

**Máster High Performance  
Computing  
HPC TOOLS**

**ACTIVIDAD 3**

**Nombre y Apellido:  
Betzabeth León**

- **Análisis con Valgrind Memcheck**

El resumen que muestra esta herramienta es la siguiente:

```
==20008==
==20008== LEAK SUMMARY:
==20008==    definitely lost: 67,125,304 bytes in 11 blocks
==20008==    indirectly lost: 0 bytes in 0 blocks
==20008==    possibly lost: 67,108,864 bytes in 2 blocks
==20008==    still reachable: 1,624 bytes in 5 blocks
==20008==    suppressed: 0 bytes in 0 blocks
==20008== Reachable blocks (those to which a pointer was found) are not shown.
==20008== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==20008==
==20008== Use --track-origins=yes to see where uninitialised values come from
==20008== For lists of detected and suppressed errors, rerun with: -s
==20008== ERROR SUMMARY: 3602 errors from 23 contexts (suppressed: 0 from 0)
```

Se encontraron problemas de los siguientes tipos:

```
==20008== Invalid read of size 8
==20008==    at 0x401A8D: gauss (dgesvmodv.c:137)
==20008==    by 0x401942: my_dgesv (dgesvmodv.c:118)
==20008==    by 0x401572: main (dgesvmodv.c:44)
==20008== Address 0xff29040 is 0 bytes after a block of size 33,554,432 alloc'd
==20008==    at 0x48335ED: malloc (vg_replace_malloc.c:307)
==20008==    by 0x4016F9: spd_generate_matrix (dgesvmodv.c:75)
==20008==    by 0x4012E3: main (dgesvmodv.c:21)
==20008==
==20008== Invalid write of size 8
==20008==    at 0x401AC1: gauss (dgesvmodv.c:139)
==20008==    by 0x401942: my_dgesv (dgesvmodv.c:118)
==20008==    by 0x401572: main (dgesvmodv.c:44)
==20008== Address 0xff29040 is 0 bytes after a block of size 33,554,432 alloc'd
==20008==    at 0x48335ED: malloc (vg_replace_malloc.c:307)
==20008==    by 0x4016F9: spd_generate_matrix (dgesvmodv.c:75)
==20008==    by 0x4012E3: main (dgesvmodv.c:21)
```

1. Invalid read of size 8:  
El problema detectado es que intentó leer 8 bytes, la primera línea que marca es con la matriz “a” que es de tipo double, el problema se arregla ajustando el tamaño en el malloc.
2. Invalid write of size 8: Intenta escribir datos en una zona de memoria donde no debería.

- **Paralelización con OpenMP**

Se ejecutó gprof para conocer las funciones más costosas:

```
[curso389@c6601 final]$ gprof ./dgesvmod
Flat profile:

Each sample counts as 0.01 seconds.
%   cumulative   self           calls   self   total    name
time  seconds    seconds               s/call  s/call
-----
71.10   26.41    26.41                1    26.41   26.45  spd_generate_matrix
28.51   37.00    10.59                1    10.59   10.59   gauss
  0.40   37.15     0.15                 4     0.04    0.04  generate_matrix
  0.00   37.15     0.00                 1     0.00    0.00  check_result
  0.00   37.15     0.00                 1     0.00   10.59  my_dgesv
```

```

granularity: each sample hit covers 2 byte(s) for 0.03% of 37.15 seconds
index % time    self  children   called    name
[1]   100.0      0.00   37.15      1/1      <spontaneous>
      26.41      0.04      1/1      main [1]
      0.00   10.59      1/1      spd_generate_matrix [2]
      0.11      0.00      3/4      my_dgesv [4]
      0.00      0.00      1/1      generate_matrix [5]
      0.00      0.00      1/1      check_result [6]
-----
[2]    71.2     26.41      0.04      1/1      main [1]
      26.41      0.04      1/1      spd_generate_matrix [2]
      0.04      0.00      1/4      generate_matrix [5]
-----
[3]    28.5     10.59      0.00      1/1      my_dgesv [4]
      10.59      0.00      1/1      gauss [3]
-----
[4]    28.5      0.00     10.59      1/1      main [1]
      0.00     10.59      1/1      my_dgesv [4]
      10.59      0.00      1/1      gauss [3]
-----
[5]     0.4      0.04      0.00      1/4      spd_generate_matrix [2]
      0.11      0.00      3/4      main [1]
      0.15      0.00      4/4      generate_matrix [5]
-----
[6]     0.0      0.00      0.00      1/1      main [1]
      0.00      0.00      1/1      check_result [6]
-----

```

Se paralelizó de la siguiente forma:

En la función spd de generar la matriz:

```
#pragma omp parallel for private(i,j,k) reduction(+:sum)
```

En la función gauss:

```
#pragma omp parallel for private(Col,C1,C2) shared(a)
```

Los tiempos obtenidos y el speedup fueron los siguientes:

Nº threads	Tamaño de la Matriz					
	2048		4096		6144	
	Tiempo en segundos	Speed up	Tiempo en segundos	Speed up	Tiempo en segundos	Speed up
Secuencial	2.069932		6.836334		22.536812	
2 Threads	2.041512	1.0139	6.498397	1.0520	20.528634	1.0978
4 Threads	2.001550	1.0200	6.226531	1.0437	21.338141	0.9621
8 Threads	1.794266	1.1155	5.128208	1.2142	20.552369	1.0382

- **Análisis con el Intel Vtune Amplifier**

En la siguiente imagen se pueden observar un listado de las funciones más activas del programa e indica que mejorando estos hotspot se puede mejorar el rendimiento.

**Elapsed Time**: 182.313s  
**CPU Time**: 165.360s  
 Total Thread Count: 1  
 Paused Time: 0s

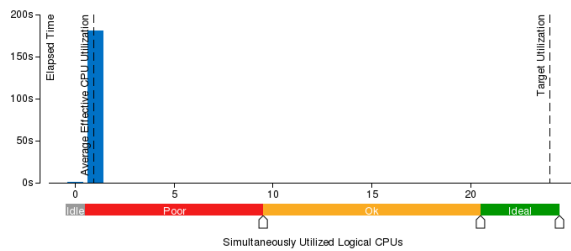
**Top Hotspots**

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time
main	dgesvmodparvtune	163.200s
random	libc.so.6	1.612s
random_r	libc.so.6	0.310s
__intel_avx_rep_memcpy	dgesvmodparvtune	0.110s
rand	libc.so.6	0.080s
[Others]		0.048s

### Effective CPU Utilization Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU utilization value.



### Collection and Platform Info

This section provides information about this collection, including result set size and collection platform data.

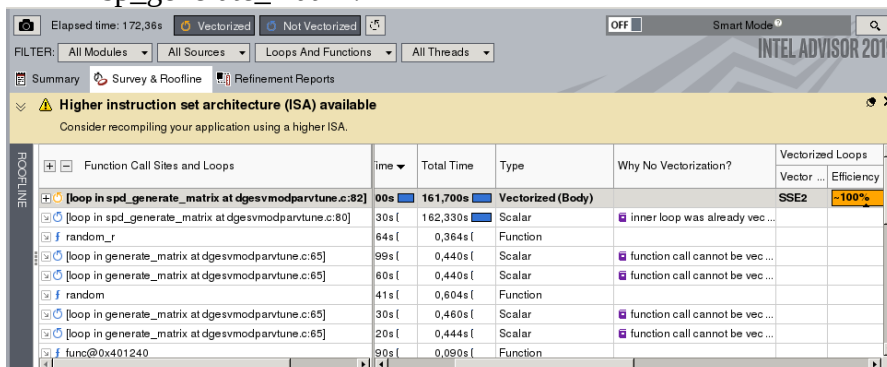
Application Command Line: /home/ulc/cursos/cursos389/hpdcolls/final/dgesvmodparvtune.6144  
Operating System: 3.10.0-852.14.4.el7.x86\_64 NAME="Red Hat Enterprise Linux Server" VERSION="7.5 (Maipo)"  
ID="rhel" ID\_LIKE="fedora" VARIANT="Server" VARIANT\_ID="server" VERSION\_ID="7.5"  
PRETTY\_NAME="Red Hat Enterprise Linux Server 7.5 (Maipo)" ANSI\_COLOR="0;31"  
CPE\_NAME="cpe:/o:redhat:enterprise\_linux:7.5:GA:server" HOME\_URL="https://www.redhat.com/"  
BUG\_REPORT\_URL="https://bugzilla.redhat.com/" REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 7" REDHAT\_BUGZILLA\_PRODUCT\_VERSION=7.5  
REDHAT\_SUPPORT\_PRODUCT="Red Hat Enterprise Linux"  
REDHAT\_SUPPORT\_PRODUCT\_VERSION="7.5"  
Computer Name: c6601  
Result Size: 5 MB  
Collection start time: 16:57:22 16/01/2021 UTC  
Collection stop time: 17:00:24 16/01/2021 UTC  
Collector Type: User-mode sampling and tracing

