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): Antonio Miguel Morillo Chica
Grupo de prácticas: D1
Fecha de entrega: 09/03/2016
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 en el seminario. Conteste a las siguientes preguntas:

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

RESPUESTA: Para indicar la cola de trabajo por donde se va a ejecutar nuestro programa.

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

RESPUESTA: Usando el comando qstat que muestra los trabajos que se están ejecutando pudiendo así ver han terminado.

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

RESPUESTA: Viendo los archivos ".e" que contienen los errores.

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

RESPUESTA: Ve tanto mensajes hello world como hebras tenga su computadora.

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

RESPUESTA: 2x3x2x2, dos núcleos, con tres cores cada uno, en total seis cores por dos hebras son 24 threads, 24 hebras.

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: Porque ya se encuentra en la cola de ejecución de ac. Ya se lo indicamos en el script.

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

RESPUESTA: Lo ejecuta 4 veces. Porque se ejecuta dentro de un bucle.

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

RESPUESTA: Serían: 12 en la primera, 6 en la segunda, 3 en la tercera y 1 en la última. Ejecuta una vez por hebra en cada iteracción es p=p/2 cuyo resultado es 12, 6, 3, 1.

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: Muestra el directorio directorio donde se ejecuta cada hebra.

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: Para obtener información de uno de los nodos he ejecutado echo 'cat /proc/cpinfo' |qsub -q ac, a continuación he mostrado por la terminas el archivo STDIN.o* y me ha mostrado la información de todos los nodos.

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: Diría 4 lógicos pero no se interpretar muy bien todo el cpuinfo.

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

RESPUESTA: Tiene un core fisico y otro lógico.

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 almacena la diferencia de tiempo entre el instante anterior a hacer el bucle y el instante postumo a este. Clock_gettime() toma la hora exacta del sistema en ese punto de ejecución y lo guarda en la variable del segundo argumento. Lo devuelve en una estructura de tipo timespec que contiene el moento EXACTO de un 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++
Reserva de memoria	Debemos usar malloc(sizeof(struct Mi estructura) para comprabar si hay espacio o no, por ello comprovamos si ha devuelto null en los tres vectores.	No hace falta controlar nada ya que new informa con una excepción si no hay tamaño suficiente.
Control de asignación de memoria	Se usa un if comprobando si alguno devuelve null, lo que indicaría que malloc no ha tenido suficiente espacio.	En c++ saltaría una excepción

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:

williams@ei141072:~/Documentos/AC$ cat STDIN.o30116

Tiempo(seg.):0.000000125 / 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: Sí, se obtienen errores de segmentación debido a que se desbordan los vectores concretamente a partir del tamaño 104857.60000.

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?

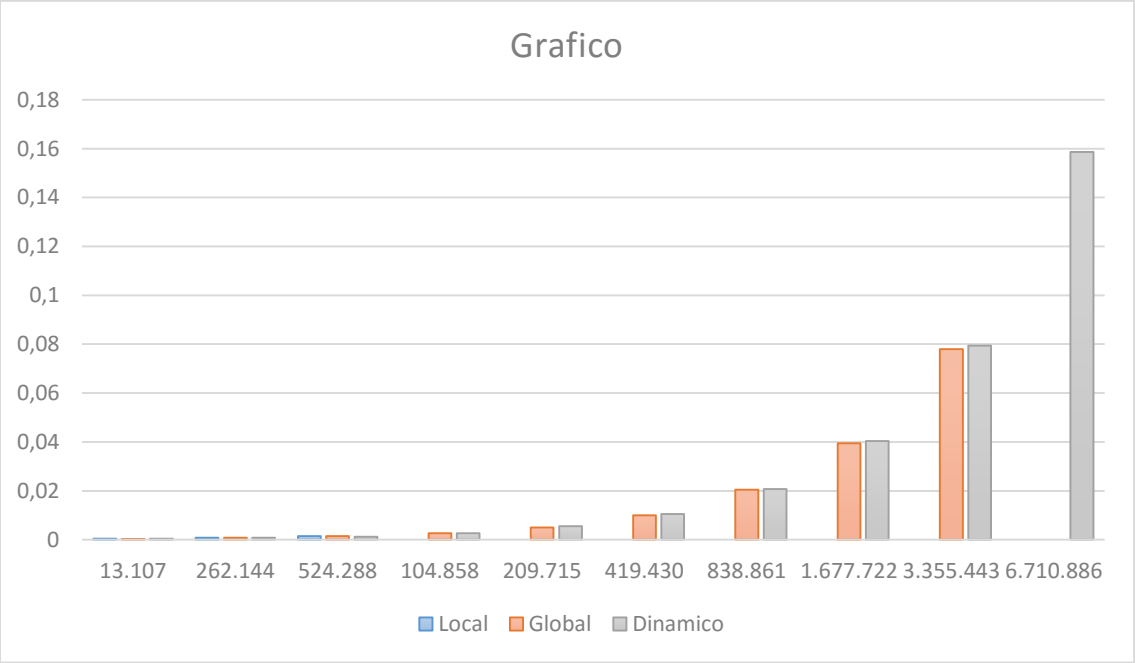
RESPUESTA: Para los vectores globales y dinámicos no se producen errores, en el local sí. Es debido a que los dinámicos usan automáticamente el tamaño que necesiten reutilizando espacio y los globales no se desborda porque no depende del tamaño de la pila de ejecución.

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: Sí que hay diferencias en los tiempos, concretamente el orden de velocidad de más a menos rápido es: dinámicos, globales y locales.

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

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	13107.20000	0.000388485	0.000336341	0.000405941
131072	26214.40000	0.000761913	0.000764915	0.000787699
262144	52428.80000	0.001529368	0.001423673	0.001219384
524288	104857.60000	-	0.002669764	0.002582073
1048576	209715.20000	-	0.005059318	0.005504403
2097152	419430.40000	-	0.009983525	0.010475343
4194304	838860.80000	-	0.020479669	0.020762251
8388608	1677721.60000	-	0.039404485	0.040414198
16777216	3355443.20000	-	0.077932405	0.079454483
33554432	6710886.40000	-	0.155612599	0.158700160
67108864	13421772.80000	-		0.318655670



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: Lo que ocurre es que se produce un error de violación de segmento debido a que N puede almacenar como máximo 4294967295 porque es un unsigned int y ese es el máximo tamaño que puede tener.