軟體分析與最佳化 HW2

612410017 林靖紳

Execution environments

- CPU information

```
ashen@Stephanie-Lin:~$ lscpu
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
  Address sizes:         39 bits physical, 48 bits virtual
  Byte Order:            Little Endian
CPU(s):                  12
  On-line CPU(s) list:   0-11
Vendor ID:               GenuineIntel
  Model name:            11th Gen Intel(R) Core(TM) i5-11500 @ 2.70GHz
    CPU family:          6
    Model:               167
    Thread(s) per core:  2
    Core(s) per socket:  6
    Socket(s):           1
    Stepping:            1
    CPU max MHz:         4600.0000
    CPU min MHz:         800.0000
    BogoMIPS:            5424.00
```

- Memory

```
ashen@Stephanie-Lin:~$ free -h
               total        used        free      shared  buff/cache   available
Mem:            31Gi       4.2Gi        12Gi       1.8Gi        14Gi        24Gi
Swap:          2.0Gi          0B       2.0Gi
```

- OS version

```
ashen@Stephanie-Lin:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.2 LTS
Release:        22.04
Codename:       jammy
```

- GCC version

```
ashen@Stephanie-Lin:~$ gcc --version
gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

- ICC version

```
ashen@Stephanie-Lin:~/Documents/Software_Analysis-git/HW2$ icc --version
icc: remark #10441: The Intel(R) C++ Compiler Classic (ICC) is deprecated and will be removed from
product release in the second half of 2023. The Intel(R) oneAPI DPC++/C++ Compiler (ICX) is the rec
ommended compiler moving forward. Please transition to use this compiler. Use '-diag-disable=10441'
 to disable this message.
icc (ICC) 2021.10.0 20230609
Copyright (C) 1985-2023 Intel Corporation.  All rights reserved.
```

Compile nsieve.c

1. Makefile generate *.spi

```
DEFINES = -DUNIX
FLAGS = $(DEFINES) -O3
SPI = -prof-gen=srcpos
NO_COMM = -diag-disable=10441

SRC =  $(wildcard *.c)
EXE = $(subst .,,$(SRC))
all: ${EXE}

ICC = icc $(FLAGS)

%c:     %.c
        @$(ICC) ${SPI} ${NO_COMM} $*.c -o $*
```

2. Execute “make”

```
ashen@Stephanie-Lin:~/Documents/Software_Analysis-git/HW2$ make
ashen@Stephanie-Lin:~/Documents/Software_Analysis-git/HW2$ ls -al
total 304
drwxrwxr-x 2 ashen ashen   4096 ┼  23 13:11 .
drwxrwxr-x 5 ashen ashen   4096 ┼  23 11:24 ..
-rwxr--r-- 1 ashen ashen    223 ┼  23 12:46 Makefile
-rwxrwxr-x 1 ashen ashen 111824 ┼  23 13:11 nsieve
-rw-rw-r-- 1 ashen ashen  22510 ┼  23 11:24 nsieve.c
-rw-rw-r-- 1 ashen ashen   5331 ┼  23 13:11 pgopti.spi
-rw-rw-r-- 1 ashen ashen      0 ┼  23 13:11 pgopti.spl
```

3. Execute “nsieve” to generate *.dyn

```
ashen@Stephanie-Lin:~/Documents/Software_Analysis-git/HW2$ ./nsieve

   Sieve of Eratosthenes (Scaled to 10 Iterations)
   Version 1.2b, 26 Sep 1992

   Array Size    Number   Last Prime     Linear    RunTime     MIPS
    (Bytes)   of Primes                Time(sec)    (Sec)
       8191       1899        16381      0.000      0.000  3772.5
      10000       2261        19997      0.001      0.001  3619.9
      20000       4202        39989      0.001      0.001  3332.9
      40000       7836        79999      0.002      0.003  3293.9
      80000      14683       160001      0.004      0.005  3372.1
     160000      27607       319993      0.009      0.010  3335.3
     320000      52073       639997      0.017      0.021  3276.4
     640000      98609      1279997      0.034      0.047  2978.4
    1280000     187133      2559989      0.069      0.105  2676.4
    2560000     356243      5119997      0.137      0.220  2573.5
    5120000     679460     10239989      0.275      0.455  2511.5
   10240000    1299068     20479999      0.549      0.913  2525.2
   20480000    2488465     40960001      1.099      2.145  2166.0
   40960000    4774994     81919993      2.198      5.127  1826.6

   Relative to 10 Iterations and the 8191 Array Size:
   Average RunTime =    0.001 (sec)
   High  MIPS      =   3772.5
   Low   MIPS      =   1826.6

ashen@Stephanie-Lin:~/Documents/Software_Analysis-git/HW2$ ls -al
total 308
drwxrwxr-x 2 ashen ashen   4096 ┼  23 13:12 .
drwxrwxr-x 5 ashen ashen   4096 ┼  23 11:24 ..
-rw-rw-r-- 1 ashen ashen   2672 ┼  23 13:12 653600b7_12005.dyn
```

4. Execute “profmerge” to generate *.dpi

```
ashen@Stephanie-Lin:~/Documents/Software_Analysis-git/HW2$ profmerge
ashen@Stephanie-Lin:~/Documents/Software_Analysis-git/HW2$ ls -al
total 316
drwxrwxr-x 2 ashen ashen   4096 ┼ 23 13:13 .
drwxrwxr-x 5 ashen ashen   4096 ┼ 23 11:24 ..
-rw-rw-r-- 1 ashen ashen   2672 ┼ 23 13:12 653600b7_12005.dyn
-rwxr--r-- 1 ashen ashen    223 ┼ 23 12:46 Makefile
-rwxrwxr-x 1 ashen ashen 111824 ┼ 23 13:11 nsieve
-rw-rw-r-- 1 ashen ashen  22510 ┼ 23 11:24 nsieve.c
-rw-rw-r-- 1 ashen ashen   5544 ┼ 23 13:13 pgopti.dpi
-rw-rw-r-- 1 ashen ashen      0 ┼ 23 13:13 pgopti.dpi.lock
-rw-rw-r-- 1 ashen ashen   5331 ┼ 23 13:11 pgopti.spi
-rw-rw-r-- 1 ashen ashen      0 ┼ 23 13:11 pgopti.spl
-rw-rw-r-- 1 ashen ashen 149654 ┼ 23 11:23 practice_codecov.pdf
ashen@Stephanie-Lin:~/Documents/Software_Analysis-git/HW2$
```

Codecov

1. Run “codecov” generate a html

```
ashen@Stephanie-Lin:~/Documents/Software_Analysis-git/HW2$ codecov -counts -prj nsieve -spi pgopti.spi -dpi pgopti.dpi
Intel(R) C++/Fortran Compiler Classic code-coverage tool, Version 2021.10.0 Build 20230609_000000
Copyright (C) 1985-2023 Intel Corporation.  All rights reserved.

33% .. 67% .. 100%
ashen@Stephanie-Lin:~/Documents/Software_Analysis-git/HW2$ ls -al
total 328
drwxrwxr-x 3 ashen ashen   4096 ┼ 23 13:16 .
drwxrwxr-x 5 ashen ashen   4096 ┼ 23 11:24 ..
-rw-rw-r-- 1 ashen ashen   2672 ┼ 23 13:12 653600b7_12005.dyn
drwxrwxr-x 2 ashen ashen   4096 ┼ 23 13:16 CodeCoverage
-rw-rw-r-- 1 ashen ashen    343 ┼ 23 13:16 CODE_COVERAGE.HTML
-rwxr--r-- 1 ashen ashen    223 ┼ 23 12:46 Makefile
-rwxrwxr-x 1 ashen ashen 111824 ┼ 23 13:11 nsieve
-rw-rw-r-- 1 ashen ashen  22510 ┼ 23 11:24 nsieve.c
-rw-rw-r-- 1 ashen ashen   5544 ┼ 23 13:13 pgopti.dpi
-rw-rw-r-- 1 ashen ashen      0 ┼ 23 13:13 pgopti.dpi.lock
-rw-rw-r-- 1 ashen ashen   5331 ┼ 23 13:11 pgopti.spi
-rw-rw-r-- 1 ashen ashen      0 ┼ 23 13:11 pgopti.spl
-rw-rw-r-- 1 ashen ashen 149654 ┼ 23 11:23 practice_codecov.pdf
```

2. Use “-txtbcvrg” to export the analysis into “result.txt”

```
ashen@Stephanie-Lin:~/Documents/Software_Analysis-git/HW2$ codecov -counts -prj nsieve -spi pgopti.spi -dpi pgopti.dpi -txtbcvrg result.txt
Intel(R) C++/Fortran Compiler Classic code-coverage tool, Version 2021.10.0 Build 20230609_000000
Copyright (C) 1985-2023 Intel Corporation.  All rights reserved.


ashen@Stephanie-Lin:~/Documents/Software_Analysis-git/HW2$ ls -al
total 328
drwxrwxr-x 3 ashen ashen   4096 ┼ 23 13:16 .
drwxrwxr-x 5 ashen ashen   4096 ┼ 23 11:24 ..
-rw-rw-r-- 1 ashen ashen   2672 ┼ 23 13:12 653600b7_12005.dyn
drwxrwxr-x 2 ashen ashen   4096 ┼ 23 13:16 CodeCoverage
-rw-rw-r-- 1 ashen ashen    343 ┼ 23 13:16 CODE_COVERAGE.HTML
-rwxr--r-- 1 ashen ashen    223 ┼ 23 12:46 Makefile
-rwxrwxr-x 1 ashen ashen 111824 ┼ 23 13:11 nsieve
-rw-rw-r-- 1 ashen ashen  22510 ┼ 23 11:24 nsieve.c
-rw-rw-r-- 1 ashen ashen   5544 ┼ 23 13:13 pgopti.dpi
-rw-rw-r-- 1 ashen ashen      0 ┼ 23 13:13 pgopti.dpi.lock
-rw-rw-r-- 1 ashen ashen   5331 ┼ 23 13:11 pgopti.spi
-rw-rw-r-- 1 ashen ashen      0 ┼ 23 13:11 pgopti.spl
-rw-rw-r-- 1 ashen ashen 149654 ┼ 23 11:23 practice_codecov.pdf
-rw-rw-r-- 1 ashen ashen   1563 ┼ 23 13:19 result.txt
```

Result

- Summary

Coverage Summary of nsieve

Files				Functions				Blocks			
total	cvrd	uncvrd	cvrg%	total	cvrd	uncvrd	cvrg%	total	cvrd	uncvrd	cvrg%
1	1	0	**100.00**	3	3	0	**100.00**	65	57	8	**87.69**

Covered Files in nsieve

Name	Functions			Blocks		
	total	cvrd	cvrg%	total	cvrd	cvrg%
nsieve.c	3	3	100.00	65	57	87.69

- Basic block example

```
236)    smips[i] = emips;
        ^ 13
237)
238)    if( ErrorFlag == 0L )
239)    {
240)    if( N_Prime != Number_Of_Primes[i] )
            ^ 13
241)    {
242)    printf("\n   Error --- Incorrect Number of Primes for Array: %ld\n",j);
        ^ 0
243)    printf("   Number of  Primes  Found is: %ld\n",N_Prime);
        ^ 0
244)    printf("   Correct Number of Primes is: %ld\n",Number_Of_Primes[i]);
        ^ 0
245)    ErrorFlag = 1L;
        ^ 0
246)    }
247)    }
248)
249)    if( ErrorFlag > 0L ) break;
            ^ 13
```