Introducción a la bioinformática para el análisis de datos de expresión genética

Mhamed Menssouri

Resumen— La comparación de genomas proporciona información sobre los procesos evolutivos que se han llevado a cabo en la aparición de los diversos seres vivos que habitan el planeta tierra. A partir de la comparación entre los genomas, se puede obtener un árbol evolutivo entre las diferentes especies. En el caso de comparación de genomas de bacterias el principal problema es el gran número de genomas a comparar. Este gran número hace que la comparación entre todos los genomas secuenciados de bacteria pueda tardar más de un año. Para superar esta barrera se ha adaptado el sistema existente para obtener el árbol filogenético. En primer lugar se lanzarán las comparaciones entre genomas en paralelo. Por otro lado, se generará el árbol evolutivo cada vez que un genoma es comparado con el resto de genomas previamente comparados, sin necesidad de esperar que se realice la comparación entre todos los genomas. Por último, el sistema dejará de guardar los resultados de las comparaciones entre genomas. En su lugar, cada vez que el usuario requiera estudiar esas comparaciones via web se relanzará el cálculo de la comparación y solo se guardarán los resultados de las comparaciones más consultadas.

Palabras clave— Genómica, Gen, Bacteria, Arquea, Eucariota, Árbol filogenético, Mummy, MUM, SMUM, Matriz de similitud, BigData, Paralelización, IBB, NCBI.

Abstract— The comparison of genomes provides information about the evolutionary processes that have taken place in the appearance of the various living beings that inhabit our planet. From the comparison between the genomes, you can obtain an evolutionary tree among the different species. In the case of the comparison between the bacteria genomes the main problem is the big amount of genomes to be compared. This huge number of genomes makes the comparison between all the sequenced bacteria genomes take longer than a year. To overtake this barrier, we have adapted the existing system to obtain the phylogenetic tree. Firstly, the comparisons between the genomes will be made in a parallel way. On the other side, the evolutionary tree will be generated every time that a genome is compared to the rest of previously compared genomes, with no need to wait for the comparison to be made between the genomes. Finally, the system will stop saving the results of the comparisons between the genomes. Instead, every time that the user requires analyzing those comparisons via web, the calculation of the comparison will be thrown and only the results of the most consulted comparisons will be saved.

Index Terms—Genomics, Gene, Bacterium, Archea, Eukaryote, Phylogenetic tree, Mummy, MUM, SMUM, Similarity matrix, BigData, Parallelization, IBB, NCBI

1 Introducción

La bioinformática con el paso del tiempo se va consolidando como un elemento esencial e indispensable en la actividad y/o investigación médica. La bioinformática es la aplicación de las tecnologías computacionales a la gestión y análisis de datos biológicos.

Día a día los problemas que se abordan en biomedicina requieren de soluciones más complejas y sofisticadas. El volumen de datos disponibles crece de manera exponencial, lo cual abre un sin fín de posibilidades a los investigadores, científicos y entusiastas de la bioinformática.

Para analizar, manipular y extraer información valiosa de estas grandes cantidades de información es necesario aplicar técnicas eficientes, escalables y robustas. El Big Data o datos a gran escala, pretende ofrecer soluciones y técnicas para hacer frente a dichos problemas. Gracias a las nuevas técnicas de procesamiento y análisis de datos masivos se han logrado grandes avances en el estudio del cáncer, alzheimer y muchas más enfermedades.

La actualidad en lo referente a la secuenciación de genomas (obtención del código genético de un organismo), el altísimo coste computacional que comporta trabajar con estos grandes volúmenes de datos (son millones de bases), llevan a considerar que estamos ante un proyecto ambicioso e innovador.

El proyecto desarrollado me ha permitido entender y hacer frente a un problema real que tendrá una aplicación productiva y beneficiosa para una gran comunidad de investigadores, científicos y entusiastas de la biomedicina.

- *E-mail de contacto: mhamed.menssouri@e-campus.uab.cat*

- *Mención realizada: Ingeniería de Computación*
- *Trabajo autorizado por: Jordi González y Mario Huerta (Departamento de ciencias de la computación)*
- *Curso 2015/16*

2 ESTADO DEL ARTE

El genoma de todo ser vivo contiene la información genética de este. Dicho genoma se compone de una secuencia de cuatro bases (A, C, T, G).

La comparación de genomas proporiciona mucha información sobre los procesos evolutivos que se han llevado a cabo en la aparición de los diversos seres vivos que habitan el planeta tierra. Además, esta comparación de genomas tiene una aplicación médica muy importante. Actualmente la asignación de funciones para nuevos genes se suele realizar mediante reconocimiento de subsecuencias funcionales conservadas de un organismo a otro.

Podemos clasificar los genomas en tres tipos: Arqueobacterias, Bacterias y Eucariotas. Los genomas de Eucariotas son muy grandes, a diferencia de los genomas de bacterias y arqueobacterias. El genoma humano, por ejemplo, tiene tres mil millones de bases a diferencia de los tres millones de bases que tendría un genoma de bacteria. En la figura 1 podemos observar los tres grupos de genomas y las diferencias de tamaño entre los grupos expresada en millones de bases. El grupo de color naranja son las Arqueobacterias, el de color azul las Bacterias y los grupos de color verde las Eucariotas.

