(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2016-524760
(P2016-524760A)
(43)公表日　平成28年8月18日(2016.8.18)

(51)Int.Cl.		F I		テーマコード（参考）
G06F 9/50	(2006.01)	G06F	9/46	462Z
G06F 9/46	(2006.01)	G06F	9/46	350
G06F 12/00	(2006.01)	G06F	12/00	535B
		G06F	12/00	545A

審査請求 有　予備審査請求 未請求　（全 34 頁）

(21)出願番号　特願2016-518066 (P2016-518066)
(86)(22)出願日　平成26年6月10日 (2014.6.10)
(85)翻訳文提出日　平成27年12月7日 (2015.12.7)
(86)国際出願番号　PCT/US2014/041724
(87)国際公開番号　WO2014/201012
(87)国際公開日　平成26年12月18日 (2014.12.18)
(31)優先権主張番号　13/914,104
(32)優先日　平成25年6月10日 (2013.6.10)
(33)優先権主張国　米国 (US)

(71)出願人　507303550
アマゾン・テクノロジーズ・インコーポレーテッド
アメリカ合衆国・８９５０７・ネバダ州・レノ・ピーオーボックス　８１０２
(74)代理人　100064621
弁理士　山川　政樹
(74)代理人　100098394
弁理士　山川　茂樹
(72)発明者　ジェンキンス，ジョージ，オリヴァー
アメリカ合衆国・９８１０９－５２１０・ワシントン州・シアトル・テリー　アヴェニュ　ノース・４１０

最終頁に続く

(54)【発明の名称】クラウドコンピューティング環境における分散ロック管理

(57)【要約】
分散ロックマネージャ（ＤＬＭ）は、仮想化されたコンピューティングリソース及び／または仮想コンピューティングサービスをクライアントに提供する分散コンピューティングシステムにおいて実装することができる。ロックは、クライアントからの作成及び管理を実行する要求に応答して、ＤＬＭによって作成かつ管理することができる。ＤＬＭの構成要素は、クライアントアプリケーション構成要素が相互に通信するか、またはロックによって保護された共有リソースにアクセスする以外のネットワーク上で相互に通信することができる。例えば、ＤＬＭ構成要素は、クラウドコンピューティング環境の制御プレーンネットワーク上で通信することができ、アプリケーション構成要素は、クラウドコンピューティング環境のデータプレーンネットワーク上で通信することができる。ＤＬＭは、クライアントにＡＰＩを公開することができ、クライアントが、様々なロック管理操作を実施するために同じノード上のＤＬＭ構成要素へのローカル呼び出しを行うことを可能にする。ロック値の意味は、クライアントアプリケーションにおけるそれらの使

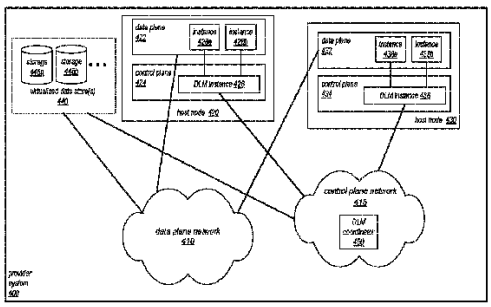


FIG. 4

【特許請求の範囲】
【請求項１】
ネットワーク上で相互に連結され、かつ仮想コンピューティングサービスを１つ以上のクライアントに集合的に提供する、複数のコンピューティングノードであって、前記コンピューティングノードの各々が、少なくとも１つのプロセッサとメモリとを備える、複数のコンピューティングノードと、
各々が前記コンピューティングノードのうちの２つ以上のそれぞれのコンピューティングノード上で実行している、２つ以上の仮想コンピュートインスタンスであって、前記仮想コンピュートインスタンスの各々が、クライアントのために分散アプリケーションのアプリケーション構成要素を実装するように構成され、前記仮想コンピュートインスタンスが、前記ネットワークの少なくとも一部上で相互に通信するように構成される、２つ以上の仮想コンピュートインスタンスと、
各々が前記２つ以上のコンピューティングノードのそれぞれのコンピューティングノード上で実行している、２つ以上の構成要素を含む分散ロックマネージャであって、前記分散ロックマネージャ構成要素が、それぞれの共有リソースの１つ以上のロックの状態を共有するように構成され、前記１つ以上のロックの状態を共有するために、前記分散ロックマネージャ構成要素が、前記仮想コンピュートインスタンスが相互に通信する前記ネットワークの前記少なくとも一部とは物理的または論理的に別個のネットワーク上で相互に通信するように構成される、分散ロックマネージャと、を備え、
前記アプリケーション構成要素のうちの１つが、前記分散アプリケーションによってアクセスされる共有リソースのロックのためのロック管理操作の実施を開始するために、前記アプリケーション構成要素を実装する前記仮想コンピュートインスタンスを実行している前記コンピューティングノード上で実行している前記分散ロックマネージャへの呼び出しを起動するように構成される、システム。
【請求項２】
前記分散ロックマネージャ構成要素が相互に通信する前記ネットワークが、サービスプロバイダコンピューティング環境の制御プレーンネットワークを備え、前記仮想コンピュートインスタンスが相互に通信する前記ネットワークの前記少なくとも一部が、前記サービスプロバイダコンピューティング環境のデータプレーンネットワークを備える、請求項１に記載の前記システム。
【請求項３】
前記分散ロックマネージャ構成要素への前記呼び出しに応答して、前記分散ロックマネージャ構成要素が、
前記ロック管理操作を実施し、かつ
前記分散ロックマネージャ構成要素が実行している前記コンピューティングノード以外のコンピューティングノード上で実行している少なくとも１つの分散ロックマネージャ構成要素へ前記ロックの結果として生じた状態を通信するように構成される、請求項１に記載の前記システム。
【請求項４】
前記２つ以上のコンピューティングノードが、前記クライアントのための仮想プライベートネットワークを実装する、請求項１に記載の前記システム。
【請求項５】
１つ以上のコンピュータによって、
複数のコンピューティングノードのうちの所与のコンピューティングノード上で実行している分散ロックマネージャの構成要素によって、共有リソースのロックのためのロック管理操作を実施する要求を受信することであって、前記受信することが、前記所与のコンピューティングノード上で実行しているコンピュートインスタンスから前記要求を受信することを含み、前記コンピュートインスタンスが、前記コンピュートインスタンスに割り当てられたネットワークリソース能力を使用して前記共有リソースにアクセスする、受信することを実施することと、

前記要求されたロック管理操作を実施することであって、前記実施することが、前記分散ロックマネージャ構成要素が、前記コンピュートインスタンスに割り当てられた前記ネットワークリソース能力以外のネットワークリソース能力を使用して、前記ロックの状態情報を共有するために、前記複数のコンピューティングノードのうちの別のコンピューティングノード上で実行している別の分散ロックマネージャ構成要素と通信することを含む、実施することと、を含む、方法。

【請求項６】

前記コンピュートインスタンスが、分散アプリケーションのアプリケーション構成要素を実装し、

前記方法が、前記アプリケーション構成要素が、前記コンピュートインスタンスに割り当てられた前記ネットワークリソース能力を使用して、前記分散アプリケーションの少なくとも１つの他の構成要素と通信することをさらに含む、請求項５に記載の前記方法。

【請求項７】

前記複数のコンピューティングノードが、１つ以上の仮想コンピューティングサービスを実装する、請求項５に記載の前記方法。

【請求項８】

前記ロック管理操作を前記実施することが、前記ロックを前記共有リソースと関連付けることを含み、前記通信することが、前記他の分散ロックマネージャ構成要素へ前記関連付けを通信することを含む、請求項５に記載の前記方法。

【請求項９】

前記ロック管理操作を前記実施することが、前記ロックの値を変更することを含む、請求項５に記載の前記方法。

【請求項１０】

前記ロックの前記値が、前記共有リソースの所有者を識別する、請求項９に記載の前記方法。

【請求項１１】

前記分散ロックマネージャの構成要素が、前記ロックの前記変更された値を永続的データストアに書き込むことをさらに含む、請求項９に記載の前記方法。

【請求項１２】

前記ロック値を前記変更することが、前記ロック値をアトミックに変更することを含む、請求項９に記載の前記方法。

【請求項１３】

前記コンピュートインスタンスに割り当てられた前記ネットワークリソース能力が、ネットワーク接続リソースまたは入出力処理能力を含む、請求項５に記載の前記方法。

【請求項１４】

前記実施することが、前記所与のコンピューティングノードの前記ロックの状態情報をキャッシュすることをさらに含む、請求項５に記載の前記方法。

【請求項１５】

１つ以上のプロセッサと、

１つまたはメモリであって、前記１つ以上のプロセッサ上で実行されると、前記１つ以上のプロセッサに、

複数のコンピューティングノードのうちの所与のコンピューティング上で実行している分散ロックマネージャの構成要素によって、共有アクセスが制御されることになる実体と関連付けられたロックのためのロック管理操作を実施する要求を受信することであって、前記受信することが、前記所与のコンピューティングノード上で実行しているリソースインスタンスから、前記要求を受信することを含み、前記複数のコンピューティングノードが、分散ロックサービスを実装し、前記リソースインスタンスが、前記ロックと関連付けられた前記実体にアクセスし、前記要求が、前記分散ロックマネージャによってサポートされた１つ以上のロック管理操作を定義するアプリケーションプログラミングインターフェースに準拠する、受信することと、

前記要求されたロック管理操作を実施することであって、前記実施することが、前記分散ロックマネージャ構成要素が、前記リソースインスタンスが前記ロックと関連付けられた前記エンティティにアクセスするネットワーク接続性以外のネットワーク接続性を使用して、前記ロックの状態情報を共有するために、前記複数のコンピューティングノードのうちの別のコンピューティングノード上で実行している前記分散ロックマネージャの別の構成要素と通信することを含む、実施することと、を実施させるプログラム命令を記憶している、１つ以上のメモリと、を備える、システム。

【発明の詳細な説明】

【技術背景】

【０００１】

多数の企業及び他の組織は、（例えば、ローカルネットワークの一部として）共存している、またはそうではなく、（例えば、１つ以上のプライベートまたは公共中間ネットワークを介して接続された）複数の別個の地理的場所に存在しているコンピューティングシステムを用いる等、その事業活動を支援するために無数のコンピューティングシステムを相互接続するコンピュータネットワークを運用する。例えば、単一の組織によりかつそのために運用されるプライベートデータセンター、及びコンピューティングリソースを顧客またはクライアントに提供するために事業として事業体により運用される公共データセンター等、相当数の相互接続されたコンピューティングシステムを収容するデータセンターが一般的になってきている。いくつかの公共データセンター運用者は、様々なクライアントによって所有されるハードウェアのためのネットワークアクセス、電力、及び安全な設置施設を提供する一方、他の公共データセンター運用者は、そのクライアントによる使用のために利用可能になったハードウェアリソースも含む「フルサービス」の施設を提供する。しかしながら、典型的なデータセンターの規模及び範囲が増加したことに伴い、物理的コンピューティングリソースをプロビジョニング、管理、及び運営するタスクはますます複雑になっている。

【０００２】

汎用ハードウェアの仮想化技術の出現は、様々なニーズを擁する多数のクライアントのための大規模コンピューティングリソースを運営する点で利点をもたらし、様々なコンピューティングリソースが、複数のクライアントによって効率的かつ安全に共有されることを可能にしている。例えば、仮想化技術は、単一の物理的コンピューティングマシンによってホストされた１つ以上の仮想マシンを各ユーザに提供することによって、単一の物理的コンピューティングマシンが、複数のユーザの間で共有されることを可能にすることができ、各々のそのような仮想マシンは、ユーザが所与のハードウェアコンピューティングリソースの唯一の運用者かつ管理者であるという錯覚をユーザに提供すると同時に、様々な仮想マシン間でアプリケーションの分離及びセキュリティも提供する、別個の論理的コンピューティングシステムとして作用するソフトウェアシミュレーションである。その上、いくつかの仮想化技術は、複数の別個の物理的コンピューティングシステムにまたがる、複数の仮想プロセッサを備える単一の仮想マシン等、２つ以上の物理的リソースにまたがる仮想リソースを提供することが可能である。別の例として、仮想化技術は、複数のデータ記憶装置にわたって分散され得る仮想化されたデータストアを各ユーザに提供することによって、データストレージハードウェアが複数のユーザの間で共有されることを可能にすることができ、各々のそのような仮想化されたデータストアは、ユーザがデータストレージリソースの唯一の運用者かつ管理者であるという錯覚をユーザに提供する別個の論理的データストアとして作用する。

【０００３】

分散コンピューティングをサポートするいくつかのシステムにおいて、分散アプリケーションのアプリケーション構成要素またはプロセスは（時々）、様々なタイプの共有リソースにアクセスすることができる。いくつかのそのようなシステムにおいて、それらの共有リソースへのアクセスを制御及び／または同期するために、分散ロックマネージャが使用される。

【図面の簡単な説明】
【０００４】
【図１】仮想コンピューティングシステムを提供する分散コンピューティングシステムにおいて分散ロックマネージャを提供するための方法の一実施形態を示すフロー図である。
【図２】少なくともいくつかの実施形態に従う、例示的プロバイダネットワーク環境を示すブロック図である。
【図３】いくつかの実施形態に従う、例示的データセンターを示すブロック図である。
【図４】いくつかの実施形態に従う、制御プレーンにおいて分散ロックマネージャを実装する例示的データセンターを示すブロック図である。
【図５】仮想化されたリソースを提供する分散コンピューティングシステムにおいて分散ロックマネージャを実装するための方法の一実施形態を示すフロー図である。
【図６】クライアントアプリケーションまたはプロセスが、クラウドコンピューティング環境において実装される分散ロックマネージャの機能にアクセスするための方法の一実施形態を示すフロー図である。
【図７】いくつかの実施形態に従う、制御プレーンにおいて分散ロックマネージャを実装し、その環境の外側で実行しているプロセスに分散ロックマネージャを公開する例示的データセンターを示すブロック図である。
【図８】クラウドコンピューティング環境の外側で実行するクライアントアプリケーションまたはプロセスが、クラウドコンピューティング環境内に実装される分散ロックマネージャの機能にアクセスするための方法の一実施形態を示すフロー図である。
【図９】異なる実施形態に従う、本明細書に記載される技法の一部または全てを実装する例示的コンピュータシステムを示すブロック図である。
【発明を実施するための形態】
【０００５】
本明細書において、実施形態はいくつかの実施形態及び例示的図面によって説明されるが、当業者は、実施形態が説明される実施形態または図面に限定されないことを認識するであろう。図面及びそれらに対する詳細説明は、開示される特定の形式に実施形態を限定することを意図せず、しかし対照的に、その意図は、添付の請求項によって定義される趣旨及び範囲に含まれる全ての変形、同等物、及び代替を網羅することであることを理解されたい。本明細書に使用される見出しは、構成上の目的のみであり、記載または請求項の範囲を限定するために使用されることを意図するものではない。本明細書全体で使用されるように、「～することができる」（ｍａｙ）は、必須の意味（すなわち、～しなければならない（ｍｕｓｔ））ではなく、許容の意味（すなわち、～する可能性を有する意味）において使用される。同様に、「含む」（ｉｎｃｌｕｄｅ、ｉｎｃｌｕｄｉｎｇ、ｉｎｃｌｕｄｅｓ）は、含むがそれらに限定されないことを意味する。
【０００６】
本明細書において、仮想化されたコンピューティングリソースをクライアントに提供するシステムにおいて分散ロック管理を実装するためのシステム及び方法の様々な実施形態が説明される。いくつかの実施形態において、分散ロックマネージャ（ＤＬＭ）の構成要素は、共有リソースにアクセスする、分散アプリケーションのアプリケーション構成要素が、相互に、及び／または共有リソースと通信する別のネットワーク（またはネットワークの一部分）とは物理的または論理的に別個のネットワーク（またはネットワークの一部分）上で共有リソースのロックを管理するために相互に通信することができる。例えば、いくつかの実施形態において、ＤＬＭ構成要素は、クラウドコンピューティング環境の制御プレーンネットワーク上で相互に通信することができ、一方、分散アプリケーションのアプリケーション構成要素は、クラウドコンピューティング環境のデータプレーンネットワーク上で相互に、及び／または共有リソースと通信することができる。いくつかの実施形態において、アプリケーションプログラミングインターフェース（ＡＰＩ）は、クラウドコンピューティング環境内部で実行しているクライアントアプリケーション（またはその構成要素）に公開されることに加えて、クラウドコンピューティング環境の外側で実行

