

# BitTorrent

Niklas Hofer <hofer@informatik.hu-berlin.de>

"Freedom of the press is limited to those who own one"  
A.J. Liebling.

# Gliederung

- 1 Einführung
  - Begriffsklärung
  - Entwickler
  - Szenario
- 2 Technik
  - up
  - down
  - Details
- 3 Real Life
  - Beispiele
  - Konkurrenz
  - Vorteile
  - Nachteile
  - Performance
- 4 Aussichten

# BitTorrent steht für

- das Protokoll



# BitTorrent steht für

- das Protokoll
- den Client (Python)



# BitTorrent steht für

- das Protokoll
- den Client (Python)
- den Dateityp `.torrent` (Metainfo)



# Bram Cohen



- Hauptentwickler

# Bram Cohen



- Hauptentwickler
- BitTorrent, Inc.

# Bram Cohen



- Hauptentwickler
- BitTorrent, Inc.
- seit 2004: Valve Software (video game patches)

# Bram Cohen



- Hauptentwickler
- BitTorrent, Inc.
- seit 2004: Valve Software (video game patches)
- offener Tracker auf <http://bittorrent.com>

# Bram Cohen



- Hauptentwickler
- BitTorrent, Inc.
- seit 2004: Valve Software (video game patches)
- offener Tracker auf <http://bittorrent.com>
- 23.Nov. 2005: Vereinbarung mit der MPAA, keine illegalen Kopien anzubieten

# Szenario



- Home Video

# Szenario



- Home Video
- Größe: ca. 300MiB

# Szenario



- Home Video
- Größe: ca. 300MiB
- Zielgruppe: die gesamte Welt oder „nur“ die Verwandtschaft (42 Personen)

# Szenario



- Home Video
- Größe: ca. 300MiB
- Zielgruppe: die gesamte Welt oder „nur“ die Verwandtschaft (42 Personen)
- hohe Nachfrage in kurzer Zeit

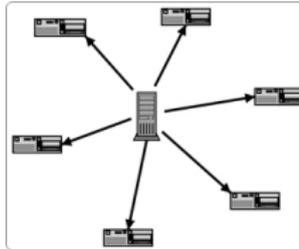
# Lösungsansätze

# Lösungsansätze

- FTP/HTTP über eigenen Webserver

# Lösungsansätze

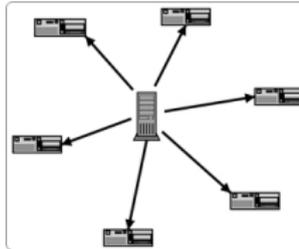
- FTP/HTTP über eigenen Webserver



- $300\text{MiB} \cdot 42 > 12\text{GiB}$

# Lösungsansätze

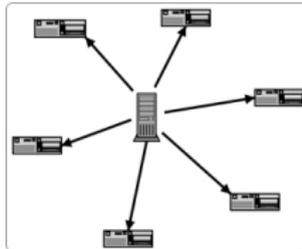
- FTP/HTTP über eigenen Webserver



- $300\text{MiB} \cdot 42 > 12\text{GiB}$
- Kosten? Geschwindigkeit?

# Lösungsansätze

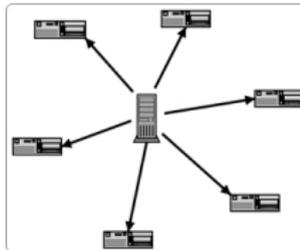
- FTP/HTTP über eigenen Webserver



- $300\text{MiB} \cdot 42 > 12\text{GiB}$
  - Kosten? Geschwindigkeit?
- Mirrors

# Lösungsansätze

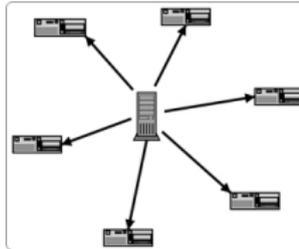
- FTP/HTTP über eigenen Webserver



- $300\text{MiB} \cdot 42 > 12\text{GiB}$
- Kosten? Geschwindigkeit?
- Mirrors
  - Verwaltung

# Lösungsansätze

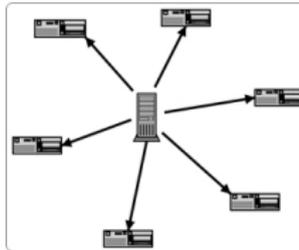
- FTP/HTTP über eigenen Webserver



- $300\text{MiB} \cdot 42 > 12\text{GiB}$
  - Kosten? Geschwindigkeit?
- Mirrors
  - Verwaltung
  - Einverständnis

# Lösungsansätze

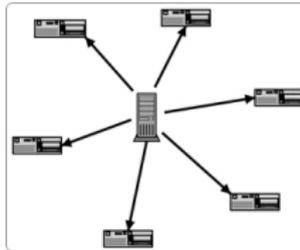
- FTP/HTTP über eigenen Webserver



- $300\text{MiB} \cdot 42 > 12\text{GiB}$
- Kosten? Geschwindigkeit?
- Mirrors
  - Verwaltung
  - Einverständnis
  - Verzögerung

# Lösungsansätze

- FTP/HTTP über eigenen Webserver



- $300\text{MiB} \cdot 42 > 12\text{GiB}$
- Kosten? Geschwindigkeit?
- Mirrors
  - Verwaltung
  - Einverständnis
  - Verzögerung
  - nur für wenige realisierbar

## Lösungsansätze 2

- traditionelle Tauschbörse

## Lösungsansätze 2

- traditionelle Tauschbörse
  - Overkill

## Lösungsansätze 2

- traditionelle Tauschbörse
  - Overkill
  - schwer zu finden (kein Meta)

## Lösungsansätze 2

- traditionelle Tauschbörse
  - Overkill
  - schwer zu finden (kein Meta)
  - Multikultur

## Lösungsansätze 2

- traditionelle Tauschbörse
  - Overkill
  - schwer zu finden (kein Meta)
  - Multikultur
- kommerzieller Anbieter

## Lösungsansätze 2

- traditionelle Tauschbörse
  - Overkill
  - schwer zu finden (kein Meta)
  - Multikultur
- kommerzieller Anbieter
  - Dateigröße oft begrenzt

## Lösungsansätze 2

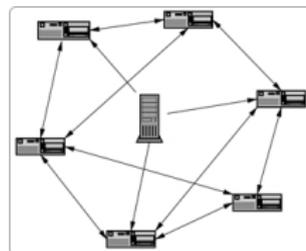
- traditionelle Tauschbörse
  - Overkill
  - schwer zu finden (kein Meta)
  - Multikultur
- kommerzieller Anbieter
  - Dateigröße oft begrenzt
  - Beschränkungen (Login, Werbung, ...)

## Lösungsansätze 2

- traditionelle Tauschbörse
  - Overkill
  - schwer zu finden (kein Meta)
  - Multikultur
- kommerzieller Anbieter
  - Dateigröße oft begrenzt
  - Beschränkungen (Login, Werbung, ...)
  - ... *kommerziell*

## Lösungsansätze 2

- traditionelle Tauschbörse
  - Overkill
  - schwer zu finden (kein Meta)
  - Multikultur
- kommerzieller Anbieter
  - Dateigröße oft begrenzt
  - Beschränkungen (Login, Werbung, ...)
  - ... *kommerziell*



- Wir brauchen also etwas ... anderes

# Tracker

Einen freien *Tracker* finden oder einen eigenen starten.  
Aufgabe:

# Tracker

Einen freien *Tracker* finden oder einen eigenen starten.  
Aufgabe:

- Liste der Benutzer

# Tracker

Einen freien *Tracker* finden oder einen eigenen starten.  
Aufgabe:

- Liste der Benutzer
- Verteilung der Fragmente

# Tracker

Einen freien *Tracker* finden oder einen eigenen starten.  
Aufgabe:

- Liste der Benutzer
- Verteilung der Fragmente
- Vermitteln

# Tracker

Einen freien *Tracker* finden oder einen eigenen starten.  
Aufgabe:

- Liste der Benutzer
- Verteilung der Fragmente
- Vermitteln
- Statistik

# Metafile

Das Metafile (.torrent) erstellen.

# Metafile

Das Metafile (.torrent) erstellen.

- Dateiname(n), Größe

# Metafile

Das Metafile (.torrent) erstellen.

