1. download SPEC2006 benchmark。Compile following the guidance

<https://www.spec.org/cpu2006/Docs/install-guide-unix.html> step 1-10

* 1. source SPEC2006/shrc 可以注册命令runspec
  2. SPEC2006/config$ cp ./Example-linux64-amd64-gcc43.cfg myconfig.cfg产生linux64 gcc43的配置文件，以备后用。
  3. Build benchmark
     1. runspec --config=myconfig.cfg --action=build --tune=base bzip2
  4. pre-run benchmark with reference workload:
     1. runspec --config=myconfig.cfg --size=ref --noreportable --tune=base –iterations=1 bzip2 在CPU2006/401.bzip2下产生run/run-base-ref-amd64-m64-gcc43-nn.0000文件夹，其中包含可执行程序，所需要的所有输入文件。

1. download pinplay-3.5
   1. 为benchmark编写config file，详见README.pinpoint。注意work directory应为482.sphinx3/run/run-base-ref-amd64-m64-gcc43-nn.0000文件夹.否则找不到相应文件。

[Parameters]

program\_name: sphinx3

input\_name: ref

command: "./sphinx\_livepretend\_base.amd64-m64-gcc43-nn ./ctlfile ./ ./args.an4"

debug: False

cutoff: 1.0

epilog\_length: 0

maxk: 5

mode: st

num\_proc: 1

prolog\_length: 0

slice\_size: 30000000

verbose: False

warmup\_length: 100000

\*benchmark使用方法(command)详见官网。

(1)./bzip2 ./input.source

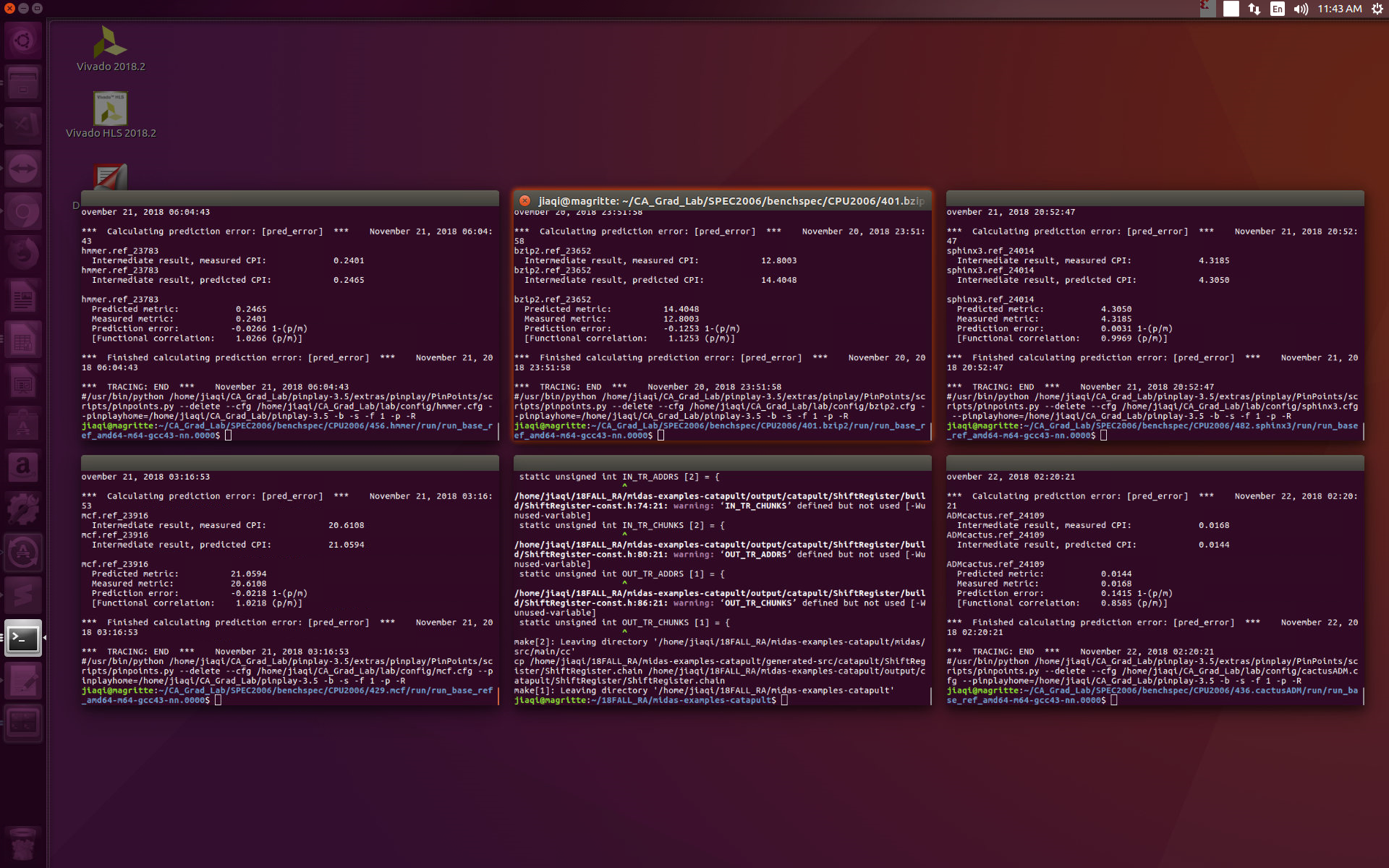
(2)./sphinx ./ctlfile ./ ./args.an4

(3)./cactusADM ./benchADM.par

(4)./mcf ./inp.in

(5)./hmmer ./nph3.hmm ./swiss41

* 1. 使用pinplay提供的pinpoint工具生成simpoints
     1. 在makefile中实现如下
     2. python pinplay-3.5/extras/pinplay/PinPoints/scripts/brpred\_pinpoints.py --delete --cfg bzip2.cfg --pinplayhome=$(pin\_dir) -lrbsp –TWc



1. clone repo of ChampSim. Build simulator with CPU configuration

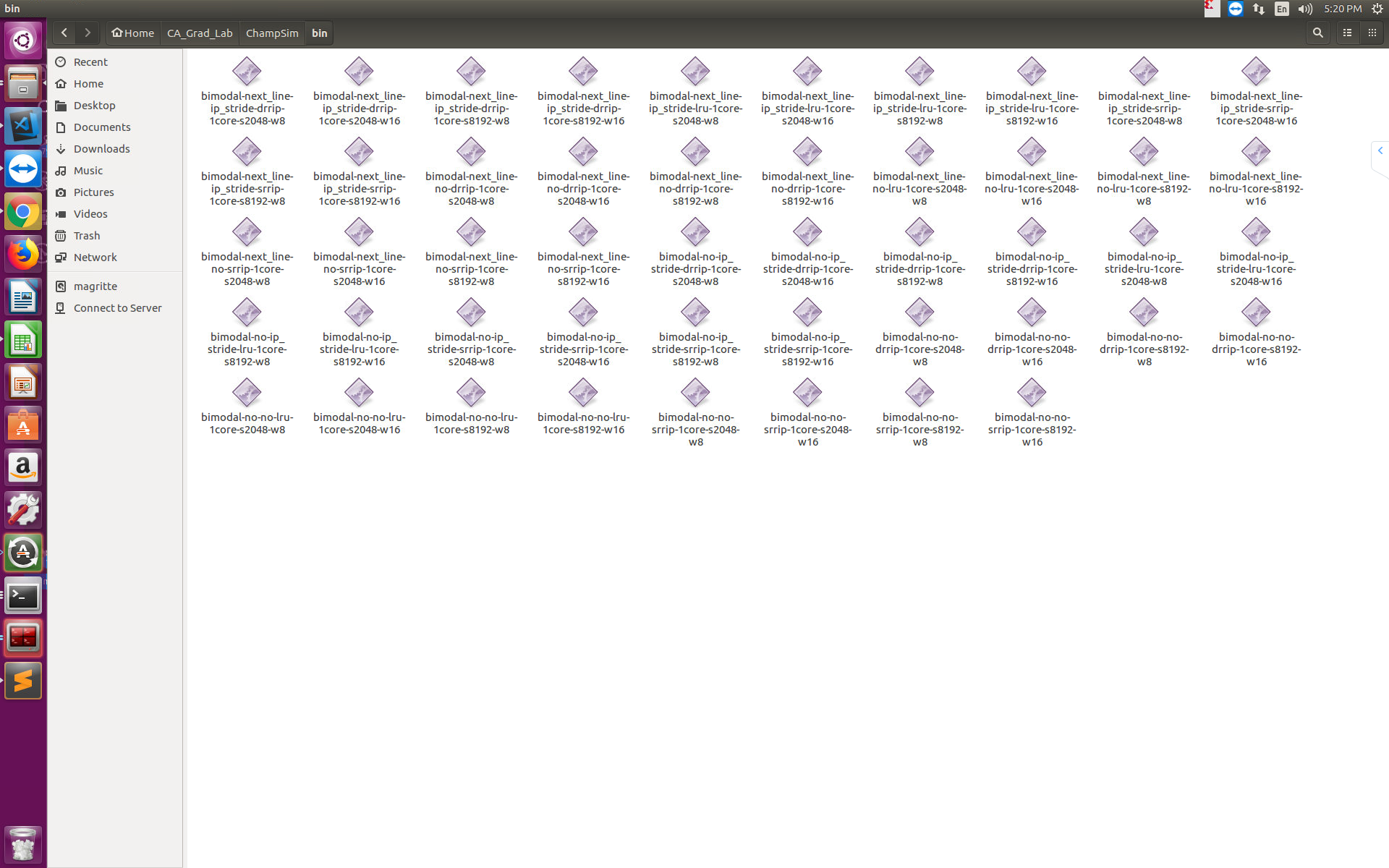
./build\_champsim.sh ${BRANCH} ${L1D\_PREFETCHER} ${L2C\_PREFETCHER} ${LLC\_REPLACEMENT} ${NUM\_CORE}

* 1. branch prediction: bimodal
  2. L1 Data Cache : no, next\_line
  3. L2 Coherent Cache: no; ip\_stride
  4. LLC replacement policy : lru; drrip;srrip
  5. Core: 1
  6. /inc/cache.h中
     1. LLC set = 2048 , 8192
     2. LLC way=8, 16

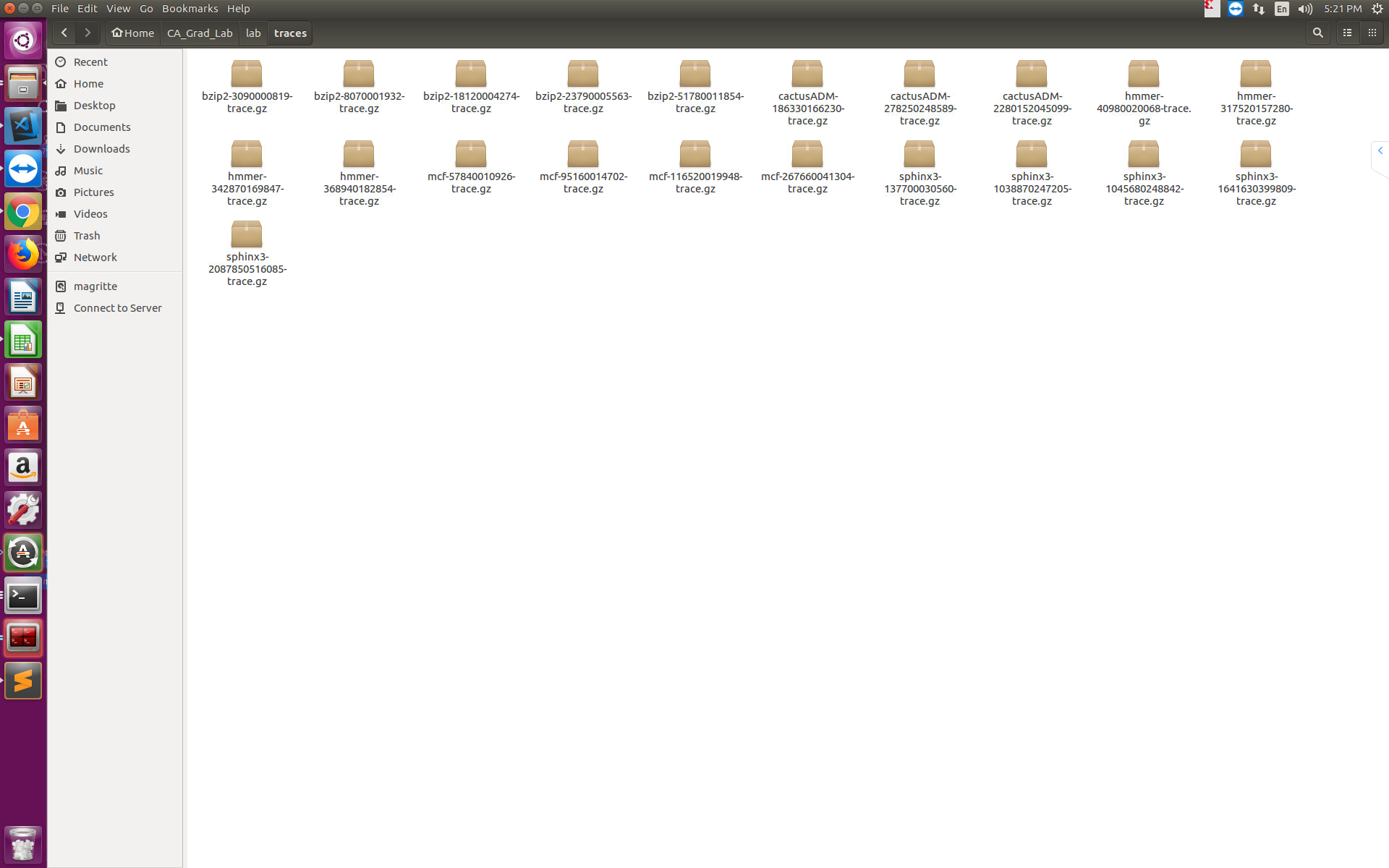
1. Generate binary files in /bin

原始命令：./build\_champsim.sh bimodal no no lru 1

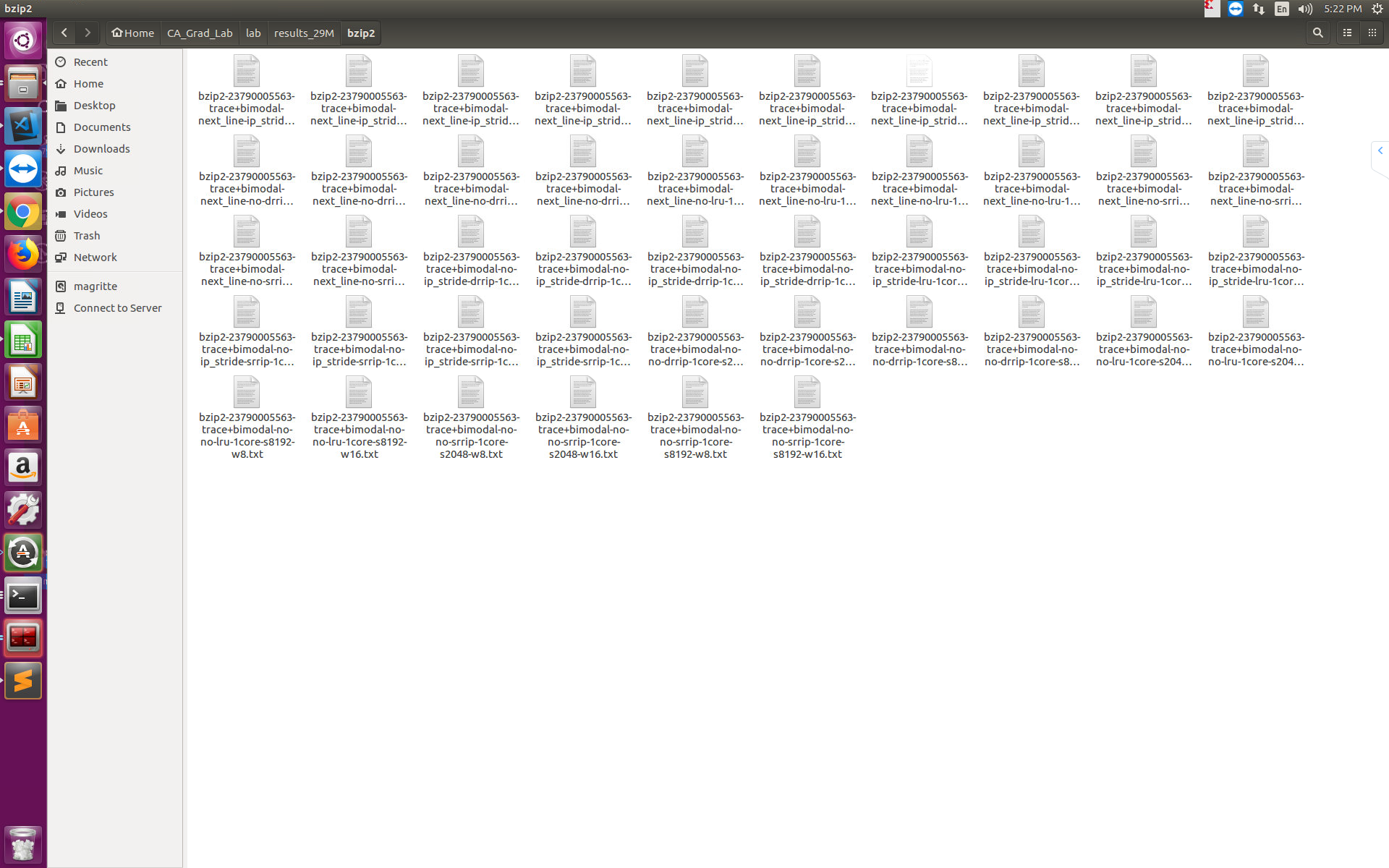
Python： compile\_champsim.py对应修改其中num\_way, num\_set



1. Generate tracer using Pin
   1. /tracer $ ./make\_tracer.sh
      1. obj\_dir = obj-intel64
      2. export PIN\_ROOT = PIN\_ROOT=/home/jiaqi/CA\_Grad\_Lab/pinplay-3.5
   2. after make tracer, the champsim\_tracer.o and champsim\_tracer.so is under /tracer/obj-intel64
2. Generate Traces by using pinpoints
   1. 摘取simpoint csv文件中每一个cluster的start，end。
   2. 调用自己编写的trace\_gen.py，来生成traces，位于/lab/traces下
   3. 使用compress\_trace.py来压缩traces –> \*.gz



1. 使用trace和champsim进行仿真
   1. 调用run\_champsim\_bzip2.py来针对不同的系统配置，不同的trace进行仿真。得到result\_29M/bzip2/下各种txt结果文件



1. 收集数据进行分析
   1. Txt文件中有不同配置、不同benchmark、不同trace对应的IPC, miss rate