

2º curso / 2º
cuatr.
Grado Ing. In-
form.
Doble Grado
Ing. Inform. y
Mat.

Arquitectura de Computadores (AC)

Cuaderno de prácticas.

Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos): Christian Andrades Molina

Grupo de prácticas: B2

Fecha de entrega: 11/03/2014 (hasta las 12 pm)

Fecha evaluación en clase: 19/03/2014

-
1. En el primer ejemplo de ejecución en atcgrid usando TORQUE se ejecuta el ejemplo `He-llOMP.c` de la página 10 del seminario usando la siguiente orden: `echo 'hello/He-llOMP' | 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: Indica el nombre de la cola a la que se mandará el trabajo.

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

RESPUESTA: Mediante el comando `qstat`.

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

RESPUESTA: Verificando el fichero generado de extensión `.e` (igualmente podría verificarse sin abrirlo si el fichero tiene un tamaño `> 0`).

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

RESPUESTA: Transfiriendo el fichero de extensión `.o` al PC local y visualizarlo con el comando `cat`.

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

RESPUESTA: Porque el cluster de prácticas atcgrid posee 2 procesadores a 12 hilos cada uno dando lugar a 24 saludos Hello World.

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: Porque en el script ya se asigna el trabajo en la cola `ac` mediante `-q`.

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

RESPUESTA: 4

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

RESPUESTA: 12 para 12 threads, 6 para 6 threads, 4 para 4 threads y 1 para 1 thread.

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: `echo "Directorio: $PBS_O_WORKDIR"`

```
Id. usuario del trabajo: B2estudiante2
Id. del trabajo: 27417.atcgrid
Nombre del trabajo especificado por usuario: helloomp
Nodo que ejecuta qsub: atcgrid
Cola: ac
Directorio: /home/B2estudiante2/hello
Nodos asignados al trabajo:
atcgrid1
No de threads inicial:12
```

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: `echo 'cat /proc/cpuinfo' | qsub -q ac` en `ssh`

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++:

$$v3 = v1 + v2; \quad v3(i) = v1(i) + v2(i), \quad i=0, \dots, N-1$$

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 devuelve la función `clock_gettime()`?

RESPUESTA: 1) `ncgt` contiene el tiempo de ejecución del bucle siguiente, medido con `clock_gettime`:

```
for(int i=0; i<N; i++)
    v3[i]=v1[i]+v2[i];
```

2) `clock_gettime(clock_realtime)` devuelve la hora del reloj del sistema.

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: En el código C utiliza la función `malloc` para crear los vectores que necesita el tamaño en bytes dado por `sizeof`. Si no hay espacio suficiente devuelve `null`. En C++ se crea a partir de `new`. Sin espacio suficiente `new` generará entonces una excepción; error.

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:

```
-bash-4.2$ cat STDIN.o27444
Tiempo(seg.):0.000000160 / Tamaño Vectores:10 / V1[0]+V2[0]=V3[0](1.000000+1.000000=2.000000) / / V1[9]+V2[9]=V3[9](1.900000+0.100000=2.000000)/
-bash-4.2$
```

7. Ejecutar en `atcgrid` el código generado en el apartado anterior usando el script del Listado 3. Generar el ejecutable usando la opción de optimización `-O2` tal y como se indica en el comentario que hay al principio del programa. Ejecutar el código también en su PC local para los mismos tamaños. ¿Se obtiene error para alguno de los tamaños? En caso afirmativo, ¿a qué se debe este error?

