(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第3627005号
(P3627005)

(45)発行日　平成17年3月9日(2005.3.9)　(24)登録日　平成16年12月17日(2004.12.17)

(51)Int.Cl.[7]	F I		
G06F 17/30	G06F	17/30	110C
G06F 9/46	G06F	17/30	170G
G06F 12/00	G06F	9/46	360C
G06F 13/00	G06F	12/00	545A
G06F 15/177	G06F	13/00	540A

請求項の数 23　(全 48 頁)　最終頁に続く

(21)出願番号　特願2000-180514(P2000-180514)
(22)出願日　平成12年6月15日(2000.6.15)
(65)公開番号　特開2001-67377(P2001-67377A)
(43)公開日　平成13年3月16日(2001.3.16)
審査請求日　平成12年6月15日(2000.6.15)
(31)優先権主張番号　09/335273
(32)優先日　平成11年6月17日(1999.6.17)
(33)優先権主張国　米国(US)

(73)特許権者　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
INTERNATIONAL BUSINESS MASCHINES CORPORATION
アメリカ合衆国10504　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(74)代理人　100086243
弁理士　坂口　博
(74)代理人　100091568
弁理士　市位　嘉宏

最終頁に続く

(54)【発明の名称】インターネット環境での負荷の分散とリソース管理を統合したシステム及び方法

(57)【特許請求の範囲】

【請求項１】

サーバの分散ネットワークでオブジェクト・リソースの量を動的に管理するシステムであって、各サーバはオブジェクト・リソースを記憶し提供する容量を有し、前記システムは、

ａ）オブジェクト・リソースに対する需要を予測する手段と、

ｂ）前記ネットワークの前記サーバの、オブジェクト・リソースを記憶しオブジェクトをクライアントにストリーミングする機能を含む容量を監視する手段と、

ｃ）前記オブジェクト・リソースに対する前記予測需要が前記利用できる容量を超えるときは前記オブジェクト・リソースのレプリカを作成し、前記オブジェクト・レプリカを前記ネットワークにあるサーバに一時的に配備する手段と、

ｄ）前記オブジェクト・リソースに対する予測需要が減少したときに一時的に配備されているオブジェクト・レプリカを削除する手段と、

ｅ）前記オブジェクト・レプリカに関連付けられた期限が切れたときに、前記オブジェクトに容量不足条件が存在するときは前記有効期限を延長し、その他の場合は、前記オブジェクト・レプリカを削除する手段と

を含む、システム。

【請求項２】

前記予測手段は、要求されたオブジェクト・リソースについて、需要の密度と量の特徴付けを含めて、異なるクライアントからのリクエストを統合して分析する手段を含む、請求

項１記載のシステム。
【請求項３】
所定コスト基準に従ってオブジェクト・レプリカの配備を管理する手段を含む、請求項１記載のシステム。
【請求項４】
前記レプリカ作成手段ｃ）の前に、要求されたオブジェクトに対する予測需要を利用できる 容量と比較する監査を開始する手段を含み、前記監査は、前記要求されたオブジェクトについて需要が容量を上回ることを示す容量不足条件を予備判定する、請求項３記載のシステム。
【請求項５】
前記削除手段ｄ）の前に、要求されたオブジェクトに対する予測需要を利用できる容量と比較する監査を開始する手段を含み、前記監査は、容量が需要を上回ることを示す容量超過条件を予備判定する、請求項４記載のシステム。
【請求項６】
前記レプリカ作成手段ｃ）は、レプリカを作成する前記オブジェクトを持つソース・サーバと、前記オブジェクト・レプリカを配備するターゲット・サーバとを見つける手段を含む、請求項１記載のシステム。
【請求項７】
前記レプリカ作成手段ｃ）は、ソース・サーバとターゲット・サーバのレプリカ作成接続の提供と、前記オブジェクトのレプリカを作成するストリーミング・レートとを含むレプリカ作成オプションを交渉する手段を含む、請求項６記載のシステム。
【請求項８】
前記レプリカ作成手段ｃ）は、交渉されたオプションに従って前記オブジェクト・レプリカを転送する手段を含む、請求項５記載のシステム。
【請求項９】
前記分散ネットワークの各オブジェクトのステータスを追跡する手段を含み、前記ステータスは、要求されたオブジェクトを記憶するサーバの現在位置とＩＤとを含む、請求項１記載のシステム。
【請求項１０】
前記レプリカ作成手段ｃ）は、前記交渉の間に前記オブジェクト・レプリカを低い品質にダウングレードする手段をさらに含む請求項９記載のシステム。
【請求項１１】
前記オブジェクトが、マルチメディア・オブジェクトである請求項１記載のシステム。
【請求項１２】
サーバの分散ネットワークでオブジェクト・リソースの量を動的に管理する方法であって、各サーバはオブジェクト・リソースを記憶し提供する容量を有し、前記方法は、
ａ）オブジェクト・リソースに対する需要を予測するステップと、
ｂ）前記ネットワークの前記サーバの、オブジェクト・リソースを記憶しオブジェクトをクライアントにストリーミングする機能を含む容量を監視するステップと、
ｃ）前記オブジェクト・リソースに対する前記予測需要が前記利用できる容量を超えるときは前記オブジェクト・リソースのレプリカを作成し、前記オブジェクト・レプリカを前記ネットワークにあるサーバに一時的に配備するステップと、
ｄ）前記オブジェクト・リソースに対する予測需要が減少したときに一時的に配備されているオブジェクト・レプリカを削除するステップと、
ｅ）前記オブジェクト・レプリカに関連付けられた期限が切れたときに、前記オブジェクトに容量不足条件が存在するときは前記有効期限を延長し、その他の場合は、前記オブジェクト・レプリカを削除するステップと
を含む、方法。
【請求項１３】
前記予測ステップは、要求されたオブジェクト・リソースについて、需要の密度と量の特

徴付けを含めて、異なるクライアントからのリクエストを統合して分析するステップを含む、請求項１２記載の方法。
【請求項１４】
所定コスト基準に従ってオブジェクト・レプリカの配備を管理するステップを含む、請求項１２記載の方法。
【請求項１５】
前記レプリカ作成ステップｃ）の前に、要求されたオブジェクトに対する予測需要を利用できる 容量と比較する監査を開始するステップを含み、前記監査は、前記要求されたオブジェクトについて需要が容量を上回ることを示す容量不足条件を予備判定する、請求項１４記載の方法。
【請求項１６】
前記削除ステップｄ）の前に、要求されたオブジェクトに対する予測需要を利用できる容量と比較する監査を開始するステップを含み、前記監査は、容量が需要を上回ることを示す容量超過条件を予備判定する、請求項１５記載の方法。
【請求項１７】
前記レプリカ作成ステップｃ）は、レプリカを作成する前記オブジェクトを持つソース・サーバと、前記オブジェクト・レプリカを配備するターゲット・サーバとを見つけるステップを含む、請求項１２記載の方法。
【請求項１８】
前記レプリカ作成ステップｃ）は、ソース・サーバとターゲット・サーバのレプリカ作成接続の提供と、前記オブジェクトのレプリカを作成するストリーミング・レートとを含むレプリカ作成オプションを交渉するステップを含む、請求項１７記載の方法。
【請求項１９】
前記レプリカ作成ステップｃ）は、交渉されたオプションに従って前記オブジェクト・レプリカを転送するステップを含む、請求項１６記載の方法。
【請求項２０】
前記分散ネットワークの各オブジェクトのステータスを追跡するステップを含み、前記ステータスは、要求されたオブジェクトを記憶するサーバの現在位置とＩＤとを含む、請求項１２記載の方法。
【請求項２１】
前記レプリカ作成ステップｃ）は、前記交渉の間に前記オブジェクト・レプリカを低い品質にダウングレードするステップをさらに含む請求項２０記載の方法。
【請求項２２】
前記オブジェクトが、マルチメディア・オブジェクトである請求項１２記載の方法。
【請求項２３】
請求項１２から２２に記載の方法をコンピュータに実行させるためのプログラムを記録した記録媒体。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、一般的には、広く分散したコンピュータ・ネットワーク環境のリソースに対する需要の監視、管理、分散を行う方法に関し、特に、Ｗｅｂやマルチメディアのサーバの負荷を分散し、複数のＷｅｂ、マルチメディア・サーバにわたってリソースを管理、分散する新規なシステム及び方法に関する。
【０００２】
【従来の技術】
図１は、代表的なコンピュータ・システム（１０）を示す。複数のクライアント（１１０、１１１、１１２）、複数のサーバ（１２０、１２１、１２２）、及び独立したオブジェクトの集合（１３０、１３１、１３２）が含まれる。これらの要素はコンピュータ・ネットワーク環境（１６０）により接続され、クライアント（１１１等）はサーバにリクエス

ト・メッセージ（１４０）を直接送ることができる。このシステムでは、サーバ（１２１等）が、要求元のクライアント（１１１等）にオブジェクトを配信するためストリーミング接続（１５０）を確立することができる。この環境は、ブラウザがクライアントを代表し、Ｗｅｂサーバがサーバを代表し、Ｗｅｂサイトがオブジェクト群を代表し、インターネットがコンピュータ・ネットワーク環境を代表するインターネットで一般的である。周知の通り、クライアントは、ＨＴＴＰプロトコルによりＵＲＬ（Ｕｎｉｖｅｒｓａｌ Ｒｅｓｏｕｒｃｅ Ｌｏｃａｔｏｒ）と呼ばれる位置ＩＤを介してサーバにオブジェクトを要求することができる。ＴＣＰ（Ｔｒａｎｓｍｉｓｓｉｏｎ Ｃｏｎｔｒｏｌ Ｐｒｏｔｏｃｏｌ）は、オブジェクト（Ｗｅｂページやビデオ・ファイル等）をＷｅｂサーバからクライアントにストリーミングできるようにする。

【０００３】

図２は、図１の環境に見られるようなサーバ（１２０等）の要素を詳しく示す。サーバは、メモリ（２１０）、ＣＰＵ（２２０）、ディスク記憶装置（２３０）、ネットワーク帯域幅（２４０）等で構成される一定量のローカル・リソース（２００）を含む。サーバはオブジェクトの集合（１３０）に関連付けられる。この例の集合は４つのオブジェクト（２８１、２８２、２８３、２８４）で構成される。再生時のＶＴＲ操作（一時停止、巻戻し、停止、早送り等）、請求処理、セキュリティ等のクライアントとの相互作用は、サーバのサービス・ロジック・コンポーネント（２５０）により処理される。サーバは、シグナリング・プロトコル（２６１）（ＨＴＴＰ等）によりクライアントからリクエスト（１４０等）を受信できる。クライアント（１１１等）がサーバの集合上のオブジェクト（２８１等）にアクセスするとき、サーバはそのリソース（２００）の一部を対応するストリーミング接続（１５０）に割当てる。リソースは有限なので、受信されるリクエストに対応できるかどうかを判定するためにローカルの受け入れ管理プロセス（２６０）が用いられる。サーバのローカル・リソース（２００）（図２に示すようなディスク記憶装置（ＨＤＤ）、帯域幅（Ｂ）、ＣＰＵサイクル（ＣＰＵ）、メモリ（ＭｅＭ）等）の予約、アクセス、監視、割当て解除のためローカル・リソース管理プロセス（２７０）が用いられる。ネットワーク・ストリーミング・プロセス（２７５）は、ストリーミング・プロトコル（２７１）により、クライアントにコンテンツを配信するため、クライアントとのストリーミング接続（１５０等）を確立／維持する。リソース管理は、（１２０等）どのサーバでも、他のサーバ（１２１等）でのリソース管理から完全に独立している。更に、サーバ上の集合（１３０、１３１等）は相互に独立している。特に、別個のサーバ（１２０、１２１）上の２つの別個の集合（１３０、１３１）に同じオブジェクト（例えばオブジェクトＯ４）のコピー（２８１、２８５）が存在することがあるが、これらのコピー（２８１、２８５）を互いに関連付ける手段は存在しない。

【０００４】

図３に示すように、分散コンピュータ・システム１０（図１）は、オブジェクト・リクエスト・ブローカ（ＯＲＢ）として実現され、オブジェクト・サイトの集合（１３０、１３１、１３２）に関してディレクトリ・サービスを提供し、位置の透過性をクライアント（１１０、１１１、１１２等）まで拡張し、分散したオブジェクト集合（１３０、１３１、１３２）にオブジェクト（例えばメディア・コンテンツ・ファイルＯ４）を要求する、オブジェクト・ディレクトリ・サービス（３００）を採用できる。オブジェクト・ディレクトリ・サービス（３００）は、コンピュータ・ネットワーク環境（１６０）の任意のオブジェクトを見つけるため必要な情報を提供する。採用されるディレクトリ（３１０）は、特にオブジェクトに関連付けられたサーバを追跡する。例えば第１ディレクトリ・エントリは、オブジェクト２８１がサーバ１２０で見つかったことを示し、第２ディレクトリ・エントリはオブジェクト２８５がサーバ１２１で見つかったことを示す。

【０００５】

広く分散したコンテンツやリソースが利用しやすくなっており、この状況を活用することが、次世代のインターネット（インターネット２等）の発展とともに重要になっている。新たに登場したインターネット・プロジェクトは、ブロードバンド・ネットワークの機能

をフルに活用する新世代アプリケーションに向けた先進的ネットワークの構築を目指している。極めて大きい帯域幅と帯域幅予約により、連続デジタル・ビデオ／オーディオ等のリソースを、研究用途からより広い用途に振り向けることができ、現在はまだ不可能な方法でイメージ、オーディオ、ビデオを追加することができる。このように広く分散した環境では、信頼でき、効率のいい、自主規制によるリソース管理が望ましく、また、最も重要なことは、そのような管理の必要性である。

【０００６】

次世代に向けたインターネットの動きに拍車をかけているのは、豊富なマルチメディア・コンテンツの商用利用である。企業が作成するデジタル・ライブラリ、映画業界が作るエンタテインメント、大学が作る対話型教育用プレゼンテーション等がまもなくインターネットで利用できるようになり、新たに幅広い収益源が創り出される。

【０００７】

新しいインターネットは、現在のインターネット・プロバイダより数倍大きい帯域幅に依存する。また、予約と監視の機構を導入することでネットワーク・リソース管理とＱｏＳ（サービス品質）管理を軽減している。しかし、現在まで、複数のメディア接続を集中的に管理し、ワイド・エリア・ネットワークで複数のサーバに及ぶリソースの共有状態を効率よく活用するような機構は登場していない。

【０００８】

こうした新しい用途の商用利用については、主に３つの条件が考えられている。第１に、サービス・プロバイダがサービスの品質を保証し、保証した品質をサポートするため必要なインフラ要素やリソースを予約するため、支払い側のユーザがサービス・プロバイダと契約を結ぶメカニズムが必要である。第２に、リソースの供給は、アーキテクチャ研究時には全く予測できない可能性のあるランダムな需要の変化に対応するに充分でなければならない。第３に、サービス・プロバイダは、動的に再構成可能な分散仮想リソースの消費に関して、効果的なセキュリティ対策、権利とロイヤルティの管理、計算、請求等を行うシステムに、安心して依存できなければならない。

【０００９】

現在のインターネットでは、リソース管理の焦点は、あるとすれば、サーバ・リソースに対する個々の独立したメディア接続の設定と管理に関係する。しかし、コンテンツの複数のプライマリ・ソースをプレゼンテーションに再利用するとき、このアプローチの危うさがはっきりする。複数のコンテンツ・ソースを再利用する際、必要な品質を保ち、使い方や配布を管理するたの方法は２つある。１つは、全てのコンテンツを作成時に１つのサーバ（またはサーバ・クラスタ）にコピーし、必要な場合は、予測需要に従って、最終結果をできるだけ多くのサーバに複製することである。プライマリ・コンテンツ・プロバイダはその場合、アプリオリな市場分析をもと著作権使用料を設定する。その利点としては、配布、セキュリティ、請求等の機能が、コンテンツが分散している場合に比べてかなり容易になる。欠点としては、需要予測が外れた場合、プライマリ、セカンダリ（つまり再利用）いずれのコンテンツのプロバイダも最大の利益が得られない。最後に、最も危険な問題は、このアプローチはリソースのオーバエンジニアリング（過剰品質）につながる一方、過剰なリクエストの脱落を防ぐことができない。しかしこアプローチは現在のインターネットの典型である。現在のマルチメディア・コンテンツは、一般的には、新しいマルチメディア・アプリケーションに比べてメモリを浪費しないからである。

【００１０】

もう１つのアプローチは、作成、配布いずれの場合についても、必要に応じてコンテンツを再構成することである。これにより、コンテンツを一度記憶すれば何回でも必要に応じて使用でき、コンテンツとリソースの使用状態に応じた使用料を設定でき、記憶域を少なくすることができる。ただし、システムは、しばしば雑多になる複数のリソースを動的に管理する必要がある。またこのアプローチでは、セキュリティとリソース・エンジニアリングが悪化する。特定のセグメントが、全く異なる垂直アプリケーションにさえも使用される可能性があるため、そのセグメントに対する需要は全く予測できない。１つのセグメ

ントに対する需要を満足できない場合、複数のアプリケーションが影響を受ける。このアプローチはしかし、将来のインターネットには唯一実際的な方法である。リソースの観点からは最も経済的であり、最大数のユーザに対応できるからである。

【００１１】

従って、主な商用利用条件３つを全て満足する装置及び方法を提供することが特に望まれる。

【００１２】

ＱｏＳドリブンのリソース管理分野に関する出版物や特許は少なくない。それらの大半は、Ｍ．Ｊ．Ｂａｕｇｈｅｒらによる１９９５年2月7日付米国特許第５３８８０９７号、Ｓｙｓｔｅｍ ａｎｄ Ｍｅｔｈｏｄ ｆｏｒ Ｂａｎｄｗｉｄｔｈ Ｒｅｓｅｒｖａｔｉｏｎ ｆｏｒ Ｍｕｌｔｉｍｅｄｉａ Ｔｒａｆｆｉｃ ｉｎ Ｃｏｍｍｕｎｉｃａｔｉｏｎ Ｎｅｔｗｏｒｋｓ及び同Ｍ．Ｊ．Ｂａｕｇｈｅｒらによる１９９６年１２月3日付米国特許第５５８１７０３号、Ｍｅｔｈｏｄ ａｎｄ Ａｐｐａｒａｔｕｓ ｆｏｒ Ｒｅｓｅｒｖｉｎｇ Ｓｙｓｔｅｍ Ｒｅｓｏｕｒｃｅｓ ｔｏ ａｓｓｕｒｅ Ｑｕａｌｉｔｙ ｏｆ Ｓｅｒｖｉｃｅに述べられているように、ネットワークに焦点を当てているか、またはＫ．Ｙ．ＹａｕとＳｉｍｏｎ Ｓ．ＬａｍによるＡｎ Ａｒｃｈｉｔｅｃｔｕｒｅ Ｔｏｗａｒｄｓ Ｅｆｆｉｃｉｅｎｔ ＯＳ Ｓｕｐｐｏｒｔ ｆｏｒ Ｄｉｓｔｒｉｂｕｔｅｄ Ｍｕｌｔｉｍｅｄｉａ（Ｐｒｏｃｅｅｄｉｎｇｓ ｏｆ ＩＳ＆Ｔ／ＳＰＩＥ Ｍｕｌｔｉｍｅｄｉａ Ｃｏｍｐｕｔｉｎｇ ａｎｄ Ｎｅｔｗｏｒｋｉｎｇ Ｃｏｎｆｅｒｅｎｃｅ ’９６、Ｓａｎ Ｊｏｓｅ、ＣＡ、Ｊａｎｕａｒｙ １９９６）に述べられているように、オペレーティング・システムに焦点を当てている。インターネットでマルチメディア・サービスが増殖してまもなく、ＩＰネットワークは、シンプル且つ最善の配信サービスを提供できるが、ＩＰプロトコルは、マルチメディア・ストリーミング、仮想現実関係のアプリケーション、分散スーパーコンピューティング等の新しいリアルタイム・アプリケーションには向いていないことがわかってきた。その結果、ＲＳＶＰ（（Ｒｅｓｏｕｒｃｅ Ｒｅｓｅｒｖａｔｉｏｎ Ｓｅｔｕｐ Ｐｒｏｔｏｃｏｌ）、Ｉａｎ ＦｏｓｔｅｒとＣａｒｌ Ｋｅｓｓｅｌｍａｎが編集したＴｈｅ Ｇｒｉｄ：Ｂｌｕｅｐｒｉｎｔ ｆｏｒ ａ Ｎｅｗ Ｃｏｍｐｕｔｉｎｇ Ｉｎｆｒａｓｔｒｕｃｔｕｒｅ、Ｃｈａｐｔｅｒ １９、ｐｐ．３７９－５０３、Ｍｏｒｇａｎ Ｋａｕｆｆｍａｎ Ｐｕｂｌｉｓｈｅｒｓ、１９９９を参照）、ＲＴＰ（Ｒｅａｌ Ｔｉｍｅ Ｔｒａｎｓｐｏｒｔ Ｐｒｏｔｏｃｏｌ）、ＲＴＣＰ（Ｒｅａｌ Ｔｉｍｅ Ｔｒａｎｓｐｏｒｔ Ｃｏｎｔｒｏｌ Ｐｒｏｔｏｃｏｌ）等の新しいネットワーク・プロトコルが開発され（例えば、Ｗｉｌｌｉａｍ StallingsのＨｉｇｈ－Ｓｐｅｅｄ Ｎｅｔｗｏｒｋｓ：ＴＣＰ／ＩＰ ａｎｄ ＡＴＭ Ｄｅｓｉｇｎ Ｐｒｉｎｃｉｐｌｅｓ、Ｐｒｅｎｔｉｃｅ Ｈａｌｌ、１９９７、Ｉ．Ｂｕｓｓｅ、Ｂ．Ｄｅｆｆｎｅｒ、及びＨ．ＳｃｈｕｌｚｒｉｎｎｅによるＤｙｎａｍｉｃ ＱｏＳ Ｃｏｎｔｒｏｌ ｏｆ Ｍｕｌｔｉｍｅｄｉａ Ａｐｐｌｉｃａｔｉｏｎｓ ｂａｓｅｄ ｏｎ ＲＴＰ、Ｃｏｍｐｕｔｅｒ Ｃｏｍｍｕｎｉｃａｔｉｏｎｓ、Ｊａｎｕａｒｙ １９９６を参照）、帯域幅、待ち時間等のネットワークＱｏＳパラメータをアプリケーションから要求し交渉できるようになっているが、こうしたプロトコルを現在のインターネットに展開する試みは成功していない。その理由は第１に、ＲＳＶＰ以外のルータやサーバのシステム・ソフトウェアを全てアップグレードする必要があるからである。第２に、現在のインターネットにＲＳＶＰを配備したとしても、帯域幅やコンピューティング・リソースがかなり制限されるため、これがリアルタイム・アプリケーションを軌道に乗せる上で障害になる。現在のインターネットはバックボーン上に構築されたもので、比較的障害のないＴ３（４５メガビット毎秒）で国際通信が可能であるが、グラフィック・ページの増加と、ストリーミング・オーディオ／ビデオのアプリケーションが、こうしたリソースをかなりのスピードで奪ってしまった。更にユーザの増加率は、新しく構築されるネットワーク・リソースのそれよりかなり高い。

