A Quick Start for ffnet

Athrun Arthur

September 17, 2012

Contents

1 Introduction

1.1 Why ffnet

Network programming is a very complicated thing. Of course it's simple to write a simple ping-pong network application using socket. But you have to consider many other possible situations in productive applications.

Consider ping-pong as an example. There is a server which replys pong message when receives ping message, and a client which replys ping message when receives pong message. To make the infinite loop start, the client need to send the first ping message when the connection is established. Now let's see what you need to consider if ping-pong is a product which means avalibility, scalibility, strong and easy to extend.

- Connection management. It's obvious as there may be multiple clients. When a client is offline, the server need to know that.
- Avalibility. Network is complex partly because you may receive any possible messages, legal or illegal. You must distinguish those illegal messages from raw messages. This means you may need to handle some hostile attacks, like DDoS.
- Performance. It's a good practice to consider response time in network programming although ping-pong is simple. Maybe you know proactor pattern, reactor pattern, asynchronized I/O (like boost.asio), parallel programming. But you may turn a blind to these solutions because of complexity
- Maintainable. Code refactoring is a normal thing in network programming. There are many situations recall code refactoring, for example, new business requirements, unexpected network behaviors, performance tunes and security ensurance. Again, you turn a blind to possible design patterns because of complexity.
- Configurable. Another burdern to adjust very network enviornments.

There are many network libraries which aim to bring simple and powerful network programming, like boost.asio, protocol buffer from Google, ACE and mudo. But ffnet is aim to provide higher level network programming enviornment with parallel, asynchronization, security , debugging and configurable features.

1.2 ffnet is ...

ffnet is a opensource framework for network programming in C++. It's based on boost.asio and provide network management, package serialization and deserialization, asynchronization, security, debugging and configurable features. Now ffnet is still under heavy development.

1.3 Build ffnet

ffnet uses CMake to organize its source code and it depends on Boost (1.40 or higher). Suppose you have got the source code of ffnet and the directory is */ffnet/root/dir/*. Here are the steps you need to build it.

1. cd */ffnet/root/dir/build*
2. cmake ../
3. make

If you didn't install Boost into system path, you will get an error in the second step. In this case, you need to specify path of your Boost in */ffnet/root/dir/CMakeLists.txt*, like this.

```
set(CMAKE_INCLUDE_PATH ${CMAKE_INCLUDE_PATH}
"/home/athrun/boost_1_46_1")
set(CMAKE_LIBRARY_PATH ${CMAKE_LIBRARY_PATH}
"/home/athrun/boost_1_46_1/stage/lib")
```

You can find two generated file in */ffnet/root/dir/lib* now if you have built ffnet successfully. One is a static library and the other is a shared library.

1.4 An example

A ping_pong example uses the ffnet framework to realize a infinite ping_pong process. Suppose you have got the source code of ping_pong and the directory is */ffnet/root/dir/example/ping_pong/*. Here are the steps you need to build it.

1. cd */ffnet/root/dir/example/ping_pong/build*
2. cmake ../
3. make

If you didn't install ffnet or Boost into system path, you will get an error in the second step. In this case, you need to specify path of your ffnet and Boost in */ffnet/root/dir/example/ping_pong/CMakeLists.txt*, like this.

```
set(CMAKE_INCLUDE_PATH ${CMAKE_INCLUDE_PATH}
"/home/athrun/boost_1_46_1")
set(CMAKE_LIBRARY_PATH ${CMAKE_LIBRARY_PATH}
"/home/athrun/boost_1_46_1/stage/lib")
set(FFNET_ROOT_DIR ${PROJECT_SOURCE_DIR}/../../network)
```

You can find two executable programs in */ffnet/root/dir/example/ping_pong/build* now if you have built ping_pong successfully. One is client and the other is server.

Before running, you need to copy clnt_net_conf.ini and svr_net_conf.ini from */ffnet/root/dir/example/ping_pong/* into */ffnet/root/dir/example/ping_pong/build*. Then you can run the example by using the command below.

1. ./server
2. ./client