

Общая информация

Одноранговая, децентрализованная или пиринговая (англ. peer-to-peer, P2P — равный к равному) сеть — это оверлейная компьютерная сеть, основанная на равноправии участников. Часто в такой сети отсутствуют выделенные серверы, а каждый узел (peer) является как клиентом, так и выполняет функции сервера. В отличие от архитектуры клиент-сервера, такая организация позволяет сохранять работоспособность сети при любом количестве и любом сочетании доступных узлов.

В сети присутствует некоторое количество машин, при этом каждая может связаться с любой из других. Каждая из этих машин может посылать запросы другим машинам на предоставление каких-либо ресурсов в пределах этой сети и, таким образом, выступать в роли клиента. Будучи сервером, каждая машина должна быть способной обрабатывать запросы от других машин в сети, отсылать то, что было запрошено.

Одним из ключевых приложений данного типа сети заключается в быстрой передаче файлов большого размера. Такой эффект достигается за счет снижения нагрузки на сервер и разбиения файла на фрагменты. Примером работы пиринговой сети для передачи файлов является приложение BitTorrent. BitTorrent является гибридным распределенным приложением, использующим как клиент-серверную архитектуру, так и пиринговую.

Архитектура BitTorrent включает в себя следующие сущности:

- 1) Статичный файл метаданных (торрент-файл)
- 2) Трекер
- 3) Пользователь, скачавший файл целиком и раздающий его (сидер)
- 4) Пользователь, скачивающий файл (личер)

Первым шагом в размещении файла с помощью BitTorrent является создание файла метаданных из файла, который вы хотите опубликовать. Файл метаданных называется торрент. Торрент-файл содержит имя файла, размер, хэш и URL торрента. Торрент необходим всем, кто хочет, чтобы загрузить файл торрент создается из. Торрент-файл может распространяться по электронной почте, IRC, HTTP и т.д. Закачка файлов посредством BitTorrent производится путем открытия торрент-файла в торрент-клиенте.

Трекер ведет лог пиров, которые в настоящее время загружают файлы, и помогает им найти друг друга. Трекер не принимает непосредственного участия в передаче данных и не имеет копии файла. Трекер и пользователи, осуществляющие загрузку, обмениваются информацией, используя простой протокол HTTP. Во-первых, пользователь дает информацию трекеру о том, какой файл он загружает, какие порты он слушает и т.д. Ответ трекера содержит список других пользователей, которые скачивают тот же файл и информацию о том, как связаться с ними. Эта группа пользователей, работающих с одним и тем же файлом, называется «роем».

При создании торрент-файла из исходного файла, исходный файл разрезается на малые части, как правило, 512 КБ или 256 КБ. Хэш значения этих частей (шифрование методом SHA-1) включаются в торрент-файл. Загруженные данные проверяются путем вычисления хэш значений и сравнения их со значениями соответствующей части в торрент-файле. Таким образом, данные проверяются на наличие ошибок и пользователям предоставляется гарантия

того, что файл существует. Всякий раз, когда часть загружается и проверяется, загружающий пир сообщает другим пирам в рое о наличии новой части.

Торрент файл

Торрент файл представляет из себя словарь в Bencode формате с расширением .torrent. Он содержит следующую информацию:

- URL трекера;
- Общую информацию о файлах (имя, длину и пр.) в данной раздаче;
- Контрольные суммы (точнее, хэш-суммы SHA1) сегментов раздаваемых файлов;
- Passkey пользователя, если он зарегистрирован на данном трекере. Длина ключа устанавливается трекером.
- (Необязательно) Хэш-суммы файлов целиком
- (Необязательно) Альтернативные источники, работающие не по протоколу BitTorrent. Наиболее распространена поддержка так называемых web-сидов (протокол HTTP).

Содержимое торрент файла является одним большим ассоциативным массивом со следующими полями:

- **info** — вложенный ассоциативный массив который собственно и описывает файлы, которые передаёт торрент;
- **announce** — URL для трекера. Наряду с info является обязательным полем, всё остальное — опционально;
- **announce-list** — список трекеров, если их несколько. В Bencode-виде — список списков;
- **creation date** — дата создания. UNIX Timestamp;
- **comment** — текстовое описание торрента. rutracker.org хранит здесь ссылку на тему форума;
- **created by** — говорит нам о том, кем создан данный торрент.

Ассоциативный массив **info** состоит из:

- **piece length** — размер одного кусочка — 512 килобайт, 1 метр, и так далее. Слишком большое число кусков будет «раздувать» .torrent-файл.
- **pieces** — строка, которая содержит конкатенацию SHA1-хешей, описывающих каждый кусочек. Длина этой строки равна 20 * количество кусков.
- **name** — рекомендательное имя файла (если файл один) или директории. Увы многие торрент-клиенты воспринимают это как аксиому.
- **length** — если файл один, то будет задано это поле, которое содержит длину файла.
- **files** — если файлов несколько, то появится список ассоциативных массивов.

Формат элементов списка **files**:

- **length** — длина файла.
- **path** — список из строк, которые задают путь. Каждая строка — элемент пути, относительно корневой директории торрента. Для пути «a/b/c/d.jpg» будет 4 строки в данном списке — ['a', 'b', 'c', 'd.jpg'].

Протокол трекера

Трекер играет существенную роль в BitTorrent, потому что без него пиры не смогли бы искать других пиров для загрузки. Трекеры используют простой протокол, базирующийся на HTTP. Трекер получает HTTP GET запросы и отправляет сообщения закодированные при помощи Bencode на запрос пира. GET запросы трекера содержат следующие ключи: *info_hash*, *peer_id*, *ip*, *port*, *uploaded*, *downloaded*, *left*, и *event*.

Ключ *info_hash* позволяет трекеру определять, в какой торрент-сессии находится клиент или к какой присоединяется. Этот ключ содержит 20 байтный хэш в формате SHA 1 для ключа *info* торрент файла.

Peer_id является идентификатором клиента и генерируется случайным образом в начале загрузки. Представляет из себя строку из 20 символов.

Следующий ключ, *ip*, не является обязательным и обычно используется для источника, если он находится на том же компьютере, что и трекер. Ключ *port* соответствует номеру порта, который слушает пир.

Ключи *uploaded*, *downloaded* и *left* соответствуют количеству отправленных байт, загруженных байт и количество оставшихся до окончания загрузки файла байтов. Они используются таким образом, чтобы трекер мог отслеживать статистику. Например, он может отслеживать число сидеров и личеров в торрент-сессии и сколько раз конкретное файл был загружен.

Следующий ключ *event* является необязательным, который может иметь возможные значения *started*, *completed*, *stopped*, или *empty*. *Started* используется, когда только начинается загрузка. *Completed* используется, когда пользователь завершает загрузку. *Stopped* используется, когда пир прекращает загрузку. *Empty* используется во время коммуникации, которую выполняют пиры через равные промежутки времени.

Ответы, которые трекер отправляет пирам, представляют из себя словари в кодировке Bencode. Этот словарь должен содержать два ключа: *interval* и *peers*. Ключ *interval* показывает количество секунд, которое пир должен ждать между запросами. Ключ *peers* содержит список словарей, которые хранят информацию о пирах: *peer id*, *ip*, и *port*. *Peer id* посылается трекеру в запросе от пира. *ip* и *port* соответствуют IP адресу и порту пира. Обычно трекер возвращает 50 случайных пиров в этом ответе на запрос.

Протокол пиров

Пиры общаться друг с другом, посылая сообщения непосредственно друг другу, используя протокол пиров BitTorrent. Этот протокол работает поверх TCP. Для того, чтобы два пира начали процесс обмена сообщениями, они должны сначала соединиться друг с другом, посылая соответствующее стартовое сообщение. Это сообщение начинается со строки «19 BitTorrent Protocol». 19 - длина. После этой строки есть восемь зарезервированных байт, которые в настоящее время не используются, но они добавляются, чтобы протокол был расширяемым. Следующие 20 байт содержат хэш-значение (SHA-1) значения *info* торрент-файла. Это то же значение, которое используется в запросах трекера. Это значение необходимо, поскольку пир может принимать участие во многих задачах торрента и когда пир передает стартовое сообщение, он должен знать, к какому рою (торрент раздаче) он присоединяется. Следующие 20 байт являются идентификатор пира, которое равно тому, что

посылается в запросах трекеру. Это завершает соединение и теперь пиры могут начать посылать другие типы сообщений подключенным пирам.

Для всех соединений пира он должен поддерживать определенную информацию о них. Все соединения находятся в одном из двух состояний: *choked* (заблокированный) или *unchoked* (свободный). Если пир забит, то ему не позволено загружать данные в рамках этого соединения. Все соединения пира должны быть также в одном из состояний: заинтересованное или нет. Пир заинтересован в другом пире, если второй имеет части файла, которых нет у первого. Только тогда, когда пир заинтересован в другом пире и находится в состоянии «свободен», он может выполнять загрузку в рамках соединения. Так как пир должен отправлять сообщения о своей заинтересованности, остальные пиры должны также знать, какие части файла есть у пира. Все соединения начинаются в состоянии «заблокированный» и «не заинтересован». Помимо стартового сообщения, есть 9 других сообщений, которые могут быть отправлены другим пирам:

- choke
- unchoke
- interested
- not interested
- have
- bitfield
- request
- piece
- cancel

Рассмотрим, как происходит обмен сообщениями в рое на примере добавления в сессию пира Peer A.

Сначала Peer A посылает запрос трекеру, чтобы получить список пиров для подключения. Peer A далее иницирует BitTorrent подключения с подмножеством из этого списка путем отправки стартовых сообщений. Обычно инициализируется не более 40 соединений. После того, как пир подключается к другому пире, он ожидает от него сообщение bitfield. Обычно это сообщение посылается только если пир уже загрузил хотя бы один фрагмент файла. Это сообщение является битовым полем, которое соответствует частям файла, которые пир уже скачал. Это необходимо, потому что все пиры должны теперь знать, какие части файла есть у остальных пиров.

Peer A должен поддерживать свой список активных пиров. Этот набор состоит из пиров, которые находятся в свободном состоянии и обычно содержат только четыре пира [1]. Этот набор должен постоянно меняться, чтобы максимизировать скорость загрузки. Это делается с помощью алгоритма блокировки (Choked Algorithm).

Доп. инфо: http://www.bittorrent.org/beps/bep_0003.html

Цель: создать каркас приложения-клиента.

Задачи:

1. Bencode-encoding, bencode-decoding (в сети куча готовых решений) torrent-файла
2. Отправка запроса трекеру
3. Получение ответа сервера и bencode-декодирование его
4. Отправка стартового сообщения пирам
5. Получение bitfield от пиров

Отчетность:

На этапе 1 – вывод инфо торрента в файл (URL трекера, дата создания трекера, размер исходного файла)

На этапе 2 – параметры запроса

На этапе 3 – декодированные параметры ответа

На этапе 4 – отображение стартового сообщения (вывод в файл или контрольное значение переменной)

На этапе 5 – отображение полученного bitfield (вывод в файл или контрольное значение переменной)