

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): Francisco Carrillo Pérez

Grupo de prácticas: B2

Fecha de entrega: !0/3/2015

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: Para indicarle cuál es la cola a la que queremos mandar el archivo.

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

RESPUESTA: Mirando con la opción `qstat`, si está significa que se sigue ejecutando. Otra forma es comprobando si se han creado los dos archivos `.o` / `.e`.

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

RESPUESTA: Si el archivo `.e` pesa más de 0KB significa que contiene algo y, por lo tanto, que ha habido errores.

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

RESPUESTA: En la ventana de stfp con el `get` (el archivo) copiamos el archivo desde el front-end al PC local. Una vez lo tenemos en el PC local hacemos un `cat` (el archivo) y lo vemos.

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

RESPUESTA: Porque tenemos 2 procesadores, con 6 cores cada uno y con multithreading por lo que son 24 hebras.

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 ya se especifica dentro del script.

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

RESPUESTA: El script se ejecuta 4 veces.

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

RESPUESTA: 12 la primera, 6 la segunda, 3 la tercera y 1 la última.
Porque en el for vas dividiendo el número máximo, que es 12, por 2.

3. Realizar las siguientes modificaciones en el script “iiiHello 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:

```
└─paco@paco-pc ~
```

```
└─$ cat helloomp.o1422
```

Id. usuario del trabajo: B2estudiante3

Id. del trabajo: 1422.atcgrid

Nombre del trabajo especificado por usuario: helloomp

Nodo que ejecuta qsub: atcgrid

Cola: ac

Nodos asignados al trabajo:

Nuevo workdir: /home/B2estudiante3/hello

atcgrid1

No de threads inicial: 12

Para 12 threads:

Para 6 threads:

Para 3 threads:

Para 1 threads:

No aparece nada ya que al no decirle el wordkir no se ejecuta.

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: Desde la pestaña de ssh mandamos a la cola de ac lo que queremos que es un cat del `/proc/cpuinfo` :

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

He listado con `ls -lag` para ver que el archivo .e tenía tamaño 0 por lo que no había ocurrido ningún error.

Luego desde la pestaña del sftp he hecho un `get STDIN.o3863` para enviarlo a mi ordenador.

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: Tiene 4 cores físicos(cpu cores) y 4 cores lógicos(siblings). Para verlo he usado `cat /proc/cpuinfo`.

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

RESPUESTA: Mi PC tiene 2 cores físicos y 4 cores lógicos. Para verlo he usado `cat /proc/cpuinfo`.

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

RESPUESTA: Un nodo del atcgrid tiene 12 cores físicos y 24 cores lógicos.

5. En el Listado 1 se puede ver un código fuente C que calcula la suma de dos vectores y en el Listado 2 una versión con C++:

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

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 información devuelve exactamente la función `clock_gettime()`? ¿en qué estructura de datos devuelve `clock_gettime()` la información?

RESPUESTA: La variable `ncgt` contiene la diferencia del tiempo final (`cgt2`) menos el tiempo inicial (`cgt1`) más una corrección de error en nanosegundos.

La función `clock_gettime()` devuelve el tiempo de reloj que hay en ese instante.

La devuelve en una estructura `timespec` y guarda el valor en una variable de ese tipo que se le pasa como segundo argumento.

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: La mayoría del código es bastante parecido, en C para el vector dinámico se declaran punteros y se necesita el tamaño en bytes para reservarles el espacio, mientras que en C++ se reserva memoria con un `new`.

En C es `PRINTF_ALL`, mientras que en C++ es `COUT_ALL`.

El resto es bastante parecido.

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:

```
Tiempo(seg.):0.000000184 / Tamaño Vectores:2 /
V1[0]+V2[0]=V3[0](0.200000+0.200000=0.400000) /
V1[1]+V2[1]=V3[1](0.300000+0.100000=0.400000) /
```

