

Willkommen bei Verteilte Systeme!

Von Datenbanken
über Webdienste
bis zu p2p und Sensornetzen.



Heute: **Peer-to-peer in der Praxis – wo und wie sich Verteilung lohnt(-e).**

Wer nicht aus der Vergangenheit lernt, ist verdammt ihre Fehler wiederholen, mit weniger Zeit, denn „die Probleme sind ja schon gelöst“.

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Warum?

Darum ich

- Seit 2004 in p2p-Entwicklung
- Seit 2013 mit Kompetenz :-)
- Aktuell Release-Manager des Freenet/Hyphanet Projektes

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Warum?

Vorträge

- 5 Minuten pro Person, Gruppen möglich.
- Weitere Ideen: Übersicht über einen [FOSEM 2023-Vortrag](#).

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Wiederholung

Fallacies of distributed Systems

- The network is reliable
 - The network is secure
 - The network is homogeneous
 - Topology does not change
 - Latency is zero
 - Bandwidth is infinite
 - Transport cost is zero
 - There is one administrator
- Hard disks don't fail
Files stay intact
 - Power is stable
 - IPs are reachable
 - Constant factors are negligible
 - APIs stay compatible
 - Textfiles are simple

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Ablauf

Ziele heute

- Sie kennen die zentralen Herausforderungen der Praxis:
 - Einstieg
 - Suche
 - Inhalte verbreiten
 - Kommunikation
- Sie können in Bezug auf zwei Herausforderungen die Eigenschaften von je zwei p2p-Systemen beschreiben, die sich in der echten Welt bewährt haben.
- Sie können einschätzen, ob ein bestimmtes Konzept diese Herausforderungen bestehen könnte.

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Ablauf

Grundprobleme in Peer-to-Peer-Netzen

- Einstieg:** Wie finde ich meinen Platz im Netz?
- Suche:** Wo gibt es, was ich brauche?
- Störungsresistenz:** Wie skaliert Gewünschtes besser als Unerwünschtes?
- Verbreitung:** Wie vermeide ich Flaschenhälse?
- Kommunikation:** Wie fließen Informationen durchs Netz?

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Darum praktisch erprobte p2p-Netze verstehen

after a few days (and especially nights) of nervous full-site tinkering, it turned a 40 minute deploy process into one that lasted just 12 seconds!¹

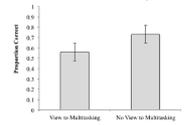
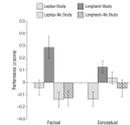
- BitTorrent-Deployment: <https://vimeo.com/11280885>
- Spoiler: Cut-through routing.

¹ <https://web.archive.org/web/20120807165933/http://engineering.twitter.com/2010/07/rauder-fast-datacenter-code-deploys.html>

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Warum?

Laptops: Eigenverantwortlich



“even when laptops are used solely to take notes ... their use results in shallower processing.”
— Mueller and Oppenheimer (2014)

“Laptop multitasking hinders classroom learning for both users and nearby peers”
— Sana et al. (2013)

- Laptops
- Möglichst zu

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Wiederholung

Wiederholung 1

- Sammlung autonomer Knoten, die als ein kohärentes System erscheinen.
- Ziele: Ressourcen, Verteilungstransparenz, Skalierbarkeit
- Skalierung: Größe, Geographie, Administration
- Latenz, Partitionierung, Replikation, Caching
- Fallacies!
- Cluster, Grid, Cloud, Ubiquitous, Mobile, DIS, Sensornetze

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Ablauf

Ablauf heute

- Grundprobleme
- Gnutella (das Erste verbreitete, komplett verteilte p2p-Netz)
- PAUSE 14:30
- Kademlia (das am weitesten verbreitete DHT)
- BitTorrent
- Freenet/Hyphanet
- Weiteres (Aktuelles, WebRTC, ...)

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Ablauf

Welche p2p-Netze kennen Sie

am FlipChart sammeln

-
-
-
-

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Ablauf

Warum p2p?

- Skalierbarkeit:** Ein einzelner Server bricht bei etwa 100k Anfragen pro Sekunde ein. *dwd* bei *Sturm Sabine 2020*?
- Mit Nutzung wachsen:** Ähnliche Infrastruktur für 1000 Leute oder 10 Millionen Leute
- Infrastrukturkosten:** 100k€ pro Jahr = Entwickler oder Entwicklerin

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Mein Ziel

Ich will, dass Sie die Fähigkeiten erwerben, unter denen zu sein, die die Deployment Zeit um Größenordnungen verringern, ohne dabei die Kosten dafür zu zahlen, Torrents als Blackbox zu sehen.

