

TK1 - Theorie Aufgabe 1

Arne Link 1582381, Lars Fritche
Gruppe 14 (S217/103)

October 29, 2013

1. Aufgabe 1

1.1. Peer-to-Peer Systems

a). Benefits when using P2P

1. Potentiell jeder kann teilnehmen und Dienste anbieten oder in Anspruch nehmen. Im Regelfall wird nur ein gewisses Programm gebraucht um an dem Netzwerk teilnehmen zu können. Dies beeinflusst überwiegend neue Systeme / Benutzer.
2. Resistent gegen Angreifer / Einflüsse von außen. Fallen einzelne Knoten aus, können andere die Hauptaufgaben übernehmen, auch wenn einzelne Dienste dann ggf. nicht mehr angeboten werden. Im Falle von Filesharing Systemen würde das bedeuten, einzelne Dateien mögen nicht mehr so oft angeboten werden, das Netzwerk als solches wird aber nicht beeinflusst. Jeder in dem Netzwerk profitiert von diesem Vorteil.
3. P2P Netzwerke sitzen als eine weitere Ebene über anderen Netzwerken, so dass P2P Systeme im Prinzip auf einer Vielzahl verschiedener Netze, Protokolle etc. laufen können, ohne das der Benutzer des Systems etwas davon mitbekommen müsste. Diese Eigenschaft würde Benutzer betreffen, wenn unser Netzwerk / Internet heterogener aufgebaut wäre. In der heutigen Zeit wohl nicht mehr so interessant.

b). Drawbacks when using P2P

1. Ggf. verwundbar gegen Angreifer von innen. Haben sich einige Knoten erst einmal als wichtige Teilnehmer des Netzwerkes etabliert, senden dann allerdings falsche Informationen (im Gegensatz zu einfach ausfallen), kann das ganze Netzwerk in Mitleidenschaft gezogen werden. Je nach Protokoll und Dichte des Netzes kann diese Art von Angriff sehr schwer, oder fast trivial sein. Betrifft alle, die dieses Netzwerk verwenden.
2. Eine Sicherung des Systems / Netzwerkes ist nicht möglich. Bietet ein Peer eine einzigartige Dienstleistung an, verlässt dann aber das System, ist auch die Dienstleistung nicht mehr verfügbar. Hat nur Auswirkungen, wenn kein andere den Platz einnehmen kann, beeinflusst aber alle, die an dieser Dienstleistung interessiert sind.
3. Da jeder Peer potentiell auch für die Weiterleitung von Paketen, für die Discovery etc, verwendet wird, können schwächere Systeme, oder Systeme mit instabiler Verbindung (mobile Netze) leicht überfordert werden. Beeinträchtigt überwiegend die (Besitzer der) schwachen Systeme.

c). Challenges when using P2P

1. Sicherheit. Jeder der das P2P Protokoll implementiert kann an dem Netz teilnehmen, auch Systeme, die ggf. böse Absichten haben. Beispielsweise Fehlerhafte Dateien anbieten (File-sharing), Flooding des Netzwerkes oder einzelner Knoten, etc.
2. Sehr Dynamische Umgebung. Peers können jederzeit neu hinzukommen und wieder aussteigen. Bisherige Routing Pfade können sich auf einmal auflösen, in kleineren Netzen kann es zum Aufteilen des Netzes führen, so dass 2 oder mehr teile des Netzes zeitweise keine Verbindung mehr zueinander haben.

1.2. Client/Server Models

a). Examples

1. Das HTTP Protokoll. Ein Client (Browser, curl, etc.) fragt einen Server über eine festgelegte Adresse (<http://www.google.de>) nach gewissen Daten (Webseite, XML Dokumente, etc.)
2. Datenbankserver. Ein Datenbankserver über ein Interface Daten an Clients zur Verfügung. Die meisten relationalen DBMS verwenden dieses System.

b). Extended Client/Server Model

Im einfachen Client/Server Modell ist der Server ein Physikalisches gerät, im erweiterten Client/Server Modell kann der Server seinerseits gewisse Aufgaben an andere Systeme verteilen, oder fallback Server ausweisen, die im Falle eines Ausfalls die Rolle des Servers übernehmen.

Der Client hat keine Informationen darüber, welches System eingesetzt wird.

c). Benefits and drawbacks

Vor- und Nachteile gegenüber welchen anderen Systemen? Ähnliches Problem bei Aufgabe 1.1 a) und 1.1 b). Angenommen, es sind P2P Systeme gemeint:

Vorteile:

1. Server haben (erwartungsgemäß) eine feste, sich nicht ändernde Adresse. Ist die Adresse eines Servers einmal bekannt, kann davon ausgegangen werden, dass der Server / diese Dienstleistung auch zukünftig unter dieser Adresse zu erreichen ist. *Trifft auch auf das erweiterte Client/Server Modell zu.*
2. Einfache Struktur. Server muss sich nur um das Bearbeiten von Anfragen kümmern, und Client muss nur wissen, wie die Antworten zu interpretieren sind. *Trifft im Falle des Clients auch auf das erweiterte Client/Server Modell zu, im Falle des Servers nur bedingt.*

Nachteile:

1. Single point of failure. Wenn der Server ausfällt, ist der Dienst der unter der Adresse angeboten wird schlicht nicht mehr erreichbar. *Trifft auf das erweiterte Client/Server Modell nur zu, wenn die rekursive Version gewählt wird. Bei der iterativen Variante tritt das Problem auf, wenn der Server ausfällt der die Anfragen verteilt oder Weiterleitet, oder wenn es keine weitere Redundanz bei sekundären Servern gibt.*
2. Server kann Clients nicht benachrichtigen bei Änderungen. Clients haben (beim reinen Client/Server Modell) keine Möglichkeit, vom Server über Änderungen informiert zu werden. *Trifft auch auf das erweiterte Client/Server Modell zu.*

d). Recursive vs. Iterative

Im rekursiven Modell bearbeitet der Server die Anfrage entweder selbst, oder leitet sie (teilweise) an genau einen weiteren Server weiter. Ist die Anfrage bearbeitet, oder antwortet der Server, an dem die Anfrage weitergeleitet wurde, wird das Ergebnis an das System weitergeleitet, dass die Bearbeitung für diesen Server in Auftrag gegeben hat.

Im iterativen Modell hingegen bearbeitet ein Server die Anfrage, oder leitet (teile) an einen **oder mehreren** anderen Server weiter. Liegen alle notwendigen Ergebnisse bereit, antwortet der Server dem Client mit dem (ggf. aggregiertem) Ergebnis. Die Server, an denen Aufgaben weitergeleitet werden bearbeiten die Anfragen im reinen iterativen Modell direkt und kommunizieren ausschließlich mit dem primären Server.

Vorteile des iterativen Modells:

1. Anfragen können parallel an mehrere sekundäre Server gesendet werden, was die Bearbeitungszeit pro Anfrage verringert.
2. Der Server kann die Anfrage selektiv an gewisse Systeme weiterleiten, so dass Systeme nur beansprucht werden, die auch wirklich gebraucht werden.

Vorteile des rekursiven Modells:

1. Jeder Server muss nur über einen weiteren Server wissen, an denen alle Anfragen weitergeleitet werden, wenn sie nicht selbst vollständig bearbeitet werden können.
2. ???