

# Task Oriented Torrent Downloading Application

## Media Companion

G.H.Achintha Reemal

Department of Computer Science and Engineering

University of Moratuwa, Sri Lanka

achintha.11@cse.mrt.ac.lk

**Abstract**—This paper identifies the evolution and current developments in Task Oriented Torrent Downloading Application, with reference to the Media Companion System, which takes into consideration different key torrent based download manager applications. This further describes the technologies being used in the implementation of Media Companion, which is the new methodology proposed in this paper, along with the reasoning as to why they were used.

**Keywords**— *Media Companion; Representational State Transfer (REST); Torrent Client; Application Program Interface (API)*

### I. INTRODUCTION

Torrent sharing has become a trending method of sharing different types of media components all around the world. It has proven to be a convenient method of transferring files between different peers without the intervention of a parent (central) node [1]. The core concept of torrent sharing lies in the method of gathering pieces of a file which the user wants and downloading these pieces simultaneously from people who already have them. This process makes popular and very large files, such as videos and television programs, download much faster than is possible with other protocols [2].

Most recent developments of file sharing through torrents have given rise to a whole new domain of entertainment, which is sharing episodes of television series. Even though this has given rise to certain legal issues related to copy write protected file sharing, torrents of TV series episodes are being downloaded and seeded in large numbers throughout the world. Different vendors have developed their own torrent clients over the years which have entered the software market leaving their mark. However, lack of specific task oriented programs has created a void, making things difficult for the users, having to take the trouble of downloading specific torrent files and then separately downloading the file using a torrent client.

This issue has been addressed by several vendors which focuses on downloading user specified TV shows and movies, making things easier for movie, TV show fanatics who are looking for convenient ways of getting their hands on TV shows and movies. Technologies used in developing such programs are critically analyzed and reviewed under this description. Furthermore, these applications will be compared against Media Companion and its functionalities.

### II. LITERATURE REVIEW

Task oriented torrent downloading applications are specifically designed to download user specified movies and TV show episodes with or without the involvement of the user. User has the ability to suggest their favorite TV show or movie and the application would take care of downloading the torrent file and then downloading the media files, without the user having to manually undertake these activities.

When developing this type of applications, identifying suitable technologies and the extent to which it is suitable should be clearly identified in order to design an efficient torrent application. When identifying these key technologies, following categorization of key components of a Task oriented torrent downloading applications would be helpful.

#### 1) *Identifying download links*

Identifying and capturing download links of a specified movie/TV series would be a critical part in the process of downloading the content of the torrent file. A convenient solution for this process is scraping web contents to capture download links [3]. However this solution has the inherent disadvantage of possible web component changes, which would make web scraping difficult and less effective.

Another possible solution would be to use a dump file of a torrent sharing web site, which would include download links of available TV shows/movies at a given time. This technique would require constant string searching in order to find a specific download link of a TV show/ movie. Therefore establishing an optimized search algorithm would be the key to develop an efficient system.

The best possible solution would be to use Rich Site Summary (RSS) feeds provided by different web sites. These feeds would provide details of availability of a certain torrent, download details etc. This method of gathering information would require minimum effort and efficiency would be considerably higher. However, RSS feeds would be generally issued for the availability of number of different torrents, but not for a specific torrent file, hence identifying the availability of a specific file and gathering its information would be somewhat difficult.

#### 2) *Check for availability*

Once a user specifies his/ her favorite set of TV shows or movies, torrent application would have to frequently check for

the availability of specified program. Similar to above mentioned component, this process has 2 different approaches.

One such method would be polling [4] for availability of a given torrent file. This is achieved through multi-threaded application development, in which a separate thread would constantly check for the availability of a certain file using any of the above stated processes. However this would be seen as a waste of processor resources.

Another possible solution would be to use interrupt driven application development [5] to check for torrent files. This method does not require constant search for the availability of certain files, hence eliminating unnecessary processor resource wastage.

### 3) *Seeding and sharing torrents*

Once all the above mentioned processes are taken care of, next big ask would be to undertake the torrent file content download process.

Since there are already developed torrent clients (Vuze [6], BitTorrent [7], µTorrent [8] etc.), developing an extension for these client would seem convenient. However, developing such extension/plugins require additional requirements and technical knowhow, which would prove difficult for a sophisticated project involving several process as stated above.

