

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): David Sánchez Jiménez

Grupo de prácticas: A3

Fecha de entrega: 08/04/2018

Fecha evaluación en clase: 09/03/2018

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

CAPTURAS:

Captura del comando `lscpu` en mi PC:

```
lun 5 mar - 17:14 ~/Escritorio/ETSIIT/2º Curso/2º Cuatrimestre/AC/Practicas/Practica_0
@david$ lscpu
Architectura:                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:                      42
Model name:                  Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz
Revisión:                    7
CPU MHz:                     2195.213
CPU max MHz:                  3100.0000
CPU min MHz:                  800.0000
BogoMIPS:                     4390.42
Virtualización:              VT-x
Cache L1d:                   32K
Cache L1i:                   32K
Cache L2:                     256K
Cache 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 t
m pbe syscall nx rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl
vmx est tm2 sse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic popcnt tsc_deadline_timer aes xsave avx lahf_lm epb pti retpoline tpr_shadow vnmi flexpriori
ty ept vpid xsaveopt dtherm ida arat pln pts
```

Captura del comando lscpu del frontend de atcgrid:

```
A3estudiante24@atcgird:~$ lscpu
Arquitectura:          x86_64
modo(s) de operación de las CPUs:  32-bit, 64-bit
Orden de los bytes:    Little Endian
CPU(s):                1
Lista de la(s) CPU(s) en línea:    0
Hilo(s) de procesamiento por núcleo: 1
Núcleo(s) por «socket»: 1
«Socket(s)»:          1
Modo(s) NUMA:         1
ID de fabricante:     AuthenticAMD
Familia de CPU:        15
Modelo:                47
Nombre del modelo:     AMD Athlon(tm) 64 Processor 3000+
Revisión:              2
CPU MHz:               1802.252
BogoMIPS:              3604.50
caché L1d:             64K
caché L1i:             64K
caché L2:              512K
CPU(s) del nodo NUMA 0: 0
Indicadores:           tpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx mmxext
t fxsr opt lm 3dnowext 3dnow rep good nopl cpuid extd apicid pni lahf lm 3dnowprefetch rdtscp lm rsvb ctvsw vmxcall
```

Captura del comando lscpu del frontend de atcgrid:

```
A3estudiante24@atcgrid-1$ echo 'lscpu' | qsub -q ac
64858.atcgrid
[A3estudiante24@atcgrid-1$ cat STDIN.064858
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                 24
CPU(s)0:                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:                1599.498
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 topology nonstop tsc cpuid aperfmperf pni dtes64 monitor ds_cpl vmx smx est tm2 ss
                        pcid dca sse4_1 sse4_2 popcnt lahf_lm epb pti retpoline tpr_shadow vnmi flexpriority ept vpid dtherm ida arat
```

Conteste a las siguientes preguntas:

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

RESPUESTA:

Mi PC tiene 4 cores físicos y 8 lógicos

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

RESPUESTA:

atcgrid tiene un nodo frontend y tres nodos de cómputo (atcgrid1, atcgrid2, atcgrid3)

- El nodo frontend tiene 1 core físico y 1 core lógico
- Cada nodo de cómputo tiene dos microprocesadores Xeon E5645
- Cada microprocesador tiene 6 cores físicos y 12 cores lógicos

Por tanto, un nodo de cómputo de atcgrid (de los tres que hay) tiene en total 12 cores físicos y 24 cores lógicos, y entre los tres nodos, 36 cores físicos y 72 cores lógicos

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 `ngct`, ¿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 variable `ngct` contiene en segundos el tiempo que ha tardado en realizarse la suma de ambos vectores. La función `clock_gettime()` devuelve en segundos el tiempo que ha pasado desde 1970. La información es devuelta en dos struct, para almacenar el tiempo antes y después de la suma del vector respectivamente.

- 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++
Reserva de memoria para los	<code>v1 = (double *)malloc(N *</code>	<code>v1 = new double[N];</code>

vectores.	sizeof(double));	
Liberación de la memoria.	free(v1);	delete[] v1;
Salida por pantalla.	printf("Faltan no componentes del vector\n");	cout_____<<"Faltan no componentes del vector\n" << endl;

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:

Primero descomentamos la definición de VECTOR_LOCAL y comentamos las definiciones de VECTOR_GLOBAL y VECTOR_DYNAMIC. A continuación compilamos y ejecutamos como se muestra la foto.

```
mar 6 mar - 19:29 ~/Escritorio/ETSIIT/2º Curso/2º Cuatrimestre/AC/Practicas/Practica_0
@david gcc -O2 SumaVectoresC.c -o SumaVectoresC -lrt

mar 6 mar - 19:29 ~/Escritorio/ETSIIT/2º Curso/2º Cuatrimestre/AC/Practicas/Practica_0
@david ./SumaVectoresC 58795
Tiempo(seg.):0.000567944 Tamaño Vectores:58795 V1[0]+V2[0]=V3[0](5879.500000+5879.500000=11759.000000) V1[58794]+V2[58794]=V3[58794](11758.900000+0.100000=11759.000000)
```

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:

Primero compilo en local el fichero SumaVectoresC.c, y lo subo con la consola sftp al atcgrid junto con el script. Una vez en el atcgrid lo ejecuto, espero a que termine y descargo el fichero de salida y de error a mi pc.

The screenshot shows three terminal windows. The top window shows the local compilation of SumaVectoresC.c with gcc -O2. The middle window shows the upload of the source file and the execution of the program on atcgrid using sftp. The bottom window shows the local execution of the program for comparison.

```
1: A3estudiante24@atcgrid:~
[A3estudiante24@atcgrid ~]$ ls
SumaVectoresC SumaVectores.sh
[A3estudiante24@atcgrid ~]$ qsub SumaVectores.sh -q ac
65897.atcgrid
[A3estudiante24@atcgrid ~]$ qstat
Job ID Name User Time Use S Queue
-----
65897.atcgrid ...resC_vlocales A3estudiante24 00:00:00 C ac

[A3estudiante24@atcgrid ~]$ ls
SumaVectoresC SumaVectoresC_vlocales.e65897
SumaVectoresC_vlocales.e65897 SumaVectores.sh
[A3estudiante24@atcgrid ~]$

3: sftp A3estudiante24@atcgrid.ugr.es
A3estudiante24@atcgrid.ugr.es's password:
Connected to atcgrid.ugr.es.
sftp> put SumaVectores.sh
Uploading SumaVectores.sh to /home/A3estudiante24/SumaVectores.sh
SumaVectores.sh 100% 827 124.9KB/s 00:00
sftp> put SumaVectoresC
Uploading SumaVectoresC to /home/A3estudiante24/SumaVectoresC
SumaVectoresC 100% 8536 766.9KB/s 00:00
sftp> get SumaVectoresC_vlocales.*
SumaVectoresC_vlocales.e65897 SumaVectoresC_vlocales.e65897
sftp> get SumaVectoresC_vlocales.*
Fetching /home/A3estudiante24/SumaVectoresC_vlocales.e65897 to SumaVectoresC_vlocales.e65897
/home/A3estudiante24/SumaVectoresC_vlocales.e65897 100% 1064 130.0KB/s 00:00
Fetching /home/A3estudiante24/SumaVectoresC_vlocales.e65897 to SumaVectoresC_vlocales.e65897
/home/A3estudiante24/SumaVectoresC_vlocales.e65897 100% 1045 161.7KB/s 00:00
sftp>

2: david@David-PC: ~/Escritorio/ETSIIT/2º Curso/2º Cuatrimestre/AC/Practicas/Practica_0
jue 8 mar - 19:08 ~/Escritorio/ETSIIT/2º Curso/2º Cuatrimestre/AC/Practicas/Practica_0
@david gcc -O2 SumaVectoresC.c -o SumaVectoresC -lrt

jue 8 mar - 19:08 ~/Escritorio/ETSIIT/2º Curso/2º Cuatrimestre/AC/Practicas/Practica_0
@david ls
BP0_Apellido1Apellido2Nombre_Y_2017_10.odt capturas SumaVectoresC SumaVectoresC.c SumaVectoresCpp.cpp SumaVectores_PC.sh SumaVectores.sh

jue 8 mar - 19:08 ~/Escritorio/ETSIIT/2º Curso/2º Cuatrimestre/AC/Practicas/Practica_0
@david ls
BP0_Apellido1Apellido2Nombre_Y_2017_10.odt SumaVectoresC SumaVectoresCpp.cpp SumaVectoresC_vlocales.e65897 SumaVectores.sh
capturas SumaVectoresC.c SumaVectoresC_vlocales.e65897 SumaVectores_PC.sh

jue 8 mar - 19:10 ~/Escritorio/ETSIIT/2º Curso/2º Cuatrimestre/AC/Practicas/Practica_0
@david
```

La salida que he obtenido es la siguiente:

[illegible]

Los errores que he obtenido son los siguientes:

```
jue 8 mar - 19:11 ~/Escritorio/ETSIIT/2º Curso/2º Cuatrimestre/AC/Practicas/Practica_0
@ david cat SumaVectores.c vlocales.atc5897
/var/lib/torque/mom_priv/jobs/65897.atcgrid.SC: line 25: 23345 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores.c $N
/var/lib/torque/mom_priv/jobs/65897.atcgrid.SC: line 25: 23348 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores.c $N
/var/lib/torque/mom_priv/jobs/65897.atcgrid.SC: line 25: 23351 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores.c $N
/var/lib/torque/mom_priv/jobs/65897.atcgrid.SC: line 25: 23355 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores.c $N
/var/lib/torque/mom_priv/jobs/65897.atcgrid.SC: line 25: 23361 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores.c $N
/var/lib/torque/mom_priv/jobs/65897.atcgrid.SC: line 25: 23365 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores.c $N
/var/lib/torque/mom_priv/jobs/65897.atcgrid.SC: line 25: 23368 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores.c $N
/var/lib/torque/mom_priv/jobs/65897.atcgrid.SC: line 25: 23371 Segmentation fault (core dumped) $PBS_O_WORKDIR/SumaVectores.c $N
```

Ahora paso a ejecutarlo en mi PC utilizando el mismo script y el resultado es este:

```
jue 8 mar - 19:17 ~./Escritorio/ETSIII/2º Curso/2º Cuatrimestre/AC/Practicas/Practica_0
$ g++ -std=c++11 ./SumaVectores_PC.sh
# of threads initial: 12
Tiempo(seg.):0.000749183 Tamaño Vectores:65536 V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) V1[65535]+V2[65535]=V3[65535](13107.100000+1.000000=13107.200000)
Tiempo(seg.):0.800836108 Tamaño Vectores:131072 V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) V1[131071]+V2[131071]=V3[131071](26214.300000+1.000000=26214.400000)
Tiempo(seg.):0.801806962 Tamaño Vectores:262144 V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) V1[262143]+V2[262143]=V3[262143](52428.700000+1.000000=52428.800000)
./SumaVectores_PC.sh: línea 9: 19483 Violación de segmento (core generado) ./SumaVectores_C.sh
./SumaVectores_PC.sh: línea 9: 19485 Violación de segmento (core generado) ./SumaVectores_C.sh
./SumaVectores_PC.sh: línea 9: 19487 Violación de segmento (core generado) ./SumaVectores_C.sh
./SumaVectores_PC.sh: línea 9: 19489 Violación de segmento (core generado) ./SumaVectores_C.sh
./SumaVectores_PC.sh: línea 9: 19491 Violación de segmento (core generado) ./SumaVectores_C.sh
./SumaVectores_PC.sh: línea 9: 19493 Violación de segmento (core generado) ./SumaVectores_C.sh
./SumaVectores_PC.sh: línea 9: 19495 Violación de segmento (core generado) ./SumaVectores_C.sh
./SumaVectores_PC.sh: línea 9: 19497 Violación de segmento (core generado) ./SumaVectores_C.sh
```

Se produce un error a partir de un tamaño de 524288 componentes debido a que se produce un desbordamiento de pila.

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:

Vectores Globales:

Comento la definición de los vectores locales y descomento la de los vectores globales. A continuación al igual que he realizado en el ejercicio anterior compilo el programa, lo subo al atcgrid, lo ejecuto y una vez ha terminado descargo el fichero de salida y el de errores.

No se han producido errores y a salida que he obtenida es esta:

[illegible]

Ahora ejecuto en local el mismo programa. La salida obtenida es la siguiente:

```

jue 8 mar - 23:08 ~/Escritorio/ETSIIIT/2º Curso/2º Cuatrimestre/AC/Practicas/Practica_0
@david ./SumaVectores_PC.sh
# of threads initial: 12
Tiempo(seg.):0.008717207 Tamaño Vectores:65536 V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) V1[65535]+V2[65535]=V3[65535](13107.180000+6553.600000=19660.780000)
Tiempo(seg.):0.00748241 Tamaño Vectores:131072 V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) V1[131071]+V2[131071]=V3[131071](26214.390000+13107.200000=39321.590000)
Tiempo(seg.):0.001076922 Tamaño Vectores:262144 V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) V1[262143]+V2[262143]=V3[262143](52428.790000+26214.400000=78643.190000)
Tiempo(seg.):0.002208920 Tamaño Vectores:524288 V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) V1[524287]+V2[524287]=V3[524287](104857.500000+104857.600000=209715.100000)
Tiempo(seg.):0.004284487 Tamaño Vectores:1048576 V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) V1[1048575]+V2[1048575]=V3[1048575](209715.100000+209715.200000=419430.300000)
Tiempo(seg.):0.008406686 Tamaño Vectores:2097152 V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) V1[2097151]+V2[2097151]=V3[2097151](419430.300000+419430.400000=838860.700000)
Tiempo(seg.):0.017856579 Tamaño Vectores:4194304 V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) V1[4194303]+V2[4194303]=V3[4194303](838860.700000+838860.800000=1677721.500000)
Tiempo(seg.):0.033242976 Tamaño Vectores:8388608 V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+1677721.600000=3355443.100000)
Tiempo(seg.):0.069143695 Tamaño Vectores:16777216 V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+3355443.200000=6710886.300000)
Tiempo(seg.):0.132766516 Tamaño Vectores:33554432 V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+6710886.400000=13421772.700000)
Tiempo(seg.):0.132689426 Tamaño Vectores:67108864 V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) V1[67108863]+V2[67108863]=V3[67108863](13421772.700000+13421772.700000=26843545.400000)
Tiempo(seg.):0.132689426 Tamaño Vectores:134217728 V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) V1[134217727]+V2[134217727]=V3[134217727](26843545.400000+26843545.400000=53687090.800000)

