Reading Report: Cohen03

Ricard Abril

May 5, 2019

1 Summary

Aquest article, explica com funciona el sistema BitTorrent, i com aquest optimitza el us de recursos i la seva robustesa millor que cap altra tècnica coneguda. L'article es divideix en les seccions següents:

1.1 What BitTorrent does.

Amb els sistemes per descàrrega d'arxius tradicionals com HTTPS, el Host, és el que suporta tot el cost de la descàrrega de l'arxiu, en canvi amb el model proposat per BitTorrent, quan diverses persones intenten descarregar un mateix arxiu, a la vegada també serveixen aquests arxius de forma que es distribueix l'esforç necessari, i es fa factible fer hosting d'un fitxer que es descarregara un nombre infinit de vegades.

1.1.1 BitTorrent interface.

La interfície que utilitza BitTorrent, és molt senzilla, consta d'una barra de progrés, la qual va acompanyada d'una tassa de descàrrega i una tassa de descàrrega, que ens indica quina quantitat de dades hem descarregat i quina quantitat hem servit.
Per afegir un nou arxiu a descarregar, únicament hem d'obrir un fitxer .torrent i afegir-lo a la llista de descàrregues.
Aquesta interfície simple, i la facilitat d'us que comporta han ajudat en gran mesura a la popularitat de BitTorrent.

1.1.2 Deployment.

El creador d'un arxiu, és el que pren la decisió d'utilitzar BitTorrent.
Els clients, que són els que desitgen descarregar l'arxiu, utilitzen BitTorrent, ja que l'arxiu està disponible mitjançant aquest sistema.
Típicament els clients, deixen de servir l'arxiu immediatament després que la seva descarrega es completi.
El nombre de clients que volen descarregar l'arxiu, augmentarà rapidament, després que aquest estigui disponible, després com que els usuaris acaben la descàrrega cau exponencialment.

1.2 Technical Framework.

1.2.1 Publishing Content.

Per a publicar contingut utilitzant el BitTorrent, el que es fa és penjar en un servidor web un arxiu amb extensió .torrent, aquest arxiu conte informació sobre l'arxiu que ens disposem a descarregar i l'enllaç a un rastrejador.
Els rastrejadors són responsables d'ajudar als clients a trobar-se els uns als altres. Aquests, s'envien informació també, com per exemple de quins arxius disposen, en quin port estan escoltant, ...
Els rastrejadors responen amb informació de contacte per a altres peers que descarregaran el mateix arxiu. Els clients usen aquesta informació per connectar-se entre ells.
Perquè un arxiu estigui disponible, una seed s'ha d'iniciar. Els requisits d'amplada de banda del rastrejador i el servidor web són baixos, mentre que la seed ha d'enviar mínim una còpia completa de l'arxiu original.

1.2.2 Peer Distribution.

Tots els problemes logístics de la descàrrega d'arxius es manegen en les interaccions entre parells. Informació sobre les taxes de càrrega i descàrrega s'envia al rastrejador per a la recopilació d'estadístiques.
La única responsabilitat dels rastrejadors es limita a ajudar als seus peer's a trobar-se entre ells, encara que els rastrejadors són l'única forma perquè els clients es trobin entre si, l'algorisme de seguiment, retorna una llista aleatòria de peers.
Per tal de realitzar un seguiment de què té cada peer, BitTorrent talla els arxius en parts de mida fixa (un quart de MB). Cada client reporta a tots els seus companys que peces té. Per verificar la integritat de les dades, s'inclouen els hashes SHA1 de totes les peces a l'arxiu .torrent, i els companys no informen de tenir una peça fins que hagin comprovat el hash. Els peers descarreguen contínuament peces de tots els companys el que poden. No poden descarregar de peers amb els que no estan connectats. Pot ser que els peers no tinguin les peces que volen o no les deixen descarregar.

