

Bits acelerados com o BitTorrent

Carga distribuída

Quanto mais popular um software, mais largura de banda é necessária para distribuí-lo. Resolva este dilema sem gastar fortunas com novos links.

POR CARSTEN SCHNOBER

Popularidade é bom, mas vem às custas da necessidade de uma imensa largura de banda para atender às hordas de usuários vorazes que querem baixar a versão mais recente de um software a partir dos servidores oficiais. A transferência de dados pode se tornar um grande problema, caso as empresas não consigam distribuir a carga.

Usuários que queiram baixar um arquivo grande, como uma imagem de CD de uma distribuição Linux, usando uma conexão discada, vão precisar de muita paciência caso o download seja interrompido no meio do caminho. Desenvolvedores realmente interessados em distribuir seu software à comunidade precisam encontrar meios de evitar esse problema.

A teoria...

O BitTorrent (literalmente “Torrente de Bits”) é um conceito que pode ajudar ambos os lados. Pessoas que precisam hospedar grandes quantidades de dados - como uma ou mais imagens de CD - são os “semeadores” (seeders), no jargão do BitTorrent, responsáveis pela distribuição inicial do arquivo. Os arquivos são divididos em partes menores e, após baixar a primeira parte, um cliente BitTorrent automaticamente se reconfigura como um “parceiro” (peer) e oferece esta mesma parte às outras máquinas na rede. É mais ou menos o comportamento



de um **servidor espelho** (mirror server), com a diferença de que o cliente só tem um pedaço do arquivo completo.

O próximo usuário que quiser baixar este arquivo pode encontrá-lo em qualquer uma das fontes acima - o semeador ou o parceiro - automaticamente tornando-se um parceiro. Após receber todas as partes do arquivo, e desde que você não feche o cliente BitTorrent, sua máquina automaticamente se transforma em um servidor, oferecendo o arquivo completo aos outros membros da rede, sem precisar baixar mais nada. Em outras palavras, sua máquina também se torna uma “semeadora”.

Da forma como foi projetado, o sistema permite que a largura total de banda seja dividida entre o grupo de usuários. Quanto mais pessoas estiverem interessadas em um arquivo, mais parceiros existirão. E quando um grupo suficientemente grande de parceiros e

semeadores se formar, o distribuidor original do arquivo não precisará mais se preocupar com o consumo de banda de sua conexão e os usuários estarão felizes com a alta velocidade de download. O número de parceiros é proporcional ao interesse gerado pelo arquivo que está sendo oferecido; dessa forma a fonte primária não deve ter dificuldades com a rede (como sobrecarga), mesmo que o número de parceiros seja pequeno.

Um “rastreador” (tracker) coordena todo este processo. Um rastreador é um servidor que conversa com os parceiros, perguntando a eles quais pacotes já foram baixados e podem ser oferecidos na rede. Quando um novo parceiro se conecta a um rastreador, recebe uma lista de parceiros ativos e semeadores, de onde pode começar a baixar o arquivo.

Para ajudar a encontrar o rastreador apropriado, há um arquivo .torrent para cada arquivo (ou conjunto de arquivos)

GLOSSÁRIO

Servidor espelho: Para ajudar a aliviar um pouco a carga de um servidor, espelhos oferecem o mesmo conteúdo (arquivos) em locais diferentes.

Hash: Uma seqüência numérica gerada a partir de um arquivo por um algoritmo matemá-

tico não-reversível. A mesma entrada (arquivo) vai produzir sempre o mesmo hash; entradas diferentes produzirão hashes diferentes. Ou seja, o hash serve como um identificador único para o arquivo, já que não há dois arquivos com hashes iguais.

Python: Linguagem de programação para Linux e sistemas Unix, além do Windows, OS/2 e Mac OS. Por ser interpretada, não é necessário um compilador para transformar o programa em código de máquina, apenas do interpretador Python para executá-lo.

oferecido na rede, contendo informações cruciais como o endereço IP do rastreador, a lista dos arquivos oferecidos e seus respectivos tamanhos e um código "Hash" único. Um rastreador não baixa nenhum arquivo e, portanto, consome muito pouca banda; só é necessária uma conexão permanente à Internet.

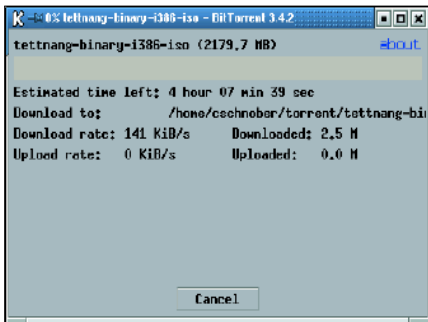


Figura 1: O BitTorrent permite downloads em alta velocidade.

...e a prática

Um usuário precisa do .torrent correspondente ao arquivo que deseja baixar antes de iniciar o download. Esses arquivos estão disponíveis em servidores espalhados pela Internet e podem ser facilmente encontrados com mecanismos de busca ou através de links em sites especializados. Supondo que você queira os CDs da distribuição Linux Fedora, basta digitar *Fedora* e *Torrent* em seu site de busca favorito para ser rapidamente levado ao endereço correto. Além disso, a página oficial do BitTorrent [1] traz alguns links.

