**Proxygen在centos上安装步骤**

**三、更新：2017-06-02 安装cpp-hocon、grpc**

**安装leatherman 0.99.0**

# git clone <https://github.com/puppetlabs/leatherman.git>  
# git checkout 0.99.0 && git checkout -b 0.99.0  
# mkdir build && cd build  
# cmake ..  
# make  
# sudo make install

(注意：默认安装的lib命名不规范，需将leatherman相关的包重命名为带lib前缀的规范命名方式，否则程序连接时找不到相关的静态lib)

**安装cpp-hocon 0.1.5**

# git checkout 0.1.5 && git checkout -b 0.1.5  
# mkdir release && cd release  
# cmake ..  
# make  
# sudo make install

**二、更新：2017-05-31 升级boost到1.63.0**

**备份并删除之前的boost和其它依赖包**

# cd /usr/local/include

# mkdir -p bak

# mv boost glog folly wangle proxygen bak/

# cd /usr/local/lib

# mkdir -p bak

# mv libboost\* libfolly\* libglog\* libwangle.a libproxygen\* bak/

**升级boost到1.63.0版本：**

1. 升级boost 从官网上下载boost 1.63.0
2. 备份 /usr/local/include/boost ： mv /usr/local/include/boost /usr/local/include/boost.bak
3. 备份 /usr/local/lib/libboost\* ： mkdir -p /usr/local/lib/bak && mv /usr/local/lib/libboost\* /usr/local/lib/bak/
4. 编译安装boost，最好直接切换成root用户编译安装，确保使用gcc 4.9 （使用admin账户运行./b2 install 时，环境变量中配置的gcc路径会失效，可能会调用gcc4.8，因此最好切换成root账户，并且显示设置CC的环境变量。）

declare -x CC="/usr/local/bin/gcc"

./bootstrap --prefix /usr/local

**升级glog到0.3.5版本：**

1. 下载glog ： wget <https://github.com/google/glog/archive/v0.3.5.tar.gz>

# tar -zxvf v0.3.5.tar.gz

# cd glog-0.3.5/

# ./configure

# make && make install

**按照原始步骤重新编译安装：folly, wangle, proxygen**

安装完成后，可以重新编译ps的源码。

**一、更新：首个版本**

**Centos6.5上安装ps**

1. **编译安装 gmp**   <ftp://ftp.gnu.org/gnu/gmp>

# tar -zxvf gmp-5.1.3.tar.gz

# cd gmp-5.1.3

# ./configure

# make && make install

1. **编译安装 mpfr** <ftp://ftp.gnu.org/gnu/mpfr>

# tar -zxvf mpfr-3.1.2.tar.gz

# cd mpfr-3.1.2

# ./configure

# make && make install

1. **编译安装 mpc** <http://www.multiprecision.org/mpc>

# tar -zxvf mpc-1.0.tar.gz

# cd mpc-1.0

# ./configure

# make && make install

1. **升级gcc到[gcc-4.9.2](http://blog.csdn.net/szq123456123/article/details/45170923)**

首先下载源代码

Wget <http://ftp.gnu.org/gnu/gcc/gcc-4.9.2/gcc-4.9.2.tar.bz2>

tar -jxvf gcc-4.9.2.tar.bz2

cd gcc-4.9.2

./contrib/download\_prerequisites

cd ..

mkdir gcc-build-4.9.2 //**建立编译输出目录**

cd gcc-build-4.9.2 //进入此目录，执行以下命令，生成makefile文件

../gcc-4.9.2/configure --enable-checking=release --enable-languages=c,c++ --disable-multilib

make -j4

make install

**安装遇到的问题**:

1. 安装gcc：[configure: error: cannot compute suffix of object files: cannot compile](http://blog.csdn.net/testcs_dn/article/details/45437149)

解决方法：export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/usr/local/lib

参考：<http://blog.csdn.net/szq123456123/article/details/45170923>

1. **安装boost1.53.0（废弃，后续己升级）**

下载地址：<http://www.boost.org/users/history/version_1_53_0.html>

解压执行./bootstrap.sh

./b2

./b2 install

1. **安装cmake 3.8.0**，wget <https://cmake.org/files/v3.8>

    免安装版本，设置PATH,和cmake命令建立软连接

问题：**[/usr/lib/libstdc++.so.6: version `GLIBCXX\_3.4.15' not found](http://www.cnblogs.com/weinyzhou/p/4983306.html)**

<http://www.cnblogs.com/weinyzhou/p/4983306.html>

1. **安装automake-1.14.tar.gz** <http://mirror.bjtu.edu.cn/gnu/automake/>
2. **安装autoconf-2.69.tar.gz** <http://mirror.bjtu.edu.cn/gnu/autoconf/>
3. **安装libtool-2.4.6.tar.xz：**<http://mirror.bjtu.edu.cn/gnu/libtool/>
4. **安装pkg-config-0.28.tar.gz** [**https://pkg-config.freedesktop.org/releases/**](https://pkg-config.freedesktop.org/releases/) **done**

./configure --with-internal-glib

make &&make install

1. **安装google-glog**

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

1. **安装google-gflags2.2.0**  <https://github.com/gflags/gflags.git>

Cmake  -DBUILD\_SHARED\_LIBS=ON

1. **安装gperf-3.1.tar.gz**  <https://ftp.gnu.org/pub/gnu/gperf/gperf-3.1.tar.gz>
2. **编译安装double-conversion 1.1.5** <https://github.com/google/double-conversion/releases>

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** [**v2017.03.06.00**](https://github.com/facebook/folly/releases/tag/v2017.03.06.00) git clone <https://github.com/facebook/folly>

    autoreconf -ivf

  ./configure

  make

  make check

  sudo make install

1. **安装libevent** [**libevent-2.1.8-stable.tar.gz**](https://github.com/libevent/libevent/releases/download/release-2.1.8-stable/libevent-2.1.8-stable.tar.gz) <http://libevent.org/>
2. **编译wangle** [**v2017.03.06.00**](https://github.com/facebook/wangle/releases/tag/v2017.03.06.00)git clone <https://github.com/facebook/wangle>

更改环境变量： export C\_INCLUDE\_PATH=/usr/local/lib:$C\_INCLUDE\_PATH

export CPLUS\_INCLUDE\_PATH=/usr/local/lib:$CPLUS\_INCLUDE\_PATH

export CC=/usr/local/bin/gcc

export CXX=/usr/local/bin/gcc

1. **安装libcap\_devel(如何机器上没有)可以直接yum install**
2. **编译安装proxygen**

git clone <https://github.com/facebook/proxygen.git>

autoreconf –ivf

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

make && make install

问题：没有外网权限需先下载googletest-release-1.8.0.zip包，重命名为release-1.8.0.zip，放到proxygen/lib/test目录下，同时注释proxygen/lib/test/Makefile的第869行：

**proxygen/lib/test/Makefile文件**

line 868      release-1.8.0.zip:

line869          #wget <https://github.com/google/googletest/archive/release-1.8.0.zip>

**wget** [**https://github.com/google/googletest/archive/release-1.8.0.zip**](https://github.com/google/googletest/archive/release-1.8.0.zip)