Reading Report: Cohen03

Albert Bausili Fernández

May 8, 2023

Upload your report in PDF format.
Use this LaTeX template to format the report, keeping the proposed headers.
The length of the report must not exceed **5 pages**.

1 Content

1.1 Identify the genre[1] of the document, its purpose, and its target audience.

It is an Article. It's purpose is to spread information about the function of the peer-to-peer systems, specifically the BitTorrent one, and its target audience are the scholars, Computer scientists or enthusiasts that want to dive deeper in this topic.

1.2 Summarize the document, indicating the key concepts[2].

Bittorrent redestributes the cost of upload to downloaders instead of placing it in the hosting machine (HTTP). Bittorrent has a problem of fairness, as the download rate must make happy all the downloaders, so it has sense to make the download rate be proportional to their upload rate.

The bandwidth requirements of the tracker and web server are very low, while the seed must send out at least one complete copy of the original file.

In order to keep track of which peers have what, BitTorrent cuts files into pieces of fixed size, typically a quarter megabyte. Each downloader reports to all of its peers what pieces it has. To verify data integrity, the SHA1 hashes of all the pieces are included in the .torrent file, and peers don't report that they have a piece until they've checked the hash. Erasure codes have been suggested as a technique which might help with file distribution, but this much simpler approach has proven to be workable.

BitTorrent facilitates this by breaking pieces further into sub-pieces over the wire, typically sixteen kilobytes in size, and always keeping some number, typically five, requests pipelined at once. Every time a sub-piece arrives a new

[1] Genres: book, article, essay, report, review, manual, white paper, data sheet, weblog, etc.
[2] The summary should help you to answer the questions about the reading in the exam.

request is sent.

Piece Selection Algorithms:
Strict Priority:
BitTorrent's first policy for piece selection is that once a single sub-piece has been requested, the remaining sub-pieces from that particular piece are requested before sub-pieces from any other piece. This does a good job of getting complete pieces as quickly as possible.

Rarest First:
When selecting which piece to start downloading next, peers generally download pieces which the fewest of their own peers have first, a technique we refer to as 'rarest first'. By replicating the rarest pieces as quickly as possible thus reducing the risk of them getting completely lost as current peers stop uploading.

Random First Piece:
An exception to rarest first is when downloading starts. At that time, the peer has nothing to upload, so it's important to get a complete piece as quickly as possible. For this reason, pieces to download are selected at random until the first complete piece is assembled, and then the strategy changes to rarest first.

Endgame Mode:
Sometimes a piece will be requested from a peer with very slow transfer rates. This isn't a problem in the middle of a download, but could potentially delay a download's finish. To keep that from happening, once all sub-pieces which a peer doesn't have are actively being requested it sends requests for all sub-pieces to all peers. Cancels are sent for sub-pieces which arrive to keep too much bandwidth from being wasted on redundant sends. In practice not much bandwidth is wasted this way, since the endgame period is very short, and the end of a file is always downloaded quickly.

Choking Algorithms:
Each peer is responsible for attempting to maximize its own download rate. Peers do this by downloading from whoever they can and deciding which peers to upload to via a variant of tit-for-tat. To cooperate, peers upload, and to not cooperate they 'choke' peers. Choking is a temporary refusal to upload; It stops uploading, but downloading can still happen and the connection doesn't need to be renegotiated when choking stops.

Pareto Efficiency:
In computer science terms, seeking pareto efficiency is a local optimization algorithm in which pairs of counter-parties see if they can improve their lot together, and such algorithms tend to lead to global optima. Specifically, if two peers are both getting poor reciprocation for some of the upload they are providing, they can often start uploading to each other instead and both get a better download rate than they had before. Peers reciprocate uploading to peers which upload to them, with the goal of at any time of having several connections which are actively transferring in both directions. Unutilized connections are also uploaded to on a trial basis to see if better transfer rates could be found using them.

BitTorrent's Choking Algorithm:
On a technical level, each BitTorrent peer always unchokes a fixed number of other peers (default is four), so the issue becomes which peers to unchoke. Decisions as to which peers to unchoke are based strictly on current download rate. To avoid situations in which resources are wasted by rapidly choking and unchoking peers, BitTorrent peers recalculate who they want to choke once every ten seconds, and then leave the situation as is until the next ten second period is up.

Optimistic Uncoking:
Simply uploading to the peers which provide the best download rate would suffer from having no method of discovering if currently unused connections are better than the ones being used. To fix this, at all times a BitTorrent peer has a single 'optimistic unchoke', which is unchoked regardless of the current download rate from it. Which peer is the optimistic unchoke is rotated every third rechoke period (30 seconds).

Anti-snubbing:
Occasionally a BitTorrent peer will be choked by all peers which it was formerly downloading from. In such cases it will usually continue to get poor download rates until the optimistic unchoke finds better peers. To mitigate this problem, when over a minute goes by without getting a single piece from a particular peer, BitTorrent assumes it is 'snubbed' by that peer and doesn't upload to it except as an optimistic unchoke. This frequently results in more than one concurrent optimistic unchoke, (an exception to the exactly one optimistic unchoke rule mentioned above), which causes download rates to recover much more quickly when they falter.

Upload only:
Once a peer is done downloading, it no longer has useful download rates to decide which peers to up- load to. The current implementation then switches to preferring peers which it has better upload rates to, which does a decent job of utilizing all available upload capacity and preferring peers which no-one else happens to be uploading to at the moment.

2 Assessment

2.1 Rate the readability of the document: easy, readable, difficult, unreadable.

Easy

2.2 Give your opinion of the reading assignment, indicating whether it should be included in next year's course or not.

I found it specially interesting, as I've been a BitTorrent user for a long time and I've always been interested in knowing how it works, so I really think it should be included in next year course.