1.安装 OpenTuner

去清华源下载，选择python3.8的x86_64版本

Miniconda3-py38_4.8.3-Windows-x86_64.exe	55.7 MiB	2020-06-17 04:06
Miniconda3-py38_4.9.2-Linux-aarch64.sh	111.8 MiB	2021-05-14 22:53
Miniconda3-py38_4.9.2-Linux-ppc64le.sh	91.9 MiB	2020-11-24 03:21
Miniconda3-py38_4.9.2-Linux-s390x.sh	102.5 MiB	2021-05-14 23:15
Miniconda3-py38_4.9.2-Linux-x86_64.sh	89.9 MiB	2020-11-24 03:21
Miniconda3-py38_4.9.2-MacOSX-x86_64.pkg	62.0 MiB	2020-11-24 03:21
Miniconda3-py38_4.9.2-MacOSX-x86_64.sh	54.5 MiB	2020-11-24 03:22
Miniconda3-py38_4.9.2-Windows-x86.exe	54.2 MiB	2020-11-24 03:22
Miniconda3-py38_4.9.2-Windows-x86_64.exe	57.0 MiB	2020-11-24 03:22
Miniconda3-py39_4.10.1-Linux-aarch64.sh	69.8 MiB	2021-06-02 07:46
Miniconda3-py39_4.10.1-Linux-s390x.sh	57.1 MiB	2021-06-02 07:46
Miniconda3-py39_4.10.3-Linux-aarch64.sh	62.6 MiB	2021-07-22 00:14
Miniconda3-py39_4.10.3-Linux-ppc64le.sh	60.6 MiB	2021-07-22 00:14
Miniconda3-py39_4.10.3-Linux-s390x.sh	57.1 MiB	2021-07-22 00:14
Miniconda3-py39_4.10.3-Linux-x86_64.sh	63.6 MiB	2021-07-22 00:14
Miniconda3-py39_4.10.3-MacOSX-x86_64.pkg	49.9 MiB	2021-07-22 00:14
Miniconda3-py39_4.10.3-MacOSX-x86_64.sh	42.3 MiB	2021-07-22 00:14
Miniconda3-py39_4.10.3-Windows-x86.exe	55.3 MiB	2021-07-22 00:14
Miniconda3-py39_4.10.3-Windows-x86_64.exe	58.1 MiB	2021-07-22 00:14
Miniconda3-py39_4.9.2-Linux-aarch64.sh	76.2 MiB	2021-05-14 22:53

```
wget https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/Miniconda3-
py38_4.9.2-Linux-x86_64.sh
```


在文件所在目录打开终端

```
chmod 777 Miniconda3-latest-Linux-x86_64.sh
sh Miniconda3-latest-Linux-x86_64.sh
```

然后一路yes

即安装成功。

查看版本号：

给conda 添加清华源，这样安装其他包的时候，下载速度会很快

```
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/bioconda/
conda config --set show_channel_urls yes
conda config --get channels
```

配置完成后 主目录文件夹多了个隐藏文件 .condarc

这里要注意：.condarc 文件打开后里面有 - default 一定要删除掉

```
1 channels:
2   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/bioconda/
3   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
4   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
5   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
6
7 show_channel_urls: true
```

创建虚拟环境:

```
conda create --name=opentuner python=3.8
```


激活环境:

```
conda activate opentuner
```


下载opentuner:

```
wget https://github.com/jansel/opentuner
```

这种方式是错误的，最后下载的是html网页。

我选择了先将zip下载到本地主机，再复制粘贴到虚拟机里
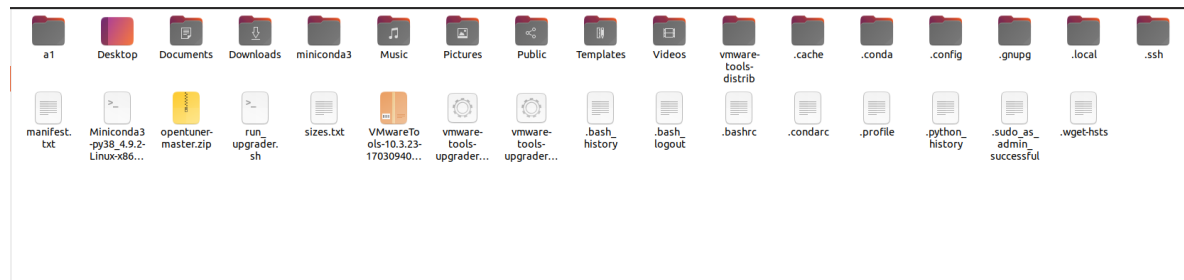
解压：

```
unzip opentuner-master.zip
```

