(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4989016号
(P4989016)
(45)発行日　平成24年8月1日(2012.8.1)　(24)登録日　平成24年5月11日(2012.5.11)

(51)Int.Cl.　F I
G06F 13/00 (2006.01)　G06F 13/00 540B

請求項の数 32 (全 33 頁)

(21)出願番号	特願2003-500759 (P2003-500759)
(86)(22)出願日	平成14年5月24日 (2002.5.24)
(65)公表番号	特表2004-536389 (P2004-536389A)
(43)公表日	平成16年12月2日 (2004.12.2)
(86)国際出願番号	PCT/US2002/016375
(87)国際公開番号	WO2002/097647
(87)国際公開日	平成14年12月5日 (2002.12.5)
審査請求日	平成17年5月17日 (2005.5.17)
審判番号	不服2009-1921 (P2009-1921/J1)
審判請求日	平成21年1月23日 (2009.1.23)
(31)優先権主張番号	09/872,589
(32)優先日	平成13年5月31日 (2001.5.31)
(33)優先権主張国	米国(US)

(73)特許権者　502303739
オラクル・インターナショナル・コーポレイション
アメリカ合衆国、９４０６５　カリフォルニア州、レッドウッド・ショアーズ、オラクル・パークウェイ、５００
(74)代理人　100064746
弁理士　深見　久郎
(74)代理人　100085132
弁理士　森田　俊雄
(74)代理人　100083703
弁理士　仲村　義平
(74)代理人　100096781
弁理士　堀井　豊

最終頁に続く

(54)【発明の名称】ページを事前に作成するための方法およびシステム

(57)【特許請求の範囲】
【請求項１】
コンピュータによりコードが実行される、情報ページを事前に作成するための方法であって、
規定可能な事前作成のポリシーに従って、他の関連付けられたページに対して相対的に事前作成のための優先順位付けがなされた関連付けられたページの情報内容に対応する情報内容の第１のページを、自動的にかつ周期的に事前に作成するステップを含み、前記第１のページは少なくとも開始ページを含み、かつユーザ用のログインページを含まず、さらに、
ログインすると、ユーザセッションにセッションＩＤを割当てるステップと、
情報要求を受取るステップと、
前記情報要求が前記第１のページに対応しているかどうかを判断するステップと、
前記情報要求が前記第１のページに対応している場合に、前記セッションＩＤを一部として含む事前に作成された前記第１のページを与えるステップと、
前記情報要求が第２のページに対応している場合に、第２のページを作成し、ページ応答上に前記セッションＩＤを置くステップとを含む、方法。
【請求項２】
事前に作成された前記第１のページが古いかどうかを判断するステップと、
事前に作成された前記第１のページが古い場合に、前記第１のページを作成するステップとをさらに含む、請求項１に記載の方法。

【請求項３】
事前に作成された前記第１のページを巡回するステップと、
追加のページが事前に作成されるべきであるかどうかを判断するステップと、
追加のページが事前に作成されるべきであると判断された場合に、追加のページを事前に作成するステップとをさらに含む、請求項１に記載の方法。
【請求項４】
前記第１のページを事前に作成するステップは、
情報をデータベースに照会するステップと、
前記情報を処理するステップと、
処理された情報を、事前に作成された前記第１のページへとパッケージ化するステップとを含む、請求項１に記載の方法。
【請求項５】
システムリソースのレベルは、前記第１のページを事前に作成する動作をスケジューリングする前に考慮され、前記システムリソースのレベルは、ＣＰＵ使用量のレベル、メモリ使用量のレベル、および待機中の事前作成の要求の数からなる群から選択されるリソースの測定値である、請求項１に記載の方法。
【請求項６】
前記規定可能な事前作成のポリシーは、特定のユーザまたはユーザのクラスに適用される、請求項１に記載の方法。
【請求項７】
前記規定可能な事前作成のポリシーは、事前に作成するページを特定する、請求項１に記載の方法。
【請求項８】
前記規定可能な事前作成のポリシーは、責任のパラメータ、アプリケーション識別子、スケジューリングのパラメータ、および／またはリフレッシュ速度のパラメータを含む、請求項７に記載の方法。
【請求項９】
前記第１のページを事前に作成する動作は、他のシステムの作業負荷への干渉を最小にするように自動調整される、請求項１に記載の方法。
【請求項１０】
前記規定可能な事前作成のポリシーは、ポリシーの階層として組織される、請求項１に記載の方法。
【請求項１１】
前記規定可能な事前作成のポリシーは、システムポリシー、アプリケーションポリシー、ユーザポリシー、および／または非常駐のポリシーを含む、請求項１０に記載の方法。
【請求項１２】
情報を事前に作成するためのシステムであって、
規定可能な事前作成のポリシーに従って、他の関連付けられたページに対して相対的に事前作成のための優先順位付けがなされた関連付けられたページの情報内容に対応する情報内容の第１のページの自動的かつ周期的な事前作成を管理する事前作成器を含み、前記第１のページは少なくとも開始ページを含み、かつユーザ用のログインページを含まず、さらに、
情報要求を傍受する傍受器を含み、前記傍受器は、ユーザインターフェイスとコンピュータアプリケーションとの間に論理的に介在し、前記傍受器は、ログインすると、ユーザセッションにセッションＩＤを割当て、前記情報要求が前記第１のページに対応している場合に、前記セッションＩＤを一部として含む事前に作成された前記第１のページを与え、前記情報要求が第２のページに対応している場合に、第２のページを動的に作成し、ページ応答上に前記セッションＩＤを置く、システム。
【請求項１３】
前記事前作成器は、事前に作成するページを特定し、事前に作成するページのリストに

優先順位をつけ、事前に作成する追加のページ用に事前に作成された前記第１のページを巡回し、および／またはシステムリソースをモニタするモジュールを含む、請求項１２に記載のシステム。

【請求項１４】

前記モジュールは、システムリソースのパラメータ、ページの事前作成の時間のパラメータ、ユーザアクセスパターンのパラメータ、および／またはページ深さのパラメータに基づいて、ページの前記リストに優先順位をつける、請求項１３に記載のシステム。

【請求項１５】

前記事前作成器は、事前作成のポリシーにアクセスして、前記第１のページの事前作成を管理する、請求項１２に記載のシステム。

【請求項１６】

前記傍受器は、ウェブサーバまたはキャッシュサーバに統合される、請求項１２に記載のシステム。

【請求項１７】

前記事前作成器および前記傍受器は、第１のネットワークノードと論理的に関連し、前記システムは、第２のネットワークノードと論理的に関連する第２の事前作成器および第２の傍受器をさらに含む、請求項１２に記載のシステム。

【請求項１８】

経路指定する構成要素は、前記第１および第２のネットワークノード間で情報要求を経路指定し、負荷分配器は、前記第１および第２のネットワークノード間で事前作成の作業負荷を分配する、請求項１７に記載のシステム。

【請求項１９】

前記事前作成の作業負荷は、前記第１および第２のネットワークノードのシステムリソースのレベルに基づいて分配される、請求項１８に記載のシステム。

【請求項２０】

前記第１および第２のネットワークノードの各々には、他のノードのリソースレベルに関係なく、その個々のリソースレベルに基づいて、前記事前作成の作業負荷から作業が割当てられる、請求項１９に記載のシステム。

【請求項２１】

事前に作成されたページは、ネットワークアクセス可能な記憶装置に記憶される、請求項１７に記載のシステム。

【請求項２２】

既存のコンピュータアプリケーションに対してコードの変更が行なわれないように、既存のコンピュータアプリケーションで非侵入的に実現される、請求項１２に記載のシステム。

【請求項２３】

前記第１のページは、第１のノードで事前に作成され、前記第２のページは、第２のノードで事前に作成される、請求項１に記載の方法。

【請求項２４】

前記第１のノードは、第２のページにアクセスして前記情報要求を満たす、請求項２３に記載の方法。

【請求項２５】

前記第１または第２のノードのいずれかに前記情報要求を経路指定するステップをさらに含む、請求項２３に記載の方法。

【請求項２６】

第１および第２のページは、ネットワークアクセス可能な記憶装置に記憶される、請求項２３に記載の方法。

【請求項２７】

事前作成の作業負荷は、前記第１および第２のノード間で分配される、請求項２３に記載の方法。

【請求項２８】
前記第１および第２のノードの各々には、他のノードのリソースレベルに関係なく、その個々のリソースレベルに基づいて、前記事前作成の作業負荷から作業が割当てられる、請求項２７に記載の方法。
【請求項２９】
前記情報要求は、セッション識別子を有するユーザからのものである、請求項１に記載の方法。
【請求項３０】
前記規定可能な事前作成のポリシーは、それに対して前記第１のページが事前に作成されるべきアプリケーションを特定する、請求項１に記載の方法。
【請求項３１】
プロセッサによって実行されると情報ページを事前に作成するためのプロセスを前記プロセッサに実行させる複数の命令を含み、前記プロセスは、請求項１から１１および２３から３０のいずれかに記載の方法を含む、コンピュータ読出し可能な媒体。
【請求項３２】
情報ページを事前に作成するためのシステムであって、前記システムは、請求項１から１１および２３から３０のいずれかに記載の方法を実施するための手段を含む、システム。
【発明の詳細な説明】
【技術分野】
【０００１】
背景および概要
この発明は、コンピュータシステムの分野に関する。より特定的に、この発明は、コンピュータシステムで情報を事前に作成する（prefabricate）ための方法およびシステムに関する。
【背景技術】
【０００２】
多くのコンピュータアプリケーションが情報を獲得して、その情報をユーザに表示する。たとえばウェブブラウザは、ウェブページを検索して表示するための、特別に構成されたアプリケーションである。ウェブブラウザのユーザが情報の特定のページを見たいことを示すと、ブラウザか、またはユーザステーションでブラウザを支援するハードウェア／ソフトウェアの構成要素が、情報要求を作成して、その情報要求を情報の位置に送る動作を行なう。この情報要求は、ネットワークを介して情報の位置に送られて遠隔地で処理され、要求された情報を作成／送信する。次に、この要求された情報は、ローカルなユーザステーションに返されて表示される。多層アーキテクチャでは、中間階層ウェブサーバに情報要求を方向付けることができ、この中間階層ウェブサーバが、次いで、１つ以上のバックエンドデータサーバからの情報の要求と、検索と、パッケージ化とを扱う。情報を検索して表示するこのような動作は、分散型データベースアプリケーション等のネットワーク化された多くのコンピュータアプリケーションに関連する。
【０００３】
例示のために、ウェブのポータルサイトに設置される典型的なユーザのホームページを考えられたい。ホームページの一部は、ユーザが所有する株の最新価格およびユーザの株券のポートフォリオの価値の概要を表示するように構成されることが考えられる。ホームページの他の部分は、ニュース、天気等の他の種類の情報を表示するように構成され得る。ユーザのブラウザがホームページを要求すると、１つ以上のバックエンドデータサーバからホームページ上の情報の関連項目を検索して（データベースサーバから株価情報等）、必要であればデータを処理して（ユーザ用に株券のポートフォリオの価値を計算する等）、さらに、そのページ情報を指定されたページフォーマットにパッケージ化して表示する責任を負うウェブサーバに対して、ページ要求が送られる。
【０００４】

従来のコンピュータシステムでは、情報のページを「作成する」このプロセスに関連する固有の遅延が常に存在する。各情報要求は、所与の時間量を消費する複数のネットワーク往復を伴ない得る。実質的な遅延は、ネットワークを介して検索またはダウンロードされなければならない大きな情報項目によっても生じ得る。情報要求が、バックエンドデータサーバでのデータベース処理を伴うことも考えられ、さらに遅延を生じる。加えて、すべての情報が収集されると、その情報は、特定のフォーマットに変換またはパッケージ化されなければならないことが考えられ、さらに遅延を生じる。これらの遅延は、コンピュータアプリケーションの性能または活用性に著しく影響を及ぼすおそれがある。今日のビジネス環境では、このような遅延が、コンピュータのユーザにとって容認することのできない生産性損失の一因となり得る。

【発明の開示】

【発明が解決しようとする課題】

【０００５】

したがって、この発明の目的は、ユーザの情報要求に対する応答速度を高めてかつ応答時間を減じるための方法およびシステムを提供することである。

【課題を解決するための手段】

【０００６】

この発明は、ユーザが要求する情報を検索およびパッケージ化するのに必要なステップがコンピュータシステムによってプリエンプティブに実行される、情報を事前に作成するための方法およびシステムを提供する。ユーザが、後に、情報に対する明示的な要求を行なうと、所望の情報は、その情報を同時に検索およびパッケージ化することに関連する遅延なく、ユーザに直ちに表示され得る。

【０００７】

一実施例において、この発明は、事前作成の要求を生成する事前作成器を含む。事前作成の要求は処理されて、事前に作成される情報を作成して記憶する。傍受器がユーザから情報要求を受取り、その情報要求に応えて、事前に作成された情報を与える。この発明の事前作成の構成要素は、ハードウェア構成要素を何ら追加して設置する必要なしに、任意の既存のコンピュータアプリケーションと共に、非侵入的に用いることができる。

【０００８】

この発明の事前作成システムは、コンピュータシステム内に存在する状況の変化に対してシステムの処理を自動的に同調させるように構成することができる。これにより、事前作成のプロセスは、リソースの使用量が少ない期間中には、事前作成の作業負荷を自動的に増やすことによって、システム内で利用することのできるどのような余剰リソースも最大限利用することができる。リソースの使用量が多い期間中には、事前作成の作業負荷が自動的に減らされて、システムが実行している他のランタイム作業に対するどのような悪影響をも最小にする。

【０００９】

この発明をスケーラブルに配備して、分散した複数の計算ノードに対し、調整された事前作成サービスを行なうことができる。多ノードの環境におけるこの発明の一実施例によると、任意の数の事前作成器のインスタンスが、別個のネットワークノードに対して事前作成機能を実行するように構成され得る。事前作成器のインスタンスの各々は、事前に作成された情報のうちの割当てられた部分を処理して、事前に作成された情報を、ネットワークアクセス可能な記憶装置に記憶する。ユーザが情報要求を行なうと、事前作成のフレームワーク内の任意のノードは、要求された情報が事前に作成されているかどうかを判断して、事前に作成された情報をユーザのために取出すことができる。作業負荷を複数のノード全体にインテリジェントに分配することによって、事前作成器の全ノードのシステムリソースを最適な態様で制御して利用し、かつ、動的に調整することができる。全ノードは、所与のノードに割当てられた事前作成の作業負荷の量に関係なく、事前に作成された情報の恩恵を等しく受ける。なぜなら、事前に作成された情報はすべて、たとえ他のノードによって事前に作成された情報であっても、ネットワーク化された記憶装置から一様に

アクセス可能なためである。

【００１０】

この発明の一実施例は、事前作成サービスに対するセッションの安全性を管理する方法を提供する。この実施例の傍受器は、ユーザに対するセッションＩＤ値の有効性を確認するように構成される。ユーザが、事前に作成された情報によって叶えられ得る情報要求を行なうと、傍受器は、事前に作成された情報に対し、その情報に接続されたユーザに対する有効なセッションＩＤを与える。

【００１１】

この発明の局面、目的および利点のさらなる詳細は、詳細な説明、図面および請求項において以下に説明される。

【００１２】

添付の図面は、この発明のさらなる理解をもたらすように、および、詳細な説明とともにこの発明の原理を説明する働きをするように、含まれる。

【発明を実施するための最良の形態】

【００１３】

詳細な説明

この発明は、ユーザが要求する情報を検索およびパッケージ化するのに必要なステップが、ユーザが明示的な情報要求を行なう前にコンピュータシステムによってプリエンプティブに実行される、情報を事前に作成するための方法およびシステムの形をとる。むしろ、この発明の実施例は、今後ユーザに表示され得る情報を予測的に特定して、その情報を事前に作成する。ユーザが、後に、情報の要求を行なうと、所望の情報は、その情報を同時に検索およびパッケージ化することに関連する遅延なく、ユーザに直ちに表示され得る。この説明は、限定のためではなく例示のために、ユーザに表示される情報の「ページ」に関して行われる。しかしながら、この発明の範囲内において、他の細かさの情報を事前に作成してよいことに注目されたい。

【００１４】

一実施例によると、コンピュータアプリケーションに対し、ページを事前に作成するために、１組の事前作成システムの構成要素が配備される。図１を参照すると、一般に、情報を生成または処理してユーザに表示するコンピュータアプリケーション１０４が示される。コンピュータアプリケーション１０４は、たとえばデータベースアプリケーション等の、ユーザに情報を表示するどのようなアプリケーションであってもよい。コンピュータアプリケーション１０４からの情報ページは、たとえば従来のウェブブラウザであってよいユーザインターフェイス１０２を介してユーザに表示される。ユーザインターフェイス１０２は、ユーザの指令に応えて、ユーザが要求した情報ページに対するページ要求を生成する。

【００１５】

傍受器１０８が、ユーザインターフェイス１０２とコンピュータアプリケーション１０４との間に論理的に介在する。ページ要求がユーザインターフェイス１０２からコンピュータアプリケーション１０４に送られると、傍受器１０８は、そのページ要求を傍受して、その要求に応えてかつ有効な、事前に作成されたページが存在するかどうかを判断する。この判断は、要求されたページが既に事前に作成されて、記憶装置１１２にキャッシュされ／記憶されているかどうかを識別することによって行われる。記憶装置１１２は、事前に作成されたページのアクセス可能なコピーを記憶するのに用いることのできる、メモリキャッシュ、ローカルディスクドライブ、またはネットワークファイルシステム（「ＮＦＳ」）装置等のどのような装置であってもよい。

【００１６】

要求されたページが既に事前に作成されている場合、記憶装置１１２から事前に作成されたページを取出して、ユーザインターフェイス１０２で表示することによって、ページ要求は直ちに叶えられる。要求されたページがまだ事前に作成されていない場合、ページ要求は、コンピュータアプリケーション１０４に送られて、要求されたページを動的に作

成することによって叶えられる。

【００１７】

事前作成器１１０は、事前に作成する情報ページを特定する。構成設定および／または発見的方法を用いて、どの情報ページが事前に作成されるべきであるかを予測的に判断することができる。特定されたこれらのページの各々に対し、事前作成器のサービスモジュール１１０から傍受器１０８に、ページ要求が送られる。傍受器１０８は、これらのページ要求をコンピュータアプリケーション１０４に送る。事前作成器１１０からのこれらのページ要求の各々は、コンピュータアプリケーション１０４によって処理されて、要求された情報ページを作成する。コンピュータアプリケーション１０４における、ページ要求に応えるプロセスは、そのページ要求が事前作成器のサービスモジュール１１０から生じたか、またはユーザインターフェイス１０２から生じたかに関係なく、同じであり得る。しかしながら、事前作成器１１０からの、事前に作成されたページに対する要求に応えて生成される情報ページは、傍受器１０８によって、たとえば記憶装置１１２においてキャッシュされる。これとは対照的に、ユーザインターフェイス１０２からの現時点でのページ要求に応えて動的に作成された情報ページは、ユーザインターフェイス１０２に直ちに送られて、表示される。同時に作成されたこれらのページもまた、後のアクセスに備えて、傍受器１０８によってキャッシュされ得る。

【００１８】

この発明の事前作成の構成要素は、コンピュータシステム内のどのような物理的な位置または論理的な位置にもインストールすることができる。物理的に、傍受器および事前作成器の構成要素は、たとえば、ユーザインターフェイスおよびコンピュータアプリケーションの物理的な位置、または中間階層ウェブサーバコンピュータ等の他のネットワーク位置に配置されてよい。論理的に、傍受器および事前作成器の構成要素は、コンピュータシステム内のどこにでもインストールすることができ、および／または他のシステム構成要素に統合することができる。一実施例において、傍受器１０８は、中間階層ウェブサーバに統合されるか、または、論理的に、中間階層ウェブサーバの前に配置されて、ユーザインターフェイス／ブラウザと、バックエンドデータサーバ上で稼動してユーザによってアクセスされる任意のアプリケーションとの間の通信のすべてを傍受する。

【００１９】

コンピュータアプリケーション１０４に対して行われるページ要求のフォーマットが、システム内で事前作成が行われているかどうかに関係なく同じままであり得ることに注目されたい。傍受器の構成要素は、ユーザインターフェイスからの通常のページ要求と事前作成器の構成要素からのページ要求とを区別して、それに応じ、作成されたページをコンピュータアプリケーションから経路指定するように構成される。したがって、コンピュータシステム内に事前作成の構成要素を導入しても、コンピュータアプリケーションに変更を生じない。むしろ、この発明を用いて、通信経路において（ウェブサーバにおいて等）コンピュータアプリケーションとそれに対応するユーザインターフェイスとの間に傍受器の構成要素を単に介在させることによって、コンピュータアプリケーションにどのような変更も、または著しい変更も加える必要なしに、どのようなコンピュータアプリケーションに対しても非侵入的にページを事前に作成することができる。

【００２０】

図２はこの発明の一実施例に従った、ページの事前作成を行なうための方法のフロー図を示す。この方法のステップ２０２は、システムに１つ以上の事前作成のポリシーを構成することに向けられる。事前作成のポリシーは、事前作成が行われる態様を支配する１組の構成パラメータである。この発明の一実施例によると、事前作成のポリシーは、以下のもの、すなわち、（１）ポリシーが適用されるコンピュータアプリケーション、（２）ポリシーが適用されるユーザ、（３）事前に作成されるべき情報ページ、（４）事前作成のスケジューリングおよびリフレッシュ間隔、（５）事前に作成する責任、および（６）事前作成の優先順位、の一部またはすべてを特定する。

【００２１】

例示するために、以下のものが、コンピュータアプリケーションに対する１組の事前作成のポリシーのパラメータの一例を示す。

【００２２】

１．ポリシー名：ゲストユーザポリシー

２．アプリケーション名：Foo

３．適用対象：ゲスト責任を有する、Fooアプリケーションの全ユーザ

４．事前作成の時間および周期性：午前２：００－毎日リフレッシュ

５．事前作成の深さ：特定されたユーザに対する開始ページおよびメニューページのすべて

これは、「Foo」というコンピュータアプリケーションに「ゲスト」責任を有するユーザに対する事前作成のポリシーを規定する、アプリケーションレベルのポリシーである。特に、この組の（「ゲストユーザポリシー」と名付けられた）ポリシーは、Fooアプリケーションのこのようなユーザの各々に対する「開始ページ」およびメニューページが毎日午前２：００に事前に作成されて記憶されることを明記している。この発明の一実施例によると、開始ページは、ユーザに表示される最初のページである。開始ページは、アプリケーションの最初に表示されるページ、組織のホームページ、ユーザのホームページ、またはユーザに表示されるべきエントリページとして指定された他の任意のページであり得る。開始ページは、しばしば、他の表示可能な情報ページを見るための、ハイパーリンクまたはユーザによる選択可能な制御を含む。「メニューページ」は、開始ページからハイパーリンクされた、タブ付きのメニューページであり得る。

【００２３】

再び図２を参照すると、その後、この方法は、規定された事前作成のポリシーに従って、ページを事前に作成する（２０４）。ページを事前に作成する厳密なステップは、大部分が、ページの特定の内容およびページ内の情報の位置に依存する。一実施例では、現時点のユーザのページ要求に応えてページを動的に作成するのに用いられたのと同じステップが、ここでも用いられて、事前作成の要求に応えてページを事前に作成する。データベースのデータにアクセスするアプリケーションに関して、事前作成は、データベースを照会して、データベースから取出したデータを処理して、結果的に得られた情報を特定のフォーマットにパッケージ化／変換してユーザに表示するステップを含み得る。事前作成のプロセスの目標は、ユーザが後に事前作成せずに「通常通り」取出すであろうページにできるだけ類似したページを、事前に作成することである。次に、事前に作成されたページは、後のアクセスに備えてキャッシュされ／記憶される。

【００２４】

動作の際に、事前作成が可能なコンピュータシステムは、ページ要求をフィルタ処理して、事前に作成されたページに対応するあらゆるページ要求を傍受する（２０８）。ページ要求に応えた、既存であってかつ有効な、事前に作成されたページが存在する場合、そのページはユーザに送られて表示される（２１０）。要求されたページが存在しないか、または、ページ要求に応える、事前に作成されたページが有効ではない場合、ページ要求は、同時作成のためにコンピュータアプリケーションに送られる（２１２）。事前に作成されたページを、たとえばそれが古くなった場合等の或る状況下で、無効であるとマークできることに注目されたい。「古い」という用語は、事前に作成されたページ全体、または事前に作成されたページ上に配置された情報の項目が、もはや有効ではないと考えられる、事前に作成されたページを指す。一実施例において、古さは、事前に作成されたページに対する作成／更新時間がしきい期間内にあるかどうかに基づく。事前に作成されたページが古くなることを避けるために作成または更新されるべき周期性を決定するために、パラメータを確立することができる。他の手法を用いて、古さを確立することもできる。たとえば、代替的な手法は、事前に作成されたページ上の情報の一部またはすべてが、関連する態様で変更されているかどうかに基づく。事前に作成されたページ上の情報の一部またはすべてが変化したかどうかを判断するために、メッセージ送信を用いることができる。バックエンドデータベースは、事前に作成されたページの位置または管理プログラム

にメッセージを送信して、古さを表示することができる。代替的に、キャッシュサーバ／事前作成器の構成要素自体がバックエンドサーバにメッセージを送信して、事前に作成されたページが古いかどうかを判断する。

【００２５】

例示的な実施例

図３は、この発明の一実施例に従った、ページを事前に作成するためのアーキテクチャ３００を示し、ここでは、傍受器のサービスモジュール３０４がウェブサーバ３０６に統合される。一例として、ウェブサーバ３０６は、アパッチ（Apache）Jservモジュールを用いた、サーバ側のJava（Ｒ）サーブレットの機能性を有するアパッチウェブサーバ（www.apache.orgから入手可能）として実現することができる。傍受器のサービスモジュール３０４は、アパッチウェブサーバのコードベースにコンパイルされるｃベースの拡張として実現することができる。ユーザのページ要求は、ブラウザ３０２から生じる。事前作成器のページ要求は、事前作成器のサービスモジュール３１０によって生成される。事前作成器のサービスモジュールは、ウェブサーバ３０６と同じハードウェアの位置か、または、コンピュータシステム内の他の任意の機器の位置のいずれかに配置することができる。一例として、事前作成器のサービスモジュールは、ウェブサーバ３０６とともに配置されたJava（Ｒ）プログラムとして実現することができる。

【００２６】

ページ要求は、統一リソースロケーター（Uniform Resource Locator）（ＵＲＬ）要求として傍受器のサービスモジュール３０４に送られる。ユーザのページ要求に対し、傍受器のサービスモジュール３０４は、要求されたＵＲＬに対するＨＴＴＰ応答が、既に事前に作成されて記憶されているかどうかを判断する。事前に作成されたＨＴＴＰ応答は、記憶装置３０８に記憶されている。

【００２７】

ユーザが要求したＵＲＬに対して、事前に作成されて有効なＨＴＴＰ応答を利用することができない場合、ＵＲＬ要求がJava（Ｒ）サーブレット３１６によって動的に処理されて、所望のＨＴＴＰ応答を生成する。Java（Ｒ）サーブレット３１６は、データベース３１２にアクセスして、ＵＲＬに関連する情報を取出すか、または処理することができる。Java（Ｒ）サーブレット３１６によって生成されたＨＴＴＰ応答は、傍受器のサービスモジュール３０４によってブラウザ３０２に経路指定される。

【００２８】

事前作成器のサービスモジュール３１０は、１組の事前作成のポリシー３１４にアクセスして、事前に作成されるべきＵＲＬを特定する。事前作成のポリシー３１４は、データベース３１２に記憶することができる。事前に作成されるべきであると特定されたＵＲＬの各々は、事前作成ページの要求として、傍受器のサービスモジュール３０４に送られる。事前作成器のＵＲＬ要求は、１つ以上のJava（Ｒ）サーブレット３１６によって処理されて、所望のＨＴＴＰ応答を生じる。傍受器のサービスモジュール３０４は、ユーザのページ要求からのＨＴＴＰ応答と、事前作成器のページ要求からのＨＴＴＰ応答とを区別する。Java（Ｒ）サーブレット３１６によって生成された、事前作成器のページ要求に対するＨＴＴＰ応答は、それらが事前作成器のサービスモジュール３１０に送り返される前に、記憶装置３０８に記憶される。

【００２９】

図４は、事前作成器のサービスモジュール３１０の一実施例の構成要素を示す。開始ローダ４０４が、事前作成器のサービスモジュール３１０に対する最初のブートストラッピングを行なう。開始ローダ４０４は、このシステムのための、規定された事前作成のポリシー４０２にアクセスして、事前に作成するページの最初の組を決定する。事前作成の候補である、特定されたページの各々が、フォーマット化されたページ要求として示されて、この明細書ではページ要求ブロック（ＰＲＢ）と呼ばれる。以下により詳細に説明するように、特定されたページの最初の組は、事前に作成されるべき他の候補ページへのリンクをさらに特定することができる。

【００３０】
１つ以上のＰＲＢが、開始ローダ４０４から便益解析器４０６に送られ、各ＰＲＢは、事前に作成される別個のページに対応する。便益解析器４０６は、事前作成のポリシー４０２に確立された優先順位付けパラメータに基づいて、ＰＲＢのリストに優先順位を付ける。優先順位付けパラメータの例には、利用可能なシステムリソース、ページの作成回数、ユーザのアクセスパターン、およびＰＲＢページのページ深さが含まれる。便益解析器はまた、ＰＲＢを処理するための好ましいバッチサイズも決定する。便益解析器が、処理のために、ＰＲＢの組に優先順位を付けてバッチ化されたＰＲＢの組を生成すると、ＰＲＢの組はページ要求フィーダ４０８に送られる。
【００３１】
ページ要求フィーダ４０８は、便益解析器４０６からＰＲＢを受取り、受取ったＰＲＢに基づいてＨＴＴＰ要求を発行する。ＰＲＢに対するＨＴＴＰ要求は、傍受器のサービスモジュールに送られて、ウェブサーバによって処理される。ＨＴＴＰ応答は、傍受器のサービスモジュール／ウェブサーバからページ要求フィーダ４０８に送られる。各ＰＲＢは、それに関連するＨＴＴＰ応答とともに、ＵＲＬコレクタ４１２に送られる。
【００３２】
ページ要求フィーダ４０８は、並行して作動する１つ以上の処理実体（スレッド等）にＰＲＢを割当てることによって、複数のＨＴＴＰ要求を並行して処理するように構成される。並行して処理されるＰＲＢの数は、システム内で利用することのできるリソースのレベルに依存する。利用することのできるシステムリソースのレベルが高いほど、システムで並行して処理することのできるＰＲＢの数は増える。事前作成のポリシー４０２は、事前作成のプロシージャが実行され得るシステムリソースのレベルを支配するか、または、事前作成のポリシーによって消費され得るシステムリソースの量を制限するパラメータを含み得る。
【００３３】
リソース管理プログラム４１０は、事前作成器の位置だけでなくウェブサーバの位置の両方における、ＣＰＵ等のシステムリソースの使用量と、メモリの使用量とをモニタして、事前作成器のサービスモジュールが利用することのできるシステムリソースのレベルを示す。リソース管理プログラム４１０は、ページ要求フィーダ４０８が用いるシステムリソース情報を与えて、ＨＴＴＰ要求が傍受器／ウェブサーバに送られる速度を下げる。リソース管理プログラム４１０はまた、ページ要求フィーダ４０８が用いる情報を提供して、負荷バランサーまたはウェブサーバのどちらにＨＴＴＰ要求が方向付けられるべきかを特定する。
【００３４】
この態様において、事前作成システムは、システム内に存在する状況の変化に対し、その処理を自動的に適合させるか、または「同調させる」。これにより、事前作成プロセスは、リソースの使用量が少ない期間中に、事前作成の作業負荷を自動的に増やすことによって、システム内で利用することのできる余剰リソースを最大限利用することができる。リソースの使用量が多い期間中は、事前作成の作業負荷を自動的に減らして、システムが行なっている他のランタイム作業から、リソースが流出することを最小限に抑える。
【００３５】
処理されたＰＲＢの各々と、傍受器／ウェブサーバからの、それに関連するＨＴＴＰ応答は、ページ要求フィーダ４０８からＵＲＬコレクタ４１２に送られる。ＵＲＬコレクタ４１２は、ＨＴＴＰ応答を巡回して（crawl）、事前に作成されるべき新規のＵＲＬを特定する。この新規のＵＲＬは、優先順位付けおよびさらなる処理のために便益解析器４０６に送られる追加のＰＲＢを形成する。事前作成のポリシー４０２は、事前に作成されるべき追加のＵＲＬを特定するために、ＵＲＬコレクタ４１２が用いるパラメータを規定する。一実施例では、ユーザの即座の要求に応えて動的に作成されたページもまた、ＵＲＬコレクタ４１２に送られて、事前に作成されるべきページを求めて巡回が行なわれ得る。
【００３６】

事前作成器のサービスモジュール３１０の動作を示すために、以下のポリシーパラメータを有する、上述の例示的な事前作成のポリシーを再び考えられたい。

【００３７】

１．ポリシー名：ゲストユーザポリシー

２．アプリケーション名：Foo

３．適用対象：ゲスト責任を有する、Fooアプリケーションの全ユーザ

４．事前作成の時間および周期性：午前２：００－毎日リフレッシュ

５．事前作成の深さ：特定されたユーザに対する開始ページおよびメニューページのすべて

以前に述べたように、これは、「Foo」というコンピュータアプリケーションに対する「ゲスト」責任を有するユーザに対する事前作成のポリシーを規定する、アプリケーションレベルのポリシーであり、Fooアプリケーションのこのようなユーザの各々に対する「開始ページ」および「メニューページ」は、毎日午前２：００に事前に作成されて記憶される。

【００３８】

この事前作成のポリシーが実行されると、事前作成器のサービスモジュール３１０内の開始ローダ４０４は、Fooコンピュータアプリケーションの、関連する全ユーザ（すなわち「ゲスト」ユーザ）に対する開始ページのＵＲＬを特定する動作を実行する。或る状況において、開始ローダ４０４はまた、関連するユーザに対するメニューページのＵＲＬの一部またはすべてを特定することもできる。特定されたＵＲＬに対応するＰＲＢは、便益解析器４０６に送られる。

【００３９】

便益解析器４０６は、ＰＲＢに優先順位を付けて、一連のＰＲＢのバッチをページ要求フィーダ４０８に送る。優先順位を付けるさまざまなプロシージャを用いてよい。この例において、適用することのできる優先順位付けのプロシージャは、開始ページのＵＲＬの方に高い優先順位を与え、ページ階層においてより深い他のＵＲＬ、すなわち、メニューページのＵＲＬの方に低い優先順位を与える。したがって、便益解析器４０６によって特定されたＰＲＢの最初のバッチは、より高い優先順位のＵＲＬに方向付けられるはずである。

【００４０】

ページ要求フィーダは、リソース管理プログラム４１０から、システムリソースの情報を受取る。このシステムリソースの情報に基づいて、ページ要求フィーダ４０８は、時間内のどの時点においても並行して処理することのできるＰＲＢの数を決定する。ＰＲＢは、ページ要求フィーダ４０８のために作動する処理実体に割当てられる。ＨＴＴＰ要求は、処理実体によって傍受器／ウェブサーバに送られる。ＨＴＴＰ要求に対するＨＴＴＰ応答は、傍受器／ウェブサーバから返されて、元のＰＲＢに付けられるか、または論理的に関連付けられる。

【００４１】

ＰＲＢおよび関連するＨＴＴＰ応答は、ＵＲＬコレクタ４１２に送られる。ＵＲＬコレクタ４１２は、事前に作成されるべき追加のＵＲＬを特定しようとする。この例において、事前作成のポリシーは、開始ページだけでなくメニューページが両方とも、事前に作成されるべきであることを示す。したがって、ＵＲＬコレクタ４１２は、各開始ページに対するＨＴＴＰ応答を巡回して、追加のメニューページに対するＵＲＬを特定しようとする。特定されたメニューページに対するＵＲＬは、便益解析器４０６に送られる新規のＰＲＢとして、パッケージ化される。

【００４２】

便益解析器４０６は、ＵＲＬコレクタ４１２から受取った新規のＰＲＢに基づいて、待機状態のＰＲＢの組に再び優先順位を付けて、ＰＲＢの別のバッチをページ要求フィーダ４０８に送る。上述のプロセスは、ＰＲＢのすべてが処理されるか、または、ポリシー用の処理時間が消滅するまで、続く。

【００４３】
スケジューラ（図示せず）は、事前作成器のサービスモジュール３１０内の構成要素の動作を調整する。一実施例において、スケジューラは、事前作成のポリシーをモニタして、どのポリシーが活性化して、どのポリシーが終了するようにスケジューリングされているかを判断するスレッドである。
【００４４】
図５は、傍受器のサービスモジュール５０２がキャッシュサーバ５０４に統合されている、この発明の代替的な一実施例を示す。キャッシュサーバ５０４は、ユーザとウェブサーバ５０６との間に介在している。この手法において、キャッシュサーバ５０４は、事前に作成されたページの記憶および取出しを管理する。これは、事前作成器のシステム構成要素が、事前に作成されてキャッシュされたページを記憶して取出す責任を負う、図３の手法とは対照的である。
【００４５】
多ノードの事前作成システム
この発明をスケーラブルに配備して、分散した複数の計算ノードに、調整された事前作成サービスを行なうことができる。一例として、この発明の実施例が、複数の中間階層機器を有するシステム構成で用いられる場合、事前作成の作業負荷を共有するために、中間階層機器の一部またはすべてにおいて事前作成サービスを開始することができる。事前作成器のインスタンスの各々は、事前に作成されたページのうちの割当てられた部分を生成し、事前に作成したページをネットワークアクセス可能な記憶装置に記憶する。ユーザが情報ページを要求すると、事前作成システム内で作動する任意のウェブサーバは、要求されたページが事前に作成されているかどうかを判断することができる。事前に作成されていると判断した場合、ウェブサーバは、事前に作成されたページが記憶されている特定のネットワークアクセス可能な装置から、事前に作成されたページを取出す。
【００４６】
図６は、「ノードがｎ個の」アーキテクチャ用のこの発明の一実施例を示し、ここでは複数の中間階層機器が分散型コンピュータシステムに存在する。図６では、第１の中間階層ノード６０６ａおよび第２の中間階層ノード６０６ｂが示される。ノード６０６ａおよび６０６ｂの各々は、データベースアプリケーション６１０等の同一のコンピュータアプリケーションにアクセスして、要求された情報ページを生成することができる。中間階層ノード６０６ａおよび６０６ｂの各々は、ノード６０６ａ上の傍受器６１６ａおよび事前作成器６２６ａ、およびノード６０６ｂ上の傍受器６１６ｂおよび事前作成器６２６ｂ等の、事前作成器の構成要素を含む。ノード６０６ａは、事前に作成されたページを記憶装置６３６ａに記憶して、ノード６０６ｂは、事前に作成されたページを記憶装置６３６ｂに記憶する。一実施例において、記憶装置６３６ａおよび６３６ｂは共に、他のネットワークノードからのアクセスが可能なＮＦＳ互換記憶装置である。
【００４７】
ユーザのページ要求がブラウザ６０２から送られると、ページ要求は、ルータ／負荷バランサー６０４によって、中間階層ノード６０６ａまたは６０６ｂの１つに経路指定される。ページ要求を現時点で扱っているノードが、要求されたページの、事前に作成されて有効なバージョンがシステム内のどこかに存在しているかどうかを判断する。存在していると判断した場合、このノードは、事前に作成されたページを取出す。ページがローカルな態様で事前に作成されておらず、事前に作成されたページが、遠隔で生成されて事前に作成されたページのようにネットワークアクセス可能なファイルシステムに記憶されている場合、このノードは、ネットワークアクセス可能なファイルシステムから、そのページを取出す。ローカルに生成されて事前に作成されたページがローカルなキャッシュまたは記憶装置に記憶されている場合、そのページに対するローカルな要求により、ローカルなキャッシュまたは記憶装置からの取出しが生じる。そのページが別のネットワークノードで事前に作成されていた場合、事前に作成されたページは、ネットワークアクセス可能な記憶装置から取出されて、ブラウザ６０２に送られて、表示される。

【００４８】
事前作成器の作業負荷が複数のノード間にどのように分配されるかを説明するために、図７では、この発明の一実施例に従ったｎ個のノードの事前作成器のシステムのための事前作成器の構成要素が示される。ｎ個のノードの事前作成器のシステムは、ページを事前に作成する作業負荷を分配する責任を負う負荷分配器７０２を含む。負荷分配器７０２は、複数の事前作成器のノード間で所与の事前作成器の作業負荷を分割する。事前作成器のノードの各々は、事前作成器のサービスインスタンス７０４を含む。図７では１つの事前作成器のサービスインスタンス７０４のみが示されているが、分散型システムにおいてこのような事前作成器の複数のサービスインスタンスが存在してよいことに注目されたい。一実施例において、負荷分配器７０２は、事前作成器のサービスインスタンス７０４の１つと同一の機器に配置される。代替的な実施例では、負荷分配器７０２が別個の機器上に配置される。

【００４９】
負荷分配器７０２は、事前作成のポリシーのすべてにアクセスして、分散型システムに対する現時点での事前作成の作業負荷を決定する。事前作成のポリシーに基づいて、負荷分配器７０２は、事前に作成するページを特定する。特定したページの各々に対してページ要求ＰＲＢが作成される。

【００５０】
負荷分配器７０２は、システム内の作成器のノード全体にＰＲＢの組を分割する。一実施例において、作業負荷の分配は、事前作成器の各ノードのリソースの使用量および利用可能性のレベルに基づく。リソースの利用可能性のレベルがより高い事前作成器のノードには、より多くの作業が割当てられて、リソースの利用可能性のレベルがより低い事前作成器のノードには、より少ない作業が割当てられる。作業負荷の分配を決定する要因として含めることのできるリソース使用量のパラメータの例は、事前作成器の各ノードにおける、ＣＰＵ、メモリ、および現時点での事前作成の要求のレベルである。このため、事前作成器の各サービスインスタンス７０４におけるリソース管理プログラム７０６は、好ましくは、リソース使用量の情報を負荷分配器７０２に通信する。

【００５１】
事前作成器のノードに作業を割当てる第１の手法は、大域的にリソースの使用レベルを見ることを含み、参加している全ノードにわたって作業負荷の全部が分配されて、システム全体におけるリソースの使用量を均衡させる。作業負荷の分配を行なうために、各ノードに対するリソースの「空き高」の総計が求められる。作業を割当てる第２の手法は、個々に事前作成ノードの各々を見て、そのノードに課されたリソース使用の正確な制約の範囲内においてのみ、そのノードに作業を割当てることを含む。

【００５２】
負荷分配器７０２は、事前に作成するＰＲＢのすべてを列挙したＰＲＢのデータ構造だけでなく、各ＰＲＢに割当てられた特定の事前作成器のノードの素性も維持する。ＰＲＢのデータ構造は、データベース７０８にテーブルとして記憶され得る。事前作成器のサービスインスタンス７０４の各々の開始ローダ７１０は、ＰＲＢのデータ構造を走査して、そのインスタンスに割当てられたＰＲＢを特定する。事前作成器のサービスインスタンス７０４に割当てられたＰＲＢは、優先順位付けおよびさらなる処理のために、そのインスタンス７０４用の便益解析器７１２に送られる。その後のＰＲＢの処理は、図４を参照して説明したＰＲＢの処理と実質的に同一であるが、例外として、ＨＴＴＰ応答が、ネットワークアクセス可能なネットワーク記憶装置に記憶される。加えて、完成したＰＲＢは、ＰＲＢデータ構造においてそのように識別される。一実施例において、ＰＲＢのデータ構造は、以下により詳細に説明するように、ユーザのページテーブルと同じである。

【００５３】
負荷分配器７０２は、多くの機会に事前作成の作業負荷を分配するように構成することができる。たとえば、負荷分配器７０２は、事前作成システムの起動時に、予め規定された間隔で、または、「作業負荷を再分配する」指令が明示的に発行されたときに、作業負

荷を再分配することができる。事前作成器のインスタンスの各々は、データベース内の１組の統計または状況メッセージを周期的にログ記録するように構成することができる。これらの統計を用いて、負荷を再分配することによって修正することのできる、システム全体の負荷の不均衡を特定することができる。加えて、周期的な報告を用いて、分散したノードでのエラー状況を特定することができる。耐故障性の目的で、負荷の再分配が、事前作成器のノードの１つでエラーまたは問題が認められたことに基づいて行われてもよい。

【００５４】

したがって、この発明は、システム内の事前作成のノード全体に事前作成の作業負荷を単に分配することによって、任意の数の事前作成器のノード全体に対して効果的にスケーリングされ得る。事前作成器の全中間階層ノードの余剰リソースは、余剰リソースを利用することのできるレベルのノードに追加の作業負荷を適切に分配または再分配することによって、制御しながら利用することができる。その一方で、利用可能なリソースがより少ないノードには、事前作成の作業負荷のうち、より小さい部分を割当てる。所与のノードに割当てられた事前作成の作業負荷の量に関係なく、すべてのノードは、事前に作成されたページの恩恵を等しく受ける。なぜなら、事前に作成されたページはすべて、たとえそれが他のノードによって事前に作成されたページであっても、ネットワーク化された記憶装置から一様にアクセスできるためである。

【００５５】

ＰＲＢ－ページ要求ブロック

一実施例によると、事前作成の候補である情報ページは、フォーマット化されたページ要求、またはＰＲＢとして内部では表わされる。図８は、この発明の一実施例に従ったＰＲＢの構造を示す。属性８０２は、特定の統一リソース識別子（ＵＲＩ）、たとえば、ＰＲＢに関連するＵＲＬ＋クッキー情報を特定する。属性８０４は、ＰＲＢ要求に関連するユーザのユーザ識別子を特定する。属性８０６は、ＰＲＢに関連するアプリケーション識別子を明示する。属性８０８は、ＰＲＢに対する責任の識別子を明示する。属性８１０は、ＰＲＢ要求に対するページ深さを規定する。開始ページまたはホームページが深さ「０」と考えられる場合、その開始／ホームページから次の追加のナビゲーション上のレベルは深さ「１」と考えられ、レベル「１」からすぐ次のナビゲーション上のレベルは、深さのレベル「２」と考えられ、以下同様である。深さの属性８１０は、ＰＲＢに関連するＵＲＩの深さのレベルを特定する。重みの属性８１２は、ＰＲＢに関連する優先順位付けの重みを特定し、これは、待機中のＰＲＢのリストを分類するために、便益解析器の構成要素によって計算され、用いられる。属性８１８は、平均的なページ生成時間を特定する。平均的な生成時間は、ＰＲＢが一旦作成されると、ページ要求フィーダの構成要素によって設定される。

【００５６】

固有の階層が、システム内の多くのＰＲＢ間に存在する。別のＰＲＢ（Ｂ）によって示されたページ内のＵＲＩを特定することによって生成されたＰＲＢ（Ａ）は、ＰＲＢ（Ｂ）の子と呼ばれる。どのようなＰＲＢも、０個またはそれより多くの子ＰＲＢを有することができる。子ＰＲＢは、親ＰＲＢの深さよりも１つ深い深さにあると考えられる。属性８１４は、親ＰＲＢのこのアプリケーションに対するエントリページを特定する。属性８１６は、現在のＰＲＢに対する子ＰＲＢを特定する。属性８２０は、現在のＰＲＢに対する子ＰＲＢの数を特定する。

【００５７】

一実施例において、システムによって処理されているＰＲＢは、以下により詳細に説明される、ユーザページテーブルにインメモリで記憶される。このテーブルは、好ましくは、データベース内に常駐する。一実施例では、履歴テーブルが維持されて、所与のアプリケーションに対し、システムによって処理された全ＰＲＢを追跡する。ＰＲＢに対する変更の頻度または速度等の統計は、履歴テーブルに維持される。これらの統計を用いて、ＰＲＢに対する重み付けアルゴリズムを変更または更新することができる。非常に古く、最近はアクセスされていないか、または頻繁にアクセスされないＰＲＢを、履歴テーブルか

ら抹消することができる。

【００５８】

事前作成のポリシー

この節は、この発明の一実施例に従った事前作成のポリシーの構造を説明する。一実施例によると、事前作成のポリシーは、事前作成器をブートストラップして調整するために用いられる構成情報を含む。事前作成のポリシーは、好ましくはアプリケーションによってストライピングされて、データベース内に常駐する情報として記憶される。事前作成のポリシーは、以下のもの、すなわち、（１）ポリシーが適用されるコンピュータアプリケーション、（２）ポリシーが適用されるユーザ、（３）事前作成されるべき情報ページ、（４）事前作成のスケジューリングおよびリフレッシュ間隔、（５）ポリシーが適用される責任、および（６）事前作成の優先順位、の一部またはすべてを特定する。ポリシーの異なるカテゴリの階層をこの発明で用いることができ、たとえば、以下のカテゴリがある。

【００５９】

・システムリソースポリシー：この種のポリシーはシステム全体のレベルに適用されて、その下で事前作成のプロセスの作動が可能になるシステムリソース条件を確立する。この種のポリシーの例には、ＣＰＵの消費限度、利用可能なメモリ、またはディスクスペースの限度を特定するポリシーが含まれる。

【００６０】

・アプリケーションレベルポリシー：この組のポリシーは、アプリケーション全体のレベルに適用され、システム内の特定のコンピュータアプリケーションに対し、事前作成のパラメータを確立する。所与のアプリケーションに対し、この組のポリシーを用いて、たとえば、事前作成のプロセスが実行される日時、ページが事前に作成される周期性（古くなることを防ぐため）、および事前に作成されるべきページの素性、種類、および／または量を確立することができる。所与のアプリケーションに対して１つ以上のアプリケーションレベルポリシーを確立することができる。

【００６１】

・責任およびユーザレベルのポリシー：これらの組のポリシーは、アプリケーション内の特定の責任およびユーザに適用される。一実施例において、これらのポリシーは、アプリケーションレベルポリシーのサブセットであり、アプリケーションレベルにおけるどのような設定も無効にする。各アプリケーションは、責任およびユーザレベルの複数のポリシーに対応することができる。

【００６２】

・非常駐のポリシー：これらは、事前作成器のサービスが、稼動する際に自動的に調整する、ランタイムのポリシーである。一実施例において、この種のポリシーは、ランタイムで計算されて、リソースの利用可能性およびシステムの現状に基づく。一例は、要求フィーダによって並行して処理されるＰＲＢの数を調整するポリシーを含む。システムリソースポリシーのパラメータを用いて、非常駐のポリシーを実現することができる。一実施例において、これらのポリシーは、データベースには持続されない。

【００６３】

java（Ｒ）層または中間階層において、ポリシーをポリシークラスのオブジェクトインスタンスとして表すことができる。ポリシーオブジェクトのプロパティの中には、常駐のものもあれば、非常駐のものもある。以下は、この発明の一実施例で用いることのできるポリシークラスに対するプロパティのリストである。

【００６４】

・applicationID：アプリケーションに関連する一意のＩＤ。（常駐）

・isEnabled：applicationIDに対応するアプリケーションに対して事前作成器が可能にされたかまたは不能にされたかを判断する。（常駐）

・startTime：このポリシーに対して事前作成器のサービスが開始されるべき時間。一実施例において、このプロパティは、責任またはユーザのレベルで変更することができな

い。（常駐）
・endTime：ポリシーに対して（必要であれば）事前作成器のサービスが終了されなければならない時間。一実施例において、このプロパティは、責任またはユーザのレベルで変更することができる。（常駐）
・interval：事前作成器のサービスがページを再び作成すべき周期性。このプロパティは、事前に作成されたページに対する「古さ」のしきい値を示す。一実施例において、このプロパティは、責任またはユーザのレベルで変更することができる。（常駐）
・depth：これは、ページが事前に作成される所望の深さを示す。一例として、このプロパティは、アプリケーションのホーム／開始ページのみ、全ページ、または規定された深さｘのページのいずれが事前に作成されるべきであるかを示す。一実施例において、このプロパティは、責任またはユーザのレベルで変更することができる。（常駐）
・currentDepth：このプロパティは、システムリソースおよび他の発見的方法の利用可能性に基づいて事前に作成するために、事前作成器のサービスが自動的に調整する実際の深さを示す。たとえば、このポリシーに対する「深さ」のパラメータが、４つのレベルのページを事前に作成すべきであることを示すものの、システムリソースのレベルによって１つのレベルのページしか事前に作成できない場合、「currentDepth」のパラメータは「１」に設定される。
【００６５】
・currentBatchSize：これは、システムリソースおよび他の発見的方法の利用可能性に基づいて、ページ要求フィーダがウェブサーバに送る、並行した要求の数を示す。
【００６６】
・cpuConsumptionLimit：このプロパティは、システムが利用するべきＣＰＵの最大率の上限を特定する。一実施例において、これは、システムワイドパラメータである。（常駐）
・memoryConsumptionLimit：このプロパティは、システムに存在すべき、使用していないメモリ量の下限を特定する。一実施例において、これはシステムワイドパラメータである。（常駐）
・diskSpace：このプロパティは、事前に作成されたページを記憶するのに利用することのできるディスクスペースの量を示す。一実施例において、これはシステムワイドパラメータである。（常駐）
・isAllResp：このプロパティは、ポリシーがアプリケーションの全責任に対して稼動すべきであるかどうかを判断する。（常駐）
・isAllUser：このプロパティは、ポリシーが全ユーザに対して稼動すべきであるかどうかを判断する。（常駐）
図９は、この発明の一実施例に従った、常駐のポリシーに対するスキーマを示す。各アプリケーションに対し、アプリケーションレベルのポリシーを常駐させる事前作成のアプリケーションポリシーテーブル９０２に１つの行が存在する。責任またはユーザレベルの任意のポリシーが構成されているかどうかに基づいて、事前作成のユーザポリシーテーブル９０４には０またはそれより多くの対応する行が存在する。テーブルアクセス機構の方法は、ＰＬ／ＳＱＬ等の手続き形の問合せ言語で実現され、たとえばＪＤＢＣを用いるJava（Ｒ）層から呼出される。
【００６７】
安全性の手段として、ポリシー情報は、好ましくは、適切な態様で許可されたアドミニストレータによってのみ、維持または変更される。アドミニストレータがポリシーを変更すると、ポリシーは、好ましくは直ちに更新され、たとえばポリシーが記憶される、関連するデータベーステーブルに伝搬される。
【００６８】
この発明の一実施例において、事前作成器のサービスが処理しているアプリケーションの各々に対し、ポリシーオブジェクトが事前作成器のサービス側においてインスタント生成される。ポリシースレッド／プロセスは、ポリシーオブジェクトの、常駐するプロパテ

ィを周期的にリフレッシュするように構成される。一実施例において、事前作成器のサービスの各々は、１つのアプリケーションにのみ専用である。すなわち、事前に作成されるべき３つのアプリケーションが存在する場合、事前作成器のサービスの３つのインスタンスが開始される。この手法において、事前作成器のサービスインスタンスは、このアプリケーションに対するポリシーオブジェクトのインスタンスにのみ対応し、ポリシースレッド／プロセスは、このアプリケーションに対するポリシーのみをリフレッシュする。代替的な実施例において、事前作成器のサービスは、複数のアプリケーションを支援するように構成される。この代替的な手法において、複数のアプリケーションに属する複数のポリシーオブジェクトが、事前作成器のサービスインスタンスと関連する。

【００６９】

事前作成器のサービスの異なる構成要素がポリシーオブジェクトにアクセスする。たとえば、開始ローダの構成要素は、ポリシーオブジェクトにアクセスして、所与のアプリケーションに対して構成された責任およびユーザを特定し、最初の作業負荷を生成する。便益解析器は、ポリシーオブジェクトにアクセスして、システム構成およびリフレッシュ間隔を特定する。便益解析器は、また、ポリシーオブジェクト内のcurrentDepthのプロパティを更新する。ページ要求フィーダは、処理することのできる、並行した要求の数に基づいて、currentBatchSizeのプロパティを設定する。ＵＲＬコレクタは、currentDepthのプロパティ情報を読出して、事前に作成されたページを巡回するかどうかを決定する。

【００７０】

ユーザページテーブル

この明細書ではユーザページテーブルと呼ばれる、インメモリのデータ構造は、この発明の一実施例で事前作成器のサービスが処理しているＰＲＢのすべてを追跡する。ユーザページテーブルの内容は、事前作成用に特定された完全な組のＰＲＢを表す。ユーザページテーブルは、まだ事前に作成されていないＰＲＢだけでなく、事前に作成されたＰＲＢの両方を含み、どのＰＲＢが事前に作成されたかを示すデータを含む。

【００７１】

一実施例において、ユーザページテーブルは、便益解析器の構成要素によって頻繁にアクセスされ、照会され、更新される。ＰＲＢの属性のいくつかを用いて、ユーザページテーブルの問合せを可能にする索引を構築することができる。索引テーブルは、ハッシュテーブルとして構築され、キーはＰＲＢのプロパティであり、値は同じキープロパティ値を有するＰＲＢのベクトルである。ユーザページテーブル自体をＰＲＢのベクトルとして実現することができる。図１０は、この発明の一実施例に従った、ユーザページテーブルの要素に高速でアクセスするために構築された索引テーブルのリストを示す。

【００７２】

一実施例において、便益解析器は、ユーザページテーブルに対して直接変更を生じる、事前作成器のサービスモジュールにおける唯一の構成要素である。便益解析器は、複製のＰＲＢが（すなわち、同一のＵＲＩを有する２つのＰＲＢが）決してテーブルに追加されないように構成され得る。

【００７３】

開始ローダ

この節は、この発明の一実施例に従った開始ローダを説明する。開始ローダは、事前作成器のサービスによって処理されるべきＰＲＢの最初の組を生成する。開始ローダのデフォルトの実施により、事前作成器のポリシーで構成されたアプリケーション、責任およびユーザに基づき、ユーザの開始／ホーム／エントリページに対する可能な全ＵＲＩを生成する機構が設けられる。一実施例において、アプリケーションは、このデフォルトの実施を無効にして、ユーザに対するエントリページのＵＲＩを決定するための、アプリケーションに固有の論理を与えることができる。開始ローダによって生成されたＵＲＩの各々は、ＰＲＢと表わされ、便益解析器の構成要素に渡される。事前作成器のサービスのための最初の作業負荷を生成するために、開始ローダに加え、他の機構を用いてもよいことに注目されたい。たとえば、ＵＲＬコレクタから以前に生成されたＰＲＢを含む、先行する事

前作成のセッションで作成されたＰＲＢの組を、現在の事前作成のセッションに対する最初の作業負荷として用いることができる。この代替的な手法は、或る期間にわたって静的な状態を保つ傾向のあるページにとって有用である。

【００７４】

図１１を参照すると、この発明の一実施例に従った開始ローダのプロセスフローが示される。開始ローダは、スレッドとしてインスタンス化されて、構築器が、アプリケーションＩＤを入力として受取る。アプリケーションＩＤは、適切な事前作成器のポリシーを取出すために用いられる。事前作成のポリシーから、それに対して作業負荷が生成されなければならない適切な組の責任およびユーザＩＤの決定が行なわれる。開始ローダのスレッドは、周期的に、たとえば１日に２回または事前作成器が起動されるたびごとに、最初の作業負荷を再生成する。これにより、システムに登録された任意の新規ユーザおよび他のこのような変更が、確実に最初の作業負荷によって捕捉される。

【００７５】

図１２は、この発明の一実施例に従った、ＵＲＬパラメータインターフェイスの実現器に常駐する適切なエントリレベルのページに対してＰＲＢを生成するための、フロー図の処理論理を示す。１２０２において、実現器が、アプリケーションＩＤに関連する事前作成器のポリシーオブジェクトをロードする。

【００７６】

この方法は、ポリシーオブジェクトを用いて、ポリシー用に構成された特別の責任およびユーザＩＤを特定する（１２０４）。特別な責任が構成されていない場合、アプリケーション内のすべての責任が事前に作成されなければならないことが考えられる。あるいは、構成された組の責任のみが用いられる。加えて、このプロセスにより、構成されたユーザＩＤのすべてが確実に最初の作業負荷で生成される。責任とは異なり、この組のユーザは、事前作成がターンオンされるユーザとは限らないことが考えられる。構成された組の責任に合致する有効な組のユーザＩＤに関する判断が行なわれる。加えて、構成されたユーザＩＤに対する責任の識別子に関する判断が行なわれる。これらの判断を行なうために、ユーザＩＤおよび責任のテーブルを維持して、それらに照会することができる。事前作成器は、信頼されたユーザとしてシステムにログインし、次に、そのユーザにページを事前に作成する際に、素性を特定のユーザに切換える。

【００７７】

この方法は、適用可能なユーザＩＤの各々に対するエントリページのＵＲＩを決定する（１２０６）。最初の作業負荷を生成する構成要素は、アプリケーションに固有の論理およびデータを用いて、事前作成を行なっている全ユーザに対するＵＲＬ値を生成する（１２０８）。これらのＵＲＬの表示は、以前に得たＵＲＩに付加される。その後、この方法は、生成されたＵＲＩの各々に対してＰＲＢを構築する（１２１０）。構築されたＰＲＢは、さらなる処理および事前作成のために、便益解析器に送られる。

【００７８】

便益解析器

この節は、便益解析器の一実施例を説明する。事前作成器のどのような所与の時点およびシステムの状態においても、便益解析器は事前に作成する次の組のページを決定する。一手法において、便益解析器は１つのスレッドとして実行される。

【００７９】

便益解析器は、待機中のＰＲＢのイン・キューを維持する。このイン・キューは、時間内の任意の時点で一度に処理されるべきＰＲＢのすべてを含む。このイン・キューは、開始ローダまたはＵＲＬコレクタのいずれかからのＰＲＢによって占められる。上で述べたように、開始ローダは、最初のブートストラッピングの作業負荷を生成する。ＵＲＬコレクタは、以前に処理されたＰＲＢの出力を巡回することによって便益解析器に新規のＰＲＢを生成する。便益解析器はまた、ユーザページテーブルを用いて、処理するＰＲＢの作業負荷を特定する。便益解析器は、ページ要求フィーダに送るためのＰＲＢのアウト・キューを維持する。

【００８０】
便益解析器は、重み付けアルゴリズムを用いて、待機中のＰＲＢのキューに優先順位を付け、システム内の各ＰＲＢに重みを関連付ける。一実施例において、重み（ｗ）は以下のものに関連する。
【００８１】
ｗ∝　１／深さ（ホームページが深さ０であり、メニューページが深さ１であり、サブメニューが深さ２である等）
　∝　ＰＲＢの処理時間（一手法では、すでに速いＰＲＢよりも、より遅いＰＲＢの方が、事前作成には優れた候補である）
　∝　パラメータ適合率（たとえば、ポリシーで構成されたユーザＩＤは、他のものに比べてより重みを有し得る）
　∝　ユーザのヒット率（より多くアクセスされるＰＲＢは、他のものよりも重要である）
ＰＲＢの重み（ｗ）は、以下のように規定することができる。
【００８２】
$w = k^{*}1/\text{深さ} + p^{*}t_{processPRB} + q + r^{*}\text{ヒットカウント}$
ここで、ｋ、ｐ、ｑおよびｒは、一定の値として選択される。ヒットカウントについては、フィードバック機構がウェブサーバから事前作成サービスに実現される。この機構が存在しない場合、定数「ｒ」を０に設定することができる。好ましい実施例では、深さの因子が、最も重要な因子である。したがって、定数「ｋ」は、「ｐ」および「ｑ」よりも高い値に設定される。厳密な重み付けの公式および重み付け定数は、この発明が方向付けられる厳密なアプリケーションおよび性能の要件の組に基づいた設計上の選択である。便益解析器のイン・キューには、ＰＲＢの重みに基づいて優先順位が付けられる。その後、便益解析器は、要望に応じて、次の優先順位が付いた、処理されるべきＰＲＢをページ要求フィーダに渡す。
【００８３】
一実施例では、適用可能なポリシーにも、便益解析器によって優先順位を付けることができる。第１のポリシーの要件には、第２のポリシーの要件よりも高い優先順位を与えることができる。たとえば、第１のポリシーが第２のポリシーよりも高い優先順位を有している場合、第１のポリシーに応答して生成されたＰＲＢの組には、第２のポリシーに応答して生成されたＰＲＢの組よりも大きな重みが与えられる。
【００８４】
一実施例において、便益解析器は、以下の統計、すなわち、（１）生成されたＰＲＢの数、（２）平均リフレッシュ速度、（３）システム内のＰＲＢの総数－ユーザページテーブルのサイズ、（４）最適なバッチサイズ、（５）現在の深さ、（６）ポリシーの開始時間、（７）ポリシーの終了時間、（８）稼動期間、および（９）ヒット率＝（アクセスされた、事前に作成されたＵＲＩの数）／（ユーザが要求したＵＲＩの総数）、を生成し、追跡し、または維持するように構成される。
【００８５】
ページ要求フィーダ
この節は、ページ要求フィーダの一実施例を説明する。ページ要求フィーダは、ＰＲＢからページ生成を開始する。ページ要求フィーダは、現在のシステムのスナップショットに基づき、ウェブサーバに送る、並行した要求の数を設定して、応答を受取り、ページ応答をＵＲＬコレクタに渡す。ページ要求フィーダは、処理されるべきＰＲＢを、便益解析器からの入力として受取る。ページ要求フィーダは、また、ポリシーオブジェクトからの最適なバッチサイズの値およびＣＰＵの消費限度の最大値を用いる。
【００８６】
ページ要求フィーダは、１つの「モニタ」処理構成要素（モニタスレッド等）および変動する数の「ワーカー」構成要素（ワーカースレッド等）を含む。モニタスレッドは、現時点でのシステムのスナップショットに基づいて、ウェブサーバに送る、並行した要求の

数を動的に変更する責任を負う。モニタスレッドは、以下のステップを用いて、要求の数を動的に調整する。すなわち、（１）モニタスレッドは、システム内の現時点でのＣＰＵの使用量が、ポリシーオブジェクトで構成されたＣＰＵの最大値よりも少ないかどうかをチェックし、（２）未処理のページ要求の数をチェックして、それが、事前作成のポリシーで一覧にされた最適値よりも少ないかどうかを判断し、（３）上述の条件が共に満たされると、１つ以上のＰＲＢが便益解析器から取出されて、使用されていないワーカースレッドに渡されるか、または、現時点でどれも利用できない場合は、作成された新規のワーカースレッドに渡される。

【００８７】

各ワーカースレッドは、以下のステップを実行する責任を負う。すなわち、（１）要求フィーダのイン・キューからＰＲＢを獲得し、（２）ＰＲＢ内のＵＲＩを処理するためにＨＴＴＰ要求を送り、（３）ＨＴＴＰ応答をＰＲＢに付加して、（４）ＰＲＢをＵＲＬコレクタに渡す。

【００８８】

この発明の一実施例において、ページ要求フィーダの並行性のレベル、または、未処理の要求の数は、便益解析器によって制御される。しかしながら、この手法は、中間階層またはデータベースの作業負荷の変化に適合する事前作成器のシステムに遅延を生じるおそれがある。したがって、代替的な実施例では、便益解析器が最適な並行性を決定するが、支援すべき適合的な並行性または現時点での並行性を、要求フィーダが計算する。これにより、ウェブサーバ上のバーストトラフィック、またはシステム負荷の突然の変化に対して、より良い反応時間がもたらされる。

【００８９】

ＵＲＬコレクタ

この節は、ＵＲＬコレクタの一実施例を説明する。ＵＲＬコレクタは、ページ要求フィーダからのイン・キューから、入力としてＰＲＢを得る。各ＰＲＢは、ＰＲＢのＨＴＴＰ要求に応えてウェブサーバによって生成されて送られたＨＴＴＰ応答のストリームを含む。ＵＲＬコレクタは、ＨＴＴＰ応答のストリームを巡回して、事前に作成する新規のＵＲＩを収集する。一実施例において、ＵＲＬコレクタはウェブクローラーとして働き、現時点での事前作成のポリシーに基づいて応答のストリームを巡回する。事前作成のポリシーは、ＵＲＬコレクタがどのページを巡回すべきであるかを決定する。たとえば、ポリシーオブジェクトのcurrentDepthのパラメータを用いて、特定のページが巡回されるべきであるかどうかを設定することができる。

【００９０】

一実施例において、ＵＲＬコレクタは、１つのスレッドとして実行される。ＵＲＬコレクタのイン・キュー内にある、関連するＰＲＢの各々に対し、ＵＲＬコレクタは、応答のストリームのストリングバッファをパーズして、＜href＞参照のすべてを探す。ＵＲＬコレクタは、関連する＜href＞の各々に対してＰＲＢを構築して、それを便益解析器に送る。便益解析器は、複製のＵＲＩが決してＰＲＢのリストに追加されないようにする。

【００９１】

ＵＲＬコレクタは、これまでに巡回されていないページか、または、最後に巡回された時点から変化したページのみを巡回するように構成される。このことを達成するために、ＵＲＬコレクタは、応答ストリームのバッファサイズが、最後に巡回された時点と同じであるかどうかをチェックする。チェックサムまたはhashIDルーチンを実行して、ページを巡回する必要があるほどページが変化したかどうかをさらに判断することができる。たとえばコンパイルまたは他のＪＳＰの例外を有するエラーページを形成するどのような応答ストリームも、巡回されない。

【００９２】

リソース管理プログラム

この節は、この発明で用いることのできるリソース管理プログラムの一実施例を説明する。リソース管理プログラムは、事前作成器のサービスが実行されている機器におけるＣ

ＰＵおよびメモリの使用量等のシステムリソースの使用量をモニタする。リソース管理プログラムは、事前作成器のサービスが関与しているアパッチサーバの、並行した要求の数をモニタすることによって、中間階層システムの負荷をモニタする責任を負う構成要素でもある。このモニタリングは、ＵＲＬ要求をmod-statusのアパッチモジュールに送ることによって実行され得る。

【００９３】

現時点のシステム使用量および予め規定されたシステムリソースの限度に基づいて、リソース管理プログラムは、予め設定されたシステムリソースの限度を超過したかどうか、および、事前作成器のサービスがアパッチサーバに送ることのできるページ事前作成の要求の数はいくつか、に関する情報を、ページ要求フィーダ等の、事前作成器の他の構成要素に与える。

【００９４】

リソース管理プログラムは、一実施例においてデータベースに記憶されるシステムポリシーオブジェクトからの以下のシステムリソースの限度および情報、すなわち、（１）中間階層システムにおける全アパッチサーバおよびそれらのポート数のリスト、（２）負荷バランサーを使用すべきであるかどうか、および、もし使用すべき場合は、負荷バランサーのポート数およびホスト名、（３）メモリの使用量の限度、（４）ＣＰＵの使用量の限度、ならびに（５）中間階層システムが、システムの応答時間に著しい影響を与えずに処理し得るべき、並行した要求の最大数、を用いる。

【００９５】

リソース管理プログラムは、構成情報にアクセスして、ページ事前作成器の要求をウェブサーバまたは負荷バランサーのいずれに送るべきであるかを決定する。一実施例において、負荷バランサーが（ｎ個のノードのアーキテクチャ用に等で）使用されている場合、ページ事前作成の要求はすべて、負荷バランサーに送られるべきである。利用することのできる負荷バランサーがない場合、ページ事前作成の要求は、ローカルウェブサーバに、すなわち、事前作成器のサービスと同一の機器上に常駐するウェブサーバに送られる。このような機器上で２つ以上のウェブサーバが稼動している場合、同数のページ事前作成の要求が、各ウェブサーバに送られる。

【００９６】

リソース管理プログラムは、また、ページ要求フィーダが送ることのできる、並行したページ事前作成のＨＴＴＰ要求の数も決定する。一実施例では、構成された以下の値が、事前作成器のサービスインスタンスがウェブサーバに送るべき並行した要求の数を計算する際に用いられる。

【００９７】

・Rmaxは、中間階層システムが、応答時間の著しい悪化なく、処理することのできる並行した要求の最大数である。

【００９８】

・Ｗは、事前作成器のサービスインスタンスが稼動する機器の相対重みである。

【００９９】

・Wtotalは、構成された全機器の相対重みの総計である。

【０１００】

負荷バランサーが用いられるべき場合、リソース管理プログラムは、構成された全ウェブサーバ（Rc1）上の並行した要求の数を規則的にモニタする。一実施例において、この事前作成器のサービスインスタンスが負荷バランサーに送ることのできる要求の数（Ｒ）は、以下のとおりである。

【０１０１】

$R = (Rmax - Rc1)^{*} (W / Wtotal)$

負荷バランサーが用いられず、ページ事前作成の要求がローカルウェブサーバに送られるべき場合、リソース管理プログラムは、ローカルウェブサーバ（Rc2）上の並行した要求の数を規則的にモニタする。一実施例において、この事前作成器のサービスインスタン

スが負荷バランサーに送ることのできる要求の数（Ｒ）は、以下のとおりになる。

【０１０２】

Ｒ＝（Rmax*（Ｗ／Wtotal））－Rc2

リソース管理プログラムは、また、ローカルシステム上のＣＰＵおよびメモリの使用量もモニタする。一実施例では、別個のスレッドがＣＰＵおよびメモリの使用量の測定用に生成される。測定は、予め規定された間隔で行なわれ、リソース管理プログラムは、測定された使用量の数および予め設定されたリソース限度を用いて、限度を超過したかどうかを判断する。リソース管理プログラムは、以下の出力、すなわち、（１）ローカル機器上の現時点でのＣＰＵの使用量、（２）ＣＰＵの使用量の限度を超過したかどうか、（３）ローカル機器上の現時点でのメモリの使用量、（４）メモリの使用量の限度を超過したかどうか、（５）ページ事前作成のＨＴＴＰ要求が送られるべきウェブサーバのホスト名およびポート数、および（６）このページ事前作成器のサービスインスタンスがウェブサーバまたは負荷バランサーに送ることのできる、並行したページ事前作成のＨＴＴＰ要求の数、を生じる。

【０１０３】

リソースの最適化

事前作成器のサービスの目標は、利用可能なシステムリソースを最適に用いて、ページを事前に作成して再び生成することである。１つの手法において、事前作成およびリフレッシュの速度は、所与のポリシーに対して構成されたリフレッシュ間隔と同程度の速度であるか、またはこのリフレッシュ間隔よりも高速であるべきである。アドミニストレータによって構成されたリフレッシュ間隔は、特定のアプリケーションが容認することのできるページの古さを示す。

【０１０４】

一例として、以下の内容は、バックエンドサーバ（データベースサーバ等）にアクセスする１つ以上の中間階層機器を有するシステムのためのリソース最適化方法を説明する。この実施例で以下の因子を用い、事前作成プロセスのためのリソースの使用を最適化する。

【０１０５】

－用いることのできるシステム［中間階層＋事前作成サービス］のＣＰＵの最大率（$CPU_{midtier}$）

－用いることのできるバックエンドサーバのＣＰＵの最大率（$CPU_{database}$）

－平均応答時間またはＰＲＢの処理時間（$T_{average}$）

上の３つのパラメータが変動するにつれて、以下の表に示すように、並行したＰＲＢが処理される態様もまた変動する。

【０１０６】

【表１】

$T_{average}$	$CPU_{middle-tier}$	$CPU_{database}$	理由	同時要求の数
遅い	高い	低い	Jserv の負荷が高すぎる	減る
遅い	低い/通常	高い	DB がボトルネックである	減るか、または無し
速い	高い	低い	Jserv の負荷が高すぎる	$CPU_{middle-tier}>CPU_{max}$ の場合減る
速い	低い/通常	高い	DB がボトルネックである	$CPU_{database}>CPU_{max}$ の場合減る

【０１０７】

上の表に基づき、ＣＰＵの使用が最大レベルに到達するまで、事前作成のサービスからウェブサーバへの並行した要求の数を増やすことができる。このレベルに一旦到達すると、最大応答時間よりも低い、ＣＰＵの使用レベルをもたらす並行性が、最適な並行性として選択される。この最適な並行性は、システム動作の変更に同調するために周期的に較正し直される。

【０１０８】

代替的な手法は、ウェブサーバによって処理され得る並行した要求の最大数を設定するパラメータ設定を用いる。このパラメータ設定は、たとえば、許可されたアドミニストレータによってもたらすことができる。この手法において、事前作成器のシステムの目標は、利用可能なリソースのすべてを用いることではなくて、むしろ、利用可能なリソースの最適量を用いて、１頁当たりの最適なリフレッシュ時間を得ることである。この手法は、発行され得る並行した要求の最大数を用いる。システムの最大の並行性または最適な並行性のための正確なパラメータ設定が、この手法の基本となる。Jserv処理に入来するトラフィックは、並行した要求が多すぎてユーザシステムを過負荷にすることを避けるように注意深く調整されるべきである。この手法の利点は、プラットフォーム依存型のシステム呼出しと、分散したＣＰＵ測定とを行なうことを避け、かつ、この手法を用いて、事前作成器が使用すべきＣＰＵのリソースの正確な量を特定できることにある。

【０１０９】

安全性およびセッションの管理

この節は、この発明の一実施例に従った事前作成のための安全管理の方法を説明する。この説明は、限定ではなく例示のために、ユーザ接続を「セッション」と呼び、ユーザ接続用の識別子を「セッションＩＤ」と呼ぶ。特定されたユーザセッションは、たとえば、ユーザがデータベースサーバに対して許可された接続を確立して、データベースアプリケーションを実行する際に形成され得る。セッションＩＤを用いてユーザセッションを確認して、ユーザのプリファランスを特定する。セッションを実現するための１つの手法では、ユーザがアプリケーションにログインすることに成功すると、有効なセッションＩＤが発行される。セッションＩＤは、ユーザがアプリケーション内のページにアクセスできるようになる前に、有効性があるかどうかチェックされる。セッションＩＤは、ユーザがログアウトするまで有効である。

【０１１０】

ユーザがセッション内でページからページへ移動すると、セッションＩＤは、見られるページと一緒に、サーバによって渡される。セッションＩＤは、クッキーが使用可能にされた際のクッキー値として、または、クッキーが不能にされた際のＵＲＬパラメータとして、設定される。この発明では、ユーザによってアクセスされるページの中には、事前に作成されたものもあれば、同時に作成されるものもあるために、ユーザに送られてサーバによって処理されるページ間の連続性が途切れるおそれがあり、したがって、セッションＩＤが、ページからページへと完全に渡されないおそれがある。この発明は、事前に作成されたページがユーザに返されたときでも、ユーザセッションに関連するセッションＩＤの有効性を確認する追加の技術を提供する。

【０１１１】

この発明の一実施例において、事前作成器は、ユーザ用のログインページの事前作成を回避するように構成される。したがって、ユーザは、通常のログイン手続きを進めて、有効なユーザセッションを確立しなければならない。ログインが成功すると、セッションＩＤがユーザセッションに割当てられる。

【０１１２】

ユーザが最初に傍受器にページを要求すると、セッションＩＤの有効性は、たとえばページ要求をバックエンドデータベースサーバに渡すことによる、通常の確認手続きを経て確認される。その時点から、傍受器は、そのセッションＩＤの有効性の記録を維持する。ユーザが、事前に作成されていないページを要求した場合、サーバは、ページ要求を動的に処理して、ページ応答上にセッションＩＤを置く。しかしながら、ユーザが、事前に作

成されたページを要求した場合、傍受器は、有効なセッションＩＤがページの一部として含まれた状態で、事前に作成されたページをユーザに返す。したがって、傍受器自体は、たとえ事前に作成されたページがユーザに返される場合でも、ユーザに返されたページ内でセッションＩＤが確実に渡されるようにする。有効なセッションＩＤが応答ページでユーザに対して常に返されるので、ユーザセッションは、事前に作成されたページと事前に作成されていないページとの間をユーザが移動する遷移中も変化しない。

【０１１３】

ｎ個のノード環境において、ユーザは、１回のセッションで複数のサーバノードにページ要求を送ることができる。一実施例によると、セッションＩＤの有効性を最初に確認する傍受器のモジュールが、そのセッションＩＤをシステム内の他のノードの各々に伝播する。そのセッションＩＤの記録は、各ノードで有効であるものとして維持される。ユーザがページ要求を発行すると、ルータまたは負荷バランサーが、分散したノードの１つにそのページ要求を経路指定する。そのノードのローカルな傍受器が、そのユーザに対するセッションＩＤの有効性を既に認識していることが考えられるため、そのローカルな傍受器は、さらに確認を行なわずにそのページ要求を直ちに処理することができる。要求されたページが既に事前に作成されている場合、添付されたセッションＩＤを有するページ応答を、ユーザに直ちに返すことができる。

【０１１４】

この発明の一実施例では、ユーザのプリファレンス、ユーザがログインしているアプリケーション、およびユーザの有する責任に関する情報を記憶するために、クッキーが用いられる。クッキーは、選択されたメニューページ等の他の状態情報を含むこともできる。クッキーが不能にされると、この情報は、ＵＲＬパラメータとして設定される。

【０１１５】

ユーザが有効なユーザ名およびパスワードを用いてログインすると、ユーザセッションを追跡するために、セッション識別子が生成される。このセッション識別子は、ユーザが自発的にログアウトするか、または、セッションが時間切れになるまで、ユーザによる後続の要求のすべてに含まれる。セッション識別子は、ログイン時に捕捉されて、共有のメモリテーブルに記憶される（したがって、アパッチ処理はすべて、それに対するアクセスを有する）。キャッシュから処理された後続の要求は、セッション識別子が共有メモリ内に存在することが確認されて初めて、送り返される。セッション識別子が共有メモリ内に存在しない場合、Jservに対するデフォルト処理およびセッションの確認が行なわれる。これにより、無効のセッション識別子を用いることによるシステムへのアクセスが確実に不可能となる。

【０１１６】

以下のステップは、この発明の一実施例に従ってユーザセッションを設定するために実行される。

【０１１７】

ａ）　このシステムは、ゲストユーザ名およびパスワードを用いるユーザを認証する。

【０１１８】

ｂ）　ユーザセッションが、事前作成器のサービスによって提出されたユーザＩＤを用いて作成される。

【０１１９】

ｃ）　ユーザの素性が、入力として渡されたユーザＩＤに切換えられる。これにより、セッションを、関連するユーザに属するものとして設定する。

【０１２０】

ｄ）　このセッションに対するセッションＩＤを決定する。

【０１２１】

作成されたセッションＩＤは、開始ローダによって用いられて、ユーザにＵＲＩを構築する。セッションの作成および確認は費用がかかるおそれがあるために、セッションＩＤは、複数のユーザに属する、事前作成器の要求によって共有される。したがって、事前作

成器によって生成されたＨＴＴＰ要求が発行されると、セッションＩＤの典型的な確認のプロセスは省略される。その代わりに、クッキーパラメータを用いて、ユーザのプリファレンスの設定を得て、ユーザのコンテキストを正確に設定する。これにより、中間階層および接続レベル両方のユーザのコンテキストを設定する。

【０１２２】

クッキーがターンオンされると、生成されたＵＲＩは、一実施例に従い、ＵＲＬパラメータおよびクッキーパラメータのすべてを含む。事前に作成されたページと事前に作成されていないページとの間での滑らかな遷移を支援するために、傍受器は、ＵＲＩ内の実際のユーザのセッションＩＤを用い、事前作成器によって生成されたセッションＩＤは用いない。これにより、アクセスされているページがどのページであるかに関係なく、ユーザは確実に、現時点でログインしたセッションのコンテキスト内に常に存在することになる。

【０１２３】

傍受器

この節は、傍受器の一実施例を説明する。傍受器は、そのキャッシュに、動的に生成されたＪＳＰ（Java（Ｒ）サーバページ）の、事前に作成されたバージョンを記憶して、ＪＳＰのこれらの静的な、キャッシュされたバージョンを用いて、真のユーザ要求を処理しようとする。傍受器は、事前に作成されてキャッシュされたページ応答を用いてユーザ要求をまず最初に処理しようとすることによって、アパッチウェブサーバのmod_jservモジュールのデフォルト動作を無効にする。ユーザ要求は、事前に作成されたページのファイル名にマッピングされて、キャッシュ内で正確な合致が発見されると、事前に作成された応答がユーザに送り返される。合致が見つからない場合、デフォルトの動的なページ生成プロセスが生じ、そのページは、ユーザに送り返される。

【０１２４】

傍受器は、ページ要求が事前作成器から生じているか、またはユーザから生じているかに依存して、異なる態様でページ要求を処理する。一実施例では、ページ要求の発信者を特定するページ要求オブジェクトにおいて、パラメータが規定される。

【０１２５】

ユーザがページ要求を送ると、一実施例に従って、傍受器は以下のステップを行なって、ページ要求を処理する。

【０１２６】

１．たとえばファイルシステム内で、事前に作成されたページを探すために用いることのできる一意の識別子に、ページ要求をマッピングする。

【０１２７】

２．要求されたページがキャッシュ内に存在するかどうかをチェックする。事前に作成されたページが存在し、容認可能な最大のキャッシュ古さの範囲内で生成されていた場合、事前に作成されたページをユーザに返す。

【０１２８】

３．要求されたページがキャッシュ内に存在しない場合、デフォルトの動的な生成を行なう。

【０１２９】

４．要求されたページがキャッシュ内に存在するものの、古さの最大要件に違反している場合、ページを動的に生成し、動的に生成したページを用いて、キャッシュされたファイルをリフレッシュする。

【０１３０】

５．ページ応答をユーザに返す。

【０１３１】

事前作成器がページ要求を送ると、傍受器は、以下のステップを行なって、ページ要求を処理する。

【０１３２】

１．ページを動的に生成する。

【０１３３】

２．事前に作成されたページを探すために用いることのできる一意の識別子に、ページ要求をマッピングする。

【０１３４】

３．ステップ２で生成された静的なファイル名で、キャッシュに応答を記憶する。

【０１３５】

４．ページ応答を事前作成器に返す。

【０１３６】

システムアーキテクチャの概要

図１３を参照すると、一実施例において、コンピュータシステム１３２０は、複数の個々のユーザステーション１３２４に接続されたホストコンピュータ１３２２を含む。一実施例において、各ユーザステーション１３２４は、限定ではなく例として、１つ以上のアプリケーション、すなわちプログラムを記憶して別個に稼動させることのできるパーソナルコンピュータ、ポータブルラップトップコンピュータ、または個人用携帯型情報端末（「ＰＤＡ」）等の適切なデータ端末を含む。例示のために、ユーザステーション１３２４の中には、ローカルエリアネットワーク（「ＬＡＮ」）１３２６を介してホストコンピュータ１３２２に接続されるものもあれば、ユーザステーション１３２４の中には、一般加入電話網（「ＰＳＴＮ」）１３２８および／または無線ネットワーク１３３０を介してホストコンピュータ１３２２に遠隔接続されるものもある。

【０１３７】

一実施例において、ホストコンピュータ１３２２は、データ記憶システム１３３１とともに作動し、データ記憶システム１３３１は、ホストコンピュータ１３２２がすぐにアクセスすることのできるデータベース１３３２を含む。複数の階層のアーキテクチャを用いて、たとえば中間アプリケーション階層（図示せず）を用いて、ユーザステーション１３２４をデータベース１３３２に接続することができる。代替的な実施例において、データベース１３３２は、たとえばホストコンピュータのＲＯＭ、ＰＲＯＭ、ＥＰＲＯＭ、または他の任意のメモリチップ、および／またはそのハードディスクに記憶されて、ホストコンピュータに常駐することが可能である。さらに代替的な実施例では、１つ以上のフロッピー（Ｒ）ディスク、フレキシブルディスク、磁気テープ、他の任意の磁気媒体、ＣＤ－ＲＯＭ、他の任意の光学媒体、パンチカード、紙テープ、または孔のパターンを有する他の任意の物理媒体、またはコンピュータがそこから読出すことのできる他の任意の媒体から、ホストコンピュータ１３２２によってデータベース１３３２を読出すことができる。代替的な実施例において、ホストコンピュータ１３２２は、上で論じたように、さまざまな媒体に記憶された２つ以上のデータベース１３３２にアクセスすることができる。

【０１３８】

図１４を参照すると、一実施例において、各ユーザステーション１３２４およびホストコンピュータ１３２２は、各々が包括的に処理装置と呼ばれて、包括的なアーキテクチャ１４０５を実現する。処理装置は、命令、メッセージおよびデータといった情報をまとめて通信するためのバス１４０６または他の通信機構と、バス１４０６に結合されて情報を処理するための１つ以上のプロセッサ１４０７とを含む。また、処理装置は、バス１４０６に結合されて、かつ、プロセッサ１４０７によって実行されるべき動的なデータおよび命令を記憶する、ランダムアクセスメモリ（ＲＡＭ）または他の動的な記憶装置等のメインメモリ１４０８も含む。メインメモリ１４０８は、プロセッサ１４０７によって命令が実行される間に、一時データ、すなわち、変数、または他の中間情報を記憶するために用いることもできる。処理装置は、さらに、バス１４０６に結合されてプロセッサ１４０７に対する静的なデータおよび命令を記憶するための読出専用メモリ（ＲＯＭ）１４０９または他の静的な記憶装置を含むことができる。磁気ディスクまたは光学ディスク等の記憶装置１４１０を設けてバス１４０６に結合し、プロセッサ１４０７に対するデータおよび命令を記憶することもできる。

【０１３９】
処理装置は、限定ではなく例として、情報をユーザに表示するための陰極線管（ＣＲＴ）等の表示装置１４１１に、バス１４０６を介して結合することができる。英数字および他の列を含む入力装置１４１２がバス１４０６に結合されて、プロセッサ１４０７に情報および指令の選択を通信する。別の種類のユーザ入力装置は、プロセッサ１４０７に方向の情報および指令の選択を通信してディスプレイ１４１１上のカーソルの動作を制御するために、限定としてではなく、マウス、トラックボール、フィンガーパッド、またはカーソル方向の列等のカーソル制御１４１３を含み得る。
【０１４０】
この発明の一実施例によると、個々の処理装置は、メインメモリ１４０８に含まれる１つ以上の命令の１つ以上のシーケンスを実行するそれぞれのプロセッサ１４０７によって特定の動作を行なう。このような命令は、ＲＯＭ１４０９または記憶装置１４１０等の別のコンピュータ使用可能な媒体から、メインメモリ１４０８内に読出すことができる。メインメモリ１４０８に含まれる命令のシーケンスを実行することにより、プロセッサ１４０７は、この明細書に記載されたプロセスを実行する。代替的な実施例では、ソフトウェア命令の代わりに、またはソフトウェア命令と組合せて、ハードワイヤード回路を用いてこの発明を実施することができる。したがって、この発明の実施例は、ハードウェア回路および／またはソフトウェアのどのような特定の組合せにも限定されない。
【０１４１】
この明細書で用いられる「コンピュータ使用可能な媒体」という用語は、情報を提供するか、またはプロセッサ１４０７による使用が可能な任意の媒体を指す。このような媒体は、限定としてではなく、不揮発性、揮発性および伝送の媒体を含む多くの形をとることができる。不揮発性媒体、すなわち、電力がなくても情報を保持することのできる媒体は、ＲＯＭ１４０９を含む。揮発性媒体、すなわち、電力がなければ情報を保持することのできない媒体は、メインメモリ１４０８を含む。伝送媒体は、バス１４０６を含む線を含む、同軸ケーブル、銅線および光ファイバを含む。伝送媒体は、情報信号を伝送するために、搬送波、すなわち、周波数、振幅または位相に関して変調可能な電磁波の形をとることもできる。加えて、伝送媒体は、電波および赤外線のデータ通信中に生じるもの等の、音波または光波の形をとることができる。
【０１４２】
コンピュータ使用可能な媒体の一般的な形には、たとえば、フロッピー（Ｒ）ディスク、フレキシブルディスク、ハードディスク、磁気テープ、他の任意の磁気媒体、ＣＤ－ＲＯＭ、他の任意の光学媒体、パンチカード、紙テープ、孔のパターンを有する他の任意の物理媒体、ＲＡＭ、ＲＯＭ、ＰＲＯＭ（すなわち、プログラマブル読出専用メモリ）、ＦＬＡＳＨ－ＥＰＲＯＭを含むＥＰＲＯＭ（すなわち、消去可能なプログラマブル読出専用メモリ）、他の任意のメモリチップもしくはカートリッジ、搬送波、または、プロセッサ１４０７がそこから情報を取出すことのできる他の任意の媒体が含まれる。コンピュータ使用可能な媒体のさまざまな形が、１つ以上の命令の１つ以上のシーケンスを実行のためにプロセッサ１４０７に与える際に関与してよい。メインメモリ１４０８によって取出された命令は、プロセッサ１４０７によって実行される前または後のいずれかに、記憶装置１４１０に任意に記憶され得る。
【０１４３】
また、各処理装置は、バス１４０６に結合された通信インターフェイス１４１４も含むことができる。通信インターフェイス１４１４は、それぞれのユーザステーション１４２４とホストコンピュータ１４２２との間の双方向の通信をもたらす。それぞれの処理装置の通信インターフェイス１４１４は、命令、メッセージおよびデータを含むさまざまな種類の情報を表わすデータストリームを含む電気的、電磁的または光学的な信号を送受信する。通信リンク１４１５は、それぞれのユーザステーション１４２４とホストコンピュータ１４２２とをリンクする。通信リンク１４１５は、ＬＡＮ１３２６であってよく、この場合、通信インターフェイス１４１４は、ＬＡＮカードであり得る。代替的に、通信リン

ク１４１５は、ＰＳＴＮ１３２８であってよく、この場合、通信インターフェイス１４１４は、統合サービスデジタル網（ＩＳＤＮ）カードまたはモデムであり得る。また、さらなる代替例として、通信リンク１４１５は、無線ネットワーク１３３０であってよい。処理装置は、そのそれぞれの通信リンク１４１５および通信インターフェイス１４１４を介して、プログラムの、すなわち、アプリケーションのコードを含むメッセージ、データおよび命令を送受信することができる。受信されたプログラムコードは、それが受信されたときにそれぞれのプロセッサ１４０７によって実行されてよく、および／または後の実行に備え、記憶装置１４１０または他の関連する不揮発性媒体に記憶されてよい。この態様で、処理装置は、メッセージ、データおよび／またはプログラムコードを搬送波の形で受信することができる。

【０１４４】

上述の明細書では、この発明の特定の実施例を参照してこの発明を説明してきた。しかしながら、この発明のより広い精神および範囲から逸脱することなく、この発明にさまざまな変更および変形を行なってよいことは明らかである。たとえば、読者は、この明細書に記載されたプロセスフロー図に示されたプロセス動作の特定の順序および組合せが、単に例示であって、異なった、もしくは追加のプロセス動作、またはプロセス動作の異なる組合せもしくは順序を用いてこの発明を実施できることを理解されるであろう。したがって、明細書および図面は、限定ではなく例示的な意味で捉えられるべきである。

【図面の簡単な説明】

【０１４５】

【図１】この発明の一実施例に従った事前作成の構成要素を示す図である。

【図２】この発明の一実施例に従った事前作成の方法のフロー図である。

【図３】この発明の一実施例に従った事前作成システムを示す図である。

【図４】この発明の一実施例に従った事前作成のサービスモジュールを示す図である。

【図５】この発明の一実施例に従った代替的な事前作成システムを示す図である。

【図６】この発明の一実施例に従った、ｎ個のノードの事前作成システムを示す図である。

【図７】この発明の一実施例に従った、ｎ個のノードの事前作成システムのための、事前作成の構成要素を示す図である。

【図８】この発明の一実施例に従ったＰＲＢのスキーマを示す図である。

【図９】この発明の一実施例に従った事前作成のポリシースキーマを示す図である。

【図１０】ユーザページテーブルに対する例示的な索引を特定する図である。

【図１１】この発明の一実施例に従った開始ローダに対するプロセスフローを示す図である。

【図１２】この発明の一実施例に従った、ＰＲＢを構築するためのプロセスのフロー図である。

【図１３】それを用いてこの発明を実施することのできるシステムアーキテクチャの図である。

【図１４】それを用いてこの発明を実施することのできるシステムアーキテクチャの図である。

【図１】

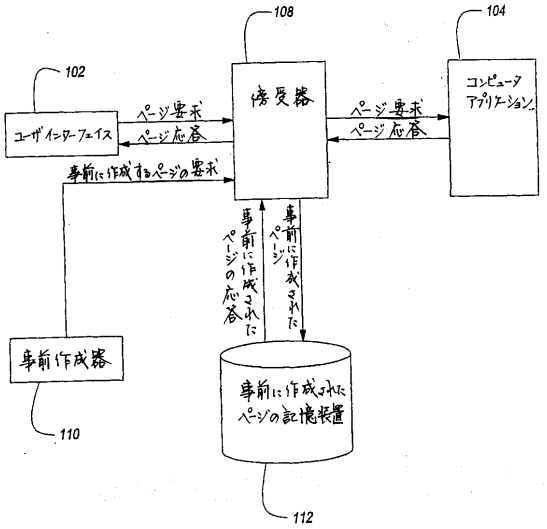

FIG. 1

【図２】

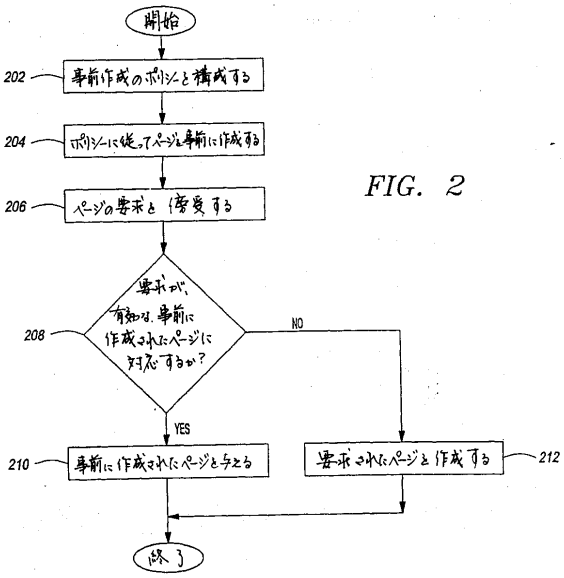

FIG. 2

【図３】

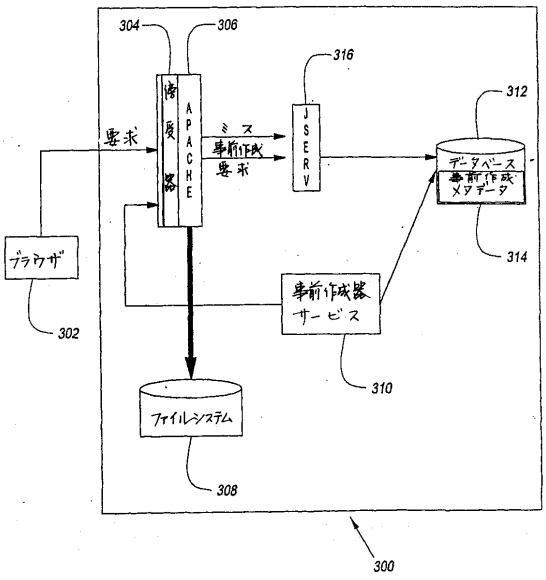

FIG. 3

【図４】

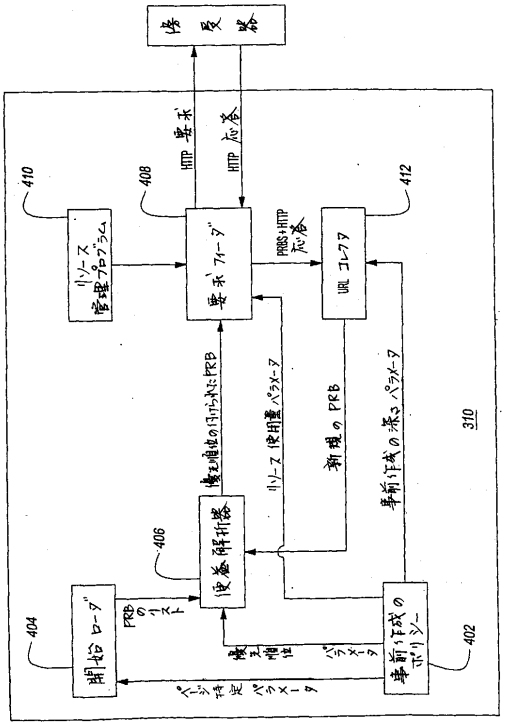

FIG. 4

【図５】

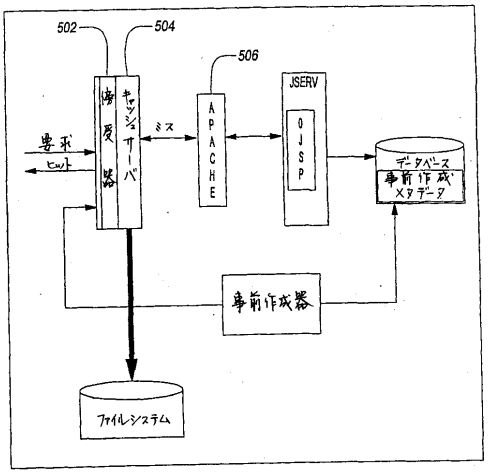

FIG. 5

【図６】

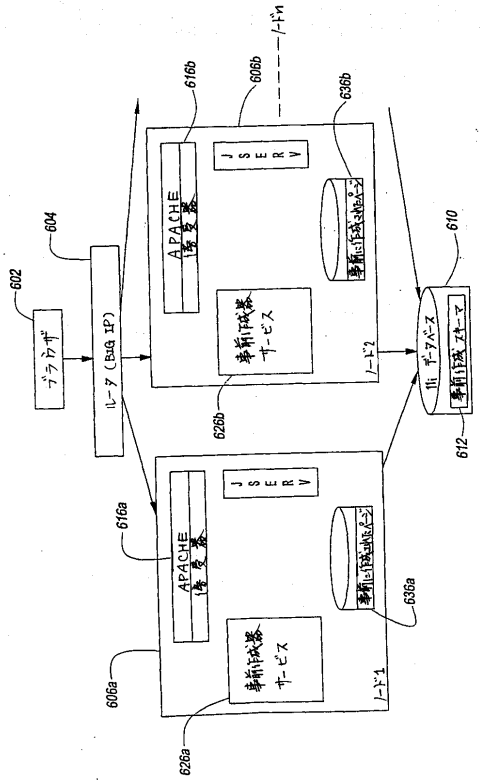

FIG. 6

【図７】

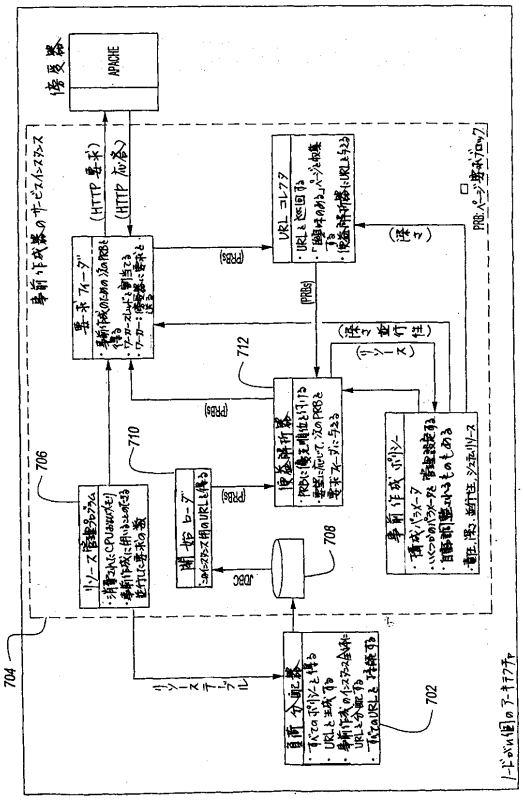

FIG. 7

【図８】

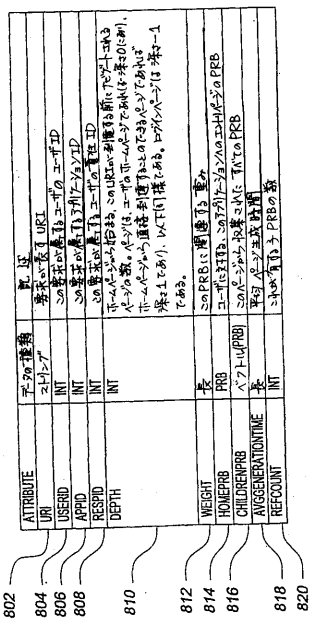

FIG. 8

【図９】

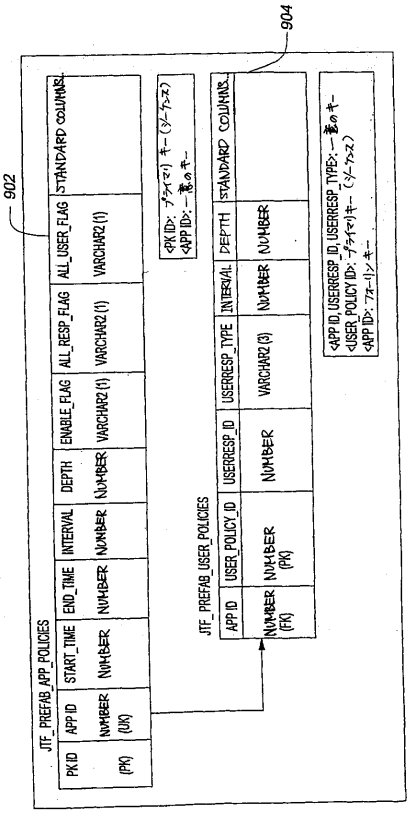

FIG. 9

【図１０】

索引 テーブル	記 述
重み 索引	PRBの重みによる、索引 PRB
深さ 索引	PRBの深さによる、索引 PRB
URI サブストリング 索引	PRBの URI サブストリングによる、索引 PRB。ここで、URI サブストリングは、PRBの .JSP 名として(推定される(E.G. JTFAVALID.JSP)。
ID 索引	PRBの APPID, RESPID, USERIDの組合せによる、索引 PRB
URI 索引	PRBの URIによる、索引 PRB。この属性は常に一意であるため、この索引のデータ構造は、ベクトルのハッシュテーブルではなく、単純な、PRBのハッシュテーブルである。

FIG. 10

【図１１】

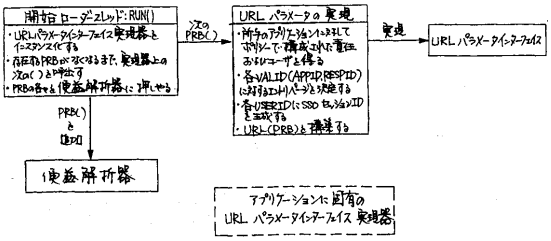

FIG. 11

【図１２】

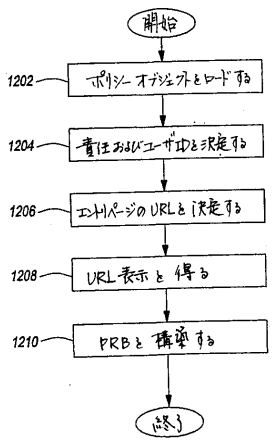

FIG. 12

【図１３】

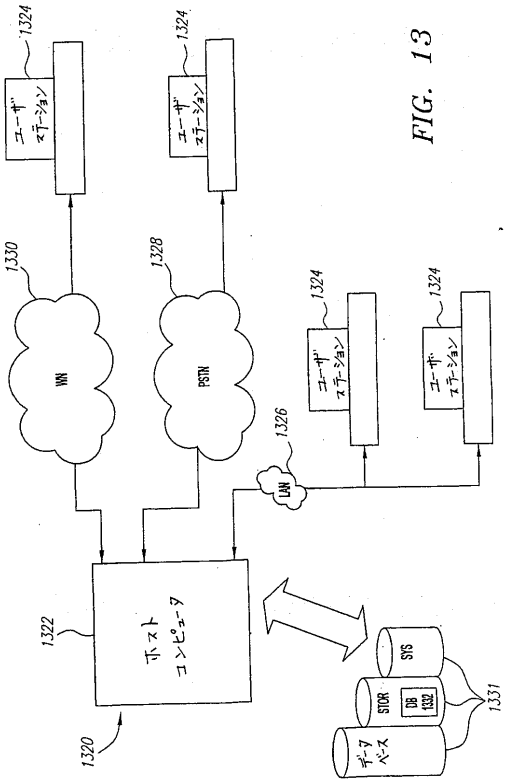

FIG. 13

【図１４】

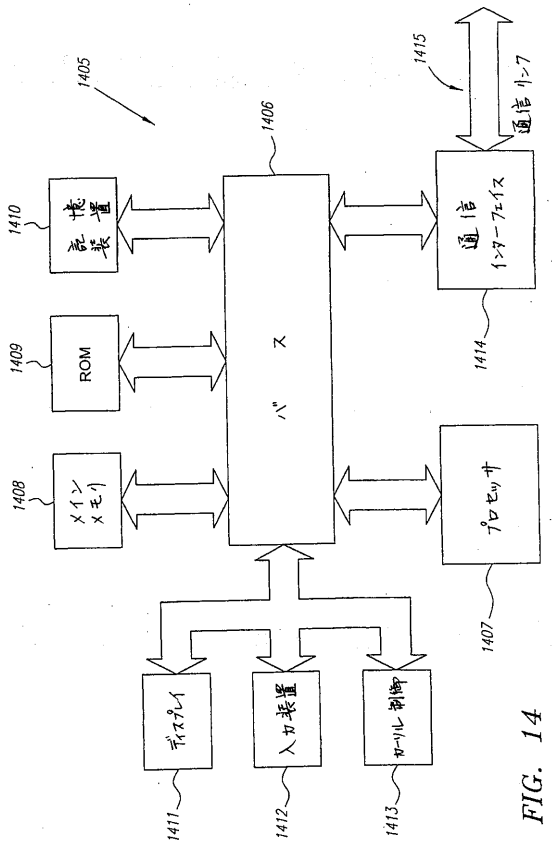
1405
1406
1415
通信リンク
1410
記憶装置
通信インターフェイス
1414
1409
ROM
バス
1408
メインメモリ
プロセッサ
1407
ディスプレイ
入力装置
カーソル制御
1411
1412
1413

FIG. 14

フロントページの続き

(74)代理人 100098316
弁理士 野田 久登
(74)代理人 100109162
弁理士 酒井 將行
(72)発明者 スブラマニアン，ソウミャ
アメリカ合衆国、９４３０１ カリフォルニア州、パロ・アルト、ホーソーン・アベニュ、５４８
(72)発明者 スンカラ，ラム
アメリカ合衆国、９４０２４ カリフォルニア州、ロス・アルトス、ヘイマン・プレイス、９４０
(72)発明者 カプール，クナル
アメリカ合衆国、９４０７０ カリフォルニア州、サン・カルロス、エルム・ストリート、７４５、ナンバー・５
(72)発明者 ライ，アンソニー
アメリカ合衆国、９５０１４ カリフォルニア州、キューパーティーノ、イートン・プレイス、２１８６０
(72)発明者 シッディクイ，サリム
アメリカ合衆国、９４５３６ カリフォルニア州、フリーモント、ディエゴ・ドライブ、３６４８１
(72)発明者 ウォン，サニー
アメリカ合衆国、９４４０３ カリフォルニア州、サン・マテオ、ディ・マリナ・コート、１７２１
(72)発明者 ビュン，ヒュンーシク
アメリカ合衆国、９４０６１ カリフォルニア州、レッドウッド・シティ、ヘス・ロード、１４９１、ナンバー・３０７

合議体
審判長 清田 健一
審判官 須田 勝巳
審判官 松尾 俊介

(56)参考文献 特開平１０－２０７７５９（ＪＰ，Ａ）
特開２０００－２５０８０３（ＪＰ，Ａ）
特開２００１－１０９７６０（ＪＰ，Ａ）
特開平１１－１４９４０５（ＪＰ，Ａ）
特表２００２－５２７８１８（ＪＰ，Ａ）
特開２０００－２０７３３１（ＪＰ，Ａ）
特開２０００－９０００１（ＪＰ，Ａ）
特開平１１－１９４９８３（ＪＰ，Ａ）
特開平１０－２１１６５（ＪＰ，Ａ）
特開２０００－８２０３９（ＪＰ，Ａ）
栗原 新，モバイルコンテンツビジネスに新機軸！，ｍｏｂｉｌｅ ｍｅｄｉａ ｍａｇａｚｉｎｅ，第6巻，第１０号，日本，株式会社シーメディア，１９９８年9月１３日，ｐ６６～６７

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F13/00