

# Clúster de Alto Retorno de Inversión para Renderización

Christian de Jesús Cruz Aguilar

Paulina Fabiola Espinosa Celis

José Enrique Álvarez Estrada

4 de diciembre de 2010

Universidad del Caribe, Cancún Quintana Roo, México

## Resumen

El constante crecimiento en la industria multimedia ha provocado la aparición de cada vez más empresas. Se trata de un negocio que requiere de tiempos de procesamiento computacional considerables, llegando en ocasiones a requerir de grandes equipos de cómputo de alto costo. En nuestro país son pocas las empresas que cuentan con tales equipos, y escasos los arrendadores de los mismos.

En el presente artículo se implementa un clúster para renderizado como propuesta de un modelo de negocio, que permita a una institución universitaria lograr un alto retorno de inversión de sus equipos, mediante el aprovechamiento de este nicho de mercado, a través de la implementación de un cluster para renderización que se ofrezca en arrendamiento a quienes lo requieran.

## 1. Introducción

Hoy en día son muchos los aspectos de nuestra vida que se ven influenciados por la tecnología de las computadoras. La televisión y el cine son una muestra clara de ello. Esto ha llevado a la aparición de empresas dedicadas a la animación, y con los avances tecnológicos que se han presentado en las últimas décadas, se ha podido mejorar la

calidad de las animaciones y los costos de producción se han abaratado para este tipo de trabajos.

De acuerdo a [ULPC], el renderizado es el proceso final por el que pasa un gráfico para generar una imagen en 2D, 3D o una animación a partir de una escena. Este proceso, llevado a cabo en una computadora común, implica un tiempo de procesamiento considerable, independientemente del motor de renderizado que se utilice.

Tres ejemplos emblemáticos de compañías mexicanas dedicadas a la industria de la animación en 2D y 3D son Anima Estudios (EL chavo del 8 y AAA: La película), Animex (La Leyenda de la Nahuala, Nicté y Roncho, el perro mala pata) y La Compañía Huevo (Huevocartoon).

A comparación de la industria de la animación norteamericana, por ejemplo, en nuestro país aún son pocas las compañías que se dediquen a tal empresa. Tal vez ello se deba a los costos que genera la renderización, y a la mano de obra especializada que se requiere para instrumentar los clústers necesarios.

Tomando en cuenta el auge que la industria de la animación ha tenido en los últimos años (comerciales, series animadas, películas, etc.), se ha convertido en un mercado potencial para el computo de alto rendimiento, ya que los tiempos y costos de renderizado son muy importantes al momento de crear estas producciones. De acuerdo a [TECHNOARQUITECTURA], tomando de base un computador Intel-2CoreDuo con 4 GB de RAM, renderizar un frame de 640x480 pixeles, con V-Ray, con una configuración media alta de render, toma de 35 a 45 minutos aproximadamente. Esto quiere decir que si se renderiza una toma de un minuto de duración se requiere de:

$$(60 \text{ segundos/toma})(24 \text{ frames/segundo})(40 \text{ minutos/frame}) = 57,600 \text{ minutos/toma}$$

$$(57,600 \text{ minutos/toma})(1 \text{ hora/ 60minutos}) = 960 \text{ horas/toma}$$

$$(960 \text{ horas/toma})(1 \text{ día/24 horas}) = 40 \text{ días/toma}$$

Por tanto, resulta inimaginable -desde el punto de vista temporal y económico- realizar un película completa en una sola máquina.

Dentro del área del diseño y la arquitectura la situación es similar, ya que las exigencias del mercado implican agregar más realismo a las imágenes, esto conlleva más tiempo de procesamiento por cada imagen al momento del renderizado.

Si bien existen supercomputadoras que ofrecen capacidades de cómputo suficiente para resolver problemas de cálculos matemáticos complejos, la inversión que requieren es muy elevada. Es por ello que reemplazarlas por clústers dedicados a trabajos de renderización,

sea una alternativa viable para este tipo de tareas, puesto que al aprovechar la tecnología existente se evitan tales inversiones.

El propósito de este proyecto tecnológico es aprovechar esta oportunidad de mercado, implementando un sistema de cómputo de alto rendimiento (clúster) basado en software libre y destinado a tareas de renderización. Con ello se pretende reducir los costos de producción, y hacer accesible esta tecnología a empresas de animación locales que no cuentan con ella.

Algunas empresas que existen en el medio son:

MarkSoft Solution. Ubicada en Cuernavaca, Morelos ofrece servicios de renderizado con las siguientes plataformas: 3D Studio Max, Maya, Blender, Cinema 4D, XSI, After Effects; y los siguientes motores de render: V-Ray, Mental Ray, Maxwell. Su granja está formada por 10 equipos con las siguientes características:

- Velocidad Total: 16.62 GHz
- Total RAM: 11056 MB
- Costo: 25 dólares/hora

Rengo Modeling Renders. Ubicada en México, Distrito Federal, ofrece servicios de renders fotorrealistas, render conceptuales, fotomontajes, modelos 3D y recorridos virtuales. Esta empresa ofrece un clúster de 5 CPU's (por el momento se desconoce las características), los costos por renderizado varían de acuerdo a la velocidad de procesamiento de un frame/segundo, con un precio máximo de \$200.00 pesos por seg/render. Es importante mencionar que, debido a su pequeño tamaño, el clúster anterior se apoya en una RenderFarm extranjera, ubicada en España. Un factor más que demuestra el déficit de servicios similares en nuestro país, y si consideramos que una escuela cuanta en promedio con dos laboratorios, cada uno con al menos 10 equipos de cómputo, se tiene un clúster teóricamente competitivo, capaz de ofrecer los servicios de renderizado a micro, pequeñas y medianas empresas.

## 2. Clúster de Alto Retorno de Inversión para Renderización

### 2.1. Propuesta

Por lo antes mencionado, nuestra propuesta pretende explotar aquellos equipos de cómputo presentes en los laboratorios de las IES, durante aquellos periodos de tiempo en que se encuentran inactivos, integrándolos en clusters para la maquila de renderizado a empresas que así lo requieran.

El motivo de plantearlo para una institución universitaria es por el retorno de inversión. De acuerdo a [ALVAREZ] en las instituciones de educación superior el uso que se les da a los equipos de cómputo es de máximo 16 horas al día, que en una semana hace un total de 80 horas. El calendario escolar es de 40 semanas al año. Por otro lado si tomamos en cuenta el tiempo de vida útil que tiene un equipo de cómputo, que es de unos 5 años, y que por la ley de Moore este plazo se reduce a año y medio, el tiempo de uso real de un equipo de cómputo en una institución universitaria es el siguiente:

$$(16 \text{ horas/día})(5 \text{ días/semana})(40 \text{ semanas/año})(1.5 \text{ años/equipo}) = 4800 \text{ horas/equipo.}$$

Si se hiciera uso de ellos las 24 horas al día, los 7 días de la semana, las 52 semanas al año y teniendo en cuenta que el de funcionamiento del equipo sería de un 99

$$(24 \text{ horas/día})(7 \text{ días/semana})(52 \text{ semanas/año})(1.5 \text{ años/equipo})(99 \%) = 12,973 \text{ horas/equipo.}$$

De donde resulta que la eficiencia de dichos equipos es de:

$$(4,800 \text{ horas/equipo} / 12,973 \text{ horas/equipo})(100) = 37 \%$$

que claramente no es la óptima. El retorno de inversión resulta extremadamente bajo, y puede incrementarse dándole uso al 63% del tiempo sobrante, por ejemplo dedicándolo a tareas de renderización bajo pedido. Con ello se ayuda a recuperar los costos de inversión derivados de la adquisición de los equipos, el mantenimiento, la energía que consumen (tanto directamente como por el aire acondicionado necesario para operarlos), y que con el sólo uso académico no se recupera.

Como se mencionó anteriormente, haremos uso de los laboratorios y talleres de cómputo, en horarios y días en los que no afecten el uso normal:

- De lunes a viernes de 22:00 a 7:00 horas.
- Sábados y domingos completos.
- Días festivos completos.

Se planea que el clúster inicie operaciones a las 22:30 hrs, tomando en cuenta que las actividades académicas concluyen a las 22:00 hrs.

## 2.2. Implementación y configuración del clúster

Laboratorio de día, render farm de noche. Para lograr la convivencia del clúster en los equipos de cómputo académico partimos de la siguiente premisa: “el día no debe saber del uso nocturno; el uso nocturno no debe afectar el uso diario”. Y para lograr esto, optamos por implementar el clúster de forma virtual, lo cual nos aporta las siguientes ventajas:

- Hardware virtualizado homogéneo. Esto quiere decir que independientemente del hardware del que dispongan los equipos, al sistema operativo se le muestra de forma homogénea, lo que nos facilita la configuración del mismo, pues será la misma para todas.
- Posibilidad de arranque y parada automáticos. La máquina virtual nos permite iniciarla de forma automática, sin necesidad de que haya una persona presente para hacerlo, bastará con que los equipos permanezcan encendidos al término de la jornada académica, de modo que inicien a una hora determinada, e igualmente se llevará a cabo la detención del clúster.
- Fácil duplicación de los nodos. Una ventaja importante que proporciona la virtualización del clúster, es que al configurar una sola máquina virtual con las herramientas y características requeridas, se puede duplicar en los demás equipos, evitando con ello la necesidad de hacerlo en cada uno de los equipos que formarán parte del clúster.

Dicho lo anterior, nuestro escenario para probar el funcionamiento del clúster dentro de una institución universitaria es el Laboratorio de Telemática de la Universidad del Caribe, el cual cuenta con 24 equipos con las siguientes características:

Cantidad	Procesador	Velocidad	Memoria RAM	HD
11	Pentium 4	2.8 GHz	512 MB	80 Gb
13	Celeron	2.4 GHz	512 MB	40 Gb

Para el nodo maestro se utilizó una laptop HP DB600, con un procesador AMD Turion X2 con una velocidad de 2 Ghz, memoria RAM de 2 Gb y del almacenamiento de 250 Gb, de los cuales se ocuparon 30 Gb para la instalación del Sistema Operativo Fedora 13. La descarga y manual de instalación de Fedora se obtuvieron de la página oficial [FEDORA].

En los nodos virtuales, se utilizó VirtualBox 3.2.8, el cual se descargó de la página oficial [VirtualBox]. Se instaló en los equipos sobre sistema operativo Windows XP, haciendo uso de 5 Gb para el disco virtual que contendría el sistema operativo, que en este caso fue Debian 5 y que se instaló siguiendo el manual de instalación de la página oficial [Debian]. En la configuración de la máquina virtual se eligió la opción de adaptador puente para poder obtener una dirección IP propia y así recrear la red de máquinas virtuales, sin afectar el sistema operativo anfitrión.

Configuración -¿Red -¿Adaptador1 -¿Conectado a -¿Adaptador Puente

En la configuración del nodo maestro y los nodos esclavos, se le instaló la biblioteca Parallel Virtual Machine (PVM), la cual nos permite realizar el cómputo paralelo y dispone de una interfaz gráfica llamada XPVM.

Instalación Parallel Virtual Machine

- Descomprimir el .tgz en /usr/local o /usr/share [root@crhistianPC local]\$ tar -xvzf pvm3.4.6.tgz
- Definimos la variable de entorno [root@crhistianPC local]\$ export PVM\_ROOT=/usr/share/pvm3
- Hacemos make dentro de la carpeta pvm3 [root@crhistianPC pvm3]\$ make
- Modificamos los permisos del directorio LINUX [root@crhistianPC pvm3]\$ cd /usr/local/pvm3/bin [root@crhistianPC bin]\$ chmod 757 LINUXI386/
- Editamos el archivo .bashrc [pvm@crhistianPC bin]# cd / [pvm@crhistianPC]# vi .bashrc
- Agregamos al archivo .bashrc lo siguiente. PVM\_ROOT=/usr/local/pvm3 PVM\_DPATH=\$PVM\_ROOT/lib/pvmd export PVM\_ROOT PVM\_DPATH
- Y descomentamos las siguientes líneas If [ -n \$PVM\_ROOT ]; then
- export PVM\_ARCH=\$PVM\_ROOT/lib/pvmgetarch

- #
- # uncomment one of the following lines if you want the PVM commands
- # directory to be added to your shell path.
- #
- export PATH=\$PATH:\$PVM\_ROOT/lib # generic
- export PATH=\$PATH:\$PVM\_ROOT/lib/\$PVM\_ARCH # arch-specific
- #
- # uncomment the following line if you want the PVM executable directory
- # to be added to your shell path.
- #
- export PATH=\$PATH:\$PVM\_ROOT/bin/\$PVM\_ARCH
- fi
- Grabamos y salimos del editor
- Ejecutamos el .bashrc [pvm@crhistianPC ]# . .bashrc
- Para ejecutar PVM simplemente ponemos pvm en la linea de comando [pvm@crhistianPC ]# pvm pvm;

Para que pvm pueda agregar nodos necesitamos agregar a /etc/hosts las direcciones IP y los nombres de los nodos esclavos. Ejemplo:

- 10.3.0.1 crhistianpc
- 10.3.0.2 debian2
- 10.3.0.3 debian3

Para la comunicación e interacción entre los nodos, PVM hace uso de RSH (Remote Shell), mecanismo por el cual el nodo maestro podrá iniciar programas en los nodos esclavos, y que en ejecución colaboran entre si para la resolución de un problema [RSH]. La instalación de este mecanismo se realizo con los siguientes comandos:

- - Caso Fedora: yum install rsh-server
- - Caso Debian: apt-get install rsh-server

Así mismo se modificaron los archivos `/etc/xinetd.d/rsh` y `/etc/xinetd.d/rlogin` y se habilitaron los demonios `rshd` y `rlogind`.

Archivo `/etc/xinetd.d/rsh`

service shell

socket\_type = stream

wait = no

user = root

log\_on\_success += USERID

log\_on\_failure += USERID

server = /usr/sbin/in.rshd

disable = no

Archivo `/etc/xinetd.d/rlogin`

service login

socket\_type = stream

wait = no

user = root

log\_on\_success += USERID

log\_on\_failure += USERID

server = /usr/sbin/in.rlogind

disable = no

Se reinicia el demonio `xinetd`

- [root@crhistianPC ~]\$ /etc/init.d/xinetd restart

En cada nodo, tanto el maestro como los esclavos, se creo un archivo en `/etc/hosts.equiv` donde se colocaron las direcciones IP de todos los nodos para que se puedan conectar entre si. Hay que mencionar que `rsh` no contempla ningún esquema de protección ni autenticación fuera del uso del archivo `/etc/hosts.equiv`. En redes publicas, el uso de `rsh` no es muy recomendado por su baja seguridad, sin embargo, en un clúster tipo Beowulf se cuenta con una red aislada de la sección publica, por lo que el uso de `rsh` no conlleva a un riesgo de seguridad.

Para las tareas de renderizado, se instaló PVMPOV (PVM+PovRay), el cual es un parche para POV-Ray, y que proporciona a POV-Ray la habilidad de distribuir un renderizado a través de múltiples sistemas heterogéneos, usando la ideología de PVM. El nodo maestro es el responsable de dividir la imagen en bloques pequeños, el cual se los



asigna a los nodos esclavos. Cuando los nodos esclavos terminan de renderizar los bloques pequeños son enviados de vuelta al nodo maestro el cual une estos pequeños bloques para formar la imagen final [PVMPOV].

La aplicación se descargo de la pagina oficial, en total fueron tres archivos:

- pvmpov-3.1g2.tgz: Contiene el parche PVMPOV
- povuni.s.tgz: Contiene el código fuente UNIX del POV-Ray
- povuni.d.tgz: Contiene archivo de datos, documentación y ejemplos que forman parte de POV-Ray

Primero descomprimos el parche pvmpov-3.1g2.tgz en /home/pvm, el cual nos crea una carpeta pvmpov-3.1g2 donde se descomprimen los archivos povuni.s.tgz y povuni.d.tgz

```
pvm@crhistianPC $ tar xzf pvmpov3_1g_2.tgz
```

```
pvm@crhistianPC $ cd pvmpov3_1g_2
```

```
pvm@crhistianPC pvmpov3_1g_2 $ tar xzf povuni.s.tgz
```

```
pvm@crhistianPC pvmpov3_1g_2 $ tar xzf povuni.d.tgz
```

una vez extraídos los .tgz, dentro de la carpeta pvmpov3\_1g\_2, le damos ”./inst-pvm” para aplicar el parche PVMPOV y nos aparecerá algo similar a lo siguiente:

```
item [pvm@crhistianPC pvmpov3_1g_2]$ ./inst-pvm
```

```
Trying to apply the patch.
```

```
Searching for rejected files
```

```
pvm@crhistianPC pvmpov3_1g_2 $
```

Ya aplicado el parche PVMPOV con éxito, se pueden construir los binarios PVMPOV, para esto nos situamos en la ruta, /home/pvm/pvmpov3\_1g\_2/povray31/source/ y aplicamos en la línea de comando .aimk newunix”

```
- [pvm@crhistianPC pvm]$ aimk newunix
```

Cabe señalar, que en esta parte se tuvo que hacer un make en la libreria png, ligar la libreria zlib y se redireccionó el makefile hacia las dos variables estáticas \*tids contenidas en los archivos pvm.h para que se pueda instalar correctamente PVMPOV. Todo esto se le aplico tanto el nodo maestro como los nodos esclavos.

### 3. Resultados

En las siguientes pruebas realizadas, el tamaño de las imagenes es de 9600x7200:

Prueba1

# nodos	Tiempo renderizado
1	37.35 min
4	13.35 min
8	9.42 min
1	4.59 min

Si tomamos en cuenta se hara uso del clúster por lo menos siete otras diarias, tenemos:

$$(420 \text{ min/noche}) / (4.46 \text{ min/frame}) = 91 \text{ frames/noche}$$

$$(94 \text{ frames/noche}) / (24 \text{ frames/seg}) = 4 \text{ seg.pelicula/noche}$$

En una noche se renderizaría 3 segundos de pelicula!! Sin embargo, esta cantidad podria aumentar si consideramos que utilizaremos por lo menos 2 laboratorios de cómputo.

Prueba2

# nodos	Tiempo renderizado
1	44.42 min
4	22.29 min
8	11.52 min
1	4.46 min

En ambas pruebas podemos ver que el tiempo de renderizado que tomo en una sola maquina con relación a 16 máquinas es considerable, por lo que el tiempo de procesamiento se optimiza al tener mas nodos trabajando en el clúster.

Lo siguiente es el retorno de inversión, para lo cual hemos establecido como precio base de arrendamiento del clúster \$100 la hora o fraccion de hora. Este precio lo sacamos en relación al mercado actual, en el cual la empresas cobran de \$100 pesos, \$250 pesos o \$25 dólares. Hay que aclarar que esta no es la única forma de cobrar ya que se puede ahcer por el tamaño de la imagen, por el número de procesadores utilizados en el renderizado o se puede negociar con el cliente.

En la la siguiente tabla, nuevamente tenemos el desgloce de las características de los equipos de Telemática, solo que ahora le agregamos una columna mas de costo aproximado.

Cantidad	Procesador	Velocidad	Memoria RAM	HD	Costo aproximado
11	Pentium 4	2.8 GHz	512 MB	80 Gb	\$4000
13	Celeron	2.4 GHz	512 MB	40 Gb	\$4000

Si se hiciera uso del clúster en los días ya estipulados, entonces:  
 $(372 \text{ horas/mes})(\$1000 \text{ MN/hora}) = \$37,000 \text{ MN}$

De acuerdo a la tabla, el precio aproximado de los equipos es de \$4000 por lo que el costo de un equipo de cómputo se recupera en una semana.

## 4. Conclusión

De acuerdo a las pruebas realizadas podemos concluir que se tuvo un buen procesamiento al renderizar y que este nivel de procesamiento puede aumentar si se aumenta el número de nodos al clúster. En lo que a la inversión respecta, el retorno de inversión es bueno, por lo que este proyecto resulta viable para aplicar en las instituciones.

### 4.1. Trabajos futuros

Ya que tenemos el clúster implementado lo siguiente será establecer las formas de contacto con el cliente, por lo que será necesario un servidor ftp para el intercambio de archivos, al igual que una página web.

Para la operación y mantenimiento del clúster será necesaria la contratación de personal, sin embargo este costo se puede ahorrar si se emplean a Becarios o Practicantes.

## Referencias

- [1] Universidad de Las Palmas de Gran Canaria, Graficos 3D por computadora, 2010, <http://www2.dis.ulpgc.es/>
- [2] Tecnoarquitectura, Pelicula: Avatar de James Cameron, 2010, <http://tecnoarquitectura.com/>
- [3] Clúster de Alto Retorno de Inversión para Aplicaciones Geofísicas”, José Enrique Álvarez Estrada, 2008
- [4] Proyecto Fedora, 2010, <http://fedoraproject.org/>

- [5] Oracle VirtualBox, 2010, <http://www.virtualbox.org/>
- [6] The universal operating system, Debian, 2010,  
<http://www.debian.org/>
- [7] PVMPOV HOWTO, 2010, <http://pvmpov.sourceforge.net/PVMPOV-HOWTO.html>