1.FASTIO\FAST前向方案实现

环境wsl (win10 + 子系统ubuntu20.04LTS)

安装cryptopp密码学库

sudo apt-get install libcrypto++-dev libcrypto++-doc libcrypto++-utils

apt-cache pkgnames | grep -i crypto++ //查看版本和安装成功无？

g++ -DNDEBUG -g3 -O2 -Wall -Wextra -o test test.cpp -l:libcryptopp.a

测试代码

|  |
| --- |
| #include <bits/stdc++.h>  #include <crypto++/integer.h>  using namespace std;  using namespace CryptoPP;  int main()  {      Integer i("11212321321312");      i -= 2;      cout << i << endl;      return 0;  } |

g++ main.cpp -o main -lcryptopp –lpthread //编译

./main //运行

安装grpc

Currently to install gRPC for C++, you need to build from source as described below.

git clone https://github.com/grpc/grpc.git

cd grpc

git submodule update --init

make

[sudo] make install

wget https://cmake.org/files/v3.17/cmake-3.17.3.tar.gz

Cmake 安装

Apt-get install cmake

Cd cmake

安装rocksdb

首先安装依赖

**Linux - Ubuntu**

Upgrade your gcc to version at least 4.8 to get C++11 support.

sudo apt-get install libgflags-dev libsnappy-dev zlib1g-dev libbz2-dev liblz4-dev libzstd-dev

Install gflags. First, try: sudo apt-get install libgflags-dev If this doesn't work and you're using Ubuntu, here's a nice tutorial: (<http://askubuntu.com/questions/312173/installing-gflags-12-04>)

Install snappy. This is usually as easy as: sudo apt-get install libsnappy-dev.

Install zlib. Try: sudo apt-get install zlib1g-dev.

Install bzip2: sudo apt-get install libbz2-dev.

Install lz4: sudo apt-get install liblz4-dev.

Install zstandard: sudo apt-get install libzstd-dev.

sudo apt install gcc-10 g++-10

下载rocksdb5.7.3 release version

解压后进入文件夹

sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-10 100 --slave /usr/bin/g++ g++ /usr/bin/g++-10 --slave /usr/bin/gcov gcov /usr/bin/gcov-10

sudo update-alternatives --config gcc

make shared\_lib -j4 CC=gcc-10 CPP=g++-10 CXX=g++-10 LD=g++-10

wsl走windows代理

在~/.bashrc里写入

export http\_proxy="socks5://127.0.0.1:7890"

export https\_proxy="socks5://127.0.0.1:7890"

**#!/bin/bash**

host\_ip**=$(**cat /etc/resolv.conf |grep "nameserver" |cut -f 2 -d " "**)**

export ALL\_PROXY**=**"http://$host\_ip:7890"

取消代理 ：

unset http\_proxy unset https\_proxy

测试代码

|  |
| --- |
| #include <cstdio>  #include <string>  #include "rocksdb/db.h"  #include "rocksdb/slice.h"  #include "rocksdb/options.h"  using namespace std;  using namespace rocksdb;  const std::string PATH = "/tmp/rocksdb\_tmp";  int main(){      DB\* db;      Options options;      options.create\_if\_missing = true;      Status status = DB::Open(options, PATH, &db);      assert(status.ok());      Slice key("foo");      Slice value("bar");      std::string get\_value;      status = db->Put(WriteOptions(), key, value);      if(status.ok()){          status = db->Get(ReadOptions(), key, &get\_value);          if(status.ok()){              printf("get %s success!!\n", get\_value.c\_str());          }else{              printf("get failed\n");          }      }else{          printf("put failed\n");      }      delete db;      return 0;  } |

动态编译：

g++ -std=c++11 -o rocksdbtest test.cpp -lrocksdb -lpthread -ldl

最后这个 -ldl很重要

运行：

./rocksdbtest

或者静态编译：

g++ -std=c++11 -o rocksdbtest test.cpp ./librocksdb.a -lpthread -lsnappy -lz -lbz2 -lzstd /usr/lib/x86\_64-linux-gnu/liblz4.a

<http://doc.oschina.net/grpc?t=57966>

<https://gitmemory.cn/repo/grpc/grpc/issues/25949>

<https://blog.51cto.com/u_15081048/2592774>

grpc测试用例 examples/cpp/helloworld

sudo cp -r /home/helen/sse/grpc/third\_party/abseil-cpp/absl /us

r/local/include/

sudo cp -r /home/helen/Desktop/grpc/third\_party/abseil-cpp/absl /usr/local/include/

sudo cp -r /home/sse/grpc/third\_party/abseil-cpp/absl /usr/local/include/

cd examples/cpp/helloworld

mkdir -p cmake/build

pushd cmake/build

cmake –DCMAKE\_PREFIX\_PATH=$MY\_INSTALL\_DIR ../..

make –j4

编译成功

FASTIO

1. memory version编译

发现找不到openssl和pkg-config，安装辅助包

Sudo apt-get install pkg-config

sudo apt install libssl-dev

cd memory-version

make –j4

运行：

先在一个终端执行 ./rpc\_server [number of threads]，比如./rpc\_server 4，这里4代表线程数量；  
然后另外一个终端开启 ./rpc\_client [db path] [number of enrties] [keyword] [number of threads]， 比如./rpc\_client ./testdb 100000 Group-10^1\_0\_1 4, 具体的参数和关键字格式，请参考rpc\_client.cc和DistSSE.db\_generator.h中的定义

1. rocksdb

make –j4

1. rocksdb-private key

make –j4

1. Sophos

先安装scons等一系列工具和库

Sudo apt-get install build-essential autoconf libtool yasm openssl scons

如果wsl，Ubuntu，关于解决E: Could not read response to hello message from hook [ ! -f /usr/bin/snap ] || /usr/ 错误

通过使用

sudo rm -rf /etc/apt/apt.conf.d/20snapd.conf

解决问题

Openssl 重新安装1.0.2g

需要那个函数

解压后进入目录

./config --prefix=/usr/local/ssl --openssldir=/usr/local/ssl shared -fPIC

sudo make –j4

sudo make test –j4

sudo make install –j4

配置共享库

cd /etc/ld.so.conf.d/

sudo vim openssl-1.0.2g.conf

写入下面的语句

/usr/local/ssl/lib

**现在使用以下命令重新加载动态链接:**

sudo ldconfig -v

**使用vim编辑'/etc/environment'文件:**

sudo vim /etc/environment

**添加新的OpenSSL二进制目录:**

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/local/ssl/bin"

**重新加载环境文件并测试新更新的二进制PATH:**

source /etc/environment

echo $PATH

**查看安装结果：**

openssl version -a

Boost库安装

选择了boost 1.63.0版本

tar -xzvf boost\_1\_63\_0.tar.gz  
cd boost\_1\_63\_0

./bootstrap.sh //准备工作  
./b2 //编译，该过程时间比较长

对sophos

我们直接修改里面的SC

各种特性和语法理解

#pragma once

  #pragma once 一般由编译器提供保证：同一个文件不会被包含多次。注意这里所说的“同一个文件”是指物理上的一个文件，而不是指内容相同的两个文件。

**你无法对一个头文件中的一段代码作pragma once声明，而只能针对文件。**

  其好处是，你不必再担心宏名冲突了，当然也就不会出现宏名冲突引发的奇怪问题。大型项目的编译速度也因此提高了一些。

  对应的缺点就是如果某个头文件有多份拷贝，本方法不能保证他们不被重复包含。当然，相比宏名冲突引发的“找不到声明”的问题，这种重复包含很容易被发现并修正。