しているクライアントアプリケーション（またはその構成要素）に公開され得る。そのような実施形態において、クライアントアプリケーション（またはその構成要素）のうちの何れかは、（例えば、それらのクライアントアプリケーションによって共有リソースのロック上のロック管理操作を開始するために）ＤＬＭのロック機構に関与することができる。

【０００７】

分散アプリケーションを実装及び／または仮想化されたコンピューティングリソースをクライアントアプリケーションに提供するようなクラスタ環境では、リソース間に発生する可能性がある様々なレベルの共有が存在し得ることに留意されたい。これらのシステムは、（例えば、コンピューティングノードの障害または他のフェイルオーバー状況に応答して）移動させる必要があるリソースを管理するために、様々な機構を実装することができる。例えば、１０の仮想化されたインスタンスが仮想プライベートクラウド（ＶＰＣ）内部で実行しているシステムでは、（例えば、１つのインスタンスからそれを切断し、別のインスタンスに再接続するために）異なるインスタンス間でネットワークインターフェース（例えば、エラスティック（Ｅｌａｓｔｉｃ）ネットワークインターフェースまたはＥＮＩ、あるいは別のタイプの仮想ネットワークインターフェース）を移動させることが所望される場合が発生し得る。

【０００８】

いくつかの既存のシステムは、共有リソースへのアクセスを制御及び同期するために、分散ロックマネージャを使用するが、これらのシステムは典型的に、ネットワーク接続性を必要とする。例えば、いくつかのクラスタ化技術は、クラスタが使用するロックを管理するために使用されるＤＬＭの相互接続機構として、ネットワークを使用する。様々な実施形態において、本明細書に記載されるシステムは、そのようなネットワーク接続性を必要としない分散ロックマネージャ（ＤＬＭ）を実装することができる。代わりに、ＤＬＭは、クラウドコンピューティング環境の制御プレーンにおいて実装することができ、様々なＡＰＩ呼び出しを使用して、（例えば、これらの分散アプリケーションによってアクセス可能であるリソースの１つ以上のロックを管理するために）クラウドコンピューティング環境内のリソースインスタンス上で実行している分散アプリケーションによってアクセスすることができる。いくつかの実施形態において、例えば、顧客が、顧客のアプリケーションが他のアプリケーションから分離される（すなわち、他のアプリケーションまたは顧客には見えず、及び／または他のアプリケーションとは異なる仮想マシン上で稼動する）ことを望む場合、分散アプリケーションのアプリケーション構成要素が実行するリソースインスタンスは、仮想プライベートクラウド（ＶＰＣ）として構成することができる。いくつかの実施形態において、仮想プライベートクラウド内で実行している各仮想マシンには、その独自のプライベートＩＰアドレスを与えることができる。いくつかの実施形態において、これらのプライベートＩＰアドレス（「仮想プライベートＩＰアドレス」とも呼ぶことができる）は、これらがクライアントの独自のプライベートネットワーク内のＩＰアドレスのいずれとも重ならないように選択することができる。いくつかの実施形態において、ＶＰＣ内部のリソースインスタンスは、公共サービスＡＰＩを使用してサービスリクエストをＤＬＭへ送信することができるが、一方、他の実施形態において、ＶＰＣ内部のリソースインスタンスは、プライベートサービスＡＰＩを使用してサービスリエクストをＤＬＭへ送信することができる。

【０００９】

クラスタ化されたシステムは一般に、少なくともいくつかの共有リソースを含むので、これらの環境において、ＤＬＭは非常に有用であり得る。上記のように、分散ロックマネージャは、様々なタイプの共有リソースへのアクセスを制御及び／または同期するために、分散アプリケーションによって使用され得る。様々な実施形態において、ロックは、任意のタイプのリソース（例えば、ディスクボリューム、ＥＮＩまたは別の種類の仮想ネットワークインターフェース、データベース行、またはファイル）、または一般に、複数のプロセスによるアクセスが制御されなければならない任意の実体（例えば、アプリケーシ

ョン構成要素またはリソースインスタンスを含む）と関連付けられ得る。
【００１０】
いくつかの実施形態において、ＤＬＭは、個々のリソース／エンティティがロックされ得るように、またはリソース／エンティティの集合体（例えば、２つ以上の「子」リソースを含む「親」リソース）がロックされ得るように、任意の数のレベルを有するロック階層を実装することができる。いくつかの実施形態において、ロックは、より下位のリソース（例えば、子リソース）上でロックを取得できる前に、より上位のリソース（例えば、親リソース）上で取得されなければならない。
【００１１】
様々な実施形態において、本明細書に記載される分散ロックマネージャは、様々なロックモードをサポートすることができ、その各々は、関連付けられたリソース／エンティティ（またはそのロック）の共有プロパティを示すことができる。例えば、異なる実施形態において、以下のロックモードのうちのいずれか、または全てがサポートされ得る。
・ヌルロックモード。このモードのロックを保持するプロセスが、関連付けられた共有リソースに対する関心を表すことを可能にするが、このプロセスまたは他のプロセスによる関連付けられた共有リソースへの一切のアクセスを防止しない。
・並列読み出しロックモード。プロセスが関連付けられた共有リソースを読み出すことを可能にし（しかし変更は可能にしない）、かつ他のプロセスが関連付けられた共有リソースを読み取る、または変更することを可能にするが、関連付けられた共有リソースに対する排他的アクセスを防止する。
・並列書き込みロックモード。プロセスが関連付けられた共有リソースを読み出す、または変更することを可能にし、かつ他のプロセスが関連付けられた共有リソースを読み出す、または変更することを可能にするが、関連付けられた共有リソースに対する排他的アクセスを防止する。
・保護読み出しロックモード。プロセスが関連付けられた共有リソースを読み出すことを可能にし（しかし変更は可能にしない）、かつ他のプロセスが関連付けられた共有リソースを読み出すことを可能にする（しかし変更は可能にしない）。
・保護書き込みロックモード。プロセスが関連付けられた共有リソースを読み出す、または変更することを可能にし、かつ並列読み出しアクセスを有する他のプロセスが、関連付けられた共有リソースを読み出すことを可能にする（しかし変更することは可能にしない）。
・排他ロックモード。プロセスが関連付けられた共有リソースを読み出す、または変更することを可能にするが、他のプロセスによる関連付けられた共有リソースへの一切のアクセスを防止する。
【００１２】
いくつかの実施形態において、共有リソースのロックを取得することは、（例えば、ＤＬＭの構成要素へ）ロックを取得する要求を通信すること、及び／またはロック上の待ち行列に入ること（例えば、要求するプロセスのインジケータを、ロックを取得するために待機しているプロセスの待ち行列に追加すること）を含むことができる。ロック要求は、異なる実施形態において、同期（例えば、プロセスは、ロックが与えられるまで待機することができる）または非同期（例えば、プロセスは、ロックを必要としない他の操作を継続することができるが、ロックが与えられる場合、ロックが与えられると中断され得る）であり得る。
【００１３】
いくつかの実施形態において、ロックデータ構造（またはＤＬＭによって作成及び／または管理されるロックの他の表現）は、ロック値を含むことができ、その意味は、アプリケーションにおけるその使用によって確立され得る。いくつかのそのような実施形態において、ＤＬＭは、ロック値の意味が何であるかを知らない場合がある（または知る必要がない場合がある）。いくつかの実施形態において、ロック値は、ロックと関連付けられた共有リソースを所有するプロセスまたはアプリケーションを特定することができる。いく

つかの実施形態において、ロック値は、ロックの、または関連付けられた共有リソースのバージョン識別子を表現する単調増分値であり得る。いくつかの実施形態において、共有リソースにアクセスする前に、プロセス（またはアプリケーション）は、関連付けられたロックの値を読み出すことができる。値が、プロセス（またはアプリケーション）がロックを最後に取得したときから変化していない場合、プロセス（またはアプリケーション）は、プロセス（またはアプリケーション）がロックを最後に取得したとき以来、共有リソースが別のプロセス（またはアプリケーション）によって変更されていないことを知ることができる。いくつかの実施形態において、本明細書に記載されるＤＬＭは、デッドロック検出をサポートまたは提供することができる。

【００１４】

上述のように、いくつかの実施形態において、ＤＬＭは、ＤＬＭ構成要素が（例えば、クラウド内で実行しているクライアントアプリケーションに割り当てられた以外のネットワーク接続性または処理能力を使用する）制御プレーンネットワーク上で相互に通信するような様式において、分散コンピューティング環境（例えば、仮想コンピューティングリソース及び／または仮想コンピューティングサービスをクライアントに提供するクラウドコンピューティング環境）において実装することができる。そのような実施形態において、ＤＬＭ構成要素間の通信は、クライアントアプリケーションの使用を目的とするネットワークリソースを消費しない、または、さもなければクライアントアプリケーションの構成要素間の通信に干渉しない場合がある。

【００１５】

本明細書に記載されるシステム及び方法は、異なる実施形態において、ネットワーク環境内部の１つ以上のコンピューティングシステム上で、またはそれらによって実装され得る。本明細書に記載される分散ロックマネージャの実施形態が実装され得る例示的なコンピュータシステムが、図９に示される。これらの分散ロックマネージャを実装するための様々なシステム及び方法の実施形態が、概して、本明細書において、インターネット等の中間ネットワークを介して、サービスプロバイダのプロバイダネットワーク上に実装された仮想化されたリソース（例えば、仮想化されたコンピューティング及びストレージリソース）をクライアントに提供するサービスプロバイダの状況において、記載される。図２～４、７及び９（ならびにそれらの対応する説明）は、本明細書に記載されるシステム及び方法の実施形態が実装され得る例示的環境を示して記載し、限定することを目的としない。少なくともいくつかの実施形態において、プロバイダネットワークを介してサービスプロバイダのクライアントに提供されるリソースのうちの少なくともいくつかは、他のクライアント（単数または複数）と共有されるマルチテナントハードウェア上、及び／または特定のクライアント専用のハードウェア上に実装された仮想化されたコンピューティングリソースであり得る。各仮想化されたコンピューティングリソースは、リソースインスタンスと呼ぶことができる。リソースインスタンスは、例えば、サービスプロバイダのクライアントに賃借またはリースされ得る。例えば、サービスプロバイダのクライアントは、リソースインスタンスを取得及び構成するため、並びに、例えば、仮想化されたプライベートネットワーク等のリソースインスタンスを含む仮想ネットワーク構成を確立及び管理するために、サービスへのＡＰＩを介して、プロバイダネットワークの１つ以上のサービスにアクセスすることができる。

【００１６】

いくつかの実施形態において、リソースインスタンスは、例えば、複数のオペレーティングシステムが、ホストコンピュータ上で同時に稼動することを可能にするハードウェア仮想化技術に従って、すなわち、ホスト上の仮想マシン（ＶＭ）として実装され得る。ホスト上のハイパーバイザ、または仮想マシンモニタ（ＶＭＭ）は、ホスト上のＶＭに仮想プラットフォームを提示することができ、ＶＭの実行を監視する。各ＶＭには１つ以上のプライベートＩＰアドレスを提供することができ、ホスト上のＶＭＭは、ホスト上のＶＭのプライベートＩＰアドレスを認識することができる。そのようなハードウェア仮想化技術を採用するシステムの実施例が、図３に示され、以下に詳細に説明する。

【００１７】
いくつかの実施形態において、ＶＭＭは、クライアントデータパケットをカプセル化し、プロバイダネットワーク内部の異なるホスト上のクライアントリソースインスタンス間のネットワークサブストレート上でルーティングするために、インターネットプロトコル（ＩＰ）トンネリング技術を使用することができる。プロバイダネットワークは、ルータ、スイッチ、ネットワークアドレストランスレータ（ＮＡＴ）等のネットワークデバイス、並びにデバイス間の物理的接続を含む物理的ネットワークサブストレートを含むことができる。プロバイダネットワークは、カプセル化されたパケット（すなわち、オーバーレイネットワーク上でルーティングするためのオーバーレイネットワークアドレス情報を含むがこれに限定されない、オーバーレイネットワークメタデータを用いてタグ付けされているクライアントパケット）が、トンネルまたはオーバーレイネットワーク経路を介して、ネットワークサブストレートを通過させられ得るオーバーレイネットワークを提供するために、ＩＰトンネリング技術を採用することができる。ＩＰトンネリング技術は、ネットワークサブストレート上にオーバーレイネットワークを作成するためのマッピング及びカプセル化システムを提供することができ、オーバーレイネットワーク層（公共ＩＰアドレス）及びネットワークサブストレート層（プライベートＩＰアドレス）のために別々の名前空間を提供することができる。少なくともいくつかの実施形態において、オーバーレイネットワーク層内のカプセル化されたパケットは、それらのトンネルサブストレートターゲット（プライベーＩＰアドレス）が何であるべきかを決定するために、マッピングディレクトリで確認することができる。ＩＰトンネリング技術は、物理的ネットワークサブストレート上にオーバーレイされた仮想ネットワークトポロジを提供することができ、クライアントに提示されるインターフェース（例えば、サービスＡＰＩ）は、オーバーレイネットワークに付加されるので、クライアントリソースインスタンスが、パケットが送信されるＩＰアドレスを提供すると、ＩＰアドレスは、ＩＰオーバーレイアドレスがどこにあるかを決定することができるマッピングサービスと通信することによって、仮想空間内で稼動する。オーバーレイネットワーク技術の例示的使用が、図３に示され、以下に詳細に説明する。