When considering the implications in developing an extension for an available client, it is significantly convenient to use freely available libraries like Ttorrent BitTorrent Java Library [9], Deluge [10] etc. and incorporate them to the developed application as an external library. Then customization of relevant classes can be done to adjust these libraries to the developer's needs and wants.

### 4) *Gathering show information for the user to evaluate*

Another key aspect of this type of application would be providing TV show related information for the user so that user could make informed decision as to download a particular TV show or not. This basically includes retrieving Internet Movie Database (IMDB) ratings or Rotten Tomatoes ratings for a particular TV show/movie. Additional information like plot, release date, genres are also considered important from a user's point of view.

Since it is considerably difficult to maintain a database of all the metadata of TV shows and movies, a practical solution to address this issue would be to request information from an already established, reliable source like IMDB, TV Rage, Trakt etc. These movie/TV show information gathering web sites maintain open ended REST APIs which provide convenient data gathering of almost all the TV shows/movies currently available in the entertainment industry.

## A. *Evolution and Current Developments in the field of Torrents and P2P File Sharing*

Peer-to-peer file sharing became popular in 1999 with the introduction of Napster [11], a file sharing application and a set of central servers that linked people who had files with those who requested files. This considered as a first generation P2P system due to the presence of central servers. However

with the limitation of sharing only music files and its shutting down in 2001, several other peer-to-peer services like Gnutella [12] and Kazaa [13] became popular, which is considered to be the second generation of P2P file sharing systems. These services also allowed users to download files other than music, such as movies and games.

Over the years, with the introduction of the BitTorrent protocol, which is considered to be a file sharing protocol of the first generation, but differently configured and networked, with the use of torrents, has revolutionized the way people used to share files, especially media files. Then it started growing out of depending on central servers and adopted a decentralized server-independent network. This adaptation has improved over the decade and has obtained its current state, where leading torrent sharing client programs like µTorrent, BitTorrent and Vuze has expand the use of torrent sharing to almost every internet user's personal computer [14]. Latest Developments in P2P files sharing include:

### 1) *Pando*

Pando is a file-sharing application for Windows platforms which allows you to share files with anyone on the Internet. To do so, all what the user need to do is download the client and upload the file or files. Then, the other person will be able to download the file. What is great about this application is that the user can send files directly to the other person's e-mail address. Pando bypasses the attachment file size limit by creating small .pando files and sending them in several different e-mails [15].

### 2) *eMule*

eMule is a free p2p client for PC to PC connection designed to help users to share files located in the incoming folder. Identified as one of the biggest and most reliable peer-to-peer file sharing clients, this program allows users to download and share files following a priority order reached by a credit system. A very interesting and convenient feature of this utility is that the files are checked for corruptions while being downloaded, with the purpose of ensuring the receipt of error free files [16].

It is evident that despite the legal issues associated with sharing files using P2P file sharing techniques, it has the potential to grow into something more than just a set of users sharing files without the intervention of a governing node. With the introduction of applications like TED, torrent users who have the interest in experiencing something additional with torrent sharing and make their life easier, has found hope. Further developments of similar applications are also likely to rise in the years to come with additional functionalities and new technologies.

The approach that is presented in this paper "Media Companion" tries to take what TED did into a new dimension with the involvement of distributed servers and smart mobile devices and provide users with a value added service. It would have its goal set on innovating how people use torrents in day to life and adding extra meaning by expanding its potential to serve user needs and wants.

### III. DESIGN AND IMPLEMENTATION

Media Companion comprises three major components as per Architectural Design Documents. They are;

1. Media Companion Web Server
2. Media Companion Desktop Application (Windows based)
3. Media Companion Android App (Media Companion Mini)

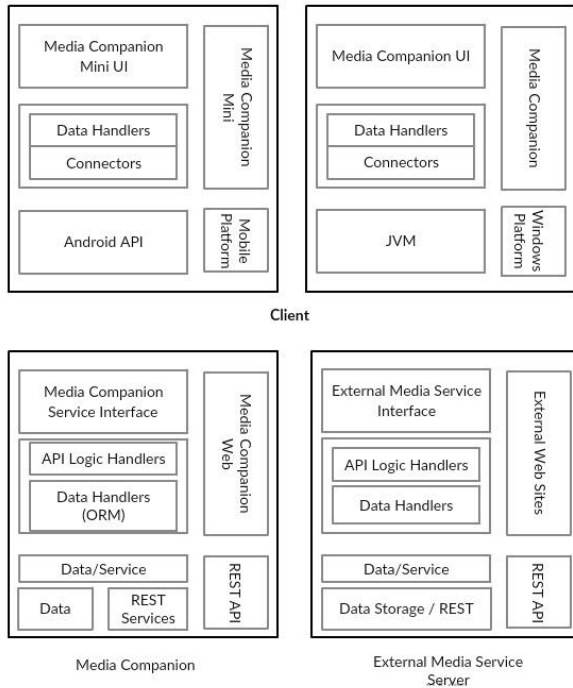


Fig. 1. Media Companion Design Overview

Each of these components possesses inherent design techniques which are unique based on the platform on which they operate and uses a common communication method which brings all the components together to create Media Companion.

#### A. Media Companion Web Server

This is the distributed component of the Media Companion System, where the user would initially register to and select his set of preferred TV series to be tracked by. Information related to different users, including their download history, preferences, Media Companion Settings, available downloads etc. would be maintained by the MySQL data base in the Media Companion Server. These information provide crucial support required by both the desktop and mobile applications.

Media Companion Web server comprise following services which are provided for the user in different stages.

1) *User Registration:* This registers users to the Media Companion system and this is the inception of the system.

2) *User preference identifier:* Once a user is registered with the system, the user will be able to set their preferred TV shows for the system to be tracked and download when available. This service includes providing useful information related to any TV series that the user searches for.

This service is obtained through the use of open ended REST API provided by *trakt.tv* TV show information gathering web site [17]. However, for this API to be used by external developers, it is required to acquire an API key which would be used in all API calls. Using this API, Media Companion is able to display Season information including details of each episode in a season, user ratings (including number of likes and number of dislikes) and other user comments regarding the searched TV series (which is a vital part in TV show download communities).

3) *User profile management:* Users registered with the Media Companion System would have their own profiles which would include their download history (last downloaded TV show and its episode) and his current preferences.

4) *Media Companion desktop/mobile application download center:* This provides the opportunity download both desktop and mobile applications for the user no matter where the user is, as long as they have access to the web server.

#### B. Media Companion Desktop Application (Windows based)

Desktop application would provide the core activity of downloading TV shows preferred by the user. When using this application, user has the ability to automatically download TV shows as well as manually download different episodes from a range of available downloads hosted in <http://kickass.to> at a given instance (polling window can be changed to users liking).

Key features and the used techniques of the desktop application would be as follows

1) *TV shows / Movie search engine (Retrieves basic information regarding the searched TV show):* This functionality is implemented using a freely available open source IMDB API [18]. Unlike the API provided by *trakt.tv*, this API does not require an API key for external accesses, hence can be used by any user without requiring authentication of any sort. This API returns JavaScript Object Notation (JSON) encoded information about the TV show requested by the user.

2) *Automatic Torrent Download Setting:* Once the user sets his preferred list of TV shows, Media Companion would keep track of their daily availability and download when available at <http://kickass.to>. Searching for torrents automatically would continually happen in the background once the desktop application is started, unless the user exits the system forcefully.

3) *Manual Torrent Download Setting:* This functionality provides the user with more autonomy when it comes to downloading TV show related downloads. It could be a single episode, latest or earlier releases, or it could also be an entire Season pack. This eliminates the amount of trouble the user has to go through if the user is to search for these downloads from a torrent sharing web site and then download them using torrent client installed in the user's computer.

For both automatic and manual downloads, Media Companion uses an Open source Java Based BitTorrent library

[9] which customized to suit the requirements of Media Companion.

### C. Media Companion Android App (Media Companion Mini)

This application provides following functionalities to the user.

1) *Display currently available downloads for the user:* Media Companion Mini would retrieve download information from Media Companion Web server and provide a list of downloads that are available for the user to be downloaded.

