Proyecto II
Cluster Beowulf para procesamiento de imágenes con filtro Rayleigh

L. Barboza, M. Davis, S. Víquez

***Resumen*—Actualmente, el uso de clusters como una alternativa para la computación paralela y el procesamiento distribuido tiene una relevancia en la computación de alto desempeño para aplicaciones que requieran el uso de dichos recursos. La presente investigación consiste en la configuración de un cluster Beowulf compuesto por un nodo maestro y 3 esclavos que se comunican por medio de una interfaz de paso de mensajes en la que se distribuye la carga de trabajo de la aplicación de un algoritmo de procesamiento de imágenes, de manera que las 4 computadoras independientes trabajan conjuntamente como si fuera una computadora de alto desempeño. Esto permite que la imagen sea procesada de una manera más rápida en comparación a una única computadora, siendo muy útil en este caso en el que se trabajan con gran cantidad de pixeles, realizando cálculos paralelos y de manera distribuida y aprovechando los recursos independientes de los 4 equipos.**

Palabras clave: algoritmos de procesamiento de imágenes, cluster Beowulf, computación de alto desempeño, interfaz de paso de mensajes, procesamiento distribuido.

I. Introducción

Hoy en día, el procesamiento de elementos como imágenes, audios, videos y demás gastan muchos recursos por la gran cantidad de procedimientos que se realizan sobre estos elementos. Al tener este problema, una de las soluciones que puede existir es aumentar el hardware existente, como la memoria. Al realizar esto los costos para los encargados se puede elevar mucho, por lo que se buscaron alternativas para lograr el objetivo. Un ejemplo para reducir tiempos y trabajar de manera más eficiente es el cluster. La idea principal es unir varios equipos de cómputo para que trabajen como uno solo, de manera que se pueda dividir la tarea y que cada una de las computadoras conectadas pueda realizar una pequeña parte del trabajo.[1]

Una forma de aplicar un cluster puede ser Beowulf. Este método se basa en computadoras personales que están conectados mediante una red.[2] El objetivo es lograr realizar una tarea compleja sobre un archivo, como por ejemplo un filtro sobre una imagen. Una imagen contiene miles de pixeles que deben ser modificados para lograr el objetivo final, que es un filtro completo sobre toda imagen. En cada uno de los pixeles se puede realizar una suma, multiplicación en conjunto con otra información de los demás elementos en la imagen, por lo que resulta muy costoso en términos de recursos computacionales.

Cuadro I
Especificaciones de los nodos del cluster

Nodo	Núcleos	Frecuencia(GHz)	RAM(Gb)
Maestro	4	1.50	8
Nodo1	4	2.20	4
Nodo2	6	2.50	4

Gracias a estos métodos no se necesitan supercomputadoras para lograr grandes cálculos, sino que un conjunto de computadoras con recursos no tan altos pueden realizar muchos cálculos de manera paralela.[3]

II. Sistema desarrollado

Se desarrolló un cluster para aplicar el algoritmo de procesamiento de imágenes Rayleigh, el cual modifica los pixeles de una imagen de entrada mediante procesamiento paralelo que se distribuye por medio de una interfaz de paso de mensajes (MPI) entre las computadoras que componen el cluster.

En este sistema se utilizó MPICH como MPI para la comunicación y distribución de trabajo entre los nodos, que se comunican mediante el protocolo SSH. Además, los datos entre los nodos se comparten mediante el protocolo NFS (Network File System), el cual es un sistema de archivos de red que permite a los nodos esclavos interactuar con el directorio creado por el nodo maestro.

El cluster se encuentra compuesto por 3 computadoras con especificaciones de hardware que se resumen en el cuadro I.

Tanto el nodo maestro como el Nodo1 poseen procesadores de la familia Intel Core i3, mientras que el Nodo2 tiene un procesador AMD Phenom II X6. Todos poseen tecnologías con múltiples núcleos (multicore) en el caso de las de Intel HyperThreading con 2 procesadores físicos y 4 procesadores lógicos.

En cuanto al software del sistema desarrollado, se implementó una aplicación que utilizando MPICH y OpenCV distribuye el procesamiento de un filtro que agrega ruido Rayleigh a una imagen. Para esto, se utilizan las colectivas de comunicación scatter y de gather sobre un arreglo de caracteres que representan pixeles con valores de 0 a 255(imagen en escala de grises).

Cuadro II
TIEMPOS PARA NODO MAESTRO

Procesos	Imagen 1 (s)	Imagen 2 (s)	Imagen 3 (s)
1	0.096762	0.809378	4.423562
2	0.127688	-	2.250418
4	0.044363	-	1.516407
5	0.055641	0.478498	-
6	0.104748	-	1.991682
9	-	-	2.193187
13	-	0.584146	2.081753
15	0.232541	-	-
20	0.448050	-	-

III. RESULTADOS

Para analizar los resultados se realizaron varias pruebas variando diferentes parámetros como lo son la cantidad de nodos y el tamaño de la imagen. Para todas las pruebas se hizo uso de 3 imágenes. La primera con un tamaño de 640x480 pixeles (**Imagen 1**), la segunda presenta un tamaño de 2119x1415 pixeles (**Imagen 2**), mientras que la última, con una dimensión más grande, de 4836x3372 pixeles (**Imagen 3**). Las tres se mantienen constantes durante las diversas ejecuciones para tener parámetros similares.

En el cuadro II se tiene la comprobación para el nodo maestro. El objetivo de esta prueba es establecer los valores de comparación para medir las diferencias cuando se le añadan los otros nodos.

Para este caso se pueden observar datos importantes. En el caso de la imagen más pequeña se nota que en varias pruebas es mejor mantener sólo un proceso para llevar a cabo el filtro, ya que la interacción entre procesos va a reducir tiempos de ejecución. Para las dos imágenes siguientes, que poseen un tamaño más grande, se nota que los otros procesos ayudan a disminuir el período de la aplicación del filtro.

En el cuadro III se presentan los tiempos con el nodo maestro y otro nodo esclavo. Para este caso, se hizo la combinación Maestro-Nodo1 y Maestro-Nodo2.

Para este caso, se nota una diferencia entre nodo 1 y nodo 2 por la razón de que tiene diferente hardware, específicamente en el procesador. Para el nodo 2 se tenían trabajando 6 procesadores, mientras que en el 1, 4 procesadores. Realizando la comparación entre esta tabla y la del maestro, se nota que la comunicación sigue siendo un factor determinante para este tipo de operación, ya que se nota que es el causante de aumentar el tiempo, lo que ocasiona una mayor latencia en el procesamiento de la imagen.

Por último, en el cuadro IV se muestran los tiempos de procesamiento con el nodo maestro y los nodos 1 y 2 trabajando de forma paralela en el procesamiento de la imagen con el filtro Rayleigh.

Cuadro III
TIEMPOS PARA MAESTRO-NODO1 Y MAESTRO-NODO2

Nodo	Procesos	Imagen 1 (s)	Imagen 2 (s)	Imagen 3 (s)
1	2	0.896153	-	42.817850
2	2	0.227942	-	14.347155
1	4	0.044363	-	28.036828
2	4	0.171569	-	10.952720
1	5	1.470113	5.580889	-
2	5	0.309158	1.250985	-
1	6	0.569278	-	21.445084
2	6	0.204036	-	7.813255
1	9	-	-	17.440918
2	9	-	-	5.717429
1	13	-	4.975904	17.536929
2	13	-	1.007248	5.378473
1	15	0.366976	-	-
2	15	0.310568	-	-
1	20	0.432598	-	-
2	20	0.295147	-	-

Cuadro IV
TIEMPOS PARA MAESTRO-NODO1-NODO2

Procesos	Imagen 1 (s)	Imagen 2 (s)	Imagen 3 (s)
2	1.135766	-	48.457385
4	1.227752	-	57.810728
5	1.054499	8.731997	-
6	1.444421	-	57.469437
9	-	-	68.529231
13	-	11.711533	63.007725
15	2.055664	-	
20	2.082865	-	

Al igual que en los casos anteriores, la comunicación sigue siendo un factor importante para el cluster. En algunos casos sirvió dividir más la tarea de procesar la imagen en otros nodos, pero en otros más bien afectó el rendimiento.

En las Fig.1 2 y 3 se muestran los gráficos con los resultados por imagen para cada uno de los análisis. Y en las Fig. 4 y 5 se puede observar tanto la **Imagen 1** original como el resultado del filtrado.

IV. CONCLUSIONES

La computación paralela y distribuida es actualmente una solución ante las necesidades de procesamiento de gran cantidad de datos y sistemas de cómputo de alto rendimiento.

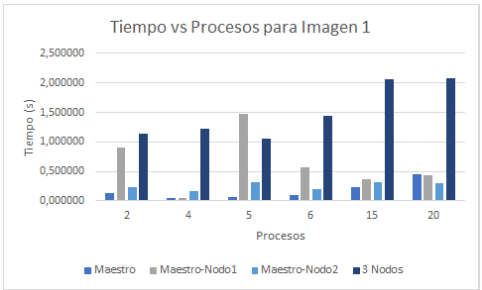

Figura 1. Gráfico con los tiempos de ejecución para la imagen 1.

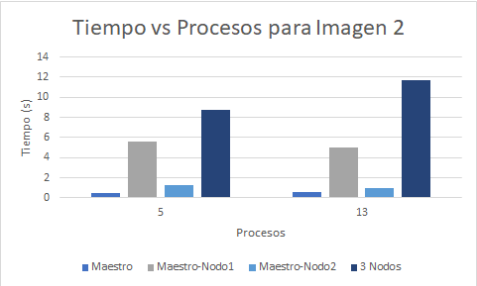

Figura 2. Gráfico con los tiempos de ejecución para la imagen 2.

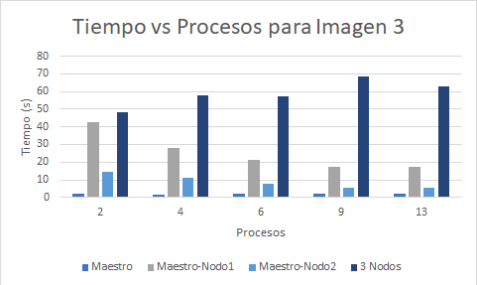

Figura 3. Gráfico con los tiempos de ejecución para la imagen 3.

Figura 4. Imagen Original.

Figura 5. Imagen Filtrada.

La implementación de un cluster Beowulf es fácil de implementar en casos en los que no se tenga grandes recursos, ya que tiene un bajo costo porque se basa en el uso de computadoras de uso diario que trabajan en conjunto para el procesamiento de datos en paralelo.

De acuerdo a los resultados obtenidos, es posible concluir que el paralelismo es muy bien aprovechado en términos de rendimiento para imágenes grandes donde se realicen muchos cálculos, mientras que para imágenes muy pequeñas donde no sean gran cantidad de cálculos se sufre de una penalidad en el tiempo de procesamiento ocasionada por la comunicación entre nodos.

Así, un cluster Beowulf es útil cuando se requiere computar muchos datos de forma paralela, no tanto así cuando se requiere una comunicación constante entre nodos.

Referencias

1 "Creación de un Beowulf", *Universitat Politécnica de Catalunya*, 2006. [Online]. Available: http://personals.ac.upc.edu/enric/PFC/Beowulf/beowulf.html. [Accessed: 06-May-2018].
2 "Clusters Beowulf", *Revista Universidad Autónoma de México*, 2009. [Online]. Available: http://www.revista.unam.mx/vol.4/num2/art3/clustb.htm [Accessed: 06-May-2018].
3 "Beowulf Cluster Computing", *Michigan Tech*, 2010. [Online]. Available: http://www.cs.mtu.edu/beowulf/ [Accessed: 07-May-2018].