

2º curso / 2º cuatr.
Grado Ing. Inform.
Doble Grado Ing.
Inform. y Mat.

Arquitectura de Computadores (AC)

Cuaderno de prácticas.

Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos):

Grupo de prácticas:

Fecha de entrega:

Fecha evaluación en clase:

1. Incorpore volcados de pantalla que muestren lo que devuelve `lscpu` en `atcgrid` y en su PC.

CAPTURAS:

```
[GuilleMuriel B3estudiante30@atcgrid:~] 2018-03-02 viernes
$cat STDIN.o63156
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                24
On-line CPU(s) list:   0-23
Thread(s) per core:    2
Core(s) per socket:    6
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                44
Model name:            Intel(R) Xeon(R) CPU           E5645  @ 2.40GHz
Stepping:              2
CPU MHz:               1637.124
CPU max MHz:           2401.0000
CPU min MHz:           1600.0000
BogoMIPS:              4800.14
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              12288K
NUMA node0 CPU(s):    0-5,12-17
NUMA node1 CPU(s):    6-11,18-23
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss
ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf pni
dtes64 monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid dca sse4_1 sse4_2 popcnt lahf_lm epb pti retpoline tpr_shadow vnmi
flexpriority ept vpid dtherm ida arat
```

```
guillesta@guillesta:~$ lscpu
Architecture:          x86_64
modo(s) de operación de las CPUs: 32-bit, 64-bit
Orden de bytes:        Little Endian
CPU(s):                8
On-line CPU(s) list:   0-7
Hilo(s) de procesamiento por núcleo: 2
Núcleo(s) por «socket»: 4
Socket(s):             1
Modo(s) NUMA:          1
ID de fabricante:      GenuineIntel
Familia de CPU:        6
Modelo:               94
Model name:            Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
Revisión:              3
CPU MHz:               2600.000
CPU max MHz:           3500.0000
CPU min MHz:           800.0000
BogoMIPS:              5184.00
Virtualización:        VT-x
Caché L1d:             32K
Caché L1i:             32K
Caché L2:              256K
Caché L3:              6144K
NUMA node0 CPU(s):    0-7
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pb
e syscall nx pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf tsc_known_freq pni
pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx f16c rdrand l
ahf_lm abm 3dnowprefetch cpuid_fault epb invpcid_single pti intel_pt tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1 hle avx2
snep bmi2 erms invpcid rtm mpx rdseed adx snap clflushopt xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp_notify hwp_act_windo
w hwp_epp
```

Conteste a las siguientes preguntas:

- a. ¿Cuántos cores físicos y cuántos cores lógicos tiene `atcgrid` de prácticas o su PC?

RESPUESTA: Mi PC tiene 8 físicos y 16 lógicos.

b. ¿Cuántos cores físicos y cuántos cores lógicos tiene un nodo de atcgrid?

RESPUESTA: Cada nodo tiene 2 Socket, cada socket tiene 6 cpu's, por cada cpu hay 2 threads. En total hay 12 cores físicos y 24 cores lógicos por cada nodo.

2. En el Listado 1 se puede ver un código fuente C que calcula la suma de dos vectores y en el Listado 2 una versión con C++:

```
v3 = v1 + v2; v3(i) = v1(i) + v2(i), i=0,...N-1
```

Los códigos utilizan directivas del compilador para fijar el tipo de variable de los vectores (v1, v2 y v3). En los comentarios que hay al principio de los códigos se indica cómo hay que compilarlos. Los vectores pueden ser:

- Variables locales: descomentando en el código `#define VECTOR_LOCAL` y comentando `#define VECTOR_GLOBAL` y `#define VECTOR_DYNAMIC`
- Variables globales: descomentando `#define VECTOR_GLOBAL` y comentando `#define VECTOR_LOCAL` y `#define VECTOR_DYNAMIC`
- Variables dinámicas: descomentando `#define VECTOR_DYNAMIC` y comentando `#define VECTOR_LOCAL` y `#define VECTOR_GLOBAL`. Si se usan los códigos tal y como están en Listado 1 y Listado 2, sin hacer ningún cambio, los vectores (v1, v2 y v3) serán variables dinámicas.

Por tanto, se debe definir sólo una de las siguientes constantes: `VECTOR_LOCAL`, `VECTOR_GLOBAL` o `VECTOR_DYNAMIC`.

