

Zweitveröffentlichung



Eisenhardt, Martin; Henrich, Andreas

Probleme und Lösungsansätze für P2P-IR-Systeme

Datum der Zweitveröffentlichung: 01.12.2023

Verlagsversion (Version of Record), Konferenzveröffentlichung

Persistenter Identifikator: urn:nbn:de:bvb:473-irb-922679

Erstveröffentlichung

Eisenhardt, Martin; Henrich, Andreas (2002): „Probleme und Lösungsansätze für P2P-IR-Systeme“. In: Sigrid Schubert (Hrsg.), Informatik bewegt : Informatik 2002 ; 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) ; 30. Sept. - 3. Okt. 2002 in Dortmund, Band 1, Bonn: Gesellschaft für Informatik e.V. S. 171–175.

Rechtehinweis

Dieses Werk ist durch das Urheberrecht und/oder die Angabe einer Lizenz geschützt. Es steht Ihnen frei, dieses Werk auf jede Art und Weise zu nutzen, die durch die für Sie geltende Gesetzgebung zum Urheberrecht und/oder durch die Lizenz erlaubt ist. Für andere Verwendungszwecke müssen Sie die Erlaubnis des/der Rechteinhaber(s) einholen.

Für dieses Dokument gilt das deutsche Urheberrecht.

Probleme und Lösungsansätze für P2P-IR-Systeme

Martin Eisenhardt
Lehrstuhl für Angewandte Informatik I
Universität Bayreuth, 95440 Bayreuth

Andreas Henrich
Lehrstuhl für Angewandte Informatik I
Universität Bayreuth, 95440 Bayreuth

{martin.eisenhardt|andreas.henrich}@uni-bayreuth.de

Abstract: Zentralisierte Information Retrieval Systeme gelangen angesichts verteilter Datenbestände an ihre Grenzen. Bisherige Ansätze zum verteilten Information Retrieval lösen einige dieser Probleme, werfen jedoch neue auf. Der dezentrale Peer-to-Peer-Ansatz verspricht, auf grundlegend andere Art und Weise Lösungen aufzuzeigen.

1 Motivation

Die traditionelle Architektur eines Information Retrieval (IR) Systems besteht aus einem zentralen IR-Server, der sowohl die Dokumentensammlung verwaltet als auch für alle mit dem IR verbundenen Aufgaben wie etwa Indexierung von Dokumenten, Ähnlichkeitssuche, Ranking von Result Sets und ähnliches mehr zuständig ist. Eine solche Architektur ist aber aus diversen Gründen in manchen Szenarien nicht wünschenswert. Zunächst impliziert die Zentralisierung die Existenz eines *single point of failure*. Hinzu kommt, dass ein zentraler Server nur begrenzt skalierbar ist. Zudem wird die benötigte Hardware mit zunehmender Leistung überproportional teurer.

Ein weiterer Grund für einen dezentralen Ansatz im IR besteht darin, dass manche Organisationen aus verschiedensten Gründen keinen zentralen Server installieren und administrieren wollen — sei es aus Kostengründen, wegen des zeitlichen Aufwands oder weil die Organisationsstruktur und die Arbeitsweise inhärent dezentral ist.

Wir werden in diesem Beitrag zeigen, wie sich mit Hilfe des Peer-to-Peer-Paradigmas ein verteiltes IR-System entwickeln lässt, welches sich weitestgehend selbst administriert und das Publizieren von Dokumenten durch den Benutzer zulässt.

2 Das P2P-Paradigma

Entwickelt haben sich P2P-Netzwerke in den letzten Jahren vor allem in rechtlichen Grauzonen — die Musiktatschbörse Napster sei nur als Beispiel genannt. Neben solchen P2P-Netzwerken zum Filesharing — weitere Beispiele sind Gnutella (gnutella.wego.com) und KaZaa (www.kazaa.com) — existieren seit Ende der 1990er Jahre Ansätze zur verteilten Analyse von Massendaten bzw. zum Metaprocessing wie etwa SETI@home.

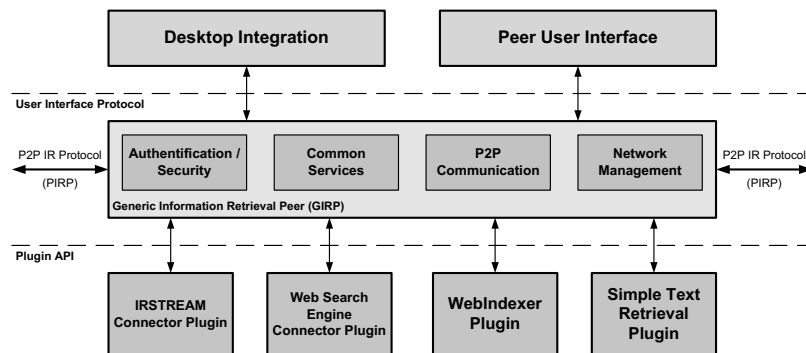


Abbildung 1: Überblick über die Architektur des P2P-IR-Systems.

Das Peer-to-Peer-Paradigma (P2P) lässt die traditionelle Einteilung von Rechnern bzw. Systemen in Klienten und Server hinter sich. Vielmehr funktionieren P2P-Netzwerke ohne zentrale Komponenten. Statt Klienten und Servern gibt es gleichberechtigte Peers, welche spontan ein vermaschtes Netz bilden. Der Grad N solcher Netzwerke bewegt sich typischerweise in einer Größenordnung um 8, wobei dies von Peer zu Peer schwanken kann und insbesondere von der Netzanbindung abhängt. Die Peers agieren zum einen als Klienten, indem sie von anderen Peers Daten und Informationen beziehen bzw. diese mit der Durchführung von Berechnungen beauftragen. Zum anderen übernimmt jeder Peer auch die typischen Aufgaben eines Servers, indem er als Anbieter von Informationen und/oder Diensten auftritt. Der P2P-Ansatz verleiht den darauf basierenden Programmen und Netzwerken große Flexibilität. In gewisser Weise sind P2P-Netzwerke selbst administrierend.

Nachteilig an P2P-Netzen ist die Tatsache, dass zu jedem Zeitpunkt einige Peers nicht online sind und somit Teile der Dokumentensammlung nicht erreicht werden können. Darüber hinaus neigen P2P-Netze dazu, mit der Zeit in Teilnetze zu zerfallen. Ein weiteres Problem liegt in der verteilten Natur von P2P-Netzen. Jedem Peer sind nur die lokalen Informationen zugänglich. Beispielsweise kennt jeder Peer nur die lokale Netzwerktopologie, nicht aber die Gesamttopologie. Zusammen mit der Tatsache, dass die Struktur eines P2P-Netzwerkes nicht vorgegeben ist, sondern spontan entsteht, kann dies zu ineffizienten Netzstrukturen führen.

Die oben beschriebenen P2P-Netze zum Filesharing beschränken ihre Retrievaldienste vornehmlich auf eine Suche nach Begriffen in den Namen der verfügbaren Dokumente. Wir gehen einen Schritt weiter und erweitern jeden Peer um Methoden des IR.

3 Überblick über die Architektur

Der hier vorgestellte P2P-IR-Peer ist modular aufgebaut. Er besteht aus Komponenten auf drei Ebenen (siehe Abbildung 1).

Die mittlere Schicht des Peers ist auch das Kernstück: der **Generic Information Retrieval**

Peer (GIRP). Der GIRP selbst besteht aus vier Subkomponenten:

Authentication and Security: Die Offenheit von P2P-Netzwerken und -Systemen verlangt nach User-Identifikation und weiteren Sicherheitsmaßnahmen. Die Komponente Authentication and Security bietet solche Verfahren an. Der Zugriff auf einzelne Dokumente oder Mengen von Dokumenten kann eingeschränkt und nur einzelnen Benutzern erlaubt werden. Hierzu können etwa *access control lists* (ACLs) verwendet werden. Nicht authentifizierte Benutzer haben keinen Zugriff auf diese Dokumente, diese werden (selbst wenn sie auf die Anfrage des Benutzers passen) auch nicht in dessen Ergebnislisten angezeigt. Solcherart zugriffsbeschränkte Dokumente sollten auch bei der Übertragung zwischen Peers gesichert sein. Unterstützung hierfür bietet die Komponente gleichfalls an.

Common Services: Diese Komponente bietet Dienste an, die von einigen anderen Komponenten genutzt werden können. An erster Stelle zu nennen ist das Dokumenten-Repository, in welchem die Dokumentensammlung des jeweiligen Peers verwaltet wird. Hierzu zählen auch Verwaltungsfunktionen zum Publizieren und Widerrufen von Dokumenten. Alle am Peer angeschlossenen Retrieval Engines haben Zugriff auf die Dokumente im Repository, um sie zu indexieren. Um die Verfügbarkeit und Leistung des Gesamtsystems zu steigern, bieten die Common Services auch Methoden zur Replikation von Dokumenten an. Hierfür gibt es vor allem zwei Gründe: zum einen sind nicht alle Peers immer online, so dass nicht alle Dokumente immer zugreifbar sind. Zum anderen kann es bei sehr „beliebten“ Dokumenten dazu kommen, dass der Peer, aus dessen Kollektion das jeweilige Dokument stammt, von Anfragen überlaufen wird und mit dem Ausliefern des Dokumentes nicht nachkommt. Eine weitere wichtige Aufgabe ist das Führen und Persistieren einer Liste aller bekannten anderen Peers. Die Informationen darüber, welche anderen Peers existieren, können aus erhaltenen Queries und Result Sets extrahiert werden. Der GIRP benutzt die Liste der *Known Hosts*, um einer Teilung des Netzwerks entgegenzuwirken.

Network Management: Die Topologie ist für die Leistungsfähigkeit eines P2P-Netzes ausschlaggebend. Sie bestimmt, wie Queries verteilt werden, welche Bandbreiten zur Verfügung stehen und welche Latenzzeiten innerhalb des Netzes auftreten. Zu beachten ist, dass hierbei nicht die physischen Verbindungen betrachtet werden, sondern die logischen Verbindungen zwischen den Peers.

P2P Communication: Die Peers kommunizieren miteinander über das *P2P Information Retrieval Protocol* (PIRP). Dieses enthält Konstrukte zur Übermittlung von Anfragen und Result Sets, zum Network Management und für weitere (administrative) Aufgaben. Neben dem Versenden und Empfangen von Queries und Result Sets ist das Query Routing und die Zusammenführung von Result Sets eine wichtige Aufgabe dieser Komponente. Ziel des Query Routings ist, Queries anhand bestimmter Merkmale (z.B. enthaltene Suchterme) in Richtung vielversprechender Peers zu routen. Die von den einzelnen Peers erhaltenen Result Sets müssen mittels geeigneter Methoden zusammengeführt werden. Dies kann entweder erst beim Absender der Query geschehen oder aber schon während des Routings der Result Sets entlang des durch die zugehörige Query erzeugten Spannbaums.

Über das *Plugin API* können mehrere **Retrieval Engines** in den Peer eingebunden werden. Dies können zum einen speziell für diesen Peer entwickelte Retrieval Engines sein, zum anderen können über Konnektoren bereits bestehende IR-Systeme eingebunden werden.

Simple Text Retrieval Engine: Diese Komponente durchsucht das lokale Repository nach indexierbaren Dokumenten (ASCII, HTML, PDF) und nimmt diese in den Index auf. Dabei wird ein an die P2P-Architektur angepasstes Vektorraummodell angewendet.

WebIndexer Plugin: Der WebIndexer ist ein einfacher Crawler, der lokale und entfernte WWW-Seiten indexieren kann.

Web Search Engine Connector Plugin: Mittels dieses Plugins kann der GIRP an bestehende Suchmaschinen im WWW angebunden werden. Anfragen des Benutzers werden in Anfragen an die jeweilige Suchmaschine übersetzt und dorthin gesendet.

IRSTREAM Connector Plugin: *IRSTREAM* ist ein an unserem Lehrstuhl entwickeltes Client/Server-System zum multimedialen Information Retrieval [HR01]. Über einen Konnektor kann dieses System an das P2P-Netzwerk angebunden werden.

Das **User Interface** (UI) des GIRP wird über ein UI Protocol angesteuert. Dadurch können verschiedene Interfaces implementiert werden. So ist es möglich, das UI auf einem anderen Rechner ablaufen zu lassen als den Peer. Ein Vorbild für die Anbindung des UI über ein Protocol ist neben dem X Protocol auch das Projekt *giFT* (gift.sourceforge.net), welches ein generisches UI Protocol für Peers des OpenFT-Netzes definiert.

4 Forschungsaspekte

Im Kontext des in Abschnitt 3 skizzierten Ansatzes ergeben sich zahlreiche interessante Forschungsaspekte, von denen im Folgenden einige wesentliche kurz angerissen werden:

Routing der Queries und Antworten: Besondere Bedeutung in P2P-Netzen hat das effektive und effiziente Routing der Queries und Responses im Netz. Einfache Broadcastansätze skalieren ab einer gewissen Netzwerkgröße sehr schlecht [ABJ01]. Daher verfolgen wir einen anderen Ansatz. Im Netzwerk werden zum einen Queries gecached, um bei erneuter Suche schneller Ergebnisse liefern zu können. Dieses Verfahren ist von Gnutella und Freenet bekannt. Des weiteren wird ein Ansatz verfolgt, wie er in [FCAB00] für das Caching im Internet vorgeschlagen wird. Dort werden *Bloom Filter* verwendet, um in komprimierter Form Auskunft über die auf einem Proxy Cache verfügbaren Dokumente zu geben. Im vorliegenden Fall können Bloom Filter benutzt werden, um in knapper Form die Dokumentensammlung eines Peers zusammenzufassen. Diese Information kann dann das effizientere Routing im P2P-Netz unterstützen.

Zusammenführung mehrerer Result Sets: Die einzelnen Result Sets aller antwortenden Peers müssen — entweder in den inneren Knoten des durch die Query erzeugten Spannbereichs oder beim Absender der Query — zusammengeführt werden. Im IR werden zur Kombination von Result Sets oft Algorithmen wie *Quick Combine* [BKG00] eingesetzt; dieser geht aber davon aus, dass die Result Sets aus unterschiedlichen Rankings auf der gleichen Kollektion stammen, weswegen er in unserem Falle nicht ohne weiteres angewendet werden kann. Stattdessen wird in [RAS01] ein Verfahren vorgeschlagen, das Result Sets aus verschiedenen Quellen zusammenführt und dabei die Relevanz der jeweiligen Kollektion berücksichtigt. Jeder Peer sendet zusammen mit dem von ihm erzeugten Result Set die geschätzte Relevanz seiner Kollektion für die Query.

Optimierung der Netzwerk-Topologie: Wie bei allen Netzwerken ist auch bei einem P2P-Netz die Topologie ein bestimmender Faktor für die Performance. Ungünstige Topologien, bei denen die logischen Verbindungen zwischen benachbarten Peers über langsame physikalische Leitungen führen, sollten in günstigere Topologien überführt werden.

Ein weiterer Ansatz zur Effizienzsteigerung besteht in der Optimierung der relativen Anordnung der einzelnen Peers innerhalb des Netzwerkes. So können Peers, welche sich aufgrund ihrer Dokumentensammlungen oder der von ihnen ausgesandten Queries ähneln, näher zueinander gruppiert werden, so dass Queries schneller zu einem Ergebnis führen. Mit der Zeit werden sich Cluster von Peers bilden, welche thematisch „gut zueinander passen“. Die meisten Queries können dann relativ lokal behandelt werden.

Sicherheit im P2P-Netz: Die offene Struktur von P2P-Netzen verlangt nach geeigneten Sicherheitsmaßnahmen. Während herkömmliche P2P-Netze die Anonymität der beteiligten Nutzer in den Vordergrund stellten, gewinnen bei unserem Ansatz zwei weitere Sicherheitsaspekte an Bedeutung: Schutz vertraulicher Daten bei der Übertragung gegen das Abhören durch Dritte und nutzerabhängige Zugriffsrechte auf Dokumente. Für die Sicherung der Übertragung sind verschiedenste Lösungen aus dem Bereich der Kryptographie bekannt. Als wesentlich schwieriger zu lösen erweist sich die verteilte Verwaltung der Zugriffsberechtigungen auf Dokumente im P2P-Netzwerk. Dabei gehen wir davon aus, dass eine große Zahl von Dokumenten im P2P-Netzwerk für alle zugänglich sein soll. Daneben muss es aber möglich sein, einzelne Dokumente nur bestimmten Benutzern oder Gruppen zugänglich zu machen. Für die Gruppenverwaltung erscheinen dedizierte Verwaltungsinstanzen sinnvoll. Denkbar ist auf dieser Basis, dass jeder Benutzer eines P2P-Systems für seine Dokumente Zugriffsrechte lokal verwaltet und Benutzer sich im Rahmen einer *Password Challenge* identifizieren müssen. Alternativ können Zertifikate verwendet werden, mit denen sich Benutzer identifizieren. Möglich wäre auch, die zu schützenden Dokumente mittels eines asymmetrischen Verfahrens zu verschlüsseln, so dass nicht autorisierte Benutzer die Dokumente nicht entschlüsseln können.

Literaturverzeichnis

- [ABJ01] Fred S. Annexstein, Kenneth A. Berman, and Mihajlo A. Jovanovic. Latency effects on reachability in large-scale peer-to-peer networks. In *Proc. of the 13th annual ACM Symposium on Parallel Algorithms and Architectures*, pages 84–92. ACM Press, 2001.
- [BKG00] Wolf-Tilo Balke, Werner Kießling, and Ulrich Güntzer. Optimizing Multi-Feature Queries for Image Databases. In *Proc. of the Intern. Conf. on Very Large Databases*, 2000.
- [FCAB00] L. Fan, P. Cao, J. Almeida, and A.Z. Broder. Summary cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293, 2000.
- [HR01] Andreas Henrich and Günter Robbert. An End User Retrieval Interface for Structured Multimedia Documents. In Sibel Adali and Satish K. Tripathi, editors, *Proc. 7th Workshop on Multimedia Information Systems*, pages 71–80, Capri, Italy, November 2001.
- [RAS01] Yves Rasolofo, Faïza Abbaci, and Jacques Savoy. Approaches to collection selection and results merging for distributed information retrieval. In *Proc. of the 10th Intl. Conf. on Information and Knowledge Management*, pages 191–198. ACM Press, 2001.