Київський національний університет імені Тараса Шевченка
радіофізичний факультет

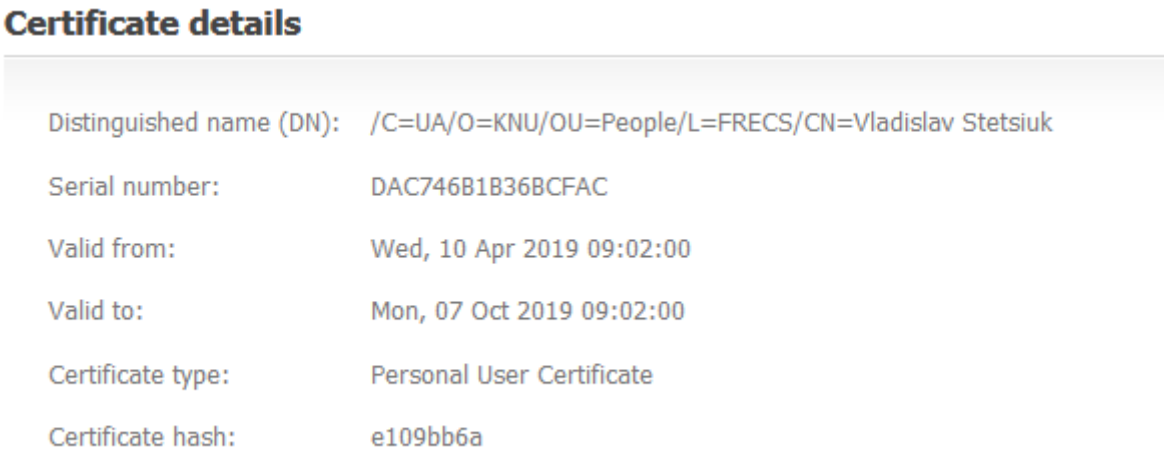
Лабораторна робота № 3
Тема:
«Дослідження оптимізації коду з використанням векторних розширень CPU»

Роботу виконав
студент 3 курсу
Комп’ютерна Інженерія
Стецюк Владислав

Київ 2019

Хід роботи

1. Отримайте доступ на обчислювальний кластер для роботи з Intel Compiler



2. Завантажте файли Intel® C++ Compiler - Using Auto-Vectorization Tutorial (https://software.intel.com/en-us/product-code-samples?topic=20813) на свій комп'ютер та в домашню директорію користувача обчислювального кластеру.

```
KNU:  :s3 [tb218 ~]$ wget https://software.intel.com/sites/default
/files/vec_samples_C_lin_20170911.tgz
```

```
[tb218@plus7 ~]$ tar -zxvf vec_samples_C_lin_20170911.tgz
./vec_samples/
./vec_samples/license.txt
./vec_samples/src/
./vec_samples/src/Multiply.h
./vec_samples/src/Driver.c
./vec_samples/src/Multiply.c
./vec_samples/build.bat
./vec_samples/vec_samples_2017.sln
./vec_samples/Makefile

[tb218@plus7 ~]$ cd vec_samples
[tb218@plus7 vec_samples]$ ls
build.bat    msvs2017      vec_samples_2013.sln
license.txt  readme.html   vec_samples_2015.sln
Makefile     resources     vec_samples_2017.sln
msvs2013     src
msvs2015     tutorial
```

3. Використовуючи інструкції в readme.html ознайомтесь та виконайте Tutorial на обчислювальному кластері.

```
KNU:  :s3 [tb218 vec_samples]$ ls
Makefile     msvs2017     vec_samples_2013.sln
build.bat    readme.html  vec_samples_2015.sln
license.txt  resources    vec_samples_2017.sln
msvs2013     src
msvs2015     tutorial
KNU:  :s3 [tb218 vec_samples]$ icc -O1 -std=c99 src/Multiply.c src
/Driver.c -o MatVector
KNU:  :s3 [tb218 vec_samples]$ ./MatVector

ROW:101 COL: 101
Execution time is 12.142 seconds
GigaFlops = 1.680282
Sum of result = 195853.999899
KNU:  :s3 [tb218 vec_samples]$ icc -std=c99 -O2 -D NOFUNCCALL -qop
t-report=1 -qopt-report-phase=vec src/Multiply.c src/Driver.c -o M
atVector
icc: remark #10397: optimization reports are generated in *.optrpt
 files in the output location
KNU:  :s3 [tb218 vec_samples]$ ./MatVector

ROW:101 COL: 101
Execution time is 4.123 seconds
GigaFlops = 4.948244
Sum of result = 195853.999899
KNU:  :s3 [tb218 vec_samples]$ cat Multiply.optrpt
Intel(R) Advisor can now assist with vectorization and show optimi
zation
  report messages with your source code.
See "https://software.intel.com/en-us/intel-advisor-xe" for detail
s.


Begin optimization report for: matvec(int, int, double (*)[*], dou
ble *, double *)

    Report from: Vector optimizations [vec]


LOOP BEGIN at src/Multiply.c(37,5)
   remark #25460: No loop optimizations reported

   LOOP BEGIN at src/Multiply.c(49,9)
      remark #25460: No loop optimizations reported
   LOOP END

   LOOP BEGIN at src/Multiply.c(49,9)
```

```
KNU:  :s3 [tb218 vec_samples]$ icc -std=c99 -qopt-report=2 -qopt-r
eport-phase=vec -D NOALIAS src/Multiply.c src/Driver.c -o MatVecto
r
icc: remark #10397: optimization reports are generated in *.optrpt
 files in the output location
KNU:  :s3 [tb218 vec_samples]$ ./MatVector

ROW:101 COL: 101
Execution time is 8.315 seconds
GigaFlops = 2.453751
Sum of result = 195853.999899
KNU:  :s3 [tb218 vec_samples]$ cat Multiply.optrpt
Intel(R) Advisor can now assist with vectorization and show optimi
zation
  report messages with your source code.
See "https://software.intel.com/en-us/intel-advisor-xe" for detail
s.


Begin optimization report for: matvec(int, int, double (*)[*], dou
ble *__restrict__, double *)

    Report from: Vector optimizations [vec]


LOOP BEGIN at src/Multiply.c(37,5)
   remark #15542: loop was not vectorized: inner loop was already
vectorized

   LOOP BEGIN at src/Multiply.c(49,9)
   <Peeled loop for vectorization>
   LOOP END

   LOOP BEGIN at src/Multiply.c(49,9)
      remark #15300: LOOP WAS VECTORIZED
   LOOP END

   LOOP BEGIN at src/Multiply.c(49,9)
   <Alternate Alignment Vectorized Loop>
   LOOP END

   LOOP BEGIN at src/Multiply.c(49,9)
   <Remainder loop for vectorization>
   LOOP END
LOOP END
```

```
KNU:  :s3 [tb218 vec_samples]$ =>> PBS: job killed: walltime 1834
exceeded limit 1800

KNU:  :s3 [tb218 vec_samples]$ icc -std=c99 -qopt-report=4 -qopt-r
eport-phase=vec -D NOALIAS -D ALIGNED src/Multiply.c src/Driver.c
-o MatVector
icc: remark #10397: optimization reports are generated in *.optrpt
 files in the output location
KNU:  :s3 [tb218 vec_samples]$ cat Multiply.optrpt
Intel(R) Advisor can now assist with vectorization and show optimi
zation
  report messages with your source code.
See "https://software.intel.com/en-us/intel-advisor-xe" for detail
s.

Intel(R) C Intel(R) 64 Compiler for applications running on Intel(
R) 64, Version 18.0.5.274 Build 20180823

Compiler options: -std=c99 -qopt-report=4 -qopt-report-phase=vec -
D NOALIAS -D ALIGNED -o MatVector

Begin optimization report for: matvec(int, int, double (*)[*], dou
ble *__restrict__, double *)

    Report from: Vector optimizations [vec]


LOOP BEGIN at src/Multiply.c(37,5)
   remark #15542: loop was not vectorized: inner loop was already
vectorized

   LOOP BEGIN at src/Multiply.c(49,9)
      remark #15388: vectorization support: reference a[i][j] has
aligned access   [ src/Multiply.c(50,21) ]
```

```
      remark #15305: vectorization support: vector length 2
      remark #15399: vectorization support: unroll factor set to 4
      remark #15309: vectorization support: normalized vectorizati
on overhead 0.594
      remark #15300: LOOP WAS VECTORIZED
      remark #15448: unmasked aligned unit stride loads: 2
      remark #15475: --- begin vector cost summary ---
      remark #15476: scalar cost: 10
      remark #15477: vector cost: 4.000
      remark #15478: estimated potential speedup: 2.410
      remark #15488: --- end vector cost summary ---
   LOOP END

   LOOP BEGIN at src/Multiply.c(49,9)
   <Remainder loop for vectorization>
      remark #15388: vectorization support: reference a[i][j] has
aligned access   [ src/Multiply.c(50,21) ]
      remark #15388: vectorization support: reference x[j] has ali
gned access   [ src/Multiply.c(50,31) ]
      remark #15335: remainder loop was not vectorized: vectorizat
ion possible but seems inefficient. Use vector always directive or
 -vec-threshold0 to override
      remark #15305: vectorization support: vector length 2
      remark #15309: vectorization support: normalized vectorizati
on overhead 2.417
   LOOP END
LOOP END
=================================================================
=========

KNU:  :s1 [tb218 vec_samples]$ icc -std=c99 -qopt-report=2 -qopt-r
eport-phase=vec -D NOALIAS -D ALIGNED -ipo src/Multiply.c src/Driv
er.c -o MatVector
```

```
KNU:  :s1 [tb218 vec_samples]$ cat ipo_out.optrpt
Intel(R) Advisor can now assist with vectorization and show optimi
zation
  report messages with your source code.
See "https://software.intel.com/en-us/intel-advisor-xe" for detail
s.


Begin optimization report for: main()

    Report from: Vector optimizations [vec]


LOOP BEGIN at src/Driver.c(152,16)
   remark #15542: loop was not vectorized: inner loop was already
vectorized

   LOOP BEGIN at src/Multiply.c(37,5) inlined into src/Driver.c(15
0,9)
      remark #15542: loop was not vectorized: inner loop was alrea
dy vectorized

      LOOP BEGIN at src/Multiply.c(49,9) inlined into src/Driver.c
(150,9)
         remark #15300: LOOP WAS VECTORIZED
      LOOP END

      LOOP BEGIN at src/Multiply.c(49,9) inlined into src/Driver.c
(150,9)
      <Remainder loop for vectorization>
         remark #15335: remainder loop was not vectorized: vectori
zation possible but seems inefficient. Use vector always directive
```

```
LOOP BEGIN at src/Driver.c(74,5) inlined into src/Driver.c(159,5)
<Remainder loop for vectorization>
LOOP END
=================================================================
=========

Begin optimization report for: init_matrix(int, int, double, doubl
e (*)[102])

    Report from: Vector optimizations [vec]


LOOP BEGIN at src/Driver.c(47,5)
   remark #15542: loop was not vectorized: inner loop was already
vectorized

   LOOP BEGIN at src/Driver.c(48,9)
      remark #15300: LOOP WAS VECTORIZED
   LOOP END

   LOOP BEGIN at src/Driver.c(48,9)
   <Remainder loop for vectorization>
   LOOP END
LOOP END

LOOP BEGIN at src/Driver.c(53,9)
   remark #15300: LOOP WAS VECTORIZED
LOOP END

LOOP BEGIN at src/Driver.c(53,9)
<Remainder loop for vectorization>
LOOP END
=================================================================
=========
```

```
KNU:  :s1 [tb218 vec_samples]$ ./MatVector

ROW:101 COL: 102
Execution time is 3.995 seconds
GigaFlops = 5.107272
Sum of result = 195853.999899
```

4. Оберіть будь-яку неінтрерактивну консольну програму мовою C/C++ (унікальну в межах групи, в гуглі більше ніж 50 програм)

a) Напишіть сценарій, що:

i) Компілює програму з різними оптимізаціями (-O) та виміряйте час її роботи. Якщо час досить малий - вимірюйте час роботи 1000 (чи 1000000) запусків алгоритму в циклі. Час роботи можна виміряти утилітою time.

ii) Отримує перелік всіх розширень процесору що підтримуються

iii) Для кожного розширення компілює Intel-компілятором окремий варіант оптимізованого коду (наприклад -x SSE2)

iv) Вимірює час виконання кожного варіанта оптимізованої програми

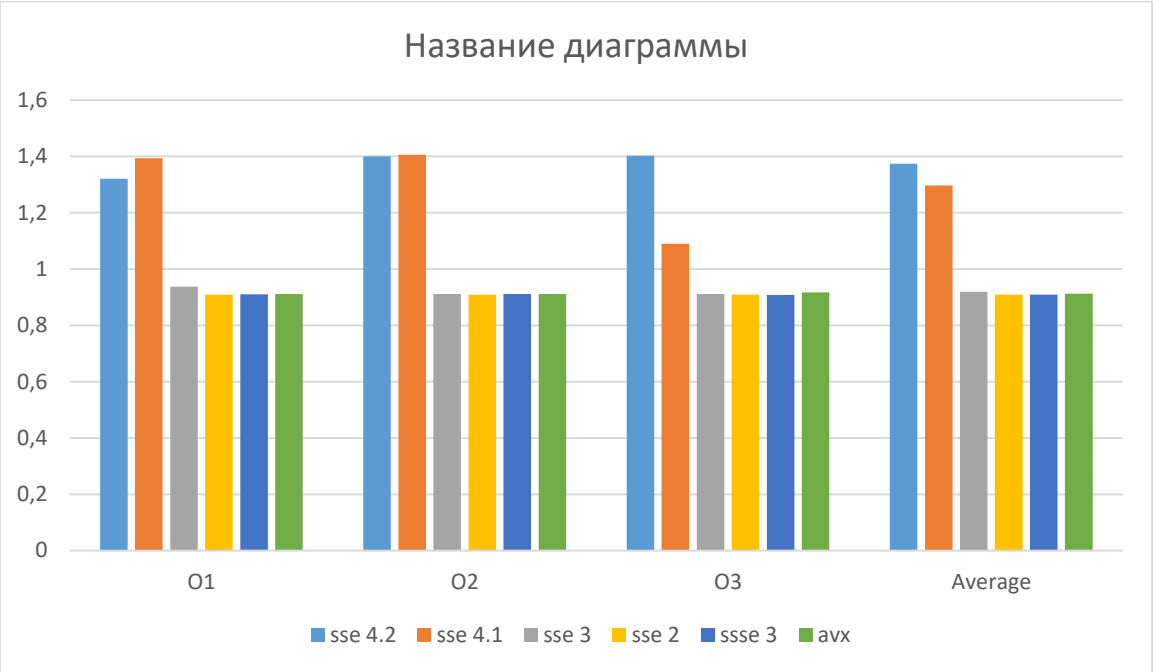
```
KNU:  :s1 [tb218 ~]$ cat date.c
#include <stdio.h>
int main()
{
 int daysInCurrentFebruary = 29;
 int daysInJanuary = 31;
 int daysInFebruary = daysInCurrentFebruary;
int daysInMarch = 31;
 int daysInApril = 30;
 int daysInMay = 31;
 int daysInJune = 30;
 int daysInJuly = 31;
 int daysInAugust = 31;
 int daysInSeptember = 30;
 int daysInOctober = 31;
int daysInNovember = 30;
int daysInDecember = 31;
int daysInFirstHalf = daysInJanuary + daysInFebruary + daysInMarch + daysInApril + daysInMay + daysInJune;
 int daysInSecondHalf = daysInJuly + daysInAugust + daysInSeptember + daysInOctober + daysInNovember + daysInDecember;
 printf("Days in the first half of the current year: %d\n", daysInFirstHalf);
printf("Days in the second half of the current year: %d\n", daysInSecondHalf);
 printf("Days in the current year: %d\n", daysInFirstHalf + daysInSecondHalf);
return 0;
 }
KNU:  :s1 [tb218 ~]$

KNU:  :s1 [tb218 ~]$ gcc date.c
KNU:  :s1 [tb218 ~]$ ls
ITAC   advisor          compiler_c  date.c      inspector  ipsxe2019_samples_lin_20190327.tgz  mkl   tbb
a.out  cluster_checker  compiler_f  index.html  ipp        licensing                          pstl  vtune_amplifier
KNU:  :s1 [tb218 ~]$ ./a.out
Days in the first half of the current year: 182
Days in the second half of the current year: 184
Days in the current year: 366
```

```
KNU:  :s1 [tb218 ~]$ time gcc date.c

real    0m0.075s
user    0m0.019s
sys     0m0.018s

KNU:  :s3 [tb218 ~]$ cat 1.sh
#!/bin/bash

flags=( "sse4.2" "sse4.1" "sse3" "sse2" "ssse3" "avx" )

for i in "${flags[@]}"; do
        for j in {1..3}; do
                gcc -O$j -m$i date.c -o Temp
                echo
                echo $i " " $j
                time `for i in {0..1000}; do ./Temp; done`
        done
done
```

```
KNU:  :s1 [tb218 ~]$ ./1.sh

sse4.2   1
./1.sh: line 10: Days: command not found

real    0m1.321s
user    0m0.200s
sys     0m1.111s

sse4.2   2
./1.sh: line 10: Days: command not found

real    0m1.400s
user    0m0.193s
sys     0m1.199s

sse4.2   3
./1.sh: line 10: Days: command not found

real    0m1.403s
user    0m0.156s
sys     0m1.237s

sse4.1   1
./1.sh: line 10: Days: command not found

real    0m1.394s
user    0m0.178s
sys     0m1.208s
```

```
ssse3   3
./1.sh: line 10: Days: command not found

real    0m0.908s
user    0m0.313s
sys     0m0.588s

avx   1
./1.sh: line 10: Days: command not found

real    0m0.911s
user    0m0.297s
sys     0m0.606s

avx   2
./1.sh: line 10: Days: command not found

real    0m0.911s
user    0m0.309s
sys     0m0.595s

avx   3
./1.sh: line 10: Days: command not found

real    0m0.917s
user    0m0.300s
sys     0m0.609s
```

	sse 4.2	sse 4.1	sse 3	sse 2	ssse 3	avx
O1	1,321	1,394	0,938	0,909	0,910	0,911
O2	1,400	1,406	0,911	0,909	0,911	0,911
O3	1,403	1,090	0,912	0,909	0,908	0,917
Average	1,3746	1,2966	0,920	0,909	0,9096	0,913



b. Запустіть задачу **в планувальник** обчислювального кластеру 5 разів (для статистики на різних нодах)

```
[tb218@plus7 ~]$ qsub -N MyJob -l nodes=1:ppn=1,walltime=00:30:00
1.sh
2789321
[tb218@plus7 ~]$ qsub -N MyJob -l nodes=1:ppn=1,walltime=00:30:00
1.sh
2789322
```

Як бачимо з отриманих даних, найшвидше програма виконується у випадку компіляції з третім методом оптимізації та розширенням процесора sse2.

Висновок: У процесі виконання лабораторної роботи було проведена ознайомлення з обчислювальним кластером та методами оптимізації виконання програми на C/C++ . За результатами виконання було написано сценарії та звіт з відповідними скріншотами.