【００１８】
ホスト上のクライアントリソースインスタンスは、伝送制御プロトコル（ＴＣＰ）等のステートフルプロトコルに従って、及び／またはユーザデータグラムプロトコル（ＵＤＰ）等のステートレスプロトコルに従って、同じホスト上または異なるホスト上の他のクライアントリソースインスタンスと通信することができる。しかしながら、クライアントパケットは、送信側ＶＭＭによって、オーバーレイネットワークプロトコルに従ってカプセル化され、受信側ＶＭＭによってカプセル化が解除される。ホスト上のＶＭＭは、ホスト上のクライアントリソースインスタンスから、別のクライアントリソースインスタンスのＩＰアドレスに向けられたクライアントパケット（例えば、ＴＣＰまたはＵＤＰパケット）を受信すると、オーバーレイネットワーク（またはＩＰトンネリング）プロトコルに従ってクライアントパケットをカプセル化またはタグ付けし、配信のためにカプセル化されたパケットをオーバーレイネットワーク上へ送信する。カプセル化されたパケットは次いで、ＩＰトンネリング技術に従って、オーバーレイネットワークを介して別のＶＭＭへルーティングされ得る。他のＶＭＭは、パケットからオーバーレイネットワークカプセル化を解除し、ターゲットクライアントリソースインスタンスを実装するホスト上の適切なＶＭへ、クライアントパケット（例えば、ＴＣＰまたはＵＤＰパケット）を配信する。すなわち、いくつかの実施形態において、サービスプロバイダのコンピューティング環境（例えば、サービスプロバイダのデータセンター）には単一の基底の物理的ネットワークが存在し得るが、本明細書に記載されるカプセル化は、それが、あたかも各クライアントアプリケーション（または１つ以上のクライアントアプリケーションが実行する各クライアントリソースインスタンス）が、その独自の仮想ネットワーク上で稼動しているように見えることを可能にすることができる（例えば、複数のクライアントアプリケーションのデータパケットは、同じ物理的ネットワーク上を移動することができるが、クライアントアプ

リケーションの各々に向かうトラフィックは、プライベートネットワーク上を移動しているかのように見えることができる）。

【００１９】

いくつかの実施形態において、オーバーレイネットワークは、コネクションレス（またはステートレス）ＩＰプロトコルに従って実装されたステートレスネットワークであり得る。いくつかのそのような実施形態において、送信側ＶＭＭは、ルーティング及び配信のために、カプセル化されたパケットをオーバーレイネットワークへ送信するが、パケットの配信に関する確認応答（ＡＣＫ）または他の応答を受信しない。他の実施形態において、ＶＭＭは、カプセル化されたパケットの配信に関するＡＣＫまたは他の応答を受信することができる。

【００２０】

分散ロックマネージャの構成要素がそれらによって管理されるロックを採用するアプリケーションと同じデータプレーン（単数または複数）内部で実行及び／または通信する既存のシステムとは異なり、本明細書に記載されるシステムのいくつかの実施形態では、分散ロックマネージャが、分散コンピューティングシステム（例えば、仮想コンピューティングリソース及び／または仮想コンピューティングサービスをクライアントに提供するシステム）のプロトコルプレーン層内に組み込むことができ、この制御プレーン層は、システム内の仮想化されたリソース上で実行しているクライアントソフトウェアとは異なるネットワーク可用性を有することができる。いくつかの実施形態において、クライアントソフトウェアは、（例えば、ＡＰＩを通じて）サービスとしてクライアントに公開され得る、制御プレーン層の、またはＤＬＭのネットワーク可用性を認識しない場合がある。

【００２１】

上記のように、ＤＬＭを含む既存のシステムにおいて、相互に通信するためにＤＬＭの構成要素によって使用される接続機構は、ロックと関連付けられたリソースがクライアントアプリケーションによってアクセスされるのと同じ接続機構である。しかしながら、本明細書に記載されるシステムのいくつかの実施形態においては、そうではない。代わりに、クライアントアプリケーションは、本明細書に記載されるロック及びロック機構に関与するためには、ローカルＡＰＩ呼び出しを行うことだけが必要であり得る。いくつかの実施形態において、ＤＬＭによって管理されるロックは、ＡＰＩの観点から、ロックの「ハンドル」と考えられ得る、固有の識別子（または特定の名前空間を含む固有の識別子）を有することができる。上記のように、いくつかの実施形態において、クライアントアプリケーションは、相互に通信するために、ＤＬＭ構成要素によって利用されるネットワーク接続について、何も知る必要がない場合がある。代わりに、各々が特定のロック管理操作に対応する、１つ以上のＡＰＩを起動することによって、ロックを操作することができる。いくつかの実施形態において、ロックの使用法、及び／またはロックの通知法を決定するのは、クライアントアプリケーションの責任であり得る。いくつかの実施形態において、プロセスまたはアプリケーションがロックを作成すると、１つ以上の他のプロセスまたはアプリケーション（時には「関与者」と呼ばれる）は、（例えば、観察者として、または他のロックモードに従って）ロックにサブスクライブすることができる。

【００２２】

いくつかの既存のＤＬＭとは異なり、本明細書で記載されるＤＬＭは、それらが実装される分散システムの制御プレーンへのフックを含むことができ、ユーザが、異なる共有レベルを含むロックを作成し、ロックを使用するアプリケーションのネットワーク接続性に依存することなく、それらのロックを管理することを可能にする。

【００２３】

いくつかの実施形態において、本明細書に記載される分散ロックマネージャは、オンプレミスのホスト（例えば、クライアントネットワーク上で実行しているホスト）と、プロバイダネットワーク（例えば、公共またはプライベートクラウドコンピューティング環境）内で実行しているホストとの間の境界にまたがることができる。いくつかの実施形態において、本明細書に記載される技法を実装することは、ＤＬＭロックが、（例えば、顧客

の施設にある）クライアントネットワーク上に存在するホストによって、及び／またはクラウドコンピューティング環境内部に存在するホストによって、取得され、操作され、管理されることを可能にする、一連のソフトウェア構成要素を構築することを含むことができる。ロックそれ自体は、相対的に単純であり得て、どのリソースがロックされるべきか、いつロックされるべきかを決定することは、クライアントアプリケーションの設計者に一任され得る。様々な実施形態において、ロックと関連付けられたリソースは、ディスクボリューム、仮想ネットワークアダプタ、ファイル、データベース内部のレコードまたは他の項目、あるいはアプリケーション開発者が、複数のプロセスによる非同期のアクセスから保護することを希望する任意の他のリソースを含むことができる。上記のように、ロックは、階層にグループ化され得る。例えば、ファイルロックと関連付けられたレコードロック、または一連のロックを共有するホスト群が存在し得る。いくつかの実施形態において、ロックの状態が変化すると（例えば、排他的書き込みアクセスのために取得されると）、そのロックのサブスクライバの全ては、一貫したある時点のロックのビューを有するようになる。すなわち、ある時点で（かつプロセスが排他的ロックを保持する場合にのみ）、１つのプロセスだけがロックの状態を変更することができ、任意の所与の時点で、ロックの関与者の全員が同じ値を見るという点で、ロック状態の変化はアトミック的であり得る。いくつかの実施形態において、ロックそれ自体は、いずれのサブスクライバが不在であっても、ロックが永続する点で耐久性があり得る。例えば、ディスクボリューム上のロックを共有している３つのホストが存在し（例えば、２つがクラウドコンピューティング環境、１つがオンプレミス）、そのうちの１つは、排他的書き込みのロックを保持している場合、全ての３つのホストに障害が発生、またはシャットダウンしてから、次いで、再起動すると、ロックの状態は、ロックマネージャによって維持することができ、排他的書き込みロックは依然として、それを取得したホストによって保持することができる。例えば、ロック状態は、ロックを使用するアプリケーション構成要素が実行しているホスト以外の、ロック機構に関与している制御プレーンのどこかで維持され得る。いくつかの実施形態において、ローカルプロセス（例えば、クラウドコンピューティング環境内及び／またはクライアントネットワーク上の様々なリソースインスタンス内で実行しているクライアントアプリケーション）は、ＤＬＭによってそれらの代わりに作成された各ロックのロック状態のキャッシュされたビューを維持することができ、クラウドコンピューティング環境内の永続的データストア内（例えば、データベース内）の各ロックのロック状態のコピーも存在し得る。

【００２４】

いくつかの実施形態において、ロックを表示及び／または変更するためにロックグループに加わるためには、またはロックにアクセスするためには、承認を必要とするセキュリティアクセスプロパティを有することができるという意味で、ロックはセキュアでもあり得る。前述のように、ロックは、その意味がロックを利用するアプリケーションまたは仮想コンピューティングサービスによって決定される値を含むことができる。いくつかの実施形態において、それらの独自のロック規則に従うことは、クライアントアプリケーションの責任であり得ることに留意されたい。例えば、クラウドコンピューティング環境内で実行しているアプリケーションが、ロックをディスクボリュームと関連付ける場合、クラウドコンピューティング環境は、この関連付けが存在することを知ることがなくても、または知る必要がなくてもよい。代わりに、関連付けが何であるか、及びロックがどのようにリソースと関連付けられているかを知ることは、アプリケーションに一任され得る。いくつかの実施形態において、ロックと対応するリソースとの間の関連付けは、クラウドコンピューティング環境の他の実体（例えば、クラウドコンピューティング環境の管理構成要素）には可視ではない場合さえあり得る。前述のように、これもまた、ロックの使用法及びそれらの値の意味を決定することは、クライアントアプリケーション（またはアプリケーションの設計者）に一任され得る（例えば、ロックと、保護する、及び／またはアクセスを制御する共有リソースとの間の関係を定義することは、クライアントアプリケーションに一任され得る）。例えば、ファイルが渡されるクラスタ化されたアプリケーション

では、ファイルと関連付けられたロックは、任意の所与の時点で、どのアプリケーション構成要素またはプロセスがファイルを所有するかを示すことができる。

【００２５】

いくつかの実施形態において、値は、上述のように、単調増加するバージョン番号であり得る。いくつかの実施形態において、本明細書に記載される技法は、ソフトウェアライブラリによって実装され、並びに／あるいは、オープンソースまたは専有（クローズ）オペレーティングシステム、及び／またはスマートフォンまたは他のモバイルデバイスのオペレーティングシステムを含む、様々なオペレーティングシステムのために開発されているコマンド行ツールによって起動され得る。

【００２６】

仮想コンピューティングシステムを提供する分散コンピューティングシステム内で分散ロックマネージャをクライアントに提供するための方法の一実施形態が、図１のフロー図によって示される。１１０に示されるように、この実施例において、方法は、クライアントにＡＰＩを公開する分散ロックマネージャ（ＤＬＭ）を実装しているクライアントに仮想コンピューティングサービス（例えば、仮想コンピューティングリソースを使用して実装されたサービス）を提供する分散システムを含むことができる。方法はまた、１２０のように、ＤＬＭの構成要素が、リソースインスタンスが分散アプリケーションの一部（例えば、プロセスまたは他の構成要素）を実装する、同じノード上で実行しているリソースインスタンスから、共有リソースのロックのためのロック管理操作を実施する要求を受信することも含むことができる。

【００２７】

この実施例に示されるように、方法は、１３０のように、ＤＬＭ構成要素が、要求された操作を実施し、データプレーンネットワーク（アプリケーションに割り当てられている一部）とは物理的または論理的に別個の制御プレーンネットワーク上でロックの状態情報（例えば、操作を実施することから生じる変更されたロック値または他の状態情報）を共有するために別のＤＬＭ構成要素と通信することを含むことができる。方法はまた、１４０のように、ＤＬＭ（例えば、要求を受信し、及び／または要求された操作を実施したＤＬＭ構成要素）が、状態情報をローカルに（例えば、このＤＬＭ構成要素が実行しているノード上）キャッシュすること、及び状態情報が、（例えば、要求を受信した、及び／または要求された操作を実施したＤＬＭ構成要素によって、あるいはＤＬＭの別の構成要素によって）永続的ストレージに書き込まれることも含み得る。

【００２８】

例示的プロバイダネットワーク環境

本項では、本明細書に記載される方法の実施形態が実装され得る、例示的プロバイダネットワーク環境を説明する。ただし、これらの例示的プロバイダネットワーク環境は限定することを目的としない。

【００２９】

図２は、少なくともいくつかの実施形態に従う、例示的プロバイダネットワーク環境を示す。プロバイダネットワーク２００は、クライアントが、１つ以上のデータセンターにおけるプロバイダネットワーク（単数または複数）内部のデバイス上に実装された、演算及びストレージリソースを含むがこれに限定されない、仮想化されたリソースのインスタンス２１２を購入、賃借、またはその他取得することを可能にする１つ以上の仮想化サービス２１０を経由して、リソース仮想化をクライアントに提供することができる。プライベートＩＰアドレス２１６は、リソースインスタンス２１２と関連付けることができ、プライベートＩＰアドレスは、プロバイダネットワーク２００上のリソースインスタンス２１２の内部ネットワークアドレスである。いくつかの実施形態において、プロバイダネットワーク２００はまた、クライアントがプロバイダ２００から取得することができる、公共ＩＰアドレス２１４及び／または公共ＩＰアドレス範囲（例えば、インターネットプロトコルバージョン４（ＩＰｖ４）またはインターネットプロトコルバージョン６（ＩＰｖ６）のアドレス）も提供することができる。

【００３０】
従来、プロバイダネットワーク２００は、仮想化サービス２１０を介して、サービスプロバイダのクライアント（例えば、クライアントネットワーク２５０Ａを運用するクライアント）が、クライアントに割り振られた、または割り当てられた少なくともいくつかの公共ＩＰアドレス２１４を、クライアントに割り振られた特定のリソースインスタンス２１２と動的に関連付けることを可能にすることができる。プロバイダネットワーク２００はまた、クライアントが、クライアントに割り当てられた１つの仮想化されたコンピューティングリソースインスタンス２１２に以前にマッピングされた、公共ＩＰアドレス２１４を、これもクライアントに割り当てられている別の仮想化されたコンピューティングリソースインスタンス２１２へ、再マッピングすることを可能にすることもできる。サービスプロバイダによって提供された仮想化されたコンピューティングリソースインスタンス２１２及び公共ＩＰアドレス２１４を使用して、クライアントネットワーク２５０Ａの運用者等のサービスプロバイダのクライアントは、例えば、インターネット等の中間ネットワーク２４０上で、クライアント専用アプリケーションを実装し、クライアントのアプリケーションを提示することができる。中間ネットワーク２４０上の他のネットワーク実体２２０は、次いで、クライアントネットワーク２５０Ａによって発行された送信先公共ＩＰアドレス２１４へのトラフィックを生成することができ、トラフィックは、サービスプロバイダのデータセンターへルーティングされ、データセンターで、ネットワークサブストレートを介して、送信先公共ＩＰアドレス２１４に現在マッピングされている仮想化されたコンピューティングリソースインスタンス２１２のプライベートＩＰアドレス２１６へルーティングされる。同様に、仮想化されたコンピューティングリソースインスタンス２１２からの応答トラフィックは、ネットワークサブストレートを介して、発信元エンティティ２２０への中間ネットワーク２４０上へ戻るようにルーティングすることができる。

