Протокол для резервного копирования в сетях с задержками и потерями пакетов

Власов Святослав
куратор: Баталов Евгений

Постановка задачи

Архитектура tcp-протокола не позволяет максимально утилизировать LFN-сети.

- LFN - Long Fat Network. Сети с высокой задержкой (RTT) и широким каналом (Bandwidth).
- BDP - Bandwidth * Delay Product, количество данных, которые могут быть одновременно отправлены через tcp-сокет не будучи подтвержденными. Неподтвержденные пакеты необходимо где-то хранить.

Постановка задачи

- Размер буфера в LFN-сетях может потребоваться очень большим. Например, в сети с RTT=250ms и Bandwidth=1GB, BDP=32Mb. Для 100 сокетов нам потребуются гигабайты памяти.
- Windows Scaling (RFC1323) - расширение для протокола tcp, позволяющее регулировать размер tcp-window. 2-16Mb for Linux, 2-4Mb for Windows.
- Даже 4Mb окна недостаточно для LFN сетей. В 1Gb сети с RTT=250ms будем иметь 12Mb/sec.

Multipath-TCP

Преимущества

- Несколько параллельных tcp-подпотоков идущих по различным сетевым маршрутам.
- Обратно совместим с tcp
- Добавление и разрыв дочерних tcp-соединений без разрыва mp-tcp соединения.
- Реализован congestion avoidance.
- Работает с большинством middlebox’ов.

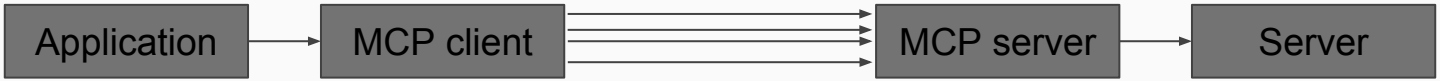
Multipath-TCP

Проблемы

- Полноценно реализован только в MacOS X и iOS.
- Для Linux существует реализация в ядре, но она не закоммичена в upstream
  - Multipath-tcp реализован интрузивно внутри tcp-уровня, требуется обширный рефакторинг, чтобы вынести его на отдельный слой над tcp.
  - Не реализован ряд опций утвержденных в стандарте.
- Реализация Multipath tcp для windows пока не планируется.

MCP (Multi-Connection Proxy)

- Более простое решение: давайте просто создадим N tcp-соединений между клиентом и сервером.
- Клиент принимает соединение от приложения и делит его на N tcp-соединений к серверу
- Сервер объединяет эти соединение в одно и передает дальше
- Если пакет с какого-то сокета приходит не в свою очередь, то данный сокет удаляется из epoll’a.



MCP (Multi-Connection Proxy)

Проблемы и задачи:

- Простое удаление сокета из epoll’а может быть не самым эффективным решением. Может быть есть способ управлять сокетами эффективнее?
- Необходимо измерить это влияние на общую скорость работы Multi-Connection Proxy и установить, есть ли смысл вообще заниматься этой задачей.
- Предложить какой-то другой алгоритм управления сокетами при задержках и потерях пакетов если это имеет смысл.

Итоги.

- Прочитано много теории по tcp и mptcp протоколам
- Получено представление о устройстве сетевого стека ядра linux.
- Получен опыт чтения и отладки низкоуровневого асинхронного сетевого кода

Вопросы?