【００１３】

米科学財団とＭＣＩは、インターネット・コミュニティの新しいニーズに応えて、ｖＢＮＳ（ｖｅｒｙ－ｈｉｇｈ－ｐｅｒｆｏｒｍａｎｃｅ Ｂａｃｋｂｏｎｅ Ｎｅｔｗｏｒｋ Ｓｅｒｖｉｃｅ）と呼ばれる新しいネットワークを構築している。これは全米規模のネットワークであり、２つの財団のバックボーンにもなっている（インターネット２と呼ばれる大学主導のプロジェクトと連邦研究機関による新世代インターネット）。ｖＢＮＳでは、接続している機関のほとんどで６億２２００万ビット毎秒（ＯＣ１２）の速度が得られる。２０００年までに２．４ギガビット毎秒（２４００メガビット毎秒）での運用が予定されている。

【００１４】

ｖＢＮＳシステムは、ＲＳＶＰプロトコルを利用して２種類のサービスをサポートする。予約帯域幅サービス、つまり帯域幅をコミットしたサービスと、従来の最善のＩＰサービス（Ｃｈｕｃｋ Ｓｏｎｇ、Ｌａｕｒａ Ｃｕｎｎｉｎｇｈａｍ、Ｒｉｃｋ ＷｉｌｄｅｒによるＱｕａｌｉｔｙ ｏｆ Ｓｅｒｖｉｃｅ Ｄｅｖｅｌｏｐｍｅｎｔ ｉｎ ｔｈｅ ｖＢＮＳ、ＭＣＩ Ｃｏｍｍｕｎｉｃａｔｉｏｎｓ Ｃｏｒｐｏｒａｔｉｏｎ、ＵＲＬはｈｔｔｐ：／／ｗｗｗ．ｖｂｎｓ．ｎｅｔ／ｐｒｅｓｅｎｔａｔｉｏｎｓ／ｐａｐｅｒｓ／ＱｏＳＤｅｖ／ｉｅｅｅｇｏｓ／ｈｔｍ等を参照）である。しかし、ｖＢＮＳのネットワーク層でのリソース管理は、オペレーティング・システム層から分離して行われ、アプリケーションのニーズから、また記憶域やコンピューティング・リソース等の末端のリソースの可用性から分離している。

【００１５】

遠隔手術、ロボット工学、遠隔計装（ｔｅｌｅ－ｉｎｓｔｒｕｍｅｎｔａｔｉｏｎ）、緊急時自動応答（ａｕｔｏｍａｔｅｄ ｃｒｉｓｉｓ ｒｅｓｐｏｎｓｅ）、衛星データのデジタル・ライブラリ、マルチメディアをサポートするＷｅｂサイトを介した遠距離学習、拡張オーディオ／ビデオ等、新世代の高性能アプリケーションが登場しているが、こうした高性能アプリケーションとその連続的メディア・フローに対応するには、ネットワーク容量の増加や予約だけでは充分ではない。これらの新しいアプリケーションには、エンド・ツー・エンドのリソース予約と受け入れ管理、及び次のような分散機能の調整が必要である。ａ）エンドシステムでのリソース・スケジューリング（ＣＰＵ、ディスク等）、ｂ）ネットワークのパケット・スケジューリングとフロー制御、ｃ）エンド・ツー・エンドの配信サービス品質の監視。サービス品質は、設定可能、予測可能、及びエンドシステム・デバイス、通信サブシステム、ネットワークを含めたシステム全体で維持できることが重要である。更に、分散システム・アーキテクチャのエンド・ツー・エンド要素は全て、求められるアプリケーション・レベルの動作を達成するために一致協力する必要がある。

【００１６】

現在まで、エンド・ツー・エンドのサービス品質サポートを開発するため多大な労力が注ぎ込まれている。例えば、ＩＢＭの欧州ネットワーク・センターのＨｅｉＰｒｏｊｅｃｔで開発され、Ｃ．Ｖｏｌｇ、Ｌ．Ｗｏｌｆ、Ｒ．Ｈｅｒｒｔｗｉｃｈ、Ｈ．ＷｉｔｔｉｇによるＨｅｉＲＡＴ－Ｑｕａｌｉｔｙ ｏｆ Ｓｅｒｖｉｃｅ Ｍａｎａｇｅｍｅｎｔ ｆｏｒ Ｄｉｓｔｒｉｂｕｔｅｄ Ｍｕｌｔｉｍｅｄｉａ Ｓｙｓｔｅｍｓ（Ｍｕｌｔｉｍｅｄｉａ Ｓｙｓｔｅｍｓ Ｊｏｕｒｎａｌ、１９９６）で説明されているＨｅｉｄｅｌｂｅｒｇ ＱｏＳモデル、コロンビア大学のＣＯＭＥＴグループが開発し、Ａ．Ｔ．Ｃａｍｐｂｅｌｌ、Ａ．Ａ．Ｌａｚａｒ、Ｈ．Ｓｃｈｕｌｚｉｎｎｅ、Ｒ．ＳｔａｄｌｅｒによるＢｕｉｌｄｉｎｇ Ｏｐｅｎ Ｐｒｏｇｒａｍｍａｂｌｅ Ｍｕｌｔｉｍｅｄｉａ Ｎｅｔｗｏｒｋｓ（Ｃｏｍｐｕｔｅｒ Ｃｏｍｍｕｎｉｃａｔｉｏｎｓ Ｊｏｕｒｎａｌ、Ｖｏｌ．２１、Ｎｏ．８、ｐｐ．７５８－７７０、Ｊｕｎｅ １９９８）に説明されているようなＸＲＭ（Ｅｘｔｅｎｄｅ Ｉｎｔｅｇｒａｔｅｄ Ｒｅｆｅｒｅｎｃｅ Ｍｏｄｅｌ）、ペンシルヴァニア大学の学際的研究プロジェクトで開発され、Ｋ．ＮａｈｒｓｔｅｄｔとＪ．ＳｍｉｔｈによるＤｅｓｉｇｎ、Ｉｍｐｌｅｍｅｎｔａｔｉｏｎ ａｎｄ Ｅｘｐｅｒｉｅｎｃｅｓ ｏｆ ｔｈｅ ＯＭＥＧＡ Ｅｎｄ－Ｐｏｉｎｔ Ａｒｃ

hitecture（Technical Report（MS－CIS－95－22）、University of Pennsylvania、May 1995）に説明されているようなOMEGAエンドポイント・アーキテクチャ、IETF（Internet Engineering Task Force）の成果で、Y. BernetらによるA Framework for End－to－End QoS Combining RSVP／Intserv and Differentiated Services（Internet Draft、IETF、March 1998）に説明されているようなイン・サーブ・アーキテクチャ（in－serv Architecture）、A. Campbellにより開発され、エンド・ツー・エンドQoSの要件を扱う統合フレームワークを提供し、Andrew T. CampbellによるA Quality of Service Architecture（PhD thesis、Lancaster University、January 1996）に説明されているような、サービス品質アーキテクチャQoS－A等がある。前記のQoS文献を分析している他の参考書として、C. Aurrecoechea、A. T. Campbell、L. HauwによるA Survey of QoS Architectures（ACM／Springer Verlag、Multimedia Systems Journal、Special Issue on QoS Architecture、Vol. 6、No. 3、pp. 138－151、May 1998）がある。

【0017】

SRI Internationalは、充分な調査研究活動の後、ERDoS（End－to－End Resource Management of Distributed Systems）を開発している。ERDoSは、ERDoS QoS Architecture（Technical Report、SRI International、May 1998）に述べられているように、分散システムのリソース管理をエンド・ツー・エンドで、適応的且つスケーラブルに行えるようにする。拡張可能なRSL（Resource Specification Language）とリソース管理アーキテクチャが、Globusメタコンピューティング・ツールキットに実装されており、K. CzajkowskiらによるA Resource Management Architecture for Metacomputing Systems（Proc. IPPS／SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing、1998）、及びI. Foster、C. KesselmanによるThe Globus Product：A Status Report（Proc. IPPS／SPDP '98 Heterogeneous Computing Workshop、pp. 4－18、1998）に述べられているように、様々なリソース管理方式を実装するため使用される。

【0018】

前記の参考書に述べられているアーキテクチャは、エンド・ツー・エンドのリソースの予約と管理を対象にしているが、一般的には、地理的に限定され、遅延やエラーを制限し、帯域幅の需要に応えるネットワーク・サブシステムと、実行時のQoSを保証できるオペレーティング・システムを想定している。しかし、次世代インターネットは、ネットワークのネットワークとしてだけでなく、まず第1に分散システムのシステムとして捉えるべきである。このパラダイムでは、通信リソースだけでなく、コンピューティング・サーバや記憶サーバも多数のユーザによって共有される。

【0019】

従って、前記のアーキテクチャは、個々のコンテンツやコンピューティング・リソースに対するリクエスト・アクティビティの機能としてシステム・リソース全体の管理を調整するものではなく、特定のサービスに予め割当てられたリソースを扱うものである。従って、サービス品質は、サービスに対するリクエストの増加に対応していくうち、増加量が設定された限度を超える結果低下するほかない。前記のアーキテクチャは、アプリケーショ

ンにより要求されるようなＱｏＳを提供することに焦点を当てており、特定のサービスに対するユーザ・リクエスト間の共通性によりリソースを統合する可能性を活かしていない。

【００２０】

例えば、特定のマルチメディア・コンテンツの利用履歴について、短い時間間隔でのリクエストの集中、リクエスト元アドレスの近接性等の共通性を発見することが望ましい。また、先に示したアーキテクチャでは、個々のサービスについても、関連サービスのグループについても、個々のクライアントにとってのサービス・コストを計算するため、リソースの消費を動的に監視し記録することはできない。

【００２１】

【発明が解決しようとする課題】

従って、適応型リソース管理機能を提供でき、次世代のインターネットに適した環境のニーズに合わせて、システム容量をオンデマンドで調整できる機構を提供することが特に望まれる。

【００２２】

また、容量調整機構を、広く分散したメディア・サーバの集合に対して負荷の分散を管理しドライブする機能と統合することのできる機構を提供することが望まれる。

【００２３】

【課題を解決するための手段】

本発明は、将来のコンピュータ・ネットワーク環境のユニークな特性を活かし、マルチメディア・コンテンツとリソースを管理するシステム及び方法を提供する。

【００２４】

特に、本発明の目的は、インターネット／Ｗｅｂ（Ｗｏｒｌｄ　Ｗｉｄｅ　Ｗｅｂ）でリソースの分散を管理し、リソースを共有、プールし、クライアントとサーバの間に、マルチメディア・オブジェクトに対するリクエストの管理とサーバへの分散／配備を行う中間管理ノードを実装する他、所定基準に従ってオブジェクトのサーバへの配備を管理するシステム及び方法を提供することである。

【００２５】

本発明の他の目的は、インターネット／Ｗｅｂ環境でリソースの分散、共有、プールを管理する装置及び方法を提供することであり、これには、ａ）オブジェクトに関連付けられたレプリカの数を管理し、ｂ）これらレプリカの配備を管理する、ことにより、（マルチメディア）Ｗｅｂオブジェクトに対する予測需要を、Ｗｅｂサーバ上の利用できる容量に合わせ、マルチメディア・オブジェクトに対する需要とマルチメディアＷｅｂサーバの容量の両方を調整する処理が含まれる。

【００２６】

従って、好適実施例に従い、分散コンピュータ環境で負荷の分散とリソース管理の機能を統合した自主規制システムが提供される。システムは、予測需要を利用できる容量に合わせ、この目的を追及するため、所定基準に従って需要と容量を調整する機構が提供される。

【００２７】

本発明の他の目的は、インターネット／Ｗｅｂ環境でマルチメディア・コンテンツを提供できるリソースの分散、共有、プールを、ユーザにとってメリットがあり、信頼でき、且つシームレスな形で管理するシステム及び方法を提供することである。

【００２８】

本発明の原理に従って、クライアント（Ｗｅｂブラウザ等）とサーバ（メディア／Ｗｅｂサーバ等）の間に、リクエストの分散とサーバへの配備を管理し、コンテンツのサーバへの配備を管理する中間管理ノード（コントローラ）が提供される。コントローラは、リクエストを受信し、所定基準に従ってリクエストの配備を調整する媒介として働く。コントローラはそのため、クライアントを代表してサーバへのリクエストの配備の活用、交渉、推奨を行う。システムは、コントローラにより、メディア／Ｗｅｂサーバに分散したオブ

ジェクトの集合のリソース管理を行う。リソース管理は、本発明の文脈では、マルチメディア・コンテンツをクライアントに効率よく提供するため必要なリソースの予約、構成、管理、例外処理、及び解放を指し示す。特にコントローラは、（１つまたは複数のサーバに及ぶ）オブジェクトに対する総予測需要をサーバの利用できる容量と合わせようとする。そのため、中間管理ノードに示されるリクエスト・ストリームの総計を分析することで、予測需要統計が生成され、サーバとコントローラ・ノード間の特別なプロトコルにより、利用可能なおおよその容量が予測される。コントローラは、容量調整機構により、サーバ上のオブジェクトの配備と数を動的に管理する。

【００２９】

本発明はそのため、更に２つの補助的なツールを用いる。利用しやすい予備の共有容量を提供するグローバル・サーバと、需要と容量の条件に対応し、スケーラブルで再配置可能なリソースとしてのマルチメディア・オブジェクトのモデルになる一時的レプリカである。これらはともに、システム全体の容量を一時的に高め、特定のマルチメディア・オブジェクトに関連付けられた予測需要に合わせることで、マルチメディア・サーバを補助するため使用できるシステムを提供する。これら補助的ツールはまた、需要と容量の間の制約に応じてインターネット環境のレプリカの配備と数を動的に管理する装置及び方法を提供するために使用する。従って、本発明のシステムは、需要と容量を監視し、グローバル・サーバとの間で一時的レプリカをいつ追加しいつ削除するか、つまりいつ容量を調整するかを判定する効率的方法を提供する。特に、同じオブジェクトに対する後のリクエストの処理中に利用可能なレプリカを見つける可能性を大きくするツールとして、オンデマンドでレプリカを作成する。

【００３０】

ここで重要なことは、本発明は、前記の事柄を達成しながら、サーバのリソースの管理に関してサーバの自主性を保護することである。リソース管理システムは、リソース管理機能（受け入れ管理、リソース予約、リソース測定、リソース・スケジューリング等）が各サーバでローカルに実装され、コントローラに集約されないという点で分散型である。コントローラは、サーバとそのリソースを直接管理することはなく、管理に関する推奨内容をサーバに送るエージェントの役を務める。これは、システムのリソースとサーバの状態に関して、コントローラに厳重な監視要件を課すことなく達成される。コントローラは、サーバとコントローラ間のシグナリング・プロトコルにより、運用時、耐障害性を保ちながらリソース管理状態を維持することができる。システムは、シグナリングのオーバヘッドと状態維持のオーバヘッドのバランスを考慮して働く。

【００３１】

【発明の実施の形態】

図４は、本発明の好適実施例に従った分散コンピュータ・システム１００を示し、クライアント（１１０、１１１、１１２等）、サーバ（１２０１、１２１１、１２２１）、オブジェクトの集合（１３０、１３１、１３２）、及びクライアントからのオブジェクト・リクエスト（５００）を含む。図４に示す通り、分散コンピュータ・システムは別に、クライアントからのリクエストを処理する中間コントローラ（５２０）を含む。コントローラ（５２０）は、特にクライアント（１１１等）からのリクエスト（５０１等）を、以下に詳述する所定基準に従ってサーバ（１２１１等）に配備する。例えば、好適実施例では、コントローラは、分散したオブジェクトの集合（１３０、１３１、１３２）に対するクライアントのリクエストの配備に関して、負荷のバランスとフォールト・トレランスを導入するため使用する。更に、後述するように、中間コントローラは、マルチメディア・オブジェクト自体のサーバへの配備を所定基準に従って管理する。

【００３２】

詳しくは後述するが、中間コントローラ（５２０）の実装、特にＯＲＢ等のオブジェクト・ディレクトリ・サービスの実装により、システム１００をサーバの集合に対して、オブジェクトの分散集合として特徴付けることができる。従って、従来のシステム１０とは異なり、独立したサーバ（１２０１、１２１１、１２２１）それぞれのオブジェクト（１３

０、１３１、１３２）の様々な集合は、ここで統合され、オブジェクトと需要と容量の条件に従う、スケーラブルで再配置可能なリソースとしてのマルチメディア・オブジェクトのモデルとなるオブジェクト・レプリカとの分散集合（１３０、１３１、１３２）になる。例えば、図４は、サーバ（１２０１）の集合（１３０）にあるレプリカ（２８１）と、サーバ（１２１１）の集合（１３１）にあるもう１つのレプリカ（２８５）とを持つオブジェクトＯ４に関連付けられたオブジェクト・レプリカ（２８１、２８５）を示す。後述する通り、サーバは、永続的（オブジェクト）レプリカを維持するローカル・サーバとみなすことができ、一時的レプリカを維持するグローバル・サーバとみなすこともできる。グローバル・サーバは、レプリカが作成された（一時的な）オブジェクトを維持する上で利用しやすい予備の共有容量を提供するための専用サーバである。

【００３３】

本発明に従って、コントローラ（５２０）に示されるクライアント・リクエストの時間シーケンスを、ここではリクエスト・ストリームまたは需要と呼ぶ。正常なクライアント・リクエスト（１４０等）の結果、ストリーミング接続（１５０等）（ここではストリームとも呼ぶ）が得られる。あるマルチメディア・オブジェクトに対してリクエストがあるとき、利用できるリソースがあれば、サーバによって利用できるようにすることのできる同時ストリームの数を、ここではマルチメディア・サーバの利用できる容量と呼ぶ。更に、コントローラ（５２０）から見た場合、システム全体で同時にサポートできる（要求されたマルチメディア・オブジェクトの）ストリームの数を、ここでは利用できるシステム容量と呼ぶ。特に、好適実施例の場合、新しいインターネット２に想定されているようなワイド・エリア・ネットワーク分散環境で、利用できるおおよその容量を効率よく予測するため、統計的測定値を使用する。

【００３４】

Ｈ３２３、ＲＴＳＰ（Ｒｅａｌ Ｔｉｍｅ Ｓｔｒｅａｍｉｎｇ Ｐｒｏｔｏｃｏｌ）等のＷｅｂ上のマルチメディア・ストリーミング・データを管理する規格は、すでに作成／実装され、目的のストリーミング機能が提供されている。例えばＨ３２３は、小さいグループ間のビデオ会議を目的に設計され、ＲＴＳＰは、大きいグループ向けにオーディオ／ビジュアル・データを効率よくブロードキャストするよう設計されている。いずれの規格も、リアルタイム・プロパティを持つデータの配信を管理するため、クライアント／サーバのアプリケーション・レベルのプロトコルを記述している。例えばＲＴＳＰは、オーディオ、ビデオ等の連続的メディアの１つまたは数個の時間同期ストリームを確立／管理し、ＵＤＰ、マルチキャストＵＤＰ、ＴＣＰ、ＲＴＰ（Ｒｅａｌ Ｔｉｍｅ Ｐｒｏｔｏｃｏｌ）等のトランスポート・プロトコルを使用して連続ストリームを配信する。

【００３５】

図５は、（マルチメディア・オブジェクトに対する）リクエストのサーバへの分散と配備を管理するため、またマルチメディア・オブジェクト自体のサーバへの配備を管理するため実装される中間コントローラ（５２０）の詳細ブロック図である。図５に示す通り、クライアントから、固有オブジェクトＩＤを含むリクエスト（６０１、６０２、６０３、６０４）を受信し、これらのリクエストを配備モジュール（６１０）に送るためリクエスト処理モジュール（６００）が使用される。配備モジュール（６１０）は、各リクエストに配備ポリシ（６１５）を適用し、リクエストに関する一時的配備照会（６２０）のセットを生成する。特に配備モジュール（６１０）は、レプリカ・ディレクトリ・サービス（６６５）（図６に関して説明するレプリカ・ディレクトリ（６６６）を維持する）とサーバ・ディレクトリ・サービス（６５５）（図７に関して説明するサーバ・ディレクトリ６５６を維持する）の両方とインタフェースをとり、一時的配備（６２０）を生成する。つまり配備モジュール（６１０）、レプリカ・ディレクトリ・サービス（６６５）、及びレプリカ・ディレクトリ（６６６）は連携して受信されたリクエストのオブジェクトＩＤに関連付けられた全てのレプリカを見つける。更に、配備モジュール（６１０）、サーバ・ディレクトリ・サービス（６５５）、及びサーバ・ディレクトリ（６５６）は連携して、後述するように、新しい配備照会（６２０）を考慮する（レプリカを保持する）位置の”意

向”を判定する。

【００３６】

図６は、レプリカ・ディレクトリ・サービス（６６５）により維持され、本発明のシステム１００により実装されるレプリカ・ディレクトリに関連付けられるスキーマ及びデータを含むレプリカ・ディレクトリの例（６６６）を示す。分散集合内のオブジェクトを一意に識別するため、分散集合の各オブジェクト（Ｏ４等）にオブジェクトＩＤ（４２０等）が割当てられる。本発明に従って、元のクライアント・リクエストを予め、クライアントからの曖昧なリクエストを一意に識別可能なオブジェクトＩＤに変換可能な補助システム（図示せず）により処理することができる。オブジェクトＩＤ毎に、分散集合内にレプリカを使用できる。例えば図６は、オブジェクトＩＤが２つ（４２０、４４０）の場合を示している。第１オブジェクトＩＤ（４２０）は現在２つのレプリカ（４２１、４２２）に関連付けられ、第２オブジェクトＩＤ（４４０）は１つのレプリカ（４４１）にのみ関連付けられている。同じオブジェクトＩＤに関連するレプリカは、別々のサーバに分散される。例えばオブジェクトＩＤ（４２０）のレプリカ（４２１、４２２等）はそれぞれ別々のサーバ（１２１１、１２２１）に位置する。各オブジェクト・レプリカには、レプリカの一時性の程度を示す存続時間タイムスタンプも関連付けられる。後述するように、存続時間が終了すると、グローバル・サーバでオブジェクト・レプリカの存続を延長するリクエストが出される。

