



tux.cl  
sitio de tux en chile

[Historia](#)

[TuxTips](#)

[Artículos](#)

[Eventos](#)

[Screenshots](#)

[Info](#)

[Links](#)

[Contacto](#)

**100% Libre de M\$**

[Principal](#) > [Artículos](#) > [clusters](#)

# Instalando un Cluster tipo MOSIX

Ricardo Muñoz A.  
rmunoz@tux.cl

MOSIX es una extensión del kernel de Linux que permite ejecutar aplicaciones "normales" (no paralelizadas) en un Cluster. Una de las posibilidades de MOSIX es la "migración de procesos", que permite migrar procesos de nodo en nodo. Si por ejemplo, cierto proceso está dominando la carga de un nodo, este será movido a otro que tiene más recursos.

A diferencia de otros tipos de Clusters, MOSIX puede ser usado en PCs de distintas características, de hecho en la instalación realizada se utilizó un Athlon, un Duron y un K6-2, todos procesadores AMD (fue una simple coincidencia...no tenemos nada contra Intel...excepto sus precios).

Se podría "planificar" a las estaciones de trabajo que tiene MOSIX, para que estos se activen periódicamente a cierta hora. Por ejemplo en una empresa donde las estaciones de trabajo se dejan de usar después de las 20:00 (todas se han ido a sus casas...excepto los trabajajicos) los nodos pueden ser usados por los servidores que realizan algún "proceso nocturno" (procesamiento, carga de datos, reindexación, etc.).

## 1. Cómo funciona y para que sirve MOSIX

Una de las características de MOSIX es que, a diferencia de otros clusters, no es necesario modificar las aplicaciones ni tampoco utilizar librerías especiales. De hecho, tampoco es necesario asignar "a mano" los procesos a los diferentes nodos que componen el cluster.

La idea es que después de la creación de un nuevo proceso (fork), MOSIX intenta asignarlo al mejor nodo disponible en ese entonces. MOSIX estará constantemente monitoreando los procesos, y si fuera necesario, migrará un proceso entre los nodos para maximizar el rendimiento promedio.

MOSIX realiza todo esto automáticamente, bajo el concepto de "**fork and forget**" al igual que en un sistema SMP. Esto significa que sólo algunas aplicaciones se beneficiarán de un cluster MOSIX, básicamente:

- Procesos que requieren de mucha CPU, aplicaciones científicas, de ingeniería, etc.
- Procesos paralelos, especialmente los que tienen tiempos de ejecución impredecibles.
- Clusters con nodos de diferentes velocidades y/o distintas cantidades de memoria.
- Entornos multi-usuario y de tiempo compartido.
- Servidores WEB escalables.

MOSIX funciona silenciosamente. Sus operaciones son transparentes para las aplicaciones. Los usuarios no necesitan saber **dónde** se están ejecutando los procesos, tampoco necesitan preocuparse de lo que están haciendo otros usuarios.

Como MOSIX está implementado en el kernel de Linux, sus operaciones son totalmente transparentes para las aplicaciones. Esto permite definir distintos tipos de clusters, incluso un cluster con diferentes CPU's o velocidades LAN.

Otra característica de MOSIX, es que sus algoritmos son descentralizados - esto significa que cada nodo puede ser el maestro de los procesos creados localmente, y un servidor de los procesos remotos que migraron desde otros nodos. Esto permite agregar o remover nodos desde el cluster en cualquier momento.

## 2. Preparando la Instalación

Los nodos destinados a MOSIX deben tener instalados Linux (obvio...). La instalación automática de MOSIX es compatible con RedHat 5.1, 6.0, 6.2, 7.0, 7.1 y SuSE 6.0, 6.1, 6.2, 6.3, 7.0. En caso de preferir otra distribución, simplemente se debería seguir un procedimiento de instalación manual que se encuentra documentado en los fuentes de MOSIX, siempre y cuando se cumplan los requisitos: **make** versión 3.77 o superior, **gcc** versión 2.95.3, 2.95.4 o 2.96.74+, y **binutils** 2.9.1.0.25 o superior.

Para la instalación necesitamos: el parche MOSIX + el código fuente del kernel de Linux. En general, cierta versión de MOSIX requiere de cierta versión de Linux.

Lo ideal es que todos los nodos del cluster tengan la misma versión de MOSIX. El requisito mínimo es que

los primeros dos dígitos de la versión sean los mismos para todos los nodos.

Cuando hicimos las pruebas instalamos la versión 1.5.2 de MOSIX, que ocupa ~400Kb y se puede bajar desde <http://www.mosix.cs.huji.ac.il/ftps/MOSIX-1.5.2.tar.gz>. Esta versión requiere del kernel 2.4.13, cuyos fuentes (~28Mb) se pueden bajar desde <ftp.mirror.psinet.cl>.

Si se realiza una instalación automática no es necesario descomprimir los fuentes del kernel de Linux, pero si por algún motivo se desea descomprimirlos esto puede hacerse en `/usr/src/linux-2.4.13`.

**OBS!** Es recomendable hacer un respaldo completo del sistema antes de instalar MOSIX. De hecho recomiendo instalar de nuevo Linux en otra partición o disco duro, donde se pueda probar libremente y recuerden que "hechando a perder se aprende!".

### 3. Instalación, configuración y puesta en marcha de un Nodo

Instalación automática:

1. Descomprimir el archivo MOSIX-1.5.2.tar.gz:

```
tar xzf MOSIX-1.5.2.tar.gz
```

2. Ejecutar:

```
cd MOSIX-1.5.2
./mosix.install
```

**mosix.install** nos guiará durante la configuración, compilación del kernel MOSIX y la posterior instalación.

3. Durante la instalación, **mosix.install** modifica varios archivos de configuración. Para deshacer estas modificaciones, los archivos originales son guardados con la extensión ".pre\_mosix". Los siguientes son los archivos modificados durante la instalación:

```
/etc/inittab
/etc/inetd.conf y/o /etc/xinetd.d/*
/etc/lilo.conf
/etc/rc.d/init.d/atd
/etc/cron.daily/slocate.cron
```

Si se desea desinstalar MOSIX (usando `./mosix.install --uninstall`), estos archivos serán restaurados con su contenido original.

### 4. Los resultados

La instalación que realizamos fue de un cluster MOSIX de 3 nodos, cada uno con las siguientes características:

Procesador	RAM	DD	Distro
AMD Athlon 1.2Ghz	756MB	30GB	Red Hat 7.1
AMD Duron 750Mhz	128MB	01GB	Red Hat 7.1
AMD K6-2 350 Mhz	96MB	01GB	Red Hat 7.1

El 1er nodo que se configuró fue el Athlon. Se procedió a instalar (en una partición nueva) la distro Red Hat 7.1 incluyendo los paquetes de desarrollo. Luego se siguieron los pasos descritos en la sección 3. Para el 2do y 3er nodo se utilizaron 2 discos duros viejos de 1GB. También se instaló la distro Red Hat 7.1 y se siguieron los pasos correspondientes (esto se pudo haber ahorrado, ver sección 6).

La instalación automática incluye la modificación al LILO, por lo que una vez finalizada la instalación y reiniciado el equipo, basta con elegir la opción linux-mosix desde el menú del LILO, para poder cargar el kernel MOSIX recién compilado.

Las pruebas que hicimos fueron bastante básicas, esto por razones de tiempo y de preparación. Nuestro X-Team programó una pequeña rutina (el proceso X!) que únicamente ejecutaba un ciclo infinito sin mostrar

nada en pantalla. Al ser ejecutada en un Linux "normal", esta rutina ocupa el 100% de la CPU. La idea era ver que pasaba si ese mismo proceso se ejecuta en un cluster...

Dentro de las aplicaciones que trae MOSIX, se encuentra el programa **mon** que se ejecuta en modo texto (consola) y que permite monitoriar la carga de los nodos del cluster, obtener información acerca de las velocidades de los nodos, cantidad de memoria, etc.

Usando el **mon** pudimos apreciar lo siguiente:

- al ejecutar nuestra rutina X! en el nodo más lento, MOSIX migraba el proceso hacía el nodo más rapido, quedando el nodo lento sin carga(!).
- al ejecutar la rutina en el nodo más rápido, toda la carga se quedaba en el mismo nodo, es decir, MOSIX no migraba el proceso hacia otros nodos.
- al modificar nuestra rutina X! agregándole una instrucción printf() y repetir las pruebas anteriores, no se pudo apreciar ningún tipo de migración entre nodos. Esto tiene una explicación bien simple, como la instrucción printf() se encarga de mostrar un texto en pantalla, este texto debe ser mostrado en el nodo desde el cual se está ejecutando la rutina, por lo tanto el proceso nunca será migrado ya que el procesamiento que requiere es mínimo (recuerden que sólo es un ciclo infinito) por lo tanto dedicará todo su tiempo a mostrar texto en pantalla...

Con estos pocos resultados y la experiencia de la instalación, hemos llegado a ciertas conclusiones...

## 5. Conclusiones

La instalación de MOSIX es muy simple. Si se usa una distro como Red Hat se puede hacer uso del instalador automático, que además de facilitar la configuración, hace que todo el proceso de instalación sea más rápido. (De hecho, nos demoramos ~ 2hrs para configurar, compilar e instalar los 3 nodos de nuestro cluster).

Con las pruebas que realizamos, nos dimos cuenta que MOSIX hace un muy buen trabajo al migrar procesos que realizan muchos cálculos o dicho de otra manera hacen mucho uso de la CPU. Los procesos más "interactivos" no sacarán mucho provecho de un cluster tipo MOSIX.

Finalmente, creo que MOSIX es una buena alternativa para el ámbito de la investigación (dónde se requiere procesar un gran volumen de datos), el ámbito académico (dónde otras alternativas son mucho más complejas de comprender/usar) y en general cuando se tiene bien claro los resultados que se desean y cómo lograrlos... (cuando teníamos el cluster funcionando nos dimos cuenta que en realidad lo más difícil es darle un uso...).

## 6. Lo que faltó probar

Por razones de tiempo nos faltó probar las siguientes herramientas (teníamos los ejecutables pero no la documentación...)

- **Clump/OS**. Mini-distribución Linux/MOSIX diseñada para permitir agregar rápidamente nodos a un cluster MOSIX. La versión más reciente es la 4 y consta de una imagen ISO de 5.3MB. Al momento de bootear el CD de ClumpOS intentará automáticamente detectar la tarjeta de red y configurarla mediante DHCP. Si la detección/configuración tuvo éxito, se creará un archivo `mosix.map` con la configuración de todos los nodos (suponiendo red clase C) y luego se mostrará una pantalla SVGA indicando el estado de todo los nodos del cluster.
- **MosixView**. Interfaz gráfica (GTK) para la administración de un cluster MOSIX. Se pueden ajustar los principales parámetros del cluster como por ejemplo la velocidad de los nodos. Es un front-end para los comandos "mosctl". [Aquí](#) hay un screenshot de cuando lo probamos con nuestro cluster.

Pueden revisar las [fotos](#) que sacamos cuando instalamos el cluster.



