End-to-end Bittorrent Publication

Andrew Wang, TJHSST Computer Systems Lab 2006-2007

Abstract

End-to-end publication through Bittorrent involves creating a .torrent metadata file, communicating with peers through a central "tracker," and an initial "seed" with a complete copy of the file. This project aims to simplify this process by providing a complete package that provides all the parts of this process.

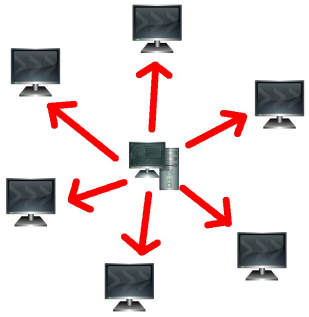

Fig A. Traditional Model

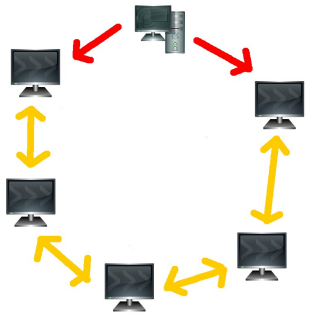

Fig B. Bittorrent Model

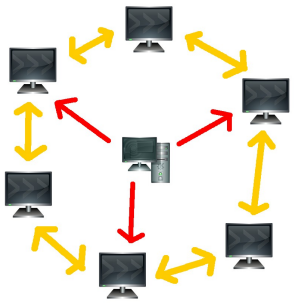

Fig C. Evertorrent Model

What is Evertorrent?

Evertorrent is my improvement upon the Bittorrent publishing system currently in place. Bittorrent itself has been recognized as an efficient means of distributing large files, but it has a single weakness: if there are no client has a complete copy of the file, the torrent "dies" because downloads can never complete.

By combining the traditional (Fig A) and Bittorrent (Fig B) models of distribution, Evertorrent (Fig C) was born. It introduces the concept of an "everseed" that is a single centralized client running on the same server as the tracker. This means that there will always be a complete copy of the file, and the torrent will never "die."

Procedure

Generation of a .torrent metadata file is the first step in Bittorrent publishing. This file tells the peers how to connect to the tracker.

The tracker handles connecting the peers with each other. A peer starting a download will request the IP addresses of other peers in the "swarm," and then connect to download the file.

The innovative "everseed" is the final step, that builds on the current system by preventing torrents from ever "dying."

Expected Results

Metadata files can be successfully generated by my bencoding algortihm and loaded by popular Bittorrent clients such as Mainline and Azureus.

The tracker handles both announce and scrape requests from the test client correctly and has a number of performance enhancements like client status-based peer list response and peer list compression.

The final step is testing and a web-based interface.