

Proposal:

Drupal is an amazing, flexible Content Management System that, on its own and through modules can present and store a large variety of information. This data can be in the form of a blog entry about one's weekend or it could be a high resolution image of our solar system. With Drupal, a vast array of information and data can be shared within a community. With the advent of broadband, the sizes of files available for download on the internet have increased dramatically. Files that are multiple gigabytes in size can be downloaded in under an hour. The increased bandwidth offered by broadband comes with a price to some website owners - the bandwidth that their site uses. For example, suppose someone has a gallery of high resolution photos of the planets in our solar system and each one is about 200MB in size. Each time someone goes to download one of these images, they use 200MB of his or her website's bandwidth. This is assuming the download completes without error, if an error occurs the file would have to be re-downloaded. Should his or her site have a high volume of traffic, the bandwidth used by the file downloads alone will exceed the bandwidth used to simply display the galleries of images. BitTorrent is a solution that will decrease the amount of download bandwidth used, increase the download speed for his clients, and ensure download quality.

Through the BitTorrent protocol multiple *peers* downloading the same file request different pieces of that file from either peers who already have the piece they need, *seeders* (peers who have all of the pieces of the file), or the server (acting as a seeder). In this method, as the peers download pieces of the file they are uploading pieces that they already have to other peers. Peers and seeders determine who to contact by contacting a *tracker* for their download. The tracker does what its name implies, it tracks. Clients contact the tracker to inform the tracker of its progress and receive a list of current peers to contact and from whom to attempt to download pieces. Once a client receives a piece of the file, it is checked against a known hash of that piece to ensure download quality. The URL of the tracker, information about the file or files, and hashes of the file pieces are contained within a *.torrent* file. The user opens the *.torrent* file with their BitTorrent client and the client contacts the tracker to receive an initial list of peers to contact and from whom to request pieces. The advantages of BitTorrent are best summarized by Wikipedia (<http://www.wikipedia.org>), "BitTorrent is a method of distributing large amounts of data widely without the original distributor incurring the whole of the corresponding costs of hardware, hosting and bandwidth resources." Wikipedia also has an excellent visual example of the BitTorrent protocol located at http://en.wikipedia.org/wiki/Image:Torrentcomp_small.gif.

Creating a tracker module for Drupal will alleviate bandwidth usage on websites with large file downloads and high traffic, as well as increase the security of the file being downloaded (through hash checking). The goal of this project is to implement a tracker module for Drupal. It will consist of announce and scrape files for the "invisible tracker" portion of a BitTorrent tracker, a Drupal module that manages current torrents and IPs as well as live tracker statistics such as number of seeders and peers. The module should also have an option for *Server Side Seeding* should a torrent not have any seeders.

Drupal benefits-

- A simple interface to create and track torrents
- A module that can easily be implemented in a variety of Drupal environments

- Focus on ease of installation and minimal database interaction.
- Adding distributed downloading technology to the Drupal CMS.

Requirements to Succeed-

- The module must be cross platform and browser compliant
 - The tracker should work on IIS and Apache with PHP 5 and MySQL 5. (I will also check for compatibility with PHP and MySQL 4)
 - The module interface must work in all major browsers.
 - IE 6 & 7
 - Gecko based browsers
 - Opera 8 and 9
 - Safari
 - The tracker must interface with most major BitTorrent Clients (hopefully all that implement the protocol)
 - Azuereus
 - BitTorrent
 - μ Torrent
 - BitComet
 - BitLord
 - BitTyrant
- Ease of installation
 - It must be easy to install the module and set up the tracker
- Ease of use
 - The user must be able to ban IP addresses (or ranges).
 - The user must be able to add and delete torrents.
 - The user must be able to view and reset statistics
 - The server should automatically seed a torrent if there are no seeders available (possibly only the pieces that are not available).

Timeline April 9th – August 20th

This project will take most of the time allotted to maximize quality code and minimize errors. I will use the interim period to become involved with the community, research the current protocols and review current solutions used in order to develop an optimized approach when the coding actually starts on May 28th.

