

# Implementación de un Cluster Beowulf y su aplicación al Método de Karatsuba

Desarrollo de Sistemas Distribuidos

Abad Freddy, Arevalo Paul, Reinozo Edison, Vizhnay Esteban

Facultad de Ingenieria, Escuela de Sistemas

Universidad de Cuenca

[freddy.abadl](mailto:freddy.abadl@ucuenca.ec), [edison.reinozo](mailto:edison.reinozo@ucuenca.ec), [paul.arevalo](mailto:paul.arevalo@ucuenca.ec), [esteban.vizhnay](mailto:esteban.vizhnay@ucuenca.ec) @ucuenca.ec

**Abstract:** One of the most complex computational operations is the multiplication of large numbers, which requires a large amount of computational resources and high performance. The Karatsuba algorithm consists of a procedure to multiply large numbers efficiently. An alternative that offers good performance at a low economic cost is the construction of a "Cluster Beowulf" that allows to emulate the behavior of a computer with several processors. This article shows the hardware requirements for the construction of the Beowulf Cluster, the implementation of the Karatsuba method and the performance results for the Beowulf Cluster.

**Keywords:** Cluster, Beowulf, Linux, Distributed, Compute.

**Resumen:** Una de las operaciones computacionales más complejas, es la multiplicación de números grandes, el cual exige gran cantidad de recursos computacionales y alto desempeño. El algoritmo de Karatsuba consiste en un procedimiento para multiplicar números grandes eficientemente. Una alternativa que ofrece buen desempeño a un bajo costo económico es la construcción de un "Cluster Beowulf" que permite emular el comportamiento de un computador con varios procesadores. En el presente artículo se muestran los requerimientos de hardware para la construcción del Cluster Beowulf, la implementación del método de Karatsuba y los resultados de rendimiento para el Cluster Beowulf.

**Palabras Claves:** Cluster, Beowulf, Linux, Distribuido, Computo.

## I. INTRODUCCIÓN.

El Clúster Beowulf brota como una herramienta con gran acogida en el campo de las ciencias de la computación, por ser una opción que busca la mejor relación costo beneficio y que se ajusta a las necesidades de un problema complejo que requiere de un poder de computación alto.

Esta herramienta es una tecnología que agrupa computadores basados en el sistema Linux (aunque existen alternativas en Windows) para recrear una máquina poderosa emulando el funcionamiento de un supercomputador de multiprocesamiento. Su origen data al año de 1994 cuando sus creadores, Thomas Sterling y Don Becker realizan el desarrollo de este como un proyecto de investigación patrocinado por el Centro de la excelencia en ciencias de los datos y de la información del espacio.

### A. Objetivo General

- Construir un Cluster con características Beowulf, implementado el método Karatsuba para cálculo de multiplicaciones matriciales de tamaños extremadamente grandes.

## II. MARCO TEORICO

Un cluster es un conjunto de computadoras construido utilizando de hardwares y softwares comunes, los cuales se comportan como una única computadora. Así un cluster Beowulf es un sistema de cómputo paralelo basado en clusters de ordenadores personales conectados a través de redes informáticas estándar, sin el uso de equipos desarrollados específicamente para la computación paralela. Es popularmente llamado, el "cluster de los pobres", ya que permite reutilizar los equipos caseros y crear sistemas de cómputos en paralelo. La primera versión fue creada por Donald Becker y Thomas Sterling, en 1994, agrupando 16 procesadores Intel DX4 de 100 MHz, interconectados con tecnología Ethernet a 10 Mbps, en equipo viejos con Linux como Sistema Operativo.

Uno de los algoritmos que reducen el número de operaciones de multiplicaciones considerablemente, es el método de Karatsuba. Este es un procedimiento para multiplicar números grandes eficientemente, su desarrollo fue a mano de Anatolii Alexeevitch Karatsuba en 1960. Este método reduce la multiplicación de dos números de  $n$  dígitos a como máximo  $3n^{\log_2 3} \approx 3n^{1.585}$  multiplicaciones de un dígito.

Así el metodo es el método más rápido que el algoritmo clásico, que requiere  $n^2$  productos de un dígito.

