

Arquitectura de Computadores (AC)

Cuaderno de prácticas.

Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos): José Antonio Álvarez Ocete

Grupo de prácticas: 2

Fecha de entrega: 8 - Marzo

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 especificar el destino. Ya sea la cola, el servidor o la cola del servidor.

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

RESPUESTA: O bien miramos que los ficheros de salida hayan sido generados o bien revisamos que el trabajo no está en la cola mediante "qstat".

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

RESPUESTA: Mirando el archivo de salida de error. Si este archivo, STDIN.eX (donde X es el número del trabajo) está vacío, entonces no hemos obtenido ningún error en la ejecución.

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

RESPUESTA: En el otro archivo de salida, STDIN.oX.

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

RESPUESTA: Porque hay 24 hebras (2 por core, 12 cores) y cada una imprime uno.

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: Si no le damos la opción -q, se enviará el trabajo al servidor por defecto (podemos encontrar esta información ejecutando "man qsub" en la terminal de ssh). Mediante `#PBS -q ac`, especificado en el script, hemos modificado el servidor por defecto al que mandamos los trabajos, asegurándonos así que el trabajo se mandará a esa cola si lo ejecutamos sin la opción -q.

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

RESPUESTA: Se ejecuta un total de 4 veces. En primer lugar fijamos el número de hebras a 12 (mediante "export OMP_THREAD_LIMIT=12") y según las operaciones del bucle ejecutamos el script para 12, 6, 3 y 1 hebras en cada ejecución.

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

RESPUESTA: Por lo explicado en la pregunta anterior, se mostrarán tantos saludos "!!!Hello World!!!" como hebras ejecutan el script pues cada hebra imprimirá uno. Es decir, 12, 6, 3 y 1 saludos para cada ejecución respectivamente.

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.

RESPUESTA: He introducido la línea de código que esta remarcado en la imagen en el guión y he sustituido la variable \$PBS_O_WORKDIR por “.”, colocando ambos archivos en el mismo directorio. Adjunto captura de la salida.

Si colocásemos el archivo en la carpeta “hello” tendríamos que sustituir la variable por “./hello” para que el proceso pueda encontrar el archivo HelloOMP.

```
[E2estudiante2@atcgrid ~]$ cat helloomp.o42360  
Id. usuario del trabajo: E2estudiante2  
Id. del trabajo: 42360.atcgrid  
Nombre del trabajo especificado por usuario: helloomp  
Nodo que ejecuta qsub: atcgrid  
Cola: ac  
Nodos asignados al trabajo:  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
atcgrid2  
  
Nº de threads inicial 12  
Directorio de trabajo /home/E2estudiante2  
  
Para 12 threads:  
8:!!!Hello world!!!3:!!!Hello world!!!7:!!!Hello world!!!11:!!!Hello world!!!2:!!!  
Hello world!!!0:!!!Hello world!!!6:!!!Hello world!!!5:!!!Hello world!!!4:!!!Hello  
world!!!9:!!!Hello world!!!10:!!!Hello world!!!1:!!!Hello world!!!  
Para 6 threads:  
3:!!!Hello world!!!0:!!!Hello world!!!5:!!!Hello world!!!4:!!!Hello world!!!1:!!!H  
ello world!!!2:!!!Hello world!!!  
Para 3 threads:  
1:!!!Hello world!!!0:!!!Hello world!!!2:!!!Hello world!!!  
Para 1 threads:  
0:!!!Hello world!!![E2estudiante2@atcgrid ~]$
```

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:

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: Tiene 4 cores lógicos (basta con contarlos), y un sólo core físico, pues todos los lógicos tienen el mismo “physical id” que nos indica a qué core físico corresponde

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

RESPUESTA: De la misma forma que en el ejercicio anterior, tenemos 24 cores lógicos y hay 2 cores físicos distintos, pues hay únicamente 2 valores de “physical id”, 0 y 1, que pertenecen a los distintos cores físicos.

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: ncgt contiene la diferencia entre los dos tiempos tomados por clock_gettime(). Esto es, el tiempo que han tardado en ejecutarse las operaciones. La función clock_gettime() devuelve como tal el valor 0 si ha habido algún error y -1 en caso contrario. En la struct timespec cgt1 pasada por referencia se almacenará el tiempo

marcado por el reloj de la `struct clockid_t` también pasada como parámetro (en este caso `CLOCK_REALTIME`, por lo que tomamos el valor de tiempo real en dicho instante).

- 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++ |
|---|---|--|
| Bibliotecas incluidas | <code>#include <stdlib.h></code> <code>#include <stdio.h></code> | <code>#include <cstdlib.h></code> <code>#include <iostream></code> |
| Constante de compilador para imprimir vectores | <code>#define PRINTF_ALL</code> | <code>#define COUT_ALL</code> |
| Reserva de memoria | <code>v1 = (double*) malloc(N*sizeof(double));</code> | <code>v1 = new double [N];</code> |
| Salida de datos por pantalla | <code>printf("Tiempo(seg.): %11.9f\t /Tamaño Vectores: %u \n",ncgt,N);</code> | <code>cout << "Tiempo(seg.):" << ncgt << "\t/ Tamaño Vectores:" << N << endl;</code> |
| Liberar memoria si hemos usado memoria dinámica | <code>free(v1);</code> <code>free(v2);</code> <code>free(v3);</code> | <code>delete [] v1;</code> <code>delete [] v2;</code> <code>delete [] v3;</code> |

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:

```
[E2estudiante2@atcgrid ~]$ ls
SumaVectores
[E2estudiante2@atcgrid ~]$ echo './SumaVectores 65536' | qsub -q ac
45466.atcgrid
[E2estudiante2@atcgrid ~]$ ls
STDIN.e45466 STDIN.o45466 SumaVectores
[E2estudiante2@atcgrid ~]$ cat STDIN.o45466
Tiempo(seg.):0.000379861 / Tamaño Vectores:65536 /V1[0]+V2[0]=V3[0](6553.600000+6
553.600000=13107.200000) / /V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
[E2estudiante2@atcgrid ~]$
```

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: Ejecución en atcgrid:

```
[E2estudiante2@atcgrid ~]$ qsub SumaVectores.sh
45519.atcgrid
[E2estudiante2@atcgrid ~]$ ls
SumaVectores                               SumaVectoresC_vlocales.o45519
SumaVectoresC_vlocales.e45519              SumaVectores.sh
[E2estudiante2@atcgrid ~]$ cat SumaVectoresC_vlocales.o45519
Id. usuario del trabajo: E2estudiante2
Id. del trabajo: 45519.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vlocales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/E2estudiante2
Cola: ac
Nodos asignados al trabajo:
atcgrid1
atcgrid1
```

```
atcgrid1
atcgrid1
atcgrid1
Tiempo(seg.):0.000375593 / 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.):0.000754539 / Tamaño Vectores:131072 /V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / /V1[1310
71]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.001525216 / Tamaño Vectores:262144 /V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / /V1[2621
43]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tiempo(seg.):0.003099574 / Tamaño Vectores:524288 /V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / /V1[524
287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tiempo(seg.):0.005043523 / Tamaño Vectores:1048576 /V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / /V1[1
048575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.010013666 / Tamaño Vectores:2097152 /V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / /V1[2
097151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.019995985 / Tamaño Vectores:4194304 /V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / /V1[4
194303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.039626546 / 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.):0.078676528 / Tamaño Vectores:16777216 /V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / /V
1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.156276784 / Tamaño Vectores:33554432 /V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / /V
1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.314880313 / Tamaño Vectores:67108864 /V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / /
V1[67108863]+V2[67108863]=V3[67108863](13421772.700000+0.100000=13421772.800000) /
[E2estudiante2@atcgrid ~]$
```

Ejecución en mi ordenador personal (modificando “\$PBS_O_WORKDIR por “.”). No he obtenido ningún problema en la ejecución para ninguno de los tamaños del vector:

```
jose@jaoopc:~/Escritorio/DGIIIM/AC/PO/SumaVectores$ ./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.):0.000430732 / 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.):0.000374477 / Tamaño Vectores:131072 /V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / /V1[1310
71]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.000889520 / Tamaño Vectores:262144 /V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / /V1[2621
43]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
Tiempo(seg.):0.001817507 / Tamaño Vectores:524288 /V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / /V1[524
287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
Tiempo(seg.):0.004528417 / Tamaño Vectores:1048576 /V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / /V1[1
048575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.008600026 / Tamaño Vectores:2097152 /V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / /V1[2
097151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.018012135 / Tamaño Vectores:4194304 /V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / /V1[4
194303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.022262233 / 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.):0.043792125 / Tamaño Vectores:16777216 /V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / /V
1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.087154255 / Tamaño Vectores:33554432 /V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / /V
1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.174850929 / Tamaño Vectores:67108864 /V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / /
V1[67108863]+V2[67108863]=V3[67108863](13421772.700000+0.100000=13421772.800000) /
jose@jaoopc:~/Escritorio/DGIIIM/AC/PO/SumaVectores$
```


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:

Ejecución en `atcgrid` con vectores globales:

```

SumaVectoresC_vdinamic.o46343      SumaVectoresC_vglobales.o46344
1 Id. usuario del trabajo: E2estudiante2
2 Id. del trabajo: 46344.atcgrid
3 Nombre del trabajo especificado por usuario: SumaVectoresC_vglobales
4 Nodo que ejecuta qsub: atcgrid
5 Directorio en el que se ha ejecutado qsub: /home/E2estudiante2
6 Cola: ac
7 Nodos asignados al trabajo:
8 Tiempo(seg.):0.000392065 / Tamaño Vectores:65536 /V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000)
9 /V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
10 Tiempo(seg.):0.000766065 / Tamaño Vectores:131072 /V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) /
11 /V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
12 Tiempo(seg.):0.001440520 / Tamaño Vectores:262144 /V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) /
13 /V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
14 Tiempo(seg.):0.002656903 / Tamaño Vectores:524288 /V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) /
15 /V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
16 Tiempo(seg.):0.005266514 / Tamaño Vectores:1048576 /V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) /
17 /V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000) /
18 Tiempo(seg.):0.010025380 / Tamaño Vectores:2097152 /V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) /
19 /V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000) /
20 Tiempo(seg.):0.019934786 / Tamaño Vectores:4194304 /V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) /
21 /V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000) /
22 Tiempo(seg.):0.039574752 / Tamaño Vectores:8388608 /V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) /
23 /V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
24 Tiempo(seg.):0.078977981 / Tamaño Vectores:16777216 /V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) /
25 /V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
26 Tiempo(seg.):0.157999013 / Tamaño Vectores:33554432 /V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) /
27 /V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
28 Tiempo(seg.):0.316351887 / Tamaño Vectores:67108864 /V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) /
29 /V1[67108863]+V2[67108863]=V3[67108863](13421772.700000+0.100000=13421772.800000) /

```

Ejecución en `atcgrid` con vectores dinámicos:

```

SumaVectoresC_vdinamic.o46343      SumaVectoresC_vglobales.o46344
1 Id. usuario del trabajo: E2estudiante2
2 Id. del trabajo: 46343.atcgrid
3 Nombre del trabajo especificado por usuario: SumaVectoresC_vdinamic
4 Nodo que ejecuta qsub: atcgrid
5 Directorio en el que se ha ejecutado qsub: /home/E2estudiante2
6 Cola: ac
7 Nodos asignados al trabajo:
8 Tiempo(seg.):0.000394304 / Tamaño Vectores:65536 /V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000)
9 /V1[65535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000) /
10 Tiempo(seg.):0.000761063 / Tamaño Vectores:131072 /V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000)
11 /V1[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000) /
12 Tiempo(seg.):0.001518601 / Tamaño Vectores:262144 /V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) /
13 /V1[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000) /
14 Tiempo(seg.):0.003140296 / Tamaño Vectores:524288 /V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) /
15 /V1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000) /
16 Tiempo(seg.):0.005119754 / Tamaño Vectores:1048576 /V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) /
17 /V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000) /
18 Tiempo(seg.):0.010063275 / Tamaño Vectores:2097152 /V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) /
19 /V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000) /
20 Tiempo(seg.):0.019820530 / Tamaño Vectores:4194304 /V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) /
21 /V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000) /
22 Tiempo(seg.):0.039165789 / Tamaño Vectores:8388608 /V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) /
23 /V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000) /
24 Tiempo(seg.):0.078659680 / Tamaño Vectores:16777216 /V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) /
25 /V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000) /
26 Tiempo(seg.):0.158775100 / Tamaño Vectores:33554432 /V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) /
27 /V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000) /
28 Tiempo(seg.):0.317966474 / Tamaño Vectores:67108864 /V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) /
29 /V1[67108863]+V2[67108863]=V3[67108863](13421772.700000+0.100000=13421772.800000) /

```

Ejecución en mi ordenador personal con vectores globales:

```

jose@jaopc:~/Escritorio/DGIIM/AC/P0/SunaVectores$ ./SunaVectoresG.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.):0.000596872 / Tamaño Vectores:65536 /V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / /V1[65535]+V2[65535]
j=V3[65535](13107.100000+0.100000=13107.200000) /
Tiempo(seg.):0.000577937 / Tamaño Vectores:131072 /V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / /V1[131071]+V2[131071]
j=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.000851488 / Tamaño Vectores:262144 /V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / /V1[262143]+V2[262143]
j=V3[262143](52428.700000+0.100000=52428.800000) /
Tiempo(seg.):0.002250217 / Tamaño Vectores:524288 /V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / /V1[524287]+V2[524287]
j=V3[524287](104857.500000+0.100000=104857.600000) /
Tiempo(seg.):0.003569262 / Tamaño Vectores:1048576 /V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / /V1[1048575]+V2[1048575]
j=V3[1048575](209715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.005541568 / Tamaño Vectores:2097152 /V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / /V1[2097151]+V2[2097151]
j=V3[2097151](419430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.013948322 / Tamaño Vectores:4194304 /V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / /V1[4194303]+V2[4194303]
j=V3[4194303](838860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.021895337 / Tamaño Vectores:8388608 /V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / /V1[8388607]+V2[8388607]
j=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.042722292 / Tamaño Vectores:16777216 /V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / /V1[16777215]+V2[16777215]
j=V3[16777215](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.085382010 / Tamaño Vectores:33554432 /V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / /V1[33554431]+V2[33554431]
j=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.172660759 / Tamaño Vectores:67108864 /V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / /V1[67108863]+V2[67108863]
j=V3[67108863](13421772.700000+0.100000=13421772.800000) /
jose@jaopc:~/Escritorio/DGIIM/AC/P0/SunaVectores$

```

Ejecución con vectores dinámicos en mi ordenador personal:

```

jose@jaopc:~/Escritorio/DGIIM/AC/P0/SunaVectores$ ./SunaVectoresD.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.):0.000597182 / Tamaño Vectores:65536 /V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / /V1[65535]+V2[65535]
j=V3[65535](13107.100000+0.100000=13107.200000) /
Tiempo(seg.):0.000674280 / Tamaño Vectores:131072 /V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) / /V1[131071]+V2[131071]
j=V3[131071](26214.300000+0.100000=26214.400000) /
Tiempo(seg.):0.001027472 / Tamaño Vectores:262144 /V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) / /V1[262143]+V2[262143]
j=V3[262143](52428.700000+0.100000=52428.800000) /
Tiempo(seg.):0.001529823 / Tamaño Vectores:524288 /V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / /V1[524287]+V2[524287]
j=V3[524287](104857.500000+0.100000=104857.600000) /
Tiempo(seg.):0.004587546 / Tamaño Vectores:1048576 /V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / /V1[1048575]+V2[1048575]
j=V3[1048575](209715.100000+0.100000=209715.200000) /
Tiempo(seg.):0.005628931 / Tamaño Vectores:2097152 /V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / /V1[2097151]+V2[2097151]
j=V3[2097151](419430.300000+0.100000=419430.400000) /
Tiempo(seg.):0.014151806 / Tamaño Vectores:4194304 /V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / /V1[4194303]+V2[4194303]
j=V3[4194303](838860.700000+0.100000=838860.800000) /
Tiempo(seg.):0.021746165 / Tamaño Vectores:8388608 /V1[0]+V2[0]=V3[0](838860.800000+838860.800000=1677721.600000) / /V1[8388607]+V2[8388607]
j=V3[8388607](1677721.500000+0.100000=1677721.600000) /
Tiempo(seg.):0.043181247 / Tamaño Vectores:16777216 /V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / /V1[16777215]+V2[16777215]
j=V3[16777215](3355443.100000+0.100000=3355443.200000) /
Tiempo(seg.):0.087782961 / Tamaño Vectores:33554432 /V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / /V1[33554431]+V2[33554431]
j=V3[33554431](6710886.300000+0.100000=6710886.400000) /
Tiempo(seg.):0.172213577 / Tamaño Vectores:67108864 /V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / /V1[67108863]+V2[67108863]
j=V3[67108863](13421772.700000+0.100000=13421772.800000) /
jose@jaopc:~/Escritorio/DGIIM/AC/P0/SunaVectores$

```

Como se puede apreciar no he obtenido ningún problema en la ejecución de ninguno de los códigos.

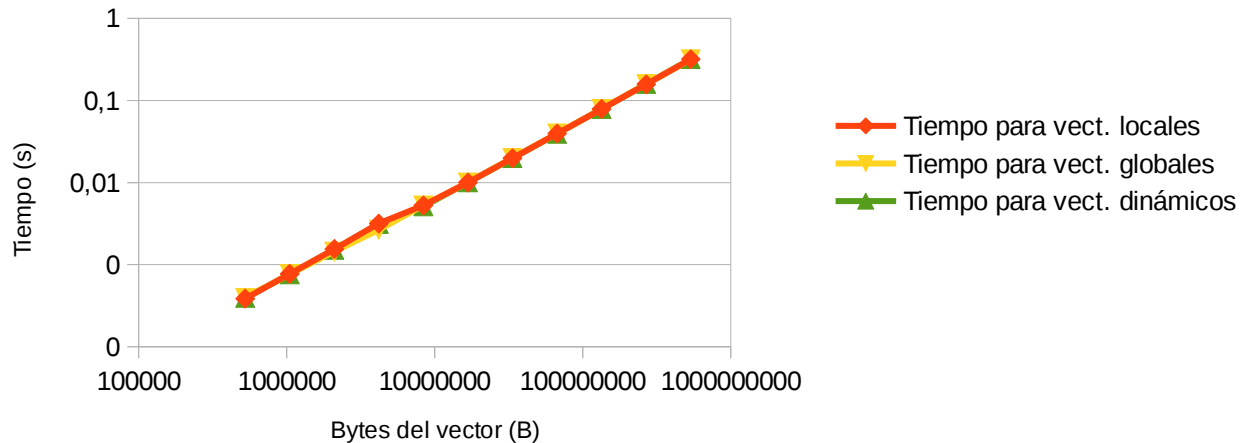
9. Rellenar una tabla como la Tabla 1 para atcgrid y otra para el PC local con los tiempos de ejecución obtenidos en los ejercicios anteriores para el trozo de código que realiza la suma de vectores. En la columna “Bytes de un vector” hay que poner el total de bytes reservado para un vector. Ayudándose 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 tablas con los datos están en el fichero de datos adjunto llamado “Gráficas”. A continuación adjunto imágenes de ambas gráficas. Aunque en el guión no se especifica como tal si el eje X ha de ir en escala logarítmica he considerado que se veían los datos de forma mucho más clara de esta forma. Aún así, en el archivo adjunto están ambas gráficas (con y sin escala logarítmica).

Como conclusión podemos decir que no se aprecian grandes diferencias entre las distintas ejecuciones. El único detalle a considerar sería la ligera mejora que representa el uso de vectores locales en la ejecución en mi ordenador personal (aunque mis compañeros han tenido ciertas dificultades en este apartado al obtener los datos para valores de N muy grandes, no ha sido así con mi ordenador).

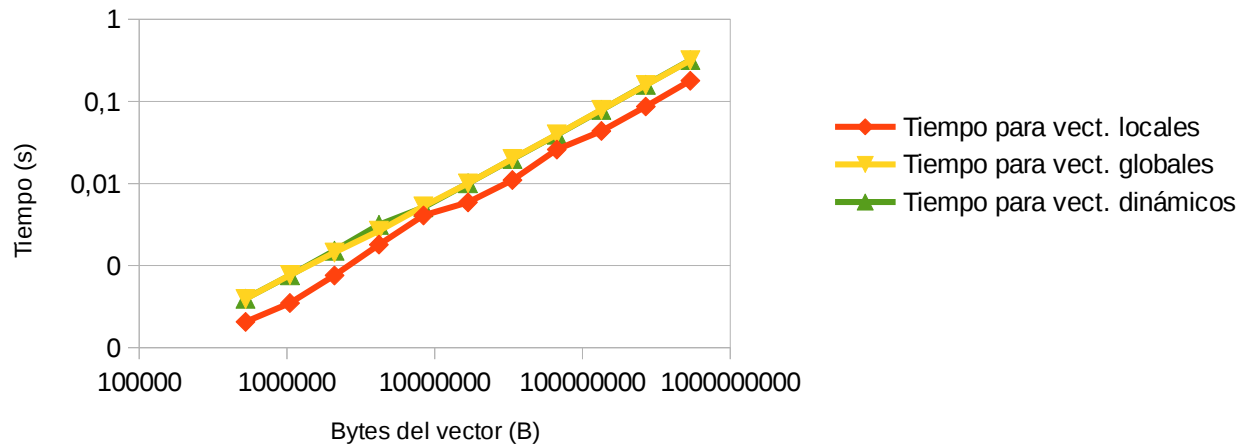
Atcgrid - con escala logarítmica en el eje X

Tiempo de ejecución en función de los bytes del vector



PC personal - con escala logarítmica en el eje X

Tiempo de ejecución en función de los bytes del vector



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 ($MAX=2^{32}-1$). 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 realizar los pasos citados en el guión se obtiene el siguiente error en distintos puntos del ejecutable y no llega a compilar:

“SumaVectores Modificado.c:(.text.startup+0x79): reubicación truncada para ajustar: R_X86_64_32S contra el símbolo `v2' definido en la sección COMMON en /tmp/ccnUymNJ.o”

Esto es porque estamos intentando generar un archivo ejecutable demasiado grande. Sabemos que las variables globales se almacenan en dicho archivo ejecutable. Al generar tres vectores de $2^{32} * 8B$, esto son tres vectores de $2^{35} B$, que son 32GB por vector, el ejecutable ocuparía aproximadamente 96GB (despreciando el resto del código frente al tamaño de estos vectores).

En cuanto al tamaño máximo del vector, para almacenar un dato de tipo “int”, C utiliza $8B = 32bits$. En general el máximo rango de un int es de $2^{32} / 2 - 1$, porque tenemos que considerar los números negativos y el cero. Sin embargo para iterar sobre un vector nunca vamos a necesitar números negativos. Teniendo en cuenta el 0 (de ahí el ‘-1’) tendremos en total 2^{32} posibles posiciones, de 0 a $2^{32} - 1 = MAX$.