BitTorrent

Ein Peer-to-Peer Datenübertragungsprotokoll

Mattias Schäffersmann

Universität Bielefeld - Technische Fakultät

2005-11-15

Übersicht

- das Problem der Datenverteilung
- drei P2P-Systeme kurz Vorgestellt
- BitTorrent im Detail
- Probleme
- Verwendung bei Organisationen
- Ausblick

Zentrale Systeme mit Servern

- zentrale(r) Server bedienen/bedient alle Clients
- skaliert schlecht
- Flash Crowds zwingen Server in die Knie
- hohe Kosten für Betreiber

Dezentrale Systeme (Peer-to-Peer)

- komplett P2P ausgelegte Netze ohne zentralem Server
- Hybridnetze mit zentralem Server
- skaliert je nach Architektur
- weitere Peers bringen neue Uploadkapazitäten mit
- keine oder geringe Betriebskosten
- instabil?

Warez und anderes Illegales

- grosse Datenmengen
  - modernes Konsolenspiel: DVD-Image 4 GiB
- fast keine finanziellen Mittel
- Lösung: Peer-to-Peer Netzwerke
  - kostenlos
  - Warez leicht zu finden
- grosse Popularität von P2P durch Warez
- Folge: P2P verrufen als illegal

Gnutella (LimeWire)

- wahres Peer-to-Peer, alle Peers gleichberechtigt
- kein angreifbarer zentraler Punkt
- Suchanfragen werden von Peer zu Peer weitergereicht
  - viel Overhead
  - gesamtes Netz praktisch nicht erreichbar
- lange Wartezeiten und geringe Geschwindigkeit
- eine Datei kann von mehreren Quellen bezogen werden
- Upload startet nach Downloadabschluss
- Peers können herrunterladen ohne hochzuladen (schummeln)

FastTrack (KaZaa)

- P2P-Protokoll der 2. Generation
- Hybrid-Netz mit zentralem Server
- “Supernodes” für bessere Performance
- Dateien und andere Peers gut zu finden
- lange Wartezeiten, akzeptable Geschwindigkeit
- eine Datei kann von mehreren Quellen bezogen werden
- Upload startet nach Downloadabschluss
- “Participation Level” gegen Schummeln
  - leicht fälschbar und somit nutzlos

eDonkey/Overnet

- eDonkey: Hybrid-Netz mit zentralen Servern
- Overnet: komplett P2P ausgelegtes Netz
  - Implementation des Kademlia Algorithmus
  - Durchsuchen des Netzes in log(n) möglich
- Dateien und andere Peers gut zu finden
- lange Wartezeiten, akzeptable Geschwindigkeit
- eine Datei kann von mehreren Quellen bezogen werden
- einzelne Dateiblöcke werden nach Fertigstellung hochgeladen
- angebliche Belohnung schneller Uploader
  - Horde

Übersicht über BitTorrent

- Hybridnetz mit zentralem Tracker
  - koordiniert Peers
- ein Schwarm von Peers per Torrent
- keine Suchfunktion
- alle Peers leicht zu finden
- keine Wartezeit, hohe Geschwindigkeit

Tauschen von Daten

- Einteilung in Blöcke von ~256KiB/16KiB
  - Herrunterladen von vielen Peers
  - neue Blöcke werden sofort selbst angeboten
- tit-for-tat
- Tauschen mit schnellsten Peers
- Peers maximieren eigene Downloadgeschwindigkeit selbst
- Schummeln nicht möglich
- skaliert sehr gut
  - Flash Crowds kein Problem

Ablauf eines BitTorrent Downloads

- Herrunterladen einer .torrent Metadatei
- automatisches oder manuelles Starten des Clients
- mit anderen Peers verbinden
- Blöcke tauschen bis Datei komplett
- Seeden nach eigenem Ermessen

BitTorrent Terminologie

- Torrent
  - .torrent Metadatei
  - Packet aus allen zusammengehörenden Dateien
- Peer
  - jeder Client
  - downloadende Clients
- Seed
- Leecher
  - unsoziale Peers
  - downloadende Clients
- Schwarm
- Tracker

.torrent Metadateien

- Adresse des Trackers
- Namen aller Dateien
- Blockgröße
- SHA1 Hashes für alle Blöcke
  - man bekommt was man will
  - Dateien nicht korrupt
- Hash der Metadatei identifiziert Torrent

Announces und Scrapes

- Announces melden Start/Ende des Downloads
- Anfrage nach mehr Peers auch zu anderen Zeiten
- Tracker liefert zufällige Menge von Peeradressen
- Peers melden Download-/Uploadmenge für Statistiken
  - leider oft gefälscht
- Scrapes liefern Anzahl Seeds + Peers

Handshake

- Peer-ID
  - enthält freiwillige Angabe der Clientversion
- Torrent-Hash zur Identifikation
- Bitfield
- bidirektionale Verbindungen
- Peers hinter Router/Firewall nicht abgeschnitten

Verbindungsflags & Geschwindigkeitsmaximierung

- Interested
- Choking
- Unchoken der vier schnellsten Peers
- Optimistic Unchoke
- Snubbing

Piece Picking

- *rarest first policy*
- *random first policy* für neue Peers ohne Blöcke
- angefangene Blöcke werden zuerst fertiggestellt
- *have piece* messages

DHT

- Distributed Hash Table
- ersetzt Tracker vollständig
- wenig Traffic
- erster Bootstrap von anderen Peers aus regulärem Schwarm
- dannach gespeicherte Adressen
- Magnet-URIs
  - enthalten keine Peer-Adressen oder IPs
  - Beispiel:
    magnet:?xt=urn:btih:MFFT6N555N6VTPSHVMNLF3AKZOSLNSF2

Kleine Demonstration

Kleine live-Demonstration um wieder wach zu werden.

Beispiel Scarywater Tracker

Tracker

- ∼55.000 Clients
- ∼13.000 Verbindungen/min
- ∼1.2 Mbit/s in, ∼1.8 Mbit/s out
- ∼30 GiB Traffic/Tag
- Pentium 4, 2.4 GHz, 512 MiB RAM
- Auslastung 10∼20% mit Hypercube-Tracker

effektive Downloadgeschwindigkeit der Peers

- ∼3.2 Gb/s ≈ 400 MiB/s
- ∼34 TiB pro Tag

- Multiplikator 1160x

Overhead

- Traffic durch Flagänderungen und *have piece* messages
- $\sim \frac{1}{20}$ von Nutzdatendurchsatz
- TCP-Overhead nicht mitgerechnet

- häufige Announces bei wenigen Peers
- GZip Kompression

mehr Probleme

- kein Index
- Firewalls/Router
- Port-Filter oder Limits von ISPs

- alte Torrents schwer zu bekommen
- "Verhungern" von Torrents ohne Seeds

Sicherheit der Peers

- IPs aller Peers beim Tracker bekannt
- IPs leicht durch Announce herrausfindbar
- kein Routen von Fremddaten
- “Verstecken” praktisch unmöglich

- Onion Router (Tor) Netzwerk
- I2P - Anonymes Peer-to-Peer Netzwerk

Verwendung im Open Source Bereich

- Open Solaris von Sun
- Red Hat 9
- openSUSE

Kommerzielle Firmen - Beispiel Blizzard

- Verbreitung des über 1 GiB großen WoW Betaclients
- Verbreitung von WoW Updates an $\sim$ 2 Mio. Spieler
  - $\sim$40 MiB/Update
  - schätzungsweise 50% der Spieler werden in 8 Stunden versorgt
    - $\sim$10 Gbit/s
  - BitTorrent Client für Updates im Spiel integriert

Web Seeding

- mehrere leistungsstarke Webserver als Seeds
- schnelle Downloads für alle Nutzer
- einfaches Load-Balancing
- Nutzer stellen eigene Bandbreite zur Verfügung
- Flash Crowds kein Problem

Avalance von Microsoft

- algorithmisches Berechnen der Dateien
  - nicht alle Blöcke benötigt
- ∼4 Peers als Nachbarn
- Upload nie mehr als das doppelte des Downloads

- hoher Resourcenaufwand für jeden Block
  - alle Daten im RAM?
- zu wenig Nachbarn
- “Bestrafungsalgorithmus” unpraktikabel

Fazit

- BitTorrent derzeit beste Möglichkeit große Mengen legaler Daten schnell zu verbreiten
- geschützt gegen Schummler
- skaliert sehr gut
- günstige Alternative

Präsentationsende

Fragen und offene Diskussion