

2º curso / 2º cuatr.

Grado en  
Ing. Informática

# Arquitectura de Computadores

## Seminario 0. Entorno de programación: atcgrid y gestor TORQUE

Material elaborado por los profesores responsables de la asignatura:

Mancia Anguita – Julio Ortega

*Licencia Creative Commons*



ugr

Universidad  
de Granada

ETSIIT

Escuela Técnica Superior  
de Ingenierías Informática  
y de Telecomunicación



ATC

Departamento de Arquitectura  
y Tecnología de Computadores  
UNIVERSIDAD DE GRANADA



# Contenidos

- Cluster de prácticas (atcgrid)
- Sistema de colas TORQUE
- Ejemplo hello OpenOMP en atcgrid usando TORQUE

# Contenidos

- Cluster de prácticas (atcgrid)
  - Componentes
  - Placa madre
  - Chip de procesamiento (procesador)
  - Acceso
- Sistema de colas TORQUE
- Ejemplo hello OpenOMP en atcgrid usando TORQUE

# Cluster de prácticas (atcgrid): componentes



**Switch: SMC8508T**

[http://www.smc.com/index.cfm?event=viewProduct&localeCode=EN\\_USA&cid=6&scid=24&pid=1147](http://www.smc.com/index.cfm?event=viewProduct&localeCode=EN_USA&cid=6&scid=24&pid=1147)



**Nodos de cómputo (dos): Servidores rack  
SuperMicro SuperServer 6016T-T**

<http://www.supermicro.com/products/system/1U/6016/SYS-6016T-T.cfm>

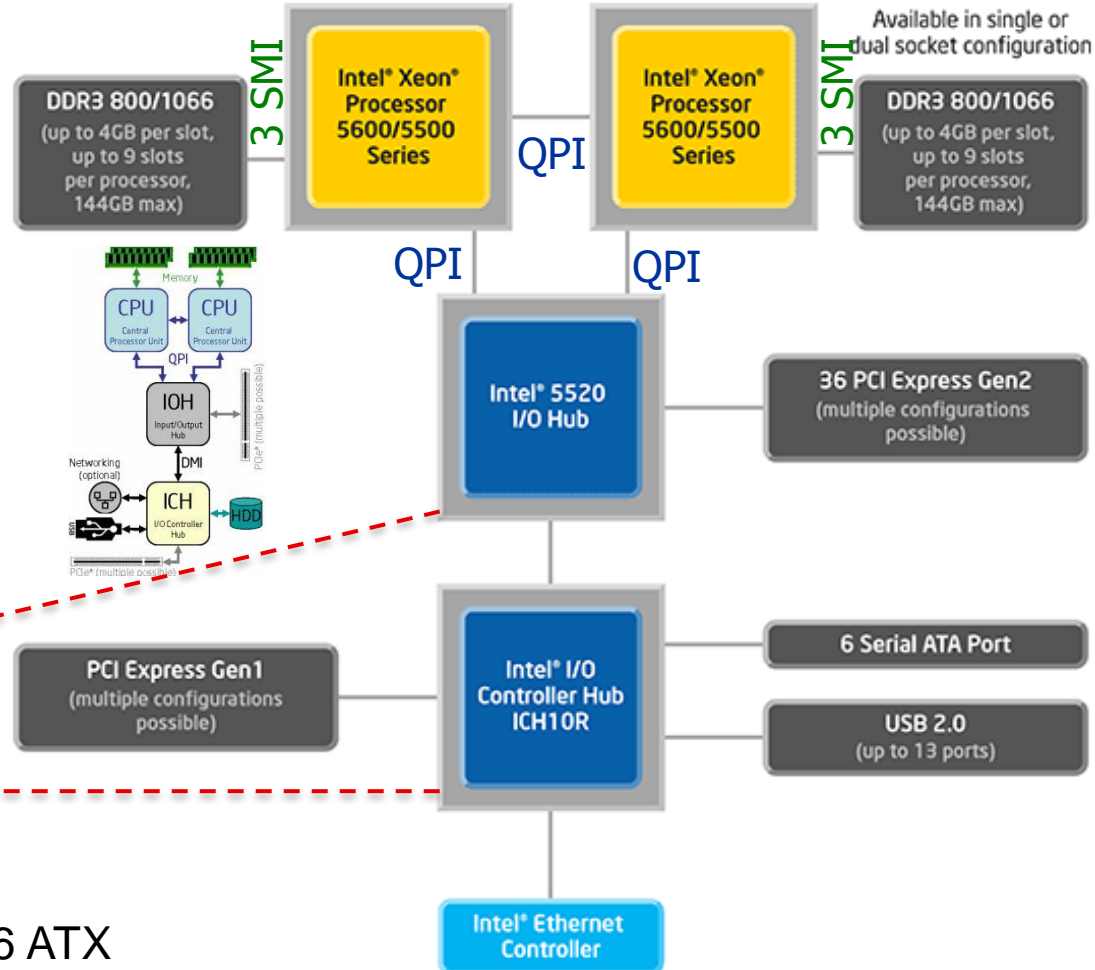
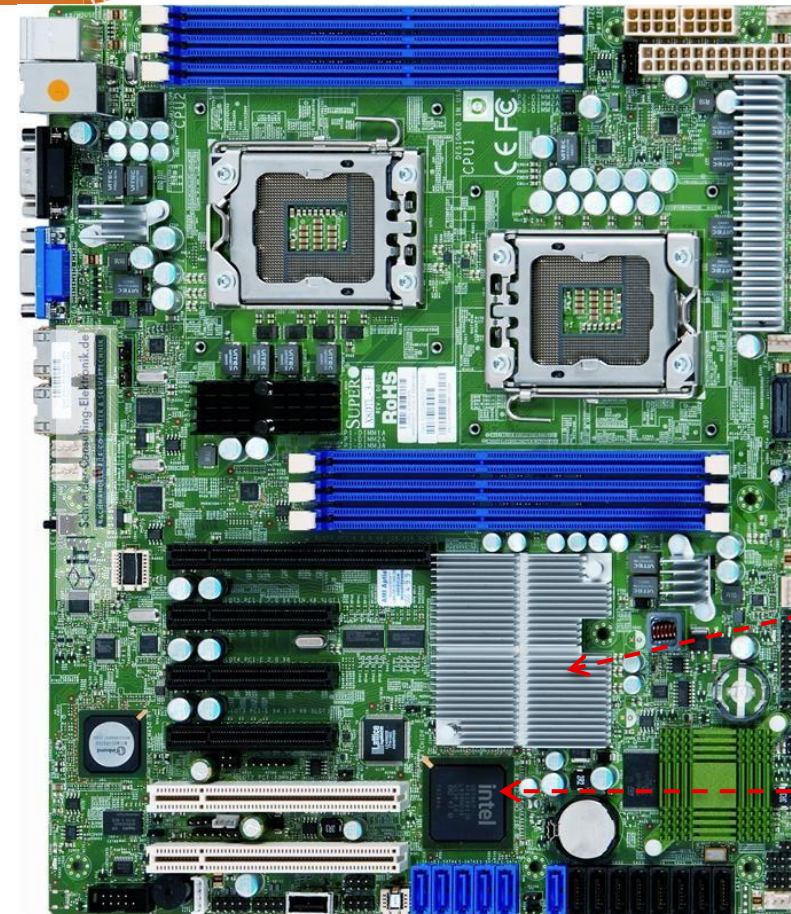