- a. En los dos códigos (Listado 1 y Listado 2) se utiliza la función `clock_gettime()` para obtener el tiempo de ejecución del trozo de código que calcula la suma de vectores. En el código se imprime la variable `ncgt`, ¿qué contiene esta variable? ¿qué información devuelve exactamente la función `clock_gettime()`? ¿en qué estructura de datos devuelve `clock_gettime()` la información (indicar el tipo de estructura de datos y describir la estructura de datos)?

RESPUESTA:

La función `clock_gettime()` obtiene el valor de un instante de tiempo determinado, lo almacena en una estructura “timespec”, que está formada por los segundos y nanosegundos en el que se llamó la función. En la variable `ncgt` se almacena la diferencia entre `cgt1` (antes de iniciar el bucle) y `cgt2` (una vez finalizado el bucle), así sabemos el tiempo que ha tardado.

La estructura `timespec` está almacenada en la cabecera `time.h` y es la siguiente:

```
struct timespec {
    time_t  tv_sec;    /* seconds */
    long    tv_nsec;   /* nanoseconds */
};
```

- b. Escribir en el cuaderno de prácticas las diferencias que hay entre el código fuente C y el código fuente C++ para la suma de vectores.

RESPUESTA:

Descripción diferencia	En C	En C++
Salida de datos	Printf()	cout<<

Definir tamaño de vectores dinámicos	Malloc()	New double[]
Liberar espacio de vectores	Free()	delete[]
Cabeceras	.h	Solamente time.h
Datos finales redondeados	Ninguno	Todos

3. Generar el ejecutable del código fuente C del Listado 1 para vectores locales (para ello antes de compilar debe descomentar la definición de VECTOR_LOCAL y comentar las definiciones de VECTOR_GLOBAL y VECTOR_DYNAMIC). Incorporar volcados de pantalla que demuestren la ejecución correcta en atcgrid o en su PC.

RESPUESTA: He compilado, subido y ejecutado el comando `echo './SumaVectores 20' | qsub -q ac .`

```
[GuilleMuriel B3estudiante30@atcgrid:~] 2018-03-08 jueves
$echo './SumaVectores 20' | qsub -q ac
65768.atcgrid
[GuilleMuriel B3estudiante30@atcgrid:~] 2018-03-08 jueves
$cat STDIN.e65768
Tiempo(seg.):0.000000215      / Tamaño Vectores:20      / V1[0]+V2[0]=V3[0](2.000000+2.000000=4.000000) / / V1[19]+V2[19]=V3[19](3.900000+0.100000=4.000000)
/
```

4. Ejecutar en atcgrid el código generado en el apartado anterior usando el script del Listado 3. Generar el ejecutable usando la opción de optimización `-O2` tal y como se indica en el comentario que hay al principio del programa. Ejecutar el código también en su PC para los mismos tamaños. ¿Se obtiene error para alguno de los tamaños? En caso afirmativo, ¿a qué se debe este error? (Incorporar volcados de pantalla)

RESPUESTA: Ejecuto en atcgrid el script con `echo './SumaVectores.sh' | qsub -q ac`. Veo que me da errores, adjunto captura del error. El error es violación del segmento. Supongo que por que se ha desbordado el tamaño de la pila.

```
[GuilleMuriel B3estudiante30@atcgrid:~] 2018-03-08 jueves
$echo './SumaVectores.sh' | qsub -q ac
65783.atcgrid
[GuilleMuriel B3estudiante30@atcgrid:~] 2018-03-08 jueves
$cat STDIN.e65783
./SumaVectores.sh: line 20: 6034 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 6037 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 6041 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 6045 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 6052 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 6056 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 6060 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 6063 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
```

En mi PC ejecuto con `./SumaVectores.sh`. También aparece el error de violación del segmento al desbordarse la pila. Adjunto captura.

```

guillesiesta@guillesiesta:~$ ./SumaVectores.sh
Id. usuario del trabajo:
Id. del trabajo:
Nombre del trabajo especificado por usuario:
Nodo que ejecuta qsub:
Directorio en el que se ha ejecutado qsub:
Cola:
Nodos asignados al trabajo:
Tiempo(seg.):0.000969408 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tiempo(seg.):0.002109792 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.004158859 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
./SumaVectores.sh: línea 20: 7127 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 20: 7129 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 20: 7131 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 20: 7133 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 20: 7135 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 20: 7137 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 20: 7139 Violación de segmento ('core' generado) ./SumaVectoresC $N
./SumaVectores.sh: línea 20: 7141 Violación de segmento ('core' generado) ./SumaVectoresC $N
guillesiesta@guillesiesta:~$

```

5. Generar los ejecutables del código fuente C para vectores globales y para dinámicos. Genere el ejecutable usando `-O2`. Ejecutar los dos códigos en atcgrid usando un script como el del Listado 3 (hay que poner en el script el nombre de los ficheros ejecutables generados en este ejercicio) para el mismo rango de tamaños utilizado en el ejercicio anterior. Ejecutar también los códigos en su PC. ¿Se obtiene error usando vectores globales o dinámicos? ¿A qué cree que es debido? (Incorporar volcados de pantalla)

RESPUESTA: No se obtiene error en ninguno de los casos. Pues en el caso de los vectores dinámicos, las variables se almacenan en el heap y en el caso de los vectores globales, éstas se almacenan en memoria. Adjunto capturas de mi PC para casos Dinámico y Global, respectivamente. Después, adjunto para atcgrid los casos Dinámico y Global respectivamente.

PC Dinámico y Global

```

guillesiesta@guillesiesta:~$ ./SumaVectoresDina.sh
Id. usuario del trabajo:
Id. del trabajo:
Nombre del trabajo especificado por usuario:
Nodo que ejecuta qsub:
Directorio en el que se ha ejecutado qsub:
Cola:
Nodos asignados al trabajo:
Tiempo(seg.):0.001033766 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tiempo(seg.):0.000593800 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.001247433 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tiempo(seg.):0.002731407 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tiempo(seg.):0.000054565 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.009242706 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.015313497 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.031207748 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.050906604 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.117261084 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.227973608 / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / / V1[67108863]+V2[67108863]=V3[13421772.700000+0.100000=13421772.800000) /
guillesiesta@guillesiesta:~$

```

```

guille@tesla:guille$ cat:~$ ./SumaVectoresGlob.sh
Id. usuario del trabajo:
Id. del trabajo:
Nombre del trabajo especificado por usuario:
Nodo que ejecuta qsub:
Directorio en el que se ha ejecutado qsub:
Cola:
Nodos asignados al trabajo:
Tiempo(seg.):0.000960166 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.100
000+0.100000=13107.200000) /
Tiempo(seg.):0.000613510 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131071]=V3[131071](2621
4.300000+0.100000=26214.400000) /
Tiempo(seg.):0.001403766 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262143]=V3[262143](5242
8.700000+0.100000=52428.800000) /
Tiempo(seg.):0.002926896 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[524287]=V3[524287](104
857.500000+0.100000=104857.600000) /
Tiempo(seg.):0.00572223 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575
](209715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.009285249 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151
](419430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.015764633 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303
](838860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.029177900 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V2[8388607]=V3[838860
7](1677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.057390228 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[16
777215](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.113061707 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33
554431](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.118630153 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33
554431](6710886.300000+0.100000=6710886.400000) /

```

Atcgrid Dinámico y Global

```

[GuilleMuriel B3estudiante30@atcgrid:~] 2018-03-08 jueves
$ echo './SumaVectoresDina.sh' | qsub -q ac
65784.atcgrid
[GuilleMuriel B3estudiante30@atcgrid:~] 2018-03-08 jueves
$ cat STDIN.o65784
Id. usuario del trabajo: B3estudiante30
Id. del trabajo: 65784.atcgrid
Nombre del trabajo especificado por usuario: STDIN
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/B3estudiante30
Cola: ac
Nodos asignados al trabajo:
atcgrid1

```

```

Tiempo(seg.):0.000453827 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13
107.200000) /
Tiempo(seg.):0.000570103 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131071]=V3[131071](26214.300000+0.1000
00=26214.400000) /
Tiempo(seg.):0.001688826 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262143]=V3[262143](52428.700000+0.1000
00=52428.800000) /
Tiempo(seg.):0.003280435 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[524287]=V3[524287](104857.500000+0.10
0000=104857.600000) /
Tiempo(seg.):0.005754238 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575](209715.100000
+0.100000=209715.200000) /
Tiempo(seg.):0.011400870 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151](419430.300000
+0.100000=419430.400000) /
Tiempo(seg.):0.022743579 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](838860.700000
+0.100000=838860.800000) /
Tiempo(seg.):0.044980780 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V2[8388607]=V3[8388607](1677721.5000
00+0.100000=1677721.600000) /
Tiempo(seg.):0.088555162 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[16777215](3355443
.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.175978828 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33554431](6710886
.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.365185938 / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / / V1[67108863]+V2[67108863]=V3[67108863](13421
72.700000+0.100000=13421772.800000) /

```

```

[GuilleMuriel B3estudiante30@atcgrid:~] 2018-03-08 jueves
$ echo './SumaVectoresGlob.sh' | qsub -q ac
65785.atcgrid
[GuilleMuriel B3estudiante30@atcgrid:~] 2018-03-08 jueves
$ cat STDIN.o65785
Id. usuario del trabajo: B3estudiante30
Id. del trabajo: 65785.atcgrid
Nombre del trabajo especificado por usuario: STDIN
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/B3estudiante30
Cola: ac
Nodos asignados al trabajo:

```


atcgrid1			
Tiempo(seg.):0.00267823	/ Tamaño Vectores:65536	/ V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) /	V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
Tiempo(seg.):0.000659835	/ Tamaño Vectores:131072	/ V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) /	V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.001119468	/ Tamaño Vectores:262144	/ V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) /	V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tiempo(seg.):0.002954518	/ Tamaño Vectores:524288	/ V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) /	V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tiempo(seg.):0.005691482	/ Tamaño Vectores:1048576	/ V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) /	V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.012699901	/ Tamaño Vectores:2097152	/ V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) /	V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.024303416	/ Tamaño Vectores:4194304	/ V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) /	V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.045519345	/ Tamaño Vectores:8388608	/ V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) /	V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.090390887	/ Tamaño Vectores:16777216	/ V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) /	V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.182692597	/ Tamaño Vectores:33554432	/ V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) /	V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.184340784	/ Tamaño Vectores:33554432	/ V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) /	V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /

6. Rellenar una tabla como la Tabla 1 para atcgrid y otra para su PC con los tiempos de ejecución obtenidos en los ejercicios anteriores para el trozo de código que realiza la suma de vectores. En la columna “Bytes de un vector” hay que poner el total de bytes reservado para un vector. Ayudándose de una hoja de cálculo represente en una misma gráfica los tiempos de ejecución obtenidos en atcgrid y en su PC para vectores locales, globales y dinámicos (eje y) en función del tamaño en bytes de un vector (los valores de la segunda columna de la tabla, que están en escala logarítmica, deben estar en el eje x). Utilice escala logarítmica en el eje de ordenadas (eje y). ¿Hay diferencias en los tiempos de ejecución?

RESPUESTA: Si hay diferencia, pero apenas es apreciable. Quizá se deberían de agrandar aún más los datos y trabajar con vectores más grandes. Curiosamente, mi pc es más lento que atcgrid con tamaños pequeños de variables, pero mientras estos aumentan, mi pc es más rápido.

(Tablas al final del documento)

7. Modificar el código fuente C para que el límite de los vectores cuando se declaran como variables globales sea igual al máximo número que se puede almacenar en la variable N ($\text{MAX}=2^{32}-1$). Generar el ejecutable usando variables globales. ¿Qué ocurre? ¿A qué es debido? Razone además por qué el máximo número que se puede almacenar en N es $2^{32}-1$.

RESPUESTA: He borrado el `#define MAX` que había y he añadido este `#define MAX 4294967295` ($= (2^{32}) - 1$). Me da el siguiente error:

```
gulliesista@gulliesista:~$ gcc -O2 SumaVectoresC.c -o SumaVectoresG32 -lrt
/tmp/cc0C3zfn.o: En la función 'main':
SumaVectoresC.c:(.text.startup+0x79): reubicación truncada para ajustar: R_X86_64_325 contra el símbolo 'v2' definido en la sección COMMON en /tmp/cc0C3zfn.o
SumaVectoresC.c:(.text.startup+0xc0): reubicación truncada para ajustar: R_X86_64_325 contra el símbolo 'v2' definido en la sección COMMON en /tmp/cc0C3zfn.o
SumaVectoresC.c:(.text.startup+0xc8): reubicación truncada para ajustar: R_X86_64_325 contra el símbolo 'v3' definido en la sección COMMON en /tmp/cc0C3zfn.o
SumaVectoresC.c:(.text.startup+0xfc): reubicación truncada para ajustar: R_X86_64_325 contra el símbolo 'v3' definido en la sección COMMON en /tmp/cc0C3zfn.o
SumaVectoresC.c:(.text.startup+0x115): reubicación truncada para ajustar: R_X86_64_325 contra el símbolo 'v2' definido en la sección COMMON en /tmp/cc0C3zfn.o
SumaVectoresC.c:(.text.startup+0x12b): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v3' definido en la sección COMMON en /tmp/cc0C3zfn.o
SumaVectoresC.c:(.text.startup+0x135): reubicación truncada para ajustar: R_X86_64_PC32 contra el símbolo 'v2' definido en la sección COMMON en /tmp/cc0C3zfn.o
collect2: error: ld returned 1 exit status
gulliesista@gulliesista:~$
```

Esto se debe a que me estoy pasando del tamaño y el compilador detecta un desbordamiento. $(2^{32})-1$ es el máximo número que se puede representar en un entero de 32bits, y le restamos 1 porque hay que tener en cuenta el 0. En otras palabras, N es un entero sin signo, por lo que tiene 4 bytes que son 32 bits. Como también se almacena el 0, el último que podemos almacenar es el $2^{32} - 1$.

Tabla 1 . Mi PC

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000642967	0.000677405	0.000664380
131072	1048576	0.001683240	0.001099099	0.001357567
262144	2097152	0.002791713	0.000972865	0.002822156
524288	4194304	-	0.004863590	0.004711733
1048576	8388608	-	0.003870953	0.003799809
2097152	16777216	-	0.007461098	0.007306248
4194304	33554432	-	0.014970263	0.015117268
8388608	67108864	-	0.029329954	0.029598085
16777216	134217728	-	0.058133822	0.058798993
33554432	268435456	-	0.114857314	0.116304655
67108864	536870912	-	0.115434053	0.234958818

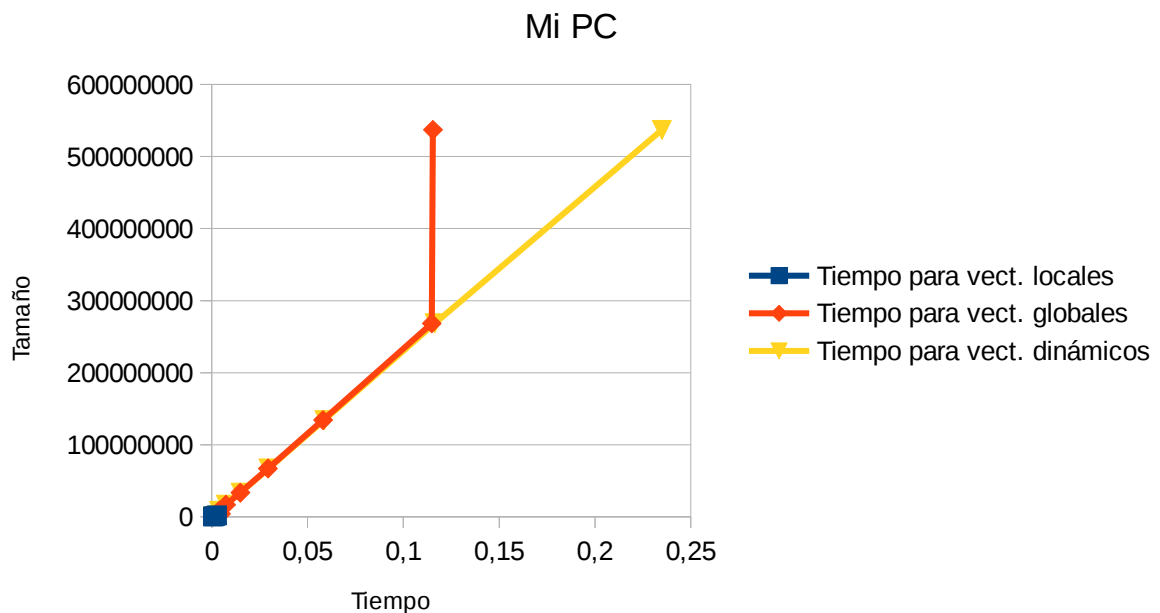


Tabla 2 .ATCGRID

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000425935	0.000575647	0.000461569
131072	1048576	0.000853508	0.000580670	0.000575647
262144	2097152	0.001692991	0.001566727	0.001890966
524288	4194304	-	0.002780018	0.002975523
1048576	8388608	-	0.005162615	0.004621280
2097152	16777216	-	0.009155163	0.008834713
4194304	33554432	-	0.017195837	0.017012498
8388608	67108864	-	0.034055641	0.033033012
16777216	134217728	-	0.069058265	0.065239206
33554432	268435456	-	0.133858142	0.129281346
67108864	536870912	-	0.135180420	0.256373192