1.2.3 Pipelining.

BitTorrent, transfereix dades a través de TCP, per tant és molt important tenir sempre diverses sol·licituds pendents a la vegada. Per tal d'evitar un retard entre peces enviades.
Això ho aconseguim trencant peces en sub-peçes de 16 KB, sempre mantenint algun nombre de sol·licituds canalitzades al mateix tema. Cada vegada que una sub-peça arriba, una nova sol·licitud es enviada. La quantitat de dades que s'envia, pot arribar a saturar una connexió.

1.2.4 Piece Selection.

Un bon ordre de selecció de peces per descarregares és important per a un bon rendiment. Un algoritme de selecció de peces pobre pot resultar en tenir totes les peces que s'ofereixen, o en no tenir cap peça per pujar als altres peers.

Strict Priority:
Una vegada que s'ha sol·licitat una única peça, les sub-peces restants d'aquesta peça es sol·liciten abans de les sub-peces de qualsevol altra peça. Així s'obtenen peces completes el més ràpid possible.

Rarest First:
Al seleccionar quina peça començar a baixar, els peers descarreguen les peces que pocs dels altres peers tenen primer. Aquesta tècnica s'anomena Rarest First, i assegura que els companys tinguin peces que tots els seus companys vulguin, de manera que es pot fer la càrrega quan es vol. També s'assegura que les peces més comunes queden per a més endavant, de manera que es reduirà la probabilitat que un peer que està oferint la pujada no tindrà interès més tard.
Per a desplegaments amb una sola seed, la capacitat de càrrega de la qual és menys que la de molts descarregadors, el rendiment és molt millor, si diferents descarregadors obtenen diferents peces de la seed, ja que les descàrregues redundants malgasten l'oportunitat de la seed per obtenir més informació. Rarest First només descarrega noves peces de la seed, ja que els descarregadors podran veure que els altres peers tenen peces que la seed ja ha carregat.

Random First Piece:
Una excepció a Rarest First és quan s'inicia la descàrrega. El peer no té res per pujar, i és important obtenir una peça completa el més ràpid possible. Les peces rares en general només es presenten en un peer, i es baixaran més lentament que les peces presents en múltiples peers, per a les quals és possible descarregar sub-peces de diferents llocs. Per aquest motiu, les peces per descarregar es seleccionen random fins que es completa la primera peça, i llavors l'estratègia canviarà a Rarest First.

Endgame Mode:
De vegades, es demanarà una peça a un peer amb taxes de transferència molt lentes. Això no és un problema enmig de la descàrrega, però en podria retardar l'acabament. Per evitar que això passi, una vegada que totes les sub-peces que un peer no té es sol·liciten activament, s'envien sol·licituds per a totes les sub-peces a tots els peers.
S'envien les cancel·lacions per a les subpeces que arriben per evitar que s'està perdent massa amplada de banda en enviaments redundants. A la pràctica, no es malgasta gaire amplada de banda d'aquesta manera, ja que el període d'endgame és molt curt i el final d'un fitxer sempre es descarrega ràpidament.

1.3 Choking Algorithms

BitTorrent no assigna recursos centrals. Cada peer és responsable d'intentar maximitzar la seva pròpia taxa de descàrrega. Els peers ho fan descarregant des de qualsevol que pugui i decidint quins peers pengen. Per cooperar, els peers pujen, i per no cooperar, ofeguen peers. L'asfíxia és el contrari de pujar: es deixa de pujar, però es pot descarregar i la connexió no ha de ser renegociada quan s'acaba l'asfíxia.
L'algorisme d'asfíxia no és part del protocol de BitTorrent, però és necessari per a un bon rendiment. Un bon algoritme d'asfíxia hauria d'utilitzar tots els recursos disponibles, proporcionar taxes de descàrrega coherents per a tothom i ser una mica resistent als peers que només descarreguen i no pugen.

1.3.1 Pareto Efficiency.

La cerca d'eficàcia de Pareto és un algorisme d'optimització local en el qual peers de contrapartides veuen si poden millorar el seu lot junts, i aquests algorismes tendeixen a conduir a una optimització global. Si els dos peers reben una reciprocitat pobra per a algunes de les càrregues que estan proporcionant, sovint poden començar a pujar-se les unes a les altres, i els dos obtenen una millor taxa de descàrrega del que tenien abans.
Els algorismes d'asfíxia de BitTorrent intenten aconseguir l'eficiència de Pareto utilitzant una versió més elaborada. Els peers pugen de forma recíproca als peers que els hi pugen, amb l'objectiu de tenir sempre diverses connexions que es transfereixen en les dues direccions. Les connexions no utilitzades també es carreguen a prova per veure si es poden trobar millors taxes de transferència utilitzant-les.

1.3.2 BitTorrent's Choking Algorithm.

Cada peer de BitTorrent sempre no ofega un nombre fix d'altres companys (4 per defecte), de manera que el problema esdevé quins peers han de no ofegar. Aquest enfocament permet controlar la congestió integrada de TCP per saturar de manera fiable la capacitat de pujada.
Les decisions sobre quins peers s'han de no ofegar es basen exclusivament en la taxa de descàrrega actual. El càlcul de la taxa de descàrrega actual és significativament difícil. La implementació actual utilitza essencialment una mitjana de 20 segons.
Per evitar situacions en què es malgasten recursos per asfixiar i no ofegar ràpidament els peers, els peers de BitTorrent calculen qui volen ofegar-se una vegada cada 10 segons, i després surten de la situació tal com fins al pròxim període de 10 segons. 10 segons és un perıode de temps prou llarg perquè TCP pugui augmentar les seves transferències fins a la seva plena capacitat.

1.3.3 Optimistic Unchoking.

Simplement pujar als peers que proporcionen la millor tarifa de descàrregues no tindria cap mètode per descobrir si les connexions actuals no utilitzades són millors que les que s'utilitzen. Per solucionar-ho, en tot moment que un peer de BitTorrent té un optimistic unchoke, que no s'ofega independentment de la taxa de descàrregues actual. La selecció de qui es el optimistic unchoke es rota cada 30 segons. 30 segons és suficient per a que la pujada aconsegueixi la màxima capacitat, i perquè la descàrrega sigui reciproca i arribi a la màxima capacitat.

1.3.4 Anti-snubbing.

De vegades, un peer de BitTorrent serà ofegat per tots els seus companys dels quals anteriorment havia descarregat. En aquests casos, normalment continuarà obtenint baixes taxes de descàrrega fins que l'optimistic unchoke trobi millors parells. Per mitigar aquest problema, quan passa més d'un minut sense obtenir una sola peça d'un peer, BitTorrent assumeix que és denegat per aquest i no hi puja, excepte en un optimistic unchoke. Amb freqüència, es produeixen concurrentment més d'un optimistic unchoke. Això fa que les taxes de descàrrega es recuperin molt més ràpidament quan es debiliten.

1.3.5 Upload Only.

Una vegada que es fa la descàrrega d'un parell, ja no té taxes de descàrrega útils per decidir a quins companys ha de penjar. L'aplicació actual canvia perquè prefereixin peers amb millors tarifes de càrrega, que fan un treball en utilitzar tota la capacitat de càrrega disponible i prefereixen peers als quals ningú més no pugui pujar en aquest moment.

1.4 Real World Experience.

BitTorrent no només està ja implementat, sinó que ja està àmpliament implementat. Serveix de forma rutinària arxius de centenars de MB i serveix centenars de descàrregues concurrents. Els desplegaments més grans coneguts han tingut més de mil descarregadors simultanis. El coll d'ampolla) sembla ser l'amplada de banda del seguidor. Actualment, es tracta d'una mil·lèsima part de l'amplada de banda que s'utilitza, i algunes extensions menors de protocol probablement és reduirà unes 10000 vegades.

2 Assessment

El text ha estat interessant, gaire tothom avui en dia utilitza BitTorrent, ja que és un dels sistemes més eficaços per compartir arxius, i per tant desperta cert interès saber com funciona. Per la qual cosa recomanaria la lectura per al curs següent.