Peer-to-Peer-Architekturen

Proseminar Virtuelle Präsenz

SS 2005

Wilhelm Eisenschmid

we1@informatik.uni-ulm.de

Inhaltsverzeichnis

Das Peer-to-Peer (kurz P2P) Denkmuster ermöglicht einen unmittelbaren Ansatz für die Entdeckung und den Austausch von Ressourcen, oftmals ohne die Notwendigkeit einer zentralen Autorität oder eines Servers. Zur Zeit richtet sich der Schwerpunkt noch auf den Austausch von Informationen und Rechenleistung, bald wird sich der Focus aber auch auf andere Services richten. P2P Systeme sind vorgesehen für Anwendungen, die vor allem Kooperation und Kommunikation beanspruchen. Das Ziel ist, die verfügbare Rechenleistung, Bandbreite oder andere Dienste möglichst effizient unter den zahlreichen Peers einzusetzen.

Diese Arbeit wird zunächst auf die allgemeine Definition von Peer-to-Peer und auf den Unterschied zu der im Parallelvortrag behandelten Client-Server-Architektur eingehen. Desweiteren werden die grundlegenden Architektur-Modelle vorgestellt. Hier unterscheidet man Hybrid-, Super- und Pure-P2P-Muster. Im Folgenden werden noch die Abhängigkeiten Bandbreite und Adressierung behandelt. Auch einige Eigenschaften sowie Vor- und Nachteile werden aufgelistet. Schließlich werden die wichtigsten Anwendungen der P2P-Architekturen beschrieben.

A. Definition

In der Literatur wird eine große Vielfalt an Definitonen vorgestellt. Wikipedia beschreibt das P2P-Paradigma wie folgt:

> „Peer-to-Peer (engl. peer "Gleichgestellter", "Ebenbürtiger" oder "Altersgenosse/in") ist eine verbreitete Lesart für P2P und bezeichnet Kommunikation unter Gleichen. Andere Interpretationen von P2P lauten Person-to-Person (Betonung der rechnergestützten zwischenmenschlichen Kommunikation) und Program-to-Program (Kommunikation zwischen "intelligenten" Agenten).“ [1]

P2P ist ein netzwerkbasiertes Modell für Anwendungen, wo Computer Ressourcen und Dienste über den direkten Austausch teilen. Alle Computer können Dienste beanspruchen oder zur Verfügung stellen, worauf besonders acht darauf gelegt wird, dass keine Abhängigkeit von zentralen Servern besteht.

