TODO TITLE

Ardhi Putra Pratama Hartono

Delft University of Technology

TODO TITLE

Master's Thesis in Computer Science

Parallel and Distributed Systems group
Faculty of Electrical Engineering, Mathematics, and Computer Science
Delft University of Technology

Ardhi Putra Pratama Hartono

September 1, 2016

Author
Ardhi Putra Pratama Hartono

Title
TODO TITLE

MSc presentation
TODO GRADUATION DATE

Graduation Committee
TODO GRADUATION COMMITTEE Delft University of Technology

Abstract

TODO ABSTRACT

Preface

TODO MOTIVATION FOR RESEARCH TOPIC

TODO ACKNOWLEDGEMENTS

TODO AUTHOR

Delft, The Netherlands
September 1, 2016

Contents

Chapter 1

Introduction

The introduction

1.1 Document Structure

This thesis is structured as follows. Chapter 2 provides problem description and answer to tackle this issue. Chapter 3 discusses related work to the problem and solution proposed. Chapter 4 presents the design of credit mining system integrated with Tribler. Implementation of the system and it's experiment will be elaborated in chapter 5. Chapter 6 shows result of the performance of credit mining system. At the end, chapter 7 concludes the work mentioning possible future work.

Chapter 2

Problem Description

Credit (share ratio or upload rate) is needed especially in private tracker. In this case, BitTorrent tit-for-tat is irrelevant [10]

2.1 Peer-to-peer networks

2.2 Economics in distributed system

Demand and supply [2]

2.3 Optimizing network cost

Anonymous Relaying performance in Tribler [14]
Significant portion when seeding million torrents [11]

Chapter 3

Related Work

3.1 BitTorrent

BitTorrent [7] is a file distributed system on top of peer-to-peer network. Static `.torrent` file, which contains information such as tracker addresses and unique hash value of this swarm, is created by peer who want to publish their files. *Tracker* responsible for monitors the distribution and progress of file and peers in a swarm. Peer uses information in `.torrent` file to connect each other.

In BitTorrent, it is desirable to have many peers upload piece of file to the swarm. This way, swarm can be *healthier* and overall download speed can increase. However, many peers become a *leecher*, which quits the swarm when his download finished. BitTorrent uses *tit-for-tat* mechanism to reward good behavior and punish bad behavior. To force cooperation of other peer, BitTorrent implement *choking algorithm*. Choking algorithm is an algorithm to temporarily refuse uploading piece of file to a particular peer. Usually, an uploader has limited number of unchoked slots. By observing other peers, choking algorithm decide which peer a particular piece will be sent or not sent to. If we unchoke a peer, it means we consider to upload a piece to that peer. For starters, it is usually useful to execute *optimistic unchoking*. Optimistic unchoking is an algorithm to unchoke a peer regardless of its activity in a swarm. This gives a peer chance to increase his upload rate by providing more content.

cite cheating peers/freeloader/hit-n-run behavior

cite choke/unchoke algorithm

3.1.1 Peer Discovery

One of the integral part in BitTorrent protocol is peer discovery. With a large number of known peers, the algorithm will have more option on which peer to unchoke. State of the swarm itself often represented by the peer belong to that swarm. In BitTorrent, there are four method to discover new or update peer.

LSD? (bep_0014.html)

Tracker Peer Announce

In original design of BitTorrent, it uses tracker to allow peer discover each other [7]. Tracker tend to use random and limited list of peers. Peer contact tracker periodically to expand their peer dictionary. This act of requesting peer to tracker is called *announce*. Usually, most tracker has a policy about recommended interval when to recontact for getting new peers. Violate this policy can result a particular peer blocked.

Distributed Hash Table (DHT)

DHT performance?

Originally, peer need to contact tracker to fetch new peer address and file information. This makes BitTorrent very dependent on centralized system which vulnerable to single point of failure. In 2008, Distributed Hash Table (DHT) is proposed [9]. Towards a "trackerless" BitTorrent system, DHT allow each peer to become a tracker. DHT stores peer contact information with defined key-space as "node ID". Each peer stored other peer's node ID and its address in their own routing table. A "distance" is measured on two node ID to define how close those two. "Distance" also can be measured between infohash of a torrent and node ID.

To enrich its peer dictionary, a node can compare a torrent's infohash and node ID in its routing table. If the distance under the threshold, it contacts that node to ask the information of the swarm, which includes the peer list. If contacted node do not know this torrent, it will respond with another node in its table which closest to the provided infohash.

Peer Exchange (PEX)

To increase the chance of getting higher downloading speed, having up to date peer is desired. This can be achieved by contacting tracker or using DHT. Reducing the interval of contacting tracker can result in getting a number of updated peer sooner, however, it will put a burden in the tracker itself. Peer Exchange (PEX)[1] is proposed to tackle this problem. PEX used list of peers that bootstrapped from another mechanism. This mechanism allow contact known peer directly to get and give up to date information on swarm. Theoretically, it can keep this swarm together if trackers are down. Specification mentioned in [1] stated a restriction such as number of request per minute and number of peer added or removed in a PEX message.

3.2 Tribler

Tribler[1] is peer-to-peer file sharing application that compatible with BitTorrent protocol [13]. Tribler focused on security, fully decentralized system, and anonymity. Starts with ABC (Another BitTorrent Client), Tribler currently provides content

[1] https://www.tribler.org/

Table 3.1: Overview of implemented Dispersy community in Tribler [8].

Community Name	Purpose
AllChannel	Used to discover new channels and to perform remote channel search operations.
BarterCast4	While currently disabled, this community was used to spread statistics about the upload and download rates of peers inside the network and has originally been created as a mechanism to prevent free-riding in Tribler.
Channel	This community represents a single channel and is responsible for managing torrents and playlists inside that channel.
Multichain	This community utilizes the blockchain technology and can be regarded as the accounting mechanism that keeps track of shared and used bandwidth.
Search	This community contains functionalities to perform remote keyword searches for torrents and torrent collecting operations.
(Hidden)Tunnel	This community contains the implementation of the Tor-like protocol that enables anonymity when downloading content and contains the foundations of the hidden seeder services protocol, used for anonymous seeding.

discovery, channels concept, and reputation management in fully distributed manner.

What information to include?
-anonymous
-end2end encryption

3.2.1 Dispersy Community

See table 3.1

Various community

Relation on peers

3.3 Libtorrent

3.3.1 Share Mode

3.4 Credit Mining Framework

Credit mining base [6].
Investment strategy in regression model [3].
Use libtorrent as helper. Multiple helper and its effect to swarm with actual downloading on the other side [5].

Inter swarm resource allocation [4].

3.5 P2P Currency

Current currency used in Tribler (Multichain)[12].
Demand and supply in bittorrent environment [2].

Chapter 4

System Design

How to measure swarm by looking at Swarm evolution [15]. Helper in swarm without need of human intervention [5]. Aligning supply and demand of bandwidth

Chapter 5

Implementation and Experiment

Chapter 6

Performance Evaluation

Chapter 7

Conclusions and Future Work

7.1 Conclusions

TODO CONCLUSIONS

7.2 Future Work

TODO FUTURE WORK

Bibliography

[1] The 8472. Peer exchange (pex). *BitTorrent.org http://www.bittorrent.org/beps/bep_0011.html. Accessed: 22 August 2016*, 2015.

[2] Nazareno Andrade, Elizeu Santos-Neto, Francisco Brasileiro, and Matei Ripeanu. Resource demand and supply in BitTorrent content-sharing communities. *Computer Networks*, 53(4):515–527, mar 2009.

[3] Mihai Capotă, Nazareno Andrade, J. A. Pouwelse, and D. H J Epema. Investment Strategies for Credit-Based P2P Communities. In *2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, pages 437–443. IEEE, feb 2013.

[4] Mihai Capotă, Nazareno Andrade, Tamás Vinkó, Flávio Santos, J. A. Pouwelse, and D. H J Epema. Inter-swarm resource allocation in BitTorrent communities. In *2011 IEEE International Conference on Peer-to-Peer Computing*, pages 300–309. IEEE, aug 2011.

[5] Mihai Capotă, J. A. Pouwelse, and D. H J Epema. Towards a peer-to-peer bandwidth marketplace. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8314 LNCS:302–316, 2014.

[6] Mihai Capotă, J. A. Pouwelse, and D. H J Epema. Decentralized credit mining in P2P systems. *Proceedings of 2015 14th IFIP Networking Conference, IFIP Networking 2015*, 2015.

[7] Bram Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72, 2003.

[8] M.A de Vos. *Identifying and Managing Technical Debt in Complex Distributed Systems*. Master thesis, Delft University of Technology, 2016.

[9] Andrew Loewenstern and Arvid Norberg. Dht protocol. *BitTorrent.org. http://www.bittorrent.org/beps/bep_0005.html. Accessed: 22 August 2016*, 2008.

[10] M. Meulpolder, L D'Acunto, and Mihai Capotă. Public and private BitTorrent communities: a measurement study. *Iptps*, page 10, 2010.

[11] Arvid Norberg. Seeding a million torrents, 2012.

[12] Steffan D Norberhuis. *MultiChain: A cybercurrency for cooperation*. Master thesis, Delft University of Technology, 2015.

[13] J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H J Epema, M. Reinders, M. R. Van Steen, and H. J. Sips. TRIBLER: A social-based peer-to-peer system. *Concurrency Computation Practice and Experience*, 20(2):127–138, 2008.

[14] Quinten Stokkink, Harmjan Treep, and Johan Pouwelse. Performance analysis of a Tor-like onion routing implementation. jul 2015.

[15] Majing Su, Hongli Zhang, Binxing Fang, and Lin Ye. A Measurement Study on Resource Popularity and Swarm Evolution of BitTorrent System. *International Journal of Communications, Network and System Sciences*, 06(06):300–308, 2013.