FAKULTA ELEKTROTECHNIKY A INFORMATIKY
SLOVENSKÁ TECHNICKÁ UNIVERZITA
Katedra Telekomunikácií

BitTorrent

Projekt číslo 1
z predmetu *Komunikačné protokoly*

Marek Banás
TLK-4 22028

2005/2006
2. 12 . 2005

Stručný popis

Bittorent je protokol určený k distribúcii súborov. Na rozdiel od bežného prenosu cez HTTP, alebo FTP, tu jednotlivé počítače sťahujúce súbor, zároveň slúžia ako servery a ponúkajú ku zdieľaniu už stiahnutú časť súboru. To znamená, že s rastúcim počtom klientov sú dáta na sieti zastúpené v stále väčšom počte kópií, čo znižuje záťaž pôvodného serveru, takže užívatelia nesťahujú súbor iba so serveru, ale aj medzi sebou. Všetky podporné aplikácie sú napísané v jazyku Python. Bittorrent je open source aplikácia vydaná pod MIT licenciou. Autorom Bittorrentu je Brian Cohen.

Architektúra

Tracker je aplikácia bežiaca spravidla na východziom serveri. Jeho úlohou je koordinovať spojenie medzi jednotlivými klientami a riadiť distribúciu súborou. Jeden Tracker môže koordinovať veľké množstvo súborov, jednotlivé relácie sú identifikované SHAl hashom svojho metasúboru. Odhaduje sa, že jeden trackej zvládne asi 10 000 užívateľov.

Metasúbor (.torrent) je súbor obsahujúci URL trackeru a informácie o súbore ku ktorým distribúcia slúžil. Jeho štruktúra bude podrobnejšie popísaná ďalej. Metasúbor pre dáta veľkosti CD-ROM je veľký zhruba 50 kB.

BitTorrent klient (downloader) je aplikácia slúžiaca k vlastnému sťahovaniu súborov. Zároveň slúži ako server a umožňuje stiahnutie už prenesených dát. Pokiaľ sú všetky dáta stiahnuté, Bittorrent klient slúži iba ako server.

Pre distribúciu súborov je potrebné zabezpečiť:

Web server - Na webovom serveri sú k dispozícii metasúbory (.Torrent) distribuovaným dátam. Spúšťanie downloaderu na klientskom je zabezpečené pomocou MIME asociácie (Application/x-Bittorrent).

Tracker - stačí jeden aj pre veľký počet súborov. Musí bežať na počítači prístupnom zvonka, PC by mal bežať podľa možností nepretržite.

zdroj - (downloader) pre každý metasúbor.

distribuované súbory

Klientský PC vyžaduje

- webový priehladač
- Bittorrent downloader

Komunikácia prebieha nasledovne:

- klient si vyžiada s web serveru metasúbor

- spustí sa downloader, užívateľ si vyberie, kam chce uložiť súbory popísané príslušným metasúborom
- pokiaľ súbory existujú, downloader si overí ich obsah (pomocou SHAl hashu). Pokiaľ nie, alokuje pre ne miestona discu (no vytvorí adresárovú štruktúru so súbormi vyplnenými samými nulami(00h))
- downloader získa od Trackeru náhodný zoznam ďalších downloaderov. Tento proces prebieha cez HTTP, alebo HTTPS.
- Downloader zasiela trackeru informácie o svojom aktuálnom stave, zároveň downloadery prenášajú dáta medzi sebou. To sa deje pomocou BitTorrent protokulu, ktorý je vybudovaný nad TCP.
- Zdroj sa chová ronako ako ostatné downloadery a kedže už má kompletné súbory, dáta, iba odosiela.

Popis protokolu

Komunikácia s Trackerem je založená na HTTP nadstavbe. Metasúbory.

Dáta sú posielané v ASCII, môžu obsahovať reťazce, celé čísla, zoznam (lists) a štruktúry (dictionaries). Kódovanie je nasledujúce:

Reťazec:	<dĺžka>:<text>
	Príklad: 10: Bittorrent
Celé číslo:	i<číslo>e
	Číslo nesmie začínať nulou, pokiaľ to nie je nula.
	Príklad: i42e
Zoznam:	l<položka><položka>…e
	Príklad: l10: Bittorrent5:pokuse
	[´Bittorrent´, ´pokus´]
Štruktúra:	d<kľúč><hodnota><kľúč><hodnota>…e
	Kľúč musí byť reťazec
	Príklad: d4: filel4: Bittorrent.txt6:lengthi3520ee
	(´file´:´bitorrent.txt´,´length´:3520)

Metasúbor (.torrent)je štruktúra s dvoma položkami:

Announce	URL trackeru
Info	štruktúra popisujúca dáta

Štruktúra info:

Name	doporučené meno súboru /východzieho adresáru
Piece lenght	dĺžka dátového bloku-menší je spravidla lepší, ale zväčšuje veľkosť metasúboru. Bežne sa používa 256KB.
Pieces	reťazec skadajúci sa z SHAl hashov jednotlivých dátových blokov. Každému bloku patrí 20 bajtov.
Lenght	dĺžka súboru v bajtoch
Files	zoznam súborov

Štruktúra info obsahuje buď položku *lenght* (pokiaľ popisuje jediný súbor) alebo položku *files* (pokiaľ popisuje adresárovú štruktúru s viacerými súbormi)

Položka files:
Zoznam štruktúr s 2 položkami:

Lenght	dĺžka súboru v bajtoch
Path	zoznam udávajúci cestu k súboru, položky sú jednotlivé pod adresáre výchozieho adresáru, posledná položka je názov súboru

Komunikácia s trackerom prebieha nasledovne:
Downloader zasiela trackeru HTTP GET žiadosť s nasledujúcimi parametrami:

Info_hash	SHAl hash štruktúry info z metasúboru
Peer_id	20 znakov dlhý identifikátor downloaderu. Generuje sa náhodne pri spustení.
Ip	IP adresa PC na ktorom downloader beží. Nepovinná položka.
Port	číslo portu na ktorom downloader beží. Implicitne prvý voľný port =>6881
Uploader	zatiaľ odoslané data
Downloader	doteraz prijaté data
Left	počet bajtov, ktoré je ešte potrebné stiahnuť
Event	nepovinná položka-viď ďalej

Sú dve zásady, ako sa klientská aplikácia a jej verzia kódujú do *peer- Id*
Azureus štýl :
‘ ‘, 2 znaky klientského id, 4 ASCII znaky pre číslo verzie, ‘ ‘, náhodné čísla
príklady klient id:

- 'AZ' - Azureus
- 'BB' - BitBuddy
- 'CT' - CTorrent
- 'MT' - MoonlightTorrent
- 'LT' - libtorrent
- 'BX' - Bittorrent X
- 'TS' - Torrentstorm
- 'TN' - TorrentDotNET
- 'SS' - SwarmScope
- 'XT' - XanTorrent
- 'BS' - BTSlave
- 'ZT' - ZipTorrent
- 'AR' - Arctic
- 'SB' - Swiftbit
- 'MP' - MooPolice
- 'lt' - libTorrent
- 'BC' - BitComet
- 'QT' - Qt 4 Torrent example
- 'UT' - µTorrent?

- 'SZ' - Shareaza

Shadow štýl je nasledovný:
Jedno ASCII písmeno pre client id, tri ASCII číslice pre verziu klienta, '---', níhodné čísla
Príklad klientov s týmto kódovaním:

- 'S' - Shadow's client
- 'U' - UPnP NAT Bit Torrent
- 'T' - BitTornado
- 'A' - ABC
- 'O' - Osprey Permaseed

Parameter *event* môže nabodúdať nasledujúce hodnoty

Started	pri začiatku stahovania
Completed	keď sú už stiahnuté všetky data- ďalej sa už neodosiala
Stopped	pri ukončení stahovania
<prázdny>	bežná správa-zasiela sa periodicky

Odpoveď je štruktúra v Pythonu zákodovaná rovnako ako metasúbor. Obsahuje buď kľúč *failure reason* s chybovým hlásením (ktoré sa zobrazí užívateľovi), alebo 2 kľúče:

Interval	doba v sekundách medzi pravidelnými žiadosťami. Downloader môže poslať žiadosť i mimo pridelený interval, pokiaľ napríklad nemá odkiaľ sťahovať
Peers	zoznam ďalších klientov pre komunikáciu

Položka *peers* je zoznam nasledujúcich štruktúr:

Id	identifikátor downloaderu
IP	IP adresa alebo DNS meno PC s downloaderom
Port	port, ktorý má downloader otvorený

Komunikácia medzi downloadermi
Spojenie je realizované na báze TCP. Je symetrické, teda správy v oboch smeroch vyzerajú rovnako a data môžu tiecť oboma smermi.
Po stiahnutí bloku a overení, že je neporušený (zhoduje sa hash) sú na to upozornené všetky downloadery zo zoznamu.

Každé spojenie má na každom konci 2 stavové bity*: interested* a *choked*.

interested	znamená, že druhá strana má nejaký blok, ktorý tento downloader ešte nemá.

choked — znamená, že spojenie je z tejto strany blokované- druhá strana nebude dostávať data, pokiaľ nebude odblokovaná. Počiatočný stav každého spojenia je *choked = 1* a *interested = 0.*

Pre zvýšenie efektivity prenosu sú bloky rozdelené do podblokov a vždy je k dispozícii poradovník žiadostí o podblok- akonáhle je jeden stiahnutý, vysiela sa žiadosť o ďalší.

Naviazanie spojenia (handshake)

- Spojenie začína znakom 19d, nasleduje reťazec „BitTorrent protocol“. Celočíselné hodnoty v ďalších fázach protokolu sú vždy big- endian na 4 bity.
- Po hlavičke nasleduje 8 bitov, ich hodnota je zatiaľ 0 a sú rezervované pre neskoršie rozšírenie.
- Ďalej nasleduje 20 bitov SHA1 hash štruktúry *info* z metasúboru.
- Nasleduje 20 bitov identifikačný kód downloaderu.

Tým končí handshake, nasledujú správy prefixované dĺžkou. Správy dĺžky 0 slúžia na udržanie spojenia, implicitne sa posielajú každé 2 minúty.

Prvý byte každej správy s dĺžkou > 0 určuje jej typ. Môže to byť

0- choke
1- unchoke
2- interested
3- not interested
4- have
5- bitfield
6- request
7- piece
8- cancel

Popis jednotlivých správ:

Správy *choke*, *unchoke*, *interested* a *not interested* nenesú žiadne ďalšie data a slúžia na oznámenie stavu bitov *choked* a *interested* na strane odosieľateľa. Posielajú sa pri zmene.

Správa *bitfield* sa zasiela na začiatku spojenia. Downloader jej oznamuje, ktoré bloky má už stiahnuté a k dispozícii. Data správy sú bitové polia. Jednotlivé bity počnúc najvýznamnejším odpovedajú jednotlivým blokom počnúc blokom s indexom 0. Pokiaľ downloader začína a nemá ešte žiadne dáta, správu *bitfield* môže vynechať.

Správa *have* obsahuje index práve stiahnutého bloku. Zasiela sa všetkým ostatným po úspešnom overení hashu.

Správa *request* slúži k vyžiadaniu podbloku od iného downloaderu. Obsahuje index bloku, začiatok a dĺžku. Ofset začiatku a dĺžka musia byť mocninou dvoch.

Správa *cancel* slúži k zrušeniu žiadosti poslanej správou *request*. Pareamtre má rovnaké.

Správa *piece* slúži k vlastnému pprenosu dát. Obsahuje index bloku, ofset začiatku a dáta.

Stratégie výberu, riadenie toku

Výber partnerov pre download
V súčastnej implementácii je uskutočňovaná náhodne. Je implementárne overené, že takto zákaný graf má vyhovujúce parametre. Alternatívou býva stromová topológia, tam sa ale nevyužije upload kapacita listov.

Pipelining
Aby počas prenosu neboli výpadky v prenose dát, sú bloky rozdelené na podbloky (spravidla veľké 16 kB). Každé spojenie má frontu spravidla 5 čakajúcich žiadostí a okamžite po obdržaní podbloku sa posiela ďaľšia žiadosť.

Začiatok downloadu
Je potrebné rýchlo získať aspoň jeden blok, aby bolo čo uploadovať (čo má za následok rýchlejší download). Snažiť sa v tomto okamihu zísať najvzácnejší blok v sieti by nebolo dobrou stratégiou, lepšie je však voliť také bloky, ktoré má viacej užívateľov (a tak je možné sťahovať viac podblokov naraz). Akonáhle je stiahnutý prvý kompletný blok, prejde sa na nasledujúcu stratégiu.

Download
Pri výbere bloku počas prvej fázy downloadu sa používa stratégia „najvzácnejší najskôr“, teda sa sťahujú tie bloky, ktoré má najmenej užívateľov v sieti. To znamená, že:

- pokiaľ zdroj neodošle všetky bloky, nikto nemôže ukončiť sťahovanie. Táto stratégia zaistí, že zdroj určite odošle všetky bloky skôr, ako sa niektoré začnú opakovať.
- vzácne bloky, v ktorých má výpadok klienta, ktorý ich poskytuje, najväčší dopad na rýchlosť celého systému, sa zálohujú prednostne.
- rozšírené bloky sa nechávajú na koniec, takže šanca, že klient s voľnou kapacitou pre upload nemá žiadne užitočné data, je menšia

Záverečná fáza
V priebehu downloadu môže nastať situácia, kedy niektorý blok je sťahovaný od užívateľa s veľmi malou prenosovou rýchlosťou. Normálne to nie je problém (je otvorených mnoho spojení súčasne a nízka rýchlosť jedného nemá na výsledok vplyv), v záverečnej fáze, kedy tento prenos môže byť posledný, to zbytočne zdržuje. Preto v okamihu, keď je na všetky ostatné podbloky odoslaná žiadosť (správa request), klient pošle žiadosť o všetky podbloky všetkým ostatným klientom v sieti. Akonáhle niektorý z nich obdrží, zruší túto žiadosť u všetkých ostatných klientov pomocou správy cancel. Táto metóda prináša veľké zrýchlenie záverečnej fázy downloadu za cenu mierneho zvýšenia záťaže siete (netrvá príliš dlho, takže negatívny dopad nie je nijako vážny).

Blokovanie a odblokovanie

Blokovanie (choking) sa používa z dvoch dôvodov:

1. pri súčasnej odchádzajúcej prevádzke cez veľký počet socketov nefungujú mecanizmy riadenia toku toku dát v TCP optimálne, je vtedy vhodné počet odchádzajúcich aktívnych spojení obmeziť
2. ako mechanizmus sa používa metóda„odmeňovania“ klientov poskytujúcich rýchly upload a naopak „trestanie“ tých, ktorí data len prijímajú

Blokované TCP spojenie zostáva otvorené , klient cez neho ale neposiela data.

BitTorrent klient štandartne používa 4 otvorené spojenia. Stav sa aktualizuje každých 10 sekúnd, kritériom pre výber je kĺzavý priemer rýchlosti downloadu z jednotlivých spojení za psledných 20 sekúnd.

V každom cykle klient odblokuje (choked=0) 4 klientov, od ktorých má najrýchlejší download a majú záujem o dáta (interested=1). Pokiaľ existuje klient s vyššou rýchlosťou downloadu, ktorý nemá záujem o dáta, je tiež odblokovaný. Pokiaľ sa v priebehu cyklu zmení jeho stav interested na 1, bude zablokovaný najpomalší zo 4 klientov vybratých predtým.

Táto metóda by sama o sebe neumožňovala zistiť, či niektoré z práve nevyužívaných spojení nie je lepšie ako tá práve využívané. Preto je v každom okamihu jeden klient odblokovaný bez ohľadu na rýchlosť downloadu od neho. To, ktorý klient to je, sa mení jenkrát za 3 cykly (30 sekúnd). Klient, ktorý zatiaľ nemá žiadne dáta, ktoré by mohol ponúkať, má 3x vyššiu šancu, že bude takto odblokovaný. Pokiaľ tento klient začne sťahovať (interested=1), počíta sa medzi 4 povolené aktívne spojenia, na rozdiel od nich ale nemôže byť zablokovaný, pokiaľ je najpomalší.

Dokončené sťahovanie

Po dokončení sťahovania sa už nie je možné rozhodovať podľa rýchlosti downloadu. Klient preto preferuje poskytovanie dát klientom, ku ktorým má najrýchlejší upload a klientom, ktorým momentálne nikto neposiela dáta.

Vytváranie .torrent súborov

- Vytvárjú sa pomocou „Make torrent“ aplikácie z c:\Program Files\BitTorrent\maketorrent.exe.
- Zvolíme si súbory, pre ktoré chceme vytvoriť .torrent súbor. Ak chceme pridať viac súborov, dáme ich do adresára a zvolíme adresár.
- Napíšeme URL trackeru. Ak chceme súbor bez URL, zvolíme „Use DHT“.
- Stlačíme tlačidlo Make torrent. Pre súbor s menom „file“ sa vytvorí „file.toorent“

Výhody a nevýhody

Výhody

Pre distribúciu súborov stačí linka s oveľa menšou priepustnosťou ako napr. pri FTP.

Výsledná prenosová rýchlosť u koncového užívateľa sa pri bežných pripojeniach blíži priepustnosti linky.
Systém dobre zvláda veľké nárazové zaťaženia (napr. pri uvolnení novej verzie programu), pretože s počtom požiadaviek rastie aj počet „serverov".
Systém by mal byť odolný voči žalobe proti výmenným sieťam, tie sú stále veľmi populárne, v BitTorrente sa však dá ľahko nastaviť, čo presne sa bude podieľať (jednalo by sa o drobnú modifikáciu zdrjového kódu trackeru, aby prijímal len spojenia, ktoré majú svoj hash uvedený v zozname).
Argumenty, že sa BitTorrent používa k nelegálnemu zdieľaniu súborov, by potom nemali šancu obstáť.

Nevýhody
BitTorrent downloader v priebehu sťahovania súboru zároveň odosiela veľké množstvo dát ostatným downloaderom. Pretože je to podmienkou pre dosiahnutie použiteľnej rýchlosti sťahovania (viď kapitola o riadení toku), nie je možné tento objem dát príliš obmedzovať. Vzhľadom k tomu, že v ČR toto často znamená obmedzenie celkvého objemu upload+download, je tento protokol v naších podmienkach pre užívateľa počítaného pripjenia značne nevýhodný (sťahovanie súboru je 1,5- 3x drahšie).

Alternatívy

Konvenčné riešenie

- väčší počet HTTP alebo FTP serverov, medzi ktorými je zaistený mirroring. Systém je doplnený webovou aplikáciou, ktorá užívateľa presmeruje na niektorý z menej vyťažených a pokiaľ možno blízky (tj. s rýchlym spojením) server. Nevýhodou je pre poskytovateľa dát nutnosť platiť pripojenie všetkých serverov na Internet. Je síce možné využiť služby špecializovaných serverov, čo je pre poskytovateľa lacnejšie (napr. download.com), v tom prípade je ale táto služba pre koncového užívateľa buď platená priamo, alebo z reklamy.

Download managery
Programy poskytujúce pokročilé funkcie pri sťahovaní súborov. Umožňujú naviazať prerušené spojenia, spúšťať sťahovanie v nastavenú dobu a mnoho ďalších funkcií.

PvP siete
K PvP (peer-to-peer) výmenným sieťam má Bittorrent najbližšie. Sám o sebe však PvP sieť nie je, je ho potrebné doplniť o systím umožňujúci zverejňovanie .torrent metasúborov bežným užívateľom a verejného trackeru (alebo viacerých).

Existujú v podstate 2 rôzne prístupy k realizácii takejto siete:
DirectConnect a jemu podobné siete
Skladá sa zo serverov a klientských aplikácii. Klientská aplikácia poskytuje ku zdieľaniu vybrané súbory z užívateľovho počítača. Po pripojení k hubu má užívateľ k dispozícii zoznam zdieľaných súborov všetkých užívateľov pripojených k hubu. Každý klient poskytuje obmedzený počet odchodzích spojení (slotov). Pokiaľ užívateľ žiada o download, čaká sa, kým sa niektorému zo zdieaľajúcich užívateľov uvoľní slot a súbor

je potom prenesený cez TCP-IP. Nevýhoda tohto riešenia je zrejmá – súbor sa vždy prenáša iba z jedného zdroja bez ohľadu na jeho rýchlosť. Výhodou je, že klientská aplikácia a software bežiaci na hub- e je Open- source.

eMule, Kazaa, Grokster...
Tieto siete sa tiež skladajú z cetnrálneho serveru a klientských aplikácií. Z pohľadu aplikácií existuje buď jediný server (Kazaa) alebo viac serverov (eMule). Na rozdiel od DirectConnectu sa tu súbor sťahuje prostredníctvom vlastného protokolu a aj z viac zdrojov naraz. Oproti BitTorrentu sú tu hlavne nasledujúce rozdiely

- užívateľ implicitne nezdieľa iba práve sťahovaný súbor, ale väčšie množstvo dát
- väčšina týchto sietí má majiteľa, ktorý teoreticky môže kedykoľvek zmeniť podmienky (eMule je výnimkou, ten je Open- Source)
- Okrem Open- source riešení tu nemožno vybudovať kompletnú infraštruktúru iba na vlastnom HW.
- Po technickej stránke je hlavný rozdiel medzi sieťami v použitých stratégiách pri riadení toku dá a zvýhodňovanie/nezvýhodňovanie užívateľov.

FTP, HTTP **BitTorrent**

Pužitá literatúra:
http://www.bittorrent.com
http://wiki.theory.org/BitTorrentSpecification
http://atm.felk.cvut.cz/mps/referaty/2003/capkao/
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd