2º curso / 2º cuatr.

Grado Ingeniería Informática

Arquitectura de Computadores (AC)

Cuaderno de prácticas.
Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos): Valentino Lugli
Grupo de prácticas y profesor de prácticas: C1, Christian Morillas
Fecha de entrega: 15/03/2021
Fecha evaluación en clase: 16/03/2021

Antes de comenzar a realizar el trabajo de este cuaderno consultar el fichero con los normas de prácticas que se encuentra en SWAD

Parte I. Ejercicios basados en los ejemplos del seminario práctico

Crear el directorio con nombre `bp0` en atcgrid y en el PC (PC = PC del aula de prácticas o su computador personal).

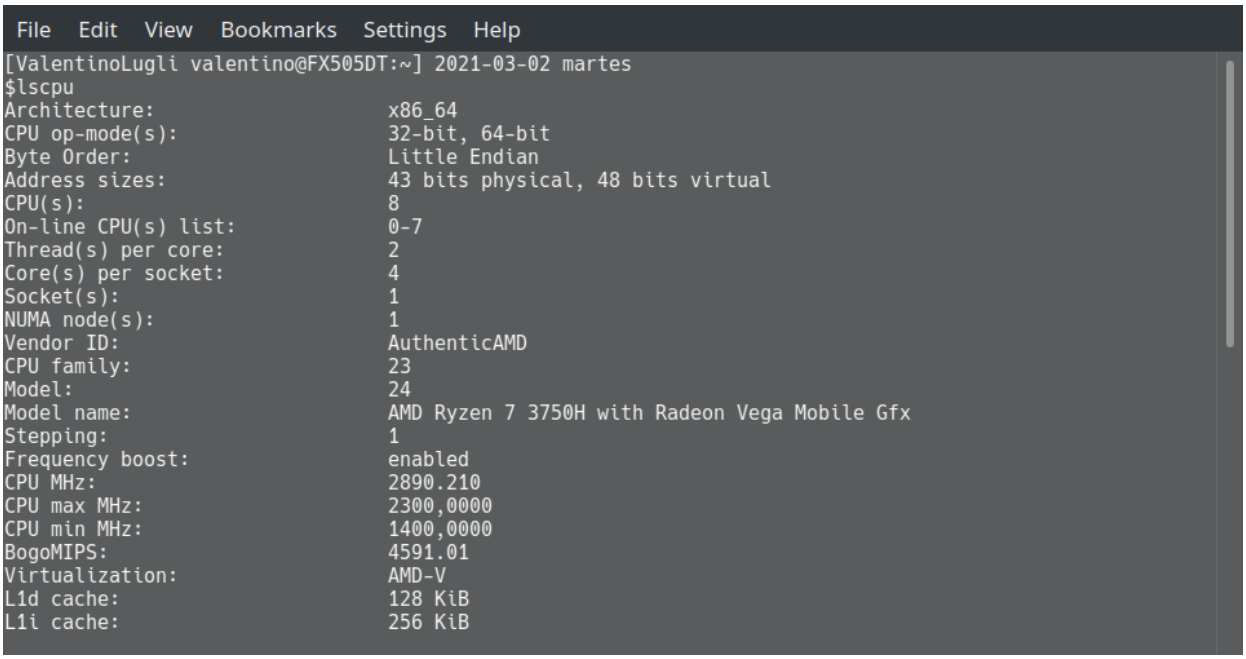
NOTA: En las prácticas se usa slurm como gestor de colas. Consideraciones a tener en cuenta:

- Slurm está configurado para asignar recursos a los procesos (llamados *tasks* en slurm) a nivel de core físico. Esto significa que por defecto slurm asigna un core a un proceso, para asignar x se debe usar con `sbatch/srun` la opción `--cpus-per-task=x (-cx)`.
- En slurm, por defecto, `cpu` se refiere a cores lógicos (ej. en la opción `-c`), si no se quieren usar cores lógicos hay que añadir la opción `--hint=nomultithread` a `sbatch/srun`.
- Para asegurar que solo se crea un proceso hay que incluir `--ntasks=1 (-n1)` en `sbatch/srun`.
- Para que no se ejecute más de un proceso en un nodo de cómputo de atcgrid hay que usar `--exclusive` con `sbatch/srun` (se recomienda no utilizarlo en los `srun` dentro de un script).
- Los `srun` dentro de un *script* heredan las opciones fijadas en el `sbatch` que se usa para enviar el script a la cola (partición slurm).
- Las opciones de `sbatch` se pueden especificar también dentro del *script* (usando #SBATCH, ver ejemplos en el script del seminario)

1. Ejecutar `lscpu` en el PC, en atcgrid4 (usar `-p ac4`) y en uno de los restantes nodos de cómputo (atcgrid1, atcgrid2 o atcgrid3, están en la cola ac). (Crear directorio `ejer1`)

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

RESPUESTA: En el PC



En la cola `ac`

```
File  Edit  View  Bookmarks  Settings  Help
[ValentinoLugli c1estudiante16@atcgrid:~] 2021-03-04 jueves
>>srun -p ac -A ac lscpu
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:              4799.93
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              12288K
```

En la cola `ac4`

```
File  Edit  View  Bookmarks  Settings  Help
[ValentinoLugli c1estudiante16@atcgrid:~] 2021-03-04 jueves
>>srun -p ac4 -A ac lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                64
On-line CPU(s) list:   0-63
Thread(s) per core:    2
Core(s) per socket:    16
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 85
Model name:            Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz
Stepping:              7
CPU MHz:               1192.785
CPU max MHz:           3200,0000
CPU min MHz:           800,0000
BogoMIPS:              4200.00
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              1024K
L3 cache:              22528K
```

(b) ¿Cuántos cores físicos y cuántos cores lógicos tiene atcgrid4?, ¿cuántos tienen atcgrid1, atcgrid2 y atcgrid3? y ¿cuántos tiene el PC? Razonar las respuestas

RESPUESTA:

- **atcgrid4** posee 32 cores físicos y 64 cores lógicos. Esto se debe que se tiene “Sockets(s): 2” y “Core(s) per socket: 16”, por lo tanto el servidor posee 2 procesadores y cada uno posee 16 cores físicos, en total 32 cores físicos. Cada core físico puede ejecutar 2 hilos; “Theads(s) per core: 2” por lo tanto en total se tienen 64 cores lógicos.
- **atcgrid1, 2 y 3** poseen 12 cores físicos (“Socket(s): 2” y “Core(s) per socket: 6”) y 24 cores lógicos (Nuevamente se tiene que “Theads(s) per core: 2”)
- **Mi PC** tiene 4 cores físicos (“Socket(s): 1” y “Cores(s) per socket: 4”) y 8 cores lógicos ya que al igual que los anteriores soporta 2 hilos por cada core.

2. Compilar y ejecutar en el PC el código `HelloOMP.c` del seminario (recordar que, como se indica en las normas de prácticas, 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`).

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

RESPUESTA:

```
File  Edit  View  Bookmarks  Settings  Help
[ValentinoLugli valentino@FX505DT:~/Documents/Git/ac2021/bp0/ejer2] 2021-03-04 jueves
>>gcc -O2 -fopenmp -o HelloOMP HelloOMP.c
[ValentinoLugli valentino@FX505DT:~/Documents/Git/ac2021/bp0/ejer2] 2021-03-04 jueves
>>./HelloOMP
(2):!!!Hello World!!!
(5):!!!Hello World!!!
(6):!!!Hello World!!!
(1):!!!Hello World!!!
(3):!!!Hello World!!!
(4):!!!Hello World!!!
(7):!!!Hello World!!!
(0):!!!Hello World!!!
[ValentinoLugli valentino@FX505DT:~/Documents/Git/ac2021/bp0/ejer2] 2021-03-04 jueves
>>
```

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

RESPUESTA:

Aparecen 8 mensajes debido a que se han generado 8 hilos diferentes, lo cual concuerda con la cantidad de cores lógicos que posee mi PC, en `lscpu` aparece como "CPU(s): 8".

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 este código en un nodo de cómputo de atcgrid (de 1 a 3) a través de cola ac del gestor de colas utilizando directamente en línea de comandos (no use ningún *script*):

(a) `srun --partition=ac --account=ac --ntasks=1 --cpus-per-task=12 --hint=nomultithread HelloOMP`

(Alternativa: `srun -pac -Aac -n1 -c12 --hint=nomultithread HelloOMP`)

Adjuntar capturas de pantalla que muestren el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas.

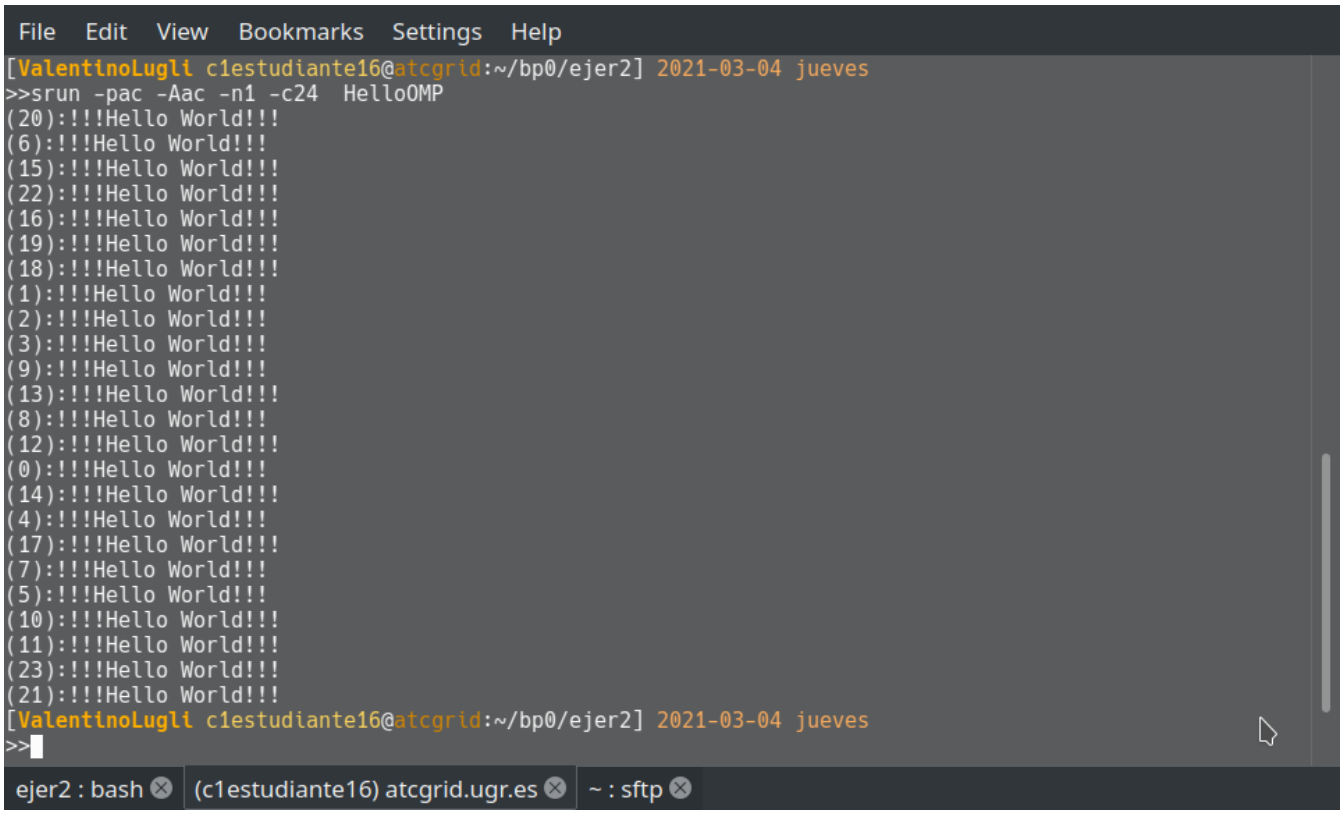
RESPUESTA:

```
File  Edit  View  Bookmarks  Settings  Help
[ValentinoLugli c1estudiante16@atcgrid:~/bp0/ejer2] 2021-03-04 jueves
>>srun --partition=ac --account=ac --ntasks=1 --cpus-per-task=12 --hint=nomultithread HelloOMP
(0):!!!Hello World!!!
(3):!!!Hello World!!!
(6):!!!Hello World!!!
(7):!!!Hello World!!!
(5):!!!Hello World!!!
(10):!!!Hello World!!!
(4):!!!Hello World!!!
(8):!!!Hello World!!!
(1):!!!Hello World!!!
(11):!!!Hello World!!!
(2):!!!Hello World!!!
(9):!!!Hello World!!!
[ValentinoLugli c1estudiante16@atcgrid:~/bp0/ejer2] 2021-03-04 jueves
>>
```

(b) `srun -pac –Aac -n1 -c24 HelloOMP`

Adjuntar capturas de pantalla que muestren el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas.

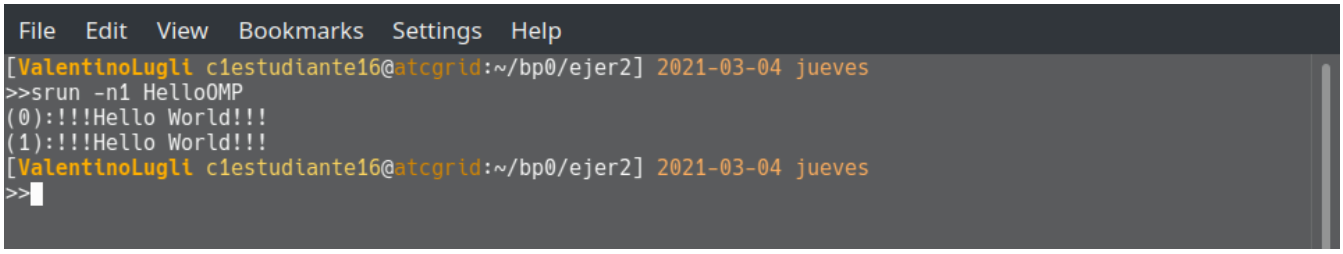
RESPUESTA:



(c) `srun -n1 HelloOMP`

Adjuntar capturas de pantalla que muestren el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas. ¿Qué partición se está usando?

RESPUESTA:



Se está utilizando la partición o cola `ac` ya que al llamar a srun sin el parámetro `-p`, se utilizará por defecto la cola `ac`.

