



Gestión de Trabajo en Cluster LAM/MPI mediante aplicativo KFRONT

Constantino A. Palacio

Grupo de Control Automático (GCA), Instituto LEICI, FI-UNLP.

Índice

1. Introducción	2
2. Especificaciones del Cluster del GCA	3
3. Preparativos	4
3.1. Requerimientos de Software	4
3.2. Acceso al Servidor Remoto	4
3.2.1. Transferencia de Archivos con WinSCP	4
3.2.2. Conexión Remota a Terminal con PuTTY	7
4. Administración del Cluster	9
4.1. Generalidades	9
4.1.1. Formato del Archivo de Configuración	10
4.2. Configuración del Cluster	10
4.2.1. Reordenar Nodos	11
4.2.2. Agregar/Quitar Nodos	12
4.2.3. Cambiar Estado de Nodo	13
4.2.4. Reasignar Maestro	13
4.3. Control de entorno LAM	14
5. Gestión de Trabajos	15
5.1. Compilar Trabajo	15
5.2. Copiar Archivo Binario	16
5.3. Ejecutar Programa	16
5.4. Compilar y Ejecutar	16
6. Ejemplo Completo	17

1. Introducción

En el presente informe se describe la operación de la herramienta de software KFRONT para gestión de trabajo en un cluster de computadoras basado en LAM/MPI.

Esta herramienta permite al operador automatizar cada paso de la ejecución de un programa distribuido, desde la preparación del cluster hasta la compilación del programa y la puesta en marcha del mismo.

En las siguientes secciones se abordará la operación del aplicativo según aparezcan las operaciones en el menú.

2. Especificaciones del Cluster del GCA

El Grupo de Control Automático (GCA) dispone de un cluster compuesto por 15 equipos monoprocesador de características heterogéneas. La tabla 1 resume las principales especificaciones de hardware y software de cada uno de los nodos del cluster.

Nombre	IP (/24)	CPU	f_{CPU} [Hz]	RAM [B]	f_M [Hz]	SO
kloster00	192.168.1.100	Celeron	300M	128M	100M	RedHat 6.2
kloster01	192.168.1.101	Celeron	300M	128M	100M	RedHat 6.2
kloster02	192.168.1.102	Celeron	300M	128M	100M	RedHat 6.2
kloster03	192.168.1.103	Celeron	300M	128M	100M	RedHat 6.2
kloster04	192.168.1.104	Celeron	300M	128M	100M	RedHat 6.2
kloster05	192.168.1.105	Celeron	300M	128M	100M	RedHat 6.2
kloster06	192.168.1.106	Celeron	300M	160M	100M	RedHat 6.2
kloster07	192.168.1.107	Celeron	300M	128M	100M	RedHat 6.2
kloster08	192.168.1.108	Celeron	300M	128M	100M	RedHat 6.2
kloster09	192.168.1.109	Pentium III	1.0G	256M	133M	RedHat 6.2
alfa00	192.168.1.200	Pentium 4	1.8G	512M	133M	Ubuntu 18
alfa01	192.168.1.201	Pentium 4	1.8G	512M	133M	Ubuntu 18
alfa02	192.168.1.202	Pentium 4	1.8G	512M	133M	Ubuntu 18
alfa03	192.168.1.203	Pentium 4	1.8G	512M	133M	Ubuntu 18
alfa04	192.168.1.204	Pentium 4	1.8G	512M	133M	Ubuntu 18

Tabla 1: Principales especificaciones del cluster.

Cada uno de los nodos está dotado de una placa de red Ethernet de 100Mbps conectada mediante un cable UTP categoría 5/e a un hub (repetidor multipuerto) modelo ESH-717, que a su vez se conecta de la misma manera a un adaptador de red de 100Mbps en un servidor disponible para proyectos, accesible mediante una dirección IP pública. En la figura 1 se puede apreciar la topología de la red a la que se conecta cada uno de los nodos del cluster. Observar que se modela también la conexión remota de un usuario a través de Internet.

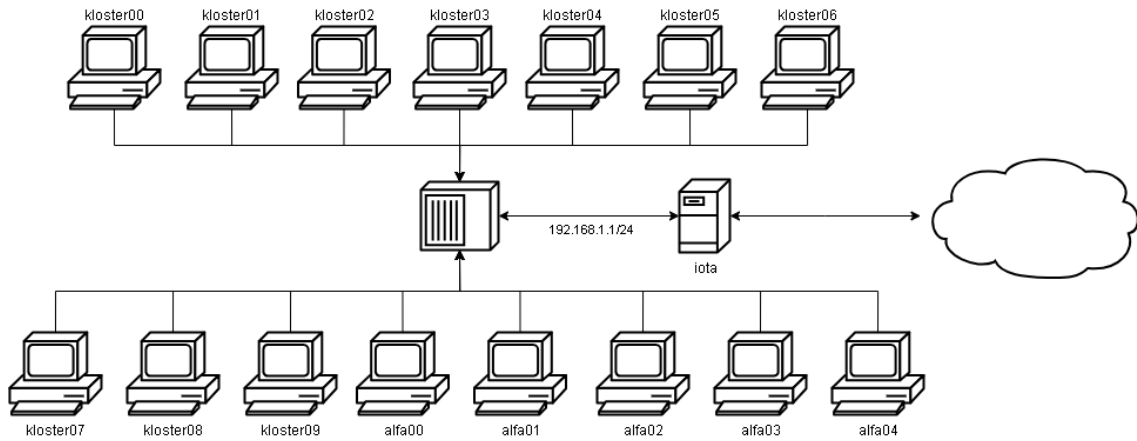


Figura 1: Topología del cluster.

3. Preparativos

3.1. Requerimientos de Software

Con el objetivo de simplificar las transferencias desde y hacia el servidor de acceso, se asume que el operador cuenta con el programa WinSCP instalado en su equipo, junto al cliente de conexión remota PuTTY. Para la instalación de estos programas se siguen las instrucciones de los respectivos autores.

3.2. Acceso al Servidor Remoto

Para el acceso al servidor remoto se contará con credenciales de acceso proporcionados por el administrador del cluster. El acceso puede ser mediante cualquier cliente SSH, aunque se recomienda el uso de PuTTY para que sea consistente con lo explicado en el presente informe.

3.2.1. Transferencia de Archivos con WinSCP

Al abrir WinSCP por primera vez se observa una pantalla similar a la ilustrada en la figura 2. De esta pantalla se pueden crear diferentes conexiones para diferentes equipos.

