



LABORATORIO:

Simulación y Mecánica Computacional

TRABAJO:

Instalación de un Super-Servidor de procesamiento paralelo basado en MPI

Idioma:

Español

Plataforma de Hardware: AMD64

<u>Sistema Operativo:</u>

Versión:

5.0 (Lenny)

Debian

Fecha (dd/mm/aaaa): 17/09/2009

Página 1 de 27

París 532 Haedo (1706) Buenos Aires - Argentina Teléfonos: (+ 5411) 4443-7466/4659-2575/4659-3857 Fax: 4443-0499 **www.frh.utn.edu.ar**





Prólogo.

La idea de este documento es la de guiar al usuario para que pueda, utilizando herramientas de software y hardware standard, montar un superservidor de procesamiento paralelo basado en el standard Open MPI.

Para esto se han utilizado seis PCs con procesadores AMD64 dual core interconectadas entre sí utilizando una red Ethernet de 100Mbps a través de un switch, luego haciendo funcionar una de ellas como servidor maestro y las otras cinco como servidores secundarios.

Se verá a continuación que los pasos son sencillos pudiendo implementarse en cualquier plataforma de procesadores y con cualquier cantidad de PCs.

Página 2 de 27





İ

- 1. Prerrequisitos de hardware.
 - **1.1.** Prerrequisitos mínimos.
 - **1.2.** Prerrequisitos requeridos.
- 2. Instalación del software necesario.
 - **2.1.** Instalación del sistema operativo Debian versión 5.0 (Lenny).
 - **2.2.** Instalación de paquetes requeridos en el servidor primario o maestro y en los servidores secundarios.
 - 2.2.1. Instalación de SSH (Secure Shell).
 - **2.2.2.** Instalación de RSSH (Restricted Secure Shell).
 - **2.2.3.** Instalación del paquete NFS-KERNEL-SERVER.
 - **2.2.4.** Instalación de los paquetes LIBOPENMPI-DEV, LIBOPENMPI1, OPENMPI-BIN, OPENMPI-COMMON y OPENMPI-DOC.
 - **2.2.5.** Instalación y desinstalación de otros paquetes.
 - **2.2.6.** Instalación de los paquetes en los servidores secundarios.
- 3. Configuración de servidores primario o maestro y secundarios.
 - **3.1.** Verificación de directorios y clave pública en el servidor primario o maestro.
 - **3.2.** Regeneración de archivos de claves públicas en el servidor primario o maestro.
 - **3.3.** Configuración de los servidores secundarios (viéndose algo similar al print screen).
 - **3.4.** Modificación de los archivos de configuración del servidor primario o maestro.
 - **3.5.** Prueba de conexión y configuración desde el servidor primario o maestro a los servidores secundarios.
- **4.** Probando nuestro superservidor.
 - **4.1.** Programa "Pi.c".
 - **4.2.** Compilación del programa "Pi.c" con MPICC.
 - **4.3.** Ejecución del programa "Pi.c" en forma local con MPIRUN.

Página 3 de 27





- **4.4.** Ejecución del programa "Pi.c" en forma multiproceso con MPIRUN.
- **4.5.** Visualización de los procesos ejecutados en forma paralela con TOP.
- 5. Fuentes consultadas y sitios de interés.
- 6. Autores.
- 7. Agradecimientos.

Esquema de conexión







1. Prerrequisitos de hardware.

- 1.1. <u>Prerrequisitos mínimos:</u>
 - ✓ Una PC con conexión a internet permanente para poder descargar el Small CD de Debian Lenny [<u>http://www.debian.org/distrib/netinst#smallcd</u>].
 - ✓ Grabadora de CD.
 - ✓ Un CD en blanco para grabar dicha imagen descargada.
 - ✓ Realizar la instalación de Linux Debian versión 5.0 (Lenny) en una partición adicional de la PC o instalarlo en una PC diferente [http://www.debian.org/releases/stable/i386/index.html.es].
 - ✓ Para todos los pasos anteriores conexión a internet permanente.

1.2. <u>Prerrequisitos requeridos:</u>

- ✓ Una PC con conexión a internet permanente para poder descargar el Small CD de Debian Lenny [<u>http://www.debian.org/distrib/netinst#smallcd]</u>.
- ✓ Grabadora de CD.
- ✓ Un CD en blanco para grabar dicha imagen descargada.
- Realizar la instalación de Linux Debian versión 5.0 (Lenny) en dos o más PCs [<u>http://www.debian.org/releases/stable/i386/index.html.es</u>].
- ✓ Para todos los pasos anteriores conexión a internet permanente.





2. Instalación del software necesario.

2.1. Instalación del sistema operativo Debian versión 5.0 (Lenny).

Debido a que está fuera del alcance del presente mini How To, no se indican los pasos para la instalación de Debian versión 5.0 (Lenny), en este punto se recomienda consultar el manual de instalación en el siguiente link [http://www.debian.org/releases/stable/i386/index.html.es].

2.2. Instalación de paquetes requeridos en el servidor primario o maestro y en los servidores secundarios.

Para esto utilizaremos el comando APT (Advanced Package Tool) en un terminal ingresando como usuario Root. En nuestro caso estamos utilizando la intefaz gráfica Gnome 2.22.3.

Estos paquetes deben instalarse en todas las máquinas que formen parte del superservidor, tener en cuenta que sólo una de las PCs funcionará como servidor primario o maestro (en nuestro caso la que tiene número IP 192.168.1.31) teniendo las demás PCs la función de servidores secundarios.

2.2.1. Instalación de SSH (Secure Shell).

Descripción del paquete:

SSH (Secure SHell, en español: intérprete de órdenes segura) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si tenemos un Servidor X (en sistemas Unix y Windows) corriendo.

Además de la conexión a otras máquinas, SSH nos permite copiar datos de forma segura (tanto ficheros sueltos como simular sesiones FTP cifradas), gestionar claves RSA para no escribir claves al conectar a las máquinas y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH.

La instalación de este programa es necesario para que las computadores intercambien información en forma segura.

Página 6 de 27





root@192.168.1.31:/# apt-get install openssh-server

Root Root
<u>A</u> rchivo <u>E</u> ditar <u>V</u> er <u>T</u> erminal <u>S</u> olapas Ay <u>u</u> da
<pre>root@192.168.1.31/#apt-get install openssh-server Leyendo lista de paquetes Hecho Creando árbol de dependencias Leyendo la información de estado Hecho Paquetes sugeridos: rssh Se instalarán los siguientes paquetes NUEVOS: openssh-server 0 actualizados, 1 se instalarán, 0 para eliminar y 0 no actualizados. Necesito descargar 295kB de archivos. Se utilizarán 791kB de espacio de disco adicional después de esta operación. Des:1 http://ftp.de.debian.org lenny/main openssh-server 1:5.1p1-5 [295kB] Descargados 295kB en 4s (61,1kB/s) Preconfigurando paquetes</pre>
Seleccionando el paquete openssh-server previamente no seleccionado. (Levendo la base de datos
441809 ficheros y directorios instalados actualmente.) Desempaquetando openssh-server (de/openssh-server_1%3a5.1p1-5_i386.deb) Procesando disparadores para man-db Configurando openssh-server (1:5.1p1-5) Restarting OpenBSD Secure Shell server: sshd. root@192.168.1.31:/#

2.2.2. Instalación de RSSH (Restricted Secure Shell).

Descripción del paquete:

RSSH es un shell restringido que provee acceso limitado a un server vía SSH incluyendo soporte a RDIST, RSYNC y CVS. Un ejemplo de utilización es: si se tuviese un servidor en el cual sólo se quiere que los usuarios copien ficheros con el somando SCP (Secure Copy) sin proveer acceso al shell, puede usarse RSSH para hacerlo.





root@192.168.1.31:/# apt-get install rssh

Root Root	-0×
<u>A</u> rchivo <u>E</u> ditar <u>V</u> er <u>T</u> erminal <u>S</u> olapas Ay <u>u</u> da	
<pre>root@192.168.1.31/#apt-get install rssh Leyendo lista de paquetes Hecho Creando árbol de dependencias Leyendo la información de estado Hecho Se instalarán los siguientes paquetes NUEVOS: rssh 0 actualizados, 1 se instalarán, 0 para eliminar y 0 no actualizados. Necesito descargar 49,1kB de archivos. Se utilizarán 213kB de espacio de disco adicional después de esta operación Des:1 http://ftp.de.debian.org lenny/main rssh 2.3.2-8 [49,1kB] Descargados 49,1kB en 5s (9810B/s) Preconfigurando paquetes Seleccionando el paquete rssh previamente no seleccionado. (Leyendo la base de datos 441819 ficheros y directorios instalados actualmente.) Desempaquetando rssh (de/archives/rssh_2.3.2-8_i386.deb) Procesando disparadores para man-db Configurando rssh (2.3.2-8) root@192.168.1.31:/#</pre>	
	3

2.2.3. Instalación del paquete NFS-KERNEL-SERVER.

Descripción del paquete:

MPI necesita encontrar específicamente el programa que se quiere ejecutar en los nodos (son nodos servidores primadior y secundarios) del superservidor. NFS (Net File System) provee un directorio público en cada nodo del superservidor para ejecutar el programa que el usuario quiera ejecutar.

Página 8 de 27





root@192.168.1.31:/# apt-get install nfs-kernel-server

🔲 Root 🗖 🗖	3)
<u>A</u> rchivo <u>E</u> ditar <u>V</u> er <u>T</u> erminal <u>S</u> olapas Ay <u>u</u> da	
<pre>root@192.168.1.31/#apt-get install nfs-kernel-server Leyendo lista de paquetes Hecho Creando árbol de dependencias Leyendo la información de estado Hecho Se instalarán los siguientes paquetes NUEVOS: nfs-kernel-server 0 actualizados, 1 se instalarán, 0 para eliminar y 89 no actualizados. Necesito descargar 150kB de archivos. Se utilizarán 291kB de espacio de disco adicional después de esta operación. Des:1 http://ftp.de.debian.org lenny/main nfs-kernel-server 1:1.1.2-6lenny1 [150 kB] Descargados 150kB en 2min58s (838B/s) Seleccionando el paquete nfs-kernel-server previamente no seleccionado. (Leyendo la base de datos 442894 ficheros y directorios instalados actualmente.) Desempaquetando nfs-kernel-server (de/nfs-kernel-server_1%3a1.1.2-6lenny1_i3 86.deb) Procesando disparadores para man-db Configurando nfs-kernel-server (1:1.1.2-6lenny1)</pre>	
Creating config file /etc/exports with new version Creating config file /etc/default/nfs-kernel-server with new version Starting NFS common utilities: statd. Exporting directories for NFS kernel daemon Starting NFS kernel daemon: nfsd mountd. root@192.168.1.31:/#	

2.2.4. <u>Instalación de los paquetes LIBOPENMPI-DEV, LIBOPENMPI1,</u> <u>OPENMPI-BIN, OPENMPI-COMMON y OPENMPI-DOC.</u>

Descripción de los paquetes:

MPI ("Message Passing Interface", Interfaz de Paso de Mensajes) es un estándar que define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes diseñada para ser usada en programas que exploten la existencia de múltiples procesadores.

El paso de mensajes es una técnica empleada en programación concurrente para aportar sincronización entre procesos y permitir la exclusión mutua, de manera similar a como se hace con los semáforos, monitores, etc.

Su principal característica es que no precisa de memoria compartida, por lo que es muy importante en la programación de sistemas distribuidos.

Página 9 de 27





root@192.168.1.31:/# apt-get install libopenmpi-dev libopenmpi1 openmpi-bin openmpi-common openmpi-doc

Root Root	
<u>A</u> rchivo <u>E</u> ditar <u>V</u> er <u>T</u> erminal <u>S</u> olapas Ay <u>u</u> da	
root@192.168.1.31/#apt-get install libopenmpi-dev libopenmpil openmpi-bin openmp	<u>^</u>
i-common_openmpi-doc	
Leyendo Lista de paquetes Hecho	
Creando arbol de dependencias	
Leyendo la información de estado Hecno	
openimpl-doc ya esta en su version mas reciente.	
libituerts	
s instalarán los siguientes paguetes NUEVOS:	
libibverbs1 libopenmpi-dev libopenmpi1 openmpi-bin openmpi-common	
0 actualizados, 5 se instalarán, 0 para eliminar y 5 no actualizados.	
Necesito descargar 1708kB de archivos.	
Se utilizarán 6517kB de espacio de disco adicional después de esta operación.	
¿Desea continuar [S/n]? S	
Des:1 http://ttp.de.debian.org lenny/main libibverbs1 1.1.2-1 [33,9KB]	
Des:2 http://ttp.de.debian.org_lenny/main_lioopenmpii_1/2/~7CZ-2 [1061KB]	
Des:4 http://ftp.de.debian.org/lenny/main/bernmpi-common 1.2./~rc2-2 [445kB]	
Des:5 http://ftp.de.debian.org/tenny/main/tioupenmpi-dev/i.z./~rc2-2 [++3kb]	
Descarados 1708kB en 28s (61.0kB/s)	
Seleccionando el paquete libibverbs1 previamente no seleccionado.	
(Leyendo la base de datos	
441837 ficheros y directorios instalados actualmente.)	
Desempaquetando libibverbs1 (de/libibverbs1_1.1.2-1_i386.deb)	
Seleccionando el paquete libopenmpi1 previamente no seleccionado.	
Desempaquetando libopenmpi1 (de/libopenmpi1_1.2.7~rc2-2_i386.deb)	
Seleccionando el paquete openmpi-common previamente no seleccionado.	
Desempaduetando openmp1-common (de/openmp1-common 1/2./~rC2-2 all.deb)	
Selectionalito el paquete troppeninje dev previamente no selectionado.	
Selectionando e logueta opennipi cho previamente no selectionado	
Desempaquetando openmpi-bin (de/openmpi-bin 1.2.7~rc2-2 i386.deb)	
Procesando disparadores para man-db	
Configurando libibverbsi (1.1.2-1)	
Configurando libopenmpil (1.2.7~rc2-2)	
Configurando openmpi-common (1.2.7~rc2-2)	
Configurando libopenmpi-dev (1.2.7~rc2-2)	
Configurando openmpi-bin (1.2.7~rc2-2)	
root@192.168.1.31:/#	

2.2.5. Instalación y desinstalación de otros paquetes.

<u>En nuestro sistema, aparte de los mencionados, se han instalado</u> <u>adicionalmente los siguientes paguetes (apt-get install NOMBRE-PAQUETE):</u>

- ✓ mc
- ✓ gfortran
- ✓ build-essentials
- ✓ popularity-contest
- ✓ ntp
- ✓ ntpdate
- ✓ sshguard
- ✓ autossh

Página 10 de 27





- ✓ iproute
- ✓ iptables
- ✓ smartmontools

Desinstalándose los siguientes paquetes (apt-get remove NOMBRE PAQUETE):

- ✓ network-manager
- ✓ avahi-daemon

2.2.6. Instalación de los paquetes en los servidores secundarios.

Realizar los pasos 2.2.1., 2.2.2., 2.2.3., 2.2.4. y 2.2.5. en las PCs que vayan a funcionar como servidores secundarios, en nuestro caso las PCs con números IP desde el 192.168.1.32 al 192.168.1.36.

Recordemos que los pasos son los siguientes:

- 2.2.1. root@192.168.1.32/36:/# apt-get install openssh-server
- **2.2.2.** root@192.168.1.32/36:/# apt-get install rssh
- **2.2.3.** root@192.168.1.32/36:/# apt-get install nfs-kernel-server
- **2.2.4.** root@192.168.1.32/36:/# apt-get install libopenmpi-dev libopenmpi1 openmpi-bin openmpi-common openmpi-doc
- **2.2.5.** Instalación y desinstalación de paquetes adicionales.





3. Configuración de servidores primario o maestro y secundarios.

La configuración se requiere tanto en el servidor primario o maestro como en los servidores secundarios. Esta configuración se requiere para que el sistema quede en funcionamiento.

3.1. <u>Verificación de directorios y clave pública en el servidor primario o maestro.</u>

Una vez finalizada la instalación todos los paquetes deberíamos poder ingresar al directorio "/root/.ssh" para ver su contenido <u>(viéndose algo similar al siguiente print screen):</u>

							Root		
<u>A</u> rchivo	<u>E</u> ditar	⊻er	Termir	nal <u>s</u>	olapa	as	Ay <u>u</u> da		
root@192	2.168.	1.31:/	# cd rc	oot				^	1
root@192	2.168.3	1.31://	~# cd	.ssh					1
root@192	2.168.	1.31:~	·/.ssh#	‡ s -la	a)				1
total 20									1
drwx	2	root	root	4096	sep	23	20:56	90	
drwxr-xr	-x 52	root	root	4096	ago	29	15:53		1
- rw	1	root	root	1675	sep	23	21:00	id_rsa	1
- rw-rr	1	root	root	402	sep	23	21:00	id_rsa.pub	1
- rw- r r	1	root	root	1768	may	12	19:54	known_hosts	1
root@192	2.168.3	1.31:~	·/.ssh#	ŧ					1
									1
									1
									f
									J

Página 12 de 27





3.2. <u>Regeneración de archivos de clave pública en el servidor primario</u> <u>o maestro.</u>

Este paso sólo debe hacerse en el servidor primario o maestro. Si bien dentro del directorio "/root/.ssh" en el servidor primario (IP 192.168.1.31) podemos ver la existencia de los archivos "id_rsa"e "id_rsa.pub", que son los que contienen la clave pública para acceder al mismo en forma segura utilizando SSH, generaremos nuevamente las mismas. Esta clave es las que servirá luego para poder hacer que las PCs que funcionan como servidores secundarios se conecten al servidor primario en forma segura y sin necesidad de tipear ningún password para ello (Passwordless Connection).

Regeneración de los archivos de clave pública en el servidor primario o maestro (viéndose algo similar al print screen):

root@192.168.1.31:/# ssh-keygen -t rsa

Root 🗖	
<u>A</u> rchivo <u>E</u> ditar <u>V</u> er <u>T</u> erminal <u>S</u> olapas Ay <u>u</u> da	
<pre>root@192.168.1.31~/.ssh#ssh-keygen -t rsa Generating public/private rsa key pair. Enter file in which to save the key (/root/.ssh/id_rsa): /root/.ssh/id_rsa already exists. Overwrite (y/n)? y Enter passphrase (empty for no passphrase): Enter same passphrase again: Your identification has been saved in /root/.ssh/id_rsa. Your public key has been saved in /root/.ssh/id_rsa.pub. The key fingerprint is: 19:10:36:33:b1:91:50:d4:cf:53:95:00:77:fd:e4:6d root@debian-notebook The key's randomart image is: +[RSA 2048]++ .+%=0000 B0 0. 00 0+ E S </pre>	
root@192.168.1.31:~/.ssh#	~

Página 13 de 27





3.3. <u>Configuración de los servidores secundarios (viéndose algo similar al print screen).</u>

Una vez generado el archivo que contiene la clave pública (id_rsa.pub) se copia a las demás PCs o servidores secundarios (en nuestro caso números IP desde el 192.168.1.32 al 192.168.1.36) para que los mismos puedan acceder a ejecutar los comandos que el usuario necesite sin requerir password (Passwordless Connection).

Este paso debe repetirse por cada servidor secundario, por lo que la sintaxis vista en el print screen más abajo cambiará en su segunda parte de "root@192.168.1.32:/home/nombre-usuario/id_rsa.server.pub" a "root@192.168.1.33:/home/nombre-usuario/id_rsa.server.pub" y así sucesivamente hasta llegar al número IP 192.168.1.36.

Root Root	
<u>A</u> rchivo <u>E</u> ditar <u>V</u> er <u>T</u> erminal <u>S</u> olapas Ay <u>u</u> da	
root@192.168.1.31/#scp_root@192.168.1.31:/root/.ssh/id_rsa.pub_root@192.1	68.1.32 🔿
:/home/nombre-usuario/id_rsa.server.pub	

Página 14 de 27





3.4. <u>Modificación de los archivos de configuración del servidor primario</u> <u>o maestro.</u>

Debemos modificar en el servidor primario o maestro el archivo "/etc/openmpi/openmpi-default-hostfile". Este archivo es el que contiene tanto los números IP de todos los nodos de nuestro superservidor (recordar que nodo es tanto el servidor primario como los secundarios) como la cantidad de procesadores que tiene cada nodo es de 2 pondremos este valor.



Página 15 de 27





uiente al archivo:
slots=2

		Root		
<u>A</u> rchivo <u>E</u> ditar <u>V</u> er	<u>T</u> erminal <u>S</u> olapas	Ay <u>u</u> da		
GNU nano 2.0.7	Fichero: ope	enmpi-default-hos	stfile	Modificado
<pre># contain hosts if # the same set of o # (such as SLURM, F # # If you are priman # should *not* simp # implementations, # created by the R/ # components were a # be disabled by ex # the value "^local # parameter).</pre>	a system adminis default hosts, an PBS, etc.). rily interested i oly list "localho such as LAM/MPI) AS component name able to find any ccluding the loca lhost" [without f	strator wants use nd is not using a ost" in here (cor). A localhost-c ed "localhost" if hosts to run on alhost RAS compor the quotes] to th	ers to always hav a batch scheduler MPI on one node, ntrary to prior M only node list is f no other RAS (this behavior c nent by specifyin ne "ras" MCA	e you PI an g
192.168.2.31 slo 192.168.2.32 slo 192.168.2.33 slo 192.168.2.34 slo 192.168.2.35 slo	ots=2 ots=2 ots=2 ots=2 ots=2_			
192.168.2.36 slo ^G Ver ayuda ^O Gua ^X Salir ^J Jus	ots=2 ardar [^] R Leer F stificar [^] W Busca	Fich <mark>^Y</mark> Pág Ant r <u>^V</u> Pág Sig	^K CortarTxt ^C ^U PegarTxt ^T	Pos actual Ortografía 🖂

Como dato adicional y a modo de ejemplo mostraremos como deben modificarse los siguientes archivos, agregando lo que figura debajo de cada título (no entraremos en detalle en cuanto a la explicación de cada uno respecta debido a que forman parte de la configuración standard de la red del sistema operativo Linux Debian versión 5.0 (Lenny)):

Archivo "/etc/resolv.conf":

domain lan.frh.utn.edu.ar

Página 16 de 27





search	lan.frh.utn.edu.ar	frh.utn.edu.ar
search	lan.frh.utn.edu.ar	frh.utn.edu.a

nameserver 192.168.2.1

nameserver 170.210.17.154

Archivo "/etc/hosts":

symc01	symc01.nn.frh.utn.edu.ar	192.168.2.31
symc02	symc02.nn.frh.utn.edu.ar	192.168.2.32
symc03	symc03.nn.frh.utn.edu.ar	192.168.2.33
symc04	symc04.nn.frh.utn.edu.ar	192.168.2.34
symc05	symc05.nn.frh.utn.edu.ar	192.168.2.35
symc06	symc06.nn.frh.utn.edu.ar	192.168.2.36

Archivo "/etc/network/interfaces":

iface eth0 inet static				
address	192.168.2.31			
netmask	255.255.255.0			
gateway	192.168.2.1			

3.5. Prueba de conexión y configuración desde el servidor primario o

Página 17 de 27





maestro a los servidores secundarios.

Ahora deberemos realizar los siguientes pasos en cada servidor secundario (en nuestro casos desde el IP 192.168.1.32 al IP 192.168.1.36):

- Realizamos una prueba de conexión vía SSH para ver si funciona la conectividad sin password (Passwordless Connection) desde el servidor primario (IP 192.168.1.31) al primer servidor secundario (IP 192.168.1.32). Una vez que SSH encuentre el servidor secundario nos solicitará el password del usuario ROOT por lo que deberemos escribirlo para ingresar al mismo.
- ✓ Modificar los permisos del directorio "/root/.ssh" poniéndolos en 700.
- ✓ Agregar el pasword público al archivo "/root/.ssh/authorized_keys".
- ✓ Cambiar los permisos del archivo "authorized_keys" poniéndolos en 644.

Una vez finalizados estos pasos deberemos hacer lo mismo con los demás



servidores secundarios viéndose algo similar al siguiente print screen:

Página 18 de 27





4. Probando nuestro superservidor.

Asumimos en este paso que ya deberíamos tener instalado apropiadamente nuestro superservidor con capacidades de procesamiento paralelo del tipo MPI, ahora usted pude probar el mismo con un programa de testeo que calculará el número PI.

El programa a utilizarse será el contenido en el archivo "Pi.c", el mismo es un programa simple escrito en lenguaje "C' que reporta el núemro PI de cada nodo sobre el superservidor usando los servicios MPI.

4.1. Para esto deberemos descargar el archivo "Pi.c" del link http://www.ps3cluster.org/distros/pi.c directorio У ponerlo en el "openmpi"en el servidor primario o maestro (se adjunta más abajo el programa del sitio original para que pueda guardarse en un archivo que deberá llamarse "Pi.c").

COMIENZO DEL PROGRAMA.

/* To run this program:							
/*	*/						
/*	*/						
/*	*/						
/* Issue: time mpirun -np [nprocs] ./p	oi (SGI, Beowulf) */						
/*	*/						
/*	*/						
/*	*/						

#include <stdio.h>

#include <stdlib.h>

Página 19 de 27





#include "mpi.h"

```
int main(int argc, char *argv[])
```

{

int i, n; double h, pi, x;

int me, nprocs;

double piece;

/* ------ */

MPI_Init (&argc, &argv);

MPI_Comm_size (MPI_COMM_WORLD, &nprocs);

MPI_Comm_rank (MPI_COMM_WORLD, &me);

/* ------ */

```
if (me == 0)
{
    printf("%s", "Input number of intervals:\n");
    scanf ("%d", &n);
}
```

Página 20 de 27





```
/* ------ */
```

MPI_Bcast (&n, 1, MPI_INT,

0, MPI_COMM_WORLD);

/* ------ */

h = 1. / (double) n;

piece = 0.;

for $(i=me+1; i \le n; i+=nprocs)$

{

 $x = (i-1)^{*}h;$

```
piece = piece + (
```

```
4/
(1+(x)*(x))
+
4/
(1+(x+h)*(x+h))
) / 2 * h;
```

}

printf("%d: pi = %25.15f\n", me, piece);

Página 21 de 27





return 0;

} FIN DEL PROGRAMA.

Página 22 de 27





4.2. Compilar el programa utilizando el comando "mpicc". Como resultado se generará un archivo binario ejecutable llamado "Pi".

	Root
<u>Archivo Editar Ver Terminal Solapas</u>	Ay <u>u</u> da
root@192.168.1.31:~# mpicc Pi Pi.c	<u>^</u>

Página 23 de 27





4.3. Para ejecutar el programa "Pi" en forma local solo basta con tipear "root@192.168.1.31:~# mpirun -np 2 Pi", el número 2 denota la cantidad de procesos que se utilizarán cuando se ejecute el programa.

	Root
<u>Archivo E</u> ditar <u>V</u> er <u>T</u> erminal <u>S</u> olapas	Ayuda
root@192.168.1.32:~# mpicc -o Pi Pi.c	
Input number of intervals:	
0: pi = 1.580762993461627 1: pi = 1.560762993461627 pi = 3.141525986923254	
root@192.168.1.31:~#	
	<u>s</u>

Página 24 de 27





4.4. Para realizar la ejecución del programa "Pi"en nuestro superservidor ahora deberemos realizar una copia del programa "Pi"en cada directorio compartido de nuestro "NFS" de cada nodo. Como paso posterior ejecutaremos desde el servidor primario nuevamente nuestro programa tecleando "root@192.168.1.31:~# mpirun -np 12 Pi". Vemos que se ha aumentado la cantidad de procesos (slots), esto es debido a que ahora estaremos realizando la ejecución con mayor cantidad de procesadores (12 en total).

					Root	
<u>A</u> rchivo	<u>E</u> ditar	⊻er	<u>T</u> erminal	<u>S</u> olapas	Ay <u>u</u> da	
root@19	2.168.1	.31:~	- # mpirun	-np 12 Pi	Pi	~
Input nu	mber o	fin	tervals:			
50						
0: pi =		Θ.	313192435	423665		
1: pi =		Θ.	309570899	760436		
2: pi =		Θ.2	266261746	100482		
10: pi =		Θ	.24023882	0018239		
5: pi =		Θ.:	257014850	811027		
4: pi =		Θ.2	260178999	022806		
3: pi =		Θ.:	263263552	108575		
11: pi =	: }	Θ	.23673988	3157596		
9: pi =		Θ.	243698753	265200		
7: pi =		Θ.:	250475054	358792		
8: pi =		Θ.	247113069	262194		
6: pi =		Θ.	253777923	634241		
pi =	3	.141	525986923	253		
root@19	2.168.1	.31:~	-#			
						3

Página 25 de 27





4.5. Para visualizar la cantidad de procesos utilizaremos el comando "TOP". Aquí veremos los datos de cada proceso paralelo ejecutado.

