

Tribler - nowe podejście do P2P

Tomasz Weksej

Uniwersytet Warszawski
Wydział MIM
Seminarium Systemy Rozproszone

13. listopada 2008

Co to Tribler?



- Nazwa pochodzi od *Tribe* - ang. plemię
- *Projekt badawczy* - eksperymentalny protokół i klient P2P
- *Działająca sieć* - 8 tys. użytkowników
- **Główna idea:** uwzględnić socjalne powiązania między użytkownikami
 - informacje o istnieniu użytkowników o podobnych gustach i znajomych wykorzystuje się przy **poszukiwaniu, rekomendowaniu i ściąganiu** plików

Co to Tribler? c.d.



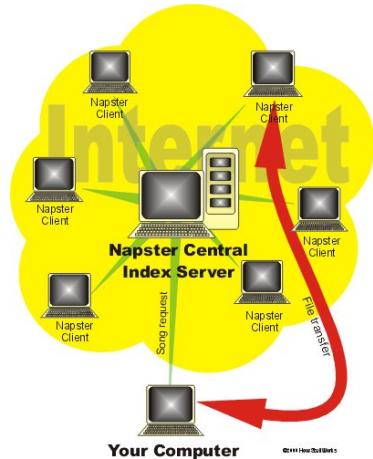
- Rozwijany przez pracowników naukowych, doktorantów i studentów holenderskich uczelni
- Oparty na protokole BitTorrent
- Służy zarówno do wymiany plików, jak i strumieniowaniu wideo

Napster - pierwszy system P2P

Jak to działa?

- 1 Użytkownik A łączy się z centralnym serwerem i umieszcza na nim listę swoich plików.
- 2 Użytkownik B wysyła do serwera zapytanie o plik.
- 3 Serwer odpowiada adresem IP komputera (np. użytkownika A).
- 4 Użytkownicy A i B łączą się i wymieniają się plikiem.

Z dzisiejszego punktu widzenia, Napster *nie jest* "prawdziwym" systemem P2P.



DHT - Distributed Hash Table

- Tablica haszująca H (plik \rightarrow adres IP komputera który posiada plik)
- W Napsterze - H jest zlokalizowane na centralnym serwerze
- W Gnutelli - brak H (= problemy z wyszukiwaniem)
- **Pytanie:** Jak rozproszyć H na wszystkie komputery w sieci?

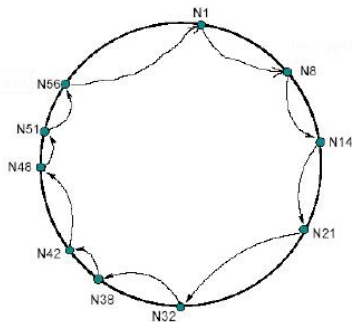
DHT - c.d.

- Pomysły: (n - liczba komputerów w sieci)
 - Każdy komputer posiada H .
 - Pierwszy komputer posiada $1/n$ wpisów, kolejny następne $1/n$ wpisów itd.
- Wpisy z H na odcinku:
 - Odcinek składa się z punktów $[0, 2^{60} - 1]$
 - Komputery opiekują się kawałkami odcinka.
 - Rozmieszczamy pliki na odcinku: bierzemy funkcję skrótu (np. SHA-1) i aplikujemy do niej nazwę pliku. Wynik modulo 2^{60} wskazuje nam miejsce na odcinku w którym umieszczamy dany wpis.

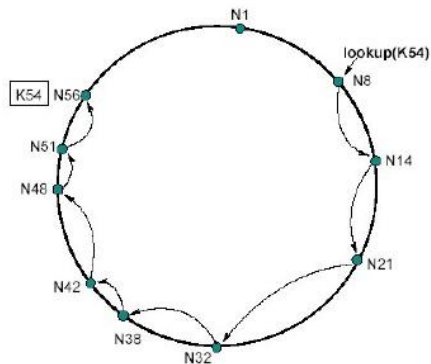
Chord

Chord (MIT, Berkeley, 2001)

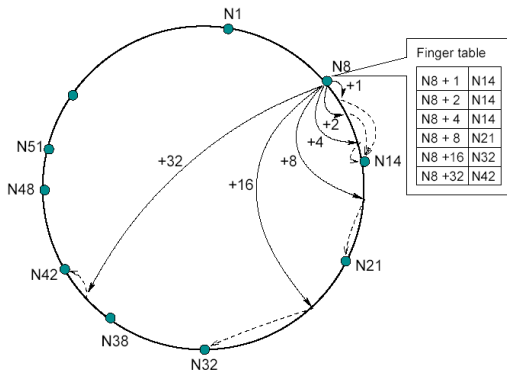
- **Założenie:** Każdy przechowywany obiekt ma unikalną nazwę znaną potencjalnym zainteresowanym
- Obiekt X przechowywany jest na komputerze $K(X)$; na różnych komputerach chcemy przechowywać pary $(X, K(X))$
- Dla każdego obiektu X i dla każdego adresu IP komputera obliczamy wartość funkcji SHA-1
- Obiekty i komputery umieszczamy na odcinku mod 2^{60} .



