

官方设计文档：[grpc/doc at master · grpc/grpc \(github.com\)](https://github.com/grpc/grpc/blob/master/doc/README.md)
[gRPC 官方文档中文版 V1.0 \(oschina.net\)](http://oschina.net/doc/4444444)
[gRPC](https://github.com/grpc/grpc)

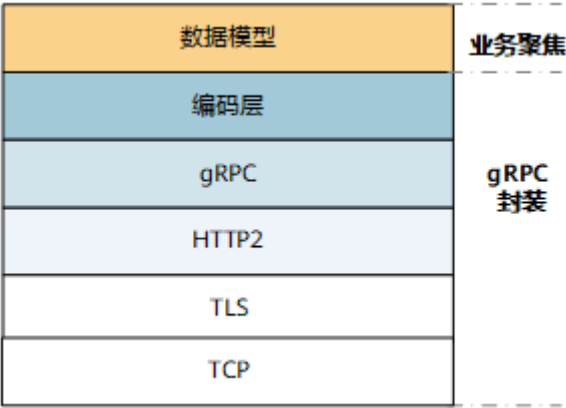
- 主要内容
- 异步和同步的线程网络模型
 - grpc应用实践，比如多个service，比如异步调用
 - 异步和同步的性能测试

- 课后扩展：如何研究线程模型
- listen
 - accept
 - epoll_ctl
 - epoll_wait

1 gRPC协议框架

gRPC协议栈分层

gRPC协议栈分层如下图所示。



各层详细说明如表 各层详细说明所示。

层次	说明
TCP层	底层通信协议，基于TCP连接。
TLS层	该层是可选的，基于TLS加密通道。
HTTP2层	gRPC承载在HTTP2协议上，利用了HTTP2的双向流、流控、头部压缩、单连接上的多路复用请求等特性。
gRPC层	远程过程调用，定义了远程过程调用的协议交互格式。

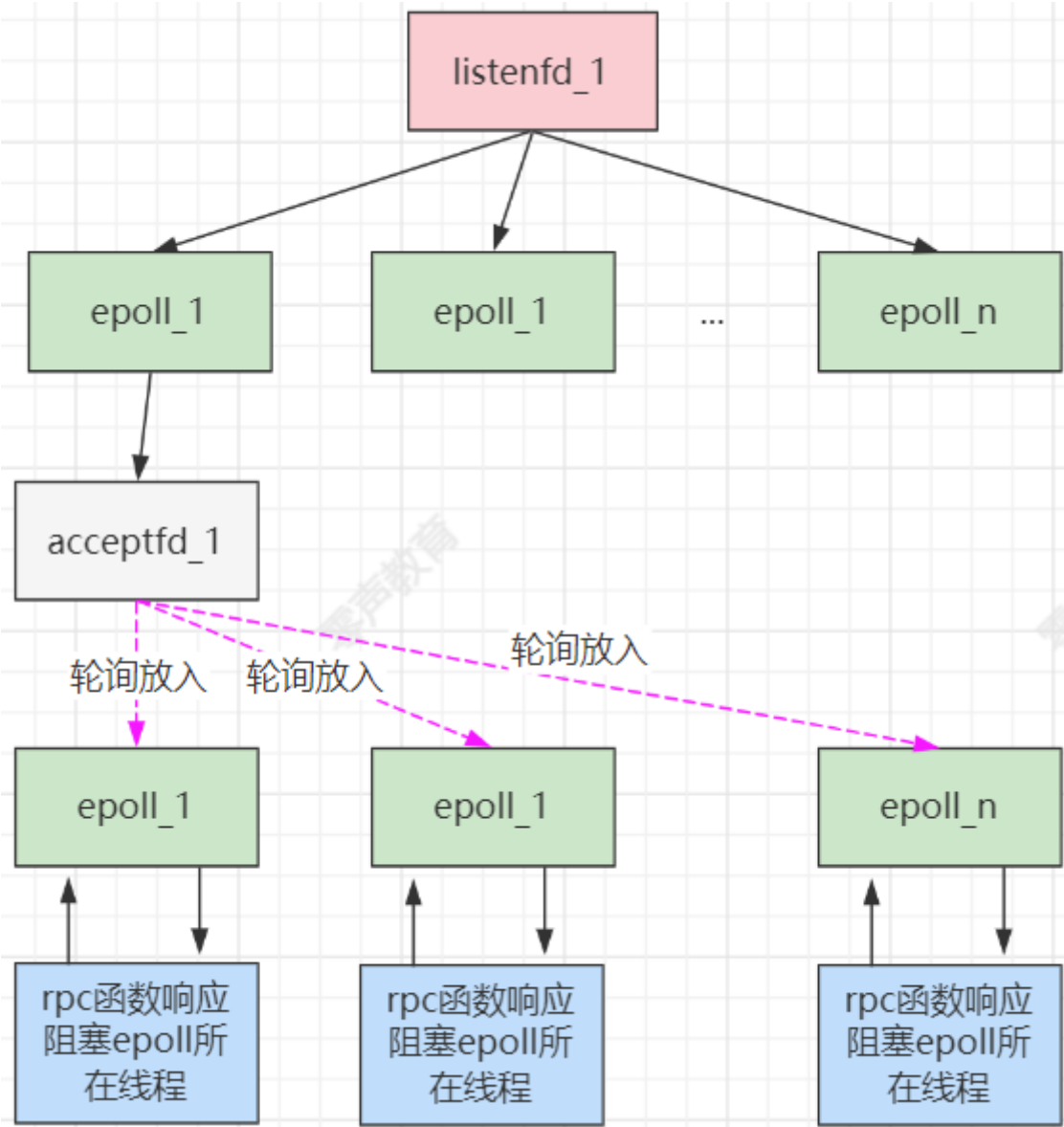
层次	说明
编码层	gRPC通过编码格式承载数据，包括GPB（Google Protocol Buffer）编码格式。
数据模型层	业务模块的数据。通信双方需要了解彼此的数据模型，才能正确调用信息。当前设备提供了订阅、配置、查询业务模块。proto文件

如果需要加密在创建channel的时候需要引入tls。

2 grpc网络模型

grpc会启动多个线程的epoll来处理描述符，不管异步还是同步，每个epoll都对应一个线程。

同步模型

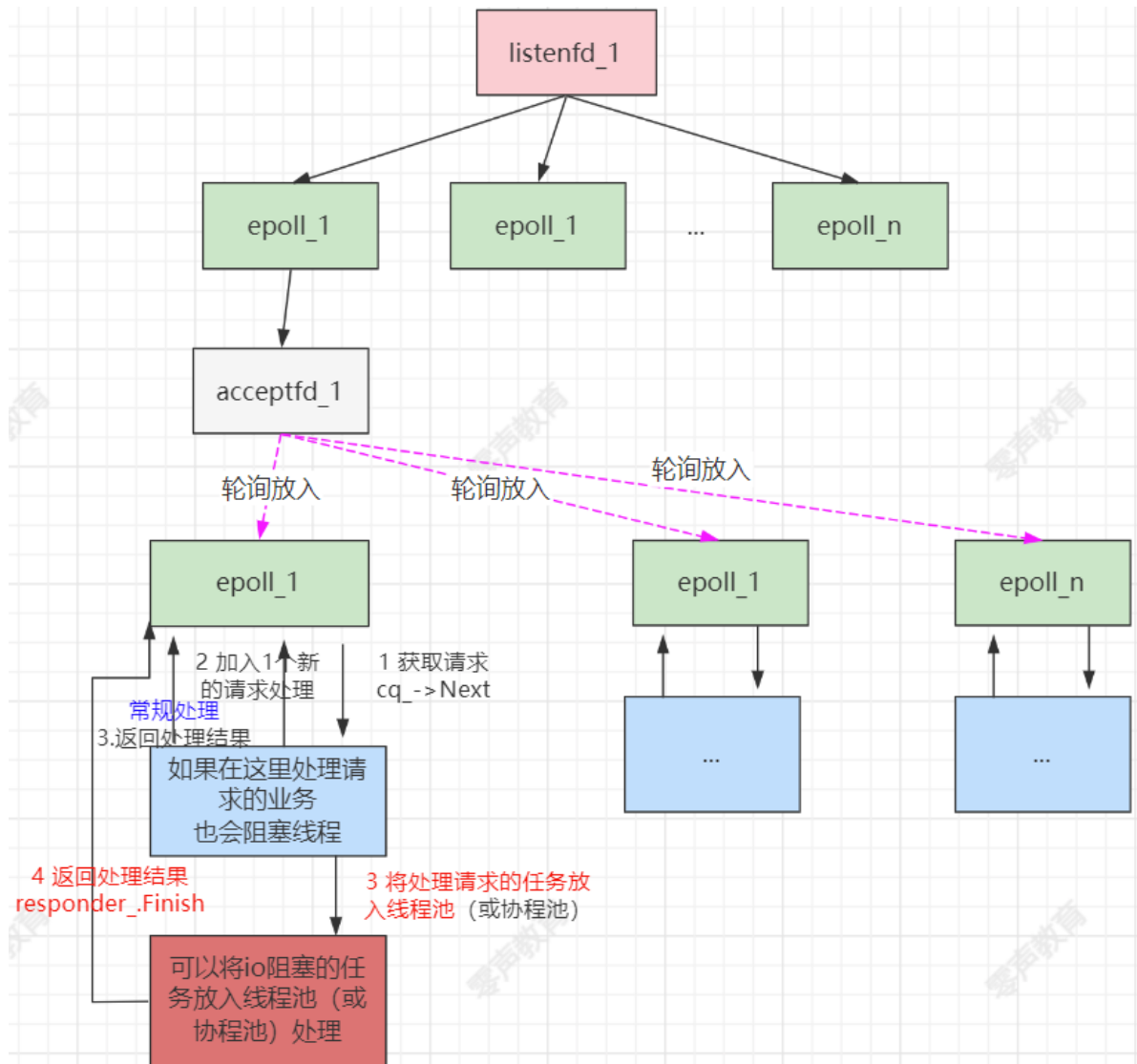


- 得益于SO_REUSEPORT参数，同一个listenfd可以被放到多个epoll中进行监听
- 当一个链接成功建立后会生成acceptfd，这个acceptfd会被随机的分配到现有的epoll中，目前grpc的分配策略是轮询（round-robin）

min poller， max poller， 自动根据调用的请求的频次 进行自动伸缩poller。

异步模型

异步处理的epoll方式和同步是类似的，但对于rpc函数的响应提供了更灵活的处理机制，可以将一些耗时的处理逻辑放到外部的线程池进行处理。



3 同步多函数多个类调用

- 服务端：一个server可以有多个service，在proto文件对应service；
- 客户端：一个channel（对应一个链路）可以供多个stub使用；就是不同的stub共用一个链路

proto文件

- 添加Greeter2类，在Greeter2类里添加2个rpc调用
- 添加HelloRequest2、HelloReply2两个请求、响应类型。

文件名：helloworld.proto

```
// Copyright 2015 gRPC authors.
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
//      http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

syntax = "proto3";

option java_multiple_files = true;
option java_package = "io.grpc.examples.helloworld";
option java_outer_classname = "HelloWorldProto";
option objc_class_prefix = "HLW";

package helloworld;

// The greeting service definition.
service Greeter {
    // Sends a greeting
    rpc SayHello (HelloRequest) returns (HelloReply) {}
}

service Greeter2 {
    // Sends a greeting
    rpc SayHello (HelloRequest) returns (HelloReply) {}
    rpc SayHello2 (HelloRequest2) returns (HelloReply2) {}
}

// The request message containing the user's name.
message HelloRequest {
    string name = 1;
}

// The response message containing the greetings
message HelloReply {
    string message = 1;
}
```

```
// The request message containing the user's name.
message HelloRequest2 {
    string name = 1;
}

// The response message containing the greetings
message HelloReply2 {
    string message = 1;
}
```

服务端重点

上一节课讲到一个server可以对应多个service，在proto文件定义的service，比如下所示：

```
service Greeter{
    // Sends a greeting
    rpc SayHello (HelloRequest) returns (HelloReply) {}
}
```

即是创建了一个service。

在原有的greeter_server.cc修改

0. 开放命名空间

```
using helloworld::Greeter2;
using helloworld::HelloReply2;
using helloworld::HelloRequest2;
```

1. 增加service

```
class GreeterServiceImpl final : public Greeter::Service {
}

// 增加对Greeter2的响应
class GreeterServiceImpl2 final : public Greeter2::Service {
    Status SayHello(ServerContext* context, const HelloRequest* request,
                    HelloReply* reply) override {
        std::string prefix("Greeter2-1 Hello ");
        reply->set_message(prefix + request->name());
        std::cout << "Greeter2 SayHello: " << reply->message() << std::endl;
        return Status::OK;
    }

    Status SayHello2(ServerContext* context, const HelloRequest2* request,
                    HelloReply2* reply) override {
        std::string prefix("Greeter2-2 Hello ");
        reply->set_message(prefix + request->name());
    }
}
```

```

        std::cout << "Greeter2 SayHello2: " << reply->message() << std::endl;
        return Status::OK;
    }
};

```

2. 将service注册到server

```

GreeterServiceImpl service;
GreeterServiceImpl2 service2; // 实例化service

builder.RegisterService(&service);
builder.RegisterService(&service2); // 注册service

```

3. 完整代码

文件名: greeter_server.cc

```

using grpc::Server;
using grpc::ServerBuilder;
using grpc::ServerContext;
using grpc::Status;
using helloworld::Greeter;
using helloworld::Greeter2;
using helloworld::HelloReply;
using helloworld::HelloReply2;
using helloworld::HelloRequest;
using helloworld::HelloRequest2;
// Logic and data behind the server's behavior.
class GreeterServiceImpl final : public Greeter::Service {
    Status SayHello(ServerContext *context, const HelloRequest *request,
                    HelloReply *reply) override {
        std::string prefix("Greeter Hello ");
        reply->set_message(prefix + request->name());
        std::cout << "Greeter SayHello: " << reply->message() << std::endl;
        return Status::OK;
    }
};

// 增加对Greeter2的响应
class GreeterServiceImpl2 final : public Greeter2::Service {
    Status SayHello(ServerContext *context, const HelloRequest *request,
                    HelloReply *reply) override {
        std::string prefix("Greeter2-1 Hello ");
        reply->set_message(prefix + request->name());
        std::cout << "Greeter2 SayHello: " << reply->message() << std::endl;
        return Status::OK;
    }

    Status SayHello2(ServerContext *context, const HelloRequest2 *request,
                     HelloReply2 *reply) override {
        std::string prefix("Greeter2-2 Hello ");

```

```

        reply->set_message(prefix + request->name());
        std::cout << "Greeter2 SayHello2: " << reply->message() << std::endl;
        return Status::OK;
    }
};

void RunServer() {
    std::string server_address("0.0.0.0:50051");
    GreeterServiceImpl service;
    GreeterServiceImpl2 service2;

    grpc::EnableDefaultHealthCheckService(true);
    grpc::reflection::InitProtoReflectionServerBuilderPlugin();
    ServerBuilder builder;
    builder.SetSyncServerOption(ServerBuilder::MIN_POLLERS, 2);
    builder.SetSyncServerOption(ServerBuilder::MAX_POLLERS, 4);
    // builder.SetSyncServerOption(ServerBuilder::MAX_POLLERS, 4); //
    // 可以设置大小 Listen on the given address without any authentication
    // mechanism.
    builder.AddListeningPort(server_address, grpc::InsecureServerCredentials());
    // Register "service" as the instance through which we'll communicate with
    // clients. In this case it corresponds to an *synchronous* service.
    builder.RegisterService(&service);
    builder.RegisterService(&service2);
    // Finally assemble the server.
    std::unique_ptr<Server> server(builder.BuildAndStart());
    std::cout << "server listening on " << server_address << std::endl;

    // Wait for the server to shutdown. Note that some other thread must be
    // responsible for shutting down the server for this call to ever return.
    server->wait();
}

int main(int argc, char **argv) {
    RunServer();

    return 0;
}

```

客户端重点

在greeter_client.cc基础上修改。

1. 开放命名空间

```

using helloworld::Greeter2;
using helloworld::HelloReply2;
using helloworld::HelloRequest2;

```

2. 新增stub

在GreeterClient类里新增stub2

```
std::unique_ptr<Greeter::Stub> stub_;  
std::unique_ptr<Greeter2::Stub> stub2_; // 新增stub2
```

3. 修改GreeterClient构造函数

注意：stub2_(Greeter2::NewStub(channel))的初始化，这里是Greeter2::NewStub，前往不要写成Greeter::NewStub(channel)

```
GreeterClient(std::shared_ptr<Channel> channel)  
    : stub_(Greeter::NewStub(channel)), stub2_(Greeter2::NewStub(channel)) {  
}
```

4. 发送请求，注意使用相应的stub

```
std::string SayHello2_1(const std::string &user) {  
    HelloRequest request;  
    request.set_name(user);  
    HelloReply reply;  
    ClientContext context;  
    Status status = stub2_->SayHello(&context, request, &reply);  
    if (status.ok()) {  
        return reply.message();  
    } else {  
        std::cout << status.error_code() << ": " << status.error_message()  
            << std::endl;  
        return "RPC failed";  
    }  
}  
  
std::string SayHello2_2(const std::string &user) {  
    HelloRequest2 request;  
    request.set_name(user);  
    HelloReply2 reply;  
    ClientContext context;  
    Status status = stub2_->SayHello2(&context, request, &reply);  
    if (status.ok()) {  
        return reply.message();  
    } else {  
        std::cout << status.error_code() << ": " << status.error_message()  
            << std::endl;  
        return "RPC failed";  
    }  
}
```

5. 测试，需要先启动 greeter_server端


```

GreeterClient greeter(
    grpc::CreateChannel(target_str, grpc::InsecureChannelCredentials()));
std::string user("world");
std::string reply = greeter.SayHello(user);
std::cout << "Greeter received: " << reply << std::endl << std::endl;

reply = greeter.SayHello2_1(user);
std::cout << "Greeter received: " << reply << std::endl << std::endl;

reply = greeter.SayHello2_2(user);
std::cout << "Greeter received: " << reply << std::endl << std::endl;

```

6. 完整的代码

```

#include <iostream>
#include <memory>
#include <string>

#include <grpcpp/grpcpp.h>

#ifdef BAZEL_BUILD
#include "examples/protos/helloworld.grpc.pb.h"
#else
#include "helloworld.grpc.pb.h"
#endif

using grpc::Channel;
using grpc::ClientContext;
using grpc::Status;
using helloworld::Greeter;
using helloworld::Greeter2;
using helloworld::HelloReply;
using helloworld::HelloReply2;
using helloworld::HelloRequest;
using helloworld::HelloRequest2;

class GreeterClient {
public:
    GreeterClient(std::shared_ptr<Channel> channel)
        : stub_(Greeter::NewStub(channel)), stub2_(Greeter2::NewStub(channel)) {}

    // Assembles the client's payload, sends it and presents the response back
    // from the server.
    std::string SayHello(const std::string &user) {
        // Data we are sending to the server.
        HelloRequest request;
        request.set_name(user);

        // Container for the data we expect from the server.
        HelloReply reply;

        // Context for the client. It could be used to convey extra information