安装opentuner：

```
pip install -r requirements.txt -r optional-requirements.txt
```



```
python setup.py develop
```

报错：

按提示安装pandoc：

```
sudo apt-get install pandoc
```

重新运行setup.py 成功：

```
reading manifest file 'opentuner.egg-info/SOURCES.txt'
reading manifest template 'MANIFEST.in'
adding license file 'LICENSE.txt'
adding license file 'AUTHORS.txt'
writing manifest file 'opentuner.egg-info/SOURCES.txt'
running build_ext
Creating /home/zhangkeer/miniconda3/envs/opentuner/lib/python3.8/site-packages/opentuner.egg-link (link to .)
Adding opentuner 0.8.7 to easy-install.pth file

Installed /home/zhangkeer/opentuner-master
Processing dependencies for opentuner==0.8.7
Searching for SQLAlchemy==1.4.25
Best match: SQLAlchemy 1.4.25
Adding SQLAlchemy 1.4.25 to easy-install.pth file

Using /home/zhangkeer/miniconda3/envs/opentuner/lib/python3.8/site-packages
Searching for numpy==1.21.2
Best match: numpy 1.21.2
Adding numpy 1.21.2 to easy-install.pth file
Installing f2py script to /home/zhangkeer/miniconda3/envs/opentuner/bin
Installing f2py3 script to /home/zhangkeer/miniconda3/envs/opentuner/bin
Installing f2py3.8 script to /home/zhangkeer/miniconda3/envs/opentuner/bin

Using /home/zhangkeer/miniconda3/envs/opentuner/lib/python3.8/site-packages
Searching for future==0.18.2
Best match: future 0.18.2
Adding future 0.18.2 to easy-install.pth file
Installing futurize script to /home/zhangkeer/miniconda3/envs/opentuner/bin
Installing pasteurize script to /home/zhangkeer/miniconda3/envs/opentuner/bin

Using /home/zhangkeer/miniconda3/envs/opentuner/lib/python3.8/site-packages
Searching for fn==0.4.3
Best match: fn 0.4.3
Adding fn 0.4.3 to easy-install.pth file

Using /home/zhangkeer/miniconda3/envs/opentuner/lib/python3.8/site-packages
Searching for argparse==1.4.0
Best match: argparse 1.4.0
Adding argparse 1.4.0 to easy-install.pth file

Using /home/zhangkeer/miniconda3/envs/opentuner/lib/python3.8/site-packages
Searching for greenlet==1.1.1
Best match: greenlet 1.1.1
Adding greenlet 1.1.1 to easy-install.pth file

Using /home/zhangkeer/miniconda3/envs/opentuner/lib/python3.8/site-packages
Finished processing dependencies for opentuner==0.8.7
(opentuner) zhangkeer@zhangkeer-virtual-machine:~/opentuner-master$
```

检查是否安装成功：

```
pytest tests/*
```

```
(opentuner) zhangkeer@zhangkeer-virtual-machine:~/opentuner-master$ pytest tests/*
================================= test session starts =================================
platform linux -- Python 3.8.12, pytest-6.2.5, py-1.10.0, pluggy-1.0.0
rootdir: /home/zhangkeer/opentuner-master
collected 23 items

tests/test_manipulator.py ...................                                    [ 82%]
tests/test_technique.py ....                                                     [100%]

================================== warnings summary ===================================
../miniconda3/envs/opentuner/lib/python3.8/site-packages/future/standard_library/__init__.py:65
  /home/zhangkeer/miniconda3/envs/opentuner/lib/python3.8/site-packages/future/standard_library/__init__.py:65: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
    import imp

../miniconda3/envs/opentuner/lib/python3.8/site-packages/fn/iters.py:2
  /home/zhangkeer/miniconda3/envs/opentuner/lib/python3.8/site-packages/fn/iters.py:2: DeprecationWarning: Using or importing the ABCs from 'collections' instead of from 'collections.abc' is deprecated since Python 3.3, and in 3.10 it will stop working
    from collections import deque, Iterable

-- Docs: https://docs.pytest.org/en/stable/warnings.html
============================ 23 passed, 2 warnings in 0.80s ============================
(opentuner) zhangkeer@zhangkeer-virtual-machine:~/opentuner-master$
```

```
./examples/rosenbrock/rosenbrock.py
```