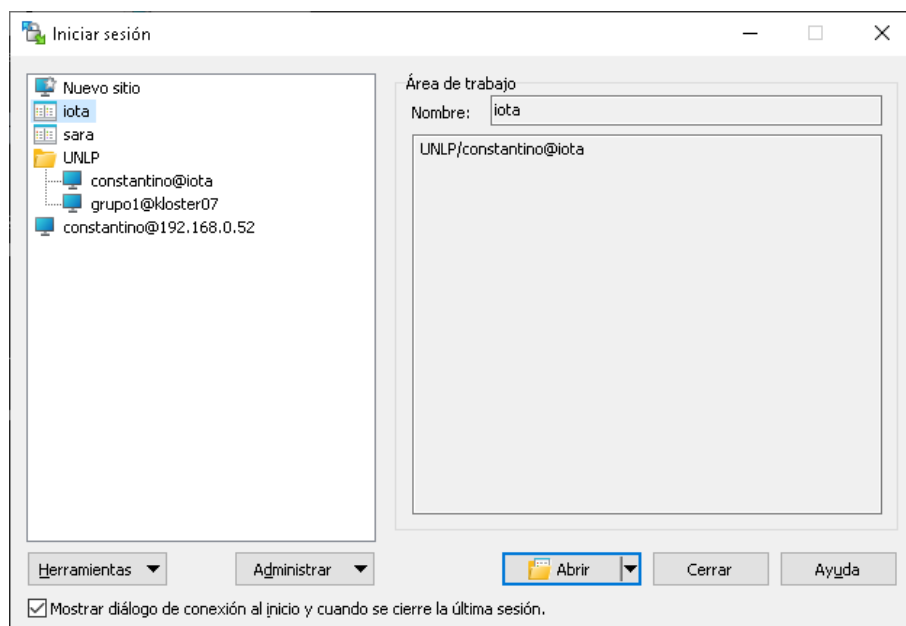


Figura 2: Pantalla inicial de WinSCP.

Para crear una nueva conexión, hacer clic en la opción “Nuevo Sitio” ubicada en el panel del lado izquierdo de la ventana. Luego, completar los datos de la conexión.

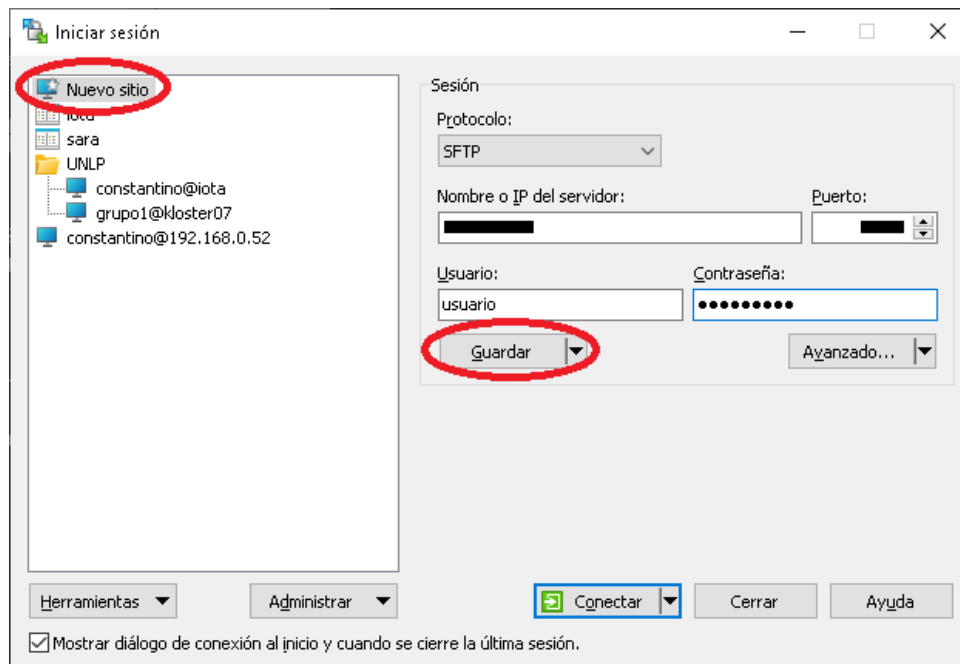


Figura 3: Carga de la información de la conexión en WinSCP.

El proceso de carga de la información se puede ver en la figura 3. Al finalizar, hacer clic en el botón “Guardar”. Aparecerá un cuadro de diálogo como el de la figura 4. Las opciones seleccionables en este cuadro no son relevantes a los efectos de la conexión al servidor y quedan a criterio del operador.

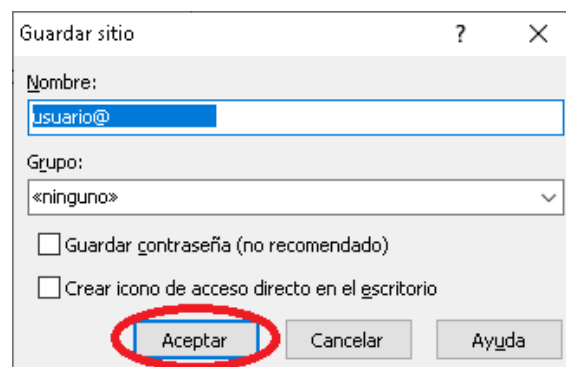


Figura 4: Terminando las configuraciones de WinSCP.

Desde la pantalla principal se puede iniciar sesión como se muestra en la figura 5. De forma alternativa, se puede invocar directamente desde un acceso directo si se seleccionó dicha opción al crear la conexión. El programa solicitará la contraseña para acceder, tal como se muestra en la figura 6.

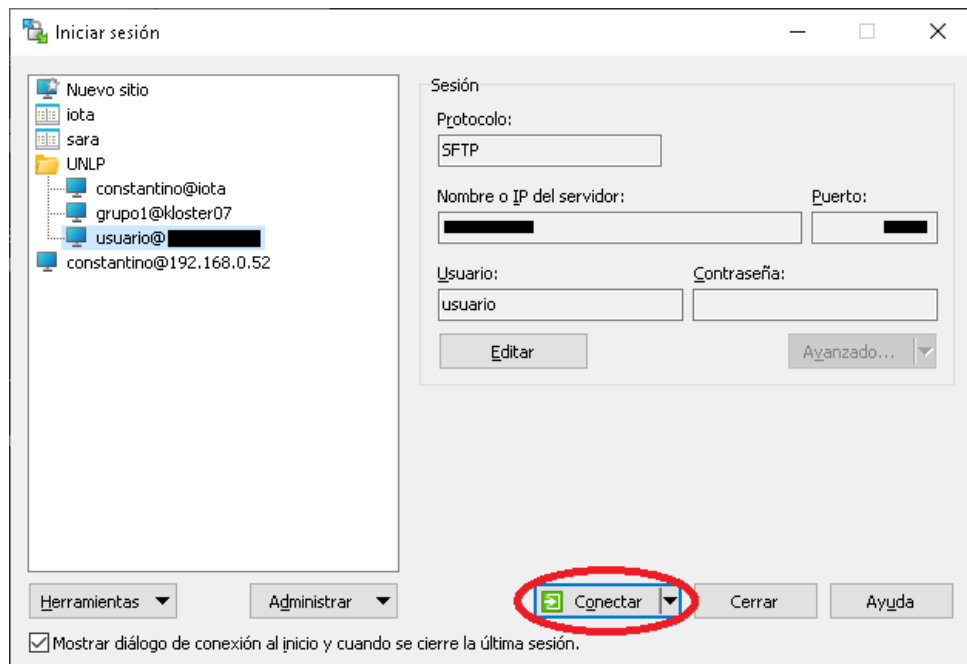


Figura 5: Abriendo la conexión realizada.

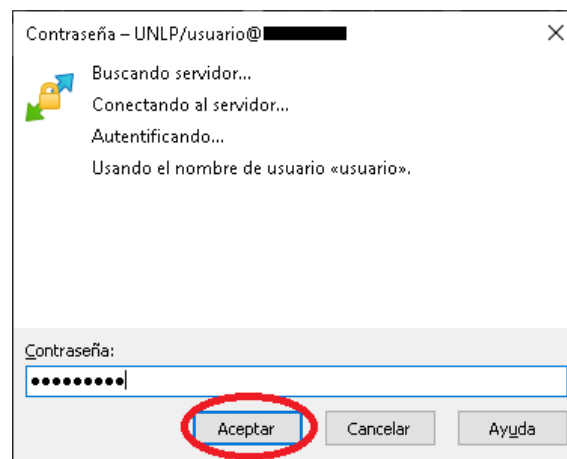


Figura 6: WinSCP solicita la clave de acceso al equipo remoto.

Al establecerse la conexión, WinSCP muestra una interfaz como la que se puede observar en la figura 7. El panel de la izquierda muestra un directorio local y, en el lado derecho, el directorio remoto.

Se pueden transferir archivos y carpetas desde y hacia el servidor simplemente arrastrándolos al panel del directorio remoto.

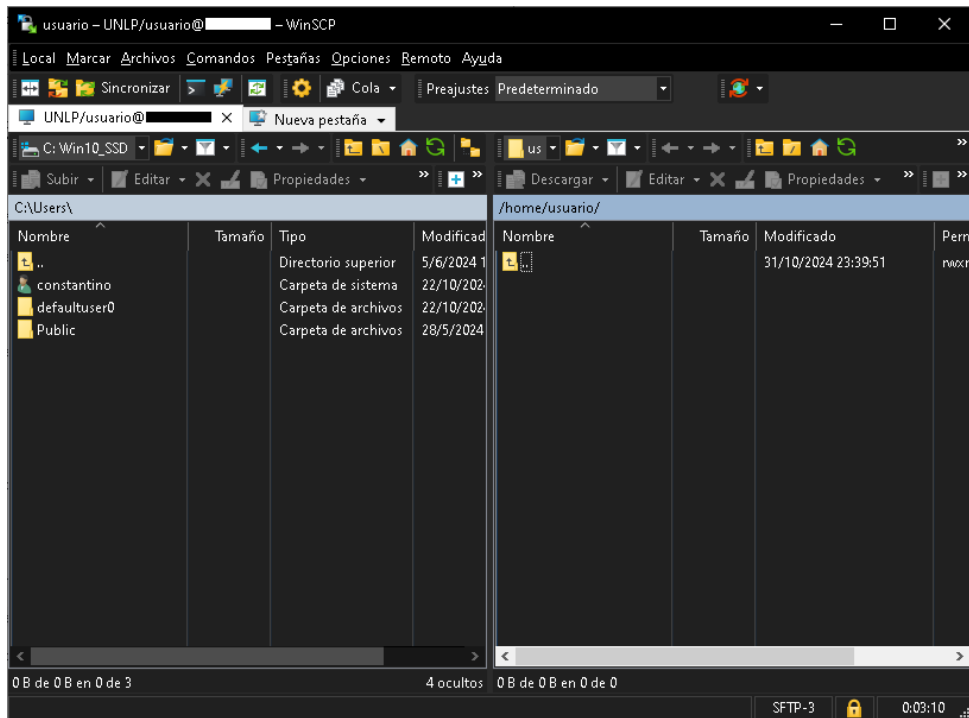


Figura 7: Ventana principal de WinSCP: directorio local (izq.) y remoto (der.).

3.2.2. Conexión Remota a Terminal con PuTTY

El programa PuTTY puede invocarse de dos maneras distintas: desde WinSCP o desde un acceso directo al ejecutable.

Desde la ventana principal de WinSCP se puede hacer clic en el ícono de conexión usando PuTTY. Se abrirá una ventana de terminal y se solicitará la clave de acceso, como se ilustra en la figura 8.



Figura 8: Sesión iniciada en PuTTY.

La principal desventaja de esta forma de acceso es que la conexión a la terminal es dependiente de la conexión de WinSCP. Es decir, si se cortara la conexión de WinSCP por cualquier motivo, se perdería también el acceso por terminal. Una forma de solventar este problema es ingresando a través del acceso directo a PuTTY creado durante la instalación.

Como puede verse en la figura 9, invocar PuTTY directamente no solo independiza el mando por consola de la ventana de WinSCP, sino que además brinda la posibilidad de una configuración más extensa de la conexión. Ingresar la dirección IP del servidor y el puerto de conexión. Luego, hacer clic en “Open”.

Se solicita ingreso de nombre de usuario y contraseña. Al ingresar los datos, observar en la figura 10 que el resultado es el mismo que al abrir la terminal desde WinSCP.

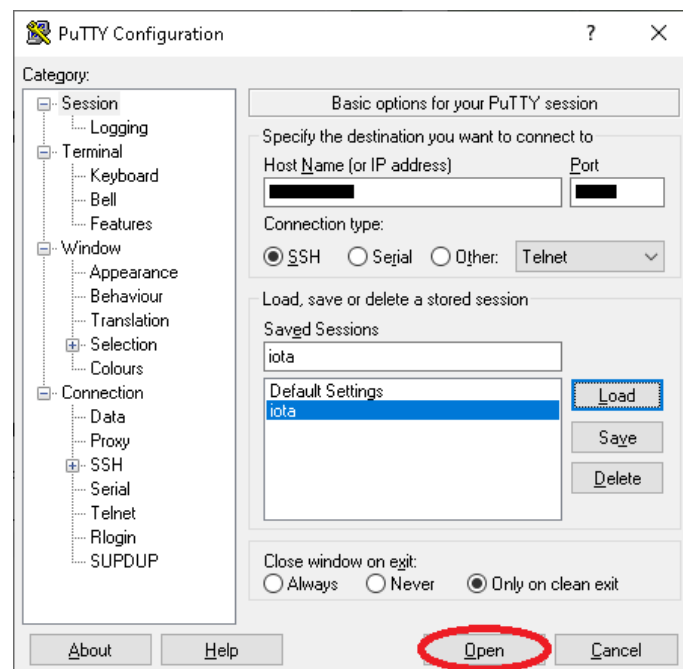


Figura 9: Abriendo una sesión en PuTTY.

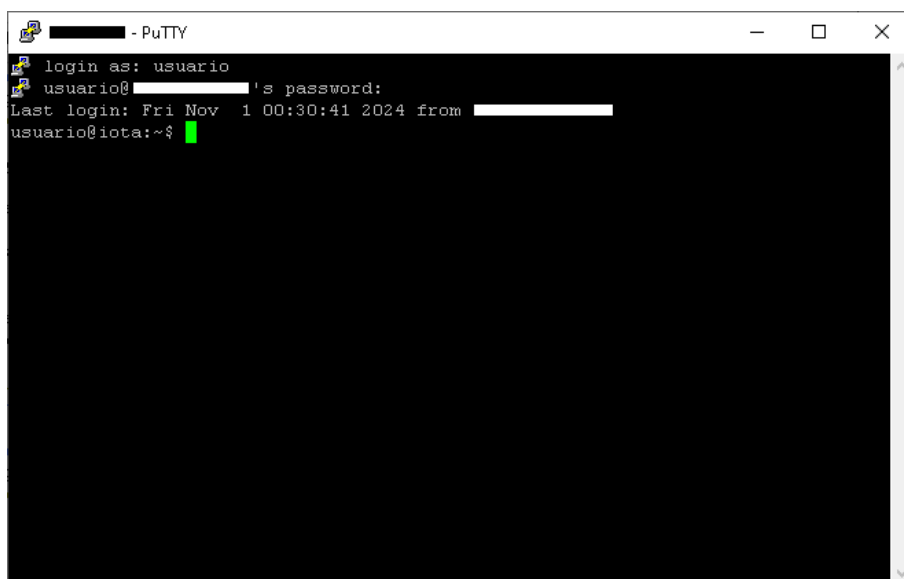


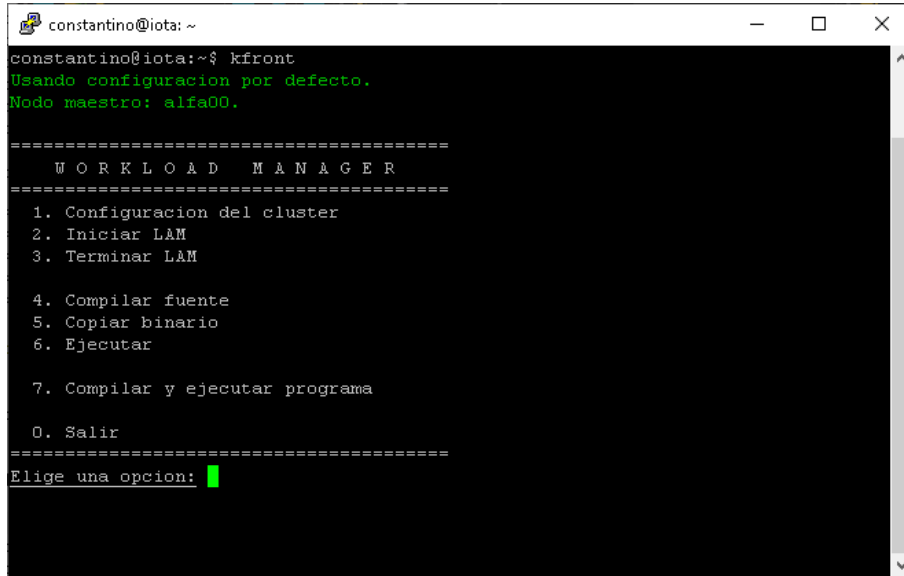
Figura 10: Terminal remota abierta manualmente.

4. Administración del Cluster

4.1. Generalidades

El aplicativo KFRONT se invoca directamente desde el *prompt* del sistema operativo remoto mediante la orden `kfront`. Se puede, de manera opcional, enviar como argumento la ruta o el nombre de un archivo sin tipo que guarde en forma de texto plano la lista de los nodos de cluster que se desea utilizar. De no especificarse, KFRONT arma un listado de los nodos `alfa00` a `alfa04`.

En la figura 11 se puede observar la pantalla de bienvenida de KFRONT.



```
constantino@iota: ~  
constantino@iota:~$ kfront  
Usando configuracion por defecto.  
Nodo maestro: alfa00.  
  
===== WORKLOAD MANAGER =====  
===== 1. Configuracion del cluster  
2. Iniciar LAM  
3. Terminar LAM  
  
4. Compilar fuente  
5. Copiar binario  
6. Ejecutar  
  
7. Compilar y ejecutar programa  
  
0. Salir  
===== Elige una opcion: █
```

Figura 11: Pantalla inicial de KFRONT.

El aplicativo utiliza un esquema de colores para diferentes significados de los mensajes impresos en la terminal, como se puede ver en la tabla 2. Los significados más específicos de los colores serán detallados a medida que surjan durante el uso de KFRONT.

Color	Significado
Blanco	Texto estándar
Verde	Mensaje del sistema
Rojo	Error del sistema
Amarillo	Resultado de ejecución
Magenta	Comando ejecutado en segundo plano
Cian	Comando clave ejecutado en segundo plano

Tabla 2: Convención de colores empleada por KFRONT.

Usando la información de la tabla 2 y la figura 11 se puede ver que la información importante presente en la pantalla principal es el nodo maestro del cluster, en este caso `kloster00`. Este mensaje aparecerá siempre para indicar al operador el nodo en el cual se compilará el programa y desde el cual se realizarán las tareas MPI.

Otros mensajes que podrían aparecer en esta pantalla (en verde) son:

- “Cargando lista de nodos”: el sistema está leyendo la lista de nodos suministrada (o la generada por defecto) de a una línea por vez y estableciendo las conexiones de red. Puede demorar un tiempo, dependiendo del estado de las conexiones a cada uno de los nodos.
- “Utilizando configuración por defecto”: el operador no suministró un archivo de configuración inicial del cluster, por lo que se utilizará la lista de nodos generada por el programa.
- “Hay una sesión previa de LAM, terminando”: se detectó una sesión de LAM previa que no fue terminada correctamente. El sistema procede a cerrarla para que el entorno de trabajo esté limpio. Esta acción puede demorar un tiempo dependiendo de la carga del sistema en el nodo maestro.

Siguiendo la convención de colores, aparecerán en color rojo los errores en la configuración de cluster. Estos mensajes son fatales, es decir, causan la terminación inmediata del aplicativo:

- “El archivo de configuración no existe”: el sistema no pudo acceder al archivo de configuración suministrado por el operador. Intente revisar el nombre o ruta al mismo o bien los permisos de acceso.
- “Listado inválido”: el archivo suministrado no contiene una lista de nodos que pueda ser utilizada por KFRONT. Este error se produce cuando el archivo no está en el formato requerido o bien cuando ninguno de los nodos del listado es accesible, ya sea por problemas de red o que el cluster se encuentre apagado.

4.1.1. Formato del Archivo de Configuración

A continuación se deja un ejemplo de un archivo de configuración del cluster en un formato aceptable por KFRONT. Este archivo es una lista de las direcciones IP o los nombres de cada uno de los nodos.

```
# Archivo de configuracion de prueba
# Las lineas que empiezan con # son comentarios
#
kloster00
kloster01
kloster02
kloster03
```

4.2. Configuración del Cluster

Desde la pantalla de bienvenida se puede elegir la opción 1 para modificar la configuración del cluster, lo que permite automatizar el armado de los archivos de configuración de LAM utilizando un menú de opciones.

En la pantalla de configuración (figura 12) se muestra un listado de nodos del cluster, donde en color verde figuran todos los nodos que están actualmente conectados a la red y en blanco (o sin atributo) los que están *offline*. Se resalta adicionalmente en video inverso el nodo maestro.

En esta pantalla se muestra también el estado de LAM para que sea tenido en cuenta al momento de realizar las diferentes operaciones disponibles al operador.

```

constantino@iota: ~/kfront

0. Salir
=====
Elige una opcion: 1

Listado de nodos:
=====
n0  alfa00  [0]  [0]
n1  alfa01  [0]  [0]
n2  alfa02  [0]  [0]
n3  alfa03  [0]  [0]
n4  alfa04  [0]  [0]
=====

LAM inactivo.

1. Reordenar nodos
2. Agregar nodo
3. Remover nodo
4. Cambiar estado de nodo
5. Reasignar maestro

0. Terminar y volver al menu

Elige una opcion: 

```

Figura 12: Pantalla de configuración del cluster.

4.2.1. Reordenar Nodos

Se puede, desde la opción 1 del menú de configuración del cluster, intercambiar la posición de dos nodos dentro del cluster. Esto afectará el orden en que los nodos aparecen en la configuración del entorno LAM. El sistema solicitará el ingreso de los identificadores de ambos nodos y luego procederá a realizar el cambio.

Si LAM está activo en el nodo maestro, KFRONT lo reinicia automáticamente para que se reflejen las configuraciones actualizadas.

Por ejemplo, dada la siguiente configuración del cluster:

n0	alfa00	[0]	[0]	<- maestro
n1	alfa01	[0]	[0]	
n2	alfa02	[0]	[0]	
n3	alfa03	[0]	[0]	
n4	alfa04	[0]	[0]	

El sistema solicita:

```

Introduzca identificador de nodo: <se ingresa n0>
Introduzca identificador de nodo: <se ingresa n3>

```

El sistema responde:

n0	alfa03	[0]	[0]	
n1	alfa01	[0]	[0]	
n2	alfa02	[0]	[0]	
n3	alfa00	[0]	[0]	<- maestro
n4	alfa04	[0]	[0]	

Tener presente que al reordenar los nodos *no se reasigna* el maestro para que sea n0, sino que seguirá siendo alfa00 (ahora nodo n3). Si se desea que el maestro sea n0, se debe reasignar el maestro tal que sea el nodo n0.

4.2.2. Agregar/Quitar Nodos

Desde el menú de configuración del cluster se pueden agregar nodos nuevos al cluster. Para ello, seleccionar la opción (2) e ingresar la IP o el nombre del nodo cuando el sistema lo solicite. El sistema informará de cualquier error siguiendo la convención de colores de la tabla 2.

De estar activo, el entorno LAM no se reinicia. KFRONT se encarga de invocar el comando `lamgrow` con la sintaxis apropiada.

Por ejemplo, dada la siguiente configuración del cluster:

n0	alfa00	[0]	[0]	<- maestro
n1	alfa01	[0]	[0]	
n2	alfa02	[0]	[0]	
n3	alfa03	[0]	[0]	
n4	alfa04	[0]	[0]	

Se desea agregar el nodo `kloster00`. El sistema solicita:

```
Introduzca nombre/direccion IP: <se ingresa kloster00>
```

Si el nodo es accesible desde la red, el sistema responde:

n0	alfa00	[0]	[0]	<- maestro
n1	alfa01	[0]	[0]	
n2	alfa02	[0]	[0]	
n3	alfa03	[0]	[0]	
n4	alfa04	[0]	[0]	
n5	kloster00	[0]	[0]	

Si, en cambio, el nodo no fuera accesible desde la red, el sistema mostrará un mensaje de error y volverá al menú de configuración sin realizar cambios.

Tener en cuenta que, al haberse agregado `kloster00`, el cluster ahora es heterogéneo. En tal caso debe reasignarse el maestro para que sea el nodo recién agregado, pues es el nodo más “viejo” en términos de hardware y sistema operativo.

Si ahora se deseara remover el nodo `n3`, se utiliza la operación 3 del menú. El sistema solicita:

```
Introduzca identificador de nodo: <se ingresa n3>
```

El sistema debe reasignar los identificadores de los demás nodos para que sean consecutivos. Por esta razón, si LAM estuviese activo, se lo detiene, se realizan los cambios y se vuelve a iniciar. La respuesta final del sistema es:

n0	alfa00	[0]	[0]	<- maestro
n1	alfa01	[0]	[0]	
n2	alfa02	[0]	[0]	
n3	alfa04	[0]	[0]	
n4	kloster00	[0]	[0]	

En caso de eliminar al nodo maestro, el sistema asigna este rol al primer nodo *online* que le siga en la lista. Al ejecutar nuevamente la herramienta 3, el sistema solicita:

```
Introduzca identificador de nodo: <se ingresa n0>
```

El sistema responde:

n0	alfa01	[0]	[0]	<- maestro
n1	alfa02	[0]	[0]	
n2	alfa04	[0]	[0]	
n3	kloster00	[0]	[0]	

4.2.3. Cambiar Estado de Nodo

Esta operación permite agregar o quitar lógicamente nodos del cluster. Cuando se inicia el entorno LAM, la lista de nodos que se arma y de envía al maestro para ejecutar el comando `lamboot -v lamhosts` es construida a partir de los nodos que son accesibles desde la red y están seleccionados para que formen parte del cluster. Esta herramienta permite modificar rápidamente los parámetros del cluster, únicamente debiendo ser reiniciado LAM cuando se modifica el estado del nodo maestro.

La operación de selección solicita al operador el identificador del nodo que se desea agregar/quitar.

Por ejemplo, se tiene la siguiente configuración:

n0	alfa00	[0]	[0]	<- maestro
n1	alfa01	[0]	[0]	
n2	alfa02	[0]	[0]	
n3	alfa03	[0]	[0]	
n4	alfa04	[0]	[0]	

Se desea que el programa se ejecute sólo en `alfa00` y `alfa01`, por lo que se tienen que remover los nodos restantes de la configuración. Para ello se utiliza la operación de selección de nodos. Debe invocarse tres veces para remover los nodos `n2`, `n3` y `n4`. El resultado es la siguiente configuración:

n0	alfa00	[0]	[0]	<- maestro
n1	alfa01	[0]	[0]	
n2	alfa02	[X]	[0]	
n3	alfa03	[X]	[0]	
n4	alfa04	[X]	[0]	

Si posteriormente se decide ejecutar el programa sobre tres nodos en vez de dos, se puede habilitar uno de los nodos previamente deshabilitados procediendo de igual forma.

4.2.4. Reasignar Maestro

En los casos donde el cluster sea heterogéneo o se desee probar la ejecución desde varios maestros, esta operación permite reasignar el rol de nodo maestro a cualquier nodo accesible desde la red. El sistema le solicitará al operador el identificador del nuevo maestro.

Por ejemplo, se tiene la siguiente configuración y se desea que el maestro sea el nodo `kloster00`:

n0	alfa00	[0]	[0]	<- maestro
n1	alfa01	[0]	[0]	
n2	alfa02	[0]	[0]	
n3	alfa03	[0]	[0]	
n4	alfa04	[0]	[0]	
n5	kloster00	[0]	[0]	

El sistema solicita:

```
Introduzca identificador de nodo: <se ingresa n5>
```

Se procederá a realizar automáticamente el intercambio entre ambos nodos y, si corresponde, reiniciar el entorno LAM para que se reflejen los cambios en la configuración, como se muestra a continuación:

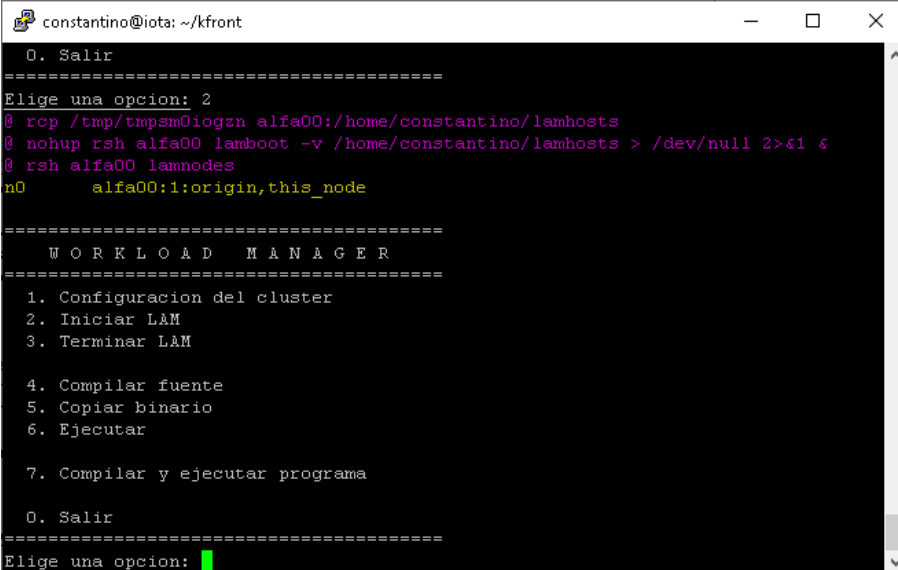
n0	kloster00	[0]	[0]	<- maestro
n1	alfa01	[0]	[0]	
n2	alfa02	[0]	[0]	
n3	alfa03	[0]	[0]	
n4	alfa04	[0]	[0]	
n5	alfa00	[0]	[0]	

4.3. Control de entorno LAM

Para iniciar el entorno LAM, elegir la opción (2) del menú principal. El programa armará un archivo de configuración con los parámetros seleccionados y lo enviará al nodo maestro, para luego invocar una serie de comandos que prepararán el entorno LAM.

Tener en cuenta que, para el caso de un cluster heterogéneo, el nodo maestro debe ser *siempre* el que tenga la versión más antigua del sistema operativo debido a la diferencia de versión de GCC incluida en cada distribución.

Se muestra en la figura 13 el resultado de iniciar el entorno con la configuración por defecto:



```
constantino@iota: ~/kfront
0. Salir
=====
Elige una opcion: 2
% rcp /tmp/tmpsm0iogzn alfa00:/home/constantino/lamhosts
% nohup rsh alfa00 lamboot -v /home/constantino/lamhosts > /dev/null 2>&1 &
% rsh alfa00 lamnodes
n0      alfa00:1:origin,this_node
=====
      W O R K L O A D   M A N A G E R
=====
1. Configuracion del cluster
2. Iniciar LAM
3. Terminar LAM

4. Compilar fuente
5. Copiar binario
6. Ejecutar

7. Compilar y ejecutar programa

0. Salir
=====
Elige una opcion: █
```

Figura 13: Entorno LAM/MPI iniciado y listo para ejecutar tareas.

El entorno LAM se puede terminar de dos formas. Se puede utilizar la opción (3) del menú principal o se puede salir de KFRONT mediante la opción (0).

5. Gestión de Trabajos

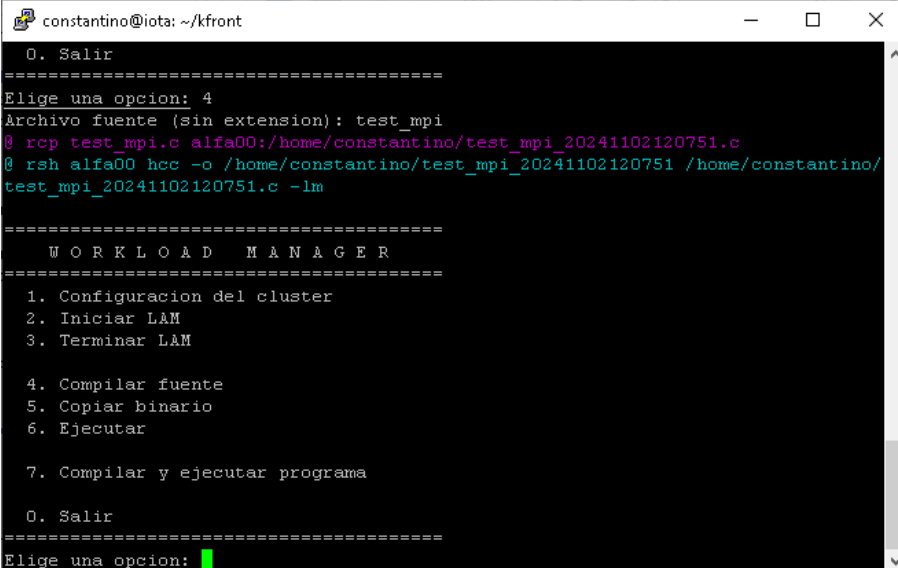
El operador dispone de las opciones (4) a (7) del menú principal para gestionar la carga de trabajos del cluster.

5.1. Compilar Trabajo

Utilizando la opción (4) del menú principal se puede enviar un programa fuente en C al nodo maestro para que lo compile. Por esta razón debe estar el entorno LAM inicializado antes de poder compilar o ejecutar un programa.

El sistema solicita al operador el nombre o ruta al archivo fuente que se desea compilar, pudiendo ingresar una ruta absoluta o relativa. *No debe indicarse* la extensión del archivo, se infiere como `.c` de forma automática. Luego, procede a copiar el archivo fuente desde la ruta indicada hacia el mismo directorio `/home` del usuario actual en el nodo maestro. Finalmente invoca al compilador e informa de cualquier error.

Por ejemplo, si se cuenta con un programa llamado `test_mpi.c`, se puede compilar en el nodo maestro de la forma que se muestra en la figura 14:



```
constantino@iota: ~/kfront
0. Salir
=====
Elige una opcion: 4
Archivo fuente (sin extension): test_mpi
@ rcp test_mpi.c alfa00:/home/constantino/test_mpi_20241102120751.c
@ rsh alfa00 hcc -o /home/constantino/test_mpi_20241102120751 /home/constantino/
test_mpi_20241102120751.c -lm
=====
      W O R K L O A D   M A N A G E R
=====
1. Configuración del cluster
2. Iniciar LAM
3. Terminar LAM

4. Compilar fuente
5. Copiar binario
6. Ejecutar

7. Compilar y ejecutar programa

0. Salir
=====
Elige una opcion: █
```

Figura 14: Compilación del programa `test_mpi.c`.

Si se llegasen a producir errores durante la compilación del programa, el binario *no se generará* y el archivo fuente será eliminado *únicamente* en el nodo maestro, con el objetivo de evitar la aglomeración de versiones anteriores de un mismo programa en el disco de dicho nodo. El sistema notificará de los errores durante cada paso del proceso, así el operador que esté más familiarizado con el lenguaje C o con LAM/MPI podrá determinar más fácilmente la causa del problema.

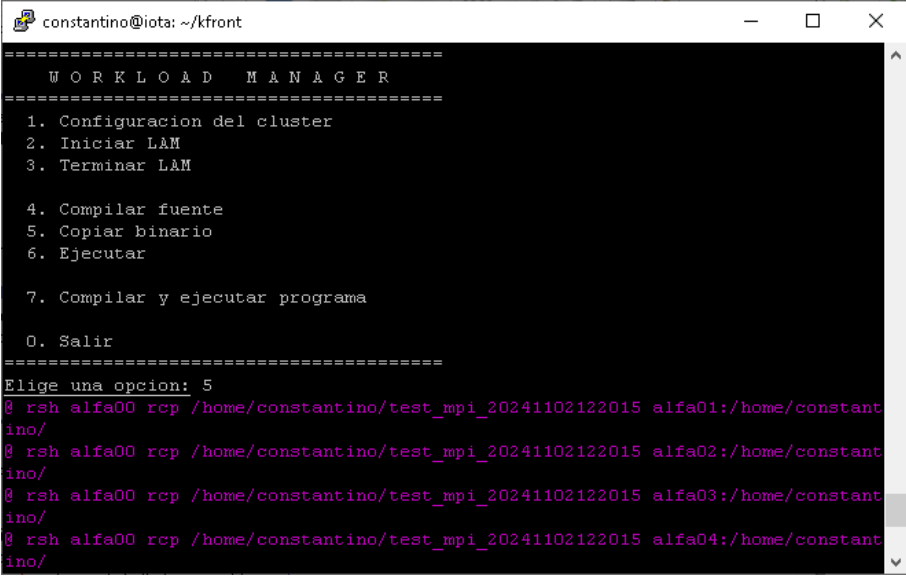
Como se mencionó anteriormente, se podría haber especificado la ruta absoluta al archivo fuente. Por ejemplo, el archivo `test_mpi.c` se ubica en `/home/usuario/`. Entonces, se le indica al sistema la ruta completa `/home/usuario/test_mpi` y este puede hallarlo sin problemas.

Si el archivo especificado no llegase a existir, el sistema informará con un mensaje de error y se volverá al menú principal.

5.2. Copiar Archivo Binario

Esta operación realiza la copia del archivo binario compilado con la opción 4 del menú principal al directorio `/home` del usuario actual en cada uno de los nodos del cluster a través del comando en segundo plano `rcp`. El único error posible es que la compilación haya fallado y no exista el binario, en cuyo caso se informará al operador y se regresará al menú principal de KFRONT.

El programa no necesita que el operador especifique ningún nombre de archivo o ruta, toma el generado por el paso anterior. Para el ejemplo de la sección anterior:



```
constantino@iota: ~/kfront
=====
W O R K L O A D   M A N A G E R
=====
1. Configuración del cluster
2. Iniciar LAM
3. Terminar LAM

4. Compilar fuente
5. Copiar binario
6. Ejecutar

7. Compilar y ejecutar programa

0. Salir
=====
Elige una opción: 5
@ rsh alfa00 rcp /home/constantino/test_mpi_20241102122015 alfa01:/home/constantino/
@ rsh alfa00 rcp /home/constantino/test_mpi_20241102122015 alfa02:/home/constantino/
@ rsh alfa00 rcp /home/constantino/test_mpi_20241102122015 alfa03:/home/constantino/
@ rsh alfa00 rcp /home/constantino/test_mpi_20241102122015 alfa04:/home/constantino/
```

Figura 15: Proceso de copia del binario generado en el paso anterior.

5.3. Ejecutar Programa

Esta operación instruye al nodo maestro para que empiece la ejecución del programa compilado con el comando `mpirun`. Verifica la existencia del binario en el maestro y realiza la copia del mismo para garantizar que, de producirse errores en la ejecución, se deben a problemas dentro del programa suministrado y no al funcionamiento de la herramienta.

Una vez copiados los archivos, el programa solicitará al usuario que ingrese los argumentos de su programa. De no requerirse, puede dejarse este campo en blanco. Se procederá con la ejecución del programa y se informará de los resultados cuando termine.

Tener en cuenta que la terminal abierta permanecerá inutilizable durante la ejecución de programa, ya que el sistema está esperando a que el programa suministrado termine de ejecutarse. Si el programa demora un tiempo significativo, abrir una segunda terminal y verificar su funcionamiento con `rsh <maestro> mpitask`, que devuelve un listado de los procesos MPI en ejecución en cada nodo. Si el listado está vacío, entonces se produjo un error grave en el entorno LAM/MPI y debe ser reiniciado manualmente.

5.4. Compilar y Ejecutar

Esta operación realiza de forma automática las operaciones de compilación, copia del binario a los nodos y puesta en ejecución del programa de usuario. Los mensajes de error que podrían observarse son los mismos que en las demás operaciones.

6. Ejemplo Completo

Para este ejemplo se cuenta con un programa `test_mpi.c` que realiza una aproximación del valor de π usando el método de Montecarlo. Este archivo se supone ubicado en el directorio `/home` de un usuario genérico.

Suponer que se invoca KFRONT con el siguiente archivo de configuración, almacenado bajo el nombre `lamhosts14` en el mismo directorio:

```
kloster00
kloster01
kloster02
kloster03
kloster04
kloster05
kloster06
kloster07
kloster08
alfa00
alfa01
alfa02
alfa03
alfa04
```

Se invoca el aplicativo mediante el comando `kfront lamhosts14`. Como KFRONT automáticamente supone que todos los nodos en la lista que sean accesibles desde la red serán usados en la ejecución, para reducir la cantidad de nodos hace falta modificar la configuración del cluster. Para ello, se selecciona la opción 1 desde el menú principal.

La configuración inicial es:

n0	kloster00	[0]	[0]	<- maestro
n1	kloster01	[0]	[0]	
n2	kloster02	[0]	[0]	
n3	kloster03	[0]	[0]	
n4	kloster04	[0]	[0]	
n5	kloster05	[0]	[0]	
n6	kloster06	[0]	[0]	
n7	kloster07	[0]	[0]	
n8	kloster08	[0]	[0]	
n9	alfa00	[0]	[0]	
n10	alfa01	[0]	[0]	
n11	alfa02	[0]	[0]	
n12	alfa03	[0]	[0]	
n13	alfa04	[0]	[0]	

Se desea ejecutar sólo en los nodos `alfa*`, por lo que deben deshabilitarse todos los nodos `kloster`. Entonces, se utiliza la operación de cambio de estado del nodo (4) 9 veces, ingresando en cada llamada el identificador correspondiente de cada nodo. Esto resulta en la siguiente configuración:

n0	kloster00	[X]	[0]
n1	kloster01	[X]	[0]
n2	kloster02	[X]	[0]
n3	kloster03	[X]	[0]
n4	kloster04	[X]	[0]
n5	kloster05	[X]	[0]
n6	kloster06	[X]	[0]

n7	kloster07	[X]	[0]	
n8	kloster08	[X]	[0]	
n9	alfa00	[0]	[0]	<- maestro
n10	alfa01	[0]	[0]	
n11	alfa02	[0]	[0]	
n12	alfa03	[0]	[0]	
n13	alfa04	[0]	[0]	

Como ya está configurado, se regresa al menú con la opción 0 y se inicia LAM con la opción 2. Al ser `alfa00` el maestro, el listado resultante será:

n0	alfa00
n1	alfa01
n2	alfa02
n3	alfa03
n4	alfa04

Ver que los identificadores del listado generado difieren de los del listado completo. Esto se debe a que el archivo `lamhosts` enviado al maestro previa la invocación de `lamboot` contiene solamente los nodos seleccionados en el paso anterior. Como se desea que los identificadores sean los mismos, se termina LAM con la opción 3 y se regresa al menú de configuración del cluster con 1. Allí se utiliza la operación de reordenamiento 1 para intercambiar los identificadores de los nodos activos con los de los nodos inactivos. Se intercambia `n9` con `n0`, `n10` con `n1`, y así sucesivamente hasta que el listado sea:

n0	alfa00	[0]	[0]	<- maestro
n1	alfa01	[0]	[0]	
n2	alfa02	[0]	[0]	
n3	alfa03	[0]	[0]	
n4	alfa04	[0]	[0]	
n5	kloster05	[X]	[0]	
n6	kloster06	[X]	[0]	
n7	kloster07	[X]	[0]	
n8	kloster08	[X]	[0]	
n9	kloster00	[X]	[0]	
n10	kloster01	[X]	[0]	
n11	kloster02	[X]	[0]	
n12	kloster03	[X]	[0]	
n13	kloster04	[X]	[0]	

Se regresa al menú principal con 0, se vuelve a iniciar LAM con 2 y ahora se puede ver que el listado devuelto por el comando `lamnodes` es idéntico al enviado como archivo de configuración por KFRONT.

Con LAM activo en `alfa00`, se procede a compilar el programa `test_mpi.c` con la operación 4 del menú principal. La figura 16 ilustra la respuesta del sistema:

```
constantino@iota: ~/kfront
=====
Elige una opcion: 4
Archivo fuente (sin extension): test_mpi
0 rcp test_mpi.c alfa00:/home/constantino/test_mpi_20241107124138.c
0 rsh alfa00 hcc -o /home/constantino/test_mpi_20241107124138 /home/constantino/
test_mpi_20241107124138.c -lm

=====
      W O R K L O A D   M A N A G E R
=====
1. Configuración del cluster
2. Iniciar LAM
3. Terminar LAM

4. Compilar fuente
5. Copiar binario
6. Ejecutar

7. Compilar y ejecutar programa

0. Salir
=====
Elige una opcion: █
```

Figura 16: Resultados de compilación de `test_mpi.c`.

Se puede ahora optar por ejecutar (5) y sólo realizar la copia de los archivos ejecutables desde el nodo maestro hacia los demás o utilizar la herramienta 6 y realizar en un único paso la copia y la llamada al proceso `mpirun`. De emplearse el primer camino, invocar luego la utilidad 6 para ejecutar el programa en el cluster.

Antes de la ejecución, el sistema solicitará al usuario la entrada de argumentos para la ejecución del programa. Éstos son de carácter opcional y puede dejarse la entrada vacía. En el caso del programa `test_mpi` no se necesitan argumentos. Al finalizar la ejecución, el sistema devolverá en pantalla la salida del programa paralelo, tal como se ilustra en la figura 17:

```
constantino@iota: ~/kfront
0. Salir
=====
Elige una opcion: 6
0 rsh alfa00 rcp /home/constantino/test_mpi_20241107124138 alfa01:/home/constant
ino/
0 rsh alfa00 rcp /home/constantino/test_mpi_20241107124138 alfa02:/home/constant
ino/
0 rsh alfa00 rcp /home/constantino/test_mpi_20241107124138 alfa03:/home/constant
ino/
0 rsh alfa00 rcp /home/constantino/test_mpi_20241107124138 alfa04:/home/constant
ino/
Argumentos del programa (opcionales):
0 rsh alfa00 mpirun C /home/constantino/test_mpi_20241107124138
3.141435 en nodo 1 con 2400000 puntos.
3.143693 en nodo 4 con 2400000 puntos.
3.139347 en nodo 3 con 2400000 puntos.
3.142813 en nodo 2 con 2400000 puntos.
3.140955 en nodo 0 con 2400000 puntos.
PI= 3.141649 (promedio) en 0.508505 seg en 5 procesadores.
0 rsh alfa00 rm -f /home/constantino/test_mpi_20241107124138*
0 rsh alfa01 rm -f /home/constantino/test_mpi_20241107124138*
0 rsh alfa02 rm -f /home/constantino/test_mpi_20241107124138*
0 rsh alfa03 rm -f /home/constantino/test_mpi_20241107124138*
0 rsh alfa04 rm -f /home/constantino/test_mpi_20241107124138*
```

Figura 17: Resultados de ejecución de `test_mpi.c`.

Supóngase ahora que se desea ejecutar el programa en los nodos `kloster00` a `kloster04` para comparar los tiempos de ejecución en un cluster homogéneo de dos arquitecturas diferentes. Para ello, se termina LAM con la operación 3 y se abre el menú de configuración del cluster.

Se habilitan los nodos `kloster00` a `kloster04` y se deshabilitan `alfa00` a `alfa04` utilizando la opción 4 del menú. Se intercambian las posiciones de los nodos tal que el resultado sea como el siguiente:

n0	kloster00	[0]	[0]	<- maestro
n1	kloster01	[0]	[0]	
n2	kloster02	[0]	[0]	
n3	kloster03	[0]	[0]	
n4	kloster04	[0]	[0]	
n5	kloster05	[X]	[0]	
n6	kloster06	[X]	[0]	
n7	kloster07	[X]	[0]	
n8	kloster08	[X]	[0]	
n9	alfa00	[X]	[0]	
n10	alfa01	[X]	[0]	
n11	alfa02	[X]	[0]	
n12	alfa03	[X]	[0]	
n13	alfa04	[X]	[0]	

Se regresa al menú principal y se inicia LAM con (2). Luego, se utiliza la opción 7 para compilar y ejecutar en un único paso. El sistema solicita:

Archivo fuente (sin extension): <se ingresa `test_mpi`>

Se procede a compilar el archivo y, al no producirse errores, a copiar el binario resultante a cada uno de los nodos del cluster. Luego, se solicita la entrada de argumentos que, como se ha mencionado, no se utilizan y puede dejarse la entrada vacía. El programa pasa luego a ejecutarse y se imprimen los resultados en la pantalla.

La figura 18 muestra los resultados de la ejecución:

```

constantino@iota: ~/kfront
@ rsh kloster00 hcc -o /home/constantino/test_mpi_20241107124950 /home/constantino/test_mpi_20241107124950.c -lm

@ rsh kloster00 rcp /home/constantino/test_mpi_20241107124950 kloster01:/home/constantino/
@ rsh kloster00 rcp /home/constantino/test_mpi_20241107124950 kloster02:/home/constantino/
@ rsh kloster00 rcp /home/constantino/test_mpi_20241107124950 kloster03:/home/constantino/
@ rsh kloster00 rcp /home/constantino/test_mpi_20241107124950 kloster04:/home/constantino/

Argumentos del programa (opcionales):
@ rsh kloster00 mpirun C /home/constantino/test_mpi_20241107124950
3.140878 en nodo 1 con 2400000 puntos.
3.141410 en nodo 2 con 2400000 puntos.
3.140762 en nodo 3 con 2400000 puntos.
3.141932 en nodo 0 con 2400000 puntos.
PI= 3.141291 (promedio) en 3.790000 seg en 5 procesadores.
3.141473 en nodo 4 con 2400000 puntos.
@ rsh kloster00 rm -f /home/constantino/test_mpi_20241107124950*
@ rsh kloster01 rm -f /home/constantino/test_mpi_20241107124950*
@ rsh kloster02 rm -f /home/constantino/test_mpi_20241107124950*
@ rsh kloster03 rm -f /home/constantino/test_mpi_20241107124950*
@ rsh kloster04 rm -f /home/constantino/test_mpi_20241107124950*

```

Figura 18: Resultados de compilación y ejecución de `test_mpi.c`.