RESPUESTA: Si. Porque se supera el tamaño de la pila y se genera así un error "Violación de Segmento".

```
-bash-4.2$ cat SumaVectoresC_vlocales.o28741
Id. usuario del trabajo: B2estudiante2
Id. del trabajo: 28741.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vlocales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/B2estudiante2/ejercicio1
Cola: ac
Nodos asignados al trabajo:
atcgrid1
Tiempo(seg.):0.000437839 / Tamaño Vectores:65536 / V1[0]+V2[0]=V3
[0](6553.600000+6553.600000=13107.200000) // V1[65535]+V2[65535]=V3[65535](131
07.100000+0.100000=13107.200000)/
Tiempo(seg.):0.000907963 / Tamaño Vectores:131072 / V1[0]+V2[0]=V3
[0](13107.200000+13107.200000=26214.400000) // V1[131071]+V2[131071]=V3[131071]
(26214.300000+0.100000=26214.400000)/
Tiempo(seg.):0.001806460 / 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)/
```

8. Generar los ejecutables del código fuente C para vectores globales y para dinámicos. Generar 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: No. No está limitada por la pila y la memoria en dinámica se reutiliza.

```
Id. del trabajo: 28744.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vlocales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/B2estudiante2/ejercicio2
Cola: ac
Nodos asignados al trabajo:
atcgrid1
Tiempo(seg.):0.000549070 / Tamaño Vectores:65536 / V1[0]+V2[
0]=V3[0](6553.600000+6553.600000=13107.200000) // V1[65535]+V2[65535]=V3[6
5535](13107.100000+0.100000=13107.200000)/
Tiempo(seg.):0.000509080 / Tamaño Vectores:131072 / V1[0]+V2[
0]=V3[0](13107.200000+13107.200000=26214.400000) // V1[131071]+V2[131071]
=V3[131071](26214.300000+0.100000=26214.400000)/
Tiempo(seg.):0.001451974 / 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)/
Tiempo(seg.):0.002721492 / 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)/
Tiempo(seg.):0.004812446 / Tamaño Vectores:1048576 / V1[0]+V2[
0]=V3[0](104857.600000+104857.600000=209715.200000) // V1[1048575]+V2[104
8575]=V3[1048575](209715.100000+0.100000=209715.200000)/
Tiempo(seg.):0.008996114 / Tamaño Vectores:2097152 / V1[0]+V2[
0]=V3[0](209715.200000+209715.200000=419430.400000) // V1[2097151]+V2[209
7151]=V3[2097151](419430.300000+0.100000=419430.400000)/
Tiempo(seg.):0.017674015 / Tamaño Vectores:4194304 / V1[0]+V2[
0]=V3[0](419430.400000+419430.400000=838860.800000) // V1[4194303]+V2[419
4303]=V3[4194303](838860.700000+0.100000=838860.800000)/
Tiempo(seg.):0.034106848 / Tamaño Vectores:8388608 / V1[0]+V2[
0]=V3[0](838860.800000+838860.800000=1677721.600000) // V1[8388607]+V2[83
88607]=V3[8388607](1677721.500000+0.100000=1677721.600000)/
Tiempo(seg.):0.067818832 / Tamaño Vectores:16777216 / V1[0]+V2[
0]=V3[0](1677721.600000+1677721.600000=3355443.200000) // V1[16777215]+V2[
16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000)/
Tiempo(seg.):0.135235767 / Tamaño Vectores:33554432 / V1[0]+V2[
0]=V3[0](3355443.200000+3355443.200000=6710886.400000) // V1[33554431]+V2[
33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000)/
Tiempo(seg.):0.134119041 / Tamaño Vectores:33554432 / V1[0]+V2[
0]=V3[0](3355443.200000+3355443.200000=6710886.400000) // V1[33554431]+V2[
33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000)/
```

```
Id. usuario del trabajo: B2estudiante2
Id. del trabajo: 28743.atcgrid
Nombre del trabajo especificado por usuario: SumaVectoresC_vlocales
Nodo que ejecuta qsub: atcgrid
Directorio en el que se ha ejecutado qsub: /home/B2estudiante2/ejercicio2
Cola: ac
Nodos asignados al trabajo:
atcgrid1
Tiempo(seg.):0.000454598 / Tamaño Vectores:65536 / V1[0]+V2[
0]=V3[0](6553.600000+6553.600000=13107.200000) // V1[65535]+V2[65535]=V3[6
5535](13107.100000+0.100000=13107.200000)/
Tiempo(seg.):0.000892297 / Tamaño Vectores:131072 / V1[0]+V2[
0]=V3[0](13107.200000+13107.200000=26214.400000) // V1[131071]+V2[131071]
=V3[131071](26214.300000+0.100000=26214.400000)/
Tiempo(seg.):0.001802932 / 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)/
Tiempo(seg.):0.001797615 / 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)/
Tiempo(seg.):0.005115716 / Tamaño Vectores:1048576 / V1[0]+V2[
0]=V3[0](104857.600000+104857.600000=209715.200000) // V1[1048575]+V2[104
8575]=V3[1048575](209715.100000+0.100000=209715.200000)/
Tiempo(seg.):0.008773302 / Tamaño Vectores:2097152 / V1[0]+V2[
0]=V3[0](209715.200000+209715.200000=419430.400000) // V1[2097151]+V2[209
7151]=V3[2097151](419430.300000+0.100000=419430.400000)/
Tiempo(seg.):0.016726603 / Tamaño Vectores:4194304 / V1[0]+V2[
0]=V3[0](419430.400000+419430.400000=838860.800000) // V1[4194303]+V2[419
4303]=V3[4194303](838860.700000+0.100000=838860.800000)/
Tiempo(seg.):0.032845956 / Tamaño Vectores:8388608 / V1[0]+V2[
0]=V3[0](838860.800000+838860.800000=1677721.600000) // V1[8388607]+V2[83
88607]=V3[8388607](1677721.500000+0.100000=1677721.600000)/
Tiempo(seg.):0.064778162 / Tamaño Vectores:16777216 / V1[0]+V2[
0]=V3[0](1677721.600000+1677721.600000=3355443.200000) // V1[16777215]+V2[
16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000)/
Tiempo(seg.):0.129148830 / Tamaño Vectores:33554432 / V1[0]+V2[
0]=V3[0](3355443.200000+3355443.200000=6710886.400000) // V1[33554431]+V2[
33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000)/
Tiempo(seg.):0.254183530 / Tamaño Vectores:67108864 / V1[0]+V2[
0]=V3[0](6710886.400000+6710886.400000=13421772.800000) // V1[67108863]+V
2[67108863]=V3[67108863](13421772.700000+0.100000=13421772.800000)/
```