```

Comento la definición de los vectores globales y descomento la de los vectores dinámicos. A continuación al igual que he realizado en el ejercicio anterior compilo el programa, lo subo al atcgrid, lo ejecuto y una vez ha terminado descargo el fichero de salida y el de errores.

No se han producido errores y a salida que he obtenida es esta:

[illegible]

Ahora ejecuto en local el mismo programa. La salida obtenida es la siguiente:

```
jue 8 mar - 23:18 ~/Escritorio/ETSIIT/2º Curso/2º Cuatrimestre/AC/Practicas/Practica_0
@david ./SumaVectores_PC.sh
Nº de threads inicial: 12
Tiempo(seg.):0.000609067      Tamaño Vectores:65536      V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) V1[65535]+V2[65535]=V3[65535](13107.10
0000+0.100000=13107.200000)
Tiempo(seg.):0.000929044      Tamaño Vectores:131072      V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) V1[131071]+V2[131071]=V3[131071](262
14.300000+0.100000=26214.400000)
Tiempo(seg.):0.002792497      Tamaño Vectores:262144      V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) V1[262143]+V2[262143]=V3[262143](524
28.700000+0.100000=52428.800000)
Tiempo(seg.):0.002128724      Tamaño Vectores:524288      V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) V1[524287]+V2[524287]=V3[524287](10
4857.500000+0.100000=104857.600000)
Tiempo(seg.):0.004320043      Tamaño Vectores:1048576      V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) V1[1048575]+V2[1048575]=V
3[1048575](209715.100000+0.100000=209715.200000)
Tiempo(seg.):0.008233679      Tamaño Vectores:2097152      V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) V1[2097151]+V2[2097151]=V
3[2097151](419430.300000+0.100000=419430.400000)
Tiempo(seg.):0.016405695      Tamaño Vectores:4194304      V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) V1[4194303]+V2[4194303]=V
3[4194303](838860.700000+0.100000=838860.800000)
Tiempo(seg.):0.031649927      Tamaño Vectores:8388608      V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) V1[8388607]+V2[8388607]=V
3[8388607](1677721.500000+0.100000=1677721.600000)
Tiempo(seg.):0.062546144      Tamaño Vectores:16777216      V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) V1[16777215]+V2[167772
15]=V3[16777215](3355443.100000+0.100000=3355443.200000)
Tiempo(seg.):0.123842249      Tamaño Vectores:33554432      V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) V1[33554431]+V2[335544
31]=V3[33554431](6710886.300000+0.100000=6710886.400000)
Tiempo(seg.):0.248084773      Tamaño Vectores:67108864      V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) V1[67108863]+V2[67108
863]=V3[67108863](13421772.700000+0.100000=13421772.800000)
```


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:

Tabla 1 . Tiempos en Local

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536		0.000748183	0.000717297	0.000609067
131072		0.000836100	0.000748241	0.000929044
262144		0.001869692	0.001077952	0.002792497
524288		-	0.002208920	0.002128724
1048576		-	0.004284487	0.004320043
2097152		-	0.008406686	0.008233679
4194304		-	0.017056579	0.016405695
8388608		-	0.033342876	0.031649927
16777216		-	0.069143695	0.062546144
33554432		-	0.132766516	0.123842249
67108864		-	0.132689426	0.123842249

Tabla 2 . Tiempos en mi atcgrid

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536		0.000379866	0.000424748	0.000401014
131072		0.000845246	0.000835870	0.000802236
262144		0.001564386	0.001791689	0.001624675
524288		-	0.003634955	0.002667146
1048576		-	0.006235772	0.006301220
2097152		-	0.012160471	0.011992833
4194304		-	0.023490829	0.023513465
8388608		-	0.047400212	0.046552130
16777216		-	0.103048226	0.092144595
33554432		-	0.186142820	0.187118826
67108864		-	0.188211444	0.369737004

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 ($MAX=2^{32}-1$). 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: