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): Rodrigo Raya Castellano
Grupo de prácticas: 2
Fecha de entrega:
Fecha evaluación en clase:

1. En el primer ejemplo de ejecución en atcgrid usando TORQUE se ejecuta el ejemplo `HelloOMP.c` de la página 10 del seminario usando la siguiente orden: `echo 'hello/HelloOMP' | qsub -q ac`. El resultado de la ejecución de este código en atcgrid se puede ver en la página 17 del seminario. Conteste a las siguientes preguntas:

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

      **RESPUESTA**:

      Define el destino de un trabajo. El destino es nombre una cola, servidor o una cola dentro de un servidor.

      Si se especifica la opción -q, el destino tendrá el siguiente formato: cola, @servidor, cola@servidor.

      Si la opción -q no se especifica, el comando qsub enviará el script el servidor por defecto.

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

      **RESPUESTA**:

      El usuario puede determinar que ha terminado una ejecución mediante el comando qstat que muestra todos los trabajos que se están ejecutando y los que están encolados en todas las colas.

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

      **RESPUESTA**:

      Tras ejecutar el comando qsub se deben obtener dos ficheros como respuesta. Un fichero STDIN.o que contiene el resultado propiamente dicho y un fichero STDIN.e que contiene la lista de errores de la ejecución.

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

      **RESPUESTA**:

Siguiendo la respuesta a la pregunta anterior, el usuario verá los resultados de la ejecución visualizando el fichero STDIN.o correspondiente mediante el comando: cat STDIN.o

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

**RESPUESTA**:

Un cluster del atcgrid contiene dos procesadores cada uno con seis núcleos. En cada núcleo puede ejecutarse hasta dos hebras por lo tanto y según las directivas que se indicaron en el programa, el mensaje será reproducido 24 veces.

2. En el segundo ejemplo de ejecución en atcgrid usando TORQUE el script `script_helloomp.sh` de la página 22 del seminario 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 la página 26 del seminario. Conteste a las siguientes preguntas:

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

**RESPUESTA**:

Está ya indicada en el propio script cuando se escribe la directiva de preprocesamiento: #PBS -q ac.

b. ¿Cuántas veces ejecuta el `script` el ejecutable `HelloOMP` en atcgrid?

**RESPUESTA**:

Se ejecuta hasta cuatro veces. Esto se deduce de la salida y de la propia sintaxis del bucle for.

c. ¿Cuántos saludos “¡¡¡Hello World!!!” se imprimen en cada ejecución? ¿Por qué se imprime ese número?

**RESPUESTA**:

Para 12 threads aparecen 12 saludos.

Para 6 threads aparecen 6 saludos.

Para 3 threads aparecen 3 saludos.

Para 1 thread aparece 1 saludo.

La variable OMP_NUM_THREADS permite fijar en todos los casos el número de hebras que se utilizarán durante la ejecución.

Mediante la directiva del fichero HelloOMP.c #pragma omp parallel se asigna a cada hebra el código entre llaves o en su defecto la línea siguiente. Es por esto que se muestra por cada hebra un saludo.

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**:

   En este caso, no se llega a ejecutar el programa. Esto se debe a que torque no encuentra el archivo o directorio.

   Nos damos cuenta de que torque necesita que se le indica la ruta absoluta dentro del sistema.

   En el volcado siguiente se presenta en primer lugar el contenido del fichero helloomp.o3330 donde se puede apreciar que mostramos el directorio de trabajo además de que el programa no produce ninguna salida por las razones previamente indicadas.

   En segundo lugar, se presenta el volcado del fichero helloomp.e3330 donde se pueden apreciar los cuatro errores que tiene el script por no haber indicado la ruta absoluta.

```
[E2estudiante9@atcgrid hello]$ cat helloomp.o3330
Id. usuario del trabajo: E2estudiante9
Id. del trabajo: 3330.atcgrid
Nombre del trabajo especificado por usuario: helloomp
Nodo que ejecuta qsub: atcgrid
Cola: ac
Nodos asignados al trabajo:
atcgrid1
Directorio de trabajo: /home/E2estudiante9/hello
No de threads inicial: 12

Para 12 threads:

Para 6 threads:

Para 3 threads:

Para 1 threads:
[E2estudiante9@atcgrid hello]$ cat helloomp.e3330
/var/lib/torque/mom_priv/jobs/3330.atcgrid.SC: line 24: $./helloOMP: No such file or directory
/var/lib/torque/mom_priv/jobs/3330.atcgrid.SC: line 24: $./helloOMP: No such file or directory
/var/lib/torque/mom_priv/jobs/3330.atcgrid.SC: line 24: $./helloOMP: No such file or directory
/var/lib/torque/mom_priv/jobs/3330.atcgrid.SC: line 24: $./helloOMP: No such file or directory
[E2estudiante9@atcgrid hello]$
```

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), del PC del aula de prácticas y de su PC (si tiene Linux instalado). Indique qué ha hecho para obtener el contenido de `/proc/cpuinfo` en atcgrid.

**RESPUESTA**:

Basta ejecutar el comando echo 'cat proc/info' | qsub -q ac.

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?

**RESPUESTA**:

De la información contenida en el archivo “cpuinfopract” deducimos que el equipo del aula de prácticas dispone de 4 cores lógicos y 4 cores físicos.

b. ¿Cuántos cores físicos y cuántos cores lógicos tiene su PC?

**RESPUESTA**:

Según la información contenida en el archivo “cpuinfolocal” se tiene un socket con dos cores físicos y cuatro cores lógicos.

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

**RESPUESTA**:

En este caso, según la información del archivo “cpuinfoatc” tengo dos sockets cada uno con seis cores físicos y doce lógicos. Eso hace un total de doce cores físicos y veinticuatro cores lógicos (contando ambos sockets).

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.`

   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?

   **RESPUESTA**:

   La variable ncgt contiene el tiempo transcurrido para sumar los vectores. Concretamente el segundo parámetro de la función clock_gettime es una estructura de tipo timespec que contiene al menos dos campos:

   1) tv_sec: número de segundos desde 1970.
   2) tv_nsec: número de nanosegundos transcurridos del segundo actual.

   Restando estos valores se obtiene en segundos el tiempo de ejecución del bucle mediante la línea de código:

```
ncgt=(double) (cgt2.tv_sec-cgt1.tv_sec)+
      (double) ((cgt2.tv_nsec-cgt1.tv_nsec)/(1.e+9));
```

   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.

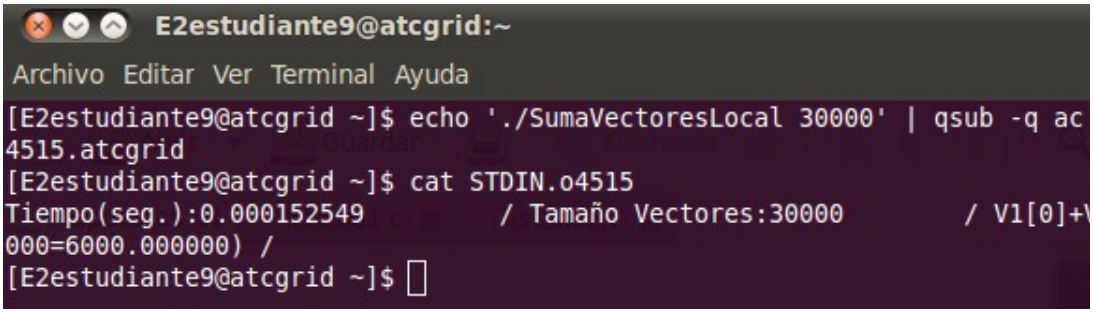
   **RESPUESTA**:

   1) Aunque no es una diferencia a nivel de código recogemos que se utilizan compiladores distintos (gcc y g++). Además las bibliotecas que se utilizan son distintas en C: `stdlib.h y stdio.h` y en C++ `cstdlib.h y iostream.h`.
   2) En el código se declaran las variables antes de usarlas en los bucles. Así: `int i;`

3) Siguiendo con lo dicho en 1) en C se utiliza printf() y en C++ cout.
4) La forma de reservar memoria en C es utilizar la función `malloc` y en C++ usando el operador new. Para liberar memoria se utilizan respectivamente las funciones `free` y `delete`.

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 la cola TORQUE. Incorporar volcados de pantalla que demuestren la ejecución correcta en atcgrid.

**RESPUESTA**:




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**:

A continuación muestro las líneas más relevantes del resultado que aparece en el fichero .o cuando la ejecución se realiza en atcgrid:

```
Nombre del trabajo: SumaVectoresC_vlocales
Nodos asignados al trabajo: atcgrid1
Tiempo(seg.):0.000357436    / Tamaño Vectores:65536
Tiempo(seg.):0.000694688    / Tamaño Vectores:131072
Tiempo(seg.):0.001296994    / Tamaño Vectores:262144
```

A partir del siguiente tamaño aparecen errores del tipo:

```
/var/lib/torque/mom_priv/jobs/4523.atcgrid.SC: line 20: 24070
Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresLocal
$N
```

El problema es que la declaración de los arrays se hace como variable local de modo que se almacenan en la pila. Lo más probable en esta situación es que si el array es demasiado grande se produzca una violación

de segmento, es decir, que se pretende escribir en posiciones de memoria protegida (hemos traspasado el segmento correspondiente del programa, en este caso la pila).

Es interesante saber que el tamaño de la pila lo fija el propio sistema operativo.

A continuación muestro los resultados resumidos de la ejecución en el ordenador local del aula de prácticas:

```
Tiempo(seg.):0.000195757    / Tamaño Vectores:65536
Tiempo(seg.):0.000378877    / Tamaño Vectores:131072
Tiempo(seg.):0.000835334    / Tamaño Vectores:262144
```

Y a continuación aparece repetidas veces el siguiente error:

```
./SumaVectores.sh: línea 21:     3620 Violación de segmento
./SumaVectoresLocal $N
```

En este caso el error se obtiene por motivos análogos.

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**:

   A continuación muestro los resultados resumidos de la ejecución en atcgrid (primero vectores globales y luego dinámicos):

```
Nombre del trabajo: SumaVectoresC_vglobales
Nodos asignados al trabajo: atcgrid1
Tiempo(seg.):0.000361707    / Tamaño Vectores:65536
Tiempo(seg.):0.000721784    / Tamaño Vectores:131072
Tiempo(seg.):0.001652240    / Tamaño Vectores:262144
Tiempo(seg.):0.002337578    / Tamaño Vectores:524288
Tiempo(seg.):0.005037839    / Tamaño Vectores:1048576
Tiempo(seg.):0.010112513    / Tamaño Vectores:2097152
Tiempo(seg.):0.020057947    / Tamaño Vectores:4194304
Tiempo(seg.):0.039884558    / Tamaño Vectores:8388608
Tiempo(seg.):0.079840965    / Tamaño Vectores:16777216
Tiempo(seg.):0.159152572    / Tamaño Vectores:33554432
Tiempo(seg.):0.159446193    / Tamaño Vectores:33554432
```

```
Nombre del trabajo: SumaVectoresC_vdinamicos
Nodos asignados al trabajo: atcgrid1
Tiempo(seg.):0.000347920    / Tamaño Vectores:65536
Tiempo(seg.):0.000597073    / Tamaño Vectores:131072
Tiempo(seg.):0.001314382    / Tamaño Vectores:262144
```

```
Tiempo(seg.):0.002418784    / Tamaño Vectores:524288
Tiempo(seg.):0.005131958    / Tamaño Vectores:1048576
Tiempo(seg.):0.010296583    / Tamaño Vectores:2097152
Tiempo(seg.):0.020340022    / Tamaño Vectores:4194304
Tiempo(seg.):0.040684178    / Tamaño Vectores:8388608
Tiempo(seg.):0.080910122    / Tamaño Vectores:16777216
Tiempo(seg.):0.161054259    / Tamaño Vectores:33554432
Tiempo(seg.):0.324236131    / Tamaño Vectores:67108864
```

A continuación mostramos los resultados resumidos de la ejecución en el ordenados local del aula de prácticas (primero con vectores globales y luego con vectores dinámicos):

```
Tiempo(seg.):0.000254949    / Tamaño Vectores:65536
Tiempo(seg.):0.000512377    / Tamaño Vectores:131072
Tiempo(seg.):0.000769406    / Tamaño Vectores:262144
Tiempo(seg.):0.001614732    / Tamaño Vectores:524288
Tiempo(seg.):0.002815389    / Tamaño Vectores:1048576
Tiempo(seg.):0.005701782    / Tamaño Vectores:2097152
Tiempo(seg.):0.011205567    / Tamaño Vectores:4194304
Tiempo(seg.):0.022454632    / Tamaño Vectores:8388608
Tiempo(seg.):0.042459965    / Tamaño Vectores:16777216
Tiempo(seg.):0.083992262    / Tamaño Vectores:33554432
Tiempo(seg.):0.083046877    / Tamaño Vectores:33554432
```

```
Tiempo(seg.):0.000261798    / Tamaño Vectores:65536
Tiempo(seg.):0.000355870    / Tamaño Vectores:131072
Tiempo(seg.):0.000771060    / Tamaño Vectores:262144
Tiempo(seg.):0.001569045    / Tamaño Vectores:524288
Tiempo(seg.):0.002985906    / Tamaño Vectores:1048576
Tiempo(seg.):0.005465719    / Tamaño Vectores:2097152
Tiempo(seg.):0.011004643    / Tamaño Vectores:4194304
Tiempo(seg.):0.021570834    / Tamaño Vectores:8388608
Tiempo(seg.):0.043341978    / Tamaño Vectores:16777216
Tiempo(seg.):0.085980014    / Tamaño Vectores:33554432
Tiempo(seg.):0.173573115    / Tamaño Vectores:67108864
```

En este ya no se obtiene error. Los segmentos donde se almacenan los vectores son ahora el segmento de datos (en el caso de las variables globales) y el heap (en el caso de las variables dinámicas). Deducimos que estos segmentos tienen un mayor tamaño capaz de alojar grandes cantidades de memoria contigua.

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). Realice otra gráfica con los tiempos obtenidos en el PC local. Utilice escala logarítmica en el eje ordenadas (eje y) en todas las gráficas. ¿Hay diferencias en los tiempos de ejecución con vectores locales, globales y dinámicos?

**RESPUESTA**:

**Tabla 1 .** Tiempos de ejecución de la suma de vectores para vectores locales, globales y dinámicos en atcgrid

| Nº de Componentes | Bytes de un vector | Tiempo para vect. locales | Tiempo para vect. globales | Tiempo para vect. dinámicos |
|---|---|---|---|---|
| 65536 | 524288 | 0.000357436 | 0.000361707 | 0.000347920 |
| 131072 | 1048576 | 0.000694688 | 0.000721784 | 0.000597073 |
| 262144 | 2097152 | 0.001296994 | 0.001652240 | 0.001314382 |
| 524288 | 4194304 | - | 0.002337578 | 0.002418784 |
| 1048576 | 8388608 | - | 0.005037839 | 0.005131958 |
| 2097152 | 16777216 | - | 0.010112513 | 0.010296583 |
| 4194304 | 33554432 | - | 0.020057947 | 0.020340022 |
| 8388608 | 67108864 | - | 0.039884558 | 0.040684178 |
| 16777216 | 134217728 | - | 0.079840965 | 0.080910122 |
| 33554432 | 268435456 | - | 0.159152572 | 0.161054259 |
| 67108864 | 536870912 | - | 0.159446193 | 0.324236131 |

**Tabla 2 .** Tiempos de ejecución de la suma de vectores para vectores locales, globales y dinámicos en el ordenador del aula de prácticas

| Nº de Componentes | Bytes de un vector | Tiempo para vect. locales | Tiempo para vect. globales | Tiempo para vect. dinámicos |
|---|---|---|---|---|
| 65536 | 524288 | 0.000195757 | 0.000254949 | 0.000261798 |
| 131072 | 1048576 | 0.000378877 | 0.000512377 | 0.000355870 |
| 262144 | 2097152 | 0.000835334 | 0.000769406 | 0.000771060 |
| 524288 | 4194304 | - | 0.001614732 | 0.001569045 |
| 1048576 | 8388608 | - | 0.002815389 | 0.002985906 |
| 2097152 | 16777216 | - | 0.005701782 | 0.005465719 |
| 4194304 | 33554432 | - | 0.011205567 | 0.011004643 |
| 8388608 | 67108864 | - | 0.022454632 | 0.021570834 |
| 16777216 | 134217728 | - | 0.042459965 | 0.043341978 |
| 33554432 | 268435456 | - | 0.083992262 | 0.085980014 |
| 67108864 | 536870912 | - | 0.083046877 | 0.173573115 |

**Nota**: realmente la ejecución para el número de componentes se ha realizado para el tamaño anterior. En el ejercicio 10 se ha realizado la modificación correspondiente.

**Tabla 3 .** Gráfica de tiempos de ejecución para los distintos tamaños en ATC-GRID

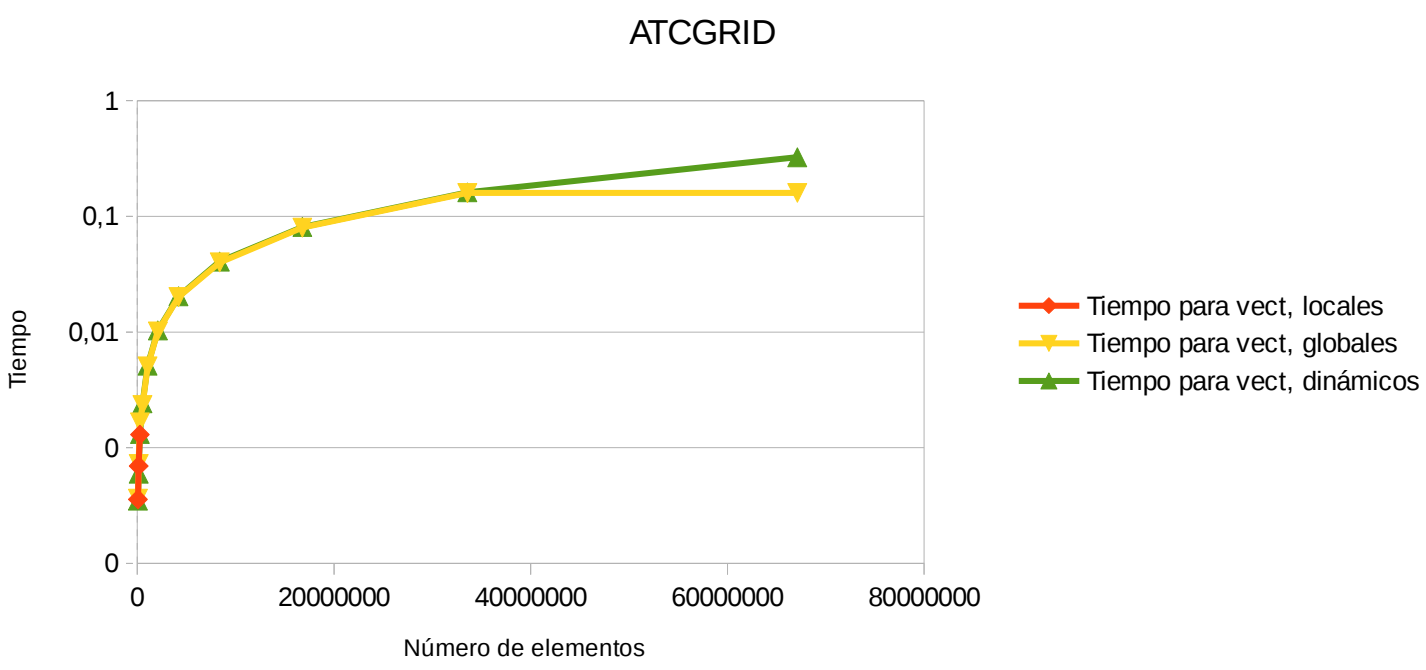
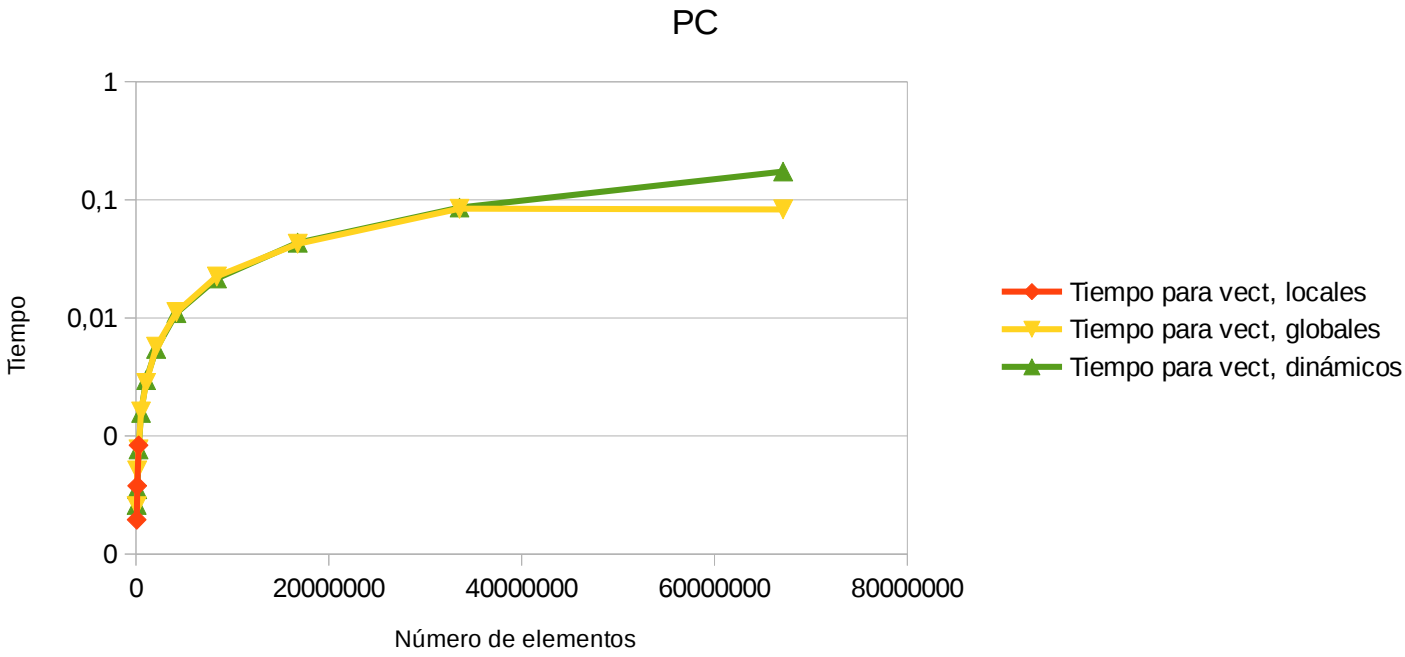


**Tabla 4 .** Gráfica de tiempos de ejecución para los distintos tamaños en el ordenador del aula de prácticas




Las diferencias entre vectores locales y globales o dinámicos son muy difíciles de determinar gráficamente. Realizando la gráfica con escala logarítmica queda ligeramente por debajo el tiempo para vectores locales. Sin embargo, esto lo atribuimos fundamentalmente a las variaciones propias de cada ejecución.

Para valores altos los vectores globales se observa que se comportan mejor que los vectores dinámicos. Esto se podría explicar porque en el caso de los dinámicos hace falta reservar y liberar la memoria utilizada por los vectores.

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**:

Devuelve el siguiente error:

```
/var/lib/torque/mom_priv/jobs/7797.atcgrid.SC: line 20: 12359
Segmentation fault (core dumped)
$PBS_O_WORKDIR/SumaVectoresGlobalgrande $N
```

Aunque a otros compañeros les aparece este otro:

```
/tmp/ccH3ssS8.o: En la función `main':
Listado1.c:(.text.startup+0x65): reubicación truncada para ajustar: R_X86_64_32S contra el símbolo `v2' definido en la sección COMMON en /tmp/ccH3ssS8.o
```

Lo que ocurre es que no puede crear los vectores ya que su tamaño es demasiado grande. En principio para los tres vectores se necesitarían 1.5 Gb. En esta página se encuentra una explicación concreta de lo que ocurre:

https://software.intel.com/en-us/articles/avoiding-relocation-errors-when-building-applications-with-large-global-or-static-data-on-intel64/

“El modelo de memoria por defecto para el compilador de Intel de 64 bits para Linux es pequeño. Esto restringe el tamaño del código y de los datos globales o estáticos a los primeros 2 Gb del espacio de direcciones y permite realizar todos los accesos mediante un direccionamiento IP (instruction pointer?) relativo. Si un programa que contiene más de 2 Gb de memoria global o estática es compilado con estas opciones por defecto, el modelo de direccionamiento seguido puede provocar un “relocation overflo” en tiempo de enlace.”

Por otro lado, el máximo número entero que se representar mediante la variable N es 2^32 - 1 porque N es de tipo unsigned int y utiliza 4 bytes.