A versão "Sarge" da distribuição Linux Debian contém um pacote com um cliente BitTorrent pronto para usar, que pode ser instalado com o comando `apt-get install bittorrent`. Se sua distribuição não contém um pacote do BitTorrent, baixe o código-fonte do programa no site oficial e descompacte-o com o comando `tar -zxvf BitTorrent-3.4.2.tar.gz`.

O cliente BitTorrent oficial é escrito na linguagem de programação **Python**, e precisa de alguns pacotes adicionais para funcionar. Além do próprio Python, presente em todas as distribuições Linux, o cliente gráfico precisa do conjunto de "widgets" (elementos de uma interface gráfica) `wxWidgets`. No Suse Linux este pacote é chamado de `python-wxGTK` e no Mandrake é chamado `wxPythonGTK`.

Salve o arquivo .torrent em seu disco rígido e execute o cliente BitTorrent gráfico,

encontrado no diretório BitTorrent-3.4.2 criado durante a descompactação do código-fonte, para iniciar o download:

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

Este exemplo inicia o download dos CDs da distribuição Linux Fedora Core 2 (codinome "Tettngang"). Usuários do Debian podem usar o comando a seguir:

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

Indique o diretório onde deseja colocar o arquivo baixado (veja a Figura 1).

Além do cliente gráfico, o BitTorrent também tem um utilitário para download em modo texto. Execute o programa `btdownloadcurses.py` em vez de `btdownloadgui.py` para uma interface com janelas em modo texto (usando a biblioteca `ncurses`), ou `btdownloadheadless.py` para a versão em linha de comando.

O comando a seguir executa múltiplos downloads em paralelo:

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

Este exemplo lê todos os arquivos .torrent no diretório `~/torrents/` e inicia todos os downloads em uma única janela em modo texto (veja a Figura 2). O cliente gráfico não possui suporte a múltiplos downloads simultâneos, mas você pode contornar o problema executando múltiplos clientes ao mesmo tempo.

A ferramenta `bbtshowmetainfo.py` lhe mostra informações sobre o conteúdo de uma torrent. Ao executar o script, especifique como parâmetro o nome de um arquivo .torrent para ver a lista e tamanho dos arquivos nele incluídos.

```
python bbtshowmetainfo.py ▶
tettngang-binary-i386-iso.torrent
```

Mercado de Trocas

As torrents se tornaram tão populares que praticamente todo pedido de download encontra ao menos um parceiro ou semeador livre. Contudo, se um parceiro recebe múltiplos pedidos de download ao mesmo tempo, ele precisa de um meio para determinar qual deles terá prioridade. O parceiro irá sempre esco-

lher o candidato que tem mais partes a oferecer em troca. Portanto, a primeira parte de um arquivo pode demorar um pouco a chegar, mas a taxa de download vai aumentar a cada parte extra que você ajudar a colocar na rede.

Esta técnica impede que os clientes baixem arquivos sem dar nada em troca. Essa é uma boa idéia, já que tal comportamento "predatório" impediria o equilíbrio de carga (*loadbalance*). As coisas ficam mais complexas quando você tenta se conectar através de um roteador ou firewall que bloqueie seu contato com o mundo exterior. Você fica impedido de oferecer partes do arquivo aos outros parceiros - isso significa menores velocidades de download, já que haverá mais pedidos que ofertas.

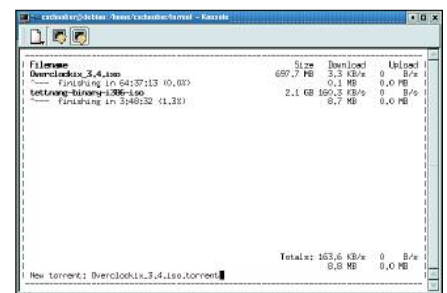


Figura 2: Múltiplos downloads em paralelo em uma janela em modo texto.

Se você puder alterar as configurações de seu firewall, pode resolver o problema permitindo conexões vindas da Internet nas portas TCP 6881 a 6889. De outra forma, parceiros que suportam acesso externo não tomarão conhecimento de seus colegas "ocultos" até que recebam um pedido deles, já que os parceiros anunciam automaticamente o que têm a oferecer quando conversam. Ou seja, clientes atrás de um firewall ainda vão receber uma resposta positiva aos seus pedidos de download, embora sejam menos conhecidos na rede e não possam desfrutar da mesma alta velocidade de download atingida por outros parceiros. ■

INFORMAÇÕES

- [1] BitTorrent: <http://bitconjur.org/BitTorrent/>
- [2] Azureus, cliente BitTorrent em Java: <http://azureus.sourceforge.net/>
- [3] Especificação do Protocolo: <http://bitconjur.org/BitTorrent/protocol.html>