

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-267009

(P2010-267009A)

(43) 公開日 平成22年11月25日(2010.11.25)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 21/22 (2006.01)	G06F 9/06 660C	5B276
G06F 9/46 (2006.01)	G06F 9/46 350	

審査請求 未請求 請求項の数 11 O L (全 15 頁)

(21) 出願番号 特願2009-116745 (P2009-116745)
 (22) 出願日 平成21年5月13日 (2009. 5. 13)

(71) 出願人 000001007
 キヤノン株式会社
 東京都大田区下丸子3丁目30番2号
 (74) 代理人 100114775
 弁理士 高岡 亮一
 (72) 発明者 速水 祥和
 東京都大田区下丸子3丁目30番2号 キ
 ヤノン株式会社内
 Fターム(参考) 5B276 FB01

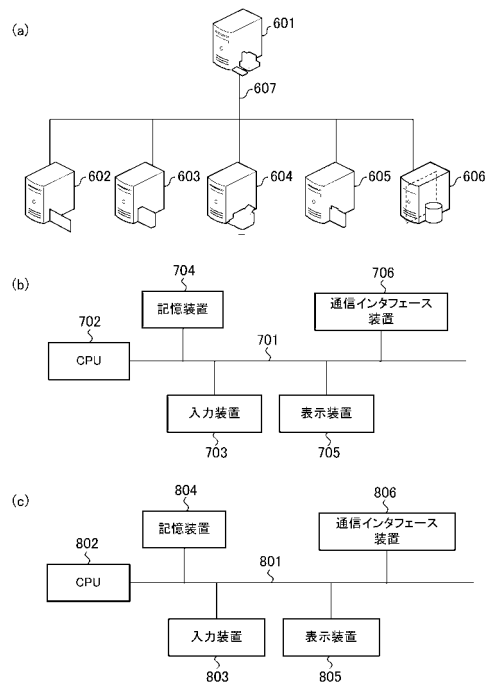
(54) 【発明の名称】 ライセンス管理システム、ライセンス管理方法及びコンピュータプログラム

(57) 【要約】

【課題】ソフトウェアベンダー側、ユーザ側双方において、ライセンスを管理できるようにすることを目的とする。

【解決手段】複数のコンピュータが接続され、アプリケーションが仮想サーバ上で起動される際に課金によってライセンスを許可するライセンス管理システムであって、アプリケーションを管理する管理モジュールは、課金のタイプを判断する手段と、判断に基づく課金のタイプに応じたライセンスの数をカウントする手段と、カウントの結果を、ライセンスを許可するサーバに送信する手段と、を有し、ライセンスを許可するサーバは、送信されたカウント数が同時に利用できるライセンス数を超えているか否かを判断する手段と、判断の結果に基づいてライセンスを許可するか否かを定める手段と、を有することを特徴とする。

【選択図】 図 1



【特許請求の範囲】**【請求項 1】**

複数のコンピュータが接続され、アプリケーションが仮想サーバ上で起動される際に課金によってライセンスを許可するライセンス管理システムであって、

前記アプリケーションを管理する管理モジュールは、前記課金のタイプを判断する手段と、

該判断に基づく課金のタイプに応じたライセンスの数をカウントする手段と、

該カウントの結果を、前記ライセンスを許可するサーバに送信する手段と、を有し、

該ライセンスを許可するサーバは、該送信されたカウント数が同時に利用できるライセンス数を超えているか否かを判断する手段と、

該判断の結果に基づいてライセンスを許可するか否かを定める手段と、を有することを特徴とするライセンス管理システム。

10

【請求項 2】

前記課金のタイプを判断する手段は、前記管理モジュールが実サーバである場合、CPU数に依存した前記課金のタイプか否かを判断し、前記ライセンスの数をカウントする手段は該判断に基づく課金のタイプに応じたライセンスの数をカウントすることを特徴とする請求項 1 記載のライセンス管理システム。

【請求項 3】

前記管理モジュールが仮想サーバである場合、前記課金のタイプが前記仮想サーバ単位であり、前記ライセンスの数が前記仮想サーバに割り当てられたCPU数分なら、前記ライセンスの数をカウントする手段は前記仮想サーバと該割り当てられたCPU数からライセンスの数をカウントすることを特徴とする請求項 1 記載のライセンス管理システム。

20

【請求項 4】

前記管理モジュールが仮想サーバである場合、前記課金のタイプが前記仮想サーバ単位であり、前記ライセンスが前記仮想サーバごとに割り当てられているなら、前記ライセンスの数をカウントする手段は前記仮想サーバの数から前記ライセンスの数をカウントすることを特徴とする請求項 1 記載のライセンス管理システム。

【請求項 5】

前記管理モジュールが仮想サーバである場合、前記課金のタイプが前記仮想サーバ単位であり、前記ライセンスの数が前記仮想サーバの数と実CPUの数とを乗じた数ならば、前記ライセンスの数をカウントする手段は前記仮想サーバと前記実CPU数とから前記ライセンスの数をカウントすることを特徴とする請求項 1 記載のライセンス管理システム。

30

【請求項 6】

前記仮想サーバが移動する場合、前記ライセンスの数をカウントする手段は、該移動した状態において前記ライセンスの数をカウントし、

前記移動の結果、前記ライセンスの数が足りなくなった場合、該足りない旨を警告する手段をさらに有することを特徴とする請求項 1 記載のライセンス管理システム。

【請求項 7】

前記仮想サーバへのCPUの割り当ての数が変更された場合、前記ライセンスの数をカウントする手段は、該移動した状態において前記ライセンスの数をカウントし、

前記変更の結果、前記ライセンスの数が足りなくなった場合、該足りない旨を警告する手段をさらに有することを特徴とする請求項 1 記載のライセンス管理システム。

40

【請求項 8】

複数のコンピュータが接続され、アプリケーションが仮想サーバ上で起動される際に課金によってライセンスを許可するライセンス管理システムの前記アプリケーションを管理する管理モジュールであって、

前記課金のタイプを判断する手段と、

該判断に基づく課金のタイプに応じたライセンスの数をカウントする手段と、

該カウントの結果を、前記ライセンスを許可するサーバに送信する手段と、を有することを特徴とする管理モジュール。

50

【請求項 9】

複数のコンピュータが接続され、アプリケーションが仮想サーバ上で起動される際に課金によってライセンスを許可するライセンス管理システムの前記ライセンスを許可するサーバであって、

該送信されたカウント数が同時に利用できるライセンス数を超過しているか否かを判断する手段と、

該判断の結果に基づいてライセンスを許可するか否かを定める手段と、を有することを特徴とするサーバ。

【請求項 10】

複数のコンピュータが接続され、アプリケーションが仮想サーバ上で起動される際に課金によってライセンスを許可するライセンス管理方法であって、

前記アプリケーションを管理する管理モジュールは、前記課金のタイプを判断するステップと、

該判断に基づく課金のタイプに応じたライセンスの数をカウントするステップと、

該カウントの結果を、前記ライセンスを許可するサーバに送信するステップと、を有し、

該ライセンスを許可するサーバは、該送信されたカウント数が同時に利用できるライセンス数を超過しているか否かを判断するステップと、

該判断の結果に基づいてライセンスを許可するか否かを定めるステップと、を有することを特徴とするライセンス管理方法。

【請求項 11】

請求項 10 記載のライセンス管理方法をコンピュータに実行させることを特徴とするコンピュータプログラム。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、ライセンス管理システム、ライセンス管理方法及びコンピュータプログラムに関し、特に、仮想サーバの性質に対応したライセンス課金を行えるライセンス管理システム、ライセンス管理方法及びコンピュータプログラムに関する。

【背景技術】**【0002】**

現在、以下に述べる理由から、サーバの仮想化を導入する企業が増えている。サーバを仮想化する最大の利点は、サーバ 1 台の中に多数の「仮想サーバ」を作ること、CPU の処理能力を無駄なく利用できる点である。

【0003】

ここで、サーバの仮想化とは、1 台のサーバコンピュータを複数台の仮想的なコンピュータに分割し、それぞれに別の OS やアプリケーションソフトを動作させる技術のことを言う。プロセッサやメモリ、ディスクをまとめて、仮想的に複数の領域に分割し、それぞれがあたかも 1 台であるかのように振る舞い、異なる OS やアプリケーションを同時に実行できる。物理的に複数のコンピュータを用意する場合に比べ、物理的資源の管理にかかる手間が省け、資源配分を需要に応じて柔軟に配分することができるという利点がある。

【0004】

しかし、仮想化にかかるオーバーヘッドの分、直接実行するよりも性能面で劣る。CPU の処理能力を無駄なく利用できるという利点を生かしたのが、多数の PC サーバを集約するサーバ統合である。企業内の PC サーバは、Web サイトから基幹業務システムまで増加の一途をたどっている。こうした PC サーバの大半は、そのリソースを最大限に生かしていなかった。平均的なサーバ利用率は 10% ~ 20% 程度に留まっているという調査がある。つまり、サーバが本来発揮できるはずの処理能力の 80% ~ 90% 程度は使われないまま無駄になっているわけである。

【0005】

理由としては、運用の安定性向上の面、システム管理の単純明快さを確保する上からも、アプリケーションごと、用務ごとにハードウェアレベルで別サーバを用意することが多かったからである。一方、台数が増加することに起因する問題としては、管理対象が増加することによる管理負担の増大や、設置スペース/電力消費/発熱量の増大などが挙げられる。こういう状況を解決する手段として、仮想化技術が注目されてきた。仮想化技術を使えば、1台につき10～20台の仮想サーバを収容でき、サーバ台数を劇的に削減できる。また、利用効率をほぼ100%に近いところまで引き上げることができ、運用コストも削減できる。

【0006】

2点目の理由としては、プロセッサのマルチコア化の進展が関連している。マルチコア化によって内部での並列演算性能は大幅に高速化されたが、一方でソフトウェアの対応がまだ追いついていないという問題がある。具体的には、従来のソフトウェアはマルチコアプロセッサに最適化された構造にはなっていないため、急激に並列度を増したプロセッサの演算リソースを使い切れないということが生じている。端的に言えば、マルチコアプロセッサで実行しても、コアを使い切ることができず、遊んでしまうコアが生じてしまう懸念がある。サーバの仮想化技術は、マルチコアプロセッサの演算性能を使いこなすための効果的な手段の一つと考えることもできる。

10

【0007】

3点目の理由としては、システムの寿命の観点からの理由が考えられる。使い慣れた業務システムがWindows(登録商標)NTのような旧型のOSで稼動しており、延命策を考えているケースは少なくない。サポートの切れた旧型ハードから新型サーバに寄せ換えたくても、旧型OSがサポート対象外のこともある。かといって、業務システムを最新のOSに寄せ換える作業は開発の手間がかかるし、移行は難しいことが多い。このようなケースで仮想化技術を使うと、新型のサーバの上で旧型OSを仮想サーバとして立てることにより動作させることができる。

20

【0008】

4点目の理由としては、耐障害性の観点からの理由が考えられる。サーバにソフトウェア的な障害が発生すると、マシンをダウンさせ再起動をする必要がある。これを速やかに行うためには、予め論理的に全く同じサーバを立ち上げておき、瞬時に切り替える必要がある。このような運用が、仮想化技術のサーバの移動機能を使えば、容易に実現可能である。

30

【0009】

5点目の理由としては、時間帯によってCPUの処理能力やメモリ容量を切り替えられるということが挙げられる。例えば、オンライン系サーバは午前中又は月曜日に負荷のピークが集中し、バッチ系サーバは夜間や土日に負荷のピークが集中することがよくある。仮想化技術を使えば、仮想マシン間で利用できるCPU数やCPUパワー、メモリ容量を変更できる。その設定は、曜日や月末といった状況に応じて細かく変更することも容易である。このような利用法は、物理的に異なるサーバでシステムを構築した場合と比べて、コストを大幅に削減できる。

【0010】

このような背景に伴い、仮想サーバ市場が急速に成長してきており、2011年にx86サーバの4割近くが仮想化されるという予測もある。また、多種多様な仮想化ソフト(VMWare、Hyper-V、Xen)が存在している。

40

【0011】

サーバの仮想化の方式には、以下の二種類がある。ホストOS方式とハイパーバイザー方式である。ホストOS方式は、仮想ハードウェアを設けずに、実際の周辺機器などをそのまま利用し、ホストOS上で複数のゲストOSを実行可能にする方式である。また、ハイパーバイザー方式は、仮想化を実現する仮想化ソフトと仮想ハードウェアのソフトウェアレイヤーをハードウェア上に実装し、その上に複数のOSをインストールして利用するものである。

50

【 0 0 1 2 】

次に、仮想サーバにおける一般的なライセンス課金方法について説明する。一般にソフトウェアの課金は、「ユーザ数」に基づくものと、サーバの「CPU数(コア数)」に基づくものがある。「ユーザ数」による課金の場合は、ユーザ数を変えない限り、ソフトウェアを仮想サーバに移行させてもライセンスコストは同じになる。このようなネットワーク環境で使用するソフトウェアでは、不特定多数の人がアクセスできるためライセンスにより使用できるコンピュータや本数を制限しているものが多い。

【 0 0 1 3 】

ソフトウェアライセンスの形態としては以下の2種類が主に使用されている。すなわち、特定のコンピュータのみで実行可能なライセンスであるノードロックライセンスとネットワーク上のどのコンピュータでも実行可能なライセンスであるフローティングライセンスである。フローティングライセンスについては、特許文献1に記載されている。フローティングライセンス方式は、複数のコンピュータをネットワークにより相互に通信可能に接続したシステムに適用される方式である。ノードロックライセンス方式は、各コンピュータのホストIDを用いてライセンスを管理する方式である。