- **April 9th – May 28th**
 - Become active in the Drupal Community
 - Join IRC and forum communities to both introduce myself and learn more about the Drupal project and environment.
 - Research the BitTorrent Protocol
 - Includes, but is not limited to:
 - Client – Tracker Requests: Announce and Scrape data, what the tracker actually receives from the client and what that information translates to in relation to the torrents.

- Tracker – How the tracker responds and what data is sent back to the client.
 - Server Seeding or WebSeeding
 - Research and evaluate current tracking solutions:
 - Look at current tracking solutions such as phpMyBitTorrent and analyze how they function, determine any shortcomings and how to create an optimized tracker for a Drupal environment.
 - Look at current Drupal modules and determine how to integrate a tracker into Drupal. Example: Banning IPs, adding torrents and removing torrents, and statistics for the tracker.
- **May 28th – July 9th**
 - Create Core Functions and Files
 - Generate the framework for the project:
 - Database schema
 - Determine number of total queries needed by the tracker to minimize database interaction
 - Create announce.php and scrape.php files
 - Concentrate on minimal database interaction
 - Create encoding/decoding functions to handle data extraction from HTTP requests
 - The majority of this time will be spent on the “invisible tracker” portion of the proposal. Towards the end of this time, I will begin work on the interface for the Drupal module.
- **July 9th – August 20th**
 - Testing for “invisible tracker” portion of the project with multiple clients and verifying usability between multiple client solutions
 - Develop management / statistics portion of the Drupal module and test interaction with the tracker portion
 - Test adding and removing torrents
 - Gather statistics (number of torrents, seeders, lechers, swarm, etc.)
 - Work on “extras” after core portions of the project are finished and tested.
 - Torrentizer – create torrents for files that meet certain criteria.
 - Server Seeding – The web server provides the initial seed or seeds pieces of the torrent which are not currently available(this may be moved to a core part of the project, but only after other work is completed)

Biography:

My name is Christopher Bradford and I am twenty years old. I am an Information Systems major at Virginia Commonwealth University located in Richmond, Virginia, USA. I am currently employed by Virginia Commonwealth University Libraries as a Web Application Developer for the Information Systems department. Presently I am pursuing a Bachelor of Science in Information Systems and a minor in Computer Science. I have been programming for six years in a variety of programming languages including Java, JavaScript, HTML, XHTML, CSS, PHP, Perl, and ColdFusion. My most recent projects include the implementation of the redesign of the VCU Libraries website

(<http://www.library.vcu.edu>) and their staff directory (http://www.library.vcu.edu/cfapps/nts/ulsdir_action.cfm?dept=all). I maintain a regular blog at <http://blog.vcu.edu/bradfordcp/> which contains snippets of code I am currently using. I am familiar with many operating systems and environments from a UNIX shell to OS X. I have a passion for coding and creating usable web applications. Last year, I created a conference management system for registration and processing of conference attendees that was written in PHP with a MySQL backend. If I am creating an application and I come to a problem, I will ask for help from others and perform further research until I can fix the issue or come up with another solution. I am proud that I have never given up - I keep working and looking until I find an answer to my problem and come up with a solution that fixes my problem.

BitTorrent greatly interests me on many levels- I have always been interested in network applications and the sharing of data between two people through a network. I have used BitTorrent for a few years now, mostly to download Linux ISO files which can range from 600MB to 4GB. One thing that intrigued me was not only how the pieces were sent between peers, but how the peers “discovered” each other. After some research, I thought trackers were neat in the way they connect peers. Following the advice of my supervisor, I started to look at Google’s Summer of Code. I started randomly clicking through projects’ idea web pages until I came to Drupal’s idea page and a suggestion had been made to implement a BitTorrent tracker module within Drupal. I jumped at the chance to work in Summer of Code as well as develop a tracker for the BitTorrent protocol. I have experience with PHP and MySQL, both of which are used by Drupal, and I believe I have the drive and determination for this project. I am floored by the idea of working on something of this scale. I have the opportunity to work on a project that genuinely interests me and on top of it all, my work will benefit an open source project. I have the skills necessary for this project along with the passion and drive to implement such a solution.