

## • **Introducción**

El campo de la supercomputación es, a mi parecer, uno de los más interesantes dentro de la informática. De hecho gracias a él, hoy en día, podemos resolver complejos problemas como por ejemplo el genoma humano, algo impensable hace unas décadas.

Aunque podemos encontrar muchas maneras de realizar supercomputación, voy a centrarme en lo que parece ser las nuevas tendencias: supercomputación a bajo coste empleando el robusto sistema operativo Linux.

## **2. ¿Qué es el procesamiento en paralelo?**

Para poder comprender la totalidad del texto, es necesario antes de nada aclarar este concepto, ya que lo emplearé en numerosas ocasiones:

Se trata de un proceso empleado para acelerar el tiempo de ejecución de un programa dividiéndolo en múltiples trozos que se ejecutarán al mismo tiempo, cada uno en su propio procesador.

En principio, un programa dividido en  $n$  trozos de esta forma, podría ejecutarse  $n$  veces más rápido, que su equivalente en un solo procesador, pero aún así queda claro que éste es su límite teórico (es decir el máximo que podremos alcanzar) pudiendo conseguir en el mejor de los casos un valor aproximado con un buen paralelismo.

### **2.1 ¿Cuándo es conveniente el paralelismo?**

Aunque, en principio, paralelizar un programa supone un incremento de su velocidad de ejecución, esto no tiene por qué ser siempre así, ya que hay muchos casos en los que o bien es imposible llevar a cabo una paralelización del mismo, o una vez llevado a cabo ésta, no se aprecia mejora alguna, o en el peor de los casos, se produce una pérdida de rendimiento.

Hay que tener claro que para realizar un programa paralelo debemos, para empezar, identificar dentro del mismo partes que puedan ser ejecutadas por separado en distintos procesadores.

Además, es importante señalar que un programa que se ejecuta de manera secuencial, debe recibir numerosas modificaciones para que pueda ser ejecutado de manera paralela, es decir, primero sería interesante estudiar si realmente el trabajo (y coste) que esto nos llevará se ve compensado por la mejora del rendimiento de la tarea después de paralelizarla.

## • **Tipos de supercomputadores de bajo costo:**

### **3.1 SMP**

Estas siglas responden a *Symmetric Multi-Processor*, es decir, multiprocesador simétrico (varios procesadores idénticos).

Este es el método más caro de los que citaré, pero, aún así, sigue siendo una opción mucho más barata que una máquina mono-procesador de potencia equivalente (si es que existe), para simplificar: siempre será más barato construir  $n$  procesadores de  $x$  bits que un procesador de  $n*x$  bits.

Este tipo de máquina se puede programar con Linux empleando los conocidos métodos de memoria compartida, debido a que hay una sola memoria principal y varios procesadores.

### 3.2 Clusters en Linux

El método de los clusters es sin duda el más extendido para computación en paralelo, debido a que unos clusters con Linux no son más que una red de ordenadores convencional, en la cual se emplea el sistema operativo Linux en cada uno de los nodos.

El procesamiento en paralelo con clusters supone muchas ventajas con respecto a otros métodos, cabe citar:

- Cada uno de los nodos es una máquina completa que podremos utilizar como una estación de trabajo normal, al mismo tiempo que lo empleamos para nuestra máquina paralela.
- Debido a que puede estar formado por PC's normales de gama media o incluso baja (Pentiums o 486), su precio es relativamente bajo (causado por el gran mercado que hay actualmente).
- Si no vamos a emplear cada una de los nodos como estación de trabajo, nos bastará con un teclado y una consola para todo el sistema, pudiéndonos evitar la compra incluso de la tarjeta de vídeo ya que el sistema operativo Linux puede ejecutarse sin todos estos elementos.
- Como bien sabemos la construcción de un supercomputador de  $n$  procesadores (tomando  $n$  como un valor superior a 4) es complejo, caro y en muchos casos irrealizable. Sin embargo, crear una red con 16, 100 o incluso más nodos es, en comparación, algo trivial.
- Cualquier problema en una máquina de este tipo es relativamente sencillo de resolver, ya que si por ejemplo se estropea un procesador en un nodo, nos bastará con sustituirlo por otro para resolverlo, esto es importante tenerlo en cuenta, ya que es infinitamente más sencillo encontrar un nuevo procesador Pentium que cualquiera de los componentes de un complejo supercomputador Fujitsu por ejemplo.

Entonces se puede concluir que un Cluster es potente y barato (muchas veces casi gratis, ya que se puede encontrar 486 antiguos que nadie quiere).

Aún así, los clusters no son la mejor solución en todos los casos ya que, entre otras, encontramos las siguientes pegas:

- Las redes de ordenadores, no están diseñadas para el procesamiento en paralelo (al menos no en principio), siendo su ancho de banda (cantidad de tráfico simultáneo) muchas veces demasiado bajo y su latencia (tiempo mínimo para transmitir un objeto) demasiado alta al menos en comparación con máquinas SMP.
- No hay mucho software que sea capaz de tratar un cluster como si fuese un único sistema.

Aún así, y para finalizar, cabe decir a su favor que cada vez encontramos más software diseñado para aumentar el rendimiento de nuestro cluster con Linux, así como su facilidad de programación. Además, día a día, aparecen nuevas redes diseñadas específicamente para este tipo de arquitectura.

### 3.3 La clase Beowulf

Aquí tenemos un tipo especial de cluster, con un costo inferior a los clusters convencionales y capaz de incrementar su rendimiento.

La clase Beowulf comenzó con un cluster en Linux diseñado en la NASA por Thomas Sterling y Don Becker en 1994. Poco a poco este proyecto creció dando lugar a un nuevo tipo de máquina.

Habiendo visto una descripción general de esta máquina además de algo de su historia, pasemos a analizar que es exactamente:

Un Beowulf es un conjunto de nodos minimalistas (con lo mínimo para funcionar), unidos por un medio de comunicaciones barato, por ejemplo una red Ethernet.

Al decir que son nodos minimalistas, nos referimos a que tienen lo mínimo para ejecutar su función, de hecho los nodos por sí solos son incapaces de ejecutar ni tan siquiera un sistema operativo.

Éstos constarán, en principio, de una placa madre, una CPU, memoria y un dispositivo de comunicaciones, por lo general una tarjeta de red Ethernet o Fast Ethernet, también es posible que dispongan de un pequeño disco duro en el que arrancar el sistema operativo remoto, que será Linux para todos ellos.

Para la programación del Beowulf disponemos de varios métodos, pero primero debemos saber que está máquina no cuenta con mecanismos de memoria compartida, luego las comunicaciones se realizan siempre mediante pasos de mensajes (sockets, PVM (Parallel Virtual Machine)...).

Otra característica importante a tener en cuenta a la hora de la programación de este tipo de máquina es su topología física: red (en anillo, árbol...), así como los tipos de máquinas dispuestas en los nodos, ya que no tienen por que ser de potencia equivalente. Esto nos lleva a la conclusión de que un supercomputador clase Beowulf debería poseer una estructura lógica diseñada específicamente a su estructura física, ésta es quizá la mayor pega, ya que la programación y el diseño del mismo se nos complica con respecto a un Cluster normal, pero, aún así, el rendimiento obtenido empleando los mecanismos correctos, podrá ser superior.

#### **4. Lenguajes de programación/compiladores**

No encontraremos ningún lenguaje de programación para máquinas paralelas tan potente como el versátil C compilado con GCC. Aún así, existen lenguajes de más alto nivel adecuados para estos menesteres, entre ellos, cabe citar:

- **Fortran:**

Fortran, tal y como es hoy es un lenguaje de alto nivel, con numerosas mejoras con respecto al ANSI del 1966, (objetos como los de C++, instrucciones para procesamiento en paralelo,...)

De echo podemos encontrar compiladores capaces de generar código para arquitecturas paralelas tipo SMP y Clusters entre otras. Como ejemplo de éstos cito: HPFC (HPF Compiler) que se encuentra en la siguiente web: <http://www.cri.ensmp.fr/~coelho/hpfc.html>.

- **Mentat:**

Mentat es un lenguaje orientado a objetos para procesamiento en paralelo, funciona en máquinas clusters y se encuentra disponible para Linux. Su sintaxis es similar a la del popular C++.

- **Parallaxis-III:**

Es un lenguaje de programación estructurada basado en Modula-2 pero con extensiones para trabajar con máquinas SIMD (Simple Instruction Multiple Data). La distribución básica de este compilador suele incluir además: un depurador de código y ejemplos de algoritmos.

- **PVM:**

La PVM (Parallel Virtual Machine) no es más que un API GNU de programación C para máquinas paralelas, capaz de crear una máquina virtual paralela, es decir, emplea los recursos libres de cada uno de los nodos sin tener que preocuparnos del cómo, haciéndonos creer que estamos ante una única super-máquina.

El que para mí es el mejor punto de la PVM es que si disponemos de una red UNIX no tenemos más que instalar el software que, además, es libre y ya dispondremos de un supercomputador paralelo con un coste nulo, y podremos ponernos a trabajar con él sin dejar de emplear cada máquina de la red para las mismas funciones que antes.

Además, PVM está portado para otras máquinas además de Linux, entre las que se encuentran distintos tipos de UNIX e incluso existe una versión para NT aunque cabe decir que alcanza unas velocidades bastante inferiores a las que alcanza con Linux.

Como conclusión creo interesante comentar que la PVM ofrece una abstracción inigualable, pero que, aún así, en muchos casos, sobre todo para tareas con alto grado de acoplamiento, la mejor solución al menos en lo que a rendimiento se refiere, es un diseño específico, siendo a veces la única manera posible.

## **5. Ejemplos en los que se emplea supercomputación.**

Cabe indicar alguno de los usos que se la da hoy mismo a la supercomputación de bajo costo.

### **• *The Stone SouperComputer (El Computador de la Sopa de piedras)***

(<http://stonesoup.esd.ornl.gov>):

Esta curiosa máquina ha sido construida con un costo igual a cero, gracias a donaciones individuales (como en el relato de la *Sopa de piedras*, de ahí su nombre). Está compuesta por máquinas que muchos considerarían ridículas: 486 DX/2 a 66Mhz con de 16 a 32Mb RAM de media y discos duros de 400–600Mb, unidos por una red Ethernet (10Mbit/s). Emplean como sistema operativo RedHat Linux, y como lenguajes de programación, desde el GCC (GNU C Compiler) con la extensión PVM hasta Fortran.

Se ha empleado, y se emplea para entre otros trabajos:

- Análisis de los cambios en la distribución de las especies según los cambios del entorno.
- Un mapa nacional a gran resolución.
- La interpolación de 39 mapas que contienen las temperaturas mínimas y máximas (anuales/mensuales) obtenidas por más de 4000 estaciones meteorológicas. Cada uno de estos mapas contiene un total de 7,7 millones de celdas de 1 kilómetro cuadrado.

### **• *Avalon:***

Es el Beowulf más famoso y más potente del mundo (de hecho está entre los 100 computadores más potentes del planeta) es *Avalon* (<http://swift.lanl.gov/avalon/>) compuesto por 140 procesadores Alpha, 36 GB de RAM total, capaz de alcanzar los 47.7 Gflops (los flops son una medida de potencia que indica el número de operaciones de punto flotante que es capaz de ejecutar la máquina en un segundo).

Se encuentra situado en el Laboratorio de cálculo no-lineal de Los Alamos, y se emplea como supercomputador de propósito general abarca un rango de aplicaciones, entre ellas:

- Astrofísica.
- Dinámica molecular.

- Dinámica no-lineal y ecuaciones diferenciales.
- Fases de transición en el universo cercano.
- 



Simulaciones de líquidos y de cristales.

*El supercomputador clase Beowulf Avalon*

## **6. Conclusión y bibliografía**

A lo largo de este trabajo he querido realizar una introducción a la supercomputación de bajo costo, una nueva tendencia dentro de ésta. Hemos tratado como pieza clave para este desarrollo el genial sistema operativo Linux, gracias al cual hoy en día podemos encontrar numerosos supercomputadores en todo el mundo, de hecho creo que todos los centros de cálculo pueden permitirse uno con el que poder experimentar.

También hemos podido analizar el gran abanico de posibilidades que una máquina paralela nos ofrece, ya que puede ser empleada para: geofísica, matemática, astrofísica y meteorología entre otras.

Bibliografía y principales sitios webs consultados para el trabajo:

*Beginning Linux Programming* de Neil Matthew y Richard Stones.

*Linux Actual*: números 6, 7, 8, 9.

<http://www.linux.org> El site principal sobre linux.

<http://www.beowulf.org> Site sobre el desarrollo de Beowulf.

<http://www.extremelinux.org> Grupo de trabajo paralelo a Beowulf.

[http://www.epm.ornl.gov/pvm/pvm\\_home.html](http://www.epm.ornl.gov/pvm/pvm_home.html) La PVM.

<http://swift.lanl.gov/avalon/> La web del Avalon.

*pphowto* Documento sobre como realizar paralelismo con linux.

Apuntes de la asignatura.

Apéndice A. Comparativa de hardware de red.

En las máquinas paralelas un elemento importante a tener en cuenta es el interface que utilizan para las comunicaciones, y antes de adquirir este debemos pararnos a pensar para qué queremos nuestro supercomputador paralelo, y el tráfico de datos que esperamos tener. Una vez tenido en cuenta esto, y atendiendo a nuestro presupuesto, seleccionaremos aquel interface que se ajuste más a lo deseado dentro de los posibles.

Podemos utilizar la siguiente tabla orientativa a la hora de esta elección. Hay que tener en cuenta que los datos pueden variar, especialmente el precio ya que tan sólo está puesto como referencia y para comparativas.

| <b>Nombre</b>    | <b>Ancho de banda</b> | <b>Latencia</b> | <b>Interface/Puerto</b> | <b>Coste/máquina</b> |
|------------------|-----------------------|-----------------|-------------------------|----------------------|
| ARCNET           | 2,5 Mb/s              | 1000 ms         | ISA                     | 200\$                |
| ATM              | 155 Mb/s              | 120 ms          | PCI                     | 3000\$               |
| CAPERS           | 1,2 Mb/s              | 3 ms            | Paralelo                | 2\$                  |
| Ethernet         | 10 Mb/s               | 100 ms          | ISA/PCI                 | 100\$                |
| Fast Ethernet    | 100 Mb/s              | 80 ms           | PCI                     | 400\$                |
| Gigabit Ethernet | 1000 Mb/s             | 300 ms          | PCI                     | 2500\$               |
| Myrinet          | 1280 Mb/s             | 9 ms            | PCI                     | 1800\$               |
| PLIP             | 1,2 Mb/s              | 1000 ms         | Paralelo                | 2\$                  |
| SCI              | 4000 Mb/s             | 2,7 ms          | PCI                     | 1000\$               |

*Algunos dispositivos hardware de red.*

Supercomputación y procesamiento en paralelo

– Pag. 12 –