Chord - wyszukiwanie



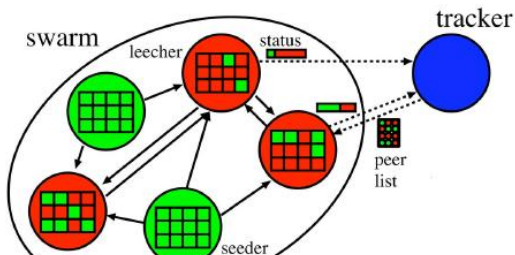
Chord - wyszukiwanie logarytmiczne



BitTorrent

Jak to działa?

- 1 Z jednego z serwisów www katalogującego zasoby ściągamy plik .torrent.
- 2 Łączymy się z trackerem i dostajemy od niego adresy IP komputerów będących w roju odpowiadającym temu plikowi.
- 3 Łączymy się bezpośrednio z poszczególnymi komputerami i wymieniamy posiadane przez nas części pliku.



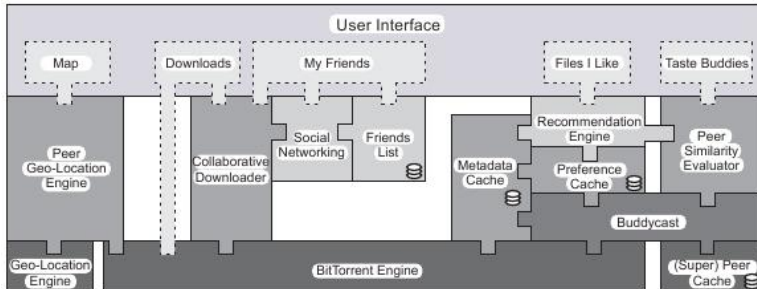
Wyzwania projektantów protokołów P2P

Zespół Triblera od 2003 roku analizował wydajność protokołu BitTorrent i wyróżnił następujące problemy którym trzeba sprostać:

- 1 **decentralizacja funkcjonalności**
- 2 **zapewnienie dostępności sieci**
- 3 **zachowanie integralności systemu i zaufania między węzłami**
- 4 **zapewnienie zachęty do udostępniania**
- 5 **zapewnienie transparentności sieci**

Tribler (zdaniem jego twórców) **taki jest.**

Architektura



Grupy społecznościowe

- Chcemy altruistycznego zachowania względem siebie - społeczności o podobnych zainteresowaniach to powodują
- Wybór pseudonimu = brak anonimowości = możliwość odróżnienia przyjaciela, wroga i nowego użytkownika
- Moduł *Social Networking* przechowuje informacje na temat grup społecznościowych (pseudonimy użytkowników, ich adresy IP, itp).

Megacache

- Jak jest:
 - Dzisiejsze protokoły P2P nie rejestrują informacji nt. poprzedniej aktywności w sieci
 - Wymieniane informacje między węzłami to tylko pliki i ew. informacje dot. ich poszukiwania
 - ... a Tribler musi pamiętać sporo innych informacji
- Cztery rodzaje Megacache:
 - *Friends List* - informacje o grupach społecznych
 - *(Super)Peer* - informacje o węzłach i superwęzłach
 - *Metadata Cache* - metadane plików
 - *Preference Cache* - listy preferencji innych węzłów
- Rozmiary - poniżej 10Mb
- Codzienna aktualizacja *Metadata Cache* - rzędu 600Kb, dzięki redukcji informacji o jednym pliku do ok. 400b

Poszukiwanie treści

- Rzecz kluczowa dla protokołu P2P. Jak to robią inni - już mówiliśmy.
- Tribler koncentruje się na *ludziach*, nie *plikach* - łączy użytkowników o podobnych zainteresowaniach (*taste buddies*).
- Moduł *Files I like* przechowuje informacje nt. lubianych plików
- Moduł *Peer Similarity Evaluator* porównuje listy preferencji i wyznacza podobieństwo gustów (cosinus wektorów ocen treści).
- Moduł *Recommendation Engine* tworzy listę plików potencjalnie interesujących użytkownika.

Pobieranie treści

- Moduł *BitTorrent Engine* pobiera pliki, używając protokołu kompatybilnego z BitTorrent.
- Możliwość korzystania z modułu *Collaborative Downloader* w celu zwiększenia prędkości pobierania.

Bootstrapping

- Bootstrapping - znajdowanie innych węzłów po uruchomieniu programu
- W BitTorrentcie - łatwo
- Kontakt z superwęzłem - pobranie listy innych węzłów
- Bycie członkiem specjalnego roju - *Overlay Swarm*

Sieci społecznościowe

- **Problem:** ograniczenie sesji
- Wprowadzono trwałe i unikalne identyfikatory węzłów (Permlid)
- Klucz publiczny

Buddycast

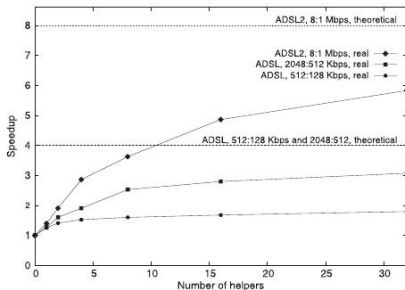
- **Problem:** znajdowanie treści - komunikacja między węzłami
- Protokół epidemiczny
- Potrzeba aktualizacji megacache
- Co 15s połączenie z osobą o podobnym guście (*taste buddy*) lub losową
- Komunikat: 50 plików z mojej listy preferencji + po 10 plików z listy preferencji 10 osób o podobnym guście

Filtry Blooma

- **Problem:** optymalizacja komunikacji między węzłami
- Friendster.com: średnio 243 znajomych, 9147 znajomych-znajomych
- filtry Blooma
- tylko 960b potrzebne, by (z dużym prawdopodobieństwem) wyznaczyć wspólnych znajomych-znajomych dwóch węzłów
- duży zysk w porównaniu z innymi algorytmami epidemicznymi!

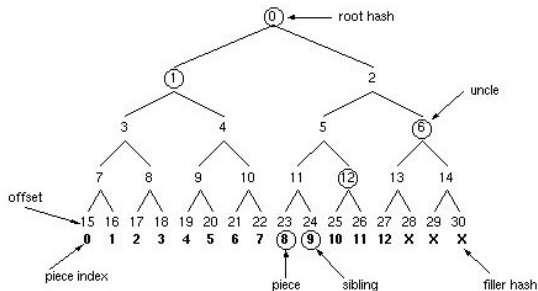
Collaborative Downloading

- **Problem:** zwiększenie wydajności pobierania
- Pomoc przy pobieraniu



Merkle Hash

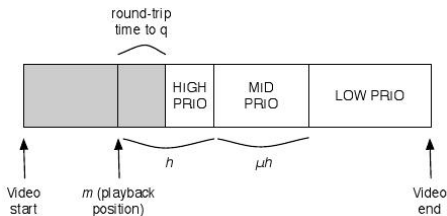
- **Problem:** zminimalizować rozmiar pliku .torrent
- "wąskie gardło" - sumy kontrolne części pliku



- minimalizacja informacji z zaufanego źródła

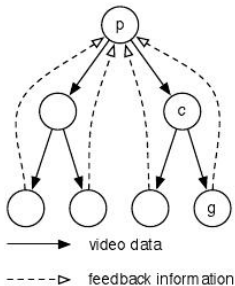
Give-To-Get

- **Problem:** dostosować protokół BitTorrent do potrzeb strumieniowania wideo
- Co trzeba poprawić:
 - wybór kawałków
 - problem z *free riding* - Tit-for-Tat
- wybór kawałków



Give-To-Get - c.d.

- Give-To-Get



Zdecentralizowany Tracker

- **Problem:** centralny Tracker = wąskie gardło
- Tracker udostępnia informacje o węzłach w roju
- Awaria Trackera - katastrofa
- Protokół epidemiczny
- Wejście do roju
- Pozyskanie informacji o większej ilości węzłów w roju
- A może DHT?

Social Overlay

- Wykorzystajmy socjalne powiązania do maksimum...
- Odległość znajomych
- Znamy PermId - jak wyznaczyć IP?

Buddycast3

- **Problem:** powolne rozprzestrzenianie się informacji o nowej treści
- Główny powód: ograniczenie do 50 plików
- Nowy komunikat nt. Torrentów:
 - preferences
 - collected torrents
 - ratings

Moderacja

- **Problem:** każdy może umieścić plik .torrent - jak zapobiegać powstawaniu śmieci?
- Dodatkowe parametry: język, napisy, opis, miniaturka, tagi
- Czarna lista moderatorów

Praca magisterska

- Zdecentralizowana radiostacja - Last.fm w Triblerze
- Give-To-Get
- Buddycast

Materiały

- <http://www.tribler.org/TriblerResearchSubjects>
- http://www.ii.uni.wroc.pl/mbi/dyd/sieci_08s

Dziękuję za uwagę

Pytania?