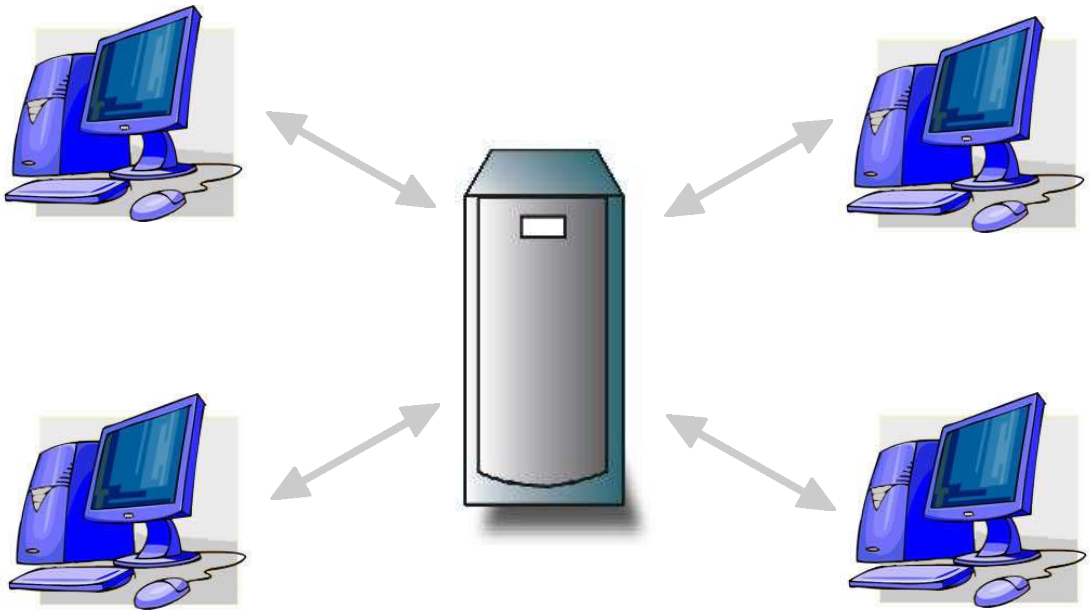
B. Gegensatz zu Client/Server

In einem Client-Server-Modell stellt der Client Anfragen an den Server, mit dem er verbunden ist. Der Server, typischerweise ein alleinstehendes und unbeaufsichtigtes System, beantwortet die Anfragen und handelt danach. Dagegen arbeitet im P2P-System jeder teilnehmende Computer (Peer) als Client, der auch Server-Funktionalität besitzt. Dies erlaubt dem Peer, sowohl als Client, als auch als Server zusammen mit der gegebenen Applikation zu handeln.
P2P-Anwendungen erfüllen verschiedene Aufgaben durch direkten Austausch, zum Beispiel das Bereitstellen von Speicherplatz oder Rechenleistung, Messaging, Gewährleistung von Sicherheit oder das Verwalten von größeren Datenmengen. Ein Peer kann Anfragen auslösen und auch Anfragen von anderen Peers im Netzwerk beantworten. Die Fähigkeit, sich direkt mit anderen Usern zu verbinden, grenzt sich von der bisherigen Abhängigkeit von einem zentralen Kontrollpunkt ab. Die Benutzer haben ein höheres Maß an Autonomie und Kontrolle über die Dienste, die sie beanspruchen. Auch der Begriff der Skalierbarkeit verwirklicht sich in einem P2P-System viel mehr als bei der Client-Server-Architektur. Auf die genauere Definition wird später noch näher eingegangen. Ein weiterer Nachteil des Client-Server-Musters ist, dass die Ressourcen (Speicher, Rechenleistung, Information) der Clients stark vernachlässigt, sogar überhaupt nicht miteinbezogen werden. Zudem besteht die Gefahr der Erscheinung eines Single-Point-of-Failure. Zum besseren Verständnis verhelfen die folgenden Definitionen:

> „Unter einem Single Point of Failure oder kurz SPoF versteht man diejenigen Komponenten eines Systems, die bei einem Ausfall den Komplettausfall eines Systems nach sich ziehen. Bei hochverfügbaren Systemen muss darauf geachtet werden, dass alle Systeme redundant ausgelegt sind.“ [2]

> „Der Begriff Redundanz (v. lat. redundare – im Überfluss vorhanden sein) bezeichnet allgemein in der Technik das zusätzliche Vorhandensein funktional gleicher oder vergleichbarer Ressourcen eines technischen Systems, wenn diese bei einem störungsfreien Betrieb im Normalfall nicht benötigt werden.“ [3]

Client-Server:



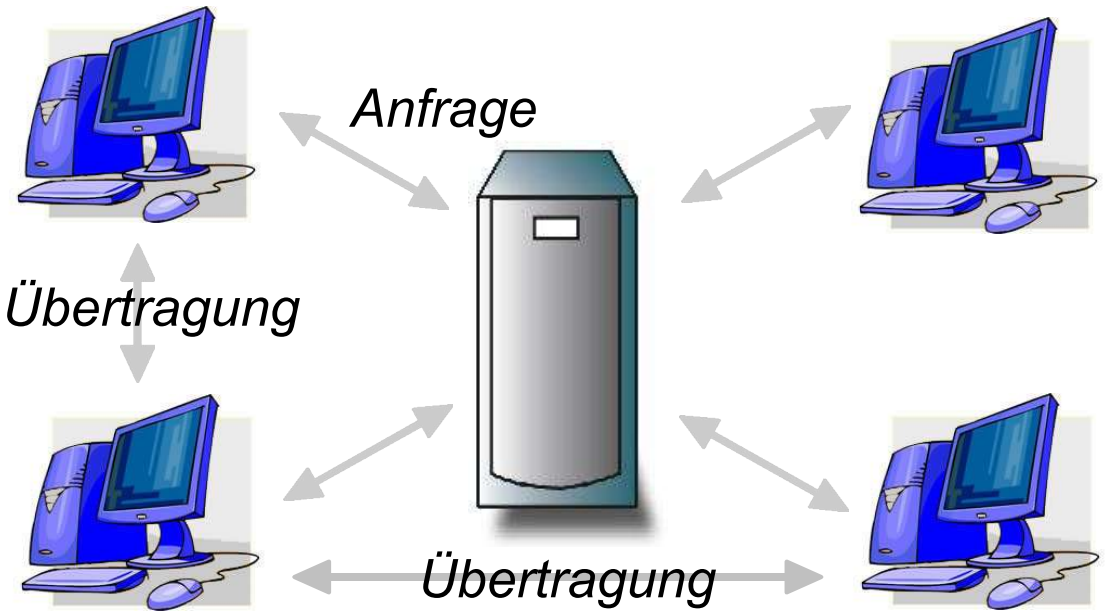
C. Architekturen

Grundsätzlich unterscheidet man zwischen Hybrid und Pure P2P-Architekturen. Hier sei aber auch noch das Super P2P-Modell erwähnt. Man differenziert ausserdem zwischen dezentralen und semi-zentralen Systemen.

I. Hybrid P2P

Das Hybrid P2P-Modell bedient sich auf der einen Seite der Client-Server-, auf der anderen der P2P-Beziehung. Es ist semi-zentral und beinhaltet mindestens einen zentralen Kontrollpunkt. Der Zweck reicht von der Kontrolle des gesamten Netzwerks bis zu einem einfachen Bezugspunkt für die verbundenen Peers. Oft existiert ein alleinstehender Peer, der für alle anderen Peers einen Index bzw. Katalog für die verfügbaren Daten darstellt. Dieser Indexserver bringt den Vorteil mit sich, dass kein spezieller Suchdienst erforderlich ist. Wenn sich ein Peer mit dem Netzwerk verbindet, ist es erforderlich, dass er dem Indexserver seinen derzeitigen Standort (IP-Adresse) mitteilt. Napster war dafür ein typisches Beispiel.
Die restlichen Peers sind entweder total frei oder stehen unter strikter Kontrolle wie zum Beispiel bei SETI@home. Es gibt auch Varianten mit mehreren Indexservern. Zum einen erhöht dies die Ausfallsicherheit, in dem ein einziger möglicher Single-Point-of-Failure ausgeschlossen wird. Auch die Performance wird erhöht, da die Last auf mehrere Server-Peers verteilt wird. Weitere Beispiele, die diese Architektur verwenden, sind WinMX und ICQ.

Hybride P2P:

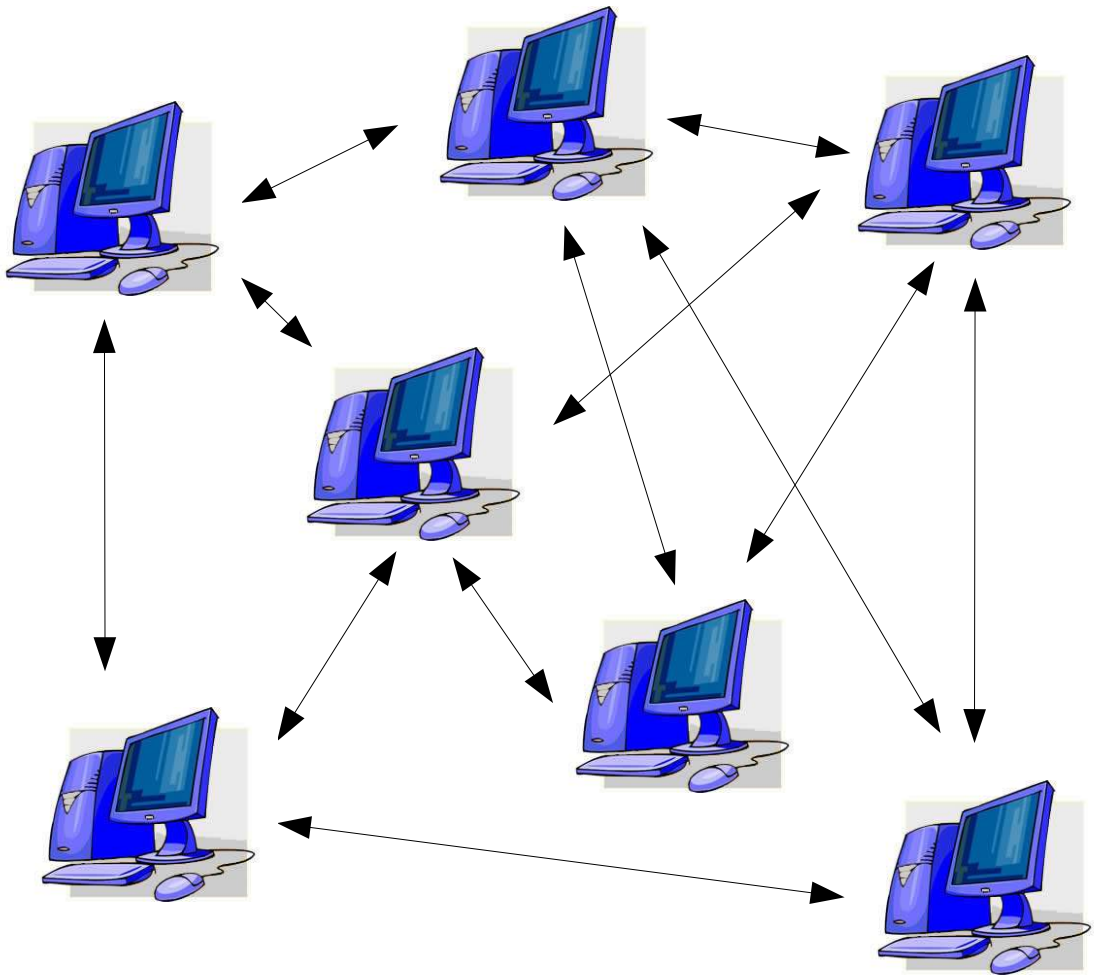


II. Super P2P

Das Super P2P-Modell ist eine Weiterentwicklung des Hybrid P2P, wo die zentrale Verwaltungsinstanz mit einem P2P-Netzwerk getauscht wird. Als Beispiel wäre hier KaZaa, das auf dem FastTrack-Protokoll aufbaut, zu nennen. Dabei werden gut angebundene Knotenpunkte im Netzwerk als Super-Nodes (Node, engl. = Knoten) eingesetzt, die als Übergangs-Indexserver dienen. Hierbei erfolgt eine „Peer-to-Peer-Interaktion zwischen „ausgezeichneten“ Teilnehmern (Super-Peers)“ [4] sowie zwischen normalen Peers. Super-Nodes und normale Peers interagieren in einen Client-Server-Beziehung.

III. Pure P2P

Das Pure P2P-Modell ist völlig dezentral ausgelegt. Es beinhaltet keinen zentralen Kontrollpunkt. Die Peers werden als völlig gleichwertig und autonom betrachtet. Daten oder Rechenleistung sollen sich über alle Peers verteilen, die direkt oder indirekt über andere Peers miteinander kommunizieren können. Infolgedessen sind sie sich über das Vorhandensein (Online-Status) anderer Knoten bewusst. Die Organisation der Peers kann entweder eine feste Struktur aufweisen oder gänzlich unstrukturiert sein. Ein unstrukturiertes Modell ersetzt die Notwendigkeit einer Netzwerkadministration, die bestimmte Konfigurationen ausführt. Beispiele für diese Architekturen sind Gnutella und Freenet.