- Dateiname(n), Größe
- Aufteilen in Fragmente a 256KiB (pieces)

# Metaverse

Das Metaverse (.torrent) erstellen.

- Dateiname(n), Größe
- Aufteilen in Fragmente a 256KiB (pieces)
- Hashes über jedes Fragment (SHA1)

# Metafile

Das Metafile (.torrent) erstellen.

- Dateiname(n), Größe
- Aufteilen in Fragmente a 256KiB (pieces)
- Hashes über jedes Fragment (SHA1)
- Adresse, Port des Trackers

# Metafile

Das Metafile (.torrent) erstellen.

- Dateiname(n), Größe
- Aufteilen in Fragmente a 256KiB (pieces)
- Hashes über jedes Fragment (SHA1)
- Adresse, Port des Trackers
- ⇒ .torrent-Datei mit  $< 1\%$  der Originalgröße

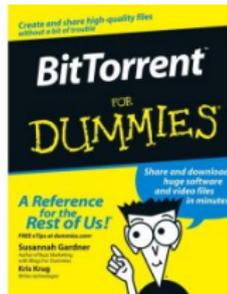
# Metaverse

Das Metaverse (.torrent) erstellen.

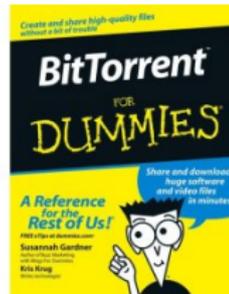
- Dateiname(n), Größe
- Aufteilen in Fragmente a 256KiB (pieces)
- Hashes über jedes Fragment (SHA1)
- Adresse, Port des Trackers
- ⇒ .torrent-Datei mit < 1% der Originalgröße
- Übertragung erfolgt in sub-pieces a 16KiB

# Publish

Publizieren des Metafiles!



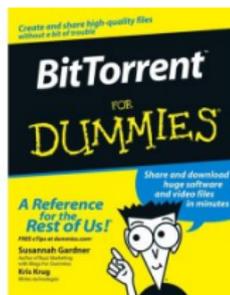
# Publish



Publizieren des Metafiles!

- Webserver, eMail, Link, ...

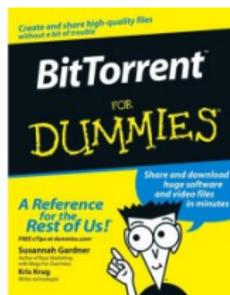
# Publish



## Publizieren des Metafiles!

- Webserver, eMail, Link, ...
- Schreibe darüber („I was with Nimrod“)

# Publish



## Publizieren des Metafiles!

- Webserver, eMail, Link, ...
- Schreibe darüber („I was with Nimrod“)
- *initial seed* (...)

# Ottormalverbraucher

# Ottonormalverbraucher

- .torrent finden

# Ottonormalverbraucher

- .torrent finden
- herunterladen

# Ottonormalverbraucher

- .torrent finden
- herunterladen
- einen Client starten

# Ottonormalverbraucher

- .torrent finden
- herunterladen
- einen Client starten
- warten.

# Ottonormalverbraucher

- .torrent finden
- herunterladen
- einen Client starten
- warten.
- warten...

# Ottonormalverbraucher

- `.torrent` finden
- herunterladen
- einen Client starten
- warten.
- warten. . .
- finished? enjoy, but keep sharing (etiquette)

# Ottonormalverbraucher

- `.torrent` finden
- herunterladen
- einen Client starten
- warten.
- warten. . .
- finished? enjoy, but keep sharing (etiquette)
- also: warten

# Behind enemy lines.

## Behind enemy lines.

- *initial seed* meldet „bin fertig“ an Tracker

## Behind enemy lines.

- *initial seed* meldet „bin fertig“ an Tracker
- neuer Client fragt Tracker nach Peers (*random list*)

## Behind enemy lines.

- *initial seed* meldet „bin fertig“ an Tracker
- neuer Client fragt Tracker nach Peers (*random list*)
- fragt Fragmente bei Peers an (*rarest first*)

## Behind enemy lines.

- *initial seed* meldet „bin fertig“ an Tracker
- neuer Client fragt Tracker nach Peers (*random list*)
- fragt Fragmente bei Peers an (*rarest first*)
- aber: *random first piece*

## Behind enemy lines.

- *initial seed* meldet „bin fertig“ an Tracker
- neuer Client fragt Tracker nach Peers (*random list*)
- fragt Fragmente bei Peers an (*rarest first*)
- aber: *random first piece*
- hash check

## Behind enemy lines.

- *initial seed* meldet „bin fertig“ an Tracker
- neuer Client fragt Tracker nach Peers (*random list*)
- fragt Fragmente bei Peers an (*rarest first*)
- aber: *random first piece*
- hash check
- vollständiges Fragment wird an Tracker gemeldet

## Behind enemy lines.

- *initial seed* meldet „bin fertig“ an Tracker
- neuer Client fragt Tracker nach Peers (*random list*)
- fragt Fragmente bei Peers an (*rarest first*)
- aber: *random first piece*
- hash check
- vollständiges Fragment wird an Tracker gemeldet
- ⇒ swarm

# Beispiele



# Beispiele



- World of Warcraft patches

# Beispiele



- World of Warcraft patches
- Filme von Fans (CC)

# Beispiele



- World of Warcraft patches
- Filme von Fans (CC)
- CC-Musik

# Beispiele



- World of Warcraft patches
- Filme von Fans (CC)
- CC-Musik
- \*BSD, Linux

# Beispiele



- World of Warcraft patches
- Filme von Fans (CC)
- CC-Musik
- \*BSD, Linux
- podcasting, videocasting, \*casting

# Konkurrenz



- Ähnlichkeit zu eDonkey2000 (Protokoll)

# Konkurrenz



- Ähnlichkeit zu eDonkey2000 (Protokoll)
- eher für kurzfristige, hohe Anforderungen

# Konkurrenz



- Ähnlichkeit zu eDonkey2000 (Protokoll)
- eher für kurzfristige, hohe Anforderungen
- kein Verschleiern des Anbieters

# Konkurrenz



- Ähnlichkeit zu eDonkey2000 (Protokoll)
- eher für kurzfristige, hohe Anforderungen
- kein Verschleiern des Anbieters
- Verantwortung: Betreiber des Trackers

# Konkurrenz



- Ähnlichkeit zu eDonkey2000 (Protokoll)
- eher für kurzfristige, hohe Anforderungen
- kein Verschleiern des Anbieters
- Verantwortung: Betreiber des Trackers
- viele verschiedene Clients

# Vorteile

## Vorteile gegenüber anderen P2P-Systemen

- mehr Teilnehmer erhöhen die Leistung des Netzes

# Vorteile

## Vorteile gegenüber anderen P2P-Systemen

- mehr Teilnehmer erhöhen die Leistung des Netzes
- *distributed copies*

# Vorteile

## Vorteile gegenüber anderen P2P-Systemen

- mehr Teilnehmer erhöhen die Leistung des Netzes
- *distributed copies*
- *leech resistance*

# Vorteile

## Vorteile gegenüber anderen P2P-Systemen

- mehr Teilnehmer erhöhen die Leistung des Netzes
- *distributed copies*
- *leech resistance*
- hocheffektiv für (stetige) Breitband-Verbindungen

# Vorteile

## Vorteile gegenüber anderen P2P-Systemen

- mehr Teilnehmer erhöhen die Leistung des Netzes
- *distributed copies*
- *leech resistance*
- hocheffektiv für (stetige) Breitband-Verbindungen
- free (as in mind) - BitTorrent Open Source License

# Vorteile

## Vorteile gegenüber anderen P2P-Systemen

- mehr Teilnehmer erhöhen die Leistung des Netzes
- *distributed copies*
- *leech resistance*
- hocheffektiv für (stetige) Breitband-Verbindungen
- free (as in mind) - BitTorrent Open Source License
- free (as in beer)

# Nachteile.

- kein graphischer Client (bzw. Suite)

# Nachteile.

- kein graphischer Client (bzw. Suite)
- kein Index

# Nachteile.

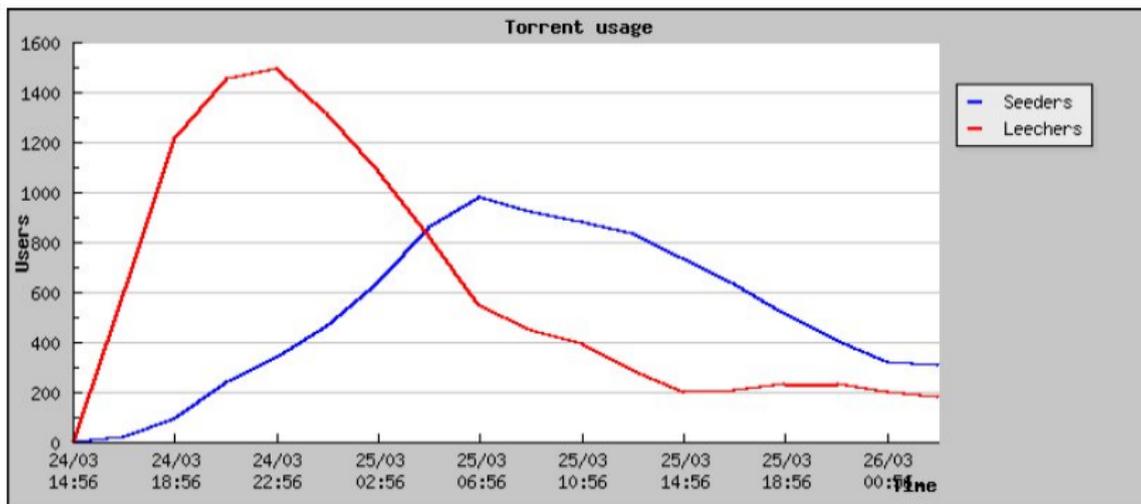
- kein graphischer Client (bzw. Suite)
- kein Index
- keine eingebaute Suche (nach Name)

## Nachteile.

- kein graphischer Client (bzw. Suite)
- kein Index
- keine eingebaute Suche (nach Name)
- deshalb Webseiten: suprnova.org († 12/2004), LokiTorrent, elitetorrents.org († 05/2005)

## Nachteile.

- kein graphischer Client (bzw. Suite)
- kein Index
- keine eingebaute Suche (nach Name)
- deshalb Webseiten: suprnova.org († 12/2004), LokiTorrent, elitetorrents.org († 05/2005)
- schlecht für Dialup



# Performance

- Up-/Downstream effizient genutzt

# Performance

- Up-/Downstream effizient genutzt
- aber:  $\sum U = \sum D$

# Performance

- Up-/Downstream effizient genutzt
- aber:  $\sum U = \sum D$
- Firewalls behindern

# Performance

- Up-/Downstream effizient genutzt
- aber:  $\sum U = \sum D$
- Firewalls behindern
- Tracker: < 10% Overhead (bottleneck)

# Performance

- Up-/Downstream effizient genutzt
- aber:  $\sum U = \sum D$
- Firewalls behindern
- Tracker: < 10% Overhead (bottleneck)
- Choking: upload verhindern

# Am Horizont.

- distributed trackers /trackerless (bt 4.1.2) mit DHT

# Am Horizont.

- distributed trackers /trackerless (bt 4.1.2) mit DHT
- Microsoft: Avalanche (paper, patents)

# Am Horizont.

- distributed trackers /trackerless (bt 4.1.2) mit DHT
- Microsoft: Avalanche (paper, patents)
- Dezentralisierung: eXeem (closed source)

# Am Horizont.

- distributed trackers /trackerless (bt 4.1.2) mit DHT
- Microsoft: Avalanche (paper, patents)
- Dezentralisierung: eXeem (closed source)
- *web seeding*

## Am Horizont.

- distributed trackers /trackerless (bt 4.1.2) mit DHT
- Microsoft: Avalanche (paper, patents)
- Dezentralisierung: eXeem (closed source)
- *web seeding*
- nur noch Links statt Dateien

# Am Horizont.

- distributed trackers /trackerless (bt 4.1.2) mit DHT
- Microsoft: Avalanche (paper, patents)
- Dezentralisierung: eXeem (closed source)
- *web seeding*
- nur noch Links statt Dateien
- Broadcatching: RSS+BitTorrent

# Quellen

- <http://en.wikipedia.org/wiki/Bittorrent>
- <http://www.bittorrent.com/>
- <http://www.heise.de/newsticker/meldung/66531>

„With BitTorrent free speech no longer has a high price.“  
Bram Cohen