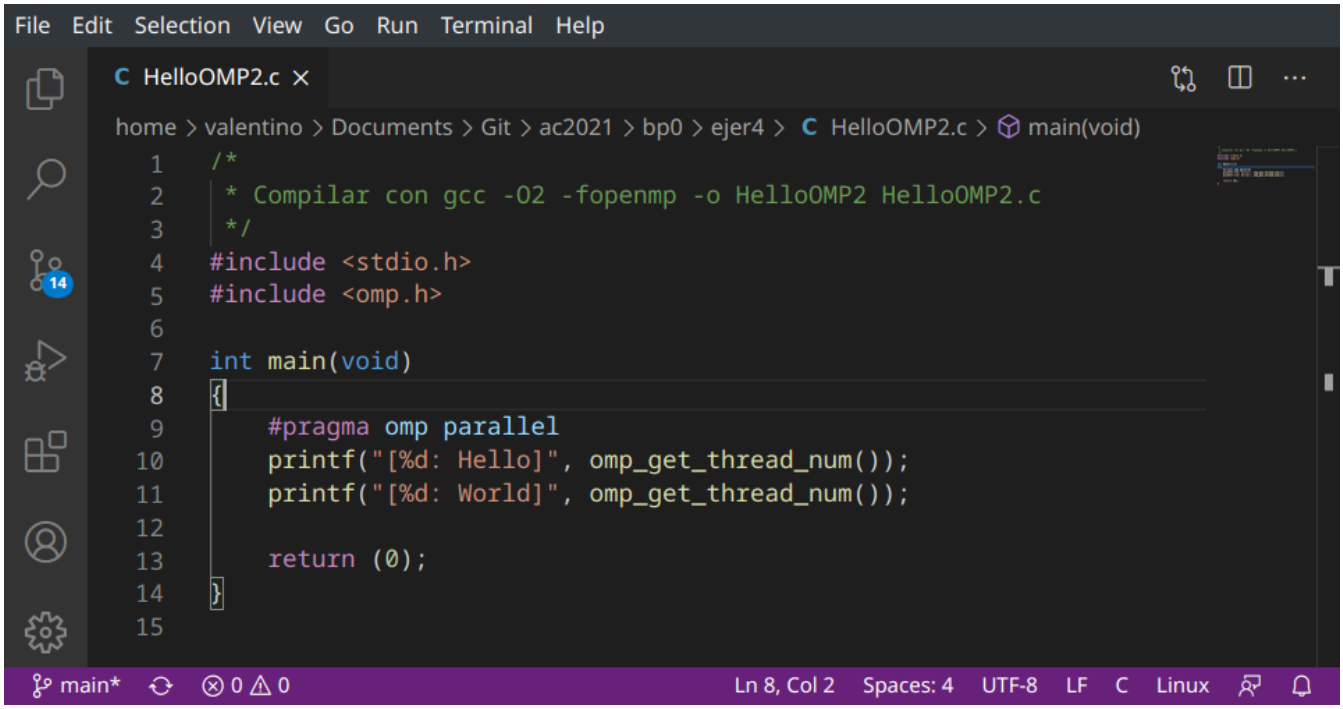
(d) ¿Qué orden `srun` usaría para que HelloOMP utilice todos los cores físicos de atcgrid4 (se debe imprimir un único mensaje desde cada uno de ellos)?

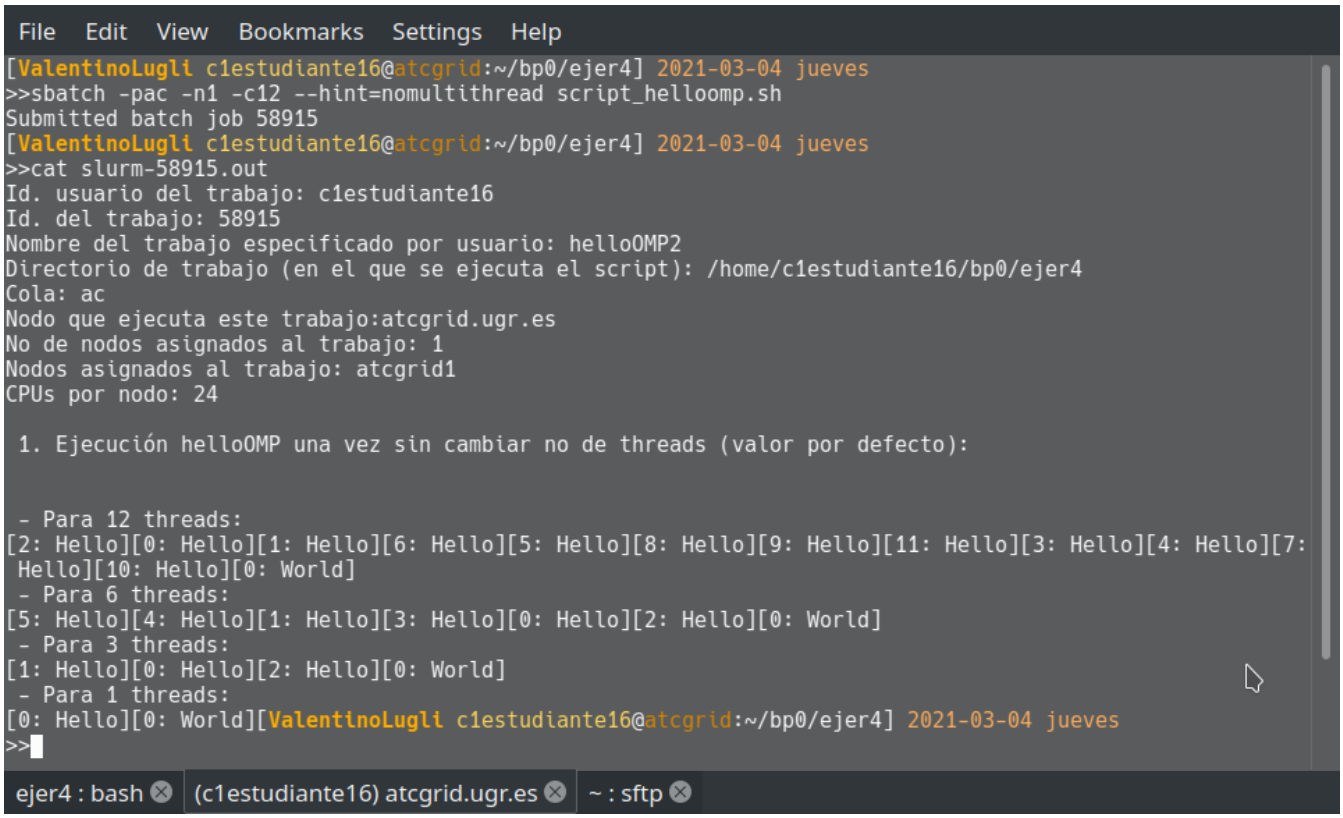
```
srun --partition=ac4 --account=ac --ntasks=1 --cpus-per-task=32 –
hint=nomultithread --exclusive HelloOMP
```

