

Clúster de Procesamiento Distribuido

Alexis Gavriel Gómez

2016085662

Área Académica de Ingeniería en Computadores
Instituto Tecnológico de Costa Rica

Resumen—En el presente documento se detalla la implementación de un clúster de procesamiento distribuido utilizado para aplicar el filtro digital de media a una imagen utilizando OpenMPI + NFS en el lenguaje de programación C.

Palabras clave—OpenMPI, C, clúster, procesamiento de imágenes digitales.

I. INTRODUCCIÓN

Un clúster es un sistema distribuido formado por un conjunto de computadoras trabajando con un mismo objetivo, en donde cada nodo realiza el mismo o parte del mismo cálculo. Generalmente, la comunicación entre las computadoras de un clúster se da por medio de la red, por lo que esta juega un papel muy importante. Dependiendo de las características de la red, se puede ver afectada la velocidad, ancho de banda y retraso en la comunicación entre los dispositivos. La biblioteca OpenMPI es una interfaz para la transmisión y recepción de mensajes entre dispositivos, lo cual permite sincronizar tareas en un sistema distribuido [1]. Un sistema de archivos de red (NFS) permite a diversos dispositivos interconectados, compartir archivos por medio del sistema de archivos del sistema [2][3].

II. AMBIENTE DE DESARROLLO

Para el ambiente de desarrollo se utilizó el lenguaje de programación C y se programó utilizando el editor de texto Visual Studio Code. Para la generación del programa ejecutable se utiliza un Makefile. Así mismo, se utilizó la biblioteca OpenMPI 4.0.3 y se configuró un sistema de archivos de red NFS para compartir una carpeta y sus contenidos. Para la ejecución del programa se utilizó un computador con procesador i5-3470, de 4 núcleos sin Hyper-Threading y otro computador con procesador i3-8130U, de 2 núcleos con Hyper-Threading.

III. DISEÑO DEL SISTEMA

Debido a que en total de núcleos y la fácil implementación de un sistema de archivos de red se plantea el diseño utilizando un nodo maestro y cinco nodos esclavos. Todos los sistemas se entrelazan también por medio de un sistema de archivos de red (NFS), en donde tienen acceso a la imagen original que se debe filtrar. En la figura 1 se ilustra un diagrama de la arquitectura de la solución.

En la figura 2 se muestran los distintos módulos del programa y se identifica cuáles son utilizados por el nodo maestro y cuáles son utilizados por los nodos esclavos.

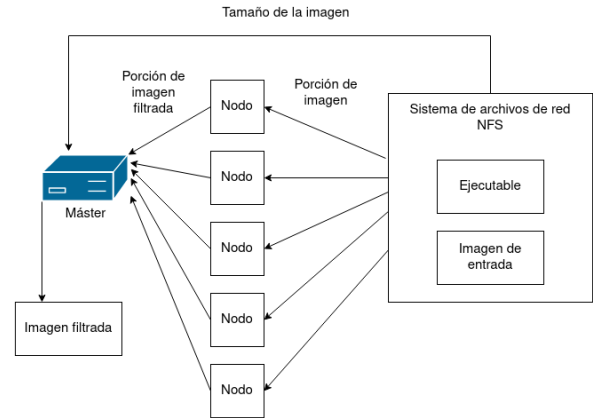


Figura 1. Diseño del sistema distribuido.

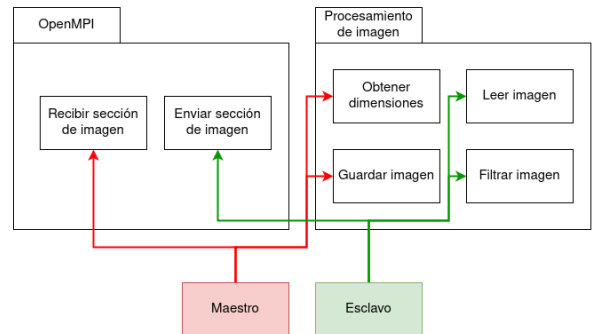


Figura 2. Diagrama de clases.

III-A. Instrucciones del sistema

III-A1. Sistema NFS: Para utilizar el sistema primero se debe configurar un sistema NFS. Para ello se debe:

- Instalar NFS kernel server.
- Crear un directorio y asignarle los permisos para compartir.
- Agregar el directorio a '/etc/exports' y definir la ip de los clientes y sus permisos sobre el directorio.
- Permitir el acceso de NFS a través del Firewall.

III-B. Clúster de OpenMPI

Para utilizar el sistema del clúster se debe:

- Conectarse desde el servidor hacia los clientes por medio de SSH.

- Generar en el maestro una llave de encriptación para la conexión por SSH.
- Agregar a la lista de llaves autorizadas en el cliente la llave generada.
- Agregar al archivo `/etc/hosts` la dirección del cliente.

III-C. Ejecución del programa

Para ejecutar el programa se debe seguir los siguientes pasos:

- Desde la raíz del programa, compilar el programa ejecutando el comando `'make'`.
- Navegar a la carpeta `'bin'` y copiar el archivo `'filter'` al directorio NFS.
- Copiar la imagen deseada a filtrar en la carpeta compartida.
- Ejecutar el comando `'mpirun -hostfile /etc/hosts -np núcleos ./filter imagen resultado ventana'`.

En donde `'núcleos'` se indica la cantidad de núcleo totales a crear, `'imagen'` indica el nombre de la imagen de entrada, `'resultado'` indica el nombre de salida de la nueva imagen. Finalmente, el parámetro `ventana` indica el tamaño del ventana con lo que se usa el filtro.

IV. CONCLUSIONES

Por medio de la biblioteca OpenMPI se puede reducir substancialmente el tiempo de procesamiento de un programa paralelizable sin muchas modificaciones del código serial. Configurar un sistema de archivos NFS permite que todos los nodos puedan acceder a la imagen de entrada, esto permite evitar transferir a cada nodo la sección de imagen que debe procesar, reduciendo así el overhead de comunicación.

V. RECOMENDACIONES

Se puede modificar el nodo maestro para que también se encargue de procesar una sección de la imagen, convirtiendo la configuración en la de un algoritmo distribuido. Para ello cada nodo debe escribir en una sección de memoria compartida el resultado correspondiente. Se pueden utilizar las funciones no bloqueantes para que el nodo maestro reciba las secciones de imagen en el orden que se completen, en lugar de un orden específico. Se puede generar una pila de tareas, en donde cada fila de la imagen es una tarea a procesar, esto permite que se aprovechen mejor los nodos con mejores características, ya que estos no quedan ociosos.

REFERENCIAS

- [1] OpenMPI, "Open mpi," 2020. [Online]. Available: <https://www.open-mpi.org/>
- [2] Linux, "Linux nfs." [Online]. Available: <http://nfs.sourceforge.net/>
- [3] J. Kiarie, "How to install and configure an nfs server," 2020. [Online]. Available: <https://www.tecmint.com/install-nfs-server-on-ubuntu/>