Torrent Bezeichnung für eine BitTorrent-Datei oder eine von BitTorrent verwaltete Datei.

BitTorrent Ein p2p-System zum Verteilen großer Datenmengen, bei dem die Verwaltung auf zentralisierten Trackern läuft

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Warum?

Projekte

- Notieren Sie bitte Ideen
- Modulplan: 39h Selbststudium
- Projekt interessanter und nützlicher als Klausur
- Projektideen sammeln.

Doing X with *libp2p* oder *libresilient*?

Auf einem der aktuellen *NLnet-Projekte* aufbauen?

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Wiederholung

Wiederholung 2

- Architektur:** Layered, Object, Resource, Event
- Schichten und Overlay Netze
- Prozesse sind isoliert, Threads teilen Speicher.
- Middleware** als Schicht: Übernimmt Verteilung, gibt Garantien.
- Messaging:** Request-Reply, Pub-Sub, Pipeline.
- Overlay metriken:** Link Stress und Stretch

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Ablauf

Was und warum?

Was?
peer-to-peer (p2p) peers (gleichberechtigte Partner) arbeiten zusammen, um sich gegenseitig einen Dienst zu erbringen.

Warum?
Sie haben ein unerwartet beliebtes Programm geschrieben. Jetzt wollen es 100 Millionen Leute herunterladen. Größe: 50GiB. Wie viel kostet die Verteilung?

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Ablauf

Ziele für Grundprobleme

Sie können die Grundprobleme beschreiben, die Peer-to-Peer-Netze lösen müssen:

- Fundamente**
- Einstieg
 - Suche
 - Verbreitung

- Aufbauend**
- Kommunikation
 - Störungsresistenz

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg Grundprobleme Gnutella Kademia BitTorrent Downloads Freenet Abschluss
 Ablauf

Warum nicht p2p?

- Gestiegende Leistung von Servern. *Sturm: dwd² hielt größtenteils Stand (durch vereinfachte Seite³)*
- Handies sind durch Batterie und Netz begrenzt → keine kontinuierliche Leistung. (Nachts möglich?)
- Viele der einfachen Lösungen unmöglich, z.B. Geld auf Probleme werfen.

²dwd: Deutscher Wetterdienst.
³→ gibt es eine einfachere Lösung?

Arne Babenhausenheide
Verteilte Systeme 1: peer-to-peer

Einstieg: Wie finde ich meinen Platz im Netz?

- **Erste Adressen:** Wie finde ich Adressen anderer Knoten?
- **Wahl der Verbindungen:** Mit wem sollte ich mich verbinden?
- **Routing-Informationen:** Welche Daten brauchen die Knoten?



Strukturiert vs. Unstrukturiert

Strukturiert

- **Erste Adressen:** Braucht Topologie⁴
- **Wahl der Verbindungen:** Nur bestimmte sinnvoll
- **Routing-Informationen:** Durch Auswahl der Partner (Peers)

Unstrukturiert

- **Erste Adressen:** Einfache Liste
- **Wahl der Verbindungen:** Beliebige Andere
- **Routing-Informationen:** Explizit austauschen

Kann ich alle direkt erreichen?

⁴Topologie: Struktur des Netzes.



Suche: Wonach suchen?

- **Schlüsselwort:** Gnutella, Skype (vor MS)
- **Inhalts-Hash:** Kademia, BitTorrent VHT, Freenet
- **Öffentlicher Schlüssel:** Freenet

Glossar:



BitTorrent VHT Verteilte Hashtabelle, ein DHT
DHT Distributed Hashtable

Öffentlicher Schlüssel public key, das Gegenstück zum privaten Schlüssel in asymmetrischer Verschlüsselung.



Suche: Wo gibt es, was ich brauche?

Zwei Konzepte:

- **Pfade zu existierenden Daten finden:** Gnutella
- **Daten an die richtigen Orte bringen:** Kademia, BitTorrent VHT, Freenet



Schlüssel zum Licht



Verbreitung: Wie vermeide ich Flaschenhalse?