					_	Root						
Archive	<u>E</u> ditar	⊻er <u>T</u> e	erminal	<u>S</u> olap	as	Ay <u>u</u> da						
Root						×	Root					X
top -	22:29:30)up 5	:21, 4	4 user	s,	load	average	e: 51.	57,	24.75,	9.70	2
Tasks:	227 tot	tal, 5	2 runna	ing, 1	75 s	leepi	ng, G) stop	oped,	0 Z0	ombie	
Cpu(s)	57.0%	15, 43.0	0%sy,	0.0%n	i,	0.0%1	d, 0.0	%wa,	0.2	%hi, €	0%si,	0.0%st 🗕
Mem:	2067112	2k tota	l, 147	70548k	use	ed,	596564	(free		39404k	buffers	5
Swap:	2658716	ik tota	ι,	764k	use	ed, 2	657952	(free	2,	519956k	cached	
	0070		0.0011	0.14514		117 0	11757	61.14 D			CONNUND	
PID	PPID	TIME+	%CPU	%MEM	PR	NI S	VIRT	SWAP	RES	UID	COMMAND	
28264	28214	0:01.0	B 5	0.2	20	ΘR	518m	514m	3936	Θ	pi	
28263	28214	0:00.9	B 3	0.2	20	ΘR	518m	514m	3932	Θ	pi	
28262	28214	0:01.0	93	0.2	20	ΘR	518m	514m	3932	Θ	pi	
28261	28214	0:00.98	B 3	0.2	20	0 R	518m	514m	3908	Θ	pi	
28260	28214	0:01.0	2 4	0.2	20	0 R	518m	514m	3920	Θ	pi	
28259	28214	0:01.1	94	0.2	20	0 R	518m	514m	3920	Θ	pi	
28259	28214	0:01.3	2 4	0.2	20	0 R	518m	514m	3920	Θ	pi	
28258	28214	0:01.3	0 4	0.2	20	0 R	518m	514m	3912	Θ	pi	
28257	28214	0:01.28	84	0.2	20	0 R	518m	514m	3920	Θ	pi	
28256	28214	0:01.1	84	0.2	20	ΘR	518m	514m	3916	Θ	pi	
28255	28214	0:01.3	2 4	0.2	20	ΘR	518m	514m	3932	Θ	pi	
28254	28214	0:01.3	4 3	0.2	20	ΘR	518m	514m	3928	Θ	pi	
28253	28214	0:01.3	9 4	0.2	20	ΘR	518m	514m	3952	Θ	pi	
28252	28214	0:01.4	4 4	0.2	20	ΘR	518m	514m	3928	Θ	pi	
28251	28214	0:01.4	4 4	0.2	20	ΘR	518m	514m	3912	Θ	pi	
28250	28214	0:01.4	9 4	0.2	20	ΘR	518m	514m	3920	Θ	pi	
28249	28214	0:01.3	2 4	0.2	20	0 R	518m	514m	3920	Θ	pi	3





Fuentes consultadas y sitios de interés.

http://www.debian.org/ http://www.open-mpi.org/ http://www.ps3cluster.umassd.edu/index.html http://www.philchen.com/2007/07/28/how-to-enable-passwordlessauthentication-with-ssh

6. Autores.

Alumnos de la Universidad Tecnológica Nacional – Facultad Regional Haedo – Laboratorio de Simulación y Mecánica Computacional:

- ✓ Andrés Trapanotto.
- ✓ Antonio Sebastián Rodríguez Capello.

7. Agradecimientos.

Profesores de la Universidad Tecnológica Nacional – Facultad Regional Haedo – Laboratorio de Simulación y Mecánica Computacional:

- ✓ Ing. Carlos Carlassare.
- ✓ Ing. Miguel Bavaro.
- ✓ Ing. Juan C. Polidoro.

Página 27 de 27