Universidad Central "Marta Abreu" de Las Villas
Facultad de Matemática – Física – Computación
Departamento de Ciencia de la Computación

Arquitectura de un centro de ciencia basado en clúster de computadoras para la visualización científica

Tesis presentada en opción al grado científico de
Máster en Ciencia de la Computación

Autor: Romel Vázquez Rodríguez

Tutor: Dr. Carlos Pérez Risquet

Santa Clara
2008

Resumen

El surgimiento de los centros de ciencia, donde se almacenan y manipulan grandes volúmenes de datos del orden de los peta bytes, compulsa el desarrollo de nuevas tecnologías, técnicas y escenarios para manipular estos datos. La visualización científica se ha desarrollado en los últimos años como un área importante de la Computación Gráfica, que simplifica el análisis, la comprensión y la comunicación de modelos, datos y conceptos en la ciencia y la ingeniería. Las diferentes técnicas de visualización proporcionan a los científicos representaciones visuales adecuadas para mostrar las correlaciones internas de los datos y constituyen una alternativa eficaz para el análisis visual de grandes volúmenes de datos. La concentración de datos de dominios específicos en centros de ciencia permite su acceso y manipulación de forma masiva por científicos de todo el mundo, incrementando las posibilidades de intercambio de información y resultados. En este trabajo se presentan las principales características de los centros de ciencia de mayor relevancia en la actualidad; se estudian los elementos fundamentales de la visualización científica, en particular la visualización distribuida, y se desarrolla un modelo de arquitectura de centro de ciencia, basada en clúster de computadoras, donde se integra la visualización científica con los centros de ciencia.

Abstract

The emerging of science centers, where huge data volumes of peta bytes order are stored and manipulated, collate the development of new technologies, techniques and sceneries to manipulate these data. The Scientific Visualization has been developed as an important area of the Computer Graphics, which simplifies the analysis, understanding and communication of models, data and concepts in science and engineering. Different visualization techniques give to the scientist the adequate visual representation for show the internal correlation of the data and it constitute an efficacious alternative for the visual analysis of huge data volumes. The collect of data of a specific domain in science centers allow the access to them and the massive manipulation by the scientist of the entire world, increasing the possibilities of information and results interchange. In this work is presented the main features of the more relevant science centers at present, the principal elements of scientific visualization, in particular distributed visualization and we develop an architecture model of a cluster-based science center that allows the integration of scientific visualization with the science centers.

Índice

1 Introducción

La Visualización Científica (VC) ha sido un área de investigación de interés creciente en los últimos años, motivado fundamentalmente por el incremento constante de los volúmenes de datos, generados en muchos campos de aplicación.

El desarrollo de las redes de comunicación y el surgimiento de Internet han ampliado la colaboración entre los científicos del mundo en determinadas áreas de la ciencia. Gracias al desarrollo del hardware, unido a la acumulación del conocimiento, se están generando a diario volúmenes de datos tan grandes y complejos, que no pueden ser analizados suficientemente en forma numérica por simples computadoras personales. Estos volúmenes de datos del orden de los peta bytes requieren de un nuevo estilo de trabajo. Como respuesta a esta problemática han surgido Centros de Ciencia, que brindan acceso tanto a los datos como a las aplicaciones que los analizan para algunos dominios científicos. En (Gray et al., 2005) se mencionan algunos ejemplos de estos centros de ciencia como: SDSS *(Sloan Digital Sky Survey)* en Fermilab (Laboratorio Nacional del Acelerador Fermi), BaBar en SLAC (*Stanford Linear Accelerator Center*), BIRN *(Biomedical Informatics research Network)* en SDSC (*San Diego Supercomputer Center*), Entrez-PubMed-GenBank en NCBI (*National Center for Biotechnology Information*).

Formulación del problema

Las aplicaciones científicas se expanden constantemente por toda la comunidad internacional; cada vez son más grandes los conjuntos de datos que necesitan manipular los investigadores, por lo que se requiere de nuevas tecnologías y métodos para lograr una efectiva y eficiente manipulación de estos datos. Por otra parte, la capacidad de cómputo requerida para almacenar y procesar estos grandes conjuntos de datos sobrepasa la capacidad de las computadoras personales y pequeñas estaciones de trabajo. Muchos de los centros de ciencia en el mundo están basados en tecnologías paralelas, principalmente en tecnología GRID. Esta tecnología es costosa y necesita redes de alto ancho de banda para la comunicación entre los distintos servicios distribuidos que se precisan para un centro de ciencia.

La visualización científica es una poderosa herramienta no explotada aún suficientemente en los centros de ciencia, por no disponer de modelos que permitan su integración con esta tecnología. Por lo tanto, se hace necesario desarrollar los servicios que brindan los centros de ciencia utilizando una arquitectura paralela más económica (clúster de computadoras) y explotar sobre ella las ventajas de la visualización científica como herramienta para el análisis de los datos.

Preguntas de investigación

1. ¿Cuáles aspectos del diseño de un centro de ciencia para la VC pueden ser soportados sobre la tecnología de clúster de computadoras?

2. ¿Cómo se pueden implementar las funcionalidades de los centros de ciencia para la VC en un clúster de computadoras?

3. ¿Qué modelo de arquitectura es aplicable para los servicios y aplicaciones que puede brindar un centro de ciencia para la VC basado en un clúster de computadoras?

4. ¿Cuáles soluciones tecnológicas satisfacen los requerimientos de un centro de ciencia para la VC basado en un clúster de computadoras?

Objetivo General

Determinar la factibilidad de implementar un centro de ciencia para la visualización científica basado en un clúster de computadoras.

Objetivos Específicos

1. Determinar las tendencias actuales en el desarrollo de centros de ciencia de áreas específicas, teniendo en cuenta aspectos tales como tecnologías, arquitectura y servicios brindados.

2. Determinar cuáles aspectos del diseño de un centro de ciencia para la VC pueden ser soportados sobre la tecnología de clúster de computadoras.

3. Proponer una implementación de las funcionalidades de un centro de ciencia para la VC basado en un clúster de computadoras.

4. Diseñar un modelo de arquitectura basada en un clúster de computadoras para los servicios y aplicaciones brindados por un centro de ciencia para la VC.

5. Proponer un conjunto de soluciones tecnológicas que satisfaga los requerimientos de un centro de ciencia basado en clúster para la VC.

Justificación de la investigación

La gran mayoría de los centros de ciencia están basados en tecnología GRID. Esta tecnología es costosa y necesita redes de alto ancho de banda para la comunicación entre los distintos servicios distribuidos, que se necesitan en el centro de ciencia. Los clúster de computadoras constituyen una solución económica para pequeñas y medianas empresas, que necesitan manipular grandes volúmenes de datos con buenos rendimientos. Por otra parte, la utilización de la VC como herramienta para el análisis de datos en los centros de ciencia se presenta como una alternativa viable ante costosos métodos y herramientas computacionales empleados tradicionalmente, que por lo general consumen gran cantidad de tiempo y recursos.

Esta investigación se desarrolla dentro del Laboratorio de Computación Gráfica del Centro de Estudios de Informática de la Universidad Central de Las Villas, donde existe un clúster de computadoras dedicado a la investigación científica; se cuenta con grandes volúmenes de datos de áreas como la bioinformática y la meteorología, y con personal científico capacitado para llevar a cabo esta investigación.

Hipótesis de investigación:

1. Es posible determinar aspectos del diseño de un centro de ciencia para la VC, que pueden ser soportados sobre la tecnología de clúster de computadoras.

2. Algunas de las funcionalidades de un centro de ciencia para la VC pueden implementarse sobre la tecnología de clúster de computadoras.

3. A partir del estudio y sistematización de los modelos de arquitectura de los centros de ciencia existentes en la actualidad es posible definir un modelo de arquitectura aplicable a los servicios y aplicaciones de un centro de ciencia para la VC basado en un clúster de computadoras, que constituye una guía para el desarrollo de futuros centros de ciencia de este tipo.

4. La determinación de un conjunto de soluciones tecnológicas que satisfacen los requerimientos de un centro de ciencia para la VC basado en un clúster de computadoras facilita la implementación del centro de ciencia

Estructura

A continuación se muestra la estructura de este trabajo y se describe brevemente el contenido de cada uno de los capítulos.

Capítulo 2. "Los centros de ciencia como herramienta para la gestión de grandes volúmenes de datos"

En este capítulo se tratan los elementos teóricos de los CC, las principales tecnologías paralelas utilizadas para la gestión de grandes volúmenes de datos en los CC y se analizan algunos de los CC de mayor relevancia en la actualidad como (SDSS, BaBar, BIRN, GenBank), con el propósito de conocer: el funcionamiento, los servicios que brindan y elementos del diseño que nos faciliten la creación de un modelo de CC basado en clúster de computadoras.

Capítulo 3. "Tecnologías para el almacenamiento y visualización de grandes volúmenes de datos científicos"

En este capítulo se muestran las facilidades que ofrece la visualización científica, en particular la visualización distribuida. Se tratan aspectos esenciales que serán utilizados en el siguiente capítulo para lograr una integración entre los CC y la visualización de grandes volúmenes de datos. Los diferentes tipos de paralelismo y los tipos de renderizado paralelo son aspectos esenciales tratados en este capítulo. Por ultimo se presentan algunas de las tecnologías para el almacenamiento de datos científicos que se utilizan en el capítulo 4.

Capítulo 4. “Diseño del modelo general de un centro de ciencia sobre un clúster de computadoras”

Finalmente, en este capítulo se parte de un conjunto de requisitos que se deben tener en cuenta para la creación del CC, después se analiza qué aspectos de los centros estudiados son soportados sobre la tecnología clúster de computadoras. Luego se crea un diseño lógico basado en capas y finalmente se brinda un conjunto de soluciones tecnológicas para la implementación futura del centro de ciencia para la VC.

2 Los centros de ciencia como herramienta para la gestión de grandes volúmenes de datos

Los centros de ciencia, donde se almacenan y manipulan grandes volúmenes de datos del orden de los peta bytes, han posibilitado el desarrollo de nuevas tecnologías, técnicas y escenarios para gestionar estos datos. En este capítulo se estudian las características generales de los centros de ciencia y las tecnologías paralelas utilizadas para procesar los grandes conjuntos de datos que ellos almacenan. Se analizan además algunos de los CC de mayor relevancia en la actualidad, como SDSS, BaBar, BIRN, GenBank, con el propósito de conocer su funcionamiento, los servicios que brindan y algunos elementos de su diseño, que facilitan la creación de un modelo de CC basado en clúster de computadoras.

2.1 Centros de ciencia

La demanda de herramientas y recursos computacionales para realizar análisis de datos científicos crece continuamente, incluso más rápido que los propios volúmenes de datos. Lo cual es consecuencia de tres fenómenos:

1. Los nuevos y sofisticados algoritmos consumen cada vez más instrucciones para analizar cada byte de datos.

2. Muchos algoritmos de análisis son súper lineales, a menudo de orden $O(N^2)$ o $O(N^3)$.

3. El ancho de banda de entrada-salida no ha mantenido el mismo ritmo de avance que la capacidad de almacenamiento. En la última década mientras la capacidad de almacenamiento ha crecido más de cien veces, el ancho de banda solo lo ha hecho diez veces.

Estos tres fenómenos provocan que el análisis de los datos necesite mucho más tiempo. Para mejorar estos problemas los científicos necesitan mejores algoritmos de análisis, que puedan manejar inmensos conjuntos de datos con algoritmos aproximados (algunos con tiempo de ejecución cercano al lineal). Según (Gray et al., 2005) son necesarios algoritmos paralelos, que utilicen muchos procesadores y discos para equiparar las demandas de densidad de CPU con el ancho de banda.

Los centros de ciencia o centros de datos científicos, que están surgiendo como estaciones de servicio para algunos dominios científicos y suministran acceso tanto a los datos como a las aplicaciones que los analizan, se presentan como una vía para solucionar estas crecientes demandas de recursos computacionales.

Conjuntos de datos del orden de los peta bytes ocupan de mil a diez mil discos, y requieren de miles de unidades de procesamiento. Un centro de ciencia es capaz de almacenar uno o más de estos grandes conjuntos de datos y ofertar aplicaciones que acceden a ellos y los analizan. Posee además, un equipo de personas que entiende los datos y está constantemente mejorándolos y agregando más información.

El nuevo estilo de trabajo en estos dominios científicos es enviar solicitudes a las aplicaciones que se ejecutan en el centro de datos y recibir de vuelta las respuestas. Esto es mejor que hacer una copia de los datos hacia el servidor local para su posterior análisis con recursos propios.

Una tendencia actual que soportan muchos centros de ciencia es ofertar la posibilidad de almacenar espacios de trabajos personales y guardar allí las respuestas a las consultas realizadas. Esto minimiza el movimiento de los datos y permite la colaboración entre grupos de científicos que hacen análisis sobre temas similares. Estos espacios de trabajo son también una vía para la colaboración entre grupos de analizadores de datos. Según (Gray et al., 2005) a largo plazo los espacios de trabajos personales de un centro de datos pueden convertirse en una forma para la publicación de datos, mostrando tanto los resultados científicos de un experimento o investigación, como los programas usados para generarlos.

Los centros de ciencia proporcionan herramientas y soporte para permitir la colaboración con otros centros de ciencia. Cuando un científico desea consultar datos de varios centros de ciencia, no queda otra opción que mover partes de los datos de un lugar a otro. Si esta práctica se torna común, los centros de datos involucrados deben colaborar para suministrar la salva de los datos de manera mutua, ya que el tráfico entre ellos justifica disponer de las copias.

Muchos científicos prefieren hacer gran parte de su análisis en los centros de datos, ya que les ahorra tener que administrar datos locales y granjas de computadoras. Algunos científicos pueden llevar extractos de datos a sus casas para hacer procesamiento, análisis y visualización de manera local. Sin embargo, esto es posible hacerlo en el centro de datos utilizando el espacio de trabajo personal. Como se mencionó anteriormente, los conjuntos de datos del orden de los peta bytes requieren de mil a diez mil discos y miles de nodos. En cualquier momento algunos de los discos o nodos pueden colapsar. Estos sistemas poseen mecanismos para evitar la pérdida de datos y proporcionar buena disponibilidad, incluso con menos de la configuración completa, realizando esta tarea mediante un sistema de auto recuperación. Es por esto que se hace necesario disponer de un sistema de replicación para replicar los datos del centro de ciencia en distintos lugares geográficos (Pacitti et al., 2005).

Para el análisis científico es de gran importancia la utilización de 3 técnicas avanzadas:

1. Uso de grandes metadatos y metadatos estándares, para facilitar el descubrimiento de los datos que existen y sea cómodo tanto para las personas como para los programas entender los datos y conocer fácilmente todos los procesos por los que han pasado.

2. Buenas herramientas de análisis, que le permitan a los científicos, de forma fácil, hacer las consultas, entender y visualizar las respuestas.

3. Permitir el acceso paralelo a los datos soportados por los nuevos esquemas indexados y los nuevos algoritmos, que permitan interactuar y explorar conjuntos de datos del orden de los peta bytes.

Como consecuencia, los centros de ciencia se mantendrán como la principal vía para la comunicación de datos e información entre las asociaciones mundiales y suministraran tanto los datos como la infraestructura para manipular, analizar, procesar, visualizar y publicar estos archivos de gran escala, así como los algoritmos y herramientas para ello.

2.2 Aspectos generales de los centros de ciencia

Según (Moore, 2006), hay que tener en cuenta un conjunto de aspectos que son fundamentales para el funcionamiento de los centros de ciencia que almacenan colecciones científicas de datos. Entre estos se presentan:

- Disponer de tecnologías de punta para el almacenamiento de los datos. Estas tecnologías deben permitir aumentar la capacidad de almacenamiento en cualquier momento que se necesite. El tamaño de los datos, actualmente del orden de los peta bytes, y el número de objetos digitales se puede cuantificar en decenas de millones de archivos, pero estos datos aumentan de forma considerable cada año.

- Los datos científicos son típicamente distribuidos a través de múltiples sitios, ya sea durante el proceso de generación, o durante el proceso de análisis. Se necesita una infraestructura independiente de designación de espacios de nombres lógicos para identificar los archivos. Cuando un archivo se mueve entre los sitios, su nombre lógico nunca cambia.

- Los términos y conceptos utilizados por una disciplina científica no describen las características de los datos creados por otra disciplina académica. Cada disciplina diseña su propio metadato descriptivo. Los mecanismos de administración de latencia son necesarios para reducir al mínimo el número de mensajes enviados a través de la red y minimizar la sobrecarga asociada con la transmisión de datos.

- Hasta que los datos de una colección no estén calibrados y verificados, se limita el acceso a los miembros del equipo. El proceso de publicación de los datos requiere de una aprobación por un equipo científico para el cual estos son una representación exacta del mundo real. La publicación de los datos puede estar sujeta a revisión por otros científicos, como ocurre con los procesos utilizados para la publicación de artículos científicos. Cada comunidad científica normalmente aplica una codificación estándar diferente para optimizar la capacidad de manipular sus estructuras de datos.

En el epígrafe 2.4 se describe como los principales centros de ciencia del mundo manejan estas características para lograr un alto aprovechamiento de todos los recursos que ellos poseen y las facilidades que brindan a los usuarios.

2.3 Tecnologías paralelas utilizadas en la gestión de grandes volúmenes de datos

La capacidad de cómputo requerida por los centros de ciencia para analizar estos grandes conjuntos de datos, supone que se utilicen arquitecturas paralelas, específicamente GRID o clúster de computadoras. A continuación se tratan las principales características de ambas tecnologías, sus ventajas y desventajas.

2.3.1 Computación GRID

Se llama GRID según (Boghosian and Coveney, 2005) al sistema de computación distribuido que permite compartir recursos no centrados geográficamente para resolver problemas de gran escala. Los recursos compartidos pueden ser ordenadores (PCs, estaciones de trabajo, supercomputadoras, PDA, portátiles, móviles, etc.), software, datos e información, instrumentos especiales (radios, telescopios, satélites etc.), personas o colaboradores.

La computación GRID ofrece muchas ventajas frente a otras tecnologías alternativas. La potencia que brinda una multitud de computadoras conectadas en red usando tecnología GRID es prácticamente ilimitada, además de que permite una perfecta integración de sistemas y dispositivos heterogéneos, por lo que las conexiones entre diferentes máquinas no generarán ningún problema. Se trata de una solución altamente escalable, potente y flexible, que evita problemas de falta de recursos (cuellos de botella) y nunca queda obsoleta, debido a la posibilidad de modificar el número y características de sus componentes. En (Moore and Jagatheesan, 2004, Watson and Paton, 2003, Rajasekar et al., 2003) se profundizan estos aspectos.

Los datos se comparten entre miles de usuarios con intereses distintos y enlazan los principales centros de súper computación de todo el mundo. Es necesario asegurar que los datos sean accesibles en cualquier lugar y en cualquier momento. Los sistemas GRID

armonizan las distintas políticas de gestión de muchos centros diferentes y proporcionan la seguridad de los datos y aplicaciones que analizan los mismos.

Los principales inconvenientes tratados en (Boghosian and Coveney, 2005) provienen de la dificultad para sincronizar los procesos de todos estos equipos, monitorizar los recursos, asignar la carga de trabajo y establecer políticas de seguridad fiables, por lo que el costo se hace inalcanzable para pequeñas y medianas empresas.

2.3.2 Clúster de computadoras

Un clúster de computadoras es un grupo de equipos independientes conectados mediante una red local, que ejecutan una serie de aplicaciones de forma conjunta y aparecen ante los clientes y aplicaciones como un solo sistema, tienen memoria RAM, discos que pueden ser compartidos y procesadores similares estrechamente acoplados.

Los componentes del clúster están usualmente conectados entre ellos por redes locales de alta velocidad. Su funcionamiento según (Pacitti et al., 2005, Shankar and DeWitt, 2006) puede ser similar al de una supercomputadora, pero es mucho más asequible, porque los procesadores no tienen que ser tan potentes, la fuerza está dada por la paralelización y el trabajo compartido entre las diferentes unidades de cómputo.

Los clústers de altos rendimientos son implementados principalmente para aumentar la eficiencia de procesamiento, así como disminuir el tiempo de ejecución, repartiendo una tarea computacional a través de los diferentes nodos que los componen. Esta tecnología tiene gran aplicación en el área de la visualización científica y es una alternativa factible para las pequeñas y medianas empresas.

La comunicación mediante el paso de mensajes como paradigma de programación tratada en (Alonso, 1997), requiere que el paralelismo se exprese de forma explicita por el programador, responsable de analizar su algoritmo o aplicaciones para identificar vías por las cuales puede repartir la carga de trabajo entre los múltiples procesadores del clúster de la forma más eficiente posible, además de poder lograr concurrencia. Como resultado, la programación mediante el paradigma de paso de mensajes, suele ser una actividad compleja, consumidora de tiempo y de alta demanda intelectual. De otro lado, las

aplicaciones diseñadas por paso de mensajes tienden a conseguir buenas prestaciones que se mantienen cuando el sistema aumenta a números grandes de tareas y/o procesadores. Algunas de las interfaces de los sistemas basados en paso de mensajes más utilizadas son: PVM (por las siglas en inglés de *Parallel Virtual Machine*) y MPI (por las siglas en inglés de *Message Passing Interface*) (Grama et al., 2003), también existen otras como BSP (por las siglas en inglés de *Bulk-Synchronous Model*, P4 (por las siglas en inglés de *Portable Programs for Parallel Processors*), PARMACS (Dongarra, 1996), y EXPRESS (Reed et al., 1990), que están en pleno desarrollo.

La tecnología GRID es la más utilizada por los grandes CC tratados en el epígrafe 2.4. No obstante, para desarrollar los modelos propuestos en esta investigación se seleccionó la tecnología clúster de computadoras. Las razones que justifican esta selección son las siguientes:

- La tecnología clúster de computadoras es más económica que la tecnología GRID, lo que la convierte en una alternativa viable para pequeñas y medianas empresas y países en vías de desarrollo.

- El Centro de Estudios de Informática de la UCLV, lugar donde se desarrolla esta investigación, cuenta con dos clústers de computadoras (uno para investigación y desarrollo y otro para producción), que están disponibles para la realización de este trabajo.

2.4 Tendencias actuales de los Centros de Ciencia

Los centros de ciencia, o centros de datos científicos como también se les conoce, son cada vez más grandes en cuanto a la capacidad de procesamiento y almacenamiento. Según (Gray et al., 2005), un centro de ciencia es capaz de almacenar uno o más conjuntos de datos y ofertar aplicaciones que acceden a ellos y los analizan. Posee además, un equipo de expertos que están constantemente mejorando los datos y agregando más información.

En este epígrafe se analizan las principales características de cuatro de los centros de ciencia más importantes del mundo, escogidos por su funcionamiento y por los servicios que brindan. Estos son de interés para la creación de un modelo de centro de ciencia, donde

se pretende integrar la visualización científica como nueva alternativa para el tratamiento de grandes volúmenes de datos.

En la siguiente sección se introduce el término "solucionador de recursos de almacenamiento" mediante el análisis de dos herramientas, una llamada *Storage Resource Broker* (SRB) y otra *integrated Rule Oriented Data Systems* (iRODS). La primera es utilizada por casi todos los centros de datos estudiados en este capítulo para manejar sus recursos de almacenamiento y procesamiento. Este término se discute con el objetivo de utilizar una de estas herramientas o definir una similar, que facilite integrar los recursos disponibles con los servicios que se pretenden brindar a los usuarios.

Es válido señalar que la documentación sobre el diseño e implementación de estos centros de ciencia es muy pobre, los principales artículos no están públicos en Internet. Las fuentes más importantes utilizadas para esta investigación fueron la interacción directa con estos centros de ciencia, que sí son de libre acceso, y la ayuda de científicos interesados en esta investigación, que facilitaron bibliografía e información. La documentación que se aporta aquí sobre este tema es un valor añadido importante de esta investigación, pues contribuye al desarrollo de pequeñas instituciones que no disponen de suficientes recursos para brindar servicios similares con un costo asequible.

2.4.1 Solucionadores de Recursos de Almacenamiento

En SDSC se ha desarrollado un sistema de gestión de datos que satisface las necesidades de la mayoría de las comunidades científicas. La tecnología se denomina "solucionador de recursos de almacenamiento" o SRB, por las siglas en inglés de *Storage Resource Broker*. El sistema actualmente en funcionamiento maneja más de 50 tera bytes de datos, incluyendo alrededor de 9 millones de archivos almacenados en repositorios de este centro y otros sitios. Existen otros sistemas solucionadores de recursos de almacenamiento para el manejo de datos del orden de los tera bytes, pero sin duda el más estandarizado hasta el momento es éste. Según (Cao et al., 2005, Rajasekar et al., 2003), el sistema SRB es utilizado por proyectos patrocinados por: la Fundación Nacional de Ciencias, la Administración Nacional de Aeronáutica y la Agencia del Espacio, el Departamento de Energía, el Consejo Nacional de Archivos y Registros de Administración, el Instituto

Nacional de Salud, el Instituto Nacional de Publicaciones Históricas y la Comisión de Registros.

SRB ha estado en producción y uso durante los últimos cuatro años y es utilizado por centros de distintas áreas de investigación, como Astronomía, Sistemas de la Tierra y Ciencias del Medio Ambiente, Ciencias Médicas, Ciencias Moleculares, Neurociencias, Física y Química, Archivos y Bibliotecas Digitales (Rajasekar et al., 2003).

SRB es un *middleware* basado en tecnología cliente-servidor, que proporciona un servicio de construcción de colecciones de archivos y su administración; brinda servicios de consulta y acceso, así como la preservación de los datos en un marco de red de datos distribuidos. Esta tecnología no es totalmente libre, solo está disponible para instituciones educacionales y dependencias gubernamentales y se rige por las leyes de exportación de los EEUU. Existen un conjunto de países que no pueden adquirirla, dentro de ellos Cuba, que no tiene acceso a este software.

Un solucionador de recursos de almacenamiento ofrece las siguientes posibilidades:

- Identificadores globales persistentes para la asignación de nombres de archivos.
- Soporte de metadatos para describir la ubicación y la propiedad de los archivos.
- Soporte de metadatos descriptivos para apoyar el descubrimiento a través de los mecanismos de consulta de bibliotecas digitales.
- Norma mecanismos de acceso a través de navegadores Web, los comandos de *shell* de UNIX, navegadores de Windows, *scripts* Python, Java, las llamadas biblioteca de C, la redirección de entrada-salida de Linux y WSDL entre otras.
- Depósito abstracto de almacenamiento para interactuar con varios tipos de sistemas de almacenamiento.
- Sistema de autenticación para el acceso seguro a datos remotos.
- El apoyo a la replicación de archivos entre sitios.

- Soporte para copias de los archivos de caché en un sistema de almacenamiento local y el apoyo para acceder a varios archivos como si estuvieran en el mismo lugar.
- Apoyo para la agregación de los archivos en los contenedores.
- Apoyo para la inscripción de nuevos ficheros en el sistema.

iRODS es otro solucionador de recursos de almacenamiento orientado a reglas para el almacenamiento de datos. Es un proyecto desarrollado con la finalidad de construir la nueva generación de ciberestructuras para la manipulación de datos científicos. Uno de los principales objetivos del iRODS es suministrar una arquitectura para manipular datos que sea flexible, adaptativa y configurable. iRODS también fue desarrollado en *San Diego Supercomputer Center* donde ya se están realizando las aplicaciones para migrar de SRB a este nuevo solucionador de recursos de almacenamiento.

Esta nueva tecnología es de código abierto bajo licencia tipo BSD (por las siglas en inglés de *Berkeley Software Distribution*). Es por eso que en (Cao, 2008) se recomendó directamente por Peter Cao[1] utilizar este software para esta investigación, el cual tiene particularidades similares al SRB (Moore et al., 2008) y puede descargarse gratuitamente de Internet. Esta es la principal razón por la que en los modelos propuestos en este trabajo se recomienda el uso de esta herramienta como solucionador de recursos de almacenamiento.

En los epígrafes siguientes se muestran las principales características de los centros de ciencia SDSS, BaBar, BIRN y GenBank, seleccionados como base de estudio para el diseño propuesto en este trabajo. Los mismos fueron escogidos por su prestigio, aporte al desarrollo de la ciencia y principalmente por sus características de diseño y servicios que brindan a la comunidad, siendo todos estos factores que contribuyen a la creación del modelo de CC.

[1] Peter Cao es un destacado investigador del *National Center for Supercomputing Applications (NCSA).*Actualmente desarrolla aplicaciones para el HDFGROUP.

2.4.2 Sloan Digital Sky Survey (SDSS)

SDSS SkyServer, es un proyecto para confeccionar un mapa de gran parte del universo. Ha revolucionado la astronomía por las facilidades que brinda a la comunidad de científicos que investigan en esta rama de la ciencia tan importante. Con la disponibilidad de acceso a los datos, los astrónomos realizan consultas complicadas y obtienen respuestas en pocos segundos, o quizás minutos en caso de que la consulta requiera buscar en toda la base de datos (Szalay et al., 2002).

SDSS brinda conjuntos de datos del orden de los tera bytes (Thakar et al., 2004), recolectados a partir de 1998 hasta la actualidad. Estos datos son cada día más dóciles, debido al crecimiento exponencial de la velocidad de procesamiento y el incremento de la capacidad de almacenamiento. El uso de la visualización distribuida, la réplica de los datos y las herramientas de programación paralela han logrado una equivalencia entre el nivel de almacenamiento y la capacidad de cómputo necesaria para poder analizar estos grandes volúmenes de datos (SDSS, 2008).

Los datos astronómicos obtenidos por los diferentes medios pasan por distintas etapas antes de ser publicados a la comunidad de científicos. Esta información primeramente es almacenada en bases de datos temporales en el Laboratorio Nacional de Fermi, que solo están disponibles para los especialistas encargados de demostrar que estos datos son una representación real de alguna parte del universo. Después del procesamiento y reexaminación de la calidad son exportados para el centro de ciencia, donde son replegados en las distintas instituciones pertenecientes a SDSS, con el archivo maestro residiendo en Fermilab.

Este centro de ciencia almacena los datos en bases de datos de SQL, organizados de forma jerárquica en multi-capas. Este repositorio es un conjunto de datos distribuido y está compuesto por miles de bases de datos. Cada base de datos es un archivo en disco y la colocación de los objetos en contenedores es configurable por el administrador de la base de datos.

El repositorio de datos puede ser movido a través de plataformas heterogéneas sin pérdida de compatibilidad. Esto facilita la implementación y el mantenimiento de archivos

distribuidos. Las herramientas administrativas brindan servicios de seguridad y gestión de los datos. La replicación de los datos permite hacerle copia, además de propagar las actualizaciones automáticamente en los sitios espejos; estas principales características posibilitan correr consultas complicadas para todo tipo de búsqueda en cualquier punto geográfico.

SDSS posee una herramienta online llamada CasJobs, que permite el acceso a la gran base de datos de este centro de ciencia a través de los catálogos científicos. Esta herramienta está diseñada para emular y mejorar el acceso libre a las consultas en un entorno Web, además permite a los usuarios abrir múltiples sesiones con servidores diferentes simultáneamente y enviar consultas a servidores en paralelo, las consultas propuestas pueden dirigir su salida de regreso a la interfaz gráfica del usuario, también pueden enviar la salida directamente a un archivo o a otra herramienta de análisis. La capacidad binaria de salida permite el flujo de los datos en forma compacta para agilizar el tráfico en la red. (CasJobs, 2008)

A continuación se presentan algunas funciones de esta aplicación:

- Permite ejecutar consultas de forma sincrónica y asincrónica, en forma de tareas rápidas o demoradas.
- Guarda un historial que registra las consultas y su estado.
- Posee una base de datos personal llamada “MyDB” en el lado del servidor que permite la creación de tablas, funciones y procedimientos. En esta base de datos personal se almacenan partes de los grandes volúmenes de datos del centro de ciencia que el usuario está investigando.
- Permite compartir datos entre los usuarios a través de un mecanismo de “grupos de usuarios”.
- Permite descargar en diferentes formatos los datos que están en la base de datos personal MyDB.

- Presenta varias opciones de interfaz, incluyendo un navegador cliente, así como una herramienta de línea de comandos.

Las consultas presentadas al sistema son tramitadas por un agente de búsqueda SDSS, que es transparente al usuario, siendo un servidor inteligente de búsqueda que:

- Maneja las sesiones del usuario.
- Analiza gramaticalmente, optimiza y ejecuta las búsquedas del usuario.
- Extrae atributos de objetos individuales conforme los solicite el usuario.
- Las salidas son puestas en las rutas especificadas por el usuario.

El primer paso del Agente de Búsqueda del centro de ciencia es analizar cada consulta propuesta para proveer una "estimación del costo de la tarea" en curso. Este costo es estimado a partir de los subconjuntos de la base de datos que debe ser registrada y mostrado al usuario como el tiempo mínimo obligado para completar la búsqueda, por lo que el usuario basado en el alcance y el costo de la consulta puede decidir abortarla o no.

El lenguaje de búsqueda SDSS es SXQL, que es un SQL simple que implementa el subconjunto básico de cláusulas y funciones necesarias para formular consultas para una base de datos objeto relacional. Reconoce las cláusulas SELECT-FROM WHERE del SQL estándar, además incluye declaraciones SELECT anidadas. También permite especificación de enlaces de asociación en el SELECT, FROM y subcláusulas WHERE. Las asociaciones son enlaces para otros objetos y pueden ser uno o varios enlaces. Además reconoce una sintaxis de búsqueda de proximidad, lo cual deja al usuario ir en busca de todos los objetos que están próximos a un objeto dado en el espacio. Además contiene un número de macros específicos de la astronomía y tiene soporte para funciones matemáticas.

Otra característica importante que no se debe dejar de mencionar es la capacidad de salida en formato XML, simplemente especificando FOR XML en la cláusula WHERE de la consulta, el usuario puede optar por recibir el resultado de la búsqueda en XML en vez de texto ASCII. Esta es una característica especialmente importante para el Observatorio

Virtual, en el cual muchas aplicaciones de análisis de datos que los astrónomos necesitan están disponibles como servicios Web.

Este centro de ciencia posee además otro conjunto de herramientas dentro de las que se encuentran, la herramienta *Famous Places*, que presenta una galería de imágenes de varias galaxias y algunas de las estrellas más distantes que se han descubierto. La herramienta *Get Images* que permite descargar imágenes individuales con varios niveles de resolución. Otras herramientas no menos importantes son *Scrolling Sky*, *Search*, *Object Crossid*, *Help* y *Download*. Además, SDSS presenta varias herramientas de visualización (*Visual Tools*) como son *Finding Chart* que devuelve una imagen jpg con varios parámetros aplicados a los datos, *Navigate* que permite la navegación interactiva por lo datos del universo y otras como *Image List*, *Quick Look*, y *Explore*. Esta última permite explorar interactivamente varias propiedades de un objeto individual.

Sloan Digital Sky Survey identificará la ubicación de más de 100 millones de galaxias. Además, mediante la combinación de los datos que se encuentran en los catálogos o el reanálisis de cada uno de los píxeles de cada imagen, se pueden obtener nuevos objetos o generar estadísticas en forma de galaxias. Este último enfoque exige el flujo de imágenes desde su ubicación de almacenamiento a través de una plataforma de procesamiento. Para facilitar el análisis las colecciones han sido replicadas en *NSF Teragrid* (Moore, 2006) que proporciona tanto los recursos de cómputo como de almacenamiento.

Los metadatos utilizados para describir los datos se basan en descriptores uniformes de contenido para los catálogos de entradas. La astronomía ha creado una comunidad estándar de formato de datos para las imágenes, llamado FITS. Este formato proporciona una manera de encapsular metadatos descriptivos acerca de la ubicación, la resolución y el alcance de cada imagen, junto con los píxeles que la componen.

2.4.3 BaBar

El Centro de ciencia BaBar en *Stanford Linear Accelerator Center* (Sullivan, 2007), se encarga de almacenar grandes volúmenes de datos del orden de los peta bytes obtenidos de experimentos de física, así como brindar servicios a la comunidad de científicos para analizar estos datos. Al ocuparse del acceso concurrente de múltiples clientes para

repositorios de datos de escala peta byte: el alto funcionamiento, la tolerancia a fallos, la robustez y la dimensionalidad son cuatro aspectos muy importantes que están presentes en la arquitectura del centro (BaBar, 2007).

La comunidad de física de altas energías, tradicionalmente, ha venido acumulando experiencias en el almacenamiento y análisis de grandes volúmenes de datos, teniendo como uno de sus principales exponentes el centro de ciencia BaBar. Sólo la comunidad del experimento BaBar aglutina a más de 600 físicos e ingenieros de 75 instituciones a escala mundial y la tendencia es que sigan aumentado cada año, además han logrado consolidar una de las bases de datos orientadas a objetos más grandes del mundo, ya que tenía en noviembre del 2004 casi 900 tera bytes. BaBar está orientado a la exploración de los principios físicos de la energía, la materia y la antimateria, utiliza recursos computacionales en modelo de GRID para aumentar la velocidad y la calidad en las investigaciones mundiales en este tema.

LHC (por las siglas en inglés de *Large Hadron Collider*) *Computing Grid Project* (LCG) ha adoptado las herramientas computacionales y ha creado grupos de evaluación y prueba conjunta con el Proyecto Europeo DataGrid. La organización para la operación de LCG es jerárquica y está compuesta por diversas Capas, donde la raíz de esta jerarquía está en la Organización Europea para la Investigación Nuclear más conocida como CERN (por sus siglas en Francés *Conseil Européen pour la Recherche Nucléaire*) (Capa 0). Luego siguen los nodos regionales (Capa 1) en IN2P3 Lyon-Fancia, PIC Barcelona-España, CNAF en Boloña-Italia, PPARC en Rutherford-Inglaterra, por mencionar los principales nodos europeos (Gordon and Boyd, 2001, Nief et al., 2005). Luego siguen nodos menores de servicios. Cada uno de estos nodos realiza labores de limpieza de datos y provee capacidades de cómputo para el análisis de esos resultados.

El experimento BaBar en *Stanford Linear Accelerator Center* produce una cantidad enorme de datos para ser accedidos por un número alto de trabajos de análisis. Por esta razón requiere un sistema confiable y escalable de acceso a datos. En el año 2002 el comité decidió migrar el sistema de almacenamiento de datos de base de datos orientada a objetos (BDOO) para un sistema de archivo plano basado en flujo de objetos (Dorigo et al., 2004). El nuevo sistema de almacenamiento se basa en un mecanismo de persistencia, desarrollado

en CERN (ROOT, 2004), que es capaz de fluir un objeto en un archivo binario similarmente como el *framework* de Java.

BaBar usa SRB como herramienta para suministrar una interfaz uniforme de acceso a sistemas de almacenamientos heterogéneos como discos, cintas, bases de datos, que están distribuidos en varios sitios (Becla and Wang, 2005), y como herramienta de soporte para compartir archivos de forma colaborativa; permite control de réplicas, por lo que es fácil duplicar una BD de un sitio a otro, y proporciona búsquedas por atributos de los datos y búsquedas por metadatos.

La implementación de técnicas de gran capacidad de procesamiento, así como sistemas de acceso a datos capaces de maniobrar millones de archivos distribuidos y tolerantes a fallos, son objetivos fundamentales en las que se basa la arquitectura de este centro, algunas características que la distingue según (Dorigo et al., 2004) son:

- Los múltiples servidores cooperan entre sí con el fin de maniobrar cantidades enormes de datos distribuidos y redundantes si es necesario, sin forzar al cliente a saber cuál servidor contactar para acceder a un conjunto particular.

- El servidor encubre las aplicaciones del cliente que están debajo del tipo de sistema de archivo, aun si maneja una o más unidades.

- Presenta un mecanismo de balanceo de carga para distribuir eficazmente la carga entre grupos de servidores.

- Presenta un alto grado de tolerancia a fallos, para minimizar el número de trabajos o aplicaciones que tienen que ser nuevamente echadas a andar después de algún problema que detecte el servidor o cualquier clase de fallo de la red.

La arquitectura diseñada por los especialistas para cumplir con los requisitos antes mencionados, permite la construcción de sencillos sitios de acceso a datos del servidor, cargar ambientes balanceados y estructuras con implementaciones iguales. Esta estructura, en cualquier caso, es una interfaz definida como un protocolo de comunicación, que define las posibles interacciones y funcionalidades dadas a los clientes.

El protocolo de comunicación define una interfaz capaz de: Pedir acceso al sistema a través de los medios de autentificación, interrogar un sistema en busca de un recurso, obtener acceso al recurso pedido en el lugar donde puede ser accedido (o sea los servidores dándole acceso a los datos locales o permitiendo interoperabilidad remota de sitios), políticas sofisticadas de comunicación del lado del cliente y capaz de manejar cualquier clase de errores de comunicación. Las peticiones que fallen son intentadas de nuevo hasta que se encuentre otro servidor de trabajo, el mismo servidor se hace disponible otra vez o hasta que se alcance un número máximo especificado de nuevos intentos.

Este sistema, por el lado del servidor, está compuesto por cuatro capas: Red y capa de administración de hilos, capa de protocolos, capa de sistema de archivos y capa de almacenamiento. Mientras que por el lado del cliente, por tres capas: La capa de interfaz, la capa de comunicación de alto nivel y la capa de comunicación de bajo nivel.

2.4.4 Biomedical Informatics Research Network (BIRN)

BIRN es otro centro de ciencia que está compuesto por gran cantidad de sitios (Moore, 2006), principalmente de Estados Unidos y Gran Bretaña, residiendo en la universidad de California, San Diego. Todos los sitios están unidos con el objetivo de diseñar e implementar una arquitectura distribuida de recursos compartidos que sean usados por los investigadores de la biomedicina para avanzar en el diagnóstico y tratamiento de enfermedades (BIRN, 2008).

El proyecto BIRN es una iniciativa encaminada a la creación de un laboratorio para los investigadores biomédicos mediante el acceso y análisis de datos ubicados en diversos sitios de varios países. Cuestiones relacionadas con el manejo de los datos en la red, autentificación de usuarios, integridad, seguridad y la propiedad de los datos son aspectos que se requieren como parte del proyecto BIRN. Son necesarias dos redes para el intercambio de datos: una para bibliotecas digitales para publicar los datos y se necesita otra para la persistencia de los archivos con el objetivo de conservar los datos. En SDSC se creó un catálogo central para la gestión de datos que lógicamente fue organizado en una colección. Los datos originales residen en los sitios participantes, mientras que los archivos están replicados en el SDSC.

Una parte importante del proyecto BIRN es administrado centralmente para desplegar el hardware personalizado que es extensible tanto en la capacidad como la ubicación. Un sistema basado en Linux, llamado BIRN RACK (Rajasekar et al., 2002), se desplegó en cada sitio para crear memorias caché de datos distribuidos. Esta memoria es un componente esencial de la red, necesario para la gestión de acceso a recursos de datos estrictamente controlados en una amplia red. BIRN RACK puede personalizarse para ejecutar *Storage Resource Broker* (SRB), al igual que BaBar. El centro coordinador de SRB, reside en la Universidad de California San Diego (UCSD) y administra la red de datos. Los datos se replican entre sitios bajo el control del SRB y este también apoya el acceso a los datos de visualización de los programas que están actualmente en uso por los diferentes asociados a BIRN.

En cierto sentido, los mecanismos de acceso uniforme de SRB hacen posible que se apliquen diferentes técnicas a los mismos datos para ver las ventajas y desventajas que tienen unas de otras (Bruch et al., 2002). El objetivo principal del proyecto BIRN es superar los desafíos y problemas en el acceso a grandes conjuntos de datos en todos los sitios al mismo tiempo y atender el estricto cumplimiento que necesita el intercambio de datos médicos.

Un requisito particular del proyecto BIRN es la capacidad de aplicar controles de acceso tanto a las imágenes como los metadatos descriptivos y administrativos registrados en un espacio de nombres lógicos. Todos los controles de acceso se aplican directamente al nombre lógico. Esto significa que cuando una imagen se mueve, automáticamente los controles de acceso se aplican en la nueva ubicación. En el ambiente SRB, todos los archivos se almacenan bajo el control de sistema de manejo de datos de este, asignando un identificador de UNIX. El sistema SRB utiliza los controles de acceso para decidir si se permite la entrada de los usuarios al sistema mediante un certificado de llave pública.

Los controles de acceso a los metadatos son más complejos. Las restricciones de acceso posibilitan que los usuarios solo puedan ver los metadatos descriptivos de los archivos a los que se le permite el acceso. No podrán ser vistos mediante consultas, los archivos sobre los cuales no se tenga permiso de acceso. También son necesarias restricciones de acceso a los metadatos administrativos, de manera que los usuarios no puedan ver ningún valor de los

atributos seleccionados. Así, los administradores pueden utilizar los valores de los atributos para administrar el repositorio sin que sea visto por los usuarios.

Uno de los objetivos del proyecto BIRN fue la necesidad de desarrollar sistemas de bajo costo para almacenar las grandes colecciones de datos. Los investigadores prefieren mantener los datos en discos giratorios para reducir al mínimo las latencias de acceso. Con el desarrollo de los discos y los procesadores, ahora es posible tener un costo en cuanto a discos de almacenamiento de alrededor de $ 3000 por tera bytes, incluyendo el acceso a la red, estas redes GRID son sistemas modulares que combinan una CPU de 1,7 Ghz, con un giga bytes de memoria, un tera bytes de disco, una controladora RAID y una conexión de red que funciona bajo el sistema operativo Linux. Para ampliar la capacidad de procesamiento se añaden nuevos nodos al sistema y se agregan más discos para aumentar la capacidad de almacenamiento.

La colaboración entre los diversos sitios que componen BIRN (Grethe, 2006), suministra al resto de la comunidad científica los siguientes aspectos:

- Intercambio de datos: La colaboración entre los múltiples sitios permiten desarrollar infraestructuras para compartir los datos y definir políticas sobre los conjuntos de datos disponibles para la comunidad científica.

- Interoperabilidad: La colaboración entre los diversos sitios motiva la interoperabilidad de las herramientas de análisis desarrolladas en los diferentes laboratorios.

- Estándares de comparación: Esta colaboración puede contribuye al desarrollo de estándares de los nuevos formatos de datos y protocolos de análisis. Estas ayudas mueven el campo hacia delante estableciendo marcas mundiales en el área de trabajo.

- Sinergia: La colaboración fomenta la sinergia que significa producir más con la unión de varios sitios de lo que se puede producir en cada uno por separado. La combinación de fuerzas de cada sitio suministra recursos más robustos que los que están disponibles en cada uno de ellos.

BIRN posee una gran cantidad de herramientas para:

- Adquisición, calibración y control de calidad.
- Almacenamiento y administración de datos.
- Análisis y procesamiento.
- Visualización y construcción de mapas cerebrales, etc.
- Construcción de estándares, terminologías y ontologías.
- Intercambio de datos y colaboración.

Las comunidades de investigación biomédicas trabajan con grandes conjuntos de datos y a menudo altamente heterogéneos. BIRN provee una manera para integrar estos datos, permitiendo analizarlos y distribuir la información para realizar nuevos descubrimientos. BVDG (por las siglas en inglés de *Virtual Data Grid*) almacena los datos a través de un ambiente distribuido, donde estos son continuamente administrados y manipulados como un simple sistema virtual de archivos. Actualmente, dentro BVDG existen más de tres millones de ficheros de datos, activos, que son accesibles directamente a través de un conjunto de interfaces estandarizadas.

A través de BIRN se puede compartir o publicar tanto datos como resultados de las investigaciones usando el repositorio de datos BIRN (BDR, por las siglas en inglés de *BIRN Data Repository*), el cual se rige por un conjunto de normas que tienen requisitos de publicación, garantía, seguridad y redistribución.

Los científicos pueden acceder desde cualquier ordenador que tenga acceso a Internet a través de un simple portal Web que provee un conjunto de herramientas integradas, infraestructura y servicios necesarios para realizar estudios de gran magnitud y en colaboración.

Además, este centro tiene la capacidad de: autentificar todas las personas que acceden a las imágenes, la gestión de los controles de acceso a todos los archivos de datos

independientemente del lugar donde estos están almacenados, la gestión de los controles de acceso a los metadatos para proporcionar pistas de auditoría, permitir todos los accesos a los datos independientemente de la ubicación de almacenamiento y apoyar el cifrado de los mismos. En la práctica, la auditoría se utiliza para demostrar la cantidad de intercambio de datos entre los sitios de colaboración. La cantidad de datos remotos visitados por un investigador es cuantificada, así como la cantidad de datos que el investigador distribuye a otros sitios.

2.4.5 EntrezPubMedGenBank

El Centro Nacional para la Información Biotecnológica (NCBI, por las siglas en inglés de *National Center for Biotechnology Information*) es parte de la Biblioteca Nacional de Medicina de Estados Unidos y una rama de los Institutos Nacionales de Salud (NIH, por las siglas en inglés de *National Institutes of Health*) (NCBI, 2008). Está localizado en Bethesda, Maryland y fue fundado el 4 de noviembre de 1988 con la misión de ser una fuente importante de información de biología molecular. Almacena y actualiza constantemente varias bases de datos biológicas de acceso público. Entre las más conocidas y populares se encuentran las bases de datos de publicaciones científicas (PubMed), de secuencias de proteínas y ADN (GenBank), de estructuras tridimensionales de proteínas y algunas otras no tan populares como OMIM (*Online Mendelian Inheritance in Man*) (GenBank, 2008).

NCBI desarrolló Entrez como una herramienta para permitir a los usuarios interaccionar con estas bases de datos. Desde el punto de vista informático, Entrez es una “interfaz de usuario”, es decir, constituye el nexo entre el usuario y las bases de datos subyacentes. Como interfaz, Entrez permite al usuario realizar consultas simples y obtener resultados, aun desconociendo la arquitectura de las bases de datos (Cannataro et al., 2004). Todas las bases de datos del NCBI están disponibles en línea de manera gratuita y pueden ser accedidas usando el buscador Entrez, este centro es conocido como EntrezPubMedGenBank en lo siguiente se le llamará GenBank.

El sistema de búsqueda PubMed es otro proyecto desarrollado por NCBI en NLM (por las siglas en inglés de *National Library of Medicine*). Permite el acceso a las base de datos

bibliográficas de NLM (NCBI, 2002) MEDLINE, PreMEDLINE (citas enviadas por los editores antes de su publicación) y AIDS. Se pueden establecer alertas de correo electrónico automático para los nuevos artículos agregados a PubMed. Además, tiene la opción de guardar las búsquedas y una vez guardada, puede escoger repetir la búsqueda en el futuro o bien que el sistema la ejecute automáticamente y le envíe los resultados por correo electrónico.

Teniendo en cuenta el objetivo de escalabilidad, este centro fue diseñado para usar una arquitectura distribuida. El sistema consta de una base de datos central de autentificación o de entrada y múltiples bases de datos distribuidas por categorías. Principalmente ocho categorías disponibles. La base de datos central de autentificación provee la URL de cada una de estas. Las ocho instancias de la base de datos comparten un esquema idéntico de base de datos, dejando una implementación genérica para el servicio Web. Las bases de datos de categorías adicionales pueden fácilmente ser incorporadas en el sistema existente, con una adición mínima en la BD central de autentificación, no requiriendo cambio en la capa de servicio Web.

La base de datos del GenBank crece de manera exponencial, este crecimiento es debido a la misma forma en que la base se actualiza. Son los mismos autores quienes se encargan de mantener la base al día, pero además de remisiones de autores, el GenBank se nutre también de las otras bases de datos existentes actualizando interactivamente sus ficheros.

Existen muchas interfaces de acceso a GenBank, pero sin duda alguna la más efectiva es Entrez. Esta es una interfaz que brinda acceso a muchas de las bases de datos mantenidas por NCBI. Desde la página http://www.ncbi.nlm.nih.gov/Entrez , se puede ir a:

1. División de publicaciones médicas (PubMed): interfaz al servicio de citas bibliográficas del MEDLINE.

2. Secuencias de nucleótidos, colección de archivos de GenBank.

3. Base de datos proteica: esta base de datos combina la información de muchas fuentes con secuencias derivadas de la traducción de secuencias GenBank.

4. Base de datos de estructuras tridimensionales: información estructural de proteínas derivada de cristalografía de rayos X y resonancia magnética nuclear.

5. Bancos de Genomas: Compilación de mapas genéticos y físicos de una gran variedad de especies.

6. Taxonomía: Se usa la misma clasificación filogenético que en el GenBank, es principalmente un recurso de navegación.

Haciendo una búsqueda a través del Entrez es posible llegar a múltiples fuentes de información acerca del mismo tema, por ejemplo es factible para una secuencia encontrar su listado de citaciones bibliográficas contenidas en el MEDLINE.

Un servicio de NCBI (http://www.ncbi.nlm.nih.gov) que vale la pena resaltar: el mapa genético del genoma humano, este es un compendio de aproximadamente 30261 secuencias.

A modo de resumen, EntrezPubMedGenBank es un centro de ciencia que se dedica fundamentalmente a la bioinformática, medicina y biotecnología, también esta compuesto por un grupo de sitios diferentes, pero con objetivos similares. Presenta un conjunto de servicios GRID que pueden ser accedidos a través de un cliente *Desktop* conocido como Taverna (Taverna, 2008). Este cliente cuando se ejecuta, automáticamente reconoce todos los servicios que hay en los diferentes sitios del centro de ciencia y permite el uso de los mismos mediante diagramas de flujo de trabajo (Wolstencroft, 2008). Los mensajes en XML son usados para la comunicación entre la capa del cliente y la capa de servicios Web. La comunicación entre el cliente y servidor se realiza mediante un protocolo simple de acceso a datos sobre HTTPS. La capa de servicio Web es la interfaz entre el cliente y las funcionalidades que brinda el servidor. Esta arquitectura permite realizar cambios en las funcionalidades internas y localizaciones de los recursos sin que afecte la interfaz del cliente. GenBank provee servicios de búsqueda y de análisis en conjuntos no comprimidos de estos datos, requiriendo una cantidad de 20 giga bytes de espacio en disco para datos, índices y las actualizaciones, así como también poderosos procesadores para poder analizarlos.

2.5 Conclusiones parciales

En este capítulo se estudiaron las principales características de los CC, las tecnologías paralelas para la gestión de grandes conjuntos de datos y aspectos específicos de los CC más importantes del mundo. A partir de este análisis se puede concluir que:

- Existe un conjunto de características generales de los CC, como uso de espacios personales de trabajo, uso de diferentes tipos de metadatos y de una herramienta solucionadora de recursos de almacenamiento. Estas características deben tenerse en cuenta en el diseño de un centro de ciencia para cualquier área específica.

- La seguridad de los datos y el tratamiento a los conjuntos que se almacenen en el centro de ciencia son otras características que requieren especial tratamiento. En el diseño que se propone en el Capítulo 4 de este trabajo para un centro de ciencia basado en un clúster de computadoras no se considera como requisito imprescindible la colaboración con otros centros de ciencia.

- Se determinó que la tecnología de clúster de computadoras es las más adecuada para el desarrollo de centros de ciencia, donde los recursos tecnológicos y financieros son limitados.

3 Tecnologías para el almacenamiento y visualización de grandes volúmenes de datos científicos

En este capítulo se muestran las facilidades que ofrece la visualización científica y se profundiza en los tipos de visualización distribuida. Se presentan las particularidades de los diferentes tipos de paralelismo que existen, así como las diferentes formas de renderizado paralelo, siendo todos estos, métodos que pueden ser utilizados para visualizar grandes volúmenes de datos. Además, se estudian los principales formatos de datos científicos como HDF (por las siglas en inglés de *Hierarchical Data Format*) y NetCDF (por las siglas en inglés de *Network Common Data Form*), teniendo en cuenta sus facultades para procesamiento paralelo y las bases de datos paralelas como otra alternativa para el almacenamiento y procesamiento de datos científicos.

3.1 Visualización científica de grandes volúmenes de datos

La visualización científica (VC) según (Morell and Pérez, 2006) significa encontrar una representación visual apropiada para un conjunto de datos, que permita mayor efectividad en el análisis y evaluación de los mismos. Simplifica el análisis, comprensión y la comunicación de modelos, conceptos y datos en la ciencia y la ingeniería.

Las aplicaciones de visualización científica de propósito general, permiten a los científicos hacer programas visuales mediante la unión de diversos módulos en una red empleando un paradigma de programación visual. Los centros de ciencia también brindan estos servicios de forma tal, que mediante un conjunto de componentes y relaciones entre ellos se definen procesos complejos en un simple bloque de construcción. A este tipo de construcción de procesos complejos a partir de componentes y servicios interrelacionados se le conoce según (Shankar, 2006) como flujo de trabajo.

La Visualización Científica ofrece grandes ventajas sobre otros métodos de análisis de datos, permitiendo representar datos de varias dimensiones o variables, logrando visualizar cuatro o más variables al mismo tiempo.

Las técnicas y tecnologías actuales permiten representar sin mucha dificultad datos no homogéneos o que no se conozca detalladamente su estructura. La exploración visual es

intuitiva, no requiere de complicados conocimientos matemáticos, estadísticos o de otra índole. Otra ventaja de la VC es la gran cantidad de conocimiento que puede ser rápidamente extraída. Consecuentemente con esto, es necesario poseer buena capacidad de cómputo para obtener los resultados en un tiempo asequible, además de tener los datos bien organizados y ubicados, por lo que los centros de ciencia surgen como una solución viable, disminuyendo en gran medida el tráfico en las redes. Todo el análisis se realiza en los centros sin necesidad de mover los datos a espacios locales. Además, con la publicación de los resultados en el centro es posible eliminar la redundancia.

Debido al tamaño de los conjuntos de datos, conviene usar la computación paralela en el centro de ciencia para realizar los cálculos de visualización y transferir imágenes en lugar de los datos crudos a las computadoras personales de los científicos. En los siguientes epígrafes se trata el tema de la visualización distribuida, principalmente la visualización orientada a red, la estructura de los sistemas de visualización y las distintas funcionalidades de los sistemas por parte del cliente y del servidor. También se profundiza en los tipos de paralelismo y renderizado paralelo, y se analizan un grupo de tecnologías para el almacenamiento de datos científicos.

3.2 Visualización distribuida

¿Qué es la visualización distribuida? Para responder esta pregunta es necesario enunciar una simple definición operacional de visualización: “Usar una computadora para darle al usuario una imagen de un conjunto de datos”. De esta forma se puede entender por visualización distribuida: “Utilizar una o más computadoras para darle a uno o más usuarios una imagen o representación visual de un conjunto de datos” (Heijmans, 2002a). Existen varias formas de realizar este mecanismo, una visión general de los distintos métodos se puede observar en la Figura 3.1, donde se usa el término CPU cuando se refiere a una computadora.

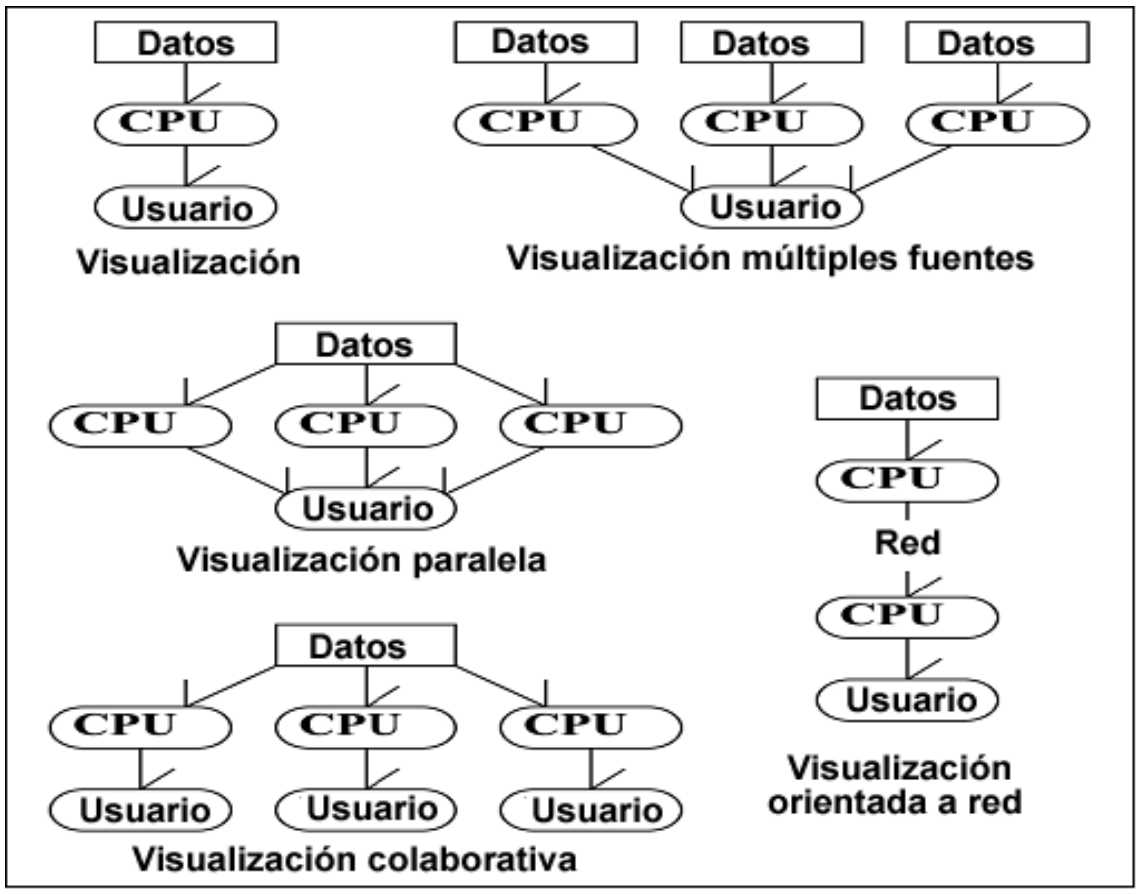

Figura 3.1 Visión general de los tipos de visualización distribuida

3.2.1 Tipos de visualización distribuida

La visualización paralela a menudo requiere un alto costo computacional, principalmente dado por la cantidad de tiempo necesario para realizar una tarea, pero gracias al avance de las tecnologías se disminuye este tiempo utilizando múltiples CPU conectados en red.

La visualización colaborativa es muy útil para facilitar el trabajo de un equipo de personas simultáneamente colaborando entre ellos, donde cada miembro del equipo tiene a la vista la misma visualización y ésta es dirigida por uno de ellos. Esta forma de trabajo elimina considerablemente la redundancia y aumenta la calidad de los resultados.

Con la visualización a partir de múltiples fuentes se combinan resultados de múltiples conjuntos de datos en una misma imagen.

La visualización orientada a la red permite al usuario crear una visualización de datos que estén en cualquier lugar fuera de su computadora, este tipo de visualización es un sistema en el cual los datos y la imagen residen en computadoras diferentes conectadas por una red (Heijmans, 2002a). Los datos y una parte del sistema se encuentran en el lado del servidor, mientras que la imagen y el resto del mismo están en el lado del cliente. La idea fundamental es que los resultados de la visualización siempre sean enviados del servidor al cliente realizando la mayor parte de la visualización en el lado del servidor.

3.2.2 Estructura del sistema de visualización

La estructura de un sistema de visualización puede ser comparada fácilmente con la estructura general de un sistema de información. La base consta de datos y metadatos descriptivos. El usuario tiene una interfaz mediante la cual observa los metadatos y los datos y con la que controla las rutinas de fondo que actúan sobre los datos. En un sistema de visualización al motor de visualización se le llama rutina de fondo. También conviene dividir la interfaz en dos partes: interfaz grafica del usuario (GUI, como indican sus siglas en ingles *Graphic User Interface*) y observador (Heijmans, 2002b). Esta estructura se puede entender mejor en la Figura 3.2.

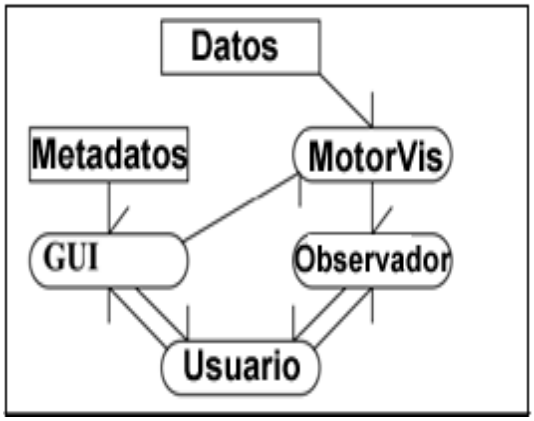

Figura 3.2 Control de flujo y visualización de datos

El usuario interactúa con la GUI para observar los metadatos y seleccionar los parámetros del motor de visualización. Los datos son transformados por el motor de visualización y mostrados a la vista del usuario por el observador.

A partir de ahora, en este epígrafe, cuando se hable de visualización, siempre se refiere a visualización distribuida y para abreviar se decidió llamar al motor de visualización del lado del servidor como (SVE) y al motor de visualización del lado del cliente como (CVE). La división no tiene que ser simétrica. Tal vez el SVE sólo envía los datos, mientras el CVE realiza la visualización, o el SVE hace todo el trabajo y el CVE sólo muestra las imágenes de información geométrica o 2D.

En la

Figura 3.3 se muestra la estructura de un sistema de visualización distribuido ilustrando el flujo de metadatos a través de él. El servidor le envía los metadatos al cliente, y estos son mostrados al usuario por la GUI.

Los metadatos tienen gran importancia, porque permiten al usuario escoger correctamente los datos y determinar los parámetros que se tienen en cuenta para la visualización. Sin embargo, muchos sistemas de visualización no tienen esto en cuenta o no especifican qué metadatos están disponibles.

Figura 3.3 Flujo de metadatos

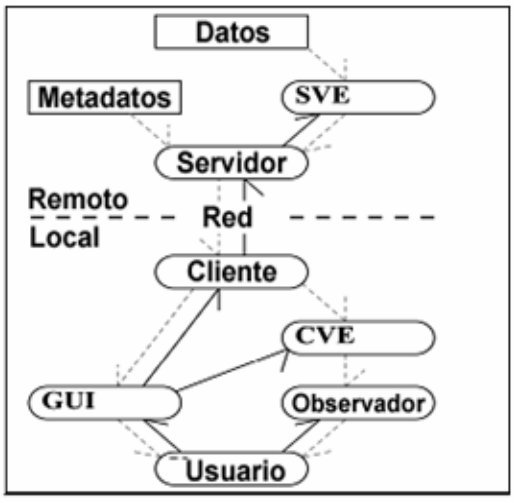

Figura 3.4 Control de flujo

En la Figura 3.4 se describe el control de flujo de un sistema de visualización. El usuario controla el observador, al CVE y al SVE a través del GUI. El usuario puede girar y escalar los objetos en el monitor y establecer los parámetros de visualización para el CVE y el SVE.

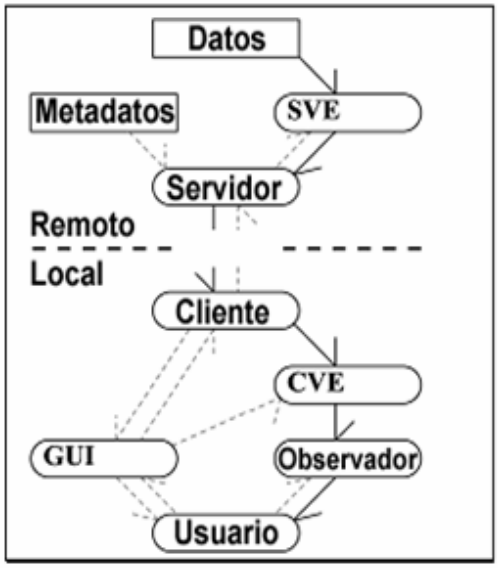

Figura 3.5 Flujo de datos

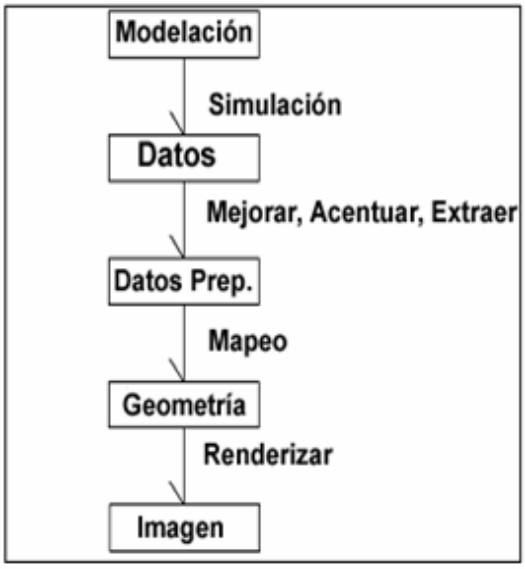

Figura 3.6 Tubería de Visualización

En la Figura 3.5 se muestra el flujo de datos a través del sistema. Los datos son procesados por el SVE, enviados del servidor para el cliente, procesados por el CVE y exhibidos por el observador para el usuario.

También se puede ilustrar el flujo de datos mediante la tubería de visualización en la Figura 3.6; para entender mejor, se muestra lo que entra y sale de las diferentes partes del sistema. La entrada del SVE son los datos, la salida del CVE es la geometría 2D/3D, la cual es también la entrada del observador. La salida del observador es una imagen 2D o una forma especial de imagen 3D siempre que esta sea soportada por el hardware.

La elección recae sobre la salida del SVE; como esta es enviada sin alteración sobre la red, también es la entrada del CVE. Es fácil concluir que para la tubería de visualización hay tres opciones: que envíe los datos, los datos preparados o la geometría 2D/3D. Esto también define la distribución de funcionalidad entre el servidor y el cliente. Si todos los datos son

enviados por la red, el SVE no hace nada, mientras el CVE realiza la visualización. Si la geometría 2D/3D es enviada, el CVE no hace nada, la visualización es realizada por el observador.

Entonces, se presentan dos criterios importantes para definir la estructura de un sistema de visualización distribuido:

¿Qué es enviado por la red, o qué se hace por el servidor y qué por el cliente?

¿Qué control tiene el usuario sobre el proceso de visualización?

Cuando los datos están en un formato que el SVE no puede comprender directamente es necesario la conversión de los mismos para realizar la visualización, por lo cual se debe disponer de una utilidad de conversión o usar un módulo especial para importar los datos en el motor de visualización; por razones de brevedad se decide no incluir los pasos de conversión en los diagramas, conceptualmente cabe colocar la conversión de datos dentro de la fase de preparación de los datos. En ese caso no hay necesidad de distinguir conversión de datos de otros pasos de la visualización.

Hay tres niveles básicos de control:

1. El usuario controla al observador: Este es el nivel más bajo de interacción, el usuario puede controlar sólo su punto de vista, pero no tiene el control de la visualización. Por consiguiente, la visualización puede llamarse apenas interactiva.

2. El usuario controla la visualización en el lado del cliente: Si se envían todos los datos al cliente, la interacción se realiza según las necesidades del usuario, por lo tanto él puede cambiar la tubería de visualización y los parámetros de la misma.

3. El usuario controla la visualización en el lado del servidor: Si no se envían todos los datos al cliente, es decir, si lo que se envía es una selección o incluso el resultado de una visualización, entonces el usuario debe poder controlar la visualización del servidor mediante un formulario para que seleccione los parámetros según su interés.

3.3 Tubería de Renderizado Paralelo

Un proceso típico de renderizado paralelo de un volumen consta de cuatro pasos. El paso de lectura de los datos en disco y la distribución para los procesadores. Cada procesador recibe un subconjunto del volumen de los datos. En el siguiente paso, cada procesador renderiza el subvolumen asignado en una imagen 2D parcial que es totalmente independiente de los demás procesadores. Después, le sigue un paso de combinación, que generalmente requiere comunicación entre los procesadores, donde se mezclan las imágenes 2D parciales para formar la imagen 2D final. En el paso final se le muestra la imagen al usuario o se almacena para su posterior análisis.

Dado un volumen genérico que debe ser renderizado en paralelo y un clúster de P-Procesadores, según (Hansen and Johnson, 2005), existen tres formas posibles de administrar los procesadores para renderizar los volúmenes de datos, teniendo en cuenta el tiempo necesario para obtener la imagen final. El primer método simplemente es correr el volumen que se va a renderizar como una secuencia de subconjuntos uno tras otro. En cualquier punto de tiempo, el clúster de P-Procesadores está dedicado completamente al renderizado de un volumen particular. El paralelismo solo se asocia con el renderizado de un solo volumen de datos, en esencia solo es explotado el paralelismo intravolumen.

El segundo método utiliza exactamente la estrategia opuesta, renderizar P volúmenes de datos simultáneamente, cada uno en un procesador. Este modus operandi sólo explota el paralelismo de intervolumen, y está limitado por el espacio principal de memoria de cada procesador.

Para lograr un renderizado óptimo, es necesario balancear dos factores en el funcionamiento: el uso eficiente de los recursos y la paralelización. Esto sugiere explotar ambos tipos de paralelismo, el paralelismo de intravolumen y el paralelismo de intervolumen. En lugar de usar todos los procesadores para el renderizado colectivamente de un volumen a la vez, se forma un proceso de tubería de renderizado dividiendo los procesadores en grupos para renderizar varios volúmenes concurrentemente. De este modo según (Hansen and Johnson, 2005), el tiempo global de renderizado puede ser minimizado

considerablemente, porque las tareas de la tubería de renderizado son superpuestas con la E/S requerida para cargar cada volumen en un grupo de procesadores.

El tercer método es un híbrido, en el cual los nodos de los P-procesadores están subdivididos en L grupos, (1<L<P), cada uno de los cuales renderea un volumen a la vez. La elección óptima de L generalmente depende del tipo y la escala del clúster así como del tamaño del conjunto de datos. La estrategia óptima de partición de discos para minimizar el tiempo global de renderizar puede ser caracterizada con el modelo de funcionamiento y revelada con un estudio experimental, lo cual demuestra que el tercer método ciertamente es el mejor de los tres.

3.4 Tipos de paralelismo

A continuación se tratan distintos métodos y algoritmos que pueden emplearse para procesar y visualizar grandes conjuntos de datos en paralelo. Los cuatro tipos de paralelismo tratados en esta sección son: flujo de datos, paralelismo de tareas, paralelismo de tubería y paralelismo de datos.

3.4.1 Flujo de Datos

Flujo de Datos (FD) tratado en (Hansen and Johnson, 2005), es el método más usado para procesar subconjuntos independientes de un gran conjunto de datos, siendo procesado un subconjunto a la vez. Éste, a menudo es el único método factible en situaciones donde el tamaño de un conjunto de datos excede la capacidad de los recursos disponibles en cuanto a memoria y espacio de intercambio. Por ejemplo, no es raro para conjuntos de datos científicos, especialmente series de tiempo, que exceda fácilmente la cantidad de memoria disponible. La ventaja crucial de este método es que todos los conjuntos de datos de cualquier tamaño pueden ser tratados exitosamente. El inconveniente de esta técnica es que a menudo requiere mucho tiempo de ejecución y no permite la exploración interactiva de los datos. El uso de un subsistema de E/S de alto rendimiento es ventajoso para mejorar el funcionamiento global de los algoritmos que utilizan esta técnica.

Este método necesita que los datos sean divididos en pedazos para irlos procesando poco a poco y debe ser invariante con respecto a la cantidad de pedazos en que se dividan los

datos. Varios algoritmos necesitan conocer los valores de datos contenidos en celtas vecinas para producir los resultados correctos, es por eso que es recomendable hacer una división adecuada.

3.4.2 Paralelismo de Tareas

Con Paralelismo de Tarea, los módulos independientes de una aplicación se ejecutan paralelamente. En la Figura 3.7, esto es logrado haciendo que los módulos A, B, y C se ejecuten al mismo tiempo. Requiere un algoritmo que resuelva tareas independientes y tener disponible múltiples recursos computacionales. La ventaja crucial de esta técnica es que permite ejecutar paralelamente tareas de visualización de múltiples porciones de datos. Su principal desventaja es que el número de tareas independientes que pueden ser tratadas tiene que ser igual al número de CPU disponibles. Además, puede ser difícil establecer el balance de carga de las tareas y por eso, a menudo se hace imposible aprovechar al máximo los recursos disponibles.

El paralelismo de tarea es usado eficazmente en la industria cinematográfica, en el que varios marcos en una producción animada son renderizados en paralelo. Las elecciones específicas del hardware para mejorar el funcionamiento del paralelismo de tarea depende de los detalles de las tareas, este tema es analizado detalladamente en (Hansen and Johnson, 2005).

3.4.3 Paralelismo de Tubería

El Paralelismo de Tubería ocurre cuando un número de módulos en una aplicación se ejecutan paralelamente, pero en subconjuntos independientes de datos. En la Figura 3.7, esto ocurriría cuando los módulos A, D, y E le hacen una operación a porciones independientes de datos. Este método es más conveniente para situaciones donde hay múltiples tareas heterogéneas. La ventaja de esta estrategia es que permite uso paralelo de los recursos de cómputo. Por ejemplo, un proceso puede estar leyendo de disco, otro computando resultados utilizando al CPU y un tercer proceso usando una tarjeta aceleradora de gráficos. La desventaja principal es que puede ser difícil balancear el tiempo de ejecución requerido por las distintas etapas; en una tubería desnivelada, la etapa más lenta afecta directamente la función global. Además, la longitud de la tubería limita

directamente la cantidad de paralelismo que puede ser logrado y se deben tener tantos procesadores disponibles como etapas haya en la tubería. Para minimizar el tiempo de ejecución, hay que mover rápidamente los datos de una etapa de la tubería a la siguiente. Esta estrategia requiere el uso de arquitecturas de memoria compartida y una red de interconexión de alta velocidad entre los procesadores.

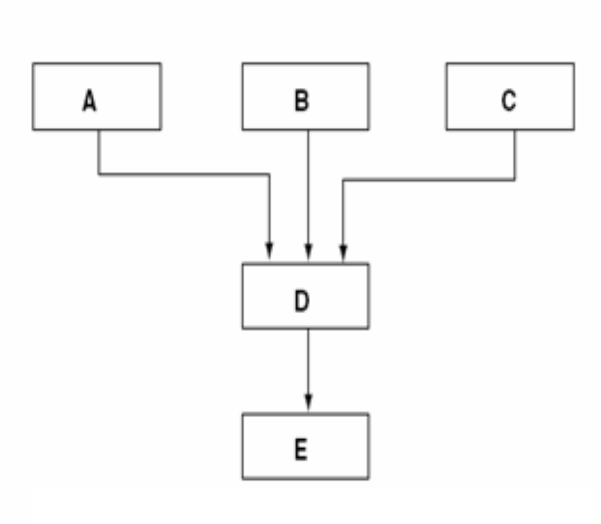

Figura 3.7 Módulos de ejecución

3.4.4 Paralelismo de Datos

Mediante Paralelismo de Datos, el código dentro de cada módulo de una aplicación se ejecuta paralelamente. En lo referente a la Figura 3.7, esto ocurre cuando el código dentro del módulo A corre en paralelo. Esto requiere que el conjunto de datos sea subdividido en múltiples procesos que corren el mismo algoritmo en las partes resultantes concurrentemente. El paralelismo de datos puede ser implementado como una extensión de la técnica de descomposición de datos descrita en la sección de flujo de datos. En este caso, los datos se subdividen de la misma forma, pero se tiene un paso extra de asignar un proceso para cada una de las partes resultantes. Este método es comúnmente llamado modelo de programa simple de múltiples datos, porque cada proceso ejecuta el mismo programa con subconjuntos diferentes de datos. La principal ventaja de este método es que se puede lograr un alto grado de paralelismo. Las soluciones tienden a escalarse adecuadamente aumentando el número de procesadores. Cuando hay un gran número de procesadores disponibles, este método es a menudo uno de los mejores para lograr alto

rendimiento. Un posible inconveniente es que la dimensionalidad puede estar limitada por el costo de la comunicación entre los procesos. Para lograr el mejor funcionamiento posible con paralelismo de datos, es a menudo importante considerar los costos de comunicación y localidades de datos entre los procesadores. Las mejores situaciones posibles para este método, son cuando no hay dependencia entre los procesos; las peores son, cuando cada proceso está obligado a compartir información con los otros. Afortunadamente, muchos algoritmos de visualización tienen pocas dependencias de comunicación entre procesos y por eso, el paralelismo de datos es a menudo una de las técnicas más efectivas para lograr alto funcionamiento, como se explica en (Hansen and Johnson, 2005).

3.4.5 Resumen de los tipos de paralelismo

Las secciones precedentes han esbozado una clasificación de técnicas basadas en la descomposición de tareas y datos. La Tabla 3.1, presenta un mapeo de las características del problema y la solución soportada. La columna de tamaño del conjunto de datos describe la cantidad de datos a visualizar. Un conjunto de datos es considerado grande cuando excede los recursos de una sola máquina. La columna de tareas describe el tipo de trabajo. Las tareas homogéneas son un mismo tipo de trabajo que se le aplica a diferentes datos. Las tareas independientes pueden ser corridas al mismo tiempo en paralelo. Las tareas secuenciales deben correr una tras otra en un orden fijo. La columna de recursos describe el número de recursos disponibles y la columna de solución identifica la técnica que debe usarse en cada caso. Por ejemplo, si usted tiene un conjunto de datos de gran escala y sólo una CPU, flujo de datos, quizás sea la única solución posible por la cual se pueden explorar la totalidad de los datos.

Tabla 3.1 Resumen de los tipos de paralelismos

Tamaño de Datos	**Tareas**	**Recursos**	**Solución**
Grandes	Cualquiera	Una CPU	Flujo de Datos
Grandes	Homogénea	Múltiples CPU	Paralelismo de datos
Cualquiera	Independiente	Múltiples CPU	Paralelismo de tareas
Cualquiera	Secuencial	Múltiples CPU	Paralelismo de Tubería

3.5 Tipos de renderizado paralelo

Cuando se trabaja visualizando conjuntos de datos del orden de los tera bytes o peta bytes, no es raro encontrarse algoritmos de visualización que produzcan billones de primitivas gráficas, lo cual sobrepasa la capacidad de una simple CPU y un acelerador gráfico para renderizar. El uso de los tipos de renderizado paralelo es necesario para lograr un adecuado funcionamiento en el procesamiento de estos grandes conjuntos de datos. En esta sección se estudia la clasificación mas conocida de este tipo de métodos que están basados en la localización dentro de la tubería de renderizado. Estos métodos son *Sort-First*, *Sort-Middle*, y *Sort-Last* (Hansen and Johnson, 2005).

3.5.1 Sort-First

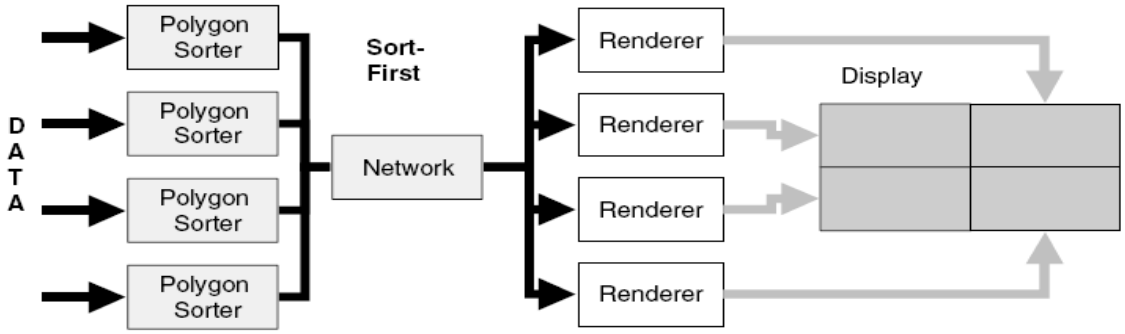

Figura 3.8 *Sort-First según* (Hansen and Johnson, 2005)

Los algoritmos *Sort-first* empiezan por distribuir primitivas de gráficos en el principio de la tubería de renderizado, asignando las primitivas a los procesadores, subdividiendo la imagen de salida y asignando un procesador para maniobrar cada región resultante. Una vez que las primitivas han sido asignadas, cada proceso completa la tubería de gráficos para producir la subimagen final. La asignación inicial de primitivas para procesadores es el paso crucial en los algoritmos *sort-first*, requiere la transformación de las primitivas en coordenadas del espacio de escritorio e introduce unos gastos computacionales fijos al algoritmo. La asignación inicial para los procesadores se realiza generalmente de forma arbitraria, excepto después de que cada procesador completa la etapa de transformación de la tubería que reasigna primitivas para los procesadores adecuados. La consideración adicional debe ser dada a las primitivas que traslapan las regiones subdivididas de la imagen final.

Sort-first es ventajoso porque los procesadores implementan la tubería de renderizado completa, permitiendo el uso de una tubería acelerada por hardware, y/o buen comportamiento del sistema de memoria caché. Además, los requisitos de comunicación entre los procesadores pueden ser bajos, resultantes de una menor sobrecarga y mejor funcionamiento. La desventaja principal es que la distribución inicial de las primitivas entre los procesadores puede fácilmente conducir a un desequilibrio de carga de trabajo. Tratar de redistribuir las primitivas durante el proceso de renderizado también puede conducir a una pobre escalabilidad debido a las cantidades de tráfico requerido por el envío de mensajes.

3.5.2 Sort-Middle

En los algoritmos *sort-middle*, la redistribución de datos ocurre entre el procesamiento de la geometría y los estados de conversión y exploración de la tubería de renderizado. En este caso, es común describir los pasos como dos conjuntos de operaciones. La primera parte manipula la porción de la geometría de la tubería (las transformaciones, la iluminación, etc.), y las primitivas están inicialmente asignadas en una forma arbitraria. El segundo grupo de operaciones es asignado a regiones continuas de la imagen final de salida y estas operaciones son responsables de convertir una imagen vectorial en una foto computarizada en el espacio de pantalla, cuyas primitivas fueron producidas por el primer conjunto de procesadores. Note que es posible dedicarle procesadores a cada uno de estos conjuntos separadamente o dejar todos los procesadores realizar ambas tareas. Si los procesadores se usan de forma separada, es posible crear un conjunto de tubería en paralelo.

La división en la tubería de renderizado es la desventaja más notable de *sort-middle*, dificulta aplacar el acelerador de hardware de renderizado, además del costo de comunicación entre las etapas de balanceo de la tubería con el número de primitivas renderizadas. Finalmente, el algoritmo puede padecer de desequilibrio de carga cuando las primitivas no son uniformemente distribuidas a través de la imagen de salida.

3.5.3 Sort-Last

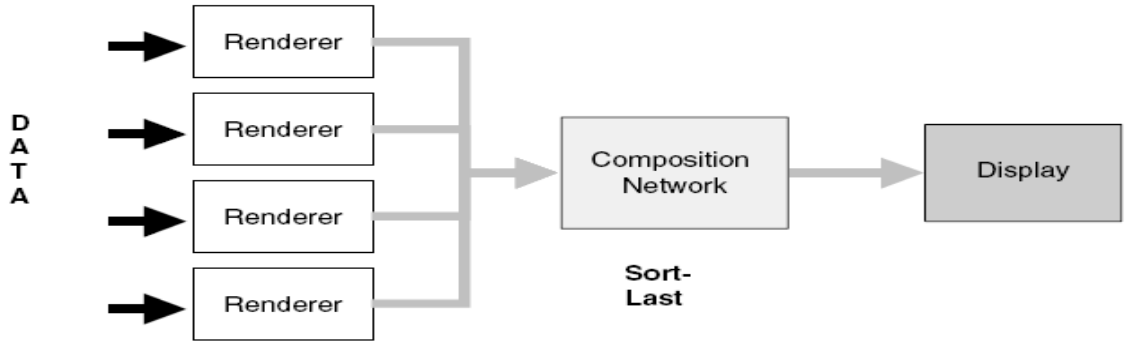

Figura 3.9 *Sort-Last según* (Hansen and Johnson, 2005)

El método *sort-last* atrasa la ordenación de las primitivas hasta las etapas finales de la tubería de renderizado. Las primitivas pueden estar inicialmente asignadas en forma arbitraria y cada procesador obtiene una porción de la imagen final. Todas estas imágenes, luego son compuestas para formar la imagen final completa. Al igual que con *sort-middle*, es posible usar dos conjuntos de procesadores. Los primeros son los responsables de completar la tubería de renderizado y los otros conjuntos de procesadores son los responsables de crear la imagen final. Esta asignación también tiene prevista la creación de una tubería paralela, que puede usarse para mejorar el funcionamiento global. Dado que las etapas de renderización de *sort-last* crean imágenes de resolución completa, la red de interconexión entre procesadores debe tener muy alto ancho de banda si se quiere renderizar interactivamente. Otras técnicas pueden usarse para reducir la cantidad de tráfico requerido para completar la etapa de composición.

Sort-last incluye la habilidad de implementar la tubería de renderizado completa. Es más fácil distribuir las primitivas uniformemente para los procesadores y así evitar asuntos de balanceo de carga. La principal desventaja es el alto costo de comunicación que introduce el envío de grandes cantidades de datos entre los procesadores. Las técnicas para reducir el costo de este tráfico de mensajes constituyen todavía un área de investigación en creciente desarrollo.

3.6 Tecnologías para el almacenamiento de datos científicos

Los archivos secuenciales sin lugar a dudas resultan eficientes en el almacenamiento de datos científicos y permiten procesar y almacenar estos datos de forma relativamente sencilla.

Los científicos demandaban para su actividad un método eficiente e independiente para la lectura y escritura de datos científicos que permitiera desarrollar bibliotecas específicas para el almacenamiento de los datos científicos. Una de las implementaciones más usadas son los archivos binarios con formatos estándares. Según (Gray et al., 2005), existe una tendencia de los formatos de datos científicos (HDF, NetCDF, FITS,..) a centrarse principalmente en aspectos de los metadatos e intercambio de datos y los sistemas tradicionales de gestión de datos (SQL y otros) a concentrarse en la gestión y el análisis de grandes conjuntos de datos almacenados principalmente en bases de datos paralelas. Los sistemas de bases de datos tradicionales tienen las virtudes de paralelismo automático, indexación y de acceso no procedural, pero tienen que adoptar los tipos de datos de la comunidad científica y necesitan coexistir con los datos en el sistema de archivos.

Estos formatos incluyen estructuras para soportar grandes cantidades de datos, arreglos multidimensionales, una amplia variedad de tipos de datos multivariados, multievaluados y métodos para la gestión de metadatos.

En este epígrafe se presentan algunas características de los principales formatos de datos científicos y las bases de datos paralelas.

Entre los formatos estándares antes mencionados, ha sido de más utilidad para esta investigación HDF, principalmente por la posibilidad del trabajo en paralelo a partir de la versión HDF5 y NetCDF, por su gran uso en los diferentes centros de ciencia. En este epígrafe se tratan de forma específica estos formatos de datos científicos, profundizando en su estructura y las funcionalidades de su biblioteca de funciones.

3.6.1 HDF

HDF (*Hierarchical Data Format*) es una biblioteca y un formato de fichero multiobjeto para la transferencia de datos gráficos y numéricos entre máquinas.

HDF se encuentra libremente disponible. La distribución consiste en la biblioteca HDF y las utilidades de línea de comando. Las utilidades de línea de comando de HDF permiten convertir de un formato a otro (por ejemplo JPEG a HDF y viceversa), analizar y ver los ficheros HDF así como manipularlos.

Las interfaces de programación de aplicaciones incluyen conjuntos de rutinas para el almacenamiento y acceso a tipos de datos específicos donde todos lo detalles de bajo nivel pueden ser ignorados. Estas bibliotecas se encuentran disponibles en C y Fortran. Los tipos de estructura de datos que soporta HDF son conjuntos de datos científicos, imágenes de puntos y paletas de colores, en (Cheng et al., 2003) se pueden encontrar detalles sobre el manejo de estas herramientas.

HDF 4.0 y versiones posteriores soportan una interfaz de compresión de bajo nivel, la cual permite que cualquier objeto de dato sea comprimido utilizando una variedad de algoritmos.

A continuación se muestran una serie de características del paquete HDF:

- **Es versátil**. HDF soporta muchos tipos diferentes de modelos de datos. Cada modelo define un conjunto específico de tipo de dato y provee una API (de las siglas en inglés de *Application Program Interface*) para lectura, escritura y organización de datos y metadatos del tipo correspondiente. Los modelos soportados incluyen arreglos multidimensionales, imágenes de puntos y tablas.

- **Es auto-descriptivo**. Posee una aplicación para interpretar con la estructura y contenidos de los archivos HDF sin requerir información alguna proveniente del exterior, esta aplicación se basa en los metadatos descriptivos para inspeccionar los archivos HDF.

- **Es flexible**. Con HDF, se pueden mezclar y asociar objetos relacionados agrupados en un fichero y acceder a ellos como un grupo o como objetos individuales. Los usuarios pueden además, crear sus propias estructuras de agrupamiento utilizando un rasgo de HDF llamado *vgroups*.

- **Es extensible**. Puede acomodar fácilmente nuevos modelos de datos, sin tener en cuenta si fueron adicionados por el equipo de desarrolladores de HDF o por los usuarios de HDF.

- **Es portable**. Los archivos HDF pueden ser compartidos a través de la mayoría de las plataformas comunes, incluyendo muchas estaciones de trabajo y computadoras de alto desempeño. Un archivo HDF creado en una computadora puede ser leído en un sistema diferente sin modificación alguna.

Además de las principales características del HDF estándar, para utilizar HDF paralelo, hay que tener en cuenta un conjunto de aspectos:

- Los programas paralelos de HDF tienen que ser compatibles con archivos HDF5 y pueden ser compartidos entre diferentes plataformas paralelas.

- La característica fundamental de esta tecnología es el trabajo con plataformas paralelas y una interfaz de Entrada-Salida paralela portátil para diferentes plataformas. HDF5 Paralelo tiene como meta inicial el soporte de la programación MPI pero no la programación de memoria compartida, pues tuvo que ser diseñado sobre la idea del trabajo con un archivo único accedido por todos los procesos en lugar de un archivo por proceso, tener un archivo por proceso puede provocar post procesamiento costoso y los archivos no son utilizables por diferentes procesos. (Cheng et al., 2003).

- Los archivos de HDF5 son tratados de manera muy similar a los directorios. Es necesario antes de crear un archivo, asegurarse de que el directorio sobre el cual se va a trabajar esté creado, de no ser así es necesario hacerlo, facilitando el trabajo con dicho archivo desde cualquier lugar del sistema sin pérdida de información. Los lenguajes de programación usados para implementar las aplicaciones de HDF5 Paralelo son Fortran, C y C++.

3.6.2 NetCDF

NetCDF es una interfaz a una biblioteca de funciones de acceso a datos diseñada para el almacenamiento y recuperación de datos en forma de arreglos, donde cada arreglo es una estructura rectangular n-dimensional (con n=0,1,2,k) en la que todos sus elementos son del mismo tipo de dato (ejemplo: caracteres de 8-bit, entero de 32-bit) y un escalar (un solo valor) es un arreglo 0-dimensional.

La biblioteca NetCDF implementa un tipo de datos abstracto, por lo que incluye todas las operaciones para acceder y manipular datos. En un conjunto de datos de NetCDF se deben usar sólo las funciones que provee la interfaz (Gao, 2008). La representación de los datos está oculta para aplicaciones que usen la interfaz, así que la forma en la cual estos se encuentran almacenados puede ser cambiada sin los programas existentes. La representación física de los datos de formato NetCDF está designada para ser independiente de la computadora en la cual sean guardados.

Una de las metas de NetCDF es soportar acceso eficiente a pequeños subconjuntos de grandes conjuntos de datos. Para lograr esto, NetCDF utiliza accesos directos en lugar de accesos secuenciales. Esto puede ser mucho más eficiente cuando el orden en que los datos son leídos sea diferente del orden en que fueron escritos, o cuando debe ser leído en orden diferente para disímiles aplicaciones.

La cantidad de costo para la representación externa portable depende de muchos factores, incluyendo el tipo de dato, el tipo de computadora, la granularidad del acceso a datos y de cuán bien se haya ajustado la implementación a la computadora en la que está corriendo. Este costo es usualmente pequeño en comparación con todo el conjunto de recursos utilizados por una aplicación. De cualquier forma, el costo de la capa de representación externa es usualmente un precio razonable a pagar por acceso de datos portables.

A pesar de que la eficiencia en el acceso a datos ha sido un asunto importante en el diseño e implementación de NetCDF, todavía es posible utilizar la interfaz NetCDF para acceder a datos de formas ineficientes: por ejemplo, mediante la petición de una porción de datos que requiere un solo valor de cada registro.

3.6.3 Bases de Datos Paralelas

Una de las implementaciones más eficientes para las Bases de Datos Científicas, debido a su exitoso desempeño basado en su gran capacidad de cómputo y de almacenamiento, son las Bases de Datos Paralelas. Para comprender mejor este concepto es necesario en primer lugar, conocer las bases de datos distribuidas. Según (DeWitt et al., 1992) una Base de Datos Distribuida es una colección múltiple de datos lógicamente relacionada a través de una red de computadoras. Un sistema de administración de bases de datos (DBMS) distribuido se define como un sistema de software que permite la fragmentación y la distribución de los datos de forma transparente para los usuarios, la forma en que se implementa esta distribución resulta interesante para esta investigación, pues es un factor determinante a la hora de gestionar los datos en esta alternativa de implementación. Una de las características principales de las Bases de Datos paralelas está determinada por la arquitectura sobre la cual basan su funcionamiento, que puede variar de acuerdo a la capacidad económica de la institución que la implementa.

La paralelización de operaciones consiste en separar una operación a procesar, en sub-operaciones; cada una de las cuales es ejecutada de forma paralela en cada procesador para luego reunir los resultados en una sola respuesta, este tipo de procesamiento se logra mediante el uso de las llamadas Bases de Datos Paralelas (DeWitt et al., 1992).

Estas Bases de Datos usualmente se utilizan en una red de alta velocidad que enlaza varios computadores (clúster de computadoras); todos ellos trabajando conjunta y simultáneamente (procesadores, memoria principal y discos de almacenamiento) para constituir una base de datos paralela.

Los Sistemas de Bases de Datos Paralelas se comenzaron a desarrollar en los años 80 sobre un sistema de procesadores de propósito general funcionando en paralelo. Sobre este sistema se basan los sistemas actuales de IMB, Oracle, Infomix entre otros. Los Sistemas Paralelos de Bases de Datos trabajan de forma conjunta ejecutando consultas y transacciones de los clientes. Esta forma de ejecutar las transacciones mejora considerablemente la productividad, ejecutando sub-transacciones de forma simultánea. La

arquitectura es sin lugar a dudas un factor determinante para la implementación de una base de datos, pues existen varios modelos para el procesamiento paralelo (Brewer, 2004):

1. **Memoria compartida:** Todos los procesadores comparten una memoria común. Es muy eficiente en cuanto a comunicación entre procesadores.

2. **Disco Compartido:** Cada procesador posee su memoria privada, pero tiene acceso directo a todos los discos. Este modelo es adecuado para grandes bases de datos de solo lectura y para aquellas donde no exista compartimiento concurrente. Este esquema no es muy efectivo para aplicaciones de bases de datos que necesiten accesos de lectura y escritura en una BD compartida.

3. **Sin compartimiento:** Cuando los procesadores no comparten ni memoria, ni discos, cada nodo contiene un procesador, memoria y sus propios discos. Debido a que los datos se transportan desde los discos hasta la memoria dentro de un mismo nodo, se minimizan las colisiones en la red. Este tipo de arquitectura es escalable a miles de nodos.

4. **Jerárquico:** Cuando los procesadores comparten memoria principal y los discos de almacenamiento.

Es importante también señalar los tipos de paralelismo que se pueden encontrar en estas arquitecturas:

Paralelismo de entrada-salida: Permite la reducción del tiempo necesario para la recuperación de datos, dividiendo la base de datos entre varios discos a través de una fragmentación horizontal.

1. **Paralelismo entre consultas:** Permite la ejecución de consultas o transacciones en paralelo.

2. **Paralelismo en consultas:** Permite que una única consulta se ejecute de forma paralela en diferentes procesadores y discos.

3. **Paralelismo entre operaciones.**

a) **Paralelismo de Entrecruzamiento:** Permite ejecutar varias operaciones sin necesidad de acceder al disco para almacenar datos intermedios.

b) **Paralelismo Independiente:** Cuando se ejecutan varias operaciones independientes que pertenecen a la misma consulta.

La arquitectura de un sistema de base de datos paralela está basada fundamentalmente en un hardware sin compartimiento, donde los procesadores se comunican entre sí enviando mensajes mediante una red de interconexión. En dichos sistemas, las tuplas de cada relación son particionadas para permitir que múltiples procesadores escaneen grandes relaciones paralelas sin necesitar dispositivos de E/S. Dicho diseño es utilizado actualmente por Teradata (Teradata, 1983), Tandem (Tandem, 1988), Oracle-nCUBE (Gibbs, 1991), y por algunos otros productos de bajo desarrollo. La comunidad de investigadores también ha abarcado esta arquitectura de flujo de datos sin compartimiento en sistemas como Arbre, Bubba, y Gamma, mencionados en (DeWitt et al., 1992).

Las bases de datos paralelas presentan importantes ventajas con respecto a otros sistemas de bases de datos pues permiten un aumento en la velocidad, necesitando menor tiempo de ejecución para cada transacción. También se logra la ampliación al poder procesar tareas más largas en el mismo tiempo. A pesar de estas posibilidades que brindan las bases de datos paralelas, el costo de inicialización es alto a la hora de la instalación y puesta a punto del sistema.

A pesar de parecer similares, los sistemas de bases de datos paralelas se diferencian de las distribuidas, ya que estas se encuentran generalmente en diferentes puntos geográficos, se administran de forma separada y poseen una interconexión más lenta. También en los sistemas distribuidos se dan dos tipos de transacciones: locales y globales. Las transacciones locales son aquellas que se ejecutan solamente en el nodo donde se inicia la transacción, mientras que las globales involucran intercambio de datos entre los distintos nodos que conforman el sistema, en un sistema paralelo todas las transacciones están al mismo nivel, ya que todas son atendidas por el sistema en conjunto (independientemente de ser paralelizadas o no). Las bases de datos paralelas trabajan con arquitectura de

multiprocesadores posibilitando altos rendimientos y disponibilidad para servidores de bases de datos con menor costo (DeWitt et al., 1992).

De forma general las Bases de Datos Paralelas superan la velocidad de las consultas de un sistema de base de datos a través de la ejecución de forma simultánea de las transacciones; sin embargo, el costo de la implementación es mucho mayor que otros tipos de Bases de Datos. Los sistemas de bases de datos paralelas están teniendo una gran aceptación comercial, pero solo en organizaciones que tienen sistemas de bases de datos grandes y complejos.

3.7 Conclusiones parciales

En este capítulo se realizó un estudio sistemático de las tecnologías para el almacenamiento de grandes volúmenes de datos, haciendo énfasis en las facilidades que brindan para la visualización científica. Se comprobó, que varios de los formatos para el almacenamiento de datos científicos que se emplean actualmente son utilizados en los centros de ciencia relacionados en el capítulo 2. Estos formatos resultan adecuados para considerar su inclusión en un modelo de centro de ciencia basado en un clúster de computadoras. Particularmente se consideran apropiados netCDF y HDF, por su libre disponibilidad y sus posibilidades para el trabajo de forma paralela.

Se analizaron las características específicas de la visualización de grandes volúmenes de datos, las ventajas de las técnicas de visualización distribuida y los diferentes tipos de paralelismo y renderizado paralelo. La visualización científica se presenta como una alternativa eficaz para el análisis de grandes volúmenes de datos. Resulta, por tanto, una excelente herramienta que pueden ofrecer los centros de ciencia para el estudio y análisis de los datos que almacenan. Los diferentes modelos de visualización distribuida descritos en este capítulo pueden implementarse, en principio, en un modelo de centro de ciencia basado en un clúster de computadoras. La combinación de estos modelos con los diferentes tipos de paralelismo fortalece la potencia de las herramientas de análisis de datos que puede ofrecer un centro de ciencia basado en un clúster de computadoras.

4 Modelo general de un centro de ciencia para la visualización científica basado en un clúster de computadoras

Después de realizar el análisis de las características de los centros estudiados en el capítulo 2 y los diferentes escenarios que se presentan en la visualización científica de grandes volúmenes de datos, tratados en el capítulo 3, se puede dar cumplimiento al objetivo principal de esta investigación: diseñar el modelo general en el que se trata las principales funcionalidades de un centro de ciencia para la visualización científica creado sobre la tecnología de clúster de computadoras. Diseño que permitirá en futuros trabajos, la implementación de un CC con las características deseadas en un clúster de computadoras.

En este capítulo se parte de una serie de requisitos indispensables para un centro de ciencia y luego se analizan los aspectos soportados sobre la tecnología de clúster de computadoras. Se crea un diseño lógico de la arquitectura del centro y se analizan las distintas variantes que se requieren para el manejo de los datos. Posteriormente, se tratan aspectos del módulo de visualización y se brinda una solución tecnológica donde se especifica el tratamiento tanto de los datos como de los metadatos.

4.1 Requisitos asumidos para el diseño del centro de ciencia

El servicio más importante en el centro es brindar la visualización científica. Hay que tener en cuenta que un CC para la visualización tiene más accesos de lectura de datos que de escritura, algo que es común en los sistemas de visualización. El tiempo de respuesta de este servicio tiene que ser lo más rápido posible, debido a que cuando se visualiza un conjunto de datos, es con el objetivo de interpretar la imagen, ajustar algunos parámetros y volver a visualizar, por lo que es posible que un usuario tenga que realizar este proceso varias veces hasta obtener el resultado deseado, si no se logra un buen tiempo de respuesta esto podría causar que el usuario desista de utilizar el sistema.

La capacidad de procesamiento y almacenamiento es un aspecto clave en un CC, condición que poseen los clústers de computadoras con discos duros. Además, la escalabilidad del

clúster permite adicionar más nodos y discos duros, en caso de que se necesite aumentar la capacidad de cómputo o de almacenamiento en cualquier momento.

Se asume que el clúster de computadoras en que se implemente este diseño tiene nodos con características similares: igual sistema operativo, el mismo sistema de archivos y en el caso de tratarse de un clúster de BD, el mismo gestor de base de datos para facilitar la implementación, aunque perfectamente pueden utilizarse varios gestores de BD.

Un aspecto fundamental es la conservación de la integridad de los datos almacenados en el CC para que el sistema sea tolerante a fallos. En caso de que ocurra algún problema de pérdida de información, es necesario disponer de mecanismos de auto recuperación que permitan mantener la integridad de los datos. El uso de la replicación es una alternativa. Con los archivos replicados, al ocurrir cualquier fallo en el sistema, ya sea la pérdida de datos por cualquier motivo, como la rotura de un disco duro, el sistema debe detectar estos problemas y si es posible solucionarlos de forma automática, restableciendo los datos a partir de un disco espejo o en caso que no se pueda, informarle al administrador el problema detectado. Estos son aspectos de los cuales se ocupa el solucionador de recursos de almacenamiento que se seleccione para el CC.

El acceso concurrente a datos es una característica importante, no obstante, este aspecto no es tan preocupante porque todos los sistemas de archivos o sistemas de gestión de bases de datos se ocupan de esta tarea, aunque no siempre de manera distribuida.

Como se mencionó en el epígrafe 2.1, un requisito esencial de los CC es brindar la posibilidad a los usuarios de tener espacios personales de trabajo, donde puedan almacenar los datos que con más frecuencia utilizan y donde puedan guardar los resultados de sus investigaciones. A través de estos espacios personales, un usuario del centro puede compartir sus datos con otros, por lo que el espacio de trabajo personal es un aspecto clave del diseño.

La manipulación de distintos tipos de metadatos es otro requisito fundamental. El manejo de catálogos de metadatos en sistemas distribuidos, facilita mantener la ubicación de cada uno de los archivos en los diferentes nodos del clúster, así como otras informaciones

importantes referentes a ellos: la fecha, el propietario y cualquier característica que sea de interés, sin entrar en detalles de los datos en sí. Los metadatos descriptivos son necesarios para comprender los datos, localizar conjuntos de datos dentro de un archivo y saber por los distintos procesos que estos han pasado.

Después de estudiar las características imprescindibles para el diseño, en el siguiente epígrafe se analizan los aspectos del diseño de los centros de ciencia tratados en el capítulo 2 que se puedan soportar en un clúster de computadoras. Esta tarea se realiza con vistas a satisfacer estos requerimientos y resaltar algunos que no se utilizan.

4.2 Diseño de un centro de ciencia sobre un clúster de computadoras

En un clúster de computadoras se pueden implementar prácticamente todas las funcionalidades de los centros de ciencia; pero es necesario excluir algunas condiciones de heterogeneidad que poseen los CC para ajustarse a las condiciones de un clúster de computadoras con similares configuraciones. Por ejemplo, limitar las funcionalidades de un *middleware* para que en lugar de manipular diferentes sistemas de archivos y sistemas operativos, solo se aplique a una configuración con condiciones homogéneas.

Con el objetivo de visualizar grandes volúmenes de datos de manera rápida y eficiente, es fundamental explotar la potencia de procesamiento del clúster (entiéndase gran volumen de datos, como un conjunto que no puede ser procesado o visualizado de manera eficiente por un solo ordenador). Es por eso que dentro del módulo de visualización se propone implementar técnicas de visualización distribuida, visualización paralela y renderizado paralelo. Las principales características de estas técnicas fueron discutidas en el capítulo 3. Se debe señalar que esta es una actividad complicada y que requiere de un análisis profundo, aunque se puede contar con software de ayuda en la construcción de clústers para renderizado. Dentro de ellos se tienen a las herramientas Chromium, ParaView, VisIt, OpenDX, todas de código abierto y el HDF5 paralelo, el cual permite la visualización y procesamiento paralelo de archivos HDF5.

En cuanto a la forma de almacenamiento se presentan dos variantes fundamentales, donde se incluyen el tratamiento de los datos, los metadatos y los espacios personales de trabajo:

1 Utilizar BD solo para almacenar los catálogos de metadatos y los archivos en formato de datos científicos (HDF, CDF, NetCDF, FITS), almacenarlos en los discos duros del clúster.

2 Tener estructuras para almacenar objetos de formatos de datos científicos (HDF, CDF, NetCDF, FITS) en un clúster de BD (Bases de datos objeto relacional, bases de datos orientadas a objetos, etc.).

Según la primera variante, se tiene una base de datos para almacenar los catálogos de metadatos. Una vez que un usuario se autentifica, le permite ubicar los datos que están replicados en los distintitos discos del clúster. Con esta variante, la BD de catálogos de metadatos tiene el control de todos los datos del sistema; en caso de que algún dato sea movido de lugar, es necesario actualizar el catálogo.

El catálogo de metadatos es el encargado de controlar todos los grandes volúmenes de datos que están almacenados en los discos duros del clúster, así como los archivos o conjuntos de datos que posee cada usuario del sistema, aspecto que es de mucha utilidad para brindar los espacios de trabajos personales.

El modelo de centro de ciencia que se propone en el epígrafe 4.3 contempla la posibilidad de incluir estas dos variantes de almacenamiento. En el caso de estudio se presenta en el epígrafe 4.5 se muestra la implementación de la primera variante, y la instalación de las herramientas necesarias para ello.

Los metadatos descriptivos según la primera variante se encuentran dentro de cada archivo. Con esta variante se pueden utilizar las facilidades que brindan algunos de los formatos de datos científicos para el procesamiento paralelo, tratados en el capítulo 2, como el HDF paralelo, disponible a partir de la versión HDF5.

En la segunda variante, la idea es disponer de estructuras para almacenar objetos de formatos de datos científicos (CDF, NetCDF, FITS), en un clúster de base de datos, que puede ser objeto relacional u orientado a objetos. Para esto se puede hacer uso del gestor de BD que sea más conveniente y de un conjunto de herramientas para la distribución de los datos, esta variante se trata más detalladamente en (Enriquez, 2008)

En esta variante, el tratamiento de los catálogos de metadatos es similar a la primera: disponer de una BD de catálogos que puede estar replicada. El tratamiento de los metadatos descriptivos es algo más complicado, ya que no se pueden explotar las facilidades de los formatos de datos científicos (HDF, NetCDF, FITS) donde los metadatos descriptivos se encuentran dentro de los archivos. Una alternativa para resolver este problema, es tener un objeto en forma de tabla que contenga los metadatos descriptivos asociados a cada uno de los archivos, los cuales también están almacenados en forma de tabla.

Brindar espacios de trabajo personales o BD personales según esta variante de almacenamiento es un problema relativamente sencillo, el cual se resuelve creando para cada usuario una BD con estructura similar a las que almacenan los grandes conjuntos de datos. Esta base de datos constituye su BD personal, donde puede almacenar los datos de mayor interés para él.

Las BD personales de los clientes son específicas para su investigación y almacenan los datos mas utilizados por ellos. Una vez definida, se copian hacia allí fragmentos de los grandes conjuntos de datos. Esta base de datos de trabajo personal usualmente requiere sucesivas actualizaciones a partir de las fuentes de datos, o de nuevos conocimientos descubiertos por los usuarios. Los datos en la BD de trabajo personal se guardan en el formato que esté disponible o en otro específico definido por el usuario. Esta forma de trabajo con la BD personal no sólo ocurre con la segunda variante de almacenamiento, igualmente sucede con la primera de ellas.

En un clúster, al igual que en los centros que utilizan tecnología GRID, se implementan buenos mecanismos de control de fallos para evitar la pérdida de información y mantener la integridad de los datos. Estableciendo políticas de seguridad se evita cualquier problema que cause un usuario ajeno al sistema, como la introducción de códigos malignos, es por eso que, implementando un buen sistema de autentificación se tiene el control de los usuarios que acceden a los datos del CC. Utilizando la replicación de los datos, se mantiene la integridad de los mismos. Efectuar una suma de chequeo de cada conjunto de datos, es el mecanismo para determinar si ocurrió algún cambio o modificación que requiera la recuperación o actualización de los mismos. Entonces, de ser necesario, los conjuntos se recuperan copiándolos desde una fuente equivalente. Disponer de un mecanismo que sea

capaz de propagar las actualizaciones de los datos a todos sus conjuntos replicados, así como a las bases de datos personales es un aspecto crucial para un CC. Estas particulares son resueltas mediante el uso del solucionador de recursos de almacenamiento seleccionado en el capítulo 2.

El hecho de contar con un clúster de computadoras implica buena capacidad de procesamiento, pero para explotar al máximo la sinergia en el clúster se necesita una buena programación de paso de mensajes (principalmente MPI) para lograr algoritmos capaces de repartir las tareas entre varios procesadores y balancear la carga luego que los nodos estén disponibles cuando terminen de realizar la tarea asignada. Por otra parte el uso de las BD paralelas tratadas en (Enriquez, 2008) constituye un buen mecanismo para aumentar la capacidad de cómputo y almacenamiento en un centro de ciencia.

Una característica importante de los centros de ciencia es la colaboración entre varios de ellos. El clúster de computadoras no interviene directamente en esta tarea, por el momento no es objetivo de esta investigación desarrollar la colaboración con otros centros de ciencia, este aspecto es una tarea a resolver en investigaciones futuras.

4.3 Modelo de arquitectura para los servicios y aplicaciones del centro de ciencia

Se realizó un diseño lógico de centro de ciencia basado en cuatro capas: capa interfaz, capa de negocio, capa manejadora de datos y la capa de almacenamiento físico, como se muestra en la Figura 4.1, en el cual se tratan aspectos tales como: catálogo de metadatos, servicios, aplicaciones, bibliotecas de formatos de datos científicos, Sistemas de Gestión de Bases de Datos (SGBD), almacenamiento de grandes volúmenes de datos y espacios personales de trabajo.

En la parte superior se encuentra la capa interfaz, en un nivel más bajo la capa de negocio seguida de la capa manejadora de datos y por último la capa de almacenamiento físico.

Los usuarios del centro de datos, desde su estación de trabajo o computadora personal, se autentifican en el sistema mediante la interfaz, localizan los datos con que deseen trabajar en el catálogo de metadatos y acceden a los servicios o aplicaciones que brinda el centro en

la capa de negocio. Una vez que están dentro del sistema, las aplicaciones seleccionadas interactúan con los componentes de la capa manejadora de datos, como son las bibliotecas de formatos de datos científicos (HDF, NetCDF, FITS) y los Sistemas de Gestión de Bases de Datos (SGBD), ambos componentes son los encargados de acceder a los datos físicos (nivel más bajo).

Los SGBD permiten el acceso a los datos científicos que están almacenados en BD. Tanto las bibliotecas de formatos de datos científicos como los SGBD, son necesarios para acceder a los espacios de trabajo personales o BD personales como se muestra en la Figura 4.1.

Los componentes de la capa inferior (capa de almacenamiento de datos físicos) son: los formatos de almacenamiento de datos científicos donde se almacenan los grandes conjuntos de datos y los espacios personales que permiten almacenar fragmentos de los grandes conjuntos de datos.

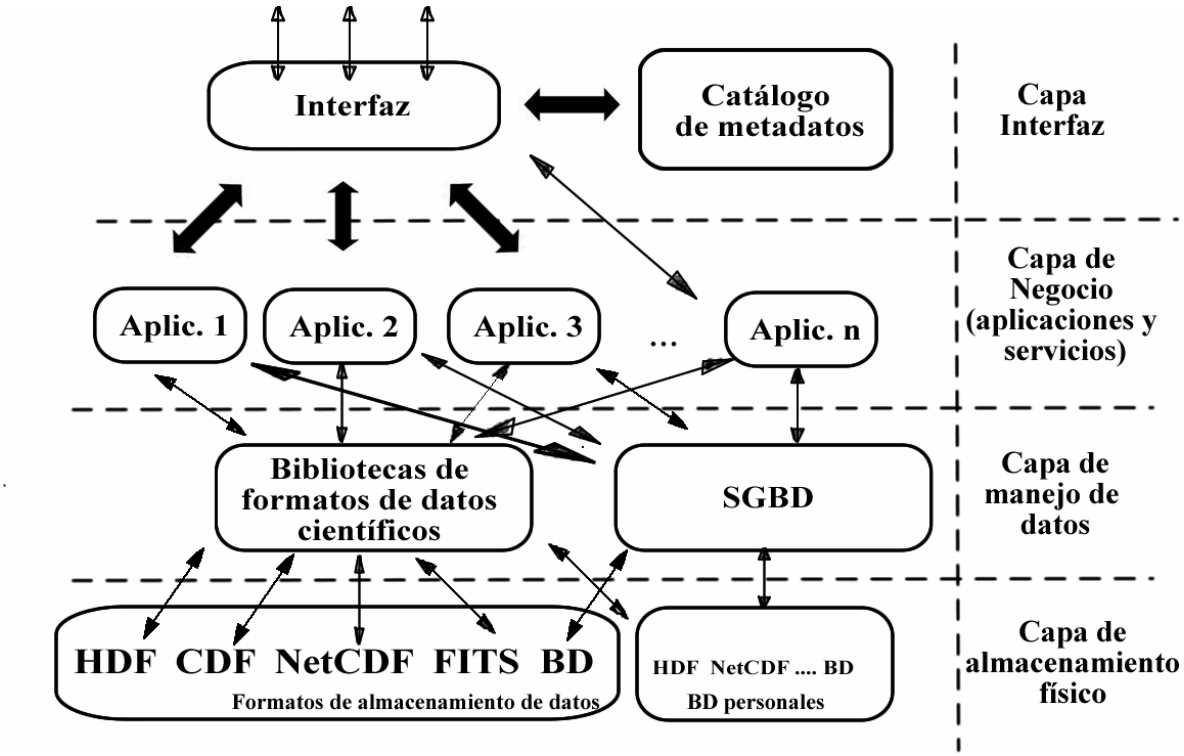

Figura 4.1 Modelo lógico del diseño del Centro de Ciencia

4.3.1 Capa Interfaz

La capa superior o capa interfaz, permite que múltiples usuarios se conecten y trabajen simultáneamente en el sistema, accediendo a través de un servicio que les permite autentificarse mediante una interfaz común para todos. Una vez que un usuario se autentifica puede ver los servicios que están disponibles en ese momento o que sus privilegios le permitan acceder.

En el sistema hay que manejar distintos grupos de usuarios: grupo de administradores, grupo de usuarios investigadores y grupo de usuarios temporales.

Los administradores del sistema tienen el control total de centro de ciencia, ellos pueden acceder directamente para publicar, mover, actualizar o eliminar datos. Una de sus tareas es la publicación de nuevos datos después de haberlos analizado minuciosamente y haber verificado que son reales y que pueden ser útiles para alguna investigación; ellos son los encargados de mover los datos en caso de que necesiten una mejor organización o para aprovechar el espacio de almacenamiento. Otra de sus tareas es la de actualizar datos que se obtienen por nuevos mecanismos, algoritmos más sofisticados o cualquier otra fuente confiable. Los datos que ya no sean de interés para los usuarios del sistema son eliminados por este grupo de usuarios. Todas estas formas de manipular los datos son invisibles para los demás grupos de usuarios del sistema.

Los usuarios investigadores son aquellos que necesitan acceder al centro para aprovechar las facilidades de los servicios que este brinda para resolver problemas de algún área científica. Ellos se registran al sistema para consultar los datos que ya están almacenados en el centro. Disponen de un espacio de trabajo personal, donde guardan los datos de mayor interés o que usan con más frecuencia. En su espacio guardan los resultados de su investigación y pueden hacerlos públicos para el resto de los usuarios. Un conjunto de investigadores que deseen colaborar en un trabajo o investigar en el mismo tema, disponen de la posibilidad de compartir un espacio de trabajo común, para de esta forma evitar la redundancia en los datos. Si estos usuarios necesitan publicar nuevos datos que no se encuentran en el sistema, tienen que ser autorizados debidamente por los administradores del centro, que son los encargados de examinar esos datos.

El último grupo, los usuarios temporales, son aquellos que necesitan utilizar el sistema ocasionalmente para realizar alguna búsqueda u obtener alguna imagen de un grupo de datos. Estos usuarios no disponen de espacios de trabajo personal, pues su principal objetivo es con fines informativos.

Otro elemento importante de este nivel es la BD de catálogo de metadatos. En ella se almacenan las direcciones de todos los recursos que están en el sistema, la localización de los datos, los espacios personales y otras facilidades que ayudan al cliente a encontrar tanto datos como recursos, para esto no es necesario ser un experto en el funcionamiento del sistema ni conocer los detalles de los datos. Cuando un usuario se autentifica, generalmente lo primero que realiza es consultar esta BD de catálogos. El caso de estudio tratado en esta investigación muestra en el epígrafe 4.5.1 el proceso de instalación del solucionador de recursos de almacenamiento iRODS, el cual incluye la instalación de la BD de catálogos de metadatos iCAT (iRODS *Metadata Catalog*), mostrada en la Figura 4.10.

Debe señalarse, que esta capa es de gran importancia para el sistema, puesto que es donde los usuarios interactúan con el centro mediante una interfaz. Las funcionalidades de los demás niveles generalmente son invisibles para los usuarios investigadores. Esta capa está enlazada directamente con la capa de negocio que se trata a continuación.

4.3.2 Capa de Negocio

El CC está diseñado para brindar diferentes servicios como son: autentificación, trabajo con metadatos y datos, procesamiento, visualización entre otros. Algunos de estos servicios pertenecen a la capa interfaz como son los relacionados con la autentificación y la búsqueda de metadatos en el catálogo. No obstante, la mayoría de las aplicaciones y servicios que brinda el centro de ciencia se encuentran en la capa de negocio.

Una vez que un usuario se autentifica, puede procesar o visualizar datos mediante el uso de estas aplicaciones, las cuales son las encargadas de acceder a los datos y hacer uso de las herramientas de la capa manejadora de datos. La idea de esta capa es disponer de un conjunto de servicios y herramientas configurables, para que el usuario prepare trabajos complicados y el centro se encargue de gestionar los datos, procesarlos y devolverle los resultados –en forma de imágenes, geometrías, o resúmenes de datos-, siendo todo este

proceso transparente para él, es decir, sin que tenga que conocer dónde están ubicados los datos, ni por cuántos procesos pasaron. Para lograr esto hay que hacer uso de herramientas, de visualización y renderizado paralelo, si se quiere obtener buenos tiempos de respuesta.

Para la preparación de los trabajos que el usuario desee realizar en el centro se hace necesario disponer de herramientas para la definición de flujos de trabajo. Este estilo de definición de tareas brinda muchas ventajas para los usuarios ya que solamente se tienen que concentrar en los resultados de su área de investigación y no tienen que preocuparse por programar sus propias aplicaciones.

En el epígrafe 4.4 se muestran un conjunto de aplicaciones, principalmente de visualización que sirven para ejemplificar la actividad de esta capa.

El centro ofrece además, la posibilidad de realizar consultas complicadas a los datos mediante el uso de un clúster de BD y de la replicación. Este trabajo puede ser muy eficiente y reducir considerablemente el tiempo con respecto a un servidor de base de datos tradicional, principalmente por las ventajas que ofrecen las bases de datos paralelas.

4.3.3 Capa Manejadora de Datos

En la Figura 4.1 se observa la capa manejadora de datos, que brinda las herramientas necesarias para manipular los distintos tipos de datos. Esta capa consta de dos partes bien definidas: una donde están las bibliotecas para el manejo de los distintos formatos de datos (HDF, NetCDF….), y la otra parte formada por los Sistemas de Gestión de Bases de Datos (SGBD). Ambos recursos permiten el trabajo en paralelo y el acceso al repositorio de datos.

En la Figura 4.2 se muestra detalladamente el modelo lógico de trabajo con los formatos de datos en paralelo. Se observa cómo las aplicaciones de los usuarios -paralelas o no- son tratadas por el sistema. En este caso se refiere a aplicaciones paralelas, ya que las aplicaciones secuenciales se realizan de manera similar y el principal objetivo de los centros de ciencia es explotar la fuerza de múltiples computadoras trabajando simultáneamente en una misma tarea.

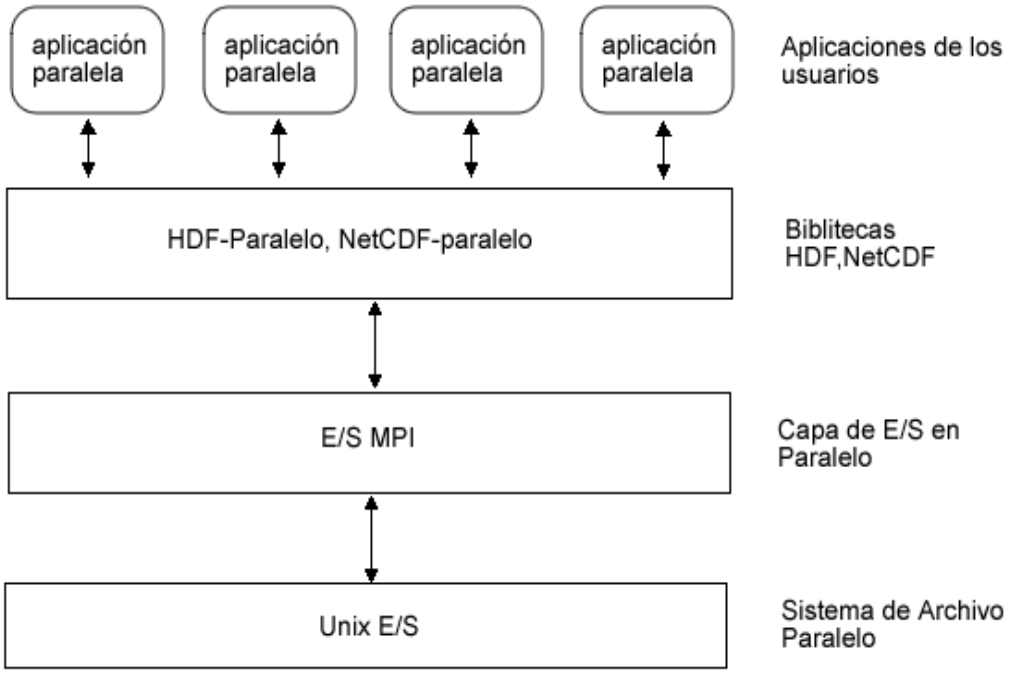

Figura 4.2 Capa de implementación de trabajo con formatos de datos en paralelo

Las bibliotecas para el manejo de los datos (HDF, NetCDF….), soportan el trabajo en paralelo con los distintos archivos almacenados en el centro de ciencia. En el nivel inferior se tienen las funciones de paso de mensajes para la entrada/salida en paralelo (E/S MPI) para poder distribuir las tareas entre los nodos de procesamiento del clúster y establecer el orden en que van a ser procesadas. En la parte inferior se encuentra el sistema de archivo paralelo, en este caso solo se muestra entrada/salida para sistemas de archivos basados en UNIX aunque para Windows el tratamiento es similar.

Estas bibliotecas para el trabajo con los distintos formatos de datos científicos permiten realizar las siguientes operaciones:

- Crear, abrir y cerrar un archivo.
- Crear, abrir y cerrar un conjunto de datos.
- Extender o reducir un conjunto de datos.
- Escribir o leer desde un conjunto de datos.

El trabajo con los datos puede ser de forma colectiva o individual. Una vez que un archivo es abierto por los procesos de comunicación:

- Todas las partes del archivo son accesibles por todos los procesos.
- Todos los objetos del archivo son accesibles por todos los procesos.
- Múltiples procesos pueden escribir en el mismo conjunto de datos.
- Cada proceso puede escribir en un conjunto de datos individual.

La comunicación mediante paso de mensajes MPI, permite la comunicación de un grupo de procesos entre sí, esta tiene que realizarse en el orden correcto. Inicialmente se abre un archivo con un comunicador, después se obtiene el manipulador del archivo que va a ser usado para los accesos posteriores.

El sistema de archivo paralelo es un sistema de ficheros para clúster que proporciona alto rendimiento y acceso a la información por múltiples computadoras simultáneamente. Los conjuntos de datos están distribuidos en varios discos, mediante este se pueden realizar lecturas y escrituras de los datos en paralelo. Estos sistemas de archivos para clústers, proporcionan acceso concurrente de alta velocidad a aplicaciones, ejecutándose en múltiples nodos del clúster. Suministra un sistema de almacenamiento de ficheros y posee herramientas para la gestión y la administración de clústers, además, permite accesos compartidos al sistema de ficheros desde otros sistemas de archivos paralelos remotos. Estos sistemas ofrecen otras funcionalidades, como alta disponibilidad, el soporte para clústers heterogéneos, recuperación en caso de fallos y seguridad.

En éste mismo nivel del modelo general mostrado en la Figura 4.1 se encuentran los SGBD, su propósito es el mismo de las bibliotecas para el trabajo con los formatos de datos científicos. Estos SGBD tienen las herramientas necesarias para acceder y consultar los datos que se encuentran almacenados en las BD del nivel inferior.

Con los SGBD el trabajo es mucho más sencillo, porque estos son más robustos que las bibliotecas de formatos de datos y la mayoría de ellos incorporan una gran cantidad de

tareas que realizan sin involucrar al usuario. Este sólo tiene que concentrarse en aprender el lenguaje de comunicación para la formulación de las consultas.

Como se ha visto esta capa, a pesar de ser invisible para el usuario, juega un papel fundamental en la interacción entre el cliente y los datos.

Se debe señalar que existen muchas herramientas disponibles de código abierto, las cuales se pueden utilizar o modificar para facilitar el trabajo de esta capa. Las bibliotecas de formatos de datos son más complicadas y requieren mayor conocimiento, pero permiten explotar un alto grado de paralelismo en las tareas, mientras que los SGBD son más robustos y por ende mucho más fáciles de usar. Ambos sistemas manejadores de datos se encuentran en pleno desarrollo.

Los solucionadores de recursos de almacenamiento SRB e iRODS tratados en el capítulo 2, controlan gran parte del funcionamiento interno del sistema: el trabajo con los archivos, la autentificación y el manejo de los diferentes tipos de metadatos.

Ambos sistemas poseen un módulo para incorporar el formato de datos científicos HDF con el solucionador, cuyos nombres son HDF-SRB y HDF-iRODS, respectivamente. En este trabajo se presenta HDF-iRODS, por ser este el solucionador seleccionado para la implementación del caso de estudio. En el epígrafe 4.5.2 se muestra como se realizó la instalación de este módulo.

Como se pudo observar, la capa manejadora de datos discutida en esta sección cuenta con 2 componentes, los SGBD y las bibliotecas de datos científicos. El caso de estudio tratado en esta tesis se centra en las bibliotecas de formatos de datos científicos, específicamente HDF con su versión HDF paralelo, para lo cual se muestra en el epígrafe 4.5.3 todo el proceso realizado para la instalación de las herramientas necesarias.

4.3.4 Capa de Almacenamiento

La capa de almacenamiento es la encargada de soportar los formatos de datos físicos y las bases de datos en sí. Como se puede observar en la Figura 4.1, esta capa esta dividida en 2 secciones: la de la izquierda significa que se tienen formatos de datos científicos o BD para

almacenar los grandes conjuntos de datos. La sección de la derecha se refiere a las BD personales que en principio poseen las mismas funcionalidades para almacenar los grandes conjuntos de datos, tanto en formatos de datos científicos como en BD; pero siendo esto en menor escala.

Los datos se almacenan en los discos duros de los nodos del clúster. En caso que estén en formatos de datos científicos, el iRODS juega también un papel importante pues en la BD iCAT se almacenan las direcciones de cada uno de los conjuntos de datos o archivos distribuidos por el clúster. El solucionador de recursos de almacenamiento se encarga también de mantener el control de los espacios personales de trabajo, permitiendo el acceso a los conjuntos particulares de los usuarios según sus privilegios. Otras de las tareas realizadas por este es la replicación de los datos, ya sea de forma automática o asistida por los usuarios administradores, y así lograr la disponibilidad del sistema en caso de que falle un nodo del clúster.

Las principales características que hay que tener en cuenta en esta capa son: gran capacidad de almacenamiento, discos duros rápidos para agilizar la comunicación entre los nodos y escalabilidad para aumentar la capacidad de almacenamiento.

4.4 Módulos de visualización

Con el surgimiento de los centros de ciencia se presentan nuevas oportunidades para el desarrollo de novedosos modelos de visualización. Algunos sistemas comerciales de visualización emplean ambientes modulares para el desarrollo de aplicaciones, compuestas generalmente, por la unión de diversos módulos en una red, empleando un paradigma de programación visual. Entre los sistemas libres de este tipo se destacan notablemente Khoros y OpenDX.

4.4.1 Khoros

Khoros es un sistema de desarrollo de software para el tratamiento y visualización de imágenes. Es un software abierto; pero no de dominio público. Esto quiere decir que se distribuye de forma gratuita, y que se puede utilizar gratis igualmente, pero si se quiere desarrollar software comercial a través de él, será necesario comprar una licencia. Esta

característica hace de este sistema una plataforma idónea para tratamiento científico en ambientes universitarios.

Khoros posee una herramienta de programación visual muy potente que hace uso de un lenguaje gráfico orientado a flujo de datos (ver Figura 4.3). Los módulos están agrupados en cajas de herramientas (*toolbox* en inglés), las cuales pueden instalarse a manera de *plug-in*, y extenderse por los propios usuarios.

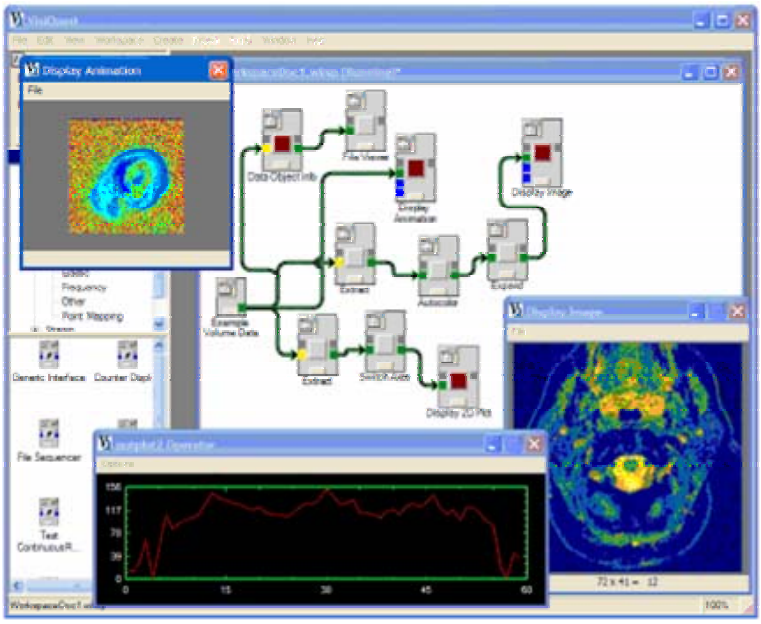

Figura 4.3 El sistema de visualización Khoros

4.4.2 OpenDX

OpenDX es una aplicación y un paquete de desarrollo de software de código abierto para la visualización de datos científicos, especialmente los adquiridos a partir de observaciones y simulaciones. OpenDX apareció en mayo de 1999 cuando IBM convirtió su sistema comercial de visualización, IBM *Visualization Data Explorer* (conocido como “DX”), en un software de código abierto. En la actualidad, científicos, ingenieros y analistas de negocios de todo el mundo, utilizan OpenDX para generar imágenes y animaciones a partir de sus investigaciones.

Después de la liberación del código, OpenDX se ha portado a distintos tipos de plataformas Linux y FreeBSD, además de ser compilado para los sistemas UNIX comerciales donde DX siempre corrió, tales como SGI, HP, Sun y Aix. De igual manera, existe una versión comercial para plataformas Windows, aunque requiere la instalación adicional de software especial.

OpenDX es un sistema de visualización de propósito general, que permite leer datos de distintas fuentes de una manera flexible y amigable. Aún cuando la organización de los datos puede ser muy diferente para cada fuente, OpenDX ofrece una manera de leer casi cualquier tipo de datos a través de la herramienta conocida como *Data Prompter*.

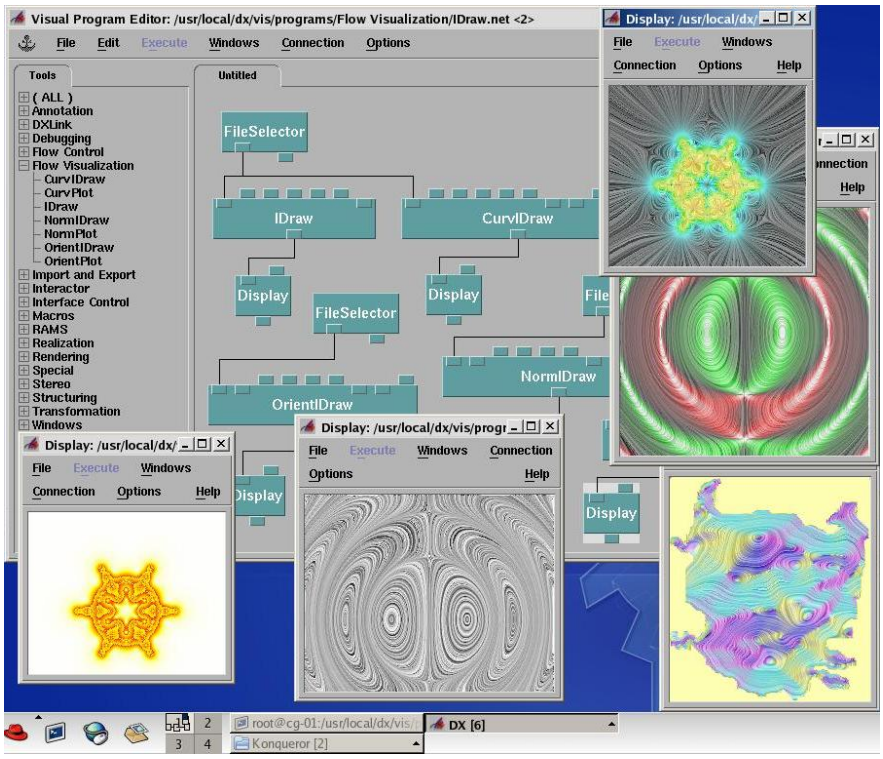

Figura 4.4 El sistema de visualización OpenDX

OpenDX ofrece una interfaz gráfica que permite crear programas de manera visual, mediante la interconexión de bloques o módulos, llamados redes (ver Figura 4.4). Por otro lado, OpenDX contiene un lenguaje *script* para crear programas de visualización. Un programa visual es convertido a este lenguaje antes de ser guardado en disco.

Entre las características destacables de OpenDX se encuentran:

- Interactividad: El software puede ser aprendido y "programado" por el mismo científico, eliminando la necesidad de agentes intermedios en el proceso de manipulación de los datos y obtención de conclusiones. Además, le permite al investigador crear programas visuales, con una interfaz de usuario amigable y sencilla, que permite modificar interactivamente las imágenes cambiando los valores de las entradas mediante controles gráficos.

- Compartible: Los programas creados por un científico pueden ser compartidos y manipulados por científicos de cualquier lugar del mundo, lo que permite el trabajo colaborativo.

- Multidimensional: Genera objetos visuales multidimensionales a partir de datos numéricos multiparamétricos.

- Temporal: El programa brinda un nivel de "animación" o movimiento. Esto permite una percepción directa de procesos temporales. Con frecuencia, el tiempo es una de las dimensiones más importantes de los procesos científicos e ingenieriles.

- Modular: Bloques de software genéricos, llamados módulos y macros, pueden ser "ensamblados" de varias maneras según la clase de dato a analizar. Una vez construidos, estos "ensamblajes", llamados redes o programas visuales, pueden ser usados para visualizar diferentes instancias de conjuntos de datos similares.

OpenDX se puede utilizar por tres tipos de usuarios:

- El usuario final, que usa un programa desarrollado por otro.

- El programador visual o de "red". Esta es la forma más común de usar OpenDX.

- El desarrollador de módulos, que desarrolla y adiciona funciones al OpenDX, o que usa las bibliotecas de DX para extender otros programas ya hechos.

Debido a la generalidad de su modelo de datos y a la flexibilidad de la programación visual, OpenDX no está optimizado para alguna rama de la ciencia en particular. Por lo tanto, mientras que es posible utilizar efectivamente OpenDX para visualizar la salida de simulaciones, mediciones, experimentos o cualquier dato generado por computadora, no tiene funcionalidades optimizadas para un área de estudio particular. Por ejemplo, no es posible realizar directamente operaciones visuales de dinámica de fluidos, aunque se pueden adicionar estas operaciones a través de la construcción de módulos especializados.

4.4.3 Arquitectura de un módulo de visualización para un Centro de Ciencia basado en un clúster de computadoras

Los nuevos modelos de visualización creados consideran diferentes escenarios para distintos tipos de clientes –Web o *Desktop*- con mayor o menor capacidad de cómputo. En cualquier caso, los datos a visualizar se concentran en el centro de ciencia.

La principal razón de utilizar un clúster de computadoras para la visualización científica es explotar al máximo las capacidades de procesamiento que estos poseen. Existen varios tipos de interacciones que se presentan para la visualización de datos científicos. Una de ellas es la interacción semántica, tratada en (Morell and Pérez, 2006) mediante la realización de un módulo de OpenDX para la visualización de fluidos bidimensionales. Esta se realiza, cuando a partir de la inteacción del usuario con la imagen que se le persenta, es posible acceder y modificar los datos de aplicación originales. Otros tipos de interacción, como la interacción de configuración y parametrización, también son posibles en este contexto (Frühauf, 1997).

Como se describió en epígrafes anteriores, algunos de los sistemas de visualización permiten definir una tubería de visualización (o diagramas de flujo de trabajo) por parte del cliente. Este es un aspecto deseable para el modelo propuesto, es decir, que el CC incorpore herramientas de este tipo dentro de sus recursos, para de esta forma aprovechar las facilidades que brinda la visualización distribuida, discutidas en el epígrafe 3.2.

Cuando se trabaja con grandes volúmenes de datos, no es raro para algoritmos de visualización producir gran cantidad de geometrías y gráficos, lo cual puede sobrepasar la capacidad de un solo CPU y de un solo acelerador gráfico, para renderizar estas imágenes.

El uso de los métodos de renderizado paralelo de datos es requerido para lograr un funcionamiento adecuado. La clasificación más usada para los algoritmos de renderizado paralelo se basa en las localizaciones de la tubería de renderizado, así como el espacio que ocupa el objeto en la pantalla. Estos métodos son: *sort-first*, *sort-middle*, y *sort-last*, los cuales fueron tratados en el epígrafe 3.5 y se detallan en el capítulo 27 y 28 de (Hansen and Johnson, 2005).

Para abordar la estructura del módulo de visualización de forma óptima utilizando las ventajas de un clúster de computadoras, es necesario utilizar algunas de las herramientas de renderizado en paralelo como Chromium, Paraview, VisIt y OpenDx.

Chromium es una biblioteca libre y de código abierto que permite desarrollar aplicaciones de renderizado en paralelo en clústers de computadoras, posee una arquitectura completamente extensible. Soporta renderizado en paralelo de tipo *sort-first*, *sort-last* y un híbrido entre ambos. Esta biblioteca coordina renderizado con OpenGL en clústers de computadoras.

ParaView es una herramienta de código abierto y multiplataforma, diseñado para visualizar conjuntos de datos de tamaño variable en un clúster de computadoras. Está montado sobre VTK (*Visualization Toolkit*) la cual es una herramienta de visualización de bajo nivel.

VisIt es otra herramienta de código abierto y multiplataforma para renderizado en paralelo. Utiliza renderizado de tipo *sort-first* y está diseñado para visualizar datos de grandes simulaciones.

OpenDX fue tratado en el epígrafe anterior, sin embargo existe una extensión de este sistema de visualización que permite renderizar en paralelo en un clúster de computadoras utilizando *sort-first*.

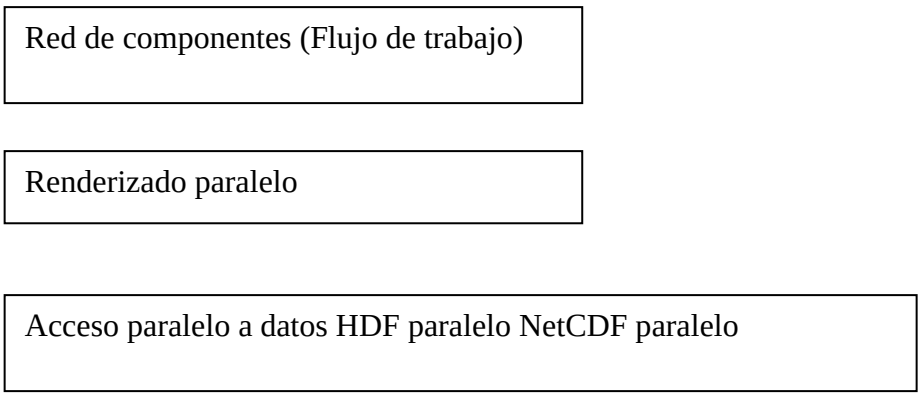

Figura 4.5 Diagrama de Visualización

La figura anterior muestra los componentes más importantes que intervienen en el módulo de visualización, de abajo hacia arriba la primera capa: Acceso paralelo a datos HDF, NetCDF, fueron analizados en el modelo lógico del diseño del centro de ciencia.

En la capa intermedia denominada “Renderizado paralelo”, intervienen las herramientas Chromium, VisIt, ParaView y OpenDX, utilizadas con el objetivo de aprovechar la capacidad del clúster para renderizar en paralelo.

La última capa es la que está más cerca de los usuarios. La llamada red de componentes (Flujo de trabajo), es la capa que sirve de interfaz entre los usuarios y el centro de ciencia. Los usuarios interactúan con el CC formulando complejas redes o flujos de trabajos y especificando las tareas que desean realizar en el clúster. OpenDX interviene en las 2 capas superiores del diagrama anterior, debido a que permite el renderizado paralelo y posee un conjunto de funciones para la creación de flujos de trabajo, siendo la principal herramienta a tener en cuenta en el diseño. Como es un paquete de código abierto, es posible modificarlo y adaptarlo para satisfacer los requerimientos del servicio de visualización.

La herramienta HDFVIEW permite la búsqueda y visualización de archivos HDF, aunque no permite el renderizado paralelo. Esta herramienta se utiliza para visualizar archivos HDF almacenados en el CC y en las estaciones locales. HDFVIEW permite acceso a datos HDF en un SRB a través del módulo HDF-SRB. Actualmente está siendo modificado para acceder a datos HDF de un servidor iRODS a través del módulo HDF-iRODS.

Para las tareas de visualización se propone instalar del lado de servidor el paquete de visualización OpenDX (o uno con posibilidades similares), y en el lado del cliente una interfaz Web o de escritorio, que permita al usuario explotar de forma remota las facilidades que proporciona este paquete para el renderizado en paralelo. También se pueden instalar en las estaciones de trabajo aplicaciones como HDFVIEW, que permiten la visualización y análisis de datos que se encuentran en el centro o en las estaciones locales de los usuarios.

4.5 Soluciones tecnológicas que satisfacen los requerimientos del CC en un clúster de computadoras

Este diseño cuenta con los requisitos de: alto funcionamiento de E/S, almacenamiento de grandes volúmenes de datos, acceso eficiente, análisis y visualización de datos. Se valoraron dos alternativas que sirven para dar solución a este problema. Almacenar los datos en archivos de formatos de datos científicos y la otra es utilizar BD para almacenar los datos científicos. Para dar solución a este problema se decidió combinar las buenas características de ambos sistemas, formatos de datos científicos y bases de datos.

Se decidió implementar un caso de estudio para almacenar los datos en formatos de datos científicos, específicamente HDF y una BD en postgreSQL para almacenar los catálogos de metadatos. También utilizar HDF paralelo para el procesamiento de los archivos HDF, de esta forma se aprovechan las buenas características que brinda la programación por paso de mensajes MPI en el procesamiento y la entrada/salida.

El sistema permite al usuario especificar nombres y otros atributos que se asocien con los conjuntos de datos. Internamente selecciona un nombre de archivo que corresponde con un conjunto de datos almacenado en el clúster. El mapeo que se establece entre los datos y los nombres de los archivos se almacena en la BD de catálogos. El usuario puede recuperar un conjunto de datos especificando atributos de los datos deseados.

Como solucionador de recursos de almacenamiento (*middleware*), se propone utilizar iRODS, escogido por las facilidades que brinda para el manejo de datos, por ser de código abierto y se ajusta perfectamente a las necesidades del diseño. El iRODS instala una BD de catálogos de metadatos llamada iCAT, que se utiliza para almacenar los catálogos en un

servidor postgreSQL. En el siguiente epígrafe se explica como instalar el servidor iRODS y la base de datos de catálogos de metadatos.

Una descripción de cómo instalar el módulo HDF-iRODS, que posee el servidor iRODS, se proporciona en el epígrafe 4.5.2.

Por otra parte debido a la potencia del trabajo en paralelo que permite la versión HDF5 paralelo se propone la instalación de esta herramienta y se brinda una descripción de cómo realizar esta tarea (epígrafe 4.5.3).

4.5.1 Instalación de iRODS

Para instalar iRODS, primero se tiene que descargar de http://irods.sdsc.edu

la licencia BSD del software y el acuerdo de registro. Luego descargue los archivos .tar de instalación, como es el caso del *script* de instalación para (Linux, Solaris, Mac OSX), la instalación de PostgreSQL y ODBC (por las siglas en inglés de *Open Database Connectivity*). Posteriormente se prosigue con la instalación de PostgreSQL, ODBC e iRODS en ese orden.

Para realizar la instalación del servidor iRODS, es necesario descompactar el archivo irods1.1.tar mediante el comando “tar xf irods1.1.tar”. Una vez descompactado, entre a la carpeta creada y ejecuta “./irodssetup”, este setup de instalación mostrará la información de la Figura 4.6 donde se deben escoger las opciones deseadas:

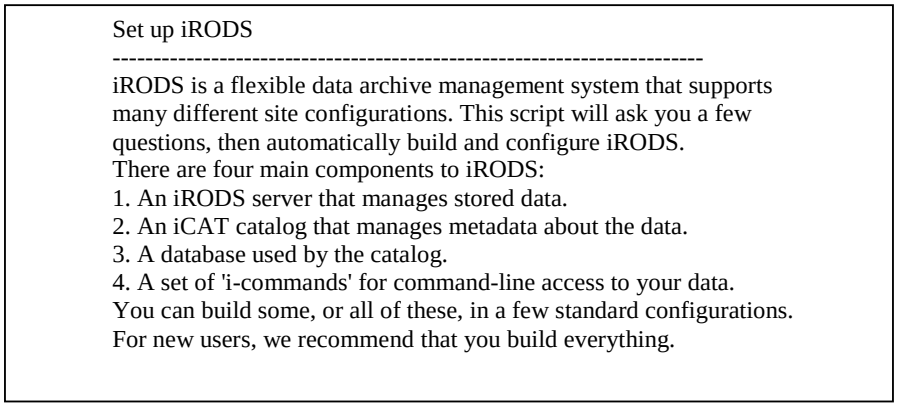

Figura 4.6 Setup de instalación de iRODS

Se recomienda instalar el servidor iRODS completamente, para ello escoja la opción 1. Luego, el programa de instalación le va preguntado si desea o no realizar alguna de las demás operaciones. Incluya la base de datos de catálogos de metadatos iCAT, diga que sí a todo hasta este paso. Para garantizar la seguridad del servidor y por ende de los datos que se administrarán en el clúster, el proceso de instalación crea una cuenta de administración de iRODS a la cual se le debe poner una contraseña.

Posteriormente, el proceso de instalación le preguntará si desea instalar el PostgreSQL o no. Diga que sí, pero si ya tiene otro servidor de postrgreSQL instalado indique el camino donde desea instalar esta otra versión. Por razones de seguridad se crea una cuenta de administración para el servidor postgreSQL.

Luego del paso anterior se le muestra al usuario las opciones seleccionadas por él, para confirmar si desea proceder o no. En caso de escoger sí, se muestra el proceso de instalación como aparece en la Figura 4.7.

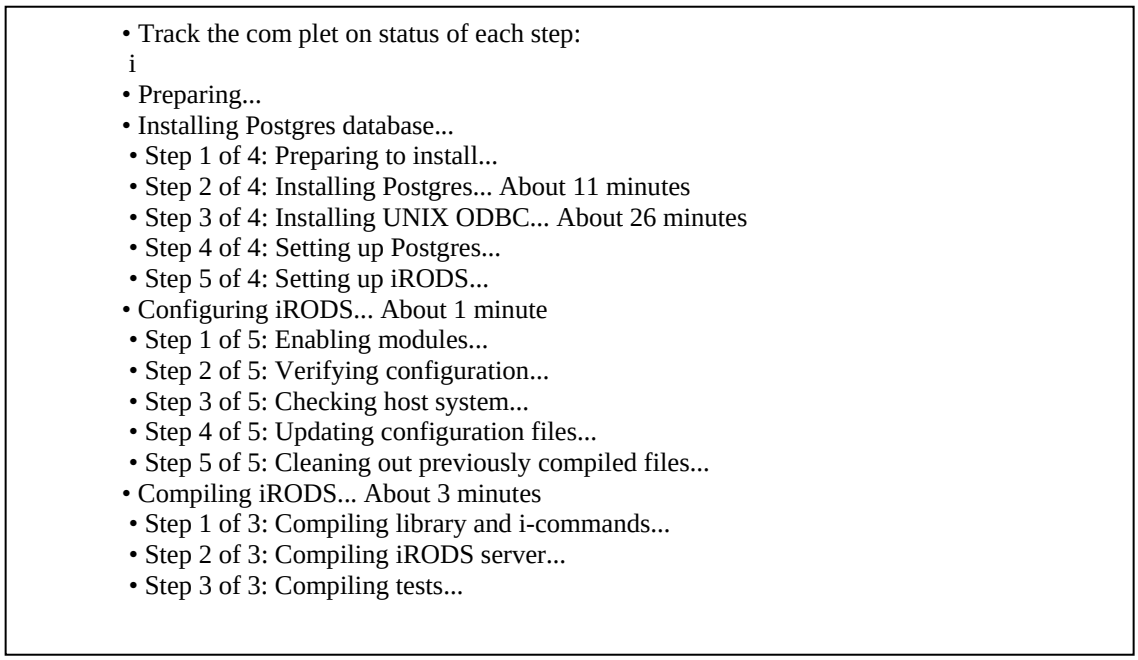

Figura 4.7 Proceso de instalación

De esta forma se puede instalar el servidor iRODS con su base de datos de catálogos iCAT. En caso de fallar esta instalación por alguna razón, entonces se debe proceder a editar los *scripts* de instalación localizados en la carpeta “scripts” de iRODS.

Una vez que el iRODS esté instalado vea el archivo finishSetup.log dentro de la carpeta installLogs, el cual muestra el proceso completo de instalación, así como los parámetros de configuración seleccionados.

Para iniciar, detener o reiniciar el servidor iRODS utilice los comandos:

• irodsctl start

• irodsctl stop

• irodsctl restart

El siguiente paso es acceder al servidor iRODS, para esto utilice cualquiera de los clientes de iRODS que se muestran a continuación:

• Cliente Web iRODS rich

https://rt.sdsc.edu:8443/irods/index.php

• Interprete de comandos de UNIX

iRODS/clients/icommands/bin

• El sistema de archivo a nivel de usuarios FUSE (*Filesystem in Userspace*)

iRODS/clients/fuse/bin/irodsFs fmount

• La biblioteca de entrada salida Jargon, desarrollada en Java

iRODS/java/jargon

• El buscador Web de PHP y la biblioteca cliente de PHP

• http://irods.sdsc.edu

• Las llamadas a la biblioteca de C

• El sistema de archivos a nivel de usuario Parrot

En esta sección se explica como instalar el buscador Web de PHP. Para instalar este cliente se necesitó instalar un servidor apache con PHP y copiar los archivos del cliente Web extrods.

En la siguiente figura se muestra el formulario de autentificación utilizado por este cliente Web para permitir el acceso al iRODS, para esto se especifica el nombre del servidor, el puerto 1247, el nombre de usuario y la contraseña.

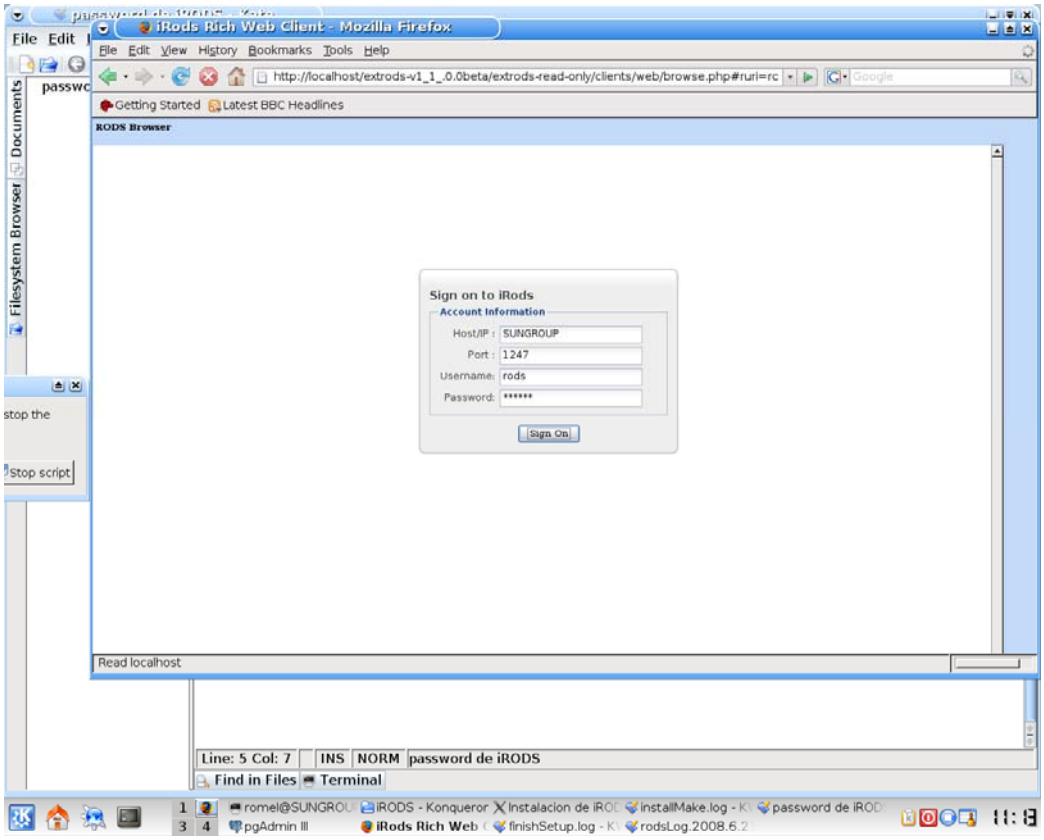

Figura 4.8 Formulario de autentificación del servidor iRODS utilizando el cliente Web extrods

En la Figura 4.9 se muestra una pantalla del cliente Web extrods cuando se navega por el servidor iRODS, a través de este se pueden crear nuevos archivos y colecciones, subir al servidor archivos o directorios completos, editar metadatos y hacer réplicas de los documentos y colecciones que se encuentren en el servidor. Observe que el archivo que está seleccionado dentro del servidor es un archivo HDF5, el cual fue subido al servidor por

el administrador. Esta es la forma que tienen los administradores del centro para crear las colecciones de grandes volúmenes de datos científicos.

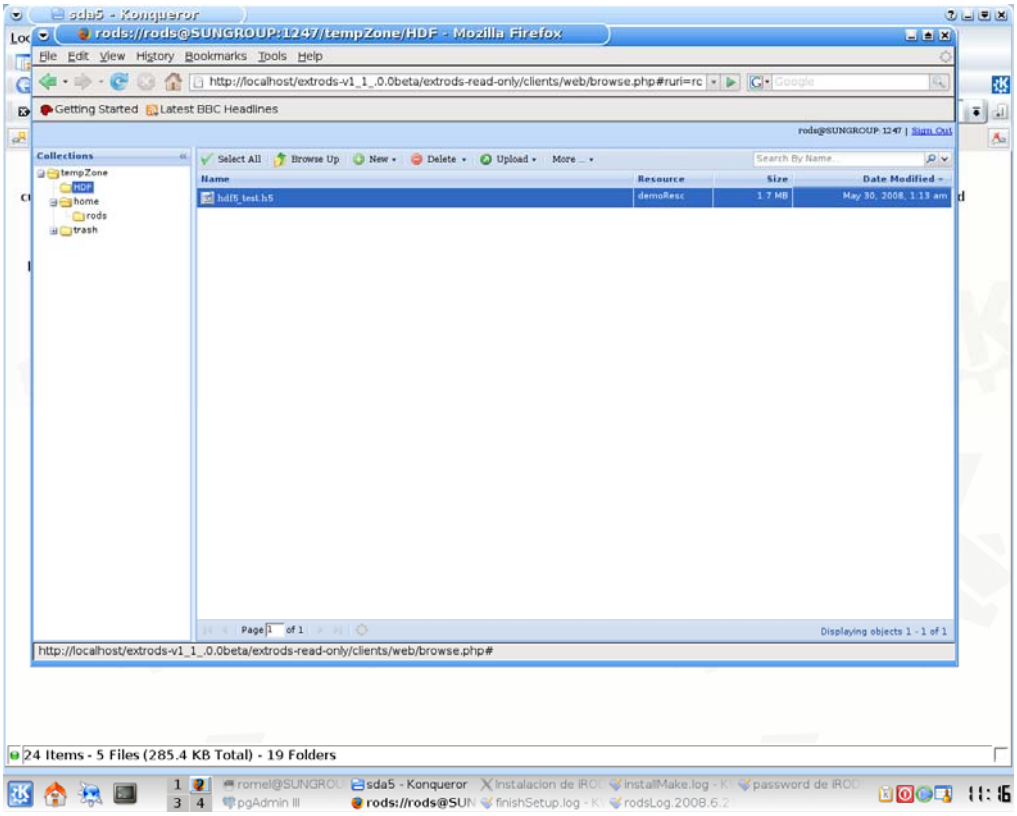

Figura 4.9 Uso del cliente Web extrods para navegar por el servidor iRODS

En la Figura 4.10 se muestra el cliente pgAdmin III de postgreSQL con las 23 tablas de la base de datos iCAT de iRODS, donde se almacenan las colecciones con las ubicaciones de los datos en los nodos del clúster, así como todo lo referente a los usuarios y los espacios personales de trabajo.

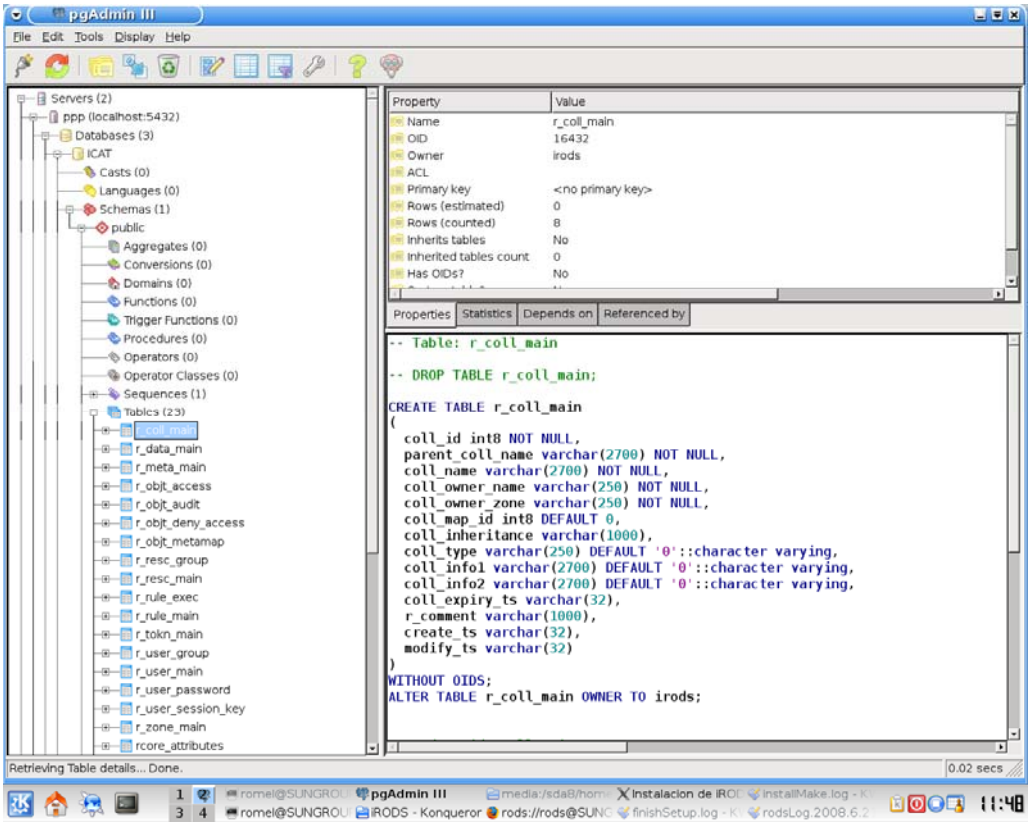

Figura 4.10 Base de datos de catálogos iCAT

4.5.2 Instalación del modulo HDF-iRODS

En este epígrafe se explican una serie de pasos que se deben seguir para instalar el módulo HDF-iRODS en el servidor iRODS, es importante destacar que este módulo está disponible a partir de la versión 1.1 de iRODS. HDF-iRODS está diseñado para soportar acceso interactivo a archivos HDF5 que estén almacenados en el servidor. Este módulo por defecto está deshabilitado por lo cual para instalarlo es necesario seguir los siguientes pasos:

1. Descargar e instalar los paquetes szip, zlib y la biblioteca HDF5, en el sitio Web http://www.hdfgroup.org/HDF5/release/obtain5.html se pueden obtener los códigos fuentes de estas aplicaciones.

   Esta versión es compatible con HDF5 1.6.x. Si se está utilizando HDF5 1.8.x. ejecute el comando "./configure ----with-default-api-version=v16" para que sea compatible con la versión 1.6.x.

2. Edite el archivo 'Makefile' dentro del módulo y ponga las siguientes variables: hdf5Dir, szlibDir y zlibDir con los caminos de donde están instalados el HDF5, szip y zlib.

   Note que hdf5Dir es el directorio de instalación del HDF5.

3. Editar la configuración del archivo config/config.mk y agregar HDF5 a la línea que define los módulos.

4. Recompile el iRODS nuevamente siguiendo los pasos mencionados en el epígrafe anterior.

5. Reinicie el servidor iRODS ejecutando 'irodsctl restart'.

Con este módulo queda instalado en el servidor iRODS, un conjunto de microservicios en forma de API "*iRODS-HDF5 client API*" que puede ser utilizada para programar aplicaciones cliente para el modulo HDF-iRODS. Estos microservicios soportan operaciones de tres tipos de objetos de HDF, como son:

H5OBJECT_FILE – archivos HDF5

H5OBJECT_GROUP – grupos HDF5

H5OBJECT_DATASET – conjuntos de datos HDF5

Entre las operaciones que permiten están:

H5FILE_OP_OPEN – abrir un archivo HDF5

H5FILE_OP_CLOSE – cerrar un archivo HDF5

H5GROUP_OP_READ_ATTRIBUTE – leer atributos de grupos HDF5

H5DATASET_OP_READ_ATTRIBUTE - leer atributos de conjuntos de datos HDF5.

Observe que a través de esta herramienta sólo se pueden realizar accesos de lectura, por lo que para lograr los accesos de escritura se deben utilizar otras herramientas, como por ejemplo las que posee el paquete HDF.

La implementación de clientes para este módulo de iRODS también incluye una interfaz para JNI (*Java Native Interface*), la cual permite el uso del buscador HDF5VIEW realizado en Java para acceder a los archivos HDF5 almacenados en el servidor iRODS.

4.5.3 Instalación de HDF5 paralelo

El paquete HDF5 se encuentra de forma binaria en todas las distribuciones de Linux, por lo que se puede instalar fácilmente mediante un administrador de paquetes. Sin embargo, para instalar HDF5 paralelo es necesario recompilar el código fuente.

La idea es tener instalado en cada una de las máquinas del clúster el HDF5 paralelo para acceder a estos archivos de forma colaborativa entre varios procesadores o para procesarlos paralelamente. Una vez que esté instalado HDF5 paralelo en el clúster entonces se puede hacer uso de las funciones que esta brinda para desarrollar aplicaciones paralelas. El acceso a los archivos HDF5 almacenados en el clúster se debe gestionar mediante peticiones al solucionador de recursos de almacenamiento iRODS con el módulo HDF-iRODS instalado.

Para instalar HDF5 paralelo es necesario instalar mpich2 como plataforma de prueba. HDF5 paralelo soporta MPI-IO colectiva para la selección irregular de archivos HDF5.

Existen un conjunto de requerimientos que se deben cumplir para instalar esta versión paralela. HDF5 paralelo requiere un compilador de MPI con soporte para MPI-IO y sistemas de archivos paralelos.

Los programas que trabajan con HDF5 paralelo pueden utilizar varios compiladores, entre los que se encuentran: mpicc, hcc, mpcc, mpcc_r. Para construir el HDF5 paralelo debe ponerse en la variable “CC” el camino donde está instalado el compilador y ejecutar “./configure”. En la siguiente línea se observa cómo se realiza la configuración para construir el HDF5 paralelo utilizando el compilador mpicc y habilitando el paralelismo.

```
CC=/usr/local/mpi/bin/mpicc ./configure --enable-parallel
```

La configuración del HDF5 paralelo se puede realizar en un solo procesador o en varios a la vez. Después que ya esté configurado, se debe ejecutar el comando “make”, seguido de “make check” y “make install”.

Las aplicaciones que se desarrollan utilizando HDF5 paralelo se compilan mediante la herramienta mpicc como se muestra en el ejemplo siguiente y se ejecutan con el comando “mpirun” especificando la cantidad de procesadores que intervienen en la ejecución del programa.

```
% mpicc Sample_mpio.c -o c.out
% mpirun -np 4 c.out
```

4.6 Conclusiones parciales

En este capítulo se desarrolló un modelo general de arquitectura de un centro de ciencia para la visualización científica basado en cluster de computadoras. Se definió un conjunto de requisitos indispensables para el diseño, y se determinaron los aspectos del diseño de un centro de ciencia para la VC que pueden ser soportados sobre la tecnología de clúster de computadoras.

Se analizaron dos variantes para el almacenamiento de los datos en el centro de ciencia y se implementó un caso de estudio que emplea la primera variante.

Se propone un conjunto de soluciones tecnológicas que satisfacen los requerimientos del diseño propuesto, y se presentan guías para la implementación real de un CC para la VC en un clúster de computadoras.

5 Conclusiones

El nuevo estilo de trabajo que se presenta con el surgimiento de los CC permite acelerar el desarrollo de las investigaciones en distintas áreas de la ciencia, pero exige una alta demanda de recursos computacionales. En esta investigación se propone un nuevo modelo de centro de ciencia basado en clúster de computadoras, que constituye una alternativa asequible para instituciones que tienen limitaciones en cuanto a la disponibilidad de recursos. El modelo presentado está orientado principalmente al análisis de datos mediante la visualización. Con este objetivo se proponen herramientas y soluciones que permiten integrar los CC con la VC.

El modelo planteado fue validado mediante un caso de estudio, que muestra la validez y eficacia de las tecnologías propuestas, y constituye una guía para el futuro desarrollo e implementación de centros similares.

De manera particular:

1. Se determinaron las tendencias actuales en el desarrollo de centros de ciencia de áreas específicas, teniendo en cuenta aspectos tales como tecnologías, arquitectura y servicios brindados, a partir del estudio de algunos de los grandes CC de mayor auge en los últimos años.

2. Se comprobó que la mayoría de las funcionalidades de los centros de ciencia pueden implementarse sobre un clúster de computadoras.

3. Se describe un caso de estudio, que muestra cómo implementar sobre un clúster de computadoras las principales funcionalidades de los CC, como espacios personales de trabajo, el trabajo con metadatos, el control de usuarios, el paralelismo, etc.

4. Se diseñó un modelo de arquitectura para los servicios y aplicaciones brindados por un CC para la VC, se proponen soluciones tecnológicas que satisfacen los requerimientos de un CC para la VC sobre clústers de computadoras y se ofrece una guía práctica para la implementación de estas soluciones.

6 Recomendaciones

Como extensión y continuación de este trabajo se propone:

1. Estudiar otras posibles herramientas y soluciones tecnológicas que faciliten la implementación de las funcionalidades de un centro de ciencia sobre un clúster de computadoras, incluyendo la eventual adaptación de herramientas de tipo libre para este fin.

2. Extender el modelo propuesto para permitir la colaboración entre distintos centros de ciencia basados en clúster de computadoras.

3. Realizar una comparación de eficiencia y rendimiento entre la solución propuesta y el uso de un clúster de BD para al almacenamiento de los datos.

7 Referencias Bibliográficas

ALONSO, J. M. (1997) Programación de aplicaciones paralelas con MPI (Message Passing Interface).

BABAR (2007) BaBar Public Web Home Page.

BECLA, J. & WANG, D. L. (2005) Lessons Learned from Managing a Petabyte *CIDR*.

BIRN (2008) Biomedical Informatics Research Network.

BOGHOSIAN, B. M. & COVENEY, P. V. (2005) Scientific Applications Of Grid Computing.

BREWER, E. A. (2004) Parallel Databases.

BRUCH, K. M., FERGUSON, C., GANNIS, M., GRAHAM, R., HART, D., MCINTOSH, L., MAISEL, M. & TOOBY, P. (2002) Building a Community Grid *ENVISION*, 18-3.

CANNATARO, M., COMITO, C., SCHIAVO, F. L. & VELTRI, P. (2004) Proteus, a Grid based Problem Solving Environment for Bioinformatics: Architecture and Experiments. *THE IEEE Computational Intelligence BULLETIN*, 3.

CAO, P. (2008) SRB or iRODS.

CAO, P., FOLK, M., WAN, M. & MOORE, R. (2005) Integration of HDF5 and the SRB for Object-level Data Access.

CASJOBS (2008) SDSS CasJobs site.

CHENG, A., KOZIOL, Q. & WENDLING, B. (2003) Flexible Parallel HDF5.

DEWITT, D. J., GRAY, J., DEPARTMENT, C. S. & CENTER, S. F. S. (1992) Parallel Database Systems: The Future of High Performance Database Processing1.

DONGARRA, J. (1996) P4 and Parmacs.

DORIGO, A., ELMER, P., FURANO, F. & HANUSHEVSKY, A. (2004) XROOTD - A highly scalable architecture for data access.

ENRIQUEZ, S. (2008) Uso de Sistemas de Bases de Datos Paralelas para la Visualización Científica

FRÜHAUF, T. (1997) Graphisch-Interaktive Strömungsvisualisierung. *Springer Verlag*.

GAO, F. (2008) REVIEW OF DATA REPRESENTATION SYSTEMS SWOT Analysis of NetCDF.

GENBANK (2008) GenBank Overview.

GIBBS, J. (1991) Massively Parallel Systems, Rethinking Computing for Business and Science.

GORDON, J. & BOYD, D. (2001) The Particle Physics Grid Programme at CLRC, GRIDS: e-Science to e-Business. *European Research Consortium for Informatics and Mathematics*, 45.

GRAMA, A., GUPTA, A., KARYPIS, G. & KUMAR, V. (2003) *Introduction to Parallel Computing*, Addison Wesley.

GRAY, J., LIU, D. T., NIETO-SANTISTEBAN, M., SZALAY, A. S., DEWITT, D. & HEBER, G. (2005) Scientific Data Management in the Coming Decade. Redmond, Microsoft Corporation.

GRETHE, J. S. (2006) The Biomedical Informatics Research Network: A National Information Infrastructure to Enable and Advance Biomedical Research. BIRN Coordinating Center - University of California San Diego.

HANSEN, C. D. & JOHNSON, C. R. (Eds.) (2005) *The Visualization Handbook*, ELSEVIER.
HEIJMANS, J. (2002a) An Introduction to Distributed Visualization, 20-22.
HEIJMANS, J. (2002b) An Introduction to Distributed Visualization, 22-29.
MOORE, R. W. (2006) Digital Libraries and Data Intensive Computing.
MOORE, R. W. & JAGATHEESAN, A. (2004) DATA GRID MANAGEMENT SYSTEMS.
MOORE, R. W., SCHROEDER, W., RAJASEKAR, A. & WAN, M. (2008) Rule--Based Distributed Data Management iRODS 1.0
MORELL, A. & PÉREZ, C. (2006) Biblioteca de módulos de visualización de fluidos para OpenDX.
NCBI (2002) The NCBI Handbook. User Services: Helping You Find Your Way
NCBI (2008) National Center for Biotechnology Information.
NIEF, J.-Y., KROEGER, W. & HASAN, A. (2005) BaBar data distribution using the Storage Resource Broker (SRB).
PACITTI, E., COULON, C., VALDURIEZ, P. & ÖZSU, M. T. (2005) Preventive Replication in a Database Cluster.
RAJASEKAR, A., WAN, M., MOORE, R., JAGATHEESAN, A. & KREMENEK, G. (2002) Real Experiences with Data Grids - Case studies in using the SRB San Diego, San Diego Supercomputer Center (SDSC), University of California at San Diego.
RAJASEKAR, A., WAN, M., MOORE, R., SCHROEDER, W., KREMENEK, G., JAGATHEESAN, A., COWART, C., ZHU, B., CHEN, S.-Y. & OLSCHANOWSKY, R. (2003) Storage Resource Broker – Managing Distributed Data in a Grid.
REED, D., AYDT, R., MADHYASTHA, T., NOE, R., SHIELDS, K. & SCHWARTZ, B. (1990) ParaSoft Express. User's Guide.
ROOT (2004) Official ROOT site.
SDSS (2008) What is the Sloan Digital Sky Survey.
SHANKAR, S. & DEWITT, D. J. (2006) Data Driven Workflow Planning in Cluster Management Systems. IN ACM (Ed. *HPDC'07*. Monterey, California, ACM.
SHANKAR, S., ET AL (2006) Data Driven Workflow Planning in Cluster management Systems.
SULLIVAN, T. (2007) Running BaBar Applications with a Web-Services Based Grid, . *Spring*.
SZALAY, A. S., GRAY, J., THAKAR, A. R., KUNSZT, P. Z., MALIK, T., RADDICK, J., STOUGHTON, C. & VANDENBERG, J. (2002) The SDSS SkyServer – Public Access to the Sloan Digital Sky Server Data. Madison.
TANDEM (1988) Tandem Performance Group: A Benchmark of Non-Stop SQL on the Debit Credit Transaction. *SIGMOD Conference.*
TAVERNA (2008) Taverna project website.
TERADATA (1983) DBC/1012 Data Base Computer Concepts & Facilities.
THAKAR, A. R., SZALAY, A. S., KUNSZT, P. Z. & GRAY, J. (2004) The Sloan Digital Sky Survey Science Archive Migrating a Multi-Terabyte Astronomical Archive from Object to Relational DBMS
WATSON, P. & PATON, N. (2003) Grid Data Management Systems & Services.
WOLSTENCROFT, K. (2008) An Introduction to Taverna Workflows.