- **Zentralisiert:** Streaming im Provider-Netz via Multicast
- **Swarming:** Nutzer übernehmen einen Teil der Verteilung
 - Koordiniert von zentraler Stelle: BitTorrent (Tracker)
 - Koordiniert durch die Nutzer: Gnutella (Download-Mesh)
 - Unabhängig verteilte Fragmente: Freenet⁵

Glossar:

Download-Mesh Name des Protokolls

Tracker Ein Server der den BitTorrent-Schwarm Koordiniert

⁵Reduziert Swarming auf Download einzelner Dateien, braucht aber caching: Zeitlich begrenzte Zwischenspeicherung.



Kommunikation

- **Vier-Augen Gespräch (PM/DM/msg/Anruf...)**
- **Unterhaltung in Gruppen (Chat, Forum, Videokonferenz, ...)**
- **Öffentliche Unterhaltung**
- **Von neuen Inhalten erfahren**
- **Informationen über Inhalte (Kommentare, Bewertung, ...)**



Störungsresistenz: Wie skaliert Gewünschtes besser als Unerwünschtes?

Störung

"Disruption", alles, was den die Qualität des Dienstes für die Nutzenden verringert



Störungsresistenz: Wie skaliert Gewünschtes besser als Unerwünschtes?

Störung

"Disruption", alles, was den die Qualität des Dienstes für die Nutzenden verringert

In Gruppen sammeln

- **Knotenauswahl:**
- **Suche:**
- **Verbreitung:**
- **Kommunikation:**



Störungsresistenz: Wie skaliert Gewünschtes besser als Unerwünschtes?

Störung

"Disruption", alles, was die Qualität des Dienstes für die Nutzer verringert

Auf jeder Ebene nötig

- **Knotenauswahl:** Verbindung mit Angreifern
- **Suche:** Spam, Falschinformationen
- **Verbreitung:** Dateien korrumpieren
- **Kommunikation:** Spam, Belästigung und Zensur⁶

⁶Das Web betrachtet Zensur als Störung und lenkt Anfragen darum herum.
The Internet treats censorship as a malfunction and routes around it. - John Perry Barlow



Störquellen

Sammeln am Flipchart



Störquellen

Sammeln am Flipchart

Quellen

- **Parasiten:** Bessere Leistung auf Kosten Anderer (leecher).
- **Trolle:** Kein Finanzinteresse, minimale Ressourcen, nutzen jegliche Lücke.
- **Spammer:** Erfolg durch Verbreitung eigener Inhalte.⁷
- **Konkurrenten:** Erfolg durch verringerte Qualität des Systems.
- **Angreifer:** Erfolg durch Schädigung von Nutzern.



Weitere Eigenschaft: Grad der Verteilung

Serverkoordinierte Teilgruppen bis vollständig dezentrale Interaktion.



Zusammenfassung

- **Einstieg:** Erste Adressen und Routing-Info
- **Suche:** Schlüsselwort, Inhalt, Public Key
- **Störung:** Parasiten, Trolle, Spammer, Konkurrenten und Angreifer
- **Verbreitung:** Tracker, Download-Mesh, Fragmente mit Caching
- **Kommunikation:** Privatnachricht, Forum, Neuigkeiten, Kommentare



Ziele

- Sie verstehen die grundlegende Funktionsweise von Gnutella als Beispiel einer effizienten, dezentralen Schlüsselwort-Suche.
- Sie erkennen, wo die für Gnutella entwickelten Techniken sinnvoll genutzt werden können.
- Sie wissen, welche Probleme ungelöst blieben.



Inhalt

- **Nutzersicht:** Das war Gnutella
- **Einstieg:** GWebCaches
- **Suche:** Slow-Start + Keyword-Multicast
- **Verteilung:** Download-Mesh
- **Kommunikation:** Neues und Sammlung zeigen
- **Störungsresistenz:** Heuristik oder Inhalts-Matrizen



Sicht der Nutzer/-innen

- 50 Millionen Knoten
- Globale Suche nach Dateinamen und ID3-Tags
- Filter für Creative-Commons-Lizenzen
- Suche nach den neusten Dateien (What's New?)
- Downloads von vielen Quellen ohne zentrale Koordination
- Audio-Streaming um 2004 („Dateivorschau“)
- LimeWire, Bearshare, Shareaza, Phex, gtk-gnutella, ...



Praktisch



Arne Babenhauserheide
Verteilte Systeme 1: peer-to-peer

Einstieg	Grundprobleme	Gnutella	Kademlia	BitTorrent Downloads	Freenet	Abschluss
○○○○○	○○○○○○○○○○	○○○○○○○○○○	○○○○	○○○	○○○	○○○○○

Implementierung (Grundlagen)

- **Adressen:** Webcaches
- **Verbindung:** HTTP-Handshake, dann Binär über TCP socket + out of band Antworten via UDP
- **Verbreitung:** HTTP + swarming
- **Kommunikation:** Via Suche

Weitere Knoten finden: X-Try

Beim Handshake (wie HTTP):

When rejecting a connection, a server **MUST**, if possible, provide the remote host with a list of other Gnutella hosts, so it can try connecting to them. This **SHOULD** be done using the X-Try header.

An X-Try header can look like:

X-Try: 1.2.3.4:1234,3.4.5.6:3456

Arne Babenhauserheide
Verteilte Systeme 1: peer-to-peer

Einstieg	Grundprobleme	Gnutella	Kademlia	BitTorrent Downloads	Freenet	Abschluss
○○○○○	○○○○○○○○○○	○○○○○○○○○○	○○○○	○○○	○○○	○○○○○

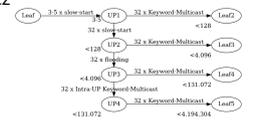
Weitere Knoten finden: Pong

Pong messages contains information about a Gnutella host. The message has the following fields

- Bytes: Description:
- 0-1 Port number. The port number on which the responding host can accept incoming connections.
 - 2-5 IP Address. The IP address of the responding host. Note: This field is in big-endian format.
 - ...
 - * When a Ping message is received (TTL>1 and it was at least one second since another Ping was received on that connection), a server **MUST**, if possible, respond with a number of Pong Messages. These pongs **MUST** have the same message ID as the incoming ping, and a TTL no lower than the hope value of the ping.

→ http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html

Suche im Netz



Nicht existente Datei durchschnittlich: $4 \times 32 \times 32 = 4.096$ Knoten.

Last (empirisch): <1kiB/s Leaf, <14kiB/s Ultrapeer

Ultrapeer (UP) Ein Hub, über den Kommunikation läuft.

Leaf Ein Randknoten, sucht über Ultrapeers.

Knoten Ein Peer oder Ultrapeer.

Arne Babenhauserheide
Verteilte Systeme 1: peer-to-peer

Einstieg	Grundprobleme	Gnutella	Kademlia	BitTorrent Downloads	Freenet	Abschluss
○○○○○	○○○○○○○○○○	○○○○○○○○○○	○○○○	○○○	○○○	○○○○○

Suche 1: Slow-Start

„Dynamic Querying“ (DQ)

- Leaf fragt einen UP nach dem anderen. Stoppt nach „genug“ Ergebnissen (um die 100).
- UP fragt Leafs und andere UPs. Stoppt nach „genug“ Ergebnissen.

Größe der Query Routing Tabellen in Gnutella

- Hashes: Normalisierte Suchwörter in der Suchanfrage oder im Dateinamen
- Größe: Variabel, Default in LimeWire 128kiB, interpolation auf größere und kleinere Tabellen möglich.
- Aktuell verfügbare Quelle: [BitSetQRTTableStorage.java](#)
- Hash-Funktion pro Suchwort: [HashFunction.java](#)

Arne Babenhauserheide
Verteilte Systeme 1: peer-to-peer

Einstieg	Grundprobleme	Gnutella	Kademlia	BitTorrent Downloads	Freenet	Abschluss
○○○○○	○○○○○○○○○○	○○○○○○○○○○	○○○○	○○○	○○○	○○○○○

Suche 4: Dateien nach Hash finden

- Zugriff auf Magnet-Links¹¹ brauchte exakte Dateisuche.¹²
- Angepasstes Kademlia \Rightarrow im Abschnitt zu Kademlia.

¹¹ Magnet-Links liefern Infos für Downloads in leicht kopierbarem Link.
¹² kt...: Suchanfrage, wurde kaum genutzt. Weiteres: https://en.wikipedia.org/wiki/Magnet_URI_scheme#Design

Verteilung: Download-Mesh

- Standard HTTP Range-Requests
- Content-Addressed: HOST/uri-res/raw/urn:sha1:HASH¹⁴
- 5 zusätzliche Header:¹⁵
 - X-Alt Bestätigte Quelle für die Datei, IP/Port
 - X-NAIt Unerrückbare Quelle oder Quelle mit Korruptierten Daten. IP/Port
 - X-Gnutella-Content-URN Merkle-Tree Root-Hash
 - X-Thex-URI /uri-res/N2X?urn:sha1:HASH;MERKLE_TREE_ROOT
 - X-Available-Ranges bytes 0-10,20-30 (Beispiel)

¹⁴ <https://www.ietf.org/rfc/rfc2169.txt> und <http://www.nuke24.net/40ca/2015/Hash08Wz.html>
¹⁵ http://rfc-gnutella.sourceforge.net/src/Partial_File_Sharing_Protocol_1.0.txt

Störungsresistenz: Objektvertrauen via Credence

- Jede korrekt bezeichnete Datei: 1.0
- Jede inkorrekt bezeichnete Datei: -1.0
- Wertungen anderer mit Korrelation der gemeinsamen Wertungen multipliziert.

→ <http://credence-p2p.org>

Wurde nie in ein Mainstream-Programm übernommen.

Arne Babenhauserheide
Verteilte Systeme 1: peer-to-peer

Einstieg	Grundprobleme	Gnutella	Kademlia	BitTorrent Downloads	Freenet	Abschluss
○○○○○	○○○○○○○○○○	○○○○○○○○○○	○○○○	○○○	○○○	○○○○○

Kommunikation: Schwachstelle

- Chat nie wirklich verlässlich
- Kein bleibender Kontakt zu anderen
- Funktionierend:
 - Was gibts Neues? (via LimeWire: Neueste Dateien sehen)
 - Sammlung durchsuchen (Alle freigegebenen Dateien sehen)



Verbleibende Schwächen 2008

- 10-20% Spam-Ergebnisse trotz 50 Millionen Nutzer.
- Credence nie weitverbreitet.
- Ein Schritt Flooding: Windows begrenzte Verbindungszahl.
- Parameter-Anpassungen beim Wachstum.
- Keine Kommentare, Peer-Chat wurde nie gut.

Arne Babenhauserheide
Verteilte Systeme 1: peer-to-peer

Einstieg	Grundprobleme	Gnutella	Kademlia	BitTorrent Downloads	Freenet	Abschluss
○○○○○	○○○○○○○○○○	○○○○○○○○○○	○○○○	○○○	○○○	○○○○○

Einstieg: Webcaches

Ursprünglich

- Liste langlebiger Hosts einkompiliert

Final

- Liste der letzten guten Knoten
- UDP Host-Caches: Minimalserver, die IP-Listen sammeln und die neuesten weitergeben
- Beispiel: GhostWhiteCrab⁹



⁹ gwc resource: <https://github.com/gtk-gnutella/gwc>

Arne Babenhauserheide
Verteilte Systeme 1: peer-to-peer

Einstieg	Grundprobleme	Gnutella	Kademlia	BitTorrent Downloads	Freenet	Abschluss
○○○○○	○○○○○○○○○○	○○○○○○○○○○	○○○○	○○○	○○○	○○○○○

Suche abschicken

```
<15 bytes GUID>0x00
0x80 ; message type: Query
0x07 ; TTL: 7
0x00 ; Hops 0
0x00,0x00,0x09 ; payload length, max: 4kiB
0x00,0x00 ; min speed
test foo ; payload: search criteria
0x00 ; null-terminator, begins extensions
```

GUID Globally Unique ID. Zufällig erstellt, um Schleifen zu vermeiden.

Arne Babenhauserheide
Verteilte Systeme 1: peer-to-peer

Einstieg	Grundprobleme	Gnutella	Kademlia	BitTorrent Downloads	Freenet	Abschluss
○○○○○	○○○○○○○○○○	○○○○○○○○○○	○○○○	○○○	○○○	○○○○○

Suche 2: Keyword-Multicast

Query Routing Protocol (QRP)

- Suchwörter normalisiert:¹⁰ lowercase, keine Akzente, ...
- Query Routing Table (QRT): Set mit schwachen Hashes von normalisierten Suchwörtern
- Automatisch hochskaliert für gewünschten Füllgrad

Intra-Ultrapeer-QRP:

- Vereinigung der Tabellen

Ähnlich: Bloom-Filter

¹⁰ ungelöst: Japanische oder Chinesische Zeichen.

Arne Babenhauserheide
Verteilte Systeme 1: peer-to-peer

Einstieg	Grundprobleme	Gnutella	Kademlia	BitTorrent Downloads	Freenet	Abschluss
○○○○○	○○○○○○○○○○	○○○○○○○○○○	○○○○	○○○	○○○	○○○○○

Verteilung in Gnutella: Out-of-Band

Ursprünglich auf dem Suchpfad zurückgereicht, aber:

- 5 Schritte
- Durchschnittliche Lebensdauer eines Knotens.¹³ 2h
- => Abbruch nach durchschnittlich 24 Minuten

Daher: Download-Mesh, unabhängig von der Suche



¹³ 2h als Lebensdauer sind erstaunlich persistent. Aktuelles bei Freenet.

Arne Babenhauserheide
Verteilte Systeme 1: peer-to-peer

Einstieg	Grundprobleme	Gnutella	Kademlia	BitTorrent Downloads	Freenet	Abschluss
○○○○○	○○○○○○○○○○	○○○○○○○○○○	○○○○	○○○	○○○	○○○○○

Störungsresistenz: Heuristiken als Spam-Filter

Ähnlich wie E-Mail-Spamfilter.

Hat Spam auf 10-20% der Ergebnisse reduziert.¹⁶

¹⁶ Ginge so ein Webshop? Wer braucht welche Garantien?

Arne Babenhauserheide
Verteilte Systeme 1: peer-to-peer

Einstieg	Grundprobleme	Gnutella	Kademlia	BitTorrent Downloads	Freenet	Abschluss
○○○○○	○○○○○○○○○○	○○○○○○○○○○	○○○○	○○○	○○○	○○○○○

Das Ende von Gnutella

2010: Die Zeit von Gnutella endete nicht durch technische Grenzen, sondern durch Klagen der Musikindustrie. Der Besitzer von LimeWire haftete mit seinem Privatvermögen. LimeWire bleibt freie Software, aber ohne große Verbreitung.¹⁷

Die entwickelte Technologie gerät in Vergessenheit.¹⁸

¹⁷ Die Geschichte von LimeWire: <https://melmagazine.com/en-us/story/an-oral-history-of-limewire-the-little-app-that-changed-the-music-industry>

¹⁸ Teile der Spezifikation: <http://rfc-gnutella.sourceforge.net/rfc-gnutella.rtp> und https://web.archive.org/web/20070429042042/http://www.the-rdf.org/index.php?title=Main_Page

Arne Babenhauserheide
Verteilte Systeme 1: peer-to-peer

Einstieg	Grundprobleme	Gnutella	Kademlia	BitTorrent Downloads	Freenet	Abschluss
○○○○○	○○○○○○○○○○	○○○○○○○○○○	○○○○	○○○	○○○	○○○○○

Gnutella Routing Experiment

- Peers: Tisch + davor + dahinter
- Letzte 2 Hops
- Suche nach Namen
- Hash = 1. Buchstabe
- QRT¹⁹: Hash der Namen der Peers
- Intra-UP QRT: QRTs der Peers, zusammengefasst

Was müsst ihr vorher austauschen?

¹⁹QRT: Query Routing Table.



Zusammenfassung Gnutella

- Effiziente Suche nach Schlüsselworten
- TCP-basiertes Binärprotokoll, 50 mio Nutzer, 1kiB/s Leaf, 14kiB/s Ultrappeer
- Einstieg: WebCache-Server + Austausch QRT (wie Bloom-Filter)²⁰
- Suche: Slow-Start + QRT Routing
- Verteilung: Download-Mesh
- Störungsresistenz: Heuristik oder Objektbasiert

²⁰Set von schwachen Hashes der Suchwörter, Anzahl keys dynamisch skaliert und interpoliert



Projektideen

- Download-Mesh implementieren
 - Nur Range-Requests + magnet für Quellen
 - Quellen-Gossip via XAlt²¹
 - Mit Merkle-Tree oder hashliste für chunks und mit XNalt
- Suche über WebRTC in Javascript
 - flooding über vereinfachtes Binärprotokoll
 - QRP / QRT
 - Sharing als Upload in local storage
 - GGEP: Generic Gnutella Extension Protocol; Binarprotokoll für beliebige Daten.

²¹XAlt/XNalt: Header, der gute / kaputte Quellen beschreibt.



PAUSE

--- PAUSE ---

Kademlia

Lookup in einer Verteilten Hash-Tabelle (DHT) mit xor-Metrik.

- Nutzersicht
- Suche
- Einstieg (nutzt die Suche)

Ziele

Sie verstehen die grundlegende Funktionsweise von Kademia als Beispiel einer effizienten, dezentralen Hash-Suche.
Sie erkennen, wo die in Kademia entwickelten Techniken sinnvoll genutzt werden können.

Sicht der Nutzer/-innen

Werkzeuge

Ursprünglich Tauschbörsen: Kad in aMule, VHT in Torrent clients
Amazon Dynamo verwendet das sehr ähnliche Chord.

Anwendung

- Suche nach exakten Dateien
- Löst Magnet-links auf
- Server-Auswahl zum Schreiben; eventual consistency



Suche in Kademia

- Jeder Knoten hat eine zufällige ID
- Suche nach Hash → Distributed Hash Table
- Distanz zwischen Hash und ID via xor-Metrik²²
- Schritt für Schritt in O(log(N)) zum richtigen Server

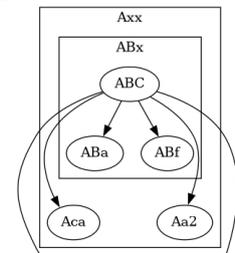


Ähnlich: Chord, Pastry.

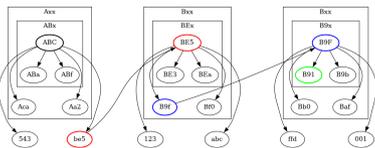
²²xor-Metrik: $4 \text{ xor } 2 \Rightarrow 100 \text{ xor } 010 \Rightarrow 110 \Rightarrow 6$.



Präfix-Buckets



Suche nach b91



Speichern

- Suche nach Knoten nahe Hash.
- STORE: Hash + Wert.

Einstieg in Kademia

- Kontakt zu mindestens einem bestehenden Knoten.
- Suche nach eigener ID: FIND_NODE (nah = zuständig für ID)
- Erhält Adressen + IDs der am nächsten liegenden Knoten
- Angefragte Knoten behalten auch die eigene Adresse und ID.



Kademia Routing-Experiment, Einstieg

IDs nach Sitzplatz:

11 - - - - - (Fenster)
10 - - - - -
01 - - - - -
00 - - - - -
000 001 010 011 100 101 110 111
(Tür) (Pult)

Präfix-Buckets (E=Eigener Bitwert):

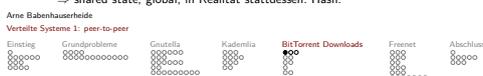
1 4 4 2 1
EEEEx EEEEx EEExx Exxxx xxxxx



Kademia Routing-Experiment, Suche

- ID berechnen (vereinfacht: zufällig²³)
- Name in ID speichern.
- Andere Person: Name abfragen

²³ shared state, global; in Realität stattdessen: Hash.



Projektideen

-
-
-
-
-

Zusammenfassung

- Distanz: key-hash XOR node-ID
- Suche: Nächstgelegenen bekannten Knoten nach besseren Knoten fragen
- Kennt mehr nahe als entfernte Knoten
- Speichern wie Suchen
- Einstieg:
 - Suche nach eigener ID
 - Erreichte Knoten nutzen Adresse und ID



BitTorrent

- Verbreitetste Lösung für Swarming
- BitTorrent, IPFS, Blizzard-Updater
- Upload für schnelleren Download
- Koordiniert durch Tracker
- Keine Suche



Ziele

- Sie kennen die grundlegende Funktionsweise von BitTorrent.
- Sie verstehen, wo BitTorrent durch teilweise Zentralisierung Komplexität vermeidet.
- Sie können erklären, warum Torrent für Twitter keine optimale Wahl war, trotzdem aber Faktor 100 schneller, als die vorherige Lösung.



Sicht der Nutzer/-innen

- Informationen von Tracker-Seiten
- Download mit torrent-Datei oder Magnet-Link
- Unterstützt Ordner
- Heute: ipfs: Webseiten über Bittorrent
- NAT-Traversing und IP Verschleiern über Tor



Konzept von BitTorrent

- Tracker: Webseite
- Koordiniert Schwärme
 - Suche, Foren, Bewertung, Verifizierung, Gemeinschaft
 - Statistiken: Seeder, Leacher
 - Liefert keine Daten
 - Aggregiert, wer wie viel hochlädt -> Anreiz



Anreiz zum Hochladen

- Uploadverhältnis wird geprüft
- Freeloader²⁵ werden von anderen clients gedrosselt (choked: niedrigere Downloadrate)
- In Literatur ist der Anreiz im Vergleich zu anderen Themen stark vertreten, in der Praxis sind die angebotenen Foren wichtig

²⁵Freeloader: Leute, die nichts hochladen. Auch „Leech“. Gegenteil: „Seed“.