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 al 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) Utilizar: `sbatch -pac –n1 -c12 --hint=nomultithread script_helloomp.sh`. Adjuntar capturas de pantalla que muestren el nuevo código, la compilación, el envío a la cola de la ejecución y el resultado de esta ejecución tal y como la devuelve el gestor de colas.

RESPUESTA:





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

RESPUESTA:

El nodo que ha ejecutado el trabajo ha sido atcgrid1. El script posee la variable de entorno `SLURM_JOB_NODELIST` el cual indica la lista de los nodos que han sido asignados al trabajo, según la documentación de Slurm.

NOTA: Utilizar siempre con `sbatch` las opciones `–n1` y `-c`, `--exclusive` y, para usar cores físicos y no lógicos, no olvide incluir `--hint=nomultithread`. Utilizar siempre con `srun`, si lo usa fuera de un script, las opciones `–n1` y `-c` y, para usar cores físicos y no lógicos, no olvide incluir `--hint=nomultithread`. Recordar que los `srun` dentro de un *script* heredan las opciones incluidas en el `sbatch` que se usa para enviar el *script* a la cola slurm. Se recomienda usar `sbatch` en lugar de `srun` para enviar trabajos a ejecutar a través slurm porque éste último deja bloqueada la ventana hasta que termina la ejecución, mientras que usando `sbatch` la ejecución se realiza en segundo plano.

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).

RESPUESTA:

```
File  Edit  View  Bookmarks  Settings  Help
[ValentinoLugli valentino@FX505DT:~/Desktop/Git/ac2021/bp0] 2021-03-06 sábado
>>gcc  -O2  SumaVectores.c -o SumaVectores -lrt
[ValentinoLugli valentino@FX505DT:~/Desktop/Git/ac2021/bp0] 2021-03-06 sábado
>>./SumaVectores 8
Tiempo:0.000000221       / Tamaño Vectores:8
/ V1[0]+V2[0]=V3[0](0.800000+0.800000=1.600000) /
/ V1[1]+V2[1]=V3[1](0.900000+0.700000=1.600000) /
/ V1[2]+V2[2]=V3[2](1.000000+0.600000=1.600000) /
/ V1[3]+V2[3]=V3[3](1.100000+0.500000=1.600000) /
/ V1[4]+V2[4]=V3[4](1.200000+0.400000=1.600000) /
/ V1[5]+V2[5]=V3[5](1.300000+0.300000=1.600000) /
/ V1[6]+V2[6]=V3[6](1.400000+0.200000=1.600000) /
/ V1[7]+V2[7]=V3[7](1.500000+0.100000=1.600000) /
[ValentinoLugli valentino@FX505DT:~/Desktop/Git/ac2021/bp0] 2021-03-06 sábado
>>
```

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?
RESPUESTA:
La variable contiene la diferencia en segundos del tiempo que ha tomado la suma de vectores; se compone de la diferencia de los tiempos `cgt1` y `cgt2`.

(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)?

RESPUESTA:

`clock_gettime()` devuelve la estructura de datos `struct timespec` la cual internamente se compone de dos atributos: `tv_sec` de tipo `time_t` y `tv_nsec` de tipo `long int`.

(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?

RESPUESTA:

`timespec` contiene el tiempo actual; el miembro `tv_sec` contiene el número de segundos en tiempo Unix, es decir, los segundos transcurridos desde la medianoche UTC del 1 de enero de 1970 hasta el instante que se llama la función y `tv_nsec` contiene el número de nanosegundos que han transcurrido dentro del último segundo desde la llamada a la función.

7. Rellenar una tabla como la Tabla 1 en una hoja de cálculo con los tiempos de ejecución del código del Listado 1 para vectores locales, globales y dinámicos (se pueden obtener errores en tiempo de ejecución o de compilación, ver ejercicio 9). Obtener estos resultados usando *scripts* (partir del *script* que hay en el seminario). Debe haber una tabla para un nodo de cómputo de atcgrid con procesador Intel Xeon E5645 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. (NOTA: Se recomienda usar en la hoja de cálculo el mismo separador para decimales que usan los códigos al imprimir –"."–. Este separador se puede modificar en la hoja de cálculo.)

RESPUESTA:

Mi PC

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000081414	0.000180841	0.000286451
131072	1048576	0.000211189	0.000499983	0.000436063
262144	2097152	0.000424802	0.000993766	0.000983607
524288	4194304	Segmentation fault	0.002234037	0.001721009
1048576	8388608	Segmentation fault	0.004666239	0.003723699
2097152	16777216	Segmentation fault	0.008968911	0.006570914
4194304	33554432	Segmentation fault	0.017548918	0.014336563
8388608	67108864	Segmentation fault	0.032473460	0.024836793
16777216	134217728	Segmentation fault	0.060135758	0.060334488
33554432	268435456	Segmentation fault	0.113444304	0.100444459
67108864	536870912	Segmentation fault	0.124786843	0.200325196

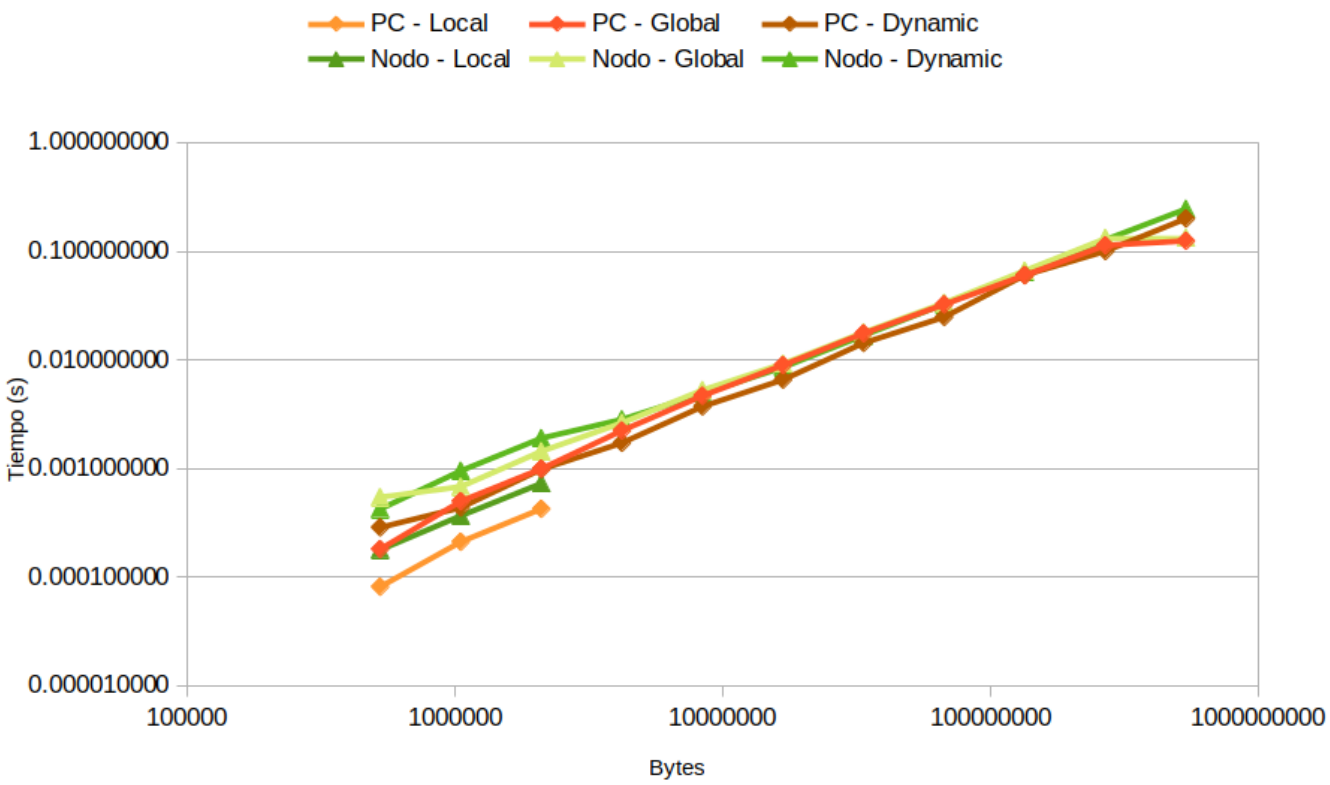
Nodo de cómputo ATCGRID

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000178204	0.000544249	0.000422847
131072	1048576	0.000367023	0.000680420	0.000952335
262144	2097152	0.000729080	0.001438869	0.001900344
524288	4194304	Segmentation fault	0.002632678	0.002828415
1048576	8388608	Segmentation fault	0.005266595	0.004889878
2097152	16777216	Segmentation fault	0.009236259	0.008556055
4194304	33554432	Segmentation fault	0.017935467	0.016817041
8388608	67108864	Segmentation fault	0.033478220	0.032767856
16777216	134217728	Segmentation fault	0.066317673	0.064050424
33554432	268435456	Segmentation fault	0.132645390	0.127837521
67108864	536870912	Segmentation fault	0.133217572	0.245578340

1. 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). ¿Hay diferencias en los tiempos de ejecución?

RESPUESTA:

Si, existe una clara distinción en los tiempos de ejecución tanto para el nodo de atcgrid como para mi PC respecto a ejecutar los vectores de manera local: es mucho más rápido que las ejecuciones para vectores globales o dinámicos; de estos últimos también se puede notar que los vectores globales son ligeramente más rápidos que los dinámicos. También, en general, el nodo de atcgrid tiene tiempos de ejecución ligeramente más altos de los que posee mi PC.



2. Contestar a las siguientes preguntas:

(a) Cuando se usan vectores locales, ¿se obtiene error para alguno de los tamaños?, ¿a qué cree que es debido lo que ocurre? (Incorporar volcados de pantalla como se indica en las normas de prácticas)

RESPUESTA:

Si, luego de 262144 componentes ocurre una violación de segmento. Esto ocurre porque los vectores con más componentes utilizan más espacio que el disponible en la pila del proceso.

```
File  Edit  View  Bookmarks  Settings  Help
[ValentinoLugli valentino@FX505DT:~/Desktop/Git/ac2021/bp0/ejer7] 2021-03-09 martes
>>bash scriptLocal_SumaVectores.sh
---LOCAL---
N: 65536 Tamaño vector: 524288 Tiempo: 0.000085431
N: 131072 Tamaño vector: 1048576 Tiempo: 0.000255632
N: 262144 Tamaño vector: 2097152 Tiempo: 0.000502719
scriptLocal_SumaVectores.sh: line 6:  7481 Segmentation fault      (core dumped) ./SumaVectores_LOCAL $(
(65536*(2**$i)))
scriptLocal_SumaVectores.sh: line 6:  7483 Segmentation fault      (core dumped) ./SumaVectores_LOCAL $(
(65536*(2**$i)))
scriptLocal_SumaVectores.sh: line 6:  7485 Segmentation fault      (core dumped) ./SumaVectores_LOCAL $(
(65536*(2**$i)))
scriptLocal_SumaVectores.sh: line 6:  7487 Segmentation fault      (core dumped) ./SumaVectores_LOCAL $(
(65536*(2**$i)))
scriptLocal_SumaVectores.sh: line 6:  7489 Segmentation fault      (core dumped) ./SumaVectores_LOCAL $(
(65536*(2**$i)))
scriptLocal_SumaVectores.sh: line 6:  7491 Segmentation fault      (core dumped) ./SumaVectores_LOCAL $(
(65536*(2**$i)))
scriptLocal_SumaVectores.sh: line 6:  7493 Segmentation fault      (core dumped) ./SumaVectores_LOCAL $(
(65536*(2**$i)))
scriptLocal_SumaVectores.sh: line 6:  7495 Segmentation fault      (core dumped) ./SumaVectores_LOCAL $(
(65536*(2**$i)))
---GLOBAL---
N: 65536 Tamaño vector: 524288 Tiempo: 0.000249541
N: 131072 Tamaño vector: 1048576 Tiempo: 0.000608138
```

(b) Cuando se usan vectores globales, ¿se obtiene error para alguno de los tamaños?, ¿a qué cree que es debido lo que ocurre? (Incorporar volcados de pantalla como se indica en las normas de prácticas)

RESPUESTA:

No, pero las últimas dos ejecuciones son la misma ya que se tiene una limitación del tamaño del vector N, esto se encuentra definido en el código como el tamaño máximo para los vectores globales.

```
File  Edit  View  Bookmarks  Settings  Help
[ValentinoLugli valentino@FX505DT:~/Desktop/Git/ac2021/bp0/ejer7] 2021-03-09 martes
>>./SumaVectores_GLOBAL 33554432
N: 33554432 Tamaño vector: 268435456 Tiempo: 0.115944844
[ValentinoLugli valentino@FX505DT:~/Desktop/Git/ac2021/bp0/ejer7] 2021-03-09 martes
>>./SumaVectores_GLOBAL 67108864
N: 33554432 Tamaño vector: 268435456 Tiempo: 0.129828450
[ValentinoLugli valentino@FX505DT:~/Desktop/Git/ac2021/bp0/ejer7] 2021-03-09 martes
>>
```

(c) Cuando se usan vectores dinámicos, ¿se obtiene error para alguno de los tamaños?, ¿a qué cree que es debido lo que ocurre? (Incorporar volcados de pantalla como se indica en las normas de prácticas)

RESPUESTA:

No, no se obtienen errores debido a que los valores se almacenan en el heap y esta no posee un límite como tal de tamaño; solo está limitada por la cantidad de memoria RAM que posea el sistema.

```
File  Edit  View  Bookmarks  Settings  Help
N: 65536 Tamaño vector: 524288 Tiempo: 0.000285597
N: 131072 Tamaño vector: 1048576 Tiempo: 0.000547670
N: 262144 Tamaño vector: 2097152 Tiempo: 0.001101171
N: 524288 Tamaño vector: 4194304 Tiempo: 0.002025258
N: 1048576 Tamaño vector: 8388608 Tiempo: 0.004448745
N: 2097152 Tamaño vector: 16777216 Tiempo: 0.009254821
N: 4194304 Tamaño vector: 33554432 Tiempo: 0.017112346
N: 8388608 Tamaño vector: 67108864 Tiempo: 0.031889783
N: 16777216 Tamaño vector: 134217728 Tiempo: 0.062433363
N: 33554432 Tamaño vector: 268435456 Tiempo: 0.124555931
N: 33554432 Tamaño vector: 268435456 Tiempo: 0.124708648
---DYNAMIC---
N: 65536 Tamaño vector: 524288 Tiempo: 0.000217208
N: 131072 Tamaño vector: 1048576 Tiempo: 0.000504628
N: 262144 Tamaño vector: 2097152 Tiempo: 0.000891025
N: 524288 Tamaño vector: 4194304 Tiempo: 0.001599618
N: 1048576 Tamaño vector: 8388608 Tiempo: 0.003885135
N: 2097152 Tamaño vector: 16777216 Tiempo: 0.007244001
N: 4194304 Tamaño vector: 33554432 Tiempo: 0.013866042
N: 8388608 Tamaño vector: 67108864 Tiempo: 0.025948771
N: 16777216 Tamaño vector: 134217728 Tiempo: 0.058960012
N: 33554432 Tamaño vector: 268435456 Tiempo: 0.103899941
N: 67108864 Tamaño vector: 536870912 Tiempo: 0.202402291
[ValentinoLugli valentino@FX505DT:~/Desktop/Git/ac2021/bp0/ejer7] 2021-03-09 martes
>>
```

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

RESPUESTA:

N es de tipo unsigned int, por lo tanto tiene un valor máximo de $2^{32}-1 = 4294967295$; esto se debe a que ese tipo de dato tiene un tamaño de 4 bytes.

(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)

RESPUESTA:

Ocurre un error al intentar compilar, específicamente en el linker, debido a que en la compilación de manera Global, lo que se está haciendo es asignar en tiempo de compilación el espacio para almacenar los datos;

lo que sucede es que por defecto el compilador no permite que se generen programas que tengan objetos con direcciones más grandes de 32 bits, y dado que este número llega a ese valor genera un error.

```
File  Edit  View  Bookmarks  Settings  Help
[ValentinoLugli valentino@FX505DT:~/Documents/Git/ac2021/bp0/ejer7.3b] 2021-03-09 martes
>>gcc  -O2  SumaVectores_MaxModificado.c -o SumaVectores -lrt
/tmp/ccPHmpMe.o: in function `main':
SumaVectores_MaxModificado.c:(.text.startup+0x68): relocation truncated to fit: R_X86_64_PC32 against sy
mbol `v2' defined in COMMON section in /tmp/ccPHmpMe.o
SumaVectores_MaxModificado.c:(.text.startup+0xbb): relocation truncated to fit: R_X86_64_PC32 against sy
mbol `v3' defined in COMMON section in /tmp/ccPHmpMe.o
SumaVectores_MaxModificado.c:(.text.startup+0x205): relocation truncated to fit: R_X86_64_PC32 against s
ymbol `v2' defined in COMMON section in /tmp/ccPHmpMe.o
collect2: error: ld returned 1 exit status
[ValentinoLugli valentino@FX505DT:~/Documents/Git/ac2021/bp0/ejer7.3b] 2021-03-09 martes
>>
```

Entrega del trabajo

Leer lo indicado en las normas de prácticas sobre la entrega del trabajo del bloque práctico en SWAD.

Listado 1. Código C que suma dos vectores. Se generan aleatoriamente las componentes para vectores de tamaño mayor que 8 y se imprimen todas las componentes para vectores menores que 10.

```
/* 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(), rand(), srand(), 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
if (N < 9)
  for (i = 0; i < N; i++)
  {
    v1[i] = N * 0.1 + i * 0.1;
    v2[i] = N * 0.1 - i * 0.1;
  }
else
{
  srand(time(0));
  for (i = 0; i < N; i++)
  {
    v1[i] = rand()/ ((double) rand());
    v2[i] = rand()/ ((double) rand()); //printf("%d:%f,%f/",i,v1[i],v2[i]);
  }
}

   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:%lu\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;
}
```