Содержание

Опция управления bittorent

BTRT - BitTorrent Proxy Tracker

Описание

BTRT является BitTorrent трекером, который позволяет оптимизировать распределение сетевого трафика в пользу уменьшения объемов трафика на внешние сети за счет увеличения объемов внутрисетевого трафика. Уменьшение объемов сетевого трафика на внешние сети осуществляется за счет приоритезации IP-адресов пиров в ответах на *announce*-запросы: в соответствии с конфигурацией в первую очередь в ответ попадают IP-адреса пиров сети оператора.

BTRT отвечает на стандартные HTTP запросы BitTorrent клиентов *announce* и *scrape*, а также предназначен для обработки нестандартных *forwarded_announce*-запросов, более подробно об этом написано здесь.

Требования к аппаратному обеспечению

Основное требование предъявляется к размеру оперативной памяти. BTRT держит в памяти всю информацию о всех torrent-файлах и пирах, а также пр. Для приблизительного расчета необходимого объема оперативной памяти можно взять за основу 512 байт на один пир.

Требования к программному обеспечению

BTRT работает в среде ОС CentOS 6. Для работы BTRT следующие пакеты должны быть установлены в системе:

- `mysql-server`, версии не ниже 5.1.73
- `mysql-connector-c++`, версии не ниже 1.1.5

Варианты использования

Локальный ретрекинг

Самый простейший способ использования BTRT - это использование в качестве локального ретрекера. В torrent-файлах существует возможность указать несколько трекеров, к которым должен обращаться BitTorrent-клиент, посылая *announce*-запросы. Для организации локального ретрекинга принято использовать специальное имя `retracker.local`. Перенаправляя запросы к `retracker.local` на BTRT обеспечивается локальный ретрекинг.

Использование данных "соседних" ретрекеров

Локальный ретрекинг позволяет перераспределить трафик только внутри сети оператора и обеспечивает уменьшения объемов трафика во внешние сети только за счет клиентов оператора. Часто у оператора существуют договорные отношения с партнерами, которые подразумевают пониженную стоимость трафика в сетях партнеров. Если в сети партнера запущен ретрекер, то использование пиров, зарегистрированных на этом ретрекере позволяет уменьшить объемы трафика во внешних сетях.

BTRT при обработке *announce*-запросов обеспечивает возможность использовать не только данные о локальных пирах, но и запрашивать информацию с ретрекеров, работающих в сетях партнеров. В ответе на *announce*-запрос BitTorrent-клиенту в этом случае посылается IP-адреса пиров зарегистрированных как в сети оператора, так и в сетях партнеров, в результате происходит перераспределение трафика генерируемого BitTorrent-клиентами.

Обработка запросов к сторонним трекерам

BTRT обеспечивает обработку *forwarded_announce*-запросов, генерируемых DPI-платформой. Под *forwarded_announce*-запросами понимаются *announce*-запросы к сторонним BitTorrent-трекерам, которые перехватываются DPI-платформой и перенаправляются на BTRT.

При обработке *forwarded_announce*-запроса BTRT при необходимости обращается к стороннему трекеру за списком пиров, но в ответе BitTorrent-клиенту формируется список пиров в соответствии с приоритетами пиров, заданными в конфигурации (локальные пиры, пиры партнеров, прочие). Таким образом при достаточном количестве пиров в сети оператора и его партнеров, возможно существенно ограничить пропускную способность при работе со сторонними пирами, тем самым перераспределить трафик в сторону увеличения внутрисетевого трафика.

Установка

Для первоначальной установки приложения необходимо выполнить установку пакета и зависимых модулей при помощи команды:

```
sudo yum install btrt
```

BTRT при работе использует БД MySQL. Для выполнения дальнейших действий по установке необходимо запустить сервис **mysqld**:

```
sudo service mysqld start
```

Перед первым запуском BTRT необходимо создать пользователя (или использовать существующего) и схему в БД MySQL. Рекомендуется создавать отдельного пользователя MySQL для работы BTRT. Чтобы сделать это, необходимо подключиться к серверу MySQL с правами суперпользователя:

```
mysql -hlocalhost -uroot -p
```

и выполнить команды создания пользователя и схемы, указав пароль пользователя:

```
CREATE USER 'btrt'@'localhost' IDENTIFIED BY 'password_for_btrt_user';
CREATE DATABASE btrt;
GRANT ALL PRIVILEGES ON btrt.* TO 'btrt'@'localhost';
```

После того, как пользователь будет создан, необходимо завершить привилегированный сеанс MySQL, нажав сочетание клавиш **Ctrl-D**. Создать объекты в схеме данных для BTRT, указав имя и пароль ранее созданного пользователя MySQL:

```
mysql -hlocalhost -ubtrt -p < /etc/btrt/btrt.db_creation.sql
```

Перед первым запуском необходимо создать файл конфигурации /etc/btrt/btrt.conf на основе шаблон конфигурации /etc/btrt/btrt.conf.default. Более подробно о конфигурации написано здесь.

Обновление

Для обновления ранее установленного приложения необходимо в первую очередь остановить сервис трекера:

```
sudo service btrt stop
```

установить обновленный пакет:

```
sudo yum update btrt
```

При обновлении с версии ниже 2.1 необходимо выполнить обновление структуры БД:

```
mysql -hlocalhost -ubtrt -p < /etc/btrt/btrt.db_upgrade_2.1.0.sql
```

Так как при переходе от версии к версии возможны изменения в конфигурации необходимо проверить ее корректность, а после запустить сервис трекера:

```
sudo service btrt start
```

Конфигурация

Основным файлом конфигурации является файл /etc/btrt/btrt.conf. Конфигурация в этом файле задается в формате YAML.

Формат YAML

Базовыми элементами формата YAML являются:

- пара "Ключ-Значение", например:
  `level: "info"`
- последовательность значений, например:
  `- "127.0.0.0/8"`
  `- "192.168.0.0/16"`

На основе базовых элементов возможно создавать сложные структуры данных.

Одним из принципов для описания сложных структур заложенных в формат YAML является правило форматирования вложенных элементов при помощи отступов от начала строки. Элементы лежащие на одном уровне иерархии должны иметь одинаковое количество ведущих пробелов, например, следующий текст:

```
#
# Logging parameters
logging:
    path:           "/var/log/btrt"     # Path to log files, default:
/var/log/btrt
    level:          "info"              # Possible levels are:
                                        #   critical, error, warning, info,
diagnostic, debug
                                        # default logging level is "info"
    switch_time:    "01:00:00"          # Log switching time in format
"HH:MI:SS", default: "01:00:00"
```

можно трактовать как структуру данных `logging`, у которой есть три поля - `path`, `level`, `switch_time`.

Учитывая выше сказанное, главным правилом при редактировании документов в формате YAML является **использование определенного количества пробельных символов для формирования отступов вместо символов табуляции**.

Основной файл конфигурации /etc/btrt/btrt.conf

Основной файл конфигурации содержит описание большинства параметров используемых при работе BTRT. Первоначально при установке пакета создается файл шаблона конфигурации /etc/btrt/btrt.conf.default, который содержит описание всех параметров с комментариями и который может использоваться для создания реальной конфигурации.

Для создания рабочей конфигурации минимально необходимо выполнить модификацию следующих параметров:

Описание подключения к БД MySQL:

```
database:
    url:        "tcp://127.0.0.1:3306"  # URL for connection to MySQL,
default: "tcp://127.0.0.1:3306"
    user:       "root"                  # User name, default: "root"
    password:   "root"                  # Password
```

```
    schema:     "btrt"                  # Schema name, default: "btrt"
```

Описание групп диапазонов IP адресов:

```
    peer_groups:
        # It is useful to have definition for peers in local network.
        # It can be used when peer from the local network is registered on
non-local tracker (neighbor, forwarded tracker)
        -   name:       "local peers"
            priority:   0
            cidr_file:  "/etc/btrt/local_cidrs.conf"
            cidr_list:
                -

        # neighboring peers is used to identifying range of IP-addresses of
peers connected to neighboring trackers
        -   name:       "neighboring peers"
            priority:   1
            cidr_file:  "/etc/btrt/neighboring_cidrs.conf"
            cidr_list:
                -
```

Описание параметров "соседних" трекеров:

```
    neighbors:
        -   name:                   "the nearest retracker"
            refresh_interval:       1800
            peers_expiry_interval:  7200
            announce_request:
"http://retracker.mgts.by/announce?info_hash={info_hash}&peer_id=01234567890
12345678A&port=6882&downloaded=0&uploaded=0&left=4194304&event=started&compa
ct=1&numwant=1000"
```

Управление

Основной исполняемый файл трекера - `/usr/sbin/btrt`.
Для вывода встроенной подсказки необходимо выполнить следующую команду:

```
btrt help
```

Трекер разработан для работы в качестве сервиса. Для старта сервиса необходимо выполнить следующую команду:

```
sudo service btrt start
```

для остановки сервиса:

```
sudo service btrt stop
```

При работе трекера создаются журнальные файлы в соответствии с настройками в конфигурационном файле. По умолчанию журнальные файлы создаются в каталоге `/var/log/btrt`.

Вопросы и решения

- как часто обновляется hashlist.bin?

crontab -l - настроено обновление hashlist раз в 10 минут

- forwarded запросы скат генерит или он только hashlist.bin забирает и все?другими словами как заставить btrt сходить на соседний трекер за пирами.

см.выше пункт - использование соседних ретрекеров
в логе статистики скат появился раздел "Detailed statistics on BitTorrent"
Первая цифра - сколько handshake обработано
2-я - сколько заблокировано в пользу локальных раздач