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: En estos casos es cuando se produce error:

```
/var/lib/torque/mom_priv/jobs/4067.atcgrid.SC: line 20: 9519
Segmentation fault (core dumped) $PBS_O_WORKDIR/sumaVectores $N
/var/lib/torque/mom_priv/jobs/4067.atcgrid.SC: line 20: 9521
Segmentation fault (core dumped) $PBS_O_WORKDIR/sumaVectores $N
```

```

/var/lib/torque/mom_priv/jobs/4067.atcgrid.SC: line 20: 9524
Segmentation fault (core dumped) $PBS_O_WORKDIR/sumaVectores $N
/var/lib/torque/mom_priv/jobs/4067.atcgrid.SC: line 20: 9528
Segmentation fault (core dumped) $PBS_O_WORKDIR/sumaVectores $N
/var/lib/torque/mom_priv/jobs/4067.atcgrid.SC: line 20: 9530
Segmentation fault (core dumped) $PBS_O_WORKDIR/sumaVectores $N
/var/lib/torque/mom_priv/jobs/4067.atcgrid.SC: line 20: 9532
Segmentation fault (core dumped) $PBS_O_WORKDIR/sumaVectores $N
/var/lib/torque/mom_priv/jobs/4067.atcgrid.SC: line 20: 9534
Segmentation fault (core dumped) $PBS_O_WORKDIR/sumaVectores $N
/var/lib/torque/mom_priv/jobs/4067.atcgrid.SC: line 20: 9536
Segmentation fault (core dumped) $PBS_O_WORKDIR/sumaVectores $N

```

Se produce una violación de segmento al llegar a esos tamaños.

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: En el vector global ocurre un error y es que se repite dos veces un tamaño y esto es debido a que hemos definido ese tamaño como el MAX por lo que no se puede superar.

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?

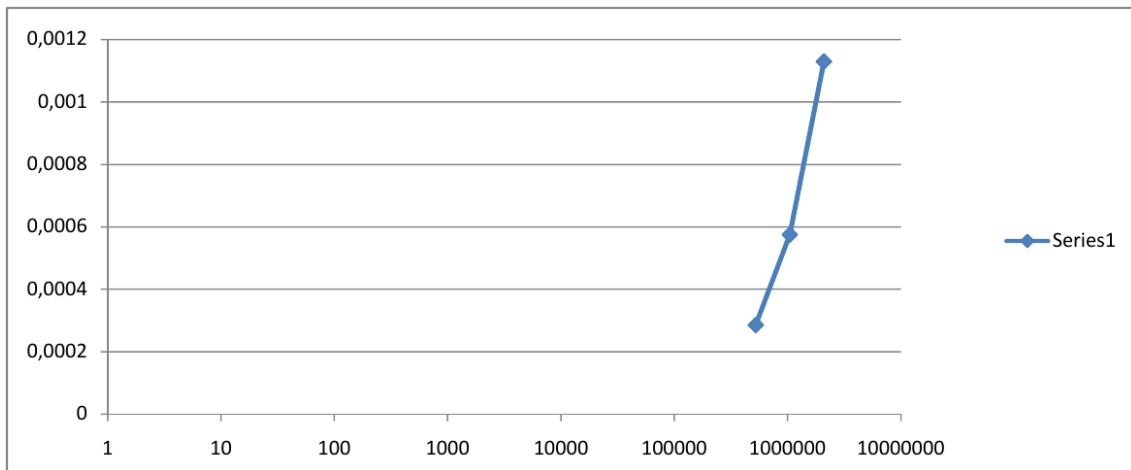
RESPUESTA:

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

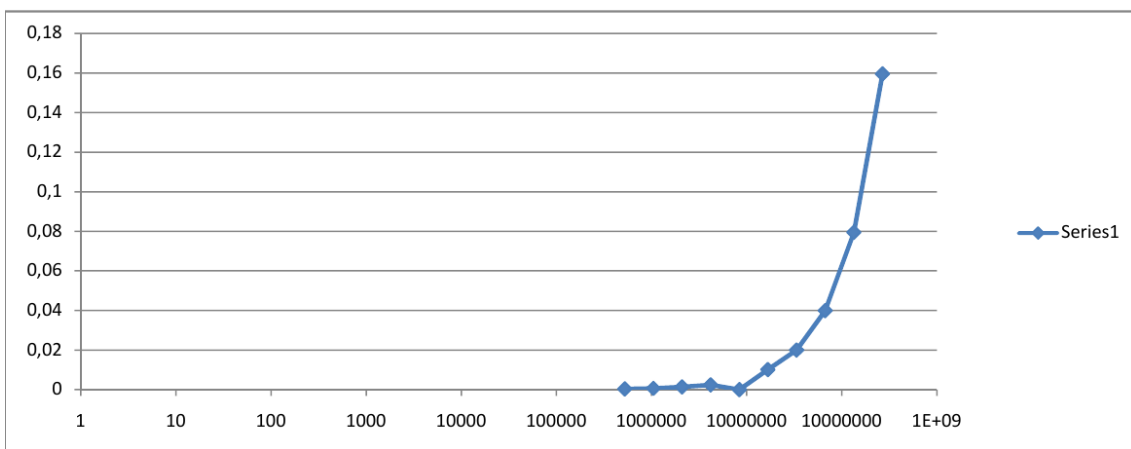
Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000285792	0.000385328	0.000347780
131072	1048576	0.000574947	0.000681580	0.000782722
262144	2097152	0.001129533	0.001389951	0.001239898
524288	4194304		0.002414614	0.002338073
1048576	8388608		0.005061032	0.005080451
2097152	16777216		0.010143238	0.010174109
4194304	33554432		0.019999342	0.020376866
8388608	67108864		0.039968389	0.041950661
16777216	134217728		0.079520224	0.081222458
33554432	268435456		0.159573642	0.160823121
67108864	536870912			0.322761087

Tabla 1 . Tiempos de ejecución de la suma de vectores para vectores locales, globales y dinámicos (PC Local)

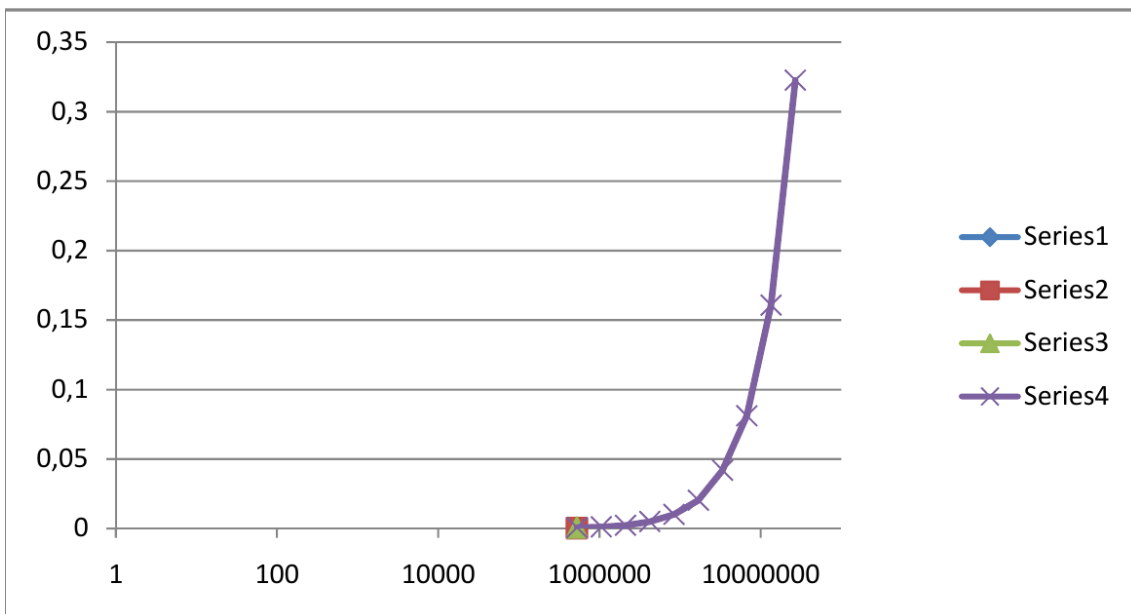
Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000205346	0.000298173	0.000207681
131072	1048576	0.000505236	0.000713875	0.000494487
262144	2097152	0.001359641	0.000989571	0.0012894
524288	4194304		0.002038076	0.00185788
1048576	8388608		0.004263208	0.0042459
2097152	16777216		0.007651413	0.00739188
4194304	33554432		0.014750745	0.0151957
8388608	67108864		0.030021554	0.0289917
16777216	134217728		0.057739504	0.0575899
33554432	268435456		0.117736564	0.120375
67108864	536870912			0.225229



Vector LOCAL



Vector GLOBAL



Vector DYNAMIC

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:

Ocurre un *relocation truncated to fit*. El problema es que el tamaño es demasiado grande, EL problema se da en el enlazado, ya que lo que se intenta es demasiado para un movimiento de 32 bits.