D. Abhängigkeiten

I. Bandbreite

Wenn sich Peers direkt miteinander verbinden, um zum Beispiel größere Datenmengen zu übertragen, so sollte das auch in einem angemessenem Zeitrahmen geschehen. Beim P2P-Modell ist die Geschwindigkeit, mit der man Daten austauschen kann, abhängig von den Anschlüssen der einzelnen Peers. Das heisst, wenn ein Peer nur über eine 56 kBit/s-Leitung verfügt und sich sein Gegenüber über eine DSL-Leitung am Netz anmeldet, so wird die Geschwindigkeit die 56 kBit/s nicht übersteigen. Die Einführung von DSL- und Kabelnetzen erfordert eine Verwaltung der Bandbreite. Bis jetzt bieten die Internet-Service-Provider Bandbreite aber nur im ungleichen Verhältnis zwischen Up- und Downstream an. Meistens werden für den Upstream nur 128 kBit/s angeboten, der Downloadkanal ist aber deutlich größer ausgelegt. Mit der Benutzung der P2P-Technologie steigt jedoch das Verlangen des Users, genauso viel Daten anderen zur Verfügung zu stellen wie selbst zu beziehen. Diesem Wunsch des Users wird derzeit noch nicht entgegengekommen, das heisst, die Anbieter müssen sich mehr der P2P-Entwicklung anpassen. Ein Ansatz für die Zukunft wäre deshalb SDSL (Symmetrical Digital Subscriber Line), dass die Up- und Downstreamraten symmetrisch teilt. Diese Zugangstechnik ist aber noch sehr kostspielig und wird noch „fast ausschließlich für den Zugang zu ISDN und zu festverschalteten Weitverkehrs-Datennetzen verwendet.“ [5] Zur Erhöhung der Geschwindigkeit kann auch Caching beitragen. Dabei wird vermieden, dass identische Daten vielfach übertragen werden. Die Daten werden nicht nur auf den Rechnern gespeichert, die sie zur Verfügung stellen. Ein Algorithmus bestimmt, dass diejenigen Daten, auf die die Nachfrage sehr groß ist, auch auf Rechnern von anderen Peers zwischengespeichert werden.

II. Adressierung

Das Adressierungssystem bei P2P beruht nicht auf dem Domain Name System (DNS), da sich die einzelnen Peers ständig an- und wieder abmelden und daher dynamisch bestimmte IP-Adressen zugeteilt bekommen. In einem P2P-Netz ohne zentralen Server muss sich ein Rechner bei einem anderen anmelden, um in das Netz einzutreten. Doch dazu benötigt er seine IP-Adresse, die ihm beispielsweise eine Suchmaschine liefert, die das Internet nach Usern durchsucht, die die gleiche P2P-Software benutzen. Um diesen Anmeldevorgang zu optimieren, wäre es sinnvoll, wenn die User ständig online wären und daher feste IP-Adressen hätten. Das aktuelle System IPv4 zur Verwaltung von IP-Adressen verwendet 32-Bit-Adressen. Das neue Ipv6 benutzt aber einen 128-Bit-Adressraum – somit das vierfache von Ipv4 - und ermöglicht, dass man an jeden vernetzten Rechner eine statische IP-Adresse vergibt.
Zur Adressierung in lokalen Netzwerken kommt häufig APIPA (Automatic Private IP Adressing) zum Einsatz. Findet der Rechner beim Start keinen DHCP-Server, verwendet der Rechner APIPA, um sich selbst eine gültige IP-Adresse zu verschaffen. Dazu sucht er sich eine IP-Adresse aus dem Class B-Netz mit der Adresse 169.254.0.0 und der Subnetmask 255.255.0.0 aus und prüft über das ARP-Protokoll nach, ob diese Adresse bereits verwendet wird.

E. Eigenschaften

Die zu verwendende P2P-Architektur hängt immer von der jeweiligen Anwendung ab. Bei der Auswahl muss man abwägen, inwieweit die Anforderungen der Anwendung mit den Eigenschaften der Architektur übereinstimmen. Im Folgenden werden einige Eigenschaften beschrieben.

Ausfallsicherheit ist auf jeden Fall die wichtigste Eigenschaft einer jeden Software. Die Qualitätserwartung der User toleriert keine langanhaltenden Software-Schwachstellen. Unzuverlässige Anwendungen können den Anwender viele Kosten aufzwingen. Ein dezentrales P2P-System neigt zu Netzwerkfehlern, so dass bestimmte Maßnahmen ergriffen werden müssen, deren negativen Einfluss auf das System zu unterbinden.

Als Skalierbarkeit bezeichnet man die Fähigkeit eines Systems, trotz Variabilität in seiner gesamten Betriebsgröße, ohne deutlichen Leistungsabfall zu operieren. Würde ein System, das ursprünglich für 10 User entwickelt worden ist, auch befriedigend operieren für 1000 User? Skalierbarkeit in P2P-Systemen wird oft unterschiedlich betrachtet, zum einen beispielsweise die Anzahl der derzeitigen User, zum anderen die Anzahl der Knoten im Netzwerk. Die Skalierbarkeit spielt also beim Entwurf eine bedeutende Rolle.

Sicherheit in P2P-Architekturen zu gewährleisten ist schwieriger als in zentralen Client-Server-Systemen. Es existiert immer der Zwispalt, dass der User seine Kommunikationspartner identifizieren will und Informationen nur mit denen teilt, denen er vertraut, aber auf der anderen Seite anonym bleiben will. Ohne Sicherheits-Tools ist ein Vergleich mit der Verbreitung von Viren per E-Mail sinnvoll. Schadhafte Daten werden von jedem Knoten an den anderen weitergegeben und somit unkontrolliert verteilt.

Datenintegrität bedeutet, dass Daten aufgrund von Netzwerkproblemen oder gleichzeitigem Zugriff nicht beschädigt werden. Durch den ständigen Austausch zwischen den Peers ist es schwierig, Datenintegrität zu gewährleisten. Daten sind besonders anfällig, während dem Transfer beschädigt zu werden, und weil alle Peers autonom handeln können, existiert oft eine Vielzahl von nicht übereinstimmenden Versionen einer Datei.

Die Anonymität versteckt die Identität des Users, gewährleistet aber auch die Authentifizierung des Users, indem zum Beispiel Pseudonyme als Benutzernamen anstatt IP-Adressen verwendet werden.

Load Balancing (Lastverteilung) ermöglicht, dass bestimmte Peers im System nicht überarbeitet bzw. ungenutzt sind. Das Ziel ist es, einen effizienten Gebrauch der Ressourcen festzustellen. Durch die Verteilung der Anfragen auf verschiedene Peers erreicht man eine Erhöhung der Ausfallsicherheit, sofern der Ausfall eines Systems erkannt und die Anfragen dann automatisch an ein anderes Teilsystem weitergegeben werden.

F. Vor- und Nachteile

Kostenteilung/-reduktion:
Zentrale Systeme bedienen viele Clients, den Großteil der entstehenden Kosten trägt jedoch der zentrale Server. Dies kann jedoch verhindert werden. P2P-Architekturen unterstützen die Ansicht, die Kosten über alle angeschlossenen Peers zu verteilen. Beispielsweise ermöglichte Napster die Verteilung der Kosten für Speicherkapazität. Deswegen hat P2P Einfluss sowohl auf Business-to-Business- (B2B) als auch auf Business-to-Costumer- (B2C) Beziehungen. Zudem können Kosten gesenkt werden, indem man auf Ressourcen stößt, die vorher nie genutzt worden sind.

Anhäufung von Ressourcen und Interoperabilität:
Ein dezentraler Ansatz eignet sich natürlich für ein System, dass die Anhäufung von Ressourcen unterstützt. Jeder Knoten im P2P-Modell steuert zusätzliche Ressourcen wie Rechenleistung oder Speicherplatz bei. Das Zusammenfassen von Daten oder Informationen zu größeren Einheiten begünstigt vor allem die Anwendungen, die zum Beispiel rechen-intensive Simulationen oder große verteilte Dateisysteme beanspruchen. Diese Vorteile der Datenanhäufung setzen jedoch Interoperabilität zwischen verschiedenen Hard- und Software-Plattformen voraus.
„Interoperabilität ist die Fähigkeit unabhängiger, heterogener Systeme möglichst nahtlos zusammen zu arbeiten, um Informationen auf effiziente und verwertbare Art und Weise auszutauschen bzw. dem Benutzer zur Verfügung zu stellen, ohne dass dazu gesonderte Absprachen zwischen den Systemen notwendig sind.“ [6]

Autonomie:
Um Kontrolle und Sicherheit zu gewährleisten, verlangt der User, dass seine Daten und deren Bearbeitung auf lokaler Ebene verwaltet werden. Die P2P-Architektur unterstützt diesen Punkt auf alle Fälle.

Privatsphäre:
Der Benutzer will auch seine Privatsphäre schützen und anonym bezüglich bestimmten Service Providern bleiben. In einer zentralen Architektur ist es schwer, Anonymität zu gewährleisten, da der Server normalerweise seinen Client identifizieren kann. Die P2P-Architektur ermöglicht dem User, seine Informationen lokal zu verwalten. FreeNet ist eines der besten Beispiele, wo Anonymität einen sehr hohen Stellenwert einnimmt.

Dynamik:
P2P-System setzen voraus, dass das Netzwerk bzw. die Umgebung der verbundenen Peers sehr dynamisch ist, da Ressourcen das System ja kontinuierlich verlassen und wieder betreten. Diese Umgebung ist besonders mit verteilten Systemen vergleichbar, deren Größe oder die Veränderungen, die später notwendig sein könnten, noch nicht im voraus bekannt sind. Zum Beispiel existiert bei Instant-Messengern eine sogenannte „Buddy-Liste“, die den User informiert, wann es ihm möglich ist, seinen Chat-Partner zu kontaktieren. Auf ähnliche Weise müssen sich auch Applikationen wie zum Beispiel SETI@home, die sich das verteilte Rechnen zu Nutze machen, einem dynamischen Teilnehmerfeld anpassen.

Als Nachteil sei hier erwähnt, dass die Geschwindigkeit des Datenaustausches natürlich von der Anbindung und der Rechner der einzelnen User abhängt. Workstations sind einfach oftmals langsamer als große Server. Desweiteren ist die Sicherheitsproblematik zu nennen, auf die bei den Eigenschaften schon eingegangen ist. Auch das Auftreten von urheberrechtlich

geschützten Daten und somit einer illegalen Nutzung von P2P-Technologien, was durch Napster ins Rollen kam, wird sehr schwer zu unterbinden sein.

G. Anwendungen

Filesharing:
Über Napster wurde als erste populäre Anwendung viel diskutiert. Napster ist ein Beispiel einer zentralen hybriden P2P-Architektur, wo eine Gruppe von Servern das Nachschlagewerk der angeschlossenen Peers erfüllt. Der User muss zuerst einen Account beim Napster-Server beantragen und die Liste seiner Musikdateien zur Verfügung stellen. Danach kann er Anfragen an den Napster-Server stellen und eine Liste von Peers, die den gewünschten Musiktitel anbieten, empfangen. Anschließend wählt er einen Peer aus, um die Datei direkt herunterladen zu können.
Nach Napster kamen jedoch noch unzählige Filesharing-Anwendungen zum Vorschein: Gnutella, FreeNet, WinMX, eDonkey2000, usw.

Internettelefonie:
Skype ist der erste VoIP-Client, der eine P2P-Architektur verwendet. Hierbei unterscheidet man normale Knoten und Super-Nodes. Super-Nodes sind sogenannte Bezugs- oder Endpunkte für die normalen Nodes. Jeder normale Knoten mit ausreichend CPU-Leistung, Arbeitsspeicher und Bandbreite kann zu einem Super-Node werden. Ein normaler Knoten muss sich zuerst mit einem Super-Node verbinden und sich dann beim Skype Login-Server registrieren bzw. einloggen. Der Login-Server speichert User-Namen und Passwörter und stellt sicher, dass User-Namen im Skype-Netzwerk nur einmalig vergeben werden. Neben dem Login-Server gibt es im Skype-Netzwerk keinen zentralen Server. Der Online-Status sowie Suchanfragen werden in dezentraler Manier verwaltet.

Instant Messaging:
P2P ermöglicht, sogenannte Echtzeit-collaborative Anwendungen zu erstellen. Das heisst, aktuelle Informationen können von Projektteilnehmern weltweit direkt und in Echtzeit übertragen werden. Es werden also direkt Informationen ausgetauscht, ohne sie auf einem Server lokal zwischenspeichern zu müssen. Dadurch wird auch der Traffic im Netz allgemein reduziert, weil man E-Mail- und Serverspeicherplatz minimiert. Inzwischen gibt es eine Vielzahl von Instant Messengern: ICQ, IRC, Jabber, usw.

Distributed Computing:
P2P kann Anwendungen der Wirtschaft oder Wissenschaft unterstützen, die hohe Rechenleistung beanspruchen kann. Der hohe Bedarf an Rechenleistung kann auf mehrere Computer verteilt werden, indem ungenutzte CPU Mips und Speicherplatz im Netzwerk verwendet werden. Die kombinierte Leistungsfähigkeit von bisher unentdeckten Ressourcen kann die normal verfügbare Rechenleistung eines unverteilten Systems leicht überschreiten. Als Beispiel ist hier das Projekt SETI@home zu nennen, das nach ausserirdischem intelligentem Leben sucht. Jeder kann sich ein freies Programm herunterladen, dass die Messdaten eines Radioteleskops empfängt und speichert. Somit kommt für das Projekt eine Kostenersparnis hinzu, da die Rechenleistung der Clients sonst vernachlässigt würde. „Die gesamte Rechenleistung beträgt gegenwärtig bis zu knapp über 200 TeraFlops.“ [7]

H. Zusammenfassung und Ausblick

Das Peer-to-Peer-Modell sieht alle Teilnehmer als gleichberechtigt und autonom, die Informationen direkt miteinander austauschen. Man unterscheidet grundsätzlich die Arten Hybrid, Super und Pure P2P. Die erste populäre Anwendung, über die viel diskutiert wurde, war das Filesharing-Tool Napster. Als wichtigster Vorteil ist die Dezentralisierung von Administration und Kosten zu nennen. Dem gegenüber stehen noch einige technische Herausforderungen wie Bandbreite (Geschwindigkeit), Adressierung und Sicherheit, die eine ständige Verbesserung verlangen.

Durchaus bekannt sind auch schon P2P-Anwendungen, die ihre Verbindungen komplett verschlüsseln, so dass es dem Internet Service Provider nicht möglich ist, den Client zu identifizieren. P2P Services werden in der nahen Zukunft enorm wachsen. 2006 werden 35% aller Online-User P2P information-sharing Services benutzen. Sobald P2P-Anwendungen auf einen gemeinsamen Standard zurückgreifen werden und die Infrastruktur sie unterstützt, wird sich einiges verändern. Das freie Protokoll JXTA, das von SUN unterstützt wird, setzt sich hier immer mehr durch. Es ist vor allem unabhängig von der Programmiersprache, dem Betriebssystem und dem Transportprotokoll. Entwickler von web-basierten Anwendungen werden begreifen, dass die Fähigkeit, sich in eine P2P-Architektur zu integrieren, ihre Anwendungen lebendiger macht. Je mehr sich die P2P-Technologie entwickelt, so wird sie traditionelle Client-Server-Muster wie das Web beeinflussen. User werden gleichzeitig kommunizieren und interaktiv Aufgaben wahrnehmen. Die Palette reicht von der Team- und Projektarbeit im Unternehmen bis zur Übertragung von Streams, also Radio und TV. 2006 werden ca. 10% der Business-Interaktionen auf P2P-Plattformen ablaufen. P2P ermöglicht die Client-to-Client Interaktion über das Web Service Modell und wird in Zukunft immer mehr im Zusammenspiel mit Web Services eingesetzt werden. Unternehmen werden sich P2P mehr anvertrauen, um ihre Strategie und weltweite Präsenz im Computerzeitalter zu verwirklichen.

I. Quellen

[1] http://de.wikipedia.org/wiki/Peer-to-Peer

[2] http://de.wikipedia.org/wiki/Single_Point_of_Failure

[3] http://de.wikipedia.org/wiki/Redundanz

[4] http://www-vs.informatik.uni-ulm.de/teach/ss04/avid/2004s-AvID-I-P2P-2-D.pdf, Folie 11

[5] http://de.wikipedia.org/wiki/SDSL

[6] http://de.wikipedia.org/wiki/Interoperabilit%C3%A4t

[7] http://de.wikipedia.org/wiki/SETI%40home

[8] http://www.zdnet.de/itmanager/tech/0,39023442,2107183,00.htm

[9] http://www1.cs.columbia.edu/~library/TR-repository/reports/reports-2004/cucs-039-04.pdf