2.阅读论文

介绍了OpenTuner这一开源框架的原理、有效性及其通用性。OpenTuner在autotuning中引入了搜索技术集成的概念，允许搜索技术协同工作来找到最佳方案，它实现了7个autotuners。同时，任何人可以往这一开源项目中添加新的技术。在程序自动调整领域，OpenTuner发挥了很大的作用。

3.练习 Optimizing Block Matrix Multiplication教程

Save the sample code below to examples/tutorials/mmm_block.cpp

```cpp
#include <stdio.h>
#include <cstdlib>

#define N 100

int main(int argc, const char** argv)
{

  int n = BLOCK_SIZE * (N/BLOCK_SIZE);
  int a[N][N];
  int b[N][N];
  int c[N][N];
  int sum=0;
  for(int k1=0;k1<n;k1+=BLOCK_SIZE)
  {
      for(int j1=0;j1<n;j1+=BLOCK_SIZE)
      {
          for(int k1=0;k1<n;k1+=BLOCK_SIZE)
          {
              for(int i=0;i<n;i++)
              {
                  for(int j=j1;j<j1+BLOCK_SIZE;j++)
                  {
                      sum = c[i][j];
                      for(int k=k1;k<k1+BLOCK_SIZE;k++)
                      {
                          sum += a[i][k] * b[k][j];
                      }
                      c[i][j] = sum;
                  }
              }
          }
      }
  }
```

```
            }
    return 0;
}
```

