Akshala Bhatnagar
2018012

OS Optional Assignment-2

- All the steps (including commands), you followed to run these benchmarks
  - Connected to FortiClient SSLVPN
  - $ scp aos@192.168.1.161:cpu2017-1_0_5.iso .
  - Provided the password aos after running above instruction
  - Extract the iso file
  - $ sudo apt install gfortran
  - $sudo apt install clang
  - $ ./cd cpu2017-1_0_5
  - $./install.sh

  Installed in ~/Documents/IIITD/fourthSem/OS/optional_assig
  nment2
  from ~/Documents/IIITD/fourthSem/OS/optional_assig
  nment2/cpu2017-1_0_5
  - $ source shrc

  1. **<u>gcc</u>**

  - Copy Example-gcc-linux-x86.cfg to gcc.cfg
  - We need to make changes at the places where EDIT is written.
  - Change the line % define label something, here something is replaced with a suitable label name.

  %define label something
  - For the Preprocessor, the number of bits are set to 64 bits.

  %ifndef %{bits}   # EDIT to control 32 or 64 bit compilation. Or,
  %   define bits   64
  - Change the value of build_ncpus from 8 to 4, this indicates the number of simultaneous compiles.
  - %   define build_ncpus 4
  - Change the path to gcc compiler to /usr. This was found by running command.
    $ which gcc

```
% define gcc_dir /usr
```

- The version of gcc compiler is changed.

```
#--------- EDIT to match your version ---------------------------------------
default:
  sw_compiler001   = C/C++/Fortran: Version 7.4.0 of GCC, the
  sw_compiler002   = GNU Compiler Collection
```

- The number of copies is kept as 1 and the number of threads is changed to 8.
- For `SPECrate Integer` suit for gcc we run the command
  $runcpu --config=gcc intrate
- For `SPECrate Floating Point` suit for we run the command
  $runcpu --config=gcc fptrate

2. **clang**

- Copy Example-gcc-linux-x86.cfg to clang.cfg
- We need to make changes at the places where EDIT is written.
- Change the line % define label something, here something is replaced with a suitable label name.

```
%define label something
```

- For the Preprocessor, the number of bits are set to 64 bits.

```
%ifndef %{bits}            # EDIT to control 32 or 64 bit compilation.  Or,
%   define  bits    64
```

- Change the value of build_ncpus from 8 to 4, this indicates the number of simultaneous compiles.

```
%   define  build_ncpus 4
```

- Change the path to gcc compiler to /usr. This was found by running command.
  $ which gcc

```
%   define  gcc_dir       /usr
```

- The version of gcc compiler is changed.

```
#--------- EDIT to match your version ---------------------------------------
```

Akshala Bhatnagar
2018012

```
default:
   sw_compiler001   = C/C++/Fortran: Version 7.4.0 of GCC, the
   sw_compiler002   = GNU Compiler Collection
```

- In the lines where CC and CXX are declared, gcc is changed to clang and remove the std flags.

```
 CC               = $(SPECLANG)clang          %{model}
CXX               = $(SPECLANG)clang++        %{model}
```

- From the line: OPTIMIZE = -g -O3 -march=native -fno-unsafe-math-optimizations -fno-tree-loop-vectorize remove -fno-tree-loop-vectorize
- The number of copies is kept as 1 and the number of threads is changed to 8.
- For `SPECrate Integer` suit for gcc we run the command
  $runcpu --config=clang intrate
- For `SPECrate Floating Point` suit for we run the command
  $runcpu --config=clang fptrate
  Upon running fptrate we get an error in files 510, 521 and 527. Upon seeing the make.out files corresponding to these changes were made in the config file.
- For 510, the following change was made.
  - In the lines where CC and CXX are declared, gcc is changed to clang and the std flags were not removed.

```
CC                 = $(SPECLANG)clang -std=c99   %{model}
  CXX              = $(SPECLANG)clang++      -std=c++03 %{model}
```

- So for 521 and 527, the following changes were made.
  - In the lines where CC and CXX are declared, gcc is changed to clang.
  - -fPIC flag was included in the line OPTIMIZE = -g -O3 -march=native -fno-unsafe-math-optimizations.

Akshala Bhatnagar
2018012

```
  OPTIMIZE       = -g -fPIC -O3 -march=native
-fno-unsafe-math-optimizations
```

- ○ The results are produced in a pdf file which are stored in a folder named “result” which is located where we had extracted the iso file.

- A table that includes the runtime (in seconds) of all the benchmarks in the `SPECrate Integer` suit for both `gcc` and `clang`.

Benchmark	gcc(in seconds)	clang
500.perlbench_r	316	339
502.gcc_r	226	227
505.mcf_r	339	364
520.omnetpp	415	416
523.xalancbmk_r	311	299
525.x264_r	382	240
531.deepsjeng_r	299	274
541.leela_r	469	452
548.exchange2_r	267	381
557.xz_r	360	378

- A table that includes the runtime (in seconds) of all benchmarks in the `SPECrate Floating Points` suit for both `gcc` and `clang`.

Benchmark	gcc(in seconds)	clang
503.bwaves_r	818	680

Akshala Bhatnagar
2018012

507.cactuBSSN_r	285	236
508.namd_r	292	202
510.parest_r	619	460
511.povray_r	467	376
519.lbm_r	262	248
521.wrf_r	701	449
526.blender_r	374	279
527.cam4_r	461	345
538.imagick_r	505	380
544.nab_r	429	311
549.fotonik3d_r	568	460
554.roms_r	451	332