gRPC研究

被<u>高建龙</u>添加,被<u>高建龙</u>最后更新于一月 14,2016

Overview

gRPC 的协议采用了 HTTP2 标准,序列化协议使用了 Protocol Buffer。

HTTP2由 HTTP1的文本协议改为二进制协议,以帧(Frame)为基本交换单元。

Frame的格式基本如下:



Type 可以是 HEADER、DATA 等等。

其中 Steam Identifier 来标识一个流,多个流可以并发地在一个连接上传输,以此实现多路复用。

服务端和客户端都可以主动发起流,客户端流ID都是奇数,服务端都是偶数。

HTTP2 同时还定义了流优先级、流量控制等流控制标准、可选的TLS加密传输以及基于HPACK的头部压缩标准。

具体在 gRPC 中,把 Steam Identifier 当作消息 ID 来使用,每一次请求都发起一个新的流。

每个请求的调用哪个服务和方法、回应的调用结果状态码都在 HEADER Frame 中指定。

请求内容和回应内容由 Protocol Buffer 序列化后使用 DATA Frame 传输。

请求:

```
▼ HyperText Transfer Protocol 2
  ▶ Stream: HEADERS, Stream ID: 1, Length 261

▼ HyperText Transfer Protocol 2

  ▶ Stream: WINDOW_UPDATE, Stream ID: 1, Length 4
  ▶ Stream: DATA, Stream ID: 1, Length 17
     00 00 00 07 3a 73 63 68 65 6d 65 00 00 00 04 68
                                                       ....:sch eme....h
     74 74 70 00 00 00 07 3a 6d 65 74 68 6f 64 00 00 ttp....: method..
     00 04 50 4f 53 54 00 00
                              00 05 3a 70 61 74 68 00
                                                        ..POST.. ..:path.
     00 00 20 2f 68 65 6c 6c
                              6f 77 6f 72 6c 64 2e 48
                                                        .. /hell oworld.H
0040 65 6c 6c 6f 53 65 72 76
                              69 63 65 2f 65 63 68 6f
                                                        elloServ ice/echo
0050 53 74 72 00 00 00 0a 3a 61 75 74 68 6f 72 69 74
                                                        Str....: authorit
                                                         y....loc alhost:9
     79 00 00 00 0e 6c 6f 63 61 6c 68 6f 73 74 3a 39
     39 39 39 00 00 00 0d 67
                              72 70 63 2d 65 6e 63 6f
                                                         999....g rpc-enco
     64 69 6e 67 00 00 00 08
                              69 64 65 6e 74 69 74 79
                                                        ding.... identity
                                                         ....grpc -accept-
     00 00 00 14 67 72 70 63
                              2d 61 63 63 65 70 74 2d
     65 6e 63 6f 64 69 6e 67
                              00 00 00 15 69 64 65 6e
                                                         encoding ....iden
     74 69 74 79 2c 64 65 66
                              6c 61 74 65 2c 67 7a 69
                                                        tity,def late,gzi
     70 00 00 00 02 74 65 00
                              00 00 08 74 72 61 69 6c
                                                        p....te. ...trail
     65 72 73 00 00 00 0c 63 6f 6e 74 65 6e 74 2d 74
                                                        ers....c ontent-t
     79 70 65 00 00 00 10 61 70 70 6c 69 63 61 74 69
                                                        ype....a pplicati
```

回应:

```
▼ HyperText Transfer Protocol 2
  ▶ Stream: DATA, Stream ID: 1, Length 29
  ▶ Stream: HEADERS, Stream ID: 1, Length 15
     0010 00 72 98 c7 40 00 40 06 a3 bc 7f 00 00 01 7f 00
                                                   .г..@.@. .....
    00 01 27 0f a1 4d ca 24 91 18 f5 c2 5c 01 80 18
                                                   ..'..M.$ ....\...
0030 01 56 fe 66 00 00 01 01
                           08 0a 02 35 08 bb 02
                                                    .hello grpc
0040 08 bb 00 00 1d 00 00 00 00 01 00 00 00 00 18
     0a 16 68 65 6c 6c 6f 20 67 72 70 63 20 66 72 6f
0050
0060
     6d 20 73 65 72 76 65 72 00 00 0f 01 05 00 00 00
                                                   m server .....
     01 40 0b 67 72 70 63 2d 73 74 61 74 75 73 01 30
                                                   .@.grpc- status.0
```

gRPC 还支持一种 Steaming RPC 的调用方式。

```
# proto 文件
service HelloService {
    rpc sayHello(HelloRequest) returns (stream HelloReply) {}
}
```

服务端可以返回多个 HelloReply, 客户端连续读取直到流关闭。

多个 HelloReply 以 Steam Identifier 相同的多个 DATA Frame 的形式传输。

gRPC实现了 SSL/TLS 和 OAuth 2.0 两种认证机制。

启用 SSL/TLS 认证机制也同时会加密所有传输数据。

性能测试

单连接20个发送线程

环境一: 本机 loopback

环境二: client 192.168.200.196

server 192.168.200.197

测试代码: http://source.jd.com/app/grpc-benchmark

测试proto文件:

```
syntax = "proto3";
package benchmark;
service BenchmarkService {
   rpc getSimpleObject(SimpleObject) returns (SimpleObject) {}
}
message SimpleObject {
   int32 id
                   = 2;
   string desc
   bool boy
                   = 3;
                   = 4;
   int64 salary
   string money
                   = 5;
   int64 date
                   = 6;
   repeated int32 intArr = 7;
    int32 oneByte = 8;
};
```

测试结果:

数据单位 qps 每秒请求数

	不启用 SSL/TLS 加密传输	启用 SSL/TLS 加密传输
环境一	3.2w	2.5w
环境二	1.3w	1.1w

无