


図20

【 図 2 1 A 】

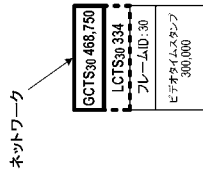
図21A

この例では、幾つかのデータコンシューマの優先度キューは、データソースにより供給されたGCTSを有する1つのビデオフレームを含み、そしてこのデータコンシューマは、既に対応するLCTSを計算している。

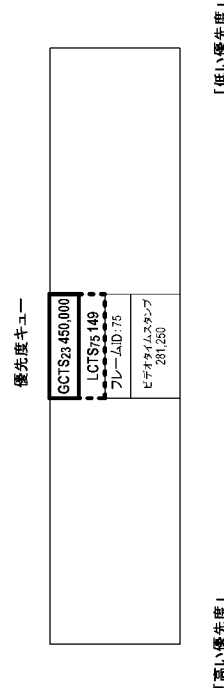


【 図 2 1 C 】

図21C

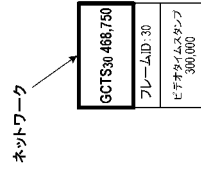


このLCTSが計算される。

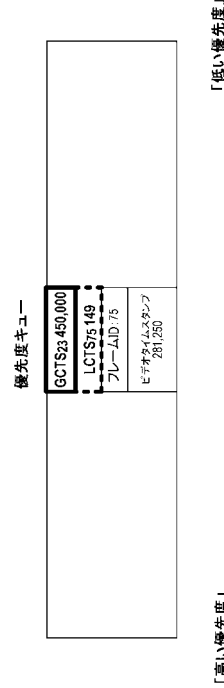


【 図 2 1 B 】

図21B



新しいビデオフレームが、GCTSと共に到着する。

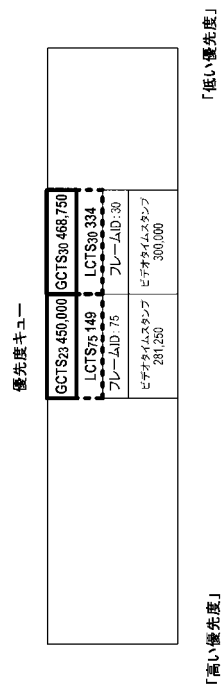


【 図 2 1 D 】

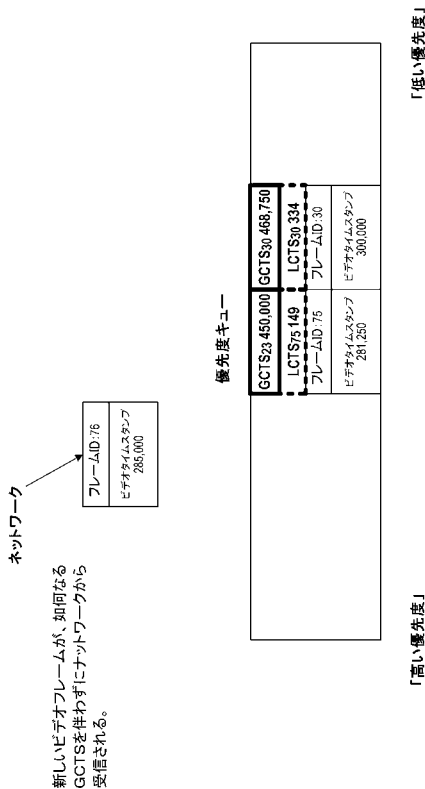
図21D

ネットワーク

このフレームが優先度キュー内に配置される。



【 図 2 1 E 】



【 図 2 1 F 】

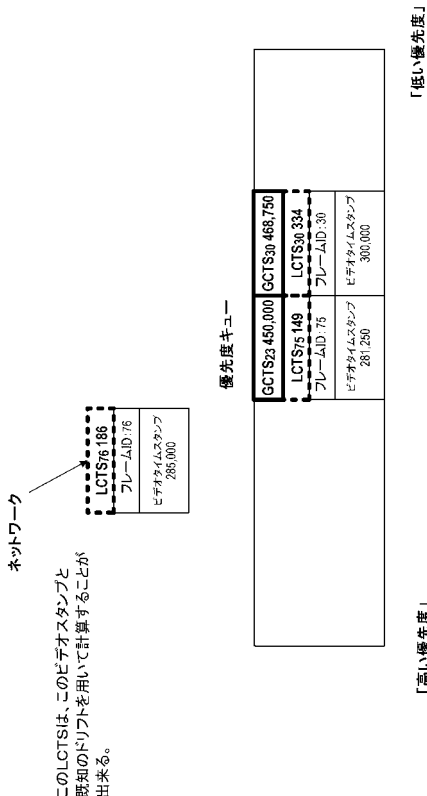
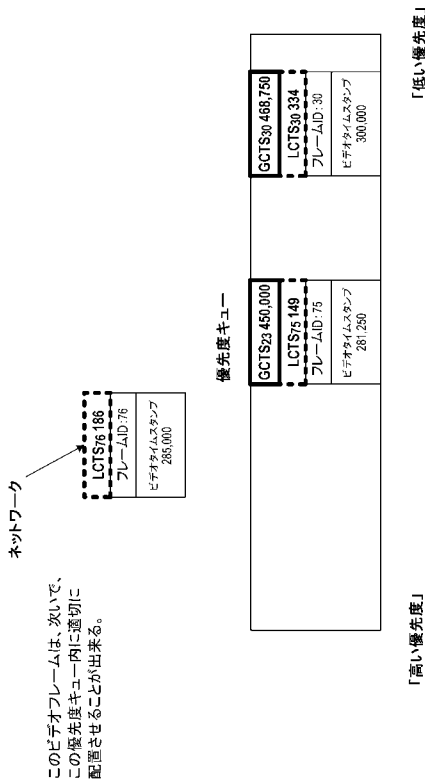


図21E

図21F

【 図 2 1 G 】



【 図 2 1 H 】

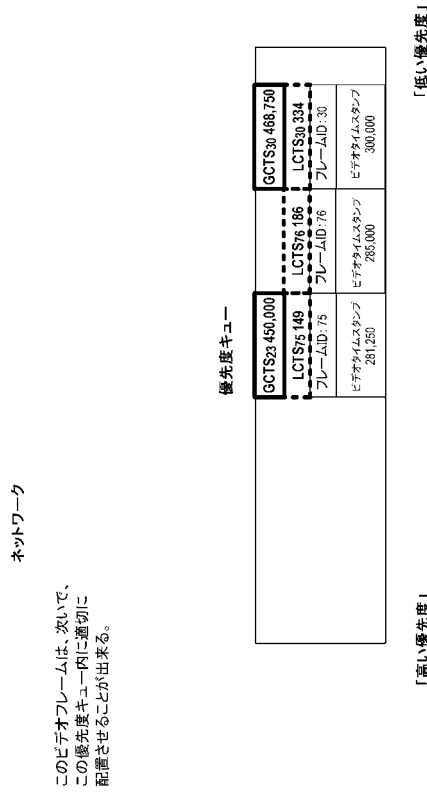
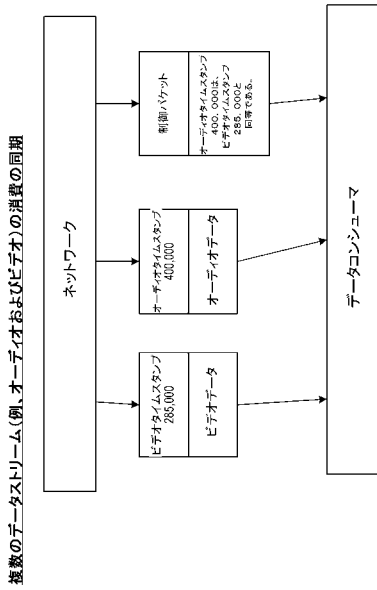


図21G

図21H

【 図 2 2 】



【 図 2 3 】

図23

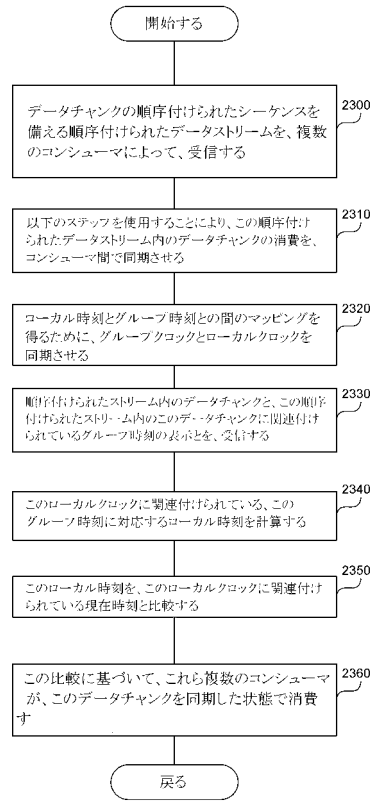
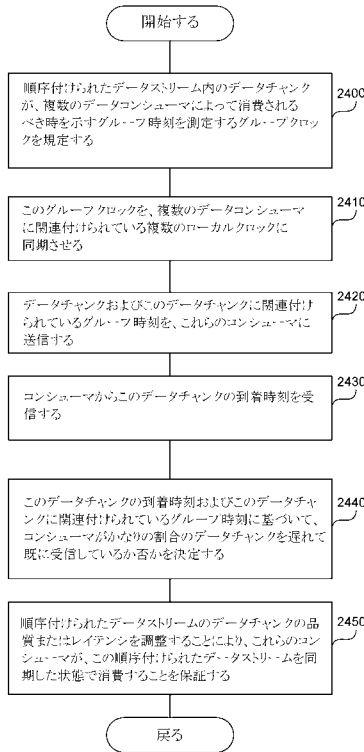


図22

【 図 2 4 】

図24



【 図 2 5 】

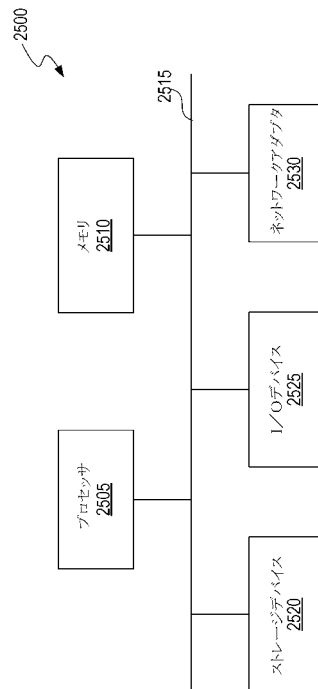


図25

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US2018/026756
A. CLASSIFICATION OF SUBJECT MATTER IPC(8) - H04N 7/173; H04N 7/025; H04N 7/16 (2018.01) CPC - H04N 7/173; H04N 21/4312; H04N 21/4788; G11B 27/105; H04N 7/15 (2018.05)		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) See Search History document		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched USPC - 725/1; 725/5; 725/34; 725/87; 725/105 (keyword delimited)		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) See Search History document		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2009/0044216 A1 (MCNICOLL) 12 February 2009 (12.02.2009) entire document	1-20
A	US 2010/0177708 A1 (PANDEY et al) 15 July 2010 (15.07.2010) entire document	1-20
A	US 2006/0056337 A1 (YOON) 16 March 2006 (16.03.2006) entire document	1-8, 14-20
A	US 2013/0014200 A1 (STOKKING et al) 10 January 2013 (10.01.2013) entire document	9-13
A	US 2011/0154417 A1 (CIVANLAR et al) 23 June 2011 (23.06.2011) entire document	1-20
A	VINGELMANN et al. "Synchronized multimedia streaming on the iphone platform with network coding." In: IEEE Communications Magazine. 07 June 2011 (07.06.2011) Retrieved from <http://kom.aau.dk/~if/documents/ccnc2011.pdf> entire document	1-20
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 11 June 2018		Date of mailing of the international search report 16 JUL 2018
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, VA 22313-1450 Facsimile No. 571-273-8300		Authorized officer Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

フロントページの続き

(81)指定国・地域 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT

(特許庁注：以下のものは登録商標)

- 1 . i P h o n e
- 2 . i P a d
- 3 . B L A C K B E R R Y
- 4 . F I R E W I R E

(72)発明者 ブスタマンテ , ファビアン
アメリカ合衆国 60611 イリノイ州 シカゴ , ノース ワバシ アベニュー 330 , フ
ロア 23

(72)発明者 ウェイナー , アンドリュー ジョセフ
アメリカ合衆国 60611 イリノイ州 シカゴ , ノース ワバシ アベニュー 330 , フ
ロア 23

Fターム(参考) 2F002 FA16 GA06
5C164 MB44S SB29S TA08S UB04P UB21S UB41S YA21
5K047 AA18 GG56 HH52