



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) DE 600 29 914 T2 2007.01.18

(12)

Übersetzung der europäischen Patentschrift

(97) EP 1 111 517 B1

(51) Int Cl.⁸: **G06F 17/30** (2006.01)

(21) Deutsches Aktenzeichen: 600 29 914.7

(96) Europäisches Aktenzeichen: 00 311 276.0

(96) Europäischer Anmeldetag: 15.12.2000

(97) Erstveröffentlichung durch das EPA: 27.06.2001

(97) Veröffentlichungstag
der Patenterteilung beim EPA: 09.08.2006

(47) Veröffentlichungstag im Patentblatt: 18.01.2007

(30) Unionspriorität:
470126 22.12.1999 US

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Xerox Corp., Rochester, N.Y., US

(72) Erfinder:
Glance, Natalie S., 38240 Meylan, FR; Huberman, Bernardo A., Palo Alto, California 94301, US; Adamic, Lada A., Redwood City, California 94062, US; Pitkow, James E., Palo Alto, California 94306, US

(74) Vertreter:
Grünecker, Kinkeldey, Stockmair & Schwanhäusser, 80538 München

(54) Bezeichnung: **System und Verfahren zum Zwischenspeichern**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingereicht, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

[0001] Diese Erfindung bezieht sich allgemein auf ein System und ein Verfahren zum Optimieren von Endbenutzer- und -Netzwerk-Ressourcen in Netzwerksystemen, wo verschiedene Niveaus eines Cache-Speichers existieren.

[0002] Netzwerksysteme, wie beispielsweise Intranet und das World Wide Web (das Web), setzen ein Cache-Speichern (Caching) (die Verwendung von Ressourcen, ähnlich einem Speicher, einer Platte, usw., mit einer sehr kurzen Zugriffszeit für häufig benutzte Datendateien) unter verschiedenen Niveaus ein, um Netzwerk-Belastungen und Ansprechzeiten zu verringern. Allgemein kann irgendeine Art einer Datendatei oder eines Dokuments cache-mäßig gespeichert werden, einschließlich Text, Graphiken, Video, Bildern, Audio-Dateien, html-Dateien, Web-Seiten, usw.. Ein Web-Cache-Speichern speichert typischerweise den Inhalt, zugeordnet zu URLs, für Web-Seiten.

[0003] In Client-Server-Informations-Systemen werden lokale Client-Ressourcen, meistens typischerweise ein Speicher (RAM), in großem Umfang verwendet, um Daten cachemäßig zu speichern und um eine zukünftige Interaktion mit Servern zu minimieren. Wenn ein Cache-Speicher verwendet wird, prüft der Client-Computer zu Anfang, ob das angeforderte Dokument in dem lokalen Cache-Speicher vorhanden ist. Falls das Dokument in dem lokalen Cache-Speicher existiert und es aktuell ist (wobei aktuell bedeutet, dass eine neuere Version des Dokuments nicht existiert), dann wird das Dokument unmittelbar zu dem Benutzer geliefert. Ansonsten ruft, wenn das Dokument nicht im Cache-Speicher vorhanden ist, der Client-Computer das Dokument von einem Server, der irgendwo auf dem System oder dem Web vorhanden ist, ab. In Abhängigkeit von der Dokumentengröße und der verfügbaren Übertragungsrate könnte die Lieferung des Dokuments an den Benutzer einen wesentlichen Umfang an Zeit benötigen.

[0004] Eine Bestimmung, welche Dokumente cachemäßig zu speichern sind, und wann sie zu ersetzen sind, wird durch einen Cache-Speicherungs-Ersetzungs-Algorithmus erreicht. Die meisten Web-Server, nach dem Stand der Technik, Proxy- und Client-Caching-Ersetzungs-Algorithmen berücksichtigen typischerweise Variablen, wie beispielsweise die Häufigkeit eines Zugriffs, die Neuheit eines Zugriffs, eine Datendatei (Dokumentengröße) und eine abgeschätzte Dokumenten-Aussuch-Latenz-Zeit, um die Priorität eines Dokuments in dem Cache-Speicher zu bestimmen. Die primäre Annahme hinter diesen Parametern ist diejenige, dass eine frühere Benutzung ein zuverlässiger Prophet (predictor) eines zukünftigen Zugriffs sein wird. Diese Parameter ermöglichen auch Cache-Ersetzungs-Policen, um Tref-

ferraten mit Byte-gewichteten Trefferraten und einer Endbenutzer-Latenz-Zeit zu optimieren. Neuere Algorithmen setzen diese Variablen in adaptiven Frameworks, da globale Gewichtungen als nicht optimal für individuelle Cache-Speicher und Verkehrsmusteränderungen über die Zeit befunden worden sind, ein.

[0005] Verschiedene andere Caching-Techniken, wie beispielsweise die Verwendung eines Rückrufs (Callback) und eines Vorabrufs (Prefetching), sind vorgeschlagen worden. Diese Techniken verbessern allerdings nicht immer die Funktionsweise. Ein Callback ist zum Beispiel nicht für Web-Objekte geeignet, die in vielen Proxies cachemäßig gespeichert werden können. Ein Prefetching ist auch dort ungeeignet, wo Cache-Trefferraten niedrig sind. Es ist oftmals schwierig, vorherzusagen, wann eine bevorrechtigte Dokumentenprüfung die Funktionsweise verbessern wird. Einige Programme sind verfügbar, die einem Benutzer ermöglichen, URLs von dem Web auf deren Client-Computern zu identifizieren und herunterzuladen und den Ablauf dieser Herunterladungen zu spezifizieren. Solche Programme erfordern allerdings, dass Benutzer spezifisch identifizieren, welche Dokumente vorab abgerufen werden sollen und cachemäßig gespeichert oder für eine spätere Benutzung heruntergeladen werden sollen, was nicht immer angenehm ist.

[0006] In Bezug auf das Problem, wie Dokumente zu priorisieren sind, wird entschieden, wie oft die Dateien zu validieren oder zu erneuern sind (d.h. um sicherzustellen, dass die Dateien aktuell sind). Allgemein sind Proxy-Server so konfiguriert, um eine Validierung entweder unter vorab eingestellten Intervallen oder auf einer auf Anforderung geleiteten Art durchzuführen. Vorab eingestellte Intervalle können unzureichend sein, um den Cache-Speicher in einer langen Suche zu aktualisieren. Und nicht alle Benutzer werden manuell den Cache-Speicher (durch Erneuern der Web-Seite) aktualisieren.

[0007] Ein anderes, wichtiges Problem in Informati onssuchsystemen ist dasjenige, wie Endbenutzer-(Client) und Netzwerk-Ressourcen in Netzwerksystemen zu optimieren sind, wo verschiedene Niveaus eines Caching existieren. In Bezug auf das Web tritt ein Caching an Client-Browsern (ein In-Speicher-Caching und ein auf einer Datei basierendes Caching) an Proxy-Cache-Speichern (z.B. ISPs, Satelliten, Corporate Firewalls, usw.) in Hierarchien von Proxy-Cache-Speichern (z.B. das National Laboratory for Applied Networks Research's caching network, @HOME's caching network, usw.) und in Web-Servern (z.B. Harvest server cache accelerator) auf. In Intranetzen kann ein Caching an den Client-Computern und den Netzwerk-Servern auftreten, die auf entfernte Datenbanken und Aufbewahrungsstellen zugreifen können. Die meisten Maßnah-

men, um ein Cache-Speichern vorzunehmen, versuchen entweder eine Endbenutzer-Latzen-Zeit zu verringern oder einen Netzwerkverkehr, verbraucht beim Zuführen des erwünschten Inhalts, zu verringern. Diese Maßnahmen sind in ihrer Art global und, wie vorstehend angegeben ist, nicht optimal.

[0008] KAISER C ET AL: „A workgroup model for smart pushing and pulling“ ENABLING TECHNOLOGIES: INFRASTRUCTURE FOR COLLABORATIVE ENTERPRISES, 1999. (WET ICE '99). PROCEEDINGS, IEEE 8TH INTERNATIONAL WORKSHOPS ON STANFORD, CA, USA, 16.–18. Juni 1999, LOS ALAMITOS, CA, USA, IEEE COMPUT. SOC, US, 16. Juni 1999 offenbart ein Caching-System, das dem Client ermöglicht, die Empfehlung von Dokumenten zu kontrollieren.

[0009] KURCEWICZ M ET AL: „A filtering algorithm for Web caches“ COMPUTER NETWORKS AND ISDN SYSTEMS, NORTH HOLLAND PUBLISHING. AMSTERDAM, NL, vol. 30, no. 22–23, 25. November 1998 (1998-11-25), Seiten 2203–2209, beschreibt einen Filterungsalgorithmus zum Steuern eines Caching basierend auf dem Konzept von nur Caching-Objekten für gemeinsam geteilte Server.

[0010] Gemäß einem ersten Aspekt der vorliegenden Erfindung weist ein Verfahren zum Bestimmen, welche Elemente in einem Cache-Speicher zu speichern sind, auf:

Erhalten von Wert-Daten für jedes Element, um in dem Cache-Speicher gespeichert zu werden, wobei die Wert-Daten ein Maß des Werts der Elemente aufweisen;

Priorisieren der Elemente, die in dem Cache-Speicher gespeichert werden sollen, entsprechend zu Gewichtungsmetriken, bestimmt für jedes Element, wobei die Gewichtungsmetriken von den Wert-Daten jedes Elements abgeleitet sind; und

Speichern solcher Elemente, die die höchste Gewichtungsmetrik haben, in dem Cache-Speicher, wobei die Elemente Web-Seiten aufweisen und wobei die Wert-Daten eine gewichtete Kombination von statistischen Informationen, die sich auf die Anzahl von eindeutigen Benutzern beziehen, die die Web-Seite, oder die Web-Stelle, die der Web-Seite zugeordnet ist, besuchen, und von Empfehlungs-Daten von einem Empfehlungssystem, das einen Wert für ein Element entsprechend zu Benutzer-Empfehlungen der Web-Seite oder der zugeordneten Web-Stelle bereitstellt, aufweisen, dadurch gekennzeichnet, dass der Priorisierungsschritt ein Evaluieren der Gewichtungsmetrik $w(t_i)$ für ein spezifisches Element (i) aufweist, wobei

$$w(t_i) = \frac{1}{MTNR_i},$$

wobei $MTNR_i = \alpha' t_i + (1 - \alpha') MTNR_{i-1}$ und

$$\alpha' = \frac{\alpha}{f(v_i)},$$

MTNR_i

die Durchschnittszeit bis zu der nächsten Anforderung für Element i ist, t_i die Zeit seit der letzten Anforderung für Element i ist, $MTNR_{i-1}$ die vorherigen Wert-Daten sind, α' der exponentielle Dämpfungskoeffizient ist, v_i die Wert-Daten von Element i sind, wie dies durch das Empfehlungssystem bereitgestellt ist, α ein Dämpfungskoeffizient ist und $f(v_i)$ eine monoton steigende Funktion der Wert-Daten ist.

[0011] Gemäß einem zweiten Aspekt der vorliegenden Erfindung weist ein System für ein Cache-Speichern von Elementen auf einem verteilten Netzwerk auf:

einen Computer zum Speichern und zum Aufsuchen von Elementen von dem Netzwerk, wobei der Computer einen Prozessor und einen Cache-Speicher zum Speichern von Elementen, aufgesucht von dem Netzwerk; und ein Wert-Modul zum Bereitstellen von Wert-Daten für Elemente, die in dem Cache-Speicher gespeichert werden sollen; umfasst;

wobei jedes Element, gespeichert in dem Cache-Speicher, Wert-Daten umfasst, die ein Maß des Werts der Elemente aufweisen;

wobei der Prozessor, auf eine Anforderung für ein bestimmtes Element hin, bestimmt, ob das Element in dem Cache-Speicher gespeichert ist, und falls nicht, das Element von dem Netzwerk aufsucht;

wobei der Prozessor, für jedes aufgesuchte Element, auf eine Gewichtungsmetrik, abgeleitet von den Wert-Daten des aufgesuchten Elements, ansprechend, das aufgesuchte Element und die Elemente, gespeichert in dem Cache-Speicher, entsprechend der Gewichtungsmetrik jedes Elements priorisiert; und solche Elemente, die die höchste Gewichtungsmetrik haben, in dem Cache-Speicher speichert, wobei

die Elemente Web-Seiten aufweisen und wobei die Wert-Daten eine gewichtete Kombination von statistischen Informationen, die sich auf die Anzahl von eindeutigen Benutzern, die die Web-Seite oder die Web-Stelle, zugeordnet zu der Web-Seite, besuchen, und von Empfehlungs-Daten von einem Empfehlungssystem aufweisen, das einen Wert für ein Element entsprechend zu Benutzer-Empfehlungen der Web-Seite oder der zugeordneten Web-Stelle bereitstellt, wobei der Prozessor so angepasst ist, um die Evaluierung der Gewichtungsmetrik $w(t_i)$ für ein spezifisches Element (i) auszuführen, wobei

$$w(t_i) = \frac{1}{MTNR_i},$$

wobei $MTNR_i = \alpha' t_i + (1 - \alpha') MTNR_{i-1}$ und

$$\alpha' = \frac{\alpha}{f(v_i)},$$

MTNR_i

die Durchschnittszeit bis zu der nächsten Anforderung für Element i ist, t_i die Zeit seit der letzten Anforderung für das Element i ist, MTNR_{i-1} der vorherige Wert für MTNR für Element i ist, αⁱ der exponentielle Dämpfungskoeffizient ist, v_i die Wert-Daten von Element i sind, wie dies durch das Empfehlungssystem bereitgestellt ist, α ein Dämpfungskoeffizient ist und f(v_i) eine monoton ansteigende Funktion der Wert-Daten ist.

[0012] Die Erfindung schafft ein verbessertes System und ein Verfahren für ein Cache-Speichern und ist besonders für ein Cache-Speichern von Informationen auf dem Web nützlich und verbessert die Funktionsweise und den Netzwerkverkehr. Ein verbessertes Cache-Speichern kommt von der Benutzung von Qualität- und Wertattributen, bereitgestellt z.B. durch ein Empfehlungssystem oder eine dynamische Analyse von Stellenzugriffen, die an cachemäßig gespeicherten Informationen angehängt sind.

[0013] Das System und das Verfahren der Erfindung erkennt, dass, auch wenn ansonsten alles gleich ist, höherwertige Dokumente wahrscheinlich öfters aufgesucht werden. Derzeitige Cache-Algorithmen berücksichtigen nicht den tatsächlich wahrgekommenen Nutzen oder Wert der Dokumenteninhalte, die cachemäßig gespeichert werden. Die Erfindung priorisiert Dokumente in dem Cache-Speicher entsprechend dem relativen Wert deren Inhalts. In einer Ausführungsform setzt das System der Erfindung ein Empfehlungssystem (oder ein Zusammenarbeiten des Filtersystems) ein, das einen Wert für ein Dokument entsprechend zu Benutzerempfehlungen liefert. In Bezug auf das Web umfasst das System der Erfindung Verfahren zum Bestimmen des Werts für individuelle Web-Seiten ebenso wie für individuelle Web-Stellen.

[0014] Das System kann explizite Empfehlungen, implizite Empfehlungen oder eine Kombination der zwei verwenden, um die höherwertigen Dokumente zu identifizieren. Das System verwendet die Maße einer Qualität, um effizienter zu priorisieren, welche Dokumente, wie beispielsweise URLs, vorzugsweise cachemäßig zu speichern sind. Das System der Erfindung kann als ein „demokratisches Caching“ Schema angesehen werden, d.h. ein Caching-Schema, das sich weg von früheren Techniken verschiebt, die Netzwerk-Ressourcen (die im Wesentlichen unabhängig von dem Wert des Dokuments für den Benutzer sind) zu einer neuen Technik optimieren, die Ressourcen für alle Benutzer optimiert. Durch Einsetzen eines demokratischen Caching-Systems werden Dokumente cachemäßig, basierend auf deren Wert für Benutzer, und nicht basierend zum Beispiel auf der Anzahl von Treffern (die aufgrund des bestimmten Indexierungs-Algorithmus, verwendet durch den Web-Crawler auftreten können) oder im Wesentli-

chen vom Benutzer unabhängigen Maßen, gespeichert. Auf diese Art und Weise ermöglicht ein demokratisches Cache-Speichern allen Benutzern des Systems, von einem Cache-Speichern bzw. Caching zu profitieren, nicht nur für solche Benutzer, die die meisten Ressourcen verbrauchen oder auf eine begrenzte Anzahl von Seiten übermäßig zugreifen.

[0015] Empfehlungssysteme werden in Intranet-Informationen teilende Anwendungen für Organisationen und für Internet-Informationen teilende Anwendungen für die Öffentlichkeit verwendet. In einem Empfehlungssystem ist die Qualität des Empfehlungsdienstes von einer primären Wichtigkeit und eine Zugriffszeit ist ebenso wichtig. Das vorgeschlagene System und das Verfahren eines Cache-Speichers kann auch für ein Verwalten des Cache-Speichers in einem Empfehlungssystem verwendet werden. Empfehlungen in einem Empfehlungssystem können entsprechend deren Wert priorisiert werden, ob sie nun explizit oder implizit abgeleitet sind. Neuere Empfehlungen, die durch Benutzer gemeinsam geteilt sind, werden für Zeitperioden cachemäßig gespeichert, die sich mit deren Werten erhöhen, was demzufolge die Funktionsweise des Empfehlungssystems für seine Benutzer durch Verringerung der Zugriffszeit zu empfohlenen Elementen verbessert. Vergange Empfehlungen werden dazu tendieren, dass sie von dem Cache-Speicher verschwinden, obwohl ein Wiederaufleben des Interesses bewirken wird, dass das Dokument (oder URL) für eine gewisse Zeit in dem Cache-Speicher existiert. Durch Verbinden der Cache-Speicher in einem Dokument-Management-System mit einem Empfehlungssystem für gemeinsam geteilte Informationen schafft die Erfindung eine verbesserte Funktionsweise für beide.

[0016] Für eine Ausführung, die ein Zugreifen auf auf dem Web-basierenden Dokumenten oder Web-Seiten einsetzt, könnte das System und das Verfahren mit einem oder einer Kombination von zwei spezifischen Verfahren zum Identifizieren des Werts der Web-Seiten (oder URLs) ausgeführt werden. Das Web ist der Bereich des Internets, der dazu verwendet wird, verknüpfte Dokumente zu speichern und dann auf diese zu zugreifen. Eine Web-Stelle ist ein generischer Ausdruck für verschiedene Typen von Web-Orten. In einem Fall ist eine Web-Stelle eine Zusammenstellung von dazu in Bezug stehenden Web-Seiten, die einer Gesamtheit zugeordnet sind, die eine Präsenz auf dem World Wide Web besitzen, wie beispielsweise eine Firma, ein Lehrinstitut, oder dergleichen. Alternativ kann eine Web-Stelle ein „Portal“ sein, um Benutzern des Webs einen Einstiegspunkt in das World Wide Web bereitzustellen. Eine Web-Stelle kann auch eine Aufbewahrungsstelle oder eine Informationsquelle sein, die Zusammenstellungen von Dokumenten enthält, auf die über das Web zugegriffen werden kann. Allgemein wird eine

Web-Stelle eine URL auf hohem Niveau (z.B. ein www.Abccompany.com) haben, wogegen die Seiten auf der Stelle Qualifizierer mit niedrigerem Niveau haben werden (z.B. www.Abccompany.com/page1.html).

[0017] In dieser Ausführungsform wird zwischen Web-Stellen und Web-Seiten unterschieden. Obwohl eine Web-Stelle eine URL auf hohem Niveau, die dazu zugeordnet ist, besitzt, wird hier zu Vereinfachung auf Web-Seiten und deren URLs gegeneinander austauschbar Bezug genommen. Es sollte angemerkt werden, dass sowohl explizite Empfehlungen als auch eine implizite Empfehlung in irgendeiner Kombination oder einer Bewertung von Web-Stellen und Web-Seiten (oder URLs) verwendet werden können.

[0018] Das erste Verfahren (ein Verfahren zum Bereitstellen einer impliziten Empfehlung) schafft eine Wert-Empfehlung auf dem Web-Stellen-Niveau (im Gegensatz zu der individuellen Web-Seite oder dem URL-Niveau) und wird garantiert, alle cachemäßig gespeicherten URLs, die den Web-Stellen zugeordnet sind, abzudecken. (Hier enthält der Cache nur Web-Stellen, die durch deren URLs aufgelistet sind.) Das erste Verfahren analysiert statistisch die Anzahl von einzigartigen Besuchern der Web-Stelle. Diese Web-Stellen in der höchsten Anzahl von eindeutigen Besuchern wird der höchste Wert zugesprochen. Der Vorteil eines Messwerts als eine Funktion von einzigartigen Besuchen im Gegensatz zu den gesamten Besuchen ist derjenige, dass die Neigung, hervorgerufen durch Spamming oder eine starke Benutzung der Web-Stelle durch nur ein paar Besucher, beseitigt wird. Eine Charakterisierung des Web-Verkehrs ergibt starke, zugeschnittene Verteilungen über den Umfang eines Verkehrs als eine Funktion der Anzahl von Benutzern, wobei ein paar Benutzer für die Mehrheit des Verkehrs steht (siehe Abdulla, C. „Analysis and Modeling of World Wide Web Traffic.“ Doctoral thesis, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, Mai 1998). Das vorgeschlagene Verfahren ist „demokratisch“ dahingehend, dass es die Wertzuweisung von individuellen Benutzern berücksichtigt und sicherstellt, dass die Vorteile eines Cache-Speichers nicht nur auf starke Benutzer beschränkt ist.

[0019] Das zweite Verfahren (ein Verfahren zum Bereitstellen einer expliziten Empfehlung) verbindet den Cache-Speicher (wie beispielsweise einen Proxy-Cache für eine Organisation) mit einer Datenbank von Benutzerbewertungen der individuellen Web-Seiten-URLs. Eine solche Datenbank kann, zum Beispiel, als eine Komponente eines Empfehlungssystems für Web-Seiten bereitgestellt werden, das Filtertechniken verwendet, um personifizierte Empfehlungen für individuelle Benutzer bereitzustellen. Das zweite Verfahren schafft Wertempfehlungen für indi-

viduelle Web-Seiten oder URLs (im Gegensatz zu Web-Stellen), allerdings nur für einen Teil der Web-Seiten-URLs in dem Cache-Speicher. Während dieser Teil klein für URLs in dem Web insgesamt sein kann, kann erwartet werden, dass dies wesentlich wird, wenn die Benutzung des Empfehlungssystems mit der Zeit zunimmt. Durch Verbinden des Proxy-Cache-Speichers mit einem Empfehlungssystem wird die Funktionsweise nicht nur des Cache-Speichers, sondern auch des Empfehlungssystems, verbessert. Durch Speichern von neueren, stark bewerteten Empfehlungen in dem Proxy-Cache-Speicher wird sich die durchschnittliche Zeit eines Herunterladens für empfohlene URLs verringern. Benutzer werden eine verbesserte Funktion sehen, was bewirkt, dass sie mehr Dokumente für das Empfehlungssystem bewerten, was dessen Effizienz ebenso verbessert.

[0020] Die zwei Verfahren können in Verbindung damit verwendet werden, den Wert einer Web-Seiten-URL zu evaluieren. Zum Beispiel wird, wenn das Empfehlungssystem einen Wert für eine bestimmte URL bereitgestellt hat, dieser Wert zuerst verwendet. Ansonsten ist der Wert, der der URL zugeordnet ist, der Web-Stellen-Wert, der durch das erste Verfahren vorhergesagt ist. Alternativ kann eine gewichtete Kombination eines Werts von jedem Verfahren verwendet werden. Ein anderes Verfahren für ein Cache-Ersetzen umfasst einen vorhergesagten Wert (durch irgendeines der vorstehenden Verfahren oder Kombinationen davon), ebenso wie die Neuheit und die Häufigkeit eines Zugriffs. Da es denkbar ist, dass es manchmal erwünscht sein kann, Netzwerk-Ressourcen zu bewahren oder die gesamte Latenz-Zeit zu minimieren, sollte es leicht ersichtlich sein, dass die vorgeschlagenen, demokratischen Cache-Techniken parallel mit traditionellere Verfahren verwendet werden können, um eine Netzwerk-Bandbreite oder Latenz-Zeit zu optimieren. Unter Verwendung dieses Schemas würde der Cache-Speicher zwischen Verfahren, basierend auf dem Netzwerk und anderen Systembedingungen, unter Verwendung von vordefinierten Schwellwerten oder adaptiven solchen, umschalten.

[0021] Ein Beispiel eines Systems und eines Verfahrens gemäß der vorliegenden Erfindung wird nun unter Bezugnahme auf die beigefügten Zeichnungen beschrieben, in denen:

[0022] [Fig. 1](#) zeigt ein Blockdiagramm eines Systems, das ein demokratisches Caching einsetzt;

[0023] [Fig. 2](#) zeigt ein Flussdiagramm, das ein Verfahren zum Zuweisen einer Priorität-Gewichtung zu einer URL beschreibt;

[0024] [Fig. 3](#) zeigt ein Flussdiagramm, das ein Verfahren zum Zuweisen von Caching-Prioritäts-Gewichtungen zu auferlegten URLs beschreibt; und

[0025] [Fig. 4](#) zeigt ein Blockdiagramm eines Empfehlungssystems, verwendet in dem System der [Fig. 1](#).

[0026] Unter Bezugnahme nun auf die Zeichnungen, und insbesondere auf [Fig. 1](#), wird ein System, das eine demokratisches Caching einsetzt, allgemein hier gezeigt und mit dem Bezugszeichen **10** identifiziert. Das System **10** umfasst einen Computer **12**, der einen Prozessor **22** und einen Cache-Speicher **24** besitzt. Auch ist in dem System **10** ein Wert-Modul **14** vorhanden, das Wert-Daten für jedes Element, gespeichert in dem Cache-Speicher **24**, liefert. Das System **10** ist unter Verwendung einer unterbrochenen Linie dargestellt, da das Empfehlungssystem **16** in dem System **10** umfasst sein kann. Das Empfehlungssystem **16** liefert Wertinformationen, die sich auf Elemente beziehen, die in dem Cache-Speicher **24**, basierend auf einer Benutzereingabe, gespeichert werden sollen. Der Computer **12** kann ein Client-Computer oder ein Proxy-Server oder irgendein anderer Computer, der einen Cache-Speicher einsetzt, sein. Der Computer **12** kann in irgendeinem verteilten Netzwerk, wie beispielsweise einem Intranet oder dem Internet (wie dies in [Fig. 1](#) dargestellt ist), arbeiten. Benutzer **40** greifen auf den Computer **12** und das Empfehlungssystem **16** über das Internet **30** zu, wie dies durch durchgezogenen Linien **31** angegeben ist. (Einige Benutzer **40** können auf den Computer **12** und das Empfehlungssystem **16** über eine bestimmte andere Form einer Kommunikation, wie durch die durchgezogene Linie **33** angezeigt, zugreifen.)

[0027] Der Computer **12** kann mit dem Wert-Modul **14** oder dem Empfehlungssystem **16** entweder direkt (dargestellt durch unterbrochene Linien **23** und **29**, jeweils) oder indirekt über das Internet **30** (dargestellt durch durchgezogene Linien **21**, **25** und **27**) kommunizieren. Das Wert-Modul **14** kann statistische Informationen, wie beispielsweise die Zahl von einzigartigen Benutzerbesuchern auf einer bestimmten Web-Stelle oder Web-Seite, bereitstellen.

[0028] Das Empfehlungssystem **16** kann Benutzerbewertungen oder einen Wert in irgendeinem oder mehreren der folgenden Punkte bereitstellen: eine Sternbewertung, die sich auf den Inhalt des Dokuments (URL's) bezieht, einen Vorgang aufgrund des Dokuments, ein Ablaufdatum von Informationen, die in dem Dokument enthalten sind, eine Expertenoption oder -bewertung über das Dokument (Web-Seite oder Web-Stelle) oder eine Bewertung von einer unabhängigen Organisation. Empfehlungen bestehen gewöhnlich aus numerischen Bewertungen, die manuell durch Benutzer eingegeben sind, allerdings können sie auch von einem Benutzerverhalten (z.B. Zeit, die beim Lesen eines Dokuments verbraucht wird, Aktionen, wie beispielsweise Drucken, Sichern oder Löschen eines Dokuments) abgeleitet werden.

Die Prämisse solcher Systeme ist diejenige, dass ein Benutzer ein Element bevorzugt, das ähnlich zu anderen Elementen, ausgewählt durch den Benutzer oder andere Benutzer, ist.

[0029] Das System und das Verfahren der Erfindung werden in größerem Detail nachfolgend unter Bezugnahme auf die nachfolgenden, bevorzugten Verfahren zum Bereitstellen von Wertempfehlungen beschrieben:

(1) implizite Stellenempfehlungen über eine statistische Analyse der Stellenbesuche; und (2) explizite URL-Empfehlungen über ein Empfehlungssystem. Das Verfahren wird auch in Bezug auf ein Modifizieren des Cache-Speichers beschrieben werden, um Wertempfehlungen zu berücksichtigen, und insbesondere wird ein Beispiel einer Caching-Methode, die den vorhergesagten Wert ebenso wie Neuheit und Häufigkeit eines Zugriffs berücksichtigt, beschrieben.

[0030] Eine dynamische Theorie von Empfehlungen, die Stellenbesuche durch Benutzer des World Wide Web vorhersagt, ist in neuerer Zeit durch einen Teil der Erfinder (Huberman, B. A. and Adamic, L. A. Novelty and Social Search in the World Wide Web. 1998), entwickelt worden. Huberman und Adamic zeigen, dass die Anzahl von Benutzern, die gegebene Stellen über Zeitperioden besuchen, ein universelles Gesetz mit einem Exponenten, der sich auf die Rate bezieht, unter der Benutzer neue Stellen selbst entdecken, ist. Eine umfangreiche, empirische Studie eines Benutzerverhaltens in dem Web, durchgeführt von Huberman und Adamic, bestätigte dieses Gesetz. Diese Ergebnisse können direkt verwendet werden, um effektive und faire Caching-Policen aufzubauen, die gegenüber einem Spamming oder einer umfangreichen Benutzung einer Stelle durch ein paar Einzelpersonen diskriminieren. Da die Anzahl von einzigartigen Stellenbesuchern einem skalierenden Gesetz folgt, bietet sich an, dass ein Caching entsprechend diesem Potenzgesetz durchgeführt werden sollte. Demzufolge werden die populärsten Stellen (für eine große Ansammlung von Benutzern) solche mit der größten Zahl von Treffern aufgrund nur ein paar Benutzern in dem Rang höher stellen.

[0031] In dem System der [Fig. 1](#) wird das System **10** berücksichtigt, in dem ein Modul **14** für einen implizierten Wert in Verbindung mit einem Proxy-Server **12** arbeitet. Das Modul **14** analysiert das Log von Client (Benutzer **40**) Zugriffen auf Web-Stellen des Internets **30**. Zuerst extrahiert es die Anzahl von einzigartigen Benutzern zu der Stelle durch Entfernen von Mehrfachbesuchen durch denselben Benutzer. Dann ordnet es eine prozentuale Bewertung jeder Stelle zu. Zum Beispiel besitzt eine Stelle mit einem prozentualen Rating von 75% viele oder mehr eindeutige Benutzer als 75% der Stellen, aufgestellt durch das Proxy. Diese prozentuale Bewertung wird zu dem

Modul **14** als ein Maß des Werts einer Stelle und/oder irgendeiner URL, zugeordnet zu der Stelle, zurückgeführt. (Der Proxy-Server **12** speichert URL-Adressen in dem Cache-Speicher **24** entsprechend dieser perszentilen bzw. prozentualen Bewertung.)

[0032] Das Modul **14** zum Berechnen implizierter Stellenempfehlungen kann getrennt vorhanden sein, wie dies in [Fig. 1](#) dargestellt ist, oder es kann innerhalb des Proxy-Servers **12** (oder des Cache-Speichers **24**) vorhanden sein und ist als Modul **28** in [Fig. 1](#) dargestellt. Das Modul **14** oder **28** bildet eine Anwendungsprogrammschnittstelle (API), die dem Prozessor **22** ermöglicht, den Cache-Speicher **24** für den vorhergesagten Wert irgendeiner URL, einer Liste von URLs, an irgendeiner Stelle oder einer Liste von Stellen, abzufragen. Zusätzlich ermöglicht die API dem Prozessor **22**, das Modul **14** oder **28** nach dem Bereich möglicher Werte, zurückgeführt durch das Modul, abzufragen.

[0033] Ein allgemeines Empfehlungssystem umfasst typischerweise drei Module: einen Datenspeicher, eine Wertvorhersage und -Bestellung. [Fig. 4](#) zeigt ein Empfehlungssystem **16** für einen Datenspeicher **52**, ein Wert-Vorhersagemodul **54** und ein Präsentations- und Bestellungsmodul **56**. Für URLs umfasst das Datenspeichermodul **52**, minimal, das Datum eines Eintritts, Benutzer-IDs und Benutzerbewertungen (oder Präferenz-Bewertungen) der URLs. Das Wert-Vorhersagemodul **54** umfasst einen Algorithmus für die Vorhersage des Werts, oder einen relativen Rang, des URL für einen Benutzer. Das Präsentation- und Bestellungsmodul **56** bestellt die Präsentation der empfohlenen URLs entsprechend der vorhergesagten Werte, oder vorhergesagter relativer Ränge.

[0034] Das Empfehlungssystem **50** umfasst auch eine Wertvorhersage **58**, die den Wert der URL für einen allgemeinen Benutzer vorhersagt. Allgemein nimmt das Modul **58** diese Vorhersage in Bezug auf eine gewichtete Summe von Bewertungen für Benutzer für eine URL vor. In dem einfachsten Fall wird diese Vorhersage einfach der Durchschnitt aller Bewertungen, die für eine URL empfangen sind, sein. Alternativ könnten die Gewichtungen zu Bewertungen von Benutzern hin gewichtet werden, deren Meinung objektiv höher durch andere Benutzer bewertet ist. Eine solche objektive Funktion für die Berücksichtigung des Werts der Meinung eines Benutzer könnte eine Funktion der Summe der Korrelation des Benutzers mit allen anderen Benutzern sein; alternativ (oder zusätzlich) könnte sie die Anzahl von Treffern von Empfehlungen eines Benutzer, die empfangen sind, berücksichtigen.

[0035] Das Modul **58** liefert eine API, die ermöglicht, den Cache-Speicher **24** nach dem vorhergesagten Wert für eine oder eine Liste von URLs, vorgesehen

als Parameter, abzufragen, ermöglicht, den Cache-Speicher **24** nach einer Liste von URLs und vorhergesagten Werten abzufragen, unter Weiterführen als einen optionalen Parameter eines Datum-Werts. In dem letzteren Fall führt, wenn ein Datum-Wert vorgesehen ist, das Modul **58** nur URLs zurück, deren Datum eines Eintritts in den Datenraum des Empfehlungssystems größer als das Datum, spezifiziert durch den Cache-Speicher, ist. Schließlich ermöglicht die API dem Cache-Speicher **24**, das Modul **58** nach dem Bereich möglicher Werte, zurückgeführt durch das Modul, abzufragen. Die < URL, Wert > Paare, zurückgeführt durch das Modul **58**, können so erweitert werden, um zusätzliche Informationen über die URLs, gespeichert in dem Datenraum, zu umfassen. Ein Beispiel von Informationen einer potentiellen Benutzung zu dem Cache-Speicher sind Benutzer-Eingabe-Ablauf-Daten oder eine Modifikations-Periodizität. Der Cache-Speicher **24** kann dann diese Werte verwenden, um effizienter URLs zu löschen und eine Validierung durchzuführen.

[0036] Es wird angenommen, dass der Proxy-Server **12** erneut eine Caching-Priorität zu jeder URL, gespeichert in dem Cache-Speicher **24**, periodisch zuführt. Wenn der Cache-Speicher **24** voll ist, gibt er dann die Elemente mit niedrigster Priorität frei, bis genug Speicher vorhanden ist, um das neue Element in dem Cache-Speicher zu speichern. Der Proxy-Server **12** kann auch den Cache-Speicher **24** in Verbindung mit den zwei Modulen (Wert-Modul **14** und Empfehlungssystem **16**) betreiben, was implizite und explizite Empfehlungen liefert. Indem die Flexibilität der API für die zwei Module gegeben ist, sind zahlreiche Variationen möglich.

[0037] Ein Flussdiagramm, das ein Verfahren zum Zuweisen einer Prioritäts-Gewichtung zu einer URL darstellt, ist in [Fig. 2](#) gezeigt. Wenn der Prozessor **22** erneut die Priorität einer URL evaluiert, fragt er zuerst das explizite Empfehlungsmodul **16** nach seinem vorhergesagten Wert der URL ab (Schritt **62**). Wenn das explizite Empfehlungsmodul in der Lage ist, eine Empfehlung vorzunehmen (Schritt **64**), dann geht der Prozessor dazu über, die Priorität, unter Verwendung dieses Wertes zu berechnen (Schritt **68**), wie dies nachfolgend unter Bezugnahme auf [Fig. 3](#) beschrieben ist. Ansonsten fragt er das implizite Empfehlungsmodul **14** nach seinem vorhergesagten Wert der URL ab (Schritt **66**) und verwendet dann diesen Wert, um die Priorität der URL zu berechnen. Der Grund zum Durchführen des Werts, vorhergesagt durch die explizite Empfehlung, gegenüber einer impliziten Empfehlung, ist derjenige, dass der erstere spezifisch für die URL ist, während der letztere über die gesamte Stelle generalisiert. Es ist allerdings anzumerken, dass gerade im Fall einer impliziten Empfehlung die Caching-Prioritäts-Berechnung URL spezifisch sein wird, da sie die Neuheit und die Häufigkeit eines Zugangs für die bestimmte URL berücksichtigt.

[0038] Reddy, M. und G. P. Fletcher „An Adaptive Mechanism for Web Browser Cache Management“, IEEE Computing Jan./Feb. 1998, beschreibt einen Caching-Priorität-Algorithmus, der eine Gesamthäufigkeit einer Benutzung, ebenso wie Neuheit, unabhängig einer Dateigröße, berücksichtigt. In einem Modul **54/58** des Empfehlungssystems **16** wird der Algorithmus von Reddy und Fletcher modifiziert, um den aufgesuchten Wert der URL zu berücksichtigen. Reddy und Fletcher schätzen die Durchschnittszeit der nächsten Anforderung (MTNR) für eine URL durch Anwenden exponentieller Glättungstechniken ab, um vorherige Anforderungen aufzuzeichnen, ebenso wie die momentane Transaktion: $MTNR_i = \alpha t_i + (1 - \alpha)MTNR_{i-1}$, wobei t_i die Zeit seit der letzten Referenz ist, $MTNR_{i-1}$ der vorherige Wert ist, und α der exponentielle Dämpfungskoeffizient ist (sie ergeben betriebsmäßige Werte zwischen 0,1 und 0,3 an). Die Gewichtungsmetrik zum Bestimmen der Priorität eines Dokuments in dem Cache-Speicher ist die wahrscheinliche Häufigkeit von Dokumentenanforderungen, die umgekehrt proportional zu MTNR:

$$w(t_i) = \frac{1}{mtnr_i}$$

ist. Hohe Werte von α heben die Neuheit von Dokumentenanforderungen hervor ($\alpha = 1$ reduziert auf priorisierende URLs durch das letzte Datum eines Zugriffs), während niedrige Werte von α vergangene Werte einer Neuheit weit mehr als momentane solche hervorheben.

[0039] In dem demokratischen Cache-Speicher kann man solche Gewichtungskriterien anwenden, allerdings modifiziert durch den abgeschätzten Wert der URL, um den Cache-Speicher zu Dokumenten eines hohen Werts hin zu gewichten. Zum Beispiel könnte man den Dämpfungskoeffizienten, α , von dem Wert abhängig machen:

$$\alpha' = \frac{\alpha}{f(v_i)}.$$

[0040] Hierbei ist v_i der Wert eines Dokuments i , wie er durch das Empfehlungssystem bereitgestellt wird, und $f(v_i)$ ist eine monoton steigende Funktion der Nutzbarkeit. Auf diese Art und Weise werden hoch bewertete Dokumente für relativ längere Zeitperioden als niedriger bewertete Dokumente cahemäßig gespeichert, wobei die Häufigkeit und die Neuheit ansonsten dieselben sind. Allerdings wird es auftreten, dass sich ein Dokument mit niedrigerem Wert mit häufigerem und neuerem Zugriff höher in der Priorität als ein höher bewertetes Dokument mit einem weniger häufigen und weniger neuen Zugriff bewegen wird. Es ist sehr wichtig, dass der Cache-Speicher dieses Verhalten anzeigt. Es ist daran zu erinnern, dass, zum Beispiel die Priorität eine einfache Funktion eines wahrgenommenen Werts war: Über der Zeit

würde der Cache-Speicher mit den am stärksten bewerteten Dokumenten in Abhängigkeit davon gefüllt werden, ob auf sie noch zugegriffen wird oder nicht.

[0041] Diese drei Gleichungen nun zusammen ermöglichen dem Prozessor **22**, die Priorität einer URL in einer Art und Weise zu bestimmen, die nicht nur die Neuheit und Häufigkeit eines Zugriffs berücksichtigt, sondern auch den abgeschätzten Wert der URL. Allerdings weist eine URL oftmals auch auf eine Seite hin, die das Herunterladen einer zusätzlichen URL an derselben Stelle mit sich bringt (z.B. URLs mit Einzelbildern und/oder gif Bildern). Wenn der abgeschätzte Wert durch das implizite Empfehlungsmodul bereitgestellt wurde, dann hat man dies getan, da dieses Modul denselben Empfehlungswerten für alle URLs von derselben Stelle bereitgestellt wird, und es wird möglich sein, die Prioritäten jeder der eingesetzten URLs separat zu berechnen. Es ist möglich, dass die URL sogar URLs an externen Stellen einsetzen kann. Um diesen Fall abzudecken, kann das Verfahren, das vorstehend beschrieben ist, so erweitert werden, um URLs mit impliziten, auf einer Stelle basierenden Empfehlungen ebenso abzudecken.

[0042] Allerdings wird, wenn der abgeschätzte Wert von dem expliziten Empfehlungsmodul kommt, die Situation komplexer. Während die explizite Empfehlung für die parent-URL existiert, existiert sie immer sicher nicht für die eingesetzten URLs, die auch cahemäßig gespeichert werden können. Demzufolge muss jede URL analysiert werden, um die eingesetzten URLs zu extrahieren. Alle eingesetzten URLs übernehmen dann den vorhergesagten Wert, der durch die explizite Empfehlung für die parent-URL zurückgeführt ist. Dieser Wert wird dann in den Prioritätsberechnungen für die eingesetzten URLs verwendet. Wenn der eingesetzten URL zuvor eine Prioritätsgewichtung innerhalb derselben Prioritäts-Umwertung-Periode zugeordnet worden ist (da, zum Beispiel, dieselbe URL durch mehr als eine parent-URL eingesetzt ist), erhält sie dann das Maximum der vorherigen Zuweisung der Momentanen. Dieser Vorgang wird anhand des Flussdiagramms der [Fig. 3](#) beschrieben.

[0043] Wie [Fig. 3](#) zeigt, ist, im Schritt **72**, die URL mit dem expliziten Empfehlungswert versehen. Im Schritt **74** werden URL-Inhalte analysiert, um die eingesetzten URLs zu extrahieren. Im Schritt **76** wird, für jede URL, die Prioritätsgewichtung W_c der eingesetzten URL, im Schritt **78**, berechnet. Wenn die vorherige Gewichtung W_p für die Umevaluierungsperiode im Schritt **80** existiert, wird eine Prüfung vorgenommen, ob sie größer (Schritt **84**) oder kleiner (Schritt **82**) ist. Es erfolgt eine Aktualisierung des Cache-Index mit dem Cache-Index im Cache-Speicher **24** mit der URL, einer Gewichtung und einem Zeitstempel.

[0044] In der Praxis wird das Verhalten des demo-

kratischen Cache-Speichers eng mit dem Empfehlungssystem verknüpft sein. Neuere Empfehlungen werden cachemäßig gespeichert werden, was demzufolge die Funktionsweise des Empfehlungssystems für seine Benutzer durch Verringern der Zugriffszeit auf die empfohlenen Elemente verbessert. Vergangene Empfehlungen werden dazu tendieren, von dem Cache-Speicher zu verschwinden, obwohl ein Wiederaufleben eines Interesses bewirken wird, dass die URL wieder für eine bestimmte Zeit in dem Cache-Speicher vorhanden ist.

[0045] Weiterhin kann das Empfehlungssystem den demokratischen Cache-Speicher mit zusätzlichen Informationen, wie beispielsweise dem Ablaufdatum der URL und/oder einer Wiederabrufrate (auch potentiell verfügbar als Meta-Daten innerhalb der HTML), bereitstellen. Für URLs, die als solche eines kurzzeitigen Interesses in dem Empfehlungssystem erklärt sind (Wiederabrufrate = 0), kann der Cache-Speicher dann die URL nach dem Ablaufdatum, zum Beispiel, löschen. Andererseits kann, für URLs mit Nicht-Null-Wiederabrufraten (täglich, wöchentlich, usw.), der Cache-Speicher ein Vorab-Abrufen der URL unter festgelegten Intervallen für URLs, die sich momentan in dem Cache-Speicher befinden, durchführen. Das Vorab-Abrufen kann während Perioden niedriger Belastung durchgeführt werden.

[0046] Es wird ersichtlich werden, dass die vorliegende Erfindung leicht in einer Software unter Verwendung von Software-Entwicklungs-Umgebungen ausgeführt werden kann, um einen portablen Quellen-Code bereitzustellen, der auf einer Vielzahl von Hardware-Plattformen verwendet werden kann. Alternativ kann das offenbare System teilweise oder vollständig in einer Hardware unter Verwendung von Standard-Logik-Schaltungen ausgeführt werden. Ob eine Software oder eine Hardware verwendet wird, um das System auszuführen, variiert in Abhängigkeit von den Geschwindigkeits- und Effektivitäts-Anforderungen des Systems und auch der bestimmten Funktion und der bestimmten Software oder den Hardwaresystemen und dem bestimmten Mikroprozessor oder den Mikrocomputersystemen, die verwendet werden.

Patentansprüche

1. Verfahren zum Bestimmen, welche Elemente in einem Cache-Speicher zu speichern sind, das aufweist:

Erhalten (62) von Wert-Daten für jedes Element, um in dem Cache-Speicher gespeichert zu werden, wobei die Wert-Daten ein Maß des Werts der Elemente aufweisen;

Priorisieren (68) der Elemente, die in dem Cache-Speicher gespeichert werden sollen, entsprechend zu Gewichtungsmetriken, bestimmt für jedes Element, wobei die Gewichtungsmetriken von den

Wert-Daten jedes Elements abgeleitet sind; und Speichern (68) solcher Elemente, die die höchste Gewichtungsmetrik haben, in dem Cache-Speicher, wobei die Elemente Web-Seiten aufweisen und wobei die Wert-Daten eine gewichtete Kombination von statistischen Informationen, die sich auf die Anzahl von eindeutigen Benutzern beziehen, die die Web-Seite, oder die Web-Stelle, die der Web-Seite zugeordnet ist, besuchen, und von Empfehlungs-Daten für ein Empfehlungssystem, das einen Wert für ein Element entsprechend zu Benutzer-Empfehlungen der Web-Seite oder der zugeordneten Web-Stelle bereitstellt, aufweisen, dadurch gekennzeichnet, dass der Priorisierungsschritt (68) ein Evaluieren der Gewichtungsmetrik $W(t_i)$ für ein spezifisches Element (i) aufweist, wobei

$$w(t_i) = \frac{1}{MTNR_i},$$

wobei $MTNR_i = \alpha' t_i + (1 - \alpha')MTNR_{i-1}$ und

$$\alpha' = \frac{\alpha}{f(v_i)},$$

$MTNR_i$

die Durchschnittszeit bis zu der nächsten Anforderung für Element i ist, t_i die Zeit seit der letzten Anforderung für Element i ist, $MTNR_{i-1}$ die vorherigen Wert-Daten sind, α' der exponentielle Dämpfungskoeffizient ist, v_i die Wert-Daten von Element i sind, wie dies durch das Empfehlungssystem bereitgestellt ist, α ein Dämpfungskoeffizient ist und $f(v_i)$ eine monoton ansteigende Funktion der Wert-Daten ist.

2. System für ein Cache-Speichern von Elementen auf einem verteilten Netzwerk, das aufweist: einen Computer (12) zum Speichern und zum Aufsuchen von Elementen von dem Netzwerk, wobei der Computer einen Prozessor (22) und einen Cache-Speicher (24) zum Speichern von Elementen, aufgesucht von dem Netzwerk; und ein Wert-Modul (14) zum Bereitstellen von Wert-Daten für Elemente, die in dem Cache-Speicher gespeichert werden sollen; umfasst;

wobei jedes Element, gespeichert in dem Cache-Speicher, Wert-Daten umfasst, die ein Maß des Werts der Elemente aufweisen;

wobei der Prozessor (22), auf eine Anforderung für ein bestimmtes Element hin, bestimmt, ob das Element in dem Cache-Speicher (24) gespeichert ist, und falls nicht, das Element von dem Netzwerk aufsucht;

wobei der Prozessor (22), für jedes aufgesuchte Element, auf eine Gewichtungsmetrik, abgeleitet von den Wert-Daten des aufgesuchten Elements, entsprechend, das aufgesuchte Element und die Elemente, gespeichert in dem Cache-Speicher, entsprechend der Gewichtungsmetrik jedes Elements priorisiert; und solche Elemente, die die höchste Gewichtungsmetrik haben, in dem Cache-Speicher speichert, wo-

bei

die Elemente Web-Seiten aufweisen und wobei die Wert-Daten eine gewichtete Kombination von statistischen Informationen, die sich auf die Anzahl von eindeutigen Benutzern, die die Web-Seite oder die Web-Stelle, zugeordnet zu der Web-Seite, besuchen, und von Empfehlungs-Daten von einem Empfehlungssystem aufweisen, das einen Wert für ein Element entsprechend zu Benutzer-Empfehlungen der Web-Seite oder der zugeordneten Web-Stelle bereitstellt, wobei der Prozessor (22) so angepasst ist, um die Evaluierung der Gewichtungsmetrik $w(t_i)$ für ein spezifisches Element (i) auszuführen, wobei

$$w(t_i) = \frac{1}{MTNR_i},$$

wobei $MTNR_i = \alpha' t_i + (1 - \alpha')MTNR_{i-1}$ und

$$\alpha' = \frac{\alpha}{f(v_i)},$$

$MTNR_i$

die Durchschnittszeit bis zu der nächsten Anforderung für Element i ist, t_i die Zeit seit der letzten Anforderung für das Element i ist, $MTNR_{i-1}$ der vorherige Wert für MTNR für Element i ist, α' der exponentielle Dämpfungskoeffizient ist, v_i die Wert-Daten von Element i sind, wie dies durch das Empfehlungssystem bereitgestellt ist, α ein Dämpfungskoeffizient ist und $f(v_i)$ eine monoton ansteigende Funktion der Wert-Daten ist.

3. System nach Anspruch 2, wobei das Wert-Modul (14) das Empfehlungssystem (16) zum Bereitstellen von Wert-Daten für ein Element entsprechend zu Benutzer-Empfehlungen oder zum Bereitstellen von Empfehlungs-Daten als eine Funktion der Qualität und der Quantität von Empfehlungen von Benutzern des Elements aufweist.

Es folgen 3 Blatt Zeichnungen

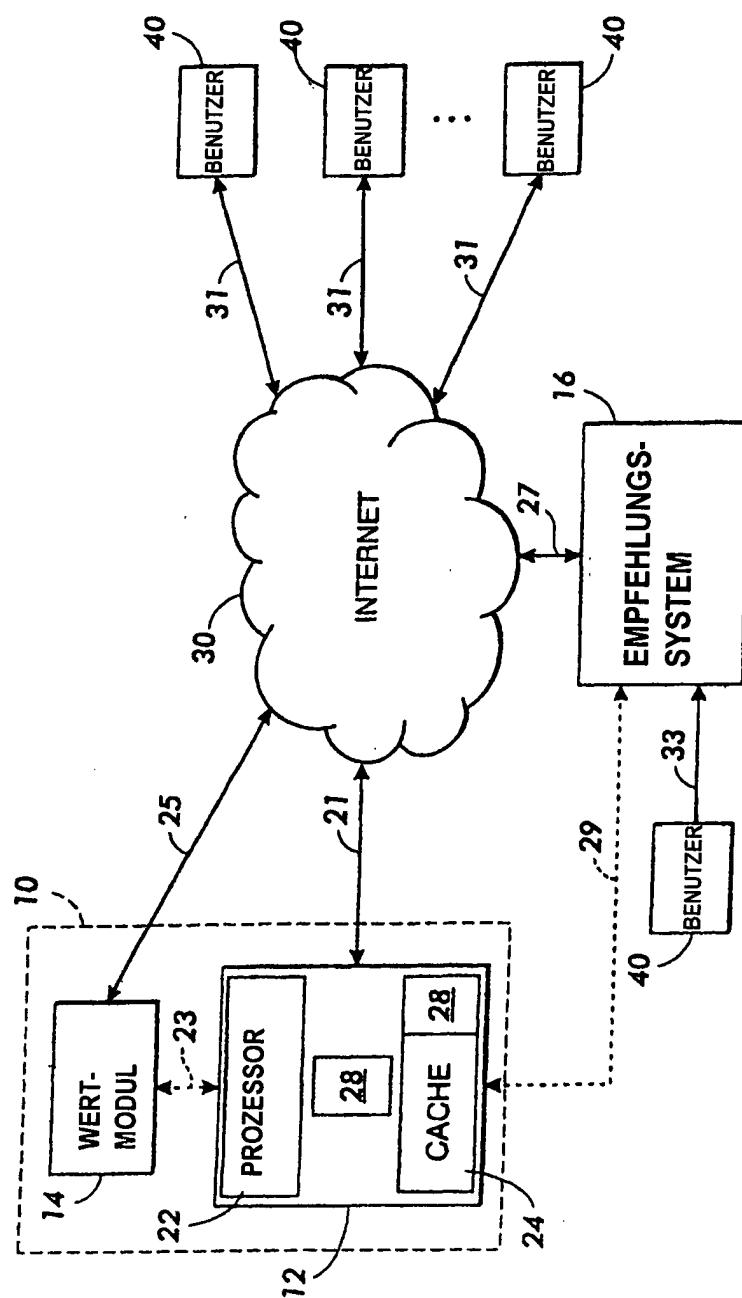
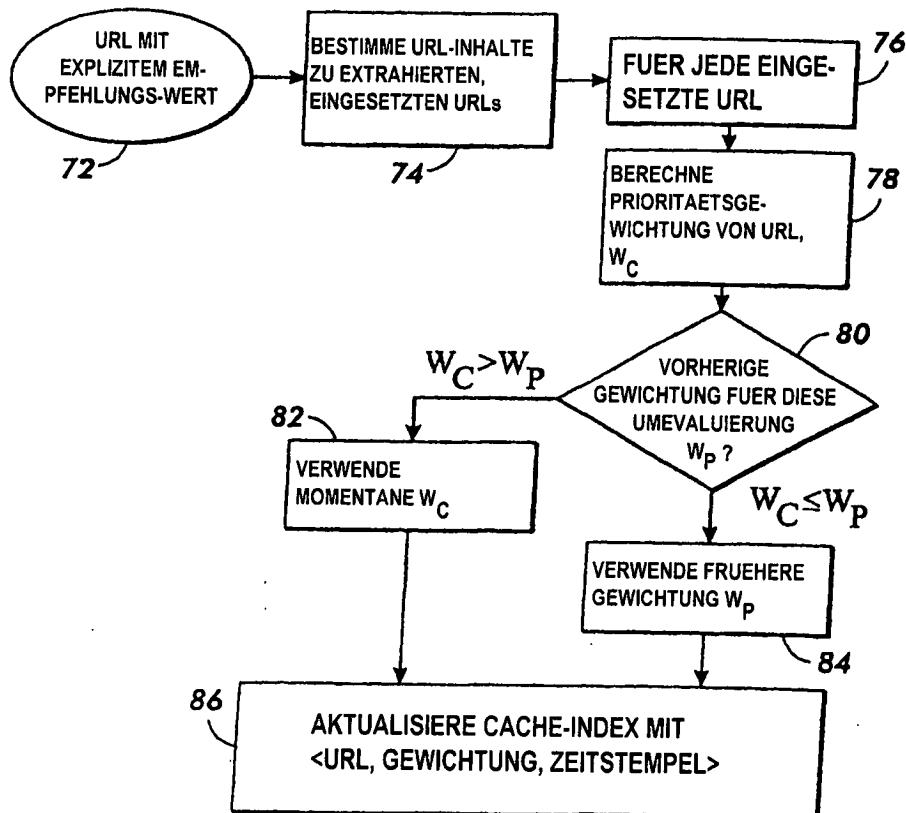
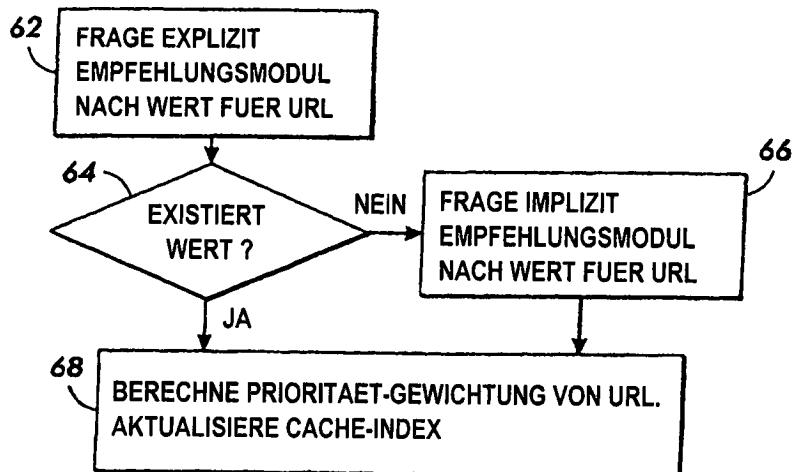


FIG. 1



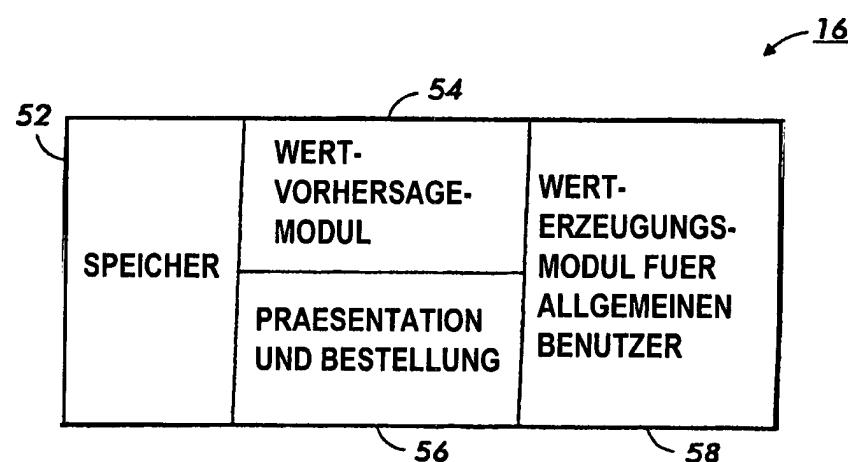


FIG. 4