2º curso / 2º cuatr.

Grado Ingeniería Informática

Arquitectura de Computadores (AC)

Cuaderno de prácticas.
Bloque Práctico 0. Entorno de programación

Félix Ramírez García
Grupo A3 con profesor Christian Morillas
Fecha de entrega: 10-03-2019
Fecha evaluación en clase: 11-03-2019

Parte I. Ejercicios basados en los ejemplos del seminario práctico

1. Ejecutar `lscpu` en el PC y en un nodo de cómputo de atcgrid.

   (a) Mostrar con capturas de pantalla el resultado de estas ejecuciones.

La salida de la ejecución de la orden lscpu en mi PC es la siguiente :

```
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/AC/Practicas/Entrega/bp0/ejer1] 2019-0
3-04 Monday
$lscpu
Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
CPU(s):              4
On-line CPU(s) list: 0-3
Thread(s) per core:  2
Core(s) per socket:  2
Socket(s):           1
Vendor ID:           GenuineIntel
CPU family:          6
Model:               142
Model name:          Intel(R) Core(TM) i3-7100U CPU @ 2.40GHz
Stepping:            9
CPU MHz:             2400.000
CPU max MHz:         2400.0000
BogoMIPS:            4800.00
Virtualization:      VT-x
Hypervisor vendor:   Windows Subsystem for Linux
Virtualization type: container
Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr s
se sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 fma cx16 xtpr pd
cm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave osxsave avx f16c rdrand
```

Y la salida de la orden lscpu en un nodo de computo de atcgrid es la siguiente:

```
[FelixRamirezGarcia A3estudiante20@atcgrid:~/bp0] 2019-03-04 Monday
$echo 'lscpu' | qsub -q ac
5613.atcgrid
[FelixRamirezGarcia A3estudiante20@atcgrid:~/bp0] 2019-03-04 Monday
$cat STDIN.e*
[FelixRamirezGarcia A3estudiante20@atcgrid:~/bp0] 2019-03-04 Monday
$cat STDIN.o*
CPU op-mode(s):        32-bit, 64-bit
CPU(s):                24
On-line CPU(s) list:   0-23
CPU family:            6
Model name:            Intel(R) Xeon(R) CPU           E5645  @ 2.40GHz
CPU MHz:               2000.000
CPU max MHz:           2401,0000
CPU min MHz:           1600,0000
NUMA node0 CPU(s):     0-5,12-17
NUMA node1 CPU(s):     6-11,18-23
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                24
On-line CPU(s) list:   0-23
Thread(s) per core:    2
Core(s) per socket:    6
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 44
Model name:            Intel(R) Xeon(R) CPU           E5645  @ 2.40GHz
Stepping:              2
CPU MHz:               1600.000
CPU max MHz:           2401,0000
CPU min MHz:           1600,0000
BogoMIPS:              4800.17
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
```

(b) ¿Cuántos cores físicos y cuántos cores lógicos tienen los nodos de cómputo de atcgrid y del PC? Razonar las respuestas

Los nodos de computo de atcgrid tienen 2 sockets con 6 cores por cada uno con dos hebras por cada core. Por lo que cada nodo de computo tiene 12 cores físicos (2 sockets * 6 cores) y 24 cores lógicos (2 sockets * 6 cores * 2 hebras).

Mi PC tiene 1 socket con dos cores y dos hebras por core. Por lo que tiene 2 cores físicos (1 socket * 2 cores) y 4 cores lógicos (1 socket * 2 cores * 2 hebras).

2. Compilar y ejecutar en el PC el código `HelloOMP.c` del seminario (recordar que se debe usar un directorio independiente para cada ejercicio dentro de `bp0` que contenga todo lo utilizado, implementado o generado durante el desarrollo del mismo, para el presente ejercicio el directorio sería `ejer2`, como se indica en las normas de prácticas).

(a) Adjuntar capturas de pantalla que muestren la compilación y ejecución en el PC.

```
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer2] 2019-03-04 Monday
$gcc -O2 -fopenmp -o HelloOMP HelloOMP.c
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer2] 2019-03-04 Monday
$
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer2] 2019-03-04 Monday
$./HelloOMP
(0:!!!Hello world!!!)
(3:!!!Hello world!!!)
(1:!!!Hello world!!!)
(2:!!!Hello world!!!)
```

(b) Justificar el número de "Hello world" que se imprimen en pantalla en ambos casos teniendo en cuenta la salida que devuelve `lscpu`.

La frase 'Hello world' se imprime 4 veces , una por cada core lógico que tiene el PC.

3. Copiar el ejecutable de `HelloOMP.c` que ha generado anteriormente y que se encuentra en el directorio `ejer2 del` PC al directorio `ejer2` de su home en el *front-end* de atcgrid. Ejecutar (desde el directorio de este ejercicio, `ejer3`) este código en un nodo de cómputo de atcgrid usando la cola ac del gestor de colas (no use ningún *script*).

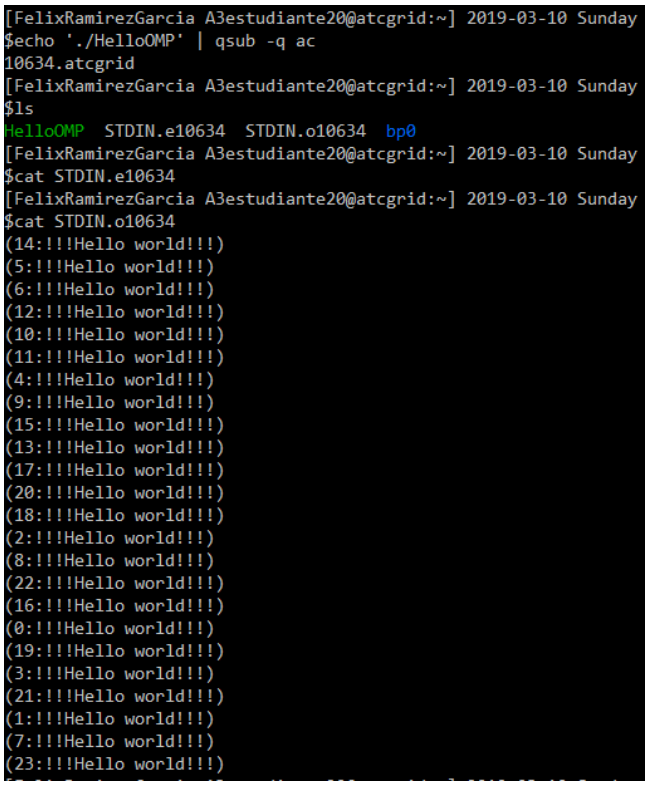
(a) Adjuntar capturas de pantalla que muestren la copia del fichero, el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas.

Para la copia del fichero se ha usado el comando rsync de la siguiente forma :

```
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer2] 2019-03-10 Sunday
$rsync -avh ./HelloOMP A3estudiante20@atcgrid.ugr.es:/home/A3estudiante20/bp0/ejer
3/
A3estudiante20@atcgrid.ugr.es's password:
sending incremental file list
HelloOMP

sent 8.79K bytes  received 35 bytes  255.74 bytes/sec
total size is 8.68K  speedup is 0.98
```

El envío a la cola de ejecución y el resultado de la ejecución se muestran en la siguiente imagen:



(b) Justificar el número de "Hello world" que se observan en el resultado teniendo en cuenta la salida que devuelve `lscpu`.

La frase 'Hello world' se imprime 24 veces , una por cada core lógico que tiene el nodo de computo de atcgrid.

4. Modificar en su PC `HelloOMP.c` para que se imprima "world" en un `printf` distinto al usado para "Hello", en ambos `printf` se debe imprimir el identificador del thread que escribe en pantalla. Nombrar al código resultante `HelloOMP2.c`. Compilar este nuevo código en el PC y ejecutarlo. Copiar el fichero ejecutable resultante en el front-end de atcgrid (directorio `ejer4`). Ejecutar el código en un nodo de cómputo de atcgrid usando el script `script_helloomp.sh` del seminario (el nombre del ejecutable en el script debe ser `HelloOMP2`).

(a) Adjuntar capturas de pantalla que muestren el nuevo código, la compilación, la copia a atcgrid, el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas.

El nuevo código es el que muestra la siguiente imagen:

HelloOMP2.c

```c
#include <stdio.h>
#include <omp.h>

int main(void) {

#pragma omp parallel
  printf("(%d: !!!Hello)", omp_get_thread_num());
  printf("(%d: world!!!)\n", omp_get_thread_num());

return(0);

}
```

La compilación y ejecución en mi PC la muestra la siguiente imagen:

```
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer4] 2019-03-10 Sunday
$gcc -O2 -fopenmp -o HelloOMP2 HelloOMP2.c
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer4] 2019-03-10 Sunday
$./HelloOMP2
(0: !!!Hello)(1: !!!Hello)(2: !!!Hello)(3: !!!Hello)(0: world!!!)
```

La copia a atcgrid del ejecutable desde mi PC la muestra la siguiente imagen:

```
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer4] 2019-03-10 Sunday
$rsync -avh ./HelloOMP2 A3estudiante20@atcgrid.ugr.es:/home/A3estudiante20/bp0/eje
r4/
A3estudiante20@atcgrid.ugr.es's password:
sending incremental file list
HelloOMP2

sent 8.79K bytes  received 35 bytes  452.51 bytes/sec
total size is 8.68K  speedup is 0.98
```

El envío a la cola de la ejecución y el resultado de esta ejecución en un nodo de computo de atcgrid lo muestra la siguiente imagen:

```
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-10 Sunday
$
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-10 Sunday
$echo './script_helloomp.sh' | qsub -q ac
10808.atcgrid
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-10 Sunday
$ls
HelloOMP2  STDIN.e10808  STDIN.o10808  bp0  script_helloomp.sh
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-10 Sunday
$cat STDIN.o10808
Id. usuario del trabajo: A3estudiante20
Id.  del trabajo: 10808.atcgrid
Nombre  del trabajo especificado por usuario: STDIN
Directorio en el que se ha ejecutado qsub: /home/A3estudiante20
Directorio de trabajo: /home/A3estudiante20
Cola: ac
Nodo que ejecuta qsub: atcgrid
Nodos asignados al trabajo:
atcgrid2
Nº de threads inicial: 12
Directorio de trabajo: /home/A3estudiante20

1.Para 12 threads:
(0: !!!Hello)(1: !!!Hello)(10: !!!Hello)(3: !!!Hello)(5: !!!Hello)(2: !!!Hello)(8: !!!Hello)(11: !!!Hello)(6: !!!Hello)(4: !!!Hello)(7: !!!Hello)(9: !!!Hello)(0: world!
!!)

1.Para 6 threads:
(0: !!!Hello)(1: !!!Hello)(3: !!!Hello)(2: !!!Hello)(4: !!!Hello)(5: !!!Hello)(0: world!!!)

1.Para 3 threads:
(1: !!!Hello)(0: !!!Hello)(2: !!!Hello)(0: world!!!)

1.Para 1 threads:
(0: !!!Hello)(0: world!!!)
```

(b) ¿Qué nodo de cómputo de atcgrid ha ejecutado el script? Explicar cómo ha obtenido esta información.

El nodo de computo atcgrid2. Esta informacion se ha obtenido en la ejecucion del script. El contenido se almacena en la variable `$PBS_NODEFILE`

(c) ¿Qué ocurre si se ejecuta el script usando `./HelloOMP2` en lugar de `$PBS_O_WORKDIR/HelloOMP2`? Razonar respuesta y adjuntar capturas de pantalla que muestren lo que ocurre.

Si se ejecuta el script usando `./HelloOMP2` se ejecuta a nivel de usuario y no se envía a la cola de procesos del nodo de atcgrid. En la imagen anterior se muestra la ejecución en la cola de atcgrid y en la siguiente imagen se muestra la ejecución a nivel de usuario.

```
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-10 Sunday
$./HelloOMP2
(6: !!!Hello)(0: !!!Hello)(7: !!!Hello)(1: !!!Hello)(3: !!!Hello)(5: !!!Hello)(2:
!!!Hello)(4: !!!Hello)(0: world!!!)
```

Parte II. Resto de ejercicios

5. Generar en el PC 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`). El comentario inicial del código muestra la orden para compilar (siempre hay que usar `-O2` al compilar como se indica en las normas de prácticas). Incorporar volcados de pantalla que demuestren la compilación y la ejecución correcta del código en el PC (leer lo indicado al respecto en las normas de prácticas).

La compilacion y ejecucion del programa SumaVectoresC se muestra en la siguiente imagen:

```
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer5] 2019-03-10 Sunday
$gcc -O2 -o SumaVectoresC SumaVectoresC.c
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer5] 2019-03-10 Sunday
$./SumaVectoresC 10000
Tamaño Vectores:10000 (4 B)
Tiempo:0.000109200       / Tamaño Vectores:10000        / V1[0]+V2[0]=V3[0](1000.0
00000+1000.000000=2000.000000) / / V1[9999]+V2[9999]=V3[9999](1999.900000+0.100000
=2000.000000) /
```

6. En el código del Listado 1 se utiliza la función `clock_gettime()` para obtener el tiempo de ejecución del trozo de código que calcula la suma de vectores. El código se imprime la variable `ncgt`,
(a) ¿qué contiene esta variable?

La variable contiene el tiempo de ejecucion de la suma de los vectores con precisión de nanosegundos.

(b) ¿en qué estructura de datos devuelve `clock_gettime()` la información de tiempo (indicar el tipo de estructura de datos, describir la estructura de datos, e indicar los tipos de datos que usa)?

La estructura de datos que devuelve es un struct timespec y se utiliza para registrar un intervalo dividido en segundos y nanosegundos. Ésta contiene dos variables de tipo time_t (tv_sec y tv_nsec) .

(c) ¿qué información devuelve exactamente la función `clock_gettime()`en la estructura de datos descrita en el apartado (b)? ¿qué representan los valores numéricos que devuelve?

La estructura de datos que devuelve es un struct timespec y se utiliza para registrar un intervalo dividido en segundos y nanosegundos. Ésta contiene dos variables de tipo time_t (tv_sec y tv_nsec) una para los segundos y otra para los nanosegundos.

7. Ejecutar en atcgrid el código generado en el apartado anterior usando el script del Listado 2. Ejecutar el código también en el PC para los mismos tamaños de los vectores. ¿Se obtiene error para alguno de los tamaños? En caso afirmativo, ¿a qué se debe este error? (Incorporar volcados de pantalla como se indica en las normas de prácticas)

Para enviar el ejecutable y el script a atcgrid se ha usado el comando que muestra la siguiente imagen:

```
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer7] 2019-03-11 Monday
$rsync -avh ./*  A3estudiante20@atcgrid.ugr.es:/home/A3estudiante20/
A3estudiante20@atcgrid.ugr.es's password:
sending incremental file list
SumaVectoresC
SumaVectoresC.c
script_suma_vectores.sh

sent 17.08K bytes  received 73 bytes  1.18K bytes/sec
total size is 16.83K  speedup is 0.98
```

Los comandos para la ejecución y salidas del script se muestran en la siguiente imagen:

```
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-11 Monday
$echo './SumaVectores.sh' | qsub -q ac
10830.atcgrid
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-11 Monday
$cat STDIN.e10830
./SumaVectores.sh: línea 19: 16034 Violación de segmento  (`core' generado) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: línea 19: 16036 Violación de segmento  (`core' generado) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: línea 19: 16039 Violación de segmento  (`core' generado) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: línea 19: 16043 Violación de segmento  (`core' generado) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: línea 19: 16045 Violación de segmento  (`core' generado) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: línea 19: 16047 Violación de segmento  (`core' generado) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: línea 19: 16049 Violación de segmento  (`core' generado) $PBS_O_WORKDIR/SumaVectoresC $N
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-11 Monday
$cat STDIN.o10830
Id. usuario del trabajo: A3estudiante20
Id. del trabajo: 10830.atcgrid
Nombre del trabajo especificado por usuario: STDIN
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/A3estudiante20
Cola: ac
Nodos asignados al trabajo:
atcgrid3
Tamaño Vectores:65536 (4 B)
Tiempo:0.000455990       / Tamaño Vectores:65536        / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.100000+0.10000
0=13107.200000) /
Tamaño Vectores:131072 (4 B)
Tiempo:0.000931112       / 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) /
Tamaño Vectores:262144 (4 B)
Tiempo:0.001754413       / 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) /
Tamaño Vectores:524288 (4 B)
Tiempo:0.002846846       / Tamaño Vectores:524288       / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[524287]=V3[524287](104857.500000+
0.100000=104857.600000) /
```

Ahora lo volvemos a ejecutar pero en local con las siguientes ordenes :

```
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/AC/Practicas/Entrega/bp0/ejer7] 2019-03-11 Monday
$chmod +x script_local.sh
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/AC/Practicas/Entrega/bp0/ejer7] 2019-03-11 Monday
$./script_local.sh SumaVectoresC
Tamaño Vectores:65536 (4 B)
Tiempo:0.001544600       / Tamaño Vectores:65536        / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.100000+0.10000
0=13107.200000) /
Tamaño Vectores:131072 (4 B)
Tiempo:0.002463100       / 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) /
Tamaño Vectores:262144 (4 B)
Tiempo:0.004287600       / 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) /
Tamaño Vectores:524288 (4 B)
./script_local.sh: line 5:   702 Segmentation fault      (core dumped) ./$1 $N
Tamaño Vectores:1048576 (4 B)
./script_local.sh: line 5:   703 Segmentation fault      (core dumped) ./$1 $N
Tamaño Vectores:2097152 (4 B)
./script_local.sh: line 5:   704 Segmentation fault      (core dumped) ./$1 $N
Tamaño Vectores:4194304 (4 B)
./script_local.sh: line 5:   705 Segmentation fault      (core dumped) ./$1 $N
Tamaño Vectores:8388608 (4 B)
./script_local.sh: line 5:   706 Segmentation fault      (core dumped) ./$1 $N
Tamaño Vectores:16777216 (4 B)
./script_local.sh: line 5:   707 Segmentation fault      (core dumped) ./$1 $N
Tamaño Vectores:33554432 (4 B)
./script_local.sh: line 5:   708 Segmentation fault      (core dumped) ./$1 $N
Tamaño Vectores:67108864 (4 B)
./script_local.sh: line 5:   709 Segmentation fault      (core dumped) ./$1 $N
```

Se producen errores en la ejecución de ambos casos. El tipo de error es un segmentation fault y se produce porque se accede a posiciones de memoria que no están reservadas.

8. Generar los ejecutables del código fuente C para vectores globales y para dinámicos. Ejecutar los dos códigos en un nodo de cómputo de atcgrid usando un script como el del Listado 2 para el mismo rango de tamaños utilizado en el ejercicio anterior. Hay que poner en el script el nombre de los ficheros ejecutables generados en este ejercicio. Ejecutar también los códigos en el PC. ¿Se obtiene error usando vectores globales o dinámicos? ¿A qué cree que es debido? (Incorporar volcados de pantalla como se indica en las normas de prácticas)

Para comenzar vamos a compilar ambos programas en local con el comando que muestra la siguiente imagen:

```
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer8] 2019-03-11 Monday
$gcc -o SumaVectoresC_dinamicos SumaVectoresC_dinamicos.c
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer8] 2019-03-11 Monday
$gcc -o SumaVectoresC_globales SumaVectoresC_globales.c
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer8] 2019-03-11 Monday
$
```