已存在:
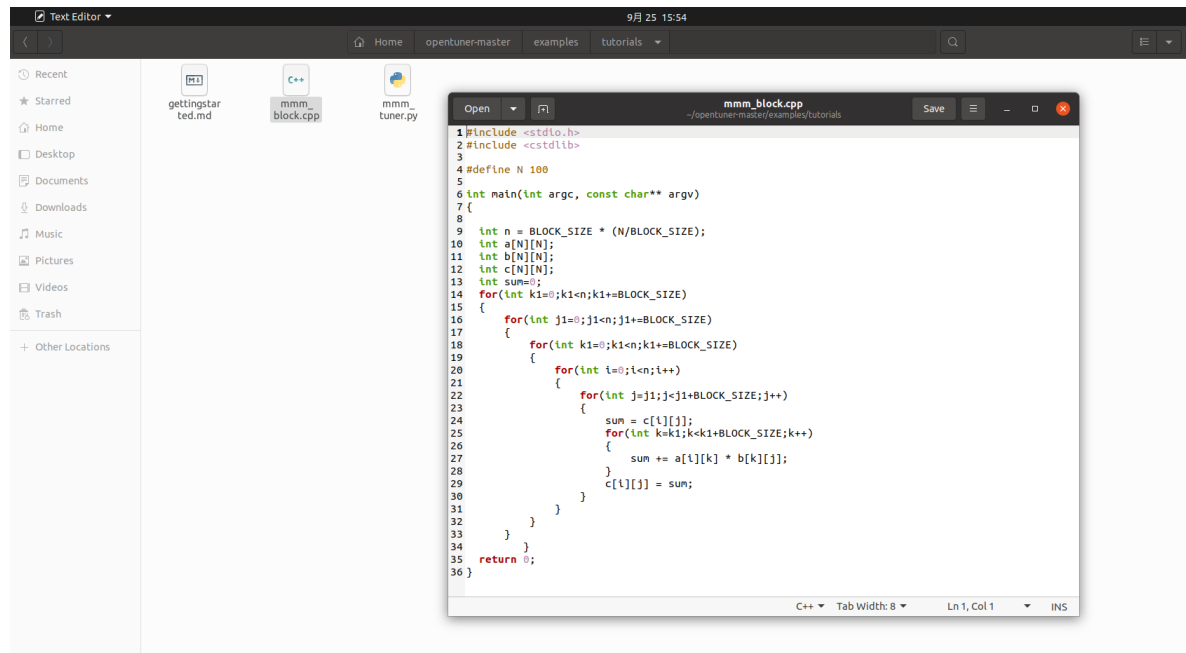
Save the following code to examples/tutorials/mmm_tuner.py

```python
#!/usr/bin/env python
#
# Optimize blocksize of apps/mmm_block.cpp
#
# This is an extremely simplified version meant only for tutorials
#
import adddeps  # fix sys.path

import opentuner
from opentuner import ConfigurationManipulator
from opentuner import IntegerParameter
from opentuner import MeasurementInterface
from opentuner import Result


class GccFlagsTuner(MeasurementInterface):

  def manipulator(self):
    """
    Define the search space by creating a
    ConfigurationManipulator
    """
    manipulator = ConfigurationManipulator()
    manipulator.add_parameter(
      IntegerParameter('blockSize', 1, 10))
    return manipulator
```

```python
    def run(self, desired_result, input, limit):
        """
        Compile and run a given configuration then
        return performance
        """
        cfg = desired_result.configuration.data

        gcc_cmd = 'g++ mmm_block.cpp '
        gcc_cmd += '-DBLOCK_SIZE='+ cfg['blockSize']
        gcc_cmd += ' -o ./tmp.bin'

        compile_result = self.call_program(gcc_cmd)
        assert compile_result['returncode'] == 0

        run_cmd = './tmp.bin'

        run_result = self.call_program(run_cmd)
        assert run_result['returncode'] == 0

        return Result(time=run_result['time'])

    def save_final_config(self, configuration):
        """called at the end of tuning"""
        print "Optimal block size written to mmm_final_config.json:",
configuration.data
        self.manipulator().save_to_file(configuration.data,
                                        'mmm_final_config.json')


if __name__ == '__main__':
    argparser = opentuner.default_argparser()
    GccFlagsTuner.main(argparser.parse_args())
```
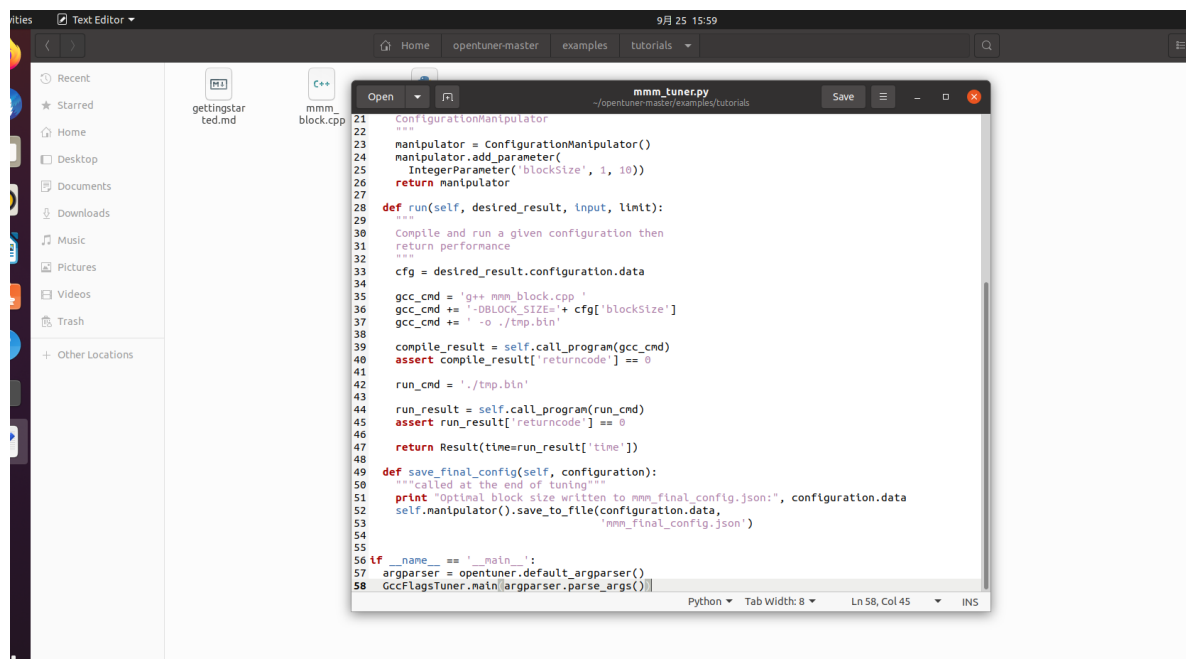

Run the following command to autotune our program(The –no-dups flag hides warnings about duplicate results and the –stop-after parameter specifies that we are running opentuner for a maximum of 30 seconds):

```
python mmm_tuner.py --no-dups --stop-after=30
```

运行报错：

因为print 没有括号是python2的语法

```
(base) zhangkeer@zhangkeer-virtual-machine:~/opentuner-master/examples/tutorials$ python mmm_tuner.py --no-dups --stop-after=30
  File "mmm_tuner.py", line 51
    print "Optimal block size written to mmm_final_config.json:", configuration.data
          ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("Optimal block size written to mmm_final_config.json:", configuration.data)?
(base) zhangkeer@zhangkeer-virtual-machine:~/opentuner-master/examples/tutorials$
```

报错：未找到模块

```
(base) zhangkeer@zhangkeer-virtual-machine:~/opentuner-master/examples/tutorials$ python mmm_tuner.py --no-dups --stop-after=30
Traceback (most recent call last):
  File "mmm_tuner.py", line 7, in <module>
    import adddeps # fix sys.path
ModuleNotFoundError: No module named 'adddeps'
(base) zhangkeer@zhangkeer-virtual-machine:~/opentuner-master/examples/tutorials$
```

安装命令：

```
pip install adddeps
```

上面对找不到模块的错误处理都不对，因为作业不是连贯完成的，这才注意到没有使用conda的opentuner环境

修改环境后报错：

```
(opentuner) zhangkeer@zhangkeer-virtual-machine:~/opentuner-master/examples/tutorials$ python mmm_tuner.py --no-dups --stop-after=30
Traceback (most recent call last):
  File "mmm_tuner.py", line 58, in <module>
    GccFlagsTuner.main(argparser.parse_args())
  File "/home/zhangkeer/opentuner-master/opentuner/measurement/interface.py", line 300, in main
    return TuningRunMain(cls(args, *pargs, **kwargs), args).main()
  File "/home/zhangkeer/opentuner-master/opentuner/tuningrunmain.py", line 201, in main
    self.search_driver.main()
  File "/home/zhangkeer/opentuner-master/opentuner/search/driver.py", line 287, in main
    self.run_generation_results(offset=-self.args.pipelining)
  File "/home/zhangkeer/opentuner-master/opentuner/search/driver.py", line 239, in run_generation_results
    self.wait_for_results(self.generation + offset)
  File "/home/zhangkeer/opentuner-master/opentuner/tuningrunmain.py", line 219, in results_wait
    self.measurement_driver.process_all()
  File "/home/zhangkeer/opentuner-master/opentuner/measurement/driver.py", line 217, in process_all
    self.run_desired_result(dr)
  File "/home/zhangkeer/opentuner-master/opentuner/measurement/driver.py", line 134, in run_desired_result
    result = self.interface.compile_and_run(desired_result, input,
  File "/home/zhangkeer/opentuner-master/opentuner/measurement/interface.py", line 131, in compile_and_run
    return self.run(desired_result, input, limit)
  File "mmm_tuner.py", line 36, in run
    gcc_cmd += '-DBLOCK_SIZE='+ cfg['blockSize']
TypeError: can only concatenate str (not "int") to str
(opentuner) zhangkeer@zhangkeer-virtual-machine:~/opentuner-master/examples/tutorials$
```

将36行的gcc_cmd += '-DBLOCK_SIZE='+ cfg['blockSize']修改为gcc_cmd += '-DBLOCK_SIZE='+ str(cfg['blockSize'])

运行结果：

```
Optimal block size written to mmm_final_config.json: {'blockSize': 3}
(opentuner) zhangkeer@zhangkeer-virtual-machine:~/opentuner-master/examples/tutorials$
```

{"blockSize": 3}