Existen alternativas más rápidas en casos donde  $n$  es lo suficientemente grande, como el algoritmo de Toom-Cook o el algoritmo Schönhage-Strassen las cuales son generalizaciones más rápidas del de Karatsuba. Estos fueron los parámetros para la elección de este método para hacerlo metodo de prueba para el cluster el cual se desarrollo, además de que este algoritmo es un ejemplo interesante del paradigma divide y vencerás.

### III. PROCEDIMIENTO

#### A. Características

Iniciamos el proceso de montaje de un cluster beowulf, empleando 4 computadores de características semejantes, de las cuales 1 máquina fue nodo master y 3 nodos slaves. Las características de estos equipos son :

<p style="text-align: center;"><i>MASTER</i></p> <p>Nombre de Usuario: rt  Nombre de Equipo: estbm  Intel Core I5 6200U 4 nucleos  RAM 8 GB  Disco Duro 256 SSD  Sistema Operativo: Linux Mint 18.3  Cinnamon 64-bits</p> <p style="text-align: center;">-----*</p> <p style="text-align: center;"><i>Nodo 1 Freddy</i></p> <p>Nombre de Usuario : rt  Nombre de Equipo : rt  Intel Core I5  RAM 4 GB  Disco Duro 600 GB  Sistema Operativo: Linux Mint 18.3  Cinnamon 64-bits</p> <p style="text-align: center;">-----*</p> <p style="text-align: center;"><i>Nodo 2 Edison</i></p> <p>Nombre de Usuario : rt  Nombre de Equipo : edzzn  Intel Core I7 2630  RAM 6 GB  Disco Duro 609 HDD  Sistema Operativo: Linux Mint 18.3  Cinnamon 64-bits</p>
--

<p style="text-align: center;">-----*</p> <p style="text-align: center;"><i>Nodo 3 Paul</i></p> <p>Nombre de Usuario : rt  Nombre de Equipo : paul  Intel Core I7 4720MQ 2.20GHz 4 Cores  RAM 12 GB  Disco Duro 1 TB  Sistema Operativo: Linux Mint 18.3  Cinnamon 64-bits</p>
--

*Figura : Características de los equipos empleados en Cluster Beowulf*

El Clúster se comporta como una sola máquina, es decir el nodo master puede acceder y controlar vía remota los nodos slave. El nodo maestro controla el Clúster prestando el sistema de archivos a los nodos esclavos. Para la comunicación se debe tener componentes con una velocidad suficiente, por ejemplo una tarjeta de red mayor o igual a 100 Mbps disminuye la colisión en la comunicación, en el caso del cluster montado, todas las maquinas contenian una tarjeta de red de 100Mbps de velocidad.

El software utilizado para distribuir los procesos por medio de la red a los demás computadores, fueron las distintas librerías MPI (Message Passing Interface), OpenMPI. Esta utiliza compiladores C para ejecutar programas paralelas.

Como se detallo en la sección I, el cluster fue idealizado para realizar multiplicación de matrices extremadamente grandes, es por eso que se debe especificar la cantidad de procesamiento en Flops (operaciones de punto flotante por segundo).

#### B. Proceso

Para realizar la conexión correctamente, sin mayores problemas, se requiere un router, que permita crear una red LAN (cableada, existen metodologías para utilizar WLAN, en este caso no aplica por requerimientos de velocidad sin interrupciones), mediante cable de red categoria 5e y 6a, así permite una comunicación a la velocidad previamente mencionada (100 Mbps sin problema alguno). Además de crear la red LAN, se debe configurar a los nodos slave y master, para tener una IP Fija, asi se evitan

problemas en los archivos etc/hosts, el cual contiene las IP's de estos nodos.

Después de implementar una red cerrada entre los nodos slave y master, se procede a la instalación de las dependencias OpenMPI en todos los nodos. Posterior a esto, se instala Openssh-server para los nodos Slave y Openssh Client, estas dependencias habilitan el control de nodo de Master a Slave.

Para la configuración de los nodos, se necesita la generación de claves vacías en todos los nodos y la configuración de las IP's en los archivos etc/hosts en todos los nodos, así todos los nodos conocen a todos los nodos, sin importar su "jerarquía". Ya que un Cluster se comporta como una Supercomputadora, siendo la unión de varios componentes pero procesando como una sola máquina, debe realizarse todas las tareas sin el ingreso de contraseñas de los nodos hijos. Es por esto que se debe modificar autorizaciones para que el nodo Master pueda loguearse con los Slave, así pueden estas loguearse.

```
Master 192.168.100.**
Nodo1 192.168.100.**
Nodo2 192.168.100.**
Nodo3 192.168.100.**
```

*Figura 2: Configuración de los archivos etc/hosts (Se encuentran en una red LAN cerrada)*

Posteriormente se crea un archivo compartido para que OpenMPI pueda correr los programas, así este archivo almacena esta configuración. Además de que este archivo contendrá los slots asignados a cada máquina nodo.

Finalmente para tener un directorio compartido entre todos los nodos, se procede a montar este directorio, así todos los cambios que se ejecute en este directorio, se visualizará en todas las máquinas participantes.

Para la ejecución de los archivos de prueba (método Karatsuba ejecutada de forma paralela), se necesitan librerías basadas en C, estas librerías se importan previamente con la instalación de todas las dependencias OpenMPI.

#### *C. Monitoreo de tareas*

Para el monitoreo de tareas se utiliza Ganglia, un software flexible, que permite mediante un servidor Apache, monitorear las acciones que ejecuta todos los procesadores que actúan en la ejecución de tareas paralelas.

Esta herramienta es de fácil instalación, configuración y empleo, para esto es necesario instalar la interfaz de monitoreo con los servidores Apache específicos para todas las tareas.

En la parte de configuración, se deben acceder al archivo gmetad.conf para la alteración de los hosts, puertos a utilizar, el nombre asignado al Cluster a utilizar.

El correcto funcionamiento del software de monitoreo depende de que los firewalls de los nodos estén desactivados, para así no tener problemas al ejecutar las tareas en paralelo. Además de agregar los puertos a utilizar Ganglia.

Así se tiene configurado correctamente las herramientas de monitoreo.

#### IV. CONCLUSIONES

El uso de un Clúster, en este caso un cluster Beowulf conlleva distintas ventajas entre las cuales están las siguientes:

- Componentes de hardware de fácil accesibilidad. (En la práctica se realizó con 4 Computadoras Laptops). Sin embargo, su montaje puede ser con equipos económico y de fácil reemplazo en caso de fallos.
- Los clusters son populares y de fácil acceso debido a que gran parte del software utilizado es de dominio público y el código fuente está disponible.
- Estas características permiten ajustar fácilmente las necesidades del usuario o los requerimientos de la aplicación en cuanto a procesamiento, memoria, almacenamiento, comunicación.
- Es una tecnología flexible ya que agregar o quitar algún componente para mejorar el desempeño se hace de manera inmediata

#### V. BIBLIOGRAFÍA

1. Colaboradores Wikipedia, “Cluster Beowulf”. 2018. Available online: [https://es.wikipedia.org/wiki/Cluster\\_Beowulf](https://es.wikipedia.org/wiki/Cluster_Beowulf)
2. Bernal C. Iván, Mejía N. David y Fernández A. Diego, “Desarrollo De Aplicaciones Paralelas Para Clusters Utilizando MPI (Message Passing Interface)”, 2006. Available online: <http://clusterfie.epn.edu.ec/ibernal/DesarrolloAplicacionesParalelas.pdf>
3. Colaboradores Wikipedia, “Algoritmo de Karatsuba”. 2017. Available online: [https://es.wikipedia.org/wiki/Algoritmo\\_de\\_Karatsuba](https://es.wikipedia.org/wiki/Algoritmo_de_Karatsuba)
4. Victorino J, Hernandez P., “Implementación de un Cluster Beowulf” (2001). Available online: [http://ciencias.bogota.unal.edu.co/fileadmin/content/geociencias/revista\\_meteorologia\\_colombiana/numero03/03\\_05.pdf](http://ciencias.bogota.unal.edu.co/fileadmin/content/geociencias/revista_meteorologia_colombiana/numero03/03_05.pdf)