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.

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	13107,200000	0,000455859	0,00057874	0,000456934
131072	26214,400000	0,000900692	0,000535638	0,000963216
262144	52428,800000	0,001820911	0,001461353	0,00181862
524288	3355443,200000		0,002695036	0,002832922
1048576	6710886,400000		0,004617448	0,004627526
2097152	6710886,400000		0,009089334	0,008854798
4194304	3355443,200000		0,017247383	0,01674772
8388608	6710886,400000		0,034506874	0,032578062
16777216	6710886,400000		0,067752759	0,064882289
33554432	3355443,200000		0,134861274	0,12861854
67108864	6710886,400000		0,134205456	0,253109604

GRÁFICA

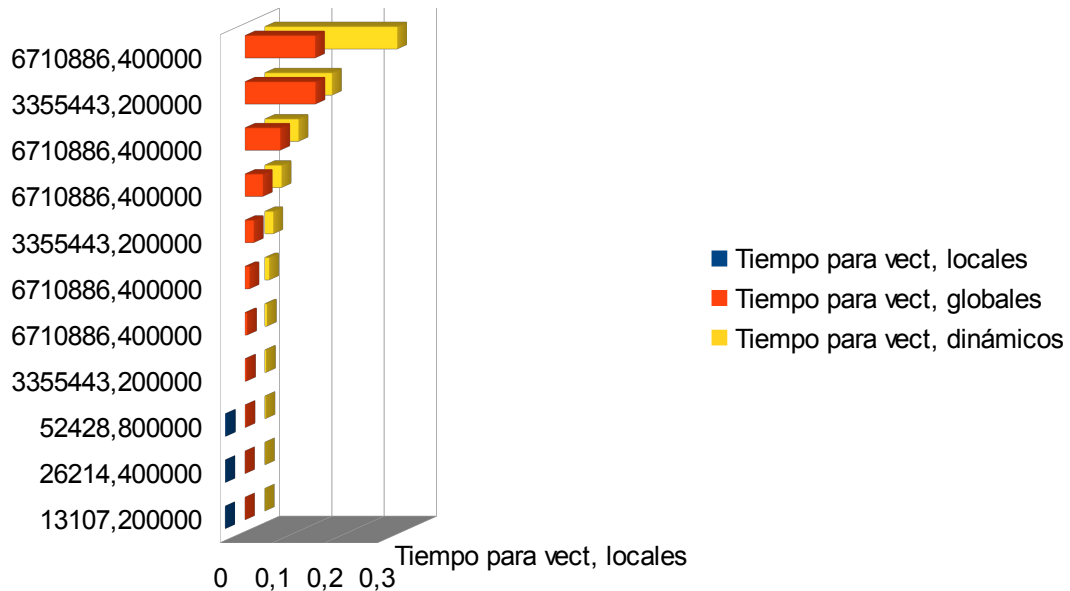
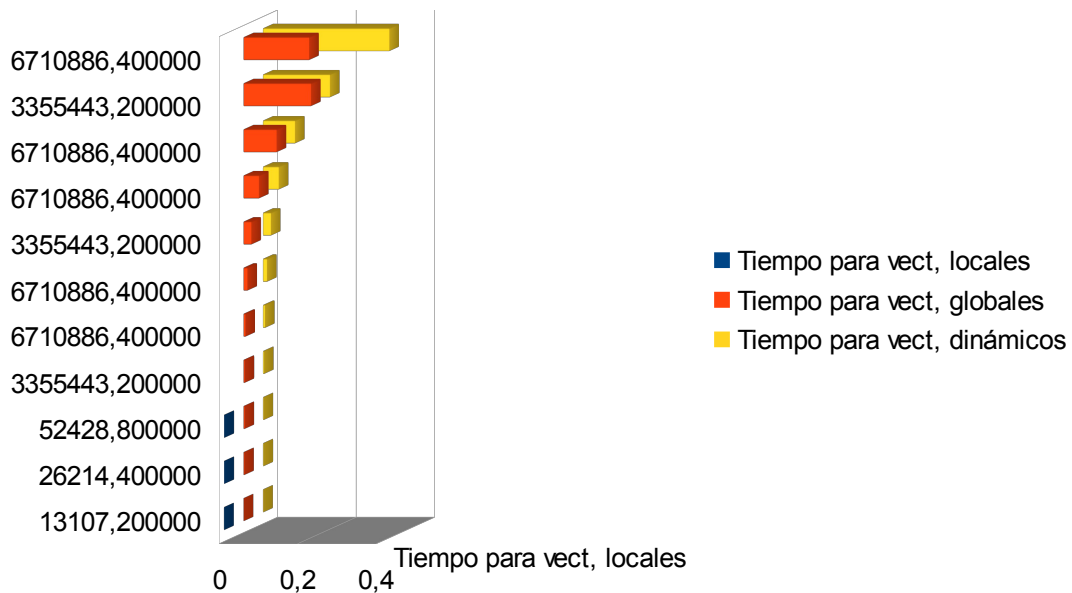


Tabla 2. Tiempos de ejecución de la suma de vectores para vectores locales, globales y dinámicos en PC

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	13107,200000	0,000662522	0,00067962	0,000677009
131072	26214,400000	0,00136331	0,002065352	0,001311833
262144	52428,800000	0,001302365	0,002828436	0,001556512
524288	3355443,200000		0,00285523	0,002670429
1048576	6710886,400000		0,005091039	0,005156976
2097152	6710886,400000		0,010244877	0,010303932
4194304	3355443,200000		0,020220256	0,020260748
8388608	6710886,400000		0,039734824	0,040385911
16777216	6710886,400000		0,084675679	0,081722806
33554432	3355443,200000		0,172293032	0,170800667
67108864	6710886,400000		0,167583511	0,322167443

GRÁFICA



- 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? Razone además por qué el máximo número que se puede almacenar en N es $2^{32}-1$.

RESPUESTA: