Sistema de Cómputo Distribuido, aplicado a la Bioinformática

Distributed Computing System, applicated to Bioinformatics

Longendri Aguilera Mendoza, Cesar Raúl García Jacas

Universidad de las Ciencias Informáticas

loge@uci.cu, crjacas@estudiantes.uci.cu

Resumen

La Bioinformática es una ciencia que acude en auxilio de la Biología cuando es necesario aplicar métodos computacionales para analizar y extraer información de un gran volumen de datos de origen biológico. Son varios los centros de investigación y universidades del país que han creado cluster de computadoras dedicadas exclusivamente para los cálculos y que pueden ser utilizados para el procesamiento de datos biológicos. Sin embargo, aun no se aprovecha toda la capacidad de cómputo en la red local de las instituciones ya que no se integran al cluster existente las máquinas que funcionan como simples estaciones de trabajo. En el presente trabajo se ofrece una alternativa de cómputo que aglutina en un solo conglomerado un conjunto de estaciones de trabajo. Para ello, se reutilizó un software existente basado en la plataforma Java al cuál se le realizaron un grupo de mejoras para adaptarlo más a la realidad de un mayor número de instituciones y lograr un mejor desempeño en los tipos de problemas a resolver. El sistema de cómputo desarrollado fue desplegado en la Facultad #6 de la Universidad de las Ciencias Informáticas y ha sido utilizado en la realización de varios cálculos distribuidos por la red.

Palabras Claves: Bioinformática, sistema de cómputo distribuido, supercomputadora

Abstract

The Bioinformatics is a science that comes in aid of the Biology when it is necessary to apply computational methods to analyze and to extract information of a great volume of biological data. Many centers of research and universities have created cluster of computers dedicated exclusively for the calculations and can be used for the processing of biological data. However, institutions do not take advantage of all the computation capacity in the local area network because they do not integrate the existent cluster with the machines that serve as simple workstation. This work offers an alternative of computation that agglutinates in one conglomerate a group of workstations. Software based on java platform was used and a group of improvements were realized to adapt it more to the reality of a major number of institutions and to achieve a better performance in the solutions of problems. The developed computing system was deployed in Faculty 6 of the University of Informatics Sciences and was used in the accomplishment of several calculations distributed by the network.

***Key words**: Bioinformatics, distributed computing system, supercomputer*

Introducción

Uno de los retos más importantes que afronta hoy la Biología es la enorme cantidad de datos disponibles que pueden ser accedidos vía Internet. La presencia de grandes volúmenes de datos biológicos ha dado lugar a una nueva rama de la ciencia: la Bioinformática, que se encarga del estudio, comprensión y análisis de estos datos utilizando herramientas informáticas. El protagonismo de la bioinformática está dado por dos razones principales: la primera es que la enorme cantidad de datos de origen biológico sólo puede ser analizada utilizando computadoras; la segunda es que los datos están tan crudos que las informaciones sólo pueden salir a la luz utilizando sofisticados algoritmos computacionales.

La realidad es que las investigaciones científicas que se realizan en nuestro país, principalmente en la rama de la Biotecnología, necesitan potencia de cómputo para el procesamiento y análisis de tantos datos, que normalmente en computadoras personales demorarían días, semanas o meses para su culminación.

Una variante para dar respuesta a estos problemas de intenso cómputo es mediante el uso de supercomputadoras, es decir, equipos con capacidades de cálculo muy superiores a las de una simple computadora de mesa. Entre las principales desventajas de estas potentes máquinas calculadoras se pueden mencionar su elevado costo. Como ejemplo tenemos la máquina IBM's ASCI White, con una velocidad de 12 TeraFLOPS (1 TeraFLOPS equivale a 10^12 operaciones por segundo) y un costo de $110 millones de dólares. Como podemos apreciar, esta variante no constituye una opción viable y económica para nuestro país, por su millonaria adquisición.

Otra alternativa, basada en redes de computadoras, es comúnmente utilizada para satisfacer la demanda de cómputo que exigen varias aplicaciones informáticas. El desarrollo acelerado de las redes de computadoras en los últimos años, ha hecho reconsiderar la utilización de las supercomputadoras para la ejecución de aplicaciones que demanden de recursos computacionales. Una simple computadora con memoria local y procesador de capacidades moderadas no es de mucha utilidad por sí misma, pero al ser conectada a otras máquinas a través de una red de interconexión suficientemente rápida, se potencia enormemente su utilidad, ya que cientos o miles de máquinas podrían trabajar como un equipo, realizando intensos cálculos para dar solución a un determinado problema.

En la Universidad de las Ciencias Informáticas (UCI), está presente la Bioinformática como una rama de investigación y producción. Varios son los proyectos que se desarrollan con centros del Polo Científico de Cuba, entre los que se destaca, el Centro de Inmunología Molecular (CIM), Centro de Ingeniería Genética y Biotecnología (CIGB), Centro de Química Farmacéutica (CQF), entre otros.

En muchos de estos proyectos se necesita realizar una gran cantidad de cálculos que demoran un tiempo excesivamente largo en una sola computadora. Sin embargo, la UCI presenta una gran potencialidad de cómputo que debe ser explotada ya que tiene el mayor número de computadoras personales (PCs), a nivel nacional, dentro de una institución. En estos momentos cuenta con más de 6 000 PCs distribuidas en toda la red Universitaria. Es válido destacar que la mayoría de estas máquinas son Pentium 4 a 2.4 GHz de velocidad y se encuentran conectadas mediante una red local cuya velocidad de transferencia es de 100 Mbps.

El objetivo principal es desarrollar y desplegar un Sistema de Cómputo Distribuido de Propósito General, que funcione como una "supercomputadora virtual", capaz de aunar y coordinar los esfuerzos entre los recursos disponibles y utilizar de esta forma, gran parte del poder computacional de la UCI, para dar respuesta a los proyectos que requieren de grandes prestaciones de cómputo.

Materiales y Métodos

La elaboración desde cero de un sistema distribuido es muy costosa, por la cual se inició una búsqueda exhaustiva y estudio de las alternativas libres de modelos distribuidos para el cómputo, y se encontró el sistema Java Based Heterogeneous Distributed Computing System [1] que se ajustaba bastante bien a lo que queríamos. Trabajamos sobre el mismo y le hicimos los cambios pertinentes, no sólo para adaptarlo a nuestros objetivos, sino para ampliarle sus funcionalidades.

Entre las deficiencias detectadas a este sistema se encontraban:

- El software no ha tenido mantenimiento ni actualizaciones hechas por sus desarrolladores desde el mes de Noviembre del año 2005.
- El usuario que dona una máquina no puede configurar el módulo cliente para decidir, por ejemplo, a que servidor debe solicitar trabajo en caso de que exista más de uno en la institución.
- Cuando el módulo cliente solicita, mediante Remote Method Invocation (RMI), acceso a los objetos remotos del módulo servidor se generan puertos aleatorios para la comunicación y esto es bloqueado por determinados corta fuegos (firewalls en inglés) que solo dejaban pasar peticiones por puertos bien conocidos.
- Una vez que el módulo cliente finaliza la ejecución de un cómputo, elimina todos los ficheros relacionados con ese procesamiento. Esto no es malo ya que no deja "basura" o ficheros innecesarios en la máquina del usuario que la dona al sistema de cómputo; pero a pesar de esto, si todas las unidades de trabajo generadas para un problema necesitan de un conjunto de ficheros que se mantienen inmutables entonces estos serán eliminados y solicitados nuevamente por el cliente para cada cómputo, lo que puede disminuir el rendimiento del sistema.
- La persistencia de todas las entidades se hace usando los métodos de serialización para objetos que brinda Java y almacenando los datos en ficheros binarios. Esto dificulta las tareas de consulta, seguimiento y generación de reportes.
- El usuario que va a interactuar con el sistema y realizar el cómputo distribuido debe tener los ficheros compilados del código Java y paquetes de clases utilizados debido a que la interfaz gráfica las exige para los cálculos. Esto conlleva a que en una etapa posterior el programador debe hacerles llegar a todos los investigadores las nuevas versiones de las clases que sean modificadas producto del ciclo de vida y mantenimiento del software.

En la elaboración del Sistema de Cómputo Distribuido se utilizó Java como lenguaje de programación, por ser un lenguaje cuya portabilidad está verdaderamente probada [1]. Otras características de dicho lenguaje son:

- Orientado a Objetos: Java trabaja con sus datos como objetos y con interfaces a esos objetos, soporta las características propias del paradigma orientado a objetos: abstracción, encapsulamiento, herencia y polimorfismo.
- Simple: Posee una curva de aprendizaje muy rápida. Ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos.
- Robusto: Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores lo antes posible en el ciclo de desarrollo. Java obliga a la declaración explícita de los tipos de los ítems de información, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de la misma.

Eclipse es un IDE para todo tipo de aplicaciones, inicialmente desarrollado por IBM, y actualmente gestionado por la Fundación Eclipse. Este IDE fue seleccionado para el desarrollo del sistema, principalmente porque es multiplataforma, tiene soporte para distintas arquitecturas, resaltado de sintaxis, auto completado, tabulador de un bloque de código seleccionado, asistentes (wizards): para la creación, exportación e importación de proyectos y para generar plantillas de códigos (templates).

El Sistema de Cómputo Distribuido está basado en el modelo Cliente – Servidor. Está dividido en tres componentes esenciales: servidor, cliente e interfaz de administración. Una vista de como está conformado el sistema, lo muestra la Figura 1.

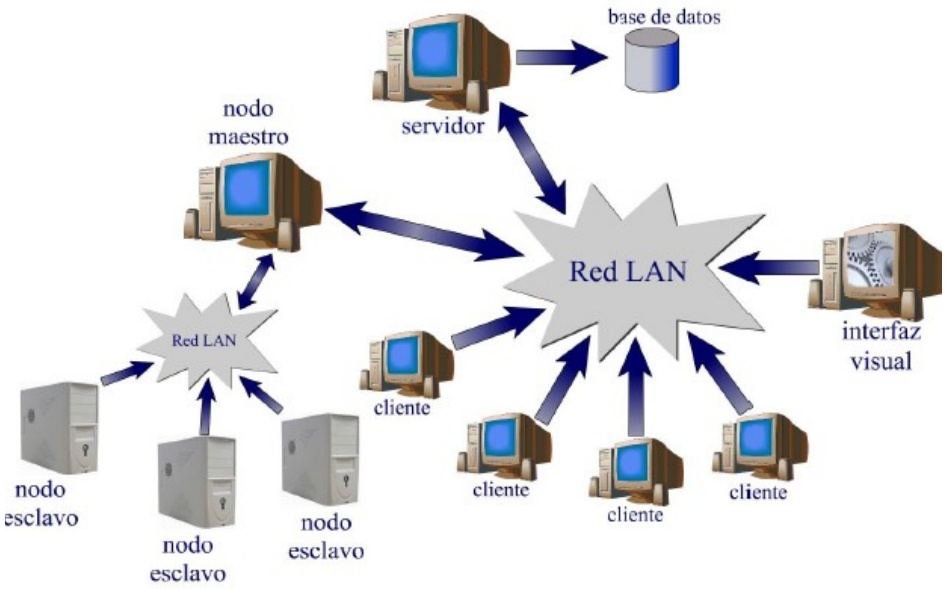

Fig. 1. Vista general del sistema de cómputo

El módulo servidor almacena el problema (datos del problema y el algoritmo que los procesará) y divide estos en pequeños subproblemas, llamados unidades de trabajo. El principal aspecto del servidor es repartir las diferentes unidades de trabajos entre los clientes (computadoras personales, cluster de computadoras, entre otros medios de cómputo), siempre y cuando estos estén autorizados a interactuar con el mismo; además de desarrollar una lógica de control de unidades pendientes de respuestas y otras que han expirado por algún error ocurrido.

El módulo cliente está instalado en cada una de las máquinas y conectadas al servidor mediante una red LAN. El cliente realiza una petición de una unidad de trabajo al servidor, hace el procesamiento de la misma y posteriormente retorna el resultado al servidor, y vuelve a pedir otra unidad de trabajo. Múltiples clientes pueden realizar peticiones de trabajo al servidor. El servidor colecciona el resultado de cada uno de los subproblemas para finalmente construir el resultado del problema original.

El módulo de la interfaz es usado para acceder a todas la funcionalidades en el servidor tales como: administración de cuentas (adicionar y/o eliminar usuarios, grupos, etc.), administración de problemas, así como su ejecución y monitorización; obtener los resultados de las ejecuciones finalizadas, configuración del servidor, entre otras.

El Sistema elaborado cumple con las siguientes características, que son parte esencial de un modelo distribuido.

- Transparencia: El sistema oculta la naturaleza distribuida permitiendo que los usuarios interactúen con una aplicación de Desktop y trabajen como si se tratara de una sola máquina. Los programadores o desarrolladores no tienen que encargarse de repartir el cálculo entre los nodos del sistema, solamente tienen que programar cómo dividir el problema y cómo integrar las soluciones parciales. El trabajo “sucio” de distribuir físicamente las unidades más pequeñas del problema y llevar el control de que estas se realicen queda a cargo del sistema.

- Eficiencia: La solución a los problemas se obtiene más rápidamente haciendo uso del sistema distribuido, que la respuesta que daría una simple computadora.
- Flexibilidad: Un proyecto en desarrollo como la implementación de un sistema distribuido, debe estar abierto a modificaciones que mejoren su funcionamiento. El presente trabajo es lo suficientemente flexible para que cualquier cambio a realizar no requiera la parada de todo el sistema y la recompilación de todo el código. Si se realizan cambios en el módulo del cliente, cambiarlo es tan simple como colocarlo en el servidor, y a medida que cada elemento de cómputo realiza peticiones de trabajo, se le envía la nueva versión para su actualización automática.
- Escalabilidad: El tamaño de una red varía en dependencia de la institución que la mantenga, el sistema debe comportarse estable tanto en redes pequeñas y grandes. También debe garantizar un adecuado mantenimiento y actualización, empleando el mínimo de personal. Ampliar el sistema en cuanto a unidades de cómputos, es tan sencillo como instalar el módulo cliente en una PC y configurarle la dirección física del servidor al cual debe reportarse.
- Fiabilidad: La protección al usuario es extrema, dada la característica que tienen los modelos distribuidos de que su arquitectura está basada en redes. Asegura al 100% que el problema del usuario será resuelto, independientemente de los problemas ajenos que existan, ya sean fallos de red, electricidad, apagado de máquinas, etc. Si un elemento de cómputo se desconecta, la unidad en la que el mismo estaba trabajando no se pierde ya que se almacena una copia de la misma y luego se envía a otro cliente pasado un tiempo.

Entre las tecnologías y herramientas utilizadas resaltan:

- Java – RMI (Remote Method Invocation): permite la comunicación objeto-objeto entre diferentes máquinas virtuales Java, Java Virtual Machines (JVM). Las JVM pueden ser entidades distintas situadas en el mismo o en diferentes ordenadores; incluso una JVM puede invocar métodos pertenecientes a un objeto almacenado en otra JVM. Esto permite a las aplicaciones invocar métodos de objetos localizados remotamente, compartir recursos y procesar cargas en sistemas. Los métodos pueden incluso aprobar objetos que una máquina virtual no ha encontrado nunca, permitiendo la carga dinámica de nuevas clases cuando sea necesario.
- Hibernate: es una herramienta de mapeo objeto/relacional para ambientes de desarrollo en Java. Hibernate no sólo realiza el mapeo de clases Java a tablas de base de datos, y desde tipos de datos Java a tipos de datos SQL, sino que también facilita el proceso de consulta y operaciones sobre base de datos, reduciendo significativamente el tiempo de desarrollo de software.
- Gestor de Base de Datos: el gestor de base de datos utilizado fue MySQL, aunque con la ayuda de Hibernate se podría cambiar de gestor sin necesidad de modificar el código ni tener que recompilar la aplicación. MySQL es un sistema de gestión de bases de datos relacional. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Resultados

El Sistema de Cómputo Distribuido fue desarrollado y desplegado en 10 laboratorios del Docente #4 de la Universidad. Los laboratorios están compuestos por 30 PCs aproximadamente, donde todas las PCs tienen la misma arquitectura y dos sistemas operativos esencialmente: Windows XP y la distribución GNU/Linux Kubuntu.

La Facultad #6 de la UCI, donde la Bioinformática está presente como una rama de investigación y producción, mantiene relaciones de trabajo con el Centro de Inmunología Molecular (CIM), el centro de Química Farmacéutica (CQF), entre otros, en la producción de software y módulos para la Biotecnología. Muchos de estos módulos y programas demandan una gran capacidad de cómputo, debido a la realización de simulaciones biológicas, potentes cálculos sobre moléculas, entre otras funcionalidades. El Sistema de Cómputo Distribuido, es utilizado en varios proyectos para disminuir considerablemente el tiempo que demoran las investigaciones. Como ejemplo de utilización tenemos:

- BioSys: Software para la simulación de Sistemas Biológicos (actualmente en fase de prueba en el CIM).
- GraphTool: Sistema Inteligente para la Predicción de Actividades Biológicas.
- Módulo para el Cálculo Distribuido de Mecánica Molecular y Mecánica Cuántica (QMMM) utilizando el programa MOPAC. Este trabajo fue presentado en el 33 Evento Internacional de Químicos Teóricos (QUITEL), con muy buena aceptación.
- Módulo de Cálculo Distribuido para la predicción de interacciones Proteína – Ligando (tamizaje virtual). Este módulo permite realizar grandes simulaciones de acoplamiento que puede tener una molécula pequeña (Ligando) y una Proteína.

A continuación se muestran las corridas realizadas del tamizaje virtual (Figura 2) y una evaluación de los resultados, que demuestran de manera simple la mejora que proporciona el uso del sistema distribuido que se propone, en cuanto al tiempo de procesamiento de los datos fundamentalmente.

PCs Windows	PCs Linux	Total Pcs	Granularidad	Tiempo (seg.)	SpeedUp
0	1	1	100 %	159585	1
3	2	5	1	42890	3,72
11	4	15	1	24012	6,64
40	0	40	1	6991	22,82
34	22	56	1	6159	25,91
68	10	78	1	5599	28,50
77	17	94	1	5152	30,97
89	18	107	1	5268	30,29
98	20	118	1	5603	28,48

Fig. 2. Resultados del procesamiento en el sistema de cómputo.

Todos estos procesamientos se hicieron con granularidad uno (cantidad de ligandos a enviar a un cliente) y el ligando utilizado fue siempre el mismo, lo que es positivo como es lógico, ya que los datos de entrada no variaron, y dicho ligando tiene el código "ZINC06850743" de la base de datos ZINC [3].

Como se pudo apreciar en la figura 2, el *speedup* crece hasta un valor límite a partir del cuál no existe un mayor crecimiento al intentar aumentar el número de procesadores, siendo el aumento de procesadores la fortaleza y debilidad del procesamiento distribuido. Por supuesto que este límite no siempre es el mismo y depende mucho del entorno y de la entrada de datos del problema. En el experimento realizado, se obtuvo un crecimiento casi lineal en el intervalo de 15 y 40 procesadores y con 94 se encontró el límite.

Conclusiones

Se lograron los dos objetivos más importantes, el desarrollo y funcionamiento del sistema, que actualmente cuenta con 300 máquinas, por el momento, para la realización de los cálculos que dan solución a diversos problemas en la Bioinformática. Se encuentra instalada también, en el Centro de Inmunología Molecular (CIM).

Además el trabajo, representa para nuestro país una solución práctica y eficiente a los problemas que presentan diariamente nuestras instituciones científicas, que necesitan un amplio poder de cómputo para desarrollar sus investigaciones.

Por supuesto no intentamos eliminar o pretender aminorar las amplias posibilidades de aplicación de los modelos paralelos o el uso de supercomputadoras, sino de complementar todos los medios disponibles en una gran "supercomputadora virtual".

Referencias Bibliográficas

Keane, Thomas. A General-Purpose Heterogeneous Distributed Computing System. National University of Ireland Maynooth. Julio, 2004.

Aguilera, Longendri. Sistema de Cómputo Distribuido aplicado a la Bioinformática. Universidad de Ciencias Informáticas. La Habana. Febrero, 2008.

ZINC - A free database for virtual screening; Revisado Junio 2007. Disponible en: http://blaster.docking.org/zinc/.

Zomaya AY. Parallel Computing for Bioinformatics and Computational Biology: Models, Enabling Technologies, and Case Studies. Wiley; 2006.

Tanenbaum A, Steen MV. Distributed Systems: Principles and Paradigms. Prentice Hall, Pearson Education, USA; 2002.

G Couloris JD, Kinberg T. Distributed Systems - Concepts and Design, 4th Edition. Addison-Wesley, Pearson Education, UK; 2001.