### CPU

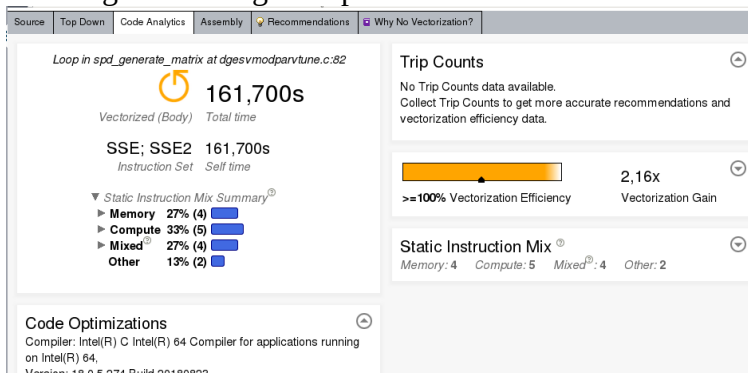
Name: Intel(R) Xeon(R) E5/E7 v3 Processor code named Haswell  
Frequency: 2.5 GHz  
Logical CPU Count: 24

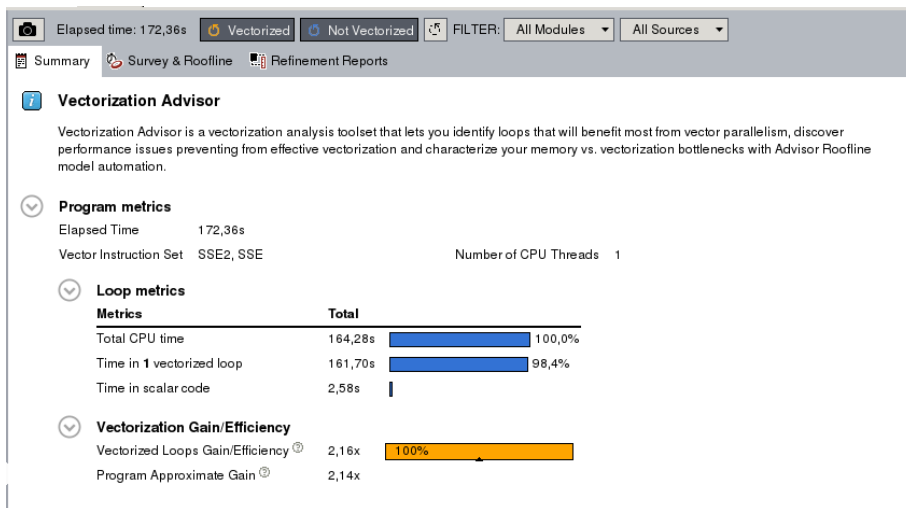
### • *Análisis con Intel Advisor*

En la siguiente imagen se puede observar el perfil por el peso de cada una de las funciones y lo que está en naranja significa que se puede vectorizar es un bucle de la función `sp_generate_matrix`:



En la siguiente imagen se puede observar la eficiencia de la vectorización:





La recomendación es vectorizar el bucle dentro de una de las funciones del programa, como se muestra en las siguientes imágenes:

