

PLATAFORMA PARA VISUALIZACIÓN EXTENDIDA Y AUMENTADA

LAURA CONSTANZA MARTINEZ MENDOZA

DEYBERTH HERNANDO RIAÑO NUÑEZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA**

2013

PLATAFORMA PARA VISUALIZACIÓN EXTENDIDA Y AUMENTADA

**LAURA CONSTANZA MARTINEZ MENDOZA
DEYBERTH HERNANDO RIAÑO NUÑEZ**

**Trabajo de Grado para optar al título de Ingenieros de
Sistemas**

Director

CARLOS JAIME BARRIOS HERNÁNDEZ

Co-Directora

LOLA XIOMARA BAUTISTA ROZO

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA**

2013

DEDICATORIA

A Dios por iluminarme para labrar este camino

*A toda mi familia por apoyarme en cada
instante vivido durante estos 5 años*

*A Luis Hernando, Daribel, Dayra y Juan David
por su compañía incondicional*

*A Paula Camila porque todos los días me daba aliento
para continuar con mi labor*

A Laura C. por acompañarme en esta etapa de mi vida

A todos mis amigos y compañeros

A la UIS, SC3, GIIB y Grupo Halley

A todos los docentes por sus enseñanzas

Mil gracias a todos

Deyberth Riaño Nuñez

A mi Padre, Madre y Hermanas por su apoyo y confianza

A Sofía por ser esa constante felicidad

A mis amigos(as) por su paciencia y ánimo para llegar a la meta

A ti Juan D. Carvajal, por esos años en los que me enseñaste tanto de la vida

A Deyberth Riaño, por ser un gran hermano y cómplice en el éxito del proyecto

A los profesores y a la UIS por los conocimientos que me ofrecieron

Laura C. Martínez Mendoza

AGRADECIMIENTOS

El éxito alcanzado de este proyecto es la suma de muchos esfuerzos, razón por la cual queremos agradecer el apoyo, tiempo y paciencia a:

Laboratorio de visualización TACC “*Texas Advanced Computing Center*” de la universidad de Austin, Texas. Con especial mención a la Ingeniera Karla Vega, por sus explicaciones y recomendaciones acerca del funcionamiento del middleware y las diversas alternativas de realizar visualización.

Al Laboratorio de Computo de Alto Rendimiento y Calculo Científico de la Universidad Industrial de Santander (UIS). Por facilitar los recursos de hardware y prestarnos soporte necesario.

Al grupo Halley de Astronomía y Ciencias Aeroespaciales, por habernos facilitado el uso de sus recursos de hardware y software.

Al Director, Carlos Jaime Barrios y codirectora Lola Bautista, por la confianza depositada en nosotros, por su tiempo y esfuerzo para hacer un seguimiento a el desarrollo del proyecto, por compartir sus conocimientos, y lo más valioso, por su constante apoyo y motivación.

A los integrantes del grupo de supercomputación y cálculo científico de la UIS, SC3 por sus asesorías, y motivación permanente.

CONTENIDO

| | | |
|-----|---|----|
| 1. | INTRODUCCIÓN | 22 |
| 2. | DESCRIPCIÓN DEL PROYECTO | 24 |
| 2.1 | JUSTIFICACIÓN | 24 |
| 2.2 | DESCRIPCIÓN DEL PROBLEMA | 25 |
| 2.3 | OBJETIVO GENERAL | 26 |
| 2.4 | OBJETIVOS ESPECÍFICOS | 26 |
| 3. | MARCO TEÓRICO | 27 |
| 3.1 | ARQUITECTURA CLÚSTER | 27 |
| 3.2 | MIDDLEWARE | 29 |
| 3.3 | INTERFAZ DE PASO DE MENSAJES | 31 |
| 3.4 | INTERFAZ HOMBRE-MAQUINA | 33 |
| 3.5 | MUROS DE VISUALIZACIÓN | 34 |
| 4. | ESTADO DEL ARTE | 50 |
| 4.1 | CONSTRUCCIÓN Y USO DE UN SISTEMA DE MURO DE VISUALIZACIÓN | 50 |
| 4.2 | JUXTAVIEW – UNA HERRAMIENTA PARA LA VISUALIZACION INTERACTIVA DE IMAGENES DE GRAN TAMAÑO EN UN MOSAICO DE MONITORES | 51 |
| 4.3 | DESARROLLO DE UN MURO DE VISUALIZACION BASADO EN GRID PARA VISUALIZACION EN RED | 52 |
| 4.4 | METODO PARA LA UTILIZACION PRACTICA DE UN MOSAICO DE MONITORES EN LA VISUALIZACIÓN CIENTÍFICA | 53 |
| 4.5 | STALLION | 54 |
| 5. | METODOLOGÍA | 56 |

| | | |
|-----|---|-----|
| 6. | IMPLEMENTACIÓN DE LA PLATAFORMA | 65 |
| 6.1 | ESPECIFICACIONES Y DEFINICIÓN DEL HARDWARE | 65 |
| 6.2 | DESCRIPCIÓN Y REQUERIMIENTOS DE SOFTWARE | 71 |
| 6.3 | FUNCIONAMIENTO DE LA PLATAFORMA | 77 |
| 6.4 | API DESARROLLADA | 82 |
| 6.5 | ILUSTRACION PASO A PASO DEL MONTAJE REALIZADO | 83 |
| 7. | LINEAMIENTOS | 86 |
| 7.1 | LINEAMIENTOS PROPUESTOS PARA LA CONSTRUCCION O MODIFICACION DE PLATAFORMAS DE VISUALIZACION | 86 |
| 7.2 | LINEAMIENTOS PROPUESTOS PARA LA CONSTRUCCION O MODIFICACION DE APLICACIONES QUE USEN LA PLATAFORMA DE VISUALIZACIÓN | 88 |
| 8. | PRUEBAS Y ANÁLISIS DE RESULTADOS | 91 |
| 8.1 | RENDIMIENTO DE LA RED | 92 |
| 8.2 | RENDIMIENTO DEL CLUSTER | 94 |
| 8.3 | PRUEBAS HARDWARE HETEROGÉNEO | 97 |
| 9. | LIMITACIONES DEL PROYECTO | 99 |
| 10. | RECOMENDACIONES | 100 |
| 11. | CONCLUSIONES | 101 |
| 12. | BIBLIOGRAFÍA | 103 |
| 13. | ANEXOS | 105 |

LISTA DE FIGURAS

| | |
|---|----|
| FIGURA 1: ARQUITECTURA DE UN CLÚSTER DE COMPUTADORAS. | 31 |
| FIGURA 2: INTERFAZ HOMBRE-MÁQUINA | 34 |
| FIGURA 3: TRABAJOS DEL CENTRO DE COMPUTACIÓN Y VISUALIZACIÓN, UNIVERSIDAD BROWN A) CAVE, B) TILED DISPLAY WALL. | 45 |
| FIGURA 4: MURO DE VISUALIZACIÓN CENTRO DE CIENCIA FORT COLLINS | 46 |
| FIGURA 5: MURO DE VISUALIZACIÓN UNIVERSIDAD DE LA GEORGIA | 48 |
| FIGURA 6: PROYECTO GRIMAGE | 49 |
| FIGURA 7: MÉTODO CIENTÍFICO | 56 |
| FIGURA 8: MURO DE VISUALIZACIÓN UNIDAD DE SUPERCOMPUTACIÓN HORIZONTAL | 59 |
| FIGURA 9: MURO DE VISUALIZACIÓN UNIDAD DE SUPERCOMPUTACIÓN VERTICAL | 63 |
| FIGURA 10: MURO DE VISUALIZACIÓN DE LA UNIDAD DE SUPERCOMPUTACIÓN | 66 |
| FIGURA 11: MURO DE VISUALIZACIÓN GRUPO HALLEY | 67 |
| FIGURA 12: PORCENTAJE DE VENTAS DE LOS DIFERENTES FABRICANTES DE TARJETAS GRÁFICAS A NIVEL MUNDIAL. | 69 |
| FIGURA 13: INTERFAZ GRÁFICA DE USUARIO DE DISPLAYCLUSTER | 78 |
| FIGURA 14: ARCHIVO DE CONFIGURACIÓN DISPLAYCLUSTER | 79 |
| FIGURA 15: MENÚ PRINCIPAL Y BARRA DE HERRAMIENTAS - DISPLAYCLUSTER | 80 |
| FIGURA 16: ILUSTRACIÓN DEL MONTAJE REALIZADO (PARTE 1) | 83 |

| | |
|--|----|
| FIGURA 17: ILUSTRACIÓN DEL MONTAJE REALIZADO (PARTE 2) | 84 |
| FIGURA 18: FLUJO DE DATOS TCP PARA UN SOLO NODO | 93 |
| FIGURA 19: FLUJO DE DATOS TCP PARA EL TOTAL DE NODOS EN EL CLÚSTER | 94 |
| FIGURA 20: CARGA DE PROCESOS EN EL CLÚSTER | 95 |
| FIGURA 19: USO DE BYTES DE MEMORIA RAM DEL CLUSTER | 96 |
| FIGURA 21: BYTES/SEGUNDOS TRANSMITIDOS POR LA RED. | 97 |

LISTA DE TABLAS

| | |
|---|----|
| TABLA 1: RESUMEN POR CRITERIOS DE EVALUACIÓN PROYECTORES CONTRA PANTALLAS LCD | 40 |
| TABLA 2: COMPARACIÓN TARJETAS DE VIDEO NVIDIA GEFORCE, NVIDIA QUADRO, AMD | 41 |
| TABLA 2: ESPECIFICACIONES MURO DE VISUALIZACIÓN UNIVERSIDAD DE GEORGIA | 47 |
| TABLA 3: ESPECIFICACIONES DE LA PLATAFORMA DE VISUALIZACIÓN DEL GRUPO DE SUPERCOMPUTACIÓN | 66 |
| TABLA 4: ESPECIFICACIONES DEL HARDWARE DEL MURO DE VISUALIZACIÓN DEL GRUPO HALLEY | 68 |
| TABLA 5: RENDIMIENTO RELATIVO DE LAS TARJETAS GRÁFICAS. | 70 |
| TABLA 6: ESPECIFICACIONES DE LAS TARJETAS DE VIDEO UTILIZADAS | 70 |
| TABLA 7: COMPARATIVO DE MIDDLEWARE | 72 |
| TABLA 8: PAQUETES BÁSICOS PARA LA INSTALACIÓN DE DISPLAYCLUSTER | 73 |

LISTA DE ANEXOS

| | |
|--|-----|
| ANEXO A: INSTALACIÓN DE DISPLAY CLUSTER | 105 |
| ANEXO B: SOFTWARE Y REVISIONES ADICIONALES | 113 |
| ANEXO C: CÓDIGO FUENTE API DESARROLLADA | 119 |
| ANEXO D: ARTÍCULO PRESENTADO EN EL CLCAR 2012 | 123 |
| ANEXO E: DISTINCIÓN OTORGADA POR EL COMITÉ ORGANIZADOR DEL CLCAR 2012 COMO POSTER GANADOR DEL EVENTO | 128 |

ACRÓNIMOS

| | |
|--------|--|
| LCD | Liquid Crystal Display (Pantalla de cristal líquido) |
| MPI | Message Passing Interface (Interfaz de paso de mensajes) |
| GPU | Graphic Processing Unit (Unidad de procesamiento gráfico) |
| TACC | Texas Advance Computing Center (Centro de Cómputo Avanzado de Texas) |
| API | Application Programming Interface (Interfaz de programación de aplicaciones) |
| GIIB | Grupo de Investigación en Ingeniería Biomédica |
| OpenGL | Open Graphics Library |
| SPU | Sound Processing Unit |
| LED | Light Emitting Diode (Diodo emisor de luz) |
| SAGE | Scalable Adaptive Graphics Environment |
| CPU | Central Processing Unit (Unidad central de procesamiento) |
| GbE | Gigabit Ethernet |
| DVI | Digital Visual Interface (Interfaz Digital Visual) |
| VGA | Video Graphics Array |
| CUDA | Compute Unified Device Architecture |
| NFS | Network File System (Sistema de archivos de red) |
| SSH | Secure Shell |
| RSA | Rivest, Shamir, Adleman |
| XML | eXtensible Markup Language |

GLOSARIO

Servicios tecnológicos: Área clave para el manejo de la información a nivel institucional y de vital importancia en el funcionamiento y desempeño del ministerio o de la empresa. Estos servicios tecnológicos para su prestación utilizan equipamiento, infraestructura y recursos humanos especializados.

Computación de alto rendimiento: Herramienta para solución de problemas complejos, que permite la reducción en el tiempo de cálculo. Comercialmente, la computación de alto rendimiento es conocido como una computadora que tiene una alta potencia comparada con otras del mercado actual.

Sistemas distribuidos de cómputo: Agrupación de computadoras independientes (cada computadora tiene hardware y software específico que puede diferir), interconectados a través de una red. Esta colección de ordenadores se comporta ante un usuario como un sistema único.

Sistema de cómputo: Conjunto de elementos, físicos (Hardware) e intangibles (Software), que interactúan entre sí para procesar y almacenar información de acuerdo a una serie de instrucciones pre-establecidas.

Tecnología estándar: Conjunto de técnicas, conocimientos, y procesos que se encuentran comúnmente en dispositivos utilizados en casas o centros de estudio y que sirven para el diseño y construcción de nuevas soluciones que facilitan la vida humana.

Middleware: Software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, redes, hardware y/o sistemas operativos. Convirtiéndose en el cerebro de una operación cliente-servidor.

MPI: (Message Passing Interface), Estándar que define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes diseñada para ser usada en programas que usan múltiples procesadores.

Supercomputación: Campo de la Ingeniería de Sistemas orientada a optimizar la duración en la ejecución de tareas, a través de arquitecturas robustas, que permiten que el tiempo de cálculo disminuya significativamente.

Hardware gama alta: Conjunto de partes tangibles de un sistema informático caracterizado por ser de altas prestaciones comparado con los componentes más comunes de mercado.

Escalabilidad: Propiedad deseable de un sistema, red, o proceso que refleja la habilidad para reaccionar y adaptarse sin perder calidad o bien manejar cambios en el tamaño de la estructura sin que este implique muchos cambios en la configuración del sistema.

Muro de visualización: Solución escalable de alta resolución de pantallas ubicadas en forma de mosaico, en otras palabras, es un conglomerado de monitores, ubicadas de forma estratégica, que logran ofrecer a un usuario una vista ampliada de sus trabajos.

Alta resolución: Calidad de la pantalla o imagen. Está basado en el número de píxeles que se utilizan para crear y formar una imagen en la pantalla. Cuanto mayor sea el número de píxeles, mayor será la resolución del dispositivo.

Visualización extendida: observación de imágenes de forma no convencional, es decir, el estudio de estas imágenes se hace a lo largo de la misma, permitiéndose recorrerla y llegar a diferentes niveles de detalle.

Visualización aumentada: estudio u observación de imágenes en dimensiones superiores a las tradicionales. Se dice, que es aumentada por la alta resolución con la que se ve la imagen, y por su tamaño.

Arquitectura: Es el diseño conceptual y la estructura operacional fundamental de un sistema de computadora. Es decir, es un modelo y una descripción funcional de los requerimientos y las implementaciones de diseño para varias partes de una computadora.

Clúster: Conjunto de computadoras independientes las cuales se encuentran interconectadas operando de forma conjunta como un único recurso computacional.

Nodo: Es un ordenador o una unidad de procesamiento ubicada en una estructura conocida como rack el cual contiene uno o más nodos. Este puede actuar como: maestro o esclavo. El primero de ellos es el responsable de coordinación de tareas y presentación de resultados. Mientras que el segundo, se caracteriza por recibir instrucciones y ejecutarlas.

Programación paralela: Forma de cómputo en la que muchas instrucciones se ejecutan simultáneamente, operando sobre el principio de que problemas grandes, a menudo se pueden dividir en unos más pequeños, que luego son resueltos simultáneamente.

Balanceo de carga: Es un concepto usado en informática que se refiere a la técnica usada para compartir el trabajo a realizar entre varios procesos, ordenadores, discos u otros recursos. Está íntimamente ligado a los sistemas de multiprocesamiento, o que hacen uso de más de una unidad de procesamiento para realizar labores útiles.

Tolerancia a fallos: capacidad de un sistema de almacenamiento de acceder a información aún en caso de producirse algún fallo. Esta falla puede deberse a daños físicos (mal funcionamiento) en uno o más componentes de hardware lo que produce la pérdida de información almacenada.

Usabilidad: rapidez y facilidad con que las personas hacen sus tareas utilizando una herramienta o cualquier objeto fabricado por los humanos con el fin de alcanzar un objetivo.

RESUMEN

Título:

PLATAFORMA DE VISUALIZACIÓN EXTENDIDA Y AUMENTADA.¹

Autores: MARTINEZ MENDOZA, Laura Constanza, RIAÑO NÚÑEZ, Deyberth Hernando.²

Palabras Claves: Visualización extendida y aumentada, middleware, plataforma de visualización, lineamientos para construcción y/o modificación de aplicaciones, visualización científica.

DESCRIPCIÓN

Las necesidades de visualizar imágenes o videos en alta resolución es una de las tareas que por años ha sido abordada por diferentes grupos de investigación y centros de tecnología de punta. Con el paso de los años se han generado soluciones en términos de supercomputación, logrando resultados prometedores.

Hoy en día la adquisición de hardware con mayores prestaciones y de bajo costo han permitido propuestas científicas de mayor alcance, por ello es que este proyecto de investigación se beneficia del uso de recursos con tecnología estándar las cuales junto con las prácticas de desarrollo de software actuales, permiten un buen nivel de poder de computo por costo. Este software encargado del control de las plataformas de visualización es llamado middleware, el cual a su vez hace uso de múltiples tecnologías comunes en la industria, por ejemplo MPI.

En la actualidad existen diferentes middleware los cuales ofrecen cada uno ventajas y desventajas según el tipo de uso que se quiera hacer de la plataforma. DisplayCluster es un middleware desarrollado por la universidad de Texas que permite un buen nivel de interacción con aplicaciones externas realizando algunas modificaciones además que puede ser alterado fácilmente al ser un software de código abierto.

Con el presente trabajo de investigación se implementó una plataforma que permite el desarrollo de aplicaciones para la investigación que requieran de visualización extendida y aumentada, además de generar lineamientos para la construcción y/o modificación de este tipo de aplicaciones, también se establecen pautas para la instalación y mantenimiento del middleware utilizado.

¹ Trabajo de Grado con modalidad de Trabajo de Investigación.

² Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.
Director: Carlos Jaime Barrios Hernández. Co-Directora: Lola Xiomara Bautista Rozo

ABSTRACT

TITLE:

PLATAFORMA DE VISUALIZACIÓN EXTENDIDA Y AUMENTADA.³

AUTHORS: MARTINEZ MENDOZA, Laura Constanza, RIAÑO NÚÑEZ, Deyberth Hernando.⁴

KEYWORDS: extended and augmented visualization, middleware, visualisation platform, generating guidelines for the construction and adaptation of applications, scientific visualization.

DESCRIPTION:

The necessity of high resolution image and video visualisation is one of the tasks conducted for years by different academic research groups and state of the art research centres. Along the years solutions have been produced based on high performance computing, with different levels of success.

The possibility of acquiring less expensive and more powerful hardware today have allowed scientific research of larger scope than before. This project profits from this possibility by using off the shelf technology hardware resources and common software development practices, attaining a good computing power per money ratio. The software used to control the visualisation platform is a middleware, using multiple, common technologies as MPI.

There are currently several middleware for that task, each one with its own advantages and disadvantages according with the specific use intended for the platform. DisplayCluster, developed by the Texas Advanced Computing Center, is a middleware that is suited, with modifications, for interaction with third party applications, and can be easily modified because it is an open source software.

This work implemented a platform that allows the development of research focused applications requiring extended and augmented visualisation, generating guidelines for the construction and adaptation of such applications. Also, maintenance guidelines for the platform are proposed.

³ Trabajo de Grado con modalidad de Trabajo de Investigación.

⁴ Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Carlos Jaime Barrios Hernández. Co-Directora: Lola Xiomara Bautista Rozo.

1. INTRODUCCIÓN

Los servicios tecnológicos dirigidos a investigadores que requieren infraestructuras de computación de alto rendimiento o sistemas distribuidos de cómputo, son hoy en día un logro a nivel regional e institucional, por ello, la Universidad Industrial de Santander, encaminada y comprometida con dicho desarrollo tecnológico, estableció el Laboratorio de Supercomputación y Cálculo Científico UIS ⁵, el cual se ha caracterizado por soportar y desarrollar la actividad científica de la UIS desarrollando ideas tecnológicas de cómputo avanzado garantizando su alto rendimiento.

La infraestructura de cómputo “GridUIS-2” integra e interconecta diferentes recursos que se encuentran distribuidos en el campus de la UIS, ubicados en: La Escuela de Ingeniería de Sistemas e Informática (EISI), el Laboratorio de Física de Materia Condensada (FICOMACO), el Laboratorio de Cálculo Numérico Luis Eduardo Arias ubicado en el CENTIC y el Laboratorio de Supercomputación en la Sede Guatigará. El Servicio se encuentra Centralizado a través del Centro de Tecnologías de Información (CENTIC) en el campus Universitario. Es tal el compromiso con el desarrollo tecnológico que a la fecha la Universidad Industrial de se ha convertido en uno de los protagonistas a través de su infraestructura tecnológica, particularmente enorgullecida por contar con la máquina más potente del país, más conocida como “GUANE 1”, la cual es un clúster tipo Beowulf de gran poder gracias a las GPU que lo integran⁶, no obstante, las propuestas fueron más ambiciosas, nació la idea de la visualización como una técnica innovadora del análisis de datos y/o aplicaciones que se ejecutan en un clúster y que una vez concluida su ejecución pueden presentarse gráficamente siendo esto de gran utilidad a investigadores y científicos.

⁵ Laboratorio de Supercomputación y Cálculo Científico UIS [En Línea]. Consultado [22/01/13] Disponible en: <<http://sc3.uis.edu.co/>>

⁶ Infraestructura de la plataforma Grid UIS-2. [En línea]. Consultado [22/01/13] Disponible en: <<http://grid.uis.edu.co/index.php/Infraestructura>>

La visualización de imágenes y/o aplicaciones de alta resolución han tomado gran valor a nivel de ciencia e investigación, esto debido al aumento en la resolución con la que hoy cuentan los dispositivos digitales y el poder de cómputo de las supercomputadoras las cuales generan imágenes de ultra alta resolución que no pueden observarse adecuadamente en una pantalla ordinaria.

Centros de Supercomputación como el de la Universidad de Texas, Texas Advance Computing Center – TACC han construido muros o mosaicos de pantallas de diferente resolución, con variedad en el número de monitores y su forma de arreglo en espacio horizontal y vertical. Igualmente se han planteado el uso de proyectores que bajo como la calibración mecánica, algoritmos de calibración geométrica, y la calibración fotométrica logran una sincronización que da como resultados imágenes casi perfectas.

Con el mejoramiento de las soluciones que hasta hoy se han propuesto en este proyecto de Investigación, y la generación de nuevos lineamientos que reorienten la construcción o modificación tanto de la plataforma como de aplicaciones que hacen uso de esta, se podrían obtener mejores resultados en este campo de la Ingeniería.

En este documento se presenta la justificación y los objetivos del trabajo de investigación, seguida de los conceptos y los proyectos afines existentes; en tercer lugar presentamos la metodología de trabajo, la plataforma construida, los lineamientos para la construcción de la plataforma y de las aplicaciones que sean ejecutadas en el laboratorio de visualización, seguido de las pruebas y análisis de los resultados; por último las limitaciones encontradas, recomendaciones y conclusiones del trabajo realizado.

2. DESCRIPCIÓN DEL PROYECTO

2.1 JUSTIFICACIÓN

La actividad investigativa exige nuevas herramientas para el tratamiento de la información de acuerdo con los requerimientos y necesidades del tratamiento específico del problema, así mismo, las posibilidades tecnológicas que existen hoy en día responden a esa exigencia, ofreciendo no sólo herramientas, sino plataformas completas que permiten el desarrollo de nuevas utilidades de acuerdo a las necesidades de los usuarios.

En esa dirección, la UIS adquirió plataformas de supercomputación que requiere sistemas de interacción de alto rendimiento para un aprovechamiento integral de acuerdo con las exigencias de los laboratorios, grupos de investigación, centros y consorcios que implican investigación y desarrollo.

Un muro de visualización es una plataforma que permite la extensión y visualización aumentada de información para múltiples usos. Desde otra perspectiva se podría definir como una alternativa que a nivel internacional está siendo usada como plataforma de apoyo a diversos tipos de aplicaciones informáticas que requieren ser visualizadas en mayor detalle, a través de los cuales se busca hacer confortable el trabajo humano.

De lo expuesto anteriormente, se implementó una Plataforma de Visualización para la Universidad Industrial de Santander, la cual facilita la integración de diversos Grupos de Investigación pertenecientes a la Universidad, gracias a que brinda soporte a todo tipo de aplicaciones, y la cual en conjunto con el desarrollo del Parque Tecnológico de Guatiguará, proporciona una solución visual para sus aplicaciones.

2.2 DESCRIPCIÓN DEL PROBLEMA

Con el paso de los años la comunidad científica ha fortalecido sus investigaciones y por ende sus descubrimientos, sin embargo se ven limitados por los recursos tecnológicos con los que cuentan, debido a dos razones primordiales: la primera, la falta de tecnología que supla sus necesidades y la segunda, la dificultad para su adquisición por su elevado costo y/o el reducido presupuesto para su obtención.

El volumen de datos ha venido creciendo significativamente, permitiendo incrementar la información que tenemos de nuestra realidad, sin embargo, no toda esta información logra ser analizada, debido a la falta de herramientas y aplicaciones para los científicos. Otra dificultad que se ha presentado, corresponde a la visualización de imágenes de alta resolución, pues requiere de infraestructuras de hardware especializadas, siendo un gran desafío a investigar en este campo de la ingeniería y la supercomputación.

Actualmente la Universidad Industrial de Santander se encuentra consolidando el Parque Tecnológico de Guatiguará, que se perfila como un centro de referencia internacional en las áreas de energía, salud, software, agroindustria y tecnología como una solución a la problemática expuesta, buscando un factor de cambio para una mejor calidad de vida de la sociedad.⁷

Paralelo al desarrollo del parque, se realizó la implementación de una plataforma de visualización de altas prestaciones que brinda un calificativo importante a los recursos con los que cuenta el área de tecnologías de la información, ya que se constituye, como un proyecto de esta área pero con finalidades de soporte y apoyo al resto de ciencias, cubriendo las necesidades de estudio de imágenes en

⁷ Visión Parque Tecnológico de Guatiguará [En línea]. Consultado [22/01/13] Disponible en: <<http://www.uis.edu.co/webUIS/es/investigacionExtension/guatiguara/index.html>>

alta resolución, en mayores dimensiones y convirtiéndose en una plataforma piloto para futuros desarrollos.

2.3 OBJETIVO GENERAL

Proponer una plataforma que permita el desarrollo de aplicaciones para la investigación que requiera de visualización extendida.

2.4 OBJETIVOS ESPECÍFICOS

1. Estudiar y elegir el uso de hardware estándar para construir una plataforma de visualización extendida determinando con esto la viabilidad de la misma.
2. Analizar y adaptar un middleware que permita la interacción de los componentes de la plataforma propuesta.
3. Proponer lineamientos que orienten la construcción o modificación, tanto de la plataforma como de las aplicaciones que harán uso del muro de visualización.
4. Validar la plataforma a través de una aplicación médica del grupo de investigación GIB probando con esto la utilidad de emplear este tipo de recursos a nivel científico y de investigación.

3. MARCO TEÓRICO

3.1 ARQUITECTURA CLÚSTER

Un clúster es un conjunto de computadoras independientes⁸ las cuales se encuentran interconectadas operando de forma conjunta como un único recurso computacional. Este arreglo de computadoras pueden conformarse a través de hardware de alta gama como también a partir de un conglomerado de equipos con partes de hardware estándar, las cuales brindan un poder de cálculo mucho más elevado que el realizado por un solo ordenador.

En la actualidad se pueden agrupar los clúster en dos categorías según sus propiedades de hardware y software, que se mencionan a continuación:

1. **Homogéneo:** todos los nodos que conforman el clúster se basan en la misma arquitectura, es decir, todos tienen la misma configuración de hardware y sistema operativo.⁹
2. **Heterogéneo:** su arquitectura se basa en nodos distintos en aspectos como el tiempo de acceso, características de hardware variado, sistema operativo distinto, rendimiento de los recursos diferente.¹⁰

Tanto para los clúster de tipo homogéneo y heterogéneo, los componentes más comunes son:

1. Múltiples nodos de computación

⁸ Multicomputadores tipo Clúster. [En línea]. Consultado [22/01/13] Disponible en: <http://www.uhu.es/josem.bravo/AD/Tema3.pdf>

⁹ Clúster Homogéneo. [En línea]. Consultado [22/01/13] Disponible en: <http://www.uhu.es/josem.bravo/AD/Tema3.pdf>

¹⁰ Clúster Heterogéneo. [En línea]. Consultado [22/01/13] Disponible en: <http://www.uhu.es/josem.bravo/AD/Tema3.pdf>

2. Sistemas operativos
3. Red de interconexión de altas prestaciones
4. Protocolos de comunicación y servicio
5. Middleware
6. Entornos y herramientas de programación paralela.

El funcionamiento de los componentes antes nombrados puede administrarse según las finalidades que se tengan con el clúster. Los dos modelos principales para gestionar los recursos son:

1. **Centralizado:** es en el cual existe un nodo maestro para configurar el comportamiento de todo el sistema y el cual facilita una mejor conducción del clúster.¹¹
2. **Descentralizado:** más conocido como un modelo distribuido donde cada nodo se administra y gestiona. En este tipo de arreglos se pueden utilizar aplicaciones de alto nivel centralizadas que guíen el arreglo. Esta arquitectura tiene mayor tolerancia a fallos sin embargo tiene mayor dificultad en su administración.¹²

3.1.1 Clúster Beowulf

Sistema de cómputo masivamente paralelo de altas prestaciones, principalmente cuyos componentes hardware son estándar. Este tipo de clúster tiene como características la ejecución de un sistema operativo de libre distribución como por ejemplo Linux o FreeBSD, sus elementos se comunican a través de una red de

¹¹ Modelo de Gestión Centralizado. [En línea]. Consultado [22/01/13] Disponible en: <<http://www.uhu.es/josem.bravo/AD/Tema3.pdf>>

¹² Modelo de Gestión Descentralizado. [En línea]. Consultado [22/01/13] Disponible en: <<http://www.uhu.es/josem.bravo/AD/Tema3.pdf>>

alta velocidad. Al igual que cualquier clúster su finalidad está dirigida a la ejecución de tareas que requieren una alta capacidad de cálculo.

Para el pleno cumplimiento de estas tareas los nodos no se encuentran en el puesto de trabajo del usuario, sino que están totalmente dedicados a las tareas asignadas.¹³

Los clúster Beowulf son muy conocidos por comportarse como un solo computador de altas prestaciones, en donde a pesar de tener varios nodos de trabajo, estos se muestran como uno solo.

Por otro lado, históricamente también son bien conocidos, ya que este tipo de clúster ha sido implementado con grandes fines y para reconocidas organizaciones. Un ejemplo de esto fue el primer clúster tipo Beowulf, Desarrollado por primera vez en 1994, por Donald Becker y Thomas Sterling en la NASA. El clúster agrupaba 16 procesadores Intel DX4 de 100MHz, interconectados con tecnología Ethernet a 10Mbps.¹⁴

3.2 MIDDLEWARE

El middleware está definido como un software intermediario que proporciona un enlace entre aplicaciones software o un conjunto de computadoras independientes, que coordina las tareas ejecutadas.

El middleware actúa como interfaz entre las tareas/procesos y el hardware o sistema operativo. En mayor detalle su papel es: ocultar detalles de la plataforma

¹³Clúster Beowulf. [En línea]. Consultado [22/01/13] Disponible en: <<http://studies.ac.upc.edu/EPSC/TFC/Beowulf/beowulf.html>>

¹⁴ Historia de la Tecnología: Clúster Beowulf, la supercomputadora de los pobres. [En línea]. Consultado [22/01/13] Disponible en: <<http://alt1040.com/2011/11/historia-de-la-tecnologia-cluster-beowulf-la-supercomputadora-de-los-pobres>>

y del hardware, automatizar las tareas más pesadas, reducir el número de errores de programación. Este software estará dedicado a proporcionar servicios y funcionalidades para el desarrollo de aplicaciones.

La funcionalidad con la que hoy cuentan los middleware debe responder a las necesidades del grupo de trabajo. Algunas de las características con la que cuentan estos para la construcción de un clúster son las siguientes:

- **SSI (Single System Image):** Propiedad de un sistema que oculta la naturaleza heterogénea y distribuida de los recursos, logrando con esto que tanto usuarios como aplicaciones cuenten con un recurso unificado y sencillo.¹⁵
- **Herramientas de optimización y mantenimiento del sistema:** realiza la migración de procesos, Checkpoint-restart, balanceo de carga, tolerancia a fallos.
- **Escalabilidad:** detecta automáticamente nuevos elementos conectados a un clúster para proceder a su utilización.

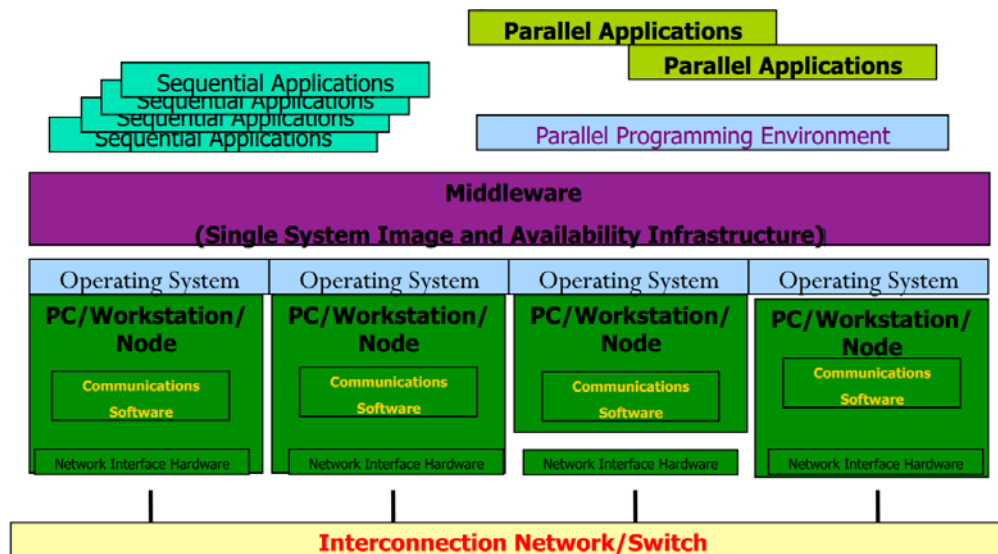
Comúnmente el middleware que se emplea en un clúster se encuentra realizando tareas de puente entre el hardware que conforma el arreglo de computadoras y las aplicaciones que son llevadas a cabo por los usuarios.

La Figura 1 es una representación gráfica de la ubicación estratégica del middleware en la arquitectura de un clúster de computadoras. Este se ubica entre los componentes estáticos y aquellos que varían según los usuarios finales y sus requerimientos, en este caso las aplicaciones y los datos que cada uno operan. En general, El middleware recibe los trabajos entrantes al clúster (aplicaciones en

¹⁵Single System Image. [En línea]. Consultado [22/01/13] Disponible en: <<http://www.buyya.com/papers/SSI-CCWhitePaper.pdf>>

paralelo, aplicaciones secuenciales) y los redistribuye en cada nodo de trabajo, a través de la red. Este a su vez coordina que el proceso se ejecute rápido y el sistema no sufra sobrecargas.

Figura 1: Arquitectura de un Clúster de Computadoras.



Fuente: Unidad de Supercomputación y Cálculo Científico UIS.¹⁶

3.3 INTERFAZ DE PASO DE MENSAJES

El paso de mensajes es un modelo de comunicación ampliamente usado en computación paralela. El crecimiento en el volumen y diversidad de aplicaciones han ido promoviendo la implementación de un sistema basado en el pase de mensajes de forma eficiente y portable. En base a esto se creó un estándar conocido hoy en día como MPI o Message Passing Interface.

MPI es entonces un estándar para la implementación de sistemas de pase de mensajes para funcionar en una amplia variedad de computadoras paralelas. Su diseño se inspira en máquinas con arquitecturas distribuidas en donde cada

¹⁶ Arquitectura de un Clúster de computadoras [En línea]. Consultado [22/01/13] Disponible en: <<http://grid.uis.edu.co/images/9/92/E-Ciencia.pdf>>

procesador es propietario de cierta memoria y la única forma de intercambiar información es a través de mensajes.

Objetivos de MPI:

1. Diseñar una API (Application Programming Interface)
2. Hacer eficiente la comunicación
3. Permitir que las aplicaciones puedan usarse en ambientes heterogéneos
4. Soportar conexiones de la interface con C y Fortran 77, con una semántica independiente del lenguaje.
5. Proveer una interface de comunicación confiable.

3.3.1 IMPLEMENTACIONES DEL DOMINIO PÚBLICO DE MPI

Algunas de las implementaciones de libre disposición que se han desarrollado en MPI son:

MPICH: Producida por el laboratorio nacional de Argonne y la Universidad de Mississippi. Es una implementación optimizada de MPI para entornos homogéneos y Myrinet¹⁷, Proporcionando un mayor rendimiento en el paso de mensajes entre nodos.

LAM: (Local Area Multicomputer), Es un ambiente de programación y un sistema de desarrollo sobre redes de computadoras heterogéneas. Desarrollada en el centro de computación de Ohio.

¹⁷ MPICH. [En línea]. Consultado [22/01/13] Disponible en:
<<http://www.mcs.anl.gov/research/projects/mpi/mpich1-old/>>

UNIFY: Provee un subconjunto de MPI dentro del ambiente PVM⁸ sin sacrificar las llamadas ya disponibles de PVM. Desarrollado por la Universidad del Estado de Mississippi. Permite que un mismo programa pueda contener código PVM Y MPI.

W32MPI: implementación de MPI para un conglomerado con MS-Win32 basado en MPICH y disponible por el instituto de Ingeniería Superior de Coímbra – Portugal y de la universidad de Coímbra.

MPI-FM: Es un puerto de alto rendimiento de MPICH para un conglomerado de SPARCstation⁹ interconectados por Myrinet. Está basado en mensajes rápidos y viene de la universidad de Illinois.

3.4 INTERFAZ HOMBRE-MAQUINA

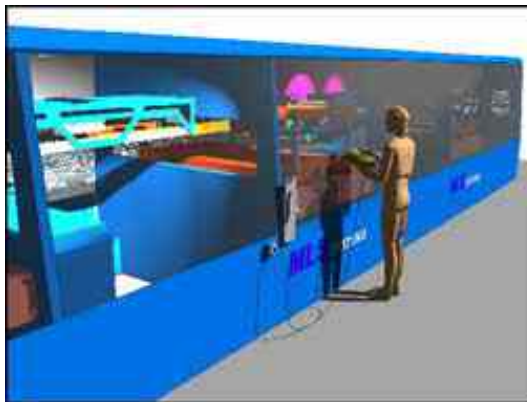
Las necesidades de entender el mundo han constituido un camino hacia la generación de nuevas ideas y tecnologías, las cuales para su desarrollo recurren al concepto de interfaz hombre – máquina. Debido al crecimiento del volumen de la información, y a la forma de interactuar con el mundo, las máquinas se convirtieron en el medio idóneo para llevar a cabo la realización exitosa y óptima de hacer una tarea.

La idea de pensar en una interfaz como un puente de comunicación entre los modelos mentales de un usuario y las capacidades de una máquina, conllevan un estudio de las expectativas de un posible usuario y cuál debe ser la correcta representación de éstas en un lenguaje disyunto al ser humano. Instruirse en lo que él espera, acoplar su forma de ver las cosas y la utilidad que percibe de una herramienta son factores que constituyen la robustez y usabilidad de un proyecto.

En el caso particular de muros de visualización, la idea de generar una interfaz, y más que eso, un mecanismo casi transparente por el cual el usuario puede recibir

la información, con gran detalle, con mayor área de visión del tamaño real del objeto, y con la posibilidad adicional de interactuar y modificar sus tareas han llevado a el éxito del crecimiento de este tipo de trabajos. La siguiente imagen, es un prototipo, de cómo la ciencia busca presentar la información de manera novedosa, logrando así descubrir cómo funciona y se comporta el mundo.

Figura 2: Interfaz Hombre-Máquina



Fuente: MLS Systems. Disponible en: <http://www.mlssystems.com/>

3.5 MUROS DE VISUALIZACIÓN

3.5.1 ¿Qué es un muro de visualización?

Un muro de visualización es una solución escalable de alta resolución de pantallas ubicadas en forma de mosaico.

Cuando se habla de alta resolución implica que es mucho mayor a la encontrada en monitores normales de escritorio, la cual es típicamente de uno a cuatro megapíxeles y no es poco común que se necesite resoluciones mucho mayores [3].

Cuando se refiere a mosaico, se quiere decir que el muro de visualización contiene múltiples monitores organizados en forma de cuadrícula, mientras que al hablar de una vista extendida, nos referimos a que el usuario se sale de su doctrina de observador “desde una sola perspectiva” y pasa a poder moverse a través de la imagen, estudiando con esto cada parte en más detalle según su interés.

Finalmente se dice que es una arquitectura escalable, es decir que a medida que las necesidades de resolución aumentan, pueden ser añadidos nuevos monitores a la cuadrícula con mínimas adecuaciones.

Un muro de visualización es entonces, un complejo mecanismo tanto de hardware como software que opera de forma transparente al usuario de la herramienta.

Algunas características de un muro de visualización son¹⁸:

1. Su tamaño permite que múltiples usuarios lo usen simultáneamente.
2. Su alta resolución permite información de alta calidad.
3. La combinación de tamaño y resolución, permite a los usuarios obtener información general del objeto estudiado y detalles a medida que se hace la exploración sobre este.

Las opciones para realizar visualización de imágenes son:

Estación de trabajo individual: Consta de una o dos pantallas de alta resolución operadas por un ordenador, es una alternativa de bajo costo y complejidad. Útil para usuarios que trabajan individualmente.

¹⁸ Visualization Clusters using Commodity Hardware Consultado [22/01/13] Disponible en: <http://i.dell.com/sites/content/business/solutions/whitepapers/en/Documents/hpc_vis_wall_011209.pdf>

Gran pantalla sencilla: (el tamaño depende de la distancia que se esté proyectando, en algunos proyectores superan las 300”) Es una opción cuyo costo y complejidad aumenta respecto a la anterior. Se lleva a cabo con el uso de Proyectores LCD. Esta solución es útil, para grupos de personas que hacen visualización de imágenes, y cuyos requerimientos en el nivel de detalle son bajos pero en donde es vital el tamaño, es decir, se sacrifica la definición de la imagen por altas dimensiones.

Sistema de múltiples proyectores sin fisuras: Es el sistema de mayor complejidad y costo entre los tipos de muros conocidos. Basado en un sistema de proyectores que logran ofrecer visualización de alta resolución y en gran tamaño. Su elevado costo se debe a que requiere de un preciso sistema de alineación, sumado al costo energético, refrigeración requerida entre otros. Este sistema es clasificado como la solución menos viable a nivel de rentabilidad [4].

Mosaico de pantallas LCD: Es un sistema de monitores LCD que varían en número, según los alcances que se quieran tener del mismo. Con un costo y complejidad óptimos para la eficiencia que ofrece. Es una solución que permite la visualización de grandes volúmenes de datos en forma simultánea y en alta resolución.

3.5.2 Software para muros de visualización

A pesar de la sencillez del montaje de un muro de visualización con un solo equipo de cómputo, el problema a tratar se enfoca en sistemas distribuidos y paralelos en muros de ultra-alta resolución. Uno de los retos en los sistemas distribuidos y paralelos es la compatibilidad de aplicaciones con el software existente. Muchos de los cambios para ejecutar aplicaciones secuenciales se deben hacer directamente sobre el código fuente de las mismas y algunas de sus bibliotecas y, con ello presentan el inconveniente de poder o no, ser usadas en el muro de

visualización. Algunos componentes de software más importantes en un sistema de visualización son: Sistema operativo, sistema de administración del clúster, middleware de visualización y aplicaciones software.

A continuación se presenta una breve descripción de diferentes tipos de middleware más utilizados para la conducción de un muro de visualización.

Chromium [12]

Chromium (basado en WireGL) es un sistema de renderización interactivo, diseñado para clúster. Utiliza unidades de procesamiento de transmisión (SPU) para serializar, de manera opcional permite modificar una secuencia de comandos OpenGL.

Una SPU toma una secuencia como entrada y produce cero o más secuencias de salida. La comunicación entre SPU ubicados en diferentes computadoras es manejada por una abstracción de red basada en conexión punto a punto. *Chromium* puede usarse para redirigir el flujo gráfico de salida de aplicaciones OpenGL al muro de visualización. La biblioteca OpenGL usada por las aplicaciones es reemplazada por la biblioteca OpenGL modificada de *Chromium*, esta biblioteca intercepta todos los comandos OpenGL y los reenvía al SPU local. Un mosaico ordenado es enviado por la SPU a los servidores *Chromium* corriendo en el muro de visualización. Las SPU locales de cada parte del arreglo reciben su secuencia de comandos correspondientes y estas son enviadas a la tarjeta gráfica.

Scalable Adaptive Graphics Environment (SAGE) [8]

SAGE es un sistema de renderización remota basado en píxeles para muros de visualización. En *SAGE*, la renderización se separa de la visualización usando un

clúster dedicado para entregar píxeles al conjunto de nodos de visualización. *SAGE* incorpora un gestor de ventanas que permiten que el flujo de píxeles sea superpuesto, movido o redimensionado en el muro de visualización. La biblioteca de interface de aplicación de *SAGE* (*SAIL*) es usada en el lado de la renderización para capturar la salida de la renderización de la aplicación y secuenciar la salida a un conjunto de receptores *SAGE* (un receptor por nodo de visualización). Un gestor de FreeSpace comunicado con *SAIL*, los receptores *SAGE* y los clientes UI para controlar la composición final de la secuencia en el muro de visualización. Como inconveniente en la utilización de *SAGE* se requiere enlaces de red de alta velocidad por que se transmiten flujos de píxeles y no se utiliza todo el potencial de renderización de las tarjetas gráficas ni las capacidades de cómputo de las CPU ubicados en los nodos de visualización.

DisplayCluster [9]

DisplayCluster es un entorno software diseñado por el TACC (Texas Advance Computing Center) de la Universidad de Texas, para manejar muros de visualización de gran escala. DisplayCluster permite administrar el contenido del muro de visualización de manera remota desde equipos como laptops y permite interactividad mediante dispositivos como Joysticks, iPhone, iPad, iTouch o algún dispositivo Android. Existe una nueva posibilidad de interactuar con el muro de visualización mediante el uso de Microsoft Kinect pero que aún se encuentra en versión experimental.

DisplayCluster es montado sobre un ambiente MPI. Una vez configurado el clúster MPI la instalación puede ser llevada a cabo de manera sencilla en comparación con los ambientes para muros de visualización presentados anteriormente. Adicionalmente, se cuenta con una interfaz de secuencia de comando en Python que permite la interacción con DisplayCluster.

3.5.3 Hardware para muros de visualización

Un muro de visualización se construye típicamente con componentes ordinarios para una PC. Sin embargo y según las ambiciones y presupuesto que se tenga, este puede ser construido como un muro de altas prestaciones y de excelente calidad, o como uno que cumple con los objetivos de visualizar sin ofrecer mayores utilidades. Existen dos modalidades muy utilizadas en su construcción, la primera y de mayor predicción actualmente es el uso de pantallas LCD o LED, y la segunda es el uso de proyectores de alta resolución. Estas dos alternativas, han presentado por años sus pros y sus contras basados en criterios como: Costo, Espacio, Mantenimiento, Resolución total ofrecida, Continuidad de la imagen, Escalabilidad.

El uso de pantallas tiene su mayor inconveniente en los bordes que por defecto tiene todo monitor¹⁹. Hoy en día, los fabricantes de estos, han reducido el espacio destinado a borde, razón por la cual poco a poco esta gran desventaja perderá peso. Otra solución ha sido a través de muros flexibles, en donde a través del soporte de cada monitor, la idea con esto es cuadrar la inclinación tanto horizontal como vertical, de tal forma que los bordes de cada uno queden parcialmente ocultos. Las ventajas en cambio son mayores, convirtiéndolo en la opción de mayor preferencia, por los centros de cómputo de hoy en día. Los criterios de costo, espacio, mantenimiento son muchas de las ventajas que tiene este tipo de implementación, ya que la vida útil de un monitor, bajo condiciones de uso normales es muy elevada, exhibiendo una imagen de alta calidad en el transcurso del tiempo. Además de esto, la configuración del arreglo es sencilla y se cuenta con buenas características según el tipo de monitor escogido.

¹⁹A Practical Guide to Large Tiled Displays. Consultado [22/01/13] Disponible en: http://tacc-web.austin.utexas.edu/bwesting/public_html/stallion.pdf

La utilización de proyectores en la construcción de muros de visualización, no es muy popular en la comunidad debido a dificultades como: Obtener la alineación de forma correcta, evitar que las imágenes se solapen, la aparición de sombras y la pérdida en la calidad de la imagen, además, el espacio que físicamente requiere hace que su futura escalabilidad sea limitada, el mantenimiento de las bombillas y en general de todo el conjunto de proyectores hacen que se eleve el costo²⁰. La resolución máxima ofrecida por un proyector, es menor comparada con la que puede alcanzar un muro de LCD. Sin embargo, su gran ventaja sobre los monitores es la ausencia de bordes y que al cambiar la distancia de proyección puede incrementarse/disminuirse el tamaño del muro de visualización.

En general, la infraestructura usada en el desarrollo de un laboratorio de visualización depende de los requerimientos y preferencias de los usuarios. La tabla 1 presenta una comparación por diversos criterios entre la construcción de un muro de visualización con proyectores y pantallas LCD.

Tabla 1: Resumen por criterios de evaluación Proyectores contra Pantallas LCD

| Criterio | Costo | Espacio | Mantenimiento | Escalabilidad | Continuidad imagen |
|---------------------|--------------|----------------|----------------------|----------------------|---------------------------|
| Proyectores | Alto | Alto | Alto | Medio | Alto |
| Pantalla LCD | Bajo | Medio | Bajo | Alto | Medio |

Fuente: Autores

Un segundo componente hardware importante son las tarjetas gráficas, las cuales hoy en día, proporcionan dos o más salidas de video, por lo cual, pueden ser configurados dos o más monitores por estación de trabajo, tal como es el caso de

²⁰ A Practical Guide to Large Tiled Displays. Consultado [22/01/13] Disponible en: <http://tacc-web.austin.utexas.edu/bwesting/public_html/stallion.pdf>

la tarjeta nVIDIA GeForce GTX 670²¹ la cual cuenta con conectividad Dual Link DVI-D. Otro tipo de tarjeta, introducida recientemente al mercado que permiten controlar más de dos monitores, son las tarjetas de video nVIDIA QUADRO[5] y la AMD junto a las tecnologías NVIDIA®Mosaic y Eyefinity[6] respectivamente, a continuación se encuentra una tabla comparativa de estas soluciones disponibles al alcance de todos.

Tabla 2: Comparación tarjetas de video nVIDIA GeForce, nVIDIA QUADRO, AMD

| | Tecnología | Cantidad máxima de monitores |
|----------------|---------------|------------------------------|
| nVIDIA GeForce | NVIDIA®Mosaic | 16 (4 GPU) |
| nVIDIA QUADRO | NVIDIA®Mosaic | 16 (4 GPU) |
| AMD | Eyefinity | 6 (1 GPU) |

Fuente: Autores

Estas tarjetas se pueden utilizar en dos tipos de configuraciones que dependen del número de monitores a desplegar en el muro de visualización:

1. Una computadora que soporta una cantidad de tarjetas gráficas y que controlará todo el muro de visualización.
2. Múltiples computadoras donde cada una de ellas contiene una tarjeta gráfica que alimenta a cada uno de los monitores.

En los muros de visualización controlados por una sola computadora, se combina el hecho de que ciertas tarjetas gráficas pueden manejar múltiples monitores y existe la posibilidad de añadir cierta cantidad de tarjetas a una máquina. El resultado contiene múltiples tarjetas gráficas conectadas a una sola tarjeta madre, esto simplifica en gran medida la configuración debido a que muchos sistemas

²¹nVIDIA GeForce GTX 670. Consultado [22/01/13] Disponible en: <http://www.nVIDIA.es/object/geforce-gtx-670-es.html>

operativos la reconocen como un recurso unificado y además, es la solución más económica para implementar y mantener [7].

Cuando una sola computadora no es una opción para manejar un muro de visualización, se debe recurrir a múltiples computadoras, interconectadas por medio de tecnologías conmutadas Ethernet de 1 Gigabit, pues se debe garantizar un ancho de banda suficiente para distribuir los datos correspondientes a la imagen eficientemente. Cada computadora maneja un número determinado de monitores los cuales se sincronizan usando intercomunicación. Estas configuraciones además de presentar alta escalabilidad en el sistema de visualización, presentan una potencia de procesamiento mucho mayor que un equipo único.

3.6. OPCIONES DE MUROS DE VISUALIZACION

La visualización de imágenes en alta resolución tomó gran importancia, debido a que a través de ésta se puede obtener gran cantidad de información. No obstante, no es ésta la única razón para que hoy en día los mosaicos de pantallas sean una alternativa para lograr la atención de determinado grupo de usuarios. Un ejemplo de esto, son los muros de visualización encontrados en aeropuertos, centros comerciales, museos, supermercados, universidades, laboratorios, etc. Con base en el ejemplo anterior, es claro precisar la diferencia, alcance y enfoque de las diversas formas de hacer visualización. Esta se divide en dos categorías, muros de visualización de uso comercial o científico.

3.6.1. Soluciones Comerciales

Los muros de visualización de tipo comercial, han alcanzado un crecimiento significativo debido a la forma como se presenta la información y lo llamativa que

puede resultar. Utilidades como la visualización de una gran ciudad en una sola pantalla, el despliegue de publicidad en una resolución mayor a la encontrada en la mayoría de televisores de un hogar, o aún más sorprendente la visualización de diversos ecosistemas en alta definición.

Dado este gran mercado, existen hoy en día empresas dedicadas a desarrollar y comercializar soluciones de visualización de alto rendimiento, según las necesidades y deseos de los clientes. A continuación se muestran algunas de ellas.

Gregintech²²: Empresa Mexicana constituida para proveer las mejores soluciones en sistemas de visualización, asistencia de proyectos inteligentes, y soporte técnico dedicado. Entre su lista de productos se encuentran muros LCD. La solución construida es ajustada totalmente a los requerimientos del cliente el cual fija: uso de monitores o televisores, tamaño, dimensiones del muro de pantallas, espacio disponible, etc.

Mechdyne²³: Empresa Estadounidense proveedora mundial de tecnologías de la visualización. Ofrecen soluciones íntegras y personalizadas que incluyen: consultorías, software, servicios técnicos, e integración de hardware. Mechdyne adicionalmente ofrece soluciones con múltiples proyectores, procesamiento gráfico y panel de control táctil.

X-ISS²⁴: Empresa Estadounidense localizada en Houston, Texas. Se especializa en la entrega de soluciones de cómputo de alto rendimiento. La construcción de muros de visualización, cuya calidad, es el alto grado de detalle que puede presentar una imagen o video de gran tamaño, adicionalmente, ofrece la instalación total de la arquitectura, soluciones especializadas según las

²² Greintech. [En línea]. Consultado [22/01/13] Disponible en: <<http://www.gregintech.com/es/es-inicio.htm>>

²³ Mechdyne. [En línea]. Consultado [22/01/13] Disponible en: <<http://www.mechdyne.com>>

²⁴ X-ISS. [En línea]. Consultado [22/01/13] Disponible en: <<http://x-iss.com/turnkey-viz-walls.html>>

necesidades del usuario final, razones que han convertido a X-ISS en una compañía muy reconocida en este tipo de soluciones.

AUTOMA²⁵: Empresa Colombiana con sedes en Bogotá y Medellín, cuenta con un equipo de expertos, especializados en la distribución mayorista de marcas mundialmente reconocidas en el área de la tecnología, enfocado principalmente en sistemas profesionales de audio, video, videocitofonía, cerraduras electrónicas y automatización tanto para hogares como oficinas. Su mayor cliente son centros comerciales, hoteles y lugares que busquen generar innovación. Actualmente ofrece variedad de productos para realizar visualización a partir de las diversas necesidades que pueda tener un usuario, sin embargo, no dejan de ser soluciones en su gran mayoría de tipo comercial.

3.6.2. Soluciones Científicas

Actualmente, los grupos de investigación han desarrollado laboratorios y centros dedicados a la visualización de imágenes en alta resolución, desarrollando así, ideas cada vez más innovadoras de la forma como percibimos y entendemos el mundo. A diferencia de las soluciones comerciales, las cuales enfocan su trabajo a la venta de un producto; la ciencia y principalmente las universidades, han dedicado sus esfuerzos a desarrollar arquitecturas cada vez más poderosas y de mayor capacidad, lo cual ha generado una retroalimentación constante de nuevas tecnologías tanto en hardware y software, y lo más importante, la agrupación de ambas, para que juntas ofrezcan mayor rendimiento.

A continuación se encuentran algunas soluciones realizadas por centros o grupos de investigación de diversas Universidades.

²⁵ AUTOMA. [En línea]. Consultado [22/01/13] Disponible en: <<http://www.automa.com.co/>>

Universidad Brown – Centro de Computación y Visualización²⁶: Universidad Estadounidense localizada en el estado de Rhode Island. Cuentan con un centro de cálculo y visualización que incluye tecnología de gama alta para realizar visualización inmersiva centralizada. Actualmente, su trabajo más destacado es: *CAVE* acrónimo de “*Automatic Virtual Environment CAVE*”. Este es un cubo con imágenes que se proyectan en la parte frontal, izquierda, derecha y suelo del arreglo (Figura 3a). Adicional a este trabajo, el centro de computación y visualización cuenta con un muro de visualización especialmente útil en el estudio de datos e imágenes de alta resolución generados en un ordenador de alto rendimiento (Figura 3b).

Figura 3: Trabajos del centro de computación y Visualización, Universidad Brown

a) CAVE

b) Tiled Display Wall.



Fuente: Brown University Center for Computation and Visualization²⁷

CAVE permite la visualización de volúmenes, visualización molecular y simple manipulación de modelos en 3D. Tiled Display Wall es usada para el estudio de datos de alta resolución o conjunto de datos de gran tamaño.

Centro de Ciencia Fort Collins: Ubicado en el estado de Colorado – Estados Unidos, este centro de ciencia, implementó un muro de 24 monitores de 27

²⁶ Brown University Center for Computation and Visualization. [En línea]. Consultado [22/01/13] Disponible en: <<http://www.brown.edu/Departments/CCV/visualization>>

²⁷ Brown University Center for Computation and Visualization [En línea]. Consultado [22/01/13] Disponible en: <<http://www.brown.edu/Departments/CCV/visualization>>

pulgadas cada uno. Permite la visualización de conjuntos de datos del sistema de información geográfica. Este muro puede ser configurado para mostrar un mapa único, es decir, una imagen a lo largo de todo el espacio disponible. O por el contrario, usar cada monitor como una salida independiente a su vecina. Cuando se hace uso de todo el mosaico este diseño ofrece 50 millones de píxeles, permitiendo así ganar resolución y mayor detalle, según el tipo que se esté realizando. El objetivo principal del proyecto es apoyar y mejorar las capacidades de cálculo y visualización de los fenómenos climáticos. La figura 4 presenta el muro antes descrito.

Figura 4: Muro de Visualización Centro de Ciencia Fort Collins



Fuente: Fort Collins Science Center ²⁸

La siguiente solución fue desarrollada por Cinemassive, una empresa especializada en muros de visualización en la ciudad de Atlanta, Estados Unidos. Esta plataforma fue creada con fines científicos los cuales son mencionados a posteriormente.

Universidad Georgia²⁹: Universidad Estadounidense, localizada en el estado de Georgia. Cuenta con un centro de visualización que ofrece una resolución de más

²⁸ Visualization Wall, Fort Collins Science Center. [En línea]. Consultado [22/01/13] Disponible en: <<http://www.fort.usgs.gov/RAM/VisWall.asp>>

de 200 millones de píxeles. Este muro es considerado uno de los más grandes del mundo y es compatible con numerosas aplicaciones de ciencias como: biología, geo ciencia, salud, computación, etc. Adicionalmente, permite la visualización y comprensión de datos complejos provenientes de la bioinformática, información biomédica, y sistemas geográficos.

La tabla 2 presenta las especificaciones de hardware y software del muro de visualización de la Universidad de Georgia, seguido de la representación gráfica de la misma (Figura 5).

Tabla 2: Especificaciones Muro de Visualización Universidad de Georgia

| | | |
|--|----------------------------|------------------------------------|
| Procesador | Intel Quad Core i7 | |
| Memoria RAM | 12 GB | |
| Sistema Operativo | Windows 7 | |
| Unidades de procesamiento gráfico | 6 | |
| Middleware | CineMassive Software Suite | |
| Aplicaciones soportadas | ArcGIS | Reason/Record |
| | Visual Molecular Dynamics | SAGE (Windows version in progress) |
| | Google Earth | Unity3D |
| | MATLAB | Zeiss AIM software |
| | Microsoft Office | |
| | | |

Fuente: Especifications, Visualization Wall, GSU³⁰

²⁹ Visualization Wall, GSU. [En línea]. Consultado [22/01/13] Disponible en: <<http://technology.gsu.edu/technology-services/it-services/research-computing/visualization-wall/>>

³⁰ Especifications, Visualization Wall, GSU. [En línea]. Consultado [22/01/13] Disponible en: <<http://technology.gsu.edu/technology-services/it-services/research-computing/visualization-wall/>>

Figura 5: Muro de Visualización Universidad de la Georgia



Fuente: Cinemassive – Advance Visualization ³¹

GRIMAGE (Grid and Image): fue desarrollado en el centro de investigación INRIA especializado en ciencias de la computación, teoría de control y matemáticas aplicadas, ubicado en Grenoble, Francia.

El objetivo principal de Grlmage es proporcionar un espacio de visualización para grandes modelos y la interacción en tiempo real con los mismos. Actualmente cuentan con un conjunto de cámaras digitales que permiten la adquisición de video en tiempo real, generando así un modelo 3D de una escena vista desde muchas perspectivas. La figura 6 muestra el proyecto Grlmage.

³¹ Visualization Wall, GSU. [En línea]. Consultado [22/01/13] Disponible en: <<http://www.cinemassive.com/solutions/visualization/Georgia-State-University/>>

Figura 6: Proyecto GrImage



Fuente: GrImage³²

La propuesta de trabajo mencionada en el siguiente capítulo abarca el desarrollo del proyecto desde tres puntos de vista, científico (visualización de datos científicos), metodológico (como implementar la plataforma) y didáctico (proceso de aprendizaje durante la implementación de la plataforma).

En base a los múltiples trabajos científicos, al crecimiento de los recursos tecnológicos para hacer visualización y a la generación de nuevos paradigmas se ha establecido un estado del arte de este tipo de ideas innovadoras de alto rendimiento. En el siguiente capítulo se indican algunos de los trabajos más destacados en el tema de visualización en alta resolución.

³² GrImage. [En línea]. Consultado [22/01/13] Disponible en: <<http://grimage.inrialpes.fr/index.php>>

4. ESTADO DEL ARTE

El desarrollo de plataformas de visualización es una tarea de años atrás, en donde cada proyecto se convierte en un trabajo innovador y lleno de propuestas futuras que enriquecen el concepto de muros de visualización. Cada año las propuestas son más ambiciosas, buscando mayor resolución a bajo costo, premisa que hoy en día no se hace imposible de alcanzar. Aunque el término “mejorar la forma de hacer visualización” es importante, este debe ir ligado a los costos y facilidad para un usuario. Es en este punto que los muros de visualización siguen ganando favoritismo, ya que se expone como una solución duradera, flexible en recursos, fácil uso y mantenimiento, bajo costo, y sin sacrificar la usabilidad de la plataforma, pensando siempre en el beneficio de los usuarios. A continuación se citan algunos trabajos implementados

4.1 CONSTRUCCIÓN Y USO DE UN SISTEMA DE MURO DE VISUALIZACIÓN

BUILDING AND USING A SCALABLE DISPLAY WALL SYSTEM [2]

Año: 2000

Desarrollador: Universidad de Princeton

Descripción: Este trabajo tuvo como meta la construcción de un espacio colaborativo que permitiera la completa visualización de imágenes en una pantalla de alto formato. Las ventajas de este proyecto era el bajo costo propuesto comparado con el uso de máquinas con aceleradores de grafico de gama alta o proyectores de gama alta.

Constituido por: clúster construido por cuatro ordenadores, un acelerador de gráficos, proyectores portables y una red Ethernet. La resolución máxima que ofreció este proyecto fue 4.096 * 1.536 Pixeles.

Desventajas: Algunas dificultades presentadas en este trabajo fueron en el ajuste de los proyectores. Estos debían alinearse tarea que implicaba mucho tiempo. Una solución para este inconveniente se desarrolló la técnica *Computational alignment*¹⁰. Esta técnica ajusta la imagen antes de ser proyectada. Solo requiere de un poco de adecuación física del hardware.

A partir de este trabajo se plantearon los siguientes interrogantes: ¿Cómo coordinar los pc y acelerador de gráficos en tiempo real?, ¿Cómo realizar una correcta comunicación entre muchos ordenadores? , ¿Cómo asignar los recursos para lograr para cumplir con la meta pensada?

4.2 JXTAVIEW – UNA HERRAMIENTA PARA LA VISUALIZACION INTERACTIVA DE IMAGENES DE GRAN TAMAÑO EN UN MOSAICO DE MONITORES

JXTAVIEW – A TOOL FOR INTERACTIVE VISUALIZATION OF LARGE IMAGERY ON SCALABLE TILED DISPLAY [11]

Año: 2004

Desarrollador: EVL Electronic Visualization Laboratory y Universidad de Illinois – Chicago

Descripción: JuxtaView es una aplicación para muros de elevada resolución, que permite recorrer un conjunto de imágenes o datos de alta resolución, que residen en un lugar remoto, junto con LambdaRam, la cual es usada para agilizar la salida de las imágenes, y cuya una de sus mayores cualidades es la reducción de la latencia fueron los mecanismos que uso el muro de visualización llamado Geowall2 para ofrecer la eficiencia que planteo como objetivo. Para este año se centró la atención en el incremento y precisión con la que debían estudiarse los datos.

Constituido por: Geowall2 fue un proyecto desarrollado por *EVL*, consistía en un arreglo de 15 monitores LCD ubicados en una malla 5X3. Y el cual ofrecía una resolución total 8000X3600 pixeles. Cada pantalla era conducida por un ordenador. A nivel de hardware cada PC tenía: Tarjeta gráfica nVIDIA Quadro FX3000, 250 GBytes de disco, conexión Ethernet. Una característica importante de esta plataforma fue la inclusión de un sistema altamente escalable, objetivo que se logró empleando JuxtaView y LambdaRam.

4.3 DESARROLLO DE UN MURO DE VISUALIZACION BASADO EN GRID PARA VISUALIZACION EN RED

DEVELOPMENT OF GRID BASED TILED DISPLAY WALL FOR NETWORKED VISUALIZATION [13]

Año: 2005

Desarrollador: Centro nacional de Computación de Alto Rendimiento de Hsinchu, Taiwán.

Descripción: Este proyecto llamado TDW (Tiled Display Wall), centró su atención en el aprovechamiento de los sistemas distribuidos a gran escala. Se logró hacer visualización de datos ubicados en diferentes instalaciones geográficamente remotas.

Constituido por: La arquitectura utilizada se basaba en un nodo que actuaba como maestro, junto a un conjunto de nodos esclavos. A nivel de hardware se formó un muro de 12 monitores en una malla 4X3, en donde cada monitor estaba controlado por un PC. Lo cual hacía que el espacio empleado por toda la plataforma fuera bastante grande. Adicional a los 12 ordenadores se encontraba el nodo maestro que estaba configurado para del usuario.

Ventajas: La visualización se podía realizar: en 3D, imágenes de alta resolución y ejecutando de algoritmos para reconstrucción geométrica en 3D.

Desventajas: Las dificultades encontradas fueron a causa de la estructura de la Grid, requería un motor gráfico exigente y no era escalable.

4.4 METODO PARA LA UTILIZACION PRACTICA DE UN MOSAICO DE MONITORES EN LA VISUALIZACIÓN CIENTÍFICA

METHOD FOR PRACTICAL UTILIZATION OF TILED DISPLAY ON SCIENTIFIC VISUALIZATION [2]

Año: 2010

Desarrollador: en: Tokio, Japón.

Descripción: El principal enfoque para el cumplimiento de este trabajo se basó en el siguiente pensamiento: “Una elevada alta resolución más una recolección de múltiples imágenes, mejoraran la eficacia de la visualización científica”. Para lograr esto se buscó mejorar la velocidad de procesamiento y realizar un incremento en el ancho de banda de la red.

A partir de proyectos realizados en 2007, el tema de desarrollo de grandes plataformas pasó a un segundo aspecto relevante y se empezó a evaluar el consumo de energía de estas plataformas al igual que el interés en las aplicaciones y las necesidades de la ciencia.

Este trabajo apoyó el descubrimiento intelectual dentro de datos numéricos con el fin de analizar eficientemente e inspeccionar datos a través de técnicas de visualización que apoyaran las visión humana y su efectiva interpretación.

Este artículo incluye una tabla comparativa entre las ventajas y desventajas del Middleware que se implementa en este tipo de plataformas, además de la descripción de las aplicaciones desarrolladas.

4.5 STALLION [1]

Año: 2011

Desarrollador: Centro de computación avanzado de Texas.

Descripción: En los últimos años este centro de investigación de la Universidad de Texas, ha logrado desarrollar múltiples propuestas de muros de visualización en donde cada nuevo proyecto tiene un punto innovador y un alcance cada vez mayor. Estos trabajos han sido desarrollados en la medida que el desarrollo tecnológico permite avanzar. Uno de sus trabajos más famosos es conocido como Stallion.

Stallion es un muro de visualización de 328 mega píxeles ofrecidos por un total de 80 monitores Dell 3007WFP de 30 pulgadas. Los cuales hacen que la resolución supere 150 veces la ofrecida por un monitor estándar de alta definición. Estos se encuentran dispuestos en una malla de 16 columnas por 5 fila, permitiendo así que la visualización se realice de forma extendida con dos cualidades muy importantes, alto detalle y calidad comparado con la resolución usual ofrecida por un proyector.

El clúster implementado por el grupo incluye 23 nodos esclavos y un nodo maestro operados bajo el sistema operativo CentOS 6.3. La cantidad de equipos es una cantidad pequeña al ver la dimensión del muro, siendo un clúster que físicamente ocupa poco espacio, comparado con los trabajos antes mencionados.

En total Stallion ofrece a los usuarios 47 unidades de procesamiento gráfico (GPU) de más de 36 GB de memoria de video. Este proyecto se enseñó a los usuarios que realizan exploración de grandes conjuntos de datos geométricos o imágenes de gran resolución, ya que ofrece examinar a gran escala características globales a una distancia considerable, incluyendo varias vistas del mismo conjunto de

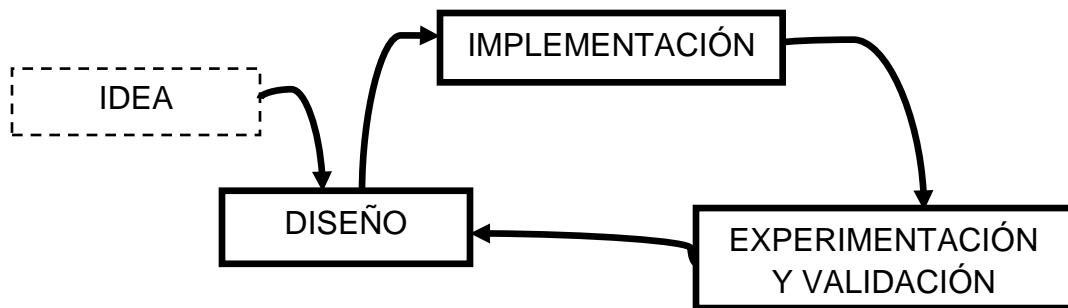
información de forma simultánea. En base a su gran alcance, a la integridad de sus componentes y al apoyo de profesionales en el área de la visualización, se escogió el proyecto Stallion como modelo de desarrollo.

En el siguiente capítulo se describen las etapas de desarrollo de la propuesta de trabajo, que abarca tres puntos de vista: científico (visualización de datos científicos), metodológico (como implementar la plataforma) y didáctico (proceso de aprendizaje durante la implementación de la plataforma).

5. METODOLOGÍA

El proceso que se siguió durante el desarrollo de este trabajo, fue una adaptación del método científico. A través de este, se logran representar las etapas a seguir para el cumplimiento de los objetivos. Las etapas se organizan así:

Figura 7: Método científico



Fuente: ACM Magazine. Disponible en:

<http://cacm.acm.org/magazines/2007/11>

En base a esto, el proyecto se realizó en 4 fases, las cuales se realimentan unas a otras.

Fase 1. IDEA (Documentación y construcción de conceptos).

Esta etapa inició con el proceso de documentación e investigación acerca de trabajos afines a esta área del conocimiento de nuevas tecnologías y supercomputación. La información recopilada permitió consolidar la idea acerca de la utilidad, logros y aportes de una herramienta de visualización de alta resolución, tanto a nivel científico como en el desarrollo de la cotidianidad en términos comerciales específicamente publicitarios. En esta fase, se realizó un mapa cronológico, que permitió a los autores indagar y conocer el proceso e implementación de diversas alternativas de visualización, enriqueciéndose con

cada una de las propuestas. Además de esto, condujo a establecer las ventajas y desventajas de ciertas arquitecturas tanto en hardware como en software.

La literatura para obtener esta información, fue principalmente tomada de bases de datos reconocidas como lo es la *IEEE* y la *ACM*. En éstas se encuentran un conjunto de artículos enmarcados en el tópico de: Comunicación, difusión y creación de redes, cómputo y procesamiento (hardware y software) e ingeniería.

En base a los resultados encontrados, se inicia la segunda etapa de esta misma fase, la cual se basó en la selección de información puntual sobre visualización, estrategias y métodos para la implementación de muros, etcétera.

Durante la selección se hizo un proceso de filtración de la información, debido a que es un tópico muy general y con gran volumen de trabajos realizados a nivel internacional, que aunque tratan el tema de visualización o alta resolución, no van enfocados a el estudio de hardware y software como tal, y a la forma de integrar estos dos grandes componentes para implementar una herramienta de visualización de calidad y valor agregado en el ámbito científico el cual es el principal impacto de ésta tesis.

Resultados Alcanzados

1. Se organizó una carpeta con los artículos afines al proyecto, seleccionados de bases de datos internacionales y ordenados cronológicamente.
2. Se obtuvo la primera definición por parte de los autores acerca del concepto de muro de visualización.
3. Se realizó una documentación acerca de la utilidad de los muros de visualización a nivel científico, la evolución y determinación de inconvenientes que se han presentado durante el desarrollo de los diferentes trabajos hasta la fecha realizados.
4. Reconocimiento de vocabulario especializado relacionado con la supercomputación.

Principales Dificultades:

Por otro lado, uno de los inconvenientes encontrados fue la falta de medios físicos, como libros o revistas científicas que presentaran el desarrollo completo de trabajos de este tipo a nivel institucional. Esta carencia no solo se presenta a nivel local, sino en general, a nivel latinoamericano, esto sustentado, en que la mayor parte de trabajos reconocidos que se encontraron en la web son de universidades y centros de investigación de países como: Estados Unidos y Japón.

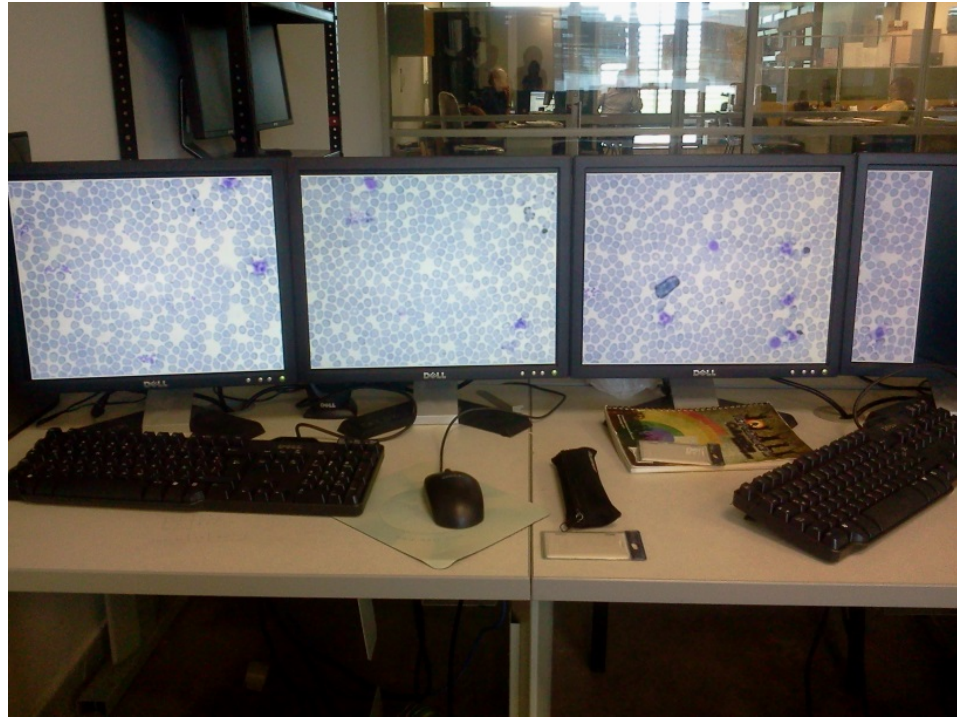
Fase 2. DISEÑO (Diseño adquisición de los recursos físicos faltantes y montaje).

Esta fase inició con la definición del tamaño del muro a implementar. Se estableció el número de filas y columnas, evaluación de los recursos hardware con los que contaba la universidad, espacio total disponible del lugar dispuesto para la plataforma, la fijación de las adecuaciones necesarias respecto las fuentes de energía y de la red de comunicación.

Resultados Alcanzados

1. Se realizó el diseño del primer mapa de infraestructura de la plataforma, el cual utilizaba 4 monitores ubicados en una matriz 2x2. Por falta de recursos para la adecuación de los monitores, este diseño se cambió a un arreglo de una sola fila de 4 monitores. La figura 8 muestra la visualización en este primer diseño de la plataforma, de un extendido de sangre periférica facilitado por el GIIB.

Figura 8: Muro de visualización Unidad de Supercomputación horizontal



Fuente: Autores

2. Obtención del hardware faltante. En base a que es un proyecto que busca usar hardware estándar, fue necesario realizar la gestión para la adquisición de tarjetas gráficas antiguas, que se adecuaron a las arquitecturas de los ordenadores destinados para el proyecto.
3. El diseño realizado sobre la organización de los componentes físicos fueron creado de tal forma que esa primera solución fuese escalable.
4. Enriquecimiento literario: Durante ésta etapa de documentación, los autores centraron los esfuerzos en obtener información de trabajos más específicos en hardware y software, que sirvieron de guía en el desarrollo del trabajo de implementación de la plataforma de visualización. Se priorizó en la realización de un estudio bibliográfico de todos los trabajos realizados por el grupo *TACC (Texas Advance Computing Center)*, principalmente en uno de sus más grandes logros, la plataforma *Stallion*.

Principales Dificultades:

El principal inconveniente de esta etapa, se centró en la selección de las tarjetas gráficas y falta de experiencia para realizar el diseño de la plataforma.

Fase 3: IMPLEMENTACIÓN (Instalación hardware, software y primeras pruebas).

Etapa entendida como el cuerpo del proyecto. Inicia con la selección del sistema operativo. Si bien existía la claridad de uso de software libre, escoger una entre la cantidad de distribuciones existentes a la fecha, fue la primera decisión a tomar para los autores.

Basados en el enfoque y la realización de trabajos que en su momento desarrollaba la unidad de Supercomputación, se seleccionó el sistema operativo *Debian 6.0* basado en el núcleo Linux. Con Debian 6.0 se presentaron problemas de instalación con paquetes bajo los cuales opera el middleware conocido como DisplayCluster, el cual ocasionó retrasos en el desarrollo del proyecto. Se realizó una búsqueda de la librería faltante (libjpeg-turbo), con la cual no se tuvo éxito en la instalación para esta distribución de Linux, razón por la que se empezó a evaluar el cambio del sistema operativo.

Se inició nuevamente el proceso, pero con la distribución de Fedora 17, con esta versión se logró la instalación de todas las librerías necesarias, el acople con la tarjeta gráfica, prestaciones aceptables en el rendimiento de las computadoras, y finalmente y la más importante la instalación total y funcional del middleware.

Se procedió a desarrollar uno de los principales objetivos de ésta fase, que era el estudio del middleware, su interacción con los recursos de hardware y medios no convencionales.

Inspección y conocimiento de los alcances del middleware

Se inició con el desarrollo de scripts que permitieron la interacción con la plataforma de manera más personalizada, según las necesidades de un usuario. Igualmente se desarrolló la construcción de lineamientos bajo los cuales la

plataforma de visualización podría operar. Se visualizaron resultados en la plataforma como aplicaciones, videos e imágenes, probando la funcionalidad de la misma.

Se realizaron video llamadas y video conferencias con la Ingeniera-Magister Karla Patricia Vega Carrillo y el Ingeniero Brandt Westing, del Centro de Computación Avanzado de la ciudad de Texas TACC, con quienes se logró despejar dudas y dar solución de algunos inconvenientes encontrados durante la instalación, configuración y operación del middleware, también se recibieron recomendaciones para el desarrollo de trabajos futuros, haciendo uso de hardware de alto rendimiento.

Resultados Alcanzados

1. Instalación correcta del Middleware.
2. Primera visualización de resultados en la plataforma (Imágenes y Videos).
3. Una vez conseguidos los recursos hardware, se procedió a conectar todas las partes de las computadoras, se organizaron los monitores en fila, y se ubicó el nodo maestro.
4. Se realizó la instalación de las tarjetas gráficas.
5. Se estableció la conexión a puntos de red.
6. Creación del API propuesto.

Principales Dificultades

Las principales dificultades presentadas durante esta fase, se dieron en el proceso de instalación del middleware, para ello se tuvo que probar en dos sistemas operativos diferentes, logrando bajo Fedora 17 el éxito de la instalación luego de un proceso de verificación y modificación de archivos de configuración, en solución a errores presentados en la puesta en marcha del mismo.

También se dio la necesidad de realizar modificaciones sobre el código fuente para poder ejecutar el API desarrollada y la demora en la adecuación de puntos de

energía para la sala donde se implementó la misma, pues se dependía de la División de Planta Física de la UIS para la realización y cumplimiento de dicha labor.

Sin embargo, se logró dar solución oportuna a estos inconvenientes, permitiendo así la continuación del desarrollo del proyecto.

Fase 4. EXPERIMENTACIÓN Y VALIDACIÓN (Validación y pruebas de rendimiento).

En esta etapa final, se realizó la prueba del script que se desarrolló, basado en las necesidades de visualización de la aplicación desarrollada por el grupo de Ingeniería Biomédica de la UIS³³.

En base al comportamiento general de la plataforma de visualización y su alcance, se realizó el desarrollo de una lista de lineamientos para la adopción de futuros trabajos en este campo, o para aquellos que buscan hacer uso centralizado de la herramienta como punto de apoyo en sus trabajos.

Finalmente se realizaron pruebas del rendimiento del clúster. Se procedió a la instalación de una herramienta de monitoreo del rendimiento de un sistema de computación de alto rendimiento, llamado *Ganglia*.

Con el uso de *Ganglia* se busca obtener estadísticas del funcionamiento y rendimiento del clúster, evaluando parámetros como:

1. Uso de la CPU
2. Utilización de la red
3. Uso de la memoria
4. Carga
5. Latencia GPU's
6. Uso de disco

³³ Herramienta computacional para la detección de glóbulos rojos en imágenes microscópicas de células sanguíneas. [En línea]. Consultado [22/01/13] Disponible en: <<http://tangara.uis.edu.co/biblioweb/tesis/2012/144104.pdf>>

Los mismos parámetros se evaluaron para cada una de las arquitecturas implementadas.

En base a esto se estableció un nivel de servicio y prestaciones de la plataforma implementada en el grupo de Supercomputación contra la implementada por grupo Halley. Y así se determinó que tan positivo y útil fue la idea de utilización de hardware estándar.

Resultados Alcanzados

1. Se logró hacer visualización extendida y aumentada de imágenes y videos, además de poder ejecutar muchos de ellos de forma simultánea, sin que esto repercutiera en el rendimiento o en la calidad de lo observado. La figura 9 muestra tres extendidos de sangre periférica proporcionados por el GIIB.

Figura 9: Muro de visualización Unidad de Supercomputación vertical



Fuente: Autores

2. Evaluación de la plataforma mediante la herramienta Gangleia, y análisis de los resultados obtenidos.
3. Cumplimiento a cabalidad de cada uno de los objetivos propuestos en el proyecto.

Principales Dificultades:

Las dificultades encontradas en esta etapa se centraron principalmente en la instalación, configuración y uso de la herramienta Gangleia, debido al desconocimiento de la misma.

A partir de la metodología mencionada anteriormente, se presenta el siguiente capítulo con los puntos clave para el desarrollo exitoso de una plataforma de visualización. Se hace distinción entre dos tipos de soluciones: según los recursos y presupuesto disponible.

6. IMPLEMENTACIÓN DE LA PLATAFORMA

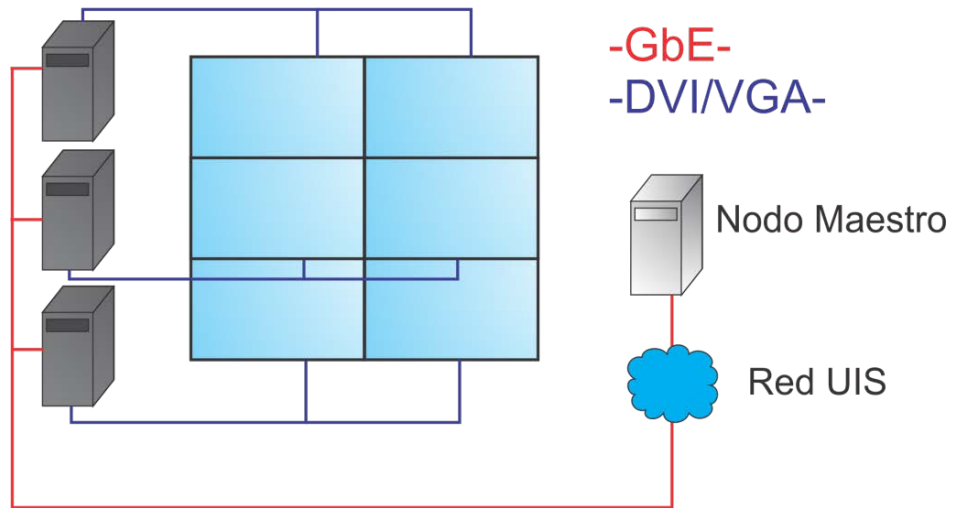
La implementación de la plataforma se divide en cuatro etapas, las cuales cada una de ellas determinan de manera significativa el desarrollo y éxito de su consecuente, estas son: definición del hardware, instalación y adecuación del software, manejo del software y desarrollo del API propuesto para la ejecución de aplicaciones. Existen, actualmente en la Universidad Industrial de Santander, dos prototipos de muros de visualización equivalentes a nivel de software y disyuntos en su configuración de hardware.

6.1 ESPECIFICACIONES Y DEFINICIÓN DEL HARDWARE

MURO DE VISUALIZACIÓN GRUPO DE SUPERCOMPUTACIÓN

La implementación de este prototipo se llevó a cabo con componentes y equipos de la Universidad Industrial de Santander y la Unidad de Supercomputación de la misma. Este muro se caracteriza por poseer recursos de hardware totalmente estándares y para concepto de muchos obsoletos. El sistema se implementó con un clúster Beowulf heterogéneo, constituido por 3 nodos esclavos y un nodo maestro, organizados en una matriz de 3 filas por 2 columnas. La figura 10 muestra una representación gráfica del muro de visualización de la Unidad de Supercomputación.

Figura 10: Muro de Visualización de la Unidad de Supercomputación
6 Monitores



3 Nodos esclavos con conexión
de 1 Puerto DVI / 1 Puerto VGA

Fuente: Autores

Las especificaciones de hardware de estas máquinas se describen en la tabla 3.

TABLA 3: Especificaciones de la plataforma de visualización del grupo de
Supercomputación

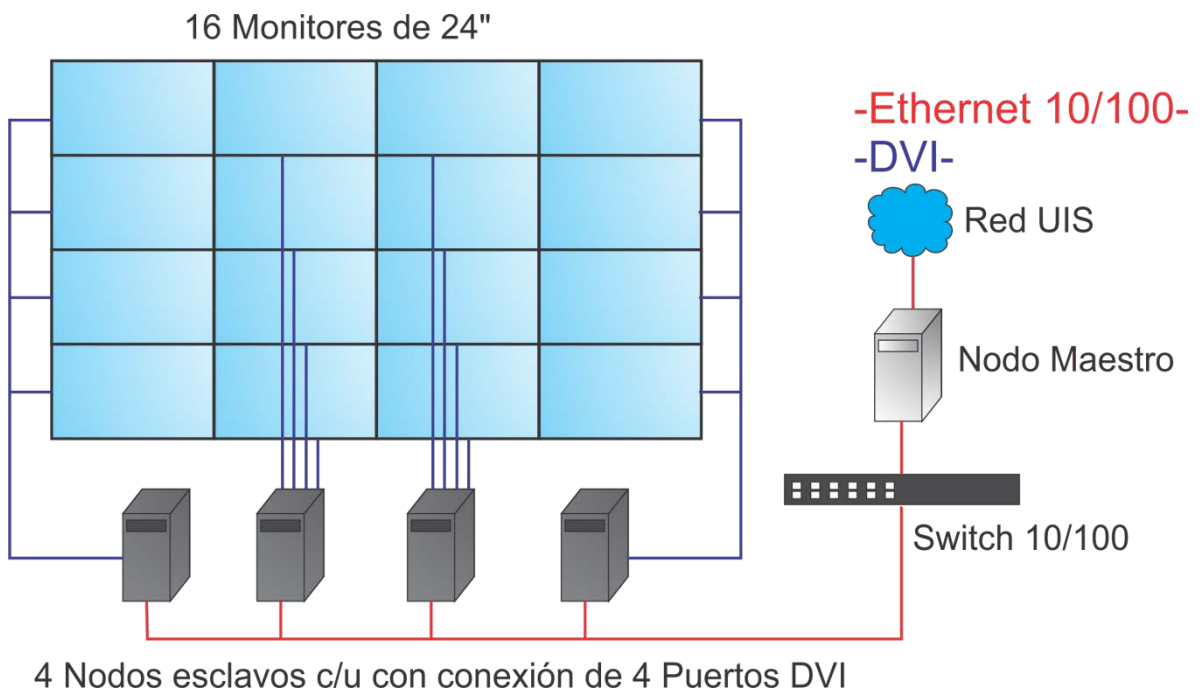
| Nombre del clúster: VisLab SC3 | | Resolución Total: | |
|--------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| | Nodo Maestro | Nodo 1 y 2 | Nodo 3 |
| S.O. | Fedora 17 | Fedora 17 | Fedora 17 |
| Monitores | 1 | 2 | 2 |
| Procesador | Intel® Pentium® 4 CPU 3.20 GHz | Intel® Pentium® 4 CPU 3.20 GHz | Intel® Pentium® 4 CPU 3.20 GHz |
| Memoria RAM | 2 GB | 2 GB | 2 GB |
| Tarjeta Grafica | Intel 82945G | nVIDIA GeForce 9500 GT | AMD Radeon HD 4350 |
| Red | 1 GB Ethernet | 1 GB Ethernet | 1 GB Ethernet |

Fuente: Autores

MURO DE VISUALIZACIÓN GRUPO HALLEY

Esta implementación consistió en un arreglo de monitores, organizados en una matriz 4 X 4 para el cual fue necesaria la compra de un soporte diseñado de acuerdo a los requerimientos de peso del total de monitores y de la formación deseada. Para el logro de este proyecto se realizó la compra de todos los componentes de hardware, buscando obtener un resultado con altas prestaciones con un presupuesto moderado. La figura 11 muestra un mapa de la infraestructura de esta plataforma.

Figura 11: Muro de Visualización Grupo Halley



Fuente: Autores

Las descripciones del hardware y los resultados obtenidos son mejores que los obtenidos con el muro de visualización de la Unidad de Supercomputación.

Catalogado como un clúster Beowulf homogéneo, constituido por cuatro nodos esclavos y un nodo maestro, los cuales tienen especificaciones idénticas que se presentan en la tabla 4.

Tabla 4: Especificaciones del hardware del muro de visualización del Grupo Halley

| Nombre del clúster: VisLab Halley | | Resolución: |
|--|-------------------------------|-------------------------------|
| | Nodo Maestro (1) | Nodo esclavos (4) |
| Sistema Operativo | Fedora 17 | Fedora 17 |
| Monitores | 2 | 4 |
| Procesador | Intel® Xeon® W3565 3.20GHz | Intel® Xeon® W3565 3.20GHz |
| Total Memoria | 4 GB | 4 GB |
| Tarjeta Grafica | nVIDIA® Quadro® NVS 420 | nVIDIA® Quadro® NVS 420 |
| Velocidad de la Red | 100 mbps | 100 mbps |

Fuente: Autores

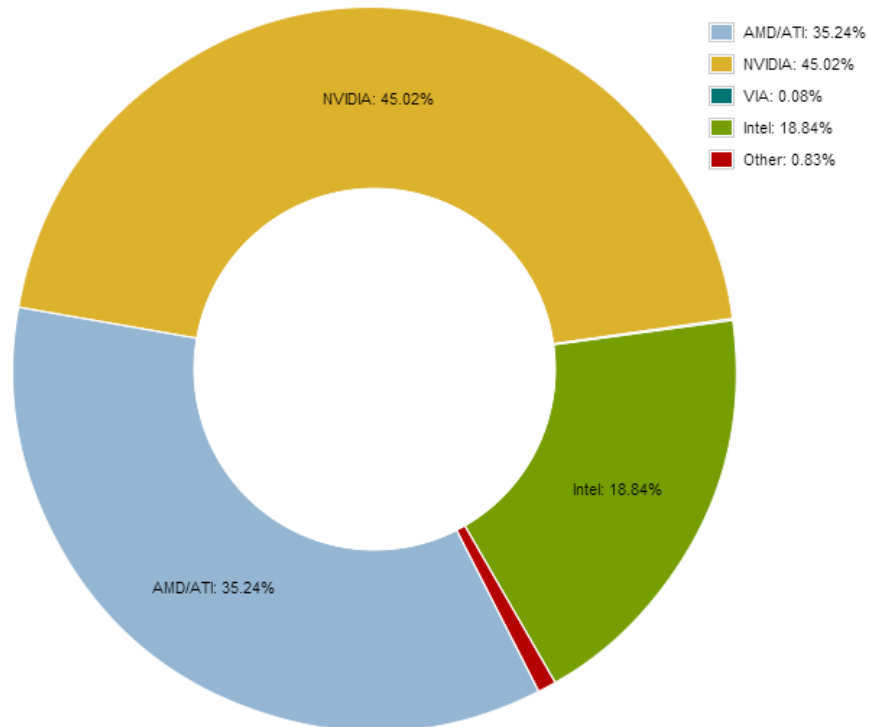
DEFINICIÓN DE LAS TARJETAS GRAFICAS

Algunos puntos que se tuvieron en cuenta para proponer una plataforma funcional y que se adecuara a los recursos disponibles de hardware, fueron pruebas de rendimiento (encontradas en la web) y especificaciones de las tarjetas.

En base a su tendencia de ventas en el mercado encontramos que nVIDIA posee mayor aceptación, siendo este solo un estadístico. No se debe descartar ninguno de los fabricantes existentes, pues todos ofrecen opciones muy competitivas. En la siguiente representación estadística (Figura 12), encontramos los porcentajes de

ventas de los diferentes fabricantes de tarjetas de video en el último mes, lo cual fue un soporte para realizar la elección de este hardware (marzo/13):

Figura 12: Porcentaje de ventas de los diferentes fabricantes de tarjetas gráficas a nivel mundial.



Fuente: PassMark Software Disponible en:

<http://www.videocardbenchmark.net/30dayshare.html>

CLASIFICACIÓN Y PUNTUACIÓN EN BASE AL RENDIMIENTO.

La información de este criterio se encuentra clasificada en cuatro categorías: Las características de hardware, el manejo de datos ofrecidas por las tarjetas, los rangos como se clasifican y categorías. La tabla 5 muestra una clasificación que

realizar PassMark³⁴ Software para las tarjetas gráficas, asignado una puntuación la cual utilizada en la tabla 6 para determinar cuál de las tarjetas utilizadas tiene mejor rendimiento.

Tabla 5: Rendimiento relativo de las tarjetas gráficas.

| Categoría | Rango de puntos |
|-------------------------------|------------------------|
| Gama alta | 8464 - 361 |
| Medianamente gama alta | 361 - 159 |
| Medianamente gama baja | 159 - 67 |
| Gama baja | 67 - 0 |

Fuente: Autores

En la tabla 6 se presentan los resultados obtenidos para las tarjetas gráficas utilizadas, las cuales se catalogan en un rango medianamente alto. Además otras características como núcleos Cuda, velocidad de reloj y versión de OpenGL.

Tabla 6: Especificaciones de las tarjetas de video utilizadas

| Tarjeta | Núcleos Cuda / Procesadores Stream | Velocidad de reloj | OpenGL | Rango de Puntos |
|----------------------------|---|-------------------------------|---------------|----------------------------|
| GeForce 9500 GT | 32 | 550 MHz | 2.1 | 291 |
| Radeon HD 4350 | 80 | 600 MHz | 3.1 | 178 |
| Quadro NVS 420 | 16 | 550 MHz | 3.0 | 362 |

Fuente: Autores

³⁴ PassMark Software [En línea]. Consultado [22/01/13] Disponible en: <http://www.videocardbenchmark.net>

6.2 DESCRIPCIÓN Y REQUERIMIENTOS DE SOFTWARE

El software necesario para el montaje de una plataforma de visualización, varía según el alcance y propósito del mismo. Este libro presenta una serie de requerimientos, que faciliten al usuario el uso de la misma y algunas descripciones básicas de los alcances que puede obtener con la herramienta.

6.2.1 ELECCIÓN DEL SISTEMA OPERATIVO

El desarrollo llevado a cabo por el grupo *Texas Advanced Computing Center* (TACC), basa sus desarrollos en software libre, sin dejar de lado la operatividad de sus proyectos en sistemas como Windows o Mac OS. En este proyecto se eligió Fedora 17, este sistema operativo se caracteriza por ser estable, cuenta con el respaldo y promoción de Red Hat. Adicional a las características anteriores, este sistema incluye en los repositorios todas las dependencias necesarias para la instalación del middleware.

6.2.2 SOFTWARE CONTROLADOR DE LAS TARJETAS

El software necesario para el correcto funcionamiento de las tarjetas gráficas está ligado a su proveedor. En el caso de las tarjetas nVIDIA y AMD, se debe descargar el driver correspondiente a la serie de la tarjeta, el cual se encuentra en la página web de nVIDIA Drivers³⁵ o AMD Graphics Driver³⁶. Allí mismo se debe seleccionar el sistema operativo bajo el cual se va operar la tarjeta y finalmente se realiza la instalación a través de un ejecutable o por consola.

³⁵ nVIDIA Drivers. [En línea]. Consultado [22/01/13] Disponible en: <http://www.nvidia.com/Download/index.aspx?lang=en-us>

³⁶ AMD Graphics Drivers. [En línea]. Consultado [22/01/13] Disponible en: <http://support.amd.com/us/gpudownload/Pages/index.aspx>

6.2.3 MIDDLEWARE – DISPLAYCLUSTER

Software que permite la interacción y conducción de un mosaico de pantallas. A diferencia de otros middleware, DisplayCluster presenta diversas utilidades, un grupo de soporte y la opción de adicionar y/o modificar código fuente, con relativa facilidad. A continuación, se presenta una tabla comparativa del middleware seleccionado junto con otros más populares y con más tiempo de existencia (Tabla 7).

Tabla 7: Comparativo de middleware

| Middleware | Basado en | Sistemas Operativos | Ejecución de Múltiples tareas | Interacción por medios no convencionales |
|-------------------|------------------|----------------------------|--------------------------------------|---|
| Chromium | Librerías OpenGL | Linux, Windows, Unix. | No | No |
| SAGE | Librerías OpenGL | Linux, Windows, Mac OSX. | Si | No |
| DisplayCluster | MPI | Linux, Mac OSX, Windows | Si | Si |

Fuente: Autores

Con base en los resultados anteriores, y la facilidad de interactuar con el middleware, ya sea desde la posición de usuario o administrador de la plataforma se realizó la elección de DisplayCluster.

El proceso de instalación de este software requiere de varias librerías para las cuales es importante revisar la versión, ya que según los repositorios encontrados en el sistema operativo podrían instalar versiones que no son acordes a los requerimientos del middleware. En la tabla 8 se muestra el resumen de las librerías y paquetes necesarios. (Para mayor detalle, de este proceso revisar el manual de instalación en el anexo 1)

Tabla 8: Paquetes básicos para la instalación de DisplayCluster

| Paquete | Versión | Descripción |
|----------------|----------------|---|
| CMAKE | > 2.8 | Herramienta multiplataforma de generación y automatización de código. Diseñada para construir, probar y empaquetar software. Es utilizada para controlar el proceso de compilación del software usando ficheros de configuración sencillos e independientes de la plataforma. |
| MPI | OpenMPI 1.4.1 | Estándar que define la sintaxis y semántica de las funciones que contiene una biblioteca de paso de mensajes diseñada para ser usada en programas que usan múltiples procesadores. Los elementos principales que intervienen en el paso de mensaje son: Proceso que se envía, quien recibe, y el mensaje. |
| QT | > 4.8.0 | Biblioteca multiplataforma, usada para desarrollar aplicaciones con interfaz gráfica de usuario o como herramienta para la línea de comandos y consolas de un servidor. Permite la reutilización de código de manera eficiente, lo cual permite el paso más ágil entre múltiples plataformas. |

| Paquete | Versión | Descripción |
|----------------------|-----------|--|
| BOOST | > 1.4.4.0 | Conjunto de bibliotecas de software libre y revisión por pares preparadas para extender las capacidades del lenguaje de programación c++. Actualmente Boost está formada por más de 80 bibliotecas individuales, incluidas las bibliotecas de algebra lineal, multi-hilos, procesamiento de imágenes, entre otras. |
| FFMPEG | > 0.8.x | Colección de software libre que puede grabar, convertir, y hacer streaming de audio y video. FFMpeg está desarrollado es GNU/Linux, pero puede ser compilado en la mayoría de sistemas operativos. Incluye libavcodec, una librería de codecs, lo cual le da la capacidad de elegir el códec según la extensión del video. |
| LIBJPEG-TURBO | 1.1.90 | Es un códec de imágenes JPEG que utiliza instrucciones SIMD para acelerar la comprensión y descompresión. Libjpeg-turbo, implementa muchas características del API libjpeg tradicional, con la diferencia que adiciona más características que lo hacen de 2 a 4 veces más rápido. |

Fuente: Autores

6.2.4 Otras Aplicaciones

En los numerales anteriores, se desplegaron los paquetes fundamentales, para el funcionamiento general del middleware. No obstante para el funcionamiento óptimo de la plataforma y la facilidad al usuario final se buscó reducir las

intervenciones que tuviesen que realizar en relación a la operación interna del muro de visualización, para esto se instalaron los siguientes programas:

NFS (Network File System): Es un protocolo de capa de aplicación, según el modelo OSI (Open System Interconnection), utilizado para sistemas de archivos distribuidos en un entorno de red de computadoras de área local. Posibilita que distintos nodos conectados a un sistema de red accedan a ficheros remotos como si se tratara de archivos locales³⁷. A continuación se enuncian el problema encontrado seguido de la justificación del porque usar NFS.

Problemática encontrada en el middleware: DisplayCluster es un middleware que basa su operación en el paso de mensajes, a través de MPI. Sin embargo, cuando se envía un trabajo, es decir, al lanzar una imagen o video, es necesario que este se encuentre ubicado en la misma ruta o directorio para cada nodo. De lo contrario el conjunto de pantallas controladas por ese nodo ejecuta la tarea de ejecutar el middleware, pero no transmitirá ninguna imagen.

Usando NFS: Al usar este protocolo se compartió una carpeta desde el nodo maestro hacia todos los nodos esclavos. Esta carpeta contiene todos los archivos que quieren ser visualizados en la plataforma. En caso de que nuevos usuarios de la plataforma deseen visualizar sus resultados o investigaciones, ellos no deben preocuparse porque estos se encuentren almacenados en cada uno de los nodos del clúster, ya que al ser copiados en la carpeta local del nodo maestro estos son actualizados de forma automática en el resto de nodos.

(Para más detalles, de la configuración del servidor NFS, revisar el manual de instalación.)

SSH (Secure Shell): Protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios

³⁷ [En línea]. Consultado [22/01/13] Disponible en: <<http://nfs.sourceforge.net/nfs-howto/>>

conectarse a un host remoto. La cualidad de SSH sobre otros protocolos con su misma funcionalidad, es que este encripta la sesión de conexión haciendo imposible que alguien pueda obtener contraseñas no encriptadas. Además de conexión a otros dispositivos, SSH permite copiar datos de forma segura, gestionar claves RSA (con el fin de evitar escribir claves al conectar a los demás nodos del clúster) y pasar los datos de cualquier otra aplicación por un canal seguro³⁸.

Problemática: VisLab, es una plataforma de visualización que debe ser operada desde una sola computadora, es decir, desde el nodo maestro. Por esto al ejecutar el middleware se presentaban demoras e incomodidades, ya que se requería ingresar la contraseña de usuario de los nodos esclavos para iniciar DisplayCluster. En caso de copiar archivos o directorios a otros nodos se requería de un dispositivo de almacenamiento externo.

Usando SSH: Se logró gran utilidad al instalar el servidor SSH. Primero, se logró un acceso transparente una vez se inicia el middleware. Segundo, se realizó de forma directa, el traspaso de archivos de configuración del middleware y otras aplicaciones, reduciendo los tiempos de instalación y configuración de la plataforma. Tercero, para la configuración futura de la plataforma se ofreció este servicio por el cual el administrador de la misma pueda controlar, modificar, y eliminar cualquier información del disco local de algún nodo esclavo de forma remota.

Ganglia Monitoring System: Es un sistema distribuido de control escalable para los sistemas de computación de alto rendimiento. Se basa en un diseño jerárquico dirigido a las federaciones de clúster. Esta herramienta hace uso de tecnologías muy usadas hoy en día como son: XML para la representación, XDR para el

³⁸ OpenSSH. [En línea]. Consultado [22/01/13] Disponible en: <<http://www.openssh.org/>>

transporte compacto y RRDtool para el almacenamiento de datos y su visualización³⁹.

Uso en VisLab: Ganglia se usó en la plataforma como herramienta para medir el rendimiento de cada nodo, y así conocer parámetros como el uso de la CPU, memoria y red en diferentes periodos de tiempo. A partir de estos resultados puede realizarse un estimativo de la operatividad brinda por el clúster. Para la obtención de los parámetros anteriores es necesario la instalación de dos demonios, estos son: Gmond y Gmetad. El primero de ellos, es un demonio multi-hilo que se ejecuta en cada nodo del clúster que se quiere supervisar. Las funciones principales de este son: monitorear cambios en el estado del nodo, anunciar cambios pertinentes, escuchar el estado del resto de nodos del clúster, y responder solicitudes XML del estado del clúster. El segundo, es un demonio que solo se ejecuta en el nodo maestro, o nodo que va a presentar los resultados de trabajo. Gmetad, tiene como función principal, la recolección periódica de los resultados de cada nodo, a su vez el análisis y almacenamiento de los mismos.

Adicional a esto, se debe realizar la instalación de Ganglia web. Esta herramienta proporciona una vista de la información obtenida, en tiempo real y en un medio cotidiano para ver información, como lo es una página web. Ganglia Web, está dirigido a administradores de sistemas y usuarios.

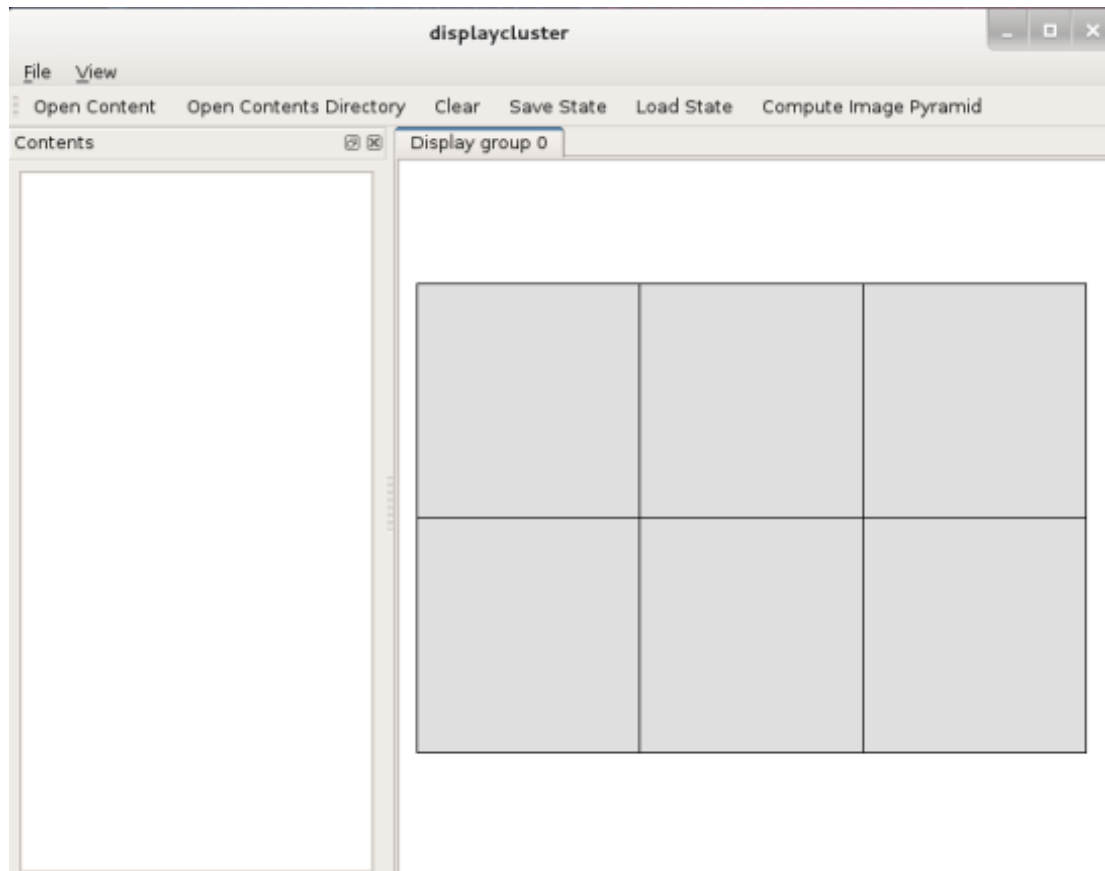
6.3 FUNCIONAMIENTO DE LA PLATAFORMA

DisplayCluster es una herramienta fácil de utilizar con una interfaz gráfica relativamente sencilla, y que posee muchas funciones. De acuerdo a la tarea que se quiere lanzar, o la forma como se desea trabajar, siempre será útil tener en

³⁹ Ganglia Monitoring System. [En línea]. Consultado [22/01/13] Disponible en: <<http://ganglia.sourceforge.net/>>

cuenta las formas de conllevar las tareas. La figura 13 enseña la interfaz inicial del middleware, a partir de la cual se realiza la interacción con el trabajo lanzado.

Figura 13: Interfaz gráfica de usuario de DisplayCluster



Fuente: Autores

La cuadrícula de color gris, dependerá del archivo de configuración, el cual en un ambiente de desarrollo debe simular la forma (orientación horizontal - vertical) y tamaño del muro a escala real. Este archivo de configuración es bastante importante, ya que incluye varios parámetros que pueden facilitar el trabajo de un usuario. En la Figura 14 se ilustra un prototipo de este archivo y posteriormente la explicación del mismo.

Figura 14: Archivo de configuración DisplayCluster

```
<configuration>

<dimensions numTilesWidth="2" numTilesHeight="2" screenWidth="400"
screenHeight="400" mullionWidth="50" mullionHeight="50" fullscreen="0"/>

<process host="localhost" display="0">

<screen x="0" y="0" i="0" j="0"/>

<screen x="400" y="0" i="1" j="0"/>

</process>
```

Fuente: Autores

PARÁMETROS

NumTilesWidth / numTilesHeight: Representan la matriz de monitores, es decir el número de pantallas horizontales y verticales totales que conforman el arreglo.

ScreenWidth / screenHeight: Representa la resolución de cada monitor. Esta debe obtenerse de las especificaciones de las pantallas.

Fullscreen: Puede tomar el valor de cero o uno, y representa si la ventana de cada monitor debe ser abierta en pantalla completa o no.

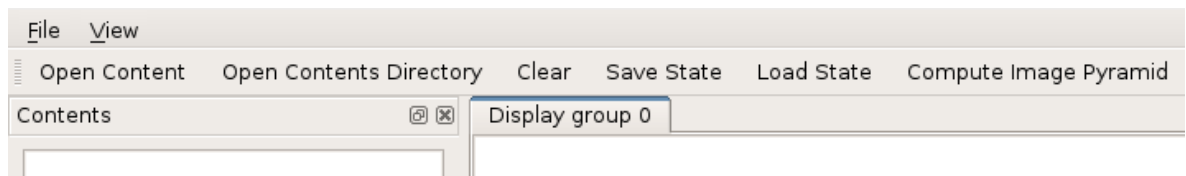
Etiqueta process: Esta etiqueta se realiza según el número de nodos de trabajo. Indica el nombre de cada nodo.

Etiqueta screen: esta etiqueta contiene la información de cada monitor y su posición. Esta etiqueta fija la representación de los píxeles por monitor, es decir, se define los límites de cada pantalla.

La correcta manipulación de los parámetros es de vital importancia, ya que un cambio en la plataforma debe quedar registrado en este archivo. Adicionalmente, el dominio de este sirve para realizar pruebas de la forma más óptima de ubicar las pantallas, según el tipo de imágenes que se van a lanzar.

Continuando con la interfaz de usuario y las funcionalidades del Middleware, se menciona a una breve descripción de la barra de herramientas con la cual los usuarios interactúan.

Figura 15: Menú principal y Barra de herramientas - DisplayCluster



Fuente: Autores

Open Content: Es la opción de mayor importancia en DisplayCluster. Esta opción permite cargar una imagen o video ubicado en cualquier posición del disco. No obstante, debe tenerse presente que el archivo seleccionado debe estar compartido en todo el clúster.

Una vez cargada una imagen o video, se pueden realizar acciones como hacerle zoom haciendo uso del Scroll del mouse, moverla por todo el arreglo de pantallas, cargar una nueva imagen, cerrar una imagen.

Open Contents Directory: A diferencia de la opción de abrir un contenido, esta opción carga es un conjunto de imágenes o videos almacenados en un directorio.

Evitando con esto que un usuario tenga que cargar uno a uno los elementos que quiere visualizar. Los parámetros de entrada son tres: directorio, dimensión en x, dimensión en y. Los dos últimos parámetros indican el tamaño de la cuadrícula. Es decir fijan el sector del muro que se va a utilizar para esa tarea específicamente.

Clear: opción útil para limpiar el área de trabajo. Aunque el middleware permite cerrar elemento por elemento esta opción resulta de mucha utilidad en los casos en los cuales el número de imágenes o videos es alto. Y cuando se tiene la seguridad que no van a ser utilizados nuevamente.

Save/Load State: Esta opción permite guardar en cualquier momento el desarrollo de un trabajo. Esta opción genera un archivo dcx el cual contiene toda la información de los elementos que se encuentran sobre el muro, su posición y todos los detalles de un momento específico de trabajo.

Adicional a las opciones antes nombradas, DisplayCluster tiene dos utilidades que deben aplicarse con cuidado.

1. **Habilitar/deshabilitar Constraint Aspect Ratio:** Esta opción permite que a una imagen se le realice zoom en un solo sentido. Es decir, que la imagen pueda aumentarse solo verticalmente u horizontalmente. Por defecto esta opción viene deshabilitada ya que es la causante de que se pierda la relación y calidad de la imagen. Sin embargo bajo cierto tipo de imagen es de mucha utilidad.
2. **Habilitar/deshabilitar Show Window Borders:** Permite delimitar una imagen de otra, al habilitarla sobre el muro de visualización cada imagen es delimitada por un borde. Esta opción viene por defecto habilitada, ya que permite claridad del elemento que se está visualizando. Sin embargo en imágenes que presentan cierta continuidad sería un inconveniente.

6.4 API DESARROLLADA

El API desarrollada tuvo como finalidad la construcción de un puente entre una aplicación y el middleware. En base a eso, se desarrolló un script en bash, que permite las siguientes opciones:

1. Listar archivos o directorios compartidos mediante el servidor NFS.
2. Agregar una imagen
3. Agregar los archivos de un directorio (imágenes).
4. Ejecutar DisplayCluster cargando los archivos agregados posteriormente.
5. Finalizar la aplicación.

A diferencia de la forma como opera DisplayCluster cuando se carga una imagen o directorio, el API desarrollada mantiene el tamaño original de la imagen para no perder resolución de la misma y así aprovechar al máximo los recursos de hardware.

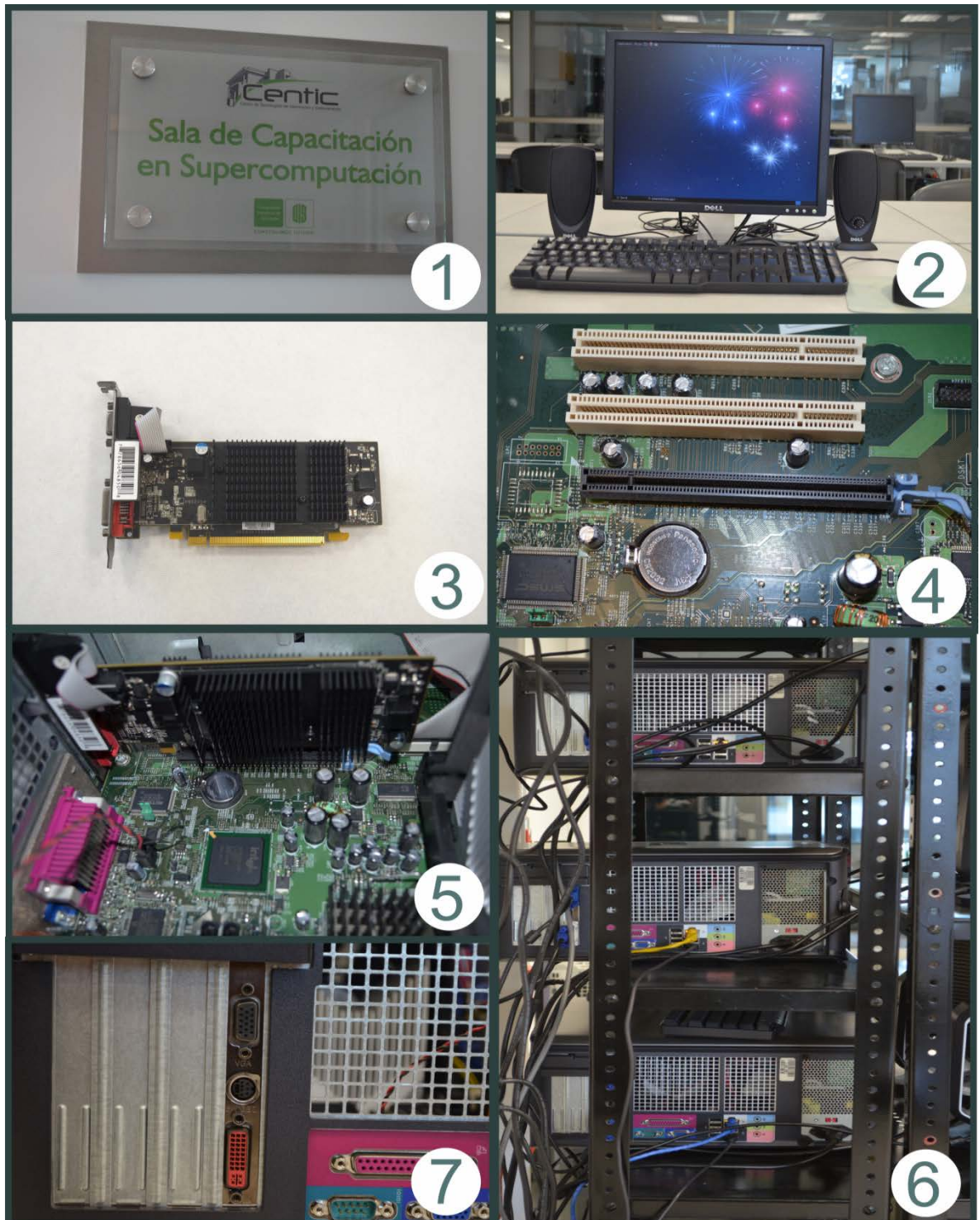
El API desarrollada fue probado por los resultados obtenidos de una aplicación realizada en el Grupo de Investigación en Ingeniería Biomédica GIIB, la cual realiza detección de glóbulos usando técnicas de procesamiento y análisis de imágenes microscópicas⁴⁰. Esta aplicación fue desarrollada en Matlab como lenguaje de programación. Se encontraron resultados satisfactorios por parte los interesados en visualizar las placas de sangre en mayor tamaño.

En el anexo 2, se encuentra el código fuente del API y la forma como debe ejecutarse.

⁴⁰ Herramienta computacional para la detección de glóbulos rojos en imágenes microscópicas de células sanguíneas. [En línea]. Consultado [22/01/13] Disponible en: <<http://tangara.uis.edu.co/biblioweb/tesis/2012/144104.pdf>>

6.5 ILUSTRACION PASO A PASO DEL MONTAJE REALIZADO

Figura 16: Ilustración del montaje realizado (Parte 1)



Fuente: Autores

Figura 17: Ilustración del montaje realizado (Parte 2)



Fuente: Autores

La figura 16 y 17, es una representación general del paso a paso seguido durante el desarrollo del proyecto. La siguiente lista es una sencilla descripción de cada una de las imágenes.

1. Laboratorio en donde actualmente se encuentra instalado el muro de visualización.
2. Nodo maestro, con todas sus partes.
3. Identificación del puerto PCI en la tarjeta gráfica.
4. Ubicación del lugar de conexión entre la tarjeta gráfica y la motherboard.
5. Instalación de la tarjeta gráfica en cada nodo esclavo.
6. Conexión de todos los componentes de hardware necesarios en los nodos esclavos.
7. Identificación de las salidas VGA y DVI de la tarjeta gráfica.
8. Ubicación de los monitores de acuerdo al nodo que los opera.
9. Conexión de dos monitores a cada nodo, en los puertos disponibles por las tarjetas gráficas.
10. Reconocimiento de los puntos de energía disponibles.
11. Instalación del software base para el funcionamiento del equipo. Aquí se incluye: sistema operativo, navegador, drivers, software para compilación, ajustes de red, entre otros.
12. Instalación de DisplayCluster y su configuración respectiva. Se incluye: middleware, SSH, servicios NFS, Ganglia, software para compilación de paquetes, entre otros.
13. Arreglo de monitores, en una matriz 3 filas por 2 columnas.
14. Plataforma completa, en modo operativo.

7. LINEAMIENTOS

La construcción y modificación de una plataforma de visualización como cualquier proyecto requiere de una previa planificación de actividades con el fin de obtener los mejores resultados de forma eficiente.

A continuación se mencionan recomendaciones con el fin de orientar a quienes deseen implementar plataformas afines a esta y ejecutar aplicaciones en la misma. Los pasos descritos se basaron en la experiencia obtenida durante la etapa investigación e implementación de la plataforma.

7.1 LINEAMIENTOS PROPUESTOS PARA LA CONSTRUCCION O MODIFICACION DE PLATAFORMAS DE VISUALIZACION

Para la construcción de la plataforma se recomienda seguir los siguientes pasos:

- 1. Definir el alcance:** antes de proponer el hardware y software de la plataforma, se debe puntualizar cuál será su uso. A partir de ello se puede aclarar qué tipo de solución implementar (monitores o proyectores) dependiendo del nivel de detalle en megapíxeles, división entre los monitores, interacción con el usuario con dispositivos no convencionales, experiencia en 3d, tamaño (que puede ser expresado en metros o megapíxeles), orientación de la matriz de monitores o proyectores, entre otros. Además es conveniente realizar un mapa de infraestructura de la plataforma.
- 2. Elegir el middleware:** en las dos soluciones (monitores o proyectores) los middleware investigados se comportan satisfactoriamente. Las principales diferencias son respecto a la interacción con el usuario, requerimientos de hardware, arquitectura requerida para su instalación y correcto funcionamiento, escalabilidad, sistema operativo, costo, entre otros. En caso que ninguna de

las soluciones existentes sean la requerida por los usuarios o se adapten a las condiciones anteriores es necesario crear el middleware que opere la plataforma de visualización.

3. **Seleccionar el hardware y otros elementos para la construcción de la**

plataforma: este paso está limitado por el alcance de la plataforma, presupuesto disponible, tecnología a utilizar, entre otros. Las siguientes recomendaciones se realizaron con el fin de obtener los mejores resultados:

- a) Monitores/Televisores con bisel delgado: Para el caso que la plataforma de visualización sea construida con monitores, se recomienda que el borde o bisel de los mismos tenga el menor ancho posible, pues el observador puede encontrar esto como una dificultad cuando desee interpretar los resultados visualizados en la plataforma. Además se debe definir el tamaño en pulgadas de los monitores, resolución deseada y conectividad.
- b) Selección de un soporte: se pueden contemplar dos opciones, soporte de piso o pared. Los soportes de pared como el caso del VESA (que son utilizados frecuentemente para televisores), deben ser instalados con gran precisión con el fin de no dejar espacios entre los monitores. Otra opción son los soportes de piso con o sin ruedas, en este paso debe ser contemplada la escalabilidad, para determinar cuál solución permite añadir o quitar monitores y/o nodos del clúster.
- c) Fijar el tipo de interconexión de red: Existen varias arquitecturas siendo *Infiniband* la que posee mayor ancho de banda y baja latencia. Se aconseja utilizar el estándar más rápido de Ethernet, 10 Gigabit. Este ítem también está delimitado por los requisitos del middleware, en caso de DisplayCluster se observó mayor uso de la CPU por el estándar utilizado para comunicarse entre los nodos, MPI.
- d) Selección de los ordenadores: en base a los resultados obtenidos favorece el uso de procesadores de alto rendimiento como Intel Xeon. Con respecto

a los demás componentes de la maquina DisplayCluster no exige altas prestaciones, aunque los resultados obtenidos se ven limitados a esto.

- e) Tarjetas Gráficas: estas se encargan de transformar las señales digitales producidas en el ordenador en una señal analógica que sea interpretada por el monitor. Poseen varias salidas como SVGA/DSub-15, DVI, HDMI, DisplayPort, S-Video, entre otras dependiendo del fabricante, las cuales son utilizadas para conectar los monitores. No se recomienda utilizar bifurcadores, debido a que la resolución en pantalla se ve afectada. La resolución también está limitada por la tarjeta de video. Se pueden utilizar varias tarjetas gráficas por nodo.

En caso que se utilice un clúster, se recomienda que este sea homogéneo con el fin de no sobrecargar ningún nodo y utilizar óptimamente todos los recursos.

Con respecto a la modificación de la plataforma esta puede ejecutarse mediante la inclusión de nodos o tarjetas gráficas de los nodos existentes con el fin de aumentar la matriz de monitores. En caso de adicionar uno o más nodos, deben ser uniformes en cuanto al sistema operativo y el middleware utilizado.

7.2 LINEAMIENTOS PROPUESTOS PARA LA CONSTRUCCION O MODIFICACION DE APLICACIONES QUE USEN LA PLATAFORMA DE VISUALIZACIÓN

Así como para la construcción de la plataforma de visualización, las aplicaciones que van a hacer uso de esta depende de la arquitectura de hardware y software, siendo la cantidad de nodos los componentes esenciales de esta.

- **Único nodo:** En este caso existe una gran ventaja, debido a que las aplicaciones se van a ejecutar sobre un único nodo, lo que permite que cualquier aplicación de escritorio sea ejecutada sobre la plataforma de

visualización, esto debido a que el driver o controlador de la tarjeta de video se encarga de esta tarea.

- **Clúster de computadoras:** de esta forma las aplicaciones que van a ser ejecutadas en la plataforma se ven limitadas al funcionamiento del middleware, debido a que este es el que va a controlar el muro de visualización.

En caso de que las aplicaciones generen como resultado archivos independientes (imágenes o videos), estos pueden ser visualizados en la plataforma, usando la interfaz del middleware o haciendo un puente entre la aplicación que genera estos resultados y el middleware que opera el muro de visualización.

DisplayCluster como se mencionó anteriormente permite la interacción con scripts de Python o cargar estados mediante archivos con estructura XML. Esto permite que las aplicaciones usen esta ventaja como alternativa para realizar el puente con la aplicación como se realizó en el API desarrollada en este proyecto de investigación. Además permite “*DesktopStreaming*”, para compartir la pantalla en la plataforma de visualización desde cualquier equipo que cuente con la aplicación dispuesta por el TACC, su principal ventaja es la compatibilidad con los sistemas operativos Windows, Linux y Mac OS X.

Desarrollo de aplicaciones con Python:

1. Asegúrese de la instalación correcta de las librerías y paquetes que requiere DisplayCluster para ejecutar programas en este software (ver anexo 1).

2. El desarrollo de un aplicativo en Python conlleva un estructura, este archivo puede ser creado con cualquier editor de notas, y guardar este como un archivo .py
3. La ejecución de un archivo .py debe realizarse ejecutando el middleware en la opción la pequeña consola que se despliega al ejecutar python de la barra de herramientas.

En conclusión las aplicaciones generalmente no están diseñadas para ser ejecutadas y visualizadas en un clúster de computadoras, por la cual, una opción muy viable es crear la aplicación que interactúe directamente con los recursos de hardware disponibles.

8. PRUEBAS Y ANÁLISIS DE RESULTADOS

La visualización de datos es una acción, que exige precisión y particularmente altas resoluciones, que faciliten la obtención de detalles y la continuidad de una imagen. Con el fin de garantizar esos requerimientos se llevaron a cabo pruebas que sirvan como medidas del rendimiento esperado, es decir, tener un antecedente de los resultados esperados bajo diversas condiciones de trabajo.

La primera prueba se enfoca a validar la correcta comunicación entre el nodo maestro y cada uno de los nodos esclavos. Igualmente, se probó el funcionamiento de la red, y la cantidad de megabits que puede transmitir. En la segunda parte, se buscó detallar el comportamiento del clúster durante un periodo de tiempo, evaluando en cada caso diversos trabajos y como la arquitectura desarrollada se comporta en esos casos. En general, las pruebas realizadas tuvieron como finalidad evaluar la plataforma de visualización y como esta cumple y satisface las necesidades de presentar datos en un monitor, permitiendo estudiar y analizar detalladamente información.

Finalmente, se fijaron los límites de uso de la plataforma, determinando el comportamiento esperado por cada una de las posibles entradas (imágenes, videos, documentos, etcétera); se llevó a cabo bajo el uso de dos herramientas:

1. Herramienta “*lperf*”: Simula el flujo de datos TCP y UDP y mide el rendimiento de la red.
2. Herramienta “*Ganglia*”: Permite obtener información periódica del rendimiento de cada nodo del clúster, y de su funcionamiento con una perspectiva holística, es decir como un conglomerado de nodos conectados.

En los numerales 8.1. Rendimiento de la Red y 8.2. Rendimiento del Clúster se muestran los resultados y respectivos análisis de los parámetros por las herramientas de evaluación de rendimiento sobre el muro de visualización desarrollado en el Grupo de Astronomía y Ciencias Aeroespaciales - Halley.

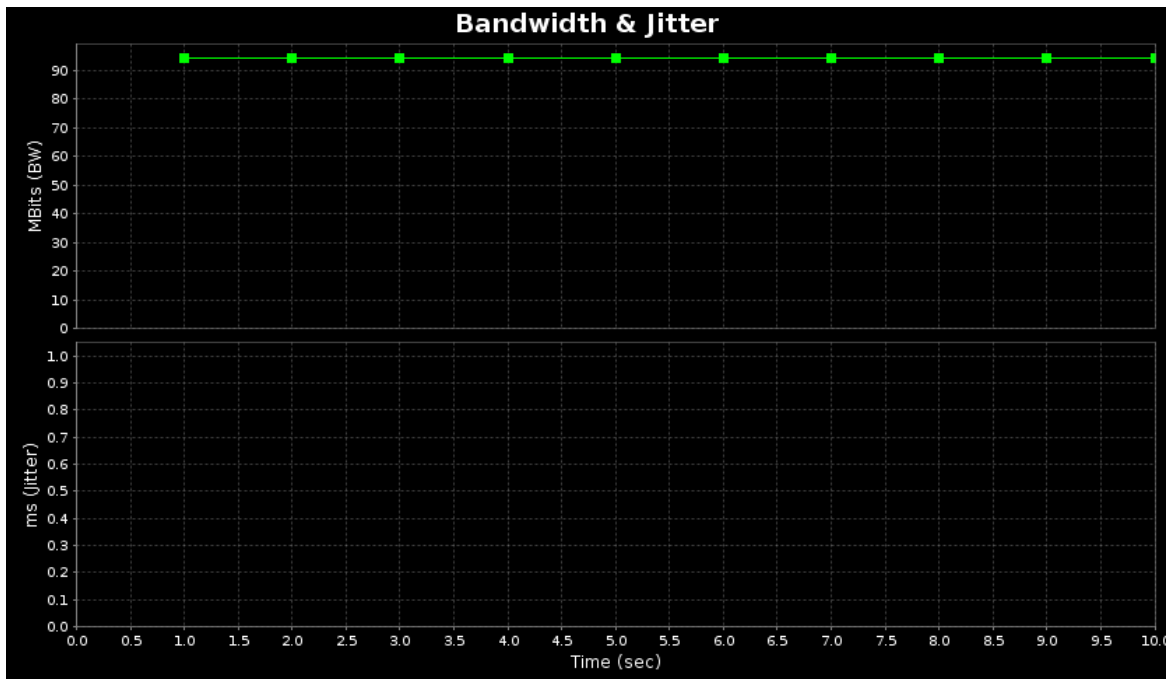
8.1 RENDIMIENTO DE LA RED

El flujo de datos que viaja por la red, junto con la capacidad de la misma, es un criterio vital para una plataforma de visualización, como la que aquí se presenta. La razón básica de esto, es que su arquitectura, integra las partes que la conforman a través de una conexión a una red. En el caso particular del muro VisLab – Halley, la red Ethernet se limitó por un switch 10/100 Mbits.

La figura 18 y 19, son el resultado de una simulación del flujo de datos TCP a través de la red descrita, para las cuales se varió el número de nodos a los cuales se les enviaban paquetes.

La figura 18 deja ver, como al tener envió de datos a un solo nodo, este toma aproximadamente el máximo permisible, es decir, 100 Mbits, manteniendo un flujo constante de datos. En el caso particular de esta simulación, los datos obtenidos, mientras el servidor escuchaba, fueron transferencias de paquetes de 11.2 MBytes, los cuales usaron un ancho de banda constante, de 94.1 Mbits/sec durante los 10 segundos que duro la simulación.

Figura 18: Flujo de datos TCP para un solo nodo

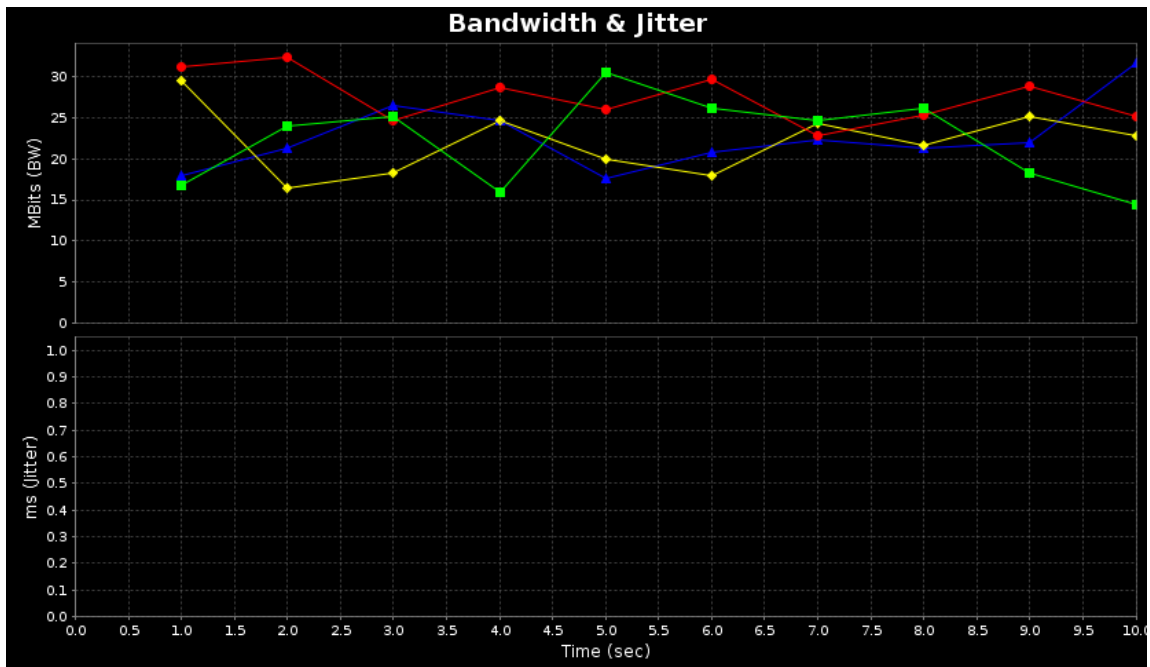


Fuente: Autores

La figura 19, muestra cómo oscila el envío de paquetes para un ancho de banda fijo. Es decir, se hace una división de los recursos de tal forma que se garantice la comunicación. Estos resultados son muy útiles, ya que sirven como respaldo y confiabilidad de la adecuada interacción de todos los nodos en cada intervalo de tiempo.

Los datos específicos para un tiempo t , en esta prueba son: El nodo 1, transfiere 2,26 MBytes en un ancho de banda de 19.0 Mbits/sec, el nodo 2, transfiere 2,50 MBytes en un ancho de banda de 21.0 Mbits/sec, el nodo 3 transfiere 2,65 MBytes en un ancho de banda de 22.2 Mbits/sec, el nodo 4 transfiere 3.95 MBytes en un ancho de banda de 33.1 Mbits/sec.

Figura 19: Flujo de datos TCP para el total de nodos en el clúster



Fuente: Autores

8.2 RENDIMIENTO DEL CLUSTER

El uso de la CPU, el comportamiento de la memoria y la red, son parámetros que ayudan a determinar el rendimiento de un clúster. Estas medidas se han convertido en una gran utilidad para administradores y usuarios de una plataforma como la que acá se implementó, debido a que VisLab Halley, cuenta con una arquitectura con capacidades limitadas dado el hardware que lo conforma. En base a esto, se realizaron pruebas con el fin de determinar la capacidad de trabajo máxima y mínima que ofrece la plataforma de visualización.

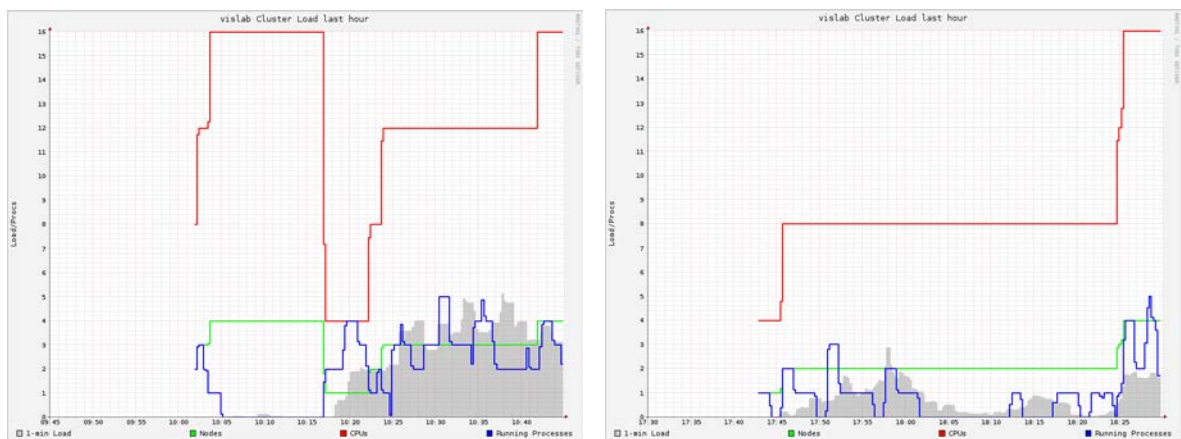
Las siguientes gráficas, son los resultados obtenidos del comportamiento del clúster desde el proceso de arranque del middleware, y la ejecución gradual del lanzamiento de diferentes actividades al muro tales como: cargar imágenes

locales, videos, conferencia en tiempo real, uso de la plataforma a través de un acceso remoto, etc.

Los parámetros observados fueron tres: Carga de procesos en el clúster, uso de Bytes de memoria RAM del cluster, Bytes/segundos transmitidos por la red. Estas medidas se estudiaron con diferentes recursos de red. La primera de ellas, es la arquitectura original con la cual opera el cluster del Grupo de Astronomía y ciencias aeroespaciales Halley, es decir con un Switch 10/100 Mbps. Para la segunda, se empleó un Router Cisco modelo WRVSS4400N de un Gigabit.

El lanzamiento de imágenes se realizó partiendo de menor a mayor en peso en Megabits, se lanzaron 12 imágenes, así: se carga la imagen se realiza un desplazamiento y zoom, se cierra y así sucesivamente. Finalmente se envía un video. Las pruebas se realizaron con imágenes y videos que oscilan entre 10 MB a 403.8 MB.

Figura 20: Carga de procesos en el clúster

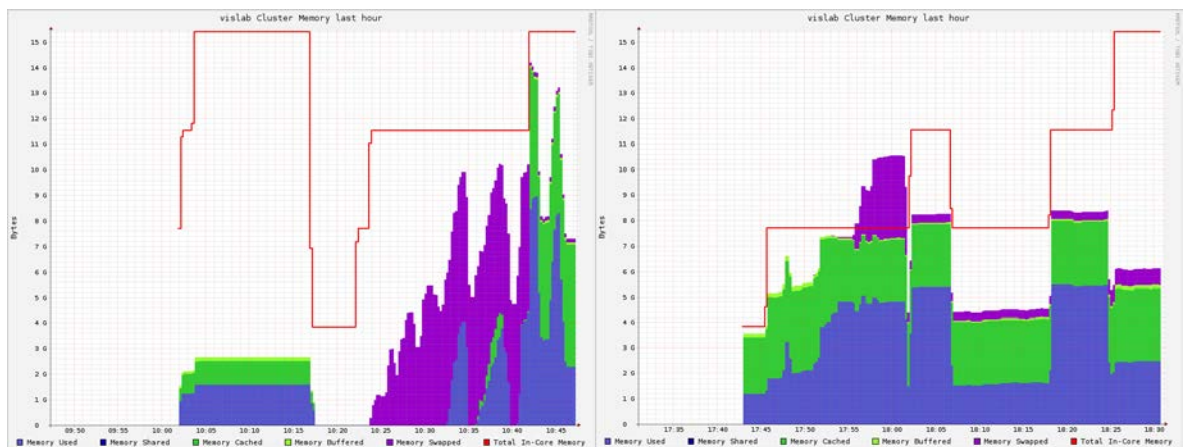


(2)

Fuente: Autores

La figura 20, muestra el uso del procesador en el clúster. En el caso 1, la carga de procesos sube a 16, con una imagen de bajo tamaño y se mantiene constante en este proceso. Solo cae en el momento en que la plataforma queda estática. A diferencia del caso dos, en el que el paso de mensajes se hace de manera más eficaz, lo cual hace que use la mitad de carga que en el caso 1. La carga de procesos se eleva a 16 al usar el streaming.

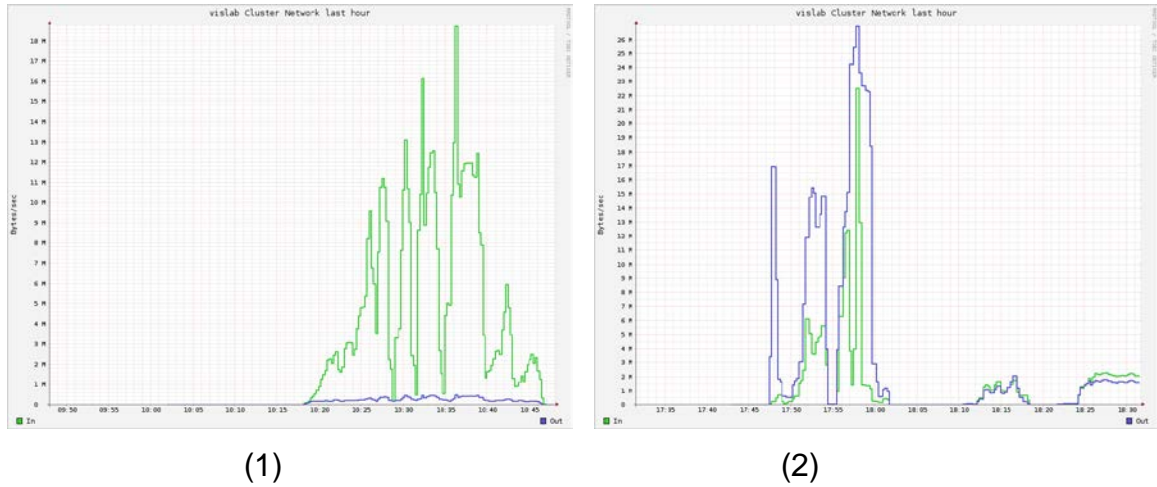
Figura 19: Uso de bytes de memoria RAM del cluster



Fuente: Autores

La figura 21, muestra el comportamiento de los diversos tipos de memoria. En el caso de la memoria swap se ve como se realiza el intercambio de un proceso que se encuentra en memoria a otro que no lo está. Razón por la cual tanto en la imagen 1 como en la 2, los valores que toma este indicador oscilan a medida que se recibe una imagen, y toma valores más grandes en coherencia con el tamaño de la imagen. El uso de memoria nos sirve como registro de que no hay problemas de capacidad con este dispositivo. Es decir que para cargar trabajos inferiores a 1GB esta no se satura.

Figura 21: Bytes/segundos transmitidos por la red.



Fuente: Autores

Los resultados de la figura 20 fueron obtenidos con un Switch 10/100 Mbps. Lo cual genera un limitante muy grande del proyecto. Ya que cuando se lanza una imagen con peso superior a 20 MB en promedio, el proceso de carga de esta en el muro, empieza a tomar tiempos más y más grandes haciendo que la plataforma pierda eficiencia. La prueba de lo antes expuesto, se realizó conectando los nodos directamente a un router de un Gigabit Ethernet. Para el cual la carga de imágenes se realizó de forma más óptima. La siguiente grafica muestra los resultados obtenidos.

8.3 PRUEBAS HARDWARE HETEROGÉNEO

Las pruebas realizadas con tarjetas gráficas de diferentes proveedores fueron exitosas, debido a que el middleware se comportó satisfactoriamente en las pruebas realizadas. Lo cual permite determinar que los requerimientos de hardware que tiene DisplayCluster son independientes o muy flexibles debido a la lógica planteada en el código fuente.

Las pruebas se llevaron a cabo en el muro de visualización del CENTIC el cual cuenta con un nodo cuya tarjeta gráfica es marca AMD. Esta no presentó ninguna incompatibilidad ni en hardware ni en software durante su proceso de instalación. Igualmente durante la ejecución del software presenta igual rendimiento a las tarjetas nVIDIA.

9. LIMITACIONES DEL PROYECTO

Las limitaciones del proyecto pueden dividirse en tres grupos: el primero de ellos es la adquisición de componentes, el segundo es la generalización de aplicaciones ejecutadas en el muro y por último la documentación de la herramienta.

La adquisición de componentes, limita la calidad y utilidad neta que puede ofrecer la plataforma, ya que son estos recursos físicos los que determinan el rendimiento de la misma. Es de vital importancia la selección de monitores y espacio de almacenamiento suficiente para que una plataforma de visualización tenga la capacidad de recibir gran número de usuarios en sus múltiples áreas del saber; pero a su vez económica para la implementación de estas herramientas.

La segunda limitante, es la dificultad de plantear aplicaciones de usuario generales sobre DisplayCluster, ya que al ser un recurso científico, las aplicaciones que pueden hacer uso de una plataforma de visualización son numerosas y cada una de ellas con unas especificaciones y necesidades propias. En base a esto, el personal encargado de la administración de estos recursos debe estar en la capacidad de brindar soluciones casi personalizadas a un usuario.

La tercera y última limitante durante el proyecto, fue la poca documentación que había para la implementación del muro de visualización usando DisplayCluster, aunque existen varios artículos sobre la historia, arquitecturas, avances, etc., sobre muros de visualización, pero no existe una amplia documentación acerca del uso de DisplayCluster y su instalación. Carencia también encontrada en las pruebas realizadas con SAGE.

10. RECOMENDACIONES

Para futuros desarrollos de plataformas de visualización, se recomienda tener suficiente experiencia en la administración y uso del sistema operativo. Igualmente la selección del sistema operativo debe realizarse de forma precavida, ya que la disponibilidad de librerías y paquetes pueden ser un inconveniente a la hora de compilar el middleware.

Para el diseño de la estructura, se recomienda el uso de monitores de alta resolución con el borde lo más delgado posible. También, se sugiere mayor inversión en recursos de almacenamiento sobre inversión en recursos de conexión. En general, desarrollar una plataforma equivalentemente poderosa tanto en hardware como en software, para que no existan limitantes por ninguno de estos factores.

Para la instalación de DisplayCluster se debe tener presente la instalación de las últimas versiones, igualmente revisar los anexos de este libro, que sirven como soporte durante el proceso de instalación y configuración del middleware, muchos de los pasos acá mencionados son obviados en el manual de instalación que se descarga del middleware.

En base a la experiencia adquirida con el libro, no es recomendable la instalación del sistema operativo Debian, ya que al momento de compilar los repositorios, drivers, etc., no se encuentran al día. Se recomienda el uso de Ubuntu, ya que es uno de los sistemas operativos más completos, y el cual actualiza constantemente sus componentes.

11. CONCLUSIONES

Para la interacción entre los componentes hardware y software de la plataforma propuesta, se requiere del desarrollo o implementación de un middleware. Durante el desarrollo de este proyecto se realizó una documentación sobre las diferentes alternativas de middleware existentes; finalmente gracias a sus características como interacción con dispositivos no convencionales, software de código abierto, soporte por el grupo de desarrolladores entre otras, se eligió entre las mejores opciones (SAGE, Chromium, DisplayCluster) a DisplayCluster como el middleware a implementar.

El desarrollo de una plataforma de visualización es un trabajo que requiere tiempo y constante aprendizaje en varios aspectos, la primera de ellas se basa en entender y establecer los recursos hardware, y la segunda en la comprensión, dominio, instalación y compilación de los paquetes necesarios para el middleware; pues al ser un software complejo, las variantes de instalación entre sistemas operativos conllevan diversos conflictos asociados a la distribución de Linux que utiliza, de allí, la importancia de fijar el tipo de aplicaciones que van a hacer uso del muro, para así documentarse y seleccionar el sistema operativo bajo el cual se deba operar.

La flexibilidad de recursos asociada a DisplayCluster permite la construcción de muros de visualización heterogéneos, lo cual permite el uso de diversas fuentes de recurso de hardware. Razón por la cual, se encuentra una alternativa útil, para pequeños grupos de investigación que trabajan con conjuntos de datos medianos. Asociando entonces recursos de rendimiento estándar, que al conformar un clúster constituyen una mejor máquina.

El funcionamiento y beneficios de un muro de visualización en su mayoría de casos quedan atados a las restricciones y en general a la lógica propia del middleware. Aunque actualmente existe gran cantidad de software controlador de este tipo de herramientas, se adaptó DisplayCluster como un software flexible a nivel de recursos de hardware y software. Al ser un software de código abierto, permite a los usuarios adaptar su modo de operar, en base a sus necesidades. Además de esto, es un software que permite la interacción con recursos no convencionales, integrando con esto nuevas tecnologías dirigidas a la ciencia.

Con la realización de este proyecto se logró proponer lineamientos para la construcción o modificación tanto de la plataforma como de aplicaciones que hacen uso de la misma. Estos están orientados a usuarios que usan ésta como un medio para hacer más eficiente la visualización de datos, y aquellos que buscan proponer y/o administrar la misma.

Gracias al desarrollo de este proyecto de investigación se obtuvo una plataforma de visualización para la Unidad de Supercomputación, la cual sirve para la visualización de proyectos científicos y de investigación dentro de la universidad, iniciando con esta labor el GIB, quienes hicieron uso del muro para observar sus aplicaciones biomédicas a gran escala, manifestando su total aceptación de usabilidad.

12. BIBLIOGRAFÍA

- [1] B. Westing et al., "A practical guide to large Tiled displays" [En línea] Consultado [22/01/13]
http://tacc-web.austin.utexas.edu/bwesting/public_html/stallion.pdf
- [2] N. Kukimoto, H. Miyachi. "Method for practical utilization of tiled Display on scientific visualization" [En línea] Consultado [22/01/13]
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5665953>
- [3] Grant Wallace et al., "Tools and Applications for Large-Scale Display Walls," IEEE Computer Graphics and Applications, vol. 25, no. 4, pp. 24-33, July 2005.
- [4] Chao Li et al., "A survey of multi-projector tiled display wall construction" Third International Conference on Image and Graphics, 452-455, 2004
- [5] NVIDIA Quadro. [En línea]. Consultado [22/01/13]
<http://www.nvidia.es/object/workstation-graphics-es.html>
- [6] AMD Eyefinity Technology. [En línea]. Consultado [22/01/13]
<http://www.amd.com/us/products/technologies/amd-eyefinity-technology/Pages/eyefinity.aspx>
- [7] Falko Kuester et al. (2012, Mayo) The Highly Interactive Parallelized Display Space project (HIPerSpace). [En línea] Consultado [22/01/13].
http://vis.ucsd.edu/mediawiki/index.php/Research_Projects:_HIPerSpace
- [8] Byungil Jeong et al., "High-performance dynamic graphics streaming for scalable adaptive graphics environment," in Proceedings of the 2006 ACM/IEEE conference on Supercomputing, New York, NY, USA, 2006, pp. 24-24.
- [9] TACC (2012, Junio) DisplayCluster. [En línea]. Consultado [22/01/13]
<http://www.tacc.utexas.edu/tacc-software/displaycluster>
- [10] Sobell Mark G, Manual práctico de Linux: Comandos, editores y programación Shell. Año:2010
- [11] Naveen K. Krishnaprasad y grupo, "JuxtaView – a Tool for Interactive Visualization of Large Imagery on Scalable Tiled Displays", 2004.

[12] Greg Humphreys et al., "Chromium: a stream-processing framework for interactive rendering on clusters," in Proceedings of the 29th annual conference on Computer graphics and interactive techniques, New York, NY, USA, 2002, pp. 693-702.

[13] S. Lin et al., "Development of Grid-Based tiled display Wall for networked visualization". [En línea] Consultado [22/01/13]

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1543224>

13. ANEXOS

ANEXO A: INSTALACIÓN DE DISPLAY CLUSTER

Manual de comandos utilizados [10]

Sistema operativo: Fedora 17

PAQUETES REQUERIDOS

A. Una vez instalado el sistema operativo, es necesaria la actualización de los repositorios de acuerdo al sistema operativo. A continuación se encuentran el procedimiento de actualización.

```
yumlocalinstall --  
nogpgcheck http://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-stable.noarch.rpm
```

```
yumlocalinstall --  
nogpgcheck http://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-stable.noarch.rpm
```

```
yumcheck-update
```

```
yumupdate
```

```
reboot
```

B. Instalación necesaria de gcc y gcc++ como compilador aplicaciones.

```
Yum install gcc.x86_64
```

```
Yum install gcc-c++.x86_64
```

C. INSTALACION DISPLAYCLUSTER

1. Instalación de OpenMPI

Buscar la última versión de MPI. Para el 10 octubre de 2012 se instaló la versión 1.5.4-5 así:

```
Yum install openmpi.x86_64
```

```
Yum install openmpi-devel.x86_64
```

IMPORTANTE

Usando el editor de preferencia, edite el siguiente archivo **/etc/profile** así:

```
export PATH=$PATH:/usr/lib64/openmpi/bin
```

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib64/openmpi/lib
```

Opcional

Para conocer la ruta de instalación completa de los paquetes *openmpi* en fedora puede consultarlo con el siguiente comando.

```
Yum provides */openmpi
```

2. Instalación de cmake:

```
yum install cmake.x86_64 o yum -y install cmake-*
```

IMPORTANTE

Siguiendo el manual propio del middleware, se presentan problemas para encontrar las librerías de ffmpeg. Para solucionar futuros problemas es necesario editar el archivo de Configuración de forma manual y colocar las rutas adecuadas.

Este archivo se encuentra dentro de los paquetes que se descomprimen al descargar el middleware. La ruta es semejante a la siguiente:

```
>>home/DisplayCluster/cmake/modules/FindLibJpegTurbo.cmake
```

El archivo queda así:

```
# Includedir
```

```
find_path(LibJpegTurbo_INCLUDE_DIR
```

```
    NAMES turbojpeg.h
```

```
    PATHS ${LibJpegTurbo_PKGCONF_INCLUDE_DIRS} /usr/include
```

```
)
```

```
# Finallythelibraryitself
```

```
find_library(LibJpegTurbo_LIBRARY
```

```
NAMES turbojpeg
PATHS ${LibJpegTurbo_PKGCONF_LIBRARY_DIRS} /usr/lib
)
```

3. Instalación Qt

```
yum -y install qt-devel.x86_64
```

Alternativo

```
Yum install qt
```

```
Yum install qt-*
```

4. Instalación Boost

```
yum -y install boost.x86_64
```

```
yum -y install boost-devel.x86_64
```

5. Instalación FFMPEG

```
yum -y install ffmpeg-devel.x86_64
```

```
yum -y install ffmpeg.x86_64
```

Opcional: De encontrarse errores futuros a causa de este paquete proceda con la instalación completa de los paquetes FFMPEG.

```
Yum install ffmpeg-*
```

6. Instalación JPEG

```
yum -y install libjpeg-turbo (libjpeg-turbo-utils.x86_64)
```

```
yum -y install turbojpeg.x86_64
```

```
yum -y install turbojpeg-devel.x86_64
```

D. DisplayCluster incluye la instalación de su herramienta para hacer visualización de manera remota. Esta es llamada **DesktopStreamer**. Para evitar problemas con

la compilación del middleware, es recomendable comentar o suprimir las líneas que contienen la sentencia Win32 del archivo MainWindow.cpp e

Edición del archivo encontrado en:

/home/vizlab/TAAC_DisplayCluster/apps/DesktopStreamer/src/MainWindow.cpp

E. Instalación y Compilación de DisplayCluster

1. Descargar el archivo fuente de la página oficial del grupo TACC:

<https://github.com/TACC/DisplayCluster>

2. Ingrese a la carpeta del middleware y cree un directorio llamado **build** así:

cd -/DisplayCluster

mkdir build

cd build

3. Compile

ccmake ../

Presione la tecla **C** para configurar y siga el menú que se despliega en la parte inferior de la pantalla.

Cambie las siguientes opciones a modo **ON**

Buildoption: BUILD_DESKTOPSTREAMER:ON

Buildoption: BUILD_DISPLAYCLUSTER:ON

Avance, seleccionando C. hasta que termine el proceso de configuración. Una vez en consola, ingrese los siguientes comandos.

make

make install

IMPORTANTE

Para que se reconozca el arreglo de monitores tal cual como queremos que se muestren las imágenes es necesario definir y editar el siguiente archivo:

```
>> /usr/local/configuration.xml
```

A continuación se presenta un ejemplo para un arreglo de 4 monitores, ubicados en línea recta.

```
<configuration>

<dimensionsnumTilesWidth="4" numTilesHeight="1" screenWidth="1280"
screenHeight="1024" mullionWidth="0" mullionHeight="0" fullscreen="1"/>

<process host="vizlab@master" display=":0">

    <screen x="0" y="0" i="0" j="0"/>

    <screen x="1280" y="0" i="1" j="0"/>

</process>

<process host="vizlab@nodo1" display=":0">

    <screen x="0" y="0" i="2" j="0"/>

    <screen x="1280" y="0" i="3" j="0"/>

</process>

</configuration>
```

F. una utilidad adicional de DisplayCluster es el uso de **PythonScriptingSupport** como herramienta para la construcción de aplicaciones que permitan la interacción y dinamismo con el middleware. Para su instalación proceda así:

1. PythonQt 2.0.1.DC (provided in dependencies/ directory)

Requerimientos

- Qt 4.6.1. o versiones superiores.

- Python y su entorno de desarrollo Python Developer.

Instalación de:

```
Yum install qt-devel.x86_64
```

```
Yum install python-devel.x86_64
```

IMPORTANTE

Edite el archivo ***python.prf*** en el directorio *build* de *PythonQt 2.0.1.DC*. Este se encuentra ubicado en *DisplayCluster/dependencias*. Esto con el fin de cambiar la versión del compilador de Python de 2.6 a 2.7, pues es el que se encuentra instalado en el sistema operativo.

```
# Change this variable to your python version (2.5, 2.6)
```

```
win32:PYTHON_VERSION=2.7
```

```
unix:PYTHON_VERSION=2.7
```

2. Compilación de PythonQt:

Ingrese a la carpeta de *DisplayCluster* y una vez allá diríjase a la carpeta *dependencias*. Con una instrucción semejante a:

```
>> /home/vizlab/DisplayCluster/dependencias
```

Y proceda así:

```
Cd PythonQtRoot
```

```
qmake-qt4
```

```
make all
```

NOTA: Más información para la instalación en <http://pythonqt.sourceforge.net/>

3. SIP 4.x

La instalación se realizó de forma manual, para esto es necesario descargar el paquete de la página oficial, y proceder como se muestra a continuación:

```
cd /home/vizlab/Downloads/sip-4.14.1/
```

```
python configure.py
```

```
make
```

```
make install
```

4. PyQt 4.x

La instalación se realizó de forma manual, para esto es necesario descargar el paquete de la página oficial, y proceder como se muestra a continuación:

```
cd PyQt-x11-gpl-4.9.5/
```

```
python configure.py -q /usr/lib64/qt4/bin/qmake
```

```
make
```

```
make install
```

5. Compile del middleware

IMPORTANTE

Antes de realizar la compilación de DisplayCluster, mueva el directorio de PythonQt (encontrado en las dependencias), a una ruta fuera de DisplayCluster. A continuación un ejemplo partiendo de que DisplayCluster se encontraba en:

```
/home/vizlab/DisplayCluster-master/dependencies/PythonQt 2.0.1
```

Nueva ruta:

```
/home/vizlab/PythonQt 2.0.1
```

Edite el archivo **FindPythonQt.cmake** ubicado en: *cmake/modules*. El archivo queda de la siguiente forma.

```
# Includedir
```

```
find_path(PythonQt_INCLUDE_DIR
```

```
    NAMES PythonQt.h
```

```
    PATHS ${PythonQt_PKGCONF_INCLUDE_DIRS}
```

```
    /home/vizlab/PythonQt2.0.1/src
```

```
)
```

```
# Finallythelibraryitself
find_library(PythonQt_LIBRARY
    NAMES PythonQt
    PATHS ${PythonQt_PKGCONF_LIBRARY_DIRS}
    /home/vizlab/PythonQt2.0.1/lib
)
```

Incluya la ruta en el siguiente archivo: */etc/profile*. Así:

```
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib64/openmpi/lib:/home/vi
zlab/PythonQt2.0.1/lib
```

Para compilar el middleware, debe hacerse de cero, es decir, cree de nuevo el directorio build, y adicione en ON la siguiente opción:

```
Buildoption: ENABLE_PYTHON_SUPPORT: ON
```

ANEXO B: SOFTWARE Y REVISIONES ADICIONALES

La configuración de red, es de vital importancia para una comunicación acorde entre los nodos y en general sobre toda la arquitectura.

1. Configuración de red y hosts

A. Edite el archivo **/etc/networks** así:

```
nano /etc/networks
```

```
auto lo
```

```
iface lo inetloopback
```

```
#default 0.0.0.0
```

```
#loopback 127.0.0.0
```

```
#link-local 169.254.0.0
```

```
auto p2p1
```

```
allow-hotplug p2p1
```

```
iface p2p1 inetstatic
```

```
address 192.168.109.183
```

```
netmask 255.255.255.0
```

```
network 192.168.109.0
```

```
broadcast 192.168.109.255
```

```
gateway 192.168.109.1
```

```
# dns-* options are implemented by the resolvconf package, if installed
```

```
dns-nameservers 192.168.19.7
```

Después edite **/etc/init.d/networkrestart** para actualizar el demonio

B. Agregue en el archivo **etc/hosts** los equipos con sus respectivas direcciones IP, de la siguiente manera.

```
127.0.0.1    localhost.localdomain localhost
::1        localhost6.localdomain6 localhost6
192.168.109.183 master.uis.edu.co  master
192.168.109.6  nodo1.uis.edu.co  nodo1
```

C. Establezca el demonio ssh con el siguiente comando:

```
Systemctl start sshd.service
```

Y confirme con comando: `servicesshd status`

D. Deshabilitamos el firewall

```
Applications->Other->Firewall->Disable
```

2. Configuración servidor ssh

A. La instalación de ssh se divide en dos partes: Para el lado del servidor y otro para el cliente. Lo primero es la instalación del paquete ssh. Así:

```
Cliente (Nodos esclavos) ->yum install openssh
```

```
Servidor (Nodo maestro) ->yum install openssh-server.x86_64
```

B. Creación de llaves

Ingrese en cada equipo en modo no administrador. Y genere la llave correspondiente así: (No en root)

```
ssh-keygen (opcional: ssh-keygen -t 1024 -b dsa)
```

NOTA: Más información en:
https://wiki.archlinux.org/index.php/SSH_Keys_%28Espa%C3%B1ol%29

C. Comparta las llaves SSH desde cada PC, para esto copiamos el archivo con la siguiente instrucción.

```
scp .ssh/id_dsa.pub usuario@IP_host_recibe
```

Guardamos la llave pública del otro pc en **authorized_keys** así:

```
cat -/id_dsa.pub >> -/ .ssh/authorized_keys
```

A continuación se muestra un ejemplo de los resultados que se deben generar.

ssh-rsa

```
AAAAB3NzaC1yc2EAAAADAQABAAQACIDgVu3/ZGiE+nTTCGdSEEu4+ybov  
pzwM9FArIXtV6SGB+4A4ALTdAyc2Qshj0DT1rv95VgDCfomsUUWhSacEh3FtvKrl  
VQf/MINE2KHAu02zKmjEDkwq8vWtqFq5K2IZ5JkiqMdTFLXUXwKX9z7+lufChVDr  
Eb081dYKJ7dl+SJBIXyr1Xvt0YOjKdDWZA0sXoqZdmj3DhSRJkkf5z8e5p9cqEjtYJ  
GXfuu0sMxYUOkWmVGabNWOJp8cvQ1j/yo6/IXXdtTPmrJtAD9nEI5V0ttoJSOD  
xjd+3i3P0kwzcZJGiTdqsfnPfdkZZJHk6k40+soNvi/Vsq5oFu3Qgf vizlab @nodo1
```

D. Añada la llave publica del propio equipo en `authorized_keys`.

El archivo debe ir quedando de la siguiente manera:

ssh-rsa

```
AAAAB3NzaC1yc2EAAAADAQABAAQACIDgVu3/ZGiE+nTTCGdSEEu4+ybov  
pzwM9FArIXtV6SGB+4A4ALTdAyc2Qshj0DT1rv95VgDCfomsUUWhSacEh3FtvKrl  
VQf/MINE2KHAu02zKmjEDkwq8vWtqFq5K2IZ5JkiqMdTFLXUXwKX9z7+lufChVDr  
Eb081dYKJ7dl+SJBIXyr1Xvt0YOjKdDWZA0sXoqZdmj3DhSRJkkf5z8e5p9cqEjtYJ  
GXfuu0sMxYUOkWmVGabNWOJp8cvQ1j/yo6/IXXdtTPmrJtAD9nEI5V0ttoJSOD  
xjd+3i3P0kwzcZJGiTdqsfnPfdkZZJHk6k40+soNvi/Vsq5oFu3Qgf vizlab @nodo1
```

ssh-rsa

```
AAAAB3NzaC1yc2EAAAADAQABAAQADQmY9hl6eDi2CdvLjIHf3Linc8hM1h4  
GCbYPHTdjjKd2p1DnMKP8Kx1+HXMZVu7LIb8u1qYEHR/KjfTHSYFP1QiWgkqDf  
GPAuc0tso0GKpG0YoxyC5Yqs9qIQwFqE1HCkBZlnLMZKEmOE1FaPp0AP9WV6  
NGIJ7B+n0YI56q2CjZN2ygXY3V4o9OgY5aK3wSCNVpB9afXuTUPzQGG+sb69k  
SMbS+FAbI+uaMW9hFEUYUS/cR8HAj8P+Wb4KsxwABHyA/L4573XBfxbPnJMxZ  
gjzqU6439LslUJdXP/B+M47Yp7N0awtOw5mCXEtJTUSqJKFVZcm6u1d2Kd+419J  
1KNn vizlab @master
```

E. Modifique los permisos de `authorizedkeys`

```
chmod 600 .ssh/authorized_keys
```

Nota: Con permiso 600 solo se da permiso de escritura a usuarios, limitando grupos y otros.

F. Para que el servidor arranque al iniciar el sistema operativo se debe incluir desde la línea de comandos con los siguientes comandos.

```
chkconfig --listsshd
```

```
chkconfigsshdon
```

```
sshd=1
```

```
reboot
```

3. Instalación Y Configuración Ganglia

A. Instale el servidor apache

```
yum -y install httpd
```

Luego actívalo con la siguiente instrucción:

```
systemctl enable httpd.service
```

Inicie el servicio con

```
Systemctl start httpd.service
```

B. Instalación ganglia: para esto se deben instalar los siguientes demonios

Maestro

```
yum -y install ganglia.x86_64
```

```
yum -y install ganglia-devel.x86_64
```

```
yum -y install ganglia-gmetad.x86_64
```

```
yum -y install ganglia-gmond.x86_64
```

```
yum -y install ganglia-web.x86_64
```

Esclavos

```
yum -y install ganglia-gmond.x86_64
```

4. Configuración servidor NFS

Para la instalación del protocolo NFS se requieren los siguientes paquetes:

```
yum -y install nfs-utils.x86_64
```

Herramienta gráfica opcional para la configuración del protocolo NFS.

```
yum -y install nfs-utils.x86_64
```

Seguido de la instalación de los paquetes, se procede a iniciar el servicio en cada uno de los nodos del clúster

```
Service start nfs-server.service
```

El siguiente comando se utiliza para ejecutar el servicio desde el arranque del sistema operativo:

```
Chkconfig nfs-server on
```

NODO MAESTRO

Se crea el directorio el cual va a ser compartido en el clúster.

```
Sudo mkdir /home/VisLab/Pictures/muro
```

Modifique los permisos del directorio antes creado:

```
chmod 777 /home/VisLab/Pictures/muro/
```

Cree o edite un archivo en la carpeta **/etc/** con el nombre “**exports**”. Este archivo debe contener 3 parámetros:

```
<Directorio><ip><permisos>
```

Directorio: Ruta del directorio que desea ser compartido.

IP: Dirección IP del servidor o equipo que desea compartir el directorio. Seguido de la dirección IP se debe establecer la máscara de la misma.

Permisos: Permisos de lectura, escritura, ejecución.

Un ejemplo del archivo **/etc/exports** se muestra a continuación:

```
/home/VisLab/Pictures/muro  
192.168.19.1/255.255.255.0(rw,sync,nohide,no_root_squash)
```

Para indicar el archivo que se va a exportar, ejecute la siguiente instrucción:

```
/usr/sbin/exportfs -rav
```

Con los siguientes comandos podemos iniciar, parar, reiniciar o conocer el estado del servicio:

```
Sudo systemctl stop nfs-server.service
```

```
Sudo systemctl startnfs-server.service
```

```
Sudo systemctl restartnfs-server.service
```

```
Sudo systemctl status nfs-server.service
```

NODOS ESCLAVOS

Para verificar los directorios compartidos por cualquier nodo usamos el comando:

```
showmount -e <nodo>
```

```
showmount -e frontend
```

Cree el directorio donde se va a montar, una buena opción es la carpeta `/mnt/` de la raíz del sistema.

Se puede montar la unidad de dos formas:

Montar la unidad de forma manual

Ejecute un comando en consola el cual tiene 5 parámetros, este procedimiento es temporal pues una vez se reinicie el sistema se desmonta el servicio.

```
mount
```

```
mount -t nfs4 192.168.19.1: /mnt/
```

Montar la unidad de forma automática

Se debe añadir al archivo `/etc/fstab` la siguiente línea:

```
192.168.19.1:/mnt/      nfs4  rw,hard,intr 0    0
```

Creamos un enlace simbólico, para que las imágenes se guarden en la misma ruta, use la siguiente instrucción:

```
ln -s /mnt/home/VisLab/Pictures/muro/ /home/VisLab/Pictures/
```

ANEXO C: CÓDIGO FUENTE API DESARROLLADA

```
#!/bin/bash
#Create log file
#

date '+%d/%m_%H:%M'> file.log
echo "LOG">>file.log

opc=0          #Auxiliar variable
i=0           #Auxiliar variable
foldernfs=$(cut -d' ' -f1 /etc/exports)      #NFS Folder

#Resolution per monitor

monitorwidth=$(cat /usr/local/configuration.xml | grep screenWidth |
awk '{print $4}' | cut -d'"' -f2)          #Width per monitor
monitorheight=$(cat /usr/local/configuration.xml | grep
screenWidth | awk '{print $5}' | cut -d'"' -f2)  #Height per
monitor

echo "Resolution per monitor(Width x Height):
"$monitorwidth"x"$monitorheight>>file.log

#Number of monitors

wallwidth=$(cat /usr/local/configuration.xml | grep screenWidth |
awk '{print $2}' | cut -d'"' -f2)          #Videowall Width
wallheight=$(cat /usr/local/configuration.xml | grep screenWidth |
awk '{print $3}' | cut -d'"' -f2)          #Videowall height

echo "Total of VideoWall monitors(Width x Height):
"$wallwidth"x"$wallheight>>file.log

#Resolution of video wall

reswall=$(echo " $wallwidth * $monitorwidth " | bc)
#Videowall Width Resolution
reswallh=$(echo " $wallheight * $monitorheight " | bc)
#Videowall Height Resolution

echo "Videowall Resolution (Width x Height):
"$reswall"x"$reswallh>>file.log

declare -a ARRAY_IMG      #ARRAY Image Names
declare -a ARRAY_WIDTH    #ARRAY Width per image
declare -a ARRAY_HEIGHT   #ARRAY Height per image
```

```

while [ $opc -ne 5 ]
do
    clear
    #Options Menu
    echo -e "\n"
    echo -e "\tUNIVERSIDAD INDUSTRIAL DE SANTANDER"
    echo -e "\tTRABAJO DE GRADO"
    echo -e "\tPLATAFORMA DE VISUALIZACION EXTENDIDA Y
AUMENTADA"
    echo -e "\tLAURA CONSTANZA MARTINEZ"
    echo -e "\tDEYBERTH RIAÑO"
    echo -e "\tSUPERCOMPUTACION Y CALCULO CIENTIFICO UIS"
    echo -e "\n"
    echo 1. Listar archivos y directorios #Lista archivos y
directorios almacenados en la carpeta compartida por servidor NFS
    echo 2. Agregar imagen o video #Agrega Imagenes o
videos contenidos en los directorios
    echo 3. Agregar un directorio #Agrega todas las
imagenes o videos del directorio
    echo 4. Ejecutar DisplayCluster #Ejecuta el
middleware, genera el archivo con las imagenes aÑadidas
anteriormente
    echo 5. Salir #Finaliza la
aplicacion
    echo -e "\n"
    echo Seleccione una opcion:
otro="s" #Variable auxiliar

    read opc
    case $opc in

    1)
        ls -x -R -p $foldernfs #Lista carpetas y el
contenido de las mismas
        echo Presione una tecla para continuar....
        read

        ;;

    2)
        folder=$foldernfs #Reasignacion de la variable
        while [ $otro = "s" ]
        do
            let i++; #Variable auxiliar
            echo "Ingrese el nombre de la imagen (junto
con la extension): "
            read ruta
            ARRAY_IMG[i]=$ruta #Se almacena en el vector
los nombres de los archivos
            ARRAY_WIDTH[i]=$(identify -format "%w"

```

```

$folder${ARRAY_IMG[$i])      #Se almacena en el vector el ancho de
la imagen
        ARRAY_HEIGHT[i]=$(identify -format "%h"
$folder${ARRAY_IMG[$i])      #Se almacena en el vector el alto de
la imagen
        echo "¿Desea añadir otra Imagen/Video?
(s/n)"
        read otro
done
;;
3)
echo "Ingrese el nombre del directorio: "
read directory
folder=$foldernfs$directory"/" #Reasignacion de la
variable NFS+directorio a visualizar+/
ls -l $folder>.temp           #Se almacena en un
archivo oculto, por filas, el nombre de los archivos del
directorio

while read file
do
    let i++;
    ARRAY_IMG[i]=$file #Se almacena en el vector los
nombres de los archivos
    ARRAY_WIDTH[i]=$(identify -format "%w"
$folder${ARRAY_IMG[$i])      #Se almacena en el vector el ancho de
la imagen
    ARRAY_HEIGHT[i]=$(identify -format "%h"
$folder${ARRAY_IMG[$i])      #Se almacena en el vector el alto de
la imagen
done < .temp
echo "Se agregaron los siguientes archivos:
"${ARRAY_IMG[*]}             #Se listan los archivos agregados
echo "Presione una tecla para continuar..."
read
;;
4)
#Generar archivo DCX, estado
echo "<!DOCTYPE state>
<state>
<version>1</version>" > format.dcx

for (( j=1; j <= $i; j++))
do
    echo $posyimage
    if [ $j -eq 1 ]
    then

```

```

        posximage=0;
        posyimage=0;
    else
        aux=$(echo "$j-1" | bc)
        posyimage=$(echo
"scale=4;$posyimage+(${ARRAY_HEIGHT[$aux]} / $reswallh)" | bc)
    fi
        widthimage=$(echo
"scale=4;${ARRAY_WIDTH[$j]} / $reswallw" | bc)
        heightimage=$(echo
"scale=4;${ARRAY_HEIGHT[$j]} / $reswallh" | bc)
        echo "<ContentWindow>

<URI> "$folder${ARRAY_IMG[$j]}" </URI>
        <x> "$posximage" </x>
        <y> "$posyimage" </y>
        <w> "$widthimage" </w>
        <h> "$heightimage" </h>
        <centerX>0.5</centerX>
        <centerY>0.5</centerY>
        <zoom>1</zoom>
        <selected>0</selected>
        </ContentWindow>" >> format.dcx

    done

    echo "</state>" >> format.dcx;

    directory_api=$(pwd)
    export VISLAB=${directory_api}/"format.dcx
    echo $VISLAB
    startdisplaycluster
;;
esac
done

```

ANEXO D: ARTÍCULO PRESENTADO EN EL CLCAR 2012

Visualization Laboratory of High Resolution Based on Scalable Architectures

Laboratorio de Visualización de Alta Resolución basado en Arquitecturas Escalables

Deyberth Hernando Riaño
Nuñez⁽ⁱ⁾⁽ⁱⁱ⁾

Laura Constanza Martínez
Mendoza⁽ⁱ⁾⁽ⁱⁱ⁾

Luis Alejandro Torres
Niño⁽ⁱ⁾⁽ⁱⁱ⁾⁽ⁱⁱⁱ⁾

{ deyberth.riano, laura.martinez2, luis.torres }@correo.uis.edu.co

Lola Xiomara Bautista
Roza^{(i)(ii)(v)}

Carlos Jaime Barrios
Hernandez^{(i)(ii)(iv)}

Luis A. Nuñez⁽ⁱ⁾⁽ⁱⁱⁱ⁾

{ lxbautis, cbarrios, lnunez }@uis.edu.co

⁽ⁱ⁾Universidad Industrial de Santander

⁽ⁱⁱ⁾Escuela de Ingeniería de Sistemas e Informática

⁽ⁱⁱⁱ⁾Grupo Halley de Astronomía y Ciencias Aeroespaciales

^(iv)Unidad de Supercomputación y Cálculo Científico

^(v)Grupo de Investigación en Ingeniería Biomédica

Abstract

The development of some science areas and their impact create new challenges to solve by engineering. Research in weather, astronomy, molecular chemistry and other domains generates thousand of data in small time intervals, which requires high performance computing to treat and extensive visualizations.

Although, Scalable Architectures ensure expected performance, sophistication in hardware and software makes expensive the platforms, as is the case of visualization walls. This work proposes the use of hardware standard and specific open source software to develop screen walls coupled with scalable platforms. The proposed platform not only ensures high resolution, massive data processing and high performance, but also remote visualization and cheapest throughput.

Resumen

El desarrollo de diferentes ramas de la ciencia y su impacto crean nuevos retos y necesidades para ser resueltos por Ingeniería. Es el caso de la investigación, en clima, astronomía, química molecular entre otras, que generan cientos de datos en intervalos de tiempo

cortos, exigiendo alto desempeño y visualización extendida.

A pesar que las arquitecturas escalables asegurar el rendimiento esperado, la sofisticación del hardware y software hacen muy costosas en términos económicos esas plataformas, como es el caso de los muros de visualización. Este trabajo, ataca el problema proponiendo el uso de hardware estándar y software abierto para desarrollar muros de visualización y acoplarlos con plataformas escalables. La plataforma propuesta, no solo asegura alta resolución, procesamiento masivo de datos y alto rendimiento, sino también, visualización remota y gestión eficiente de demanda a bajo costo.

1. Introducción

Los científicos a través de los avances tecnológicos han logrado acercarse más a la realidad, como consecuencia de esto se generó un aumento del volumen de datos por almacenar y procesar. Un ejemplo es el conjunto de imágenes de alta resolución del planeta Tierra [1] que son albergadas por la NASA (National Aeronautics and Space Administration), las

cuales requieren herramientas no convencionales para visualizarse en mayor detalle.

Actualmente se construyen muros de visualización utilizando monitores LCD (Liquid Crystal Display) que proporcionan cientos de megapíxeles [2], ofreciendo costos de mantenimiento relativamente bajos pero que requieren un software de control que puede llegar a ser complicado de configurar e implementar.

En este artículo se describen las experiencias de construcción de muros de visualización utilizando monitores LCD y software de código abierto en la Universidad Industrial de Santander (Colombia).

2. Muros de Visualización

Un muro de visualización es una solución escalable de alta resolución de pantallas, ubicadas en forma de mosaico. Cuando se habla de alta resolución implica que es mucho mayor a la encontrada en los monitores estándares, la cual es típicamente de uno a cuatro megapíxeles. Cuando se refiere a mosaico quiere decir que el muro de visualización contiene múltiples monitores organizados en forma de cuadrícula y finalmente cuando se implica que es una arquitectura escalable, a medida que las necesidades de resolución aumentan pueden ser añadidos nuevos monitores a la cuadrícula.

Las principales características de un muro de visualización son:

- i. Su tamaño permite que múltiples usuarios lo usen simultáneamente.
- ii. Su alta resolución permite gran cantidad de información de alta calidad.
- iii. La combinación de tamaño y resolución permite a los usuarios obtener información general del objeto estudiado y detallar el mismo a medida que se está más cerca.



Fig. 1. 307 Mpixel Tiled-Display System (Stallion) The University of Texas (Texas Advance Computing Center)

2.1 Hardware para Muros de Visualización

Un muro de visualización se construye típicamente con componentes ordinarios para una PC [3][4]. Existen dos modalidades muy utilizadas en su construcción: pantallas LCD o LED (Light-Emitting Diode) y proyectores, cada una de las cuales muestra ciertas ventajas y desventajas. Cuando se utilizan pantallas se tiene un inconveniente con respecto a los bordes que se presenta entre ellas y por lo tanto es de suma importancia la utilización de monitores con bordes delgados para evitar la pérdida de ilusión de manejo de una imagen unificada, pero a pesar de esto, la configuración del arreglo es sencilla y se cuenta con buenas características según el tipo de monitor escogido. La utilización de proyectores en la construcción de muros de visualización no es muy popular en la comunidad debido a la dificultad que presenta al alinearlos de forma correcta, evitando que las imágenes se solapen, produzcan sombras y se pierda calidad en la imagen, además, uno de los mayores inconvenientes es la calibración de cada uno de ellos [5] y el tamaño de la estructura, pero su gran ventaja sobre los monitores es la ausencia de bordes y que al cambiar la distancia de proyección puede incrementarse/disminuirse el tamaño del muro de visualización. En general, la infraestructura usada en el desarrollo de un laboratorio de visualización depende de los requerimientos y preferencias de los usuarios.

Un muro de visualización contiene múltiples monitores y para cada uno de ellos es necesario una señal de video. Las tarjetas gráficas estándares actuales proporcionan dos o más salidas de video por lo cual, pueden ser configurados dos o más monitores por estación de trabajo, tal como es el caso de la tarjeta NVIDIA GeForce GTX 670 la cual cuenta con conectividad Dual Link DVI-D [6]. Existen otro tipo de tarjetas introducidas recientemente al mercado que permiten controlar más de dos monitores como las tarjetas de video NVIDIA QUADRO [7] y la AMD Eyefinity [8]. Además dos tipos de configuraciones que dependen del número de monitores a desplegar en el muro de visualización:

- i. Una computadora que soporta una cantidad de tarjetas gráficas y que controlará todo el muro de visualización.
- ii. Múltiples computadoras donde cada una de ellas contiene una tarjeta gráfica que alimenta a cada uno de los monitores.

En los muros de visualización controlados por una sola computadora se combina el hecho de que ciertas

tarjetas gráficas pueden manejar múltiples monitores y existe la posibilidad de añadir cierta cantidad de tarjetas a una máquina. El resultado contiene múltiples tarjetas gráficas conectadas a una sola tarjeta madre. Esto simplifica en gran medida la configuración debido a que muchos sistemas operativos la reconocen como un recurso unificado y además, es la solución más económica para implementar y mantener. Para pequeñas y medianas configuraciones esto es más que suficiente, pero para laboratorios de visualización de ultra-alta resolución no lo es [9].

Cuando una sola computadora no es una opción para manejar un muro de visualización, se debe recurrir a múltiples computadoras interconectadas por medio de tecnologías conmutadas Ethernet de 1 gigabit, pues se debe garantizar un ancho de banda suficiente para distribuir los datos correspondientes a la imagen eficientemente. Cada computadora maneja un número determinado de monitores los cuales se sincronizan usando intercomunicación. Estas configuraciones además de presentar alta escalabilidad en el sistema de visualización, presentan una potencia de procesamiento mucho mayor que un equipo único.

Una configuración recomendada por el TACC, para un muro de Visualización conformado por 6 Monitores (12,1 Megapíxeles) es la siguiente:

| | |
|----------------------|---|
| Equipo (1 Unidad) | Dell Precision T3600, Six Core Xeon, 32GB de Memoria, 635W Fuente de Poder, 2TB Disco Duro |
| Monitor (6 Unidades) | Samsung 460UX – 3 |
| Tarjeta de Video | Nvidia GeForce GTX 690 4GB GDDR5 PCI Express 3.0 x16 (x2) ó MSI R7970 Lightning Radeon HD 7970 3GB 384-bit GDDR5 PCI Express 3.0 x16 HDCP Ready CrossFireX Support (x1) |

3. Software para Muros de Visualización

A pesar de la sencillez del montaje de un muro de visualización con un solo equipo de cómputo, el problema a tratar se enfoca en sistemas distribuidos y paralelos en muros de ultra-alta resolución.

Uno de los retos en los sistemas distribuidos y paralelos es la compatibilidad de aplicaciones con el software existente. Muchos de los cambios para ejecutar aplicaciones secuenciales se deben hacer directamente sobre el código fuente de las mismas y algunas de sus bibliotecas y, con ello presentan el

inconveniente de poder o no ser usadas en el muro de visualización.

A continuación se presenta una breve descripción de diferentes tipos de software utilizados para manejar un muro de visualización.

3.1. Chromium

Chromium [10] (basado en WireGL [11]) es un sistema de renderización interactivo diseñado para clúster y utiliza unidades de procesamiento de transmisión (SPUs) para serializar y de manera opcional modificar una secuencia de comandos OpenGL. Una SPU toma una secuencia como entrada y produce cero o más secuencias de salida. La comunicación entre SPU's ubicados en diferentes computadoras es manejada por una abstracción de red basada en conexión punto a punto. Chromium puede usarse para redirigir el flujo gráfico de salida de aplicaciones OpenGL al muro de visualización. La biblioteca OpenGL usada por las aplicaciones es reemplazada por la biblioteca OpenGL modificada de Chromium. Esta biblioteca intercepta todos los comandos OpenGL y los reenvía al SPU local. Un mosaico ordenado es enviado por la SPU a los servidores Chromium corriendo en el muro de visualización. Las SPU locales de cada parte del arreglo reciben su secuencia de comandos correspondientes y estas son enviadas a la tarjeta gráfica.

3.2. Scalable Adaptive Graphics Environment (SAGE)

SAGE [12] es un sistema de renderización remota basado en píxeles para muros de visualización. En SAGE, la renderización se separa de la visualización usando un clúster dedicado para entregar píxeles al conjunto de nodos de visualización. SAGE incorpora un gestor de ventanas que permiten que el flujo de píxeles sea superpuesto, movido o redimensionado en el muro de visualización. La biblioteca de interface de aplicación de SAGE (SAIL) es usada en el lado de la renderización para capturar la salida de la renderización de la aplicación y secuenciar la salida a un conjunto de receptores SAGE (un receptor por nodo de visualización). Un gestor de FreeSpace comunicado con SAIL, los receptores SAGE y los clientes UI para controlar la composición final de la secuencia en el muro de visualización. Como inconveniente en la utilización de SAGE se requiere enlaces de red de alta velocidad por que se transmiten flujos de píxeles y no se utiliza todo el potencial de renderización de las

tarjetas gráficas ni las capacidades de cómputo de las CPUs ubicados en los nodos de visualización.

3.3 DisplayCluster

DisplayCluster [13] es un entorno software diseñado por el TACC (Texas Advanced Computing Center) de la Universidad de Texas, para manejar muros de visualización de gran escala. DisplayCluster permite administrar el contenido del muro de visualización de manera remota desde equipos como laptops y permite interactividad mediante dispositivos como Joysticks, iPhone, iPad, iTouch o algún dispositivo Android. Existe una nueva posibilidad de interactuar con el muro de visualización mediante el uso de Microsoft Kinect pero que aún se encuentra en versión experimental.

DisplayCluster es montado sobre un ambiente MPI. Una vez configurado el clúster MPI la instalación puede ser llevada a cabo de manera sencilla en comparación con los ambientes para muros de visualización presentados anteriormente. Adicionalmente, se cuenta con una interfaz de secuencia de comando en Python que permite la interacción con DisplayCluster.

4. Muro de Visualización en Desarrollo

La Universidad Industrial de Santander (Colombia) actualmente, construye tres muros de visualización para diferentes laboratorios de ciencia. El primero de ellos se desarrolla para el Grupo Halley de Astronomía y Ciencias Aeroespaciales, el cual se implementa sobre un pequeño clúster construido en base a 5 Dell WorkStation T3500 con tarjetas QUADRO NVS 420, [14] con salida para 4 monitores cada una y con capacidad de soportar un muro de visualización de 16 monitores. Los monitores utilizados son DELL P2412H de 24" cada uno con resolución máxima de 1920x1080 a 60Hz, organizados en una matriz de 4x4 obteniendo una resolución total de 30720x17280 (530 Megapíxeles). El WorkStation restante es quien realiza las labores de Equipo Maestro. El sistema operativo del clúster es Debian GNU/Linux 6.0 y el software que sirve de frontend al muro de visualización es DisplayCluster.

Los restantes dos muros de visualización se encuentran en desarrollo y se está evaluando el hardware y el software necesario para su implementación.

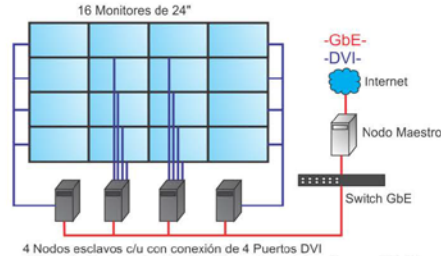


Fig. 2. Laboratorio de Visualización Grupo Halley- Universidad Industrial de Santander

DisplayCluster fue elegido más por su sencillez de instalación y fácil uso que por otras razones. SAGE es una excelente alternativa pero, es mucho más compleja su configuración.

El laboratorio de visualización del Grupo Halley se encuentra vinculado a un telescopio de Meade LX400 de 20" de gran capacidad y a un CORONADO SolarMax, los cuales entrarán en funcionamiento realizando estudios de dinámica solar y las imágenes serán enviadas al muro de visualización para su estudio y análisis. El clúster implementado proporciona capacidad de cómputo de alto rendimiento para modelado, análisis y procesamiento de los datos adquiridos con la instrumentación del observatorio, debido a que cuenta con procesadores Intel Xeon Quad Core y con 16 procesadores CUDA por tarjeta gráfica.

6. Conclusiones y Trabajos Futuros

La elaboración de un Laboratorio de Visualización no requiere de componentes especiales, se puede realizar con Hardware estándar el cual es muy asequible. Además se cuenta con variedad de Middleware el cual posibilita visualizar la imagen eficientemente. Para esto requerimos de alta velocidad de transferencia de datos, la cual se facilita con la tecnología Gigabit Ethernet.

Una de las aplicaciones del Grupo de Investigación de Informática Biomédica (GIIB) de la Universidad Industrial de Santander, será ejecutada para Detectar Glóbulos rojos en imágenes microscópicas de células sanguíneas.

7. Agradecimientos

Este proyecto es apoyado por la Universidad Industrial de Santander (UIS), y realizado por la Unidad de Supercomputación y Cálculo Científico SC3-UIS y el Grupo Halley de Astronomía y Ciencias Aeroespaciales, igualmente de la UIS. Agradecemos a la Universidad de Texas Austin, El Centro de Computo Avanzado de la Universidad de Texas (TACC-Texas Advance Computing Center), especialmente a Karla Vega y su grupo de trabajo por su constante apoyo.

Este trabajo es igualmente realizado con el apoyo de NVIDIA Corporation dentro del programa de NVIDIA Teaching Center, agradeciendo especialmente a Michael Lasen por su colaboración.

8. Referencias

- [1] NASA. (2012, Junio) EARTH OBSERVATORY <http://earthobservatory.nasa.gov>
- [2] DeFanti, T.A., Leigh, J., Renambot, L., Jeong, B., Verlo, A., Long, L., Brown, M., Sandin, D.J., Vishwanath, V., Liu, Q., Katz, M.J., Papadopoulos, P., Keefe, J.P., Hidley, G.R., Dawe, G.L., Kaufman, I., Glogowski, B., Doerr, K.U., Singh, R., Girado, J., Schulze, J.P., Kuester, F., Smarr, L.: The OptiPortal, a Scalable Visualization, Storage, and Computing Interface Device for the OptiPuter. *Future Generation Computer Systems* 25 (2009) 114-123
- [3] K. Li et al., "Building and Using A Scalable Display Wall System," *IEEE Computer Graphics and Applications*, vol. 20, no. 4, pp. 29-37, July/August 2000.
- [4] Grant Wallace et al., "Tools and Applications for Large-Scale Display Walls," *IEEE Computer Graphics and Applications*, vol. 25, no. 4, pp. 24-33, July 2005.
- [5] Chao Li et al., "A survey of multi-projector tiled display wall construction" *Third International Conference on Image and Graphics*, 452-455, 2004
- [6] NVIDIA. (2012, Junio) NVIDIA GeForce GTX 670. [Online]. <http://www.nvidia.es/object/geforce-gtx-670-es.html>
- [7] NVIDIA. (2012, Junio) NVIDIA Quadro. [Online]. <http://www.nvidia.es/object/workstation-graphics-es.html>
- [8] AMD. (2012, Junio) AMD Eyefinity Technology. [Online]. <http://www.amd.com/us/products/technologies/amd-eyefinity-technology/Pages/eyefinity.aspx>
- [9] Falko Kuester et al. (2012, Mayo) The Highly Interactive Parallelized Display Space project (HIPerSpace). [Online]. http://vis.ucsd.edu/mediawiki/index.php/Research_Projects:_HIPerSpace
- [10] Greg Humphreys et al., "Chromium: a stream-processing framework for interactive rendering on clusters," in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2002, pp. 693-702.
- [11] Greg Humphreys et al., "WireGL: a scalable graphics system for clusters," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2001, pp. 129-140.
- [12] Byungil Jeong et al., "High-performance dynamic graphics streaming for scalable adaptive graphics environment," in *Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, New York, NY, USA, 2006, pp. 24-24.
- [13] TACC (2012, Junio) DisplayCluster. [Online]. <http://www.tacc.utexas.edu/tacc-software/displaycluster>
- [14] NVIDIA. (2012, Junio) NVIDIA Quadro NVS 420. [Online]. http://www.nvidia.es/object/product_quadro_nvs_420_es.html

ANEXO E: DISTINCIÓN OTORGADA POR EL COMITÉ ORGANIZADOR DEL CLCAR 2012 COMO POSTER GANADOR DEL EVENTO



La Universidad Tecnológica de Panamá y la Conferencia Latinoamericana de Computación de Alto Rendimiento 2012, otorga el presente certificado a:

LUIS A. TORRES - CARLOS J. BARRIOS H.- LUIS A. NUÑEZ - LOLA X. BAUTISTA R. - DEYBERTH H. RIAÑO -LAURA C. MARTINEZ M.

Por haber presentado el poster ganador: **"Visualization Laboratory of High Resolution Based on Scalable Architectures "**

Conferencia Latinoamericana de Computación de Alto Rendimiento 2012

Realizada en el Campus Universitario Dr. Victor Levi Sasso de la Universidad Tecnológica de Panamá, en la Ciudad de Panamá, del 27 al 31 de agosto de 2012.

Dr. Carlos J. Barrios
Presidente del Comité Internacional CLCAR

Dr. Martín Candanedo
Presidente del Comité Local CLCAR

Dr. Armando A. Jipsion P.
Coordinador General CLCAR 2012