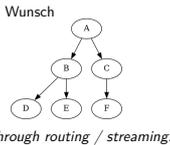


Weiteres

- VHT statt Tracker möglich (Kademlia)
- Freies Protokoll mit vielen Implementierungen
- Weiterentwicklung in der Community
- IPFS nutzt Torrents für dezentral gecachte Webseiten

Torrent für Twitter-Deployment

- Kosten bei Twitter: Übertragung über viele Schritte
- Torrent überträgt in Fragmenten.



Projektideen

-
-
-
-
-

Freenet/Hyphanet

Zensur-Resistente Kommunikation auf Freund-zu-Freund Netzwerk
Dezentrale Datenbank mit pubkey-Zugriff

- Ziele
- Verwendung
- Einstieg
- Small-World
- Suche
- Verteilung
- Mutability
- Kommunikation
- Schnittstellen



Ziele für den Freenet-Abschnitt

- Sie kennen Ähnlichkeiten und Unterschiede zwischen Kademia und Freenet
- Sie erkennen feste Freund-zu-Freund Verbindungen
- Sie erkennen die Small-World-Anforderung
- Sie verstehen, wie Freenet Daten versioniert und neue Versionen findet, ohne existierende Daten ändern zu können
- Sie kennen die dezentrale Spam-Abwehr in Freenet

Einstieg in Freenet

- Opennet:
 - Ähnlich Kademia: Wähle bekannte Seednode²⁶, Seednode sucht nach ID → Referenzen
 - Unterschied zu Kademia: Nicht nur IP, sondern Referenz mit Schlüssel
- Friend-to-Friend:
 - Feste Verbindungen
 - Knoten tauschen ihre IDs, um das soziale Small-World-Netzwerk zu rekonstruieren → Overlay-Kosten minimieren.



²⁶Seednode: Bekannter Knoten, der Verbindungen zu anderen vermittelt.

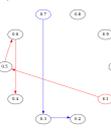


Small-World-Netzwerk (skalenfrees Netzwerk)

- Viele kurze und wenige lange Verbindungen.
- 6 degrees of separation via Post: Unsere Bekanntschaften bilden ein small-world Netzwerk
- Kleinberg-Netzwerk: Wahrscheinlichkeit verbunden zu sein: $\frac{1}{d^x}$, d = Distanz, x = Dimension.
- Freenet: x = 1

Freenet Suche

- Wie Kademia, aber hop für hop weitergeleitet → keine Globale Erreichbarkeit oder Sichtbarkeit
- Suche nach public key möglich
- Keyspace: [0.0 : 0.1)



Arten von Schlüsseln

- CHK: Content Hash
- KSK: Keyword Subspace: Passwort
- SSK: Signed Subspace: Public Key
- USK: Updatable Subspace: SSK mit Version

Format:
XXX@routing, encryption/tarball-name/path/to/file.ext
Ohne Pfad und Name möglich (kleiner → Optimierung).



Torrent-Datei

- Tracker URL(-s)
- Hashes für Chunks
- Namen der Datei(-en)
- Kann Ordner enthalten²⁴

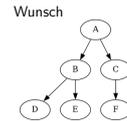


²⁴http://www.bittorrent.org/beps/bep_0003.html



Torrent für Twitter-Deployment

- Kosten bei Twitter: Übertragung über viele Schritte
- Torrent überträgt in Fragmenten.



Cut-through routing / streaming.



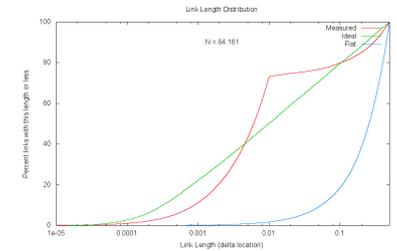
Zusammenfassung

- Tracker und Clients
- Tracker: Statistiken und Koordination
- Torrent-Datei mit Chunk-Infos

Verwendung

- Web-Schnittstelle
- Plugins mit E-Mail
- Externe Programme wie Chat und Foren mit Freenet als Datenbank via HTTP-ähnlicher API (FCP)

Theoretische und gemessene Link-Längen



Verteilung in Freenet

- Netz speichert Inhalte → verteilter Cache
- Dateien verschlüsselt, als 32 KiB Fragmente mit 100% Redundanz abgelegt
- Manifest enthält Schlüssel der Fragmente als CHKS
- Effektive LRU-Cache:²⁷
 - Speichern überschreibt zufällig gewählte Fragmente
 - Zugriff stellt überschriebene Fragmente wieder her
- Upload auf existierenden Schlüssel+Pfad: Kollision → In der Praxis immutable



²⁷LRU: Least Recently Used. Ältestes wird zuerst gelöscht.

