

Kiedy jakaś dystrybucja Linuksa zaczyna zyskiwać popularność, coraz więcej użytkowników chce ją ściągnąć z Internetu. Dla producentów dystrybucji (zwłaszcza mniejszych) może to oznaczać problemy finansowe – należy zamówić i opłacić szybkie łącza, które pozwolą użytkownikom na bezproblemowe pobieranie plików.

CARSTEN SCHNOBER

Pobieranie plików przy pomocy BitTorrent

Rozproszone obciążenie



Popularność to dobra sprawa, ale do obsługi ogromnej liczby użytkowników, chcących natychmiast pobrać najnowszą wersję systemu, potrzebne jest niezwykle szerokie pasmo dostępu. Jeżeli administrator nie rozproszy jakoś tego obciążenia, może ono „zabić” serwis.

Użytkownicy, którzy chcą pobrać plik o dużym rozmiarze (np. obraz płyty CD-ROM z dystrybucją Linuksa) przez łącze komutowane, będą musieli uzbroid się w cierpliwość i wydać sporo pieniędzy, szczególnie, gdy pobieranie pliku utknie w martwym punkcie, mimo wystarczającej szerokości pasma. Twórcy oprogramowania, którym naprawdę zależy na sprawnej dystrybucji swojego oprogramowania wśród społeczności użytkowników Linuksa, muszą unikać tego typu sytuacji.

Teoria mówi...

Lekarstwem na te problemy może być BitTorrent. Osoby chcące udostępnić na serwerach duże ilości danych (np. wiele obrazów płyt CD) to pierwsze tzw. „ziarna” (ang. seed) w terminologii BitTorrent. Najpierw wszystkie pliki są dzielone na mniejsze pa-

kiety. Po pobraniu pierwszego pakietu klient BitTorrent (tzw. downloader) ustawia się w tryb „peer” i udostępnia pobrany pakiet innym użytkownikom. Zasada działania jest podobna do **serwerów lustrzanych**, chociaż komputer lokalny jest w posiadaniu tylko małej części pliku.

Kolejny użytkownik, chcący pobrać ten plik, może skorzystać albo z „ziaren” (seed) albo z dostępnych maszyn równorzędnych (peer), stając się jednocześnie kolejnym „peer”. Po pobraniu wszystkich niezbędnych pakietów i pod warunkiem, że nie przerwie pracy klienta BitTorrent, nasz komputer stanie się automatycznie serwerem, oferując pobranie pliku w całości, bez konieczności ściągnięcia poszczególnych pakietów. Nasz komputer stanie się zatem „ziarnem” (seed).

Taka architektura umożliwia rozproszenie szerokości pasma na całą grupę użytkowników. Im więcej osób będzie zainteresowanych danym plikiem, tym więcej będzie komputerów pracujących w trybie „peer”. Kiedy ustanowiona zostanie już wystarczająca liczba komputerów „seed” i „peer”, serwis pierwotnie udostępniający dany plik zostanie znacząco odciążony, a użytkownicy będą

z pewnością zadowoleni z dużej prędkości pobierania pliku. Liczba komputerów „peer” jest odzwierciedleniem zainteresowania daną ofertą. Serwis macierzysty nie będzie przeciążony, nawet jeśli liczba komputerów pracujących jako „peer” jest ograniczona.

Cały proces rozpraszania jest koordynowany przez tzw. szperacza (ang. tracker). Szperacz to serwer komunikujący się z komputerami w trybie „peer” i uzyskujący informacje na temat pobranych pakietów, które można zaoferować innym użytkownikom. Gdy nowy użytkownik typu „peer” kontaktuje się ze szperaczem, otrzymuje on listę aktywnych komputerów typu „seed” i „peer”, co umożliwia rozpoczęcie pobierania wybranego pliku.

Aby znaleźć właściwego szperacza dla danego pliku, tworzony jest plik Torrent zawierający istotne informacje na temat adresów internetowych innych szperaczy, dostępnych plików i ich rozmiarów oraz skrót (**Hash**) służący do jednoznacznej identyfikacji plików. Mimo że sam szperacz nie potrafi pobierać plików, wymaga jednak stałego dostępu do Internetu (używa niewielkiej szerokości pasma).

W praktyce

Aby rozpocząć pobieranie pliku, użytkownik musi pobrać plik Torrent zawierający definicję dla pliku/plików, które chcemy ściągnąć. Pliki Torrent znajdują się na serwerach sieciowych, rozproszonych w Internecie, które można łatwo zlokalizować przy pomocy wyszukiwarek lub odnośników. Załóżmy, że szukamy popularnej dystrybucji Linuksa – Fedora. Wpisujemy w ulubionej wyszukiwarce słowo Fedora wraz ze słowem Torrent. W ten sposób szybko trafimy pod właściwy adres. Jeżeli nasze próby okażą się bezowocne, zawsze możemy przejść na stronę główną BitTorrent [1] i tam szukać odpowiednich adresów.

Dystrybucja Debian Sarge i jej nowsze wersje zawierają już gotowy do użycia pakiet programu BitTorrent, który można zainstalować, wykonując polecenie `apt-get install bittorrent`. Użytkownicy innych dystrybucji mogą rozpakować do katalogu domowego pobrane archiwum, wpisując `tar -xzf BitTorrent-3.4.2.tar.gz`.

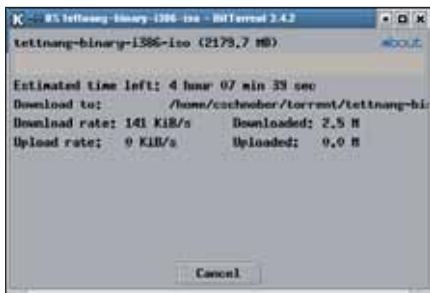
Klient programu BitTorrent został napisany w języku **Python** i wymaga obecności kilku dodatkowych zasobów. Poza podstawowym pakietem Pythona, klient GUI wymaga dodatków `wxWidgets` dla Pythona, które zarządzają wyglądem okien. W dystrybucji `SUSE` pakiet ten nosi nazwę `python-wxGTK`, a w `Mandrake` – `wxPythonGTK`.

Zapisujemy teraz plik Torrent na dysku twardym oraz uruchamiamy program klienta BitTorrent z graficznym interfejsem użytkownika w nowym katalogu `BitTorrent-3.4.2` i rozpoczynamy pobieranie pliku:

```
python btdownloadgui.py
tettngang-binary-i386-iso.torrent
```

W naszym przykładzie pobierzemy z sieci edycję Fedory „Tettngang” (Core 2). Użytkownicy Debiana mogą zamiast powyższego wykonać następujące polecenie:

```
btdownloadgui
```



Rysunek 1: BitTorrent pozwala na uzyskiwanie wysokich transferów przy pobieraniu plików.

```
tettngang-binary-i386-iso.torrent
```

Wskazujemy wybrany katalog docelowy i rozpoczyna się pobieranie pliku (patrz Rysunek 1).

Klient BitTorrent oferuje także tekstowy interfejs do pobierania plików, wystarczy zamiast `btdownloadgui.py` uruchomić plik `btdownloadcurses.py`. Istnieje też wersja do pracy z linią poleceń: `btdownloadheadless.py`.

Pokazana poniżej składnia powoduje uruchomienie równoległego pobierania plików:

```
python
btlaunchmanycurses.py ~/torrents
```

W tym przypadku wszystkie pliki Torrent przechowywane są w katalogu `~/torrents/`, a pobieranie plików uruchamiane jest w tym samym oknie tekstowym (patrz Rysunek 2). Nie istnieje obsługa równoległego pobierania plików dla klienta GUI. Możliwe jest jednak uruchomienie kilku kopii klienta GUI w tym samym czasie.

Narzędzie `btshowmetainfo.py` umożliwia uzyskanie dodatkowych informacji na temat zawartości pliku definicji Torrent. Aby zobaczyć nazwy i rozmiar poszczególnych plików dostępnych w pliku Torrent, uruchamiamy skrypt i podajemy nazwę pliku Torrent:

```
python btshowmetainfo.py
tettngang-binary-i386-iso.torrent
```

Coś za coś

Popularne oferty Torrent mają zawsze wolne miejsce dla nowego żądania pobierania pliku i umożliwiają uzyskiwanie znacznych prędkości przesyłania. Jeżeli jednak komputer „peer” otrzyma kilkanaście żądań pobierania plików w tym samym czasie, musi podjąć decyzję, które z nich obsłużyć w pierwszej kolejności. Komputer „peer” zawsze wybiera jako pierwsze żądanie z komputera oferującego największą liczbę pakietów w zamian. Tak więc pobranie pierwszego pakietu



Rysunek 2: Równoległe pobieranie plików w oknie tekstowym.

dla nowego komputera „peer” może zabrać sporo czasu, ale prędkość pobierania pliku będzie rosła z każdym pobranym pakietem, który będzie on udostępniał.

Technika ta zapobiega pobieraniu plików przez klientów, którzy niczego nie oferują w zamian – jest to uczciwe podejście. Jeżeli będziemy chcieli pobrać plik z komputera chronionego zaporą sieciową (ang. firewall) lub ruterem, sprawa komplikuje się, ponieważ tak chroniony komputer nie będzie mógł oferować żadnych pakietów do pobrania, a to oznacza niskie prędkości pobierania plików.

Jeżeli mamy możliwość zmiany konfiguracji zapor sieciowej, możemy rozwiązać ten problem, zezwalając na połączenia przychodzące na portach TCP 6881-6889 wykorzystywanych przez BitTorrent. W innym przypadku komputery „peer” nigdy nie skontaktują się ze swoimi ukrytymi za zaporami sieciowymi partnerami, do czasu, gdy sami wysłamy żądanie pobrania pliku. Komputery ukryte za zaporami sieciowymi mogą ciągle otrzymywać pozytywne odpowiedzi na żądania pobrania pliku, ale nie będą odpowiednio identyfikowane, zatem nie uzyskają tak dobrej prędkości transmisji danych jak komputery w „żywym” Internecie. ■

INFO

[1] BitTorrent:
<http://bitconjuror.org/BitTorrent/>

SŁOWNICZEK

Serwer lustrzany: Serwery lustrzane (tzw. mirror) przechowują te same pliki co serwer główny, aby go odciążać. Znajdują się one z reguły w różnych miejscach na świecie.

Hash: Skrót – ciąg znaków generowany z pliku przy pomocy nieodwracalnego algorytmu matematycznego. Te same dane wejściowe spowodują uzyskanie takiego samego skrótu (a więc różne dane wej-

ściowe spowodują uzyskanie różnych wyników). Tak więc skrót umożliwia nieomylną identyfikację plików.

Python: Język programowania dla Linuksa i innych systemów typu Unix oraz Windows, OS/2 i Mac OS. Python to język skryptowy, wymaga zatem jedynie interpretera do zamiany kodu programu do formatu zrozumiałego przez komputer.