自动驾驶中的SLAM技术环境配置(18.04)

1 前言

本pdf记录的是本人使用**ubuntu 18.04**运行课程代码时，环境配置的过程以及一些bug的解决思路（见“部分问题解决思路”），希望能给到同学们一些帮助。如果同学们使用的是**ubuntu 20.04**，请参考高博的github地址：https://github.com/gaoxiang12/slam_in_autonomous_driving，里面有详细说明。或者使用专门为本课程定制的docker：https://github.com/zijiechenrobotics/slam_in_autonomous_driving_docker（推荐）

环境配置是十分烦人的事情，请同学们有多一点耐心，有问题多在群里交流、讨论~💪

2 安装gcc-9

本课程用到c++17的新特性以及TBB库进行并发加速，因此需要适配gcc编译器

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
sudo apt update
sudo apt-get install gcc-9
sudo apt-get install g++-9

# 实现版本切换
sudo update-alternatives --remove-all gcc
sudo update-alternatives --remove-all g++

# 命令最后的10是优先级，如果使用auto选择模式，系统将默认使用优先级高的
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-9 10
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-9 10

# 切换版本
sudo update-alternatives --config gcc
sudo update-alternatives --config g++

# 检查版本
gcc -v
g++ -v
```

3 关于TBB

若本机已经通过源码安装了TBB2018及以上版本，可以修改cmake/packages.cmake中第74行为

```
if(BUILD_WITH_UBUNTU1804)
  find_package(TBB REQUIRED)
  set(third_party_libs
    ${catkin_LIBRARIES}
    ${g2o_libs}
    ${OpenCV_LIBS}
```

```
    ${PCL_LIBRARIES}
    ${Pangolin_LIBRARIES}
    glog gflags
    ${yaml-cpp_LIBRARIES}
    yaml-cpp
    TBB::tbb
  )
else()
```

取消自动安装TBB2019, 从而避免环境存在多个TBB版本

4 安装ROS

本课程主要使用ROS来解析bag数据集，安装请参考ROS官网：

http://wiki.ros.org/melodic/Installation/Ubuntu

5 安装Pangolin

本课程将使用Pangolin进行可视化，而不是ROS中的rviz

```
git clone https://github.com/stevenlovegrove/Pangolin.git
cd Pangolin
mkdir build
cd build
cmake ..
make -j8 (-j8 表示用8个core编译, 根据自己电脑性能给定)
sudo make install
sudo ldconfig # 执行一下，不然有error while loading shared libraries: libpango_windowing.so
```

6 安装其余依赖

缺哪个装那个

```
sudo apt-get install ros-melodic-pcl-ros ros-melodic-velodyne-msgs libopencv-dev libgoogle-glog-dev libeigen3-dev libsuitesparse-dev libpcl-dev libyaml-cpp-dev libgtest-dev
```

7 编译程序

编译g2o时可以不用新建 `build` ，直接 `cmake .` ，再 `make -j` 也可以

```
git clone https://github.com/gaoxiang12/slam_in_autonomous_driving

# 首先编译g2o
```

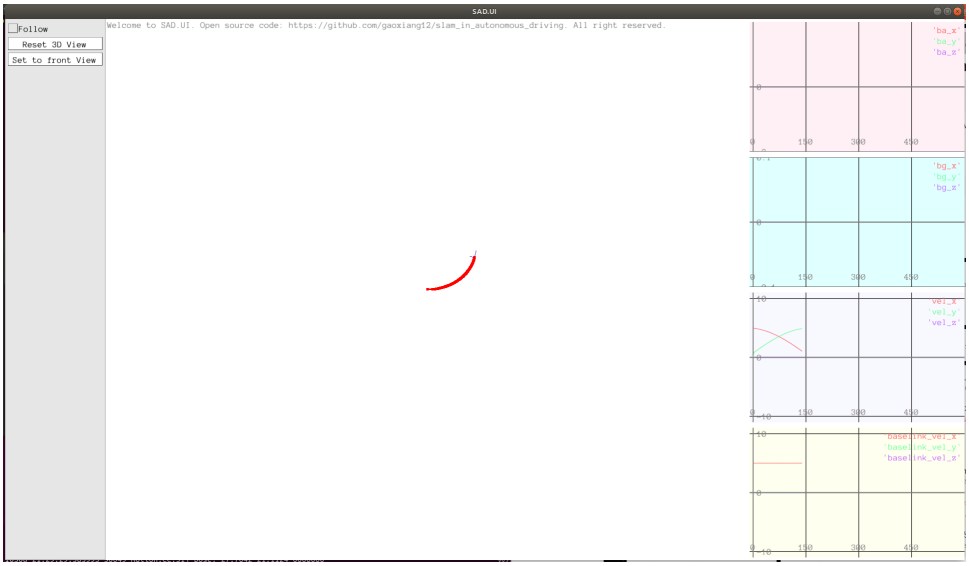
```
cd slam_in_autonomous_driving/thirdparty/g2o/
mkdir build
cd build
cmake ..
make -j8 (-j8 表示用8个core编译, 根据自己电脑性能给定)

# 然后正式编译课程代码
cd ../../../
mkdir build
cd build
cmake .. -DBUILD_WITH_UBUNTU1804=ON
make -j8 (-j8 表示用8个core编译, 根据自己电脑性能给定)
```

编译完成就可以尝试运行一下代码:

```
./bin/motion
```

可以看到如下界面则证明编译成功，可以开始愉快地学习了😄



8 部分问题解决思路

8.1 问题1

报错:

```
CMakeFiles/test_nn.dir/test_nn.cc.o：在函数‘std::result_of<std::integral_constant<bool, true> ()>::type __ps
tl::__internal::__except_handler<__pstl::__internal::__pattern_walk1<__pstl::execution::v1::parallel_unseque
nced_policy const&, __gnu_cxx::__normal_iterator<unsigned long*, std::vector<unsigned long, std::allocator<u
nsigned long> > >, sad::GridNN<2>::GetClosestPointForCloudMT(boost::shared_ptr<pcl::PointCloud<pcl::PointXYZ
I> >, boost::shared_ptr<pcl::PointCloud<pcl::PointXYZI> >, std::vector<std::pair<unsigned long, unsigned lon
g>, std::allocator<std::pair<unsigned long, unsigned long> > >&)::{lambda(unsigned long const&)#2}, std::int
egral_constant<bool, true> >(__pstl::execution::v1::parallel_unsequenced_policy const&, __gnu_cxx::__normal_
iterator<unsigned long*, std::vector<unsigned long, std::allocator<unsigned long> > >, __pstl::execution::v1
::parallel_unsequenced_policy const&, sad::GridNN<2>::GetClosestPointForCloudMT(boost::shared_ptr<pcl::Point
Cloud<pcl::PointXYZI> >, boost::shared_ptr<pcl::PointCloud<pcl::PointXYZI> >, std::vector<std::pair<unsigned
 long, unsigned long>, std::allocator<std::pair<unsigned long, unsigned long> > >&)::{lambda(unsigned long c
onst&)#2}, std::integral_constant<bool, true>, std::integral_constant)::{lambda()#1}>(std::integral_constant
<bool, true>)’中：
/media/xuyong666/C4B28107B280FEE4/workspace/slam_in_autonomous_driving/thirdparty/tbb/oneTBB-2019_U8/oneTBB-
2019_U8/include/tbb/task_arena.h:157：对‘tbb::interface7::internal::isolate_within_arena(tbb::interface7::in
ternal::delegate_base&, long)’未定义的引用
CMakeFiles/test_nn.dir/test_nn.cc.o：在函数‘std::result_of<std::integral_constant<bool, true> ()>::type __ps
tl::__internal::__except_handler<__pstl::__internal::__pattern_walk1<__pstl::execution::v1::parallel_unseque
nced_policy const&, __gnu_cxx::__normal_iterator<unsigned long*, std::vector<unsigned long, std::allocator<u
nsigned long> > >, sad::GridNN<3>::GetClosestPointForCloudMT(boost::shared_ptr<pcl::PointCloud<pcl::PointXYZ
I> >, boost::shared_ptr<pcl::PointCloud<pcl::PointXYZI> >, std::vector<std::pair<unsigned long, unsigned lon
g>, std::allocator<std::pair<unsigned long, unsigned long> > >&)::{lambda(unsigned long const&)#2}, std::int
egral_constant<bool, true> >(__pstl::execution::v1::parallel_unsequenced_policy const&, __gnu_cxx::__normal_
iterator<unsigned long*, std::vector<unsigned long, std::allocator<unsigned long> > >, __pstl::execution::v1
::parallel_unsequenced_policy const&, sad::GridNN<3>::GetClosestPointForCloudMT(boost::shared_ptr<pcl::Point
Cloud<pcl::PointXYZI> >, boost::shared_ptr<pcl::PointCloud<pcl::PointXYZI> >, std::vector<std::pair<unsigned
 long, unsigned long>, std::allocator<std::pair<unsigned long, unsigned long> > >&)::{lambda(unsigned long c
onst&)#2}, std::integral_constant<bool, true>, std::integral_constant)::{lambda()#1}>(std::integral_constant
<bool, true>)’中：
/media/xuyong666/C4B28107B280FEE4/workspace/slam_in_autonomous_driving/thirdparty/tbb/oneTBB-2019_U8/oneTBB-
2019_U8/include/tbb/task_arena.h:157：对‘tbb::interface7::internal::isolate_within_arena(tbb::interface7::in
ternal::delegate_base&, long)’未定义的引用
```

解决思路:

需要修改课程代码中的CMakeLists

```
# 修改src/ch5/CMakeLists.txt 中的第30，45行
target_link_libraries(${PROJECT_NAME}.ch5
# tbb
TBB::tbb
)

target_link_libraries(test_nn
gtest pthread glog gflags ${PROJECT_NAME}.ch5 ${PROJECT_NAME}.common ${PCL_LIBRARIES}
# tbb
TBB::tbb
)
```

8.2 问题2

报错:

```
                 from /media/xuyong666/C4B28107B280FEE4/workspace/slam_in_autonomous_driving/src/ch6/test_2d
_icp_s2s.cc:10:
/usr/include/lz4.h:249:72: error: conflicting declaration ‘typedef struct LZ4_streamDecode_t LZ4_streamDecod
e_t’
  249 | typedef struct { unsigned long long table[LZ4_STREAMDECODESIZE_U64]; } LZ4_streamDecode_t;
      |                                                                        ^~~~~~~~~~~~~~~~~~
In file included from /usr/include/flann/util/serialization.h:9,
                 from /usr/include/flann/util/matrix.h:35,
                 from /usr/include/flann/flann.hpp:41,
                 from /usr/include/pcl-1.8/pcl/kdtree/flann.h:50,
                 from /usr/include/pcl-1.8/pcl/kdtree/kdtree_flann.h:45,
                 from /usr/include/pcl-1.8/pcl/search/kdtree.h:44,
                 from /media/xuyong666/C4B28107B280FEE4/workspace/slam_in_autonomous_driving/src/ch6/icp_2d.
```

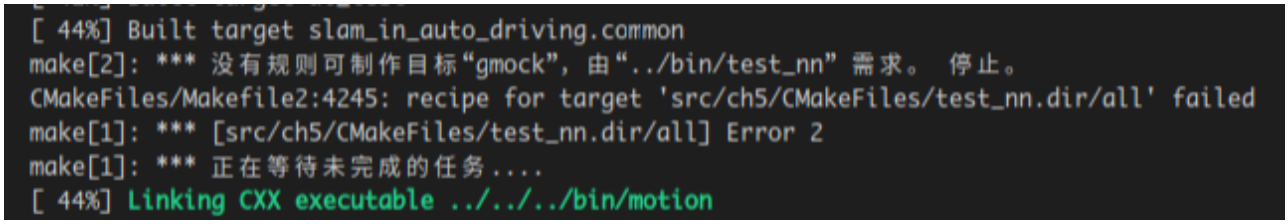
解决思路:

```
sudo mv /usr/include/flann/ext/lz4.h /usr/include/flann/ext/lz4.h.bak
sudo mv /usr/include/flann/ext/lz4hc.h /usr/include/flann/ext/lz4.h.bak

sudo ln -s /usr/include/lz4.h /usr/include/flann/ext/lz4.h
sudo ln -s /usr/include/lz4hc.h /usr/include/flann/ext/lz4hc.h
```

8.3 问题3

报错:



ubuntu 18.04正常`sudo apt-get install libgtest-dev`好像也会出现这个问题，猜测是版本太低的原因

解决思路:

源码安装最新的gtest

```
git clone https://github.com/google/googletest

cd googlegtest
mkdir build
cd build
cmake ..
make -j
sudo make install
```

修改代码的CMakeLists

```
# src/ch5/CMakeLists.txt 的第45行
target_link_libraries(test_nn
GTest::gtest pthread glog gflags ${PROJECT_NAME}.ch5 ${PROJECT_NAME}.common
${PCL_LIBRARIES}
TBB::tbb
)

# 上面提示找不到GTest::gtest就试下这段
target_link_libraries(test_nn
GTest::GTest pthread glog gflags ${PROJECT_NAME}.ch5 ${PROJECT_NAME}.common
${PCL_LIBRARIES}
TBB::tbb
)

# src/ch4/CMakeLists.txt中同理
```

重新编译一次课程代码

```
# 清除一下之前课程代码的编译结果
rm -rf /build

# 重新执行上面课程代码的编译步骤
```

9 一些第一期同学环境配置总结

9.1 gfags 安装

参考https://blog.csdn.net/u011622208/article/details/119823779

这里要注意的是cmake .. 命令之后需要加的参数：-DGFLAGS_NAMESPACE=google -DCMAKE_CXX_FLAGS=-fPIC .. 这会在之后程序链接 libglog.so.x. 的时候会避坑

9.2 glog安装

参考https://blog.csdn.net/peijian1998/article/details/117655872