10

【 先行技術文献 】

【 特許文献 】

【 0 0 1 4 】

【 特許文献 1 】 特開 2 0 0 1 - 0 9 2 5 5 0 号 公 報

【 発明の概要 】

20

【 発明が解決しようとする課題 】

【 0 0 1 5 】

しかしながら、従来のフローティングライセンス技術では、上記した仮想サーバを含む、仮想化特有のライセンス課金体系を考慮したものではなかった。また、仮想化の機能として、仮想サーバに割り当てるCPU数を昼と夜とで変更するといった設定をできるということが挙げられる。仮想サーバに割り当てたCPU数に依存したライセンス体系の場合は、このCPUの割り当てスケジューリングのことを考慮する必要がある。このような仮想化特有の技術により、複雑になった課金方法は、現状ソフトウェアベンダーとユーザ側で、課金体系を紳士協定的に遂行されていることが多い状況である。しかしながら、成長著しい仮想サーバ環境を考えると、ソフトウェアベンダーにとって、正確で、かつ多様な仮想化ソフトに対応できる汎用的なライセンス管理方法の必要性が増してくる。また、ユーザにとっても、ライセンス使用状況の把握の負担が増してきている。そこで、本発明は、ソフトウェアベンダー側、ユーザ側双方において、ライセンスを管理できるようにすることを目的とする。

30

【 課題を解決するための手段 】

【 0 0 1 6 】

上記の課題を解決するため、本発明は、複数のコンピュータが接続され、アプリケーションが仮想サーバ上で起動される際に課金によってライセンスを許可するライセンス管理システムであって、前記アプリケーションを管理する管理モジュールは、前記課金のタイプを判断する手段と、該判断に基づく課金のタイプに応じたライセンスの数をカウントする手段と、該カウントの結果を、前記ライセンスを許可するサーバに送信する手段と、を有し、該ライセンスを許可するサーバは、該送信されたカウント数が同時に利用できるライセンス数を超えているか否かを判断する手段と、該判断の結果に基づいてライセンスを許可するか否かを定める手段と、を有することを特徴とする。

40

【 発明の効果 】

【 0 0 1 7 】

本発明によれば、仮想サーバを含むシステム全体で、課金方法が混在したような環境、又は動的にライセンスが変わるような環境において、同時に利用可能なライセンスカウントが可能となり、システム全体のライセンス管理を行うことが可能となる。

【 図面の簡単な説明 】

50

【 0 0 1 8 】

【 図 1 】 本発明の一実施形態の構成を示す図である。

【 図 2 】 仮想化方式を示す図である。

【 図 3 】 仮想サーバにおける課金方法を示す図である。

【 図 4 】 仮想サーバの移動機能を示す図である。

【 図 5 】 端末装置 6 0 2 - 6 0 6 に内蔵される仮想サーバの様子を示す図である。

【 図 6 】 仮想環境ライセンス管理モジュール及びライセンスサーバがもつ情報内容を示す図である。

【 図 7 】 仮想サーバ上でアプリケーションが起動されるとき処理を示すフローチャートである。

10

【 図 8 】 仮想サーバの移動機能を実行したときの全体構成を示す図である。

【 図 9 】 仮想サーバの移動機能を実行したときの処理を示すフローチャートである。

【 図 1 0 】 C P U 割り当て数に変更されたときの処理を示すフローチャートである。

【 図 1 1 】 実サーバと仮想サーバが混在したライセンス管理システムの全体構成を示す図である。

【 図 1 2 】 仮想サーバ及び実サーバ上でライセンス管理モジュールがもつ情報を示す図である。

【 発明を実施するための形態 】

【 0 0 1 9 】

以下、本発明を実施するための最良の形態について、添付図面等を参照して詳細に説明する。なお、以下の説明では、具体的な数値、構成、動作等を示して説明を行うが、これらは、適宜変更することができる。

20

【 0 0 2 0 】

< 実施形態 1 >

図 1 (a) は、本発明の一実施形態としてのネットワークにおけるライセンス管理システムの全体構成図である。図 1 (a) に示すように、ネットワークにおけるライセンス管理システムは、ユーザのサーバ各種端末装置 6 0 2 - 6 0 6 と、ライセンスサーバ 6 0 1 とを備えている。端末装置 6 0 2 - 6 0 6 とライセンスサーバ 6 0 1 とはネットワーク 6 0 7 を介して相互に通信可能に接続されている。なお、ネットワーク 6 0 7 は、例えば LAN から構成されている。

30

【 0 0 2 1 】

図 1 (b) は、ライセンス管理装置の一例であるライセンスサーバ 6 0 1 の構成を示す図である。図 1 (b) に示すように、ライセンスサーバ 6 0 1 は、ライセンスサーバ 6 0 1 全体を制御する CPU 7 0 2 を有している。CPU 7 0 2 が、記憶装置 7 0 4 に記憶されているプログラムに基づき動作することによって、例えば、後記するライセンスサーバ 6 0 1 に関するフローチャートの各処理を実行する。CPU 7 0 2 には、バス 7 0 1 を介して、キーボードやマウス等からなる入力装置 7 0 3 と、記憶装置 7 0 4 と、表示装置 7 0 5 と、通信インタフェース装置 7 0 6 とが接続されている。記憶手段の一例である記憶装置 7 0 4 は、例えば、ROM、RAM、ハードディスク装置等からなる。記憶装置 7 0 4 は、端末装置 6 0 2 - 6 0 6 で使用可能なアプリケーションの種類、アプリケーションが使用できるライセンス数、残ライセンス数を記憶している。表示装置 7 0 5 は、CRT 又は LCD 等からなる。通信インタフェース装置 7 0 6 は、端末装置 6 0 2 - 6 0 6 との間でデータや信号の授受を可能にする。

40

【 0 0 2 2 】

図 1 (c) は、情報処理装置の一例である端末装置 6 0 2 - 6 0 6 の構成を示す図である。図 1 (c) に示すように、端末装置 6 0 2 - 6 0 6 は、端末装置全体を制御する CPU 8 0 2 を有している。CPU 8 0 2 が、記憶装置 8 0 4 に記憶されているプログラムに基づき動作することによって、例えば、後記する端末装置に関するフローチャートの各処理を実行する。CPU 8 0 2 には、バス 8 0 1 を介して、キーボードやマウス等からなる入力装置 8 0 3 と、記憶装置 8 0 4 と、表示装置 8 0 5 と、通信インタフェース装置 8

50

06とが接続されている。記憶手段の一例である記憶装置804は、例えば、ROM、RAM、ハードディスク装置等からなる。記憶装置804は、端末装置602-606で実行されるアプリケーションコンポーネントや動作機能追加に関するプログラム(プラグイン)を記憶している。表示装置805は、CRT又はLCD等からなる。通信インタフェース装置806は、ライセンスサーバ801との間でデータや信号の授受を可能にする。

【0023】

ここで、サーバの仮想化方式について説明する。サーバの仮想化方式には、代表的なものとして、ホストOS方式とハイパーバイザー方式とがある。図2は、これらの仮想化方式を示す図である。

【0024】

図2(a)は、ホストOS方式を示す図である。ホストOS102のミドルウェアとして仮想化ソフト103が動作する。仮想化ソフトは、物理ハードウェアをもとに仮想ハードウェア104を作り、そこにOS105やアプリケーション106が導入できるようにするソフトウェアである。OS105には、ホストOSと同一又は異なるOSをゲストOSとしてインストールすることが可能である。ホストOS方式では、ハードウェアとの入出力にホストOSの機能を活用する。ただし、仮想マシンとハードウェアの間にフル機能のOSが介在するため、オーバーヘッドがやや大きくなる傾向にある。製品としては、VMWare Server、Virtual Server、Virtual PCなどがある。

【0025】

図2(b)は、ハイパーバイザー方式を示す図である。上記のホストOS方式で利用していた仮想化レイヤーをホストOSの代わりに直接物理ハードウェア上に展開したようなイメージである。このハイパーバイザーは、一種のOSともいえるが、ユーザインタフェースやユーザコマンドなどの要素は省略し、ハードウェアの管理と仮想環境の管理に特化しているため、ごく小規模で簡略化されたものとなっている。製品としては、VMWare ESX、Xen Server、Hyper-Vなどがある。

【0026】

次に、仮想サーバにおける一般的なライセンス課金方法について説明する。上記したように、一般にソフトウェアの課金は、「ユーザ数」に基づくものと、サーバの「CPU数(コア数)」に基づくものがある。「ユーザ数」による課金の場合は、ユーザ数を変えない限り、ソフトウェアを仮想サーバに移行させてもライセンスコストは同じになる。仮想環境に移行させるときに注意が必要なのは、CPUやサーバ数に基づく課金体系の場合である。このとき、ソフトウェアは「仮想サーバ単位」又は「物理サーバ単位」で課金される。

【0027】

一方、コンピュータのネットワーク化が進み、分散した資源にネットワークを介してアクセスすることが可能になってきている。コンピュータで実行するソフトウェアに関してもネットワーク環境での使用を前提にしたものが増え、ネットワーク上のマシンであれば何処からでも実行できるようになってきている。このようなネットワーク環境で使用するソフトウェアでは、不特定多数の人がアクセスできるためライセンスにより使用できるコンピュータや本数を制限しているものが多い。

【0028】

ソフトウェアライセンスの形態としては特定のコンピュータのみで実行可能なライセンス(ノードロックライセンス)と、ネットワーク上のどのコンピュータでも実行可能なライセンス(フローティングライセンス)の2種類が主に使用されている。

【0029】

フローティングライセンス方式は、複数のコンピュータをネットワークにより相互に通信可能に接続したシステムに適用される方式である。そのコンピュータのうちサーバとなるものにライセンス確認プログラムが常駐しており、クライアントでアプリケーションを起動すると、そのアプリケーションがサーバのライセンス確認プログラムを呼び出す。呼

10

20

30

40

50

び出されたライセンス確認プログラムは、クライアントのアプリケーションから送られてくる情報に基づいて動作環境をチェックし、適正であればクライアントのアプリケーションに動作許可を与える。したがって、この方式によりネットワークに接続された複数のコンピュータにおいて同時に同じアプリケーションを動作させるためには、同時に起動させるアプリケーションの数だけライセンスが必要となる。

【0030】

ノードロックライセンス方式は、各コンピュータのホストIDを用いてライセンスを管理する方式である。該当ホストで実行されるならば、ユーザ名やユーザ数に制限がない方式である。つまり、アプリケーションは、予め登録されたコンピュータのみで起動される。この方式により複数のコンピュータにおいて同時に同じアプリケーションを動作させるためには、そのコンピュータの数だけライセンスが必要となる。

10

【0031】

次に、仮想サーバで稼動する現在一般的になっているソフトウェアのライセンスについて説明する。

【0032】

図3は、仮想サーバにおける課金方法を示す図である。図3に示すような5CPU301をもつサーバPCに、仮想サーバ3台を立て(302)、それぞれのサーバ上にアプリケーション303をインストールした構成で考えると、以下のような課金タイプが一般的である。

1. 実マシンのCPU数(コア数) 5ライセンス(304)
2. アプリケーションをインストールした仮想サーバに割り当てたCPU数(コア数)分 4ライセンス(305)
3. アプリケーションをインストールした仮想サーバの台数分 3ライセンス(306)
4. 実マシンのCPU数(コア数)分×アプリケーションをインストールした仮想サーバの台数分。 15ライセンス(307)

20

【0033】

現状このような課金方法がアプリケーションによって異なるというのが実情であり、このことがユーザにとってライセンス体系が複雑でわかりにくいものとしている。また、仮想化特有の機能として、移動機能というものがある(VMWareのVMotion、XenのLiveMigrationなど)。移動機能を使うと、あるサーバ上で稼動中の仮想サーバをオンライン状態で別のサーバに移動できる。本番サーバに対してバックアップサーバを用意しておき、フェイルオーバーや計画停止などに利用される。この機能を利用する際に、ライセンスの考え方としてバックアップサーバの分も考慮に入れるのが一般的である。

30

【0034】

図4は、仮想サーバの移動機能を示す図である。例えば、図4(a)のように、本番サーバ401で、仮想サーバA402が2CPU403割り当てられており、そこで、アプリケーション404がインストールされているとする。移動先のバックアップサーバ405では割り当てるCPU数が3CPU406になっていたとすると、最大利用CPUの3ライセンスが必要となる。図4(b)のように、物理サーバ単位で課金されるアプリケーションでは、バックアップサーバにもライセンスが必要になるケースがある。この例の場合、実CPU501を四つもつ本番サーバ502に仮想サーバA503、仮想サーバB504が複数台あり、それらにアプリケーション505がインストールされているとする。そのうちの一つの仮想サーバAが、実CPU506を四つもつバックアップサーバ507に移動する場合を考える。その場合、移動先の仮想サーバA508のライセンスも必要となり、本番サーバの実CPU四つとバックアップサーバの実CPU四つとを併せて、8ライセンスが必要となる。

40

【0035】

図5は、上記のサーバ各種端末装置602-606に内蔵される仮想サーバの様子を示

50

す図である。図 5 に示す仮想サーバは上記のホスト OS 方式である。各端末装置 602 - 606 には、実サーバで運用されているサーバ 901 と、仮想サーバ 904 をそれぞれ二つたてた運用を行っているサーバ 902 及び 903 とが内蔵されている。実サーバ 901 及び仮想サーバ 904 には、アプリケーション 905 がインストールされているものとする。各仮想サーバを含むサーバ上に仮想環境ライセンス管理モジュール 906 を配置する。このモジュールが全ての仮想サーバ上で動作しているアプリケーションのライセンス数をカウントする。そのために、仮想環境ライセンス管理モジュール 906 は、図 6 (a) に示すような情報 1001、1002 をもつ。

【0036】

図 6 は、仮想環境ライセンス管理モジュール及びライセンスサーバがもつ情報内容を示す図である。情報 1001 は予め仮想環境ライセンス管理モジュールに設定されている情報であり、その仮想環境上で管理しているアプリケーションのライセンス課金タイプ、仮想環境ライセンス管理モジュールが管理しているアプリケーション情報などである。サーバ 902、サーバ 903 のような異なるサーバでは、この課金タイプが異なってもよい。また、情報 1002 は仮想化ソフトを通じ、実 CPU 数、仮想サーバ数、各仮想サーバに割り当てられている CPU 数、バックアップサーバの有無などの情報である。これらの情報からライセンスカウントを計算することが可能である (1003)。実サーバ 901 のような仮想サーバをもたない実サーバ上にインストールされているアプリケーションは、それ自身がライセンスカウント情報をシステム全体のライセンスをカウントするライセンスサーバに送信し、利用の可否を問い合わせる (907)。同様に、サーバ 902、サーバ 903 のような仮想サーバをもつサーバ上でライセンスをカウントし、このライセンスカウント情報をシステム全体のライセンスをカウントするライセンスサーバに送信し、利用の可否を問い合わせる (908)。図 6 (b) はライセンスサーバがもつ情報を示す。ライセンスサーバは、システム全体で同時利用可能なライセンス数の情報 1101 をもつ。また、現在利用されているライセンス数を把握している。アプリケーション又は仮想環境ライセンス管理モジュールから利用の可否が問い合わせされると、送信されてきたライセンスカウント情報から、現在利用されているライセンス数を更新する。その値と予め設定されている同時利用可能ライセンス数とを比較、利用の可否の判断を行い (1102)、その結果をアプリケーション又は仮想環境ライセンス管理モジュールに戻す。

【0037】

図 7 は、仮想サーバ上でアプリケーションが起動されることを示すフローチャートである。まず、アプリケーションが起動されると (S 1201)、アプリケーションは仮想環境ライセンス管理モジュールに利用の可否を問い合わせる (S 1202)。仮想環境ライセンス管理モジュールは、自己の課金タイプに応じたライセンスカウントを行う。以下、仮想環境ライセンス管理モジュールの課金タイプに応じたライセンスカウントを数える方法について詳細に説明する。

【0038】

まず、課金タイプを判断する (S 1203)。仮想サーバ単位なら仮想サーバの数を判断する (S 1204)。ライセンス数がアプリケーションをインストールした仮想サーバに割り当てられた CPU 数分なら、仮想サーバと割り当てられた CPU 数からライセンスをカウントする (S 1206)。ライセンスが仮想サーバごとに割り当てられているなら、仮想サーバの数からカウントする (S 1207)。ライセンス数が仮想サーバ数と実 CPU 数とを乗じた数ならば、仮想サーバと実 CPU 数とからカウントする (S 1208)。また、課金タイプが物理サーバ単位なら、実 CPU 数からライセンス数をカウントする (S 1205)。その結果をライセンスサーバに送信し、利用の可否を問い合わせる (S 1209)。ライセンスサーバは、ライセンス数を受信し、現在利用されているライセンス数を更新する (S 1210)。同時利用数に上限がある場合、カウント数の累積が同時利用可能数の上限を超えているかどうかを判断する (S 1211)。その結果が NG の場合、仮想環境ライセンス管理モジュールにその結果を返却する (S 1212)。さらに、その結果をアプリケーションに返却する (S 1213)。アプリケーションは起動に失敗する (

10

20

30

40

50

S 1 2 1 4)。一方、カウント数の累積が同時利用可能数の上限を超えている場合、仮想環境ライセンス管理モジュールにその結果を返却する (S 1 2 1 5)。さらに、その結果をアプリケーションに返却する (S 1 2 1 6)。アプリケーションは起動に成功する (S 1 2 1 7)。

【 0 0 3 9 】

以上、説明したように本実施形態によれば、仮想サーバを含むネットワークシステムにおけるフローティングライセンスサーバにおいて使用することができる。なお、本実施形態では、仮想サーバはホストOS方式で行っていることとしたが、ハイパーバイザー方式の場合でも同様に適用可能である。

【 0 0 4 0 】

< 実施形態 2 >

図 8 は、仮想サーバの移動機能を利用したバックアップサーバ運用を想定した仮想サーバ (ホストOS方式) を含むネットワークにおけるライセンス管理システムを示したものである。本実施形態のライセンス管理システムは、ライセンスサーバ 6 0 1 でネットワーク全体を管理している。本番サーバ 1 3 0 1 は、仮想サーバ 1 3 0 2、1 3 0 3 をたてた運用を行っている。バックアップサーバ 1 3 0 4 でバックアップをとる。仮想サーバ 1 3 0 2、1 3 0 3 には、アプリケーション 1 3 0 6 がインストールされているものとする。各仮想サーバを含むサーバ上に仮想環境ライセンス管理モジュール 1 3 0 7 を配置する。仮想サーバ 1 3 0 2 がバックアップサーバへ移動されると (1 3 0 8)、ライセンスカウント数の状況が変更される可能性があるため、仮想環境ライセンス管理モジュールは、自己が管理しているアプリケーションのライセンス数を再カウントする。このモジュールが全ての仮想サーバ上で動作しているアプリケーションのライセンス数をカウントする。このライセンスカウント情報をシステム全体のライセンスをカウントするライセンスサーバに送信し、利用の可否を問い合わせる (1 3 0 9)。

【 0 0 4 1 】

図 9 は、仮想サーバの移動機能が使用されたときを示すフローチャートである。まず、仮想サーバが移動すると (S 1 4 0 1)、仮想環境ライセンス管理モジュールは、自己が管理しているアプリケーションのライセンス利用状況を再確認する (S 1 4 0 2)。仮想環境ライセンス管理モジュールは、自己の課金タイプに応じたライセンスカウントを行う。以下、ライセンス管理モジュールの課金タイプに応じたライセンスカウントを数える方法について詳細に説明する。まず、課金タイプを判断する (S 1 4 0 3)。仮想サーバ単位なら、仮想サーバの数え方を判断する (S 1 4 0 4)。ライセンス数がアプリケーションをインストールした仮想サーバに割り当てられた CPU 数分なら、仮想サーバと割り当て CPU 数からライセンスをカウントする (S 1 4 0 6)。ライセンスが仮想サーバごとに割り当てられているなら、仮想サーバの数からカウントする (S 1 4 0 7)。ライセンス数が仮想サーバ数と実 CPU 数とを乗じた数ならば、仮想サーバと割り当て CPU 数とからカウントする (S 1 4 0 8)。また、課金タイプが物理サーバ単位なら、実 CPU 数からライセンス数をカウントする (S 1 4 0 5)。その結果をライセンスサーバに送信し、利用の可否を問い合わせる (S 1 4 0 9)。ライセンスサーバは、ライセンス数を受信し現在利用されているライセンス数を更新する (S 1 4 1 0)。同時利用数に上限がある場合、カウント数の累積が同時利用可能数の上限を超えているかどうかを判断する (S 1 4 1 1)。一方、カウント数の累積が同時利用可能数の上限を超えている場合、仮想環境ライセンス管理モジュールにその結果を返却され、仮想サーバの移動が行われる (S 1 4 1 2)。そして、移動の結果、ライセンスが足りなくなった旨の警告を表示する (S 1 4 1 3)。超えていない場合、仮想環境ライセンス管理モジュールにその結果を返却する (S 1 4 1 4)。

【 0 0 4 2 】

以上説明したように本実施形態によれば、仮想サーバの移動機能を実施したときでもライセンス管理を行うことが可能である。

【 0 0 4 3 】

10

20

30

40

50

< 実施形態 3 >

CPU数の時間配分による割り当て機能により、CPU割り当て数に変更された場合も、ライセンスカウント数の状況が変更される可能性がある。

【0044】

図10は、仮想サーバに割り当てているCPU割り当て数に変更されたときを示すフローチャートである。まず、CPU割り当て数に変更されると(S1501)、仮想環境ライセンス管理モジュールは、自己が管理しているアプリケーションのライセンス利用状況を再確認する(S1502)。S1503で、仮想環境ライセンス管理モジュールは、自己の課金タイプに応じたライセンスカウントを行う。以下、ライセンス管理モジュールの課金タイプに応じたライセンスカウントを数える方法について詳細に説明する。まず、課金タイプを判断する(S1503)。仮想サーバ単位なら、仮想サーバの数え方を判断する(S1504)。ライセンス数がアプリケーションをインストールした仮想サーバに割り当てられたCPU数分なら、仮想サーバと割当CPU数からライセンスをカウントする(S1506)。ライセンスが仮想サーバごとに割り当てられているなら、仮想サーバの数からカウントする(S1507)。ライセンス数が仮想サーバ数と実CPU数とを乗じた数ならば、仮想サーバと割当CPU数とからカウントする(S1508)。また、課金タイプが物理サーバ単位なら、実CPU数からライセンス数をカウントする(S1505)。その結果をライセンスサーバに送信し、利用の可否を問い合わせる(S1509)。ライセンスサーバは、ライセンス数を受信し、現在利用されているライセンス数を更新する(S1510)。同時利用数に上限がある場合、カウント数の累積が同時利用可能数の上限を超えているかどうかを判断する(S1511)。一方、カウント数の累積が同時利用可能数の上限を超えている場合、仮想環境ライセンス管理モジュールにその結果を返却され、仮想サーバのCPU数の割り当てが変更される(S1512)。そして、変更の結果、ライセンスが足りなくなった旨の警告を表示する(S1513)。上限を超えていない場合、仮想環境ライセンス管理モジュールにその結果を返却する(S1514)。

10

20

【0045】

以上、説明したように本実施形態によれば、仮想サーバのCPU割り当て機能を施したときでもライセンス管理を行うことが可能である。

【0046】

< 実施形態 4 >

図11は、実サーバと仮想サーバが混在したネットワークシステムにおけるライセンス管理システムを示す全体構成図である。本実施形態のライセンス管理システムは、実サーバで運用されているサーバ1601と、仮想サーバ1604をそれぞれ二つたてた運用を行っているサーバ1602及び1603とを備えている。実サーバ1601、仮想サーバ1604には、アプリケーション1605がインストールされているものとする。上記の実施形態1とは異なり、実サーバ及び各仮想サーバ両方に各クライアントサーバ上のライセンスを把握するライセンス管理モジュール1606を配置する。

30

【0047】

図12は、仮想サーバ及び実サーバ上でライセンス管理モジュールがもつ情報を示す図である。ライセンス管理モジュール1606は、仮想サーバ上では、図12(a)に示すような情報1701、1702をもつ。情報1701は予めライセンス管理モジュールに設定されている情報であり、その仮想環境上で管理しているアプリケーションのライセンス課金タイプ、ライセンス管理モジュールが管理しているアプリケーション情報などである。サーバ1602、1603のような異なるサーバでは、この課金タイプが異なってもよい。また、情報1701は仮想化ソフトを通じ、実CPU数、仮想サーバ数、各仮想サーバに割り当てられているCPU数、バックアップサーバの有無などの情報である。これらの情報からライセンスカウントを計算することが可能である(1703)。ライセンス管理モジュール1606は、実サーバ1601上では、図12(b)に示すような情報1801をもつ。情報1801は予め仮想環境ライセンス管理モジュールに設定されている情報であり、その仮想環境上で管理しているアプリケーションのライセンス課金タイプ、

40

50

ライセンス管理モジュールが管理しているアプリケーション情報などである。このときの課金タイプとしては、CPU数又はCPUコア数に依存しているか、又はCPU数には依存していないなどが考えられる。ライセンス管理モジュールは予めCPU数又はコア数を把握しているので、課金タイプに応じてライセンス数をカウントする(1802)。このライセンスカウント情報をシステム全体のライセンスをカウントするライセンスサーバに送信し、利用の可否を問い合わせる(1607)。同様に、サーバ1602、1603のような仮想サーバをもつサーバ上でのライセンスをカウントし、このライセンスカウント情報をシステム全体のライセンスをカウントするライセンスサーバに送信し、利用の可否を問い合わせる(1608)。

【0048】

また、本発明の各工程は、ネットワーク又は各種記録媒体を介して取得したソフトウェア(コンピュータプログラム)をパーソナルコンピュータ等の処理装置(CPU、プロセッサ)にて実行することで実現することでもできる。

【産業上の利用可能性】

【0049】

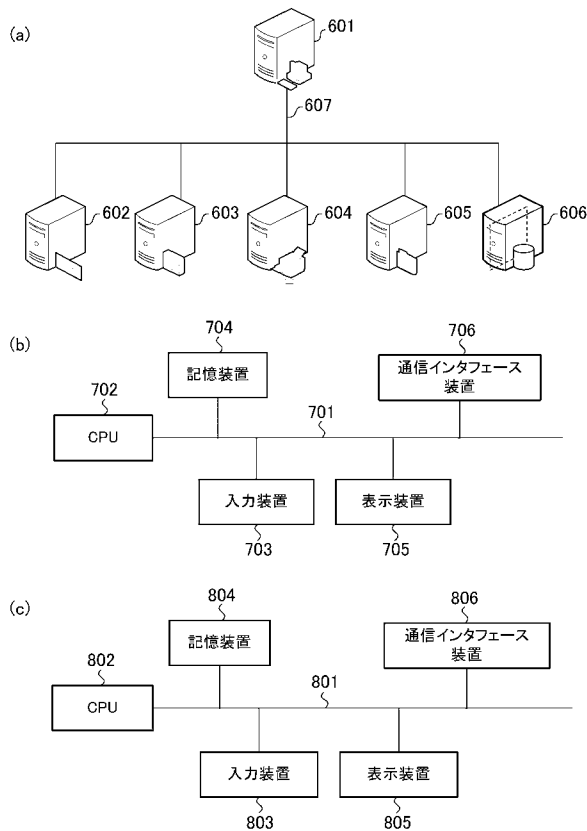
本発明は、ライセンス管理システムをもつネットワークなどに利用可能である。

【符号の説明】

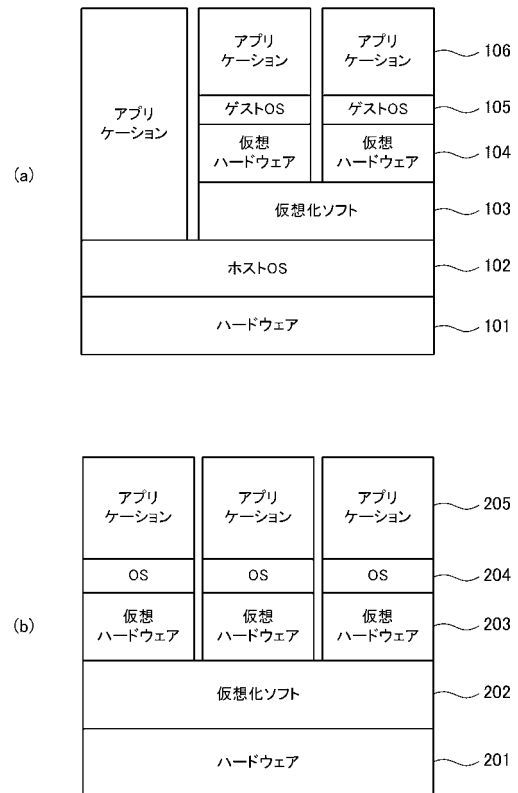
【0050】

- 601 ライセンスサーバ
- 602 端末装置
- 603 端末装置
- 604 端末装置
- 605 端末装置
- 606 端末装置
- 607 ネットワーク

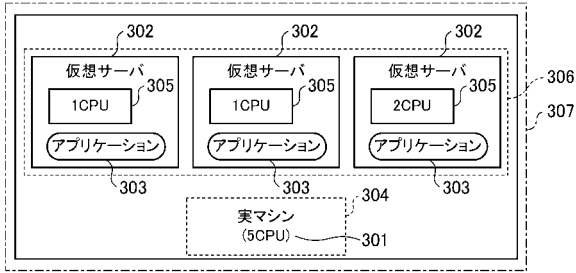
【図1】



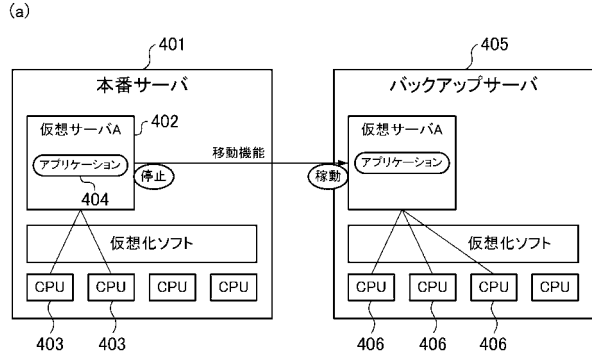
【図2】



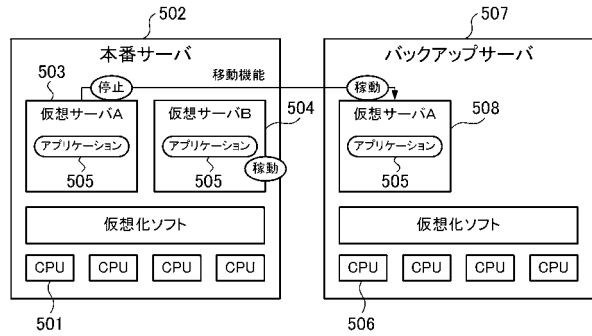
【 図 3 】



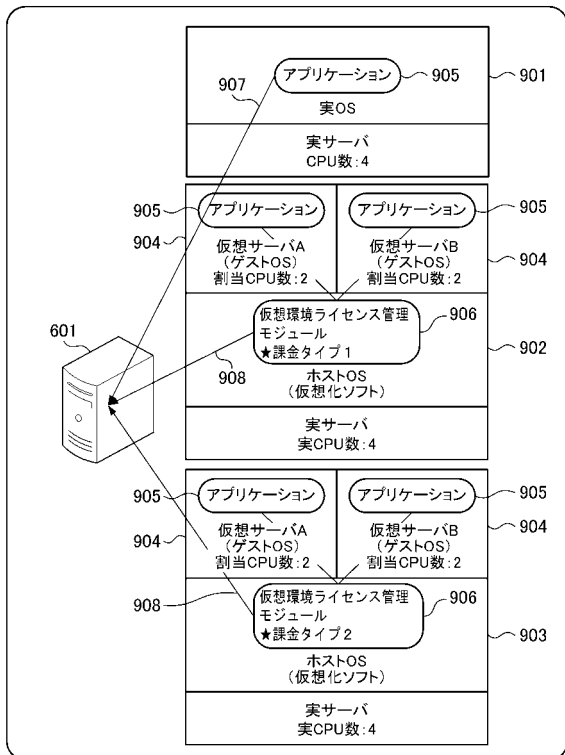
【 図 4 】



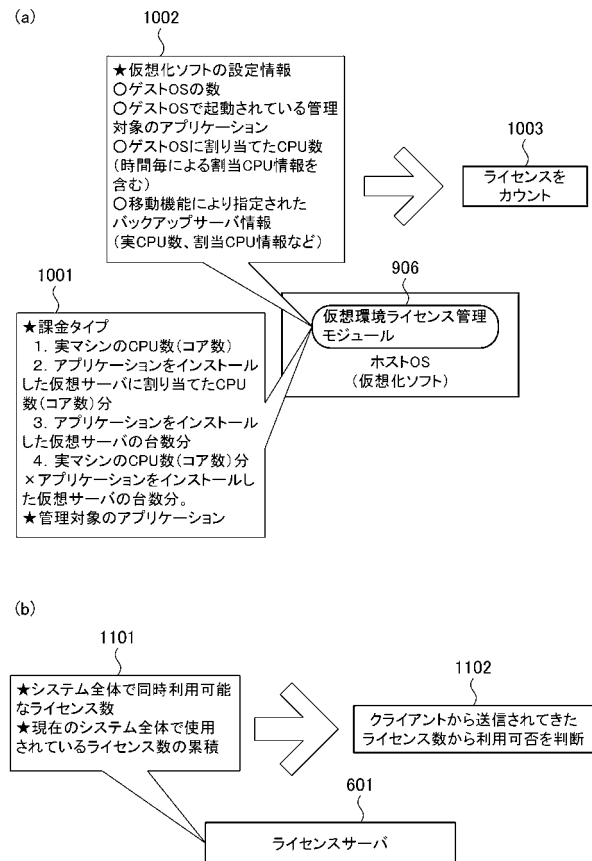
(b)



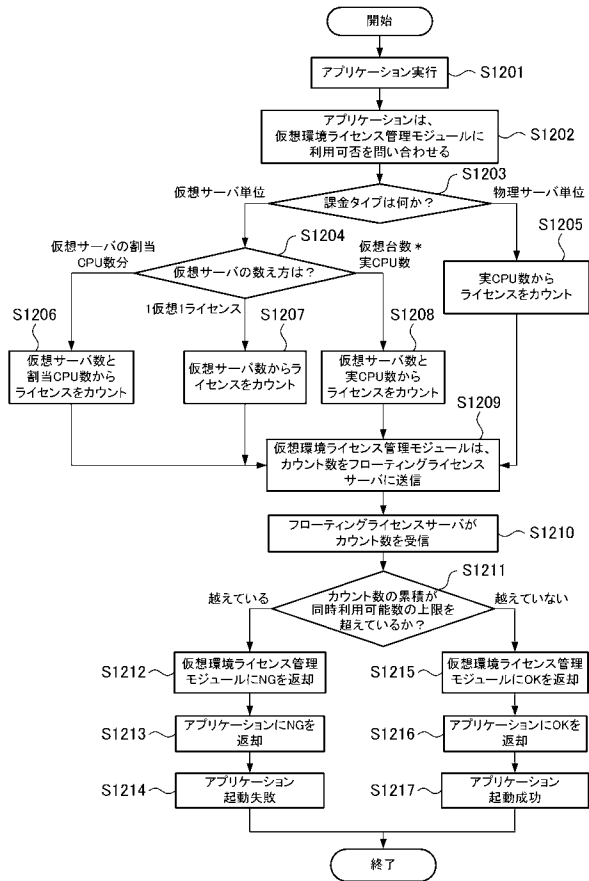
【 図 5 】



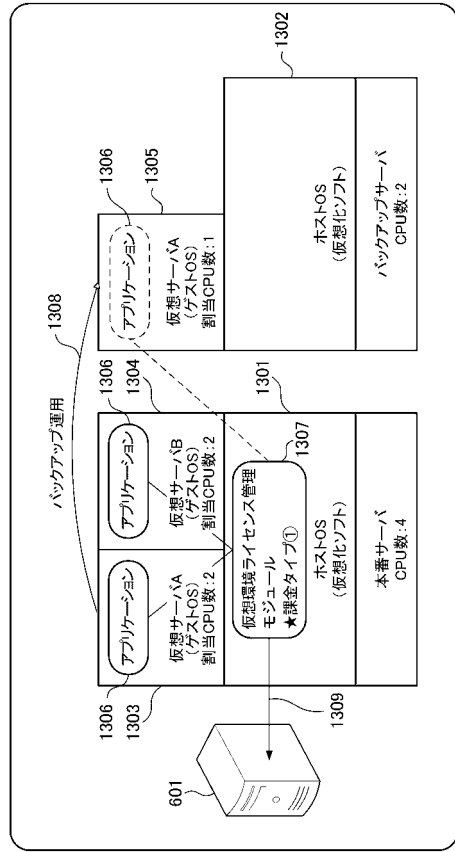
【 図 6 】



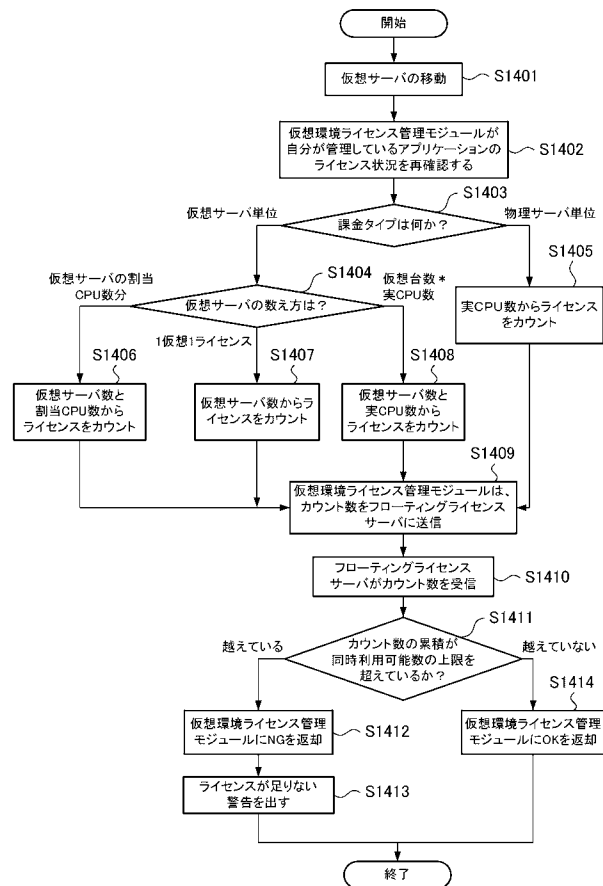
【図7】



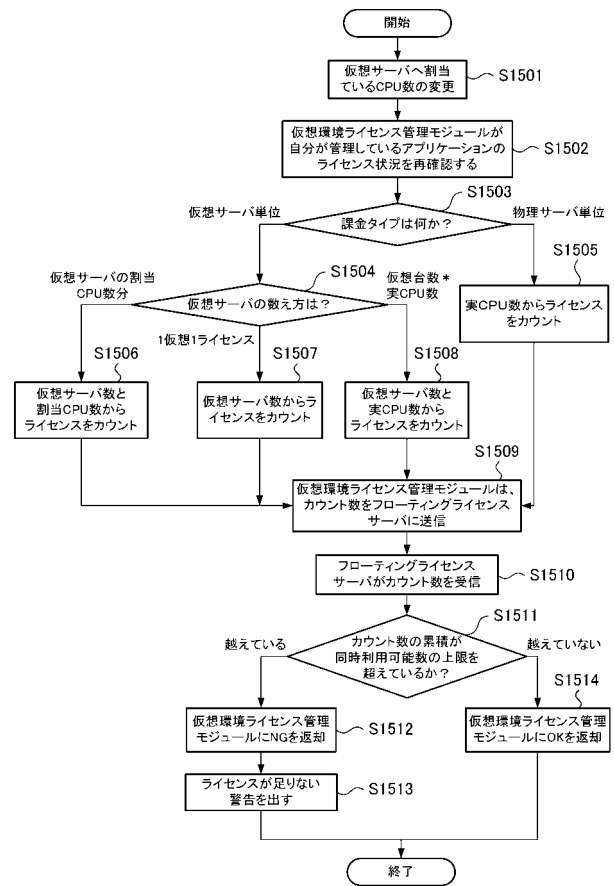
【図8】



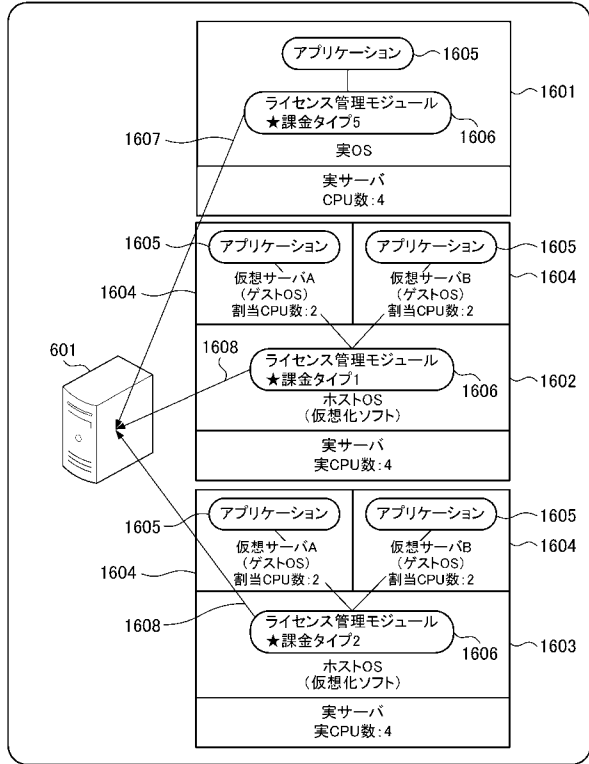
【図9】



【図10】



【図 1 1】



【図 1 2】