【００３７】

レプリカ・ディレクトリ（６６６）は障害に耐える必要があり、永続的レプリカに関するデータとこれに関連するローカル・サーバをチェックポイントとして使用しても、データが事実上、失われる恐れはなく問題はない。ただし一時的レプリカに関するデータだけは揮発性のデータである。一時的レプリカに関するデータの喪失から復帰するとき、コントローラ（５２０）は単にグローバル・サーバに照会し、現在まだ期限の切れていない一時的レプリカのリストをチェックするだけである。前記のサーバ・ディレクトリ（６５６）（図７）では、コントローラが全てのグローバル・サーバのＩＤを見つけることができる。コントローラは、コントローラのドメイン内の各グローバル・サーバに照会することで、レプリカ・ディレクトリの再書込みがオンデマンドでできる。データが失われる恐れをなくすために、グローバル・サーバのリストをチェックポイントにしてもよい。当業者には明らかなように、レプリカ・ディレクトリ（６６６）の再書込みの後にアクセスされていないレプリカは、コントローラによってそのグローバル・サーバに対してリクエストが出されなくなるため、使用率が次第に減っていくことになる。

【００３８】

更に、図１５に関して説明するが、レプリカ・ディレクトリは、リクエストに関連付けられた統計をオブジェクトＩＤ毎に維持し、これには予測需要ｄ、リクエストに関連付けられた優勢な地域を示す優勢地域インディケータｇ、及び需要統計またはレートｒが含まれる。更に、各レプリカに存続時間タイムスタンプが関連付けられる。タイムスタンプの期限が切れると、一時的レプリカを現在所有しているグローバル・サーバはそのコントローラ（つまりこのレプリカを配備したコントローラ）に更新を要求する。この時点でコントローラは、レプリカの更新を拒否することでレプリカを削除するか、またはレプリカの存続時間を延長することでレプリカを更新する（従ってこの新しいレプリカでデータベースの再書込みを行う）ことができる。コントローラが更新を拒否した場合、一時的レプリカは、そのグローバル・サーバにより削除されて終わる。

【００３９】

図７は、サーバ・ディレクトリ・サービス（６６５）により維持され、スキーマとデータが関連付けられたサーバ・ディレクトリの例（６５６）を示す。分散コンピュータ環境（１６０）の各サーバにサーバＩＤ（１２１１等）が割当てられる。サーバＩＤは固定とみなされ、変更されない。サーバＩＤの例として、サーバの固定ＩＰアドレスやホスト名等がある（２０９．０９．９．１２７（１２１１）、１２８．０．０．１（１２２１）等）。サーバＩＤ毎に、サーバの容量評価と呼ぶ特別なフィールドにより、サーバ全体の容量

を評価する。つまり容量評価は、コントローラが、事実上異なるリソースを持つサーバを区別するために使用する。好適実施例の場合は２段階評価を行う。好適実施例は、ＨＩＧＨ（スーパーコンピュータ／メインフレーム等）とＬＯＷ（ＮＴクラスのサーバ等）の２つの容量評価を区別する。ただし当業者には明らかなように、代わりに他の評価方式を使用してもよい。容量評価は、サーバ固有のパラメータであり、初期化時に設定する。例えばサーバのデフォルト評価がＬＯＷ容量とする。本発明に従って、コントローラは、サーバの実際の利用できる容量を追跡する必要なしに、容量評価により容量がＨＩＧＨとＬＯＷのサーバを区別することができる。レプリカが作成された（一時的）オブジェクトの受信を継続するグローバル・サーバは、通常はＨＩＧＨ容量サーバである。

【００４０】

また、サーバＩＤ毎、そのサーバの最後にレポートされた使用率／意向状態を記憶するため特別なフィールドを使用する（例えばサーバ（１２１１）は赤、サーバ（１２２１）は緑である）。またサーバ毎に、コントローラにより受信された最後の使用率／意向状態レポートの時刻も記憶する。例えば図７でサーバ（１２１１）にはタイムスタンプｔ１が関連付けられ、サーバ（１２２１）にはタイムスタンプｔ２が関連付けられる。最後に、サーバがグローバル・サーバかローカル・サーバかをフィールドで示す。例えばサーバ（１２１１）はコントローラによりローカル・サーバとして認識され、サーバ（１２２１）はグローバル・サーバとして認識される。サーバはグローバル、ローカル両方であり得、その場合、２つのエントリを使用する。１つのエントリで仮想ローカル・サーバを、もう１つのエントリで仮想グローバル・サーバを記述する。

【００４１】

図５に戻る。コントローラ（５２０）は他に、一時的配備（６２０）を選択し、これら一時的配備に関連付けられたサーバに照会する方式を実行する交渉モジュール（６３０）を含む。探索ポリシ（６３５）と交渉ポリシ（６３６）に従って照会方式を作成する。交渉ポリシは複数の一時的配備を洗練し、コスト等の一定の基準による選択を可能にするため実装する。様々なポリシ（６１５、６３５、６３６等）を記憶、ロードし、ここで説明しているコントローラ・アルゴリズムをカスタマイズするためポリシ・バンク（６４０）を使用する。これらのポリシは初期化時またはオンデマンドでロードできる。

【００４２】

図５に示す通り、コントローラ（５２０）は他に、サーバを監視するグローバル・リソース監視モジュール（６５０）を備える。サーバ・ディレクトリ・サービス（６５５）によりリソース監視データが提供される。レプリカのライフサイクルを管理するため、特にレプリカの作成、破壊、または移動を決定するため、ヒューリスティクスを適用するレプリカ作成管理モジュール（６６０）を使用する。レプリカ・ディレクトリ・サービス（６６５）によりレプリカ・データが提供される。リソース管理プロトコル（６７１）、レプリカ管理プロトコル（６７２）、配備管理プロトコル（６７３）の３つのシグナリング・プロトコルを介してサーバとのインタフェースが管理シグナリング・モジュール（６７０）により提供される。配備モジュール（６１０）は、本発明に従って配備管理プロトコル（６７３）と連携し、配備ポリシ（６１５）または探索ポリシ（６３５）に従って配備照会を作成し、意向があり対応可能な位置に送る。サーバの受け入れ検査に合格した配備照会は、受け入れ候補と呼ばれる。受け入れ候補は、受け入れ保証という従来の概念とは異なり、比較的短い時間（数秒のオーダ）のみサーバによってリソースが一時的に予約され、その後、受け入れ候補がサーバによって確保されていない場合、受け入れ候補は削除される。後述する通り、交渉モジュール（６３０）、交渉ポリシ・モジュール（６３６）、及び配備管理プロトコル（６７３）は連携し、肯定応答した受け入れ候補群から受け入れ候補を選択／確保して受け入れを保証し、また、前に選択されたサーバ以外のサーバに対する他の全ての受け入れ候補を無効にし、他の全ての保留中の配備照会を無効にする。

【００４３】

コントローラ（５２０）の一部として他に、リクエストのストリームを調べ、最もリクエストの多いオブジェクト（６８１）（以下、ホット・オブジェクトと呼ぶ）と、これらオ

ブジェクトの最も優勢な地域（６８２）のリストを生成し、それらの需要を予測する需要分析モジュール（６８０）がある。これらの統計はレプリカ管理モジュール（６６０）に送られる。容量分析モジュール（６９０）は、最もリクエストの多いオブジェクトそれぞれについて利用できる容量を調べ、利用できる容量をレプリカ管理モジュール（６６０）に送る。

【００４４】

図５に示す通り、本発明のシステムは、リソース管理プロトコル（６７１）、レプリカ管理プロトコル（６７２）、配備管理プロトコル（６７３）の３つのインタフェースを利用して、コントローラ（５２０）とサーバの間で管理情報をリレーする。当業者には明らかなように、これらのインタフェースの実装を容易にするため使用できるプロトコル規格がある。例えばリソース管理プロトコルは、ＲＳＶＰ（Ｒｅｓｅｒｖａｔｉｏｎ　Ｐｒｏｔｏｃｏｌ）、及びＲＴＣＰにより得られる機能をベースに構築することで実装することができる。ＲＳＶＰは、統合サービス・インターネットワーク向けのリソース予約設定プロトコルである。アプリケーションはＲＳＶＰを呼び出して、データ・ストリームに対する具体的なエンド・ツー・エンドＱｏＳを要求する。ＲＳＶＰは、ユニキャストとマルチキャストのルーティング・プロトコルをサポートし、大きいマルチキャスト配信グループにも充分対応するＱｏＳ保証リソース予約を効率よく設定するためのものである。ＲＳＶＰプロトコルは、予想通りマルチメディア・サーバからそのクライアントに、接続ベースでエンド・ツー・エンドのリソース予約を行うため使用されよう。一方、ＲＴＣＰは、ＲＴＰと連携する測定／管理プロトコルである。ＲＴＰセッションの各参加者によりＲＴＣＰ管理パケットが他の全ての参加者に定期的に転送される。アプリケーション・レベルの情報をフィードバックすれば、本発明により必要になるようなパフォーマンス管理や診断が行える。ＲＴＣＰプロトコルは、マルチメディア・サーバとそのコントローラ間で測定値（つまりリソース管理プロトコルのＭＯＮ＿ＳＴＡＴＵＳ、ＭＯＮ＿ＲＥＱＵＥＳＴメッセージ）をフィードバックするために使用されよう。ＲＳＶＰは、分散したパーティ間でサービス品質の交渉を実装する機構を提供するが、ＲＴＣＰは、分散したパーティ間で測定値とパフォーマンス・フィードバック情報をリレーする機構を提供する。同様に、レプリカ管理プロトコルは、インターネットのファイル転送プロトコル（ＦＴＰ）とＲＳＶＰプロトコルにより得られる抽象性をベースに構成できる。ＦＴＰでは、サーバ間のパイプで最善を期してコンテンツの移動が可能になるのに対して、ＲＳＶＰでは、統合サービス・ネットワークのパイプを指定することができる。

【００４５】

前記の通り、本発明のシステムは、分散コンピュータ環境で負荷の分散とリソース管理の統合機能を提供する。コントローラは、好適には、需要を利用できる容量に、ある程度まで自由に適合させる。第１に、リクエストの分散とサーバへの配備を管理し調整する。第２にサーバ（つまり容量）間のレプリカの分散と配備を、一定の基準に従って管理／調整する。また、コントローラは、本発明の機構により、本発明の目的を達成するため必要とみなされるときは、サーバ間のレプリカの動的な作成、破壊、移動を行える。以下、これらの機能について詳しく説明する。

【００４６】

リクエスト管理システム：

図４、図５に関して説明する通り、本発明によれば、固有オブジェクトＩＤがあるとき、リクエスト（５０１、６０１等）を、要求されたオブジェクトのレプリカを持つサーバ（１２１１等）に、中間コントローラ（５２０）を介して配備できる。コントローラは、本発明の好適実施例に従って、すぐに利用できる容量に従って需要を動的に再調整する、次のような機構を実装する。１）第１のアプローチでは、コントローラのニーズとリクエストのパラメータに応じてリクエストをアップグレードまたはダウングレードすることができる。特にコントローラは、元のリクエストとわずかに異なるパラメータ値をもとに配備オプションを探索することができる。２）第２のアプローチでは、コントローラのニーズと個々のリクエストの制約に応じて、時間的に近接した同種のリクエストを遅らせ、グル

ープにまとめることができる。特にコントローラは、任意の時間間隔で同種のリクエストのグループを、例えばマルチキャスト機能を持つグローバル（マルチメディア）サーバに一時的に記憶し、再構成し、バッチ処理を行うことができる。３）他のアプローチの場合、同種の地域特性を共有するリクエストは、コントローラのニーズと地域で最もコスト効果の大きいリソースの可用性に応じて、グループ化することができる。特にコントローラは、クライアント及びサーバを、ＥＳＴ等の時間帯、利用できる帯域幅（Ｔ１ライン等）を含めた地理上の制約に関連付け、これらの基準からリクエストの配備を調整することができる。

【００４７】

そのためコントローラは、統計収集ポイントとして機能する。特に２種類の統計（需要統計と容量統計）がコントローラにより維持される。需要統計は、過去のリクエストに関する特性を記述するためコントローラ（５２０）により使用される。好適実施例の場合、特定のコントローラにより認識される異なるクライアントから集められたリクエスト・ストリームを分析することで、コントローラにより予測需要統計が生成される。例えば、需要の密度と量、及び需要に関連付けられた優勢な地域を特徴付けるため統計が生成される。一方、容量統計は、マルチメディア・サーバの容量に関する特性を記述し、マルチメディア・オブジェクトに対する配備を受け入れるためコントローラにより使用される。好適実施例の場合、利用できるおおよその容量がサーバにより予測され、サーバにより必要とみなされるときはコントローラ（５２０）に送られる。

【００４８】

システムは、本発明の好適実施例に従い、受け入れ管理が各サーバでローカルに実装されるという点で分散される。図１２は図４のグローバル・サーバ１２１１の詳細である。各サーバ（例えばサーバ１２１１）は受け入れ管理機構（１０４０、１０４１）を備え、配備照会（クエリ）リクエストの受け入れ候補を承認または否認し、その旨をコントローラに伝える（オファ）。受け入れ管理機構（１０４０、１０４１）は更に、受け入れ候補を受け入れ保証に昇格させ、また受け入れ候補を無効にするよう機能する。コントローラ（５２０）は受け入れやリソース予約の作業は行わない。当業者には明らかなように、本発明は、サーバの集合及びサーバ・クラスタに適用できる。特に、各サーバ・クラスタに集中受け入れ管理が関連付けられるので、各クラスタは、コントローラからは、１つのＨＩＧＨ容量サーバに見える。そのためコントローラは、サーバとサーバ・クラスタを特に区別することはない。

【００４９】

コントローラ、クライアント、サーバ間のシグナリング・プロトコルは、ここでは、図８に関して詳述しているが、配備管理プロトコルと呼ばれ、これら分散パーティ間で、コントローラ（５２０）がサーバにクライアント・リクエストを配備するために使用される。配備管理プロトコルは、少なくとも次のメッセージの実装を含む。クライアントがリクエストをコントローラに送るために使用するＣＩＤ＿ＲＥＱＵＥＳＴメッセージ、コントローラがサーバ間の配備候補を探索するために使用するＣＩＤ＿ＱＵＥＲＹメッセージ、及びコントローラが受け入れ候補から受け入れ保証への昇格を要求するために使用するＣＩＤ＿ＰＬＡＣＥメッセージである。またこれらメッセージはそれぞれ、シグナリング・パーティが前記の非同期メッセージ（ＣＩＤ＿ＲＥＱＵＥＳＴ、ＣＩＤ＿ＱＵＥＲＹ、ＣＩＤ＿ＰＬＡＣＥ）のいずれかに応答するため使用する確認メッセージＣＸ＿ＡＣＫに関連付けられる。従ってＣＩＤ＿ＱＵＥＲＹの場合、ＣＱ＿ＡＣＫメッセージは、受け入れ候補が承認されたことを示す肯定応答を返す。このメッセージは受け入れの有効期限を示す。当業者には明らかなように、有効期限はサーバ単位で設定し、新しい配備を求めるサーバのアグレッシブネスを区別することができる。更に、実施例によっては、この期限をサーバで利用できる容量に応じて、時間に関して可変とすることも可能である。ＣＩＤ＿ＰＬＡＣＥメッセージの場合、ＣＰ＿ＡＣＫメッセージは、受け入れ候補が受け入れ保証に昇格したかどうかを示すフラグをリレーする。

【００５０】

一般に、リクエストをサーバにマッピングするプロセスは、コントローラにより３段階に分けられる。第１にコントローラは、受け入れ照会を考慮する意向のあることがわかっている要求されたオブジェクトのレプリカを含むサーバからサーバの識別を始める。第２に、コントローラは、ＣＩＤ＿ＲＥＱＵＥＳＴメッセージでコントローラに与えられる可能性のある選択済みパラメータにより、これらのサーバに受け入れの照会を始める。このプロセスは、本発明では、サーバとコントローラ間で、可能ならクライアントが介入して合意が得られるまで繰り返すことができる。最後にコントローラは、最後のステップで対応可能なことがわかったサーバの１つにリクエストを送り始める。前者２つの段階は、最後の段階に入る前に繰り返すことができるので、本発明に従って、リクエストのサーバへのマッピングは、有望な受け入れ候補の動的セットに関する探索と交渉を含む繰り返しプロセスになる。

【００５１】

ここで述べている通り、コントローラのレプリカ・ディレクトリ・サービス（６６５）を使って各レプリカの位置を追跡する機構が提供される。固有オブジェクトＩＤがあるとき、レプリカ・ディレクトリ（６６６）の検索により、システム上の対応する全てのレプリカの位置が返される。好適実施例の場合、レプリカの位置は単にサーバ・アドレスとして表される（ホスト名、ＩＰアドレス等）。サーバ当たり１つのレプリカで充分であることに注意されたい。

【００５２】

図８は、配備管理プロトコルのタイムライン図である。時間ｔでコントローラ（５２０）はクライアント（１１０等）からＣＩＤ＿ＲＥＱＵＥＳＴメッセージ（７１０）を受け取る。時間ｔ１でコントローラ（５２０）はＣＩＤ＿ＱＵＥＲＹメッセージ（７３９、７４０）を、受け入れ照会を考慮する意向のあることがわかっている２つのサーバ（１２０１、１２１１）に送る。ＣＩＤ＿ＱＵＥＲＹメッセージはそれぞれ、解像度、品質、コスト、最大遅延等の名前／値のペアの形の他のパラメータ（７４２）の他に、要求されているオブジェクトの固有ＩＤ（７４１）を含む。パラメータ（７４２）は、ＣＩＤ＿ＲＥＱＵＥＳＴでクライアントにより指定されたパラメータ（７３１）またはそれらを洗練したものに対応することがある。パラメータ（７４２）は、このリクエストに関連付けられた交渉ポリシ（６３６）に従って各サーバ（１２０１、１２１１）と個別に交渉できる。

【００５３】

時間ｔ２でサーバ（１２０１）はＣＱ＿ＡＣＫメッセージ（７６０）でコントローラ（７２０）に応答する。このメッセージは、受け入れ候補がサーバ（１２０１）により承認されているかどうかを示すフラグ（７７０）を含む。メッセージはまた受け入れ候補の有効期限と、ある場合は、このサーバ（１２０１等）が受け入れ候補について提供する意向のある対応するＣＩＤ＿ＱＵＥＲＹメッセージ・パラメータ（７７３）について交渉された名前／値ペアも含む。

【００５４】

同様に、タイムラインは、時間ｔ３で第２サーバ（１２１１）が別のＣＱ＿ＡＣＫメッセージ（７６１）を介してコントローラ（５２０）に応答することを示す。ＣＱ＿ＡＣＫメッセージが、受け入れ候補がサーバにより保持されていることを示すとき、ＣＱ＿ＡＣＫメッセージのパラメータ・フィールド（７７３等）は、対応するサーバが提供する意向のある旨のパラメータ値を示す。コントローラは、数秒のオーダ等、相応の時間内でＣＱ＿ＡＣＫ（７６０、７６１）を待機する。次に、コントローラ（５２０）は、その交渉ポリシ（６３６）をもとに、これまでに受信したＣＱ＿ＡＣＫ応答（７６０、７６１）から１つ（例えば７６０）を選択することができる。

【００５５】

また、受け入れ候補が確保されない可能性もある。これには次のような理由が考えられる。ａ）全てのサーバでリソースが足りない、ｂ）交渉パラメータに関して同意が得られない、ｃ）交渉期限が切れている、またはｄ）前記の組み合わせである。好適には、全てのリクエストを公平に処理するため交渉期限を設定する。これによりコントローラは、他の

リクエストを犠牲にして、有望ではないリクエストのために無駄な検索を行って時間を浪費することがなくなる。交渉期限をシステム・パラメータとする必要があることは明らかである。この期限は、特にサーバとオブジェクトの距離のため、数１０秒のオーダにする必要がある。いずれにしろコントローラは、リクエストに関連付けられた交渉ポリシ（６３６）を参照して、条件に適した処理を決定する。新しいパラメータ・セット（７３１）を要求する、探索ポリシ（６３５）を再評価する等、受け入れ候補を探し、次に新しいセットでサーバにＣＩＤ＿ＱＵＥＲＹを再発行するため、いくつかの処理を適用できる。

【００５６】

タイムラインは、時間ｔ４でコントローラ（５２０）がサーバ（１２０１）にＣＩＤ＿ＰＬＡＣＥメッセージ（７９０）を送ることを示している。このメッセージは、固有ＩＤ（７３０）を含み、サーバはこれにより受け入れ候補を受け入れ保証にアップグレードする。サーバはこのアップグレードをＣＰ＿ＡＣＫメッセージ（７９１）を介して確認応答する。図５に戻る。配備推奨（６２０）はそれぞれマッピング（配備と呼ばれる）・ポリシ（６１５）に関連付けられる。例えば配備／マッピング・ポリシ（６１５）は、ＬＡＲＧＥ容量のサーバの緑のレプリカには常にリクエストを送る、というように指定できる。配備に関連付けられた受け入れ候補がコントローラによって選択されると、配備はコントローラにより保証される。そのためコントローラは、受け入れ保証を配備のため確保し、配備を管理下に置く。このような保証の条件は、コントローラの配備ポリシにより指定される（例えば、緑のレプリカに常にリクエストを出す、ＨＩＧＨ容量サーバを優先する、配備がサーバ障害の影響を受けないようにする等）。その結果、このようなバインディングのパフォーマンスがその存続時間の間コントローラにより監視される。更にバインディングの存続時間中、バインディングのパフォーマンスを危うくするような条件が発生し得る。例えばユーザ側の対話は非線形であるため（ＶＴＲの停止、巻戻し、一時停止、継続等）、代表的なビデオ・オンデマンド（ＶｏＤ）・サーバの場合のように、線形の再生モデルを想定して作成されるバインディングは危うくなる可能性がある。同様に、サーバ障害が発生した場合、通常はバインディングが中断される。コントローラが選択する配備ポリシによるが、コントローラは、このようなバインディングのパフォーマンスを、そのような条件が発生するかどうかにかかわらず保証することができる。例えば、フォールト・トレランスを保証したバインディングでは、サーバがバインディングを削除した場合、コントローラは新しい配備を自主的に探そうとし、そこから処理を再開する。

【００５７】

図５を参照する。可能な配備推奨をそれぞれ個別に探索する代わりに、コントローラが複数の配備推奨（６２０）を同時に探索できることは、本発明の側面である。この動作は、探索ポリシ（６３５）を介してコントローラに指定される。例えばこのような探索ポリシは、最大Ｋ回は繰り返し、各繰り返しで少なくともＬ個、最大Ｍ個のサーバに及ぶ配備を探索する、というように指定できる。探索ポリシは、結果的に失敗するリクエストについては高コストになる可能性がある。例えば、このポリシでは、リクエストが成功する度に、少なくとも２つのメッセージのそれぞれについて合計で少なくともＮ＝Ｋ＊Ｌ個のプロトコル交渉が開始されることになる。本発明では、各リクエストを探索ポリシに関連付けることができる。このようなサーバとレプリカの同時探索によって、同じリクエストが異なるサーバにマップされる可能性があることは明らかである。また、リクエストがあるとき、コントローラは０個以上の配備推奨を作成してもよい。例えば利用できる容量がリクエストに応えるには不充分なとき、配備推奨は作成されない。

【００５８】

図８に戻る。保留されているＣＩＤ＿ＱＵＥＲＹメッセージはコントローラにより棄却することができる。例えば本発明では、コントローラは、（例えば照会の応答が最大待ち時間のしきい値を超えた場合に）対象ではなくなった配備照会を単に棄却するだけである。この交渉の構成は、交渉ポリシ（６３６）によりコントローラに指定される（図５）。例えばこのような交渉ポリシは、パラメータｙと元のパラメータｘ（７３１）のコスト差が最大ｚになるようにパラメータ・セットｘをサーバと交渉する、というように指定できる

。
【００５９】
統合ドリブンの応答：
前記の通り、本発明のシステムは、サーバ容量と予測需要の適合を目指している。そのため本発明は、需要を監視し、分散コンピュータ・システムの容量を監視することによって、リソース管理のためのコントローラ（５２０）を実装する。特にコントローラは、レプリカに対する総予測需要をサーバの利用できる容量とサーバ上のレプリカの配備に合わせようとする。異なるクライアントからのリクエストのストリームを分析することによって予測需要統計を生成する。利用できる容量はサーバの監視と照会により予測する。
【００６０】
コントローラ（５２０）は、オブジェクト、レプリカ、サーバ、リクエスト、及びそれらの配備に関する永続的で動的な状態とデータを記憶する。例えば好適実施例の場合、ある特定のオブジェクトに対する需要、そのレプリカの位置、サーバの容量、及び時間上のリクエストの分散についてのデータを記憶するため、ディレクトリ・サービスを使用する。好適実施例の場合、これらのデータ構造は、データの喪失及びデータの破壊から復帰しやすくされる。
【００６１】
有効なレプリカ候補（以下、ホット・オブジェクトと呼ぶ）であるオブジェクトの選択は、オブジェクトＩＤに対する予測需要等の基準に従って行われる。好適実施例の場合、レプリカ候補の選択は、利用できる容量に対する総予測需要をオンラインで分析することによってドライブする。つまり、レプリカ管理は、第１実施例に従って、予測需要を利用できる容量に合わせようとする。或いはまたレプリカ管理は、クライアントからオブジェクトＩＤまでの地理上の近さに従って、予測需要を利用できる容量に合わせようとする。
【００６２】
本発明のシステムはそのため、リクエストを同種の特性を持つリクエスト・グループに統合する。例えば、有限な時間間隔で独立したクライアントから受信されたリクエストは、管理サーバにより配備するため、所定基準に従ってグループとして管理することができる。リクエストに関する基準の例としては、１）同じオブジェクトＩＤを要求するクライアントの地理上の近さ、２）解像度、品質等の要求される制約の共通性、３）同じオブジェクトＩＤに対するリクエストの到着時間の時間的近接性等がある。
【００６３】
リクエストは、本発明に従って、同じオブジェクトＩＤに対するわずかに異なるリクエストを同種のリクエストとしてプールするため、そのコントローラにより自主的にアップグレードまたはダウングレードすることができる。これは例えば、地理上の近さによるグループ化基準の対応性を低くする、要求されたオブジェクトの品質を低くする、時間的な近さのグループ化基準の対応性を低くする、或いはそれらの組み合わせにより達成することができる。コントローラは、このオプションを行使するか、またはクライアント、リクエスト、オブジェクトのコスト等の特性を基準にしないことも可能である。ただしコントローラは、同じオブジェクトＩＤに対するリクエストを、他のリクエストに関しては関連付けることができない形で処理することを選択することもできる。
【００６４】
先に述べた通り、コントローラは、時間上のリクエストの分散を監視する。好適実施例の場合、オブジェクトＩＤそれぞれについてリクエストの分散に関する統計が維持される。需要統計はオブジェクトＩＤに対する相対需要について情報を与え、需要の点からオブジェクトＩＤを評価することができる。好適には、各オブジェクトについて特徴付けられる統計は、１）需要の密度と、２）需要の量を含む。これらの他、各オブジェクトについて特徴付けられる統計に、３）需要に関連付けられた優勢な地域を追加してもよい。コントローラは、分散集合上のオブジェクト毎に需要統計を維持する。特定のオブジェクトの需要統計は、オブジェクトに対するリクエスト毎に更新される。特にコントローラは、需要の多いことがわかったオブジェクト（ホット・オブジェクト）をフラグで指示する。図５

の需要分析モジュール（６８０）で特定のオブジェクトＯ１について実行される需要統計の計算を、図１３乃至図１５に示す。当業者には明らかなように、予測の信頼性と精度を高めるためこれらの統計を計算する方法は多数ある。

【００６５】

図１３は、コントローラから見た時間間隔t_{n-3}、t_{n-2}、t_{n-1}、ｔに及ぶリクエスト・ストリーム（１１１０ａ、．．．、１１１０ｄ）を示す。リクエストはそれぞれオブジェクトＩＤ（オブジェクトＯ１）と、要求側クライアントに関連付けられた地域を一意に識別するため使用する地域インディケータＧ１、Ｇ２、Ｇ３、またはＧ４に関連付けられる。例えば図９は、レプリカＯ１、Ｏ２及びＯ３の様々なグローバル・サーバ８４０、８５０、８６０への配備を動的に管理する分散コンピュータ・システムを示す。ここでは需要と地域の傾向により、永続的レプリカと一時的レプリカの区別が促進される。一時的レプリカは常にグローバル・サーバ上にあり、コントローラにより決まる動的存続時間を持つ。コントローラは、後述するように、コスト、需要、容量等の所定基準に従って、レプリカのグローバル・サーバへの動的配備を管理する。好適には、関連付けられる地域Ｇ１、Ｇ２、Ｇ３、またはＧ４は、東部標準時間帯、太平洋標準時間帯、米北東部、米南西部等、既知の地域に適合するようにシステム管理者がアプリオリ（ａ－ｐｒｉｏｒｉ）に設定する。ただし地域は、コントローラにより動的に設定することもできる。例えばコントローラはクライアントを属性や特性に応じてグループ分けし、この基準を、同種のリクエストの配備に役立てることができる。

【００６６】

図１３に戻る。リクエスト・ストリームの要求されたオブジェクトＩＤに対するリクエスト統計の生成が示してある。図１３の例は、４つの時間間隔（例えばＴ（ｎ－２）（１１１０ｂ）とＴ（ｎ）（１１１０ｄ））に対する統計の計算を示す。需要統計は、有限の時間間隔Ｔ（ｊ）当たりのリクエスト数として計算される。この例は、需要統計が、最初の間隔（Ｔ（ｎ－３））の１０／Ｔから次の間隔（Ｔ（ｎ－２））の１３／Ｔまで、３つ目の時間間隔の１５／Ｔまで、最後の時間間隔の１６／Ｔまで変化することを示す。需要統計は、安定性を高めるために、コントローラにより使用される前に平滑化することができる。

【００６７】

図１４は、図１３に対応する優勢な地域の統計の計算を示す。この例ではシステムは、アプリオリに知られる４つの地域（Ｇ１、Ｇ２、Ｇ３、Ｇ４）に分けられる。リクエストがコントローラにより分析される前に、リクエストが入る地域のタグがリクエストに付けられる。各時間間隔Ｔ（ｊ）で、あるオブジェクトに対して入るリクエスト毎に、コントローラが地域別にそれらのリクエストをソートし、各地域に関連付けられたリクエスト・カウンタ（図示せず）を更新する。従って、各時間間隔でリクエスト・ストリームの需要の傾向が監視される。図１４の例の地域Ｇ４は、前の時間間隔でのオブジェクトＯ１に対するリクエストの安定した増加をもとに、時間ｔ（ｐ）でのオブジェクトＯ１に対する予測需要の伸びを示している。つまり、地域Ｇ４は、４つの時間間隔で、他の地域Ｇ１、Ｇ２、Ｇ３に対して着実に優勢な地域になっている。ここでも、優勢な地域のインディケータは、安定性を高めるため、コントローラにより使用する前に平滑化することもできる。

【００６８】

図１５は、各レプリカについて維持される需要統計を記憶するためのスキーマとデータ構造（６９６）の例を示す。コントローラは、各オブジェクトＩＤについて、その予測需要に関する統計を記憶する。予測と円滑化の現在の慣行に従って、移動ウィンドウ統計が使用される。需要統計を円滑にするため時間間隔（ここではＴと呼ぶ）が使用される。需要統計を予測するためサイズＫ＊Ｔの移動ウィンドウが使用される。使用する平滑化方式（指数平滑、均一スムーザ等）と所望の安定性またはスムーザの信頼性により、平滑にする間隔の数（Ｋ）が求められる。需要統計の更新に使用する時間間隔のサイズは、ａ）オーバヘッドを少なくする、ｂ）遷移効果を平滑にする、またｃ）充分大きい数のリクエストを対象とするために必要なので、充分大きくする必要がある。一方、ＴとＫが比較的小さ

いとき、コントローラは変化に対してオンデマンドでより速やかに反応することができる。インターネット分野で現在一般的で合理的な値は、Ｋ＝２、Ｔ＝［６０、．．．、３６００］秒の指数スムーザである。従って、図１５に示した例のデータ構造では、リクエスト統計（１１５０）が、最後のＫ個の時間間隔Ｔ（つまりｊ、ｊ－１、．．．、ｊ－ｋ＋１）について、このコントローラから見た、対応するオブジェクトＩＤに関連付けられたリクエストの密度を示す。優勢地域インディケータ（１１６０）は、対応するオブジェクトＩＤに関連付けられた優勢な地域を示す。需要統計（１１７０）は、最後のＫ個の時間間隔Ｔのリクエスト数の移動合計を示す。コントローラは、あるオブジェクトに対する需要が多いかどうかを評価するため、密度と量の統計を確認する。そのためコントローラは高率（量）、高密度、及び好適には高密度で高率（量）のオブジェクトを探す。需要の多いことがわかったオブジェクトは、ホット・オブジェクト・ブール・フィールド（１１８０）を介してコントローラによりそのように識別される。このフィールドのＹＥＳは、オブジェクトＩＤに対する需要が多いことを示す。最後にタイムスタンプ（１１９０）は、最後の需要評価の時間を追跡するため使用される。当業者には明らかなように、古すぎる測定値の評価は、その信頼性が低いので（好適には）避ける必要がある。コントローラは、好適にはオブジェクトＩＤだけをベースにして優勢な地域を追跡するのではない。例えば優勢な地域の統計はレプリカ単位でも追跡できる。コントローラは、このような追跡により、特定の（ある地域に関連付けられた）レプリカが（長期的には）異なる優勢な地域からのリクエストに対応するかどうかを効率よく検出することができる。そのような状況を訂正するため、ある地域のサーバ上に位置するレプリカを、過去の配備に関連付けられた優勢な地域をそのレプリカに合わせる新しいグローバル・サーバに移行する機能を提供するレプリカ移行機構を使用することができる。レプリカ移行機構は、後述する容量調整機構の一部として実装する。更に優勢な地域の最近の履歴から、あるレプリカまたはオブジェクトＩＤに関連付けられた優勢な地域のシフトの安定性を評価できる。

【００６９】

利用できるシステム容量を監視し予測するため採用する手段は、リソース監視サブシステムを配備することで得られる。リソース監視システムは、サーバがその利用できる容量をＭＯＮ＿ＳＴＡＴＵＳメッセージを通してレポートできるリソース管理プロトコルを通してサーバとのインタフェースをとる。特にＭＯＮ＿ＳＴＡＴＵＳメッセージはその将来の可用性の予測をコントローラにリレーする。この予測はバインディングではなく、コントローラにより契約とみなされることもない。予測は、図８に関して説明している将来のＣＩＤ＿ＱＵＥＲＹメッセージを考慮するサーバの意向を示すとみなされる。好適実施例の場合、新しいリクエストを考慮するサーバの意向は、後述するようにその利用できる容量の関数である。そのためコントローラはこれをサーバの使用率／意向状態と呼ぶ。

【００７０】

本発明の好適実施例に従い、図７に関して説明している使用率／意向状態の目的は２つある。第１に、コントローラの観点から、使用率／意向インディケータは、サーバのリソース使用率に関する多様性に対応するため使用される。特に好適実施例は、３色のウォータマークを、サーバから独立したその使用率／意向インディケータとして使用する。この方式と、サーバの容量評価とにより、コントローラは、異なるサーバの相対的使用率を比較して将来の配備照会を処理する。その結果、システム内のサーバの使用率／意向は、使用状態（緑、黄、赤）と２つの容量評価（ＨＩＧＨ、ＬＯＷ）による６つ（３×２＝６）条件に関して測定される。

【００７１】

一方、各サーバは独立して、そのニーズに合わせて使用率／意向インディケータを設定することができる。特にサーバは、その赤、緑、黄のしきい値を、特定の使用レベルを反映する値ではなく、それぞれの意向に合った値に設定することができる。更に、このような意向値は動的に変更できると考えられる。変更はその場合、将来の需要に対するサーバの応答での動作の変更を表す。この第２の側面は、ここでは、配備照会の取得に向けたサーバのアグレッシブネスまたは意向と呼んでいる。つまりサーバの観点からは、使用率／意

向インディケータ方式を、そのサーバの管理者が、コントローラからの配備を求めるサーバのアグレッシブネスまたはその欠如をカスタマイズするため使用できる。言い換えるとリソースが同じで物理位置も同じ２つのサーバは、それぞれに割当てられる使用状態がどの程度アグレッシブかに応じて配備照会が異なることがある。

【００７２】

特にサーバの管理者は、３つの使用状態（緑、黄、赤）を、サーバ管理者のサービスに関するニーズに合ったしきい値に設定することができる。例えば、信頼性が高く、サービス保証が安定していると認知されることに関心を持つサーバが考えられる。その場合、サーバ管理者は緑の領域を控えめな値にカスタマイズする（例えば、実際の容量の５０％等の低い値にし、１００％の容量のオーバエンジニアリングを犠牲にしてサービスを保証する）。一方、ジターがある、サービスが拒否される等、サービス品質を犠牲にして安価なサービスを提供するとの認知を得たいというサーバもあり得る。その場合サーバ管理者は、緑の領域をアグレッシブな値にカスタマイズする（例えば、実際の容量の８５％といった高い値にし、黄と赤の領域は１５％とする）。その結果、黄と赤の領域は、レプリカ作成リクエストとリソース利用の無作為性に対応するため使用される余剰リソースを表すので、この余剰リソースを小さくすることは、緑の領域で受け入れられるストリーミング接続に対するサービス保証が影響を受けることになる。ただし、このサーバは、コントローラからの配備照会の数が増える。このようなサーバはそこで、照会を収益や統計等の”貪欲な”基準に従って選択することができる。

【００７３】

図１０、１１は、例として、好適実施例により指定されるようなウォータマーク方式（９００）を示す。ウォータマーク方式（９００）は、サーバがその使用状態を、コントローラが全てのサーバに依存できるような正規化した意向インディケータ（９９０）にマップするため使用する。

【００７４】

特に図１０は、使用負荷がかかるとき、特定のサーバ（サーバ１等）の使用率／意向プロファイル（９６０）を示す。赤の条件（９１０）は、サーバがそのコントローラに、サーバがＣＩＤ＿ＱＵＥＲＹメッセージを考慮できなくなっていることを通知するため使用する。黄の条件（９２０）は、サーバがそのコントローラに、ＣＩＤ＿ＱＵＥＲＹメッセージを考慮できなくなっているが、保留中のＰＬＡＣＥメッセージは考慮できることを通知するため使用する。最後に緑の条件（９３０）は、サーバがそのコントローラに、ＣＩＤ＿ＱＵＥＲＹメッセージを考慮することを通知するため使用する。このフラグは、各サーバがその使用率／意向状態を示すため更新する。サーバは、その意向インディケータを変更したときにのみＭＯＮ＿ＳＴＡＴＵＳメッセージをコントローラにディスパッチする。３つのフラグでは６つの条件が考慮されるが、１）９４０に示すような緑から黄／赤への変更と、２）赤／黄から緑への変更（９５０）の２つの条件が重要とみなされる。

【００７５】

同様に図１１は、使用率がサーバ１と同じとき、別のサーバ（サーバ２）から得られる別の使用率／意向プロファイル（９６１）を示す。各サーバは独立にその赤、緑、黄のウォータマークを、ＣＩＤ＿ＱＵＥＲＹメッセージを受け取るそれぞれの意向に合った値に設定できる。使用率／意向状態に加えて、サーバがＨＩＧＨ容量かＬＯＷ容量かを示すため容量評価が使用される。サーバの容量評価の決定は、サーバがその緑状態で提供する同時ストリームの最大数の点で、直接的なしきい値チェックをもとに行える。ここで、緑のレプリカは緑のサーバ上のレプリカを指し示している。同様に緑のサーバは、その最後の使用率／意向状態（９９０）が緑（９３０）とレポートされたサーバを指し示す。

【００７６】

図５に戻る。サーバは、リソース監視／管理プロトコル（６７１）によりその使用率／意向状態、例えばその緑、容量評価（低容量等）等をコントローラにレポートする。その使用率／意向と容量をコントローラにレポートするため、サーバは、レポート側サーバ（例えばそのＩＰアドレスやホスト名を介して）、コントローラ、サーバ側のレポートの時間

、新しい使用率／意向状態、及び容量評価を識別するＭＯＮ＿ＳＴＡＴＵＳメッセージを使用する（後述）。

【００７７】

サーバとコントローラの間でメッセージが順に受信されるように、ＴＣＰ等のＦＩＦＯ型トランスポート機構を使用できる。新しいＭＯＮ＿ＳＴＡＴＵＳメッセージは、コントローラ側の対応するサーバについてそれぞれ最後にレポートされた状態を無効にするが、容量評価がサーバにより空白になっている場合、そのサーバの容量についてコントローラによって変更が記録されることはない。更に、ＭＯＮ＿ＳＴＡＴＵＳメッセージが失われた場合、システムは次のように復帰する。失われたメッセージが赤への変更（９４０）（図１０）を示していた場合、後の配備（つまりＣＩＤ＿ＱＵＥＲＹメッセージ）はサーバ側の受け入れ検査に合格しない。そのイベントはサーバにより、コントローラとサーバ間の配備契約に対する違反とみなされる。その結果その赤のサーバは、他のＣＩＤ＿ＱＵＥＲＹ配備メッセージの受信を避けるため、必要なら赤のＭＯＮ＿ＳＴＡＴＵＳメッセージを対応するコントローラに再送する。

【００７８】

コントローラは、必要ならリソース監視状態を特定のサーバに要求することができる。コントローラは、ＭＯＮ＿ＲＥＱＵＥＳＴメッセージを介して、リソース監視プロトコルにより使用率／照会状態を照会し、また、特定のオブジェクトＩＤの要件について評価されるときに任意のサーバの実際の使用できる緑の容量を判定することもできる。特定のサーバをプールできることは、後述するように、レプリカ配備プロセス（１４００）により実現されるように、グローバル・サーバ上のオブジェクトの新しいレプリカを配備するかどうかを決定するときに、コントローラにとって便利である。

【００７９】

図１５に関して先に説明した需要統計とデータ構造（６９６）は更に、コントローラがそのサーバのレポートされた容量と使用率／意向を追跡するため使用できる。図１５に示す通り、需要統計とデータ構造は、各オブジェクトＩＤに対するリクエストに関連付けられた統計（予測需要ｄ、需要統計またはレートｒつまり時間間隔当たりのオブジェクト・レプリカに対するリクエストの状況、要求されたオブジェクトがホット・オブジェクトかどうかを示し、その状況の要旨を表す指標等）を維持する。オブジェクトＩＤには、オプションとしてオブジェクト・リクエストに関連付けられた優勢な地域を表す優勢地域インディケータが関連付けられる。更に、存続時間タイムスタンプが各レプリカに関連付けられる。タイムスタンプの期限が切れると、一時的レプリカを現在所有しているグローバル・サーバがそのコントローラ（つまりこのレプリカを配備したコントローラ）に更新を要求する。この時点でコントローラは、レプリカの更新を拒否することによってレプリカを削除するか、またはレプリカの存続時間を延長して更新する（従ってこの新しいレプリカでデータベースの再書込みを行う）ことができる。コントローラが更新を拒否した場合、一時的レプリカは、最後にはそのグローバル・サーバにより削除される可能性がある。当業者には明らかなように、これらデータ構造は、フォールト・トレランスのため定期的チェックポイントにすることが望ましい。データが失われた場合、そのデータは、コントローラがＭＯＮ＿ＲＥＱＵＥＳＴメッセージを介して各サーバに照会することで再構成する。レポートが利用できないサーバについては、ＭＯＮ＿ＳＴＡＴＵＳメッセージがそのサーバから受信されるまで、対応する使用率／意向状態のデフォルトを赤にし、その容量評価は空白にする。この方法では、サーバの使用率／意向状態が再び緑になるまでそのサーバに新しい配備が割当てられないので、サーバの使用率が下がった状態でサーバ障害に対するコントローラのフォールト・トレランスが向上する。

【００８０】

地理上の容量調整：

先に述べた通り、コントローラ（５２０）は、ある程度自由に需要と容量を合わせることができる。第１にリクエストの分散とサーバへの配備を管理／調整する。第２に、一定の基準に従ってサーバに対するレプリカの分散と配備を管理／調整する。最後にコントロー

ラは、本発明の機構により、その目的を達成するため必要とみなされるときサーバ間のレプリカの動的な作成、配備、移動を行える。

【００８１】

サーバ上のレプリカの動的な作成、配備、移動の機能に関しては、オブジェクト・レプリカが、予測需要と利用できる容量の予測に応答してサーバ間で移行できるようにされる。そのため本発明は、ネットワーク全体でリクエストだけでなく、それより重要であるが、レプリカの配備も調整する機構を提供する。このレプリカ管理方式は、予測リクエスト需要と利用可能な容量の特性をベースにしてもよい。

【００８２】

図９は、１例として、好適実施例に示しているように、需要、地域等、一定の基準に従って、グローバル・サーバへのレプリカの配備を動的に管理できる分散コンピュータ・システムが必要なことを示す。この例では、おおよそ米国の北東部（Ｇ１）、南東部（Ｇ４）、北西部（Ｇ２）、南西部（Ｇ３）に対応する４つの地域（Ｇ１、Ｇ２、Ｇ３、及びＧ４）がある。ローカル・サーバ（８３０）はＧ１にある。グローバル・サーバ（８４０）はＧ４にある。システムは、他に２つのグローバル・サーバ（８５０、８６０）を含む。グローバル・サーバ（８５０）はＧ２地域をカバーし、グローバル・サーバ（８６０）はＧ３地域をカバーする。システムは分散オブジェクトの集合を含む。この場合、集合は３つのオブジェクト（Ｏ１、Ｏ２、Ｏ３）から構成される。これらオブジェクトはそれぞれ、システムに分散し得る、つまりグローバル・サーバとローカル・サーバの両方にあり得るレプリカに関連付けられる。本発明では、ローカル・サーバにあるレプリカは永続的レプリカと呼ばれ、グローバル・サーバにあるレプリカは一時的レプリカと呼ばれる。分散集合のオブジェクト毎に、システム全体で一時的レプリカが存在し得る。

【００８３】

図９の例では、３つの永続的レプリカ、オブジェクトＯ１（８２０）、オブジェクトＯ２（８００）、オブジェクトＯ３（８１０）がある。これらのレプリカは全てＧ１ローカル・サーバ（８３０）にある。一方オブジェクトＯ１には、Ｇ４グローバル・サーバ（８４０）にある１つの一時的レプリカ（８０１）があり、オブジェクトＯ２には、Ｇ４（８４０）とＧ３（８６０）のグローバル・サーバにある２つの一時的レプリカ（８１１、８１２）がある。オブジェクトＯ３は、Ｇ１ローカル・サーバとその永続的レプリカ（８１０）を介して充分な容量が与えられるオブジェクトを表す。その結果、システムにオブジェクトＯ３の一時的レプリカは（現在は）ない。更に、この例ではオブジェクトＯ２は、需要の多いと予測されるオブジェクトつまりホット・オブジェクトを表す。この例は、オブジェクトＯ２に対するかなりの数のリクエストがＧ３地域から来ており、オブジェクトＯ２に優勢な地域が関連付けられていることが過去の履歴の分析から示されていることを想定している。システムは、オブジェクトＯ２に関連付けられた優勢な地域と需要統計をもとに、Ｇ３地域にレプリカを配備するのが望ましいと判定する。従ってオブジェクトＯ２の一時的レプリカ（８１２）がＧ３グローバル・サーバ（８６０）に一時的に配備されている。

【００８４】

一時的レプリカは、コントローラにより決定される動的存続時間を持つ。一時的レプリカの存続時間は、例えば対応するオブジェクトに関連付けられた利用可能な容量に対する需要、及びオブジェクトの予測セッション時間により異なる。例えば９０分の映画には２時間とあまり余裕のない期限を設定でき、そのうち３０分は、ユーザの対話と、コントローラからの期限更新のための時間を想定して割当てられる。例えば９０分の映画に２４時間の期限というように、かなり余裕のある期限も設定できることは明らかである。このような方法は、そのような時間にオブジェクト需要があると予測され、ただし需要は、その時間帯に多いことを保証するほど多くはないときに使用できる。更にレプリカの前記の存続時間の期限は、一時的レプリカに対して新しいリクエストが出される毎にリセットすることができる。

【００８５】

図９に示す通り、一時的レプリカは、（８００、８１０、８２０）等のオブジェクトの集合に関連付けられる（８０１、８１１、８１２）等の動的レプリカ・セットに記憶域とストリーミング・リソースを提供するグローバル・サーバ（８４０、８５０、８６０）に位置する。この例でオブジェクトＯ１（８００）の一時的レプリカ（８０１）はグローバル・サーバ（８４０）に、オブジェクトＯ２（８１０）の一時的レプリカ（８１２）はグローバル・サーバ（８６０）にあり、オブジェクトＯ３（８２０）の一時的レプリカは存在しない。更に、図９の例では、第３のグローバル・サーバ（８５０）が利用できる形で示してあるが、一時的レプリカはここにない。コントローラは、コスト、需要、容量等の一定の基準に従って、グローバル・サーバに対するレプリカの動的配備を管理できるようにされる。

【００８６】

図１２は、図９のサーバ（サーバ（８３０）または（８４０）等）のモデリングを詳しく示す。前記のように、サーバは記憶域やストリーミング・リソースをオブジェクトに提供する。ただしサーバ（１０００）はここでは、ローカル部（１０１０）とグローバル部（１０２０）の２つの独立した部分に分けられる。どちらの領域も、独立した記憶域やストリーミング・リソースを集合（１０１１）、及び（１０２１）上のオブジェクトに提供する。これらの集合（１０１１、１０２１）は独立に管理される。ただしローカル部（１０１０）ではメンバシップ集合（１０１１）が閉じているのに対して、グローバル部（１０２０）ではメンバシップ集合（１０２１）が開いている。前記のように、グローバル部については、メンバシップはコントローラが管理し、ローカル部についてはサーバがローカルに管理する。サーバは１つの部分にしか専用にすることはできない（つまり１つの領域に１００％、もう１つの領域には０％）。

【００８７】

好適実施例に従い、分散システムは、（１００％）ローカル・サーバの集合と（１００％）グローバル・サーバの集合と２種類のサーバで構成することができる。従って、図２に関して説明しているサーバの実施例に加え、図１２の各領域（１０１０、１０２０）は、ここで説明する５つのソフトウェア・モジュールまたはプロセスで構成することができる。

【００８８】

サービス・ロジック（図２のものと同じ）は、サーバ上のアプリケーション指向の機能を提供する。アプリケーション指向機能の例として、請求処理、ストリーミング・セッションでのクライアントの対話の処理等がある。ストリーミング・プロセス（２７５）は、サーバからクライアントにマルチメディア・コンテンツを配信するネットワーク・ストリーミング機能を提供する。この機能は通常、ＲＴＳＰ等の標準プロトコルに従って実行される。受け入れ管理プロセス（１０４０）は、コントローラからの照会に適用される代表的な受け入れ管理作業を実行する。受け入れ管理プロセス（１０４０）はリクエストを評価し、コントローラへの受け入れオファ（以下、コントローラによる候補配備と呼ぶ）を生成する。リソース管理プロセス（１０５０）は拡張リソース監視機能を提供し、コントローラはこれにより、サーバの使用率／意向状態やその容量等のサーバに関する統合指向属性を判定することができる。リソース管理プロトコル（６７１）は、サーバの状態を監視し照会するためのシグナリング（ＭＯＮ＿ＳＴＡＴＵＳ、ＭＯＮ＿ＲＥＱＵＥＳＴ等）メッセージを指定する。最後に、レプリカ作成管理プロセス（１０３０）は、グローバル・サーバに対する一時的レプリカの作成と削除を可能にするため、サーバに追加される新しいプロセスを代表する。ここで述べるレプリカ作成管理プロトコルは、オブジェクトのレプリカのオンデマンド作成を可能にするシグナリング要件を提供する。シグナリング・インタフェース（１０３１、１０４１、１０５１、２７１）はそれぞれ、サーバがコントローラ上の対応する配備管理、リソース監視、ストリーミング、及びレプリカ管理のプロセスに従えるようにする。前記の通り、グローバル・サーバ上のオブジェクトの集合のメンバシップはオープンである。オブジェクトは、例えば、特定のオブジェクトに対する予測需要等のファクタをもとに集合に参加することができる。同様にオブジェクトは、例えば

集合の他のオブジェクトと比較した特定のオブジェクトの相対的使用率、収益等のファクタをもとに集合から離れることができる。この動的メンバシップの管理は、サーバ間のオブジェクトのレプリカ作成、及びグローバル・サーバ間のレプリカの移動を行うためのレプリカ管理シグナリング・プロトコルを介してコントローラにより自主的に管理できる。

【００８９】

例えば、あるオブジェクトに、最大数Ｎの一時的レプリカを実装することができる。この数は、アプリオリに決定するかまたは実行時に動的に設定でき、オブジェクトの一時的レプリカの最大数はオブジェクト毎に変更可能である。また、あるオブジェクトに関連付けられる一時的レプリカの数は時間とともに変わり、例えば需要が伸びたとき、オブジェクトがホットなとき、容量が少ないとき等には自主的に増やし、或いは需要が低下したとき、オブジェクトがホットでなくなったとき、容量が予測需要に対して充分大きいことがわかったとき等には少なくすることができる。

【００９０】

具体的には、レプリカ管理システムは、４つのプロセスと補助的シグナリング・プロトコル（つまりレプリカ管理プロトコル）を含み、オブジェクト・レプリカのオンデマンド作成を実装するように機能する。レプリカ管理システムは、サーバのリソース容量等の属性を示す所定の制約をもとに特定のサーバに向けられるような規制応答での、サーバに対するコントローラの規制応答（つまりレプリカやリクエストの配備）に責任を持つ。特に同じグローバル・サーバに対するリクエストとレプリカの配備は、クライアントとコンテント作成者により示される明示的な共同割当て制約を満足することに重点を置いて行える。

【００９１】

本発明では、リクエスト（つまり需要）とレプリカ（つまり容量）の管理システム間の対話は、それぞれ予備不足チェック、予備供給過剰チェックと呼ぶ需要から容量（つまり一方向）への２つのトリガで構成される。予備不足チェックは、リクエスト管理システムが、あるオブジェクトに対する需要が伸びると予測されるときには、レプリカ管理システムに容量不足監査を要求するため行い、特定のオブジェクトに対する需要が少なくなると予測されるときには、容量過剰監査をレプリカ管理システムに要求するため行う。予備テストにより、需要から容量への条件が明らかになった場合、レプリカ管理システムから総合分析が要求される。これによりレプリカの作成や削除が行われることがある。これらのチェックが予備と呼ばれるのは、その目的が、レプリカ管理のオーバヘッドとアグレッシブネスのバランスをとることだからである。

【００９２】

容量調整機構は、本発明で述べている一定の条件に応じてアクティブにされる。特にリクエスト管理システムによる予備チェックは、不足条件（ある予測需要に対して利用可能な容量が供給不足と定義される）を確認するため行われる。このチェックは、容量調整機構のアクティブ化をトリガするため行われ、レプリカ管理システムと呼ばれる。同様に、予備チェックは、供給過剰条件（ある予測需要に対して利用可能な容量が供給過剰と定義される）を確認するため行われる。このチェックは、容量調整機構のアクティブ化をトリガするため行われる。先に述べたレプリカ管理とリクエスト管理の各システムの統合（つまり需要と容量の調整）について、ここで図１６に関して詳しく説明する。図１６は、後述するようにレプリカ管理システムの様々なプロセス間の対話を示す。

【００９３】

図１６に示す通り、予備不足チェック（１４０５）は最初に、不足条件を確認するためリクエスト管理システムにより実行される。リクエスト管理システムでサービス・バインディングが確立された後、予備不足監査チェック（１４０５）が行われる。好適実施例の予備不足チェック（１４０５）を図１８に詳しく示す。

【００９４】

図１８に示した好適実施例の場合、予備不足チェック（１４０５）により、要求されたオブジェクトの緑のレプリカが１つしか残っていないかどうか判定される（１４１０）。当業者には明らかなように、不足チェック（１４０５）は、安定性やアグレッシブネスを変

えていくつかの方法で実装することができる。例えば、アプリオリに選択したオブジェクト・セットに優先権を与えることができる。その場合、予備不足チェック（１４０５）は、このバイアスを反映するよう変更できる。不足条件が確認されると、レプリカ作成プロセス（１３００）に監査リクエスト（１４１５）が出される。監査リクエスト（１４１５）は当該オブジェクトを識別する。その目的は、指定されたオブジェクトについてレプリカ作成プロセス（１３００）に、より総合的な評価を要求することである。好適実施例のレプリカ作成プロセス（１３００）の詳細を図１９に示す。特にレプリカ作成プロセス（１３００）は、そのような監査リクエスト（１４１５）が出されたときのみ実行され、他の場合、監査はトリガされない（１４９９）。レプリカの作成により、対応するレプリカ・ディレクトリ（６５６）が更新される。

【００９５】

図１６で、レプリカ作成プロセス（１３００）の目的は、監査リクエスト（１４１５）で指定されたオブジェクト（例えばオブジェクトＯ１）について新しいレプリカのニーズが本当にあるかどうかを判定することである。新しいレプリカのニーズがある場合、レプリカ配備プロセス（１４００）に配備リクエスト（１７１０）が登録される。好適実施例のレプリカ配備プロセス（１４００）の詳細を図２０に示す。配備リクエスト（１７１０）は、指定オブジェクト（Ｏ１等）がレプリカ作成基準を満たしており、レプリカを作成する必要のあることを示す。好適実施例のレプリカ作成基準（１８００）の詳細を図２０に示す。特にレプリカ作成基準は、需要統計（６９６）、レプリカ・ディレクトリ（６６６）、サーバ・ディレクトリ（６５６）等のコントローラ・ベースのデータ構造に依存する需要／容量評価を実装する。

【００９６】

レプリカ配備プロセス（１４００）は、保留されている配備リクエスト（１７１０等）を選択し、そのリクエストについて、可能なら新しいレプリカの配備を決定する。当業者には明らかなように、レプリカ作成リソースが少ないときは、保留中のいくつかの配備リクエストを登録し、コスト基準（配備ポリシ）（１７４５）によりオブジェクトのレプリカ作成に優先順位を付けることもできる。好適実施例ではＦＩＦＯで順序付けを行う。

【００９７】

図１６、図２０に示す通り、レプリカ配備プロセス（１４００）の目的は、所定基準（１４４０）をもとに、ソース・サーバ（１７２０参照）とターゲット・サーバ（１７３０）を識別することである。好適実施例の場合、コントローラが、ここで説明しているリクエストの配備と同様な形でレプリカ作成オプションを調べて交渉する。これは、図１６に示し、図２３に関して説明するレプリカ管理プロトコル（１２００）により提供される照会機能を呼び出すことにより行う。

【００９８】

ソース（１７２０）とターゲット（１７３０）のサーバが識別されると、つまりオプションがステップ（１４５０）で受け入れられると、レプリカ管理プロトコル（１２００）が、対応する配備リクエスト（１７１０）に対するレプリカ作成リクエスト（１７４０）を登録する。これはレプリカ管理プロトコル（１２００）により提供されるレプリカ作成機能を呼び出して行う。当業者には明らかなように、レプリカ作成時にいくつか条件が発生することがある。好適実施例の場合、レプリカ作成ポリシ（１７６５）により。レプリカ管理プロセス下で例外処理をカスタマイズすることができる。

【００９９】

同様に、レプリカ管理システムはレプリカを削除する機能を提供する。リクエスト管理システムは、予備供給過剰チェック（１５０５）を行い、供給過剰条件を確認する。好適実施例の予備供給過剰チェック（１５０５）の詳細を図１７に示す。予備供給過剰監査チェック（１５０５）は、サービス・バインディングの終了時に行われる。更に、サーバが一時的レプリカの更新を要求したときにも適用される。供給過剰条件が確認されたとき、監査リクエスト（１５１５）リクエストがレプリカ削除プロセス（１５００）に出される。好適実施例のレプリカ削除プロセス（１５００）の詳細を図２１に示す。

【０１００】
レプリカ削除プロセス（１５００）の目的は、特定のレプリカに関連付けられたオブジェクトに対するグローバル需要対容量基準（１８００）をもとに、そのレプリカを削除するかどうかを判定することである。好適実施例の場合、一時的レプリカが、その存続期限、需要対容量、及びオブジェクトがホットかどうかにもとづく複雑な基準から削除候補とみなされる。好適実施例の削除基準（１８００）を図２２に示す。特に削除基準は需要／容量評価を実装する。
【０１０１】
好適実施例の場合、削除基準とレプリカ作成の基準は補い合う関係にある。つまりレプリカを更新する条件は、レプリカを初めて作成することと同じである。好適実施例のレプリカ削除プロセス（１５００）の詳細を図２１に示す。レプリカ削除プロセス（１５００）によりレプリカを削除する理由が見つかった場合、レプリカ管理システムは、ただレプリカの更新（つまりレプリカの存続期限の更新）を拒否するだけである。レプリカが削除されると、対応するレプリカ・ディレクトリ（６５６）が更新される。
【０１０２】
図１９は、レプリカ作成プロセス（１３００）のフローチャートである。先に述べた通り、レプリカ作成プロセス（１３００）の目的は、要求されたオブジェクトの新しいレプリカを作成する必要があるかどうかを判定することである。ステップ（１３００）で監査リクエスト（１４１５）（図１６）が受信されると、レプリカ作成プロセスは、要求されたオブジェクトがホット・オブジェクトのセットのメンバかどうかを判定する（１３０４）。オブジェクトがホットなら、包括的不足チェックがステップ（１３５０）で要求される。このチェックは需要対容量チェックと呼ばれ、詳細を図２２に示している。一方、レプリカがホット・オブジェクトのセットのメンバではない場合（１３５５）、コントローラは、まだレプリカを作成する時期ではないと判断し（１３６０、１３７０）、従ってレプリカは作成されない（１３７５）。ただし、ステップ（１３６０）では、オブジェクトがホットでない場合にもコントローラがレプリカを作成することを決定できる。例えば、選択されたオブジェクトには、コスト基準をもとに優先権を与えることができ、従って、例えば大きいコスト・メリットに関連付けられたオブジェクトに対して優先的にレプリカを作成できる。従ってステップ（１３６５）で、図２０に示すようにレプリカ配備アルゴリズムが呼び出される。
【０１０３】
図２２は、需要対容量をチェックするプロセス（１８００）のフローチャートである。ステップ（１８３０）で、特定のオブジェクト（Ｏ（Ｒ））に対する予測需要（ステップ１８１０で判定）が利用可能な容量（ステップ１８２０で判定）を超えるかどうか判定される。ステップ（１８３０）では、予測需要が利用可能な容量を超える場合、コントローラは、ステップ（１８３１）に示すように、要求された（ホット）オブジェクトをレプリカ作成候補とみなす。その場合、要求されたオブジェクトはレプリカ作成キューに置かれ、レプリカ配備プロセス（１４００）が呼び出される（図１５に示す）。一方、予測需要が利用可能な容量を下回る場合、コントローラは、ステップ（１８３５）に示すように、まだレプリカを作成する時期ではないと判断する。
【０１０４】
緑のレプリカの不足（図１８の１４１５）は、レプリカ作成プロセス（１３００）に備えるためのトリガにすぎない。新しいレプリカを作成する実際の決定は、ステップ（１８３０）で調べられ、予測需要（１８１０）が図２２に示すように利用可能な容量（１８２０）をかなり上回るときにのみオンにされる（１８３１）。先に述べた通り、平滑化した需要統計により、各オブジェクトの予測需要が安定に予測される。一方、サーバ使用率／意向状態（９９０）（図７等）とそのサーバ容量評価は、おおよそのサーバ容量（例えば残っている緑のサーバの数等）を予測するために使用される。残りの容量を安定に予測するため緑のサーバに照会するには、ＭＯＮ＿ＲＥＱＵＥＳＴメッセージも使用できる。その予測は、コントローラにとっては、レプリカが評価されているオブジェクトの特定の要件

に翻訳された場合にのみ有益である。これにより、そのオブジェクトに対して排他的に予約された場合にサーバが供給できる配備数が別に得られる。図２２の好適実施例では、この数（１８２０）が、要求されたオブジェクトに関連付けられた予測需要（１８１０）により少ない場合、新しいレプリカの作成が決定される（１８３１）。

【０１０５】

レプリカ作成プロセスで、あるオブジェクトに新しいレプリカが必要なことが確認されると、図２０に関して説明しているレプリカ配備プロセス（１４００）が呼び出され、このレプリカの配備を検出できるかどうか判定される。

【０１０６】

図２０は、ステップ（１４３０）に示すように、容量等のファクタをもとにターゲットのグローバル・サーバを見つけるため実装されるレプリカ配備プロセス（１４００）のフローチャートである。レプリカ配備プロセス（１４００）は、ターゲット・サーバを見つける他、ステップ（１４２０）に示すように、レプリカ作成を開始するソース・サーバを見つけようとする。そのためコントローラは、ステップ（１４４０）に示すように、候補のソース・サーバとターゲット・サーバの探索と交渉のプロセスにかかわる。探索と交渉は、好適には、基本的にステップ（１４２０）に戻るプロセスの繰り返してよく、ステップ（１４５０）に示すように、オプションが受け入れられない場合は、新しいソース・サーバとターゲット・サーバが見つけられる。この探索と交渉は、図２３に例を示しているように、レプリカ管理プロトコルにより提供される照会機能（ＲＥＰ＿ＱＵＥＲＹ／ＲＱ＿ＡＣＫメッセージ）を通して達成される。例えば、ソース（１４２０）とターゲット（１４３０）のサーバの検索時、レプリカ配備プロセス（１４００）は、例えば、レプリカを作成するストリーミング・レートの決定（及び対応する帯域幅の要件）等のレプリカ作成オプションを交渉する（１４４０）。レプリカが作成されるオブジェクトはレプリカ作成時に変更することができる。例えば、レプリカを作成するオブジェクトは、図２０に示すオプション交渉ステップ（１４４０）の間に低い品質にダウングレードできる。同様に、オブジェクトは、異なるフォーマットにエンコードしたり、オブジェクトの元のコンテンツとの関係にかかわらずコンテンツで拡張したりできる。

【０１０７】

プロセスは、レプリカ作成オプションに同意すると、ステップ（１４６０）に進み、レプリカ作成リクエストが出される。レプリカ作成リクエストは、好適には、選択されたオブジェクトの別のレプリカを現在保持していない緑のグローバル・サーバに出される。そのような緑のグローバル・サーバが２つ以上存在する場合、好適実施例に従って、要求されているオブジェクトに関連付けられた利用可能な容量等のコスト基準をもとに、最も近いグローバル・サーバに優先権が与えられる。

【０１０８】

好適実施例では、予備不足基準（図１８）により、ほとんどの場合、残る緑のレプリカ（つまり可能性のあるソース・サーバ）は０か１つである。更に、緑のターゲット・サーバが残っていない可能性もある。また、好適実施例の場合、レプリカ配備プロセスでは、ソース・サーバでもターゲット・サーバでもリソースは明示的に予約されない。そうした理由から、本発明の好適実施例は、どのサーバについても（ソースまたはターゲット）、残っている黄／赤の容量の特権的利用に依存する。レプリカ配備プロセス（１４００）は、サーバの範囲外の過剰な容量（つまり黄／緑の容量）に依存して、レプリカ作成リクエスト（１４６０）を出す。すなわち、レプリカ配備プロセス（１４００）は、選択されたオブジェクトに利用できる緑のサーバが見つからない場合は、任意の対応できる黄のサーバにレプリカ作成リクエスト（１４６０）を出すことができる。そのため、サーバ側の受け入れ管理プロセス（１０４０）は、レプリカ作成配備とリクエスト配備を区別できるように拡張する必要がある。或いはまた、残りの緑のレプリカが利用できない場合、レプリカ配備プロセスは、そのような緑のレプリカが利用できるようになるまでレプリカ作成リクエスト（１４６０）を登録しておくこともできる。

【０１０９】

図２０に戻る。ソースとターゲットのサーバが選択された後、レプリカ作成リクエストは、ステップ（１４６０）に示すように両方のサーバに出される。そのようなレプリカ作成リクエストの処理に必要なシグナリングの例を図２３に示す。ステップ（１４７０）でコントローラは、選択されたターゲット・サーバから、レプリカ作成が完了したとの肯定応答を待つ。次にステップ（１４７５）で有効期限がレプリカに割当てられる。これはここではレプリカの存続期限と呼ぶ。存続期限は、グローバル・サーバ上の一時的レプリカの存続時間に（更新可能な）上限を課す。次にステップ（１４８０）でコントローラはそのレプリカ・ディレクトリ（６５６）を更新する。その後、新しいレプリカが将来の配備に利用できるようになる（１４９０）。

【０１１０】

交渉されたオプション（１４４０）によるが、レプリカ作成は時間のかかることがある。従って本発明に従い、レプリカ作成時に出されるリクエストは、（時間的に）延長される、（別のコントローラに）渡される、または所定基準に応じてコントローラにより拒否されることがある。

【０１１１】

当業者には明らかなように、その間、サーバについてレポートされる使用率／意向状態が変わる可能性がある。一時的レプリカが作成されているとき（１４７０）、サーバが利用可能になり（つまり緑になり）、容量が需要を上回る可能性がある。その場合、新しく作成されたレプリカの存続時間により、過剰供給の期間が決まる。すなわち、先に述べたように、レプリカ作成機構は、それが作成する全てのレプリカに有効期限を割当てる。レプリカの有効期限が切れると、そのグローバル・サーバはレプリカの有効期限の更新を要求する。その要求が受け入れられない場合、レプリカはグローバル・サーバから削除され、そのリソースが利用できるようになる。一方、新しいレプリカが利用できる時間までに、利用可能な容量が残っておらず（つまり緑のサーバが見つからない）、需要が容量を大きく上回る可能性もある。その場合、不足時に行われる配備により、別にレプリカ作成監査がトリガされる。そのため、どのようなオブジェクトについても作成されるレプリカの最大数を制限する必要がある。当業者には明らかなように、新しい監査リクエストは所定時間の間に登録できる。その時間が経過した後、予備不足条件を再チェックし、それが続いているかどうか（つまり登録時間の間、緑のレプリカは利用できるようにならないかどうか）判定する。この時間が長すぎる場合、そのオブジェクトに対するリクエストは削除されるか別のコントローラに渡されることは明らかである。

【０１１２】

図２３は、コントローラ（５２０）によりグローバル・サーバ上の一時的レプリカを作成、削除するためのレプリカ管理プロトコル（１２００）の１実施例を示す。コントローラは、新しいレプリカのニーズと配備を確認すると、ソース・サーバ（１７２０等）の探索を開始し、そのようなサーバとのレプリカ作成接続を交渉する。これはＲＥＰ＿ＱＵＥＲＹ（２０２０）／ＲＱ＿ＡＣＫ（２０２５）メッセージの交換として示している。ＲＥＰ＿ＱＵＥＲＹ（２０２０）メッセージは、交渉パラメータとともに、レプリカを作成するオブジェクトのオブジェクトＩＤを含む。

【０１１３】

サーバ（１７２０等）は、ＲＥＰ＿ＱＵＥＲＹ（２０２０）メッセージを受信すると、受け入れ管理を適用し、レプリカ作成接続を受け入れるかどうか判定する。当業者には明らかなように、ＲＥＰ＿ＱＵＥＲＹ（２０２０）メッセージは、機能的にＣＩＤ＿ＱＵＥＲＹメッセージと似ている。ただし前記の通り、レプリカ作成リクエストに対して特権的受け入れ管理（つまり黄の受け入れ）を行うためには、配備とレプリカ作成の照会を区別する必要がある。レプリカ作成リクエスト（２０２０）に特権的受け入れ管理を適用した後、対応するソース・サーバ（１７２０等）はそれぞれその受け入れ応答をＲＱ＿ＡＣＫメッセージ（２０２５）を介してコントローラにリレーする。そのようなサーバ応答（２０２５）はそれぞれ、（ｉ）コントローラ（５２０）によりＲＥＰ＿ＱＵＥＲＹメッセージ（２０２０）に含まれる交渉パラメータの受け入れ、（ｉｉ）交渉、または（ｉｉｉ）コ

ントローラ（５２０）によりＲＥＰ＿ＱＵＥＲＹメッセージ（２０２０）に含まれる交渉パラメータの拒否を示すことがある。

【０１１４】

その時間の間、コントローラはまた、前記のように、有望なターゲット・サーバのセット（１７３０参照）を別のＲＥＰ＿ＱＵＥＲＹ（２０２１）／ＲＱ＿ＡＣＫ（２０２６）メッセージ交換を介して探索する。可能性のあるターゲット・サーバ（１７３０等）は特権的受け入れ管理（前記の通り）をレプリカ作成照会のＲＥＰ＿ＱＵＥＲＹ（２０２１等）に適用し、その受け入れ決定をＲＱ＿ＡＣＫメッセージ（２０２６）を介してコントローラ（５２０）にリレーする。

【０１１５】

コントローラは（５２０）は、ＣＩＤ＿ＱＵＥＲＹメッセージの配備と同様な方法で、ソースとターゲットのサーバからＲＱ＿ＡＣＫメッセージ（２０２５、及び２０２６等）を集め、レプリカ作成配備候補から選択を行う。同様に、作成配備候補が受信されない場合、コントローラはそのレプリカ作成ポリシ（１７６５参照）（図１６）に従って次の探索に進む。

【０１１６】

次に、コントローラ（５２０）がソースとターゲットのサーバを選択すると、ＲＥＰ＿ＬＩＳＴＥＮメッセージ（２０４０）がターゲット・サーバ（１７３０）に送られる。ＲＥＰ＿ＬＩＳＴＥＮメッセージは、レプリカを作成するオブジェクト（オブジェクトＯ１等）、ソース・サーバ（１７２０等）、及びターゲット・サーバ（１７３０等）を識別する。またＲＥＰ＿ＬＩＳＴＥＮメッセージ（２０４０）は、コントローラにより決まるレプリカの存続期限を含む。先に述べた通り、存続期限は、サーバ上の一時的レプリカの存続時間を決定し、これにより、使用されなくなったコンテンツを削除できる。グローバル・サーバ（２０１０）は、ＲＥＰ＿ＬＩＳＴＥＮメッセージ（２０４０）により、ＲＥＰ＿ＬＩＳＴＥＮメッセージ（２０４０）で識別されたサーバ（１７２０等）からのレプリカ作成接続を待機しリスンする。ターゲット・サーバは、このレプリカ作成接続のリソースを設定し、準備ができたことをＲＬ＿ＡＣＫメッセージ（２０４５）を介してコントローラに通知する。

【０１１７】

コントローラ（５２０）は、ターゲット・サーバ（１７３０）がＲＬ＿ＡＣＫメッセージ（２０４５）を介して準備ができたことを通知した後、ＲＥＰ＿ＰＬＡＣＥメッセージ（２０５５）を選択されたソース・サーバ（１７２０）に発行する。ＲＥＰ＿ＰＬＡＣＥメッセージ（２０５５）はソース・サーバに、ターゲットのグローバル・サーバとのレプリカ作成接続を確立することを指示する。ソース・サーバ（１７２０）は、ターゲット・サーバ（１７３０）とのレプリカ作成接続をスケジュールし設定する。

【０１１８】

ソース・サーバ（１７２０）は、レプリカ作成接続を設定した後、レプリカ作成接続の確立をＲＥＰ＿ＡＣＫメッセージ（２０７５）を介してコントローラ（５２０）に通知する。同時にコンテンツのレプリカ作成が、先に交渉されたパラメータで、ＲＥＰ＿ＴＲＡＮＳＦＥＲメッセージ（２０６０）を介して開始される。ＲＥＰ＿ＴＲＡＮＳＦＥＲメッセージ（２０６０）はそれぞれ、ターゲット・サーバで断片化されたコンテンツを再構成するため必要なデータ（シーケンス番号、バイト数等）を含む。

【０１１９】

コンテンツのレプリカが作成された後、ターゲット・サーバ（１７３０）は、新しいレプリカの作成をＲＥＰ＿ＣＯＭＰＬＥＴＥＤメッセージ（２０９０）を介してコントローラ（５２０）に発表する。ＲＥＰ＿ＣＯＭＰＬＥＴＥＤメッセージはレプリカが作成されたオブジェクトのオブジェクトＩＤを含む。コントローラは、ＲＥＰ＿ＳＥＴＬＩＦＥメッセージ（２０８５）を使い、更新とこれに関連する新しい期限を一時的レプリカを持つグローバル・サーバにリレーする。コントローラは、当該グローバル・サーバがＲＳ＿ＡＣＫメッセージ（２０９５）を介して更新の確認応答を返すまで待機する（１５５５）。コ

ントローラは次に、グローバル・サーバ（１７３０）で新しく作成されたこの一時的レプリカを将来の配備に利用するため、そのレプリカ管理ディレクトリ（６５６）を更新する（１４８０）。

【０１２０】

新しい一時的レプリカの作成では、グローバル・サーバにリソースが予約されることはない。その代わり、オンデマンドのレプリカ作成により、同じオブジェクトに対する後のリクエストの間、利用可能な容量が見つかる可能性が大きくされる。

【０１２１】

レプリカは永続的ではなく、存続期限に関連付けられるので、レプリカ作成機構は、それが作成する全てのレプリカに有効期限を割当てる。レプリカの有効期限が切れると、そのグローバル・サーバは、レプリカの有効期限の更新を要求する。この要求が受け入れられない場合、レプリカはグローバル・サーバから削除され、そのリソースが利用できるようになることがある。

【０１２２】

好適実施例に従い、レプリカ管理システムは、オブジェクトに対する需要が大きく減少し続ける間は、容量を節約する。本発明では、レプリカ管理システムは、レプリカを削除する必要があるかどうかを自主的に判定する。特に、サービス・バインディングが終了する毎に、サービス・バインディングに関連付けられたサーバは、ＣＩＤ＿ＣＯＭＰＬＥＴＥＤメッセージをコントローラに送る。とりわけＣＩＤ＿ＣＯＭＰＬＥＴＥＤメッセージにより、コントローラは予備供給過剰チェックをこのレプリカに適用する。また予備供給過剰チェック（１５０５）（図１７）は、レプリカの更新が要求されるときにもサーバによりトリガされる。

【０１２３】

過剰チェックは、好適には、ある特定のオブジェクトについて、利用可能な容量が予測需要を大きく上回るかどうかを判定するため実行する。特に好適実施例では、レプリカの使用率が低い、ホットではないとされるオブジェクトのレプリカが存在する、レプリカの存続期限が切れている等、供給過剰の様々な兆候を認識する。従って、当業者には、ホット・オブジェクトの最大数と一時的レプリカの存続時間の相互作用は明らかであろう。例えば、選択された一時的レプリカの存続時間が長すぎる場合、ホットでなくなる可能性のあるオブジェクトのレプリカがグローバル・サーバに存在する間、現在新しくホットになったオブジェクトのレプリカは、利用可能な緑のグローバル・サーバを待機する可能性がある。

【０１２４】

好適実施例の予備供給過剰チェック（１５０５）を図１７に示す。一時的レプリカは、その存続期限が切れたとされる場合（１５１０）、削除対象になるかどうか監査される。その場合、ステップ（１５１５）でレプリカ削除プロセス（１５００）が呼び出され、より総合的な分析が行われる。他の場合に処理は行われない（１５９９）。

【０１２５】

図２１は、監査されたレプリカを削除する必要があるかどうかを判定するレプリカ削除プロセス（１５００）のフローチャートである。逆に言えば、レプリカ削除プロセス（１５００）は、レプリカを更新する必要があるかどうかも判定する（１５３５）。レプリカ削除プロセス（１５００）は最初、レプリカがホット・オブジェクトに関連付けられているかどうかを確認する（１５２５）。このとき、コントローラは、レプリカの更新を拒否することでレプリカを削除するか、レプリカの存続時間を延長することでレプリカを更新することができる。レプリカがホットである場合、図２２に関して説明しているように総合的な需要対容量チェックが呼び出される（１５３０）。一方、レプリカがホットでない（１５２５）か、需要対容量チェックのとき（１５３０）供給過剰とされた場合、レプリカは削除候補とみなされる。

【０１２６】

実際の削除は、現在の最適な慣習に従い、当該レプリカが続けて２回削除候補とされる（

１５７０）まで延期される（１５６５）。当業者には、この慣習が、ＰｅｔｅｒｓｏｎとＳｉｌｂｅｒｓｃｈａｔｚによりＯｐｅｒａｔｉｎｇ Ｓｙｓｔｅｍｓ Ｃｏｎｃｅｐｔｓ（Ｐｒｅｎｔｉｃｅ Ｈａｌｌ）に説明されているセカンド・チャンス置換アルゴリズムの例であることは明らかであろう。これが、このレプリカが存続する最初のチャンスである場合（１５９０）、レプリカはそこで一時的に更新される（１５４０）。当業者には、２つ目の存続期限（１５５０）を元の期限より短くできることは明らかであろう。ＲＥＰ＿ＳＥＴＬＩＦＥメッセージ（２０８５）（図２３）は、更新とこれに関連する新しい期限を、一時的レプリカを持つグローバル・サーバにリレーするため使用する。コントローラは、当該グローバル・サーバが更新の確認応答をＲＳ＿ＡＣＫメッセージ（２０９５）（図２３）を介して返すまで待機する（１５５５）。存続する２回目のチャンスがレプリカに与えられた場合、レプリカはステップ（１５７０）（図２１）で削除される。コントローラはそのとき、レプリカの更新を取りやめ（１５７５）、これによりグローバル・サーバはレプリカを削除する。レプリカを削除する決定にまで至ると（１５７０）、レプリカ・ディレクトリが更新され（１５８０）、このレプリカが後の配備に利用される。レプリカのデータ構造は、複数のスレッドまたはプロセスがアクセスするので、実施例によっては保護する必要がある。

【０１２７】

当業者には明らかなように、他の機構により、一度に１つのレプリカではなく、また別の形で容量を調整することができる。特に容量の調整は、適合型レート管理の問題であり、現在の最適な慣習は、その範囲内で非対称型の補足方式によっている。例えば、倍数的に増加する（例えば最初に１つのレプリカが作られ、２回目には２つのレプリカ、３回目は４つのレプリカが作られる等）他、線形に減少する（例えば最初は１つのレプリカが削除され、２回目も１つのレプリカが削除される等）方法も使用できる。また、補足作業（つまり作成するレプリカの数）を需要の伸びに合わせることも可能である。そのアプローチでは、作成するレプリカの数は、Ｎ．ＭａｎｏｈａｒらによりＡｐｐｌｙｉｎｇ Ｓｔａｔｉｓｔｉｃａｌ Ｐｒｏｃｅｓｓ Ｃｏｎｔｒｏｌ ｔｏ ｔｈｅ Ａｄａｐｔｉｖｅ Ｒａｔｅ Ｃｏｎｔｒｏｌ Ｐｒｏｂｌｅｍ（Ｐｒｏｃ ｏｆ ＳＰＩＥ ＭＭＣＮ’９８、Ｊａｎｕａｒｙ １９９８）に説明されているような過去の需要と予測需要の違いにもとづいて決定される。しかしその場合でも、レプリカ作成作業を制限するように、レプリカの最大数に制限を課すことが望ましいことは明らかである。

【０１２８】

更に、需要と容量の調整機構の好適実施例は、分散実装形態が可能である。例えば、１実施例では、コントローラ機能を各サーバに置ける。そこで代理コントローラが、サーバの需要と容量の監視作業を行い、重要なしきい値を超えたときに、グローバル・サーバの利用を開始する。特に複数のコントローラ間で同じグローバル・サーバを使用できる。

【０１２９】

当業者には明らかなように、ここで説明しているトリガ管理の実施例は、需要と地理上のパターンがわかっているかまたは安定に予測できる特定の環境に合わせて最適化できる。

【０１３０】

本発明には有益な用途が多数ある。例えば前記の通り、本発明のシステム及び方法は、インターネット・サービス・プロバイダ（ＩＳＰ）が提供し、ケーブル・テレビ・ネットワーク等のブロードキャスト・コンテント・プロバイダをサポートし、需要を、例えばそのケーブル・ネットワークのサーバの容量に動的に適合させる付加価値サービスとして使用できる。ＩＳＰは、必要なら、（ケーブル・ネットワークのコンテンツの）一時的レプリカを、ＩＳＰに示されるような、そのコンテンツに対する需要に関する特性をもとにＩＳＰが所有するグローバル・サーバに配備する。ここではそのような変形例について説明する。

【０１３１】

例えば、本発明は、複数のブロードキャスト・コンテンツ・プロバイダをサポートするとき、ＩＳＰリソースの統計的共有（つまりグローバルに共有されるサーバ）を実現するた

めにも使用できる。これによりＩＰＳコントローラは、収益やコンテンツ・プロバイダの最大損失等のコスト基準に従って、レプリカ作成リクエストの割当てを管理し、通常はＩＳＰの最大利益を保護する。

【０１３２】

独立したコンテンツ・プロバイダは、それぞれ、特定の分野を実装する場合、そのファイアウォールの背後に、代理コントローラと呼ばれるサーバを運用する。代理コントローラは、コンテンツ・プロバイダのために需要と容量の監視を行い、重要なしきい値を超えたとき、ＩＳＰのコントローラにレプリカ作成リクエストを送る。ＩＳＰコントローラはそこで、保留中のレプリカ作成リクエスト間の調停を行い、どのレプリカ作成リクエストをどのグローバル・サーバに割当てるかを決定する。特に代理コントローラ間で同じグローバル・サーバを共有できる。

【０１３３】

当業者には明らかなように、コンテンツ・プロバイダは、セキュリティを理由に、その内部の使用状況統計を開示しようとせず、使用状況統計へのアクセスを許可することもないと思われる。そのため、コンテンツ・プロバイダが、ＩＰＳグローバル・サーバへの特定のレプリカの配備を直接要求できるようにする必要があろう。更に、そのような要求は、地域やターゲットのグローバル・サーバの容量等の特定の要件を満たすため（コントローラではなくコンテンツ・プロバイダにより）条件付きにすることができる。ＩＳＰコントローラはその場合、そのような要件を満足する特定のグローバル・サーバを１つ見つけようとする。例えば、あるコンテンツ・プロバイダは、”次のコンテンツのレプリカを５分以内に作成し、その一時的レプリカを米南東部にあるＨＩＧＨ容量サーバに設定して下さい”といったレプリカ作成リクエストを出す。

【０１３４】

更に、ＩＳＰがレプリカ作成リクエストを適切なグローバル・サーバを持つ他のＩＳＰに渡せるとした契約により、ＩＳＰ間の連携が可能である。例えば、あるコンテンツ・プロバイダが出したリクエストに関連付けられた要件をＩＳＰのグローバル・サーバがどれも満足しない場合に、ＩＳＰは、レプリカ作成リクエストをフレンドリなＩＳＰに渡す等である。

【０１３５】

更に、そのようなレプリカ作成リクエストの配備は、中間パーティにより実装することができる。その場合、中間パーティは、レプリカ作成リクエストの、最も有能且つ適切なパーティへの受け渡しを、コスト、有効期限等の何らかのコスト基準に従って交渉する。

【０１３６】

当業者には明らかなように、需要特性に対応しなくなったレプリカはグローバル・サーバに配備することが考えられる。そのために、レプリカ移行管理をシステム全体で一時的レプリカの安定性とその分散の微調整を受け持つ定期的健全性チェックにより拡張することができる。レプリカ移行プロセスは、ガーベッジ・コレクション及び一時的レプリカの数と配備の最適化を行う。レプリカ移行プロセスは、ガーベッジ・コレクションのプロセスと同じ形で呼び出す。レプリカ移行プロセスは、全ての一時的レプリカのリストを辿り、使用率の低いレプリカを見つける。

【０１３７】

レプリカ移行プロセスは、ホット・オブジェクトの優勢な地域が変化する場合等（例えば、クライアントからのリクエストが米東海岸から西海岸にシフトした場合）地理上の区域でレプリカを移行するのがコスト基準から効率的かどうかを判定することはできない。ただしレプリカの移行は、予測需要に対して地理上の近接性が増したためにネットワーク・コストを下げる場合に利用できる。例えば、すでに配備された一時的レプリカをグローバル・サーバ間で移動するのは、システム全体の容量が少なく、予測需要の分析から将来の需要の特性がシフトすることが明らかになった場合には望ましい。特に過去の需要により、一時的レプリカが地域Ｇ１に配備されている可能性があるが、地域Ｇ３からの予測需要を見ると、一時的レプリカをＧ１からＧ３に移行するのが望ましい場合もある（図９参照

）。従って、その場合、現在の配備を推奨配備と比較したときに、コスト・メリットを予測したコスト基準により、別のグローバル・サーバへのレプリカの移行がトリガされる。

【０１３８】

レプリカ移行プロセスは、例えば一時的レプリカをグローバル・サーバ間で移行できる等、所定基準をもとにした様々なヒューリスティックに依存する。更に、別々のグローバル・サーバ上の２つの一時的レプリカは、マージして１つのグローバル・サーバのレプリカにすることができる。例えば、別のサーバに配備した場合の組み合わせ特性（使用状態、統計等）を予測するため、２つ以上のサーバをマージすると有益なことがある。更に、そのような決定は、需要統計に関するグローバル・サーバの適切性等の所定基準をもとに行える。そのため本発明は、レプリカのマージをレプリカ配備の加法的投影と呼ぶ。

【０１３９】

また、一時的または非一時的レプリカを別の一時的または非一時的レプリカに移行する、つまりあるサーバから別のサーバにレプリカの配備をオフロードすることが望ましい場合もある。例えば、これは、特定のグローバル・サーバ上の容量をオープンにするとき、または、複数のサーバにまたがる一時的レプリカの使用率が低い場合に、それらを低使用率の永続的レプリカにまとめるためには望ましい。

【０１４０】

レプリカの統計の不安定さを避けるため、過去の配備の目的に関して、レプリカ移行手段には慎重なトリガ管理が必要である。当業者には明らかなように、自己規制管理を実装し、前記の移行ヒューリスティクスの安定性管理を評価するためには、前記の手法に加えて、または前記の手法に代えて、オンライン・プロセス最適化やニューラル・ネットワーク等の手法を採用できる。

【０１４１】

まとめとして、本発明の構成に関して以下の事項を開示する。

【０１４２】

（１）独立したクライアントから、ネットワークに分散しマルチメディア・オブジェクトを有するサーバに該オブジェクトを要求するリクエスト管理システムであって、各サーバがマルチメディア・オブジェクトを記憶する容量と、要求されたマルチメディア・オブジェクトをクライアントに配信するストリーミング・リソースとを持ち、該システムは、
前記クライアントから前記マルチメディア・オブジェクトの配備照会を受信し、要求されたオブジェクトには固有ＩＤが関連付けられる、中間管理手段と、
前記所与のオブジェクトＩＤに関連付けられ、前記サーバに記憶されたオブジェクト・レプリカを見つける手段と、
前記オブジェクト・レプリカの前記配備照会を考慮するサーバの意向を判定する手段と、
所定のポリシに従って、意向のあるサーバに対する配備照会を生成する手段と、を含み、
前記中間管理手段は、配備照会を前記意向のあるサーバに送る、
システム。

（２）オブジェクト・レプリカを見つける前記手段は、オブジェクトＩＤと前記サーバでの関連付けられたオブジェクト・レプリカの位置とのマッピングを含む第１ディレクトリ・サービス手段を含む、前記（１）記載のシステム。

（３）サーバの意向を判定する前記手段は、利用できる分散サーバを、配備照会を受信する意向の程度を示すインディケータにマップする第２ディレクトリ・サービスを含む、前記（２）記載のシステム。

（４）前記第２ディレクトリ・サービスにより、分散サーバと、前記サーバで利用できるリソースの量を示す利用可能容量評価インディケータとのマッピングが可能になる、前記（３）記載のシステム。

（５）前記管理手段は、サーバに対するリクエストの一時的配備を生成する配備ポリシを実装し、該一時的配備を決定するため前記第１と第２のディレクトリ・サービスにアクセスする、前記（４）記載のシステム。

（６）前記所定ポリシは、交渉ポリシを含み、前記管理手段は、一時的配備を選択し、関

連付けられた各サーバと交渉し、所定基準を基に該サーバに対するリクエストの配備を可能にする、前記（５）記載のシステム。
（７）前記所定基準は、ストリーミング・オブジェクトの解像度、コスト、ストリーミング・オブジェクトの品質、クライアントとサーバの距離、ストリーミング・オブジェクト受信時の最大遅延を含むグループから選択される基準を含む、前記（６）記載のシステム。
（８）前記管理手段は、サーバに対する複数の配備機会を繰り返し判定する探索ポリシを実装する、前記（６）記載のシステム。
（９）前記管理手段は、有限時間間隔で受信済みリクエスト数を監視し、所定基準を基に該リクエストをリクエスト・グループにまとめる手段を含む、前記（６）記載のシステム。
（１０）有限時間間隔でデマンド・リクエストを総合する前記所定基準は、同じオブジェクトＩＤを要求するクライアントの地理上の近接性を含む、前記（９）記載のシステム。
（１１）有限時間間隔でデマンド・リクエストを総合する前記所定基準は、同じオブジェクトＩＤについて解像度と品質を含めて、要求されたストリーミング制約の共通性を含む、前記（１０）記載のシステム。
（１２）有限時間間隔でデマンド・リクエストを総合する前記所定基準は、同じオブジェクトＩＤに対するリクエストの到着時間の時間的近接性を含む、前記（１０）記載のシステム。
（１３）前記管理手段は、サーバに対する同種のリクエストのグループをマルチキャスト機能でバッチ処理する手段を含み、該サーバはクライアントにマルチキャスト・サービスを提供する、前記（１２）記載のシステム。
（１４）前記管理手段は、統合のためリクエストを自主的にアップグレードまたはダウングレードする手段を含む、前記（１２）記載のシステム。
（１５）オブジェクト・リクエストをアップグレードまたはダウングレードする前記手段は、地理上の近接性によりグループ分けする基準の適用性を加減する、前記（１４）記載のシステム。
（１６）オブジェクト・リクエストをアップグレードまたはダウングレードする前記手段は、要求されたオブジェクトの品質を加減する、前記（１４）記載のシステム。
（１７）オブジェクト・リクエストをアップグレードまたはダウングレードする前記手段は、時間的近接性によりグループ分けする基準の適用性を変更する、前記（１４）記載のシステム。
（１８）前記サーバはそれぞれその利用できるシステム容量を前記管理手段にレポートする、前記（９）記載のシステム。
（１９）前記管理手段は、オブジェクト・レプリカを生成し、前記オブジェクトに対する現在のデマンド・リクエストに従ってサーバに対するオブジェクト・レプリカの配備を動的に管理する、前記（９）記載のシステム。
（２０）前記オブジェクトに対する過去の需要を基にオブジェクト・リクエストに対する将来の需要を予測する手段を含み、オブジェクト・レプリカの前記配備は予測された需要にもとづく、前記（９）記載のシステム。
（２１）前記ディレクトリ・サービスの前記意向インディケータは、前記分散ネットワークで機能するサーバのアグレッシブネス（進取性）の程度を表し、前記サーバのアグレッシブネスをドライブするため使用される、前記（３）記載のシステム。
（２２）前記分散ネットワークのサーバは、１つのエンティティとしてアドレス指定できるサーバ・デバイスを含むサーバ・クラスタを含む、前記（３）記載のシステム。
（２３）前記ネットワークで利用できるオブジェクト・リソースの量を動的に管理する手段を含み、該動的管理手段は、
オブジェクト・リソースに対する需要を予測する手段と、
要求されたオブジェクト・リソースを記憶しストリーミングする前記ネットワークの前記サーバの容量を監視する手段と、

前記オブジェクト・リソースに対する前記予測需要と前記利用できる容量を基に前記オブジェクト・リソースのオブジェクト・レプリカを動的に生成し、需要が増加すると予測されたときは前記ネットワークのサーバに前記オブジェクト・レプリカを自動的且つ一時的に配置し、需要が減少すると予測されたときはレプリカを削除する手段と、
を含む、前記（１）記載のシステム。
（２４）前記予測手段は、要求されたオブジェクト・リソースの需要の密度と量の特徴付けを含め、異なるクライアントからのリクエストを統合して分析する手段を含む、前記（２３）記載のシステム。
（２５）所定基準に従って動的レプリカの配備を管理する手段を含む、前記（２４）記載のシステム。
（２６）要求された特定のオブジェクトについて利用できる容量を予測需要と比較する監査をトリガする手段を含み、該監査は、需要が容量を上回ることを示す容量不足条件または容量が需要を上回ることを示す容量過剰条件を予備判定する、前記（２５）記載のシステム。
（２７）前記生成手段は、前記オブジェクトのレプリカを作成するソース・サーバと、該オブジェクト・レプリカを配備するターゲット・サーバとを見つける手段を含む、前記（２６）記載のシステム。
（２８）ソース・サーバとターゲット・サーバの間のレプリカ作成接続の提供と前記オブジェクトのレプリカを作成するストリーミング・レートを含め、レプリカ作成オプションを交渉する手段を含む、前記（２７）記載のシステム。
（２９）前記レプリカが作成されるオブジェクトには、有効期限が関連付けられ、該有効期限は、前記オブジェクトに容量不足条件が存在する場合には延長され、他の場合、該有効期限に達したときに前記オブジェクト・レプリカが削除される、前記（２８）記載のシステム。
（３０）ネットワークに分散したサーバに記憶されるマルチメディア・オブジェクトの需要を調整する方法であって、各サーバ・デバイスにマルチメディア・オブジェクトを記憶する容量と、要求されたマルチメディア・オブジェクトをクライアントに配信するストリーミング・リソースとを有し、該方法は、
独立したクライアントから前記マルチメディア・オブジェクトに対するリクエストを受信し、要求された各オブジェクトに固有ＩＤが関連付けられる、ステップと、
前記要求されたオブジェクトのＩＤに関連付けられ、前記サーバに記憶されたオブジェクト・レプリカを見つけるステップと、
前記オブジェクト・レプリカに対する前記リクエストに関連付けられた配備照会を考慮するサーバの意向を判定するステップと、
所定ポリシに従って、意向のあるサーバへの前記オブジェクトに対する配備照会を生成し、該意向のあるサーバに配備照会を送るステップと、
を含む、方法。
（３１）オブジェクトＩＤと、前記サーバでの関連付けられたオブジェクト・レプリカの位置とのマッピングを維持するステップを含む、前記（３０）記載の方法。
（３２）サーバの意向を判定する前記ステップは、利用できる分散サーバを、配備照会を受信するサーバの意向の程度を示すインディケータにマップするディレクトリ・サービスを実装するステップを含む、前記（３０）記載の方法。
（３３）サーバの意向を判定する前記ステップは、分散サーバを、前記サーバで利用できるリソースの量を示す利用可能容量評価インディケータにマップするディレクトリ・サービスを実装するステップを含む、前記（３０）記載の方法。
（３４）各分散オブジェクトに関連付けられ、需要の密度と量の特徴付けを含み、優勢な需要分野を示す地域インディケータを含む需要統計を維持するステップを含む、前記（３０）記載の方法。
（３５）前記生成ステップは、一時的配備を選択し、関連付けられた各サーバと交渉して、該サーバに対するリクエストの配備を所定基準を基に可能にし、該所定基準は、ストリ

ーミング・オブジェクトの解像度、ストリーミング・オブジェクトの品質、クライアントとサーバの距離、及びストリーミング・オブジェクト受信時の最大遅延を含むグループから選択される基準を含む、前記（３４）記載の方法。
（３６）有限時間間隔で受信されたリクエストの数を監視し、所定特性を基に該リクエストをリクエスト・グループに統合するステップを含む、前記（３４）記載の方法。
（３７）統合のため、特定のリクエストを自主的にアップグレードまたはダウングレードするステップを含む、前記（３６）記載の方法。
（３８）前記ネットワークのサーバにおけるオブジェクト・リソースの量を動的に管理するステップを含み、該動的管理ステップは、
オブジェクト・リソースに対する需要を予測するステップと、
前記ネットワークの前記サーバの、オブジェクト・リソースを記憶しクライアントにオブジェクトをストリーミングする機能を含む容量を監視するステップと、
前記オブジェクト・リソースの前記予測需要が前記利用可能な容量を超えたときは前記オブジェクト・リソースのレプリカを作成し、前記ネットワーク内に位置するサーバに該オブジェクト・レプリカを一時的に配備するステップと、
前記オブジェクト・リソースの予測需要が減少したときは一時的に配備されているオブジェクト・レプリカを削除するステップと、
を含む、前記（３０）記載の方法。
（３９）前記予測ステップは、要求されたオブジェクト・リソースに対する需要の密度と量の特徴付けを含めて、異なるクライアントからのリクエストを統合して分析するステップを含む、前記（３８）記載の方法。
（４０）所定コスト基準に従ってオブジェクト・レプリカの配備を管理するステップを含む、前記（３８）記載の方法。
（４１）前記レプリカ作成ステップの前に、要求されたオブジェクトに対する予測需要を利用できる容量と比較する監査を開始するステップを含み、該監査は、該要求されたオブジェクトについて需要が容量を超えることを示す容量不足条件を予備判定する、前記（３８）記載の方法。
（４２）前記削除ステップの前に、要求されたオブジェクトに対する予測需要を利用できる容量と比較する監査を開始するステップを含み、該監査は、容量が需要を超えることを示す容量超過条件を予備判定する、前記（４１）記載の方法。
（４３）前記レプリカ作成ステップは、前記オブジェクトのレプリカを作成するソース・サーバと、該オブジェクト・レプリカを配備するターゲット・サーバとを見つけるステップを含む、前記（４２）記載の方法。
（４４）前記レプリカ作成ステップは、ソース・サーバとターゲット・サーバのレプリカ作成接続の提供と、前記オブジェクトのレプリカを作成するストリーミング・レートとを含むレプリカ作成オプションを交渉するステップを含む、前記（４３）記載の方法。
（４５）前記レプリカ作成ステップは、レプリカが作成された各オブジェクトに有効期限を割当てるステップを含み、前記方法は、オブジェクト・レプリカに関連付けられた期限が切れたとき該オブジェクト・レプリカを削除するステップを含む、前記（４４）記載の方法。
（４６）マルチメディア・オブジェクトを記憶し、該オブジェクトのストリーミングをクライアントに提供するサーバを有するインターネット環境にて、マルチメディア・オブジェクト・リソースをリアルタイムに管理する統合システムであって、
オブジェクト・リソースに対する需要を監視し、需要統計とデマンド・リクエスト位置までのサーバの地理上の近接性とを基に将来の需要を予測する手段と、前記インターネット環境でオブジェクト・リソースを記憶し該リソースをストリーミングするリソースを割当てるサーバ・デバイスの現在の意向を確認する手段と、
前記需要統計、前記現在の意向、及びデマンド・リクエスト位置までのサーバの前記地理上の近接性を基に前記インターネット環境のオブジェクト・リソースの容量を調整し、前記オブジェクト・リソースに関連付けられたオブジェクトのレプリカを作成し、該オブジ

ェクト・レプリカを、予測された地理上のデマンド位置にあり利用できる容量を持つサーバ・デバイスに一時的に配備する手段を含む調整手段と、
クライアントから受信されたオブジェクト・リソースに対するリクエスト入力を配備し、受信されたリクエストに関係する配備照会を生成し、前記地理上の位置にあり前記要求されたオブジェクト・リソースを持つサーバ・デバイスに送る手段と、
を含む、システム。
（４７）容量を調整する前記手段は、前記需要に関連付けられた特性を基にオブジェクト・レプリカの数と地理上の配備を評価する手段を含む、前記（４６）記載の統合システム。
（４８）受信されたリクエスト入力を配備する前記手段は、何らかのポリシに従って、意向及び容量のあるサーバにリクエストの受け入れを照会する手段を含む、前記（４６）記載の統合システム。
（４９）ネットワークに分散したサーバに記憶されるマルチメディア・オブジェクトに対する需要を管理する方法のステップを実行するため、機械で読み取ることができ、機械で実行可能な命令のプログラムを具体化したプログラム記憶装置であって、各サーバ・デバイスはマルチメディア・オブジェクトを記憶する容量と、要求されたマルチメディア・オブジェクトをクライアントに配信するストリーミング・リソースとを有し、該方法のステップは、
前記マルチメディア・オブジェクトに対するリクエストを独立したクライアントから受信し、要求された各リクエストに固有ＩＤが関連付けられる、ステップと、
前記サーバに記憶された要求されたオブジェクトのＩＤに関連付けられたオブジェクト・レプリカを見つけるステップと、
前記オブジェクト・レプリカに対する前記リクエストに関連付けられた配備照会を考慮するサーバの意図を判定するステップと、
所定のポリシに従って、意図のあるサーバに対する前記オブジェクトの配備照会を生成し、配備照会を該意図のあるサーバに送るステップと、
を含む、装置。
（５０）サーバの分散ネットワークでオブジェクト・リソースの量を動的に管理するシステムであって、各サーバに、該オブジェクト・リソースを記憶しクライアントにスケジューリングする容量があり、該システムは、
ａ）オブジェクト・リソースに対する需要を予測する手段と、
ｂ）要求されたオブジェクト・リソースを記憶しストリーミングするため前記ネットワークの前記サーバの容量を監視する手段と、
ｃ）前記オブジェクト・リソースに対する前記予測需要と前記利用できる容量をもとに前記オブジェクト・リソースの動的レプリカを生成し、需要が増加すると予想されるときは前記ネットワークのサーバにオブジェクト・レプリカを自動的且つ一時的に配備し、需要が減少すると予想されるときはレプリカを削除する、手段と、
を含む、システム。
（５１）サーバの分散ネットワークでオブジェクト・リソースの量を動的に管理する方法であって、各サーバはオブジェクト・リソースを記憶し提供する容量を有し、該方法は、
ａ）オブジェクト・リソースに対する需要を予測するステップと、
ｂ）前記ネットワークの前記サーバの、オブジェクト・リソースを記憶しオブジェクトをクライアントにストリーミングする機能を含む容量を監視するステップと、
ｃ）前記オブジェクト・リソースに対する前記予測需要が前記利用できる容量を超えるときは前記オブジェクト・リソースのレプリカを作成し、該オブジェクト・レプリカを前記ネットワークにあるサーバに一時的に配備するステップと、
ｄ）前記オブジェクト・リソースに対する予測需要が減少したときに一時的に配備されているオブジェクト・レプリカを削除するステップと、
を含む、方法。
（５２）前記予測ステップは、要求されたオブジェクト・リソースについて、需要の密度

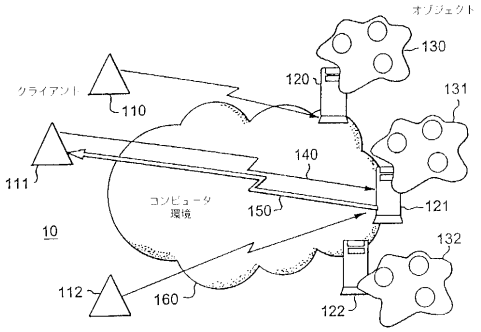
と量の特徴付けを含めて、異なるクライアントからのリクエストを統合して分析するステップを含む、前記（５１）記載の方法。
（５３）所定コスト基準に従ってオブジェクト・レプリカの配備を管理するステップを含む、前記（５１）記載の方法。
（５４）前記レプリカ作成ステップｃ）の前に、要求されたオブジェクトに対する予測需要を利用できる容量と比較する監査を開始するステップを含み、該監査は、該要求されたオブジェクトについて需要が容量を上回ることを示す容量不足条件を予備判定する、前記（５１）記載の方法。
（５５）前記削除ステップｄ）の前に、要求されたオブジェクトに対する予測需要を利用できる容量と比較する監査を開始するステップを含み、該監査は、容量が需要を上回ることを示す容量超過条件を予備判定する、前記（５４）記載の方法。
（５６）前記レプリカ作成ステップｃ）は、レプリカを作成する前記オブジェクトを持つソース・サーバと、該オブジェクト・レプリカを配備するターゲット・サーバとを見つけるステップを含む、前記（５５）記載の方法。
（５７）前記レプリカ作成ステップｃ）は、ソース・サーバとターゲット・サーバのレプリカ作成接続の提供と、前記オブジェクトのレプリカを作成するストリーミング・レートとを含むレプリカ作成オプションを交渉するステップを含む、前記（５６）記載の方法。
（５８）前記レプリカ作成ステップｃ）は、レプリカが作成された各オブジェクトに有効期限を割当てるステップを含み、前記方法は、オブジェクト・レプリカに関連付けられた期限が切れたときに該オブジェクト・レプリカを削除するステップを含む、前記（５６）記載の方法。
（５９）前記オブジェクトに容量不足条件が存在するときは前記有効期限を延長するステップを含む、前記（５８）記載の方法。
（６０）前記レプリカ作成ステップｃ）は、交渉されたオプションに従って前記オブジェクト・レプリカを転送するステップを含む、前記（５７）記載の方法。
（６１）前記分散ネットワークの各オブジェクトのステータスを追跡するステップを含み、該ステータスは、要求されたオブジェクトを記憶するサーバの現在位置とＩＤとを含む、前記（５１）記載の方法。
（６２）前記監視ステップｂ）は、前記サーバが前記サーバへのオブジェクト・レプリカの動的配備を管理できるようにするため、前記サーバ、クライアント、及びコントローラ・デバイスの間でシグナリング・プロトコルを実装するステップを含む、前記（５１）記載の方法。
（６３）サーバの分散ネットワークにあるオブジェクト・リソースの量を動的に管理する方法のステップを実行するため、機械で読取ることができ、機械で実行可能な命令のプログラムを具体化したプログラム記憶装置であって、各サーバはオブジェクト・リソースを記憶する容量と、要求されたオブジェクトをクライアントに配信するストリーミング・リソースとを有し、該方法のステップは、
ａ）オブジェクト・リソースに対する需要を予測するステップと、
ｂ）前記ネットワークの前記サーバの、オブジェクト・リソースを記憶し、オブジェクトをクライアントにストリーミングする機能を含めた容量を監視するステップと、
ｃ）前記オブジェクト・リソースに対する前記予測需要が前記利用できる容量を超えるときは前記オブジェクト・リソースのレプリカを作成し、前記ネットワークにあるサーバに該オブジェクト・レプリカを一時的に配備するステップと、
ｄ）前記オブジェクト・リソースに対する予測需要が減少したときは一時的に配備されているオブジェクト・レプリカを削除するステップと、
を含む、装置。
【図面の簡単な説明】
【図１】クライアント、サーバ、及びサーバに記憶され、クライアントに配信するため要求されることがあるオブジェクトを含む代表的な分散コンピュータ・システムの図である。

【図２】図１の代表的な分散コンピュータ・システムに見られるサーバ・デバイスの構成要素を詳しく示す図である。
【図３】分散集合内の任意のオブジェクトの検出と管理を可能にするオブジェクト・リクエスト・ブローカ・システムを含む代表的な分散コンピュータ・システムの図である。
【図４】本発明の好適実施例に従った、クライアントからのリクエストを処理する中間コントローラ・デバイスを含む分散コンピュータ・システム（１００）の図である。
【図５】コントローラ・デバイスの主な構成要素を示すブロック図である。
【図６】スキーマとデータが関連付けられたレプリカ・ディレクトリの例（６６６）を示す図である。
【図７】スキーマとデータが関連付けられたサーバ・ディレクトリの例（６５６）を示す図である。
【図８】配備管理プロトコルのタイムラインの１例を示す図である。
【図９】グローバル・サーバへのレプリカの配備を動的に管理し、需要と地域の傾向により永続的レプリカと一時的レプリカの区別が促進される分散コンピュータ・システムの図である。
【図１０】サーバがその使用状態を、全てのサーバでコントローラが使用できる正規化した意向インディケータに関連付けるため使用する３色ウォータマーク方式の図である。
【図１１】別のサーバによる同じウォータマーク方式の応用を示す図である。
【図１２】本発明に従ったサーバ・デバイスの変形例を詳しく示す図である。
【図１３】需要統計を生成するためのコントローラから見たリクエストのストリームと有限時間間隔の使用を示す図である。
【図１４】図１３のリクエスト・ストリームについて地域密度インディケータを生成するため好適実施例に用いる方法の例を示す図である。
【図１５】図１３、図１４に従ってコントローラにより記憶される需要統計の例を示す図である。
【図１６】レプリカ管理プロセスとそのリクエスト管理システムとのトリガ・ベースの対話を示す図である。
【図１７】容量調整機構（つまりレプリカ管理システム）をアクティブにするためリクエスト管理システムにより実装される予備供給過剰チェックのフローチャートである。
【図１８】容量調整機構をアクティブにするためリクエスト管理システムにより実装される予備不足チェックのフローチャートである。
【図１９】レプリカ作成プロセスのフローチャートである。
【図２０】レプリカ配備プロセスのフローチャートである。
【図２１】レプリカ削除プロセスのフローチャートである。
【図２２】好適実施例の需要対供給チェックのフローチャートである。
【図２３】レプリカ作成管理プロトコルのタイムラインを示す図である。
【符号の説明】
１００　分散コンピュータ・システム
１１０、１１１、１１２　クライアント
１３０、１３１、１３２　オブジェクトの集合
１４０　クライアント・リクエスト
１５０　ストリーミング接続
１６０　コンピュータ環境
２７１、１０３１、１０４１、１０５１　シグナリング・インタフェース
２８１、２８５　オブジェクト・レプリカ
４２０、４４０　オブジェクトＩＤ
４２１、４２２　レプリカ
５００　オブジェクト・リクエスト
５０１、６０１、６０２、６０３、６０４　リクエスト
５２０、７２０　中間コントローラ

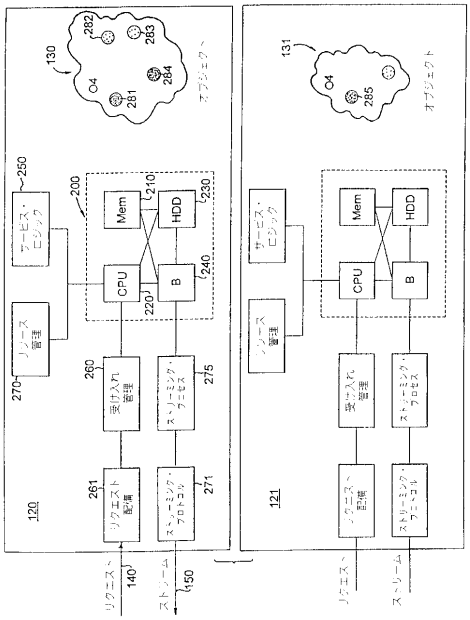
６００　リクエスト処理モジュール
６１０　配備モジュール
６１５　配備ポリシ
６２０　一時的配備照会
６３０　交渉モジュール
６３５　探索ポリシ
６３６　交渉ポリシ
６４０　ポリシ・バンク
６５０　リソース監視モジュール
６５５　サーバ・ディレクトリ・サービス
６５６　サーバ・ディレクトリ
６６０　レプリカ作成管理モジュール
６６５　レプリカ・ディレクトリ・サービス
６６６　レプリカ・ディレクトリ
６７０　管理シグナリング・モジュール
６７１　リソース管理プロトコル
６７２　レプリカ管理プロトコル
６７３　配備管理プロトコル
６８０　需要分析モジュール
６８１　オブジェクト
６９０　容量分析モジュール
６９６　データ構造
７１０　ＣＩＤ＿ＲＥＱＵＥＳＴメッセージ
７３０、７４１　固有ＩＤ
７３１、７４２　パラメータ
７３９、７４０　ＣＩＤ＿ＱＵＥＲＹメッセージ
７６０、７６１　ＣＱ＿ＡＣＫメッセージ
７７０　フラグ
７７３　ＣＩＤ＿ＱＵＥＲＹメッセージ・パラメータ
７９０　ＣＩＤ＿ＰＬＡＣＥメッセージ
７９１　ＣＰ＿ＡＣＫメッセージ
８００　オブジェクトＯ２
８１０　オブジェクトＯ３
８１１、８１２　一時的レプリカ
８２０　オブジェクトＯ１
８３０　Ｇ１ローカル・サーバ
８４０、８５０、８６０　グローバル・サーバ
９００　ウォータマーク方式
９１０　赤の条件
９２０　黄の条件
９３０　緑の条件
９６０、９６１　使用率／意向プロファイル
９９０　意向インディケータ
１０１０　ローカル部
１０１１、１０２１　集合
１０２０　グローバル部
１０３０　レプリカ作成管理プロセス
１０４０、１０４１　受け入れ管理機構
１１５０　リクエスト統計
１１６０　優勢地域インディケータ

１１７０　需要統計
１１８０　ホット・オブジェクト・ブール・フィールド
１１９０　タイムスタンプ
１２００　レプリカ管理プロトコル
１２０１、１２１１、１２２１、１７２０、１７３０　サーバ
１３００　レプリカ作成プロセス
１４００　レプリカ配備プロセス
１４０５　予備不足チェック
１４１５　監査リクエスト
１４４０　所定基準
１４６０　レプリカ作成リクエスト
１５００　レプリカ削除プロセス
１５０５　予備供給過剰チェック
１５５０　存続期限
１７１０　配備リクエスト
１７２０　ソース・サーバ
１７３０　ターゲット・サーバ
１７４５　コスト基準（配備ポリシ）
１７６５　レプリカ作成ポリシ
１８００　レプリカ作成基準
１８１０　予測需要
１８２０　利用可能な容量
２０２０、２０２１　ＲＥＰ＿ＱＵＥＲＹメッセージ
２０２５、２０２６　ＲＱ＿ＡＣＫメッセージ
２０４０　ＲＥＰ＿ＬＩＳＴＥＮメッセージ
２０４５　ＲＬ＿ＡＣＫメッセージ
２０５５　ＲＥＰ＿ＰＬＡＣＥメッセージ
２０６０　ＲＥＰ＿ＴＲＡＮＳＦＥＲメッセージ
２０７５　ＲＥＰ＿ＡＣＫメッセージ
２０８５　ＲＥＰ＿ＳＥＴＬＩＦＥメッセージ
２０９０　ＲＥＰ＿ＣＯＭＰＬＥＴＥＤメッセージ
２０９５　ＲＳ＿ＡＣＫメッセージ

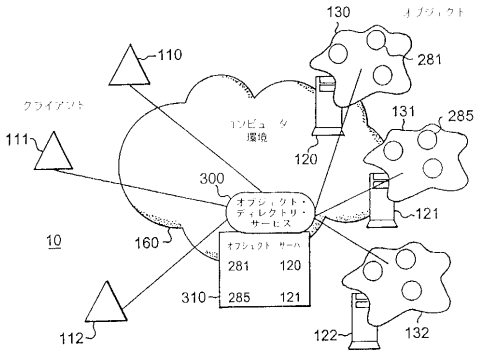
【図１】



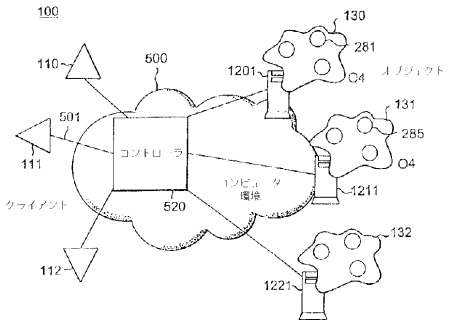
【図２】



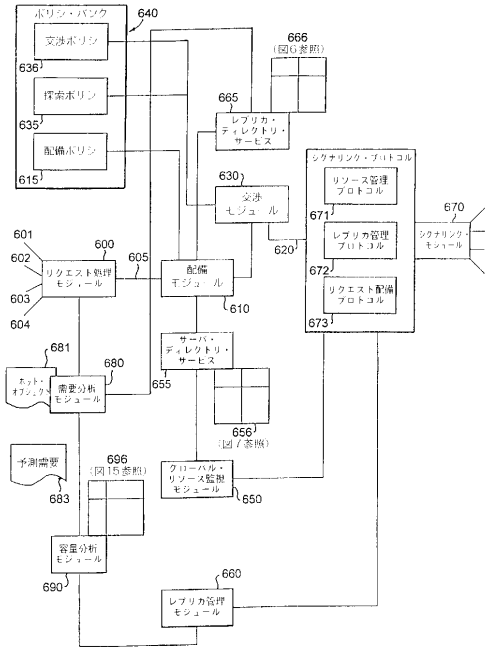
【図３】



【図４】



【図５】



【図６】

666

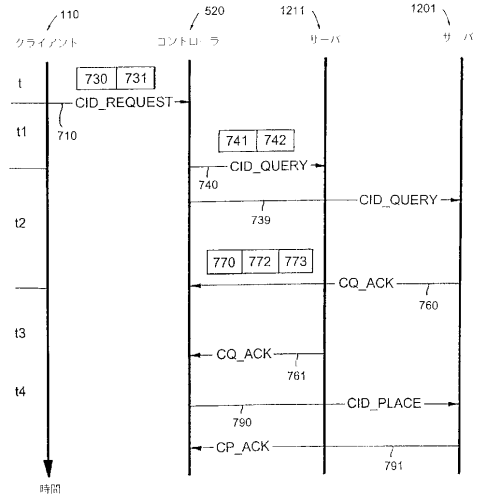
オブジェクトID	レプリカ	サーバ	一時的レプリカ	存続時間
420	421	1211	NO	
	422	1221	YES	060599-133000
440	441	1211	NO	

【図７】

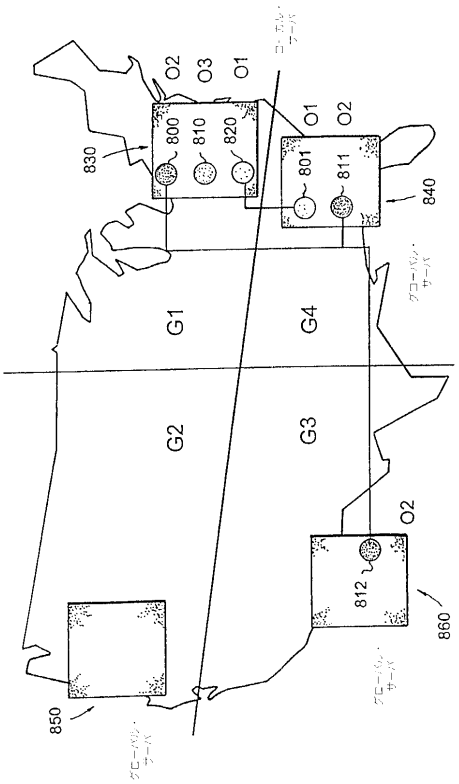
656 657

サーバ	IPアドレス	容量評価	使用状態	タイムスタンプ	クローバリティ
1211	209.09.9.127	Low	赤	t1	ローカル
1221	128.0.0.1	High	緑	t2	クローバル

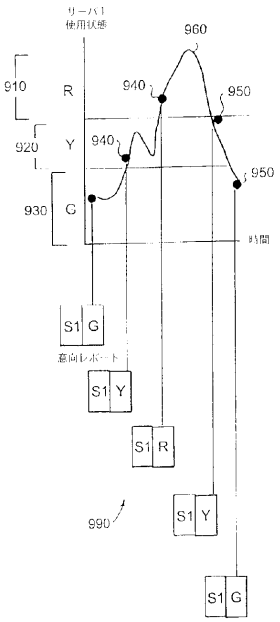
【図８】



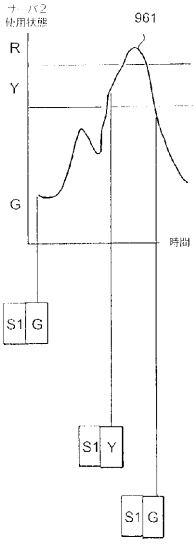
【図９】



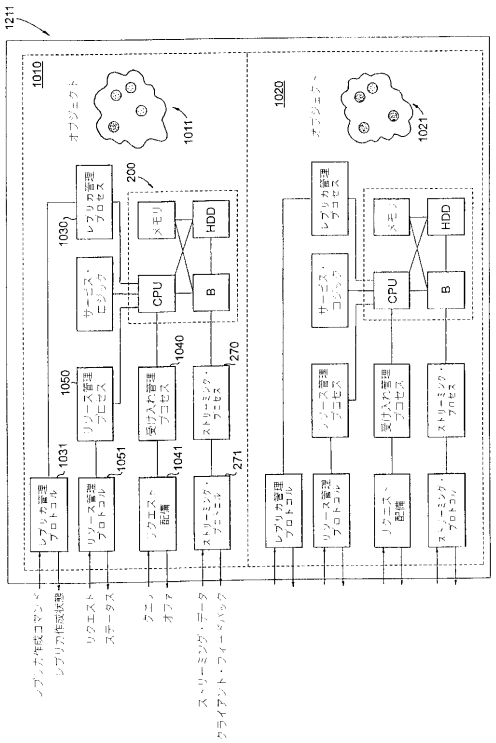
【図１０】



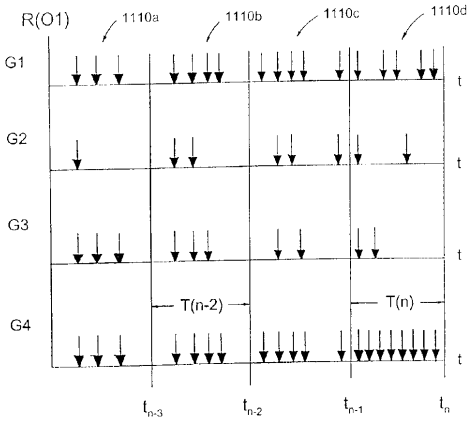
【図１１】



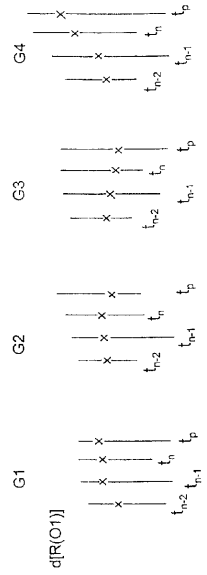
【図１２】



【図１３】



【図１４】

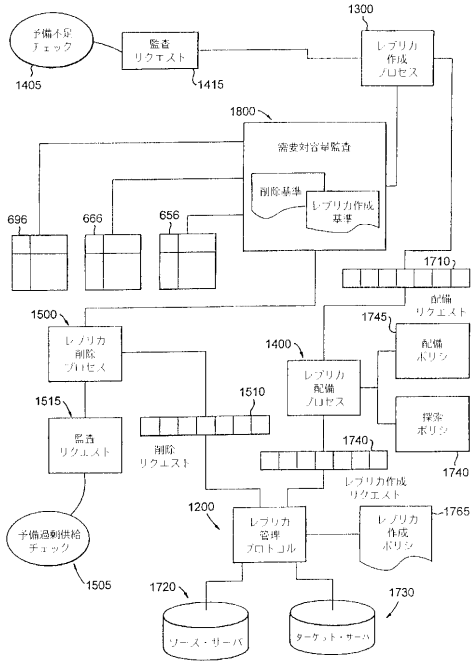


【図１５】

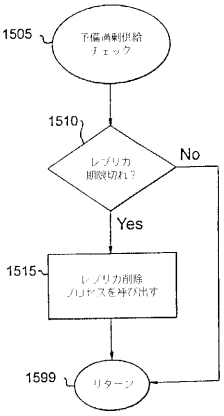
オブジェクトID	需要（リクエスト統計／秒）	リクエスト量／t_1	リクエスト量／t_0	ホット・オブジェクト	タイムスタンプ
420	10	120	60	yes	t_1
425	5	60	55	no	t_1
428	5	30	62	no	t_0

1150 696 1170 1180 1190

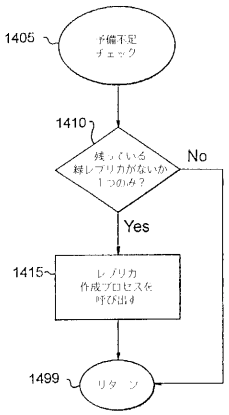
【図１６】



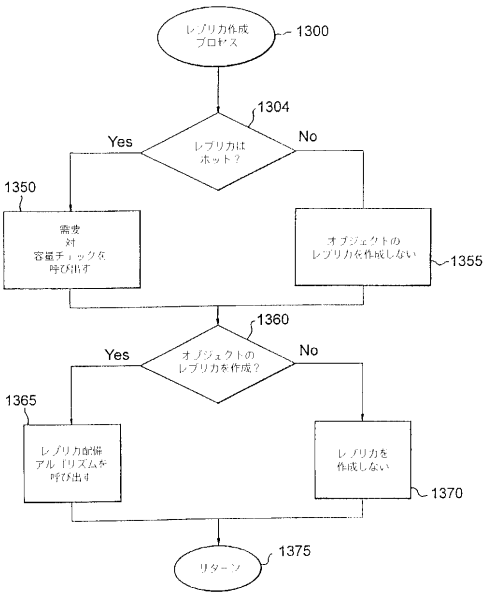
【図１７】



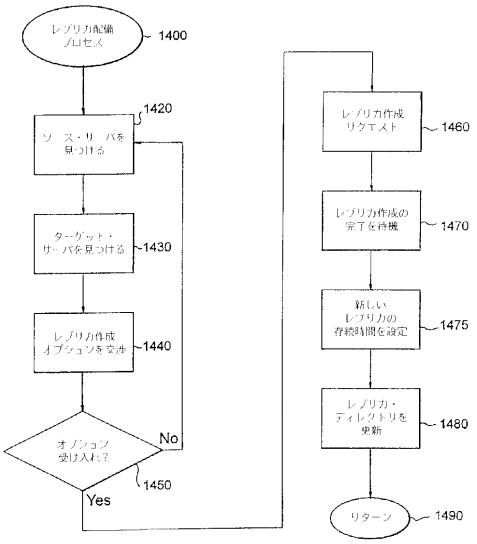
【図１８】



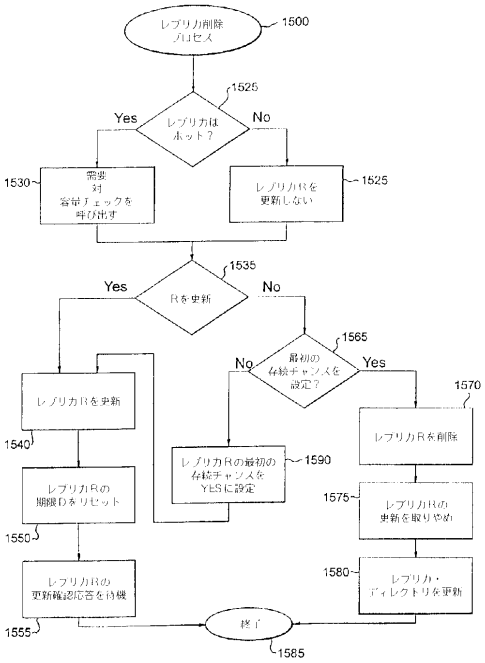
【図１９】



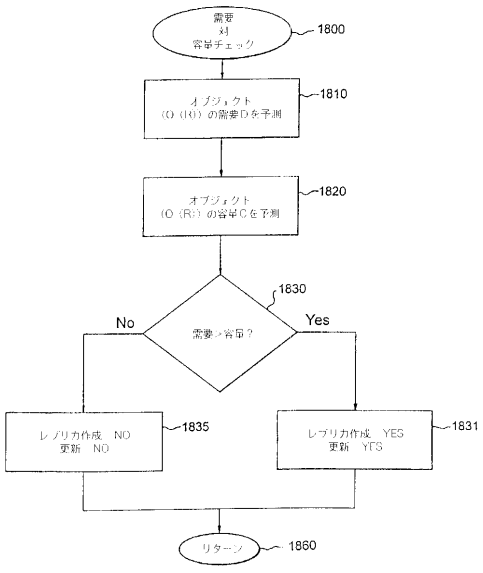
【図２０】



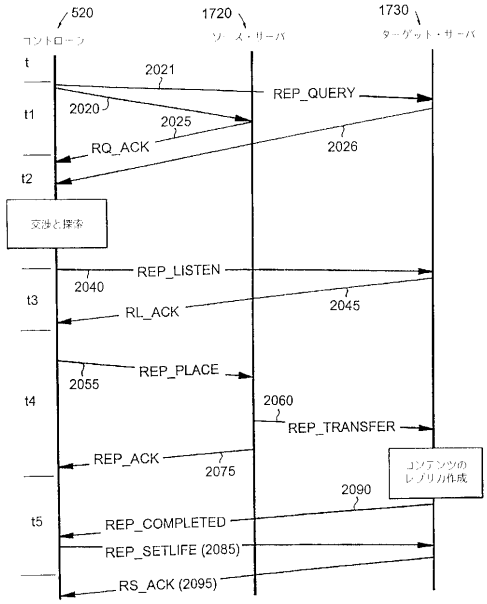
【図２１】



【図２２】



【図２３】



フロントページの続き

(51)Int.Cl.[7] FI
G06F 15/177 674B

(72)発明者 レオン・エル・ルメルスキー
アメリカ合衆国０６９０５、コネチカット州スタンフォード、ガクストン・ロード ３０
(72)発明者 ネルソン・アール・マノアー
アメリカ合衆国１００２４、ニューヨーク州ニューヨーク、エイティセブンス・ストリート、ウエスト ３２７

審査官 野崎 大進

(56)参考文献 特開平１１－０４１５８７（ＪＰ，Ａ）
宇野 哲史 他，バースト型情報転送を用いた映像配信方式の提案，電子情報通信学会技術研究報告，日本，社団法人 電子情報通信学会，１９９８年１２月１７日，Vol.98,No.478(SSE98-152,PS98-42)，pp.19-24.
高倉 健，アクセス負荷の予測を用いた動的コンテンツ再配置方式，情報処理学会論文誌，日本，社団法人 情報処理学会，１９９９年 3月１５日，Vol.40,No.3，pp.1285-1293.
高倉 健 他，分散コンテンツ管理基盤におけるコンテンツ配置の検討，第５５回（平成９年後期）全国大会講演論文集，日本，社団法人 情報処理学会，１９９７年 9月２４日，Vol.3,No.2U-1，pp.3-264～3-265.

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F 17/30 110
G06F 17/30 170
G06F 9/46 360
G06F 12/00 545
G06F 13/00 540
G06F 15/177 674
JSTPlusファイル(JOIS)