【００３１】
本明細書に使用されるように、プライベートＩＰアドレスは、プロバイダネットワークにおけるリソースインスタンスの内部ネットワークアドレスを指す。プライベートＩＰアドレスは、プロバイダネットワーク内部のみでルーティング可能である。プロバイダネットワークの外側から発信されているネットワークトラフィックは、プライベートＩＰアドレスへ直接ルーティングされず、その代わりに、トラフィックは、リソースインスタンスにマッピングされている公共ＩＰアドレスを使用する。プロバイダネットワークは、公共ＩＰアドレスからプライベートＩＰアドレスへ、及びこの逆のマッピングを実施するために、ネットワークアドレス変換（ＮＡＴ）または類似の機能性を提供するネットワークデバイスまたは機器を含むことができる。

【００３２】
本明細書に使用されるように、公共ＩＰアドレスは、サービスプロバイダによって、またはクライアントによってのいずれかで、リソースインスタンスに割り振られる、インターネットをルーティング可能なネットワークアドレスである。公共ＩＰアドレスへルーティングされたトラフィックは、例えば、１：１ネットワークアドレス変換（ＮＡＴ）を介して変換され、リソースインスタンスのそれぞれのプライベートＩＰアドレスへ転送される。

【００３３】
いくつかの公共ＩＰアドレスは、プロバイダネットワーク基礎構造によって、特定のリソースインスタンスに割り振ることができ、これらの公共ＩＰアドレスは、標準公共ＩＰアドレス、または簡単に標準ＩＰアドレスと呼ぶことができる。少なくともいくつかの実施形態において、リソースインスタンスの標準ＩＰアドレスからプライベートＩＰアドレスへのマッピングは、リソースインスタンスタイプ全てのデフォルトの起動構成である。

【００３４】
少なくともいくつかの公共ＩＰアドレスは、プロバイダネットワーク２００のクライアントによって割り当てられる、または取得され得て、クライアントは次いで、それらの割

り当てられた公共ＩＰアドレスを、クライアントに割り当てられた特定のリソースインスタンスに割り振ることができる。これらの公共ＩＰアドレスは、クライアント公共ＩＰアドレス、または簡単にクライアントＩＰアドレスと呼ぶことができる。標準のＩＰアドレスの場合のように、プロバイダネットワーク２００によってリソースインスタンスに割り振られるのではなく、クライアントＩＰアドレスは、例えば、サービスプロバイダによって提供されるＡＰＩを介して、クライアントによってリソースインスタンスに割り振られる場合がある。標準のＩＰアドレスとは異なり、クライアントＩＰアドレスは、クライアントアカウントに割り当てられ、必要または所望に応じて、それぞれのクライアントによって、他のリソースインスタンスに再マッピングされ得る。クライアントＩＰアドレスは、特定のリソースインスタンスではなく、クライアントのアカウントと関連付けられ、クライアントは、クライアントがそれを解放することを選択するまで、そのＩＰアドレスを制御する。従来の静的ＩＰアドレスとは異なり、クライアントＩＰアドレスは、クライアントが、クライアントの公共ＩＰアドレスを、クライアントのアカウントと関連付けられたいずれかのリソースインスタンスに再マッピングすることによって、リソースインスタンスまたは可用性ゾーンの障害をマスクすることを可能にする。クライアントＩＰアドレスは、例えば、クライアントが、クライアントＩＰアドレスを交換リソースインスタンスに再マッピングすることによって、クライアントのリソースまたはソフトウェアで問題を回避するように設計することを可能にする。

【００３５】

図３は、少なくともいくつかの実施形態に従う、例示的データセンター（例えば、ＩＰトンネリング技術を使用してネットワークサブストレート上にオーバーレイネットワークを実装するデータセンター）を示す。この実施例に示されるように、プロバイダデータセンター３００は、ルータ、スイッチ、ネットワークアドレストランスレータ（ＮＡＴ）等のネットワークデバイス３１２を含むネットワークサブストレートを含むことができる。少なくともいくつかの実施形態は、カプセル化されたパケットがトンネルを使用してネットワークサブストレート３１０を通過され得るオーバーレイネットワークを提供するために、インターネットプロトコル（ＩＰ）トンネリング技術を採用することができる。ＩＰトンネリング技術は、ネットワーク（例えば、図３のデータセンター３００内のローカルネットワーク）上にオーバーレイネットワークを作成するためのマッピング及びカプセル化システムを提供することができ、オーバーレイ層（公共ＩＰアドレス）及びネットワークサブストレート３１０層（プライベートＩＰアドレス）に別々の名前空間を提供することができる。オーバーレイ層内のパケットは、それらのトンネルサブストレートターゲット（プライベートＩＰアドレス）が何であるべきかを決定するために、マッピングディレクトリ（例えば、マッピングサービス３３０によって提供される）と照合することができる。ＩＰトンネリング技術は、仮想ネットワークトポロジ（オーバーレイネットワーク）を提供し、クライアントに提示されるインターフェース（例えば、サービスＡＰＩ）は、オーバーレイネットワークに付加されるので、クライアントが、クライアントがパケットを送信したいＩＰアドレスを提供すると、ＩＰアドレスは、ＩＰオーバーレイアドレスがどこにあるか知っているマッピングサービス（例えば、マッピングサービス３３０）と通信することによって、仮想空間内で稼動する。

【００３６】

少なくともいくつかの実施形態において、ＩＰトンネリング技術は、ＩＰオーバーレイアドレス（公共ＩＰアドレス）をサブストレートＩＰアドレス（プライベートＩＰアドレス）にオーバーレイし、２つの名前空間の間のトンネルでパケットをカプセル化し、パケットを、トンネルを介して正しいエンドポイントに配信することができ、カプセル化がパケットから解除される。図３には、ホスト３２０Ａ上の仮想マシン（ＶＭ）３２４Ａから中間ネットワーク３５０上のデバイスへの例示的なオーバーレイネットワークトンネル３３４Ａ、及びホスト３２０Ｂ上のＶＭ３２４Ｂと、ホスト３２０Ｃ上のＶＭ３２４Ｃとの間の例示的なオーバーレイネットワークトンネル３３４Ｂが示される。いくつかの実施形態において、パケットは、送信前にオーバーレイネットワークパケット形式にカプセル化

することができ、オーバーレイネットワークパケットは、受信後に解除することができる。他の実施形態において、パケットをオーバーレイネットワークパケットにカプセル化する代わりに、オーバーレイネットワークアドレス（公共ＩＰアドレス）を、送信前にパケットのサブストレートアドレス（プライベートＩＰアドレス）に埋め込むことができ、受信後にパケットアドレスから解除することができる。実施例として、オーバーレイネットワークは、公共ＩＰアドレスとして、３２ビットＩＰｖ４（インターネットプロトコルバージョン４）アドレスを使用して実装することができ、ＩＰｖ４アドレスは、プライベートＩＰアドレスとして、サブストレートネットワーク上で使用される１２８ビットＩＰｖ６（インターネットプロトコルバージョン６）アドレスの一部として埋め込むことができる。

【００３７】

図３を参照すると、本明細書に記載される分散ロックマネージャの実施形態を実装することができる少なくともいくつかのネットワークは、複数のオペレーティングシステムが、ホストコンピュータ（例えば、図３のホスト３２０Ａ及び３２０Ｂ）上で同時に、すなわち、ホスト３２０上の仮想マシン（ＶＭ）３２４として、稼動することを可能にするハードウェア仮想化技術を含むことができる。ＶＭ３２４は、例えば、ネットワークプロバイダのクライアントに賃借またはリースされ得る。ハイパーバイザ、または仮想マシンモニタ（ＶＭＭ）３２２は、ホスト３２０上で、ホスト上のＶＭ３２４に仮想プラットフォームを提示し、ＶＭ３２４の実行を監視する。各ＶＭ３２４には、１つ以上のプライベートＩＰアドレスを提供することができ、ホスト３２０上のＶＭＭ３２２は、ホスト上のＶＭ３２４のプライベートＩＰアドレスを認識することができる。マッピングサービス３３０は、ローカルネットワーク上のＩＰアドレスに対応しているルータまたは他のデバイスの全てのネットワークＩＰプレフィックス及びＩＰアドレスを認識することができる。これは、複数のＶＭ３２４に対応しているＶＭＭ３２２のＩＰアドレスを含む。マッピングサービス３３０は、例えば、サーバーシステム上で集中管理されてもよく、または代替として、ネットワーク上の２つ以上のサーバーシステムまたは他のデバイス間で分散されてもよい。ネットワークは、例えば、データセンター３００ネットワーク内部の異なるホスト３２０上のＶＭ３２４間でデータパケットをルーティングするために、例えば、マッピングサービス技術及びＩＰトンネリング技術を使用することができ、内部ゲートウェイプロトコル（ＩＧＰ）は、そのようなローカルネットワーク内部のルーティング情報を交換するために使用することができることに留意されたい。

【００３８】

加えて、プロバイダデータセンター３００ネットワーク（時には自律システム（ＡＳ）と呼ばれる）のようなネットワークは、パケットをＶＭ３２４からインターネットの送信先へ、及びインターネットの発信元からＶＭ３２４へルーティングするために、マッピングサービス技術、ＩＰトンネリング技術、及びルーティングサービス技術を使用することができる。外部ゲートウェイプロトコル（ＥＧＰ）またはボーダゲートウェイプロトコル（ＢＧＰ）は一般に、インターネット上の発信元と送信先との間のインターネットルーティングのために使用されることに留意されたい。図３は、少なくともいくつかの実施形態に従う、リソース仮想化技術を提供し、かつインターネットトランジットプロバイダに接続するエッジルータ３１４を介して完全なインターネットアクセスを提供するネットワークを実装する例示的なプロバイダデータセンター３００を示す。プロバイダデータセンター３００は、例えば、ハードウェア仮想化サービスを介して、仮想コンピューティングシステム（ＶＭ３２４）を実装する能力、及びストレージ仮想化サービスを介して、ストレージリソース３１８上に仮想化されたデータストア３１６を実装する能力をクライアントに提供することができる。

【００３９】

いくつかの実施形態において、データセンター３００は、例えば、データセンター３００内のホスト３２０上のＶＭ３２４からインターネット送信先へ、及びインターネット発信元からＶＭ３２４へパケットをルーティングするために、仮想化されたリソースへ及び

仮想化されたリソースからのトラフィックをルーティングするために、ＩＰトンネリング技術、マッピングサービス技術、及びルーティングサービス技術を実装することができる。インターネットの発信元及び送信先は、例えば、中間ネットワーク３４０に接続されたコンピューティングシステム３７０、及び（例えば、ネットワーク３５０を中間トランジットプロバイダに接続するエッジルータ（単数または複数）３１４を介して）中間ネットワーク３４０に接続するローカルネットワーク３５０に接続されたコンピューティングシステム３５２を含むことができる。プロバイダデータセンター３００ネットワークはまた、例えば、データセンター３００内のホスト３２０上のＶＭ３２４から同じホスト上またはデータセンター３００内の他のホスト３２０上の他のＶＭ３２４へ、データセンター３００内のリソース間でパケットをルーティングすることもできる。

【００４０】

データセンター３００を提供するサービスプロバイダはまた、データセンター３００に類似するハードウェア仮想化技術を含み、中間ネットワーク３４０にも接続することができる、追加のデータセンター（単数または複数）３６０を提供することができる。パケットは、例えば、データセンター３００内のホスト３２０上のＶＭ３２４から、別の類似のデータセンター３６０内の別のホスト上の別のＶＭへ、及びこの逆等、データセンター３００から他のデータセンター３６０へ転送することができる。

【００４１】

上記に複数のオペレーティングシステムが、ホスト上の仮想マシン（ＶＭ）としてホストコンピュータ上で同時に稼動することを可能にし、ＶＭは、ネットワークプロバイダのクライアントに賃借またはリースされ得る、ハードウェア仮想化技術を説明したが、ハードウェア仮想化技術はまた、同様な様式でネットワークプロバイダのクライアントに対する仮想化されたリソースとして、例えば、ストレージリソース３１８等、他のコンピューティングリソースを提供するためにも使用され得る。

【００４２】

公共ネットワークは、広義には、複数のエンティティ間へのオープンアクセス及び相互接続を提供するネットワークとして定義され得ることに留意されたい。インターネット、またはワールドワイドウェブ（ＷＷＷ）は、公共ネットワークの一例である。共有ネットワークは、広義には、アクセスが一般に限定されない公共ネットワークとは対照的に、アクセスが２つ以上のエンティティに限定されるネットワークとして定義され得る。共有ネットワークは、例えば、１つ以上のローカルエリアネットワーク（ＬＡＮ）及び／またはデータセンターネットワーク、あるいは広域ネットワーク（ＷＡＮ）を形成するために相互接続される２つ以上のＬＡＮまたはデータセンターネットワークを含むことができる。共有ネットワークの例として、企業ネットワーク及び他のエンタープライズネットワークを挙げることができるが、これらに限定されない。共有ネットワークは、ローカルエリアを対象とするネットワークからグローバルネットワークまでの範囲のいずれかであり得る。共有ネットワークは、少なくともいくつかのネットワーク基礎構造を公共ネットワークと共有することができ、共有ネットワークは、他のネットワーク（単数または複数）と共有ネットワークとの間のアクセスを制御して、公共ネットワークを含むことができる、１つ以上の他のネットワークに連結され得ることに留意されたい。共有ネットワークはまた、インターネットのような公共ネットワークとは対照的に、プライベートネットワークとして考えることもできる。実施形態において、共有ネットワークまたは公共ネットワークのいずれかが、プロバイダネットワークとクライアントネットワークとの間の中間ネットワークとして役目を果たすことができる。

【００４３】

いくつかの実施形態において、本明細書に記載されるＤＬＭは、図２または図３に示され、前述した例示的なプロバイダネットワーク環境のうちの１つ等、分散コンピューティング環境（例えば、仮想コンピューティングリソース及び／またはサービスをクライアントに提供するクラウドコンピューティング環境）において実装され得る。いくつかの実施形態において、そのようなシステムに実装されたＤＬＭの構成要素は、（例えば、クラウ

ドコンピューティング環境で実行しているクライアントアプリケーションに割り当てられた以外の、あるいは相互に通信する及び／またはロックによって保護された共有リソースにアクセスするためにクライアントアプリケーションの構成要素によって使用される、ネットワーク接続性及び／または処理能力を使用して）制御プレーンネットワーク上で相互に通信することができる。

【００４４】

いくつかの実施形態において、複数のリソースインスタンスが、クライアントの代わりに分散アプリケーションを実装するためにクラウドコンピューティング環境において実行している場合がある。前述のように、クラウドコンピューティング環境は、各アプリケーション（及び／または各仮想プライベートネットワーク）がその独自の名前空間を有することができる、マルチテナント環境であり得る。いくつかの実施形態において、各クライアントは、ネットワーク接続及び／または処理能力（帯域幅）のその独自の割当を有することができる。例えば、データプレーンネットワークのネットワーク接続性及び／または処理能力は、様々なクライアントの使用のためにプロビジョニング（例えば、専用または予約）され得る。いくつかの実施形態において、ＤＬＭの１つ以上の構成要素（またはインスタンス）はまた、リソースインスタンスのうちの１つが実行している各ノード上でも実行されている場合があり、これらの構成要素は、相互に通信するためにクライアントアプリケーションに割り当てられた以外のネットワーク接続及び／または処理能力を使用することができる。例えば、様々な実施形態において、ノードあたり１つのＤＬＭ構成要素（またはインスタンス）、または各ノード上の顧客あたり１つのＤＬＭ構成要素（またはインスタンス）が存在し得る。

【００４５】

いくつかの実施形態において、クライアントアプリケーションの構成要素が、クライアントアプリケーションの構成要素がそれらの通常の作業（例えば、ロック管理以外の作業）の一部として、相互に通信するネットワーク接続とは異なる接続機構上から同じノード上のＤＬＭ構成要素／インスタンスへのローカルＡＰＩ呼び出しを行うことができる。様々なロック管理操作（例えば、ロックを作成、ロックの１つ以上のプロパティを指定、ロックのリストを取得、ロックをサブスクライブ、ロックを取得、ロックを解放、またはロックを消去する操作）を起動するためにローカルＡＰＩ呼び出しを行うことによって、クライアントアプリケーションの構成要素は、ＤＬＭによって管理されるロックに関与することができる。

【００４６】

いくつかの実施形態において、本明細書に記載されるクライアントアプリケーション及び他のプロセスの全てを実行している物理的コンピュータが存在する一方、クライアントアプリケーションは、物理的コンピュータ上で仮想マシンとして稼動している場合がある。例えば、これらの仮想マシンの作成を管理する、これらの仮想マシンのリソースをプロビジョニングする、及び／またはクライアント及び／またはそれらのアプリケーションの代わりに他の管理タスク（例えば、リソース使用量、顧客アカウント、サービスの請求等を監視する）を実施するように構成されるクラウドコンピューティング環境の内部プロセスは、クラウドコンピューティング環境内の制御プレーン層（またはハイパーバイザ）内で実行することができる。対照的に、クライアントアプリケーション（例えば、アプリケーション構成要素を実装する各リソースインスタンス）は、クラウドコンピューティング環境のデータプレーン層内で実行することができる。これらの層の下には、いくつかの実施形態において、各ホストノードに対して（または複数のホストノードに対して）１つの物理的ネットワークカードだけが存在し得るが、各リソースインスタンスは、あたかもそれがその独自のネットワーク（例えば、仮想ネットワーク）を有するように実行することができる。いくつかの実施形態において、各リソースインスタンスは、その独自のデータプレーンネットワーク接続（単数または複数）を有することができるが、これらのデータプレーンネットワーク接続に依存する必要なく、ローカルＡＰＩ呼び出し（例えば、同じノード上のＤＬＭ構成要素への呼び出し）を行うことができる。

【００４７】
いくつかの実施形態において、ＤＬＭはまた、制御プレーン（ハイパーバイザ）層上で稼動しているプロセスとして実装することができる。したがって、それは、クライアントプロセスが認識せず、アクセスを有さないネットワークアクセスを有することができる。そのような実施形態において、ＤＬＭは、データプレーンのいかなるリソース（例えば、コンピュートインスタンスまたはネットワーク接続／大域幅）も消費しない場合があり、それらのリソースに対してクライアントアプリケーションと競争しない場合があり、制御プレーンのリソースだけを消費する場合がある。様々な実施形態において、ＤＬＭ構成要素（またはインスタンス）は、様々な基底のネットワーク及びネットワーク機構のいずれかを使用して、作成したロックのロック状態情報を共有することができる。一実施形態において、ＤＬＭ構成要素は、ロック状態情報が共有される機構として、クラスタ通信（例えば、ＩｎｆｉｎｉＢａｎｄ（登録商標）アーキテクチャ仕様に準じる通信リンク）のために設計された高速相互接続を採用することができる。例えば、ロックの状態に変化が生じると、ロックを変更したＤＬＭ構成要素（または別のＤＬＭ構成要素）は、ロック状態が変化したことを、１つ以上の他のＤＬＭ構成要素（例えば、ロックのサブスクライバである任意のクライアントアプリケーション構成要素と同じノード上で実行しているＤＬＭ構成要素）に通知することができ、及び／または変更されたロック状態値をサブスクライバに通信する。様々な実施形態において、ロックを変更するＤＬＭ構成要素（または別のＤＬＭ構成要素）は、永続的データストア（例えば、クラウドコンピューティング環境内）内のロックのロック状態情報のコピーを更新することに責任を負う場合がある。

【００４８】
図４は、いくつかの実施形態に従う、制御プレーンにおいて分散ロックマネージャを実装する例示的なサービスプロバイダシステムを示すブロック図である。いくつかの実施形態において、プロバイダシステム（図４においてプロバイダシステム４００として示される）は、図３に示されるプロバイダデータセンター３００に類似し得る。例えば、図４の各ホストノード上の制御プレーンは、図３に示されるハイパーバイザまたは仮想マシンモニタの機能性のうちのいくつかまたは全てを実装することができる。同様に、図４の各ホストノード上のデータプレーンで実行しているインスタンスは、図３に示された仮想マシンの機能性のうちのいくつかまたは全てを実装する仮想コンピュートインスタンスであり得る。

【００４９】
より具体的には、示された実施例において、プロバイダシステム４００は、複数の仮想化されたデータストア（単数または複数）４４０と、ホストノード４２０及び４３０（その各々がデータプレーン部分と制御プレーン部分とを含む）と、データプレーンネットワーク４１０と、制御プレーンネットワーク４１５（異なる実施形態において、データプレーンネットワーク４１０と異なる物理的ハードウェア上に実装される場合または実装されない場合がある）とを含むことができる。

【００５０】
この実施例において、インスタンス４２８ａ～４２８ｂ、及び４３８ａ～４３８ｂは、システムの制御プレーンのそれぞれの部分（４２４及び４３４として示される）で実行し、１つ以上のクライアントアプリケーションまたはプロセスを実装することができ、そのうちの少なくともいくつかは、ロック（例えば、分散ロックマネージャ（ＤＬＭ）によって管理されるロック）によって保護される共有リソースにアクセスするように構成される。この実施例において、アプリケーション／プロセスを実装するために、これらのインスタンスは、データプレーンネットワーク４１０上で相互に、及び／または他のアプリケーション構成要素（例えば、仮想化されたデータストア（単数または複数）４４０のストレージデバイス４４５）と通信するように構成することができる。

【００５１】
図４に示される実施例において、ホストノード４２０上で実行しているインスタンス４２８ａ～４２８ｂは、様々なロック管理操作を開始するために、ＤＬＭインスタンス４２

６へのＡＰＩ呼び出しを行うように構成することができ、ホストノード４３０上で実行しているインスタンス４３８ａ～４３８ｂは、様々なロック管理操作を開始するために、ＤＬＭインスタンス４３６へのＡＰＩ呼び出しを行うように構成することができる。この実施例において、ＤＬＭ（ＤＬＭインスタンス４２６及び４３６を含み、かつ仮想化されたデータストア（単数または複数）４４０へのアクセスを有する）は、システムの制御プレーンに内（例えば、４２４及び４３４として制御プレーンのそれぞれの部分内）で実行し、その構成要素は、インスタンス４２８ａ～４２８ｂ、及び４３８ａ～４３８ｂのために、共有リソースの１つ以上のロックを管理するように、制御プレーンネットワーク４１５上で相互に通信するように構成することができる。

【００５２】

図４に示されるように、いくつかの実施形態において、サービスプロバイダシステムは、（例えば、制御プレーンネットワーク４１５上の）制御プレーン内にＤＬＭコーディネータ構成要素（例えば、ＤＬＭコーディネータ４５０）を含むことができる。例えば、ＤＬＭコーディネータ構成要素（制御プレーンのクラウドマネージャ構成要素のサブ構成要素であり得る）は、分散ロックサービスを提供するために一体に機能する際、ＤＬＭインスタンスのアクティビティのうちの少なくともいくつかを管理及び／または調整することができる。様々な実施形態において、ＤＬＭインスタンスは、制御プレーンネットワーク４１５上で相互に及び／またはＤＬＭコーディネータ４５０と通信することができる。例えば、ＤＬＭコーディネータ４１５は、例えば、分散ロックマネージャが（例えば、最新状態のＤＬＭインスタンスの各々によってローカルにキャッシュされたロック状態情報の全てを保つために）ＤＬＭによって管理されたロックの状態の一貫した（または最終的に一貫した）ビューを維持することを促進するために、ＤＬＭインスタンス４２６及び４３６へ、及び／または４２６と４３６との間でメッセージを仲介することができる。他の実施形態において、ＤＬＭインスタンスは（少なくとも時々）、ロック状態情報を共有するために、及び／またはＤＬＭインスタンスの各々にローカルに記憶されたロック状態情報が最新状態に保たれていることを保証するために、制御プレーンネットワーク４１５上で直接相互に通信することができる。いくつかの実施形態において、ＤＬＭコーディネータ４５０は、ロック状態情報の永続的データストアを維持するように構成することができる（図示せず）。

【００５３】

仮想化されたリソースを提供する分散コンピューティングシステム内において分散ロックマネージャを実装するための方法の一実施形態が、図５のフロー図によって示される。５１０に示されるように、この実施例において、方法は、分散コンピューティングシステムが、クライアントから仮想コンピューティングサービスの要求を受信することを含むことができる。要求に応答して、方法は、５２０のように、分散コンピューティングシステムが、分散アプリケーションを実装するために、それぞれのコンピューティングノード上でクライアントのための２つ以上のリソースインスタンスをプロビジョニングすることと、これらのインスタンスが、データプレーンネットワーク上で相互に通信するように、システム内でそれらを構成することとを含むことができる。例えば、システムは、様々なコンピュートインスタンスまたは実行プラットフォームインスタンスをプロビジョニングすることができ、いくつかの実施形態において、データプレーンネットワーク上のプロビジョニングされたコンピューティングリソース、プロビジョニングされた記憶能力、プロビジョニングされたネットワーク接続及び／またはプロビジョニングされた処理能力（例えば、帯域幅）を含むことができる。データプレーンネットワークは、分散アプリケーションの構成要素がその作業（例えば、ロック管理以外の作業）を実行するために、相互に通信するネットワークであってもよい。

【００５４】

この実施例に示されるように、方法は、分散コンピューティングシステムが、分散ロックマネージャ（ＤＬＭ）を実装するために、コンピューティングノードの各々の上でリソースインスタンスをプロビジョニングすることと、これらが制御プレーンネットワーク上

で相互に通信するなど、５３０のようにシステム内のこれらのＤＬＭインスタンスを構成することとを含むことができる。例えば、システムは、様々なコンピュートインスタンスまたは実行プラットフォームインスタンスをプロビジョニングすることができ、いくつかの実施形態において、制御プレーンネットワーク上のプロビジョニングされたコンピューティングリソース、プロビジョニングされた記憶能力、プロビジョニングされたネットワーク接続及び／またはプロビジョニングされた処理能力（例えば、帯域幅）を含むことができる。制御プレーンネットワークは、クライアントプロセス以外のプロセス（例えば、リソース使用量、クライアント請求、認証サービス、及び／または分散コンピューティングシステムの他の管理タスク）が実行及び／または相互に通信するネットワークであり得ることに留意されたい。

【００５５】

５４０に示されるように、方法は、ロック管理操作の実施を開始するために、ローカルＤＬＭインスタンス（例えば、呼び出しが公共またはプライベートネットワーク接続上を移動する必要がないように、アプリケーション構成要素と同じコンピューティングノード上で実行しているＤＬＭインスタンス）へのＡＰＩ呼び出しを使用する分散アプリケーション（例えば、分散アプリケーションのアプリケーション構成要素）を含むことができる。例えば、アプリケーション構成要素は、ロックの作成、ロックの取得、ロックの解放、またはこれら及び他の操作を定義するＡＰＩに従う別のロック管理操作を開始することができる。いくつかの実施形態において、方法は、５５０のように、ローカルＤＬＭインスタンスが、要求されたロック管理操作を実施することと、他のＤＬＭインスタンスとロック状態情報（例えば、変更されたロック値または操作を実施したことから生じる他の状態情報）を共有することとを含むことができる。本明細書に記載されるように、ＤＬＭの構成要素は、いくつかの実施形態において、分散コンピューティングシステムのデータプレーンネットワーク上ではなく、システムの制御プレーンネットワーク上で相互に通信することができる。方法はまた、５６０のように、ＤＬＭインスタンス（単数または複数）が、ロックの状態情報に対する何らかの変化について、ロックのサブスクライバに通知することも含むことができる。例えば、各ＤＬＭインスタンスは、それらがサブスクライブされているロックのロック情報における何らかの変化について同じノード上のプロセスに通知すること、及び／またはロックの他のサブスクライバが実行しているノード上で実行している他のＤＬＭインスタンスへのロック状態情報の変化を伝播することに責任を負う場合がある。

【００５６】

本明細書に記載される分散ロックマネージャのアプリケーションプログラミングインターフェースは、以下を含むが、これらに限定されない、共有リソースのロックのための様々なロック管理操作を定義することができる。

・ロックを作成すること（これは、ロックとリソースの関連付けを含まない場合があり、呼び出し側の責任となる場合があることに留意されたい）。

・ロックの所定のプロパティの値を設定すること（例えば、ロックの共有プロパティまたはロックモードを指定する）。

・クライアントがサブスクライブすることができるロックのリストを取得すること。

・ロックの状態情報を表示すること（例えば、そのようなＡＰＩが、ロックの複数のサブスクライバが、ロックの状態を表示することを可能にすることができ、状態がアトミックに変化するため、サブスクライバは、ロック状態の一貫してビューを得るようになる）。

・ＤＬＭがロックのハンドルまたはロックの状態情報を返すことに応答して、所与のロックのロックグループのメンバーになる（すなわち、所与のロックをサブスクライブする）ことを要求すること。

【００５７】

いくつかの実施形態において、ロックは、エラスティックネットワークインターフェース（ＥＮＩ）または別のタイプの仮想ネットワークインターフェースを管理するために使用することができる。例えば、いくつかの既存システムにおいて、ネットワークインター

フェースを移動することが可能である場合、これによって、いくつかの既存のシステムのように、システムのロックを管理するための媒体としてネットワーク自体を使用しようとすることがより困難である。すなわち、ネットワークインターフェースが、特定のアプリケーションまたはリソースインスタンスの唯一のネットワークインターフェースである場合、かつそれを移動するために切断されなければならない場合、アプリケーションまたはリソースインスタンスは、ロックマネージャから封鎖され得る。しかしながら、本明細書に記載されるシステムにおいて、分散ロックマネージャは、ネットワークインターフェースのロックを管理するために（例えば、ネットワークインターフェースまたはロックの所有権とのロックの関連付けを管理するために）同じ物理的または論理的ネットワークを使用しない場合がある。その代わりに、ロックマネージャは、その接続機構として、クラウドコンピューティング環境の制御プレーンネットワークを使用することができ、制御プレーンネットワークは、クライアントアプリケーションから隠すことができる。一実施例において、ＥＮＩは、インスタンスの障害に応答して、（そのＩＰアドレス及び接続されたそのクライアントと共に）別のホストへ移動され得る。ＥＮＩを別のホストへ移動させると、それに伴いそのＩＰアドレスをもたらすため、クライアントは、１つのＩＰアドレスだけを知ることが必要であり、クライアントは、ＥＮＩが接続されている場所を知る必要はない。従来のＤＬＭソリューションでは、そのＥＮＩがそれ自体データプレーン接続機構であった場合、それを移動させることはできない（移動させると、そのデータプレーンへの接続がすべて失われるため）。

【００５８】

別の例において、クラスタ化されたファイルシステムタイプのアプリケーションは、ディスクボリューム上にロックを作成することができ、予備のデータベースを起動させて稼動状態に保つことができる（例えば、一次ホスト上のデータベースをミラリングする）。この例において、一次ホストに障害が発生すると、対応するディスクボリュームを、一次ホストから切断し、予備のうちの１つ（例えば、二次ホスト）に再接続することができる。一般に、分散コンピューティングシステムに共有リソースが存在する場合は必ず、ロックの関与者に、関与者のうちの１つが特定の容量のロックを保持していることを示し、及び／またはそのロックの所有者の識別子を他の関与者に通信するためにロックを使用することが可能であるため、ロックをそれらのリソースと関連付けることが有用であり得る。

【００５９】

クライアントアプリケーションまたはプロセスが、クラウドコンピューティング環境において実装される分散ロックマネージャの機能にアクセスするための方法の一実施形態が、図６のフロー図によって示される。６１０に示されるように、この実施例において、方法は、クラウドコンピューティング環境で実行しているクライアントアプリケーションまたはプロセスが、共有リソースのロックを作成するためにローカルＤＬＭインスタンス（例えば、同じコンピューティングノード上で実行しているＤＬＭインスタンス）へのＡＰＩ呼び出しを行うことと、ロックが作成されていることに応答して、新しく作成されたロックの識別子（例えば、ロックハンドル）を受信することとを含むことができる。異なる実施形態において、アプリケーションは、分散アプリケーション、または別のアプリケーションまたはプロセスによってもアクセス可能であるリソースにアクセスする、単一のノード上で稼動しているアプリケーションまたはプロセスであり得る。いくつかの実施形態において、ロックを作成することは、ロックの１つ以上のプロパティ（例えば、共有プロパティ）の値を設定するために、１つ以上の追加のＡＰＩ呼び出しを行うことも含むことができる。

【００６０】

この実施例に示されるように、方法は、６２０のように、クライアントアプリケーション／プロセスが、共有リソースのロックを取得するために、ローカルＤＬＭインスタンスへのＡＰＩ呼び出しを行うことを含むことができる。例えば、要求は、ロックの識別子（またはハンドル）を含むことができ、ロックを作成した同じアプリケーション／プロセスまたはロックに参加もする異なるアプリケーション／プロセスから受信され得る。クライ

アントアプリケーションは、任意の他のＤＬＭインスタンスの存在または場所に対する可視性を有さない場合、または任意の他のＤＬＭインスタンスとの通信能力を有さない場合があることに留意されたい。

【００６１】

６３０からの肯定終了として示される、ロックが別のアプリケーションプロセスによって保持されている場合、方法は、６３５のように、クライアントアプリケーション／プロセスが、それが解放される（またはその他取得のために有効になる）までロック上の待ち行列に入ることまたはロックをポーリングすることを含むことができる。ロックが別のアプリケーション／プロセスによって保持されていない場合（６３０からの否定終了として示される）またはロックを保持する別のアプリケーション／プロセスによって解放された後）、方法は、６４０のように、クライアントアプリケーション／プロセスにロックが承諾されることと、共有リソースにアクセスすることとを含むことができる。いくつかの実施形態において、ロックを承諾することは、永続的データストア内のロック状態のコピーをアトミックに更新すること、及び／または他のＤＬＭインスタンス（これらの全ては制御プレーンネットワーク上で相互に通信する）のロック状態のローカルにキャッシュされたコピーを更新することを含むことができる。いくつかの実施形態において、アプリケーション／プロセスは、クラウドコンピューティング環境内のデータプレーンネットワーク上で共有リソースにアクセスする。

【００６２】

図６に示されるように、何らかの時点で（例えば、アプリケーションまたはプロセスが、共有リソースに対するアクセスをもはや必要としないとき）、方法は、６５０のように、クライアントアプリケーション／プロセスが、ロックを解放するために、ローカルＤＬＭインスタンスへのＡＰＩ呼び出しを行うことを含むことができる。その後、方法は、６６０のように、別のクライアントアプリケーションまたはプロセスが、ロックの状態を表示及び／または変更するために、１つ以上のローカルＤＬＭインスタンスへの１つ以上のＡＰＩ呼び出しを行うことを含むことができる。例えば、別のクライアントアプリケーションまたはプロセスは、ロックをクエリする（例えば、共有リソースがロックされているかを決定、及び／または現在の所有者を決定するために）、あるいはロックを取得（そしてその後解放）するために、ＡＰＩ呼び出しを行うことができる。

【００６３】

いくつかの実施形態において、本明細書に記載される分散ロックマネージャは、ロックが、クラウドコンピューティング環境の制御プレーンの外側で拡張されることを可能にすることができる。例えば、いくつかの実施形態において、顧客は、クラウドコンピューティング環境の外側でロックを拡張するために、（例えば、クライアントＤＬＭエージェントをダウンロードしてインストールするために）その独自の施設上でソフトウェアプロセスをダウンロード及びインストールすることが可能であってもよい。そのような実施形態において、クライアントネットワーク上のホストコンピューティングノード上で稼動しているアプリケーションは、様々なロック管理操作を実施するために、クライアントＤＬＭへのＡＰＩ呼び出しを行うことができる。いくつかの実施形態において、クライアントがＡＰＩ呼び出しを行う機構は、セキュアチャンネルであり、クライアントは、クラウドコンピューティング環境の外側でエージェントからのＡＰＩ呼び出しを行うことができる。

【００６４】

一例において、顧客は、クラウドコンピューティング環境内部にアプリケーションを構築することができるが、アプリケーションコントローラ（ロックマネージャを含む）をオンプレミス（例えば、クライアントネットワークのローカルにあるホストノード上で実行しているマシン上）に所在させたい場合がある。より具体的な例として、顧客は、クラウドコンピューティング環境内で使用するために１０のリソースインスタンスをプロビジョニングすることを要求する場合があるが、顧客は、クラウドコンピューティング環境の外側に所在するリソース（例えば、そのローカルマシン上のファイル）と関連付けられたロックを有することも希望する場合がある。本明細書に記載される分散ロックマネージャは

、顧客が、そのファイルのロックを作成し、ロック（またはファイル）の所定の共有プロパティを設定することを可能にすることができ、そのロックに対する全てのサブスクライバが、ロックの一貫した状態情報を表示することを可能にすることができる。

【００６５】

別の例において、顧客は、ローカルマシン上で稼動しているアプリケーションを有する場合があるが、ローカルマシン上の障害の場合に使用されるように、クラウドコンピューティング環境内のいくつかのリソースインスタンスをプロビジョニングすることを所望する場合がある。この例では、アプリケーションと関連付けられたロックが存在する場合があり、障害の場合に、ロックは、クラウドコンピューティング環境内のリソースインスタンスに移動され得る。

【００６６】

ＤＬＭがクラウドコンピューティング環境とローカルクライアントネットワークとの間に及ぶいくつかの実施形態において、クライアントアプリケーション及び／またはＤＬＭは、ＶＣＰ（仮想プライベートクラウド）内部に実装され得ることに留意されたい。

【００６７】

図７は、いくつかの実施形態に従う、制御プレーンに分散ロックマネージャを実装し、その環境の外側（例えば、サービスプロバイダシステムの外側）で実行しているプロセスに対して、分散ロックマネージャを公開する例示的サービスプロバイダシステムを示すブロック図である。様々な実施形態において、プロバイダシステム７００は、図３に示されるプロバイダデータセンター３００及び／または図４に示されるプロバイダシステム４００に類似し得る。例えば、図７の各ホストノード上の制御プレーンは、図３に示されたハイパーバイザまたは仮想マシンモニタの機能性のうちのいくつかまたは全てを実装することができる。同様に、図７の各ホストノード上のデータプレーンで実行しているインスタンスは、図７に示された仮想マシンの機能性のうちのいくつかまたは全てを実装する仮想コンピュートインスタンスであり得る。

【００６８】

より具体的には、示された実施例において、プロバイダシステム７００は、複数の仮想化されたデータストア（単数または複数）７４０と、ホストノード７２０及び７３０（その各々がデータプレーン部分と制御プレーン部分とを含む）と、データプレーンネットワーク７１０と、制御プレーンネットワーク７１５（異なる実施形態において、データプレーンネットワーク７１０と異なる物理的ハードウェア上に実装される場合または実装されない場合がある）とを含むことができる。

【００６９】

この実施例において、インスタンス７２８ａ～７２８ｂ、及び７３８ａ～７３８ｂは、システムの制御プレーンのそれぞれの部分（７２４及び７３４として示される）で実行し、１つ以上のクライアントアプリケーションまたはプロセスを実装することができ、そのうちの少なくともいくつかは、ロック（例えば、分散ロックマネージャ（ＤＬＭ）によって管理されるロック）によって保護される共有リソースにアクセスするように構成される。この実施例において、アプリケーション／プロセスを実装するために、これらのインスタンスは、データプレーンネットワーク７１０上で相互に、及び／または他のアプリケーション構成要素（例えば、仮想化されたデータストア（単数または複数）７４０のストレージデバイス７４５）と通信するように構成することができる。

【００７０】

図７に示される実施例において、ホストノード７２０上で実行しているインスタンス７２８ａ～７２８ｂは、様々なロック管理操作を開始するために、ＤＬＭインスタンス７２６へのＡＰＩ呼び出しを行うように構成することができ、ホストノード７３０上で実行しているインスタンス７３８ａ～７３８ｂは、様々なロック管理操作を開始するために、ＤＬＭインスタンス７３６へのＡＰＩ呼び出しを行うように構成することができる。この実施例において、ＤＬＭ（ＤＬＭインスタンス７２６及び７３６を含み、かつ仮想化されたデータストア（単数または複数）７４０へのアクセスを有する）は、システムの制御プレ

ーンにおいて（例えば、７２４及び７３４として制御プレーンのそれぞれの部分において）実行し、その構成要素は、インスタンス７２８ａ～７２８ｂ、及び７３８ａ～７３８ｂのために、共有リソースの１つ以上のロックを管理するように、制御プレーンネットワーク７１５上で相互に通信するように構成することができる。

【００７１】

図７に示されるように、いくつかの実施形態において、サービスプロバイダシステムは、（例えば、制御プレーンネットワーク７１５上の）制御プレーンにＤＬＭコーディネータ構成要素（例えば、ＤＬＭコーディネータ７５０）を含むことができる。例えば、ＤＬＭコーディネータ構成要素（制御プレーンのクラウドマネージャ構成要素のサブ構成要素であり得る）は、分散ロックサービスを提供するために一体に機能する際、ＤＬＭインスタンスのアクティビティのうちの少なくともいくつかを管理及び／または調整することができる。様々な実施形態において、ＤＬＭインスタンスは、制御プレーンネットワーク７１５上で相互に及び／またはＤＬＭコーディネータ７５０と通信することができる。例えば、ＤＬＭコーディネータ７１５は、例えば、分散ロックサービスが（例えば、最新状態のＤＬＭインスタンスの各々によってローカルにキャッシュされたロック状態情報の全てを保つために）ＤＬＭによって管理されたロックの状態の一貫した（または最終的に一貫した）ビューを維持することを促進するために、ＤＬＭインスタンス７２６及び７３６へ、及び／または７２６と７３６との間でメッセージを仲介することができる。他の実施形態において、ＤＬＭインスタンスは（少なくとも時々）、ロック状態情報を共有するために、及び／またはＤＬＭインスタンスの各々にローカルに記憶されたロック状態情報が最新状態に保たれていることを保証するために、制御プレーンネットワーク７１５上で直接相互に通信することができる。いくつかの実施形態において、ＤＬＭコーディネータ７５０は、ロック状態情報（図示せず）の永続的データストアを維持するように構成することができる。

【００７２】

図４に示された実施例とは異なり、図７に示されたプロバイダシステム７００のＤＬＭに対するインターフェースは、プロバイダシステム７００の外側で実行しているプロセスに公開され得る。この実施例において、クライアントネットワーク７７０上のホストノード７７５上で実行している様々なアプリケーション及び／またはプロセスは、様々なＡＰＩ７６０を通じて、（例えば、それらのアプリケーション／プロセス及びホストノード７２０またはホストノード７３０上で実行しているアプリケーション／プロセスによって共有リソースのロック上でロック管理操作を開始するために）ＤＬＭのロック機構に関与することが可能であり得る。本明細書に記載されるように、いくつかの実施形態において、ＤＬＭクライアントエージェントは、ホスト７７５上で実行しているアプリケーションまたはプロセスが、ＤＬＭクライアントエージェントへのローカルＡＰＩ呼び出しを使用して、ＤＬＭによって管理されたロックにアクセスすることを可能にするために、ホストノード７７５上でインスタンス化され得る。そのような実施形態において、ローカルＤＬＭクライアントエージェントが様々なローカルＡＰＩ呼び出しに応答して、これらの構成要素との通信を処理するので、プロバイダシステムの外側で実行しているアプリケーション／プロセスは、プロバイダシステム７００内部に実装されたＤＬＭ構成要素のネットワークアドレス、ネットワーク接続、及び／またはネットワークリソースについて全く知る必要なく、これらのロックの管理をサブスクライブ及び／または関与することができる。

【００７３】

図７に示された実施例において、分散ロックマネージャ（または分散ロックマネージャの構成要素によって提供された分散ロックサービス）に対して、クライアントネットワーク７７０上のホストノード７７５上で実行しているアプリケーション／プロセスによって行われたＡＰＩ呼び出し（例えば、ＡＰＩ７６０に準拠するＡＰＩ呼び出し）は、ＤＬＭコーディネータ７５０によって仲介され得て、ＡＰＩ呼び出しがルーティングされるべき特定のホストノード（または、より具体的には、特定のホストノード上のＤＬＭインスタンス）を決定することができる。他の実施形態において、クライアントネットワーク７７

０上のホストノード７７５上で実行しているアプリケーション／プロセスによって行われたＡＰＩ呼び出しは、ＤＬＭコーディネータ７５０を通じてルーティングされることなく、特定のホストノード上の特定のＤＬＭインスタンスへ向けることができる。例えば、いくつかの実施形態において、通信チャンネル（例えば、制御プレーンネットワーク７１５上のオーバーレイネットワークトンネル、または制御プレーンネットワーク７１５上の別の種類の通信チャンネル）は、（例えば、ＤＬＭコーディネータ７５０または別の制御プレーン構成要素によって仲介された通信を通じて）ホストノード７７５上で実行しているアプリケーション／プロセスと、特定のロックをサブスクライブしているアプリケーション／プロセスの結果としての特定のＤＬＭインスタンスとの間に確立され得る。サブスクリプションが承諾された、及び／または通信チャンネルが確立された後、ホストノード７７５上で実行しているアプリケーション／プロセスと特定のＤＬＭインスタンス（例えば、ＡＰＩ７６０に準拠する様々なＡＰＩ呼び出し）との間のその後の通信は、ＤＬＭコーディネータ７５０を通じてではなく、この通信チャンネル上で（例えば、直接）発生することができる。

【００７４】

クラウドコンピューティング環境の外側で実行するクライアントアプリケーションまたはプロセスが、クラウドコンピューティング環境において実装される分散ロックマネージャの機能にアクセスするための方法の一実施形態が、図８のフロー図によって示される。８１０に示されるように、この実施例において、方法は、クラウドコンピューティング環境の外側で実行しているクライアントプロセスが、クラウドコンピューティング環境内で実行している１つ以上のプロセスと共有されるリソースのロックをサブスクライブするために、クラウドコンピューティング環境内で実行する分散ロックマネージャのローカルエージェントへのＡＰＩ呼び出しを行うことを含むことができる。例えば、クライアントプロセスは、呼び出しプロセスと同じコンピューティングノード上で実行しているクライアントＤＬＭエージェントへのＡＰＩ呼び出しを行うことができる。その呼び出しに応答して、方法は、８２０のように、クラウドコンピューティング環境の外側で実行しているクライアントプロセスが、ロックの識別子（例えば、ロックハンドル）またはロックの値を受信することを含むことができ、その後、共有リソースのロックを取得するために、分散ロックマネージャのローカルエージェントへのＡＰＩ呼び出しを行うことができる。いくつかの実施形態において、これによって、（例えば、ロックのロック状態情報を他のサブスクライバと共有するために）ローカルクライアントＤＬＭエージェントと、クラウドコンピューティング環境で実行しているＤＬＭの構成要素との間の通信を開始することができる。

【００７５】

この実施例に示されるように、８３０からの肯定終了として示される、ロックが別のアプリケーションプロセスによって保持されている場合、方法は、８３５のように、クライアントプロセスが、それが解放される（またはその他取得のために有効になる）までロック上の待ち行列に入ることまたはロックをポーリングすることを含むことができる。ロックは、クラウドコンピューティング環境の外側で実行している別のプロセスによって、またはクラウドコンピューティング環境の内側で実行しているプロセスによって保持され得ることに留意されたい。ロックが別のプロセスによって保持されていない場合（８３０からの否定終了として示される）またはロックを保持する別のプロセスによって解放された後）、方法は、８４０のように、クライアントプロセスにロックを承諾することと、共有リソースにアクセスすることとを含むことができる。いくつかの実施形態において、ロックを承諾することは、永続的データストア内のロック状態のコピーをアトミックに更新すること、並びに／あるいはローカルクライアントＤＬＭエージェント及び／またクラウドコンピューティング環境内の他のＤＬＭ構成要素（これらの全ては制御プレーンネットワーク上で相互に通信する）のロック状態のローカルにキャッシュされたコピーを更新することを含むことができる。いくつかの実施形態において、ロックを承諾することは、ＤＬＭが、プロセスにロックが承諾されたことを示すロック状態値を返すことを含むことがで

きる。

【００７６】

図８に示されるように、何らかの時点で（例えば、プロセスが、共有リソースに対するアクセスをもはや必要としないとき）、方法は、８５０のように、クライアントプロセスが、ロックを解放するために、ローカルクライアントＤＬＭエージェントへのＡＰＩ呼び出しを行うことを含むことができる。いくつかの実施形態において、これによって、（例えば、ロックの状態情報を他のサブスクライバと共有するために）ローカルクライアントＤＬＭエージェントと、クラウドで実行しているＤＬＭの１つ以上の構成要素との間の通信を開始することができる。その後、方法は、８６０のように、別のプロセスが、ロックの状態を表示及び／または変更するために、クラウドコンピューティング環境の外側で実行しているローカルクライアントＤＬＭエージェントまたはクラウドコンピューティング環境の内側で実行しているローカルＤＬＭインスタンスへの１つ以上のＡＰＩ呼び出しを行うことを含むことができる。すなわち、他のプロセスは、そのプロセスに対してローカルであるＤＬＭエージェントまたはインスタンス（例えば、クラウドコンピューティング環境内またはクラウドコンピューティング環境外側の、そのプロセスと同じホストノード上で実行しているＤＬＭエージェントまたはインスタンス）へのＡＰＩ呼び出しを行うことができる。例えば、別のクライアントアプリケーションまたはプロセスは、ロックをクエリする（例えば、共有リソースがロックされているかを決定、及び／または現在の所有者を決定するために）、あるいはロックを取得（そしてその後解放）するために、ＡＰＩ呼び出しを行うことができる。

【００７７】

例示的システム

少なくともいくつかの実施形態において、本明細書に記載されるような分散ロックマネージャを実装するための技法のいくつかまたは全てを実装するサーバーが、図９に示されるようなコンピュータシステム９００等、非一時的コンピュータアクセス可能媒体を含む、またはこれにアクセスするように構成される汎用コンピュータシステムを含むことができる。示される実施形態において、コンピュータシステム９００は、入出力（Ｉ／Ｏ）インターフェース９３０を介して、システムメモリ９２０に連結される１つ以上のプロセッサ９１０を含む。コンピュータシステム９００は、Ｉ／Ｏインターフェース９３０に連結されたネットワークインターフェース９４０をさらに含む。

【００７８】

様々な実施形態において、コンピューターステム９００は、１つのプロセッサ９１０を含む単一プロセッサシステム、または複数（例えば、２、４、８、または別の適切な数）のプロセッサ９１０を含むマルチプロセッサシステムである。プロセッサ９１０は、命令を実行することが可能である任意の適切なプロセッサであり得る。例えば、様々な実施形態において、プロセッサ９１０は、ｘ８６、ＰｏｗｅｒＰＣ、ＳＰＡＲＣ、またはＭＩＰＳ　ＩＳＡ、あるいは任意の他の適切なＩＳＡ等、様々な命令セットアーキテクチャ（ＩＳＡ）のうちのいずれかを実装する汎用または埋め込みプロセッサであり得る。マルチプロセッサシステムにおいて、プロセッサ９１０の各々は一般に、同じＩＳＡを実装することができるが、必ずしも実装しなくてもよい。

【００７９】

システムメモリ９２０は、プロセッサ９１０によってアクセス可能な命令及びデータを記憶するように構成され得る。様々な実施形態において、システムメモリ９２０は、静的ランダムアクセスメモリ（ＳＲＡＭ）、同期動的ＲＡＭ（ＳＤＲＡＭ）、不揮発性／フラッシュタイプメモリ、または任意の他の種類のメモリ等、任意の適切なメモリ技術を使用して実装され得る。示された実施形態において、分散ロックマネージャを実装するための上記の方法、技法、及びデータ等、１つ以上の所望される機能を実装するプログラム命令及びデータは、コード９２５及びデータ９２６として、システムメモリ９２０内部に記憶されて示される。

【００８０】

一実施形態において、Ｉ／Ｏインターフェース９３０は、ネットワークインターフェース９４０または他の周辺インターフェースを含めて、プロセッサ９１０と、システムメモリ９２０と、デバイス内の任意の周辺機器との間で、Ｉ／Ｏトラフィックを調整するように構成され得る。いくつかの実施形態において、Ｉ／Ｏインターフェース９３０は、１つの構成要素（例えば、システムメモリ９２０）からのデータ信号を、別の構成要素（例えば、プロセッサ９１０）によって使用されるために適切な形式に変換するために、任意の必要なプロトコル、タイミング、または他のデータ変換を実施することができる。いくつかの実施形態において、Ｉ／Ｏインターフェース９３０は、例えば、周辺構成要素相互接続（ＰＣＩ）バス基準またはユニバーサルシリアルバス（ＵＳＢ）基準等、多種の周辺バスを通じて接続されたデバイスのサポートを含むことができる。いくつかの実施形態において、Ｉ／Ｏインターフェース９３０の機能は、例えば、ノースブリッジ及びサウスブリッジ等、２つ以上の別々の構成要素に分割され得る。また、いくつかの実施形態において、システムメモリ９２０へのインターフェース等、Ｉ／Ｏインターフェース９３０の機能のいくつかまたは全ては、プロセッサ９１０内に直接組み入れることができる。

【００８１】

ネットワークインターフェース９４０は、データが、コンピュータシステム９００と、例えば、図面に示される他のコンピュータシステムまたはデバイス等、ネットワーク（単数または複数）９５０に接続された他のデバイス９６０との間で交換されることを可能にするように構成され得る。様々な実施形態において、ネットワークインターフェース９４０は、例えば、イーサネット（登録商標）ワークのタイプ等、任意の適切な有線または無線汎用データネットワークを介して、通信をサポートすることができる。加えて、ネットワークインターフェース９４０は、ファイバチャンネルＳＡＮ等のストレージエリアネットワークを介して、あるいは任意の他の適切な種類のネットワーク及び／またはプロトコルを介して、アナログ音声ネットワークまたはデジタルファイバー通信ネットワーク等の電気通信網／テレフォニーネットワークを介して、通信をサポートすることができる。

【００８２】

いくつかの実施形態において、システムメモリ９２０は、本明細書に記載される分散ロックマネージャの様々な実施形態を実装するめに、図１～１２の上記のプログラム命令及びデータを記憶するように構成されたコンピュータアクセス可能媒体の一実施形態であり得る。しかしながら、他の実施形態において、プログラム命令及び／データは、受信、送信、または異なる種類のコンピュータアクセス可能媒体上に記憶され得る。一般に、コンピュータアクセス可能媒体は、磁気または光学式媒体等の非一時的記憶媒体、例えば、Ｉ／Ｏインターフェース９３０を介して、コンピュータシステム９００に連結されたディスクまたはＤＶＤ／ＣＤを含むことができる。非一時的コンピュータアクセス可能記憶媒体は、システムメモリ９２０または別のタイプのメモリとしてコンピュータシステム９００のいくつかの実施形態に含むことができる、ＲＡＭ（例えば、ＳＤＲＡＭ、ＤＤＲ　ＳＤＲＡＭ、ＲＤＲＡＭ、ＳＲＡＭ等）、ＲＯＭ等の任意の揮発性または不揮発性媒体も含むことができる。さらに、コンピュータアクセス可能媒体は、ネットワークインターフェース９４０を介して実装されてもよいように、ネットワーク及び／または無線リンク等の通信媒体を介して伝達される、電気、電磁気、またはデジタル信号等の伝送媒体または信号を含むことができる。

【００８３】

前述の説明は、以下の付記の観点からさらに理解され得る。

１．ネットワーク上で相互に連結され、かつ仮想コンピューティングサービスを１つ以上のクライアントに集合的に提供する、複数のコンピューティングノードであって、コンピューティングノードの各々が、少なくとも１つのプロセッサとメモリとを備える、複数のコンピューティングノードと、

各々がコンピューティングノードのうちの２つ以上のそれぞれのコンピューティングノード上で実行している、２つ以上の仮想コンピュートインスタンスであって、仮想コンピュートインスタンスの各々が、クライアントのために分散アプリケーションのアプリケー

ション構成要素を実装するように構成され、仮想コンピュートインスタンスが、ネットワークの少なくとも一部上で相互に通信するように構成される、２つ以上の仮想コンピュートインスタンスと、

各々が２つ以上のコンピューティングノードのそれぞれのコンピューティングノード上で実行している、２つ以上の構成要素を含む分散ロックマネージャであって、分散ロックマネージャ構成要素が、それぞれの共有リソースの１つ以上のロックの状態を共有するように構成され、１つ以上のロックの状態を共有するために、分散ロックマネージャ構成要素が、仮想コンピュートインスタンスが相互に通信するネットワークの少なくとも一部とは物理的または論理的に別個のネットワーク上で相互に通信するように構成される、分散ロックマネージャと、を備え、

アプリケーション構成要素のうちの１つが、分散アプリケーションによってアクセスされる共有リソースのロックのためのロック管理操作の実施を開始するために、アプリケーション構成要素を実装する仮想コンピュートインスタンスを実行しているコンピューティングノード上で実行している分散ロックマネージャへの呼び出しを起動するように構成される、システム。

２．分散ロックマネージャ構成要素が相互に通信するネットワークが、サービスプロバイダコンピューティング環境の制御プレーンネットワークを備え、仮想コンピュートインスタンスが相互に通信するネットワークの少なくとも一部が、サービスプロバイダコンピューティング環境のデータプレーンネットワークを備える、付記１に記載のシステム。

３．分散ロックマネージャ構成要素への呼び出しに応答して、分散ロックマネージャ構成要素が、

ロック管理操作を実施し、かつ

分散ロックマネージャ構成要素を実行しているコンピューティングノード以外のコンピューティングノード上で実行している少なくとも１つの分散ロックマネージャ構成要素へロックの結果として生じた状態を通信するように構成される、付記１に記載のシステム。

４．２つ以上のコンピューティングノードが、クライアントのための仮想プライベートネットワークを実装する、付記１に記載のシステム。

５．１つ以上のコンピュータによって、

複数のコンピューティングノードのうちの所与のコンピューティングノード上で実行している分散ロックマネージャの構成要素によって、共有リソースのロックのためのロック管理操作を実施する要求を受信することであって、該受信することが、所与のコンピューティングノード上で実行しているコンピュートインスタンスから要求を受信することを含み、コンピュートインスタンスが、コンピュートインスタンスに割り当てられたネットワークリソース能力を使用して共有リソースにアクセスする、受信することを実施することと、

要求されたロック管理操作を実施することであって、該実施することが、分散ロックマネージャ構成要素が、コンピュートインスタンスに割り当てられたネットワークリソース能力以外のネットワークリソース能力を使用して、ロックの状態情報を共有するために、複数のコンピューティングノードのうちの別のコンピューティングノード上で実行している別の分散ロックマネージャ構成要素と通信することを含む、実施することと、を含む、方法。

６．コンピュートインスタンスが、分散アプリケーションのアプリケーション構成要素を実装し、

方法が、アプリケーション構成要素が、コンピュートインスタンスに割り当てられたネットワークリソース能力を使用して、分散アプリケーションの少なくとも１つの他の構成要素と通信することをさらに含む、付記５に記載の方法。

７．複数のコンピューティングノードが、分散ロックサービスを実装する、付記５に記載の方法。

８．複数のコンピューティングノードが、１つ以上の仮想コンピューティングサービスを実装する、付記５に記載の方法。

９．ロック管理操作を該実施することが、ロックを共有リソースと関連付けることを含み、該通信することが、他の分散ロックマネージャ構成要素へ関連付けを通信することを含む、付記５に記載の方法。

１０．ロック管理操作を該実施することが、ロックの値を変更することを含む、付記５に記載の方法。

１１．ロックの値が、共有リソースの所有者を識別する、付記１０に記載の方法。

１２．分散ロックマネージャの構成要素が、ロックの変更された値を永続的データストアに書き込むことをさらに含む、付記１０に記載の方法。

１３．ロック値を該変更することが、ロック値をアトミックに変更することを含む、付記１０に記載の方法。

１４．コンピュートインスタンスに割り当てられたネットワークリソース能力が、ネットワーク接続リソースまたは入出力処理能力を含む、付記５に記載の方法。

１５．該実施することが、所与のコンピューティングノードのロックの状態情報をキャッシュすることをさらに含む、付記５に記載の方法。

１６．１つ以上のコンピュータ上で実行されると、１つ以上のコンピュータに、

複数のコンピューティングノードのうちの所与のコンピューティングノード上で実行している分散ロックマネージャの構成要素によって、共有アクセスが制御されることになる実体と関連付けられたロックのためのロック管理操作を実施する要求を受信することであって、該受信することが、所与のコンピューティングノード上で実行しているリソースインスタンスから要求を受信することを含み、複数のコンピューティングノードが、分散ロックサービスを実装し、リソースインスタンスが、ロックと関連付けられた実体にアクセスし、要求が、分散ロックマネージャによってサポートされる１つ以上のロック管理操作を定義するアプリケーションプログラミングインターフェースに準拠する、受信することと、

要求されたロック管理操作を実施することであって、該実施することが、分散ロックマネージャ構成要素が、リソースインスタンスがロックと関連付けられたエンティティにアクセスするネットワーク接続以外のネットワーク接続を使用して、ロックの状態情報を共有するために、複数のコンピューティングノードのうちの別のコンピューティングノード上で実行している分散ロックマネージャの別の構成要素と通信することを含む、実施することと、を実施させるプログラム命令を記憶する、非一時的コンピュータ可読記憶媒体。

１７．要求が、ロックを作成する要求を含み、要求されたロック管理操作を該実施することが、ロックを作成することと、要求が受信されたリソースインスタンスへロックの識別子を返すこととを含み、該通信することが、分散ロックマネージャの他の構成要素へロックの識別子を通信することを含む、付記１６に記載の非一時的コンピュータ可読記憶媒体。

１８．要求が、ロックをサブスクライブする要求、またはロックのプロパティの値を設定する要求を含む、付記１６に記載の非一時的コンピュータ可読媒体。

１９．要求が、ロックを取得する要求、またはロックを解放する要求を含み、要求された操作を該実施することが、ロックのロック値を変更することを含み、該通信することが、変更されたロック値を分散ロックマネージャの他の構成要素に通信することを含む、付記１６に記載の非一時的コンピュータ可読記憶媒体。

２０．複数のコンピューティングノードが、ネットワーク上で相互に連結され、仮想コンピューティングサービスを１つ以上のクライアントに集合的に提供し、

１つ以上のコンピュータ上で実行されると、プログラム命令が、１つ以上のコンピュータに、

ロックのためのロック管理操作を実施する第２の要求を受信することをさらに実施させ、第２の要求が、ネットワーク上で相互に連結され、１つ以上のクライアントに仮想コンピューティングサービスを集合的に提供する、複数のコンピューティングノードのうちのコンピューティングノード以外のコンピューティングサービスから受信され、要求が、分散ロックマネージャによってサポートされた１つ以上のロック管理操作を定義するアプリ

ケーションプログラミングインターフェースに準拠する、付記１６に記載の非一時的コンピュータ可読記憶媒体。

２１．ロックと関連付けられたエンティティが、仮想ネットワークインターフェースを備える、付記１６に記載の非一時的コンピュータ可読記憶媒体。

【００８４】

様々な実施形態は、コンピュータアクセス可能媒体に関する前述の説明に従って実装された命令及び／またはデータを受信すること、送信すること、または記憶することをさらに含むことができる。一般的に、コンピュータアクセス可能媒体として、磁気または光学式媒体（例えば、ディスクまたはＤＶＤ／ＣＤ－ＲＯＭ）、ＲＡＭ（例えば、ＳＤＲＡＭ、ＤＤＲ、ＲＤＲＡＭ、ＳＲＡＭ等）、ＲＯＭ等の揮発または不揮発媒体等の記憶媒体またはメモリ媒体、並びにネットワーク及び／またはワイヤレス連結等の通信媒体を介して伝達される、伝送媒体または、電気、電磁気、またはデジタル信号等の信号を挙げることができる。

【００８５】

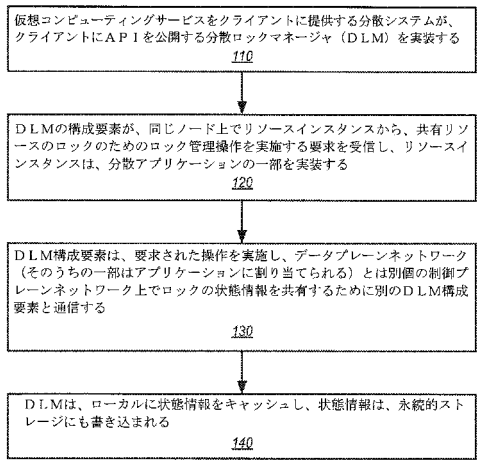
本明細書の図面及び説明において例示された様々な方法は、方法の例示的実施形態を表す。方法は、ソフトウェア、ハードウェア、またはこれらの組み合わせにおいて実装され得る。方法の順序は変更されてもよく、様々な要素が追加、順番を変更、組み合わせ、省略、修正等されてもよい。

【００８６】

本開示の利点を有する当業者には明らかであるように、様々な修正及び変更が行われ得る。全てのそのような修正及び変更を網羅すること、したがって、上記の説明は、制限的意味ではなく、例示として解釈されることが意図される。

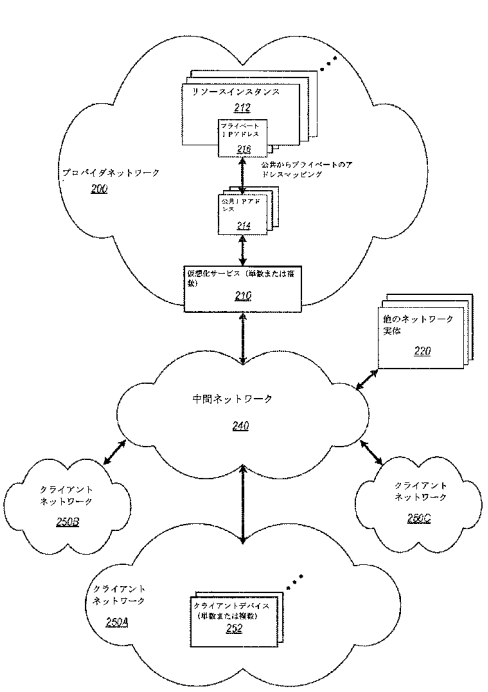
【図１】

図１



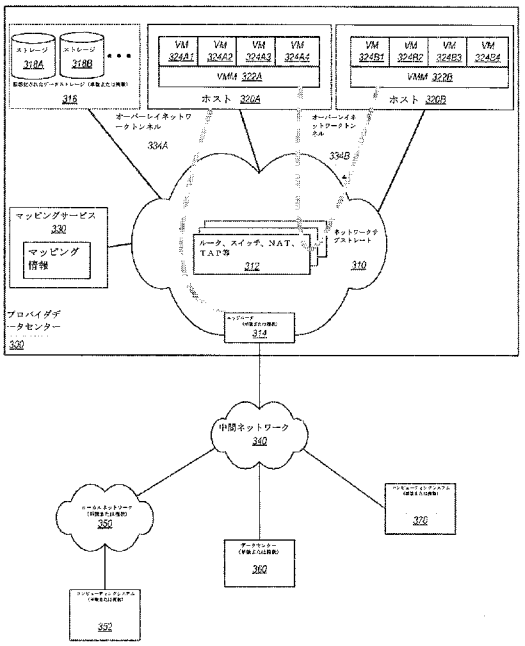
【図２】

図２



【図３】

図３



【図４】

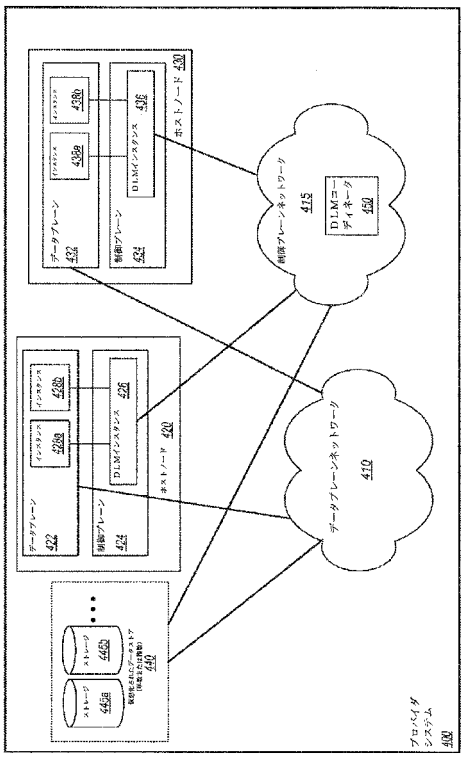
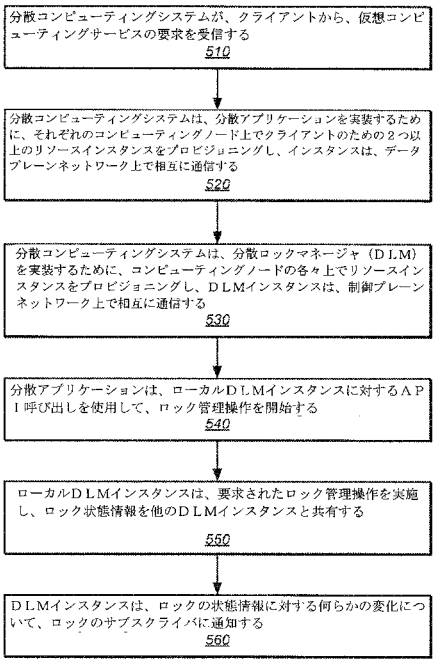


図４

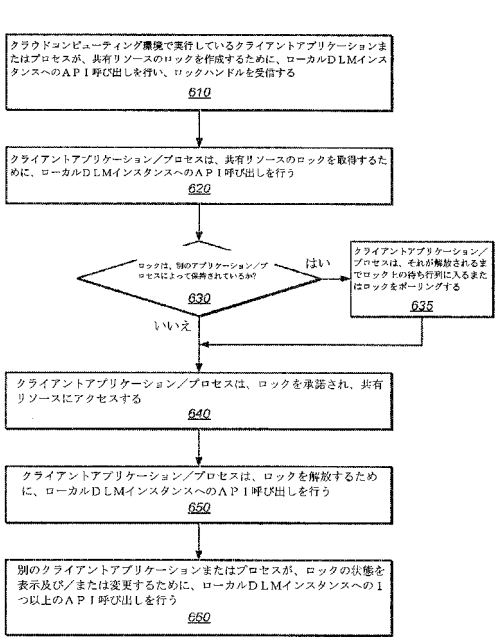
【図５】

図５



【図６】

図６



【図７】

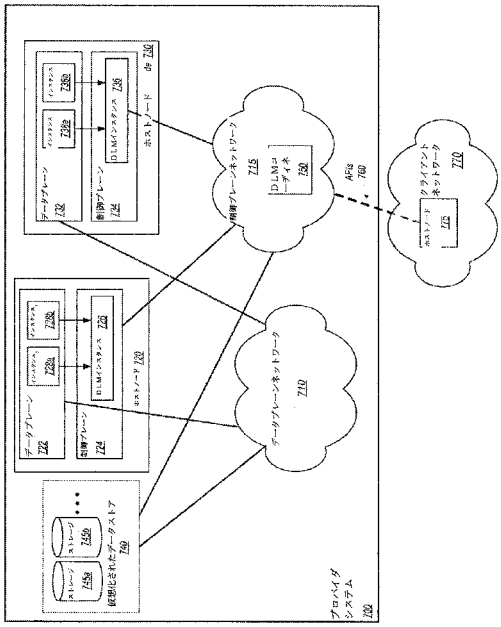
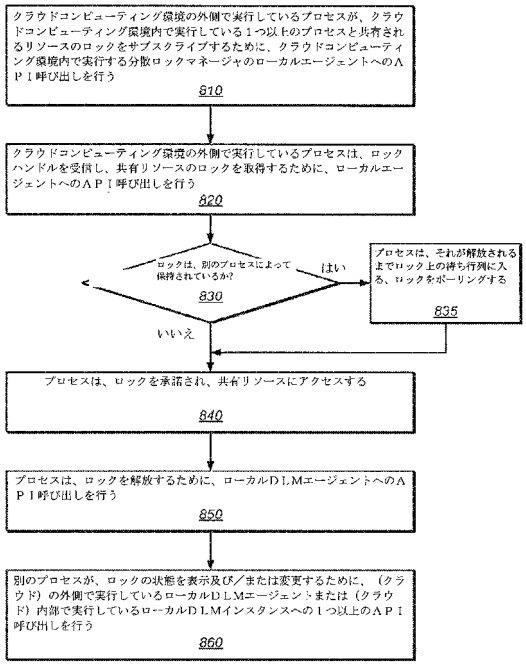


図７

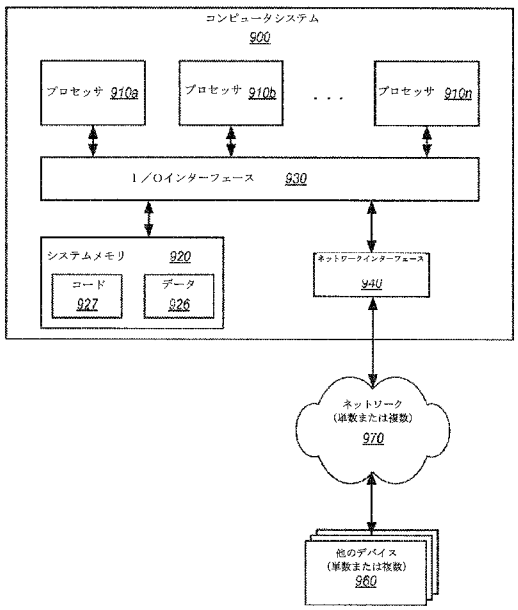
【図８】

図８



【図９】

図９



【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 14/41724

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 7/00 (2014.01)
CPC - G06F 17/30008
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)
IPC(8): G06F 7/00 (2014.01)
CPC: G06F 17/30008

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 707/704 or 707/713 or 707/687 or 707/609; IPC(8): G06F 7/00 (2014.01); CPC: G06F 17/30008 or G06F 9/526 or G06F 9/52 or G06F 9/466 or G06F 17/30067

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Patbase; Google (Web); distributed lock manager, lock, virtual, instance, resource, services, cloud, DLM, network, LAN, WAN

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2012/0310881 A1 (Shadmon) 06 December 2012 (06.12.2012) entire document (especially para [0046]-[0049], [0053]-[0059], [0067]-[0075], [0081]-[0090])	1-15
Y	US 8,458,517 B1 (Vermeulen et al.) 04 June 2013 (04.06.2013) (col. 3, ln 50-65; col. 4, ln 32-33; col. 5, ln 52-55; col. 8, ln 59 to col. 9, ln 3; col. 14, ln 9-13; col. 12, ln 35-37)	1-15
Y	US 2008/0109807 A1 (Rosenbluth) 08 May 2008 (08.05.2008) para [0042]-[0043]	2

☐ Further documents are listed in the continuation of Box C. ☐

* 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	Date of mailing of the international search report
24 September 2014 (24.09.2014)	22 OCT 2014
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (July 2009)

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,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,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,JP,KE,KG,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LT,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,TZ,UA,UG,US

【要約の続き】
用によって確立することができる。
【選択図】図４