CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

SEDE ZACATENCO

DEPARTAMENTO DE COMPUTACIÓN

Minería de datos visual sobre una pared de video

Tesis que presenta

Laura Patricia Ramírez Rivera

para obtener el Grado de

Maestra en Ciencias

En Computación

Director de la Tesis

Dr.Sergio Víctor Chapa Vergara

México, D.F. **Octubre 2008**

Agradecimientos

Agradezco a todos los que me acompañaron durante este trayecto, a mis padres, a mis hermanos, a mis profesores, a mis amigos, al conacyt y al cinvestav.

Resumen

Nunca antes en la historia se han generado tal cantidad de datos como en estos días. Explorar y analizar todos estos datos empieza a ser increíblemente difícil. La visualización y la minería visual pueden ayudar a tratar este enorme flujo de datos. La ventaja de la exploración visual es que el usuario es envuelto en el proceso de minería, lo cual permite una manera más intuitiva en el descubrimiento de la información.

Ese problema implica buscar la manera de aprovechar el uso de técnicas de minería visual, las cuales permiten el apoyo de la tecnología en pro de un buen entendimiento de los datos, además es posible aprovechar las ventajas que ofrece una pared de video. Podemos usar una pared de video para desplegar múltiples gráficas de los mismos datos y así permitir su visualización en diferentes planos.

El objetivo del proyecto es aprovechar la tecnología para permitir el descubrimiento del conocimiento, a través del uso de una minería visual con el apoyo de una pared de video, es decir se trata de aplicar un enfoque matemático-computacional, enlazando una correlación de Pearson con el manejo de una pared de video.

El propósito de este escrito es mostrar el desarrollo teórico y práctico realizado para el funcionamiento de este proyecto.

El sistema se construyó con 6 Mac mini, 6 pantallas de 23 pulgadas, un servidor de video Mac pro, y un servidor de base de datos G5. El manejador de base de datos que se empleo es PostgreSQL. La aplicación fue desarrollada con C-objetivo con las bibliotecas de libpq (para la comunicación con PostrgeSQL) y el framework de OpenGL (para la generación de los gráficos).

El sistema consiste de una interfaz principal ubicada en el servidor de video, que es donde se calcula la correlación de las variables. En cada Mac mini se ubica un proceso cliente para permitir la manipulación del despliegue dentro de su área.

Palabras clave-:< visualizador >,< servidor>, < intérpretes > .

Abstract

The history never before has seen the generation so much data as in these days. Explore and analyze all this data is becoming incredibly difficult. The visualization and visual mining can help deal with this enormous data flow. The advantage of the exploration vision is that the user is involved in the process of mining, which allows a more intuitive in the discovery of information.

That problem involves looking for ways to exploit the use of visual mining techniques, which allow the support of technology towards a good understanding of the data, and it is possible to exploit the advantages offered by a video wall. We can use a video wall to display multiple graphs of the same data and thus allow their visualization at different levels.

The target project is harness technology to enable the discovery of knowledge, through the use of a visual mining is supported by a wall of video, the principal idea is applying a mathematical-computational approach, linking a Pearson correlation managing a video wall.

The purpose of this work is to show the theoretical and practical done for the operation of this project.

The system was built with 6 Mac mini, 6 screens of 23-inch , a video server for Mac, and a database server G5. The handler database that employment is PostgreSQL. The application was developed with objetive-C with libraries libpq (for communication with PostrgeSQL) and the OpenGL framework (for the generation of graphics).

The system consists of a main interface located in the video server, where it is estimated the correlation of variables. Each Mac mini is located a process to allow the customer manipulation deployment inside their area.

Índice general

Índice de figuras

Capítulo 1

Introducción

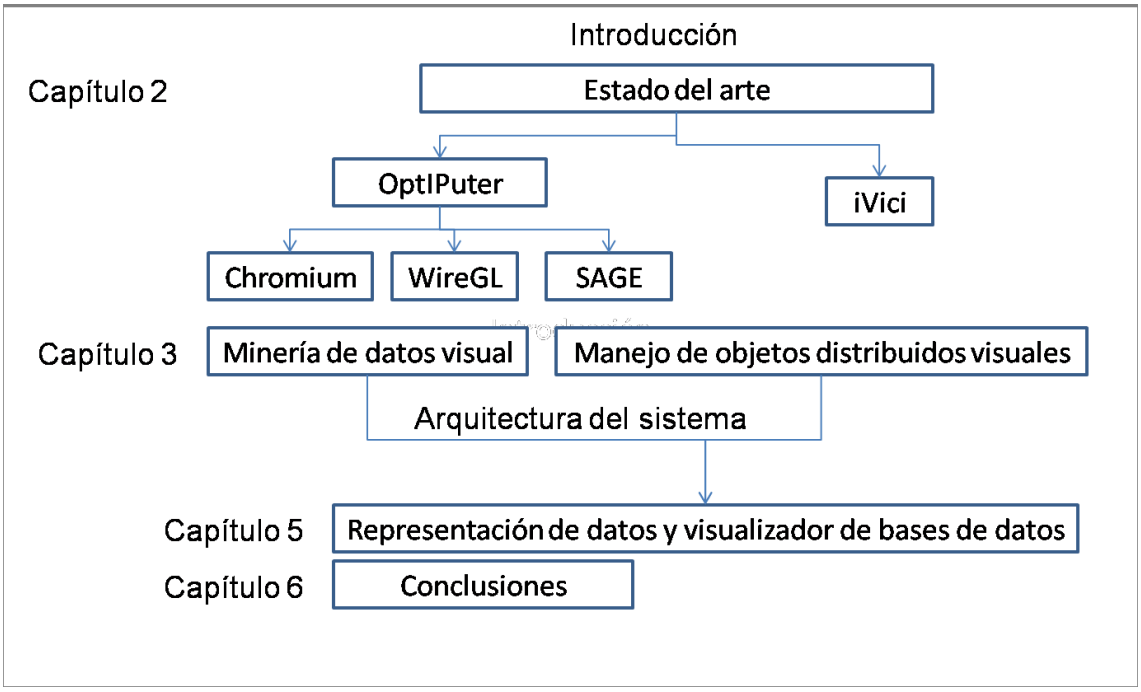

Figura 1.1: Estructura del documento de tesis

Para que la minería de datos sea efectiva, es importante incluir al humano en el proceso de exploración de los datos, en general es deseable combinar el conocimiento humano con la enorme capacidad de almacenamiento que se tiene en las máquinas así como su poder de procesamiento.

Las exploraciones visuales de datos tienen como objetivo la integración del humano en el proceso de exploración, gestión que tienen los sistemas computacionales.

Esto con el fin de aprovechar las capacidades humanas como la deducción y la intuición, permitiendo así obtener una minería más completa, tomando en cuenta las capacidades inherentes a nuestra especie.

Para apoyar este tipo de exploraciones se propone el uso de una pared de video que permita una amplia visualización de los datos al mismo tiempo que se realiza el proceso de minería. Es por eso que el objetivo final es crear un sistema que permita visualizar bases de datos científicas y mostrar los gráficos obtenidos de sus atributos sobre una pared de video.

Cabe recalcar que la pared de video es un medio tecnológico eficaz, que aun no ha sido desarrollado, ni estandarizado, por lo que involucra la investigación sobre su manejo, es decir requiere conocimiento computacional que todavía no ha sido desarrollado, y del cual solo se han encontrado ejemplos de uso, e ideas y sugerencias para su implantación.

Existen diversas investigaciones para crear visualizadores de bases de datos, que varían en la forma en la que despliegan o manipulan los datos.

1.1. Planteamiento del problema

Desde hace tiempo que dentro de diversas areas del conocimiento se han generado grandes cantidades de datos.

En ocasiones son tan numerosos que su manejo es difícil.

Tenemos el problema de localizar patrones que nos ayuden a ver como es el comportamiento de los datos y de ahí tratar de comprender la información contenida en ellos.

Existe también el problema del transporte en un tiempo relativamente corto. Se debe mencionar el tiempo de procesamiento que incrementa en proporción a la cantidad de datos que se presente.

Si unimos todos estos problemas tenemos un problema que básicamente consiste en obtener conocimiento de grandes bases de datos científicas, disminuyendo en la medida de lo posible el costo computacional que esto representa.

En cuanto a la solución propuesta para estos problemas planteamos lo siguiente. Podemos aprovechar las ventajas que la visualización nos proporciona para la mejor y más fácil apreciación de los datos. Desplegando los gráficos en una área mayor como es una pared de video, podemos observar más de un gráfico a la vez, permitiendo una mejor apreciación del comportamineto de los datos.

Si permitimos el manejo de los objetos gráficos a disposición del usuario por toda el área de despliegue, el podrá hacer conjeturas mediante comparaciones evidentes. Se propone la correlación de Pearson como el método que mostrará evidencia de comportamiento similar en los datos. Se propone el manejo de objetos distribuidos para evitar la interpretación centralizada.

Se han encontrado diversos trabajos que con tecnología especializada como redes ópticas y grandes cantidades de memoria ram, obtienen información de las bases de datos, la idea de este proyecto es aprovechar en la medida de lo posible las características del manejo de objetos distribuidos en la plataforma Mac.

La idea es implementar un sistema que conste de 4 partes principales, que solucionarán en conjunto el problema que se describió anteriormente:

- Parte 1. En cuanto a la obtención de la información de la base de datos, debemos tomar en cuenta el manejador de la base de datos. Debemos generar la conexión con el servidor de forma que emplee el menor tiempo posible. Se debe verificar la forma en la que se manejarán los datos.
- Parte 2. En cuanto a las operaciones que se realizarán para la obtención de la correlación de Pearson. Se debe tomar en cuenta que los datos pueden ser manipulados desde una archivo ya que la consulta se haya terminado. Debemos tener una forma segura para realizar las operaciones y evitar desbordes. Tomando en cuenta que no sabemos que tan grande es la base a manipular se debe ser cuidadoso en ese aspecto.
- Parte 3. La parte que realiza el manejo de los objetos distribuidos. Contiene una variante devido a que el tipo de objetos que se manipularán son gráficos. El protocolo que se utiliza es cliente servidor. Donde los servidores ponen a disposición sus objetos al cliente quien los manipula de manera transparente una ves que este funcionando la conexión.
- Parte 4. La generación del gráfico OpenGL representa la parte que ayudará al usuario con la apreciación de la información. Es una parte muy importante dentro de la solución.

Si unimos estas partes podemos generar un sistema que pueda obtener información de bases de datos de gran tamaño generando su correlación y un gráfico que muestre su comportamiento y permita una buena apreciación.

Se realizará un sistema que permita el manejo de bases de datos científicas, para obtener sus atributos y generar la correlación de datos sobre 18 variables distintas. Los resultados de las correlaciones se mostrarán en una matriz de colores, la cual permitirá generar las gráficas de los pares de variables, que el usuario desee. Los contenedores de gráficos serán de tipo OpenGL, y podrán moverse por el área de las pantallas que formen parte del cluster de visualización.

1.1.1. Cluster de visualización

Un cluster de visualización es un conjunto de nodos que comparten su salida de video con el objetivo de formar un dispositivo visual de mayor dimensión y resolución.

Un cluster de visualización consta de un servidor de video que es el encargado de hacer disponibles los objetos visuales, para que los procesos clientes accedan a ellos, y los nodos que permitirán la interpretación y despliegue de los mismos.

1.1.2. Manejo de objetos distribuidos

La computación distribuida, es un nuevo modelo para resolver problemas de computación masiva utilizando un gran número de computadoras organizadas en grupos incrustados en una infraestructura de telecomunicaciones distribuida que ha sido diseñado para resolver problemas demasiado grandes para cualquier supercomputadora, mientras se mantiene la flexibilidad de trabajar en múltiples problemas más pequeños. Para que un cluster funcione como tal, no basta solo con conectar entre sí los ordenadores, sino que es necesario proveer un sistema de manejo del cluster, el cual se encargue de interactuar con el usuario y los procesos que corren en él para optimizar el funcionamiento.

Los mensajes remotos en C-objetivo proporcionan un sistema en tiempo de ejecución que permite establecer conexiones entre objetos en diferentes espacios de direcciones, reconociendo cuando un mensaje es invocado por un objeto en una dirección remota y transferir los datos de una dirección a otra.

Usando objetos distribuidos, se pueden enviar mensajes en C-objetivo a objetos en otras tareas o tener mensajes ejecutados en otros hilos de la misma tarea. Para enviar un mensaje remoto, una aplicación debe primero establecer la conexión con el objeto receptor. Después del establecimiento de la conexión se comunica con el objeto remoto a través del proxy. El proxy asume la identidad del objeto remoto.

Dentro de este contexto se emplean ciertas clases que a continuación se definen: NSView que es una clase abstracta que define las bases para dibujar, imprimir y manejar eventos de una aplicación.

NSConnection son objetos que permiten la comunicación entre objetos en diferentes hilos o tareas, en un host local o sobre una red. Son la parte más importante en el mecanismo de los objetos distribuidos y normalmente operan detrás de ellos. NSPort es una clase abstracta que representa un canal de comunicación.

Una derivación de NSPort para la conexión es NSSocketPort el cual permite la comunicación tanto a nivel local como para la comunicación a distancia.

1.1.3. Generación de objetos gráficos

Un objeto grafico debe ser eficiente en cuanto a su manipulación desde la interfaz, debe además permitir la manipulación de diversos tipos de eventos.

Podemos aprovechar las ventajas conocidas en cierto tipo de objetos visuales, que permiten ciertas características, como manejo de eventos del ratón, para desplegar la información necesaria dentro de una aproximación a la minería de datos visual.

En esta parte es necesario considerar los aspectos que incluyen su interpretación desde los datos, el tipo de gráfico que se generará, así como la forma en la que puede manipularse en tiempo real desde el servidor.

Otro aspecto importante incluye la dimensión que se manejará, ya que depende en gran medida del tipo de datos contenidos por la base de datos, pero puede tambien obtenerse como una simulación de su comportamiento, es decir, a partir de los datos obtenidos inferir un comportamiento y generar un gráfico representativo.

1.2. Organización de la tesis

Esta tesis está organizada en 7 capítulos que describen ampliamente cada una de las fases del sistema, la figura 1 muestra la estructura del documento.

- Introducción (Capítulo 1).- Se describe el problema a resolver con esta tesis, un diagrama a bloques del trabajo a desarrollar, así como un resumen general de las técnicas y procesos que se emplearán en cada una de las fases del sistema; se presentan algunas definiciones sobre cluster de visualización, así como la organización de la tesis en su totalidad y algunos conceptos que se manejarán a lo largo de la tesis.

- Estado del arte (Capítulo 2).- En este capítulo se muestra un resumen de diversos proyectos que tratan de resolver el mismo problema, dichos problemas fueron tomados en la literatura de: ACM, IEEE y laboratorios de University of Illinois at Chicago, Stanford University and Intel Corporation, al terminar el capítulo se encontrarán las conclusiones, asi como una tabla comparativa.

- Minería de datos visual (Capítulo 3).- . En este capítulo podemos encontrar la base teorica de minería de datos, y algunos conceptos importantes para su entendimiento.

- Manejo de objetos distribuidos visuales (Capítulo 4).- . Definición del manejo de objetos distribuidos sobre la plataforma Mac, y la idea del manejo de objetos visuales, se define el manejo de la herramienta de administración, y la configuración de la red local para el manejo de los intérpretes del cluster.

- Representación de datos y visualizador de base datos. (Capítulo 5).- Se explica cómo se maneja el sistema y que bloques de software y arquitectura en hardware lo componen.
- Conclusiones y trabajo a futuro (Capítulo 6).- Se muestra en suma cual es la aportación del trabajo desarrollado y las propuestas de los trabajos a futuro.
- Caso de estudio: Sinac (Apéndice A).- Se muestra como es el funcionamiento de la solución propuesta sobre la práctica en una base de datos real.

Capítulo 2

Estado del arte

El ver es conocer aunque limitarse a ver no es suficiente. Cuando se entiende lo que se ve, entonces se empieza a creer. Hace tiempo que los científicos descubrieron que ver y entender al mismo tiempo permite descubrir conocimiento y obtener una visión más profunda en la teoría del análisis de grandes cantidades de datos.

Podemos usar un enfoque que integra la capacidad de exploración de la mente humana con el enorme poder de procesamiento de las computadoras para formar un poderoso descubrimiento sobre el ambiente de desarrolló.

El propósito de este capítulo es mostrar varios trabajos relacionados con la visualización de bases de datos, así como algoritmos mejorados basados en el uso del coeficiente de correlación de Pearson. Esta sección contiene un breve análisis de los principales trabajos encontrados.

2.1. Visualización y minería de datos

La exploración de información en espacios heterogéneos requiere métodos de minería tan eficientes como sus interfaces visuales [12]. Hace tiempo que la mayoría de los sistemas se concentraban o en los algoritmos de minería o en las técnicas de visualización. La desventaja que se tiene al separar el análisis de la visualización es que en general no tiene tanto apoyo una en otra, es decir, son aplicaciones diferentes que tratan de unirse, muy diferente a una aplicación cuya idea implique el uso de ambas desde un inicio. Es por eso que en el año 2002 se describió un ambiente de trabajo para la minería de datos visual, la cual combinó el análisis de datos y la visualización de los datos, para permitir el mejor entendimiento del espacio de información.

Se desarrollaron técnicas de visualización para datos complejos. Donde una de las características principales del sistema fue un nuevo paradigma para visualizar estructuras de información sin importar su marco de referencia. Otras interfaces visuales que se han desarrollado para visualizar e interactuar de forma similar son Cone Trees o Disc Trees.

El objetivo del pre-procesado es ayudar a determinar la información relevante para la visualización. Básicamente se ofrecen las siguientes ventajas:

- Reducción del número de dimensiones.
- Filtrado de datos.

Ese algoritmo buscó similitud entre los objetos de información, mediante la distancia euclidiana o el coeficiente de correlación.

2.1.1. OptIPuter

Es una arquitectura-infraestructura de redes ópticas paralelas para la exploración en parejas de datos, visualización y colaboración de tecnologías pensadas en velocidades IP de multigygabit.
En este proyecto se remarca que los esfuerzos de investigación se dirigen hacia los modelos y abstracciones que simplifiquen las aplicaciones distribuidas.

- Protocolos de alta velocidad y capas de comunicación que ayuden al desempeño de la red.
- Redes dinámicas ópticas, configuración, manejo, etc.
- Abstracción de la comunicación (paralela y multicast) y APIs que permitan elementos de computo escalable para generar conexiones de área ancha.
- Almacenamiento y sistemas de archivos que soporten acceso a alta velocidad de datos remotos.
- Protocolos de seguridad y modelos para ancho de banda altos, gran latencia en un ambiente de millones de recursos.
- Adaptabilidad y acceso a datos punto a punto, e integración de protocolos.
- Modelo de objetos en tiempo real permitiendo el desempeño predecible a través del sistema.

De grandes colecciones de datos reales o simulados se genera la correlación los cuales se representan visualmente y pueden ser distribuidos en pantallas individuales.
Se propone el siguiente paradigma:

- Tomar los datos
- Generar la correlación de los datos
- Emplear el sistema de visualización
- Mostrar la imagen mediante el sistema de despliegue

En el contexto de exploración colaborativa[1] se necesita multicast.

La conectividad entre los componentes de cómputo en gran escala de exploración de datos es estática, las conexiones ya están establecidas.
Basados en las ventajas que ofrece OptIPuter se han creado algunas aplicaciones de visualización y herramientas de colaboración.
En la figura 2.1.1, se muestra una imagen del sistema funcionando, en este caso se muestran imágenes de celulas ampliadas.

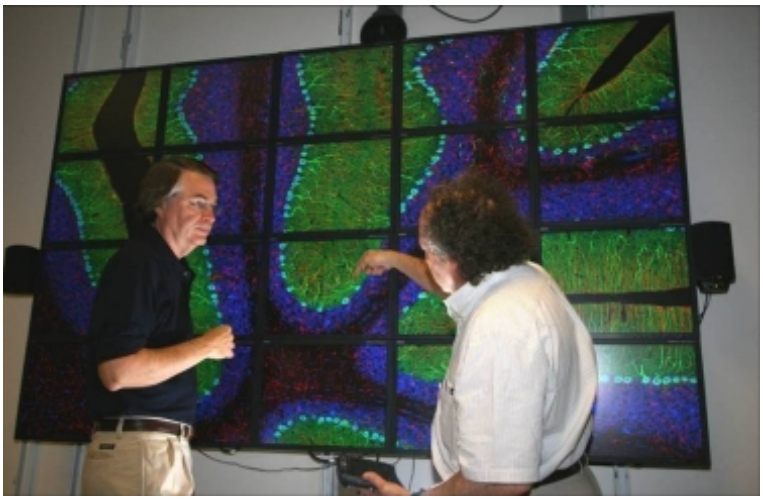

Figura 2.1: OptIPuter, Ref. http://www.optiputer.net

- Terascope.- Monitoreando el desempeño de sistemas de disco remotos, redes y cluster de minería de datos e interpretación, este framework adaptativo e inteligente selecciona una estrategia de visualización apropiada por el tamaño de los datos que inician a ser visualizados[1].
- Juxtavision.- Se toma el concepto de terascope, y esta herramienta es diseñada para desplegar en alta resolución. Incorporando una red de trabajo, un sistema de memoria llamado lambdaram para proveer datos de imágenes para visualizaciones. Se usa OptIPuter para crear grandes bloques de memoria ayudadas por caches en largas distancias de red.
- Teravision.- Diseñado en un formato de alta resolución de gráficos en un cluster de visualización, esta aplicación colaborativa usa una red multicast para distribuir las visualizaciones.

Se concluye que dentro de OptIPuter la infraestructura del visualizador distribuido tenga como elemento principal la red y no las computadoras.

[1]Software colaborativo o groupware se refiere al conjunto de programas que integran el trabajo en un sólo proyecto con muchos usuarios concurrentes encontrados en diversas estaciones de trabajo.

2.1.2. iVici

Es una herramienta desarrollada por el departamento de Bioquímica de la Universidad de Montreal, en Québec, Canada, esta diseñada para la visualización y comparación del análisis de células de genes o redes de proteínas representadas en matrices de dos dimensiones.

El típico flujo de datos consiste de la creación de un primer conjunto de datos que contienen una matriz de asociación describiendo, por ejemplo, interacciones proteína-proteína.

Los nombres de los genes y las proteínas en el conjunto de datos son nombres de las filas y las columnas.

Una visualización previa, por ejemplo, podemos tomar las bases de datos y clasificar con simetría jerárquica o usar otros métodos que puedan revelar patrones de asociación entre los genes o las proteínas[8].

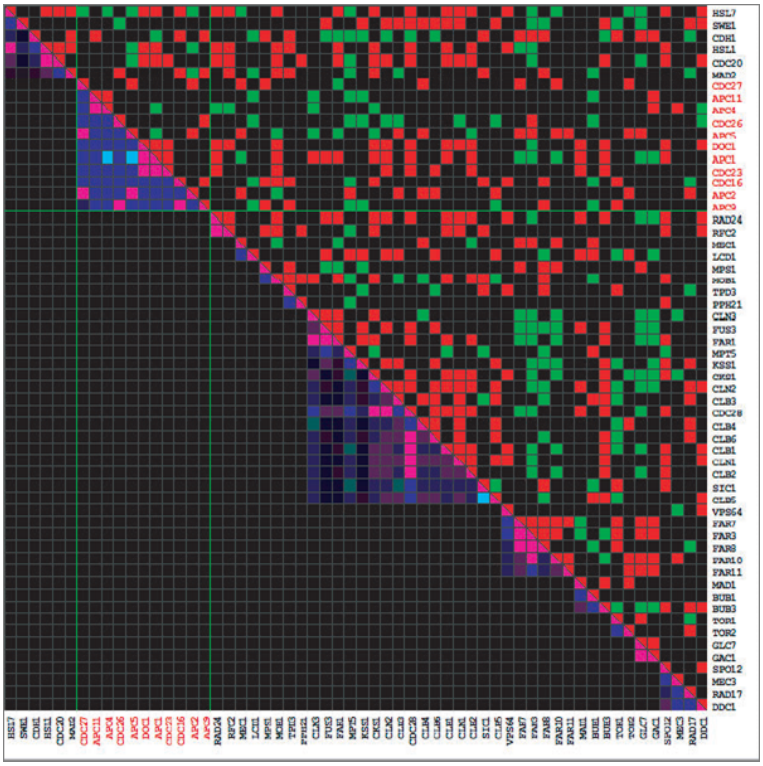

Figura 2.2: iVici, Ref. http://michnick.bcm.umontreal.ca

La figura 2.1.2 muestra la matriz de colores que genera el sistema, donde la intensidad del color esta definida por el valor de la correlación, además cada color representa los diferentes grupos de datos.

iVici, es un método para buscar una correlación en dos dimensiones para analizar datos presentados con múltiples variables, con lo cual se pueden mostrar los datos con mayor facilidad. Se usa la desviación estándar para determinar magnitud de la variación en la intensidad, la cual se muestra dentro de una matriz de colores.

Se muestran los cambios de la correlación tomando en cuenta que el signo indica la dirección, así que podemos combinar el signo del coeficiente de correlación con la magnitud de la desviación estándar para obtener mayor información. iVici puede seleccionar diferentes colores para interpretar los valores de dos conjuntos de datos. Las propiedades y la elegancia de las representaciones de matrices en dos dimensiones sugiere múltiples caminos por los cuales podemos permitirnos un rápido e intuitivo análisis visual de grandes bases de datos, otros trabajos similares son infoscope y weka[2].

2.1.3. Chromium

Es un sistema para manipular comandos de gráficos sobre cluster. Sus filtros pueden crear arreglos de arquitecturas paralelas de gráficos. Este sistema se construyó para proveer un mecanismo genérico para manipular grupos de gráficos con comandos de una API. Se puede usar como un mecanismo para implementar algoritmos de gráficos en cluster, se usa la biblioteca de OpenGL para mover los gráficos.

Además existen aplicaciones en OpenGL que pueden usarse en clusters con pocas modificaciones porque este sistema provee una API estándar que permite utilizar los recursos presentes en un cluster para la interpretación de los gráficos.

La compatibilidad con las aplicaciones existentes puede acelerar la adopción del cluster de interpretación y despliegues de alta resolución, de tal forma que se pueda explotar la resolución y el paralelismo.

Dentro de Chromium los procesadores son implementados como módulos que pueden ser interconectados y combinados de cualquier forma. Para modificar la configuración de estos equipos de procesadores, se crearon arquitecturas paralelas de gráficos que pueden en muchos casos soportar las mismas aplicaciones sin recompilar.
El enfoque es tener componentes cómodos del cluster, por eso solo se consideran arquitecturas que no requieren comunicación entre los estados del bus que no son expuestos en la aplicación.

2.1.4. WireGL

Es un sistema basado en cluster que dirige unas aplicaciones para desplegar imágenes sobre una red. Usa ténicas de interpretación paralela[2] para mostrar tamaños escalables con el mínimo impacto en el desempeño de la aplicación.
WireGL intercepta los comandos hechos por una aplicación y genera multiples nuevas secuencias de comandos cada una representada en un compacto protocolo. Cada secuencia es transmitida sobre una red o diferentes servidores.

Este sistema fue diseñado tomando como ventaja la alta velocidad de un servidor de área de red, común en el uso de cluster, mientras soportamos infraestructuras heterogeneas en la red.
WireGL usa técnicas de interpretación paralelizada que minimizan la transmisión de los graficos. Podemos definir este sistema por medio de sus partes, las cuales se mencionan a continuación:

- Una interfaz distribuida y paralela de OpenGL.
- Una interfaz general de OpenGL que genere máquinas en la red para mostrar la imagen distribuidamente.
- Un mecanismo eficiente para la API de OpenGL.
- Un estado robusto que permita la actualizaciónes del modelo de OpenGL.
- Una abstración de la red que permita el soporte de diferentes tecnologías de la red con la API.
- Un servidor de red que maneje recepción asíncrona del trabajo y resuelva el orden de las restricciones basadas en las extenciones de la API de OpenGL.

Los componentes de wireGL están ocultos de la aplicación y de la extensión desarrollada por WireGL. La capa de interfaz de la aplicación está completamente oculta, es decir enmascarada como OpenGL, de la aplicación en desarrollo[3].

2.1.5. Collaborative Visualization using High Resolution Tiled Displays

Este trabajo trata de desplegar información de forma heterogénea sobre una pared de video, para trabajar de forma colaborativa, se buscó la forma de cómo mostrar en alta resolución que pueda ser usado para mostrar el detalle y el contexto de los datos simultáneamente.

[2]Trata de distribuir paralelamente la carga de trabajo implicada por la generación del gráfico en OpenGL

Se tomaron algunos trabajos funcionales para construirlo usando LambdaVision con una pantalla total de 100 megapixeles y con el desarrollo de SAGE (Scalable Adaptative Graphics Enviroment). La meta es habilitar la exploración científica de datos y compartir aplicaciones sobre una pantalla de alta resolución. La arquitectura de gráficos de SAGE se convierte en un problema poco trivial de visualización científica.
La figura 2.1.5 muestra el sistema permitiendo el trabajo colaborativo entre un grupo de usuarios.

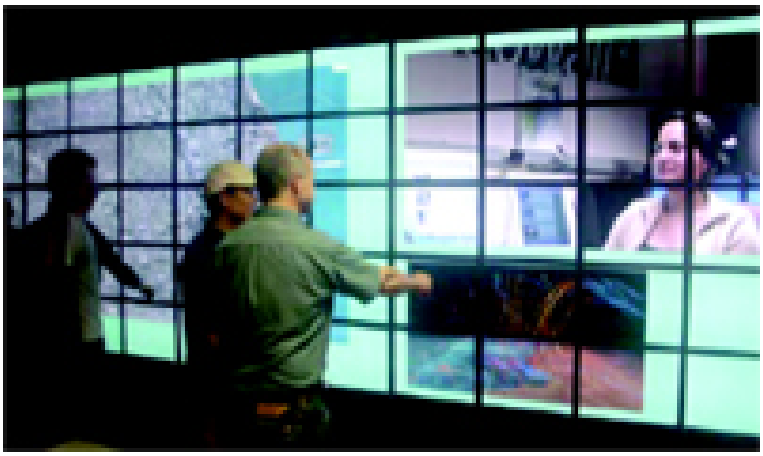

Figura 2.3: Funcionamiento del sistema, Ref. http://www.evl.uic.edu/

Uno de los problemas principales es la heterogeneidad ya que la mayoría de las aplicaciones de visualización son cerradas en cuanto al desarrollo de sus gráficos, por lo que es difícil integrar varias aplicaciones de visualización en un solo desarrollador unificado de gráficos.
El otro problema es la escalabilidad, la habilidad del software de visualización y los sistemas escalables en términos de la cantidad de datos que pueden ser visualizados y su resolución. Existen muchos otros sistemas con interpretación paralela remota que se mencionan en SAGE. WireGL usa la interpretación paralela llamada sort-first. Esta aproximación es muy similar a una aplicación serial que construye gráficas de forma primitiva que serán interpretadas de forma paralela por los nodos.
Tiene una pobre escalabilidad en los datos lo cual se puede tomar como limitación. Los usuarios podrán discutir y conocer los datos frente a las pantallas donde cada aplicación corre en una instancia de la GUI de SAGE.

Cuando un usuario quiere manejar algún contenido en la pantalla, debe invocar la aplicación de la UI que soporta el tipo de datos, selecciona un conjunto de datos y una posición en la pantalla. Posee una comunicación básica, chat capaz de unir en una lista a los usuarios que estén conectados a la pantalla. Un visor multimedia básico dentro de la interfaz de usuario. Se pueden conectar múltiples usuarios y controlar la misma aplicación.

2.1.6. SAGE

Este trabajo aprovecha el uso de pantallas de alta resolución y un servicio de red de alta velocidad. Esta plataforma provee de una noción de escritorio con acceso remoto. Existen herramientas como Remote Desktop que son diseñadas para transmitir pantallas simples de escritorios a computadoras remotas sobre redes lentas. Son diseñados para operar con eventos que no pueden ser generados en tiempo real como las aplicaciones colaborativas.
La figura 2.1.6 muestra el manejo de las diferentes áreas generadas por el sistema.

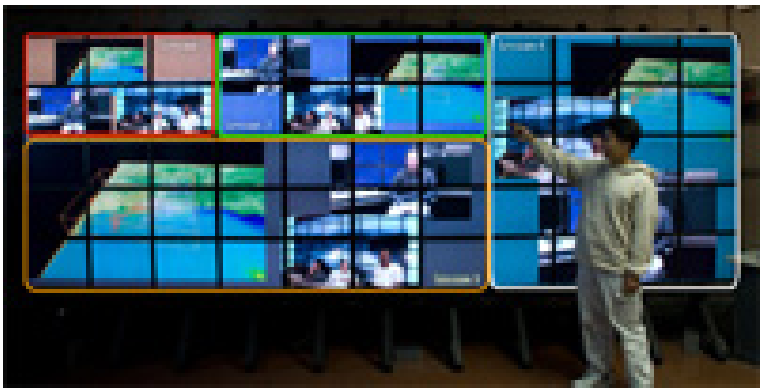

Figura 2.4: Funcionamiento de SAGE, Ref. http://www.evl.uic.edu/cavern/sage/index.php

La motivación para generar a SAGE viene de dos áreas principales. La primera es que las aplicaciones escritas para el desarrollo de gráficos tienen que ser rediseñados antes de poder correr sobre otros desarrollos. Por ejemplo, herramientas para visualizar que son desarrolladas por escritorios de computadoras son raramente usados para tomar ventajas del poder del procesamiento en un cluster de gráficos, si son convertidas entonces no podrán usarse de nuevo en el escritorio.

La segunda, es la habilidad del software de visualización y sistemas escalables en términos de la cantidad de datos que pueden ser visualizados, así como la resolución deseable, lo cual permanece siendo un problema de investigación dentro del área de visualización.

El direccionamiento de SAGE es necesario para soportar la heterogeneidad y escalabilidad para la interpretación de los gráficos tomando ventaja del ancho de banda de la red.

La arquitectura de SAGE consiste de un número de computadoras o cluster de interpretación, cada uno conectado sobre una red escalable. LamdaVision pone el concepto de tecnología de despliegue y anticipa necesidades para manejo masivo de datos en una pared de 55 pantallas conducidas cada una por maquinas en una red de 10 Gygabytes, con un total de 100Megapixeles.
El desarrollo de SAGE con el software de visualización, control de red, manejo de datos e interfaces humano computadora serán un trabajo colaborativo para científicos.

Los principales componentes de visualización del sistema son diseñados en alta resolución, conjuntos de datos geo-referenciados y modelos en 3d. El trabajo de visualización es invocado por un usuario, automáticamente se genera la visualización optima sobre la red disponible en ese momento.

Cada demanda de las aplicaciones de visualización necesita gran ancho de banda para acceder a los datos almacenados en forma remota. El trabajo colaborativo es habilitado en dos niveles: cada uno comparte los conjuntos de datos usando LambdaRam o el resultado será generado por una aplicación de visualización distribuida por Teravision[10].

2.1.7. Lambda Table: Hig Resolution Tiled Display Table for Interacting with Large Visualizations

Para explorar nuevos métodos de interacción con grandes conjuntos de datos científicos, se desarrolló una mesa de visualización de alta dimensión. Este sistema mezcla la interacción de la mesa con interfaces tangibles para el usuario y la tecnología de despliegue en pantallas.

La investigación de visualización científica ha producido sistemas capaces de interpretar grandes bases de datos sin sacrificar el contexto ni el detalle. Soportar el trabajo colaborativo, requiere en este caso una red óptica con un alto ancho de banda.

El LambdaTable esta construido con tres resultados de diversas investigaciones, una interacción con el humano y la mesa, un sistema con una gran área para su despliegue, así como la compartición de manejo de espacio.

En este trabajo se han utilizado pantallas LCD. Ya que estas pantallas son extremadamente necesarias en la solución para ver imágenes grandes porque son baratas, y pueden proporcionar un gran campo de visión como lo muestra la figura 2.1.7.

Figura 2.5: Funcionamiento de LambdaTable, Ref. http://www.evl.uic.edu

Mientras que construirlo con proyectores tendría un costo prohibitivo. El dibujado que se realiza con este sistema de despliegue es para visualización científica y manejo colaborativo, en pocas palabras se construyó una mesa que es capaz de facilitar la interacción con grandes bases de datos. La meta de LambdaTable es proveer a los científicos de un método para visualizar, entender manipulando y compartir sus datos en alta resolución.

2.1.8. Software Enviroment For Cluster based Display System

Una solución evidente para construir una pared de video escalable es usar un cluster de computadoras personales con la comodidad de usar el acelerador de gráficos para mostrar las proyecciones.

Una pantalla es el artefacto más común con el cual el humano inicia a visualizar información dentro de una computadora o una red. La escala y resolución del artefacto de despliegue define cuanta información un usuario puede ver al mismo tiempo.

Este proyecto está enfocado en el uso del desarrollo de herramientas, de aplicación secuencial en una pared de video escalable construido con componentes que funcionan eficientemente. Muchas aplicaciones se pueden ejecutar como películas MPEG y modelos VRML.

Para muchas otras aplicaciones, es extremadamente difícil aplicar esta aproximación. Una razón es que el desempeño de esas aplicaciones son frecuentemente codificadas en el software mismo. Un ejemplo de esto es el manejo de juegos de video para computadoras personales. Las escenas en el juego se describen usando gráficos en formatos estándar. Pero la manipulació de las escenas y la interpretación de las entradas del usuario, y de otros manejos, son frecuentemente tratados en secreto y ocultos en millones de líneas de código a nivel máquina.

Un importante objetivo de la pared de video escalable es diseñar un desarrollo para la ejecución del código y para la generación de aplicaciones de alta resolución.

2.1.9. The Metabuffer: A Scalable Multiresolution Multidisplay 3D Graphics System Using Commodity Rendering Engines

Muchas aplicaciones generadoras de gráficos requieren múltiples despliegues de visualizaciones interpretadas en tiempo real. Lo que hacen es dividir y balancear la interpretación del trabajo en el número de computadoras, enviando su parte a cada una, lo cual es una tarea difícil, cuando este varia.

Este sistema es una construcción multiescalable, que toma la composición de la imagen y la divide entre el número de máquinas.

Este hardware soporta un número escalable de computadoras y un número de pantallas que puede ser independiente del número de máquinas que se tengan.

Cada máquina en la red tiene el mismo acceso a todas las partes del despliegue y sobre el espacio de despliegue como un espacio uniforme.

Se ha seleccionado la interpretación de un esquema sort-last con características únicas. La mayor dificultad del problema es el manejo de los datos a ser procesados. Cada pixel necesita RGB colores, el eje Z y la información de alpha.

Un solo gráfico tiene millones de pixeles, una animación en tiempo real necesita aproximadamente 30 gráficos por segundo.

2.1.10. Parallel Graphics and Interactivity with the Scaleable Graphics Engine

Esta es una implementación de un interpretador en paralelo usando SGE, la meta de este software incluye encontrar caminos eficientes para producir y desplegar gráficos generados en nodos SP.

Una alternativa a la generación de gráficos desde datos sobre muchos nodos SP, pueden ser consolidadas a través de una red en una terminal cualquiera.

El SGE es un componente de hardware que se conecta al switch SP, puede conectarse a un cluster Linux en interfaces Ethernet Gygabyte. Recibe fragmentos de pixeles directamente de algún número de nodos.

El enrutador mantiene múltiples direcciones con los datos del pixel de la entrada ligada a las interfaces del banco de memoria, habilitando las pantallas para los datos almacenado en las memorias paralelas. Los gráficos pueden soportar 16 millones de pixeles y pueden tener una salida múltiple. Las lecturas y escrituras concurrentes con ese ancho de banda en el banco de memoria son de 45 megapixeles por segundo. Cada una de las 8 tarjetas de memoria contiene 2 bancos.

El SGE es capaz de actualizar 8 pantallas manipuladas por su propia tarjeta con un desempeño de 720 megapixeles por segundo. Las tarjetas de despliegue son habilitadas por salidas analógicas y digitales.

2.2. Conclusiones

Se han mostrado aquí algunos de los trabajos encontrados que coinciden en la idea o la implementación del proyecto que se plantea para la tesis. La parte más representativa en cuanto a su desarrollo abarca aspectos como el modo de visualización, el tipo de minería y la manera en la que se trabaja con la pared de video, en las que es el caso.
Cabe hacer mención que la mayoría de las técnicas aquí mostradas emplean como base para su desarrollo la tecnología OptIPuter (basada en redes opticas), entre otras tecnologías que implican una infraestructura que todavía no es posible implementar fuera de un ambiente experimental. El manejo de la pared de video no se ha definido de forma definitiva, solo podemos implementar diferentes paradigmas que nos permiten su uso.
En la figura 2.2 se muestra una tabla comparativa con algunos proyectos que coinciden en algunos aspectos considerados representativos del proyecto, seguida de la figura se explica a que hace referencia cada uno de ellos.

Nombre	Múltiple Despliegue	Generalizado	Procesamiento en cluster	Objetos Distribuidos	Requisitos Especiales
Terascope [Chong et al.,2003]	si	si	si	no	si
iVici [Tarassov et al.,2005]	no	no	si	no	no
InfoScope [MacroFocus]	p	si	no	no	no
Weka4WS [Domenico et al.,2005]	no	si	si	no	no
VDBCSPV [Ramírez 08]	si	si	si	si	no

Figura 2.6: Comparación entre algunos de los trabajos mencionados anteriormente

- Múltiple despliegue.- Hace referencia a la característica que posee el sistema para desplegar información sobre diferentes pantallas, p (parcial) significa que es capaz de mostrar mas de un gráfico al mismo tiempo en la misma pantalla.
- Generalizado.- En este aspecto se caracteriza la manera en la que puede acceder a cualquier tipo de información, específicamente, obtener datos de cualquier base de datos relacional.
- Procesamiento en cluster.- Característica que hace referencia a la capacidad que tiene el sistema para distribuir entre las maquinas, la carga que representa la visualización de los datos.

- Objetos distribuidos.- Definido por la capacidad del sistema para permitir la manipulación de los objetos sobre toda la pared de video, mediante algún tipo de comunicación entre el servidor y los clientes.
- Requisitos especiales.- Esta parte hace énfasis en la tecnología experimental que esta fuera de los estándares de un sistema de cómputo común.

Capítulo 3

Minería de datos visual

La minería de datos visual es un proceso el cual puede extraer conocimiento, sin tener antecedentes sobre los datos iniciales. Es un proceso que consume mucho tiempo de procesamiento y requiere una tecnología compleja, además se puede encontrar con mucho ruido en los datos.

En el proceso de minería de datos nosotros debemos usar un camino efectivo para observar la distribución y la estructura de los datos claramente, entendiendo la mutua relación y desempeño de los datos.

La minería de datos visual combina la visualización de datos y la minería de datos con la finalidad de proporcionar un camino efectivo para resolver este problema.

Ver es conocer, aunque solamente ver en ocasiones no resulta suficiente. Cuando se entiende que es lo que se ve se empieza a creer.

Desde hace un tiempo los científicos descubrieron que ver y entender al mismo tiempo permite a los hombres conocer más sobre los datos con una visión más profunda sobre todo tratandose de grandes cantidades de datos.

Esta aproximación integra las habilidades de exploración de la mente humana con la enorme capacidad de procesamiento de las computadoras.

La tecnología de visualización y análisis de procesos se han desarrollado en varias disciplinas incluyendo la visualización científica, minería de datos, estadística y machine learning que manejan datos muy grandes, con muchas variables, en múltiples dimensiones.

La metodología está basada en ambas funcionalidades que caracterizan las estructuras y muestran datos, apoyándose con las capacidades que tienen los hombres para percibir patrones y relaciones.

La visión de un sistema de minería de datos visual proviene de los siguientes principios:

- Simplicidad
- Autonomía del usuario
- Reusabilidad
- Seguridad.

Un sistema de minería de datos visual debe ser sintácticamente simple para ser útil. Debe ser simple para aprenderlo de modo intuitivo y amigable, con mecanismos de entrada tan buenos como fáciles de interpretar, con el fin de obtener como salida de todo el proceso algún conocimiento.

Simple de aplicar que permita una unión entre la información y el usuario, que facilite una búsqueda veloz del conocimiento. Ejecutando el mínimo de pasos obteniendo los mejores resultados. Un genuino sistema de minería de datos visual no debe imponer conocimiento al usuario, pero debe guiarlo en el proceso de minería dibujando las conclusiones dentro de los gráficos. Los humanos deben estudiar los gráficos y decidir qué es lo que se puede concluir.

Un sistema de minería de datos visual debe proveer una estimación del error o aproximación proyectada en cada paso del proceso de minería, este error puede compensarse por la deficiencia que causa la imprecisión del análisis visual de los datos.

Un sistema reusable debe ser adaptable a una variedad de sistemas y desarrollos que reduzcan el esfuerzo del cliente, proporcionando desempeño y portabilidad en el sistema.Un minero visual práctico debe ser generalmente robusto. Como se usa para la búsqueda de conocimiento nuevo no podemos planear la existencia del conocimiento.

Un sistema portable no puede tener ningún dominio específico de información. Se requiere que el conocimiento recibido de un dominio se pueda adoptar a cualquier otro sin pensar en los requerimientos físicos ni las conexiones electrónicas.

Finalmente, un sistema minero visual debe incluir medidas de seguridad que protejan los datos, el conocimiento descubierto y la identidad del usuario por si el sistema es accedido por muchos usuarios. Sin contar con las técnicas matemáticas y de visualización. Esta parte está limitada por el espacio y por los incrementos tecnológicos que surgen en este campo.

La visualización ha sido usada por la minería de datos como una herramienta de presentación para la vista inicial, la navegación en los datos con estructuras complicadas, y para convertir los resultados del análisis. Generalmente, los métodos de análisis no implican visualización.

3.1. Adquisición de datos

Una parte muy importante en cuanto a la minería de datos, es la forma en la que se realizará la adquisición de datos. Se deben tomar en cuenta varios aspectos como son el tipo de manejador de bases de datos que se usará, la comunicación que se tendrá con la interfaz.

Además el manejo de los atributos, la cantidad de información que se manejará, si es necesario buscar el modo más eficiente en cuanto a la velocidad de la transacción. El formato de los datos contenidos en la fuente de datos nunca es el correcto, y la mayoría de las veces no es posible ni siquiera utilizar algún algoritmo de minería sobre los datos iniciales sin que requieran alguna transformación.

En este paso se filtran los datos con el objetivo de eliminar valores incorrectos, no válidos o desconocidos; según las necesidades y el algoritmo a utilizar. Además se obtienen muestras de los datos en busca de mayor velocidad y eficiencia de los algoritmos, o se reducen el número de valores posibles para los atributos de análisis.

Después de realizar la limpieza de los datos, en la mayoría de los casos se tiene una gran cantidad de variables o atributos. La selección de características reduce el tamaño de los datos, sin apenas sacrificar la calidad del modelo de conocimiento obtenido del proceso de minería; seleccionando las variables más influyentes en el problema. Los métodos para la selección de los atributos que más influencia tienen en el problema son básicamente dos:

- Aquellos basados en la elección de los mejores atributos del problema.
- Aquellos que buscan variables independientes mediante pruebas de sensibilidad, algoritmos de distancia o heurísticos.

Las transformaciones de datos pueden ser de dos tipos básicos: conversión de formatos particulares a tablas de datos y manipulación de dichas tablas. En primer lugar, los conjuntos de datos, generalmente multivariados y de gran tamaño, se encuentran en formatos específicos del domino de aplicación y deben ser organizados con una estructura relacional para lograr mayor flexibilidad. Luego, contando con tablas de datos, se pueden aplicar distintas transformaciones que actúan sobre los valores de las tablas o sobre sus estructuras (eliminación de variables, derivación de nuevas variables, clasificación, ordenamiento, etcétera).

La fase de preparación de los datos es muy importante dentro del proceso más general de descubrimiento de conocimiento a partir de los datos (KDD). Esta fase tiene como objetivo localizar las fuentes de datos, caracterizarlas completamente en estructura y tipo, fusionarlas (opcionalmente) en un almacén central de datos (datawarehouse) para poder visualizarlos y tratarlos mejor, mejorar la calidad de los datos eliminando datos faltantes, corrigiendo datos erróneos, etc. y obtener una vista explorable sobre la cual se puedan aplicar las técnicas de minería de datos.

Esta vista minable puede requerir un procesado adicional para seleccionar las variables más significativas dentro de un conjunto posible, reducir dichas variables a un número adecuado para la técnica que queremos aplicar (la escalabilidad de algunas es limitada), combinar ciertas variables que están muy relacionadas o que funcionan mejor combinadas, adaptar los datos al tipo adecuado para el algoritmo(pasar de numéricos a literales o viceversa, escalado, centrado, etc).

Las herramientas de preprocesamiento de datos son llamadas filtros y junto al análisis forman dos procesos principales en minería de datos. Los filtros de atributos realizan el preprocesamiento en dirección a los rasgos del conjunto de datos, significando que los mismos hacen cambios en el número o definición de los atributos.

Por otro lado, los filtros de instancias realizan un preprocesamiento orientado a las mismas, por lo que no afectan los atributos del conjunto de datos. Permiten realizar acciones como adicionar, eliminar o modificar instancias.

El funcionamiento de los filtros de forma general se basa en tomar las instancias en lotes, preprocesarlas y ubicarlas en una cola de salida de instancias ya filtradas, por lo que todos hacen uso de esta cola y de una variable bandera que indica la terminación o no de un lote de entrada. Un filtro no debe cambiar los datos de entrada, ni agregar instancias al conjunto de datos usado para definir el formato de entrada.

Análisis de datos es la actividad de transformar un conjunto de datos con el objetivo de extraer información útil y facilitar así la formulación de conclusiones. En función del tipo de datos y de la cuestión planteada, puede involucrar la aplicación de métodos estadísticos, ajuste de curvas, selección o rechazo de determinados subconjuntos de datos, y otras técnicas.

En contraste con la técnica de minería de datos, el análisis de datos se utiliza no tanto para el descubrimiento de patrones ocultos, sino para la verificación o rechazo de un modelo existente o para la extracción de parámetros necesarios para el ajuste de un modelo teórico a la realidad.

3.2. Minería de datos

La minería de datos no surge a partir de nuevas tecnologías, sino que, se crea por nuevas necesidades y, especialmente, por el reconocimiento de un nuevo potencial en la experimentación computacional.

Además cabe notar el valor, hasta ahora subutilizado, de la gran cantidad de datos almacenados en los sistemas de información de instituciones, empresas, gobiernos y particulares.

Los datos pasan de ser un producto (el resultado histórico de los sistemas de información) a ser una materia prima que hay que explotar para obtener el verdadero producto elaborado, el conocimiento; un conocimiento que ha de ser especialmente valioso para la ayuda en la toma de desiciones sobre el ámbito en el que se han recopilado o extraído los datos.

En principio la estadística, ha considerado la descripción y el análisis de grandes volúmenes de datos que cumplen con cierto comportamiento y tipo. sin embargo, a partir del manejo simbolico de la información y nuevas necesidades y, en particular, las nuevas características de los datos (en volumen y tipología) hacen que las disciplinas que integran lo que se conoce como minería de datos sean numerosas y heterogéneas. El aumento del volumen y variedad de información que se encuentra en bases de datos digitales y otras fuentes, han crecido espectacularmente en las últimas décadas. Gran parte de esta información es histórica, es decir, representa transacciones o situaciones que se han producido.

Aparte de su función de memoria de la organización, la información histórica es útil para explicar el pasado, entender el presente y predecir la información futura. La mayoría de las decisiones de empresas, organizaciones e instituciones se basan también en información sobre experiencias pasadas extraídas de fuentes muy diversas.

Además, ya que los datos pueden proceder de fuentes diversas y pertenecer a diferentes dominios, parece clara la inminente necesidad de analizar los mismos para la obtención de información útil para la organización.

En muchas situaciones, el método tradicional de convertir los datos en conocimiento consiste en un análisis e interpretación realizada de forma manual. El especialista en la materia, digamos por ejemplo un médico, analiza los datos y elabora un informe o hipótesis que refleja las tendencias y pautas de los mismos.

Por ejemplo un grupo de médicos puede analizar la evolución de enfermedades infecto contagiosas entre la población para determinar el rango de edad más frecuente de las personas afectadas. Este conocimiento validado convenientemente, puede ser usado en este caso por la autoridad sanitaria competente para establecer políticas de vacunaciones. Esta forma de actuar es lenta, cara y altamente subjetiva.

De hecho, el análisis manual es impracticable en dominios donde el volumen de los datos crece exponencialmente: la enorme abundancia de datos desborda la capacidad humana de comprenderlos sin la ayuda de herramientas potentes. Consecuentemente, muchas decisiones importantes se realizan, no sobre la gran cantidad de datos disponibles, sino siguiendo la propia intuición del usuario al no disponer de las herramientas necesarias. Éste es el principal cometido de la minería de datos: resolver problemas analizando los datos presentes en las bases de datos.

Por ejemplo, supongamos que una cadena de supermercados quiere ampliar su zona de actuación abriendo nuevos locales. Para ello, la empresa analiza la información disponible en la base de datos de clientes para determinar el perfil de los mismos y hace uso de diferentes indicadores demográficos que le permiten determinar los lugares más idóneos para los nuevos emplazamientos. La clave para resolver este problema es analizar los datos para identificar el patrón que define las características de los clientes más fieles y que se usa posteriormente para identificar el número de futuros buenos clientes de cada zona[7].

3.2.1. Análisis de datos

Hasta no hace mucho, el análisis de los datos de una base de datos se realizaba mediante consultas efectuadas con lenguajes generalistas de consulta, como el SQL, y se producía sobre la base de datos operacional, es decir, junto al procesamiento transaccional en línea de las aplicaciones de la gestión. No obstante, esta manera de actuar sólo permitía generar información resumida de una manera previamente establecida, poco flexible y, sobre todo, poco escalable a grandes volúmenes de datos.

La tecnología de bases de datos han respondido a este reto con una nueva arquitectura surgida recientemente: el almacén de datos. Se trata de un repositorio se fuentes heterogéneas de datos, integrados y organizados bajo un esquema unificado para facilitar su análisis y dar soporte a la toma de decisiones. Esta tecnología incluye operaciones de procesamiento analítico en la línea, es decir, técnicas de análisis como pueden ser el resumen, la consolidación o la agregación, así como la posibilidad de ver la información desde distintas perspectivas.

Sin embargo, a pesar de que las herramientas soportan cierto análisis descriptivo y de sumarización que permite transformar los datos en otros datos agregados o cruzados de manera sofisticada, no generan reglas, patrones, pautas, es decir, conocimiento que pueda ser aplicado a otros datos. Sin embargo, en muchos contextos, como los negocios, la medicina o la ciencia, los datos por sí solos tienen un valor relativo.

Lo que de verdad es interesante es el conocimiento que puede inferirse a partir de los datos y, más aún, la capacidad de poder usar este conocimiento. Por ejemplo, podemos saber estadísticamente que el 10 por ciento de los ancianos padecen Alzheimer.

Esto puede ser útil, pero seguramente es mucho más útil tener un conjunto de reglas que a partir de los antecedentes y los hábitos y otras características del individuo nos digan si un paciente tendrá o no Alzheimer.Es encontrar relaciones entre ellos que nos proporcionen conocimiento mediante un sistema de inferencia.

Existen otras herramientas analíticas que han sido empleadas para analizar los datos y que tienen su origen en la estadística, algo lógico teniendo en cuenta que la materia prima de esta disciplina son precisamente los datos.

Aunque algunos paquetes estadísticos son capaces de inferir patrones a partir de los datos (utilizando modelización estadística paramétrica o no paramétrica), el problema es que resultan especialmente crípticos para los no estadísticos, generalmente no funcionan bien para la talla de las bases de datos actuales (cientos de tablas, millones de registros, talla de varios gigabytes y una alta dimensionalidad) y algunos tipos de datos frecuentes en ellos (atributos nominales con muchos valores, datos textuales, multimedia, etc.), y no se integran bien con los sistemas de información.

Todos estos problemas y limitaciones de las aproximaciones clásicas han hecho surgir la necesidad de una nueva generación de herramientas y técnicas para soportar la extracción de conocimiento útil desde la información disponible, y que se engloban bajo el enfoque y definición de la minería de datos.

La minería de datos se distingue de las aproximaciones anteriores por que no obtiene información extensional sino intensional y, además, el conocimiento no es, generalmente, una parametrización de ningún modelo preestablecido o intuido por el usuario, sino que es un modelo novedoso y original, conjuntos de reglas, ecuaciones, árboles de decisión, redes neuronales, grafos probabilísticos, análisis matricial.

Se define la minería de datos como el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos. Es decir, la tarea fundamental de la minería de datos es encontrar modelos inteligibles a partir de los datos. Para que este proceso sea efectivo debería ser automático o semi automático y el uso de los patrones descubiertos debería ayudar a tomar decisiones más seguras que reporten, por tanto, algún beneficio a la organización.

Por lo tanto, dos son los retos de la minería de datos: por un lado, trabajar con grandes volúmenes de datos, procedentes mayoritariamente de sistemas de información, con los problemas que ello conlleva, y por el otro usar técnicas adecuadas para analizar los mismos y extraer conocimiento novedoso y útil. En muchos casos la utilidad del conocimiento descubierto está íntimamente relacionada con la comprensibilidad del modelo inferido. No debemos olvidar que, generalmente, el usuario final no tiene por qué ser un experto en las técnicas de minería de datos, ni tampoco puede perder mucho tiempo interpretando los resultados. Por ello, en muchas aplicaciones es importante hacer que la información descubierta sea más comprensible por los humanos.

De una manera simplista pero ambiciosa, podríamos decir que el objetivo de la minería de datos es convertir datos en conocimiento. Este objetivo no es sólo ambicioso sino muy amplio.

Ahora hay que ver a qué tipo de datos podemos aplicar la minería de datos, en principio, puede aplicarse a cualquier tipo de información, siendo las técnicas de minería diferentes para cada una de ellas. En esta sección damos una breve introducción a algunos de estos tipos. En concreto, vamos a diferenciar entre datos estructurados provenientes de bases de datos relacionales, otros tipos de datos estructurados en bases de datos y datos no estructurados provenientes de la web o de otros tipos de repositorios de documentos.

3.3. Tecnología de base de datos

Una base de datos relacional es una colección de relaciones (tablas). Cada tabla consta de un conjunto de atributos(columnas o campos) y puede contener un gran número de tuplas (registros o filas). Cada tupla representa un objeto, el cual se describe a través de los valores de sus atributos y se caracteriza por poseer una clave única o primaria que lo identifica.

Una de las principales características de las bases de datos relacionales es la existencia de un esquema asociado, es decir, los datos deben seguir una estructura y son, por tanto, estructurados.

La integridad de los datos se expresa a través de las restricciones de integridad. Éstas pueden ser de dominio (restringen el valor que puede tomar un atributo respecto a su dominio y si puede tomar valores nulos o no), de identidad (por ejemplo la clave primaria que debe ser única) y referencial (los valores de las claves ajenas se deben corresponder con uno y solo un valor de la tabla referenciada).

Aunque las bases de datos relacionales (recogidas o no en un almacén de datos, normalizadas o estructuradas de una manera multidimensional) son la fuente de datos para la mayoría de aplicaciones de minería de datos, muchas técnicas de minería de datos no son capaces de trabajar con toda la base de datos, sino que sólo son capaces de tratar con una sola tabla a la vez.

Lógicamente mediante una consulta podemos combinar en una sola tabla o vista explorable aquella información de varias tablas que requiramos para cada tarea concreta de minería de datos. Por tanto, la presentación tabular, también llamada atributo valor, es la más utilizada por las técnicas de minería de datos.

En esta presentación tabular, e importante conocer los tipos de los atributos y, aunque en bases de datos existen muchos tipos de datos (enteros, reales, fechas, cadenas de texto, etc.), desde el punto de vista de las técnicas de minería de datos más habituales nos interesa distinguir sólo entre dos tipos, numéricos y categóricos. Para tratar otras representaciones mas complejas, como las cadenas de caracteres, los tipos textuales memo,

los vectores, y otras muchas, harán falta técnicas especificas.

- Los atributos numéricos contienen valores enteros reales. Por ejemplo, atributos como el salario o la edad.
- Los atributos categóricos o nominales toman valores en un conjunto finito y preestablecido de categorías. Por ejemplo, atributos como el sexo (H, M).

Incluso considerando únicamente estos dos tipos de datos, no todas las técnicas de minería de datos son capaces de trabajar con ambos tipos. Este hecho puede requerir la aplicación de un proceso previo de transformación o preparación de los datos.

3.3.1. Tipos de modelos

La minería de datos tiene como objetivo analizar los datos para extraer conocimientos. Este conocimiento puede ser en forma de relaciones, patrones o reglas inferidos de los datos.

Estas relaciones o resúmenes constituyen el modelo de los datos analizados. Existen muchas formas diferentes de representar los modelos y cada una de ellas determina el tipo de técnica que puede usarse para inferirlos.

En la práctica, los modelos pueden ser de dos tipos: predictivos y descriptivos. Los modelos predictivos pretenden estimar valores futuros o desconocidos de variables de interés, que denominamos variables objetivos o dependientes, usando otras variables o campos de la base de datos, a las que nos referiremos como variables independientes o predictivas.

Por ejemplo, un modelo predictivo sería aquel que permite estimar la demanda de un nuevo producto en función del gasto de publicidad.

Los modelos descriptivos, en cambio, identifican patrones que explican o resumen los datos, es decir, sirven para explorar las propiedades de los datos examinados, no para predecir nuevos datos.

Por ejemplo, una agencia de viaje desea identificar grupos de personas con unos mismos gustos, con el objeto de organizar diferentes ofertas para cada grupo y poder así remitirles esta información; para ello analiza los viajes que han realizado sus clientes e infiere un modelo descriptivo que caracteriza estos grupos.

Ahora regresemos un poco a la conceptualización de la minería de datos, ya que existen términos que se utilizan frecuentemente como sinónimos de la minería de datos y es necesario marcar la diferencia entre ellos. Iniciamos con el análisis de datos, que suele ser un mayor hincapié en las técnicas de análisis estadístico.

Otro termino muy utilizado, y el más relacionado con la minería de datos, es la extracción o descubrimiento de conocimiento en base de datos. De hecho, en muchas ocasiones ambos términos se han utilizado indistintamente, aunque existen claras diferencias entre los dos.

Así, últimamente se ha usado el término KDD para referirse a un proceso que consta de una serie de fases, mientras que la minería de datos es solo una de estas fases.

Se ha definido el KDD como el proceso no trivial de identificar patrones validos, novedosos, potencialmente útiles y, en última instancia, comprensibles a partir de los datos. En esta definición se resumen cuales deben ser las propiedades deseables del conocimiento extraído.

- Válido: hace referencia a que los patrones deben seguir siendo precisos para datos nuevos, y no sólo para aquellos que han sido usados en su obtención.
- Novedoso: que aporte algo desconocido tanto para el sistema y preferiblemente para el usuario.
- Potencialmente útil: la información debe conducir a acciones que reporten algún tipo de beneficio para el usuario.
- Comprensible: la extracción de patrones no comprensibles dificulta o imposibilita su interpretación, revisión, validación y uso en la toma de decisiones. De hecho, una información incomprensible no proporcionan conocimiento.

Como se deduce de la anterior definición, el KDD es un proceso complejo que incluye no sólo la obtención de los modelos o patrones, sino también la evaluación y posible interpretación de los mismos.

Así, los sistemas de KDD permiten la selección, limpieza, transformación y proyección de los datos; analizar los datos para extraer patrones y modelos adecuados; evaluar e interpretar los patrones para convertirlos en conocimiento; consolidar el conocimiento resolviendo posibles conflictos con conocimiento previamente extraído; y hacer el conocimiento disponible para su uso.

Esta definición del proceso clarifica la relación entre el KDD y la minería de datos: el KDD es el proceso global de descubrir conocimiento útil desde las bases de datos mientras que la minería de datos se refiere a la aplicación de los métodos de aprendizaje y estadísticos para la obtención de patrones y modelos.

3.3.2. Relación con otras disciplinas

La minería de datos es un campo multidisciplinario que se ha desarrollado en paralelo o como prolongación de otras tecnologías. Por ello, la investigación y los avances en la minería de datos se nutren de los que se producen en estas áreas relacionadas. Podemos destacar como disciplinas más influyentes las siguientes:

- Las bases de datos: conceptos como los almacenes de datos y el procesamiento analítico en línea tiene una gran relación con la minería de datos, aunque en este último caso no se trata de obtener informes avanzados a base de agregar los datos de cierta manera compleja pero predefinida, sino de extraer conocimiento novedoso y comprensible. Las técnicas de indización y de acceso eficiente a los datos son muy relevantes para el diseño de algoritmos eficientes de minería de datos.

- La recuperación de información: consiste en obtener información desde datos textuales, por lo que su desarrollo histórico se ha basado en el uso efectivo de bibliotecas y en búsqueda por internet. Una tarea típica es encontrar documentos a partir de palabras claves, lo cual puede verse como un proceso de clasificación de los documentos en función de estas palabras clave. Para ello se usan medidas de similitud entre los documentos y la consulta. Muchas de estas medidas se han empleado en aplicaciones más generales de minería de datos.

- La estadística: esta disciplina ha proporcionado muchos de los conceptos, algoritmos y técnicas que se utilizan en minería de datos, como por ejemplo, la medida, la varianza, las distribuciones, el análisis univariante y multivariante, la regresión lineal y no lineal, la teoría del muestreo, la validación cruzada, la modelización paramétrica y no paramétrica, las técnicas bayesianas, etc. De hecho, algunos paquetes de análisis estadístico se comercializan como herramientas de minería de datos.

- La visualización de datos: el uso de técnicas de visualización permite al usuario descubrir, intuir o entender patrones que serían más difíciles de ver a partir de descripciones matemáticas o textuales de los resultados. Existen técnicas de visualización, como, por ejemplo, las gráficas, las icónicas, las basadas en pixeles, las jerárquicas y muchas otras.

3.3.3. Proceso de descubrimiento de conocimiento

La existencia de voluminosas bases de datos conteniendo grandes cantidades de datos, que exceden en mucho las capacidades humanas de reducción y análisis a fin de obtener información útil, actualmente son una realidad en muchas organizaciones.

Debido a esto, frecuentemente, las decisiones importantes se toman en base a la intuición y experiencia del decisor más que considerando la rica información almacenada. Esta situación se intenta solucionar a través del proceso de KDD, el cual implica la realización de tres etapas: preprocesamiento, minería de datos (dataminig), y postprocesamiento, que se explican brevemente a continuación.

- La etapa de preprocesamiento tiene por objetivo preparar los datos para que puedan ser sometidos a la etapa siguiente del proceso. Dentro de las técnicas para realizar el preprocesamiento cabe mencionar : limpieza de datos, a fin de remover ruido e inconsistencias; integración de datos, para generar un único almacén de datos coherente

en aquellos casos donde los datos provienen de diferentes fuentes; transformaciones de datos, para normalizarlos; y reducción de datos, a fin de reducir el tamaño de los datos, por ejemplo, eliminando características redundantes. La importancia del preprocesamiento de los datos se debe a que la calidad de los datos sobre los que se aplican técnicas de KDD impacta de manera directa en la calidad del conocimiento que se descubre a partir de ellos.

- La etapa de minería de datos puede definirse sobre la base de un conjunto de primitivas diseñadas especialmente para facilitar un descubrimiento de conocimientos eficiente y fructífero. Tales primitivas incluyen: la especificación de las porciones de la base de datos o del conjunto de datos en los que se quiere trabajar; la clase de conocimiento a ser descubierto; los conocimientos existentes que podrían resultar útiles para guiar el proceso de KDD; las métricas de interés para llevar a cabo la evaluación de patrones en los datos analizados; y finalmente, las formas en que el conocimiento descubierto podría ser visualizado.

- La etapa de postprocesamiento implica la realización de algún tipo de reformulación de los resultados obtenidos producto de la minería de datos realizada. Se pretende, así, que los conocimientos encontrados sean más fáciles de entender y utilizar por el usuario a quien finalmente están destinados.

Capítulo 4

Sistemas Distribuidos

Los sistemas distribuidos están diseñados para que muchos usuarios trabajen en forma conjunta, lo cual difiere de los sistemas paralelos, ya que ellos están diseñados para lograr la máxima rapidez en un único problema.

En general un sistema distribuido es en el cual existen varios CPU conectados entre sí que cumplen con las siguientes características:

- Los distintos CPU trabajan de manera conjunta.
- Se debe distribuir la carga de trabajo en todas las máquinas.
- La falla de una de ellas no afectara a las demás.
- Pueden añadirse procesadores al sistema, permitiendo un desarrollo gradual.

Existen algunas desventajas de los sistemas distribuidos, en cuanto a el diseño, implantación y uso del software distribuido, ya que se presentan ciertos inconvenientes. Otro problema potencial tiene que ver con las redes de comunicación, ya que se deben considerar los problemas debidos a pérdidas de mensajes, saturación en el tráfico, expansión, etc. El hecho de que sea facíl compartir los datos es una ventaja pero se puede convertir en un gran problema, por lo que la seguridad debe organizarse adecuadamente.

Todos los sistemas distribuidos constan de varios CPU, organizados de diversas formas, especialmente respecto de la forma de interconectarlas entre sí y los esquemas de comunicación utilizados.

Existen diversos sistemas de clasificación para los sistemas con varios CPU, uno de los más conocidos es la taxonomia de Flynn, considera como características esenciales el número de flujo de instrucciones y el número de flujos de datos.

La clasificación incluye equipos:

- SISD(un flujo de instrucciones y un flujo de datos)
- SIMD(un flujo de instrucciones y varios flujos de datos)
- MISD(varios flujos de instrucciones y un flujo de datos)
- MIMD(varios flujos de instrucciones y varios flujos de datos)

Todos los sistemas distribuidos son de este tipo. Un avance sobre la clasificación de Flynn incluye la división de las computadoras MIMD en dos grupos:

- Multiprocesadores: poseen memoria compartida, es decir los distintos procesadores comparten el mismo espacio de direcciones virtuales.
- Multicomputadoras: no poseen memoria compartida, como ejemplo un grupo de máquinas conectadas mediante una red.

Cada una de las categorías indicadas se puede clasificar según la arquitectura de la red de interconexión en:

- Esquema de bus.-Es donde existe una sola red, bus, cable u otro medio que conecta todas las máquinas. squema de conmutador.- No existe una sola columna vertebral de conexión, hay múltiples conexiones y varios patrones de conexionado, los mensajes se mueven a través de los medios de conexión, se decide explícitamente la conmutación en cada etapa para dirigir el mensaje a lo largo de uno de los cables de salida.

  Otro aspecto de la clasificación considera el acoplamiento entre los equipos:

  - Sistemas fuertemente acoplados: donde el retraso al enviar un mensaje de una máquina a otra es corto y la tasa de transmisión es alta, generalmente se utiliza en sistemas paralelos.
  - Sistemas débilmente acoplados: donde el retraso entre las máquinas es grande y la tasa de transmisión es baja, generalmente se utiliza como sistemas distribuidos.

  Los sistemas distribuidos en cuanto al software se pueden clasificar en dos tipos:

  - Débilmente acoplados.
  - Fuertemente acoplados.

  El software débilmente acoplado de un sistema distribuido:

  - Permite independencia entre las máquinas en lo fundamental.
  - Facilita que interactúen en cierto grado cuando sea necesario.
  - Los equipos individuales se distinguen fácilmente.

Combinando los distintos tipos de hardware distribuido con software distribuido se logran distintas soluciones una de ellas es el Software débilmente acoplado en hardware débilmente acoplado, que es una solución muy utilizada usando una red de estaciones de trabajo conectadas mediante una LAN. Cada usuario tiene una estación de trabajo para su uso exclusivo con su propio S. O., la mayoría de los requerimientos se resuelven localmente.

Otra solución es tener un sistema de archivos global compartido, accesible desde todas las estaciones de trabajo, donde una o varias máquinas soporten al sistema de archivos. Los servidores de archivos aceptan solicitudes de los programas usuarios, las solicitudes se examinan, se ejecutan y la respuesta se envía de regreso, generalmente tienen un sistema jerárquico de archivos.

NFS es uno de los más conocidos y aceptados, fue desarrollado por Sun Microsystems, y algunas de sus características mas interesantes son su arquitectura, protocolo e implantación.

En cuanto a los aspectos claves en el diseño de un sistema distribuido estan los siguientes:

- Transparencia: Trata de lograr la imagen de un único sistema, tal que se pueda percibir que la colección de máquinas conectadas son un sistema de tiempo compartido de un solo procesador, esto se logra cuando los pedidos de los usuarios se satisfacen con ejecuciones en paralelo en distintas máquinas.
- Confiabilidad: significa que si una máquina falla, alguna otra debe encargarse del trabajo. La disponibilidad se mejora con un diseño que no exija el funcionamiento simultáneo de un número sustancial de componentes críticos y con la redundancia, es decir la duplicidad de componentes clave del hardware y del software.

Dentro de la plataforma de MAC OSX existe una solución para el manejo de objetos distribuidos, aquí esta su definición, tomada de un documento que explica los tópicos de los objetos distribuidos en el lenguaje objetive-C.

The Objective-C runtime supports an interprocess messaging solution called distributed objects. This mechanism enables a Cocoa application to call an object in a different Cocoa application (or a different thread in the same application). The applications can even be running on different computers on a network.

En español significa que:

En tiempo de ejecución Objetive-C soporta una solución en el proceso de mensajería llamado Objetos Distribuidos. Este mecanismo habilita una aplicación de cocoa para llamar un objeto desde otra aplicación de cocoa. Las aplicaciones pueden ser ejecutadas en diferentes computadoras sobre una red[4].

Los objetos distribuidos funcionan de forma que se puede disponer de ciertos objetos públicos, es decir un objeto al cual procesos clientes pueden acceder. Una vez que la conexión se realiza, el proceso cliente invoca uno de los métodos del objeto público, como si el proceso existiera en el proceso cliente, la sintaxis no cambia.

Cocoa y el sistema de ejecución de objetive-C se encargan de la transmisión de datos entre los procesos. En la figura se muestran los objetos que son usados en un sistema de objetos distribuidos y como un mensaje pasa del objeto cliente al objeto servidor.

El objeto público del proceso servidor es conectado mediante el objeto NSConnection el cual contiene un objeto NSPort. El puerto puede ser registrado con un objeto NSPortNameServer para un acceso fácil desde los procesos clientes. Los procesos clientes se conectan a los objetos públicos por su propio objeto NSConnection al objeto NSPort del servidor como lo muestra la figura4.

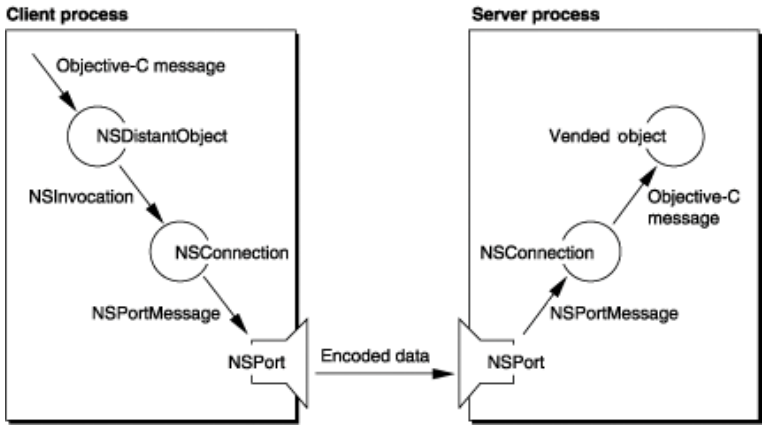

Figura 4.1: Manejo de objetos distribuidos en la plataforma Mac OS X, Ref. http://developer.apple.com

El objeto proxy es instanciado de NSDistantObject. Cuando el proceso cliente envía un mensaje al objeto NSDistantObject, el proxy captura el mensaje de objetive-C en forma de un objeto NSInvocation y muestra el objeto NSConnection. El objeto NSConnection codifica la NSInvocation dentro de un objeto NSPortMenssage, usando un objeto NSPortCoder, y lo pasa a un objeto NSPort conectado a un objeto NSPort en el proceso servidor.

Los objetos distribuidos de cocoa habilitan objetos en diferentes hilos y tareas, sobre diferentes máquinas, con transparencia en el envio de los mensajes de cada uno hacia el otro. Hay muchos caminos para comunicar las tareas entre ellas, los cuales trabajan como objetos distribuidos ocultos en el mecanismo estándar del lenguaje Objetive-C. La sintaxis de los mensajes remotos y locales es la misma.

Los mensajes remotos pueden ser enviados sincrónicamente y asincrónicamente. Cuando enviamos un mensaje síncronamente, el que envía espera la respuesta, bloqueando su ejecución, justamente como un mensaje local. Cuando se envía un mensaje asíncronamente, el que envía continua con su ejecución sin esperar la respuesta, cualquier respuesta del objeto remoto es ignorada.

Los objetos distribuidos pueden ser usados para dividir complejas tareas dentro de código separado en diferentes direcciones que pueden ejecutarse independientemente, pero puede permanecer cooperando como si estuvieran corriendo juntos.

Por ejemplo, una aplicación puede ser dividida dentro de la parte gráfica y las partes computacionales. Donde la parte gráfica puede aceptar las entradas del usuario y la parte computacional generará los cálculos necesarios.Como las dos partes funcionan de manera independiente no importa que los cálculos requieran mucho tiempo, la parte gráfica seguirá funcionando.

Los objetos distribuidos pueden también ser usados para implementar computación distribuida, o proceso paralelo. Si un trabajo es muy largo podemos dividirlo dentro de trabajos pequeños en múltiples máquinas. Los objetos distribuidos pueden simplificar las arquitecturas de las aplicaciones y la comunicación entre las partes distribuidas.

Antes de que un objeto pueda recibir mensajes de otro procesador, este debe localizar un puerto y habilitarlo con los otros. Cocoa proporciona acceso a los nombres de los servidores para soportar cada tipo de comunicación. Un objeto puede registrar su puerto con un nombre para su servidor con el cual los demás procesos podrán encontrarlo.

Se pueden controlar algunos factores de comunicación de los objetos distribuidos configurando los objetos NSConnection. Se puede limitar el tiempo de la conexión para esperar un mensaje remoto.

4.1. Definición del cluster de visualización

El término cluster se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora. Hoy en día juegan un papel importante en la solución de problemas de las ciencias, las ingenierías y del comercio moderno.

La tecnología de clusters ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y software de misiones críticas, servidores Web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos. El cómputo con clusters surge como resultado de la convergencia de varias tendencias actuales que incluyen la disponibilidad de microprocesadores económicos de alto rendimiento y redes de alta velocidad, el desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, así como la creciente necesidad de potencia computacional para aplicaciones que la requieran.

Simplemente, cluster es un grupo de múltiples ordenadores unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único ordenador, más potente que los comunes de escritorio. Los clusters son usualmente empleados para mejorar el rendimiento y la disponibilidad por encima de la que es provista por una sola computadora, típicamente siendo más económico que computadoras individuales de rapidez y disponibilidad comparables. De un cluster se espera que presente combinaciones de los siguientes servicios:

- Alto rendimiento (High Performance).
- Alta disponibilidad (High Availability).
- Equilibrio de carga (Load Balancing).
- Escalabilidad (Scalability).

La construcción de los ordenadores del cluster es más fácil y económica debido a su flexibilidad: pueden tener todos la misma configuración de hardware y sistema operativo (cluster homogéneo), diferente rendimiento pero con arquitecturas y sistemas operativos similares (cluster semi-homogéneo), tener diferente hardware y sistema operativo (cluster heterogéneo) , lo que hace más fácil y económica su construcción.

Para que un cluster funcione como tal, no basta sólo con conectar entre sí los ordenadores, sino que es necesario proveer un sistema de manejo del cluster, el cual se encargue de interactuar con el usuario y los procesos que corren en él para optimizar el funcionamiento.

Las aplicaciones paralelas escalables requieren: buen rendimiento, baja latencia,

comunicaciones que dispongan de gran ancho de banda, redes escalables y acceso rápido a archivos. Un cluster puede satisfacer estos requerimientos usando los recursos que tiene asociados a él. Los clusters ofrecen las siguientes características a un costo relativamente bajo:

- Alto Rendimiento (High Performance).
- Alta Disponibilidad (High Availability).
- Alta Eficiencia (High Throughput).
- Escalabilidad (Scalability).

La tecnología cluster permite a las organizaciones incrementar su capacidad de procesamiento usando tecnología estándar, tanto en componentes de hardware como de software que pueden adquirirse a un costo relativamente bajo.

4.1.1. Clasificación de los Clusters

El término cluster tiene diferentes connotaciones para diferentes grupos de personas. Los tipos de clusters, establecidos en base al uso que se dé a los clusters y los servicios que ofrecen, determinan el significado del término para el grupo que lo utiliza. Los clusters pueden clasificarse con base en sus características. Se pueden tener clusters de alto rendimiento (HPC High Performance Clusters), clusters de alta disponibilidad (HA High Availability) o clusters de alta eficiencia (HT High Throughput).

- High Performance: Son clusters en los cuales se ejecutan tareas que requieren de gran capacidad computacional, grandes cantidades de memoria, o ambos a la vez. El llevar a cabo estas tareas puede comprometer los recursos del cluster por largos periodos de tiempo.
- High Availability: Son clusters cuyo objetivo de diseño es el de proveer disponibilidad y confiabilidad. Estos clusters tratan de brindar la máxima disponibilidad de los servicios que ofrecen. La confiabilidad se provee mediante software que detecta fallos y permite recuperarse frente a los mismos, mientras que en hardware se evita tener un único punto de fallos.
- High Throughput: Son clusters cuyo objetivo de diseño es el ejecutar la mayor cantidad de tareas en el menor tiempo posible. Existe independencia de datos entre las tareas individuales. El retardo entre los nodos del cluster no es considerado un gran problema.

Los clusters pueden también clasificar como Clusters de IT Comerciales (High Availability, High Throughput) y Clusters Científicos (High Performance). A pesar de las discrepancias a nivel de requerimientos de las aplicaciones, muchas de las características de las arquitecturas de hardware y software, que están por debajo de

las aplicaciones en todos estos clusters, son las mismas. Más aun, un cluster de determinado tipo, puede también presentar características de los otros.

4.1.2. Componentes de un Cluster

En general, un cluster necesita de varios componentes de software y hardware para poder funcionar, estos son los siguientes:

- Nodos
- Conexiones de Red

Nodos

Pueden ser simples ordenadores, sistemas multi procesador o estaciones de trabajo (workstations). En informática, de forma muy general, un nodo es un punto de intersección o unión de varios elementos que confluyen en el mismo lugar.

Ahora bien, dentro de la informática la palabra nodo puede referirse a conceptos diferentes según el ámbito en el que nos movamos.

Si hablamos de redes de computadoras cada una de las máquinas es un nodo, y si la red es Internet, cada servidor constituye también un nodo.

También tenemos el caso de estructuras de datos dinámicas donde, un nodo es un registro que contiene un dato de interés y al menos un puntero para referenciar (apuntar) a otro nodo. Si la estructura tiene sólo un puntero, la única estructura que se puede construir con él es una lista, si el nodo tiene más de un puntero ya se pueden construir estructuras más complejas como árboles o grafos.

El cluster puede estar conformado por nodos dedicados o por nodos no dedicados.

En un cluster con nodos dedicados, los nodos no disponen de teclado, mouse ni monitor y su uso está exclusivamente dedicado a realizar tareas relacionadas con el cluster.

Mientras que, en un cluster con nodos no dedicados, los nodos disponen de teclado, mouse y monitor y su uso no está exclusivamente dedicado a realizar tareas relacionadas con el cluster, el cluster hace uso de los ciclos de reloj que el usuario del computador no está utilizando para realizar sus tareas.

Cabe aclarar que a la hora de diseñar un Cluster, los nodos deben tener características similares, es decir, deben guardar cierta similaridad de arquitectura y sistemas operativos.

Si el cluster generado tiene nodos totalmente heterogéneos (existe una diferencia grande entre capacidad de procesadores, memoria, HD)será ineficiente debido a que el middleware delegará o asignará todos los procesos al Nodo de mayor capacidad de Computo y solo distribuirá cuando este se encuentre saturado de procesos, por eso es recomendable construir un grupo de ordenadores lo mas similares posible.

Conexiones de Red

Los nodos de un cluster pueden conectarse mediante una simple red Ethernet con placas comunes (adaptadores de red o NICs), o utilizarse tecnologías especiales de alta velocidad como Fast Ethernet, Gigabit Ethernet, Myrinet, Infiniband, SCI, etc.

Las redes Ethernet son las redes más utilizadas en la actualidad, debido a su relativo bajo costo. No obstante, su tecnología limita el tamaño de paquete, realizan excesivas comprobaciones de error y sus protocolos no son eficientes, y sus velocidades de transmisión pueden limitar el rendimiento de los Clusters.

Para aplicaciones con paralelismo de grano grueso puede suponer una solución acertada.

La opción más utilizada en la actualidad es Gigabit-Ethernet (1000Mbps), siendo emergente la solución 10Gigabit-Ethernet. La latencia de estas tecnologías está en torno a los 30-100 us, dependiendo del protocolo de comunicación empleado.

En todo caso, es la red de administración por excelencia, así que aunque no sea la solución de red de altas prestaciones para las comunicaciones, es la red dedicada a las tareas administrativas.

Para el control del clúster de visualización debemos generar una red local en la que estarán conectadas las máquinas.

Aprovechando que el servidor de video es una Mac Pro la cual contienen dos salidas a Ethernet, la primera la conectamos al servidor de red, con la dirección de darwin8.

La segunda salida Ethernet nos sirve para generar la red local de maquinas a las cuales les dimos la dirección 128.0.0.0 para el servidor, y el incremental para las demás máquinas.

La figura 4.1.2 , muestra la arquitectura de las redes usadas en este proyecto, donde

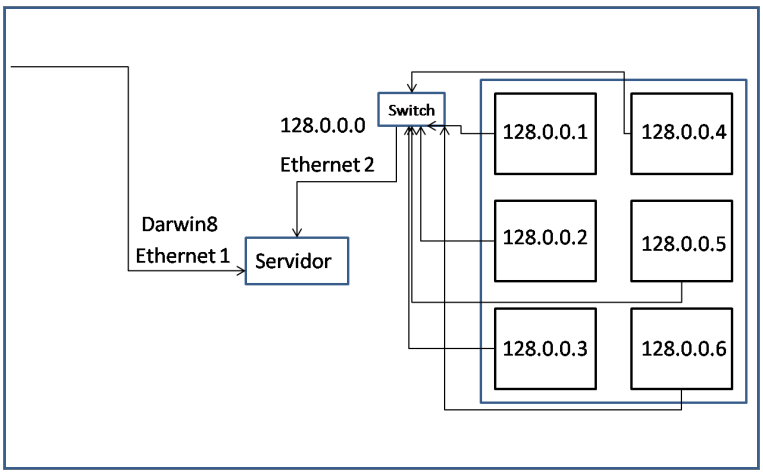

Figura 4.2: Arquitectura de la red local

la salida Ethernet 2 funciona como servidor de la red local, donde estarán conectadas las mac mini, la otra salida se usa para conectarse al servidor de la base de datos.

En esta parte es posible concluir que la manipulación de los objetos distribuidos que nos proporciona la plataforma Mac OS X, es una muy buena opción para el manejo de la pared de video, debido a las características antes mencionadas. La arquitectura que se emplea en la implantación de la red es una idea obtenida por rock cluster que es un proyecto que desde hace tiempo trabaja en el manejo de clusters[9].

4.1.3. Desempeño

Los sistemas de bases de datos pueden ser optimizados de acuerdo al desarrollo en el cual estén corriendo. Así, es Muy difícil dar una comparación en su desempeño sin tomar en cuenta la comfirguración y el ambiente de desarrollo. PostgreSQL y MySQL emplean varias tecnologías que pueden ser aprovechadas para mejorar su desempeño.

MySQL inicia con desarrollos enfocados en la velocidad mientras PostgreSQL se enfoca en el desarrollo de características y estándares.

Es por eso que MySQL es conocido frecuentemente por ser el más rápido de los dos. PostgreSQL con una configuración básica puede correr con un sistema con poca memoria. Aunque MySQL tiene un desempeño más veloz que PostgreSQL, no soporta algunas transacciones, y frecuentemente no garantiza durabilidad de los datos.

PostgreSQL puede comprimir y descomprimir sus datos de una forma veloz, almacenando la mayoría de los datos en poco espacio de disco. La ventaja de la compresión de datos, reside en la manera en la que guardamos los datos en el disco, hace que la lectura de los datos sea menor, lo cual resulta en una rápida lectura de datos.

PostgreSQL tiene una ventaja sobre MySQL ya que puede trabajar sobre los multicore. Otras ventajas de PostgreSQL contra MySQL es sobre la indización parcial que este permite. De hecho PostgreSQL en muchas medidas de desempeño muestra ser el mejor comparado con MySQL.

En cuanto a las características que tienen ambos, tenemos el incremento de la integridad de datos, funcionalidad, y desempeño. Las características incluidas en una base de datos pueden ser mejoradas por ellos.

Ambos soportan procedimientos almacenados, PostgreSQL es similar a Oracle y soporta también muchos procedimientos de otros lenguajes, incluyendo Python, Perl, TCL, Java.

PostgreSQL no tiene los tipos de datos de enteros sin signo, pero tiene unos tipos mucho más ricos que soportan muchos aspectos: fundamentos lógicos de los tipos booleanos, tipos definidos por el usuario.

Además permite usar columna de una tabla para definirla como un valor de arreglos multidimensionales. Los arreglos pueden ser de cualquier tipo. MySQL no tiene un tipo de datos para almacenar direcciones IP como PostgreSQL.

Dentro de los manejadores de bases de datos existentes, PostgreSQL, sobresale no solo por su distribución gratuita, sino además por sus bibliotecas que permiten un manejo muy bueno y una comunicación con múltiples lenguajes de desarrollo.

Para permitir que alguna otra máquina pueda acceder a la base de datos debemos modificar el archivo pghba.conf agregando la IP de la máquina que se quiera agregar dentro de las permitidas.

Este archivo se encuentra en la ruta /usr/local/pgsql/data y se debe tener privilegios de root para manipularlo. Una vez realizado el cambio hay que reiniciar el servicio de PostgeSQL.

Capítulo 5

Diseño

5.1. Diseño del sistema distribuido

El manejo de una pared de video es una cuestion que puede manejarse de diversas formas, en este caso se ha diseñado un sistema distribuido que permite su manipulación desde un servidor de visualización.

La manera en que trabaja este sistema es considerando que los intérpretes son un grupo de ususarios que quieren trabajar de manera conjunta, ya que cada uno de ellos tendra las funciónes que le permitirán el despliegue y la interpretación de los datos sobre la pared de video.

Se planea usar un grupo de máquinas que puede variar de 3 a 12, las cuales trabajarán de manera conjunta, quienes recibirán una carga de trabajo similar, donde la falla de alguna de ellas no permita la perdida de la información, así permitir la escalabilidad.

Se toma en cuenta que existen ciertas complicaciones en el diseño de los sistemas distribuidos, en cuanto a las redes de comunicación y software apropiado. Es por eso que se planea usar las ventajas que nos ofrece la plataforma MAC OS X mediante su solución de objetos distribuidos.

La forma en la que se planea tener el sistema es con multicomputadoras, es decir un grupo de máquinas conectadas mediante una red de Ethernet o por airport. En los proyectos mostrados en un inicio dentro del estado del arte hay uno que plantea una arquitectura de memoria compartida, la cual no es posible implementar en este diseño por que para eso deberiamos contar con otro tipo de infraestructura.

La solución propuesta en este caso consta de multicomputadoras que no poseen memoria compartida.

Este sistema se planea con un esquema conmutador, por que permitirá varios esquemas de conexión, en este caso alámbrico e inalámbrico, además se hará la elección explícita desde el servidor a cada intérprete.

Este sistema sera débilmente acoplado para permitir su desarrollo de manera no tan complicada en cuanto a la adquisición de los componentes hardware, para que sea una solución factible, sin un costo tan elevado, y flexible, evidentemente se tratará de minimizar en la medida de los posible el retraso de los mensajes enviados entre las máquinas.

La estratégia que se busca usar en este sistema en cuanto al manejo de archivos, no es la de un sistema de archivos global, debido a que esto puede verse reflejado en el tiempo de acceso de cada máquina, y en un retardo para todas. Es por eso que como antes se menciono la idea es que solo la inicialización requiera la mayoría de los datos, después el manejo solo sera por inidices.

En cuanto a la Transparencia del sistema, se planea que la manera en la que se maneje la pared de video sea como si fuera una sola pantalla, se tratará de lograr satisfacer las peticiones del usuario sobre la pared como si fuera una sola máquina.

Se planea tener un sistema confiable que cumpla con el requisisto de mantener la información sin importar en número de máquinas que fallen, aunque como ya se menciono con anterioridad, la falla de una máquina disminuye el área de visualización. Esta característica se cumple con la redundancia de información a desplegar la cual permite que su manejo sobre la pared sea de un valor numérico.

En este sistema se han definido los objetos visuales distribuidos, como objetos que pueden ser manipulados dentro de una extensión de pantallas, que además participen en la interpretación del gráfico. Además este objeto debe poder contener un objeto grafico, que acepte eventos del ratón, para facilitar su manipulación.

La idea es apoyarse de la base que proporciona Cocoa para el manejo de objetos distribuidos, agregando que sean además gráficos que acepten eventos del usuario.

El plan es unir este concepto de objetos distribuidos a otro concepto de objeto que permite el manejo de objetos gráficos de diversos tipos, para que de esta forma se permita el manejo de gráficos sobre la pared de video.

5.1.1. Gestíon remota distribuida

Como es necesario manipular 3 o más máquinas que no tienen dispositivo de entrada, es necesario el uso de una herramienta que permita su administración remota, esta herramienta permite realizar el manejo de las máquinas para suspenderlas, reiniciarlas, o apagarlas.

Permite también actualizar el software de todas las máquinas, así como ejecutar alguna aplicación, o simplemente su configuración desde el servidor.

El proceso de comunicación es el siguiente, primero debemos generar un programa cliente para instalar, indicando la IP del cliente y los privilegios del servidor sobre ellas, después se instala en las máquinas clientes.

Desde una herramienta que nos permite ubicar las máquinas desde su dirección IP, la seleccionamos y damos el nombre de usuario y contraseña que nos permita la conexión.

Ya que la máquina es habilitada para el uso del servidor, es posible controlarla, observarla o simplemente bloquearla. Es posible instalar programas automáticamente a un grupo formado por varias máquinas, o actualizar la información del cliente en cuanto a privilegios y conectividad.

5.2. Diseño de la minería visual

Si recordamos algunos conceptos vistos con anterioridad [**?**], es claro que la importancia de este sistema para la minería visual, no solo radica en el manejo del despliegue de la información, sino que además es importante verificar cual es la información que se debe mostrar y cual sería la mejor manera de visualizarla.

5.2.1. Coeficiente de correlación de Pearson

La correlación entre dos variables refleja el grado en que las puntuaciones están asociadas. La formulación clásica, conocida como correlación producto momento de Pearson, se simboliza por la letra griega ρ, cuando ha sido calculada en la población. Si se obtiene sobre una muestra, se designa por la letra r.

Este tipo de estadístico puede utilizarse para medir el grado de relación de dos variables si ambas utilizan una escala de medida a nivel de intervalo razón (variables cuantitativas).

Su fórmula es la siguiente:

$$\frac{\sum_{i=1}^{n} Z_{xj} Z_{yi}}{n-1}$$

La primera expresión se resuelve utilizando la covarianza y las desviaciones típicas de las dos variables (en su forma insesgada). La segunda forma se utiliza cuando partimos de las puntuaciones típicas empíricas.

Este estadístico, refleja el grado de relación lineal que existe entre dos variables. El resultado numérico fluctua entre los rangos de $+1$ a -1.

Una correlación de $+1$ significa que existe una relación lineal directa perfecta (positiva) entre las dos variables, como lo muestra la figura 5.2.1. Es decir, las puntuaciones bajas de la primera variable (X) se asocian con las puntuaciones bajas de la segunda variable (Y), mientras las puntuaciones altas de X se asocian con los valores altos de la variable Y.

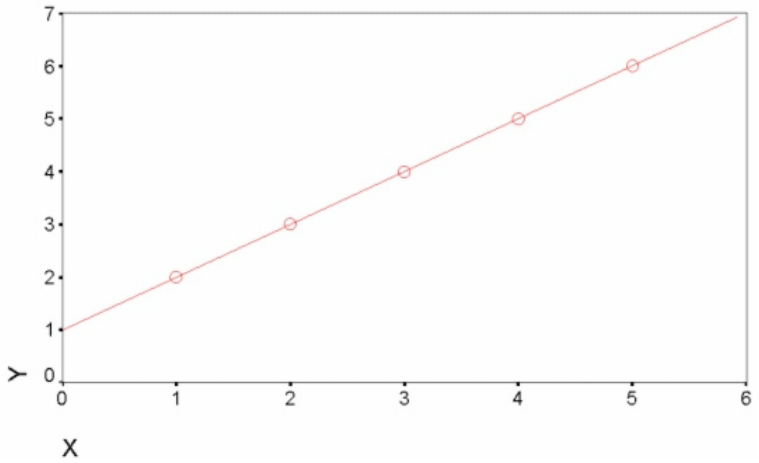

Figura 5.1: Correlación de Pearson con valor 1

Una correlación de -1 significa que existe una relación lineal inversa perfecta (negativa) entre las dos variables, como lo muestra la figura 5.2.1.

Lo que significa que las puntuaciones bajas en X se asocian con los valores altos en Y, mientras las puntuaciones altas en X se asocian con los valores bajos en Y.

Una correlación de 0 se interpreta como la no existencia de una relación lineal entre las dos variables estudiadas, como lo muestra la figura 5.2.1. El coeficiente de correlación posee las siguientes características :

- El valor del coeficiente de correlación es independiente de cualquier unidad usada para medir las variables.

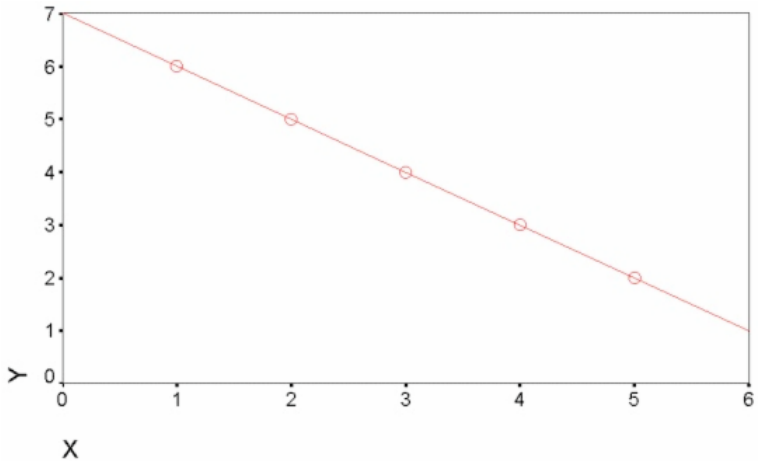

Figura 5.2: Correlación de Pearson con valor -1

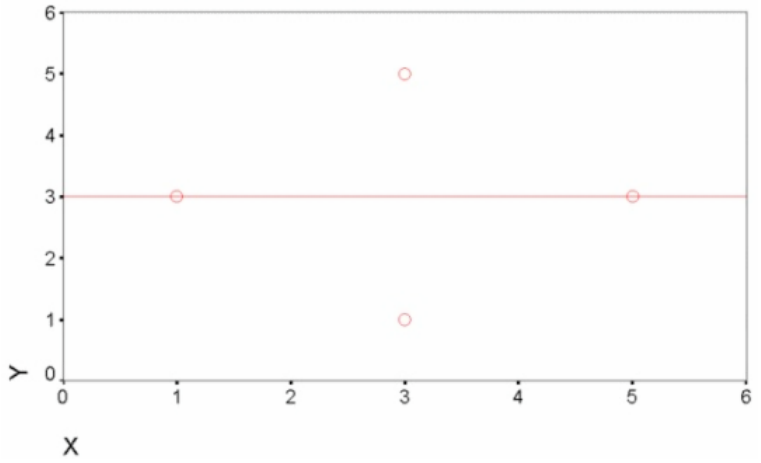

Figura 5.3: Correlación de Pearson con valor 0

- El valor del coeficiente de correlación se altera de forma importante ante la presencia de un valor extremo, como sucede con la desviación típica.

  Ante estas situaciones conviene realizar una transformación de datos que cambia la escala de medición y modera el efecto de valores extremos (como la transformación logarítmica).

- El coeficiente de correlación mide sólo la relación con una línea recta. Dos variables pueden tener una relación curvilínea fuerte, a pesar de que su correlación sea pequeña. Por tanto cuando analicemos las relaciones entre dos variables debemos representarlas gráficamente y posteriormente calcular el coeficiente de correlación.

- El coeficiente de correlación no se debe extrapolar más allá del rango de valores observado de las variables a estudio ya que la relación existente entre X e Y puede cambiar fuera de dicho rango.
- La correlación no implica causalidad. La causalidad es un juicio de valor que requiere más información que un simple valor cuantitativo de un coeficiente de correlación.
- El coeficiente de correlación de Pearson (r) puede calcularse en cualquier grupo de datos, sin embargo la validez del test de hipótesis sobre la correlación entre las variables requiere en sentido estricto :
  - Que las dos variables procedan de una muestra aleatoria de individuos.
  - Que al menos una de las variables tenga una distribucin normal en la población de la cual la muestra procede.

Para el cálculo válido de un intervalo de confianza del coeficiente de correlación de r ambas variables deben tener una distribución normal.

Si los datos no tienen una distribución normal, una o ambas variables se pueden transformar (transformación logarítmica) o si no se calcularía un coeficiente de correlación no paramétrico (coeficiente de correlación de Spearman) que tiene el mismo significado que el coeficiente de correlación de Pearson y se calcula utilizando el rango de las observaciones[11].

5.2.2. Interpretación de la correlación

El coeficiente de correlación como previamente se indicó oscila entre 1 y +1 encontrándose en medio el valor 0 que indica que no existe asociación lineal entre las dos variables a estudio.

Un coeficiente de valor reducido no indica necesariamente que no exista correlación ya que las variables pueden presentar una relación no lineal como puede ser el peso del recién nacido y el tiempo de gestación. En este caso el r infraestima la asociación al medirse linealmente. Los métodos no paramétricos estarían mejor utilizados en este caso para mostrar si las variables tienden a elevarse conjuntamente o a moverse en direcciones diferentes.

El significado de la estadística de un coeficiente debe tenerse en cuenta conjuntamente con la relevancia del fenómeno que estudiamos ya que coeficientes de 0,5 a 0,7 tienden ya a ser significativos. Es por ello muy útil calcular el intervalo de confianza del r ya que en muestras pequeñas tenderá a ser amplio.

La estimación del coeficiente de determinación (r2) nos muestra el porcentaje de la variabilidad de los datos que se explica por la asociación entre las dos variables.

Como previamente se indicó la correlación elevada y estadísticamente significativa no tiene que asociarse a causalidad. Cuando encontramos que dos variables están correlacionadas diversas razones pueden ser la causa de dicha correlación:

- Puede que X influencie o cause Y.
- Puede que Y influencie o cause X.
- X e Y pueden estar influenciadas por terceras variables que hace que se modifiquen ambas a la vez.

El coeficiente de correlación no debe utilizarse para comparar dos métodos que intentan medir el mismo evento, como por ejemplo dos instrumentos que miden la tensión arterial.

El coeficiente de correlación mide el grado de asociación entre dos cantidades pero no muestra el nivel de acuerdo o concordancia. Si los instrumentos de medida obtienen sistemáticamente cantidades diferentes uno del otro, la correlación puede ser 1 y su concordancia ser nula.

El coeficiente de correlación de Pearson es una medida de la mutua relación entre dos variables. El coeficiente de correlación muestral r es una expresión cuantitativa de la similitud comúnmente observada en la evolución de las variables.

El coeficiente de correlación muestral es una medida del grado de estrechez de la relación lineal entre 2 variables. Es necesario hacer notar dos propiedades por las cuales se toma como solución para este trabajo:

- R es un número sin unidades ni dimensiones, ya que la escala de su numerador y de su denominador son el producto de las escalas en que se miden ambas variables. Una consecuencia útil es que r puede calcularse a partir de valores codificados de ambas variables. No es necesario descodificar.
- R siempre cae entre -1 y +1. Los valores positivos de r indican una tendencia de aumento de x1 y x2 conjuntamente. Cuando r es negativa, entonces grandes valores de la primera variable están asociados con pequeños valores de la segunda variable.

5.2.3. Manejador de bases de datos PostgreSQL

Para mostrar por que fue seleccionado PostgreSQL para manejador de base de datos es necesario ver las características que proporcionan cada uno de ellos, es por eso que se hace una comparación entre el servidor de SQL, MySQL y PostgreSQL.

Dentro de las carácterísticas que son relevantes en el manejo de este tipo de servidores tenemos el sistema operativo que podemos usar para cada uno de ellos.

- Microsoft SQL Server 2005 trabaja solamente sobre los sistemas de windows XP y windows 2000. Es comercial, código fuente oculto, tiene varios niveles de carácterísticas en sus versiones.
  Requiere un alto consumo de tiempo en la instalación y en el mantenimiento, además tiene grandes recursos que realmente no son tan necesarios. Tiene una autenticación estándar.

- MySQL 5 funciona en windows, linux, unix y Mac. Usa un modelo de desarrollo cerrado. Nadie puede obtener código fuera del que se acepte por la licencia. Es decir se controla absolutamente todo el código y se mantienen como dueños de él. Fácil la instalación y el mantenimiento, inserción de valores en múltiples columnas. tiene una autenticación estándar.

- PostgreSQL funciona en windows, linux, unix, Mac. Código fuente libre. Es moderada la instalación y el mantenimiento. Permite índices funcionales, es decir índices basados en una función. Permite el uso de índices parciales, es decir uno puede crear in índice únicamente considerando los valores no nulos. Inserción de valores en múltiples columnas. Tienen una autenticación extensiva. Es posible regresar los valores deseados dependiendo del orden del cliente.

Es por lo anterior que elegir MySQL o PostgreSQL es una decisión que se debe tomar frecuentemente al buscar un open-source de gestión de bases de datos relacionales.

Ambas proporcionan soluciones que compiten fuertemente en el manejo de las bases de datos. MySQL ha sido durante mucho tiempo conocido como el más rápido, mientras que PostgreSQL se describe como una solución de open-source a la versión de Oracle.

MySQL ha sido popular entre los diversos proyectos de software debido a su velocidad, mientras que PostgreSQL es preferido por desarrolladores que provienen de un Oracle o SQL Server.

MySQL ha recorrido un largo camino en la adición de funcionalidad avanzada, mientras que PostgreSQL ha mejorado enormemente su velocidad en los últimos grandes lanzamientos. Muchos, sin embargo, no son conscientes de la convergencia y todavía se aferran a los estereotipos sobre la base de 4,1 MySQL y PostgreSQL 7,4. Si la comparación es entre las últimas versiones de InnoDB y PostgreSQL, PostgreSQL es a menudo más rápido.

5.2.4. Funciones de Conexión a la Base de Datos

Las siguientes rutinas permiten realizar una conexión al servidor de Postgres. El programa de aplicación puede tener abiertas varias conexiones a servidores al mismo tiempo. (Una razón para hacer esto es acceder a más de una base de datos). Cada conexión se representa por un objeto PGconn que se obtiene de PQconnectdb () o PQsetdbLogin ().

Nótese que estas funciones siempre devolverán un puntero a un objeto no nulo, a menos que se tenga demasiada poca memoria incluso para crear el objeto PGconn.

Se deberá llamar a la función PQstatus para comprobar si la conexión se ha realizado con éxito antes de enviar consultas a traves del objeto de conexión.

PQconnectdb Realiza una nueva conexión al servidor de base de datos. Esta rutina abre una conexión a una base de datos utilizando los parámetros que se dan en la cadena conninfo.

Contra lo que ocurre más abajo con PQsetdbLogin(), los parámetros fijados se pueden extender sin cambiar la firma de la función, por lo que el uso de PQconnectStart y PQconnectPoll resulta preferible para la programación de las aplicaciones.

La cadena pasada puede variar para utilizar así los parámetros de defecto, o puede contener uno o más parámetros separados por espacios. Cada fijación de un parámetro tiene la forma keyword = value. (Para escribir un valor nulo o un valor que contiene espacio vacio, se emplearán comillas simples.

Los espacios alrededor del signo igual son opcionales). Los parámetros reconocidos actualmente son:

- **Host**.- Nombre del ordenador al que se conecta. Si se da una cadena de longitud distinta de cero, se utiliza comunicación TCP/IP. El uso de este parámetro supone una búsqueda del nombre del ordenador.
- **Hostaddr**.-Dirección IP del ordenador al que se debe conectar. Deberá estar en el formato estándar de números y puntos, como se usan en las funciones de BSD y otras. Si se especifica una cadena de longitud distinta de cero, se emplea una comunicación TCP/IP.

  El uso de hostaddr en lugar de host permite a la aplicación evitar la búsqueda del nombre de la máquina, lo que puede ser importante en aplicaciones que tienen una limitación de tiempo. Sin embargo la autenticación Kerberos necesita el nombre de la máquina. En este caso se aplica la siguiente secuencia. Si se especifica host sin hostaddr, se forza la búsqueda del nombre de la máquina.

Si se especifica hostaddr sin host, el valor de hostaddr dará la dirección remota; si se emplea Kerberos, se buscará de modo inverso el nombre del ordenador. Si se dan tanto host como hostaddr, el valor de hostaddr dará la dirección remota; el valor de host se ignorará, a menos que se emplee Kerberos, en cuyo caso ese valor se utilizará para la autenticación Kerberos. Nótese que libpq fallará si se pasa un nombre de ordenador que no sea el nombre de la máquina en hostaddr.

Cuando no se empleen ni uno ni otro, libpq se conectará utilizando un socket de dominio local, usando lo siguiente:

- **Port**.-Número del puerto para la conexión en el ordenador servidor, o extensión del nombre de fichero del socket para conexión de dominio Unix.
- **Dbname**.-Nombre de la base de datos.
- **User**.-Nombre del usuario que se debe conectar.
- **Password**.-Password que se deberá utilizar si el servidor solicita una autenticación con password.
- **Options**.-Se pueden enviar las opciones Trace/debug al servidor.
- **Tty**.-Un fichero o tty para la salida de la depuración opcional desde el servidor. Si no se especifica ningún parámetro, se comprobarán las correspondientes variables de entorno.

Si no se encuentran fijadas, se emplerán los valores de defecto codificadas en el programa. El valor devuelto es un puntero a una estructura abstracta que representa la conexión al servidor.

Una vez que se ha establecido correctamente una conexión con un servidor de base de datos, se utilizan las funciones que se muestran a continuación para realizar consultas y comandos de SQL:

- PQexec Emite una consulta a Postgres y espera el resultado.
- PGresult Devuelve un puntero PGresult o, posiblemente, un puntero NULL. Generalmente devolverá un puntero no nulo, excepto en condiciones de "fuera de memoria"(out-of-memory) o errores serios tales como la incapacidad de enviar la consulta al servidor. Si se devuelve un puntero nulo, se deberá tratar de la misma forma que un resulado PGRESFATALERROR. Para conseguir más información sobre el error, utilice PQerrorMessage.

La estructura PGresult encapsula el resultado devuelto por el servidor a la consulta. Los programadores de aplicaciones con libpq deberán mostrarse cuidadosos de mantener la abstracción de PGresult. Prohiban la referencia directa a los campos de la estructura PGresult, porque están sujetos a cambios en el futuro. (Incluso a partir de la versión 6.4 de Postgres, ha dejado de proporcionarse la definición de

PGresult en libpq-fe.h.). PQresultStatus Devuelve la situación (status) resultante de una consulta.

5.2.5. Procesamiento Asíncrono de Consultas

La función PQexec es adecuada para emitir consultas en aplicaciones síncronas sencillas. Sin embargo, tiene una porción de definciencias importantes:

- PQexec espera hasta que se completa la consulta. La aplicación puede tener otro trabajo para hacer (como por ejemplo mantener una interfaz de usuario), en cuyo caso no se querrá bloquear esperando la respuesta.

  Una vez que el control se pasa a PQexec, la aplicación cliente tiene muy dificil intentar cancelar la consulta en curso. (Se puede hacer con un manipulador de señales, pero no de otra forma).

- PQexec sólo puede devolver una estructura PGresult. Si la cadena de la consulta emitida contiene múltiples comands SQL, se perderán todos excepto el último.

  Las aplicaciones que no se quieren encontrar con estas limitaciones, pueden utilizar en su lugar las funciones que subyacen bajo PQexec: PQsendQuery y PQgetResult.

Para los programas antiguos que tenían esta funcionalidad utilizando PQputline y PQputnbytes y esperaban bloqueados el envío de datos del servidor, se añade la función PQsetnonblocking.

Las aplicaciones antíguas pueden rechazar el uso de PQsetnonblocking y mantener el comportamiento anterior potencialmente bloquante. Los programas más nuevos pueden utilizar PQsetnonblocking para conseguir una conexión con el servidor completamente no bloqueante.

5.3. Diseño de los gráficos con OpenGL

OpenGL (Open Graphics Library) es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos.

Fundamentalmente OpenGL es una especificación, es decir, un documento que describe un conjunto de funciones y el comportamiento exacto que deben tener. Partiendo de ella, los fabricantes de hardware crean implementaciones, que son bibliotecas de funciones que se ajustan a los requisitos de la especificación, utilizando aceleración hardware cuando es posible.

Dichas implementaciones deben superar unas pruebas de conformidad para que sus fabricantes puedan calificar su implementación como conforme a OpenGL y para poder usar el logotipo oficial de OpenGL.

OpenGL tiene dos propósitos esenciales:

- Ocultar la complejidad de la interfaz con las diferentes tarjetas gráficas, presentando al programador una API única y uniforme.
- Ocultar las diferentes capacidades de las diversas plataformas hardware, requiriendo que todas las implementaciones soporten la funcionalidad completa de OpenGL (utilizando emulación software si fuese necesario).

El funcionamiento básico de OpenGL consiste en aceptar primitivas tales como puntos, líneas y polígonos, y convertirlas en píxeles. Este proceso es realizado por una pipeline gráfica conocida como la Máquina de estados de OpenGL.

La mayor parte de los comandos de OpenGL o bien emiten primitivas a la pipeline gr'áfica o bien configuran cómo la pipeline procesa dichas primitivas.

OpenGL es una API basada en procedimientos de bajo nivel que requiere que el programador dicte los pasos exactos necesarios para renderizar una escena. Esto contrasta con las APIs descriptivas, donde un programador sólo debe describir la escena y puede dejar que la biblioteca controle los detalles para representarla.

El diseño de bajo nivel de OpenGL requiere que los programadores conozcan en profundidad la pipeline gráfica, a cambio de darles libertad para implementar algoritmos gráficos novedosos.

OpenGL ha influido en el desarrollo de las tarjetas gráficas, promocionando un nivel básico de funcionalidad que actualmente es común en el hardware comercial; algunas de esas contribuciones son:

- Primitivas básicas de puntos, líneas y polígonos rasterizados[1] .
- Una pipeline de transformación e iluminación.-Es un término informático utilizado en graficación, empleado normalmente en el contexto de la aceleración hardware.

[1]Es el proceso por el cual una imagen descrita en un formato gráfico vectorial se convierte en un conjunto de pixeles o puntos para ser desplegados en un medio de salida digital

- Z-buffering.-Es la parte de la memoria de un adaptador de video encargada de gestionar las coordenadas de profundidad de las imágenes en los gráficos en tres dimensiones (3-D), normalmente calculados por hardware y algunas veces por software.
- Mapeado de texturas.- Se refiere a la utilización de texturas es determinante para dar una apariencia real al material del que estén constituidos los modelos de la escena. Una textura es una imagen que se pega a un modelo tridimensional de forma que parezca que forma parte del objeto.
- Alpha blending.-Es el proceso de combinación de una imagen con su fondo para crear la apariencia de una trasparencia parcial.

5.4. Diseño de los objetos distribuidos visuales

5.4.1. Objetos Visuales

Dentro del lenguaje de C-objetivo tenemos la ventaja de trabajar sólo con objetos, podemos aprovechar esta ventaja si tomamos en cuenta la clase que nos pueda permitir el máximo de manipulación posible. Esta clase es NSView, de la cual derivan la mayoría de los objetos usados en la interfaz.

Esta clase se caracteriza por permitir la generación de gráficos, servir como contenedor del mismo tipo de objetos y además aceptar eventos del usuario. Básicamente NSView es una clase abstracta que define dibujo básico, el manejo de eventos y la impresión de la arquitectura de una aplicación.

Típicamente no se interactúa directamente, sino que se usa una clase heredada, sobre escribiendo unos pocos métodos. Sus principales atributos son:

- Manejo de eventos.
- Despliegue integrado de pantalla e impresora.
- Sistema flexible de coordenadas.

Este tipo de objetos se organizan dentro de un objeto NSWindow. Un objeto NSView es un área rectangular, responsable de todos los dibujos y eventos del ratón que se producen en ella.

Además de las ventajas antes mencionadas, esta clase tiene una derivación que permite la manipulación de objetos creados con la biblioteca de Open GL. Con esto se planea tener facilidades para la creación y manipulación de objetos en 3D.

Dentro de la arquitectura inicial del sistema se contemplarón los siguientes componentes:

- Mac Pro (Procesador Xeon con núcleo cuádruple a 2.8 GHz , 12 MB de caché)
- 9 mini Mac (Procesador Core Duo de Intel a 1.83, 2 GB de memoria Ram, Procesador gráfico GMA 950 de Itel, Disco Duro de 160 GB, Ethernet 1 GB integrada)
- 1 switch (Ethernet 1 Gigabit)
- 9 pantallas (display 23 pulgadas, 1920x1200 pixeles, 2.25MP)

En la figura 6.1, se muestra la arquitectura en hardware del sistema. La canti-

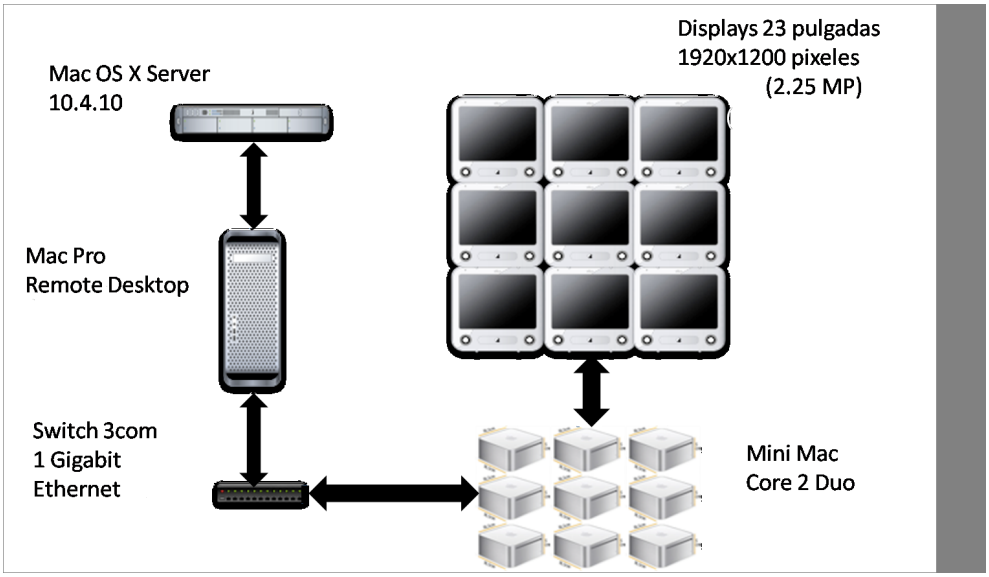

Figura 5.4: Arquitectura general del sistema

dad de máquinas puede variar ya que este prototipo es escalable, los modulos de software utilizados para el funcionamiento del sistema, estan divididos en dos bloques principales, el primero es el que se alojará en el servidor 5.4.1, es aquí donde se desarrollará el procesamiento de la correlación de Pearson, y es desde este módulo donde se envia la información que los módulos clientes 5.4.1, quienes van a permitir la generación y manipulación de los gráficos.

Figura 5.5: Modulos de software alojados en el servidor de video.

Otro software necesario:

Figura 5.6: Modulo alojado en cada cliente.

- Sistema Operativo MAC OSX 10.4
- Apple Remote Desktop
- Aplicación Servidor
- Aplicación Cliente

Capítulo 6

Implementación

6.1. Implementación del sistema distribuido

Para responder a las necesidades planteadas en este proyecto es necesario usar un sistema distribuido que permita el acoplamiento de las máquinas para trabajar en forma conjunta en el despliegue de información y permitir así el manejo de una pared de video.

Se implantó un sistema con 12 máquinas que trabajan de manera conjunta permitiendo la manipulación de la pared de video, bajo las ordenes de la máquina que trabaja como servidor de visualización, este sistema asigna una carga de trabajo igual a cada máquina, permite que la falla de alguna de ellas no provoque perdida en la información y permite la escalabilidad de manera casi inmediata.

Se implantó el sistema apoyandonos en la tecnología de la plataforma MAC OSX, mediante el uso de la solución que da para el manejo de objetos distribuidos, que en este caso se transformaron en objetos contenedores de imagen.

Existen clases especiales dentro del lenguaje de programación objetive-C que permiten la manipulación de objetos visuales, los cuales se implementarón junto con los objetos distribuidos para permitir la solución de manipulación de los objetos visuales sobre la pared de video.

El sistema de multicomputadoras inicial se implantó con 6 máquinas conectadas en una Lan con Ethernet, después se incremento el número de máquinas a 12 y las otras 6 entraron a la red mediante airport.

Es por eso que este sistema se implantó con el esquema de conmutador, incluyendo el manejo de los mensajes desde el servidor de visualización.

Esta implementación se considera un sistema débilmente acoplado que cubre las necesidades que en este momento se propone cumplir, para minimizar el retraso en los mensajes y tomando en cuenta que la tasa de transmisión es baja, se propone enviar en la medida de lo posible datos pequeños y significativos, que no requieran grandes cantidades de tiempo y generen retrasos significativos.

El manejo de archivos que se implantó en este proyecto no involucra un sistema de archivos global compartido, si no que cada mensaje contendrá ciertos valores que permitan el manejo de la pared de video.

El tipo de implantación que se hizo permite que el sistema sea un poco transparente, en cuanto al manejo, aun sin mantener un sistema de archivos compartido se logró un sistema que permita el manejo de la pared como si fuera una sola pantalla.

Este sistema cumple a su manera con la confiabilidad, ya que la replica de la información en cada máquina permite que aún con fallas, la información desplegada no pierda.

Se unio el concepto de dos clases principales de objetive-C, los objetos distribuidos y la clase NSView con lo cual podemos manejar las gráficas por toda el área de la pared de video

Dentro de las clases que permiten el manejo de gráficos se eligió NSView por que permite una manipulación más completa de su contenido.

El sistema permite la creación y manipulación de objetos en 3D, ya que los contenedores gráficos que se implementarón, permiten el uso de la biblioteca de OpenGL, en este caso los gráficos que se generán son de dos tipos de datos, es decir solo se muestran gráficas simples en el plano x,y.

Esta es la arquitectura que se implementó:

- Mac Pro (Procesador Xeon con núcleo cuádruple a 2.8 GHz , 12 MB de caché)
- 12 mini Mac (Procesador Core Duo de Intel a 1.83, 2 GB de memoria Ram, Procesador gráfico GMA 950 de Itel, Disco Duro de 160 GB, Ethernet 1 GB integrada)
- 1 switch (Ethernet 1 Gigabit)
- 12 pantallas (display 23 pulgadas, 1920x1200 pixeles, 2.25MP)

En la figura 6.1, se muestra la arquitectura implementada del sistema.

Figura 6.1: Arquitectura general del sistema

6.2. Implementación de la minería visual

6.2.1. Biblioteca Libpq

Para comunicarnos desde el sistema al manejador de bases de datos PostgreSQL, se decidió utilizar la biblioteca Libpq.a que es la que permite la conexión desde C++.

La consulta tiene por objetivo acceder a 18 atributos de cualquier tabla de la base de datos elegida.

Como el lenguaje de desarrollo seleccionado es C-objetivo, una opción viable que permite la conexión con PostgreSQL es generar un objeto que será de tipo C++, de esta forma podemos adquirir los datos desde este objeto y manipularlo después desde la interfaz, creada en Cocoa.

El proceso de comunicación inicia con la conexión a la base de datos, una vez que es aceptada por el servidor, se realizan las consultas a los atributos elegidos. Se obtienen todos los registros de cada atributo y se almacenan en un archivo para su uso posterior.

libpq es la interfaz para los programadores de aplicaciones en C para PostgreSQL. libpq es un conjunto de rutinas de biblioteca que permiten a los programas cliente trasladar consultas al servidor de Postgres y recibir el resultado de esas consultas. libpq es también el mecanismo subyacente para muchas otras interfaces de aplicaciones de PostgreSQL, incluyendo libpq++ (C++), libpgtcl (Tcl), Perl, y ecpg.

Los programas cliente que utilicen libpq deben incluir el fichero de cabeceras libpq-fe.h, y deben enlazarse con la biblioteca libpq.

6.2.2. Coeficiente de Pearson

Anteriormente se hablo de las ventajas del uso de la correlación de Pearson, ahora se explicará como se realizan los cálculos.

Primero debemos tomar en cuenta que obtenemos los datos de una base de datos, por lo que mantener los valores en memoria dinámica durante la obtención puede resultar en un desborde de pila, ya que recordemos que no se conoce la cantidad de valores con los que se tratarán.

Es por eso que se opta por generar archivos que contendrán los valores para cada variable, esto permite tener cierta seguridad en los datos. Una vez que terminan las consultas a la base de datos y tenemos la seguridad de contar con todos los datos, procedemos a generar arreglos que los contendrán, para realizar las operaciones.

Los cálculos que se realizan son en base al cuadrado del producto y al producto de la combinación de todas las variables, después se deben generar las medias para cada variable y todas se operan en un cociente que es quien genera el coeficiente que esperamos, los cálculos no son complicados pero si son repetitivos y requieren muchas evaluaciones debido a la cantidad de valores que se van a trabajar.

Una vez realizado el cálculo para cada una de las variables se procede a colorear la matriz que permitira saber cual par de variables tiene mayor correlación. El color intenso va a indicar una correlación completa y la baja intensidad significará correlación nula. Para que el usuario vea la gráfica que desee sobre las variables que elija, debe dar clic en el boton que intersecte ambos valores en la matriz.

6.3. Implementación de objetos distribuidos visuales

Un objeto distribuido visual, (ver figura 6.3) es aquel objeto distribuido que permite la contención de un objeto del mismo tipo dentro de su área de control, con todo lo que esto implica.

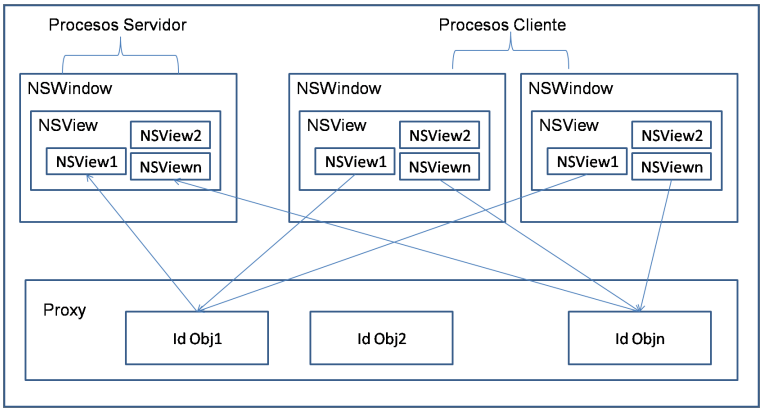

Figura 6.2: Comportamiento de un objeto visual distribuido

Dentro de la parte que incluye la comunicación con los objetos tenemos que mediante el protocolo, indicamos los dos métodos sobre los que necesitamos trabajar:

- Genera. Debe indicar, la posición inicial del objeto y los datos para generarlo.
- Coloca. Debe indicar, la coordenada en la que se encuentra actualmente el objeto visual distribuido, y cuál de todos es.

Dentro del método mueve, debemos notar que se debe saber sobre que objeto se está trabajando.

Además debemos actualizar continuamente su posición, mientras el ratón lo mueva.

La propuesta implica el envio en una ocasión de una cadena de caracteres, que se convertirá, dentro de cada cliente en una imagen, después de esta inicialización lo único que se necesita recibir es un entero indice del arreglo de los objetos visuales. Es así como se hará la manipulación. Esto ocasiona que dentro del método genera se deba generar el arreglo en el servidor y en los clientes.

El problema radica en cómo enviar la petición, como enviar el objeto o solo los datos. Esperar la respuesta y replicar la información a todos los clientes. La evidencia indica que la mejor opción es generar cada objeto visual en los clientes, pero por otro lado el gasto en enviar arreglos grandes es un tema para reflexionar.

6.4. Sistema

Este sistema va a permitir la manipulación de una gran variedad de bases de datos, permitirá además la manipulación de 18 variables como mínimo. El flujo del sistema se puede denotar bajo los 4 módulos que se ejecutan uno después de otro, con cierta dependencia, por ejemplo el módulo conecta debe ser ejecutado obligatoriamente antes que la correlación, así como el método coloca se ejecuta siempre después de haber utilizado el método genera en al menos una ocasión, como se muestra en la figura6.4. El método conecta debe tener

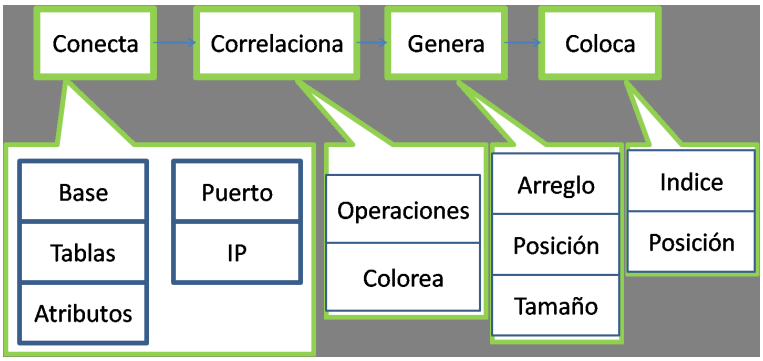

Figura 6.3: Flujo del manejo de los métodos del sistema

como atributos el nombre de la base, las tablas y los atributos que se desean correlacionar. Dentro de correlaciona se tendrán las 18 variables y se realizarán las operaciones y el coloreado de los botones, dependiendo del resultado de la correlación.

Dentro del método genera se va a crear un arreglo que contendrá los objetos visuales, para la generación de estos objetos debemos tener su tamaño, los arreglos de los valores para el gráfico y la posición de origen. El método coloca es el que permitirá el movimiento de los objetos sobre el área total de despliegue, y requiere como parámetros el índice y la posición donde se moverá el objeto visual.

6.5. Definición del sistema

Es importante mostrar el manejo de los apuntadores desde el control hacia el objeto visual y viceversa, así como del contenedor visual hacia los objetos distribuidos visuales, los cuales cabe mencionar se generan dinámicamente.

El hecho de que un objeto se genere desde el inicio de la aplicación permite que sea más sencillo generar un apuntador que nos permita su manejo. Cabe mencionar que dentro de C-objetivo contamos con un apuntador genérico que nos permite la manipulación de todos los objetos que se requieren como es el caso del objeto control, del objeto ventana, del objeto visual, del objeto boton, etc (dichos objetos se muestran en la figura6.5).

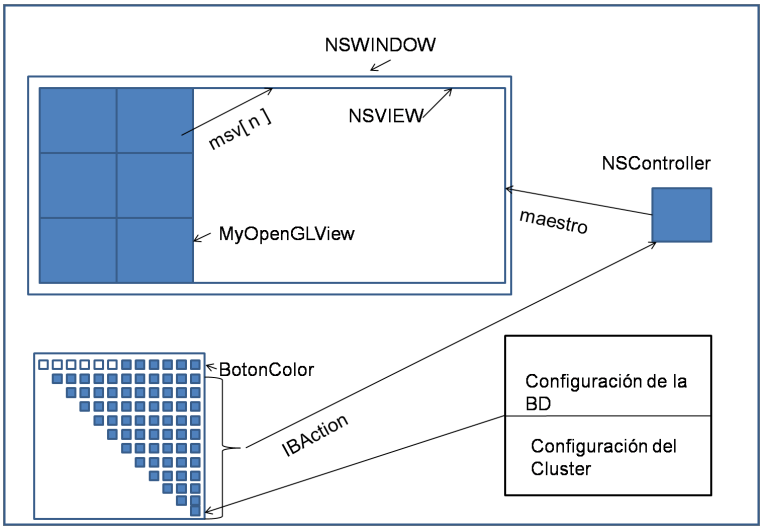

Figura 6.4: Objetos que conforman la vista principal

Tomando en cuenta que la manipulación de los objetos se realizará en una pantalla mucho menor que el área de la pared de video, es necesario definir una escala para el sistema de coordenadas, dicha escala está dada por un cuarto del total de las pantallas, es decir el área total de las pantallas es 1920x1200 por las 3 pantallas, la proporción está decidida a ser un cuarto del total, por lo que en el servidor tendremos las mismas 3 áreas pero de 480x300 pixeles como se muestra en la figura 6.5.

Además se debe definir el área entre las tres pantallas, para que cada despliegue sepa cuándo debe mostrar el objeto, esto se realiza mediante una resta a los valores recibidos desde el servidor, los cuales tiene la forma de coordenada es decir poseen un valor x y un valor y, como nuestra pared está formada

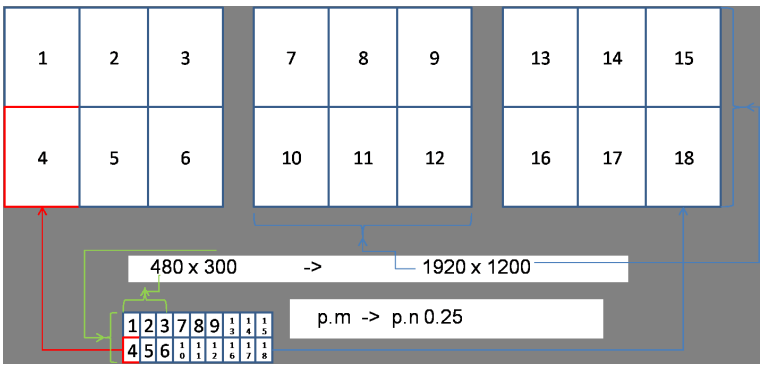

Figura 6.5: Cambio de coordenadas de la pantalla principal a la pared de video

por 3 pantallas de forma horizontal debemos restar el valor de cada una en el eje x, por ejemplo para la máquina uno vamos a tener que x debe ser menor o igual a 480 para el área del servidor, para el 2 x se encuentra acotada de la siguiente forma, debe ser mayor a 480 y menor a 960, y así sucesivamente para las máquinas que continúen, en forma práctica podemos restar el valor de 480 y 960 respectivamente para que desplieguen las máquinas cuando les corresponda su turno.

6.5.1. Generación de gráficos en Mac

El modelo de pintura en Cocoa, para imágenes consiste en que cada operación de dibujo se aplica como una capa de pintura a un lienzo de salida. A medida que nuevas capas de pintura se añaden, el dibujo puede irse obscureciendo. Este modelo permite la construcción de imágenes sofisticadas con un número pequeño de primitivas.

El medio ambiente del dibujo, lienzo y ajustes de los gráficos determinan la vista final contenida en el objeto. El lienzo de dibujo determina donde estará contenido el dibujo y mientras, el control del gráfico se ocupa del tamaño, color y calidad. El contexto del gráfico se puede pensar como el destino del dibujo.

Un contexto de gráfico encapsula toda la información necesaria para el dibujo sobre el lienzo incluyendo los atributos del dibujo y la representación del dibujo sobre el lienzo. El objeto que lo representa es de la clase NSGraphicsContext. Para la mayoría de las operaciones de dibujo, Cocoa crea el contexto del gráfico según sea necesario y lo configura antes de llamar el código del dibujo[6][5].

Todas las coordenadas son especificadas usando valores de punto flotante, su código dibuja en el espacio de coordenadas del usuario. Los comandos de dibujo

hacen la conversión al espacio de coordenadas del artefacto donde se hará la interpretación. Como el nombre lo implica, el espacio de coordenadas del artefacto se refiere al espacio de coordenadas del intérprete.

6.5.2. Generación de gráficos OpenGL

Una clase que deriva directamente de la clase NSView, es OpenGLView, que permite la manipulación de objetos en 2d y 3d, mediante las sentencias de OpenGL. OpenGL es una poderosa interface de software para hardware gráfico, desarrollada por silicon graphics.

OpenGL es una interface que consiste de aproximadamente 120 comandos, que se utilizan para especificar los objetos y las operaciones necesarias para producir aplicaciones tridimensionales interactivas.

OpenGL está diseñada como una librería independiente del hardware para lograr esto, no tiene comandos para realizar tareas de manejo de ventanas o de entradas del usuario.

Lo que tienen es un conjunto de primitivas geométricas: puntos, líneas y polígonos, así como comandos que actúan sobre estas primitivas y que permiten realizar tareas como sombreado, iluminación y texturizado. Dentro de las diversas funciones que puede realizar OpenGL sobre las primitivas podemos mencionar:

- Descripción y despliegue de primitivas tales como puntos, líneas y polígonos, así como el cálculo de sus normales.
- Manejo del objeto y de la cámara en un espacio tridimensional. (Rotaciones, Escalamientos, Translación)
- Creación de listas de despliegue, esto es la agrupación de comandos de OpenGL.
- Iluminación, posicionamiento de fuentes de luz.

La instrucción GLPoints nos sirve para dibujar puntos. La información más importante acerca de los vértices son sus coordenadas, que se especifican con el comando glVertex. Por cada uno de los vértices se puede especificar un color y una textura.

Capítulo 7

Pruebas

7.1. Pruebas del sistema distribuido

Una de las primeras pruebas que se realizarón sobre el sistema es la de permitir la manipulación de la pared de video, esto se hizo, aislando solo la parte que permite el manejo de objetos distribuidos visuales.

Para verificar la escalabilidad, se implantó el sistema de prueba sobre 3,6 y 12 máquinas, con mínimos cambios en el programa cliente de cada una de ellas.

La falla de cada máquina no altera la información que poseen las demás, pero evidentemente es notada por que se trata de una herramienta de visualización.

Que una máquina deje de funcionar no implica que la imagen no se pueda mostrar, solo que tendremos que moverla a la parte donde las máquinas funcionen correctamente, es decir solo reduce el area de despliegue, pero no se pierde la información.

Una de las pruebas mas interesantes desarrolladas en este proyecto, fue la de la implantación de los objetos distribuidos visuales, los cuales permitieron contener objetos en 3D creados con la biblioteca Open GL. Las gráficas que muestran los datos correlacionados por el sistema, son creados en este tipo de objetos, y que permiten un despliegue que promete ser en un futuro mas complicado.

Cuando se instaló la red inicial con Ethernet con 3 y 6 máquinas la velocidad de manejo fué relativamente corto, después se incremento el número de máquinas a 12, donde las ultimas 6 máquinas se conectaron a través de la red inalámbrica, con lo cual se noto el decremento de velocidad en cuanto a la manipulación de la pared de video, en la parte de inicialización.

Cuando el sistema esta ya en funcionamiento el retraso ya no es significativo y en cambio el costo es por el número de mensajes que se envian.

La implantación de las 6 primeras máquinas nos permitio el uso de un manejo con esquema de bus, cuando llegarón las otras 6 el esquema cambio a conmutador, aunque en el sentido estrito, ambos son de conmutador por que el servidor es siempre quien envia los mensajes a los intérpretes.

Este sistema tiene retrasos en el envio de los mensjaes debido a sus características de diseño en hardware, es por eso que una de las pruebas que se hizo, es la de enviar en la inicialización la mayoría de la información, y durante el manejo solo un indice que la caracterize, asi no importa que la tasa de transmisión sea baja, la información de peso esta en cada intérprete.

Cuando se ejecutá la solución, sin un sistema de archivos global, ciertamente fue necesario enviar información replicada a cada máquina, pero se muestra que el retraso no es significativo y que la información esta almacenada y respaldada en cada una de las máquinas, lo cual parece buena opción en caso de fallos masivos.

Durante la ejecución del sistema se puede percibir el retraso de los mensajes y la velocidad de despliegue de cada máquina, lo cual impide tener una transparencia real del sistema, pero como primera prueba se considera como una buena aproximación a esta característica.

La confiabilidad del sistema se probó con fallos en algunas máquinas, lo cual no evita que el funcionamiento del sistema continue, si perdida de información, solo disminución del área de despliegue.

La union de los objetos distribuidos y los objetos derivados de la clase de NSView funcionan correctamente permitiendo un buen manejo de gráficas sobre la pared.

La prueba de creación de un modelo complejo no se hizo por que los datos que se estan ingresando son en 2D, pero es posible mostrar cualquier modelo complicado posteriormente.

7.2. Pruebas con la minería visual

En este respecto se probó con una base de datos llamada Sinac, aunque sería deseable obtener una base de datos que contenga más datos númericos, con

comportamientos irregulares.

A continuación se muestra el estudio realizado sobre la base de datos de prueba.

7.2.1. Caso de estudio: SINAC

El Sistema de Informacón Académica (SINAC) es una base de datos integrada de información académica administrativa que abarca personal académico, productividad y alumnos. También es un sistema de consulta con aplicaciones que permiten realizar y controlar operaciones.

SINAC esta dividido en cuatro módulos:

- Capturas
- Catálogos
- Investigadores
- Servicios Escolares

El modelo de datos de SINAC

La base de datos académica surge a partir de un diseño conceptual basado en la metodología entidad-vínculo que deriva en un modelo relacional de datos que se prepara para el esquema físico en el manejador de bases de datos correspondiente. En un diagrama entidad-vínculo el modelo integra las diversas entidades académicas en grupos de vistas parciales:

- Vista de catálogos.
- Vista de adscripciones.
- Vista de alumnos.
- Vista de investigadores.
- Vista de proyectos.
- Vista de productividad.
- Vista de formación de recursos humanos.

El modelo lógico de la base de datos es el modelo relacional que está normalizado en tercera forma normal y genera un esquema físico para Progress.

A continuación se mecionarán las tablas que tienen potencial para la realización de la minería sobre ellas y sus atributos.

- Adscripciones. Contienen información referente a las adscripciones de los alumnos en los programas de maestría o doctorado que maneja el Cinvestav.

- ◇ Identificador de la unidad del cinvestav a la cual se va a accesar, cveUnidad, de tipo carácter.
- ◇ Identificador de la especialidad del Cinvestav a la cual se va a accesar, cveEspec, de tipo carácter.
- ◇ Nivel académico del alumno adscripto, de tipo carácter.
- ◇ Fecha en que el alumno fue aceptado en el programa.

- Alumnos. Contiene los datos personales del los alumnos.
  - ◇ Nombre completo del alumno, de tipo carácter.
  - ◇ Nombre del alumno, de tipo carácter.
  - ◇ Apellido paterno del alumno, de tipo carácter.
  - ◇ Apellido materno del alumno, de tipo carácter.
  - ◇ Sexo del alumno, de tipo carácter.
  - ◇ Fecha de nacimiento del alumno, de tipo carácter.
  - ◇ Dirección electrónica del alumno, de tipo carácter.
  - ◇ RFC, de tipo carácter.
  - ◇ CURP, de tipo carácter.
  - ◇ Fotografía del alumno.
  - ◇ Nacionalidad, de tipo carácter.
  - ◇ País, de tipo carácter.
  - ◇ Número de hijos, de tipo entero.
  - ◇ Clave provisional del alumno en el sistema SINAC, de tipo entero.
  - ◇ Identificador de la unidad, de tipo carácter.
  - ◇ Identificador del departamento, de tipo carácter.
  - ◇ Identificador de la sección, de tipo carácter.
  - ◇ Estado Civil, de tipo carácter.
  - ◇ Entidad federativa, de tipo entero.
- Inscripciones2. Contiene informacion sobre las inscripciones de los alumnos a los diferentes cursos que se imparten en el cinvestav o alguna institución externa.
  - ◇ Identificador en el sistema SINAC de la inscripción,de tipo carácter.
  - ◇ Identificador en el sistema SINAC del curso, de tipo carácter.
  - ◇ Cuatrimestre a cursar, de tipo carácter.
  - ◇ Número de inscritos, de tipo numerico.
  - ◇ Fecha de inscripción, de tipo data.
  - ◇ Tipo de curso, de tipo carácter.
- InscripcionesDet2. Contiene el detalle de las inscripciones de los alumnos a los diferentes cursos que se imparten en el cinvestav o alguna institución externa.

- ◇ Identificador en el sistema SINAC de la inscripción,de tipo carácter.
- ◇ Identificador en el sistema SINAC del curso, de tipo carácter.
- ◇ Calificación obtenida.

Ahora ya que tenemos una idea de cómo está constituida la base de datos es necesario pensar en cómo se obtendrán los datos de forma que puedan ser operados correctamente en el visualizador.

Nótese que hasta ahora las tablas anteriores contienen en su mayoría datos de tipo carácter, por ejemplo de la tabla alumnos, parecería que no se puede obtener nada útil para una minería numérica como la propuesta en este sistema.

Basta recordar que una interpretación interesante nos podría decir si es verdad que un alumno más joven tiene más posibilidades de terminar la maestría.

Existen muchas interpretaciones que nos parecen interesantes para descubrir, pero en este caso se buscará sobre las calificaciones que se obtienen por los alumnos en los distintos cuatrimestres, tratando de verificar, cual es en verdad el cuatrimestre que le cuesta más trabajo a todos los alumnos.

Recordemos que es posible generar consultas que permitan obtener tablas con campos apropiados para su uso, en este caso serán realizadas de modo que se puedan obtener los valores de las calificaciones de cada alumno en los diferentes cuatrimestres que ha cursado.

Una vez obtenida la consulta podemos generar tablas que puedan ser accesibles para el visualizador.

Resultados

Las tablas que vamos a utilizar en este caso para la obtención de resultados de la base de datos de Sinac, son 2, la primera se llama inscripcionesDet2, e inscripciones2.

Lo que se propone es tomar solo la información que podemos correlacionar en el visualizador, es decir se tomarán las calificaciones por cuatrimestre de las tablas anteriores.

Cuando se tengan por tabla por catrimestre se generara una base que contendra las tablas de manera que el visualizador pueda obtener los datos y pueda ser utilizado.

Esta minería nos muestra como los valores de las calificaciones con respecto a los diferentes cuatrimestres no es significativamente variable, como para decidir

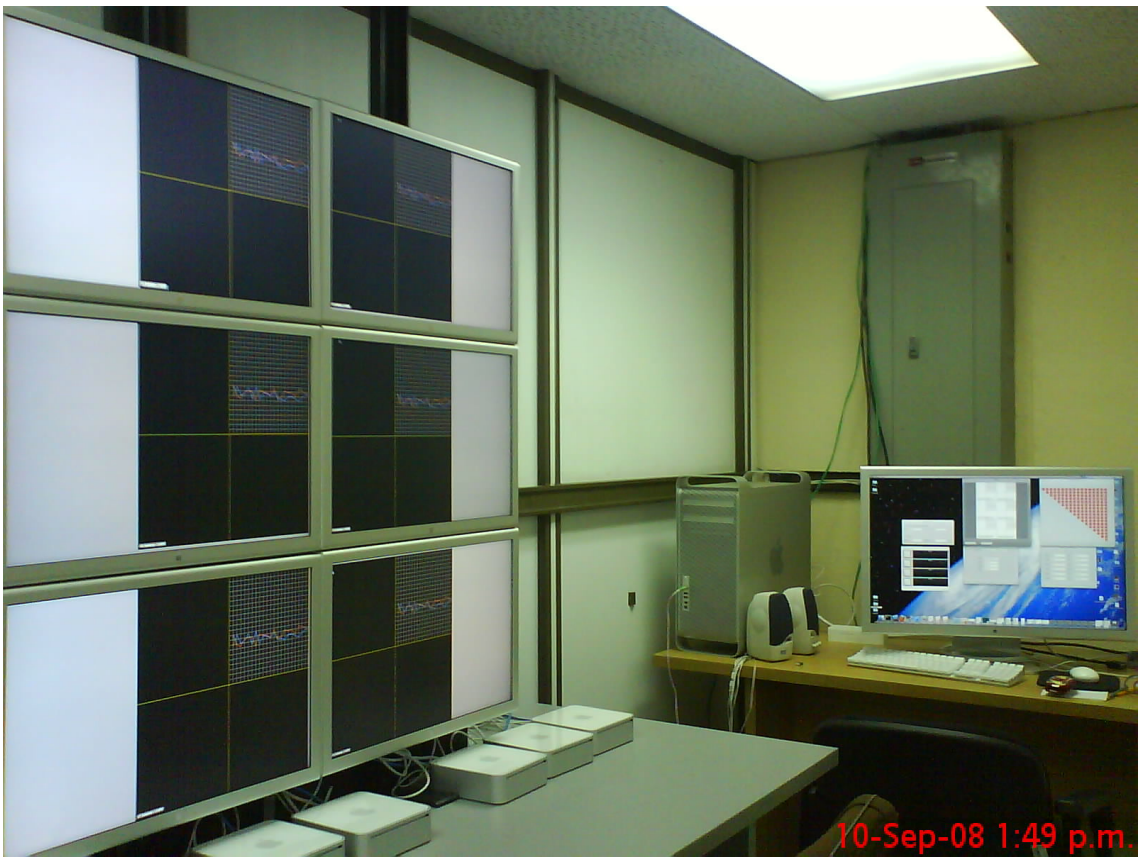

Figura 7.1: Graficas obtenidas de la mineria visual sobre Sinac

que alguno de los cuatrimestres, sea el mas complicado para los alumnos de maestría del cinvestav.

7.2.2. Problemas a resolver

La minería de datos visual es una modalidad en desarrollo, cuyo objetivo esta diversificado y puede incrementarse de tal manera que es muy dificíl definir sus aplicaciones o los métodos que deben ser implementados como basicos.

Tomando como experiencia el prototipo básico dessarrollado en este proyecto, se hace necesario mencionar los multiples métodos de minería, y de visualización de la información, los cuales quedan como propuestas para un trabajo posterior.

7.3. Pruebas con los objetos distribuidos visuales

Para verificar la validez de la idea de los objetos distribuidos visuales, se generó una aplicación visualizador de imagenes, que permite su movimiento por la pared de video.

Esta aplicación contiene las mismas funciones de comunicación que el proyecto, así como las mismas clases manejadoras de gráficos, es básicamente parte del proyecto.

Esta aplicación solo muestra la funcionalidad de la parte del sistema distribuido, sin manejo de bases de datos y sin la minería de datos, conservando las propiedades de manejo de la pared de video.

Figura 7.2: Graficas obtenidas de la mineria visual sobre Sinac

Capítulo 8

Conclusiones y Trabajo a futuro

8.1. Conclusiones

8.1.1. Conclusiones del sistema distribuido

Tomando en cuenta el diseño de este sistema distribuido se llegó a una primera conclusión que indica que este tipo de manejo distribuido de objetos, es en si una buena forma de control para una pared de video, aunque se debe tener en cuenta que algunas características deben ser mejoradas para beneficiar el desempeño.

En cuanto a la escalabilidad, el sistema es funcional, debido a que es facíl agregar máquinas que permitan la ampliación de la pared de video, lo cual se probo con el incremento de máquinas paulatinamente.

Se obtuvo una solución que permite tener objetos visuales distribuidos, que pueden contener visualizaciones complicadas, elaboradas con tecnología de Open GL.

Lo cual trae consigo que este tipo de objetos pueda contener gráficos simples en formatos comunes como jpg, bmp, png.

Tomando en cuenta lo anterior, obtuvimos una herramienta que permite la manipulación de imagenes sobre la pared de video.

La implantación de la red Ethernet fué más veloz que la de red inalámbrica en el inicio de la ejecución, pero una ves que estan conectados el retraso ya no es por la red sino por el número de mensajes enviados.

Apesar de haber perdido un poco de velocidad, este tipo de implantación per-

mite el uso de menos recursos para su desempeño, por lo menos en hardware, lo cual puede darle cierta ventaja en ciertos casos.

La idea de usar indices en lugar de compartir todo un bloque de imagen, a resuelto satisfactoriamente la desventaja del retraso en tiempo por el envio de mensajes, ya que el datos es un simple entero, aunque por otro lado si se considera que la replicación de la información puede llegar a ser un problema en cierto momento.

No se implementó un sistema de archivos compartido, por que la estrategia de solución implica el manejo de la menor cantidad de información a través de los mensajes, la conclusión de esto es que si es funcional, con sus respectivas mejoras.

Se logra tener la transparencia parcial dentro del sistema, demeritada solamente por la velocidad de despliegue de cada máquina.

La confiabilidad del sistema fue comprobada y es buena aún que implique no ser tan bueno en otros aspectos.

La idea de aprovechar las ventajas de los objetos distribuidos y los objetos contenedores de gráficos de la plataforma de MAC OSX fué buena y se concluye que con esto solo se ha puesto la primera piedra para la construcción de un sistema que permita más opciones en un futuro.

Esta primera implementación solo muestra datos en 2D, pero deja abierta las posibilidades de crear cualquier modelo complicado, con un mínimo cambio dentro de la clase del objeto visual.

8.1.2. Conclusiones de la minería visual

La minería que se desarrolló en este proyecto permite el manejo de los datos de manera que el usuario desida que parte quiere ver desplegada.

La interfaz que se emplea tratá de evitar ambiguedades en la medida de lo posible. El manejo de una pantalla principal permita el movimiento de las gáficas en la pared de video.

La obtención de la correlación de Pearson nos permite obtener una información muy básica de los datos, es buena si se trata de una primera exploración.

Las gráficas que se generan en este caso no son las más deseables pero cumplen con su función informativa, y son buenas para ejemplificar valores simples.

8.1.3. Conclusiones de los gráficos

Los gráficos que se obtuvieron en este sistema no son de alta calidad y no usan ni la mitad de las capacidades que tiene, por su generación con la biblioteca de OpenGL. Se pueden mejorar de una manera no tan compleja, ya que están implementados.

Los contenedores que se usan para el desliegue demostrarón ser eficientes en cuanto a la capacidad de manejo que permiten, así como por las capacidades gráficas con las que cuentan.

Se puede pensar en funciones como zoom o como resize que no serían tan complicadas de implementar gracias a la implementación que está ahora en funcionamiento.

8.2. Trabajo a futuro

8.2.1. Etapa de minería de datos

La minería de datos está lejos de ser un área totalmente descubierta ya que dentro de ella se engloban muchos aspectos muy importantes, por ejemplo la manera en la que se puede descubrir conocimiento dentro de un conjunto de datos no explorados.

El tipo de datos que se pueden manipular son tan variados que es necesario encontrar maneras para su manejo. Si hablamos de multimedia no solo debemos encontrar una forma para identificarlo sino además para su procesamiento y análisis preliminar.

Hablamos de memoria, de tiempo en el procesador, y de técnicas en el envío de red para distribuir esta carga que tiende a ser cada vez más pesada, por el tamaño de la información y por el tipo. Para ejemplificar un poco estas consideraciones basta hablar de las manipulaciones de imágenes de gran tamaño, ahora si en lugar de imágenes tenemos video el incremento en el procesamiento y almacenamiento es mayor.

Cabe notar que dentro del manejo de diversos tipos de información se ha dado solución a muchos problemas mediante el uso de metadatos, lo cual en si tiene su propio problema anexado con respecto a cómo se deben utilizar.

Es decir cómo tratar los datos que no tienen una estructura lineal, evitando en lo posible las ambigüedades que la misma estructura puede traer consigo.

Una vez que se pueda resolver el problema del tipo de datos que se manipulará y disculpando el problema que implica la estructura que los contendrá, está el problema de la comunicación de la aplicación con la base de datos, así como las consultas que se deberán hacer.

Cuál sería el manejador de bases de datos que pueda realmente beneficiar la comunicación, mediante que protocolo sería bueno transferir los resultados.

Otro problema que surge es la búsqueda del modelo más adecuado que contendrá la base de datos, aunque cabe mencionar que la mayoría de las bases de datos usan el modelo relacional, es necesario revisar si se podría usar un modelo que pueda ser mas benéfico en cuanto a lo que antes se mencionó.

Si hablamos de los métodos de minería se puede quedar abierta la pregunta de cuál sería el mejor para el tipo de base de datos que se tiene en el momento, pero es necesario recordar que para esta propuesta de solución del problema el más conveniente para una primera aproximación a la base de datos se hace mediante la obtención del coeficiente de correlación.

De ahí que se proponga para un trabajo a futuro anexar más técnicas de minería que puedan usarse después de la correlación para que así el experto pueda usar el método que más le convenza, y así podemos ayudar en la búsqueda de conocimiento dentro de cualquier área de forma más específica, dependiendo de las necesidades de cada una.

8.2.2. Etapa de comunicación con la base de datos

En este trabajo nos aprovechamos de las ventajas de la definición de objetos distribuidos dentro del lenguaje de programación objetive-C. Es por eso que la comunicación entre los mensajes la hacemos mediante el protocolo ip, definido por el lenguaje.

Realmente la forma en la que se comunican en este trabajo no causa problema debido a que los mensajes que se envían, no contiene datos más grandes que un entero o una cadena.

Debido precisamente al paradigma de solución que se implantó, pero si lo vemos desde un punto de vista más amplio, es decir que la aplicación realmente envía datos más grandes como imágenes o video.

La solución debe cambiar para utilizar un tipo de protocolo que permita la transmisión de paquetes de datos más grandes. Un ejemplo de esto es un proyecto que planea generar mapas dinámicamente desde una base de datos

geográfica, hablamos de obtener en tiempo real la información de cada uno de los puntos que formen el mapa.

En otras palabras trataremos de obtener la descripción de cada pixel de forma que pueda verse y manipularse cada uno de ellos con sus propias características, hablamos de 1920x1200 pixeles por cada pantalla conectada.

Además hablamos de una manera tal de comunicación que permita a cada máquina estar consciente de su posición con respecto a las demás para así obtener los datos que requiere para desplegar su parte.

Se trata de evitar el uso de un servidor centralizado que envíe órdenes a cada cliente, se trata de que cada nodo sepa que parte de imagen debe desplegar con respecto a las demás pantallas conectadas.

De esta forma se generará el mapa en forma distribuida, el problema se queda abierto, definido como la manera en la que un cluster de visualización puede generar un mapa sin necesidad de un control externo.

8.2.3. Etapa de generación de gráficas

En esta solución se generan gráficos simples en dos dimensiones, debido a que el tipo de método de minería elegido maneja pares de variables.

Aun así se han usado objetos que poseen las capacidades de OpenGL, con lo cual surgen muchas ideas que pueden realizarse en trabajos posteriores, mediante la generación de modelos más complejos, en 3D, con texturas que permitan una mejor apreciación del comportamiento de los datos.

En un futuro se propone tomar en cuenta más de dos grupos de variables, se propone la búsqueda de modelos adecuados que permitan el mejor uso de esta tecnología de visualización.

Existen características que pueden ser aprovechadas de mejor modo si se toma otra posición en cuanto al desarrollo de las graficas, se hace evidente que usarlo en este caso parece un desperdicio de tecnología.

Pero si tomamos en cuenta que este trabajo se puede tomar como parte de uno a futuro que requiera emplear este tipo de gráficos, podemos aceptar el hecho de que este sea el primer paso, y es mejor darlo con lo mejor que se tiene, aunque esté un poco desaprovechado.

Podría evitarse el retraso en la generación del grafico si utilizamos primitivas

básicas del contenedor estándar y generamos una imagen en baja resolución la cual puede ser manipulada de forma más fácil.

8.2.4. Etapa de manejo de objetos distribuidos

Recordando un poco podemos encontrar que:

Sistema distribuido es aquel en que los componentes de hardware y software, localizados en computadoras unidos mediante una red, comunican y coordinan sus acciones mediante el paso de mensajes.

Lo cual trae consigo las siguientes consecuencias:

- Concurrencia
- Inexistencia de un reloj global
- Fallos independientes

Por otro lado podemos encontrar una de las mas aceptadas definiciones de sistema distribuido como:

Un sistema distribuido es un grupo de computadoras independientes que son percibidas por los usuarios como una sola.

De ahí encontramos que los objetos distribuidos son módulos de software que son diseñados para trabajar juntos, pero están ubicados en múltiples máquinas, conectadas mediante la red. Un objeto envía un mensaje a otro objeto en una máquina remota. El resultado se envía al objeto que lo llamó. Ahora bien, sobre la solución que se le dio a este concepto, cabe recordar que la idea fue aprovechar las ventajas que ofrece la tecnología del lenguaje objetive-C, es por eso que se manejan tal y como son propuestos.

Podemos mejorar este tipo de mensajería si conocemos explícitamente como se manejan, podríamos en otro contexto tratar de implementar este concepto con otro protocolo, evitando en lo posible los retrasos, pero esto claro requiere de una amplia investigación en este respecto.

Las características que debe cumplir un objeto distribuido se pueden limitar a las que podemos realmente ofrecer, sabemos que con respecto al concepto más general, un objeto distribuido no está en una máquina sino en varias y de ahí que surjan la concurrencia y los demás problemas inherentes al concepto de distribuido.

Una idea que surge de la implementación realizada me hace sugerir una manera en la que el objeto realmente permanezca en varias máquinas a la vez.

Esta idea es con respecto a la generación de mapas desde una base de datos geográfica, la cual se menciono al inicio del capítulo.

Si tomamos al mapa como un objeto y consideramos que su área va a estar distribuida por todas las pantallas, además de permitir la manipulación del usuario sobre dicho objeto, se abren más problemas a su alrededor como la manipulación del ratón por el área, que maquina lo maneja, que tal si todas lo hacen, como hacer posible eso, que tal si una de todas, pero todas saben lo que sucede con estos eventos.

Bibliografía

[1] Thomas A. DeFanti Chong Zhang, Jason Leigh. Terascope: Distributed visual data mining of terascale data sets over photonic networks. 19:935 – 943, 2003.

[2] Oreste Verta Domenico Talia, Paolo Trunfio. Weka4ws: a wrsf-enabled weka toolkit for distribuited data mining on grids. *Data Mining Grid digital library*, 2005.

[3] Matthew Eldridge Greg Humphreys. Wiregl: A scalable graphics system for clusters.

[4] Apple Inc. Distributed objects programming topics cocoa interapplication communication. © *2003, 2007 Apple Inc.*, 2007.

[5] Apple Inc. Nsopenglview class reference cocoa user experience. © *2003, 2007 Apple Inc.*, 2007.

[6] Apple Inc. Nsview class reference cocoa graphics and imaging. *Apple Inc.* © *2003, 2007 Apple Inc.*, 2008.

[7] Cesar Ferri Ramírez José Hernández Orallo, Maria José Ramírez Quintana. *Introducción a la minería de datos*, volume 19. Marcel Dekker. INc., 2004.

[8] Stephen W Michnick Kirill Tarassov. ivici: Interrelational visualization and correlation interface.

[9] Mason J. Katz William J. Link Philip M. Papadopoulos, Caroline A. Papadopoulos and Greg Bruno. Configuring large high-performance clusters at lightspeed: A case study. *IEEE Computer Graphics and Applications*, 2005.

[10] Singh Rajvikra. Sage: the scalable adaptive graphics environment.

[11] George W. Snedecor and William. *Métodos estadísticos*. CECSA, 1977.

[12] Pak Chung Wong. Visual data mining. *IEEE Computer Graphics and Applications*, pages 2–3, 2002.