

Arquitectura de Computadores (AC)

Cuaderno de prácticas.

Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos): Álvaro Fernández García

Grupo de prácticas: A2

Fecha de entrega:

Fecha evaluación en clase:

Ejercicios basados en los ejemplos del seminario práctico

1. En el primer ejemplo de ejecución en atcgrid usando TORQUE se ejecuta el ejemplo HelloOMP.c usando la siguiente orden: `echo 'hello/HelloOMP' | qsub -q ac`. El resultado de la ejecución de este código en atcgrid se puede ver en el seminario. Conteste a las siguientes preguntas:

- a. ¿Para qué se usa en qsub la opción -q?

RESPUESTA: Para poder asignar la tarea que desea ejecutarse a una determinada cola, en nuestro caso asignamos a la cola ac.

- b. ¿Cómo sabe el usuario que ha terminado la ejecución en atcgrid?

RESPUESTA: podemos comprobarlo desde la terminal ssh ejecutando el comando `qstat`, si en el campo S (status o estado) de la tarea que hemos enviado, (la cual se reconoce mediante un identificador) aparece la letra “C”, significa que la tarea ya está completa.

- c. ¿Cómo puede saber el usuario si ha habido algún error en la ejecución?

RESPUESTA: Tras la ejecución de la tarea, se generan dos archivos. En el archivo con extensión que comienza con .e (error) contiene la salida de error, podemos comprobar en este si se ha producido o no un error.

- d. ¿Cómo ve el usuario el resultado de la ejecución?

RESPUESTA: Tras la ejecución de la tarea, se generan dos archivos. En el archivo con extensión que comienza con .o (output) contiene la salida del programa, podemos comprobar en este el resultado

- e. ¿Por qué en el resultado de la ejecución aparecen 24 saludos “¡¡¡Hello World!!!”?

RESPUESTA: En la práctica ya no aparecen 24, aparecen únicamente 12, puesto que se ha impuesto una restricción para evitar el uso de cores lógicos (24 en total) utilizando así los físicos (12).

2. En el segundo ejemplo de ejecución en atcgrid usando TORQUE el script `script_helloomp.sh` usando la siguiente orden: `qsub script_helloomp.sh`. El script ejecuta varias veces el ejecutable del código HelloOMP.c. El resultado de la ejecución de este código en atcgrid se puede ver en el seminario. Conteste a las siguientes preguntas:

- a. ¿Por qué no acompaña a al orden qsub la opción -q en este caso?

RESPUESTA: Por que esta ya aparece especificada dentro del propio script, concretamente en la línea `#PBS -q ac`

- b. ¿Cuántas veces ejecuta el script el ejecutable HelloOMP en atcgrid? ¿Por qué lo ejecuta ese número de veces?

RESPUESTA: El ejecutable HelloOMP se ejecuta 4 veces, esto se debe a que se encuentra dentro de un bucle for, el cual empieza en 12, mientras sea mayor que 0, y en cada iteración divide el índice entre dos, por tanto 12 se puede dividir 4 veces entre 2.

- c. ¿Cuántos saludos “!!!Hello World!!!” se imprimen en cada ejecución? (indique el número exacto) ¿Por qué se imprime ese número?

RESPUESTA: En la primera ejecución se imprime 12 veces, en la segunda 6, en la tercera 3 y en la cuarta y última 1, esto se debe a que en cada iteración del bucle modificamos el número de hebras que va a ejecutar el programa HelloOMP, concretamente en la sentencia “export OMP_NUM_THREADS=\$P” asignándole el valor del índice, el cual toma los valores de 12, 6, 3 y 1 en cada una de las respectivas iteraciones.

3. Realizar las siguientes modificaciones en el script “!!!Hello World!!!”:

- Eliminar la variable de entorno \$PBS_O_WORKDIR en el punto en el que aparece.
- Añadir lo necesario para que, cuando se ejecute el script, se imprima la variable de entorno \$PBS_O_WORKDIR.

Ejecutar el script con estas modificaciones. ¿Qué resultados de ejecución se obtienen en este caso? Incorporar en el cuaderno de trabajo volcados de pantalla que muestren estos resultados.



```

A2estudiante6@atcgrid:~/hello
...nte6@atcgrid:~/hello
...nte6@atcgrid.ugr.es
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
Nº de threads inicial: 12

Para 12 threads:

Para 6 threads:

Para 3 threads:

Para 1 threads:
/home/A2estudiante6/hello
[A2estudiante6@atcgrid hello]$
    
```

RESPUESTA: En este caso, el script no ejecuta correctamente el programa, esto se debe a que al eliminar esa variable no se encuentra la ruta donde está el ejecutable (la cual es la impresora) y en consecuencia no hay programa que ejecutar, de ahí que no muestre los resultados.

Resto de ejercicios

4. Incorporar en el fichero .zip que se entregará al profesor el fichero /proc/cpuinfo de alguno de los nodos de atcgrid (atcgrid1, atcgrid2, atcgrid3), y del PC del aula de prácticas o de su PC. Indique qué ha hecho para obtener el contenido de /proc/cpuinfo en atcgrid.

RESPUESTA: Para obtener el fichero /proc/cpuinfo, únicamente tenemos que crear un script que contenga la orden cat /proc/cpuinfo, de esta forma cuando el script se mande a atcgrid se ejecutará y mostrará su información. Por otra parte se proporciona también las especificaciones del PC del aula de prácticas. El script tiene el nombre de ejercicio4.sh

Teniendo en cuenta el contenido de cpuinfo conteste a las siguientes preguntas (justifique las respuestas):

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

RESPUESTA: Cores Físicos : 4 ; Cores Lógicos : 4 ;

Los cores físicos se observan en la línea `cpu cores:` y los lógicos en `siblings:` por tanto podemos afirmar que cada core físico tiene un solo core lógico.

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

RESPUESTA: Cores físicos : 6 ; Cores Lógicos : 12 ;

Los cores físicos se observan en la línea `cpu cores:` y los lógicos en `siblings:` por tanto podemos afirmar que cada core físico tiene dos cores lógicos.

5. 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()` sirve para obtener el tiempo actual que marca el reloj especificado como argumento, es este caso `CLOCK_REALTIME` (un reloj de tiempo real común al sistema). El resultado se almacena en una estructura de tipo `timespec` la cual tiene dos campos: `tv_sec` (que contiene los segundos) y `tv_nsec` (que contiene los nanosegundos). Por último en la variable `ncgt`, se encuentra almacenado el tiempo transcurrido en completar la suma de los vectores, como `clock_gettime()` devuelve el tiempo transcurrido desde una fecha determinada, para poder medir cuanto tarda la operación es necesario medir primero el tiempo antes de ejecutar la suma y luego después. Por tanto en la variable se guarda el resultado del tiempo final menos el tiempo inicial, además todo pasado a segundos.

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++
Utilización de bibliotecas y namespaces	<code>stdlib.h</code> <code>stdio.h</code> No hay namespace	<code>cstdlib</code> <code>iostream</code> Namespace <code>std</code>
Flujo de salida	Utilización de <code>printf</code>	Utilización de <code>cout</code>

Reserva dinámica	Mediante malloc	Mediante operador new
Dstrucción de memoria dinámica	Mediante free	Mediante delete[]
Comprobación de errores en la reserva.	Si el puntero a la memoria reservada es NULL	No hay comprobación

6. 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). Ejecutar el código ejecutable resultante en atcgrid usando el la cola TORQUE. Incorporar volcados de pantalla que demuestren la ejecución correcta en atcgrid.

RESPUESTA:

```

A2estudiante6@atcgrid:~/SumaVectoresC
+ Práctica0: gcc x ...rid:~/SumaVectoresC ...nte6@atcgrid.ugr.es
[A2estudiante6@atcgrid SumaVectoresC]$ ls
SumaVectoresC
[A2estudiante6@atcgrid SumaVectoresC]$ echo 'SumaVectoresC/SumaVectoresC 15' | qsub -q ac
41662.atcgrid
[A2estudiante6@atcgrid SumaVectoresC]$ cat STDIN.o41662
Tiempo(seg.):0.000000217 / Tamaño Vectores:15 / V1[0]+V2[0]=V3[0](1.500000+1.500000=3.000000) / / V1[14]+V2[14]=V3[14](2.900000+0.100000=3.000000) /
[A2estudiante6@atcgrid SumaVectoresC]$

```

7. 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 local para los mismos tamaños. ¿Se obtiene error para alguno de los tamaños? En caso afirmativo, ¿a qué se debe este error?

RESPUESTA: Si se obtienen errores, concretamente, para los tamaños superiores a 262144. Esto se debe a que las variables locales están almacenadas en la pila, y para tamaños superiores la pila se desborda. Los resultados que no han dado error para atcgrid son:

```

A2estudiante6@atcgrid:~/SumaVectoresC
+ Práctica0: gcc x ...rid:~/SumaVectoresC ...nte6@atcgrid.ugr.es
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
atcgrid1
Tiempo(seg.):0.000372833 / 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.):0.000749379 / 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.):0.001505867 / 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) /
[A2estudiante6@atcgrid SumaVectoresC]$

```

Por otra parte, los resultados obtenidos en mi PC son:

```

x          Práctica0: ./listado3.sh
+  x  ...: ./listado3.sh  ...maVectoresC  ...tcgrid.ugr.es  ...maVectoresC

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.):0.000709242 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.200000+0.100000=13107.200000) /
Tiempo(seg.):0.001768723 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131071]=V3[131071](26214.400000+0.100000=26214.400000) /
Tiempo(seg.):0.003768340 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262143]=V3[262143](52428.800000+0.100000=52428.800000) /
./listado3.sh: línea 21: 14760 Violación de segmento ('core' generado) ./SumaVectoresC $N

```

8. 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 local. ¿Se obtiene error usando vectores globales o dinámicos? ¿A qué cree que es debido?

RESPUESTA:

Vector Dinámico: Atcgrid:

```

x          A2estudiante6@atcgrid:~/SumaVectoresC
+  x  Práctica0: gcc  x  ...rid:~/SumaVectoresC  ...nte6@atcgrid.ugr.es

Tiempo(seg.):0.000387652 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.200000+0.100000=13107.200000) /
Tiempo(seg.):0.000768238 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131071]=V3[131071](26214.400000+0.100000=26214.400000) /
Tiempo(seg.):0.001563822 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262143]=V3[262143](52428.800000+0.100000=52428.800000) /
Tiempo(seg.):0.003131408 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[524287]=V3[524287](104857.600000+0.100000=104857.600000) /
Tiempo(seg.):0.00525459 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575](209715.200000+0.100000=209715.200000) /
Tiempo(seg.):0.010073394 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151](419430.400000+0.100000=419430.400000) /
Tiempo(seg.):0.019988288 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](838860.800000+0.100000=838860.800000) /
Tiempo(seg.):0.039795798 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V2[8388607]=V3[8388607](1677721.600000+0.100000=1677721.600000) /
Tiempo(seg.):0.079475893 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[16777215](3355443.200000+0.100000=3355443.200000) /
Tiempo(seg.):0.157361890 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33554431](6710886.400000+0.100000=6710886.400000) /
Tiempo(seg.):0.318068867 / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / / V1[67108863]+V2[67108863]=V3[67108863](13421772.800000+0.100000=13421772.800000) /
[A2estudiante6@atcgrid:~/SumaVectoresC]$

```

Vector Global Atcgrid:

```

x          A2estudiante6@atcgrid:~/SumaVectoresC
+  x  Práctica0: gcc  x  ...rid:~/SumaVectoresC  ...nte6@atcgrid.ugr.es

Tiempo(seg.):0.000400364 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.200000+0.100000=13107.200000) /
Tiempo(seg.):0.000782427 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / / V1[131071]+V2[131071]=V3[131071](26214.400000+0.100000=26214.400000) /
Tiempo(seg.):0.001535041 / Tamaño Vectores:262144 / V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / / V1[262143]+V2[262143]=V3[262143](52428.800000+0.100000=52428.800000) /
Tiempo(seg.):0.003155250 / Tamaño Vectores:524288 / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[524287]=V3[524287](104857.600000+0.100000=104857.600000) /
Tiempo(seg.):0.005079773 / Tamaño Vectores:1048576 / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575](209715.200000+0.100000=209715.200000) /
Tiempo(seg.):0.009942763 / Tamaño Vectores:2097152 / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151](419430.400000+0.100000=419430.400000) /
Tiempo(seg.):0.019794680 / Tamaño Vectores:4194304 / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](838860.800000+0.100000=838860.800000) /
Tiempo(seg.):0.039364023 / Tamaño Vectores:8388608 / V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / / V1[8388607]+V2[8388607]=V3[8388607](1677721.600000+0.100000=1677721.600000) /
Tiempo(seg.):0.079031517 / Tamaño Vectores:16777216 / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[16777215](3355443.200000+0.100000=3355443.200000) /
Tiempo(seg.):0.157707541 / Tamaño Vectores:33554432 / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33554431](6710886.400000+0.100000=6710886.400000) /
Tiempo(seg.):0.315836520 / Tamaño Vectores:67108864 / V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / / V1[67108863]+V2[67108863]=V3[67108863](13421772.800000+0.100000=13421772.800000) /
[A2estudiante6@atcgrid:~/SumaVectoresC]$

```


Vector Dinámico Mi PC:

Práctica0: ./listado3.sh			
+	Práctica0: ./listado3.sh	...rid~/SumaVectoresC	...nte6@atcgrid.ugr.es
Tiempo(seg.):0.000826500	/ Tamaño Vectores:65536	/ V1[0]+V2[0]=V3[0](6553.6000	
00+6553.60000=13107.200000)	/ V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.20		
0000) /			
Tiempo(seg.):0.001577555	/ Tamaño Vectores:131072	/ V1[0]+V2[0]=V3[0](13107.200	
000+13107.200000=26214.400000)	/ V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=262		
14.400000) /			
Tiempo(seg.):0.003536499	/ Tamaño Vectores:262144	/ V1[0]+V2[0]=V3[0](26214.400	
000+26214.400000=52428.800000)	/ V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=524		
28.800000) /			
Tiempo(seg.):0.006629241	/ Tamaño Vectores:524288	/ V1[0]+V2[0]=V3[0](52428.800	
000+52428.800000=104857.600000)	/ V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=1		
04857.600000) /			
Tiempo(seg.):0.009814630	/ Tamaño Vectores:1048576	/ V1[0]+V2[0]=V3[0](104857.60	
000+104857.600000=209715.200000)	/ V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.100		
000=209715.200000) /			
Tiempo(seg.):0.019383903	/ Tamaño Vectores:2097152	/ V1[0]+V2[0]=V3[0](209715.20	
000+209715.200000=419430.400000)	/ V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.100		
000=419430.400000) /			
Tiempo(seg.):0.035460852	/ Tamaño Vectores:4194304	/ V1[0]+V2[0]=V3[0](419430.40	
000+419430.400000=838860.800000)	/ V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.100		
000=838860.800000) /			
Tiempo(seg.):0.083908020	/ Tamaño Vectores:8388608	/ V1[0]+V2[0]=V3[0](838860.80	
000+838860.800000=1677721.600000)	/ V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.1		
00000=1677721.600000) /			
Tiempo(seg.):0.178422502	/ Tamaño Vectores:16777216	/ V1[0]+V2[0]=V3[0](1677721.6	
00000+1677721.600000=3355443.200000)	/ V1[16777215]+V2[16777215]=V3[16777215](3355443.10000		
0+0.100000=3355443.200000) /			
Tiempo(seg.):0.263902457	/ Tamaño Vectores:33554432	/ V1[0]+V2[0]=V3[0](3355443.2	
00000+3355443.200000=6710886.400000)	/ V1[33554431]+V2[33554431]=V3[33554431](6710886.30000		
0+0.100000=6710886.400000) /			
Tiempo(seg.):0.528215010	/ Tamaño Vectores:67108864	/ V1[0]+V2[0]=V3[0](6710886.4	
00000+6710886.400000=13421772.800000)	/ V1[67108863]+V2[67108863]=V3[67108863](13421772.700		
000+0.100000=13421772.800000) /			

Vector Global mi PC:

Práctica0: ./listado3.sh			
+	Práctica0: ./listado3.sh	...rid~/SumaVectoresC	...nte6@atcgrid.ugr.es
Tiempo(seg.):0.002197887	/ Tamaño Vectores:65536	/ V1[0]+V2[0]=V3[0](6553.6000	
00+6553.600000=13107.200000)	/ V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.20		
0000) /			
Tiempo(seg.):0.002431660	/ Tamaño Vectores:131072	/ V1[0]+V2[0]=V3[0](13107.200	
000+13107.200000=26214.400000)	/ V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=262		
14.400000) /			
Tiempo(seg.):0.005192766	/ Tamaño Vectores:262144	/ V1[0]+V2[0]=V3[0](26214.400	
000+26214.400000=52428.800000)	/ V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=524		
28.800000) /			
Tiempo(seg.):0.010529890	/ Tamaño Vectores:524288	/ V1[0]+V2[0]=V3[0](52428.800	
000+52428.800000=104857.600000)	/ V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=1		
04857.600000) /			
Tiempo(seg.):0.019954765	/ Tamaño Vectores:1048576	/ V1[0]+V2[0]=V3[0](104857.60	
000+104857.600000=209715.200000)	/ V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.100		
000=209715.200000) /			
Tiempo(seg.):0.034446440	/ Tamaño Vectores:2097152	/ V1[0]+V2[0]=V3[0](209715.20	
000+209715.200000=419430.400000)	/ V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.100		
000=419430.400000) /			
Tiempo(seg.):0.067455726	/ Tamaño Vectores:4194304	/ V1[0]+V2[0]=V3[0](419430.40	
000+419430.400000=838860.800000)	/ V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.100		
000=838860.800000) /			
Tiempo(seg.):0.126017071	/ Tamaño Vectores:8388608	/ V1[0]+V2[0]=V3[0](838860.80	
000+838860.800000=1677721.600000)	/ V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.1		
00000=1677721.600000) /			
Tiempo(seg.):0.259029248	/ Tamaño Vectores:16777216	/ V1[0]+V2[0]=V3[0](1677721.6	
00000+1677721.600000=3355443.200000)	/ V1[16777215]+V2[16777215]=V3[16777215](3355443.10000		
0+0.100000=3355443.200000) /			
Tiempo(seg.):0.574549386	/ Tamaño Vectores:33554432	/ V1[0]+V2[0]=V3[0](3355443.2	
00000+3355443.200000=6710886.400000)	/ V1[33554431]+V2[33554431]=V3[33554431](6710886.30000		
0+0.100000=6710886.400000) /			
Tiempo(seg.):0.516855903	/ Tamaño Vectores:33554432	/ V1[0]+V2[0]=V3[0](3355443.2	
00000+3355443.200000=6710886.400000)	/ V1[33554431]+V2[33554431]=V3[33554431](6710886.30000		
0+0.100000=6710886.400000) /			

Con respecto a los errores, la ejecución con vectores dinámicos no presenta ningún problema, sin embargo cuando se realiza con vectores globales podemos observar que no se realiza la última iteración con valor 671088863, si no que las dos últimas tienen el mismo valor: 33554431, esto se debe a que en el código hemos definido que cuando se esté ejecutando con vectores globales el tamaño máximo para el vector sea 2^{25} (`#define MAX=33554431;`).

9. Rellenar una tabla como la Tabla 1 para atcgrid y otra para el PC local 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 para vectores locales, globales y dinámicos (eje y) en función del tamaño en bytes de un vector (eje x). Utilice escala logarítmica en el eje de ordenadas (eje y) en todas las gráficas. ¿Hay diferencias en los tiempos de ejecución con vectores locales, globales y dinámicos?

RESPUESTA: Las gráficas y la tabla de mi PC se encuentran en la hoja de cálculo Tablas y gráficas.ods. Además no se presenta prácticamente ninguna diferencia entre los tres tiempos de ejecución, coincidiendo prácticamente, unas líneas sobre otras, (aunque en mi pc si que hay una ligera desviación de los vectores globales aunque posiblemente se deba a la carga del sistema en ese instante).

Tabla 1. Tiempos de ejecución de la suma de vectores para vectores locales, globales y dinámico

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0,000380620	0,000364329	0,000393320
131072	1048576	0,000763693	0,000776877	0,000764788
262144	2097152	0,001517513	0,001519844	0,001531676
524288	4194304	error	0,003144566	0,002725632
1048576	8388608	error	0,005297078	0,005187736
2097152	16777216	error	0,010033637	0,010036647
4194304	33554431	error	0,019896295	0,019888861
8388608	67108864	error	0,039554854	0,039594262
16777216	134217728	error	0,079190334	0,079166288
33554432	268435456	error	0,158160931	0,158830611

10. 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: Al haber realizado esta modificación el programa directamente no puede compilarse, apareciendo el siguiente error en el linker `SumaVectoresC.c: (.text.startup+0x79): reubicación truncada para ajustar: R_X86_64_32S` contra el símbolo ``v2'` definido en la sección `COMMON` en `/tmp/cc1ndas4.o`, para el vector `v2` y `v3`. Este error en el linker se debe fundamentalmente a que se está intentando enlazar el programa de tal forma que el esquema de direccionamiento relativo está muy alejado de lo que se puede soportar con el desplazamiento de 32 bits del modo de direccionamiento relativo elegido. Una de las posibles causas de este error es que el programa es demasiado grande, lo cual concuerda con la desmesurada reserva de datos que se está realizando. En definitiva la forma más segura de almacenar datos grandes es dinámicamente, evitándonos problemas de espacio. Por otra parte se utiliza el tamaño $2^{32}-1$ porque la cantidad de elementos a reservar de un vector se realiza con un número entero, que ocupa 4 bytes (32 bits), por tanto el mayor entero positivo que puede representarse con 32 bits es $2^{32}-1$.