proxygen需要c++14的特性支持，因此centos6.5环境下要编译proxygen需升级gcc，本机gcc版本升级为gcc4.9.2。安装路径为/home/zhangmaolin/usr/local/gcc\_4\_9，gcc4.9.2的环境变量声明如下：

export PATH=/home/zhangmaolin/usr/local/gcc\_4\_9/bin:/home/zhangmaolin/usr/local/bin:$PATH

export LD\_LIBRARY\_PATH=/home/zhangmaolin/usr/local/gcc\_4\_9/lib:/home/zhangmaolin/usr/local/gcc\_4\_9/lib64:$LD\_LIBRARY\_PATH

export LIBRARY\_PATH=/home/zhangmaolin/usr/local/gcc\_4\_9/lib:/home/zhangmaolin/usr/local/gcc\_4\_9/lib64:$LIBRARY\_PATH

export C\_INCLUDE\_PATH=/home/zhangmaolin/usr/local/gcc\_4\_9/include:$C\_INCLUDE\_PATH

export CPLUS\_INCLUDE\_PATH=/home/zhangmaolin/usr/local/gcc\_4\_9/include:$CPLUS\_INCLUDE\_PATH

export CC=/home/zhangmaolin/usr/local/gcc\_4\_9/bin/gcc

export CXX=/home/zhangmaolin/usr/local/gcc\_4\_9/bin/gcc

本机己安装boost1.53.0，安装路径为：/home/zhangmaolin/usr/local

其有些依赖库使用cmake编译，并且也需c++11及c++14的支持，因此也需安装cmake，版本>=2.8.12。

proxygen写了deps.sh，在ubuntu上可以自动编译运行，但是在centos6.5上不行，依据deps.sh内的编译执行顺序，需下载安装google-glog、google-gflags、gperf、double-conversion、folly/folly、wangle/wangle，最后编译proxygen。

编译google-glog时，需要高版本的autoconf和automake，因此需先安装automake-1.14.tar.gz和autoconf-2.69.tar.gz。

编译folly时，需要两个autoreconf的相关库libtool和pkg-config，因此下载安装libtool-2.4.6.tar.xz和pkg-config-0.28.tar.gz。

1. 编译安装automake-1.14.tar.gz和autoconf-2.69.tar.gz

automake-1.14.tar.gz下载网址：<https://ftp.gnu.org/gnu/autoconf/autoconf-2.69.tar.gz>

解压，./configure –prefix=/home/zhangmaolin/usr/local，make && make install

automake-1.14.tar.gz下载网址：<https://ftp.gnu.org/gnu/autoconf/autoconf-2.69.tar.gz>

解压，./configure –prefix=/home/zhangmaolin/usr/local，make && make install

编译完成后，修改环境变量PATH，export PATH=/home/zhangmaolin/usr/local/bin:$PATH

执行autoconf –version，版本应为2.69；

1. 编译安装google-glog

glog下载地址：<https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/google-glog/glog-0.3.3.tar.gz>

./configure --prefix=/home/zhangmaolin/usr/local/

make & make install

注：使用git下载的glog会报错：relocation R\_X86\_64\_32S against `.rodata' can not be used when making a shared object; recompile with –fPIC，暂时未找到原因；

1. 编译安装google-gflags

下载源码：<https://github.com/gflags/gflags.git>

mkdir build && cd build

cmake .. -DCMAKE\_INSTALL\_PREFIX=/home/zhangmaolin/usr/local -DBUILD\_SHARED\_LIBS=ON

make && make install

注：若不加-DBUILD\_SHARED\_LIBS=ON，则后面编译folly时会报错：

configure: error: "gflags invalid, see config.log for details"

1. 编译安装gperf

下载源码gperf-3.1.tar.gz，网址：<https://ftp.gnu.org/pub/gnu/gperf/gperf-3.1.tar.gz>

解压，./configure –prefix=/home/zhangmaolin/usr/local，make && make install

编译安装double-conversion

git clone https://github.com/floitsch/double-conversion.git double-conversion

cd double-conversion

mkdir build && cd build

cmake .. -DBUILD\_SHARED\_LIBS=ON -DCMAKE\_INSTALL\_PREFIX=/home/zhangmaolin/usr/local

make && make install

1. 编译安装folly/folly

首先需先编译两个相关库

libtool-2.4.6.tar.xz(<https://ftp.gnu.org/gnu/libtool/libtool-2.4.6.tar.xz>)

解压，./configure --prefix=/home/zhangmaolin/usr/local/, make && make install

若无该库，folly运行autoreconf –ivf时会报错：error: possibly undefined macro: AM\_PROG\_LIBTOOL

pkg-config-0.28.tar.gz（<https://pkg-config.freedesktop.org/releases/pkg-config-0.28.tar.gz>）

解压，./configure --prefix=/home/zhangmaolin/usr/local/, make && make install

若无该库，folly运行autoreconf –ivf时会报错：configure:17178: error: possibly undefined macro: AC\_MSG\_NOTICE

安装folly需要高版本的gcc，因此需先重置与之相关的环境变量，具体见本文开头。

下载源码：git clone <https://github.com/facebook/folly>

autoreconf –ivf

./configure --prefix=/home/zhangmaolin/usr/local/ --with-boost-libdir=/home/zhangmaolin/usr/local/lib

make && make install

编译wangle

注：wangle编译需要使用高版本gcc，因此在wanglecmake前，需先声明环境变量CC及CXX指向高版本gcc，具体可见开头环境变量的声明。

下载源码：git clone <https://github.com/facebook/wangle>

mkdir build && cd build

cmake .. -DCMAKE\_INSTALL\_PREFIX=/home/zhangmaolin/usr/local -DBUILD\_TESTS=off

make && make install

1. 编译proxygen

下载源码：git clone https://github.com/facebook/proxygen.git

autoreconf –ivf

./configure --prefix=/home/zhangmaolin/usr/local

make && make install

注：因为之前的folly等库未装在系统默认目录，./configure时可能会报错找不到folly等库，修改configure文件将报错的代码注释即可

.configure时报错：

syntax error near unexpected token `proxygen-config.h,'

`AX\_PREFIX\_CONFIG\_H(proxygen-config.h, proxygen, config.h)'

解决方案，注释掉相关语句

make时会报错/home/zhangmaolin/usr/local/lib/libfolly.so: undefined reference to `vtable for boost::program\_options::error\_with\_option\_name'

解决方案为进入/usr/lib64，mkidr boost\_bak，将所有libboost\*相关的文件move到boost\_bak文件夹内