2) *Ability to remotely initiate desired download from the list of available downloads:* From the list provided by the application, based on currently available downloads of user's preference, the user has the ability to select one or many downloads to be downloaded remotely. This would trigger a download in the Media Companion Desktop application. For this to take effect, user has to keep Media Companion Desktop Application running.

### D. Core Libraries Used

When developing Media Companion, two key libraries are used which are listed below.

1) *Torrent (Java implementation of the BitTorrent protocol):* Ttorrent [9] is a pure-Java implementation of the BitTorrent protocol, providing a BitTorrent tracker, a BitTorrent client and the related Torrent metainfo files creation and parsing capabilities. It is designed to be embedded into larger applications, but its components can also be used as standalone programs. This tool suite was implemented as part of Turn's [19] release distribution and deployment system and is used to distribute new build tarballs to a large number of machines inside a datacenter as efficiently as possible

This implementation aims at providing a down-to-earth, simple to use library. Without the implementation of fancy protocols: just the basics that allows for the exchange and distribution of files through the BitTorrent protocol. Although the write performance of the BitTorrent client is currently quite poor (~10MB/sec/connected peer), it has been measured that the distribution of a 150MB file to thousands of machines across several datacenters took no more than 30 seconds, with very little network overhead for the initial seeder [20].

2) *Apache HttpComponents:* The Apache HttpComponents™ [21] project is responsible for creating and maintaining a toolset of low level Java components focused on Hyper Text Transfer Protocol (HTTP) and associated protocols. This project functions under the Apache Software Foundation [22], and is part of a larger community of developers and users.

HTTP is perhaps the most significant protocol used on the Internet today [23]. Web services, network-enabled appliances and the growth of network computing continue to expand the role of the HTTP protocol beyond user-driven web browsers,

while increasing the number of applications that require HTTP support.

Designed for extension while providing robust support for the base HTTP protocol, the HttpComponents may be of interest to anyone building HTTP-aware client and server applications such as web browsers, web spiders, HTTP proxies, web service transport libraries, or systems that leverage or extend the HTTP protocol for distributed communication [21].

### E. Communication Methodology

For a distributed system like the Media Companion, it is important to maintain a unified way of communicating so that applications in different platforms can adhere to a common set of rules and methodologies when communicating with each other. To make this happen, Media Companion introduces its REST API which could be accessed through HTTP protocol without authentication while there is no requirement for an API key.

Media Companion REST API is developed using PHP Slim Micro Framework [24], which provides a convenient way of establishing GET, POST, PUT, UPDATE and DELETE methods.

Both the desktop and mobile applications would access the central database hosted in Media Companion using the API. This provides convenient access to user related information which are used in tracking and managing torrent downloads.

### F. Developing responsive SWING application with Swing Worker

Swing event handling code runs on a special thread known as the event dispatch thread. Most code that invokes Swing methods also runs on this thread. This is necessary because most Swing object methods are not "thread safe": invoking them from multiple threads risks thread interference or memory consistency errors [25].

Due to this design constraint, implementing heavy duty process within Swing components leads to unresponsive Swing components. To overcome this issue and add long-running task to Swing components, SwingWorker [25] can be used. Each task running on a worker thread is represented by an instance of `javax.swing.SwingWorker`. `SwingWorker` itself is an abstract class; where the user must define a subclass in order to create a `SwingWorker` object; anonymous inner classes are often useful for creating very simple `SwingWorker` objects.

## IV. CONCLUSION

Considering Media Companion's growth potential and its usability for almost anyone who possess the know-how to use the internet for their daily work, it is clear that Media Companion should be torrent lover's favorite application. The amount of contribution that the Media Companion could pass onto media related applications is enormous. Hence, Media Companion would not just be an application that would sit on a user's PC, it would be someone's inspiration to develop

something more than the Media Companion itself and support its consumers to enjoy something more and new.

#### REFERENCES

- [1] Wikipedia. "Bittorrent", 2014. [Online]. Available: <https://en.wikipedia.org/wiki/BitTorrent>. [Accessed: 13 Sep 2014].
- [2] howstuffworks.com. "Bittorrent", 2014. [Online]. Available: <http://computer.howstuffworks.com/bittorrent.htm>. [Accessed: 13 Sep 2014].
- [3] webharvy.com. "Web Scraping", 2014. [Online]. Available: <https://www.webharvy.com/articles/what-is-web-scraping.html>. [Accessed: 13 Sep 2014].
- [4] Wikipedia. "Polling", 2014. [Online]. Available: [https://en.wikipedia.org/wiki/Polling\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Polling_(computer_science)). [Accessed: 13 Sep 2014].
- [5] Wikipedia. "Event Driven Programming", 2014. [Online]. Available: [https://en.wikipedia.org/wiki/Event-driven\\_programming](https://en.wikipedia.org/wiki/Event-driven_programming). [Accessed: 13 Sep 2014].
- [6] Vuze. "Vuze", 2014. [Online]. Available: <http://www.vuze.com/>. [Accessed: 13 Sep 2014].
- [7] bittorrent.com. "BitTorrent", 2014. [Online]. Available: <http://www.bittorrent.com/>. [Accessed: 13 Sep 2014].
- [8] utorrent.com. "µTorrent", 2014. [Online]. Available: <http://www.utorrent.com/>. [Accessed: 13 Sep 2014].
- [9] GitHub. "Ttorrent", 2014. [Online]. Available: <http://mpetazzoni.github.io/ttorrent/>. [Accessed: 13 Sep 2014].
- [10] deluge-torrent.org. "Deluge", 2014. [Online]. Available: <http://dev.deluge-torrent.org/wiki/About>. [Accessed: 13 Sep 2014].
- [11] Wikipedia. "Napster", 2014. [Online]. Available: <https://en.wikipedia.org/wiki/Napster>. [Accessed: 13 Sep 2014].
- [12] Wikipedia. "Gnutella", 2014. [Online]. Available: <https://en.wikipedia.org/wiki/Gnutella>. [Accessed: 13 Sep 2014].
- [13] Wikipedia. "Kazaa", 2014. [Online]. Available: <https://en.wikipedia.org/wiki/Kazaa>. [Accessed: 13 Sep 2014].
- [14] Wikipedia. "Peer-to-Peer File Sharing", 2014. [Online]. Available: [http://en.wikipedia.org/wiki/Peer-to-peer\\_file\\_sharing](http://en.wikipedia.org/wiki/Peer-to-peer_file_sharing). [Accessed: 13 Sep 2014].
- [15] Pando Software. "Pando", 2014. [Online]. Available: <http://pando.software.informer.com/>. [Accessed: 13 Sep 2014].
- [16] eMule Software. "eMule", 2014. [Online]. Available: <http://emule.software.informer.com/>. [Accessed: 13 Sep 2014].
- [17] trakt.tv.com. "trakt.tv", 2014. [Online]. Available: <http://trakt.tv/api-docs>. [Accessed: 13 Sep 2014].
- [18] imdbapi.com. "IMDB API", 2014. [Online]. Available: <http://www.imdbapi.com>. [Accessed: 13 Sep 2014].
- [19] turn.com. "Turn", 2014. [Online]. Available: <http://www.turn.com>. [Accessed: 13 Sep 2014].
- [20] GitHub. "Torrent Java Library", 2014. [Online]. Available: <https://github.com/mpetazzoni/torrent>. [Accessed: 13 Sep 2014].
- [21] apache.org. "HTTP Components", 2014. [Online]. Available: <http://hc.apache.org/>. [Accessed: 13 Sep 2014].
- [22] apache.org. "Apache", 2014. [Online]. Available: <http://www.apache.org>. [Accessed: 13 Sep 2014].
- [23] www8.org. "Key Differences Between HTTP/1.0 and HTTP/1.1.", 2014. [Online]. Available: <http://www8.org/w8-papers/5c-protocols/key/key.html>. [Accessed: 13 Sep 2014].
- [24] slimframework.com. "Slim Framework", 2014. [Online]. Available: <http://www.slimframework.com/>. [Accessed: 13 Sep 2014].
- [25] Oracle. "Java SwingWorker", 2014. [Online]. Available: <http://docs.oracle.com/javase/tutorial/uiswing/concurrency/dispatch.html>. [Accessed: 13 Sep 2014].