(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5868466号
(P5868466)
(45)発行日　平成28年2月24日(2016.2.24)　(24)登録日　平成28年1月15日(2016.1.15)

(51)Int.Cl.　F I
G06F 11/20 (2006.01)　G06F 11/20 310A

請求項の数 12　外国語出願　(全 37 頁)

(21)出願番号　特願2014-168666(P2014-168666)
(22)出願日　平成26年8月21日(2014.8.21)
(62)分割の表示　特願2012-523039(P2012-523039)の分割
原出願日　平成22年7月29日(2010.7.29)
(65)公開番号　特開2014-238885(P2014-238885A)
(43)公開日　平成26年12月18日(2014.12.18)
審査請求日　平成26年8月22日(2014.8.22)
(31)優先権主張番号　12/831,119
(32)優先日　平成22年7月6日(2010.7.6)
(33)優先権主張国　米国(US)
(31)優先権主張番号　61/230,226
(32)優先日　平成21年7月31日(2009.7.31)
(33)優先権主張国　米国(US)

(73)特許権者　392026693
株式会社ＮＴＴドコモ
東京都千代田区永田町二丁目１１番１号
(74)代理人　100088155
弁理士　長谷川　芳樹
(74)代理人　100113435
弁理士　黒木　義樹
(74)代理人　100121980
弁理士　沖山　隆
(74)代理人　100128107
弁理士　深石　賢治

最終頁に続く

(54)【発明の名称】信頼性保証のある仮想化インフラストラクチャのためのリソース割振りプロトコル

(57)【特許請求の範囲】
【請求項１】
物理ノードと物理リンクからなる物理リソースを、プライマリ仮想ノード及び冗長仮想ノードを有する仮想インフラストラクチャに割り振るための方法であって、
１組の物理ノード、前記物理ノードを接続している物理リンク、及び前記仮想インフラストラクチャに対して要求された信頼性要件についての第１の要求を受信するステップと、
前記要求された信頼性を提供するための冗長仮想ノードの数を計算するステップであって、ｎがプライマリ仮想ノードの数、ｋが冗長仮想ノードの数である場合、ｎ個のプライマリ仮想ノードについてｋ個の冗長仮想ノードを供給して、ｎ：ｋの複製を達成するステップと、
１つの仮想インフラストラクチャのｎ＋ｋ個の仮想ノードをそれぞれ異なる物理ノードに割り振るステップと
を含む物理リソース割振り方法。
【請求項２】
前記物理リソースのそれぞれが物理ノードを備える請求項１に記載の方法。
【請求項３】
前記物理リソースが計算容量及び帯域幅を備える請求項１または２に記載の方法。
【請求項４】
前記ｎ個のプライマリ仮想ノードのうちの任意のものについて、前記ｋ個の冗長仮想ノ

ードがバックアップされる請求項１から３のいずれか一項に記載の方法。
【請求項５】
仮想ノード間に挿入するための仮想冗長リンクを決定するステップをさらに含む請求項１から４のいずれか一項に記載の方法。
【請求項６】
既存の割り振られた冗長仮想ノードが前記第１の要求により要求された冗長仮想ノードと共有されるとき、前記第１の要求により要求された信頼性、及び前記既存の割り振られた冗長仮想ノードの信頼性のそれぞれを計算するステップと、
前記計算された信頼性及び使用可能な物理リソースに基づいて、前記第１の要求により要求された冗長仮想ノードを前記既存の割り振られた冗長仮想ノードと結合するかどうかを決定するステップと、
前記第１の要求により要求された冗長仮想ノードと前記既存の割り振られた冗長仮想ノードとを結合するかどうかを決定する結果に基づいて、前記第１の要求により要求された冗長仮想ノードと前記既存の割り振られた冗長仮想ノードとを結合するステップと
をさらに含む請求項１から５のいずれか一項に記載の方法。
【請求項７】
前記物理リソースがサーバリソース及びこれらのサーバリソースを接続する物理リンクを含む請求項１から６のいずれか一項に記載の方法。
【請求項８】
前記信頼性要件が少なくとも１つの用途に関連する請求項１から７のいずれか一項に記載の方法。
【請求項９】
仮想ノード間の帯域幅予約をフローとしてモデル化するステップをさらに含む請求項１から８のいずれか一項に記載の方法。
【請求項１０】
物理ノードと冗長仮想ノード又は冗長物理ノードとの間の双方向のマッピングを二値変数によってモデル化するステップをさらに含む請求項１から９のいずれか一項に記載の方法。
【請求項１１】
各仮想インフラストラクチャの１つの仮想ノードが単一の物理ノードにのみマップされ、１つの仮想インフラストラクチャのｎ＋ｋ個の仮想ノードについて、１つの物理ノードに１つ以下の仮想ノードしかマップされない請求項１から１０のいずれか一項に記載の方法。
【請求項１２】
コンピュータによって実行されると、前記コンピュータが物理ノードと物理リンクからなる物理リソースをプライマリ仮想ノード及び冗長仮想ノードを有する仮想インフラストラクチャに割り振る方法を実行するプログラムを格納するコンピュータ読み取り可能な記録媒体であって、前記方法が、
１組の物理ノード、前記物理ノードを接続している物理リンク、及び前記仮想インフラストラクチャに対して要求された信頼性要件についての第１の要求を受信するステップと、
前記要求された信頼性を提供するための冗長仮想ノードの数を計算するステップであって、ｎがプライマリ仮想ノードの数、ｋが冗長仮想ノードの数である場合、ｎ個のプライマリ仮想ノードについてｋ個の冗長仮想ノードを供給して、ｎ：ｋの複製を達成するステップと、
１つの仮想インフラストラクチャのｎ＋ｋ個の仮想ノードをそれぞれ異なる物理ノードに割り振るステップと
を含むコンピュータ読み取り可能な記録媒体。
【発明の詳細な説明】
【優先権】

【０００１】
[0001]本特許出願は、２００９年７月３１日に出願された「信頼性保証のある仮想化インフラストラクチャのためのリソース割振りプロトコル（Ａ Ｒｅｓｏｕｒｃｅ Ａｌｌｏｃａｔｉｏｎ Ｐｒｏｔｏｃｏｌ ｆｏｒ Ｖｉｒｔｕａｌｉｚｅｄ Ｉｎｆｒａｓｔｒｕｃｔｕｒｅ ｗｉｔｈ Ｒｅｌｉａｂｉｌｉｔｙ Ｇｕａｒａｎｔｅｅｓ）」という名称の対応する特許仮出願第６１／２３０，２２６号の優先権を主張し、参照によって組み込む。
【発明の分野】
【０００２】
[0002]本発明は、信頼性、仮想化インフラストラクチャ、及びリソース割振りの分野に関し、より詳細には、本発明は、信頼性保証のある仮想インフラストラクチャにおけるリソースの割振りに関する。
【発明の背景】
【０００３】
[0003]通信ネットワークは、物理から仮想へとシフトしつつある。従来、通信ネットワークは、所与のネットワークをサポートするために物理的なインフラストラクチャを使用して構築されている。インフラストラクチャは、ますます仮想になっている。すなわち、専用の物理ネットワークを構築する代わりに、又は、特定の意図によって設計されていないネットワークを他者と共有する代わりに、共有の物理的な基体の上に構築された、特定のカスタマイズされたプロトコルを備える専用ネットワークの外観をそのユーザに提供するカスタマイズされたネットワークの仮想ネットワークが使用されている。仮想ネットワークは、その（仮想の）オペレータにとってのプライベートネットワークであり、アンダーレイは、様々なオペレータの間で共有される。
【０００４】
[0004]仮想化は、物理リソースが現在使用されている方法を急速に変えている。元々、サーバを切り離し、物理サーバにわたってリソースを共有するために設計されており、仮想化は、サーバをソフトウェアによって完全に定義できるようにすることによって、高速で素早い配置及びマイグレーションを提供する。これは、演算処理を弾性のあるリソースに変え、他の商業的なエンティティの間でも急速に人気が出つつある。仮想化のパラダイムは、ネットワーキングに及んでいる。例えば、仮想化のパラダイムによって、複数の研究グループは、惑星規模のネットワークの異なる仮想スライスにわたって複数のオーバーレイの叩き台を稼働させることができる。事実、仮想化インフラストラクチャがサービスとして提供される場合、仮想化によってもたらされる機敏さ及び柔軟性によって、インフラストラクチャプロバイダとサービスプロバイダとの間の効率の良い分離によって、次世代のインターネットを融通が利くものにすることができることを研究者は確信している。
【０００５】
[0005]こうした仮想化されたアーキテクチャの１つの鍵となる側面は、アンダーレイのリソースを上部の仮想ネットワークに適切に割り当てることである。使用するリソースは、仮想化されるので、物理的なアンダーレイにおける異なるスポットに配置することができ、ネットワークの最高のパフォーマンスのために、物理に対する仮想リソースの慎重な割振りが重要である。適切になされると、各仮想ネットワークはよりよく機能し、物理的なアンダーレイの使用率が増加する（及びしたがって、経費を低減する）。
【０００６】
[0006]インフラストラクチャが急速に仮想化され、共有され、動的に変化している状態で、強い信頼性を物理的なインフラストラクチャに提供することが重要である。その理由は、単一の物理サーバ又はリンクの故障がいくつかの共有される仮想化されたエンティティに影響を及ぼすからである。信頼性は、冗長性を使用することによって提供される。現在では、信頼性は、リソースを複製することによって提供される。この理由は、信頼性が物理層に設けられているからである。したがって、物理コンポーネントの故障は、別の物理的な要素を持ち出すことによって処理される。仮想化インフラストラクチャにおいて、

物理コンポーネントや物理的な要素はバックアップすることを必要とする仮想要素であり、物理コンポーネントの故障は、何らかの仮想コンポーネントの消失を意味し、これらの仮想コンポーネントを他の物理コンポーネント上に配置し直さなければならない。

【０００７】

[0007]信頼性を提供することは、多くの場合、計算、ネットワーク、及びストレージの能力を過度に供給し、付加的な堅牢性のために負荷バランシングを使用することにつながる。こうした高可用性システムは、例えば、リンク又はノードの故障をルート変更する間のネットワークフローの再起動、又は、部分的なジョブがノード故障時に再開するなど、大きい不連続性が許容され得る用途に適している。高レベルのフォールトトレランスは、何らかの故障がシステムの現在の状態にかなりの影響を及ぼす用途で必要とされる。例えば、アドミッションコントロール、スケジューリング、負荷バランシング、帯域幅ブローキング（ｂａｎｄｗｉｄｔｈ　ｂｒｏｋｉｎｇ）、ＡＡＡ、又はネットワーク状態のスナップショットを維持する他のＮＯＣ動作を実行するサーバを有する仮想ネットワークは、全体的な故障を許容することができない。例えばＭａｐＲｅｄｕｃｅ、ＰＶＭなど、マスター－スレーブ／ワーカーアーキテクチャ（ｍａｓｔｅｒ－ｓｌａｖｅ／ｗｏｒｋｅｒ　ａｒｃｈｉｔｅｃｔｕｒｅ）では、マスターノードでの故障は、スレーブ／ワーカーにおいてリソースを浪費する。

【０００８】

[0008]ネットワークの仮想化は、ネットワークの運転費及び管理の複雑さを低減する有望な技術であり、研究対象の増加を受けている。インフラストラクチャのプロバイダがより単純な、より安価なコモディティハードウェア（ｃｏｍｍｏｄｉｔｙ　ｈａｒｄｗａｒｅ）上に各自のネットワークを仮想化する方向に進むにつれて、信頼性はますます重要な問題にならざるを得ない。

【０００９】

[0009]ネットワークの信頼性を研究するために、「ｓｈａｄｏｗ　ＶＮｅｔ」、すなわち、パラレルの仮想化スライスの使用を考慮に入れているものもある。しかし、こうしたスライスは、バックアップとして使用されるのではなく、監視ツールとして使用され、故障時にはネットワークをデバッグする方法として使用される。

【００１０】

[0010]一方、サーバ仮想化レベルでのノードのフォールトトレランスに目標が定められる研究がいくつかある。少なくとも１つは、ハイパーバイザにおけるフォールトトレランスを導入した。同じ物理ノードにある２枚の仮想スライスがハイパーバイザを介して同期して動作するようにできる。しかし、これは、せいぜいソフトウェアの故障に対する信頼性を提供するだけである。その理由は、スライスが同じノードにあるからである。

【００１１】

[0011]仮想スライスがネットワークにわたって複製され、遷移されるようにするために発達したものもある。異なるタイプの用途（ウェブサーバ、ゲームサーバ、及びベンチマークの用途）のために、様々な複製技術及びマイグレーションプロトコルが提案された。別のシステムは、ある期間にわたる２つの仮想ノードの間の状態の同期を可能にする。したがって、信頼性のためにネットワークにわたって分散される冗長な仮想ノードを有することは実際に可能である。しかし、これらのソリューションは、ネットワークのどこかに冗長ノードが存在するのに、（計算能力における）リソース割振り問題に対処していない。

【００１２】

[0012]基本的なレベルで、ノード及びリンクの信頼性に対処する冗長ノードのトポロジを構成するための方法がある。一部の入力グラフに基づいて、追加のリンク（又は帯域幅予約）は、最小数だけで良いように、最適に導かれる。しかし、これは、大部分はステートレスであるマルチプロセッサシステムのフォールトトレランスを設計することに基づく。この場合、ノードの故障は、最初のトポロジを保護するために、残りのノードの中のマイグレーション又は回転を伴う。これは、マイグレーションが故障によって影響を受けな

いネットワークの一部分に中断を引き起こす場合がある仮想化ネットワークのシナリオにおいて適切でない場合がある。

【００１３】

[0013]フォールトトレランスは、データセンターにおいても設けられる。冗長性は、過剰な大量のノード及びリンクに関する。故障の回復のために、いくつかのプロトコルが定義されるが、信頼性保証はほとんどない。

【発明の概要】

【００１４】

[0014]本明細書において、リソース割振りプロトコルのための方法及び装置が開示される。一実施形態において、この装置は、物理リソースをプライマリ及び冗長仮想インフラストラクチャに割り振るためのリソース割振りエンジンを備え、リソース割振りエンジンが仮想インフラストラクチャを割り振るとき、冗長な仮想インフラストラクチャの物理リソースが複数のプライマリ仮想インフラストラクチャにわたって共有される。

【００１５】

[0015]本発明は、以下の詳細な説明から、及び本発明の様々な実施形態の添付の図面から、より完全に理解されるが、こうした図面及び実施形態は、本発明を特定の実施形態に制限するものと見なされないものとし、説明及び理解のためのものである。

【図面の簡単な説明】

【００１６】

【図１】フォールトトレラントアーキテクチャの比較を示す図である。

【図２】９９．９９９％のノードの信頼性のために必要とされる冗長ノードの数を示す図である。

【図３】ｎ：ｋの複製がサポートできるノードの数を示す図である。

【図４】１つのバックアップノード及びそれぞれ割り振られたフェールオーバ帯域幅を備えるＶＩの一例を示す図である。

【図５】冗長性をプールし、仮想ノードを拡散させる一例を、仮想化データセンターにある４つのＶＩを示すことによって示す図である。

【図６】２つのＶＩのバックアップノードをプールするときのこのトレードオフを示す図である。

【図７】管理アーキテクチャの一実施形態を示す図である。

【図８】入ってくる各要求にサービスを提供するプロセスの一実施形態のフロー図である。

【図９】重複する帯域幅予約の一例を示す図である。

【図１０】冗長ノードの位置が固定された拡張グラフの一例を示す図である。

【図１１】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】

【００１７】

[0016]以下、ｎ：ｋの冗長アーキテクチャが開示され、この場合、ｋ個の冗長リソースは、ｎ個のプライマリリソースのうちの任意のものについてのバックアップであり、複数の仮想インフラストラクチャ（ＶＩ）にわたってバックアップを共有することができる。例えば、n_1及びn_2のコンピューティングノードを有する２つのＶＩは、k_1及びk_2の冗長性がそれぞれr_1及びr_2の保証された信頼性となることを要求する。バックアップを共有することは、同じレベルの信頼性を有する$k_0 < k_1 + k_2$の冗長性を達成し、フォールトトレランスのために供給されるリソースを低減する。さらに、冗長ノードが、保証された接続性、帯域幅で、ほとんど中断なく、故障したノードを引き継ぐことができるように、ジョイントノード及びリンク冗長性がある。リンクの故障は、同じ機構を介して回復することができる。

【００１８】

[0017]物理リソース（例えば、計算容量、ストレージ、及び帯域幅）を同時にプライマリ及び冗長ＶＩに静的に割り振る方法も本明細書に開示される。この方法は、既存の冗長

ノードを使用して、冗長仮想リンクの帯域幅をできるだけ多く重複させることによって冗長性に割り振られるリソースを低減することを試みる。

【００１９】

[0018]さらに、物理リソースの使用を最低限に抑える、又は大幅に低減し、対応することができる仮想リソースの数を最大にすることを試みる方法で信頼性保証を提供する、物理的な基体上に仮想インフラストラクチャのリソースを割り振る機構が開示される。

【００２０】

[0019]信頼性を物理リソースの割振りに組み込み、冗長ノードをいくつかの仮想ネットワーク中で共有することは、信頼性のためのリソースの量を大幅に低減する。

【００２１】

[0020]以下の説明では、本発明のより完全な説明を提供するために、多数の詳細が記載される。しかし、これらの具体的な詳細なしに本発明を実践できることは、当業者であれば明らかである。他の例において、周知の構造及びデバイスは、本発明を不明瞭にすることを回避するために、詳細にではなく、ブロック図の形式で示される。

【００２２】

[0021]以下の詳細な説明の一部は、コンピュータメモリ内のデータビット上の動作のアルゴリズム及び記号表現の形で示される。これらのアルゴリズムの説明及び表現は、他の当業者に最も効果的にその作業の要旨を伝えるために、データ処理技術に熟練した人々によって使用される手段である。アルゴリズムは、ここでは、また一般的に、所望の結果をもたらす自己矛盾のない一連のステップであると考えられる。こうしたステップは、物理量の物理的操作を必要とするものである。必須ではないが、通常、これらの量は、格納され、移動され、結合され、比較され、そうでなければ操作され得る電気又は磁気の信号の形をとる。主に一般的な使用の理由で、これらの信号をビット、値、要素、シンボル、文字、項、数字等と呼ぶことが時として便利であることがわかっている。

【００２３】

[0022]しかし、これらの及び類似の用語はすべて、適切な物理量に関連付けられたものであり、こうした物理量に適用される便宜上のラベルに過ぎないことを理解されたい。特に明記しない限り、以下の説明からわかるように、説明の全体にわたって、「処理する」、「計算する」、「算出する」、「決定する」、又は「表示する」等の用語を使用している議論は、コンピュータシステムのレジスタ及びメモリ内の物理（電子）量として表されるデータを操作し、コンピュータシステムメモリ若しくはレジスタ、又は他のこうした情報ストレージ、送信又は表示デバイス内で物理量として同じように表される他のデータに変換する、コンピュータシステム又は類似の電子コンピューティング装置の動作及びプロセスを指すことを理解されたい。

【００２４】

[0023]本発明は、本明細書において動作を実行するための装置にも関する。この装置は、要求される目的のために特別に構成することができ、又は、コンピュータに格納されたコンピュータプログラムによって選択的に稼働される、又は再構成される汎用コンピュータを備えることができる。こうしたコンピュータプログラムは、例えば、それだけには限定されないが、コンピュータシステムバスにそれぞれ結合される、フロッピーディスク、光ディスク、ＣＤ－ＲＯＭ、及び光磁気ディスクを含む任意のタイプのディスク、読み取り専用メモリ（ＲＯＭ）、ランダムアクセスメモリ（ＲＡＭ）、ＥＰＲＯＭ、ＥＥＰＲＯＭ、磁気又は光学カード、又は電子命令の格納に適した任意のタイプの媒体など、コンピュータ可読記憶媒体に格納することができる。

【００２５】

[0024]本明細書において提示されるアルゴリズム及び表示は、任意の特定のコンピュータ又は他の装置に本質的に関連がない。様々な汎用システムは、本明細書における教示によるプログラムによって使用することができ、又は、必要な方法ステップを実行するためにより専門の装置を構成することは便利であることがわかり得る。様々なこれらのシステムに必要な構造が下記の説明から明らかになる。さらに、本発明は、任意の特定のプログ

ラミング言語を参照して記載されていない。本明細書において記載されている本発明の教示を実施するために、様々なプログラミング言語を使用することができることを理解されたい。

【００２６】

[0025]機械可読媒体は、マシン（例えば、コンピュータ）によって可読の形式で情報を格納又は送信するための任意の機構を含む。例えば、機械可読媒体は、読み取り専用メモリ（「ＲＯＭ」）、ランダムアクセスメモリ（「ＲＡＭ」）、磁気ディスク記憶媒体、光記憶媒体、フラッシュメモリ記憶デバイスなどを含む。

概要

【００２７】

[0026]一実施形態において、リソースは、仮想インフラストラクチャ要求の信頼性要件を考慮する割振り方法を使用して割り振られる。物理リソースを仮想リソース要求に割り振るための方法が存在するのに対して、本明細書に記載される割振り方法は、まず、明確な信頼性保証を提供する。

【００２８】

[0027]一実施形態において、割振り機構は、１組のリソース（例えば、サーバリソース）又はその一部、これらのリソースを接続しているリンクについての要求、及び信頼性要件、例えば９９．９９９％の動作可能時間を受信する。一実施形態において、要求は、（Ｇ＝（Ｖ，Ｅ），ｒ）として表され、この場合、Ｖはノード、Ｅはノードを接続しているリンク、ｒは信頼性である。次いで割振り機構は、要求された信頼性を提供するために要求に追加する冗長ノードの数を計算する。この要求を別のものと結合することができ、そうすることからの利点がある場合、割振り機構は２つの要求を結合する。一実施形態において、割振り機構は、割振り要求を結合することが有益かどうかを決定する。したがって、一実施形態において、割振り機構は、要求を集約し、冗長性のために取っておかれる物理リソースの量を低減し、場合によっては最低限に抑えることを試みる方法で物理的な冗長リソースを割り振る。

【００２９】

[0028]一実施形態において、要求に追加する冗長ノードの量、及びエッジの間に挿入するリンクを決定した後、割振り機構は、新しい要求Ｇ’＝（Ｖ’，Ｅ’）を計算し、従来のマルチフローコモディティ問題（ｔｒａｄｉｔｉｏｎａｌ　ｍｕｌｔｉ－ｆｌｏｗ　ｃｏｍｍｏｄｉｔｙ　ｐｒｏｂｌｅｍ）を使用して、この要求を割り振る。マルチフローコモディティ問題は、公知技術である。

［ネットワークモデル］

【００３０】

[0029]本明細書のために、コンピューティング及びネットワークのリソースを仮想化し、分離し、いくつかのエンティティにわたって共有することができる物理的なネットワークインフラストラクチャが使用される。物理的なインフラストラクチャからのリソースについての要求が、計算ノードの容量及び好ましい場所、ノード間の帯域幅、及び要求されたノード（及びそれらのリンク）のサブセットにおけるあるレベルの信頼性に関して定義される。入ってくる各リソース要求は、冗長なインフラストラクチャと共に静的に割り振られる。

【００３１】

[0030]一実施形態において、物理ネットワークは、無向グラフ$G^P=(N^P, E^P)$としてモデル化され、この場合、N^Pは物理ノードの組であり、E^Pはリンクの組である。各ノード$u \in N^P$は、M_uの使用可能な計算容量を有する。各無向リンク$(u, v) \in E^P$、$u, v \in N^P$は、H_{uv}の使用可能な帯域幅容量を有する。マルチフローコモディティ問題を単純化するために、物理ノードにおける故障は、確率ｐで、独立しており一様であると見なされる。

【００３２】

[0031]リソース要求は、追加の特性を有する無向グラフ$G^V=(N^V, E^V)$としてモ

デル化される。N^Vは、１組の計算ノードであり、E^Vは１組のエッジである。μ_xは、ノードごとの計算能力要件、$x \in N^V$であり、ノードの間の帯域幅要件はη_{xy}、（ｘ，ｙ）$\in E^V$及びｘ，ｙ$\in N^V$である。さらに、Φ［ｘ］$\subseteq N^P$は、仮想ノードｘをマップすることができる追加の制約である。すなわち、仮想ノードを物理ノード上に何らかの特定のマッピングを課すために、ｘを物理ノードのサブセットにしかマップできないように、制約Φ［ｘ］として指定される。このことは物理的位置の選好（本文で述べるように）又は物理ノードタイプ（ＣＰＵノード、ストレージノード、ルータノード）に起因し得ることに留意されたい。これは、任意の物理的位置の選好、例えば、入口及び出口の仮想ルータ、他のノードとの近さなどを表す。後述するように、この組は、すでに設けられている別のＶＩから冗長ノードを再利用する／共有するためにも利用される。各要求も１組のクリティカル仮想ノード$C^V \subseteq N^V$及びその関連のリンク｛（ｃ，ｘ）｜（ｃ，ｘ）$\in E^V$，ｃ$\in C^V$，ｘ$\in N^V$｝から成り、信頼性ｒで保護される。本明細書のために、冗長ノードの組は、N^Kと示される。

【００３３】

[0032]整合性のために、ｉ，ｊを使用して任意のタイプのノードを表し、ｘ，ｙ，ｚ$\in N^V$を使用して仮想ノードを表し、ｕ，ｖ，ｗ$\in N^P$を使用して物理ノードを表し、ｃ，ｄ$\in C^V$を使用してクリティカルノードを表し、ａ，ｂ$\in N^K$を使用して冗長ノードを表す。

［冗長性のための仮想アーキテクチャ］

【００３４】

[0033]一実施形態において、冗長性のためのアーキテクチャは、以下の特性を有する。

【００３５】

ｎ：ｋの冗長性。最高で

【数１】

$$\frac{1}{k+1},$$

の使用率をもたらす１：ｋレベルの冗長性を有することに対して、ｎ個のプライマリリソースの任意のものについて、ｋ個の冗長ノードをバックアップすることができるとき、より良い粒度及び使用率を達成することができる。

【００３６】

ジョイントノード及びリンク冗長性。ノードが故障すると、冗長ノードが保証された接続性、帯域幅で、ほとんど中断なく引き継ぐように、冗長ノード及びリンクが一緒に供給される。

【００３７】

交わらない位置。同じ物理ノードにおいてホストすることができる仮想又は冗長ノードは１つ以下である。

【００３８】

[0034]このように、｜N^V｜＋ｋ個から、物理ノードの故障の数がｋ個を上回る確率が１－ｒを下回らないように、ｋ個の冗長仮想ノードが供給される。言い換えれば、信頼性は、

【数２】

$$r \le \sum_{i=0}^{k} \binom{n+k}{i} p^i \overline{p}^{n+k-i} \quad (1)$$
$$= I_{\overline{p}}(n, k+1)$$

として得られ、この場合、

【数３】

$$n = \left|N^V\right|, \overline{p} = 1 - p.$$

である。右辺の和は、公知技術である、規則化された不完全なベータ関数Ｉ$_{x}$（・，・）に相当する。

【００３９】

[0035]一実施形態において、故障時にｋ個の冗長ノードすべてに対して、十分な量の計算能力及び帯域幅が使用可能である。したがって、リンク及びノードの故障について、回復手順は、ｋ個の冗長ノードのうちの１つ又は複数を持ち出し、確保された冗長リソースを使用するように動作する。一実施形態において、これによってさらに中断が生じる場合があるため、仮想ノードのマイグレーション又はスワッピングは回復を援助することができない。さらに、冗長ノードも故障する場合があるため、冗長ノードは、式（１）において述べられる信頼性を達成するために、任意のノードｃ∈Ｃ^{V}の代わりとなることができなければいけない。上述したように、帯域幅予約においてパス分割が使用され、リンクについての別の層の保護及びグレースフルデグラデーションが提供される。

【００４０】

[0036]冗長性のための帯域幅予約は、Ｎ^{K}のノードから発せられる１組の重み付けされた無向仮想リンクＬとしてモデル化される。

【００４１】

Ｌ⊆Ｎ^{K}×（Ｎ^{V}∪Ｎ^{K}）＝（Ｎ^{K}×Ｎ^{V}）∪（Ｎ^{K}×Ｎ^{K}）（２）

すなわち、Ｌは、Ｎ^{V}の頂点を有する、それ自体の間の冗長ノードからのリンクを含む、２つの２部グラフの結合である。これらのリンクは、仮想ネットワークの埋め込みのためにＧ^{V}に追加される。より形式的に、Ｌは、下の２つの定理によって定義される。

【００４２】

定理１。ａ∈Ｎ^{K}及びｘ∈Ｎ^{V}が与えられる。その場合、

【数４】

$$(a,x)\notin L \ iff \ \nexists (c,x)\in E^{V}, c\in C^{V}.$$

である。これは、クリティカルリンク（ｃ，ｘ）をＬにおけるリンク（ａ，ｘ）によってバックアップする必要があり、したがって、故障のためにｃがａに遷移された場合、ｘは、リソースの新しい位置にまだ接続されていることを示す。

【００４３】

証明：

【数５】

$$(a,x)\notin L$$

及び仮想リンク（ｃ，ｘ）∈Ｅ^{V}が存在し、ｃ∈Ｃ^{V}であると仮定する。
その時、アーキテクチャは、ｎ：ｋの冗長性を有していない。その理由は、ｃが故障した場合、ａは、ｘに供給される帯域幅を有していないからである。同様に、

【数６】

$$\nexists (c,x)\in E^{V}, c\in C^{V}$$

であり、（ａ，ｘ）∈Ｌである場合、ｃが故障した場合、（ａ，ｘ）のために供給される帯域幅は決して使用されない。

【００４４】

推論１。

【数７】

$$(a,c)\notin L \ iff \ (c,d)\notin E^{V}$$

であり、この場合、ａ∈Ｎ^{K}及びｃ，ｄ∈Ｃ^{V}である。

【００４５】

証明：これは、ｘの領域をＣ^{V}に限定することによる定理１からの直接的な結果である。

【００４６】
[0037]上記は、Ｌが２部グラフＬ^{1}から成ることを明示する。
【００４７】
Ｌ^{1}＝｛（ａ，ｘ）｜∀ａ∈Ｎ^{K}，∀ｃ∈Ｃ^{V}，∃（ｃ，ｘ）∈Ｅ^{V}，ｘ∈Ｎ^{V}｝（３）
【００４８】
定理２。ａ，ｂ＜∈Ｎ^{K}が与えられた場合、
【数８】
$$(a,b)\notin L \; iff \; \int(c,d)\in E^{V}$$
∀ｃ，ｄ∈Ｃ^{V}である。これは、クリティカルノードの間にリンクがある場合、各冗長ノード間のリンクが存在しなければならないことを示す。
【００４９】
証明：Ｌにおいてａ及びｂが接続されておらず、しかし、リンク（ｃ，ｄ）∈Ｅ^{V}が存在すると仮定する。その時、ｃ及びｄが故障し、ａ及びｂに遷移される場合、帯域幅保証はない。逆に、（ａ，ｂ）∈Ｌであり、
【数９】
$$\int(c,d)\in E^{V}$$
である場合、（ａ，ｂ）のために供給される帯域幅は決して使用されない。
【００５０】
[0038]これによって、任意の２つのクリティカルノードの間にリンクがある限り、Ｎ^{K}の冗長ノード間に完全グラフを含むＬが得られる。冗長ノード間の完全グラフをＬ^{2}によって示す。
【００５１】
Ｌ^{2}＝｛（ａ，ｂ）｜ａ≠ｂ，∀ａ，ｂ∈Ｎ^{K}｝（４）
【００５２】
[0039]Ｌ＝⊆（Ｎ^{K}×Ｎ^{V}）∪（Ｎ^{K}×Ｎ^{K}）であるため、冗長リンクの最小の組は、
【数１０】
$$L=\begin{cases} L^{1} & ,(x,y)\in E^{V},\forall x,y\in C^{V} \\ L^{1}\cup L^{2} & ,\text{それ以外の場合} \end{cases} \quad (5)$$
によって与えられる。この結果は、他の提案されたアーキテクチャより多くのリンクを必要とする。しかし、後者の結果は、故障後の回復されたグラフがＧ^{V}を含むという仮定に基づく。このことは、故障によって影響を受けないノードをＧ^{V}の最初のトポロジの回復のために遷移する必要がないことを確実にしない。この追加の制約は、Ｌを構成する際、考慮に入れられる。それにもかかわらず、この制約が必要ない場合、Ｌを他のソリューションと置き換えることができる。
【００５３】
[0040]マルチコモディティフロー（ＭＣＦ）によって帯域幅が供給されている下記の場合、冗長フローを可能な限り重複させることによって、帯域幅が低減され、又は最低限に抑えられる。これらの重複は、ＭＣＦモデルへの制約とし捕捉される。
［冗長性の共有］
【００５４】
[0041]以下、ｎ：ｋのフォールトトレラントアーキテクチャの利点、及び冗長ノードの共有が使用率をどれだけ増加させ得るかの表示が開示される。説明を簡単にするために、本明細書ではＣ^{V}＝Ｎ^{V}と仮定する。
【００５５】
[0042]図１Ａの小さい３ノードの仮想ネットワークについて考慮する。図１Ｂに示されるように、１ノードの故障を許容する簡単で端的な方法は、１：ｋの許容値を使用するこ

とであり、すなわちあらゆる仮想マシンをいったん複製し、複製及びプライマリノードへの論理リンクを作成することである。ｋ$_s$ノードの故障のフォールトトレランスは、ｋ$_s$層の複製によって達成することができる。各物理ノードの故障の確率がｐであると仮定すると、ｎ個のノード及びｅ個のリンクのネットワークについて、ノードの信頼性ｒを達成するために必要な冗長性の層の数は、

【数１１】

$$k_s = \left\lceil \frac{\log\left(1 - r^{\frac{1}{n}}\right)}{\log p - 1} \right\rceil \quad (6)$$

となる。残念なことに、簡単な複製は、システムにあまりに多くの冗長ノード及び論理リンクを追加し、それぞれｋ$_s$ｎ及びｋ$_s$ｎ＋３ｋ$_s$ｅとなる。

【００５６】

[0043]この方法と、冗長ノードが、すなわちｎ：ｋである、図１Ｃにおけるの３つのノードのいずれかについてのバックアップである他の手法とを比較する。ノード信頼性ｒは、より細かい粒度、及びより少ない数の冗長ノード及びリンクによって提供することができ、（１）によって得られる。冗長ノードの数及びリンク（最大）は、ｋ及び

【数１２】

$$kn + \frac{k}{2}(k - 1).$$

である。図２は、９９．９９９％のノードの信頼性に必要とされる冗長ノードの数について、２つの手法の間の比較を示す。

【００５７】

[0044]予想されるように、冗長ノードの数は、同じレベルの信頼性について、ｎ：ｋの複製より１：ｋの複製でかなり速く増える。実際に、図３でわかるように、ｎ：ｋの手法では、十分に拡縮される。１つの興味深い傾向は、ｋの値が小さい場合、ｎが超線形であるということである。例えば、故障の確率ｐ＝０．０１の曲線について、９５ノードのＶＩは、９９．９９９％の信頼性のためにｋ＝７を必要とし、１９０ノードのＶＩは、ｋ＝１０を必要とする。一見したところ、２つの９５ノードＶＩを割り振るときに、７つずつの冗長ノードを供給するよりも、ｋ＝１０の冗長ノードを共有することに価値がある。

【００５８】

[0045]大きいｋでは、ｎが直線的に増えることに留意されたい。共有はもはや、冗長ノードの数を減らさない。ｎ対ｋの線形の挙動は、結合も有害でないことを意味することにも留意されたい。冗長リンクの数が少なくともｎｋであると想定すれば、冗長ノードを共有するときに、より多くの帯域幅が確保される。一方で、小さいｋでは、より多くの冗長リンクについて、冗長ノードの数の減少がトレードオフされる。

【００５９】

[0046]冗長ノードを共有するのに価値がある方法が２つある。

【００６０】

１）ｋにおける離散型のジャンプ（ｄｉｓｃｒｅｔｅ　ｊｕｍｐ）の使用。例えば、１２ノードのＶＩは、９９．９９９％の信頼性のために４つの冗長ノードを使用する。同じ４つのノードは、同じレベルの信頼性のために、別の１３個のプライマリノードをサポートすることができる。

【００６１】

２）異なるレベルの信頼性を必要とするＶＩ間での非対称の共有。例えば、２１ノード

のＶＩは、９９．９９９％の信頼性のために、４つの冗長ノードを使用する。４つの冗長ノードのうちの１つを、９９．９％の信頼性のために、別の５ノードのＶＩと共有することができる。（ｍ個の他のＶＩと共有するとき）より大きいＶＩの信頼性は、下記のように計算することができる。

【数１３】

$$r'_0 = 1 - \sum_{x=0}^{k'} Pr\{\text{k'個のうちのx個のバックアップがダウンするまたはVInf-1,..., VInf-mによって使用されている}\} \times$$

$$Pr\{\text{k0-k'個のバックアップを有するVInf-0からk0-x個超のノードが故障する}\}$$

【００６２】

[0047]第１の方法と比較すると、共有後、ｋは不変のままであるため、共有のこれら２つの方法はより良好である。このことは、ＶＩが順次割り振られる場合、稼働しているＶＩが再構成を必要としないことを確実にする。

［仮想データセンターに対するリソース割振りの管理及び適用］

【００６３】

[0048]一実施形態では、最初の管理アーキテクチャは、仮想化データセンターにおいて、仮想エンティティ（例えば、ホストされたサービス）の信頼性保証及びリソースを自律的に管理する。このアーキテクチャにおいて、追加の仮想バックアップノード及びその関連のリンクは、任意のレベルの信頼性保証のために適切に調整される。アイドルの冗長ノードを有するにもかかわらず、より多くの物理リソースを入ってくる新しいサービスが利用できるように、データセンター全体への冗長性のプールは集合的に管理される。さらに、一実施形態において、いくつかのコンポーネントの故障がデータセンター全体を低下させないように、アーキテクチャは、故障に対して障害許容力があるように設計される。

【００６４】

[0049]一実施形態において、冗長機構は、仮想化データセンターにおいて、顧客当たりのレベルで、フォールトトレランスをサポートする。以下は、主要な要求に使用されるリソースの管理の概要、及び仮想化データセンターに適用するための追加の冗長性を提供する。しかし、これらの技術を他の仮想化された環境に適用することができること、及びこうしたことは当業者にとって明らかであることに留意されたい。

仮想化データセンターのためのリソース要求モデルの一例

【００６５】

[0050]リソース要求モデルは、例えばアマゾンＥＣ２クラウドサービス及び他のクラウドサービスプロバイダなど、その物理リソースをリースする仮想化データセンターのものなど、リソースを要求するためのものである。独立サーバインスタンスをリースするより、一実施形態において、リソース要求のモデルは、

【００６６】

１）最低限のＣＰＵ容量の要件を有するワーカー及びマスターノード、並びに

【００６７】

２）これらのノード間の帯域幅保証

を含む全仮想インフラストラクチャ（ＶＩ）に対応する。

【００６８】

[0051]一実施形態において、ワーカーノードは基本的にデータプロセッサ／ナンバークランチャであり、マスターノードはワーカーノードの機能を調整するサーバである。複数のサーバを有するＶＩは、複数のマスターノードを有する。さらに、マスターノードが故障のクリティカルポイントであるため、各ＶＩ要求がマスターノードに関する信頼性保証を要求する。これは、帯域幅保証を加重されたノード間のエッジとする加重グラフとしてモデル化することができ、マスターノードはサブグラフを形成する。このモデルは、様々なニーズを表すのに十分包括的である。

【００６９】
[0052]データセンターのオペレータは、すべての現在のリース及び入ってくる新しいＶＩ要求を管理することを必要とする。
信頼性のための仮想バックアップノード
【００７０】
[0053]一実施形態において、クリティカルなマスターノードに関する信頼性を保証するために、管理アーキテクチャは、空きのあるＣＰＵ及び帯域幅の容量を有する追加のバックアップノードを確保する。すべてのクリティカルなマスターノードの状態を複製し、周知の最適化された同期技術を使用してすべてのバックアップノードに同期することができる。ノードの故障の場合、どのバックアップノードも、故障したノードを置き換えるために「ホットスワップ」の用意ができている。
【００７１】
[0054]図４は、１つのバックアップノード及びそれぞれ割り振られたフェールオーバ帯域幅を備える４ノードのＶＩの一例を示す。図４を参照すると、クリティカルノードＡ及びＢに関して、バックアップノード（黒）及び信頼性のために確保されている帯域幅（点線）が示されている。リンク上の数字は、確保されている帯域幅を表す。このようにして、バックアップノードｅは、クリティカルノードが故障した場合、クリティカルノードに取って代わることができる。
【００７２】
[0055]これは、任意のｋ個のバックアップノードがｎ個のクリティカルノードをカバーするように、容易に拡張することができる。例えば、ｐによって物理ノードの故障の確率を定義する。ｐが物理ノードごとに独立同分布であると推定する。この場合、ｎ個のクリティカルノードに関する信頼性ｒは、以下のように計算される。
【数１４】

$$r \le \sum_{i=0}^{k} \binom{n+k}{i} p^i \overline{p}^{n+k-i} \quad (1)$$

ｎ＋ｋ個のノードのうちの１つ以下が同じ物理ノード上にホストされると仮定される。下記の表Ｉは、２％の物理ノードの故障率の場合に様々な信頼性保証の下でバックアップノードの数によってサポートすることができるクリティカルノードの最大数を示す。
【表１】

TABLE I
NUMBER OF BACKUP NODES REQUIRED, $p = 2\%$

Max. no. of critical nodes (n)				No. of backups (k)
99.9%	99.99%	99.999%	99.9999%	
1	0	0	0	1
8	3	1	0	2
19	10	4	2	3
34	20	11	6	4

バックアップノードの数は、バックアップノードの増加が準直線であるため、表Ｉに示される範囲について十分拡縮する。したがって、冗長性を低減することができるように、いくつかのＶＩにわたってバックアップノードを一緒にプールすることは有益であり、このことは、より良いリソースの使用率につながる。
分散された割振り、冗長マッピング、及び同期

【００７３】
[0056]一実施形態において、効果的に信頼性を管理するために、同じＶＩのすべての仮想ノード、及びそれぞれのバックアップノードは、同じ物理ノード上にホストされない、すなわち、可能な限りデータセンターにわたって分散される。図５は、冗長性をプールし、仮想ノードを分散させる一例を、仮想化データセンターにある４つのＶＩを示すことによって示す。仮想ノードは、同じ物理ノードにスタックされない。クリティカルノードは、その状態をバックアップノードに同期させる。一実施形態において、仮想ノードでのほぼシームレスな動作のために、同期機構がハイパーバイザ層に設けられる。信頼度のレベルに応じて、あるクリティカルノードは、図５のＶＩ－４の場合と同様に、１を超える仮想バックアップノードを必要とする、又はＶＩ－３のようにまったく必要としない場合がある。時として、逆もあり得る。仮想バックアップノード１は、複数のＶＩからの複数のノードをカバーし、ＣＰＵリソースを保護する。このことは、本明細書において、冗長性のプールと呼び、後述する。
仮想データセンターの冗長性のプール
【００７４】
[0057]冗長ノードを一緒にプールし、それらをいくつかのＶＩにわたって共有することによって、バックアップノードの総量及びしたがってアイドルのＣＰＵ容量を低減するために、表Ｉのｎとｋの間の準線形の関係が利用される。例示のために、９９．９９９％の信頼性保証のある５つのクリティカルノードを有するＶＩは、４つのバックアップノードが確保されていることを必要とする。同じレベルの信頼性のために、同じ４つのノードが最高１１個のクリティカルノードをサポートすることができるため、最高６つのクリティカルノードを有する別のＶＩは、追加のバックアップを確保することなく、同じ４つのノードを使用することができる。
【００７５】
[0058]しかし、冗長性のプールは、必ずしも「無料」ではない。無計画なプールは、バックアップノードに関係するフェールオーバ帯域幅を確保する際、かなりのコストをもたらす。ｎ個のクリティカルノード及びｋ個のバックアップノードを含む新しいＶＩ要求に追加される追加のリンクの数は、少なくとも
【数１５】

$$kn + \frac{k}{2}(k - 1),$$

であり、第１の項は、バックアップノードとクリティカルノードとの間に確保されるすべての帯域幅を表し、後者は、バックアップノードを相互接続している帯域幅を表す。したがって、冗長性をプールしながらバックアップノードの数を増加させることは、フェールオーバ帯域幅も増加させるため、逆効果である。
【００７６】
[0059]図６は、２つのＶＩのバックアップノードをプールするときのこのトレードオフを示す。これらの領域の境界は、表Ｉのすべてのｎについて、式ｘ＋ｙ≦ｎ及びｘ，ｙ＞ｎである。見やすいように、これらの線は、境界点がどの線上にもないように、定数によってシフトされる。領域６０１は、ＶＩ－１及びＶＩ－２のバックアップノードをプールすることが完全に価値があるケースを示す。例えば、（５，６）では、５ノードのＶＩ及び６ノードのＶＩがいずれもそれぞれ４つのバックアップノードを必要とする。両方のＶＩに同じ４つのバックアップノードを割り当てることによって、９９．９９９％の同じ信頼性を保証する。これは、領域６０２における場合では、両方のＶＩに対するバックアップノードの数を増やすことなしに行うことができず、例えば、（３，２）では、各ＶＩは、個々に３つのバックアップノードのみを必要とする。しかし、２つを結合すると、５つのノードが追加のバックアップノードを必要とすることになるため、より多くのフェールオーバ帯域幅を必要とする。領域６０３は、１つのＶＩだけが必要より多くのバックアップノードを必要とするケースを表す。

【００７７】
[0060]異なる信頼性保証をサポートしているバックアップノードは、一緒にプールすることもでき、類似のトレードオフ領域を有する。それはすべて、表Ｉに示されるバックアップノードの残りの「サポート容量」に依存する。
仮想化データセンターのための障害許容力のあるアーキテクチャ
【００７８】
[0061]図７は、仮想化データセンターの物理リソースの上にある管理アーキテクチャの一実施形態を示しており、リソース要求を管理する集中化したコントローラとして作用する。各コンポーネントは、個々に機能しているエンティティとなるように設計されており、故障に対して障害許容力を確実にするための方策を有する。
【００７９】
[0062]物理リソースアカウンティングコンポーネント（ｐｈｙｓｉｃａｌ　ｒｅｓｏｕｒｃｅ　ａｃｃｏｕｎｔｉｎｇ　ｃｏｍｐｏｎｅｎｔ）７０６は、入ってくる新しい要求に対するリソースの割振り時に必要な、仮想化データセンターにおいて割り振られていない残りのリソースを追跡する。価格付けポリシー（ｐｒｉｃｉｎｇ　ｐｏｌｉｃｙ）７０５は、動的な価格付けを容易にするために、物理リソースアカウンティングコンポーネント７０６からのその入力を引き出す。リソース割振りエンジン７００及びリソース解放モジュール７０９のみが、物理リソースアカウンティングコンポーネント７０６を更新することができる。一実施形態において、更新は、要求及びリーブイベントに応答して起きる。
【００８０】
[0063]一実施形態において、物理リソースアカウンティングコンポーネント７０６のための障害許容力を確実にする方法が２つある。（ｉ）周知の障害許容力のあるデータベースが使用される、又は（ｉｉ）データの複数のコピーは、データに対する書き込み及び読み取りをそれぞれマルチキャスト及びエニーキャストとして、別々に格納される。一実施形態において、データは、（ＰｈｙＮｏｄｅ，ｒＣＰＵ）及び（ＰｈｙＬｉｎｋ，ｒＢＷ）の形の鍵－値の組として格納され、この場合、ＰｈｙＮｏｄｅ及びＰｈｙＬｉｎｋは、物理ノード及びリンクをそれぞれ一意に識別し、ｒＣＰＵ及びｒＢＷは、使用可能なＣＰＵ及び帯域幅リソースの量をそれぞれ提供する。
【００８１】
[0064]上述したように、時として、ＶＩにわたってバックアップノードをプールするとき、ＣＰＵ及び帯域幅を保護する際のトレードオフが存在する。一実施形態において、信頼性ポリシー７０５は、入ってくる新しいＶＩのバックアップノードをデータセンターにおける別の既存のＶＩと共にプールするべきかどうかを指定する決定ルールのリストを備える。図では、これらのルールは、トレードオフ領域の境界を表す（一例として図６を参照）。故障に対する障害許容力を確実にするための戦略は、物理リソースアカウンティングコンポーネント７０６と同様である。
【００８２】
[0065]リソース割振りエンジン７００は、入ってくる要求に対するリソースをマップし、確保する役割を果たす。図８は、入ってくる各要求にサービスを提供するプロセスの一実施形態のフロー図である。プロセスは、ハードウェア（例えば、論理回路、回路など）、ソフトウェア（汎用コンピュータシステム又は専用マシン上で稼働されるものなど）、又は両方の組み合わせを備えることができるリソース割振りエンジンにおいて処理論理回路によって実行される。図８を参照すると、プロセスは、処理論理回路が入ってくる要求を受信することによって開始する（処理ブロック８０１）。入ってくる要求を受信することに応答して、処理論理回路は、入ってくる要求に対応するために必要なバックアップノードの数を計算する（処理ブロック８０２）。その後、処理論理回路は、バックアップノードを既存のＶＩと共にプールするかどうか検査する（処理ブロック８０３）。処理論理回路がバックアップノードを既存のＶＩと共にプールすると決定する場合、処理論理回路は現在のバックアップに新しい制約及び帯域幅を追加し（処理ブロック８０４）、プロセ

スは処理ブロック８０７に移行する。そうでない場合、処理論理回路は、新しいバックアップノード及び帯域幅を作り（処理ブロック８０５）、これらの新しいリソースを、リソース割振りのための最初のリソース要求に追加し、プロセスは、処理ブロック８０７に移行する。処理ブロック８０７で、処理論理回路は、外部ツールを介して線形最適化を解決する。こうしたツールの例には、それだけには限定されないが、ＣＯＩＮ　ＣＢＣ、ｉＬｏｇ　ＣＰＬＥＸ、及びＭＯＳＥＫなどがある。

【００８３】

[0066]次に、処理論理回路は、外部ツールの出力を介した使用可能なリソースに基づいてソリューションが実行可能かどうかを決定する（処理ブロック８０８）。より詳細には、ソルバー（solver）が稼働され、ソルバーが所与の制約を有するソリューションを探し出すことができない場合、ソリューションは実行可能ではない。ソルバーがソリューションを戻す場合、ソリューションは実行可能である。そうである場合、処理論理回路は、他のコンポーネントを更新し（処理ブロック８０９）、プロセスが終了する。そうでない場合、処理論理回路は、リソース要求を拒否し（処理ブロック８１０）、プロセスが終了する。

【００８４】

[0067]一実施形態において、仮想ノードから物理ノード、及び仮想リンクから物理パスへのマッピングの問題は、仮想ノード間の帯域幅予約が物理ノード間のフローであり、仮想ノードと物理ノードとの間のフローの存在がマッピングを示すというマルチコモディティフロー問題として、当分野で周知の方法で定式化される。これは、以下の目的に関する線形最適化問題である。

【数１６】

$$P1:\ \min\ \sum_{u}\sum_{x}\alpha_{ux}\rho_{ux}\mu_{ux}+\sum_{u,v}\sum_{l}\beta_{uv}^{l}f_{uv}^{l}\quad(2)$$

式中、ρ及びｆ$^{l}{}_{uv}$は変数、ρはブール変数であり、仮想ノードｘが物理ノードｕにマップされる場合、真である。ｆ$^{l}{}_{uv}$は、物理リンク（ｕ，ｖ）上に「流れている」仮想リンク１の帯域幅の量であり、負ではない。ｆ$^{l}{}_{uv}$についての従来のフロー保護制約が適用され、従来のフロー保護制約（ｆｌｏｗ　ｃｏｎｓｅｒｖａｔｉｏｎ　ｃｏｎｓｔｒａｉｎｔｓ）は、当業者によって十分理解されている。さらに、制約

【数１７】

$$\sum_{x}\rho_{ux}=1\ \text{及び}\ \sum_{u}\rho_{ux}\le 1\quad(3)$$

は、仮想ノードと物理ノードと間の１対１のマッピングを確実にし、仮想ノードは、上述したように分散される。ＣＰＵ及び帯域幅に関して新しいＶＩによって消費される全リソースは、残っている物理リソースの量に供され、すなわち、

【数１８】

$$\sum_{x}\rho_{ux}\mu_{x}\le \mathrm{rCPU}_{u}\ \ \text{及び}\ \ \sum_{l}f_{uv}^{l}\le \mathrm{rBW}_{uv}\quad(4)$$

であり、μｘは、仮想ノードｘによって必要とされるＣＰＵ容量である。入力α$_{\mu x}$及びβ$^{l}{}_{uv}$はそれぞれ、リソースがリースされるとき、データセンターのオペレータに対して、ＣＰＵ及び帯域幅当たりの純コスト（マイナス収入）を表す。これらは、後述する価格付けポリシーから導出される。

【００８５】

[0068]バックアップノード及びフェールオーバ帯域幅の追加は、信頼性ポリシー、すなわちバックアップノードがプールされるかどうかに依存する。そうでない場合、問題は、新しいバックアップノード及び帯域幅を含んでＰ１を解くほど単純である。そうでなければ、マッピング変数ρ$_{\mu x}$への追加の制約は、新しいＶＩの仮想ノードと現在の仮想ノー

ドとの間の重複がないことを確実にするために、Ｐ１に追加され、すなわち、すべての占有されたｕについて、$\rho_{ux}=0$である。

【００８６】

[0069]Ｐ１に対する実行可能なソリューションがある場合、新しい要求にのみ対応することができる。次いで、細い両方向矢印を介してリソース割振りエンジン７００にリンクされているコンポーネント、すなわちＶＩマップ７０７、ホットスワップマップ７０８、及びアカウンティングコンポーネント７０６がＰ１からのソリューションによって更新される。そうでなければ、単に不十分な物理リソースのために、更新が拒絶される。

【００８７】

[0070]このメイン制御コンポーネントが障害許容力を有するようになる簡単な戦略は、複数のインスタンスにわたって同じ要求を実行することである。より効果的な方法は、いくつかの要求を処理しているが、低いｒＣＰＵｕ及びｒＢＷｕｖの値を使用して競合状態を防止するインスタンスを複数有することである。しかし、過度の拒否のリスクがある。

価格付けポリシー

【００８８】

[0071]価格付けポリシーは、Ｐ１の入力α_{ux}及び$\beta^{1}{}_{uv}$に影響するリソースの価格を指定する。ここで使用する価格付け戦略に固定する必要はなく、むしろ、価格付けモジュール７０５が設けられ、できる限り一般的である。特に、動的な価格付けがサポートされ、この価格付けは、需要を抑えることができ、リソースのより効率的な使用率をもたらすことができる。ある期間にわたる物理リソースアカウンティングモジュール７０６からの入力及びリソース割振りエンジン７０５からのフィードバックにより、価格付けモジュール７０５は、信頼性保証、物理リソースのタイプ（リンク、ノード）、受容率、及びリース期間の面において仮想ＣＰＵ及び帯域幅の価格を動的に設定することができる。

仮想インフラストラクチャマップ及びホットスワップマップ

【００８９】

[0072]ＶＩマップ７０７は、認められるすべてのＶＩ、及び仮想エンティティのその物理リソースに対するマッピング、すなわち、その物理サーバ及び確保されたＣＰＵの量に対する仮想ノードのマップ、及び物理パス及びそのパスに沿って確保された帯域幅の量に対する仮想リンクのマップを記録する。さらに、ＶＩが使用するバックアップノードのプールも格納される。

【００９０】

[0073]ホットスワップマップ７０８は、バックアップノードのすべての現在のプール及びそれぞれの残りのサポート容量を記録する。この情報は、ＶＩマップと共に、入ってくる新しいＶＩがバックアップノードの既存のプールを使用することができるか、又は新しいＶＩのための別の新しいプールを作ることができるかどうかをリソース割振りエンジン７００が決定するのを助ける。一実施形態において、新しいＶＩのためのマッピングのソリューションがいったん取得されると、リソース割振りエンジン７００はこれらの２つのマップに書き込む。

【００９１】

[0074]一実施形態において、これら２つのコンポーネントに対する障害許容力のある戦略は、物理リソースアカウンティングモジュール７０６のものと同じである。その理由は、これらのコンポーネントがデータベースのようなコンポーネントだからである。

リソースの解放

【００９２】

[0075]ＶＩによって使用されるリソースは、リースの終了時に解放されなければならない。競合状態を防止するために、リソース割振りエンジン７００による同期ロックが解除されるまで、リソース解放モジュール７０９は、一時的にこれらのリソースを保持するガベージコレクタとして働く。このコンポーネントが故障する場合、２つのマップに関する簡単なチェック、及び物理リソースのアカウンティング検証がこのコンポーネントを回復する。

同期及び回復機構

【００９３】

[0076]これら２つの機構は、分散された方法で機能しているあらゆる物理ノードにおけるローカルなサービスである。一実施形態において、ノード間の同期は、物理ノードのハイパーバイザで管理され、物理ノード間の監視は、ハートビート、同期信号、又は当業者に周知である他の分散監視方法を介することができる。故障が検出されると、回復手順が開始し、コントロールアーキテクチャで進行中の動作すべての代わりをする。ホットスワップノードは、均一のランダム化によって分散された方法で、各ＶＩの仮想の隣接者によって選択され、結合は任意に中断される。

【００９４】

[0077]したがって、データセンターにおいてホストされる仮想インフラストラクチャにおける信頼性保証を自律的に管理することができるフォールトトレラントアーキテクチャが開示される。ここで、信頼性は、仮想バックアップノードのプール及び確保されたフェールオーバ帯域幅によって保証される。アイドルのＣＰＵ容量を保護するために、バックアップがプールされ、帯域幅に対するトレードオフが定義される。バックアップを含めて、すべての仮想エンティティの物理リソースは、線形最適化フレームワークを介して割り振られる。データセンターのリソース使用率を追跡し、考慮に入れる他のコンポーネントも定義される。個々のコンポーネントは、個々に動作するように設計されており、障害に対する許容力を確実にするための方策を有する。

［リソースの割振り：混合整数計画問題（ＭＩＸＥＤ　ＩＮＴＥＧＥＲ　ＰＲＯＧＲＡＭＭＩＮＧ　ＰＲＯＢＬＥＭ）］

【００９５】

[0078]ＶＩリソース割振り問題は、マルチコモディティフロー問題（ＭＣＦ）に類似する混合整数計画問題として定式化することができる。ノード間の帯域幅の要求は、フローとしてモデル化される。物理ノードと仮想ノードとの間のマッピングは、余分な「マッピング」エッジを追加し、当分野で周知の方法で、フロー問題を解決する際に、仮想ノード当たりこうした１つのエッジだけが使用されることを確実にすることによって構築される。

【００９６】

[0079]一実施形態において、ＭＣＦは、ＶＩノード及びリンクを物理的なインフラストラクチャにマップするために使用される。しかし、ＭＣＦは、（ｉ）バックアップリンクＬが可能な限り重複することができ、（ｉｉ）バックアップノードのマッピングは、好適な１組の物理ノードΦに限定されるように制約する。アルゴリズム１は、Φを取得し、ＶＩ、及びそのバックアップノード、並びにリンクを物理的なインフラストラクチャにマップして、信頼性ｒを保証するための手順を列挙する。

【００９７】

[0080]図９は、帯域幅予約がどのように重複し得るかの一例を示す。図９を参照すると、左は、２つの冗長ノード（黒）を有する２ノード仮想トポロジである。ノードｃは、クリティカルノードである。したがって、ノードｘを冗長ノードにリンクしている１単位の帯域幅が確保されていなければならず、それは結果としてリンクＤＥ上の２単位の予約となり得る。しかし、冗長ノードは常にノードｃの１つのインスタンスしか引き継ぐことができないので、過度の予約が生じる。本明細書において使用されるＭＣＦの制約形式において、任意のトポロジ上のこれらの重複を最大にする試みがある。

【００９８】

[0081]上記のように、既存のＶＩのバックアップノードを、入ってくる新しいＶＩと共有しながら、確実に不変にすることは価値がある。リソース割振り手順が以下に提供される。行６～１４は、そのバックアップノードを共有する適切なＶＩを貪欲に探す。これらのＶＩ候補を、「サポート容量」に関して配列することができる。例えば、ｋ＝３のバックアップノードは、ｒ＝９９．９９％のために、８～２１の仮想ノードの間をサポートすることができる。８ノードのＶＩは、２０ノードのＶＩより多くのサポート容量を有して

おり、したがって好ましい。この配列は、すでに共有されているＶＩを考慮に入れなければならない。行１１は、そのバックアップノードが好適な物理的位置Φに限定されるＶＩを埋め込むことを試みる。共有が可能でない場合、バックアップノードは行１５のようにN^Pにおけるどこからでも選択される。

【表２】

アルゴリズム1: リソース割付け手順の一例

```
1: procedure ALLOCATE(VirInf, PhyInf, r)
2:     n←|VirInf.N^V|
3:     Compute k from n, r, p.              > inverse of (1)
4:     Compute L from VirInf given k
5:     VirInf.augment (N^K, L)
6:     for all ς in PhyInf.getVIs()do        > orderd
7:             if |ς.N^K| ≠ k then
8:                 continue
9:             end if
10:            Φ ← phyLoc(ς.N^K)
11:            if MCF_OL(VirInf, PhyInf, Φ, ς) = True then
12:                return True
13:            end if
14:    end for
15:    return MCF_OL(VirInf, PhyInf, N^P, NULL)
16: end procedure
```

【００９９】

[0082]ＭＣＦ問題は、以下の通りに定義される。マッピングのための拡張エッジの組をR^Pによって示し、

【０１００】

$$R^P=\{(u, x),(x,u) \mid \forall x \in N^V \cup N^K, u \in \Phi[x]\} \quad (7)$$

となり、この場合、各エッジは無限の帯域幅を有する。Φ［ｘ］$\subseteq N^P$は、仮想ノードｘをホストすることができる物理ノードの組である。ｘがバックアップノードであり、別のＶＩのバックアップと共有されることになっている場合、Φ［ｘ］はアルゴリズム１で定義されたΦに等しい。図１０は、この拡張構造の一例を示す。ノードａ及びｂは、別のＶＩの冗長ノードである。これらのノードは、物理ノードのＣ及びＤにあり、ノードｘ及びｙを有する新しいＶＩの冗長ノードであり得る。同じＶＩのノードが同じ物理ノードにホストされない場合があるので、Φ［ａ］＝｛Ｃ｝，Φ［ｂ］＝｛Ｄ｝、及びΦ［ｘ］＝Φ［ｙ］＝｛Ａ，Ｂ，Ｅ｝である。

【０１０１】

[0083]３つの組は、以下の通りに定義される。

【０１０２】

$$N^A=N^P \cup N^V \cup N^K \quad (8)$$

【０１０３】

$$E^A=E^P \cup R^P \quad (9)$$

【０１０４】

$$C^K=C^V \cup \{x \mid x \in N^V, \exists (c, x) \in E^V, \forall c \in C^V\} \quad (10)$$

この場合、N^Aは、すべての仮想、物理、及び冗長ノードの組であり、E^Aは物理及びマッピングエッジの組であり、C^Kは、冗長ノードがL^1においてリンクされるノードの組である。

【０１０５】

[0084]一実施形態において、仮想ノードとバックアップノードとの間の帯域幅予約は、フローとしてモデル化される。これらのフローによって使用される帯域幅の量は、ＭＣＦ問題に対する変数である。一実施形態において、４種類のフローがある。

【０１０６】

２つの仮想ノードの間のフロー。ｘ，ｙ∈Ｎ^{V}。リンク（ｉ，ｊ）∈Ｅ^{A}において使用する帯域幅の量は、ｆ^{xy}［ｉｊ］によって示される。

【０１０７】

Ｌ^{1}:冗長ノードａ∈Ｎ^{K}と仮想ノードｙ∈Ｃ^{K}との間のフロー。冗長ノードａが何らかの仮想ノードｘを引き継ぐ場合を除いて、これらのフローのいずれかにおける実際の帯域幅は、ゼロである。こうした回復が生じるときに、リンク（ｉ，ｊ）∈Ｅ^{A}において使用される帯域幅の量は、

【数１９】

$$f_{L^1}^{axy}[ij].$$

によって示される。これによって、冗長なフロー間の重複のモデル化が可能になる。

【０１０８】

冗長ノードＮ^{K}と仮想ノードｘ∈Ｃ^{K}との間のリンクにおける集計フロー。これは、リンク（ｉ，ｊ）上での重複後確保された帯域幅の実際の量を反映する。これは、ｆ$_{0}^{x}$［ｉｊ］で示される。

【０１０９】

Ｌ^{2}：２つの冗長ノードａ，ｂ∈Ｎ^{K}間のフロー。リンク（ｉ，ｊ）∈Ｅ^{A}において使用される帯域幅の量は、

【数２０】

$$f_{L^2}^{ab}[ij].$$

によって示される。

【数２１】

$$f_{L^1}$$

のフローとは異なり、これらのフローは、重複しない。これは、Ｌ^{1}リンクに対する追加の信頼性を提供しない、仮想ノードｘを介してノードａとｂとをリンクするパス（ａ，ｘ，ｂ）を有することの取るに足らないソリューションを回避するためである。

【０１１０】

[0085]一実施形態において、物理ノードと、仮想又は冗長ノードとの間の双方向のマッピングは、二値変数ｐ$_{ij}$，（ｉ，ｊ）∈Ｒ^{P}によってモデル化される。リンク（ｉ、ｊ）及び（ｊ，ｉ）の間を流れているフローの総量が正である場合、ρ$_{ij}$＝１、そうでない場合、０である。したがって、ＭＣＦに対するソリューションがρ$_{xu}$＝１を提供する場合、仮想ノードｘは、物理ノードｕ上にホストされる。

【０１１１】

[0086]ＭＣＦの目的関数は、

【数２２】

$$\min \sum_{w\in N^P}\alpha_w \sum_{x\in N^V}\rho_{xw}\mu_x + \sum_{(u,v)\in E^P}\beta_{uv}\times\left[\sum_{x\in C^K}f_o^x[uv]+\sum_{(a,b)\in L^2}f_{L^2}^{ab}[uv]+\sum_{(x,y)\in E^V}f^{xy}[uv]\right] \quad (11)$$

として定義され、この場合、α$_{w}$及びβ$_{uv}$はそれぞれ、ノード及びリンクの重みである。これによって、割り振られる計算及び帯域幅の加重和が最低限に抑えられる。負荷バランシングを達成するために、重みはそれぞれ、

【数２３】

$$\frac{1}{M_w+\grave{o}}$$

及び

【数２４】

$$\frac{1}{H_{uv}+\grave{o}}$$

と設定することができる。ＭＣＦに対する制約は、以下の通りである。
マッピング制約：
【数２５】

$$\sum_{u\in\Phi[x]}\rho_{ux}=1,\quad \forall x\in N^V\cup N^K \qquad (12)$$

【数２６】

$$\sum_{x\in N^V\cup N^K}\rho_{xu}\le 1,\quad \forall u\in N^P \qquad (13)$$

【０１１２】
$\rho_{ij}\leqq H_{ji}$，∀（ｉ，ｊ）∈R^P（１４）
【０１１３】
$\rho_{ij}=\rho_{ji}$，∀（ｉ，ｊ）∈R^P（１５）
制約（１２）及び（１３）は、各仮想ノードが単一の物理ノードのみにマップされ、１つの物理ノードに１つ以下の仮想ノードしかマップできないようにすることを確実にする。
制約（１４）及び（１５）は、実行可能なフローがリンク（ｉ，ｊ）上にマップされるとき、二値変数ρ_{ij}を強制的に１にし、そうでなければ０にする。
（共有を条件とした）計算容量の制約：
【０１１４】
$p_{iu}\mu_u\leqq M_u$，∀ｕ∈N^P，∀ｉ∈$N^V\cup N^K$　（１６）
これは、マップされた仮想及び冗長ノードが物理ノードｕ上の使用可能な容量M_uを超えないことを確実にする。冗長ノード、ａ∈N^Kについて、供給される最大能力は、
【数２７】

$$\max_{u\in C^V}\mu_u .$$

である。さらに、この容量が共有の冗長ノードのものを超える場合、バランスが供給されるだけで良い。
２つの仮想ノード間の帯域幅予約のためのフロー保護制約：
【数２８】

$$\sum_{u:(x,u)\in R^P}\left[f^{xy}[xu]-f^{xy}[ux]\right]=\eta_{xy},\quad \forall(x,y)\in E^V \qquad (17)$$

【数２９】

$$\sum_{u:(u,y)\in R^P}\left[f^{xy}[yu]-f^{xy}[uy]\right]=-\eta_{xy},\quad \forall(x,y)\in E^V \qquad (18)$$

【数３０】

$$\sum_{i\in N^A}\left[f^{xy}[ui]-f^{xy}[iu]\right]=0,\quad \forall u\in N^P,\forall(x,y)\in E^V \qquad (19)$$

【０１１５】
[0087]制約（１７）及び（１８）は、仮想ノードｘから仮想ノードｙに発せられる仮想リンク（ｘ，ｙ）の総帯域幅η_{xy}を定義する。制約（１９）は、フローが中間の物理ノードで保護されることを確実にする。すなわち、ノードｕから流れている総帯域幅がそのノード内に流れ込む総帯域幅に等しい。

Ｌ[1]リンク上の帯域幅を確保するためのフロー保護及び重複制約：

【数３１】

$$\sum_{u:(a,u)\in R^P}\left[f_{L^1}^{acy}[au]-f_{L^1}^{acy}[au]\right]=\eta_{cy},\quad \forall(a,y)\in L^1,\forall c\in C^V \qquad (20)$$

【数３２】

$$\sum_{u:(u,y)\in R^P}\left[f_{L^1}^{acy}[yu]-f_{L^1}^{acy}[uy]\right]=-\eta_{cy},\quad \forall(a,y)\in L^1,\forall c\in C^V \qquad (21)$$

【数３３】

$$\sum_{i\in N^A}\left[f_{L^1}^{acy}[ui]-f^{acy}_{L^1}[iu]\right]=0,\quad \forall u\in N^P,\forall(c,y)\in E^V,\forall a\in N^K \qquad (22)$$

【数３４】

$$\sum_{a\in N^K,c\in F,(c,y)\in E^V} f_{L^1}^{acy}[ij]\le f_O^y[ij],\quad \forall(i,j)\in E^A,\forall y\in C^K,\forall F\subseteq C^V,|F|\le k \qquad (23)$$

冗長ノードａがクリティカルノードｘの代わりをする仮想ノードｙへの各フローについて、制約（２０）～（２２）は、（１７）～（１８）における仮想フローのものと似たフロー保護モデルを定義する。制約（２３）は、すべてのａにわたって合計される代わりに、重複し得る冗長なフローを処理する。１つの冗長ノードａのみが、いつでもクリティカルノードｃの代わりとなり得る。次いで、フロー

【数３５】

$$f_{L^1}^{acy}[ij]$$

及び

【数３６】

$$f_{L^1}^{bcy}[ij]$$

は、重複する可能性があり、すなわち、リンク（ｉ，ｊ）について、

【数３７】

$$\max_{a\in N^K}\ f_{L^1}^{acy}[ij]\le f_o^y[ij]$$

である。しかし、重複は、フロー

【数３８】

$$f_{L^1}^{acy}[ij]$$

及び

【数３９】

$$f_{L^1}^{bdy}[ij]$$

では起こらない可能性があり、この場合、冗長ノードａは、クリティカルノードｃの代わりとなり、別のノードｂは、クリティカルノードｄの代わりとなる。これは、最大ｋ個の置き換えまで生じる。制約（２３）は、これらの関係を捕捉する。

（共有及びＣ[V]を条件とした）Ｌ[2]リンク上の帯域幅予約のためのフロー保護制約：

【数４０】

$$\sum_{i\in N^A}\left[f_{L^2}^{ab}[ui]-f_{L^2}^{ab}[iu]\right]=0,\quad \forall u\in N^P,\forall(a,b)\in L^2 \qquad (24)$$

【数４１】

$$\sum_{u:(a,u)\in R^P}\left[f_{L^2}^{ab}[au]-f_{L^2}^{ab}[ua]\right]=\max_{x,y\in C^V}\eta_{xy},\quad \forall(a,b)\in L^2 \qquad (25)$$

【数４２】

$$\sum\left[f_{L^2}^{ab}[bu]-f_{L^2}^{ab}[ub]\right]=-\max_{x,y\in C^V}\eta_{xy},\quad \forall(a,b)\in L^2 \qquad (26)$$

２つの冗長ノードａとｂとの間のフロー保護制約は、（１７）～（１９）における仮想フローのものと変わらない。供給される帯域幅は、C^Vのノードを相互接続する仮想リンクの最大数である。しかし、これらの制約は、２つのケースにおいてのみ必要である。

【０１１６】

１）

【数４３】

$$L^2\neq\varnothing.$$

定理２から、クリティカルノードを相互接続している仮想リンクがある場合、L^2についての帯域幅を供給するだけで良い。

【０１１７】

２）共有。帯域幅は、共有されることになっているＶＩにおいて、すでに供給されている。帯域幅の供給が十分でない場合、これらの制約は、バランスを供給するために存在する。

物理リンクにおけるリンク容量制約：

【数４４】

$$\sum_{x\in C^K}[f_o^x[uv]+f_o^x[vu]]+\sum_{(a,b)\in L^2}[f_{L^2}^{ab}[uv]+f_{L^2}^{ab}[vu]]+\sum_{(x,y)\in E^V}[f^{xy}[uv]+f^{xy}[vu]]\le H_{uv},\quad \forall(u,v)\in E^P \qquad (27)$$

制約（２７）は、両方向の物理リンク（ｕ、ｖ）上のすべてのフローを考慮に入れる。これは、物理的に残っている帯域幅H_{uv}未満でなければならない。

拡張マッピングリンクにおけるリンク容量制約：

【数４５】

$$\sum_{x\in C^K}[f_o^x[ij]+f_o^x[ji]]+\sum_{(a,b)\in L^2}[f_{L^2}^{ab}[ij]+f_{L^2}^{ab}[ij]+f_{L^2}^{ab}[ji]]+\sum_{(x,y)\in E^V}[f^{xy}[ij]+f^{xy}[ji]]\le H_{ij}\rho_{ij},\quad \forall(i,j)\in R^P \qquad (28)$$

厳密に言えば、帯域幅H_{ij}が無限であるため、マッピングリンクにおける制約はないはずである。しかし、この制約は、マッピング制約（１４）及び（１５）と連動して、いずれかの方向のそのリンクにおける任意の正のフローがある場合、マッピング二値変数ρ_{ij}を強制的に１にし、そうでない場合０にする。

領域制約：

$f^{xy}[ij]\geqq 0$，$\forall i, j\in N^A$，$\forall(x, y)\in E^V$（２９）

【数４６】

$$f^{xy}[ij] \geq 0, \quad \forall i,j \in N^A, \forall (x,y) \in E^V \tag{29}$$

$$f_{L^1}^{ax}[ij] \geq 0 \quad \forall i,j \in N^A, \forall (a,x) \in L^1 \tag{30}$$

$$f_{L^2}^{ab}[ij] \geq 0, \quad \forall i,j \in N^A, \forall (a,b) \in L^2 \tag{31}$$

$$f_o^x[ij] \geq 0, \quad \forall i,j \in N^A, \forall x \in C^K \tag{32}$$

$$\rho_{ij} \in \{0,1\}, \quad \forall (i,j) \in R^P \tag{33}$$

【０１１８】

$f_0{}^x[ij] \geqq 0, \forall i, j \in N^A, \forall x \in C^K$（３２）

【０１１９】

$\rho_{ij} \in \{0, 1\}, \forall (i, j) \in R^P$（３３）

これらは、この変更されたＭＣＦ問題のすべての変数における領域制約であり、すべてのフローは、非ゼロでなければならず、マッピング変数はバイナリである。

【０１２０】

[0088]インフラストラクチャが急速に仮想化されるにつれて、信頼性保証を仮想化インフラストラクチャに提供する必要性が増加している。上記には、仮想化層自体における信頼性保証について記載されている。冗長ノードは、ネットワークにわたって分散される仮想ノードとすることができる。このために、供給された帯域幅を完備したｎ：ｋの冗長アーキテクチャ、及び物理ネットワークにわたって仮想化されたネットワークを割り振る方法が提案される。冗長ノード及びリンクによって使用されるリソースを保護するために、これらの冗長ノードをＶＩにわたって共有することができ、供給の間、それらの帯域幅が重複する。

コンピュータシステムの一例

【０１２１】

[0089]図１１は、本明細書において記載されている１つ又は複数の動作を実行することができるコンピュータシステム例のブロック図である。図１１を参照すると、コンピュータシステム１１００は、典型的なクライアント又はサーバコンピュータシステムを備えることができる。コンピュータシステム１１００は、情報を伝えるための通信機構又はバス１１１１、及びバス１１１１に結合されて情報を処理するためのプロセッサ１１１２を備える。プロセッサ１１１２は、それだけには限定されないが、ペンティアム（Ｐｅｎｔｉｕｍ）（商標）、パワーＰＣ（ＰｏｗｅｒＰＣ）（商標）、アルファ（Ａｌｐｈａ）（商標）など、マイクロプロセッサを含む。

【０１２２】

[0090]システム１１００は、ランダムアクセスメモリ（ＲＡＭ）、又はバス１１１１に結合されてプロセッサ１１１２によって実行される命令及び情報を格納するための他の動的な記憶デバイス１１０４（メインメモリと呼ばれる）をさらに備える。プロセッサ１１１２による命令の実行中に一時的数値変数又は他の中間情報を格納するために、メインメモリ１１０４を使用することもできる。

【０１２３】

[0091]コンピュータシステム１１００は、読み取り専用メモリ（ＲＯＭ）及び／又はバス１１１１に結合されてプロセッサ１１１２のための静的情報及び命令を格納するための他の静的記憶デバイス１１０６、並びに磁気ディスク又は光ディスクなどのデータ記憶デバイス１１０７及びその対応するディスクドライブも備える。データ記憶デバイス１１０７は、バス１１１１に結合されて情報及び命令を格納するためのものである。

【０１２４】

[0092]コンピュータシステム１１００は、バス１１１１に結合されて、コンピュータユ

ーザに対して情報を表示するために、例えばブラウン管（ＣＲＴ）又は液晶式ディスプレイ（ＬＣＤ）などのディスプレイデバイス１１２１にさらに結合することができる。英数字及び他のキーを含む英数字入力デバイス１１２２も、バス１１１１に結合されてプロセッサ１１１２に情報及びコマンドの選択を伝えるようにすることができる。追加のユーザ入力デバイスは、バス１１１１に結合されて、プロセッサ１１１２に指示情報及びコマンドの選択を伝える、及びディスプレイ１１２１上のカーソルの動きを制御するための、例えばマウス、トラックボール、トラックパッド、スタイラス、又はカーソル方向キーなどのカーソル制御１１２３である。

【０１２５】

[0093]バス１１１１に結合することができる他のデバイスは、ハードコピーデバイス１１２４であり、このデバイスを使用して、例えば紙、フィルム、又は類似のタイプの媒体に情報を表すことができる。バス１１１１に結合することができる他のデバイスは、電話又はハンドヘルドパームデバイスに対して通信を行うための有線／無線通信機能１１２５である。

【０１２６】

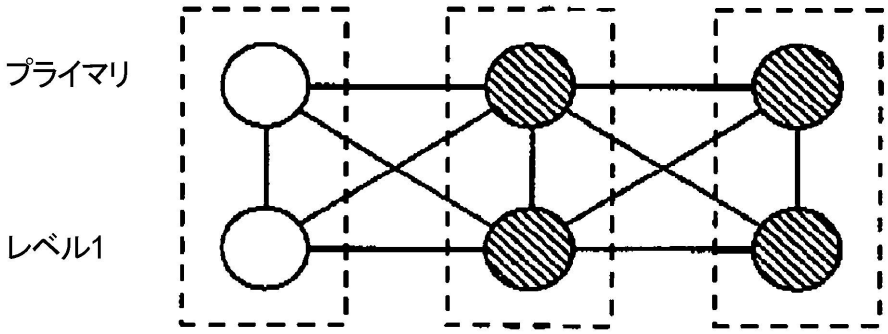
[0094]システム１１００及び付随するハードウェアのコンポーネントの任意のもの又は全部を本発明において使うことができることに留意されたい。しかし、コンピュータシステムの他の構成がデバイスの一部又は全部を含むことができることを理解されよう。

【０１２７】

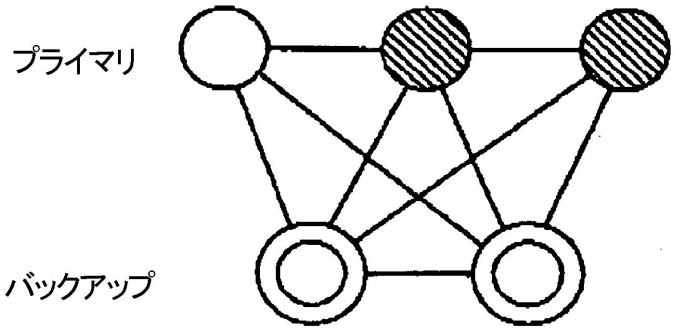
[0095]本発明の多くの変更及び修正は、間違いなく前述の説明を読んだ後に当業者にとって明らかになるが、例として示され、記載されるいかなる特定の実施形態も決して制限と見なされないことを理解されたい。したがって、様々な実施形態の詳細への言及は、本発明にとって重要であると見なされる特徴だけを詳述する請求項の範囲を制限することを目的としない。

【図１】

(a) ベース仮想ネットワーク



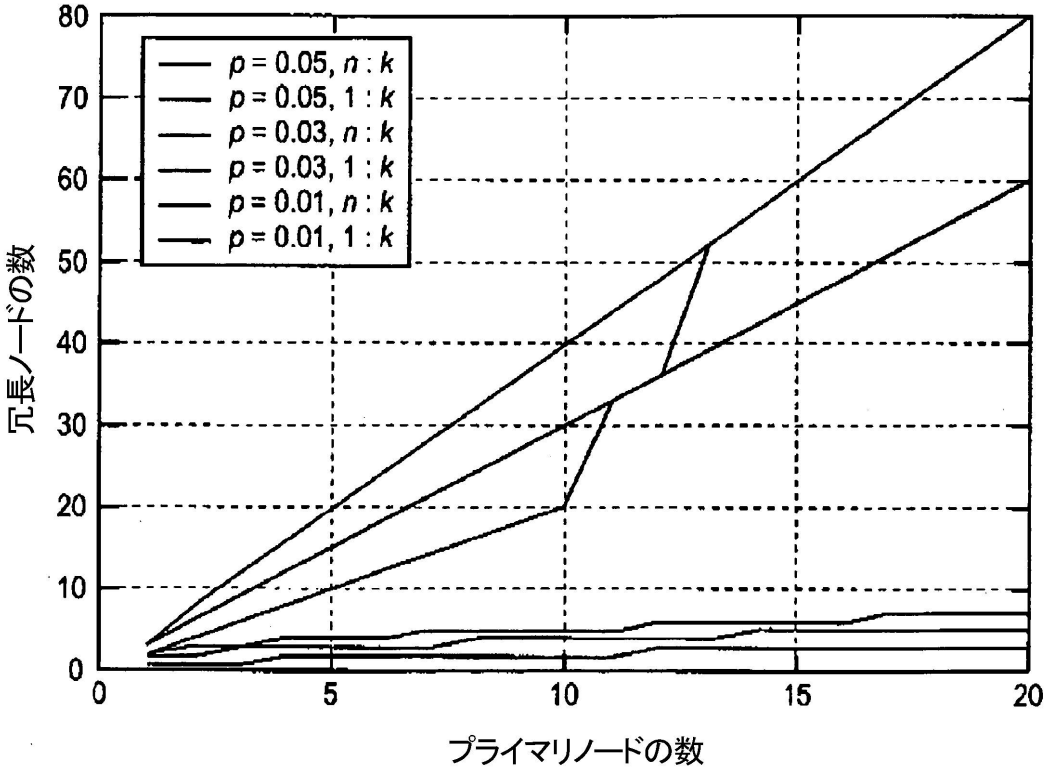
(b) シンプルな1:kの複製



(c) n:kの複製

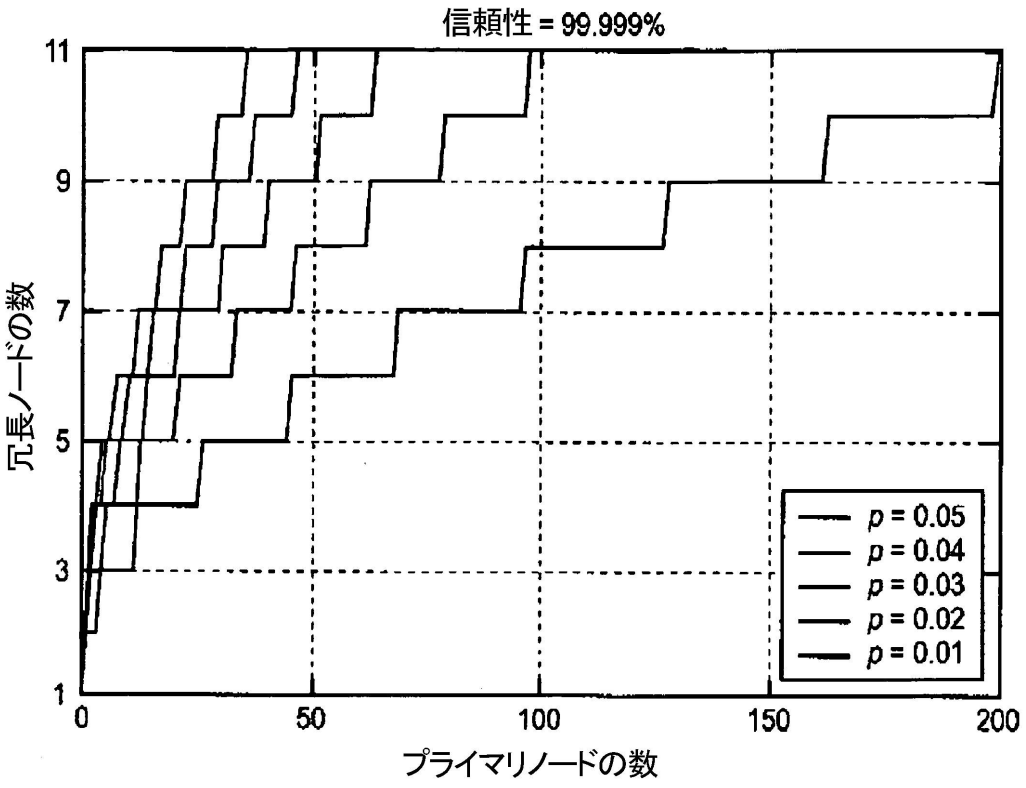
フォールトトレラントアーキテクチャの比較

【図２】



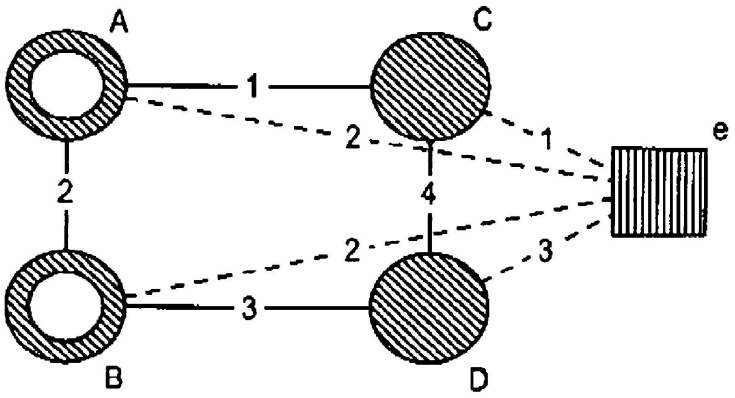
99.999%のノード信頼性のために必要な冗長ノードの数。
pは、各ノードの個別の故障の確率である。

【図３】

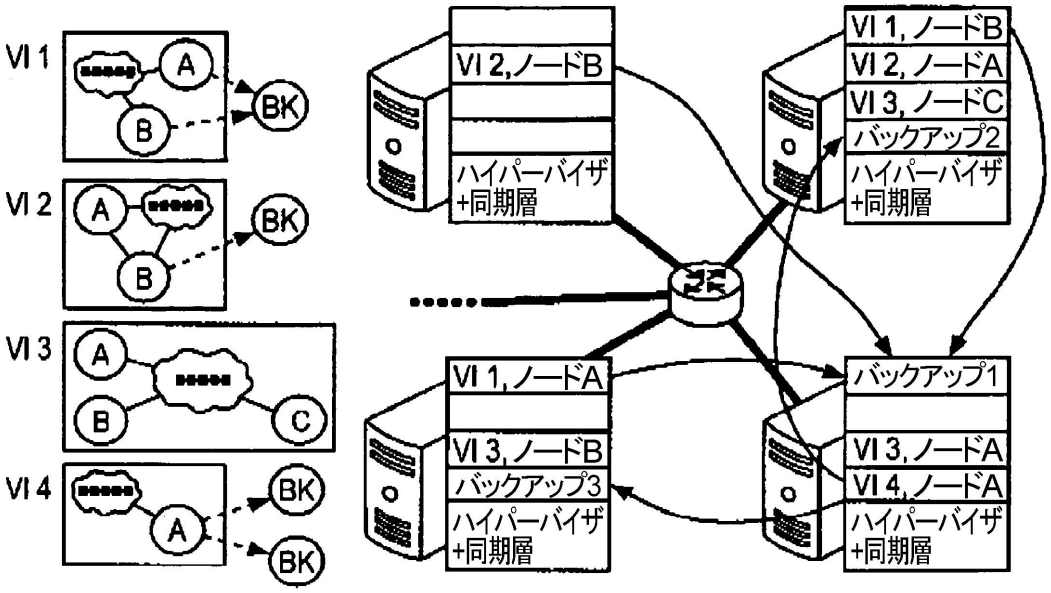


n:kの複製がサポートできるノードの数

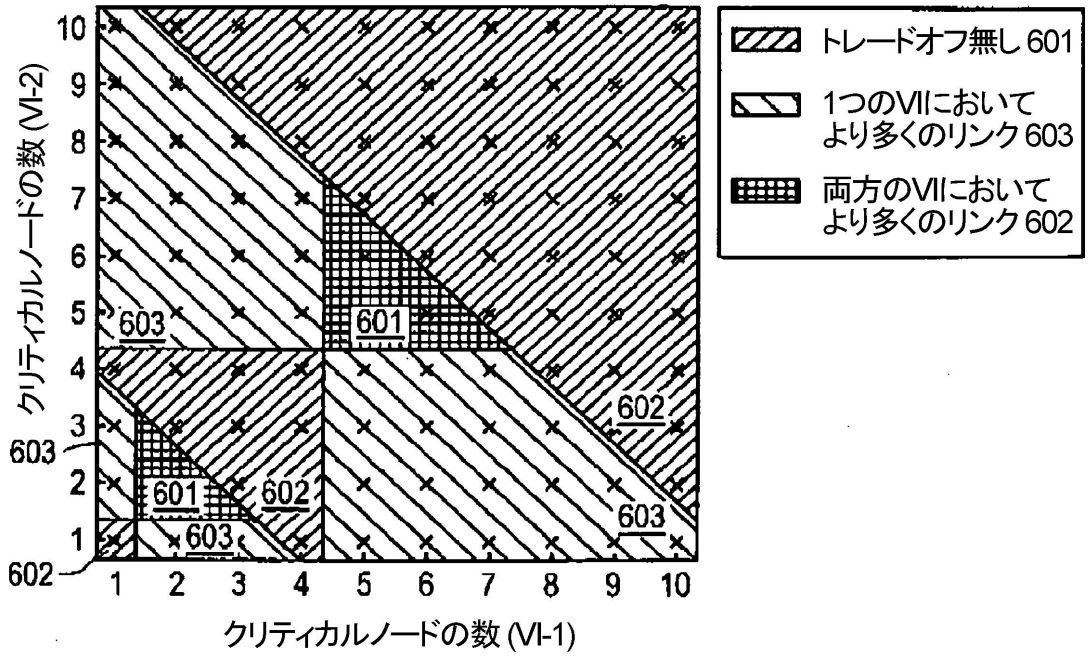
【図４】



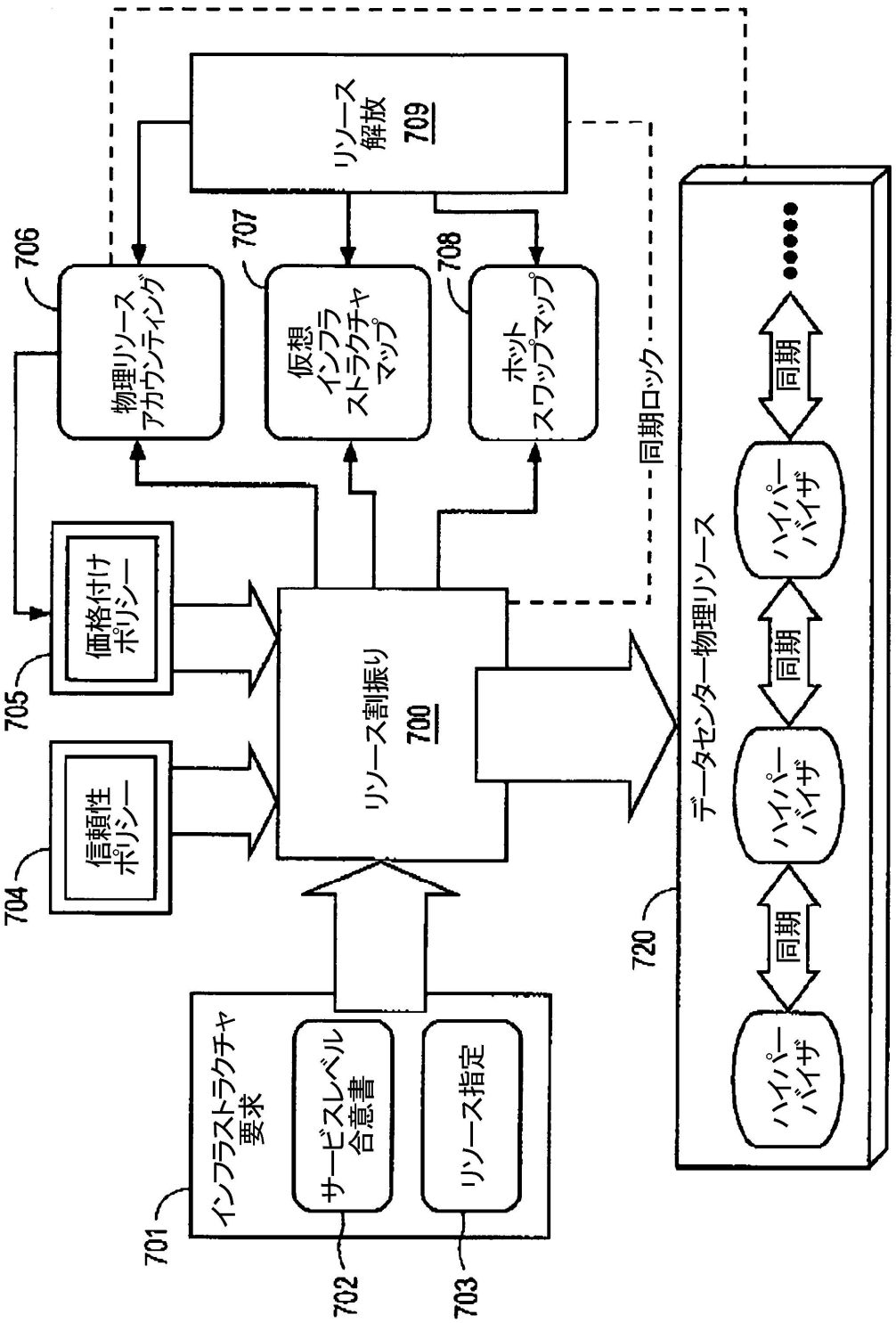
【図５】


VI 1
A
B
BK
VI 2
A
B
BK
VI 3
A
B
C
VI 4
A
BK
BK
VI 2,ノードB
ハイパーバイザ
+同期層
VI 1,ノードB
VI 2,ノードA
VI 3,ノードC
バックアップ2
ハイパーバイザ
+同期層
VI 1,ノードA
VI 3,ノードB
バックアップ3
ハイパーバイザ
+同期層
バックアップ1
VI 3,ノードA
VI 4,ノードA
ハイパーバイザ
+同期層

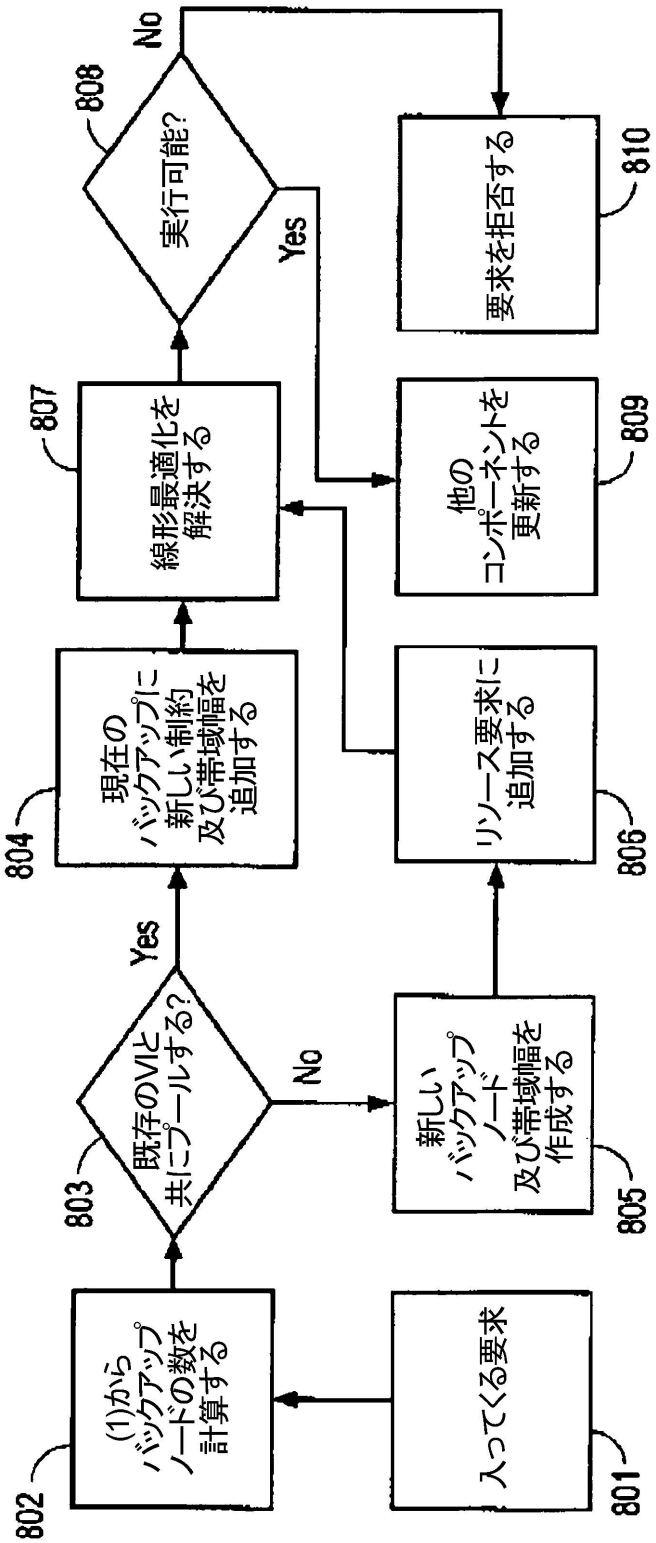
【図６】


トレードオフ無し 601
1つのVIIにおいて
より多くのリンク 603
両方のVIIにおいて
より多くのリンク 602
クリティカルノードの数 (VI-2)
クリティカルノードの数 (VI-1)
601
602
603

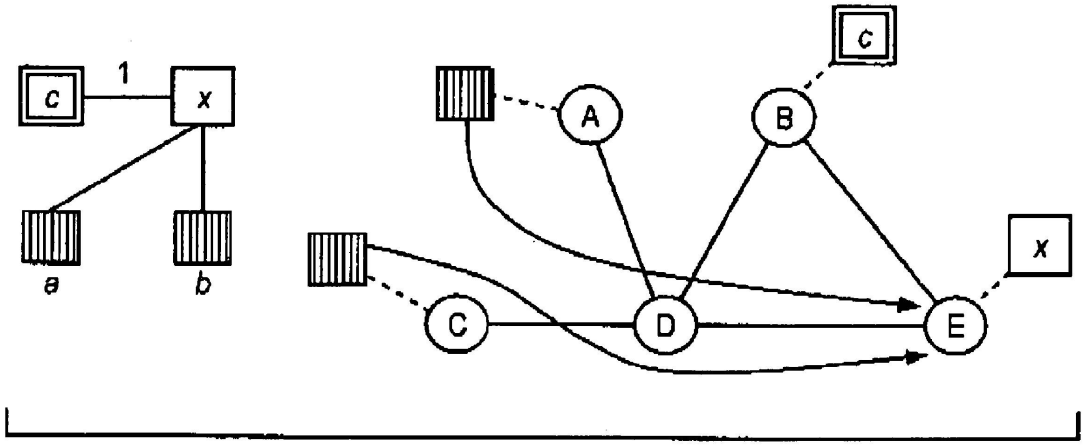
【図７】


インフラストラクチャ要求
サービスレベル合意書
リソース指定
701
702
703
信頼性ポリシー
704
価格付けポリシー
705
物理リソースアカウンティング
706
仮想インフラストラクチャマップ
707
ホットスワップマップ
708
リソース解放
709
リソース割振り
700
同期ロック
データセンター物理リソース
720
ハイパーバイザ
同期
ハイパーバイザ
同期
ハイパーバイザ
同期
ハイパーバイザ

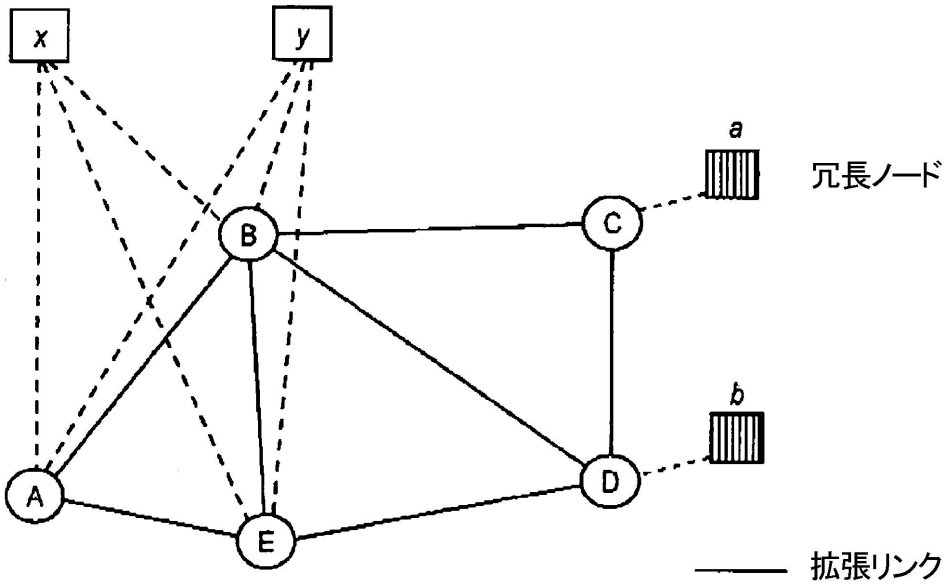
【図８】


801
入ってくる要求
802
(1)から
バックアップ
ノードの数を
計算する
803
既存のVIと
共にプールする?
Yes
No
804
現在の
バックアップに
新しい制約
及び帯域幅を
追加する
805
新しい
バックアップ
ノード
及び帯域幅を
作成する
806
リソース要求に
追加する
807
線形最適化を
解決する
808
実行可能?
Yes
No
809
他の
コンポーネントを
更新する
810
要求を拒否する

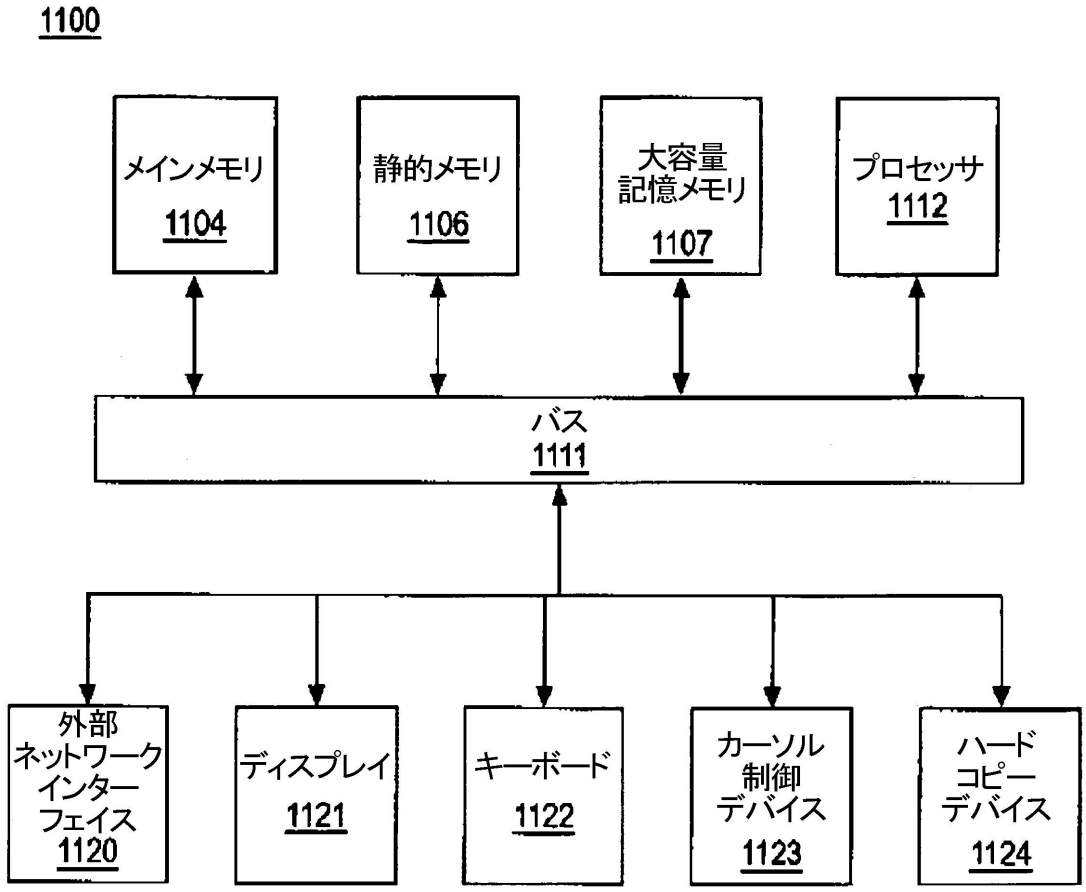
【図９】



【図１０】



【図１１】


1100
メインメモリ
1104
静的メモリ
1106
大容量
記憶メモリ
1107
プロセッサ
1112
バス
1111
外部
ネットワーク
インター
フェイス
1120
ディスプレイ
1121
キーボード
1122
カーソル
制御
デバイス
1123
ハード
コピー
デバイス
1124

フロントページの続き

(72)発明者 ヤオ, ワイーレオン
アメリカ合衆国, カリフォルニア州, マウンテン ビュー, アパートメント ３０２, カリフォルニア アヴェニュー ２６８５
(72)発明者 ウェストファル, セドリック
アメリカ合衆国, カリフォルニア州, サン フランシスコ, ゲレーロ ストリート １０１９
(72)発明者 コザット, ウラス
アメリカ合衆国, カリフォルニア州, サンタ クララ, フローラ ヴィスタ アヴェニュー ナンバー３４９ ３６１２

審査官 三坂 敏夫

(56)参考文献 特開平０６－０２８３３０（ＪＰ，Ａ）
国際公開第２００９／０８１７３６（WO，Ａ１）
米国特許出願公開第２００９／０１３８７５２（ＵＳ，Ａ１）
Song FU, "Failure-Aware Construction and Reconfiguration of Distributed Virtual Machines for High Availability Computing", CCGRID '09. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009., 米国, IEEE, ２００９年 5月２１日, pages:372-379

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０６Ｆ １１／２０