```

```

// to the server and/or tweak certain RPC behaviors.
ClientContext context;

// The actual RPC.
Status status = stub_>SayHello(&context, request, &reply);

// Act upon its status.
if (status.ok()) {
    return reply.message();
} else {
    std::cout << status.error_code() << ": " << status.error_message()
              << std::endl;
    return "RPC failed";
}
}

std::string SayHello2_1(const std::string &user) {
    // Data we are sending to the server.
    HelloRequest request;
    request.set_name(user);

    // Container for the data we expect from the server.
    HelloReply reply;

    // Context for the client. It could be used to convey extra information
    // to the server and/or tweak certain RPC behaviors.
    ClientContext context;

    // The actual RPC.
    Status status = stub2_>SayHello(&context, request, &reply);

    // Act upon its status.
    if (status.ok()) {
        return reply.message();
    } else {
        std::cout << status.error_code() << ": " << status.error_message()
              << std::endl;
        return "RPC failed";
    }
}

std::string SayHello2_2(const std::string &user) {
    // Data we are sending to the server.
    HelloRequest2 request;
    request.set_name(user);

    // Container for the data we expect from the server.
    HelloReply2 reply;

    // Context for the client. It could be used to convey extra information
    // to the server and/or tweak certain RPC behaviors.
    ClientContext context;

    // The actual RPC.
    Status status = stub2_>SayHello2(&context, request, &reply);

```

```

        // Act upon its status.
        if (status.ok()) {
            return reply.message();
        } else {
            std::cout << status.error_code() << ": " << status.error_message()
                << std::endl;
            return "RPC failed";
        }
    }
}

private:
    std::unique_ptr<Greeter::Stub> stub_;
    std::unique_ptr<Greeter2::Stub> stub2_; // 新增stub2
};

int main(int argc, char **argv) {
    // Instantiate the client. It requires a channel, out of which the actual
    // RPCs are created. This channel models a connection to an endpoint
    // specified by the argument "--target=" which is the only expected
    // argument. We indicate that the channel isn't authenticated (use of
    // InsecureChannelCredentials()).
    std::string target_str;
    std::string arg_str("--target");
    if (argc > 1) {
        std::string arg_val = argv[1];
        size_t start_pos = arg_val.find(arg_str);
        if (start_pos != std::string::npos) {
            start_pos += arg_str.size();
            if (arg_val[start_pos] == '=') {
                target_str = arg_val.substr(start_pos + 1);
            } else {
                std::cout << "The only correct argument syntax is --target="
                    << std::endl;
                return 0;
            }
        } else {
            std::cout << "The only acceptable argument is --target="
                << std::endl;
            return 0;
        }
    } else {
        target_str = "localhost:50051";
    }
    GreeterClient greeter(
        grpc::CreateChannel(target_str, grpc::InsecureChannelCredentials()));
    std::string user("world");
    std::string reply = greeter.SayHello(user);
    std::cout << "Greeter received: " << reply << std::endl << std::endl;

    reply = greeter.SayHello2_1(user);
    std::cout << "Greeter received: " << reply << std::endl << std::endl;

    reply = greeter.SayHello2_2(user);
    std::cout << "Greeter received: " << reply << std::endl << std::endl;
}

```

```
    return 0;
}
```

4 异步多函数多类调用

参考本文第4章节，以及上一节课补充的《1-3-grpc for c++使用案例-手把手.pdf》

5 性能测试

- 同步异步客户端、同步异步服务器交叉测试。
- 同步服务端不同线程数量测试
- 异步服务端不同线程数量测试

先思考服务器线程数量对于性能的影响。

具体见课上测试。

查看线程数

```
grep 'processor' /proc/cpuinfo | sort -u | wc -l
```

注意，此处查看的线程数是总的线程数，可以理解为逻辑cpu的数量，**这里本人获取的结果为4**

服务端：

- 同步服务端：greeter_server_test.cc 可以设置最小、最大poller数量
 - 范例：
- 异步服务端：
 - greeter_async_server_test.cc 可以设置epoll线程数量
 - 范例：greeter_async_server_test 2 即是开启2个epoll线程
 - greeter_async_server_test2.cc 可以设置epoll线程数量，**线程池线程数量**
 - 范例：./greeter_async_server_test2 2 4 即是开启2个epoll线程，线程池有4个线程

客户端：

- 同步客户端：greeter_client_test.cc 可以设置线程数量以及每个线程的请求数量
 - 范例：./greeter_client_test 1 10000 即是开启1个发送线程每个线程发送10000次请求
- 异步客户端：greeter_async_client2_test.cc 单个线程发送异步请求，可以设置请求数量，在另一个线程获取结果

- 范例: `./greeter_async_client2_test 10000` 即是异步的方式发送10000次请求, 在另一个线程接收返回结果

测试环境, 逻辑cpu数量为4, 并且程序是debug方式编译

异步客户端: `greeter_async_client2_test`

服务端程序 (epoll数量,线程池数量)	1, 1	1, 2	1, 4	2, 2	2, 4	4, 4	8, 8	
<code>greeter_async_server_test2</code>	8865	9699	9871	10471	10330	11001	8438	
<code>greeter_async_server_test2</code>	8496	9803	9708	10615	9960	10683	9842	
<code>greeter_async_server_test2</code>	9980	9871	9861	11750	9910	11013	9560	
服务端程序 (epoll数量)	1			2		4	8	
<code>greeter_async_server_test</code>	11415			11560		11641	9910	
<code>greeter_async_server_test</code>	11049			11600		12062	10775	
<code>greeter_async_server_test</code>	10570			12376		11402	10582	
服务端程序 (min epolls,max epolls)	1, 1	1, 2	1, 4	2, 2	2, 4	4, 4	1, 8	8, 8
<code>greeter_server_test</code>	5733	9970	10256	6165	9861	6796	10319	5973
<code>greeter_server_test</code>	5740	10010	10822	6297	9587	5938	10319	6199
<code>greeter_server_test</code>	4911	10341	10741	5549	9596	6596	10729	5970

同步服务端: 能够自动调整epoll线程数量 2~cpu逻辑数量

同步客户端: `./greeter_client_test 1 10000` 这里简单测试, 单线程发起10000请求。

同步客户端而言, 不能充分使用带宽

req ->

等待 这段时间不能发请求

<- resp

req ->

等待 这段时间不能发请求

服务端程序 (epoll数量,线程池数量)	1, 1	1, 2	1, 4	2, 2	2, 4	4, 4	8, 8	
<code>greeter_async_server_test2</code>				3299				
<code>greeter_async_server_test2</code>				2656				
<code>greeter_async_server_test2</code>				3257				
服务端程序 (epoll数量)	1			2		4	8	

服务端程序 (epoll数量,线程池数量)	1, 1	1, 2	1, 4	2, 2	2, 4	4, 4	8, 8	
greeter_async_server_test				3020				
greeter_async_server_test				2578				
greeter_async_server_test				2525				
服务端程序 (min epolls,max epolls)	1, 1	1, 2	1, 4	2, 2	2, 4	4, 4	1, 8	8, 8
greeter_server_test		3709.2					3467	2957
greeter_server_test		2955.96					3474	2787
greeter_server_test		3557.45					3384	2876

异步客户端。

6 以下内容供debug分析做参考

如何研究线程模型：

- listen
- accept/ grpc_accept4 , 建议在grpc_accept4打断点
- epoll_ctl
- epoll_wait

异步 greeter_async_server

listen

2 grpc_tcp_server_prepare_socket

```
Thread 1 "greeter_async_s" hit Breakpoint 2, 0x00007ffff6bc5310 in listen () from /lib/x86_64-linux-gnu/libc.so.6
(gdb) bt
#0  0x00007ffff6bc5310 in listen () from /lib/x86_64-linux-gnu/libc.so.6
#1  0x00000000006f76ee in grpc_tcp_server_prepare_socket(grpc_tcp_server*, int, grpc_resolved_address const*, bool, int*) ()
#2  0x00000000006f6fe7 in add_socket_to_server(grpc_tcp_server*, int, grpc_resolved_address const*, unsigned int, unsigned int, grpc_tcp_listener**) ()
```

```

#3 0x00000000006f73e6 in grpc_tcp_server_add_addr(grpc_tcp_server*,
grpc_resolved_address const*, unsigned int, unsigned int, grpc_dualstack_mode*,
grpc_tcp_listener**) ()
#4 0x00000000006f54a5 in add_wildcard_addrs_to_server(grpc_tcp_server*, unsigned
int, int, int*) ()
#5 0x00000000006f5db7 in tcp_server_add_port(grpc_tcp_server*,
grpc_resolved_address const*, int*) ()
#6 0x00000000006f447e in grpc_tcp_server_add_port(grpc_tcp_server*,
grpc_resolved_address const*, int*) ()
#7 0x0000000000662913 in grpc_core::(anonymous
namespace)::Http2ServerListener::Create(grpc_core::Server*,
grpc_resolved_address*, grpc_channel_args*, std::function<grpc_channel_args*
(grpc_channel_args*, grpc_error**)>, int*)::(lambda()#1)::operator()() const
()
#8 0x0000000000662b56 in grpc_core::(anonymous
namespace)::Http2ServerListener::Create(grpc_core::Server*,
grpc_resolved_address*, grpc_channel_args*, std::function<grpc_channel_args*
(grpc_channel_args*, grpc_error**)>, int*) ()
#9 0x00000000006640a2 in grpc_core::Http2ServerAddPort(grpc_core::Server*, char
const*, grpc_channel_args*, std::function<grpc_channel_args* (grpc_channel_args*,
grpc_error**)>, int*)::(lambda()#1)::operator()() const ()
#10 0x000000000066473c in grpc_core::Http2ServerAddPort(grpc_core::Server*, char
const*, grpc_channel_args*, std::function<grpc_channel_args* (grpc_channel_args*,
grpc_error**)>, int*) ()
#11 0x0000000000664e72 in grpc_server_add_http2_port ()
#12 0x000000000042fa3b in grpc::(anonymous
namespace)::InsecureServerCredentialsImpl::AddPortToServer(std::__cxx11::basic_s
tring<char, std::char_traits<char>, std::allocator<char> > const&, grpc_server*)
()
#13 0x0000000000440f53 in
grpc::Server::AddListeningPort(std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> > const&, grpc::ServerCredentials*)
()
#14 0x0000000000431d2c in grpc::ServerBuilder::BuildAndStart() ()
#15 0x000000000041041f in ServerImpl::Run (this=0x7fffffffddcf0)
--Type <RET> for more, q to quit, c to continue without paging--
    at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:91
#16 0x000000000040bad0 in main (argc=1, argv=0x7fffffffde58)

```

4 grpc_tcp_server_prepare_socket

```

Thread 1 "greeter_async_s" hit Breakpoint 2, 0x00007ffff6bc5310 in listen () from
/lib/x86_64-linux-gnu/libc.so.6
(gdb) bt
#0 0x00007ffff6bc5310 in listen () from /lib/x86_64-linux-gnu/libc.so.6
#1 0x00000000006f76ee in grpc_tcp_server_prepare_socket(grpc_tcp_server*, int,
grpc_resolved_address const*, bool, int*) ()
#2 0x00000000006f58e9 in clone_port(grpc_tcp_listener*, unsigned int) ()
#3 0x00000000006f6133 in tcp_server_start(grpc_tcp_server*,
std::vector<grpc_pollset*, std::allocator<grpc_pollset*> > const*, void (*)
(void*, grpc_endpoint*, grpc_pollset*, grpc_tcp_server_acceptor*), void*) ()

```

```

#4 0x00000000006f4444 in grpc_tcp_server_start(grpc_tcp_server*,
std::vector<grpc_pollset*, std::allocator<grpc_pollset*> > const*, void (*)
(void*, grpc_endpoint*, grpc_pollset*, grpc_tcp_server_acceptor*), void*) ()
#5 0x0000000000663265 in grpc_core::(anonymous
namespace)::Http2ServerListener::StartListening() ()
#6 0x00000000006631cb in grpc_core::(anonymous
namespace)::Http2ServerListener::Start(grpc_core::Server*,
std::vector<grpc_pollset*, std::allocator<grpc_pollset*> > const*) ()
#7 0x0000000000745def in grpc_core::Server::Start() ()
#8 0x0000000000749bd5 in grpc_server_start ()
#9 0x0000000000441605 in grpc::Server::Start(grpc::ServerCompletionQueue**,
unsigned long) ()
#10 0x0000000000431e47 in grpc::ServerBuilder::BuildAndStart() ()
#11 0x000000000041041f in ServerImpl::Run (this=0x7fffffffdfcf0)
    at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:91
#12 0x000000000040bad0 in main (argc=1, argv=0x7fffffffde58)

```

accept

grpc_accept4

```

Thread 5 "greeter_async_s" hit Breakpoint 4, 0x000000000094ab33 in
grpc_accept4(int, grpc_resolved_address*, int, int) ()
(gdb) bt
#0 0x000000000094ab33 in grpc_accept4(int, grpc_resolved_address*, int, int) ()
#1 0x00000000006f4eb9 in on_read(void*, grpc_error*) ()
#2 0x00000000006e6d9f in exec_ctx_run(grpc_closure*) ()
#3 0x00000000006e6ea1 in grpc_core::ExecCtx::Flush() ()
#4 0x000000000093cfc5 in end_worker(grpc_pollset*, grpc_pollset_worker*,
grpc_pollset_worker**) ()
#5 0x000000000093d2e7 in pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#6 0x00000000006e660c in pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#7 0x00000000006e91e1 in grpc_pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#8 0x00000000007410e9 in cq_next(grpc_completion_queue*, gpr_timespec, void*)
()
#9 0x000000000074166b in grpc_completion_queue_next ()
#10 0x000000000042d60c in grpc::CompletionQueue::AsyncNextInternal(void**, bool*,
gpr_timespec) ()
#11 0x000000000040ca6a in grpc::CompletionQueue::Next (this=0x13d1210,
tag=0x7ffff52b8cc8, ok=0x7ffff52b8cc7)
    at /usr/local/include/grpcpp/impl/codegen/completion_queue.h:185
#12 0x0000000000410c90 in ServerImpl::HandleRpcs (this=0x7fffffffdfcf0, cq=...)
    at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:240
#13 0x000000000041074f in ServerImpl::_run (this=0x7fffffffdfcf0, i=0)
    at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:126

```



```

#14 0x00000000004199eb in std::__invoke_impl<void, void (ServerImpl::*)(int),
ServerImpl*, unsigned long> (__f=
    @0x13d1608: (void (ServerImpl::*)(ServerImpl * const, int)) 0x4106ea
<ServerImpl::_run(int)>, __t=@0x13d1600: 0x7fffffffdfcf0)
    at /usr/local/gcc-11.2/include/c++/11.2.0/bits/invoke.h:74
#15 0x0000000000419944 in std::__invoke<void (ServerImpl::*)(int), ServerImpl*,
unsigned long> (__fn=
    @0x13d1608: (void (ServerImpl::*)(ServerImpl * const, int)) 0x4106ea
<ServerImpl::_run(int)>)
    at /usr/local/gcc-11.2/include/c++/11.2.0/bits/invoke.h:96
#16 0x00000000004197b3 in std::thread::_Invoker<std::tuple<void (ServerImpl::*)(
int), ServerImpl*, unsigned long> >::_M_invoke<0ul, --Type <RET> for more, q to
quit, c to continue without paging--
1ul, 2ul> (this=0x13d15f8) at /usr/local/gcc-
11.2/include/c++/11.2.0/bits/std_thread.h:253
#17 0x0000000000418f36 in std::thread::_Invoker<std::tuple<void (ServerImpl::*)(
int), ServerImpl*, unsigned long> >::operator() (
    this=0x13d15f8) at /usr/local/gcc-
11.2/include/c++/11.2.0/bits/std_thread.h:260
#18 0x00000000004181a8 in
std::thread::_State_impl<std::thread::_Invoker<std::tuple<void (ServerImpl::*)(
int), ServerImpl*, unsigned long> > >::_M_run (this=0x13d15f0) at
/usr/local/gcc-11.2/include/c++/11.2.0/bits/std_thread.h:211
#19 0x00007fffff7478e80 in std::execute_native_thread_routine (__p=0x13d15f0) at
../../../../../libstdc++-v3/src/c++11/thread.cc:82
#20 0x00007fffff7b56ba in start_thread () from /lib/x86_64-linux-
gnu/libpthread.so.0
#21 0x00007fffff6bc451d in clone () from /lib/x86_64-linux-gnu/libc.so.6

```

epoll_ctl

1 pollset_global_init

```

Thread 1 "greeter_async_s" hit Breakpoint 5, 0x00007fffff6bc4ab0 in epoll_ctl ()
from /lib/x86_64-linux-gnu/libc.so.6
(gdb) bt
#0 0x00007fffff6bc4ab0 in epoll_ctl () from /lib/x86_64-linux-gnu/libc.so.6
#1 0x000000000093b982 in pollset_global_init() ()
#2 0x000000000093e1e0 in grpc_init_epoll1_linux(bool) ()
#3 0x00000000006e5c81 in try_engine(char const*) ()
#4 0x00000000006e5ebf in grpc_event_engine_init() ()
#5 0x00000000006e8c62 in iomgr_platform_init() ()
#6 0x00000000006e8b6e in grpc_iomgr_platform_init() ()
#7 0x0000000000948552 in grpc_iomgr_init() ()
#8 0x0000000000743b62 in grpc_init ()
#9 0x000000000042df43 in grpc::internal::GrpcLibrary::init() ()
#10 0x000000000041bed2 in grpc::GrpcLibraryCodegen::GrpcLibraryCodegen
(this=0x13cde50, call_grpc_init=true)
    at /usr/local/include/grpcpp/impl/codegen/grpc_library.h:48
--Type <RET> for more, q to quit, c to continue without paging--
#11 0x0000000000458e49 in grpc::ServerCredentials::ServerCredentials() ()

```

```
#12 0x000000000042fab0 in grpc::(anonymous
namespace)::InsecureServerCredentialsImpl::InsecureServerCredentialsImpl() ()
#13 0x000000000042fafe in grpc::InsecureServerCredentials() ()
#14 0x0000000000410331 in ServerImpl::Run (this=0x7fffffffdfcf0)
    at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:76
#15 0x000000000040bad0 in main (argc=1, argv=0x7fffffffde58)
    at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:261
```

3 fd_create

```
Thread 1 "greeter_async_s" hit Breakpoint 5, 0x00007ffff6bc4ab0 in epoll_ctl ()
from /lib/x86_64-linux-gnu/libc.so.6
(gdb) bt
#0 0x00007ffff6bc4ab0 in epoll_ctl () from /lib/x86_64-linux-gnu/libc.so.6
#1 0x000000000093b2af in fd_create(int, char const*, bool) ()
#2 0x00000000006e6107 in grpc_fd_create(int, char const*, bool) ()
#3 0x00000000006f71bd in add_socket_to_server(grpc_tcp_server*, int,
grpc_resolved_address const*, unsigned int, unsigned int, grpc_tcp_listener**)
()
#4 0x00000000006f73e6 in grpc_tcp_server_add_addr(grpc_tcp_server*,
grpc_resolved_address const*, unsigned int, unsigned int, grpc_dualstack_mode*,
grpc_tcp_listener**) ()
#5 0x00000000006f54a5 in add_wildcard_addrs_to_server(grpc_tcp_server*, unsigned
int, int, int*) ()
#6 0x00000000006f5db7 in tcp_server_add_port(grpc_tcp_server*,
grpc_resolved_address const*, int*) ()
#7 0x00000000006f447e in grpc_tcp_server_add_port(grpc_tcp_server*,
grpc_resolved_address const*, int*) ()
#8 0x0000000000662913 in grpc_core::(anonymous
namespace)::Http2ServerListener::Create(grpc_core::Server*,
grpc_resolved_address*, grpc_channel_args*, std::function<grpc_channel_args*
(grpc_channel_args*, grpc_error**)>, int*)::{{lambda()#1}}::operator()() const
()
#9 0x0000000000662b56 in grpc_core::(anonymous
namespace)::Http2ServerListener::Create(grpc_core::Server*,
grpc_resolved_address*, grpc_channel_args*, std::function<grpc_channel_args*
(grpc_channel_args*, grpc_error**)>, int*) ()
#10 0x00000000006640a2 in grpc_core::Http2ServerAddPort(grpc_core::Server*, char
const*, grpc_channel_args*, std::function<grpc_channel_args* (grpc_channel_args*,
grpc_error**)>, int*)::{{lambda()#1}}::operator()() const ()
#11 0x000000000066473c in grpc_core::Http2ServerAddPort(grpc_core::Server*, char
const*, grpc_channel_args*, std::function<grpc_channel_args* (grpc_channel_args*,
grpc_error**)>, int*) ()
#12 0x0000000000664e72 in grpc_server_add_http2_port ()
#13 0x000000000042fa3b in grpc::(anonymous
namespace)::InsecureServerCredentialsImpl::AddPortToServer(std::__cxx11::basic_s
tring<char, std::char_traits<char>, std::allocator<char> > const&, grpc_server*)
()
```

```

#14 0x0000000000440f53 in
grpc::Server::AddListeningPort(std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> > const&, grpc::ServerCredentials*)
()
#15 0x0000000000431d2c in grpc::ServerBuilder::BuildAndStart() ()
--Type <RET> for more, q to quit, c to continue without paging--
#16 0x000000000041041f in ServerImpl::Run (this=0x7fffffffdfcf0)
    at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:91
#17 0x000000000040bad0 in main (argc=1, argv=0x7fffffffde58)
    at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:261

```

5 fd_create

Thread 1 "greeter_async_s" hit Breakpoint 5, 0x00007ffff6bc4ab0 in epoll_ctl () from /lib/x86_64-linux-gnu/libc.so.6

(gdb) bt

```

#0 0x00007ffff6bc4ab0 in epoll_ctl () from /lib/x86_64-linux-gnu/libc.so.6
#1 0x000000000093b2af in fd_create(int, char const, bool) ()
#2 0x00000000006e6107 in grpc_fd_create(int, char const, bool) ()
#3 0x00000000006f5a54 in clone_port(grpc_tcp_listener, unsigned int) ()
#4 0x00000000006f6133 in tcp_server_start(grpc_tcp_server, std::vector<grpc_pollset,
std::allocator<grpc_pollset> > const, void ()(void, grpc_endpoint, grpc_pollset,
grpc_tcp_server_acceptor), void) ()
#5 0x00000000006f4444 in grpc_tcp_server_start(grpc_tcp_server, std::vector<grpc_pollset,
std::allocator<grpc_pollset> > const, void ()(void, grpc_endpoint, grpc_pollset,
grpc_tcp_server_acceptor), void) ()
#6 0x0000000000663265 in grpc_core::(anonymous namespace)::Chttp2ServerListener::StartListening()
()
#7 0x00000000006631cb in grpc_core::(anonymous
namespace)::Chttp2ServerListener::Start(grpc_core::Server, std::vector<grpc_pollset,
std::allocator<grpc_pollset> > const*) ()
#8 0x0000000000745def in grpc_core::Server::Start() ()
#9 0x0000000000749bd5 in grpc_server_start ()
#10 0x0000000000441605 in grpc::Server::Start(grpc::ServerCompletionQueue**, unsigned long)
()
#11 0x0000000000431e47 in grpc::ServerBuilder::BuildAndStart() ()
#12 0x000000000041041f in ServerImpl::Run (this=0x7fffffffdfcf0)

```

epoll_wait

do_epoll_wait

```
Thread 5 "greeter_async_s" hit Breakpoint 6, 0x00007ffff6bc4ae0 in epoll_wait ()
from /lib/x86_64-linux-gnu/libc.so.6
(gdb) bt
#0  0x00007ffff6bc4ae0 in epoll_wait () from /lib/x86_64-linux-gnu/libc.so.6
#1  0x000000000093c235 in do_epoll_wait(grpc_pollset*, grpc_core::Timestamp) ()
#2  0x000000000093d257 in pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#3  0x000000000006e660c in pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#4  0x000000000006e91e1 in grpc_pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#5  0x000000000007410e9 in cq_next(grpc_completion_queue*, gpr_timespec, void*)
()
#6  0x0000000000074166b in grpc_completion_queue_next ()
#7  0x0000000000042d60c in grpc::CompletionQueue::AsyncNextInternal(void**, bool*,
gpr_timespec) ()
#8  0x0000000000040ca6a in grpc::CompletionQueue::Next (this=0x13d1210,
tag=0x7ffff52b8cc8, ok=0x7ffff52b8cc7)
    at /usr/local/include/grpcpp/impl/codegen/completion_queue.h:185
#9  0x00000000000410c90 in ServerImpl::HandleRpcs (this=0x7fffffdcf0, cq=...)
    at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:240
#10 0x0000000000041074f in ServerImpl::_run (this=0x7fffffdcf0, i=0)
    at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:126
#11 0x000000000004199eb in std::__invoke_impl<void, void (ServerImpl::*)(int),
ServerImpl*, unsigned long> (__f=
    @0x13d1608: (void (ServerImpl::*)(ServerImpl * const, int)) 0x4106ea
<ServerImpl::_run(int)>, __t=@0x13d1600: 0x7fffffdcf0)
    at /usr/local/gcc-11.2/include/c++/11.2.0/bits/invoke.h:74
#12 0x00000000000419944 in std::__invoke<void (ServerImpl::*)(int), ServerImpl*,
unsigned long> (__fn=
    @0x13d1608: (void (ServerImpl::*)(ServerImpl * const, int)) 0x4106ea
<ServerImpl::_run(int)>))
    at /usr/local/gcc-11.2/include/c++/11.2.0/bits/invoke.h:96
#13 0x000000000004197b3 in std::thread::_Invoker<std::tuple<void (ServerImpl::*)(
int), ServerImpl*, unsigned long> >::_M_invoke<0ul, 1ul, 2ul> (this=0x13d15f8)
    at /usr/local/gcc-11.2/include/c++/11.2.0/bits/std_thread.h:253
#14 0x00000000000418f36 in std::thread::_Invoker<std::tuple<void (ServerImpl::*)(
int), ServerImpl*, unsigned long> >::operator() (
    this=0x13d15f8) at /usr/local/gcc-
11.2/include/c++/11.2.0/bits/std_thread.h:260
--Type <RET> for more, q to quit, c to continue without paging--
#15 0x000000000004181a8 in
std::thread::_State_impl<std::thread::_Invoker<std::tuple<void (ServerImpl::*)(
int), ServerImpl*, unsigned long> > >::_M_run (this=0x13d15f0) at
/usr/local/gcc-11.2/include/c++/11.2.0/bits/std_thread.h:211
#16 0x00007ffff7478e80 in std::execute_native_thread_routine (__p=0x13d15f0) at
.././././././././libstdc++-v3/src/c++11/thread.cc:82
#17 0x00007ffff77b56ba in start_thread () from /lib/x86_64-linux-
gnu/libpthread.so.0
#18 0x00007ffff6bc451d in clone () from /lib/x86_64-linux-gnu/libc.so.6
```

SayHello

```
Thread 6 "greeter_async_s" hit Breakpoint 7, ServerImpl::CallData::Proceed
(this=0x7fffe80008c0) at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:173
173             reply_.set_message(prefix + request_.name());
(gdb) bt
#0  ServerImpl::CallData::Proceed (this=0x7fffe80008c0)
    at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:173
#1  0x0000000000410d0d in ServerImpl::HandleRpcs (this=0x7fffffffdfcf0, cq=...)
    at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:242
#2  0x000000000041074f in ServerImpl::_run (this=0x7fffffffdfcf0, i=1)
    at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_async_server_test.cc:126
#3  0x00000000004199eb in std::__invoke_impl<void, void (ServerImpl::*)(int),
ServerImpl*, unsigned long> (__f=
    @0x13d64a8: (void (ServerImpl::*)(ServerImpl * const, int)) 0x4106ea
<ServerImpl::_run(int)>, __t=@0x13d64a0: 0x7fffffffdfcf0)
    at /usr/local/gcc-11.2/include/c++/11.2.0/bits/invoke.h:74
#4  0x0000000000419944 in std::__invoke<void (ServerImpl::*)(int), ServerImpl*,
unsigned long> (__fn=
    @0x13d64a8: (void (ServerImpl::*)(ServerImpl * const, int)) 0x4106ea
<ServerImpl::_run(int)>))
    at /usr/local/gcc-11.2/include/c++/11.2.0/bits/invoke.h:96
#5  0x00000000004197b3 in std::thread::_Invoker<std::tuple<void (ServerImpl::*)(
int), ServerImpl*, unsigned long> >::_M_invoke<0ul, 1ul, 2ul> (this=0x13d6498)
    at /usr/local/gcc-11.2/include/c++/11.2.0/bits/std_thread.h:253
#6  0x0000000000418f36 in std::thread::_Invoker<std::tuple<void (ServerImpl::*)(
int), ServerImpl*, unsigned long> >::operator() (
    this=0x13d6498) at /usr/local/gcc-
11.2/include/c++/11.2.0/bits/std_thread.h:260
--Type <RET> for more, q to quit, c to continue without paging--
#7  0x00000000004181a8 in
std::thread::_State_impl<std::thread::_Invoker<std::tuple<void (ServerImpl::*)(
int), ServerImpl*, unsigned long> > >::_M_run (this=0x13d6490) at
/usr/local/gcc-11.2/include/c++/11.2.0/bits/std_thread.h:211
#8  0x00007ffff7478e80 in std::execute_native_thread_routine (__p=0x13d6490) at
.././.././.././../libstdc++-v3/src/c++11/thread.cc:82
#9  0x00007ffff77b56ba in start_thread () from /lib/x86_64-linux-
gnu/libpthread.so.0
#10 0x00007ffff6bc451d in clone () from /lib/x86_64-linux-gnu/libc.so.6
```

同步

listen

2 grpc_tcp_server_prepare_socket

```
#0 0x00007ffff6bc5310 in listen () from /lib/x86_64-linux-gnu/libc.so.6
#1 0x000000000070c142 in grpc_tcp_server_prepare_socket(grpc_tcp_server*, int,
grpc_resolved_address const*, bool, int*) ()
#2 0x000000000070ba3b in add_socket_to_server(grpc_tcp_server*, int,
grpc_resolved_address const*, unsigned int, unsigned int, grpc_tcp_listener**)
()
#3 0x000000000070be3a in grpc_tcp_server_add_addr(grpc_tcp_server*,
grpc_resolved_address const*, unsigned int, unsigned int, grpc_dualstack_mode*,
grpc_tcp_listener**) ()
#4 0x0000000000709ef9 in add_wildcard_addrs_to_server(grpc_tcp_server*, unsigned
int, int, int*) ()
#5 0x000000000070a80b in tcp_server_add_port(grpc_tcp_server*,
grpc_resolved_address const*, int*) ()
#6 0x0000000000708ed2 in grpc_tcp_server_add_port(grpc_tcp_server*,
grpc_resolved_address const*, int*) ()
#7 0x0000000000677367 in grpc_core::(anonymous
namespace)::Http2ServerListener::Create(grpc_core::Server*,
grpc_resolved_address*, grpc_channel_args*, std::function<grpc_channel_args*
(grpc_channel_args*, grpc_error**)>, int*)::(lambda()#1)::operator()() const
()
#8 0x00000000006775aa in grpc_core::(anonymous
namespace)::Http2ServerListener::Create(grpc_core::Server*,
grpc_resolved_address*, grpc_channel_args*, std::function<grpc_channel_args*
(grpc_channel_args*, grpc_error**)>, int*) ()
#9 0x0000000000678af6 in grpc_core::Http2ServerAddPort(grpc_core::Server*, char
const*, grpc_channel_args*, std::function<grpc_channel_args* (grpc_channel_args*,
grpc_error**)>, int*)::(lambda()#1)::operator()() const ()
#10 0x0000000000679190 in grpc_core::Http2ServerAddPort(grpc_core::Server*, char
const*, grpc_channel_args*, std::function<grpc_channel_args* (grpc_channel_args*,
grpc_error**)>, int*) ()
#11 0x00000000006798c6 in grpc_server_add_http2_port ()
#12 0x0000000000443e9b in grpc::(anonymous
namespace)::InsecureServerCredentialsImpl::AddPortToServer(std::__cxx11::basic_s
tring<char, std::char_traits<char>, std::allocator<char> > const&, grpc_server*)
()
#13 0x000000000045556b in
grpc::Server::AddListeningPort(std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> > const&, grpc::ServerCredentials*)
()
#14 0x000000000044618c in grpc::ServerBuilder::BuildAndStart() ()
--Type <RET> for more, q to quit, c to continue without paging--
#15 0x000000000040b88a in RunServer () at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_server.cc:77
#16 0x000000000040b9a6 in main (argc=1, argv=0x7fffffffde78)
```

accept

grpc_accept4

```
Thread 6 "grpcpp_sync_ser" hit Breakpoint 9, 0x000000000095f5c9 in
grpc_accept4(int, grpc_resolved_address*, int, int) ()
(gdb) bt
#0  0x000000000095f5c9 in grpc_accept4(int, grpc_resolved_address*, int, int) ()
#1  0x000000000070990d in on_read(void*, grpc_error*) ()
#2  0x00000000006fb7f3 in exec_ctx_run(grpc_closure*) ()
#3  0x00000000006fb8f5 in grpc_core::ExecCtx::Flush() ()
#4  0x00000000009518b4 in end_worker(grpc_pollset*, grpc_pollset_worker*,
grpc_pollset_worker**) ()
#5  0x0000000000951d7d in pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#6  0x00000000006fb060 in pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#7  0x00000000006fdc35 in grpc_pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#8  0x0000000000755b7f in cq_next(grpc_completion_queue*, gpr_timespec, void*)
()
#9  0x0000000000756101 in grpc_completion_queue_next ()
#10 0x0000000000473ede in grpc::CompletionQueue::AsyncNextInternal(void**, bool*,
gpr_timespec) ()
#11 0x000000000045ba7e in grpc::CompletionQueue::NextStatus
grpc::CompletionQueue::AsyncNext<gpr_timespec>(void**, bool*, gpr_timespec
const&) ()
#12 0x000000000045a19d in
grpc::Server::SyncRequestThreadManager::PollForWork(void**, bool*) ()
#13 0x000000000046edf4 in grpc::ThreadManager::MainWorkLoop() ()
#14 0x000000000046e6b5 in grpc::ThreadManager::WorkerThread::Run() ()
#15 0x000000000046e56e in
grpc::ThreadManager::WorkerThread::WorkerThread(grpc::ThreadManager*)::
{lambda(void*)#1}::operator()(void*) const ()
#16 0x000000000046e58e in
grpc::ThreadManager::WorkerThread::WorkerThread(grpc::ThreadManager*)::
{lambda(void*)#1}::_FUN(void*) ()
--Type <RET> for more, q to quit, c to continue without paging--
#17 0x0000000000a7a797 in grpc_core::(anonymous
namespace)::ThreadInternalsPosix::ThreadInternalsPosix(char const*, void (*)
(void*), void*, bool*, grpc_core::Thread::Options const&)::
{lambda(void*)#1}::operator()(void*) const ()
#18 0x0000000000a7a7cc in grpc_core::(anonymous
namespace)::ThreadInternalsPosix::ThreadInternalsPosix(char const*, void (*)
(void*), void*, bool*, grpc_core::Thread::Options const&)::
{lambda(void*)#1}::_FUN(void*) ()
#19 0x00007ffff7b756ba in start_thread () from /lib/x86_64-linux-
gnu/libpthread.so.0
#20 0x00007ffff6bc451d in clone () from /lib/x86_64-linux-gnu/libc.so.6
```

epoll_ctl

1 pollset_global_init

```
#0 0x00007ffff6bc4ab0 in epoll_ctl () from /lib/x86_64-linux-gnu/libc.so.6
#1 0x0000000000950418 in pollset_global_init() ()
#2 0x0000000000952c76 in grpc_init_epoll1_linux(bool) ()
#3 0x00000000006fa6d5 in try_engine(char const*) ()
#4 0x00000000006fa913 in grpc_event_engine_init() ()
#5 0x00000000006fd6b6 in iomgr_platform_init() ()
#6 0x00000000006fd5c2 in grpc_iomgr_platform_init() ()
#7 0x000000000095cfe8 in grpc_iomgr_init() ()
#8 0x00000000007585f8 in grpc_init ()
#9 0x0000000000458a1d in grpc::internal::GrpcLibrary::init() ()
#10 0x000000000040f222 in grpc::GrpcLibraryCodegen::GrpcLibraryCodegen
(this=0x13f4b50, call_grpc_init=true)
    at /usr/local/include/grpcpp/impl/codegen/grpc_library.h:48
#11 0x000000000046e453 in grpc::ServerCredentials::ServerCredentials() ()
#12 0x0000000000443f1c in grpc::(anonymous
namespace)::InsecureServerCredentialsImpl::InsecureServerCredentialsImpl() ()
#13 0x0000000000443f5e in grpc::InsecureServerCredentials() ()
#14 0x000000000040b833 in RunServer () at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_server.cc:72
--Type <RET> for more, q to quit, c to continue without paging--
#15 0x000000000040b9a6 in main (argc=1, argv=0x7fffffffde78)
```

3 fd_create

```
#0 0x00007ffff6bc4ab0 in epoll_ctl () from /lib/x86_64-linux-gnu/libc.so.6
#1 0x000000000094fd45 in fd_create(int, char const*, bool) ()
#2 0x00000000006fab5b in grpc_fd_create(int, char const*, bool) ()
#3 0x000000000070bc11 in add_socket_to_server(grpc_tcp_server*, int,
grpc_resolved_address const*, unsigned int, unsigned int, grpc_tcp_listener**)
()
#4 0x000000000070be3a in grpc_tcp_server_add_addr(grpc_tcp_server*,
grpc_resolved_address const*, unsigned int, unsigned int, grpc_dualstack_mode*,
grpc_tcp_listener**) ()
#5 0x0000000000709ef9 in add_wildcard_addrs_to_server(grpc_tcp_server*, unsigned
int, int, int*) ()
#6 0x000000000070a80b in tcp_server_add_port(grpc_tcp_server*,
grpc_resolved_address const*, int*) ()
#7 0x0000000000708ed2 in grpc_tcp_server_add_port(grpc_tcp_server*,
grpc_resolved_address const*, int*) ()
#8 0x0000000000677367 in grpc_core::(anonymous
namespace)::Http2ServerListener::Create(grpc_core::Server*,
grpc_resolved_address*, grpc_channel_args*, std::function<grpc_channel_args*
(grpc_channel_args*, grpc_error**)>, int*)::lambda()#1::operator()() const
()
#9 0x00000000006775aa in grpc_core::(anonymous
namespace)::Http2ServerListener::Create(grpc_core::Server*,
grpc_resolved_address*, grpc_channel_args*, std::function<grpc_channel_args*
(grpc_channel_args*, grpc_error**)>, int*) ()
```



```

#10 0x0000000000678af6 in grpc_core::Http2ServerAddPort(grpc_core::Server*, char
const*, grpc_channel_args*, std::function<grpc_channel_args* (grpc_channel_args*,
grpc_error**)>, int*):{lambda()#1}:operator()() const ()
#11 0x0000000000679190 in grpc_core::Http2ServerAddPort(grpc_core::Server*, char
const*, grpc_channel_args*, std::function<grpc_channel_args* (grpc_channel_args*,
grpc_error**)>, int*) ()
#12 0x00000000006798c6 in grpc_server_add_http2_port ()
#13 0x0000000000443e9b in grpc::(anonymous
namespace)::InsecureServerCredentialsImpl::AddPortToServer(std::__cxx11::basic_s
tring<char, std::char_traits<char>, std::allocator<char> > const&, grpc_server*)
()
#14 0x000000000045556b in
grpc::Server::AddListeningPort(std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> > const&, grpc::ServerCredentials*)
()
#15 0x000000000044618c in grpc::ServerBuilder::BuildAndStart() ()
#16 0x000000000040b88a in RunServer () at /home/lqf/0voice/rpc/grpc-
v1.45.2/examples/cpp/helloworld/greeter_server.cc:77
#17 0x000000000040b9a6 in main (argc=1, argv=0x7fffffffde78)

```

on_read

```

Thread 6 "grpcpp_sync_ser" hit Breakpoint 12, 0x00007ffff6bc4ab0 in epoll_ctl ()
from /lib/x86_64-linux-gnu/libc.so.6
(gdb) bt
#0 0x00007ffff6bc4ab0 in epoll_ctl () from /lib/x86_64-linux-gnu/libc.so.6
#1 0x000000000094fd45 in fd_create(int, char const*, bool) ()
#2 0x00000000006fab5b in grpc_fd_create(int, char const*, bool) ()
#3 0x0000000000709bed in on_read(void*, grpc_error*) ()
#4 0x00000000006fb7f3 in exec_ctx_run(grpc_closure*) ()
#5 0x00000000006fb8f5 in grpc_core::ExecCtx::Flush() ()
#6 0x00000000009518b4 in end_worker(grpc_pollset*, grpc_pollset_worker*,
grpc_pollset_worker**) ()
#7 0x0000000000951d7d in pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#8 0x00000000006fb060 in pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#9 0x00000000006fdc35 in grpc_pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#10 0x0000000000755b7f in cq_next(grpc_completion_queue*, gpr_timespec, void*)
()
--Type <RET> for more, q to quit, c to continue without paging--
#11 0x0000000000756101 in grpc_completion_queue_next ()
#12 0x0000000000473ede in grpc::CompletionQueue::AsyncNextInternal(void**, bool*,
gpr_timespec) ()
#13 0x000000000045ba7e in grpc::CompletionQueue::NextStatus
grpc::CompletionQueue::AsyncNext<gpr_timespec>(void**, bool*, gpr_timespec
const&) ()
#14 0x000000000045a19d in
grpc::Server::SyncRequestThreadManager::PollForWork(void**, bool*) ()
#15 0x000000000046edf4 in grpc::ThreadManager::MainWorkLoop() ()
#16 0x000000000046e6b5 in grpc::ThreadManager::WorkerThread::Run() ()

```

```

#17 0x000000000046e56e in
grpc::ThreadManager::WorkerThread::WorkerThread(grpc::ThreadManager*)::
{lambda(void*)#1}::operator()(void*) const ()
#18 0x000000000046e58e in
grpc::ThreadManager::WorkerThread::WorkerThread(grpc::ThreadManager*)::
{lambda(void*)#1}::_FUN(void*) ()
#19 0x0000000000a7a797 in grpc_core::(anonymous
namespace)::ThreadInternalsPosix::ThreadInternalsPosix(char const*, void (*)
(void*), void*, bool*, grpc_core::Thread::Options const&)::
{lambda(void*)#1}::operator()(void*) const ()
#20 0x0000000000a7a7cc in grpc_core::(anonymous
namespace)::ThreadInternalsPosix::ThreadInternalsPosix(char const*, void (*)
(void*), void*, bool*, grpc_core::Thread::Options const&)::
{lambda(void*)#1}::_FUN(void*) ()
#21 0x00007ffff77b56ba in start_thread () from /lib/x86_64-linux-
gnu/libpthread.so.0
#22 0x00007ffff6bc451d in clone () from /lib/x86_64-linux-gnu/libc.so.6

```

epoll_wait

4 grpcpp_sync_ser

```

Thread 6 "grpcpp_sync_ser" hit Breakpoint 4, 0x00007ffff6bc4ae0 in epoll_wait ()
from /lib/x86_64-linux-gnu/libc.so.6
(gdb) bt
#0 0x00007ffff6bc4ae0 in epoll_wait () from /lib/x86_64-linux-gnu/libc.so.6
#1 0x0000000000950ccb in do_epoll_wait(grpc_pollset*, grpc_core::Timestamp) ()
#2 0x0000000000951ced in pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#3 0x00000000006fb060 in pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#4 0x00000000006fdc35 in grpc_pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#5 0x0000000000755b7f in cq_next(grpc_completion_queue*, gpr_timespec, void*)
()
#6 0x0000000000756101 in grpc_completion_queue_next ()
#7 0x0000000000473ede in grpc::CompletionQueue::AsyncNextInternal(void**, bool*,
gpr_timespec) ()
#8 0x000000000045ba7e in grpc::CompletionQueue::NextStatus
grpc::CompletionQueue::AsyncNext<gpr_timespec>(void**, bool*, gpr_timespec
const&) ()
#9 0x000000000045a19d in
grpc::Server::SyncRequestThreadManager::PollForWork(void**, bool*) ()
#10 0x000000000046edf4 in grpc::ThreadManager::MainWorkLoop() ()
#11 0x000000000046e6b5 in grpc::ThreadManager::WorkerThread::Run() ()
#12 0x000000000046e56e in
grpc::ThreadManager::WorkerThread::WorkerThread(grpc::ThreadManager*)::
{lambda(void*)#1}::operator()(void*) const ()

```

```

#13 0x000000000046e58e in
grpc::ThreadManager::workerThread::workerThread(grpc::ThreadManager*)::
{lambda(void*)#1}::_FUN(void*) ()
#14 0x0000000000a7a797 in grpc_core::(anonymous
namespace)::ThreadInternalsPosix::ThreadInternalsPosix(char const*, void (*)
(void*), void*, bool*, grpc_core::Thread::Options const&)::
{lambda(void*)#1}::operator()(void*) const ()
#15 0x0000000000a7a7cc in grpc_core::(anonymous
namespace)::ThreadInternalsPosix::ThreadInternalsPosix(char const*, void (*)
(void*), void*, bool*, grpc_core::Thread::Options const&)::
{lambda(void*)#1}::_FUN(void*) ()
#16 0x00007ffff7b56ba in start_thread () from /lib/x86_64-linux-
gnu/libpthread.so.0
#17 0x00007ffff6bc451d in clone () from /lib/x86_64-linux-gnu/libc.so.6

```

4

```

#0 0x00007ffff6bc4ae0 in epoll_wait () from /lib/x86_64-linux-gnu/libc.so.6
#1 0x0000000000950ccb in do_epoll_wait(grpc_pollset*, grpc_core::Timestamp) ()
#2 0x0000000000951ced in pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#3 0x00000000006fb060 in pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#4 0x00000000006fdc35 in grpc_pollset_work(grpc_pollset*, grpc_pollset_worker**,
grpc_core::Timestamp) ()
#5 0x0000000000755b7f in cq_next(grpc_completion_queue*, gpr_timespec, void*)
()
#6 0x0000000000756101 in grpc_completion_queue_next ()
#7 0x0000000000473ede in grpc::CompletionQueue::AsyncNextInternal(void**, bool*,
gpr_timespec) ()
#8 0x000000000045ba7e in grpc::CompletionQueue::NextStatus
grpc::CompletionQueue::AsyncNext<gpr_timespec>(void**, bool*, gpr_timespec
const&) ()
#9 0x000000000045a19d in
grpc::Server::SyncRequestThreadManager::PollForWork(void**, bool*) ()
#10 0x000000000046edf4 in grpc::ThreadManager::MainWorkLoop() ()
#11 0x000000000046e6b5 in grpc::ThreadManager::workerThread::Run() ()
#12 0x000000000046e56e in
grpc::ThreadManager::workerThread::workerThread(grpc::ThreadManager*)::
{lambda(void*)#1}::operator()(void*) const ()
#13 0x000000000046e58e in
grpc::ThreadManager::workerThread::workerThread(grpc::ThreadManager*)::
{lambda(void*)#1}::_FUN(void*) ()
#14 0x0000000000a7a797 in grpc_core::(anonymous
namespace)::ThreadInternalsPosix::ThreadInternalsPosix(char const*, void (*)
(void*), void*, bool*, grpc_core::Thread::Options const&)::
{lambda(void*)#1}::operator()(void*) const ()
#15 0x0000000000a7a7cc in grpc_core::(anonymous
namespace)::ThreadInternalsPosix::ThreadInternalsPosix(char const*, void (*)
(void*), void*, bool*, grpc_core::Thread::Options const&)::
{lambda(void*)#1}::_FUN(void*) ()

```

```
#16 0x00007ffff7b56ba in start_thread () from /lib/x86_64-linux-  
gnu/libpthread.so.0  
#17 0x00007ffff6bc451d in clone () from /lib/x86_64-linux-gnu/libc.so.6
```

SayHello

```
Thread 7 "grpcpp_sync_ser" hit Breakpoint 6, GreeterServiceImpl::SayHello  
(this=0x7fffffffdfcf0, context=0x7fffec007798, request=0x7fffec00c820,  
reply=0x7ffff42b69a0) at /home/lqf/0voice/rpc/grpc-  
v1.45.2/examples/cpp/helloworld/greeter_server.cc:45  
45         std::string prefix("Hello ");  
(gdb) bt  
#0 GreeterServiceImpl::SayHello (this=0x7fffffffdfcf0, context=0x7fffec007798,  
request=0x7fffec00c820, reply=0x7ffff42b69a0)  
    at /home/lqf/0voice/rpc/grpc-  
v1.45.2/examples/cpp/helloworld/greeter_server.cc:45  
#1 0x00000000040c577 in operator() (__closure=0x13f1838,  
service=0x7fffffffdfcf0, ctx=0x7fffec007798, req=0x7fffec00c820,  
resp=0x7ffff42b69a0) at /home/lqf/0voice/rpc/grpc-  
v1.45.2/examples/cpp/helloworld/build/helloworld.grpc.pb.cc:70  
#2 0x00000000040dc26 in std::__invoke_impl<grpc::Status,  
helloworld::Greeter::Service::Service()::<lambda(helloworld::Greeter::Service*,  
grpc::ServerContext*, const helloworld::HelloRequest*,  
helloworld::HelloReply*)>&, helloworld::Greeter::Service*, grpc::ServerContext*,  
const helloworld::HelloRequest*, helloworld::HelloReply*>(std::__invoke_other,  
struct {...} &) (__f=...)  
    at /usr/local/gcc-11.2/include/c++/11.2.0/bits/invoke.h:61  
#3 0x00000000040d742 in std::__invoke_r<grpc::Status,  
helloworld::Greeter::Service::Service()::<lambda(helloworld::Greeter::Service*,  
grpc::ServerContext*, const helloworld::HelloRequest*,  
helloworld::HelloReply*)>&, helloworld::Greeter::Service*, grpc::ServerContext*,  
const helloworld::HelloRequest*, helloworld::HelloReply*>(struct {...} &)  
    (__fn=...)  
    at /usr/local/gcc-11.2/include/c++/11.2.0/bits/invoke.h:143  
#4 0x00000000040d30a in  
std::_Function_handler<grpc::Status(helloworld::Greeter::Service*,  
grpc::ServerContext*, const helloworld::HelloRequest*, helloworld::HelloReply*),  
helloworld::Greeter::Service::Service()::<lambda(helloworld::Greeter::Service*,  
grpc::ServerContext*, const helloworld::HelloRequest*, helloworld::HelloReply*)>>  
>::_M_invoke(const std::_Any_data &, helloworld::Greeter::Service *&&,  
grpc::ServerContext *&&, const helloworld::HelloRequest *&&,  
helloworld::HelloReply *&&) (__functor=...,  
    __args#0=@0x7ffff42b6878: 0x7fffffffdfcf0, __args#1=@0x7ffff42b6870:  
0x7fffec007798, __args#2=@0x7ffff42b6868: 0x7fffec00c820,  
    __args#3=@0x7ffff42b6860: 0x7ffff42b69a0) at /usr/local/gcc-  
11.2/include/c++/11.2.0/bits/std_function.h:292
```

```

#5 0x000000000041fd80 in std::function<grpc::Status
(helloworld::Greeter::Service*, grpc::ServerContext*, helloworld::HelloRequest
const*, helloworld::HelloReply*)>::operator()(helloworld::Greeter::Service*,
grpc::ServerContext*, helloworld::HelloRequest const*, helloworld::HelloReply*)
const (this=0x13f1838, __args#0=0x7fffffffdfcf0, __args#1=0x7fffec007798,
__args#2=0x7fffec00c820,
__args#3=0x7ffff42b69a0) at /usr/local/gcc-
11.2/include/c++/11.2.0/bits/std_function.h:560
#6 0x000000000041dc00 in
grpc::internal::RpcMethodHandler<helloworld::Greeter::Service,
helloworld::HelloRequest, helloworld::HelloReply, google::protobuf::MessageLite,
google::protobuf::MessageLite>::RunHandler(grpc::internal::MethodHandler::Handle
rParameter const&)::{lambda()#1}::operator()() const (__closure=0x7ffff42b6a10)
at /usr/local/include/grpcpp/impl/codegen/method_handler.h:116
#7 0x000000000041fdb5 in
grpc::internal::CatchingFunctionHandler<grpc::internal::RpcMethodHandler<hellowo
rld::Greeter::Service, helloworld::HelloRequest, helloworld::HelloReply,
google::protobuf::MessageLite,
google::protobuf::MessageLite>::RunHandler(grpc::internal::MethodHandler::Handle
rParameter const&)::{lambda()#1}>
(grpc::internal::RpcMethodHandler<helloworld::Greeter::Service,
helloworld::HelloRequest, helloworld::HelloReply, google::protobuf::MessageLite,
google::protobuf::MessageLite>::RunHandler(grpc::internal::MethodHandler::Handle
rParameter const&)::{lambda()#1}&&) (handler=...) at
/usr/local/include/grpcpp/impl/codegen/method_handler.h:44
--Type <RET> for more, q to quit, c to continue without paging--
#8 0x000000000041dc93 in
grpc::internal::RpcMethodHandler<helloworld::Greeter::Service,
helloworld::HelloRequest, helloworld::HelloReply, google::protobuf::MessageLite,
google::protobuf::MessageLite>::RunHandler (this=0x13f1830, param=...)
at /usr/local/include/grpcpp/impl/codegen/method_handler.h:113
#9 0x0000000000459cb8 in
grpc::Server::SyncRequest::ContinueRunAfterInterception() ()
#10 0x0000000000459ac5 in
grpc::Server::SyncRequest::Run(std::shared_ptr<grpc::Server::GlobalCallbacks>
const&, bool) ()
#11 0x000000000045a2c5 in grpc::Server::SyncRequestThreadManager::DoWork(void*,
bool, bool) ()
#12 0x000000000046f01f in grpc::ThreadManager::MainWorkLoop() ()
#13 0x000000000046e6b5 in grpc::ThreadManager::WorkerThread::Run() ()
#14 0x000000000046e56e in
grpc::ThreadManager::WorkerThread::WorkerThread(grpc::ThreadManager*)::
{lambda(void*)#1}::operator()(void*) const ()
#15 0x000000000046e58e in
grpc::ThreadManager::WorkerThread::WorkerThread(grpc::ThreadManager*)::
{lambda(void*)#1}::_FUN(void*) ()
#16 0x0000000000a7a797 in grpc_core::(anonymous
namespace)::ThreadInternalsPosix::ThreadInternalsPosix(char const*, void (*)
(void*), void*, bool*, grpc_core::Thread::Options const&)::
{lambda(void*)#1}::operator()(void*) const ()
#17 0x0000000000a7a7cc in grpc_core::(anonymous
namespace)::ThreadInternalsPosix::ThreadInternalsPosix(char const*, void (*)
(void*), void*, bool*, grpc_core::Thread::Options const&)::
{lambda(void*)#1}::_FUN(void*) ()

```

```
#18 0x00007ffff77b56ba in start_thread () from /lib/x86_64-linux-  
gnu/libpthread.so.0  
#19 0x00007ffff6bc451d in clone () from /lib/x86_64-linux-gnu/libc.so.6
```