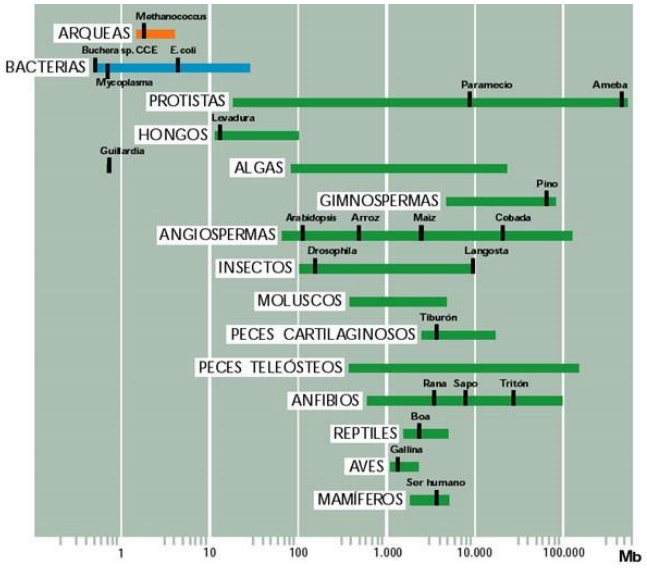


Figura 1: Los tres grupos de genomas y las diferencias de tamaño entre los grupos expresada en millones de bases. El grupo de color naranja son las arqueobacterias, el de color azul las bacterias y los grupos de color verde las eucariotas.

Existen diversos métodos para comparar genomas. La técnica que utiiza la línea de investigación del Instituto de biomedicina y biotecnología (IBB) [1] para la comparación de genomas completos se basa en el cálculo de los Maximal Unique Matchings (MUMs) entre las secuencias de genomas comparados [2]. Los MUMs son las subsecuencias comunes de bases nitrogenadas de mayor longitud y únicas a los genomas comparados. En la figura 2 se muestra un ejemplo de subsecuencia de MUM entre dos genomas.

ACTGACTTAGAT**GATGACCA**TAGTAATGCGCGATCGATCGATCGTA

CGTCAGTCATGCATGCTATATACGATCGA**GATGACCA**ACTGACGAT

Figura 2. Ejemplo de subsecuencia de MUM entre dos genomas.

A partir de los MUMs se calculan los SuperMUMs (SMUMs) que permiten tratar con mayor facilidad los genomas muy grandes como en el caso de las eucariotas. Un SuperMUM es una agrupación de MUMs consecutivos con pequeños "gaps" (subsecuancias diferentes) entre ellos.

Una estructura de datos utilizada para obtener los MUMs entre dos genomas son los suffix tres [3], utilizados por el algortimo MUMS-OL que obtiene los MUMS mientras lee el genoma a comparar [4]. Este algortimo es utilizado por servidores de aplicaciones que comparan secuencias genómicas [5]. El servidor del IBB [6] destinado a la comparación de genomas utiliza el algoritmo MUMS-OL para realizar los cálculos de MUMs.

Las subsecuencias de genomas pueden estar codificadas de izquierda a derecha o de derecha a izquierda. De esta manera se pueden encontrar fragmentos del código genéticio que durante su proceso evolutivo invierten su orden de una especie a otra manteniendo la funcionalidad. Dado este caso los MUMs pueden clasificarse en directos y en inversos.

Todos los genomas secuencializados hasta el momento se pueden obtener accediendo al servidor del National Center for Biotechnology Information (NCBI) [7] del gobierno de los EEUU. En este mismo servidor contiene los genes de cada genoma con las posiciones que ocupa cada gen dentro del genoma.

Para representar las dependencias evolutivas entre genomas se utilizan los árboles filogenéticos, los cuales son árboles que muestran las relaciones evolutivas entre diversas especies con ascendencia común. Estos arboles filogenéticos suelen basarse en rasgos conservados entre las especies. Los arboles filogenéticos proporcionan resultados mucho más precisos cuando utilizan la información obtenida a partir de las comparaciones entre los genomas de los diversos organismos.

El IBB [1] sigue una línea de investigación para la comparación de genomas enteros. Para este propósito se está trabajando para ofrecer un servidor público (http://www.platypus.uab.cat/) que permita consultar vía web las comparaciones entre genomas.

Para poder construir el árbol filogenético basado en la comparación de genomas es necesario disponer de los cálculos previos que permitan obtener la información necesaria acerca de la similitud entre los genomas.
En la línea de invertigación del IBB [1] se denomina preproceso a todos los cálculos previos que luego mostrará la aplicación web. De esta manera se consigue que cuando el usuario final utilize la web todos los cálculos previos ya estén realizados y toda la información necesaria ya esté disponible.

En la línea de investigación del IBB [1] el preproceso se divide en dos fases. La primera fase corresponde a la descarga de genomas desde el servidor del NCBI [7], el cálculo de los MUMs y los SMUMs entre todos los geno-

mas y a partir de estos cálculos la creación matriz de similitud entre genomas. La matriz de similitud tiene como finalidad almacenar la similitud entre cada par de genomas. La similitud entre dos pares de genomas se obtiene a partir de la suma de los MUMs o SMUMs resultantes tanto en dirección directa como en inversa. La segunda fase correponde a la creación del árbol filogenético entre los genomas comparados, la creación de los clústers de genomas más similares y el layout (ubicación en el plano 2D) del árbol filogenético. Su propósito es ser mostrado en el aplicativo web.

Uno de los principales problemas es que la primera fase del preproceso es muy costosa en tiempo de ejecución. En el caso de las Bacterias, existen más de 2700 genomas secuenciados lo cual implica que la fase 1 del preproceso sea muy lenta. La comparación entre todos los genomas de bacterias y arqueas secuenciados hasta el momento, con el sistema actual, puede tardar más de un año en finalizar lo cual es totalmente inviable.

El principal factor que aumenta el coste temporal de la fase 1 del preproceso es que los genomas deben compararse todos contra todos y, además, en inversa y directa. Por otro lado, otro problema importante, es el almacenamiento de los resultados de las comparaciones. Se generan miles de ficheros de resultados por lo que se podría sobrepasar la capacidad de almacenamiento del servidor.

En la segunda fase del preproceso se trabaja a partir de la matriz de similitud entre genomas por lo que las limitaciones de almacenamiento de resultados ya no son un problema y su tiempo de cálculo es reducido.

La descarga de genomas del NCBI [7] debería ser automática para que la base de datos local del aplicativo esté siempre actualizada. Actualmente el sistema de descarga solamente se activa bajo petición manual.

En el servidor del IBB [6] se puede encontrar la interficie gráfica web, bautizada como Mummy, que permite la visualización de las comparaciones entre múltiples genomas. La inteficie gráfica permite estudiar y analizar en detalle los MUMs y SMUMs entre pares de genomas. En el apéndice 1 se puede observar un ejemplo de la interficie web Mummy para la comparación de los genomas Arquea. Las líneas verdes corresponden a los MUMs y las líneas rojas corresponden a los SMUMs y, por útlimo, también se pueden observar los genes de los genomas comparados.

En el servidor del IBB [6] una interficie web diferente muestra el árbol filogenético de todos los genomas comparados. En el apéndice 2 se puede observar un árbol filogenético mostrado desde la aplicación web.

Actualmente para la interfaz Mummy es necesario almacenar los MUMs y SMUMs de cada comparación para la muestra de resultados. Tal y como se ha comentado anteriormente el almacenmiento de todos los MUMs entre los genomas comparados implica un gran volumen de ficheros lo cual no es viable.

3 OBJETIVOS

Los objetivos que pretendemos entonces se separan en dos bloques. Los cálculos del pre-proceso y el "pipeline" de ejecución.

En el pre-proceso se generan las estructuras de datos relativas a MUMs, SuperMUMs, genes e información estadística necesaria para las interfaces web que mostrarán graficamente los resultados de la comparaciónes entre genomas y el árbol filogenético. En cuanto al pipeline nos referimos al flujo y/o orden de ejecución de los procgramas que componen el backend del servi-dor.

De esta forma pretendemos,

- Paralelizar la comparación de genomas. Se cuenta con 24 núcleos de proceso los cuales se deben aprovechar de manera eficiente. Por ello es necesario gestionar cuantos cálculos se lanzan en paralelo de forma que la eficiencia sea máxima sin colapsar el sistema.
- Reestructurar el preproceso. Construcción del árbol filogenético a partir de los resultados parciales de la fase 1. Es decir, sin disponer todavía de la matriz de similitud entre todos los genomas. De esta forma el usuario siempre podrá disponer del árbol filogenético entre los genomas comparados hasta el momento
- Que todas las modificaciones aplicadas sean correctas. El correcto funcionamiento de los procesos del pipeline implica la generación de los resultados esperados en el formato esperado ya que las aplicaciones que utilizarán dichos resultados los re-quieren con un formato concreto.
- No guardar ficheros resultados de las comparaciones con los MUMS y SMUMs entre genomas. Cuando el usuario solicite via web el detalle de una comparación entre genomas, esta comparación se calculará de nuevo y solo se almacenarán los resultados de las consultas habituales.
- Crear un proceso automático de descargas de nuevos genomas de manera periódica desde el servidor del NCBI [7] para incluirlos en los datos locales a la aplicación. Además de la descarga de los genomas se deberá lanzar todo el pre-proceso para los nuevos genomas y de esta manera incluirlos a los resultados que se muestran mediante las interfaces web.

4 RESULTADOS Y DISCUSIÓN

En este apartado analizaré todo el trabajo realizado para obtener los resultados esperados. Detallaré porqué las decisiones tomadas son mejores o más adecuadas que las ya existentes o porqué son mejores que las alternativas disponibles.

En cada una de la secciones realizaré un análisis a nivel funcional de cada uno de los procesos trabajados a lo largo del proyecto. En dichas secciones expondré las alternativas probadas y/o planteadas que se descartaron y porqué motivo.

4.1 Métodos de evaluación de los resultados de los procesos

Desde el incio del proyecto he definido cual será el mecanismo a utilizar para la evaluación de los resultados obtenidos antes y después de aplicar cualquier modificación en los procesos. Tal y como se ha indicado anteriormente, cada proceso genera sus propios resultados en una ubicación y formato concreto. Cada proceso requiere de los resultados de su proceso antecesor. Esto significa que en el momento en el que un resultado se genere incorrectamente toda la cadena de procesos se vería afectada. Por esto es muy importante la fase de validación de resultados después de cada modificación, por muy pequeña que pueda resultar la modificación aplicada.

Es obvio que si las validaciones no se realizan en cada paso del desarrollo, cada vez resulta más complicado dar con el error o punto que provoca la generación de resultados incorrectos.

Para validar las diferentes versiones de los procesos decidí comparar los resultados generados por cada versión mediante la comparación de los ficheros y directorios generados. Dicha comparación se realizaba de manera automática utilizando comandos del sistema que permiten la comparación de directorios y ficheros e indican las diferencias entre estos. Este sistema de evaluación de resultados resulta más práctico que realizar pruebas de caja negra o de caja blanca. El motivo de esta elección es debido a que lo realmente importante es que los procesos generen los resultados correctos, con los datos correctos y con el formato correcto. En el caso de las pruebas de caja negra y caja blanca se evalúa también lo robusto que es el sistema y la manera en la cual se ha implementado la solución que genera dichos resultados, lo cual para mi caso no era tan relevante.

Antes de aplicar una nueva funcionalidad en los procesos se barajaban cuales eran las alternativas disponibles para conseguir dicha funcionalidad. En el caso de que fuese una modificación de algún elemento ya existente, primero analizaba porqué la alternativa existente no era aceptable. Una vez analizados todos los “pros” y “contras”, junto a las respectivas pruebas, de las diferentes alternativas escogía la que más se adecuaba a las necesidades del proceso en conjunto.

La primera tarea de validación de resultados fue validar los procesos de los cuales partiría mi trabajo. Disponía de una carpeta con resultados correctos, los cuales, ya se utilizaban para mostrar la información en las aplicaciones web. Lanzé los procesos de la aplicación y validé que los resultados de esta versión eran los mismos que la carpeta de resultados fiables. Esto me ha permitido estar seguro de partir desde una versión funcionalmente correcta. Para validar los resultados he utilizado el mismo sistema de comparación de ficheros y directorios.

4.2 Lanzamiento en paralelo de las comparaciones entre genomas

Este ha sido el primer paso que he dado en el proyecto. La versión desde la cual partía era una versión de comparaciones secuenciales. Las operaciones secuenciales cuentan con la ventaja de que no necesitan una gestión compleja y se pueden aplicar medidas de control de errores sencillas y eficaces. El gran problema de las operaciones secuenciales es que no aprovechan los recursos de la máquina en la cual se ejecutan. En mi caso dispongo de un servidor de 24 núcleos de procesamiento. En las comparaciones secuenciales de esta cantidad de núcleos de procesamiento solamente se está utilizando uno ya que solamente existe un proceso de comparación en curso.
En el caso de las comparaciones secuenciales de 56 Bacterias, el proceso de comparación tardaba casi unas 3 horas en finalizar. Este es el motivo principal por el cual resulta inviable mantener le metología secuencial para obtener los resultados de las comparaciones.

Dada esta situación y el objetivo inicial de paralelizar el lanzamiento de las comparaciones entre genomas decidí ponerme manos a la obra. Es importante tener en cuenta que ha cada comparación le sigue un post-proceso de los resultados generados, es decir, cuando una comparación entre genomas ha finalizado y se han generado los MUMs correspondientes se les debe aplicar un post-tratamiento. Es obligatorio esperar a que la comparación finalize para que a sus respectivos resultados se les aplique dicho tratamiento. Por lo tanto, la comparación y el post-tratamiento de los resultados de la comparación se pueden comprender como una única unidad con dependencias entre sí. Primero la comparación y después el tratamiento de los resultados.

Pasar de un flujo secuencial a un flujo paralelo implica muchos cambios. Uno de los factores diferenciales entre el flujo sencuencil y paralelo es la gestión de los recursos. Por recursos me refiero a todos los ficheros que se utilizan, directorios a los que se accede, objetos almacenados en memoria, escritura y modificación de ficheros en disco, etc.

El sistema de paralelización que utilicé es el de creación de subprocesos clones. Esta estrategia consiste en que llegados a un punto del proceso principal (a partir de ahora le llamaremos proceso padre) se crean procesos identificados como hijos del actual. Estos procesos son clones del padre. Por lo tanto, la línea que he seguido ha sido la creación de procesos hijos en cada momento de lanzar una nueva comparación. Dichos procesos hijos serían los encargados de la comparación de un par de genomas, ya conocidos y comunicados por el proceso padre, y el respectivo post-tratamiento de los resultados generados de esa comparación. Este sistema permite lanzar tantos procesos independientes como se desee y además favorece a la gestión del sistema operativo, lo cual no sería posible con una estrategia basada en hilos de ejecución.

Inicialmente los genomas se emparejaban siguiendo el orden que se seguiría al llenar una matriz cuadrada por columnas, es decir, emparejando el genoma 1 con todos los restantes y el 2 con todos los restantes menos el 1, y así sucesivamente. En el caso de la comparación de 3 genomas, primero se compararía el genoma 1 con el 2 y con el 3 y en la siguiente y última iteración se compararía el 2 con el 3. Este orden de emparejamiento de genomas no

era el más adecuado para nuestros objetivos ya que las primeras iteraciones del proceso de lanzamiento de comparaciones eran las más costosas debido a que eran las iteraciones en las cuales mas pares de genomas se comparaban. Para mis objetivos lo que buscaba era justo lo contrario, poder calcular de la manera más rápida posible las primeras iteraciones y de esta manera poder tener resultados parciales que mostrar al usuario de la aplicación web. Recordemos que las dos interficies web, tanto la Mummy como la que muestra el árbol filogenético, se alimentan de los resultados proporcionados por el proceso de comparación de genomas. Por lo tanto finalmente procedí a modificar el sistema de emparejamiento de genomas a un sistema que permita llenar una matriz cuadrara por filas. Este nuevo sistema emparejaría el genoma de la iteración activa con sus anteriores, es decir, el genoma 2 con el 1, el genoma 3 con el 2 y con el 1, el genoma 4 con el 3, con el 2 y con el 1 y así sucesivamente. Este sistema ha permitido obtener resultados parciales para las interficies web mucho más rápido debido a que las primeras iteraciones finalizan muy rápido gracias a los pocos emparejamientos resultantes.

En resumen, lanzado en un proceso independiente la comparación entre un par de genomas he permitido que el proceso padre pueda desvincularse de las comparaciones, encargando a los procesos hijos dichos cálculos. Por otro lado he invertido el orden de los emparejamientos de los genomas para su comparación siguiendo un orden de llenado de una matriz cuadrada por filas. Esta última modificación me ha permitido preparar el proceso de comparaciones de genomas para conseguir el objetivo de lanzar la fase 2 del pre-proceso cada vez que se finalizen las comparaciones de un genoma con sus anteriores.

4.3 Gestión y control de errores de las comparaciones lanzadas en paralelo

En este apartado analizaremos cual ha sido el método empleado para la gestión de los procesos de comparación lanzados en paralelo, además de ver cual ha sido la clave para el control de los procesos de comparación con problemas durante la ejecución.

Para esta fase del proyecto partímos de la versión que permite lanzar las comparaciones en paralelo. Es obvio, que la versión de partida para esta fase ha sido validada previamente.

Para conseguir el objetivo de paralelizar los lanzamientos de comparaciones entre genomas no es suficiente con lanzar procesos que se ejecutan en paralelo. Es indispensable implantar medidas de control que permitan controlar el flujo de ejecución de las comparaciones. La primera medida de control que se debía implantar venía precedida de la necesidad de saber en que momento todos los lanzamientos paralelos habían finalizado. Inicialmente, para facilitar el desarrollo de la tarea, consideré que todos los procesos finalizarían correctamente. Para este control inicialmente utilicé operaciones del sistema operativo que permiten al proceso padre esperar la finalización de todos sus hijos. El mecanismo que proporciona el sistema operativo es sencillo de utilizar, pero propenso a errores. Un posible caso en el cual este sistema no funcionaría sería cuando un proceso hijo jamás finalizase por algún fallo en la ejecución. La función de espera se quedaría esperando a la finalización del hijo, lo cual jamás ocurriría. Otra posible situación que podría provocar el fallo de este sistema de espera sería cuando dos hijos finalizan en el mismo instante de tiempo, lo cual es perfectamente posible en lanzamientos de procesos paralelos. En el caso del BigData hay que analizar con sumo cuidado todas las situaciones que podrían darse ya que al ser BigData, si una situación puede darse, se dará. Para hacer frente a esta situación característica del BigData decidí replantear la forma mediante la cual realizaba la espera de los procesos hijos pendientes de finalizar. La alternativa que analizé se basaba en comprobar la existencia de los ficheros esperados después de las comparaciones lanzadas para una fila de la matriz de similitud. Para cada nueva fila de la matriz de similitud podía saber cuantos ficheros de resultados se debían generar una vez finalizados todos los lanzamientos paralelos. En el caso de estar procesando la fila 3 de la matriz de similitud, el número de ficheros esperados sería de 4 ya que el genoma 3 se emparejaría con el 2 y con el 1 y para cada emparejamiento se realizaría una comparación en dirección inversa y directa. Gracias a este mecanismo de cálculo de ficheros esperados siempre se puede saber exactamente cuantos ficheros se esperan para el momento en el que todas las comparaciones han finalizado. Una vez validado este sistema de cálculo de ficheros esperados he procedido a modificar el sistema de control basado en herramientas del sistema operativo. Este nuevo método de control permitirá estar seguros al 100% de que todos los ficheros esperados se han generado correctamente ya que es lo que realmente se necesita. Por otro lado este nuevo mecanismo nos permitirá saber que resultados no se han generado y a que comparación corresponden. La condición para que el padre continúe con la siguiente fila de la matriz de similitud será que se hayan generado todos los ficheros esperados o que se haya sobrepasado el tiempo límite permitido para la generación de nuevos resultados. En el caso de las comparaciones de bacterias o arqueas las comparaciones tardan entre 1 y 4 segundos en finalizar. Si en el momento de comprobar los resultados en 6 segundos no se ha generado ningún resultados nuevo, y además no se ha alcanzado el número de resultados esperados, el proceso padre pasa a la siguiente fase de lanzamiento secuencial de comparaciones erroneas, la cual explicaremos a continuación.

Antes de analizar la fase de lanzamientos secuenciales de las comparaciones fallidas explicaré como se gestiona la finalización de una comparación que ha fallado y que no ha podido finalizar de la manera esperada. Los procesos que se lanzan en paralelo se lanzan con un temporizador activado. Dicho temporizador actúa como alarma, la cual está programa para que pasado un tiempo establecido se activé y se trate la señal emitida. Cuando un temporizador se agota o expira el sistema de control de procesos “mata” al proceso que se le ha expirado el tiempo. Este mecanismo permite que ningún proceso se quede “colgado” y por lo tanto consuma recursos o pueda pro-

vocar problemas en el servidor además de asegurarnos que pasado un cierto tiempo todos los procesos lanzados finalizarán su ejecución por alguna vía.

En la fase de lanzamiento secuencial de comparaciones solamente se entrará cuando no se han generado todos los resultados esperados. En esta fase se identifica cuales son los resultados que faltan y que comparaciones son las asociadas a dichos resultados para poder relanzarlas. Una comparación puede finalizar con problemas por diversos motivos. Controlar cada uno de los casos sería imposible ya que el trabajar con BigData los posibles casos serían infinitos. El hecho de relanzar en solitario un proceso de comparación, en muchos casos, permitirá que el proceso finalice correctamente debido a que el contexto en el que se lanza es totalmente diferente al contexto en el que falló. Por lo tanto, todas aquellas comparaciones fallidas se identifican mediante los resultados que se esperan pero que no existen y se lanzan secuencialmente. Cada comparación se intenta relanzar un máximo de 3 veces.

4.4 Creación de la matriz de similitud a medida que se comparan los genomas

Inicialmente en la versión de lanzamiento secuencial de las comparaciones la matriz de similitud no se generaba hasta que no se hubiesen comparado todos los genomas almacenados. Esta estrategia corresponde a un sistema de procesamiento de datos offline. En este proyecto uno de los objetivos era lograr implantar sistemas que trabajen bajo demanda o que generen resultados parciales para siempre disponer de resultados para las interfaces web y de esta manera poder acortar la espera del usuario de la aplicación web.

Para la versión de lanzamiento de comparaciones en paralelo la matriz de similitud se pasa a generar por filas, siguiendo la línea de emparejamiento de los genomas explicado en el apartado anterior. La generación por filas de la matriz de similitud permite que a cada iteración del proceso de comparaciones la aplicación disponga de la matriz para poder lanzar la fase 2 del preproceso y de esta manera poder generar el árbol filogenético con resultados parciales. La matriz de similitud necesita que todos los resultados de las comparaciones existan lo cual se consigue gracias a las medidas de control de procesos paralelos descritos en el apartado anterior.

Una vez generada la matriz de similitud todos los programas de la fase 2 del preproceso pueden ser lanzados, ya que lo único de la fase 1 que necesitan para realizar sus respectivos cálculos.

4.5 Control del uso de los procesadores del servidor

Esta tarea ha sido la más compleja y costosa del proyecto. Para la paralelización de procesos es indispensable disponer de algún sistema de control del consumo de los recursos del servidor. Para mi caso que he trabajado con Bacterias y Arqueas la complejidad no residía en la comparación en sí, si no en la cantidad de genomas existentes y en la cantidad de comparaciones en paralelo que se podrían llegar a lanzar si no existiese ningún mecanismo de control. El servidor en el cual se aloja todo el aplicativo, tanto el entorno de desarrollo como el de producción, es compartido con otras aplicaciones y procesos. La hipotética caída o colapso del servidor tendría consecuencias graves para todas las aplicaciones y procesos activos en el servidor, además del coste que conllevaría poner en marcha desde 0 un servidor de estas características.

Desde el inicio los requisitos no funcionales para esta tarea eran claros: fiabilidad y eficacia. El mínimo error en la estrategia de control de consumo de recursos podría provocar el colapso del servidor. Por la tipología de las comparaciones entre genomas de mi proyecto, el recurso que había que controlar eran núcleos de procesamiento (Cpu) ya que para las Bacterias y Arqueas lo realmente crítico es la cantidad de comparaciones a lanzar en vez del consumo de memoria para los cálculos. El servidor cuenta con 24 núcleos de procesamiento.

Inicialmente utilicé un método ya utilizado en otros módulos del aplicativo. El sistema utilizado, aparentemente, recuperaba el consumo de cpu. La información que se requería para el cálculo se obtenía de archivos del sistema operativo. En dichos recursos del sistema operativo se guarda la información del estado actual de todos los recursos físicos del sistema. Para el cálculo del consumo actual de cpu se obtenían los datos de las cpus en un instante t0, seguidamente, se realizaba una espera de 1 segundo y se volvían a recuperar los datos de las cpus. Al final se aplicaba una fórmula para calcular el consumo en función de los valores recuperados en t0 y en t1.

Los valores obtenidos mediante la aplicación de este método basado en el incremento de consumo de la CPU no eran fiables. El sistema operativo ofrece una funcionalidad que recupera el estado actual del uso de la memoria y de la cpu. Los valores de mi fórmula se contrastaron con los valores de dicho proceso del sistema y no coincidían en muchos casos. Inlcuso había momentos en los que el método basado en incremento de uso del procesador provocaba errores de división entre 0. Esto se debe a que en las dos lecturas secuenciales del archivo de sistema no se reflejaban diferencias y los valores recuperados en t0 y t1 eran los mismos.

Finalmente se encontró un modo alternativo para la obtención del consumo de la cpu. La función del sistema operativo recupera mediante un refresco de 2-3 segundos el uso de memoria y CPU, además de otra información. Lo que decidí hacer es invocar a dicho proceso y procesar el resultado generado para almacenar solamente el valor alojado en el campo “CPUs”. Este método ha permitido poder trabajar con datos fiables pero con una penalización en el tiempo de ejecución notable. Para recuperar los valores fiables se debe esperar al refresco de los datos, lo cual implica una espera de 2 o 3 segundos en el peor de los casos. Al trabajar con una cantidad enorme de genomas la penalización aumenta considerablemente. Para un lanzamiento de comparación de 100 genomas se realizarían como mínimo 100 consultas de consumo de cpu lo cual implica una espera de 300 segundos en el peor de los casos. Para 1000 genomas la penalización sería de 3000 segundos en el peor de los casos. Esta penalización se asumió debido a que colapsar el servidor tendría conse-

cuencias más graves.

Tal y como se ha descrito en apartados anteriores, es el proceso padre el cual lanza los n procesos de comparación en paralelo. El coste de la tarea del proceso padre es muy pequeña ya que solamente indica que pares de genomas se deben comparar y lanza el proceso de comparación. El padre por otro lado, también es el encargado de controlar el uso de la cpu para no lanzar demasiados procesos y en consencuencia colapsar el servidor. Para decidir si se debe pausar el lanzamiento de procesos paralelos se ha utilizado un método de estimación de consumo de cpu por cada proceso lanzado. Para obtener el valor estimado del consumo de cpu de cada hijo se lanzaron 80 procesos paralelos, lo cual es el número mínimo de hijos necesarios para colapsar el servidor. Dicho número de procesos se definió gracias a la monitorización del estado del servidor mediante las funciones del sistema operativo. Finalmente para obtener el consumo de cpu por proceso lanzado se dividió el 100% del uso de la cpu entre los 80 procesos, obteniendo así, un consumo estimado de 1.25% por proceso lanzado.

He fijado un valor constante que indica cual es el máximo de consumo de cpu permitido. Este valor es de 80. El proceso padre cada vez que inicia el cálculo de una nueva fila de la matriz de similitud recupera cual es el estado de la cpu en dicho momento. Cada vez que se lanza una nueva comparación en paralelo se incrementa el uso de la cpu con el valor estimado de consumo de cpu por proceso lanzado. Cuando se ha superado el límite de consumo de cpu el proceso padre entra en un bucle del cual no sale hasta que el consumo de cpu no es inferior al máximo permitido.

4.6 Fase 2 del pre-proceso. Creación del árbol filogenético a medida que se comparan los genomas

La fase 2 del pre-proceso ha implicado pocos cambios en la versión original pero me ha ayudado a solucionar pequeños problemas en los resultados generados por la fase 1.

He partido de una versión ya existente del programa de lanzamiento de los programas de la fase 2. Lo primero que he realizado es validar el correcto funcionamiento de esta versión comparando los resultados generados con los resultados ya exsitentes, los cuales si eran correctos.

La fase 2 es la encargada de generar los resultados necesarios para que la interface web que muestra el árbol filogenético pueda mostrar la información correctamente. De entre los procesos que se lanzan desde la fase 2 solamente he necesitado modificar el encargado de dividir el factor de similitud. En cuanto al proceso principal que invoca al resto de procesos de la fase 2 (a partir de ahora proceso principal) he realizado varios cambios. Uno de los cambios ha consistido en mover el proceso de mapeado de genes a la fase 1 del pre-proceso, junto al proceso de descarga de genomas del servidor del NCBI. Otro cambio ha implicado indicar al proceso principal el número de genomas procesados hasta el momento, ya que esta información la necesitan los procesos que se invocan. Este cambio es necesario ya que ahora el árbol filogenético se genera cada vez que se añade una fila en la matriz de similitud. El anterior "pipeline" de ejecución no generaba el árbol hasta que no hubiesen finalizado todas las comparaciones entre genomas lo cual facilitaba la gestión. El último cambio aplicado en el proceso principal ha sido mover el proceso que calcula la matriz de similitud en el caso que no exista a la fase 2 del pre-proceso. Esto ha sido necesario ya que ahora la fase 1 comprende únicamente la comparación entre genomas. Es desde el proceso de comparación de genomas desde donde se invoca la fase 2 del pre-proceso.

En el apéndice 1 se puede observar el resultado de la interficie que muestra el árbol filogenético para genomas Arquea.

4.7 Generación de MUMs y SMUMs cuando se solicita la comparación en detalle vía web

Uno de los problemas a solucionar respecto a los sitemas existentes es la posibilidad de ofrecer sistemas que actúan y operan bajo demana, ofreciendo respuestas y resultados bajo petición.

En el caso de la interfaz Mummy, la cual muestra en detalle los MUMs entre dos pares de genomas y sus respectivos genes, requiere de los ficheros de MUMs para mostrar dicha información. Al inicio del proyecto los ficheros de MUMs resultantes de la comparación de todos los genomas se guardaban en el servidor para asegurar su disponibilidad. Esta estrategia es inviable cuando se trata de BigData ya que la cantidad de datos que se almacena es muy grande. Para solucionar este problema de poner al límite la capacidad de almacenamiento del servidor decidí que los archivos de MUMs ya no se guardarían más. Cuando un usario de la aplicación web solicitase los resultados de una comparación entre genomas de Bacterias o Arqueas la web invocaría al proceso de comparación indicándole que genomas se deben comparar. Las comparaciones de este tipo de organismos, tal y como hemos indicado a lo largo del artículo, no son muy costosas en tiempo por lo que se podría ofrecer al usuario una respuesta en un tiempo aceptable y sin la necesidad de almacenar los MUMs de todas las comparaciones. Además, basándome en el principio de ingeniería del software que afirma que sí se ha realizado una consulta por una persona es casi 100% seguro que se volverá a realizar de nuevo, ya sea por la misma persona o no. Aplicando este principio decidí que los ficheros de MUMs generados a partir de las peticiones via web no se borrarían si no que se guardarían para futuras consultas y de esta manera se evitaría lanzar las mismas comparaciones varias veces. En el apéndice 2 se puede observar un ejemplo de la interificie gráfica Mummy para la comparación de dos genomas Arquea. Las líneas rojas corresponden a los MUMs y SMUMs inversos y las líneas verdes a los directos. En el mismo resultado también se pueden observar los genes de los genomas comparados. Las Arqueas que se han comparado para obtener dichos resultados corresponden a Pyrococcus_horikoshii_OT3_uid57753 y Pyrococcus_furiosus_COM1_uid169620.

4.8 Robot de descarga periódica de genomas del servidor del NCBI

Para la creación de este sistema automático y periódico de descarga de nuevos genomas desde el servidor del NCBI he utilizado el mismo mecanismo de alarmas empleado para "matar" los procesos de comparación fallidos. El robot de descarga de genomas pasa a ser un proceso que siempre está operativo pero sin realizar ninguna tarea funcional. El periodo de descarga de nuevos genomas lo he establecido mensual. En todo el mes el robot de descarga de genomas permanece en StandBy hasta que el temporizador expira. Una vez la alarma envía la señal de finalización del periodo de espera se lanza el proceso de descarga de genomas del servidor del NCBI y el posterior tratamiento. Una vez finalizado el proceso al completo el programa reestablece el temporizador y vuelve al estado de espera.

5 METODOLOGÍA

En este apartado explicaré los programas que he modificado y que modificaciones he realizado.

5.1 Lanza_mums.cc

En este programa he aplicado todos los cambios referentes a la paralelización de las comparaciones, el control de la finalización de los procesos paralelos, el lanzamiento secuencial de procesos erróneos, creación de la matriz de similitud y lanzamiento de la fase 2 del pre-proceso.

Para la creación de procesos hijos e utilizado forks de C++. Para las esperas de procesos hijos he utilizado la función wait(). Para las pausas y esperas de tiempo de los procesos he utilizado la función sleep(X). Para saber si todos los ficheros esperados se han generado he utilizado una función implementada por mí, llamada boolean file_exists(fileName). Esta función utiliza el comando "stat" para saber si un fichero existe o no. Para saber si tenemos el número de resultados esperados cuanto la cantidad de ficheros alojados en el directorio de salida con el comando de Unix "ls directorio | wc –l". Para la suma de los fichero de mums directors e inversos utilizo varios comandos concatenados y comunicados mediante Pipes. Para el control de alarmas de los procesos lanzados he utilizado la función alarm() la cual lanza la señal SIGALARM y activa la función alarma() que mata al proceso que ha lanzado la señal. Para ejecutar la fase 2 después de completar una nueva fila de la matriz de similitud he utilizado la función system() para invocar al programa lanzadera.sh y pasarle los parámetros que necesita. He añadido una nueva fase en la cual el fichero genes.txt se genera justo antes de lanzar la fase 2 del preproceso obteniendo solamente el nombre de los genomas comparados.

En cuanto a los parámetros del programa lanza_mums.cc se ha realizado una modificación leve. Antes se requería recibir por parámetro la ruta absoluta de la carpeta donde se encontraban los genomas. Ahora le programa únicamente necesita que se le indique cual es la carpeta principal de trabajo, es decir, Bacteria56 o Bacteria770, etc.

5.1 Lanzadera.sh

En este programa las modicaciones han sido meramente estructurales. He alterado el orden en el cual se llaman a los programas y algunos parámetros en las invocaciones a los distintos programas de la segunda fase del preproceso.

La primera modificación ha consistido en mover el programa de mapeo de genes a la fase previa al lanzamiento del lanza_mums.cc . Esta modificación se debe a que el mapgenes.cc forma parte de la fase de descarga de los genomas del servidor del NCBI. Otra modificación ha sido pasar la llamada del programa crea_factors.cc a la fase 2 del pre-proceso para poder invocar a este programa cada vez que se necesite después de la llamada desde el lanza_mums al programa lanzadera.

En cuanto a los parámetros del programa he añadido un parámetro opcional para que el programa lanzadera pueda recibir desde las llamadas del lanza_mums.cc el número de genomas que se han procesado hasta el momento y de esta manera poder comunicárselo a los programas que necesitan de esta información.

6 CONCLUSIONES

A pesar de las dificultades que se han dado a lo largo de todo el desarrollo del proyecto se puede concluir que todos los objetivos establecidos al inicio del proyecto han sido logrados.

La paralelización en los lanzamientos de los procesos de comparación se ha realizado con éxito. El proceso de comparación de genomas queda totalmente paralelizado y con todos los sistemas de control necesarios para garantizar la robustez del sistema. Se ha logrado que la comparación que las comparaciones de genomas de Bacterias y Arqueas se realicen en tiempos muy inferiores en comparación a las versiones originales. Se ha logrado optimizar el uso de los recursos que ofrece el servidor del proyecto.

En cuando a la fase 2 del pre-proceso se ha logrado integrar perfectamente los lanzamientos desde el proceso de comparación de genomas. Ahora ya es posible generar el árbol filogenético a partir de resultados parciales de la fase 1 del pre-proceso.

La interfaz Mummy ya dispone de un sistema de generación de resultados bajo demanda, lo que pemitirá no sobrecargar el servidor con grandes cantidades de ficheros.

El pre-proceso ya cuenta con un robot de descarga automática de nuevos genomas desde el servidor del NCBI.

En cuanto al proyecto en conjunto ha sido un éxito y me ha supuesto una gran satisfacción el hecho de haber sido el protagonista de todas estas mejoras en el proyecto del IBB.

Gracias a este proyecto he descubierto que la bionfirmática es una disciplina muy interesante y atractiva. He aprendido mucho sobre BigData y sobre las dificultades que conlleva trabajar con volúmenes tan grandes de datos.

AGRADECIMIENTOS

Me gustaría mostrar mi más sincero agradecimiento a todas aquellas personas que me han apoyado y ayudado a lograr la finalización del presente proyecto.

Especial mención al co-tutor del proyecto Mario Huerta ya que ha sido mi principal apoyo y guía en los momentos más delicados del proyecto. Siempre ha estado a mi disposición para ayudarme con todo aquello que he necesitado.

Mi familia y amigos más cercanos han sido la fuente de energía y motivación que he necesitado para superar todos los obstáculos. Sin su presencia todo el proceso habría sido mucho más difícil.

BIBLIOGRAFIA

[1] http://ibb.uab.cat/ibb, Web del Instituto de Biotecnología i Biomedecina de la Universitat Autònoma de Barcelona.

[2] Alignment of Whole Genomes. A.L. Delcher, S. Kasif, R.D. Fleischmann, J. Peterson, O. White, and S.L. Salzberg, Nucleic Acids Research, 27:11 (1999), 2369-2376.

[3] Suffix Tree Construction with slide nodes. Mario Huerta. Technical report LSI-02-63-R. Dept. Llenguatge i Sistemes Informàtics, Universitat Politècnica de Catalunya (2002).

[4] Efficient space and time multicomparison of genomes. Mario Huerta, Xavier Messeguer. Research Report LSI-02-64-R. Llenguatge i Sistemes Informàtics, Universitat Politècnica de Catalunya (2002).

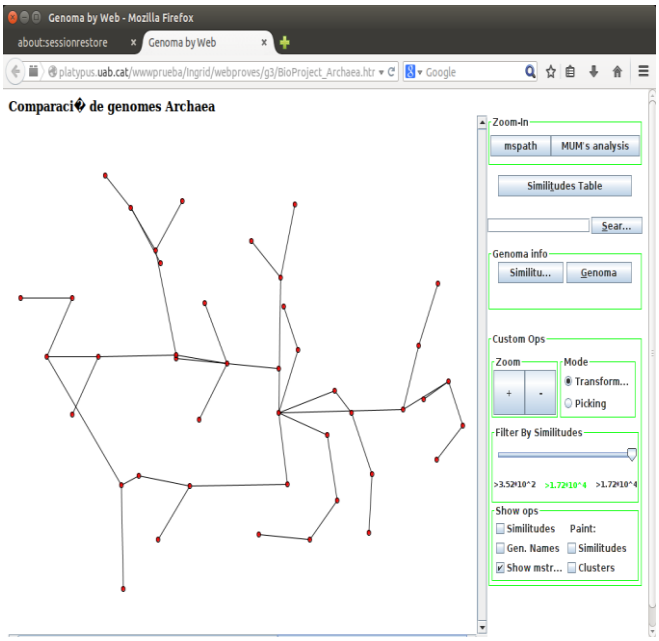
[5] Identification of patterns in biological sequences at the ALGGEN server. PROMO and MALGEN. Domènec Farré, Romà Roset, Mario Huerta, José E. Adsuara, Llorenç Roselló, M. Mar Albà, Xavier Messeguer. Nucleic Acids Research 31(13): 3651-3653 (2003).

[6] [6] http://platypus.uab.cat/, Web server for the all-known-genomes comparison by web. Server supported by the Institute of Biotechnology and Biomedicine of the Autonomous University of Barcelona (IBB-UAB).

[7] [7] WebServer del NCBI (National Center for Biotechnology Information). http://www.ncbi.nlm.nih.gov/

APÉNDICE

A1. INTERFAZ GRÁFICA DEL ÁRBOL FILOGENÉTICO



A2. INTERFAZ GRÁFICA MUMMY