**Cables**



**Nodo front-end  
(host, master)**

# Cluster de prácticas (atcgrid): placa madre



Supermicro X8DTL-i Dual Socket 1366 ATX

Server Mainboard Intel 5500 chipset

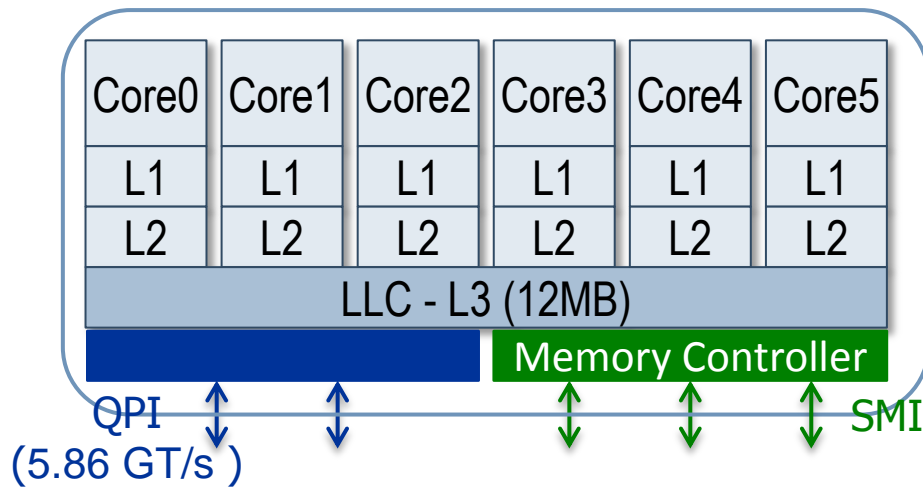
<http://www.supermicro.com/products/motherboard/QPI/5500/X8DTL-i.cfm>

Intel® 5520 Chipset

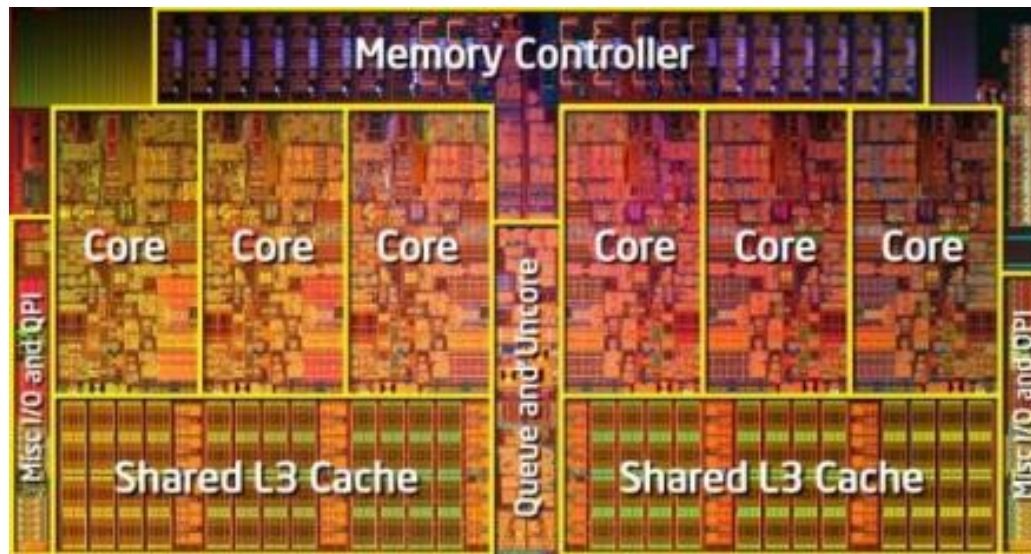
[http://www.intel.com/p/en\\_US/embedded/hwsw/hardware/xeon-5600-5500/overview](http://www.intel.com/p/en_US/embedded/hwsw/hardware/xeon-5600-5500/overview)



# Cluster de prácticas (atcgrid): chip de procesamiento

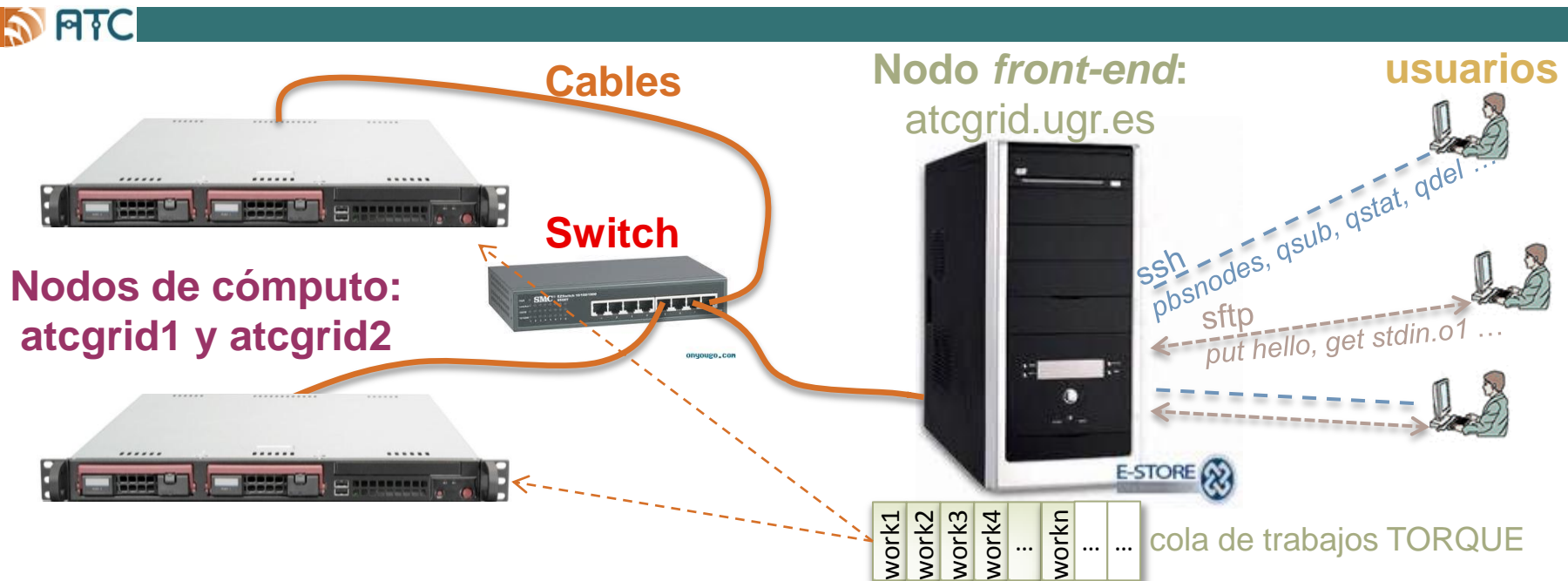


6 cores  
hyperthreading  
2.4 GHz (12  
threads)



Intel Xeon E5645 (6 cores/12 threads, 12M L2 Cache compartida, 2.40 GHz cada core, 5.86 GT/s Intel® QPI)  
[http://ark.intel.com/products/48768?wapkw=\(E5645\)](http://ark.intel.com/products/48768?wapkw=(E5645))

# Cluster de prácticas (atcgrid): acceso



- Cada **usuario** tiene un home en el nodo **front-end** de atcgrid al que accederá:
  - Para ejecutar comandos (`pbsnodes`, `qsub`, `qstat`, `qdel` ...), con un cliente **ssh** (*secure shell*):
    - Linux: `ssh username@atcgrid.ugr.es` (pide *password* del usuario “username”)
  - Para cargar y descargar ficheros (`put hello`, `get stdin.o1`, ...), con un cliente **sftp** (*secure file transfer protocol*):
    - Linux: `sftp username@atcgrid.ugr.es` (pide *password* del usuario “username”)

# Contenidos

- Cluster de prácticas (atcgrid)
- Sistema de colas TORQUE
  - ¿Qué es TORQUE?
  - Ejemplo con comandos TORQUE
- Ejemplo hello OpenOMP en atcgrid usando TORQUE



# ¿Qué es TORQUE?

- Es un gestor de colas y de recursos distribuidos basado en PBS (*Portable Batch System*)
- TORQUE: *Terascale Open-source Resource and QUEue manager*
- Comandos TORQUE que se usarán en prácticas
  - pbsnodes: información de nodos PBS
  - qsub: enviar un trabajo a ejecutar. Devuelve dos ficheros:
    - Salida del programa: en un fichero con extensión que comienza por “.o”
    - Errores : en un fichero con extensión que comienza por “.e”
  - qstat: chequear el estado de los trabajos
  - qdel: eliminar un trabajo
- Otros comandos:
  - <http://docs.adaptivecomputing.com/torque/4-1-3/help.htm#topics/12-appendices/commandsOverview.htm>

# Ejemplos con comandos TORQUE/PBS

➤ Se ejecutarán en el *front-end* con conexión **ssh**

Ejemplo	Explicación
<code>echo 'hello'   <b>qsub</b> -q ac</code>	Envía a ejecutar el trabajo, formado por el ejecutable “hello”, por la cola “ac”.
<code><b>qsub</b> script.sh -q ac</code>	Envía a ejecutar el trabajo, formado por el <i>script</i> “script.sh”, por la cola “ac”
<code>echo 'cat /proc/cpuinfo'   <b>qsub</b> -q ac</code>	Envía a ejecutar el comando “cat /proc/cpuinfo” a través de la cola “ac”. Devuelve en el fichero de salida el contenido del fichero “/proc/cpuid” de los nodos a los que se ha enviado el trabajo.
<code><b>qstat</b></code>	Muestra todos los trabajos que se están ejecutando y los que están encolados en todas las colas
<code><b>qstat</b> -n -u username</code>	Muestra todos los trabajos del usuario “username” y los nodos asignados a cada trabajo (-n)
<code><b>qdel</b> jobid</code>	Elimina el trabajo con identificador “jobid”
<code><b>pbsnodes</b> [-a]</code>	Lista los atributos de todos los nodos

# Contenidos

- Cluster de prácticas (atcgrid)
- Sistema de colas TORQUE
- Ejemplo hello OpenOMP en atcgrid usando TORQUE
  - Ejemplo hello OpenMP
  - Protagonistas de la ejecución
  - Ejecución de hello en atcgrid sin *script*
  - Ejecución de hello en atcgrid con *script*

# Ejemplo hello OpenMP

## HelloOMP.c

- Cada *thread* imprime su identificador
- El identificador se obtiene con la función OpenMP `omp_get_thread_num()`

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

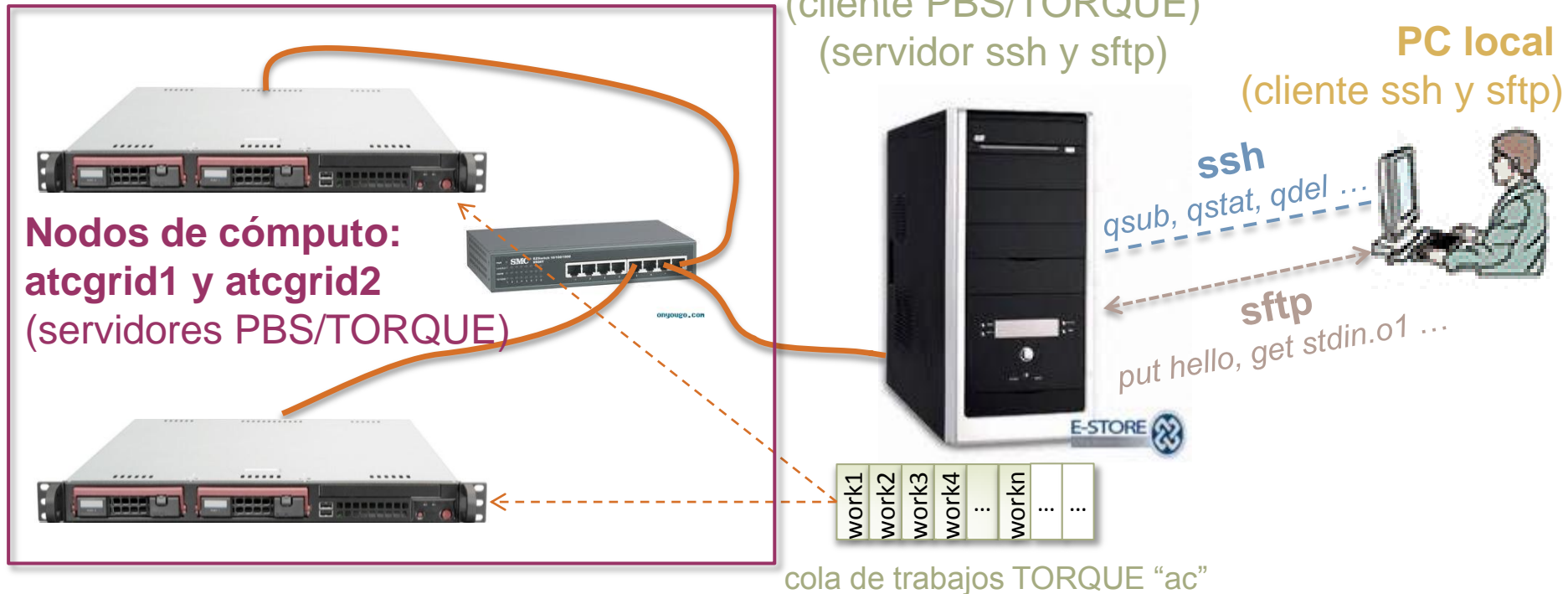
int main(void) {

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

return(0);

}
```

# Protagonistas de la ejecución





# Contenidos

- Cluster de prácticas (atcgrid)
- Sistema de colas TORQUE
- Ejemplo hello OpenOMP en atcgrid usando TORQUE
  - Ejemplo hello OpenMP
  - Protagonistas de la ejecución
- ➔ Ejecución de hello en atcgrid sin *script*
  - Paso 1: Conectar con `ssh` y `sftp` al *front-end* desde PC local
  - Paso 2: Generar el ejecutable en PC local y transferirlo al *front-end*
  - Paso 3: Encolar el ejecutable en la cola “ac” para su ejecución en los servidores
  - Paso 4: Copiar el fichero con resultados desde el *front-end* al PC local
  - Paso 5: Visualizar el fichero de salida en PC local
  - Paso 6: Eliminar el fichero de salida y el de error en *front-end*
- Ejecución de hello en atcgrid con *script*

# Paso 1: Conectar con `ssh` y `sftp` al *front-end* desde PC local

## Ventana `ssh`

```
Archivo  Editar  Ver  Terminal  Ayuda  1
$ ssh mancia@atcgrid.ugr.es
mancia@atcgrid.ugr.es's password:
Last login: Wed Feb 22 13:49:49 2012 from e
[mancia@atcgrid ~]$ mkdir hello
[mancia@atcgrid ~]$ cd hello
[mancia@atcgrid hello]$
```

- 1 (conexión `ssh`):
  - Ejecuta en un terminal `ssh` (introduce password)
  - Crea directorio 'hello' con `mkdir`
  - Pasa al directorio 'hello' con `cd`

## Ventana `sftp`

```
Archivo  Editar  Ver  Terminal  Ayuda  2
$ sftp mancia@atcgrid.ugr.es
Connecting to atcgrid.ugr.es...
mancia@atcgrid.ugr.es's password:
sftp> cd hello
sftp>
```

- 2 (conexión `sftp`):
  - Ejecuta en otro terminal `sftp` (introduce password)
  - Pasa al directorio 'hello' con `cd`

# Paso 2: Generar el ejecutable en PC local y transferirlo al *front-end*

- 3 (genera ejecutable en PC local):
  - Lista contenido directorio con `ls`
  - Genera ejecutable con `gcc` y lo ejecuta

## Ventana ssh

```
Archivo Editar Ver Terminal Ayuda
[mancia@atcgrid hello]$ ls
HelloOMP
```

## Ventana sftp

```
Archivo Editar Ver Terminal Ayuda
sftp> ll
helloomp pi
sftp> lcd helloomp
sftp> ll
HelloOMP HelloOMP.c
sftp> put HelloOMP
Uploading HelloOMP to /home/mancia/hello/HelloOMP
HelloOMP      100% 8696      8.5KB/s   00:00
sftp> ls
HelloOMP
```

## Ventana comandos PC local

```
Archivo Editar Ver Terminal Ayuda
$ ls
HelloOMP.c
$ gcc -fopenmp -O2 -o HelloOMP HelloOMP.c
$ ls
HelloOMP HelloOMP.c
$ ./HelloOMP
(1:!!!Hello World!!!)(0:!!!Hello World!!!)$
```

- 4 (trasferencia de PC local a *front-end*):
  - Lista con `lls` el directorio actual en PC local y cambia de directorio con `lcd` en PC local
  - Transfiere con `put` el ejecutable generado en 3 de PC local a *front-end*
  - Lista con `ls` el directorio actual en *front-end*

# Paso 3: Encolar el ejecutable en la cola ac para su ejecución en los servidores

AC ATC

Ventana ssh

```
Archivo  Editar  Ver  Terminal  Ayuda  6
```

```
[mancia@atcgrid hello]$ ls
HelloOMP
[mancia@atcgrid hello]$ echo 'hello/HelloOMP' | qsub -q ac
195.atcgrid
[mancia@atcgrid hello]$ qstat
```

Job id	Name	User	Time Use	S	Queue
195.atcgrid	STDIN	mancia	00:00:00	C	ac

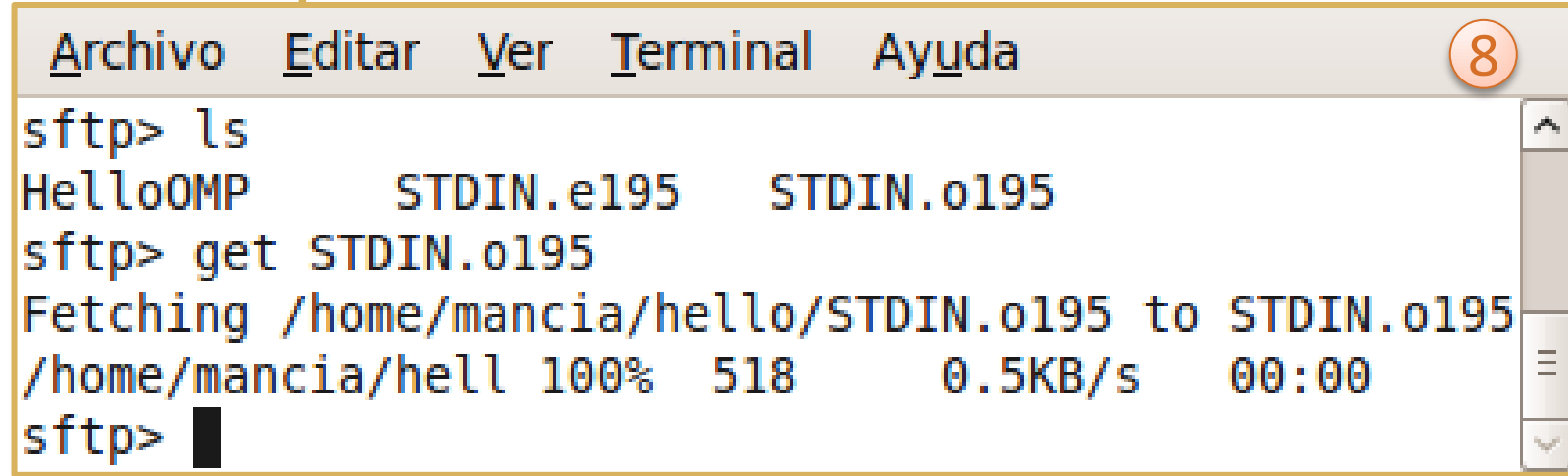
```
[mancia@atcgrid hello]$ ls -lag
total 32
drwxrwxr-x  2 mancia  4096 feb 22 18:21 .
drwx-----  6 mancia 12288 feb 22 13:53 ..
-rwxr-xr-x  1 mancia  8696 feb 22 13:56 HelloOMP
-rw-----  1 mancia    0 feb 22 18:20 STDIN.e195
-rw-----  1 mancia  518 feb 22 18:20 STDIN.o195
```

ficheros con la respuesta de qsub: “.o” (output) y “.e” (errors)

- 6 (ejecución del ejecutable HelloOMP en los servidores):
  - Usa `qsub` para enviar a ejecutar “HelloOMP” a los servidores a través de la cola “ac”
  - Usa `qstat` para ver el estado del trabajo enviado a ejecución (id.: 195.atcgrid)
  - Lista con `ls` el contenido del directorio actual antes y después de la ejecución

# Paso 4: Copiar el fichero con resultados desde el *front-end* al PC local

## Ventana sftp

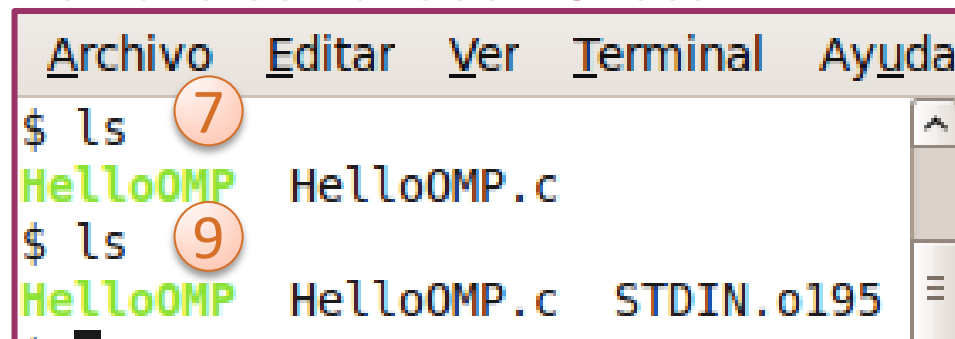


A screenshot of an sftp terminal window. The menu bar at the top includes 'Archivo', 'Editar', 'Ver', 'Terminal', and 'Ayuda'. A circled number '8' is in the top right corner. The terminal shows the following commands and output:

```
sftp> ls
HelloOMP      STDIN.e195    STDIN.o195
sftp> get STDIN.o195
Fetching /home/mancia/hello/STDIN.o195 to STDIN.o195
/home/mancia/hell 100% 518      0.5KB/s   00:00
sftp>
```

- 8 (trasferencia de resultados de *front-end* a PC local):
  - Transfiere con `get` el fichero con la salida de la ejecución (STDIN.o195) de *front-end* a PC local

## Ventana comandos PC local



A screenshot of a local PC terminal window. The menu bar at the top includes 'Archivo', 'Editar', 'Ver', 'Terminal', and 'Ayuda'. Two circled numbers, '7' and '9', are placed next to the first and second `ls` commands respectively. The terminal shows the following commands and output:

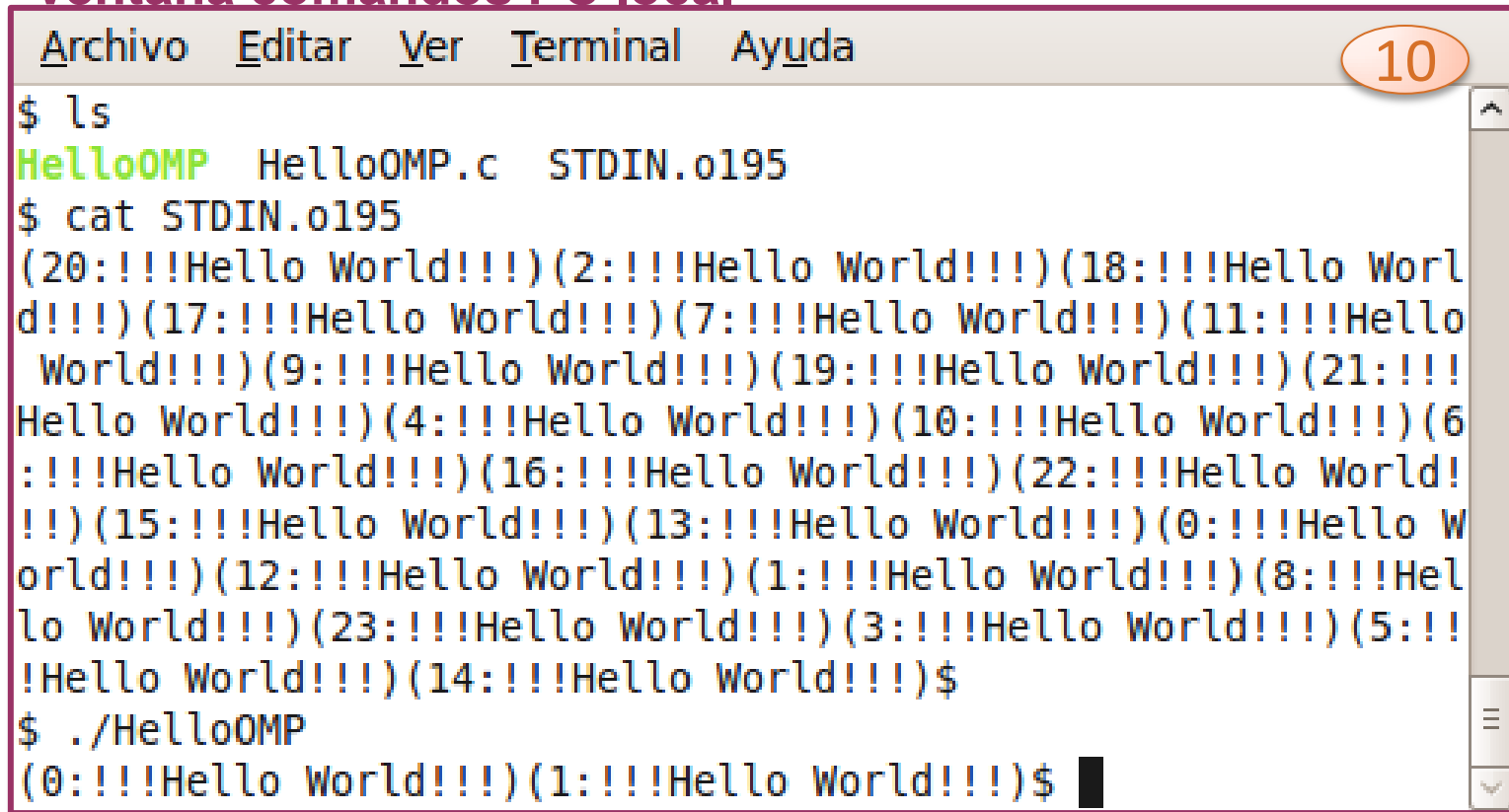
```
$ ls
HelloOMP  HelloOMP.c
$ ls
HelloOMP  HelloOMP.c  STDIN.o195
```

- 7: lista el directorio actual en PC local antes de ejecutar `get` en la ventana de `sftp`
- 9: lista el directorio actual en PC local después de ejecutar `get` en la ventana de `sftp`



# Paso 5: Visualizar el fichero de salida en PC local

## Ventana comandos PC local

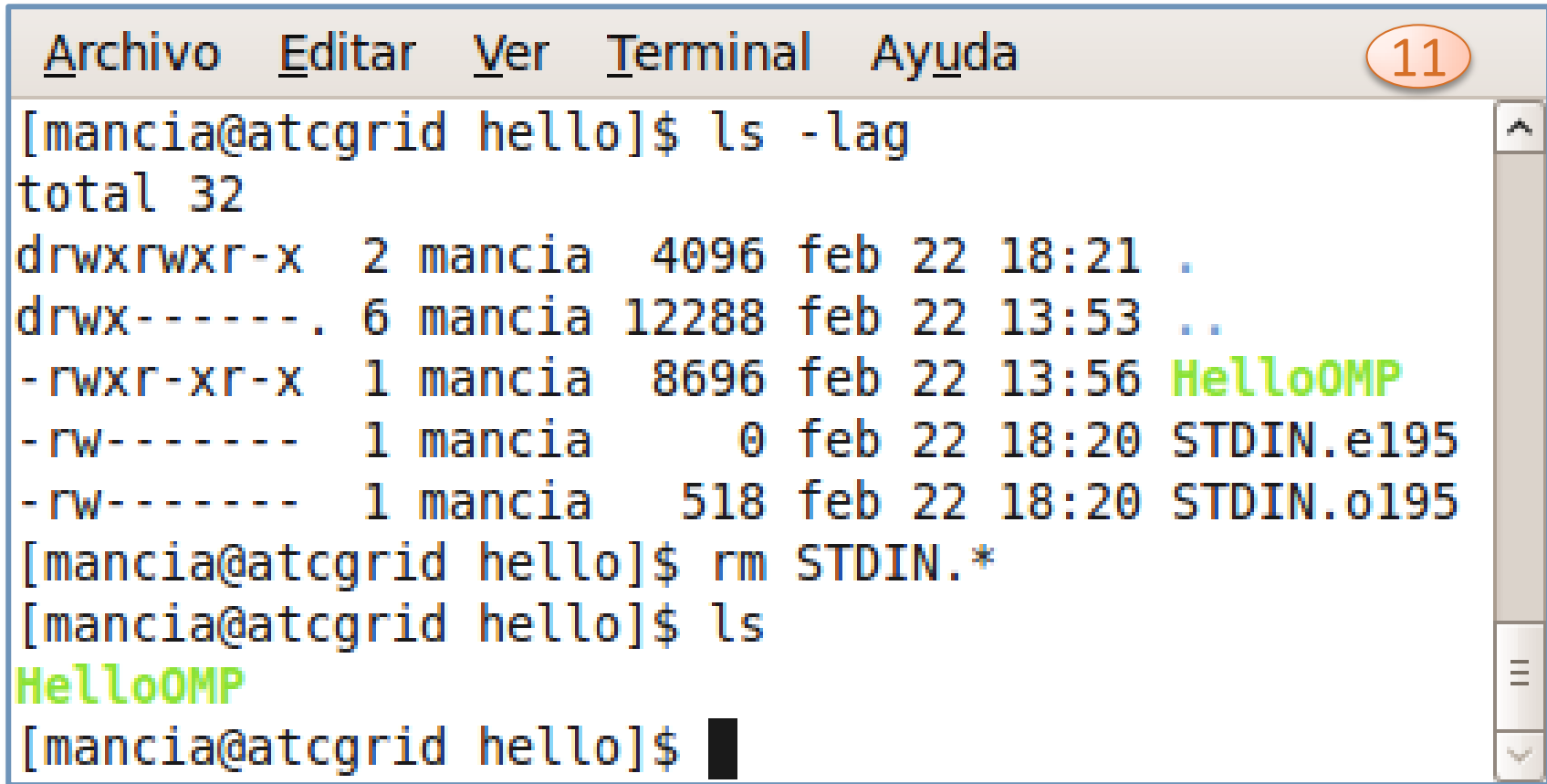


```
Archivo  Editar  Ver  Terminal  Ayuda
$ ls
HelloOMP  HelloOMP.c  STDIN.o195
$ cat STDIN.o195
(20:!!!Hello World!!!)(2:!!!Hello World!!!)(18:!!!Hello World!!!)(17:!!!Hello World!!!)(7:!!!Hello World!!!)(11:!!!Hello World!!!)(9:!!!Hello World!!!)(19:!!!Hello World!!!)(21:!!!Hello World!!!)(4:!!!Hello World!!!)(10:!!!Hello World!!!)(6:!!!Hello World!!!)(16:!!!Hello World!!!)(22:!!!Hello World!!!)(15:!!!Hello World!!!)(13:!!!Hello World!!!)(0:!!!Hello World!!!)(12:!!!Hello World!!!)(1:!!!Hello World!!!)(8:!!!Hello World!!!)(23:!!!Hello World!!!)(3:!!!Hello World!!!)(5:!!!Hello World!!!)(14:!!!Hello World!!!)$
$ ./HelloOMP
(0:!!!Hello World!!!)(1:!!!Hello World!!!)$
```

- 10 (visualización en PC local del contenido del fichero de salida):
  - Visualiza en PC local con `cat` el fichero con la salida de la ejecución (STDIN.o195) que se transfirió previamente desde *front-end* a PC local
  - Ejecuta en PC local “HelloOMP” para comparar la salida de PC local y atcgrid

# Paso 6: Eliminar el fichero de salida y el de error en *front-end*

## Ventana ssh



```
Archivo  Editar  Ver  Terminal  Ayuda  11
[mancia@atcgrid hello]$ ls -lag
total 32
drwxrwxr-x   2 mancia  4096 feb 22 18:21 .
drwx-----  6 mancia 12288 feb 22 13:53 ..
-rwxr-xr-x   1 mancia  8696 feb 22 13:56 HelloOMP
-rw-----   1 mancia    0 feb 22 18:20 STDIN.e195
-rw-----   1 mancia   518 feb 22 18:20 STDIN.o195
[mancia@atcgrid hello]$ rm STDIN.*
[mancia@atcgrid hello]$ ls
HelloOMP
[mancia@atcgrid hello]$
```

- 11 (borrado de ficheros devueltos por `qsub` en *front-end*):
  - Elimina en *front-end* con `rm` los dos ficheros generados por el comando `qsub` (el fichero de salida `STDIN.o195` y el fichero de error `STDIN.e195`)

# Contenidos

- Cluster de prácticas (atcgrid)
- Sistema de colas TORQUE
- Ejemplo hello OpenOMP en atcgrid usando TORQUE
  - Ejemplo hello OpenMP
  - Protagonistas de la ejecución
  - Ejecución de hello en atcgrid sin *script*
  - ➔ Ejecución de hello en atcgrid con *script*
    - *Script* para la ejecución del ejemplo HelloOMP en atcgrid
    - Paso 1: Traslencia del *script* desde PC local al *front-end* usando `sftp`
    - Paso 2: Ejecución del *script* en los servidores del cluster con TORQUE
    - Paso 3: Traslencia del fichero de salida desde *front-end* a PC local usando `sftp`
    - Paso 4: Visualización en PC local del contenido del fichero de salida

# Script para la ejecución del ejemplo HelloOMP en atcgrid

## script\_helloomp.sh

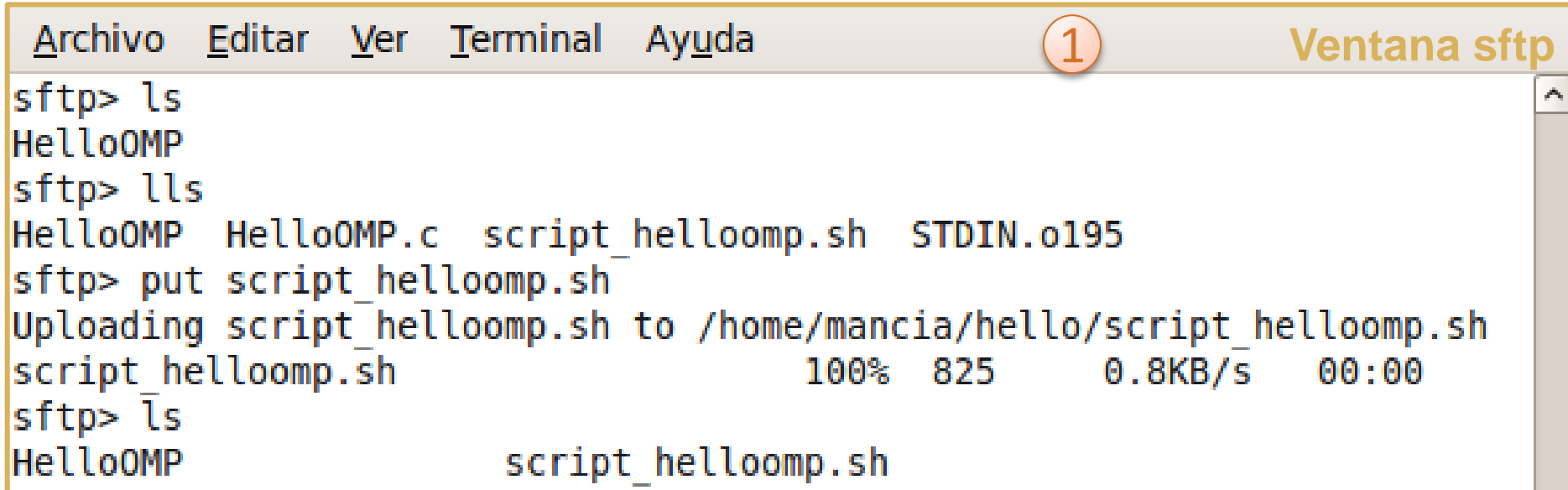
```
#!/bin/bash
#Se asigna al trabajo el nombre helloomp
#PBS -N helloomp
#Se asigna al trabajo la cola ac
#PBS -q ac
#Se imprime información del trabajo usando variables de entorno de PBS
echo "Id. usuario del trabajo: $PBS_O_LOGNAME"
echo "Id. del trabajo: $PBS_JOBID"
echo "Nombre del trabajo especificado por usuario: $PBS_JOBNAME"
echo "Nodo que ejecuta qsub: $PBS_O_HOST"
echo "Cola: $PBS_QUEUE"
echo "Nodos asignados al trabajo:"
cat $PBS_NODEFILE

#Se fija a 12 el nº de threads máximo (tantos como cores en un nodo)
export OMP_THREAD_LIMIT=12
echo "Nº de threads inicial: $OMP_THREAD_LIMIT"
#Se ejecuta HelloOMP, que está en el directorio en el que se ha ejecutado qsub
for ((P=OMP_THREAD_LIMIT;P>0;P=P/2))
do
    export OMP_NUM_THREADS=$P
    echo -e "\nPara $OMP_NUM_THREADS threads:"
    $PBS_O_WORKDIR /HelloOMP
done
```

Para imprimir  
variables de  
entorno PBS  
Órdenes  
para PBS

Instrucciones del  
script

# Paso 1: Transferencia del *script* desde PC local al *front-end* usando `sftp`



The screenshot shows a terminal window titled "Ventana sftp" with a menu bar containing "Archivo", "Editar", "Ver", "Terminal", and "Ayuda". A red circle with the number "1" is positioned above the terminal area. The terminal output is as follows:

```
sftp> ls
HelloOMP
sftp> ll
HelloOMP  HelloOMP.c  script_helloomp.sh  STDIN.o195
sftp> put script_helloomp.sh
Uploading script_helloomp.sh to /home/mancia/hello/script_helloomp.sh
script_helloomp.sh                100% 825      0.8KB/s   00:00
sftp> ls
HelloOMP                          script_helloomp.sh
```



# Paso 2: Ejecución del *script* en los servidores del cluster con TORQUE

Archivo Editar Ver Terminal Ayuda 2 Ventana ssh

```
[mancia@atcgrid hello]$ qstat
[mancia@atcgrid hello]$ ls
HelloOMP
[mancia@atcgrid hello]$ ls
HelloOMP script_helloomp.sh
[mancia@atcgrid hello]$ qsub script_helloomp.sh
200.atcgrid
[mancia@atcgrid hello]$ qstat
```

Job id	Name	User	Time Use	S	Queue
200.atcgrid	helloomp	mancia	00:00:00	C	ac

```
[mancia@atcgrid hello]$ ls -lag
total 36
drwxrwxr-x  2 mancia  4096 feb 22 19:46 .
drwx-----  6 mancia 12288 feb 22 13:53 ..
-rwxr-xr-x  1 mancia  8696 feb 22 13:56 HelloOMP
-rw-----  1 mancia    0 feb 22 19:45 helloomp.e200
-rw-----  1 mancia   753 feb 22 19:45 helloomp.o200
-rw-r--r--  1 mancia   825 feb 22 19:46 script_helloomp.sh
[mancia@atcgrid hello]$
```

# Paso 3: Traslado del fichero de salida desde *front-end* a PC local usando `sftp`

```
Archivo  Editar  Ver  Terminal  Ayuda  Ventana sftp
sftp> ls
HelloOMP
sftp> ll
HelloOMP  HelloOMP.c  script_helloomp.sh  STDIN.o195
sftp> put script_helloomp.sh
Uploading script_helloomp.sh to /home/mancia/hello/script_helloomp.sh
script_helloomp.sh          100% 825      0.8KB/s   00:00
sftp> ls
HelloOMP  ----- script_helloomp.sh -----
sftp> ls
HelloOMP          helloomp.e200          helloomp.o200
script_helloomp.sh
sftp> get helloomp.o200
Fetching /home/mancia/hello/helloomp.o200 to helloomp.o200
/home/mancia/hello/helloomp.o200  100% 753      0.7KB/s   00:00
sftp> 
```

3

# Paso 4: Visualización en PC local del contenido del fichero de salida

```
$ ls
HelloOMP HelloOMP.c helloomp.o200 script_helloomp.sh STDIN.o195
$ cat helloomp.o200
Id. usuario del trabajo: mancia
Id. del trabajo: 200.atcgrid
Nombre del trabajo especificado por usuario: helloomp
Nodo que ejecuta qsub: atcgrid
Cola: ac
Nodos asignados al trabajo:
atcgrid1
Nº de threads inicial: 12

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