Después subimos toda la carpeta a atcgrid con el siguiente comando:

```
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer8] 2019-03-11 Monday
$rsync -avh ./*  A3estudiante20@atcgrid.ugr.es:/home/A3estudiante20/
A3estudiante20@atcgrid.ugr.es's password:
sending incremental file list
SumaVectoresC_dinamicos
SumaVectoresC_dinamicos.c
SumaVectoresC_globales
SumaVectoresC_globales.c
SumaVectores_dinamicos.sh
SumaVectores_globales.sh
compilacion.PNG
script_local.sh

sent 50.68K bytes  received 168 bytes  5.35K bytes/sec
total size is 50.09K  speedup is 0.99
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer8] 2019-03-11 Monday
$
```

Ahora ejecutamos ambos scripts en un nodo de atcgrid, las siguientes 4 capturas muestran el envío a la cola de procesos y las salidas de los scipts.

```
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-11 Monday
$echo './SumaVectores_dinamicos.sh' | qsub -q ac
10831.atcgrid
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-11 Monday
$cat STDIN.e10831
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-11 Monday
$cat STDIN.o10831
Id. usuario del trabajo: A3estudiante20
Id. del trabajo: 10831.atcgrid
Nombre del trabajo especificado por usuario: STDIN
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/A3estudiante20
Cola: ac
Nodos asignados al trabajo:
atcgrid1
Tamaño Vectores:65536 (4 B)
Tiempo:0.000887221       / Tamaño Vectores:65536        / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.100000+0.10000
0=13107.200000) /
Tamaño Vectores:131072 (4 B)
Tiempo:0.001812317       / 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) /
Tamaño Vectores:262144 (4 B)
Tiempo:0.003618923       / 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) /
Tamaño Vectores:524288 (4 B)
Tiempo:0.005871458       / Tamaño Vectores:524288       / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[524287]=V3[524287](104857.500000+
0.100000=104857.600000) /
Tamaño Vectores:1048576 (4 B)
Tiempo:0.010260892       / Tamaño Vectores:1048576      / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575](209715.10
0000+0.100000=209715.200000) /
Tamaño Vectores:2097152 (4 B)
Tiempo:0.016826394       / Tamaño Vectores:2097152      / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151](419430.30
0000+0.100000=419430.400000) /
Tamaño Vectores:4194304 (4 B)
Tiempo:0.031151032       / Tamaño Vectores:4194304      / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](838860.70
0000+0.100000=838860.800000) /
```

```
Tamaño Vectores:8388608 (4 B)
Tiempo:0.061541236      / 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) /
Tamaño Vectores:16777216 (4 B)
Tiempo:0.122479129      / Tamaño Vectores:16777216     / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[16777215](335
5443.100000+0.100000=3355443.200000) /
Tamaño Vectores:33554432 (4 B)
Tiempo:0.244281960      / Tamaño Vectores:33554432     / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33554431](671
0886.300000+0.100000=6710886.400000) /
Tamaño Vectores:67108864 (4 B)
Tiempo:0.487515351      / Tamaño Vectores:67108864     / V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / / V1[67108863]+V2[67108863]=V3[67108863](13
421772.700000+0.100000=13421772.800000) /
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-11 Monday
```

```
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-11 Monday
$echo './SumaVectores_globales.sh' | qsub -q ac
10832.atcgrid
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-11 Monday
$cat STDIN.e10832
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-11 Monday
$cat STDIN.o10832
Id. usuario del trabajo: A3estudiante20
Id. del trabajo: 10832.atcgrid
Nombre del trabajo especificado por usuario: STDIN
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/A3estudiante20
Cola: ac
Nodos asignados al trabajo:
atcgrid2
Tamaño Vectores:65536 (4 B)
Tiempo:0.000822726      / Tamaño Vectores:65536        / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.100000+0.10000
0=13107.200000) /
Tamaño Vectores:131072 (4 B)
Tiempo:0.001658879      / 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) /
Tamaño Vectores:262144 (4 B)
Tiempo:0.002297300      / 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) /
Tamaño Vectores:524288 (4 B)
Tiempo:0.004743774      / Tamaño Vectores:524288       / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[524287]=V3[524287](104857.500000+
0.100000=104857.600000) /
Tamaño Vectores:1048576 (4 B)
Tiempo:0.008580876      / Tamaño Vectores:1048576      / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575](209715.10
0000+0.100000=209715.200000) /
Tamaño Vectores:2097152 (4 B)
Tiempo:0.014565946      / Tamaño Vectores:2097152      / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151](419430.30
0000+0.100000=419430.400000) /
Tamaño Vectores:4194304 (4 B)
Tiempo:0.025809342      / Tamaño Vectores:4194304      / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](838860.70
```

```
Tiempo:0.025809342      / Tamaño Vectores:4194304      / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](838860.70
0000+0.100000=838860.800000) /
Tamaño Vectores:8388608 (4 B)
Tiempo:0.051248287      / 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) /
Tamaño Vectores:16777216 (4 B)
Tiempo:0.101761180      / Tamaño Vectores:16777216     / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[16777215](335
5443.100000+0.100000=3355443.200000) /
Tamaño Vectores:33554432 (4 B)
Tiempo:0.202484151      / Tamaño Vectores:33554432     / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33554431](671
0886.300000+0.100000=6710886.400000) /
Tamaño Vectores:67108864 (4 B)
Tiempo:0.201736276      / Tamaño Vectores:33554432     / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33554431](671
0886.300000+0.100000=6710886.400000) /
[FelixRamirezGarcia A3estudiante20@atcgrid:~] 2019-03-11 Monday
$
```

Por ultimo incluimos 2 capturas de la ejecución de ambos scripts en mi PC .

```
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/AC/Practicas/Entrega/bp0/ejer8] 2019-03-11 Monday
$./script_local.sh SumaVectoresC_globales
Tamaño Vectores:65536 (4 B)
Tiempo:0.000432200       / Tamaño Vectores:65536       / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.100000+0.10000
0=13107.200000) /
Tamaño Vectores:131072 (4 B)
Tiempo:0.000793600       / 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) /
Tamaño Vectores:262144 (4 B)
Tiempo:0.001201100       / 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) /
Tamaño Vectores:524288 (4 B)
Tiempo:0.003670600       / Tamaño Vectores:524288      / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[524287]=V3[524287](104857.500000+
0.100000=104857.600000) /
Tamaño Vectores:1048576 (4 B)
Tiempo:0.005656700       / Tamaño Vectores:1048576     / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575](209715.10
0000+0.100000=209715.200000) /
Tamaño Vectores:2097152 (4 B)
Tiempo:0.012189000       / Tamaño Vectores:2097152     / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151](419430.30
0000+0.100000=419430.400000) /
Tamaño Vectores:4194304 (4 B)
Tiempo:0.044123700       / Tamaño Vectores:4194304     / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](838860.70
0000+0.100000=838860.800000) /
Tamaño Vectores:8388608 (4 B)
Tiempo:0.049230500       / 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) /
Tamaño Vectores:16777216 (4 B)
Tiempo:0.090062800       / Tamaño Vectores:16777216    / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[16777215](335
5443.100000+0.100000=3355443.200000) /
Tamaño Vectores:33554432 (4 B)
Tiempo:0.184863000       / Tamaño Vectores:33554432    / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33554431](671
0886.300000+0.100000=6710886.400000) /
Tamaño Vectores:67108864 (4 B)
Tiempo:0.213959900       / Tamaño Vectores:33554432    / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33554431](671
0886.300000+0.100000=6710886.400000) /
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/AC/Practicas/Entrega/bp0/ejer8] 2019-03-11 Monday
```

```
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/AC/Practicas/Entrega/bp0/ejer8] 2019-03-11 Monday
$./script_local.sh SumaVectoresC_dinamicos
Tamaño Vectores:65536 (4 B)
Tiempo:0.001515900       / Tamaño Vectores:65536       / V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) / / V1[65535]+V2[65535]=V3[65535](13107.100000+0.10000
0=13107.200000) /
Tamaño Vectores:131072 (4 B)
Tiempo:0.002913000       / 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) /
Tamaño Vectores:262144 (4 B)
Tiempo:0.006766900       / 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) /
Tamaño Vectores:524288 (4 B)
Tiempo:0.011223000       / Tamaño Vectores:524288      / V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) / / V1[524287]+V2[524287]=V3[524287](104857.500000+
0.100000=104857.600000) /
Tamaño Vectores:1048576 (4 B)
Tiempo:0.024847400       / Tamaño Vectores:1048576     / V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715.200000) / / V1[1048575]+V2[1048575]=V3[1048575](209715.10
0000+0.100000=209715.200000) /
Tamaño Vectores:2097152 (4 B)
Tiempo:0.051171000       / Tamaño Vectores:2097152     / V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430.400000) / / V1[2097151]+V2[2097151]=V3[2097151](419430.30
0000+0.100000=419430.400000) /
Tamaño Vectores:4194304 (4 B)
Tiempo:0.097002200       / Tamaño Vectores:4194304     / V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860.800000) / / V1[4194303]+V2[4194303]=V3[4194303](838860.70
0000+0.100000=838860.800000) /
Tamaño Vectores:8388608 (4 B)
Tiempo:0.236623600       / 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) /
Tamaño Vectores:16777216 (4 B)
Tiempo:0.610111500       / Tamaño Vectores:16777216    / V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355443.200000) / / V1[16777215]+V2[16777215]=V3[16777215](335
5443.100000+0.100000=3355443.200000) /
Tamaño Vectores:33554432 (4 B)
Tiempo:0.913293200       / Tamaño Vectores:33554432    / V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33554431](671
0886.300000+0.100000=6710886.400000) /
Tamaño Vectores:67108864 (4 B)
Tiempo:2.158136600       / Tamaño Vectores:67108864    / V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=13421772.800000) / / V1[67108863]+V2[67108863]=V3[67108863](13
421772.700000+0.100000=13421772.800000) /
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/AC/Practicas/Entrega/bp0/ejer8] 2019-03-11 Monday
```

En ninguno de los casos se obtienen errores usando vectores globales o dinámicos, ya que en el caso de la variable global se crea una constante con un tamaño suficiente para realizar las operaciones y en el caso de los vectores dinámicos se crean punteros con suficiente capacidad.

9. Rellenar una tabla como la Tabla 1 **en una hoja de cálculo** con los tiempos de ejecución obtenidos en los ejercicios anteriores para el trozo de código que realiza la suma de vectores. Debe haber una tabla para atcgrid y otra para su PC en la hoja de cálculo. En la columna "Bytes de un vector" hay que poner el total de bytes reservado para un vector. Con ayuda de la hoja de cálculo representar **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 (por tanto, los valores de la segunda columna de la tabla, que están en escala logarítmica, deben estar en el eje x). Utilizar escala logarítmica en el eje de ordenadas (eje y). (NOTA: Se

recomienda usar en la hoja de cálculo el mismo separador para decimales que usan los códigos. Este separador se puede modificar en la hoja de cálculo.)

(a) Copiar las tablas y la gráfica en el cuaderno de prácticas.

Tabla para atcgrid:

Nº componentes	Tamaño en B	T local	T global	T dinamic
65536	262144	0.000455990	0.000822726	0.000887221
131072	524288	0.000931112	0.001658879	0.001812317
262144	1048576	0.001754413	0.002297300	0.003618923
524288	2097152	0.002846846	0.004743774	0.005871458
1048576	4194304		0.008580876	0.010260892
2097152	8388608		0.014565946	0.016826394
4194304	16777216		0.025809342	0.031151032
8388608	33554432		0.051248287	0.061541236
16777216	67108864		0.101761180	0.122479129
33554432	134217728		0.202484151	0.244281960
67108864	268435456		0.201736276	0.487515351

Tabla para local:

Nº componentes	Tamaño en B	T local	T global	T dinamic
65536	262144	0.000903700	0.000307600	0.000935300
131072	524288	0.001822700	0.000583200	0.001889700
262144	1048576	0.003878800	0.001126000	0.003819000
524288	2097152		0.002662400	0.009329100
1048576	4194304		0.006907300	0.015297200
2097152	8388608		0.016021800	0.035208500
4194304	16777216		0.021264700	0.061193800
8388608	33554432		0.046041200	0.131784500
16777216	67108864		0.083106500	0.248037800
33554432	134217728		0.193552500	0.539875000
67108864	268435456		0.137038400	1.896302500

(b) ¿Hay diferencias en los tiempos de ejecución?

Hay pocas diferencias, siendo mas rápida la ejecución en mi PC para global . Asi como mas lenta en mi PC para local y dinamic.

10. **(a)** ¿Cuál es el máximo valor que se puede almacenar en la variable N teniendo en cuenta su tipo? Razonar respuesta.

El máximo valor para N es 4294967294, asociado a el tamaño de un unsigned int. (4B)

(b) Modificar el código fuente C (en el PC) 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 y generar el ejecutable. ¿Qué ocurre? ¿A qué es debido? (Incorporar volcados de pantalla que muestren lo que ocurre)

```
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer10] 2019-03-11 Monday
$gcc -o SumaVectoresC SumaVectoresC.c
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer10] 2019-03-11 Monday
$./SumaVectoresC
Tamaño Vectores:4294967295 (4 B)
Tiempo:0.157522200       / Tamaño Vectores:33554432     / V1[0]+V2[0]=V3[0](335544
3.200000+3355443.200000=6710886.400000) / / V1[33554431]+V2[33554431]=V3[33554431]
(6710886.300000+0.100000=6710886.400000) /
[FelixRamirezGarcia felix@DESKTOP-8PO8DVP:/mnt/c/Users/felix/Desktop/Google-Drive/
AC/Practicas/Entrega/bp0/ejer10] 2019-03-11 Monday
$
```

Lo que ocurre es que vuelve a ajustar el tamaño a 33554432, por lo que se ejecuta correctamente.

Listado 1 . Código C que suma dos vectores

```
/* SumaVectoresC.c
 Suma de dos vectores: v3 = v1 + v2

 Para compilar usar (-lrt: real time library, no todas las versiones de gcc necesitan que se incluya
-lrt):
         gcc  -O2 SumaVectores.c -o SumaVectores –lrt
         gcc  -O2 –S SumaVectores.c –lrt   //para generar el código ensamblador

 Para ejecutar use: SumaVectoresC longitud
*/

#include <stdlib.h> // biblioteca con funciones atoi(), malloc() y free()
#include <stdio.h> // biblioteca donde se encuentra la función printf()
#include <time.h>       // biblioteca donde se encuentra la función clock_gettime()

//Sólo puede estar definida una de las tres constantes VECTOR_ (sólo uno de los ...
//tres defines siguientes puede estar descomentado):
//#define VECTOR_LOCAL    // descomentar para que los vectores sean variables ...
                          // locales (si se supera el tamaño de la pila se ...
                          // generará el error "Violación de Segmento")
//#define VECTOR_GLOBAL// descomentar para que los vectores sean variables ...
                          // globales (su longitud no estará limitada por el ...
                          // tamaño de la pila del programa)
#define VECTOR_DYNAMIC    // descomentar para que los vectores sean variables ...
                          // dinámicas (memoria reutilizable durante la ejecución)
#ifdef VECTOR_GLOBAL
#define MAX 33554432         //=2^25
double v1[MAX], v2[MAX], v3[MAX];
#endif

int main(int argc, char** argv){

   int i;
   struct timespec cgt1,cgt2; double ncgt; //para tiempo de ejecución

   //Leer argumento de entrada (nº de componentes del vector)
   if (argc<2){
      printf("Faltan nº componentes del vector\n");
      exit(-1);
   }

   unsigned int N = atoi(argv[1]); // Máximo N =2^32-1=4294967295 (sizeof(unsigned int) = 4 B)
   #ifdef VECTOR_LOCAL
   double v1[N], v2[N], v3[N];   // Tamaño variable local en tiempo de ejecución ...
                                 // disponible en C a partir de actualización C99
   #endif
   #ifdef VECTOR_GLOBAL
   if (N>MAX) N=MAX;
   #endif
   #ifdef VECTOR_DYNAMIC
   double *v1, *v2, *v3;
   v1 = (double*) malloc(N*sizeof(double));// malloc necesita el tamaño en bytes
   v2 = (double*) malloc(N*sizeof(double)); //si no hay espacio suficiente malloc devuelve NULL
   v3 = (double*) malloc(N*sizeof(double));
```

```
      if ( (v1==NULL) || (v2==NULL) || (v3==NULL) ){
      printf("Error en la reserva de espacio para los vectores\n");
      exit(-2);
   }
   #endif

   //Inicializar vectores
   for(i=0; i<N; i++){
      v1[i] = N*0.1+i*0.1; v2[i] = N*0.1-i*0.1; //los valores dependen de N
   }

   clock_gettime(CLOCK_REALTIME,&cgt1);
   //Calcular suma de vectores
   for(i=0; i<N; i++)
      v3[i] = v1[i] + v2[i];

   clock_gettime(CLOCK_REALTIME,&cgt2);
   ncgt=(double) (cgt2.tv_sec-cgt1.tv_sec)+
          (double) ((cgt2.tv_nsec-cgt1.tv_nsec)/(1.e+9));

   //Imprimir resultado de la suma y el tiempo de ejecución
   if (N<10) {
   printf("Tiempo(seg.):%11.9f\t / Tamaño Vectores:%u\n",ncgt,N);
   for(i=0; i<N; i++)
      printf("/ V1[%d]+V2[%d]=V3[%d](%8.6f+%8.6f=%8.6f) /\n",
                i,i,i,v1[i],v2[i],v3[i]);
   }
   else
      printf("Tiempo(seg.):%11.9f\t / Tamaño Vectores:%u\t/ V1[0]+V2[0]=V3[0](%8.6f+%8.6f=%8.6f) / /
                V1[%d]+V2[%d]=V3[%d](%8.6f+%8.6f=%8.6f) /\n",
                ncgt,N,v1[0],v2[0],v3[0],N-1,N-1,N-1,v1[N-1],v2[N-1],v3[N-1]);


   #ifdef VECTOR_DYNAMIC
   free(v1); // libera el espacio reservado para v1
   free(v2); // libera el espacio reservado para v2
   free(v3); // libera el espacio reservado para v3
   #endif
   return 0;
}
```

Listado 2 . Script para la suma de vectores (SumaVectores.sh). Se supone en el script que el fichero a ejecutar se llama SumaVectorC.

```
#!/bin/bash
#Todos los scripts que se hagan para atcgrid deben incluir lo siguiente:
#Se asigna al trabajo el nombre SumaVectoresC_vlocales
#PBS -N SumaVectoresC_vlocales
#Se asigna al trabajo la cola ac
#PBS -q ac
#Se imprime información del trabajo usando variables de entorno de PBS
echo "Id. usuario del trabajo: $PBS_O_LOGNAME"
echo "Id. del trabajo: $PBS_JOBID"
```

```
echo "Nombre  del trabajo especificado por usuario: $PBS_JOBNAME"
echo "Nodo que ejecuta qsub: $PBS_O_HOST"
echo "Directorio en el que se ha ejecutado qsub: $PBS_O_WORKDIR"
echo "Cola: $PBS_QUEUE"
echo "Nodos asignados al trabajo:"
cat $PBS_NODEFILE
# FIN del trozo que deben incluir todos los scripts

#para N potencia de 2 desde 2^16 a 2^26
for ((N=65536;N<67108865;N=N*2))
do
    Poner_el_camino_al_ejecutable/SumaVectoresC $N
done
```