

COMPUTACIÓN DE ALTO RENDIMIENTO PARA LA VISUALIZACIÓN E
INTERACCIÓN REMOTA DE APLICACIONES CIENTÍFICAS

DEYBERTH HERNANDO RIAÑO NUÑEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
MAESTRÍA EN INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2017

COMPUTACIÓN DE ALTO RENDIMIENTO PARA LA VISUALIZACIÓN E
INTERACCIÓN REMOTA DE APLICACIONES CIENTÍFICAS

DEYBERTH HERNANDO RIAÑO NUÑEZ

Trabajo de grado para optar al título de
Magister en Ingeniería de Sistemas

Director

CARLOS JAIME BARRIOS HERNÁNDEZ PhD.
Universidad Industrial de Santander

Codirector

JOSÉ TIBERIO HERNÁNDEZ PEÑALOZA PhD.
Universidad de los Andes

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
MAESTRÍA EN INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2017

TABLA DE CONTENIDO

INTRODUCCIÓN	13
1 JUSTIFICACIÓN	15
2 OBJETIVOS	17
2.1 OBJETIVO GENERAL	17
2.2 OBJETIVOS ESPECÍFICOS	17
3 MARCO TEÓRICO	18
3.1 Computación remota	18
3.2 Visualización remota	18
3.3 Virtualización	19
3.3.1 Virtualización de GPU	20
3.3.2 Virtualización de GPUs para cálculo científico	21
3.3.3 Virtualización de GPUs para procesamiento gráfico	21
3.4 Procesamiento en paralelo	24
3.5 Muro de visualización	25
3.6 Analítica Visual	27
3.7 Imágenes de Tractografías	27
4 METODOLOGÍA	29
5 ARQUITECTURA COMPUTACIONAL	32
5.1 Componentes	32
5.1.1 Usuarios y almacenamiento	32
5.1.2 Procesamiento	33
5.1.3 Visualización e interacción	34
5.2 Integración de los componentes de la arquitectura	38
6 LINEAMIENTOS	41
6.1 Lineamientos para la diseño e implementación de la arquitectura compu- tacional	41
7 PRUEBAS DE RENDIMIENTO	46
7.1 Hardware	46
7.2 Software	48
7.3 Casos de estudio	48

7.3.1	Caso UNO: Tractografías	49
7.3.2	Caso DOS: Simulación de fluidos	54
8	CONCLUSIONES	60
	REFERENCIAS BIBLIOGRÁFICAS	61
	BIBLIOGRAFÍA	64
	ANEXOS	66

TABLA DE TABLAS

3.1	Comparación de formas de virtualización de GPUs	24
3.2	Taxonomía de Flynn	25
5.1	Infraestructura GUANE	34
5.2	Especificaciones nodo de procesamiento gráfico	35
5.3	Especificaciones de NVIDIA GRID K2	35
5.4	Especificaciones de NVIDIA GRID K2 TM	36
5.5	Especificaciones nodo de procesamiento gráfico	36
5.6	Especificaciones nodo vCenter Server	37
5.7	Especificaciones nodo View Connection Server	37
5.8	Especificaciones nodo View Security Server	37
5.9	Especificaciones nodo Microsoft Active Directory	38
7.1	Infraestructura GUANE	47
7.2	Especificaciones configuración Máquinas Virtuales	47
7.3	Especificaciones Cliente Remoto	48
7.4	Ejemplo grupo de datos iniciales	50
7.5	Resumen de datos simulación de fluidos	55
7.6	Parámetros para el renderizado del modelo	56
7.7	Tiempos promedios de ejecución utilizando RayCast	57
A.1	Ejemplo grupo de datos iniciales	67

TABLA DE FIGURAS

3.1	Esquema computación remota	18
3.2	Esquema visualización remota	18
3.3	Esquema Virtualización basada en Hipervisor	19
3.4	Esquema Virtualización basada en Host	20
3.5	Arquitectura de rCuda	21
3.6	Esquema vDGA	22
3.7	Esquema vGPU	23
3.8	Esquema vSGA	23
3.9	Ley de Amdahl	25
3.10	Arquitectura muro de visualización	26
3.11	Esquema visual analytics	27
3.12	Imágenes de Tractografías	28
4.1	Metodología de trabajo	31
5.1	Elementos principales de la arquitectura	32
5.2	Servicios LDAP y NFS implementados en la arquitectura	33
5.3	Arquitectura para virtualización de escritorios con vGPU	38
5.4	Arquitectura Global	39
5.5	Arquitectura YAJE	40
7.1	Escenarios para medición de pruebas	46
7.2	Imagen de tractografía del cerebro	50
7.3	Región de interés seleccionada para la búsqueda de tractos	51
7.4	Tractos en región de interés con método probabilístico - Camino	51
7.5	Tractos en región de interés con método probabilístico - DIPY	52
7.6	Clustering de tractos en el cuerpo calloso - DIPY	53
7.7	Ancho de banda utilizado en interacción con Clustering de tractos	54
7.8	Volumen en un periodo de tiempo método Splatting	55
7.9	Rendering del modelo con método RayCast con 4000 muestras	56
7.10	Tiempos Raycast Rendering	57
7.11	Promedio uso de CPU Raycast Rendering	58
7.12	Promedio uso de memoria Raycast Rendering	58
7.13	Ancho de banda utilizado en prueba de video	59
A.1	Anisotropía fraccional sin vectores de orientación	68
A.2	Anisotropía fraccional con vectores de orientación	69
A.3	Imagen de tractografías del cerebro	70

A.4	Región de interés seleccionada para la búsqueda de tractos	71
A.5	Tractos encontrados en la región de interés usando método probabilístico	72

LISTA DE ANEXOS

Anexo A: Generación de imágenes de tractografía con Camino	67
Anexo B: Poster presentado al GPU Technology Conference 2016,	73
San Jose, California	

RESUMEN

Título: COMPUTACIÓN DE ALTO RENDIMIENTO PARA LA VISUALIZACIÓN E INTERACCIÓN REMOTA DE APLICACIONES CIENTÍFICAS ¹

Autor: Deyberth Hernando Riaño Nuñez ²

Palabras clave: Visualización, HPC, Servicio, Interacción, Supercomputación, vGPU

DESCRIPCIÓN

Con los avances tecnológicos y con ello la capacidad para captura de información con cientos de variables temporales que pretenden dar respuesta a fenómenos naturales, del comportamiento humano, de ciencias básicas, sociales, económicas, entre otras; la comunidad científica está demandando capacidad de cómputo y visualización de grandes volúmenes de datos de manera ágil, rápida y amigable con el fin de compilar algoritmos complejos que ayuden a la toma de decisiones y enriquezcan la comunidad. Es por lo anterior que se da la oportunidad de creación de un nuevo esquema o arquitectura con dichas características a la cual se le da el nombre de YAJE (Arquitectura ofrecida como servicio para la computación y visualización de alto rendimiento), la cual favorece la investigación poniendo a disposición de los usuarios la posibilidad de acceso remoto mediante escritorios virtuales con tarjeta gráfica dedicada y acceso directo a la plataforma de supercómputo, en un mismo entorno de manera que favorece así la interacción Cliente/Aplicación, disminuye el tiempo de búsqueda de soluciones óptimas para realizar procesamiento y visualización de resultados como un flujo de trabajo único, reduce costos asociados a la adquisición de hardware de altas prestaciones y favorece la colaboración científica.

¹Trabajo de Grado con modalidad de Trabajo de Investigación.

²Trabajo de Grado con modalidad de Trabajo de Investigación. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Carlos Jaime Barrios Hernández. Co-Director: José Tiberio Hernández Peñaloza.

ABSTRACT

Title: HIGH PERFORMANCE COMPUTING FOR REMOTE VISUALIZATION AND INTERACTION WITH SCIENTIFIC APPLICATIONS ³

Autor: Deyberth Hernando Riaño Nuñez ⁴

Keywords: Visualization, HPC, Service, Interaction, Supercomputing, vGPU

DESCRIPTION

With the technological advances and the ability for capture information with hundreds of temporal variables that aim to respond to natural phenomena, human behavior, basic sciences, socials, economics, among other; the scientific community demands computation and visualization of large datasets in an agile, fast and friendly way in order to compile complex algorithms that help the decision making and the community

Because of the above there is an opportunity to create a new scheme or architecture with these characteristics called YAJE (Architecture offered as a service for high performance computing and visualization), supporting the research, available for users using remote access with virtual desktops with dedicated graphic card and direct access to the supercomputing platform, in the same environment improving the interaction Client/Application, decreasing the search time for optimal solutions to run processing and visualization of results in a single workflow, reduces costs associated with the acquisition of high performance computing hardware and enhances the scientific collaboration.

³Master's degree research work.

⁴Department of Physical-mechanical Engineering. School of Systems Engineering and Computer Science. Advisor: Carlos Jaime Barrios Hernández, Ph.D. in Computer Science. Co-advisor: José Tiberio Hernández Peñaloza

INTRODUCCIÓN

La computación a gran escala y por ende la visualización de resultados es una actividad indispensable y una necesidad en el campo de la investigación. Los expertos en las áreas de la ciencia e investigadores demandan constantemente la realización de simulaciones o computación de modelos apoyados de plataformas de cómputo, los cuales generalmente requieren el apoyo del proceso de visualización de resultados producto del cómputo realizado.

Por su parte, la gran cantidad de datos y el tiempo dedicado al procesamiento hace que las plataformas de cómputo de alto rendimiento sean una herramienta necesaria para los investigadores, sin embargo hoy por hoy cuando se desarrolla una aplicación en computación de alto rendimiento la visualización de salidas es comúnmente considerada como un paso independiente, situación que limita al usuario e implica la búsqueda de herramientas paralelas que apoyen su actividad de visualización.

La visualización es el resultado de un largo proceso de investigación el cual genera una extensa cantidad de datos, por lo cual, el tamaño de estos datos y el procesamiento que se realiza sobre estos, hacen del cómputo de alto rendimiento una necesidad para obtener sus resultados.

Si bien, hoy día el uso computación científica hace parte ineludible de la investigación, es aún evidente que se encuentran dificultades de acceso a la misma dada la falta de disponibilidad de infraestructuras de HPC ⁵ en cada centro de investigación, bien por la demanda de uso, la falta de equipos y en ocasiones el costo de acceso a dichos servicios.

En Colombia para poner en contexto, la Universidad Industrial de Santander y puntualmente su Centro de Investigación CAGE(Computo Avanzado y a Gran Escala) provee Cómputo de Alto Rendimiento como servicio para la comunidad usando GUANE [1], un supercomputador híbrido basado en tarjetas GPU ⁶ NVIDIA TESLA TM [2], siendo éste uno de los más utilizados por la comunidad científica no solo en el oriente Colombiano, sino también con presencia en proyectos internacionales.

Debido al requisito de realizar visualización, nace la necesidad de crear e implementar una arquitectura computacional, que integre cómputo de alto rendimiento y visualización al mismo tiempo, estos ofrecidos como servicio de forma ágil a todos los usuarios.

El presente trabajo consiste en el desarrollo de la arquitectura,YAJE ⁷(Arquitectura ofrecida como servicio para la Computación y Visualización de Alto Rendimiento), que permite integrar tanto los requerimientos de hardware para visualización remota como los de HPC.

⁵High Performance Computing / Computación de Alto Rendimiento

⁶Graphics processing unit / Unidad de procesamiento gráfico

⁷<https://yaj.sc3.uis.edu.co/>

La arquitectura propuesta tiene como finalidad ser una alternativa de solución en el desarrollo de aplicaciones que requieren HPC y visualización usando escritorios virtuales como herramienta para observar los resultados provenientes del cálculo científico. Estos escritorios, contienen GPU dedicada para propósitos gráficos, la cual genera una característica muy importante para garantizar gráficos con mejor calidad.

La arquitectura adicional al cómputo de datos propone la inclusión de gráficos como servicio (Graphics as a service GaaS) la cual es una nueva tendencia en las ciencias de la computación y juegos en la nube, que le da a este proyecto un enfoque que remueve o disminuye a una mínima expresión la necesidad de adquirir costosas tarjetas gráficas para realizar tareas de renderización.

Adicionalmente se propone a la comunidad científica un flujo de trabajo el cual pretende eliminar la necesidad de adquirir hardware de alto costo para realizar visualización y cómputo de alto rendimiento, situación que resulta no solo viable sino interesante desde el punto de vista de disminución de costos asociados a la investigación por eliminación de presupuestos para adquisición de equipos para realización de procesamiento temporales, eliminando así la compra de hardware que tendrá a futuro una depreciación y obsolescencia asociada a tiempo de uso.

Dando un valor agregado para el usuario, el hecho de que esta propuesta tenga un enfoque orientado como servicio permitirá a los usuarios finales el acceso a ambientes en escritorios remotos configurados con altas prestaciones de hardware por un bajo costo y bajo los pilares de usabilidad y facilidad de usuario; dado que se garantiza la disminución del retardo que se genera en la interacción Usuario/Aplicación lo cual normalmente afecta la experiencia del usuario en la interacción con la aplicación disminuyendo la usabilidad y por ende la calidad del servicio.

Finalmente se presenta el resultado de esta investigación que consta de una arquitectura de cómputo la cual integra supercomputación como una herramienta para acelerar la solución de un algoritmo, visualización remota con el fin de eliminar la necesidad a los investigadores de adquirir hardware de gama alta para visualizar sus resultados e interacción remota donde los usuarios finales que administran o usan la aplicaciones.

1.JUSTIFICACIÓN

Con los avances tecnológicos y la búsqueda de respuestas o patrones de comportamiento a los fenómenos naturales, el comportamiento humano y de las diferentes áreas del conocimiento se han generado a su vez no solo cientos de preguntas por resolver sino grandes volúmenes de datos que requieren ser procesados mediante algoritmos complejos que ameritan no solo potentes equipos para su tratamiento sino disponibilidad de tiempo de compilación y mecanismos que presenten los datos de manera gráfica, amigable y atractiva para la comunidad científica.

Con la capacidad de almacenamiento y el poder de captura de información a gran escala que se ha dado en la era digital, se han logrado contar con series históricas de datos que demandan ser estudiadas y procesadas en aplicaciones científicas las cuales requieren un procesamiento de altas prestaciones para garantizar realismo en la visualización, apoyando la toma de decisiones o análisis de fenómenos específicos.

Por otra parte esta visualización, igualmente requiere una interacción que permita a los observadores o comunidad científica manipular los fenómenos observados o modificar los parámetros de la simulación de acuerdo a tiempos reales específicos.

Actualmente en Instituciones como el INRIA (Francia), los actores y los recursos tanto de procesamiento como de visualización e interacción, se encuentran remotos (por ejemplo FlowVR [3] aplicación de interacción remota). Este tipo de aplicaciones permiten garantizar colaboración y uso eficiente de recursos de cómputo, los cuales apoyan a los expertos en tareas de observación de información, comparación y estudio de problemáticas de la sociedad.

Con base en las necesidades previamente expuestas surgió la propuesta que da origen a esta tesis en la cual se pretende no solo ofrecer los servicios tanto de visualización como de cómputo a gran escala, sino ofrecerlos de manera segura, ágil, amigable y en un mismo entorno o plataforma disminuyendo el tiempo que el usuario desgasta en la comprensión, instalación o configuración de escenarios computacionales, tiempo que resulta más útil y valioso para análisis de resultados que enriquezcan los descubrimientos de la comunidad científica.

La solución desarrollada en este trabajo de investigación consta de una arquitectura orientada al servicio tanto para visualización como para computación de alto rendimiento a la cual se dio el nombre YAJE.

YAJE ayudará a los grupos de Investigación pertenecientes a la Universidad Industrial de Santander y en general a la comunidad científica asociada a la Universidad (bien a nivel nacional e internacional debida a la posibilidad de acceso remoto), gracias a que el esquema propuesto puede ser adoptado para uso de gran cantidad de algoritmos o aplicaciones científicas que requieran de computación de alto rendimiento, visualización

de información e interacción con la información presentada haciendo de la misma un esquema amigable y viable para la comunidad científica en general.

2.OBJETIVOS

2.1.OBJETIVO GENERAL

Diseñar un esquema de procesamiento, visualización de datos y colaboración en aplicaciones científicas.

2.2.OBJETIVOS ESPECÍFICOS

- Establecer una arquitectura que permita la colaboración científica y toma de decisiones mediante aplicaciones, utilizando visualización en gran formato y técnicas de interacción.
- Definir lineamientos para implementar una arquitectura híbrida de hardware/software para el procesamiento, visualización y colaboración en aplicaciones científicas.
- Desarrollar una aplicación que permita validar los lineamientos planteados en el contexto de imágenes diagnosticas de cerebro, utilizando técnicas de clasificación en una cohorte de sujetos.

3.MARCO TEÓRICO

3.1.Computación remota

Computación remota se relaciona al acceso a un servidor mediante una terminal en la cual se realizan procesos y únicamente se observa la salida del equipo remoto. Se obtiene acceso a recursos como el procesador, memoria y almacenamiento en disco. En este tipo de conexión no es posible acceder a la tarjeta gráfica del servidor, esto quiere decir que cualquier tipo de visualización de los datos transmitidos debe hacerse localmente. Para realizar una simulación en tiempo real utilizando computación remota, se requiere transferir los datos para realizar el rendering localmente mediante una red de altas prestaciones.

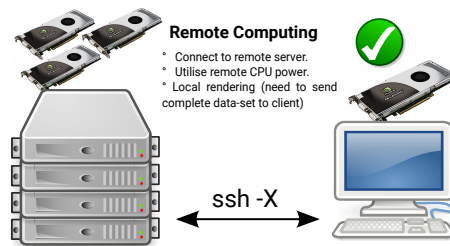


Figura 3.1: Esquema computación remota. Fuente: Autor

3.2.Visualización remota

Mediante la visualización remota los datos se mantienen en el servidor remoto, donde se realiza el renderizado de las imágenes. La visualización de datos se realiza localmente usando la tarjeta gráfica del servidor.

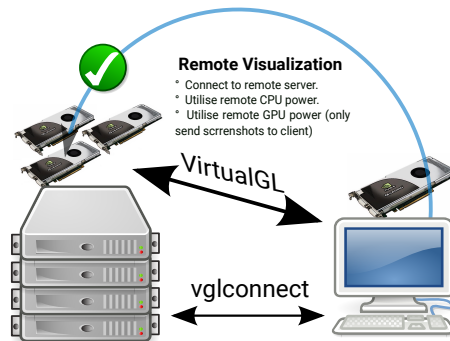


Figura 3.2: Esquema visualización remota. Fuente: Autor

3.3.Virtualización

La virtualización permite simular varios computadores lógicos, usando un único computador. Esta simulación se realiza dividiendo los recursos entre los computadores lógicas que se desean crear.

Generalmente los recursos a dividir son procesador, memoria y disco duro. Para esto existen dos maneras principales de realizar virtualización, virtualización basada en hipervisor y virtualización basada en host.

- Virtualización basada en hipervisor: este tipo de virtualización no requiere un sistema operativo base. Los recursos se dividen mediante un hipervisor el cual es el responsable de su administración. En la figura 3.3 se pueden observar las capas utilizadas en esta virtualizacion, iniciando con la capa hardware la cual es administrada por hipervisores comerciales como VMWARE[4] o gratuitos como XenServer[5] y KVM[6]. Finalmente la ultima capa contiene las computadoras lógicas simuladas (Linux, Windows, MAC OS).

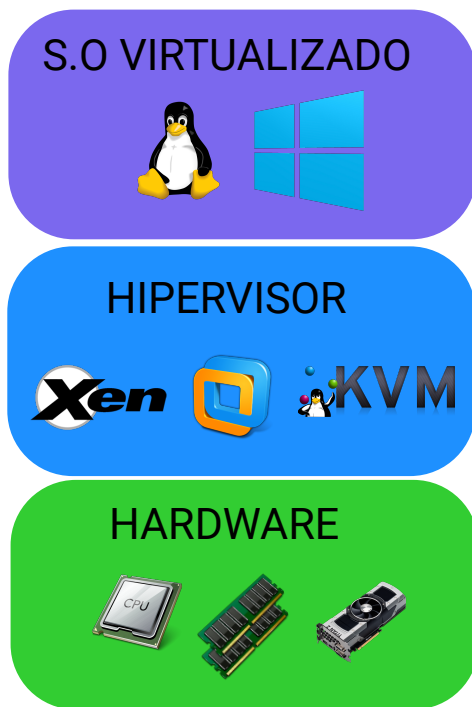


Figura 3.3: Esquema Virtualización basada en Hipervisor. Fuente: Autor

- Virtualización basada en host: En este tipo de virtualización se requiere un sistema operativo base (Windows, Linux , MAC OS) en el que se instala un software para la creación de computadores lógicos o máquinas virtuales. En el mercado existen soluciones comerciales y privadas para realizar virtualización por host, como VMWARE, Qemu[7], KVM, XenServer y VirtualBox entre otras.

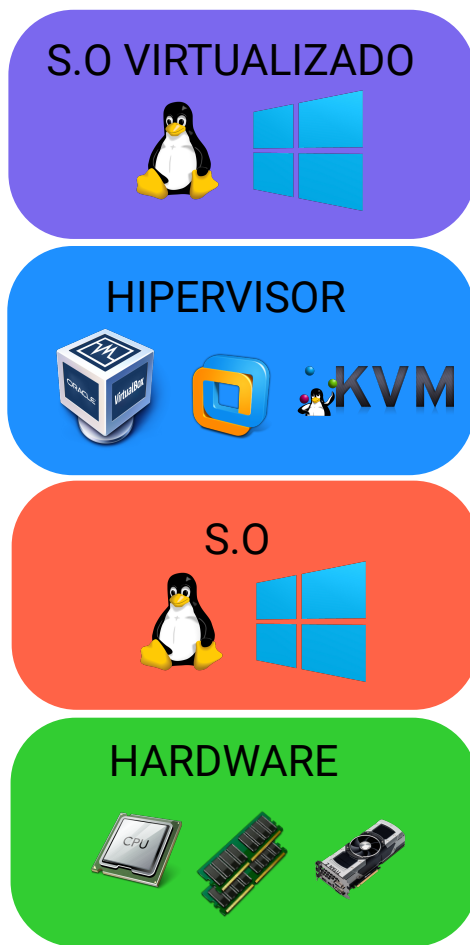


Figura 3.4: Esquema Virtualización basada en Host. Fuente: Autor

Otro concepto importante tiene que ver con la forma como el hipervisor comparte los recursos. Los recursos pueden ser divididos como el procesador, la memoria RAM y disco duro, o pueden ser utilizados totalmente por la máquina virtual. Para utilizar los recursos directamente y exclusivamente por una única máquina virtual se utiliza una tecnología llamada I/O Passthrough (Input/Output Passthrough), ésta tecnología permite que el dispositivo sea utilizado directamente en la máquina virtual, como si se estuviese físicamente conectado a ella.

3.3.1. Virtualización de GPU

Un concepto reciente respecto a la virtualización es la virtualización de GPUs. En las unidades de procesamiento gráfico se pueden diferenciar dos tipos, las que se usan para realizar tareas de cálculo científico y las que se utilizan para realizar visualización o procesamiento gráfico. En el primer tipo, se pueden encontrar tarjetas con miles de núcleos que son aprovechados para realizar cálculo científico en paradigmas de computación paralela, algunas de las tarjetas más conocidas son fabricadas por NVIDIA, como Tesla

con arquitecturas Fermi , Kepler y Maxwell. En el segundo tipo se utiliza la GPU para realizar procesamiento gráfico, ocupando la mayor parte del mercado dos fabricantes AMD y NVIDIA.

En los dos tipos de unidades de procesamiento gráfico es posible realizar virtualización del dispositivo, permitiendo que múltiples usuarios accedan a una sola GPU, dejando a un lado la subutilización de los recursos y generando que más usuarios trabajen simultáneamente.

3.3.2.Virtualización de GPUs para cálculo científico

Para realizar virtualización de GPU para realizar cálculo científico se pueden utilizar manejadores de recursos como SLURM[8] o implementaciones como rCuda[9].

- SLURM¹ es un manejador de trabajos que tiene la habilidad de compartir recursos heterogeneos para ejecutar trabajos en un cluster para su ejecución. SLURM puede compartir recursos no genericos como GPUs [8].
- rCUDA: Adicionalmente al uso de vGPU para realizar tareas de visualización, se pueden crear perfiles de GPU para tareas de cálculo científico. rCuda permite a los usuarios acceder a perfiles de GPU remotamente, donde el usuario final puede usar tarjetas de cálculo científico como NVIDIA TeslaTM desde su ordenador personal sin necesidad de acceder a un supercomputador mediante conexión remota. Esta conexión es realizada por el administrador de la plataforma, siendo transparente para el usuario final.[9]

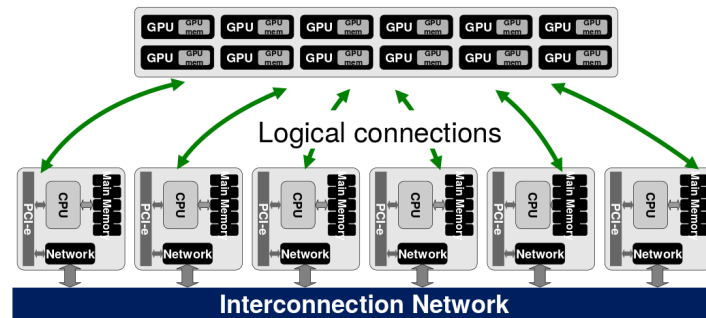


Figura 3.5: Arquitectura de rCuda. Fuente: rCuda

3.3.3.Virtualización de GPUs para procesamiento gráfico

La virtualización de GPU para procesamiento gráfico es un nuevo concepto, permite que la tarjeta gráfica sea utilizada por múltiples máquinas virtuales, incrementando la cantidad de usuarios que pueden usar un único dispositivo. Para esto se puede utilizar virtualización basada en hipervisor, en la que el hipervisor administra la tarjeta gráfica

¹<https://slurm.schedmd.com/>

dependiendo de la configuración prevista (por ejemplo Passthrough). Hipervisores como KVM [6], VMware[4] o XenServer [5] permiten el uso compartido de estos dispositivos.

Utilizando este enfoque se pueden ofrecer a los usuarios gráficos de alto desempeño, los cuales son utilizados remotamente sin la necesidad de adquirirlos localmente.

Consecuente con lo anterior, existen diferentes maneras en las cuales se pueden dividir las tarjetas gráficas, esto depende de las necesidades del usuario. A continuación se mencionan algunos perfiles o configuraciones utilizadas [10].

- **vDGA Virtual Dedicated Graphics Acceleration:** Es una tecnología empleada para realizar virtualización de GPUs, proporciona a los usuarios acceso sin restricciones y completamente dedicado a una única GPU. vDGA ofrece un nivel mas alto de rendimiento para los usuarios que requieren computación gráfica de mayor nivel. Adicional a lo anterior permite el uso de aplicaciones que se ejecutan con OpenGL 4.4, Microsoft DirectX 9, 10, or 11, and NVIDIA CUDA 5.0. En la figura 3.6 se presenta el esquema vDGA.

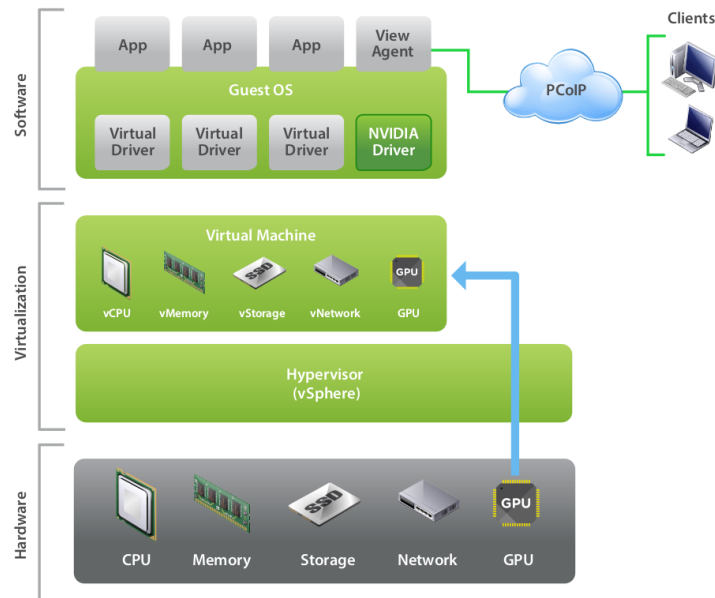


Figura 3.6: Esquema vDGA. Fuente: VMware

- **vGPU Virtual Graphics Processing Unit:** NVIDIA ofrece una tecnología para realizar virtualización por GPU que crea GPUs virtuales (vGPU) con ciertas características de hardware iguales o inferiores a la tarjeta gráfica que está siendo compartida, estas vGPU permiten ser accedidas remotamente, utilizando un esquema como el que se presenta en la figura 3.7.

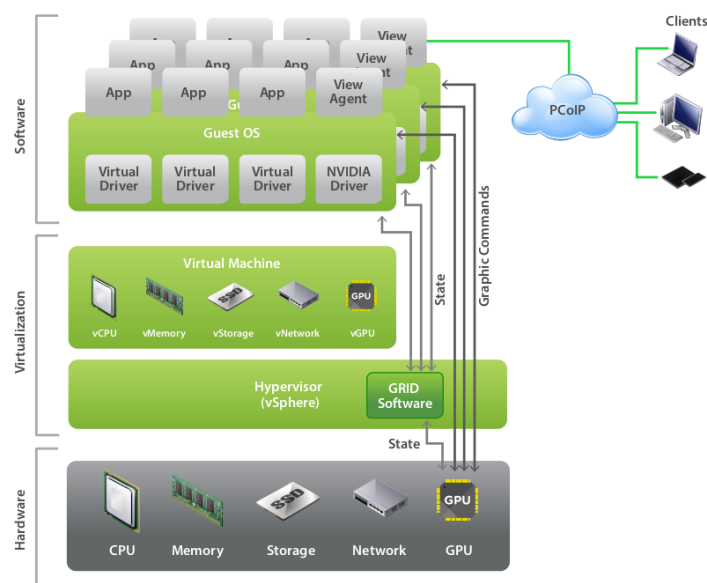


Figura 3.7: Esquema vGPU. Fuente: VMware

Existen implementaciones en la que se puede dividir un único dispositivo PCI express entre ocho máquinas virtuales como la NVIDIA GRID K2. Estos perfiles son creados y administrados mediante virtualización basada en hipervisor.

- **vSGA Virtual Shared Graphics Acceleration:** Esta tecnología permite a la GPU ser compartida por múltiples escritorios virtuales. Es una solución atractiva para usuarios que requieren todo el potencial de la capacidad de la GPU durante cortos periodos. En todo caso, vSGA puede generar cuellos de botella dependiendo de cuales aplicaciones se están usando y los recursos necesarios de la GPU. A continuación se presenta el esquema vSGA en la figura 3.8.

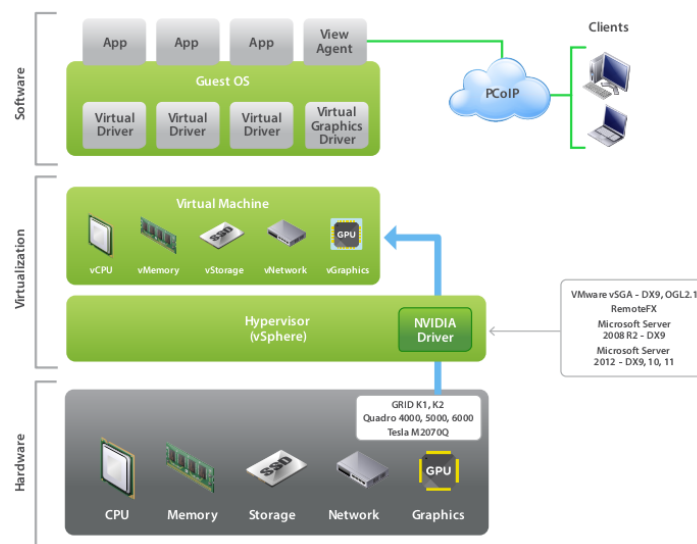


Figura 3.8: Esquema vSGA Fuente: VMware

A continuación se muestra una tabla consolidada de comparaciones entre las tres tecnologías expuestas para virtualización de GPUs.

vDGA	vGPU	vSGA
GPU Dedicada para un usuario	GPUs compartidas entre usuarios, aunque pueden ser dedicadas	GPUs compartidas entre usuarios
Ratio de consolidación 1:1 (1 usuario por GPU física)	Buen ratio de consolidación (8 usuarios por GPU física)	Alto ratio de consolidación (Limitación para memoria de vídeo en tarjetas gráficas)
Rendimiento a nivel de estación de trabajo	Rendimiento a nivel de Estación de trabajo de nivel básico	Buen desempeño para aplicaciones livianas sin certificación de drivers.
Soporta aplicaciones con CUDA u OpenCL	No soporta aplicaciones con CUDA ni OpenCL	No soporta aplicaciones con CUDA ni OpenCL

Tabla 3.1: Comparación de formas de virtualización de GPUs Fuente: Autor

3.4. Procesamiento en paralelo

El procesamiento en paralelo hace referencia a la habilidad de ejecutar múltiples tareas simultáneamente, utilizado para la resolución de problemas computacionales a gran escala. Existen diferentes clases de sistemas para procesamiento en paralelo: Computación multi-core, distribuida, clúster, procesamiento masivo en paralelo, Grid Computing, GPU, entre otras. [11]

Existen dos términos fundamentales en procesamiento en paralelo: La Taxonomía de Flynn y la Ley de Amdahl.

- La taxonomía de Flynn es una clasificación de arquitecturas de computadores propuesta por Michael Flynn, contiene cuatro clasificaciones basadas en el número de instrucciones y los flujos de datos disponibles en la arquitectura. [12]

	Una Instrucción	Múltiples Instrucciones
Un dato	SISD: computador secuencial, no existe paralelismo de datos ni instrucciones.	MISD: flujo de instrucciones ejecutadas sobre un solo dato.
Múltiples datos	SIMD: un solo flujo de instrucciones ejecutadas sobre un flujos de datos.	MIMD: flujo de instrucciones ejecutadas sobre varios flujos de datos. Ej.: Arreglos híbridos

Tabla 3.2: Taxonomía de Flynn Fuente: Autor

- La Ley de Amdahl es un modelo matemático que describe la relación entre la aceleración esperada de la implementación paralela de un algoritmo y la implementación serial del mismo algoritmo. [13] La aceleración de un algoritmo en el cual se usan múltiples procesadores se ve afectada por la cantidad de código secuencial, es decir, que no se pueda paralelizar. Ver la figura 3.9.

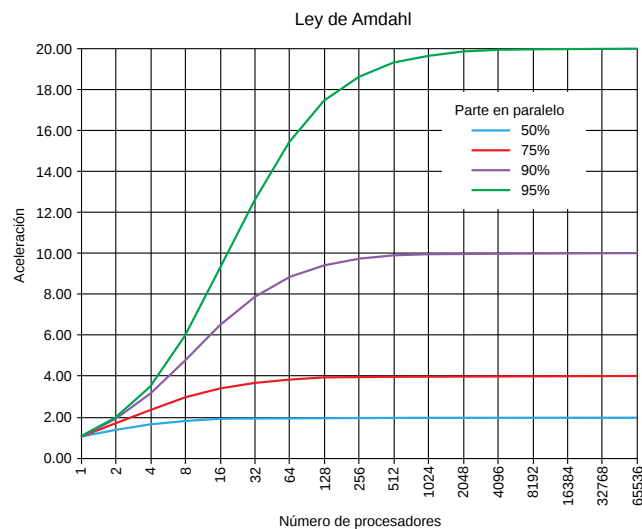


Figura 3.9: Ley de Amdahl Fuente: Wikipedia

3.5. Muro de visualización

Un muro de visualización (videowall) es una solución escalable de alta resolución de pantallas o proyectores ubicados en forma de mosaico. Cuando se habla de alta resolución implica que es mucho mayor a la encontrada en pantallas convencionales, la cual es normalmente de uno a cuatro megapíxeles. Al hablar de mosaico se refiere a que el muro de visualización contiene múltiples pantallas en forma de cuadrícula [14]. Algunas características de un muro de visualización son [15]:

- Su gran tamaño permite que múltiples usuarios lo usen simultáneamente.

- Al poseer alta resolución permite visualizar imágenes de alta calidad.
- Permite visualizar (dependiendo de su tamaño) gran cantidad de imágenes simultáneamente.
- La combinación de tamaño y resolución, permite a los usuarios obtener información general del objeto estudiado y realizar comparaciones visuales con otros objetos.

La construcción de muros de visualización se puede realizar por los siguientes métodos [15]

- Estación de trabajo individual: consta de una o dos pantallas de alta resolución operadas por un ordenador, alternativa de bajo costo útil para trabajo individual.
- Gran pantalla sencilla: se usan proyectores para obtener resoluciones superiores a 300“, aumenta el costo respecto a la estación de trabajo individual. Generalmente se ve afectada la resolución de las imágenes sacrificada por la dimensión de las mismas.
- Sistema de múltiples proyectores: Es el sistema de mayor complejidad y costo, ofrece alta resolución sin fisuras o biseles. Su complejidad está asociada a la alineación de los proyectores.
- Mosaico de pantallas: se crea mediante arreglos de pantallas LCD/LED con bisel delgado. Ofrece alta resolución a bajo costo.

Los métodos mencionados anteriormente pueden ser configurados mediante una estación de trabajo individual o un clúster de computadores.

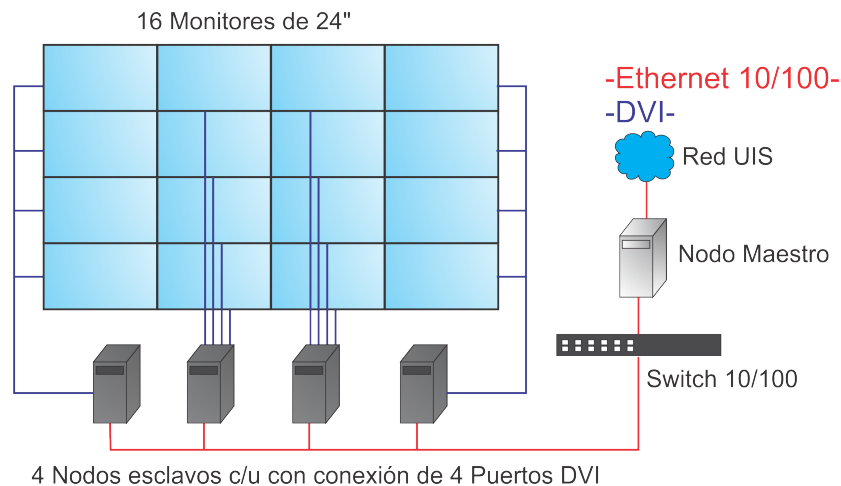


Figura 3.10: Arquitectura muro de visualización Fuente: Autor

3.6. Analítica Visual

La Analítica visual es la ciencia que estudia el uso de interfaces visuales interactivas para el apoyo del análisis de datos complejos. Utiliza un conjunto de algoritmos y métodos de estudio de datos con la finalidad de facilitar la toma de decisiones. Los objetivos de la analítica visual son [16]

- Sintetizar información y derivar conocimiento de datos masivos, dinámicos, ambiguos y algunas veces conflictivos.
- Detectar lo esperado y predecir lo inesperado.
- Proporcionar oportuna, defendible y entendiblemente valoraciones.
- Comunicar efectivamente las valoraciones para tomar las acciones correspondientes.

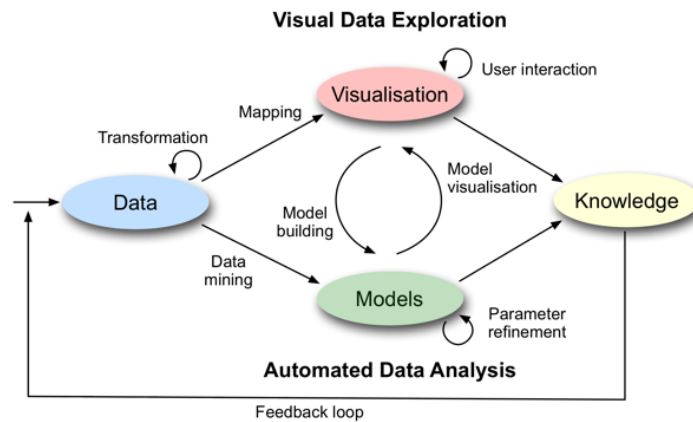


Figura 3.11: Esquema visual analytics

Analítica visual es importante en áreas del conocimiento donde gran cantidad de información debe ser procesada y analizada, como por ejemplo la física, astrofísica, meteorología, seguridad, geografía, biología, medicina, entre otras. El proceso de analítica visual combina métodos de análisis visual y automático acoplado, a la interacción humana con el fin de obtener conocimiento de los datos iniciales. La siguiente figura muestra los estados (representados por óvalos) y sus transiciones (flechas) en el proceso de analítica visual. Se caracteriza a través de interacciones entre los datos, visualizaciones, modelos sobre los datos y usuarios para descubrir conocimiento.

3.7. Imágenes de Tractografías

En neurociencias una tractografía, es un método para representar los tractos neuronales que son las conexiones anatómicas en el cerebro humano. Este método utiliza los

datos generados por una imagen de tensor de difusión (DTI)², la cual se obtiene de un procesamiento realizado a imágenes de resonancia magnética (MRI)³.

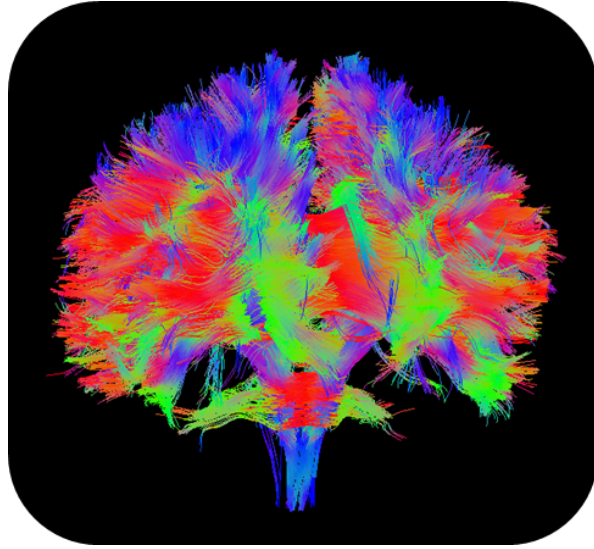


Figura 3.12: Imágenes de Tractografías Fuente: Grupo IMAGINE

Clustering (o agrupamiento) es una técnica de aprendizaje de máquina (no supervisado), que permite crear automáticamente grupos a partir de atributos en un conjunto de datos. Las fibras hacen referencia a un conjunto de polilíneas que se forman a partir de los puntos contenidos en los archivos de partida. Son fibras nerviosas compuestas de diferentes tipos de células y componen las estructuras cerebrales.

En el siguiente capítulo se presenta la metodología de trabajo utilizada, iniciando con un proceso de Revisión Bibliográfica, seguido del Diseño e Implementación de la arquitectura, por ultimo se desarrollaron dos casos de estudio para realizar pruebas sobre la arquitectura propuesta.

²Diffusion Tensor Imaging

³Magnetic Resonance Imaging

4.METODOLOGÍA

El desarrollo del siguiente trabajo de investigación se llevó a cabo haciendo uso de una metodología propia, de carácter híbrido, que combinó el desarrollo del método científico y las bondades de las metodologías ágiles para el desarrollo de proyectos del área de la informática por sus beneficios al observar las salidas en cada uno de los incrementos de cada iteración.

La metodología constó del desarrollo de cuatro etapas dentro de las cuales cada una siguió un proceso de análisis, observación, desarrollo y evaluación para dar paso al siguiente paso o en su defecto realizar correcciones o replanteamiento de soluciones que permitieran el refinamiento del trabajo.

- Revisión bibliográfica: Se realizó la búsqueda, estudio y documentación bibliográfica relacionada a los temas de investigación previamente descritos en el marco conceptual u otros temas relevantes para poner en funcionamiento la arquitectura de forma ágil, asequible y amigable para el usuario. Para ello se accedió a artículos en revistas indexadas.

Durante la revisión bibliográfica no se encontró como tal una herramienta o plataforma que permita integrar ambas actividades utilizando maquinas virtuales con vGPU, situación que dificulta y obliga a los usuarios expertos de la comunidad científica a trabajar de manera aislada o paralela en diferentes entornos ambas actividades las cuales están directamente relacionadas y deberían ser una consecuencia de la otra sin necesidad de cambiar de plataforma, servicio o herramienta computacional. Lo cual creó una oportunidad en este trabajo de investigación. En esta revisión se encontraron soluciones que requieren instalaciones de entornos de desarrollo para una sola aplicación, limitando la cantidad de usuarios que pueden usar el clúster bajo esta implementación [17]. Respecto a vGPU se encontraron soluciones independientes pero no utilizando computación de alto desempeño [18].

- Diseño de la arquitectura: Se realizó el diseño y la diagramación de la arquitectura YAJE, arquitectura ofrecida como servicio para la Computación y Visualización de Alto Rendimiento, que permite integrar tanto los recursos de hardware y software requeridos para el cómputo de alto rendimiento y la visualización e interacción remota.

La tesis finalmente logró integrar los tres pilares fundamentales que fueron el desafío del desarrollo de este trabajo "Lograr la integración en una única arquitectura: Computación de alto rendimiento, Visualización, e Interacción".

En concordancia con la metodología se logró durante las iteraciones el replanteamiento de soluciones y la evolución del proyecto de investigación logrando de esta manera diferenciar que la arquitectura resultante logra condensar los tres pilares

o ejes principales de la tesis:

1. Usuarios y almacenamiento: Enfocados en el usuario se otorga la facilidad para el acceso, transferencia de información y almacenamiento de datos.
2. Procesamiento: Capacidad de cómputo para realizar tareas de cálculo científico, tales como procesamiento de datos, imágenes, simulaciones y actividades con fines académicos o científicos que requieran de equipos de alta capacidad o HPC para minimizar tiempos de respuesta en la obtención de resultados.
3. Interacción y Visualización: Enfocado en el usuario, la plataforma permite la interacción y visualización utilizando escritorios virtuales, con tarjeta gráfica dedicada y visualización de sus resultados remotamente. Las máquinas virtuales tienen acceso a la plataforma de HPC.

En el capítulo de Arquitectura se presentaran a fondo los tres tópicos propuestos.

- Implementación de la arquitectura planteada: Se llevó a cabo la implementación de la arquitectura diseñada. Durante este paso se generaron incrementos que permitieron replantear soluciones añadiendo o suprimiendo elementos previamente establecidos en la arquitectura. Se consolidaron las especificaciones planteadas respecto a hardware y software requeridos para el completo funcionamiento de cada uno de los componentes.
- Desarrollo de Casos de estudio: Consistió en el desarrollo de dos casos de estudio uno de ellos propuesto en el los objetivos del proyecto sobre la realización de tratamiento de imágenes médicas, y posterior visualización de sus resultados. Adicional se incluyó un segundo caso de estudio sobre simulación de fluidos en el cual se realizaron tareas de rendering de modelos 3D las cuales requerían cómputo de alto rendimiento para acelerar la ejecución del algoritmo.

Con ambos casos de estudio se plantearon escenarios de pruebas que permitieran aportar a los resultados y lineamientos de la arquitectura realizada. En el primer caso se realizó un algoritmo para el tratamiento de imágenes médicas el cual se ejecutaba en GUANE y del cual se visualizaban resultados en las máquinas virtuales, lo anterior con fines de establecer los lineamientos del desarrollo de aplicaciones bajo esta arquitectura. En el segundo caso se utilizó un software llamado VIsit (Visualize It) el cual es un software libre para visualización interactiva y análisis de gráficos con implementación paralela de datos científicos; para ello se utilizaron datos de fluidos otorgados por la Universidad Técnica de Kaiserslautern con el ánimo de probar la implementación paralela de un algoritmo usando la arquitectura YAJE mediante el servicio otorgado por GUANE.

Se elaboró el análisis de los resultados obtenidos estableciendo métricas de rendimiento de la aplicación y la creación de los lineamientos para el uso de la arquitectura.

La figura 4.1 representa la metodología planteada, como se puede observar en la figura se puede presentar una iteración entre el diseño e implementación de la misma, debido

a requerimientos que pueden variar durante el desarrollo.

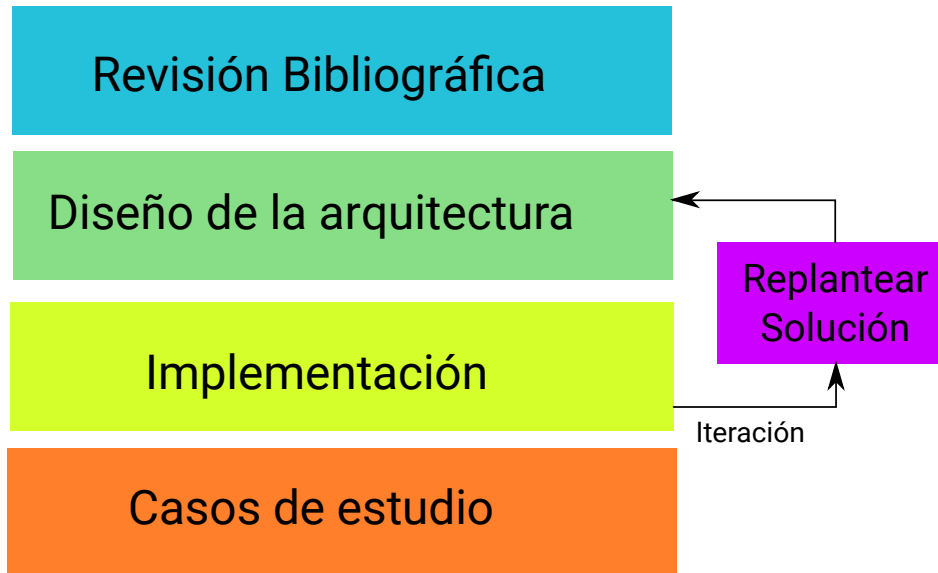


Figura 4.1: Metodología de trabajo Fuente: Autor

A partir de la metodología mencionada anteriormente se inicia con el desarrollo del proyecto, en el siguiente capítulo se presenta la arquitectura desarrollada resaltando tres elementos fundamentales: Administración (Usuarios y Almacenamiento), Procesamiento y Visualización (Visualización e Interacción).

5.ARQUITECTURA COMPUTACIONAL

En el siguiente capítulo se presenta la arquitectura propuesta. En el desarrollo de este capítulo se observan los tres elementos fundamentales juntos a sus especificaciones de hardware y software.

El diseño de la arquitectura se dio de manera iterativa obteniendo incrementos que permitieron finalmente consolidar la arquitectura y las especificaciones presentadas en este capítulo.

Durante todo el desarrollo de la arquitectura, la finalidad y eje principal fue proveer a los usuarios expertos o comunidad científica en general, de los recursos de hardware y software (algunos ya disponibles para HPC), entregando una arquitectura y sus lineamientos para llevar a cabo la tarea de computación de alto rendimiento y procesamiento de imágenes de manera tal que favorezca la interacción y colaboración científica, y a su vez que resulte amigable y fácil de usar.

La arquitectura diseñada provee a los usuarios finales escritorios virtuales con tarjeta gráfica dedicada y acceso directo a la plataforma de supercómputo.

Para el diseño e implementación de la arquitectura se encontraron tres elementos principales (Figura 5.1).

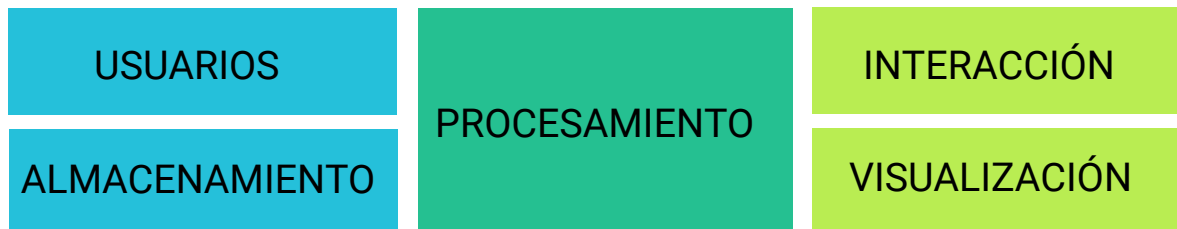


Figura 5.1: Elementos principales de la arquitectura Fuente: Autor

5.1.Componentes

5.1.1.Usuarios y almacenamiento

El primer elemento permite que los usuarios interactúen de manera transparente, sin realizar autenticaciones o transferencia de datos recurrentemente entre los nodos de cálculo y las máquinas virtuales. Se utilizan protocolos que mantienen estados homogéneos tanto en los nodos de visualización como los de cómputo. Este elemento está compuesto por dos servicios principales que son un sistema de archivos en red y un protocolo de acceso a directorios para el almacenamiento de datos. Se encuentran en

los nodos de administración de GUANE [1] y son accedidos por cada uno de los nodos de cálculo y las máquinas virtuales para realizar visualización.

NFS Network File System ¹: Es un protocolo que permite compartir un directorio localizado en una computadora en red con computadores o dispositivos de la misma red; donde el equipo donde se localiza el directorio es el servidor y los computadores/dispositivos conectados son los clientes. Los clientes normalmente se unen a un directorio compartido el cual está sincronizado en cada uno de los nodos. Este protocolo es adecuado para crear NAS² en entorno Linux/Unix.

LDAP Lightweight Directory Access Protocol ³: Es un protocolo de red para acceso a directorios ligeros que permite acceder a los servicios de un directorio y unir bases de datos de usuarios Unix. Se utilizó para realizar la autenticación de usuarios en cada uno de los nodos de GUANE y las máquinas virtuales.

Con la implementación de estos servicios se asegura la consistencia de datos y usuarios. La figura 5.2 muestra la disposición de elementos en la arquitectura.

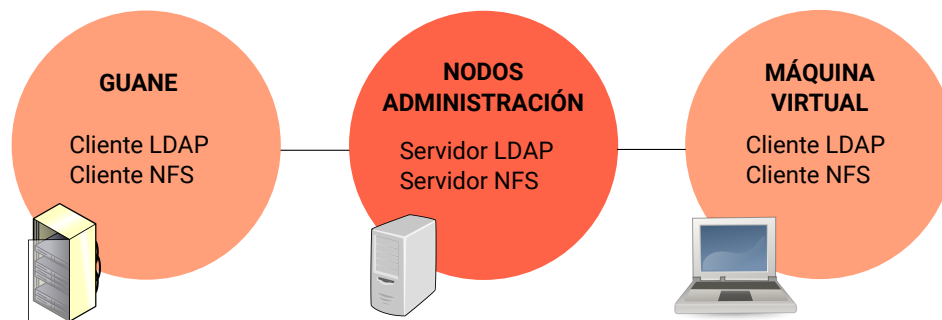


Figura 5.2: Servicios LDAP y NFS implementados en la arquitectura. Fuente: Autor

Ambas implementaciones funcionan bajo el modelo cliente-servidor, en este caso los servicios desplegados se encuentran en los nodos de administración de GUANE (correspondiente a la parte del servidor). Respecto a los clientes, estos se conectan a los demonios de LDAP y NFS tanto desde las máquinas virtuales como de los nodos de GUANE.

5.1.2. Procesamiento

El segundo elemento hace referencia a la plataforma de cómputo donde se realizan las tareas de cálculo científico, que en este caso es un supercomputador.

En el supercomputador se pueden ejecutar algoritmos bajo paradigmas de computación paralela como multicpu (lenguajes de programación como OpenMP y OpenMPI) y multigpu (lenguajes de programación como CUDA y OpenCL).

¹Network File System

²Networked Attached Storage

³Lightweight Directory Access Protocol

El Centro de Supercomputación y Cálculo Científico de la Universidad Industrial de Santander SC3-UIS, proporciona el supercomputador mas rápido de Colombia a investigadores de diferentes campos de la ciencia. La infraestructura de GUANE [1] está compuesta por 16 nodos de cómputo, la configuración de los nodos se encuentra en la tabla 5.1 .

8 nodos con	
Procesador	2 Intel(R) Xeon(R) CPU E5645 @ 2.40GHz. (12 Cores 2 Hebra)
Memoria RAM	104 GB
Almacenamiento	SAS 200 GB
GPU	8 Tesla M2075
3 nodos con	
Procesador	2 procesadores Intel(R) Xeon(R) CPU E5645 @ 2.40GHz. (12 Cores 2 Hebra)
Memoria RAM	104 GB en RAM
Almacenamiento	SAS 200GB
GPU	8 Tesla S2050
5 nodos con	
Procesador	2 procesadores Intel(R) Xeon(R) CPU E5640 @ 2.67GHz (8 Cores 2 Hebra)
Memoria RAM	104 GB
Almacenamiento	SAS 200GB
GPU	8 Tesla S2050

Tabla 5.1: Infraestructura GUANE. Fuente: CAGE

En GUANE se ejecutan los diferentes algoritmos que tienen como resultado una visualización de datos, mostrados en las maquinas virtuales. El envío de los algoritmos se realiza mediante un manejador de recursos llamado SLURM.[8]

5.1.3. Visualización e interacción

El tercer y último elemento es el lugar donde el usuario realiza la visualización e interacción con la misma.

Se tomó la decisión de trabajar con NVIDIA GRID TM, por las capacidades que ofrecía para compartir los recursos entre los usuarios y a su vez el buen desempeño para realizar visualización remota. Se inicia con implementación en Centos la cual no permite crear maquinas virtuales y es menos flexible para el usuario, se toma la decisión de cambiar a la implementación usando hipervisor.

El servidor permite creación de escritorios virtuales, accediendo a la tarjeta gráfica

y visualizando sus resultados remotamente. Además permite que múltiples usuarios accedan a la misma GPU, aprovechando los recursos eficientemente.

Durante la interacción el usuario tiene acceso a los nodos de supercomputo que se encuentran en la misma red local, obteniendo de esta manera una plataforma de supercomputo para procesamiento y una GPU dedicada para propósitos gráficos, la cual aporta una característica adicional muy importante. Es importante aclarar que la conexión a las máquinas virtuales se realiza remotamente. El acceso a GUANE desde las máquinas virtuales se realiza mediante conexión SSH.

La GPU utilizada para el servidor de procesamiento gráfico es la tarjeta NVIDIA GRIDTM K2 instalada en un servidor HP ML350R el cual tiene las siguientes características (Tabla 5.2).

Modelo	HP ML350R
Procesador	Intel Xeon 6 Cores E5-2609
Memoria RAM	16 GB
Almacenamiento	500 GB HDD
GPU	1 Nvidia GRID K2

Tabla 5.2: Especificaciones nodo de procesamiento gráfico Fuente: Autor

En la tabla 5.3 se muestran las características de la tarjeta gráfica [19].

Modelo	K2
Numero de GPUs	2 procesadores GPUs Kepler
CUDA cores	3072
Memoria	8192 MB DDR5
Potencia	225 W
PCIe	x16

Tabla 5.3: Especificaciones de NVIDIA GRID K2 Fuente: AutorTM

Adicionalmente a las especificaciones de hardware de la tarjeta gráfica, como se dijo anteriormente la NVIDIA GRID K2 se puede dividir entre múltiples máquinas virtuales. En la tabla 5.4 se observan los perfiles GPU que puede adoptar esta tarjeta [10].

Perfil vGPU	Memoria gráfica	Max. Pantallas	Max. Resolución	Max. Usuarios por tarjeta	Uso
K280Q	4096 MB	4	2560x1600	2	Diseño avanzado o Ingeniería
K260Q	2048 MB	4	2560x1600	4	Diseño, Ingeniería o Usuario avanzado
K240Q	1024 MB	2	2560x1600	8	Diseño de Ingeniería, Usuario avanzado
K220Q	512 MB	2	2560x1600	16	Diseño o Usuario avanzado

Tabla 5.4: Especificaciones de NVIDIA GRID K2 TM Fuente: Autor

Los clientes remotos no requieren de hardware especializado para tareas de cómputo o visualización de de alto rendimiento. Requieren de una conexión a internet estable, pueden interactuar desde computadores de escritorio o equipos portátiles como laptops o tablets. Además los clientes remotos pueden aprovechar una alta resolución, utilizando muros de visualización como herramienta para apoyar la colaboración científica debido a que múltiples usuarios pueden trabajar en un mismo lugar.

El servidor de procesamiento gráfico requiere de algunos nodos adicionales dependiendo de la solución que se implemente. En este caso se utiliza VMWARE[4] como hipervisor, para la creación de máquinas virtuales.

A continuación se mencionan los servicios requeridos por VMWARE para su implementación en la arquitectura YAJE. [20].

- vSphere ESXi: Es una plataforma para virtualización a nivel de centro de datos, en otras palabras un hipervisor que se ejecuta directamente en el hardware (a nivel inferior a la capa de virtualización) sin la necesidad de un sistema operativo. Está conformado de un sistema operativo autónomo que le proporciona un entorno de gestión, administración y ejecución al software hipervisor, y los servicios y servidores que permiten la interacción con el software de gestión y administración y las máquinas virtuales.

Modelo	HP ML350R
Procesador	Intel Xeon 6 Cores E5-2609
Memoria RAM	16 GB
Almacenamiento	500 GB HDD
GPU	1 Nvidia GRID K2
Tipo	Servidor Físico

Tabla 5.5: Especificaciones nodo de procesamiento gráfico. Fuente: Autor

- vCenter Server: Este servicio actúa como un administrador central para los servidores VMware ESXi conectados a la red. Proporciona también un punto central para configuración, provisión y administración de máquinas virtuales en el data-center.

Procesador	Intel Xeon 6 Cores E5-2650
Memoria RAM	8 GB
Almacenamiento	50 GB HDD
Tipo	Máquina virtual

Tabla 5.6: Especificaciones nodo vCenter Server. Fuente: Autor

- View Connection Server: Este servicio de software actúa como un intermediario para las conexiones de clientes. Autentica a los usuarios a través del servicio de directorio activo de Windows y dirige la solicitud a la máquina virtual apropiada, al Computador físico o al host de Escritorio remoto en sesión.

Procesador	Intel Xeon 6 Cores E5-2650
Memoria RAM	2 GB
Almacenamiento	50 GB HDD
Tipo	Máquina virtual

Tabla 5.7: Especificaciones nodo View Connection Server. Fuente: Autor

- View Security Server: Es un servicio que permite conectarse a las máquinas virtuales fuera de la red local. Se puede instalar y configurar como un servidor de seguridad fuera del firewall de la red local de la universidad en la DMZ ⁴, para ello se añaden reglas de firewall donde los puertos requeridos son aceptados garantizando así el único tráfico remoto de escritorios y aplicaciones que puedan ingresar. y a su vez los usuarios solo pueden acceder a los recursos que estén autorizados.

Procesador	Intel Xeon 6 Cores E5-2650
Memoria RAM	2 GB
Almacenamiento	50 GB HDD
Tipo	Máquina virtual

Tabla 5.8: Especificaciones nodo View Security Server. Fuente: Autor

- Microsoft Active Directory: Servicio utilizado por Microsoft para realizar la administración de acceso a escritorios y aplicaciones remotas en una red distribuida. En este caso Microsoft Active Directory se usa para realizar la autenticación de

⁴Demilitarized Zone

los usuarios en los servicios de VMWare como Vcenter Server, View Connection Server y View Security Server; adicionalmente se utiliza para administrar usuarios entre las máquinas virtuales Windows que se crean.

Procesador	Intel Xeon 6 Cores E5-2650
Memoria RAM	2 GB
Almacenamiento	50 GB HDD
Tipo	Máquina virtual

Tabla 5.9: Especificaciones nodo Microsoft Active Directory. Fuente: Autor

- Horizon Client: Software utilizado para conectarse a las máquinas virtuales. Está públicamente disponible para su descarga y puede instalarse en muchos dispositivos diferentes.

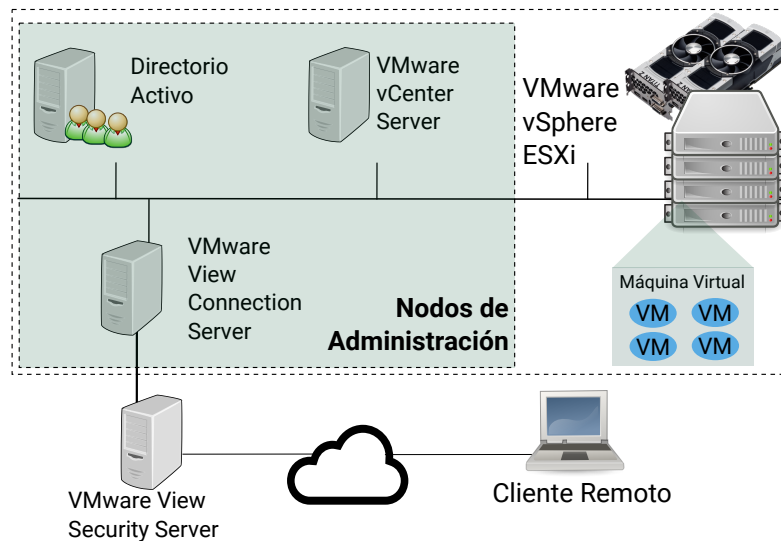


Figura 5.3: Arquitectura para virtualización de escritorios con vGPU. Fuente: Autor

5.2.Integración de los componentes de la arquitectura

En la gráfica 5.4 se presentan una síntesis de los componentes de la arquitectura. Esta gráfica permite ilustrar de manera global la disposición y conformación de la arquitectura desde cada uno de sus componentes: Nodos administrativos, Supercomputador, Servidor de procesamiento gráfico, Máquinas virtuales y Cliente remoto.

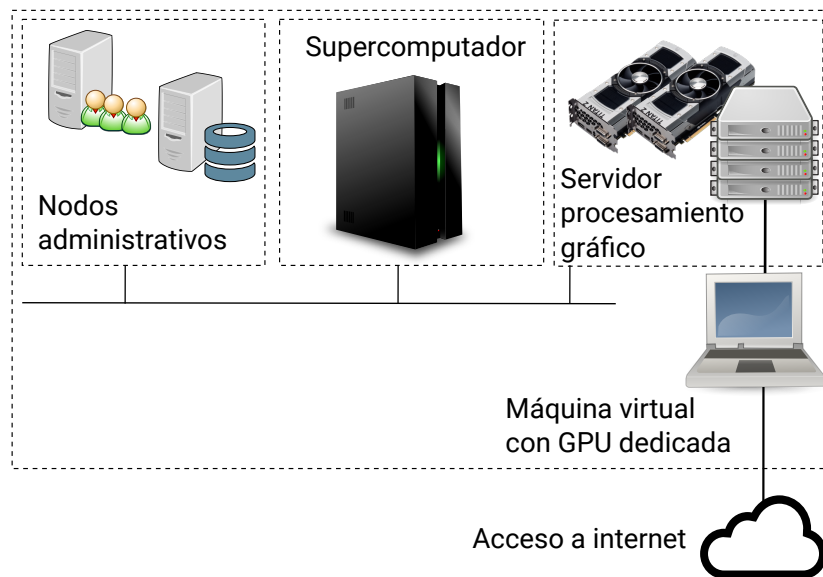


Figura 5.4: Arquitectura Global. Fuente: Autor

Para entrar en detalle de la disposición de los componentes de YAJE, en la figura 5.5 se desglosan los nodos requeridos por el servidor de procesamiento gráfico. La figura muestra las posibles alternativas de implementación ya sea en servidores físicos o máquinas virtuales para cada uno de los nodos requeridos por la arquitectura.

Durante este capítulo se presentaron cada uno de los componentes de la arquitectura propuesta, así como sus especificaciones de hardware y software y la importancia de cada uno dando cumplimiento al primer objetivo escrito anteriormente. En el siguiente capítulo se desarrolla pruebas de rendimiento ejecutadas en YAJE, utilizando visualización, interacción y cómputo de alto rendimiento en cada una de ellas.

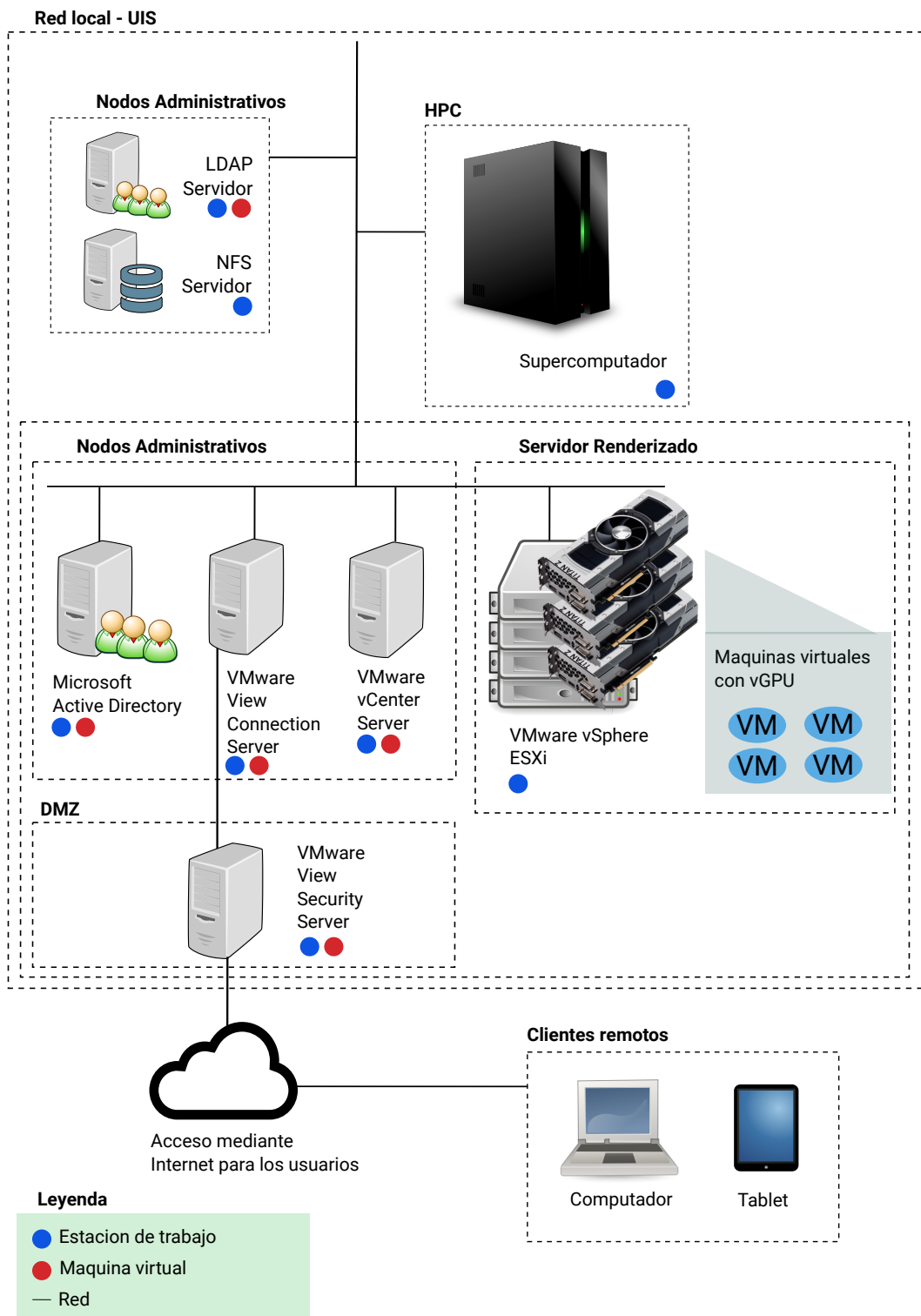


Figura 5.5: Arquitectura YAJE. Fuente: Autor

6.LINEAMIENTOS

En este capítulo se establecieron lineamientos o pautas con el fin de orientar a quienes deseen implementar y ejecutar aplicaciones en este tipo de arquitecturas, generados durante el desarrollo de este proyecto. Los lineamientos descritos a continuación se basan en la experiencia obtenida durante el desarrollo del proyecto, la revisión de manuales, marco teórico y pruebas ejecutadas. A continuación se mencionan estos lineamientos.

6.1.Lineamientos para la diseño e implementación de la arquitectura computacional

Los lineamientos pretenden ser una declaración de principios o directrices que le ayuden al investigador a desarrollar implementaciones a su medida, cuando se requiriera de cómputo de alto rendimiento acompañado de procesos de visualización de resultados.

A continuación se da paso a la descripción de los lineamientos encapsulados en cuatro etapas metodológicas: Definición del alcance, Planificación de la Arquitectura, Diagrama de la Arquitectura e Implementación de la Arquitectura. Se agregarán a sus vez notas del caso de aplicación que lleven al contexto real la experiencia durante el desarrollo de YAJE y sirvan como ejemplo para el lector.

1. **Definición del Alcance:** Es la primera y mas importante de las actividades a realizar en el proceso de elaboración de una arquitectura para cómputo y visualización, pues de esta fase depende que todos los pasos subsiguientes estén alineados a los requerimientos definidos.

En la definición del alcance es necesario justificar el por qué, para qué y para quién se requiere la arquitectura; se deben identificar los componentes generales a usar dependiendo del uso que se le vaya a dar a la solución. Los componentes hacen referencia a los pilares sobre los cuales se construye la arquitectura computacional: Administración (Usuarios y almacenamiento), Procesamiento (Computo de alto rendimiento) y Visualización (Visualización e Interacción).

En esta sección se pueden formular varias preguntas respecto a los requerimientos mínimos de la arquitectura o los pilares de la misma, recomendado obtener respuestas en principio de forma general que permitan posteriormente llegar a realizar la planificación y diagramación específica.

Preguntas básicas para determinar el alcance de la arquitectura computacional:

- ¿Cómo se realiza el acceso de los usuarios?
- ¿Qué cantidad de recursos para visualización necesita el usuario?

- ¿Qué cantidad de recursos de cómputo necesita el usuario?
- ¿Cuál es la cantidad de datos que se van a procesar?
- ¿El software requerido es el mismo para todos los usuarios?

Nota del Caso de Aplicación: El alcance del trabajo de investigación fué ofrecer a investigadores de múltiples áreas de la ciencia una herramienta/plataforma/arquitectura que permitiera realizar visualización y cómputo en arquitecturas de alto desempeño, sin afectar la experiencia de usuario ni el tiempo de procesamiento.

2. Planificación de la Arquitectura: La planificación hace referencia al cómo se va a dar cumplimiento al alcance definido.

Si bien, al hablar de 'planificación' inmediatamente se asocia con cronogramas y presupuestos para cumplir el objetivo o alcance definido, en este contexto se abordará la planificación como el momento en que se deben establecer los componentes de hardware y software que requiere la arquitectura, los cuales no necesariamente deben ser los componentes finales pues es una etapa temprana del desarrollo.

En cuanto a los tiempos y actividades se recomienda establecer un listado de chequeo o etapas a cumplir en el proceso de construcción, implementación y pruebas de la arquitectura; de manera tal que se pueda llegar a realizar auditoría al plan para observar los avances y el cumplimiento del mismo. Se sugiere agregar a las actividades listadas los requerimientos hardware y software, e inclusive los del talento humano, para así tener un estimativo del costo del desarrollo de la arquitectura computacional.

Con la finalización de la etapa de planificación se logrará obtener una visión mas global de la solución y se estará listo para continuar con la diagramación.

Nota del Caso de Aplicación: En la solución planteada en este documento se ofrece visualización y cómputo como servicio, el cual permite a los usuarios acceder de forma remota. Se tomó también la decisión de usar máquinas virtuales debido a la flexibilidad que estas presentan para la personalización del sistema operativo, lo cual permite que el usuario instale sus aplicaciones para realizar visualización de resultados sin afectar a otros usuarios ni el esquema de procesamiento que tiene GUANE.

3. Diagrama de la Arquitectura: Es el paso previo antes de iniciar con la implementación. Para ello se debe realizar un diagrama que describa los componentes requeridos para la puesta en marcha de la solución. Este proceso está en continuo desarrollo, pues existen componentes que pueden añadirse o eliminarse de la arquitectura. Cabe mencionar que no existe un diagrama absoluto, tampoco uno que este del todo mal ni un perfecto, cada vez pueden encontrarse mejoramientos sobre el mismo que lo van enriqueciendo.

Nota del Caso de Aplicación: Inicialmente se planteó un diagrama que omitía muchos de los elementos existentes en YAJE, sin embargo con el paso del tiempo

y de la investigación se fue enriqueciendo hasta obtener el diagrama final como se muestra en la figura 5.5.

4. **Implementación de la Arquitectura:** La implementación es la ejecución del modelo realizado en la fase de diagramación. Es importante resaltar que el modelo debe contar con los componentes básicos establecidos para el funcionamiento exitoso de la arquitectura los cuales son alusivos a los pilares sobre los cuales se construye la arquitectura computacional: Administración (Usuarios y almacenamiento), Procesamiento (Computo de alto rendimiento) y Visualización (Visualización e Interacción).

Para la implementación se debe contemplar:

- Administración: Definición y configuración de protocolos para manejo de archivos y usuarios. Evaluar en primera instancia cuáles son los protocolos mas utilizados y que brindan soluciones de código abierto con soporte multiplataforma.

Nota del Caso de Aplicación:(USUARIOS/ALMACENAMIENTO) En el caso desarrollado se escogieron los protocolos tanto para mejor de archivos como usuario debidos a las bondades y ventajas que presentan al garantizar la sesión y la disponibilidad de la información almacenada. LDAP para manejo de usuarios en red y para manejo de archivos en red NFS.

- Procesamiento: Establecer o utilizar los manejadores de recursos para enviar trabajos al clúster o nodos de procesamiento.

Nota del Caso de Aplicación:(PROCESAMIENTO) Para el procesamiento se utiliza GUANE, un clúster heterogeneo con 16 nodos. GUANE administra sus trabajos mediante un manejador de recursos llamado SLURM. Esta implementacion ya se encontraba en GUANE por lo que no se requirio realizar cambios dentro del cluster ni dentro de las maquinas virtuales, pues SLURM recibe los trabajos desde el frontend de GUANE el cual es accedido por conexion SSH ¹ desde la maquina virtual.

- Visualización: Escoger e instalar la herramienta de visualización que se quiere usar. La virtualización en GPU se selecciona para obtener mejor rendimiento gráfico con NVIDIA GRID. Tiene soluciones comerciales disponibles en el mercado como Citrix o VMware, que proveen el software necesario para usar los perfiles de vGPU en máquinas virtuales con diversos sistemas operativos. No obstante, si ninguna de las soluciones comerciales satisface las necesidades del cliente, se puede optar por el uso de hipervisores gratuitos KVM u OpenXen que no tienen soporte por parte del fabricante (NVIDIA); para ello se deben desarrollar los controladores de la tarjeta de manera tal que funcionen en el hipervisor, pues éstos drivers sólo existen para soluciones comerciales. Por otro lado dependiendo de las necesidades del usuario

¹Secure Shell - <https://www.openssh.com/>

respecto al rendimiento gráfico se escoge el tipo de perfil de GPU que se va a virtualizar.

Nota del Caso de Aplicación:(VISUALIZACIÓN E INTERACCIÓN) Para la implementación de la arquitectura propia YAJE se encontraron soluciones comerciales para realizar virtualización. Se optó por la creación de máquinas virtuales con acceso directo a los dispositivos hardware de la máquina mediante el uso de la tecnología Passthrough usando el perfil vDGA. La tecnología Passthrough hace posible el acceso directo de una máquina virtual a un dispositivo, en este caso una tarjeta gráfica conectada mediante un dispositivo PCI Express. Este perfil es escogido ya que da el mejor rendimiento gráfico, la tarjeta NVIDIA GRID K2 permite hasta dos usuarios trabajando simultáneamente con este perfil. El hipervisor utilizado fue VMWARE, el cual requiere algunos nodos adicionales para su administración.

Lineamientos generales propios de YAJE

Adicionales a los lineamientos descritos anteriormente para elaboración de arquitecturas computacionales para dar solución a requerimientos de cómputo y visualización similares a YAJE, se encontró factible incluir esta sección para listar lineamientos generales encontrados durante el desarrollo de YAJE con aras de facilitar la comprensión de los requerimientos utilizados y a su vez permitan dar continuidad a la misma en trabajos futuros.

- El uso de los nodos de administración (LDAP para el manejo de usuarios y NFS para datos) permiten que en las maquinas virtuales se puedan acceder a los mismos usuarios utilizados en GUANE y a sus respectivos archivos. Cualquier usuario registrado en GUANE puede usar las maquinas virtuales.
- Se requieren nodos de administración para los nodos de procesamiento gráfico. En este caso VMware utiliza 4 nodos que pueden ser implementados en maquinas virtuales con características de hardware convencionales.
- No existen en la actualidad controladores para la instalación de máquinas virtuales Linux con vGPU con las tarjetas NVIDIA GRID de la familia Kepler, debido a que los fabricantes no han desarrollado los controladores. Por ello si se desean implementar se deben desarrollar los drivers para los perfiles vGPU en Linux. Para la familia Maxwell si existen los controladores de la tarjeta gráfica (Ejemplo NVIDIA M60).
- Se pueden crear maquinas virtuales Linux con perfiles vDGA, estos drivers si existen por parte de NVIDIA.
- La interacción del usuario con la visualización requiere de un ancho de banda dedicado superior a 8 Mbps.
- El uso de maquinas virtuales provee al investigador hardware para realizar pruebas pequeñas que no requieran HPC o computacion paralela.

Con el desarrollo de estos lineamientos se da cumplimiento al segundo objetivo. En el siguiente capítulo se da desarrollo a los casos de estudio los cuales apoyaron el proceso o modificación de los lineamientos anteriormente nombrados.

7.PRUEBAS DE RENDIMIENTO

En este capítulo se presentan las pruebas de rendimiento ejecutadas en la plataforma que tenían como finalidad medir el uso de los recursos en los principales componentes de la arquitectura en diferentes escenarios.

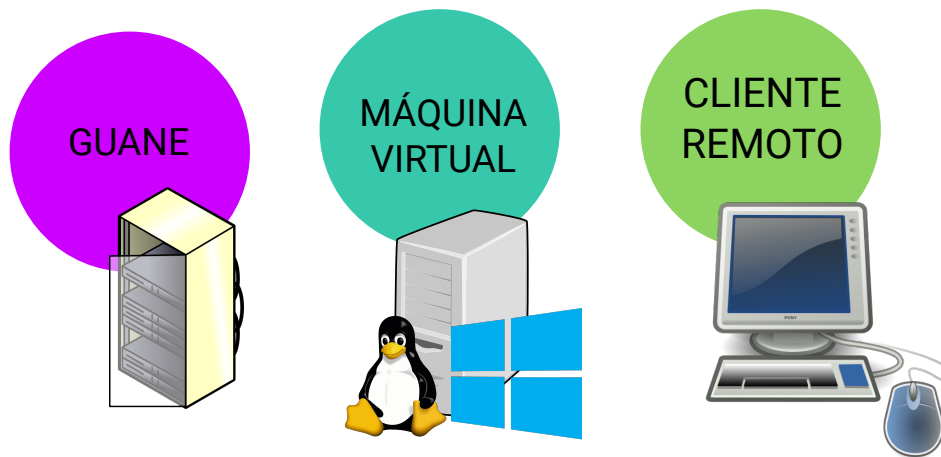


Figura 7.1: Escenarios para medición de pruebas. Fuente: Autor.

Las mediciones en estos escenarios es para comprobar que el uso de recursos sea adecuado en cada uno de ellos. Para ello se realizaron dos casos de estudio donde se enviaban trabajos al clúster y se visualizaban remotamente. En estas pruebas no se espera uso de recursos en la máquina del cliente (desde donde se conectaba a la máquina virtual) ni para cálculo ni para procesamiento gráfico. Además tampoco se debían utilizar recursos en la máquina virtual para cálculo científico. A continuación se mencionan los elementos hardware y software utilizados en estas pruebas, seguido de la explicación de los casos de estudio y resultados obtenidos.

7.1.Hardware

Para la realización de las pruebas se utilizaron las configuraciones de hardware que se presentan en las siguientes tablas. En GUANE se utilizaron hasta tres nodos dedicados para realizar las pruebas de rendimiento de los casos de estudio, las configuraciones de los nodos se presentan a continuación en la tabla 7.1

GUANE08	
SO	Debian Jessie 8
Procesador	2 procesadores Intel(R) Xeon(R) CPU E5640 @ 2.67GHz (8 Cores 2 Hebra)
Memoria RAM	104 GB
Almacenamiento	SAS 200GB
GPU	8 Tesla S2050
GUANE14	
SO	Debian Jessie 8
Procesador	2 Intel(R) Xeon(R) CPU E5645 @ 2.40GHz. (12 Cores 2 Hebra)
Memoria RAM	104 GB
Almacenamiento	SAS 200 GB
GPU	8 Tesla M2075
GUANE15	
SO	Debian Jessie 8
Procesador	2 procesadores Intel(R) Xeon(R) CPU E5645 @ 2.40GHz. (12 Cores 2 Hebra)
Memoria RAM	104 GB en RAM
Almacenamiento	SAS 200GB
GPU	8 Tesla S2050

Tabla 7.1: Infraestructura GUANE. Fuente: Autor.

Se realizó la creación de múltiples máquinas virtuales en el servidor, dicha configuración se mantiene durante todas las pruebas ejecutadas. A continuación se presenta la configuración en la tabla 7.2

SO	Ubuntu 16.04
Procesador	Intel Xeon 6 Cores E5-2609
Memoria RAM	4 GB
Almacenamiento	100 GB HDD
GPU	1 NVIDIA GRID K2 (Perfil vDGA)

Tabla 7.2: Especificaciones configuración Máquinas Virtuales. Fuente: Autor.

Por último el cliente remoto tiene la siguiente configuración, la cual es detallada en la tabla 7.3.

SO	openSUSE Leap 42.1
Procesador	Intel Core i7 4700 MQ 2.40 GHZ
Memoria RAM	6 GB
Almacenamiento	1000 GB HDD
GPU	1 NVIDIA GeForce 755M

Tabla 7.3: Especificaciones Cliente Remoto. Fuente: Autor.

7.2. Software

El software utilizado para realizar las pruebas de rendimiento es el siguiente:

- Camino: Librería en JAVA para el estudio de tractografías imágenes de tensor de difusión [21].
- DIPY: Es un software de código abierto para el estudio de imágenes de difusión de resonancia magnética (dMRI). Esta librería esta implementada en PYTHON [22].
- VisIt: Visualize It es un software libre para visualización científica. Se puede utilizar en modo cliente-servidor apoyándose de clusters de computadoras para el procesamiento [23].
- top: Permite realizar el monitoreo de CPU y RAM en tiempo real de los procesos que se están ejecutando. [24]
- nethogs: Herramienta para monitoreo de red por aplicaciones (PID), permitiendo comprobar el ancho de banda que esta consumiendo cada proceso [25].

Esta software se utiliza tanto para el desarrollo de las pruebas como el monitoreo de las mismas.

7.3. Casos de estudio

Las pruebas de rendimiento se realizaron con dos casos de estudio, el primero sobre procesamiento de imágenes médicas donde se buscó hacer énfasis en una aplicación enfocada en el usuario, donde se pudiera realizar visualización e interacción con las salidas generadas a partir del procesamiento. En este escenario se lograron mediciones sobre ancho de banda requerido para que la interacción remota del usuario fuera fluida y en tiempo real; a demás que al realizar el procesamiento éste pudiera hacerse sin la necesidad hacer transferencia de datos recurrentes.

Por otro lado en el caso dos, sobre generación de modelos 3D de fluidos, el trabajo fue enfocado en la realización pruebas y obtención de métricas que permitieran observar y monitorear el uso de los recursos en la máquina virtual y el supercomputador; de

manera tal que se lograran establecer comparaciones y patrones de procesamiento local versus el procesamiento remoto.

7.3.1.Caso UNO: Tractografías

En el primer caso de estudio se contó con el apoyo brindado por el grupo de investigación IMAGINE de la Universidad de los Andes, quienes realizaron el acompañamiento durante el proceso de análisis con usuarios expertos y proporcionando las imágenes para realizar el procesamiento [26].

Este caso de estudio se realizó con imágenes médicas correspondientes a imágenes diagnósticas de cerebro, particularmente con imágenes de tensor de difusión llamadas tractografías.

En neurociencias una tractografía, es un método para representar los tractos neuronales que son las conexiones anatómicas en el cerebro humano. Este método utiliza los datos generados por una imagen de tensor de difusión (DTI)¹, la cual se obtiene de un procesamiento realizado a imágenes de resonancia magnética (MRI)².

Durante la ejecución de este caso de estudio, se hallaron diferentes secciones del algoritmo donde es necesaria la interacción del usuario la cual es realizada en la máquina virtual, cada uno de los detalles se menciona a continuación.

Generación de una imagen de Tractografía

La generación de la imagen de tractografía se realizó con dos métodos, el primero utilizando la librería Camino [21] y el segundo con la librería DIPY [22]. Usando Camino el usuario tiene mayor interacción con la aplicación, el cual era una de las tareas que se quería realizar en la investigación. Con DIPY el programa es enviado al cluster esperando únicamente la salida, en este caso la interacción usuario/aplicación es previa a realizar el procesamiento.

1. Camino

- Grupo de datos inicial: Para generar las imágenes de tractografías se inicia el procesamiento con imágenes de resonancia magnética. Estas imágenes son obtenidas mediante la técnica de difusión por resonancia magnética (DWI)³, están representadas en archivos con formato Nifti[27].

Adicional a la imagen DWI el escáner de resonancia magnética genera la dirección de los vectores, que es la dirección que recorre un tracto. Por último se utiliza una máscara binaria de imagen DWI para separar el cerebro del fondo. La tabla 7.4 presenta un ejemplo de los datos iniciales.

¹Diffusion Tensor Imaging

²Magnetic Resonance Imaging

³Diffusion Weighted Imaging

Archivo	Descripción
4Ddwi_b1000.nii.gz	RAW DWI en formato NifTI-1
grad_dirs.txt	Lista de direcciones de los gradientes
brain_mask.nii.gz	Máscara binaria que separa el cerebro del fondo

Tabla 7.4: Ejemplo grupo de datos iniciales. Fuente: Autor.

- Generación de la imagen de tractografía: Se genera la visualización 3D de la imagen de tractografía de todo el cerebro. Para esto se trabaja con librerías de visualización científica como VTK ⁴ cuyo formato de archivo puede ser visualizado en VisIt, Paraview o software especializado para imágenes de tractografías, en este caso se usa la librería para visualización científica de imágenes médicas (SMILI) ⁵. Para generar la imagen de tractografía se debe establecer una semilla (ROI ⁶), que calcula los tractos que están en el área donde la semilla se encuentra. En este caso se toma la imagen de todo el cerebro como la semilla para obtener todos los tractos. Este método no requería de gran poder de cómputo, por esta razón se puede generar la imagen en la máquina virtual sin la necesidad de hacer uso de GUANE.

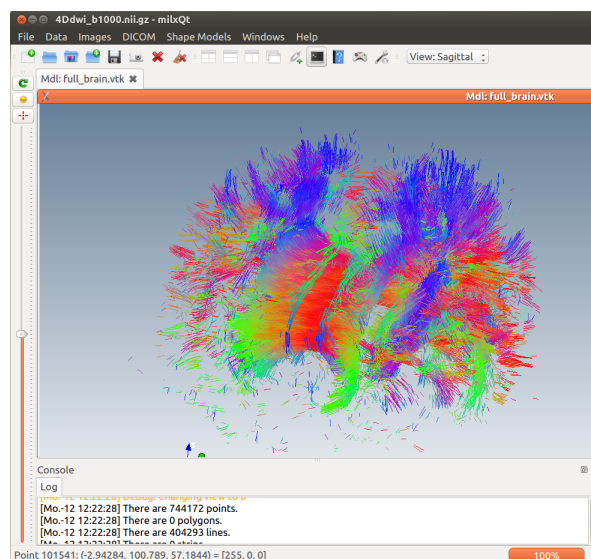


Figura 7.2: Imagen de tractografía del cerebro. Fuente: Autor.

Ademas de obtener la imagen de tractografía de todo el cerebro, se pueden generar los tractos de una sola región. El primer paso es establecer una región de interés en la que se van a buscar los tractos. En este paso el investigador

⁴Visualization Toolkit

⁵Simple Medical Imaging Library Interface

⁶Region of interest/Región de interés

debe interactuar con la imagen, seleccionando la semilla o ROI ⁷. La región de interés es seleccionada de la imagen MRI.

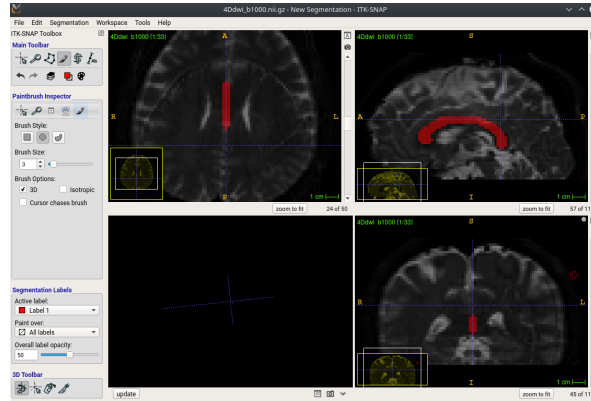


Figura 7.3: Región de interés seleccionada para la búsqueda de tractos. Fuente: Autor.

En la imagen 7.3 se selecciona el cuerpo calloso, que es la estructura que une los dos hemisferios del cerebro. Al realizar el cálculo de fibras con esta región de interés se espera una gran cantidad de datos.

- Cálculo de fibras utilizando un método probabilista para una región de interés [28]: este método probabilístico es una implementación del método de Bayesiano propuesto por Friman et al [29]. En este caso se hace uso de GUA-NE debido al tiempo requerido por este método.

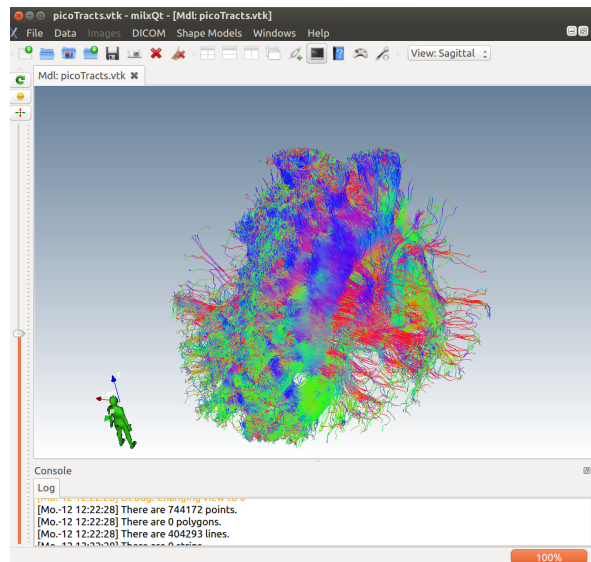


Figura 7.4: Tractos en región de interés con método probabilístico. Fuente: Autor

2. DIPY

⁷Region of interest

El proceso inicia con imágenes de resonancia magnética (MRI) utilizando un método probabilista para generar la imagen de tractografía. Esta tarea fue ejecutada en GUANE, enviando el trabajo al cluster usando el manejador de recursos SLURM desde la máquina virtual. Una vez generada la imagen se procede a realizar el clustering de tractos y finalmente su visualización en la máquina virtual. Cabe resaltar que los colores observados en la tractografía hacen referencia a la orientación de la fibras (izquierda-derecha, arriba-abajo, adelante-atrás).

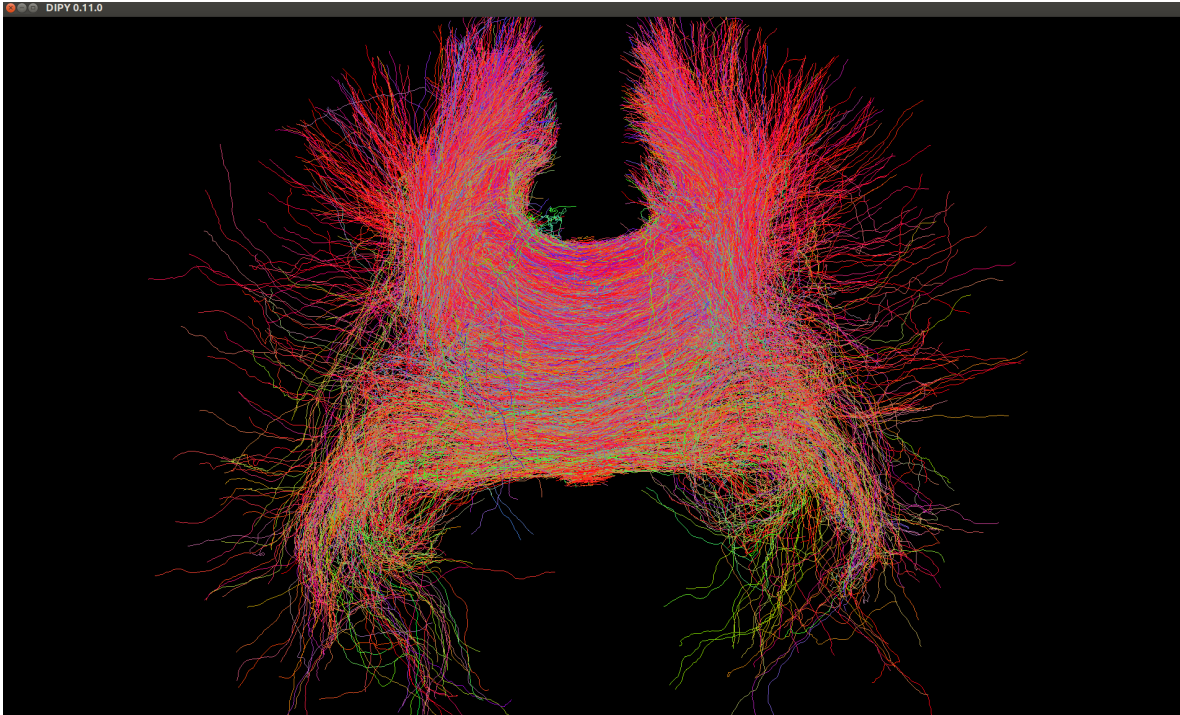


Figura 7.5: Tractos en región de interés con método probabilístico. Fuente: Autor

Durante este ejercicio se dividió el algoritmo en 3 pasos:

- Generación de imagen de tractografía (Procesamiento en GUANE)
- Clustering de tractografía (Procesamiento en la máquina virtual)
- Visualización de tractografía (Visualización en la máquina virtual)

Esta división se planificó desde el inicio del desarrollo, ya que las tareas que se envían al cluster no pueden tener salidas gráficas.

Clasificación de una imagen de Tractografía

La clasificación de tractos se llevó a cabo con DIPY, utilizando un método llamado QuickBundles [30]. Quickbundles es un método rápido de clustering por lo cual no se requiere GUANE para realizar este procesamiento.

En una de las pruebas se obtuvieron 188 clusters de fibras de la imagen de tractografías generada anteriormente. Los clusters se pueden diferenciar por colores en la imagen 7.6.

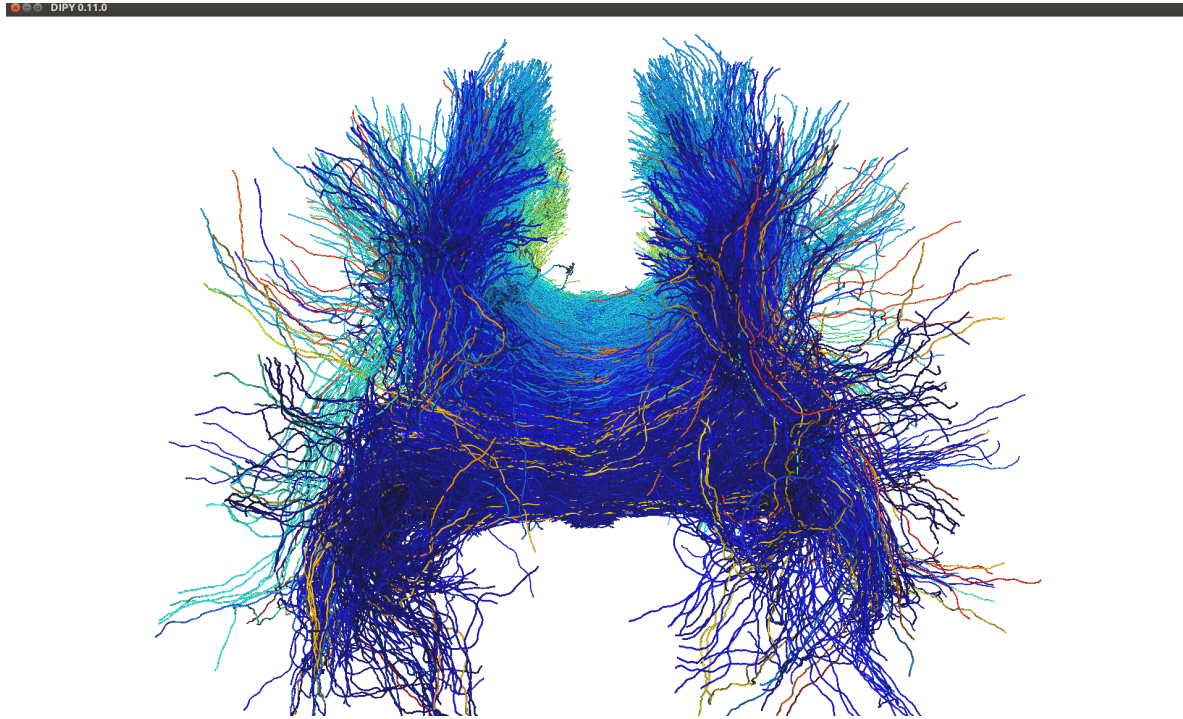


Figura 7.6: Clustering de tractos en el cuerpo calloso - DIPY. Fuente: Autor

Utilizando DIPY el investigador puede cambiar múltiples parámetros para realizar el clustering, y en caso de ser necesario usar GUANE para su procesamiento, unicamente enviando el trabajo al manejador de recursos.

Resultados obtenidos

En este caso de estudio se encontraron secciones que requerían interacción del usuario, computo de alto rendimiento y visualización de resultados, pudiendo realizar estas tareas sin la necesidad de realizar transferencias constantes de datos ni afectar la experiencia de usuario.

Ademas se monitorea la interacción del usuario con los resultados presentados en la visualización. Para ello se calcula el ancho de banda utilizado en el cliente (equipo del usuario remoto) y la maquina virtual, obteniendo los resultados de la figura 7.7

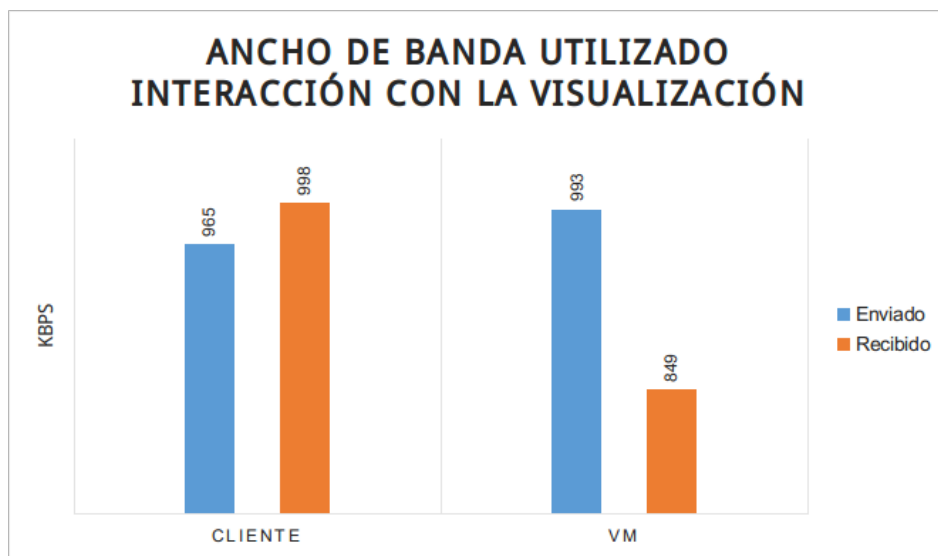


Figura 7.7: Ancho de banda utilizado en interacción con Clustering de tractos. Fuente: Autor

Como se puede observar el ancho de banda utilizado para el envío de paquetes de datos (visualización remota) de la maquina virtual es muy cercano al recibido por el cliente, permitiendo concluir que ambos nodos deben tener al menos una conexión superior a 8 Mbps para realizar la visualización remota del escritorio.

Durante el desarrollo de esta aplicación el usuario podía interactuar desde la maquina virtual, modificando los parámetros de los algoritmos, estos experimentos apoyan la toma de decisiones por parte del investigador sin la necesidad que sea experto en el uso de la arquitectura.

7.3.2.Caso DOS: Simulación de fluidos

El segundo caso de estudio corresponde a la simulación de fluidos. En este caso de estudio los datos fueron provistos por la Technische Universität Kaiserslautern ⁸ (Universidad Técnica de Kaiserslautern, Alemania).

Los datos representan la simulación de dos fluidos a partir de los cuales se pueden generar diferentes tipos de visualizaciones. La simulación está compuesta por una serie de imágenes 3D.

Los datos fueron visualizados en la máquina virtual utilizando VisIt como software de visualización científica. VisIt tiene la habilidad de ejecutarse en paralelo para generar vistas de los modelos a partir del conjunto de datos. El funcionamiento de VisIt en paralelo es realizado mediante un paradigma multicore utilizando MPI, lo cual permite la ejecución en múltiples nodos de GUANE.

⁸<https://www.uni-kl.de>

En la tabla 7.5 se presenta un resumen de una serie de tiempos del conjunto de datos.

Tamaño	361 megabytes
Número de nodos	35'152.000
Metodo de renderizado	RayCast
Samples per ray	4000
Memoria utilizada	2674.43 megabytes

Tabla 7.5: Resumen de datos simulación de fluidos. Fuente: Autor.

En las siguiente gráfica se presentan una de las visualizaciones que se pueden generar con VisIt en esta serie de tiempos.

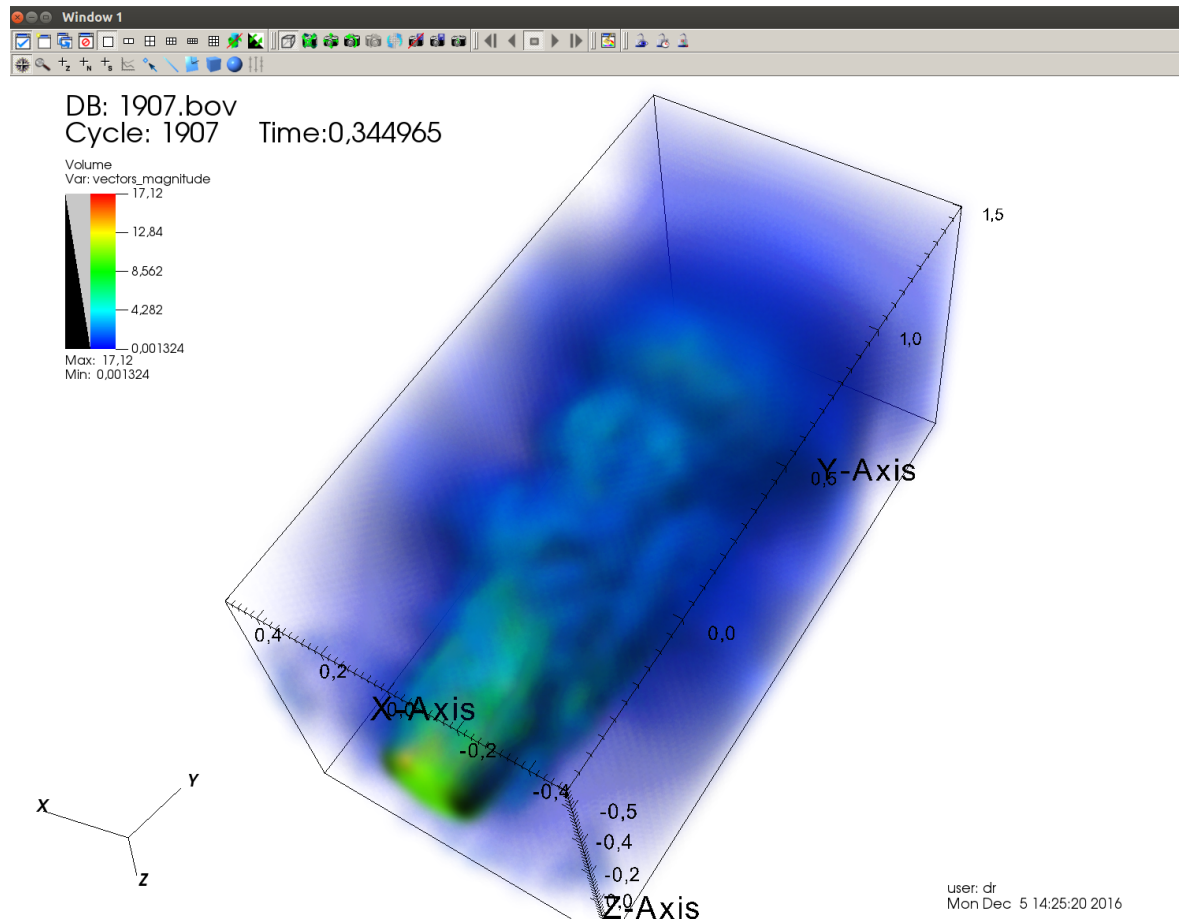


Figura 7.8: Volumen en un periodo de tiempo método Splatting. Fuente: Autor

Para las pruebas ejecutadas se realizó la generación de un volumen 3D utilizando un método de rendering llamado RayCast. En la tabla 7.6 se muestran los parámetros seleccionados y en la figura 7.9 se muestran los resultados usando esta parametrización.

Método de Renderizado	RayCast
Método de muestreo	Rasterización
Samples per ray	4000

Tabla 7.6: Parámetros para el renderizado del modelo. Fuente: Autor.

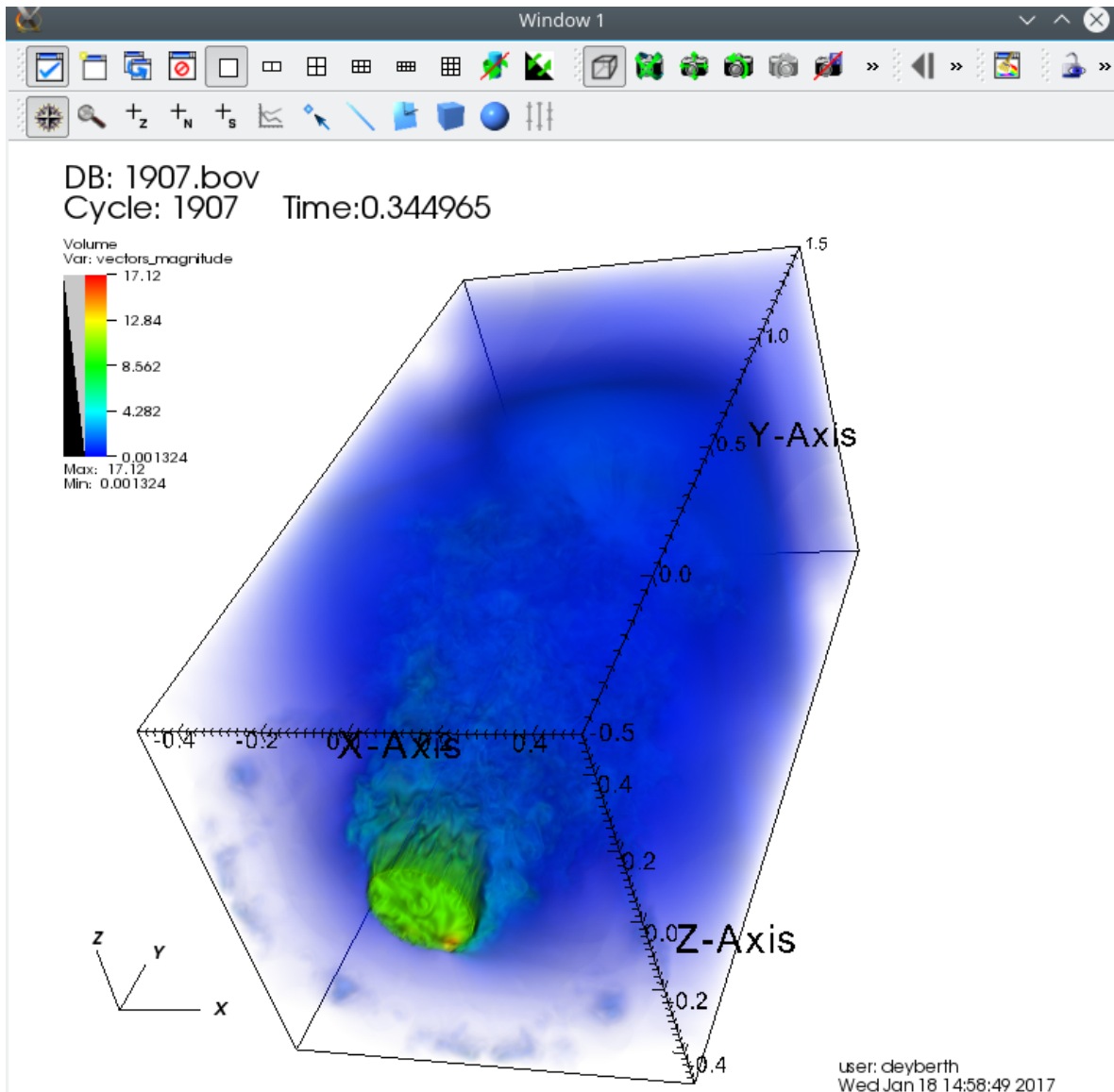


Figura 7.9: Rendering del modelo con método RayCast con 4000 muestras. Fuente: Autor

Durante las pruebas ejecutadas se utilizaron hasta 3 nodos de GUANE para realizar el rendering del volumen modificando la cantidad de procesadores utilizados para esta tarea. En la siguiente sección se detalla este proceso.

Resultados obtenidos

Durante la captura de resultados se realizó el monitoreo de recursos en la máquina virtual y los nodos de GUANE utilizados.

La finalidad de esta prueba era disminuir el tiempo que tardaba la generación del volumen utilizando GUANE como herramienta de procesamiento, encontrando que estos tiempos se reducen significativamente sin afectar el rendimiento de la máquina virtual ni el rendimiento del cliente que utilizaba el servicio.

La tabla 7.7 presenta el resumen de tiempos de ejecución del modelo utilizando RayCast con 500,1000,2000 y 4000 puntos usando la máquina virtual y hasta 3 nodos GUANE.

	Maquina Virtual	GUANE		
RayCast Samples	1 Core [s]	1 Nodo 24 Cores [s]	2 Nodos 48 Cores [s]	3 Nodos 54 Cores [s]
500	18.38	3.73	3.87	4.01
1000	32.56	6.73	6.77	6.53
2000	62.18	12.43	12.15	11.98
4000	139.58	25.74	25.67	23.59

Tabla 7.7: Tiempos [s] de ejecución utilizando RayCast. Fuente: Autor.

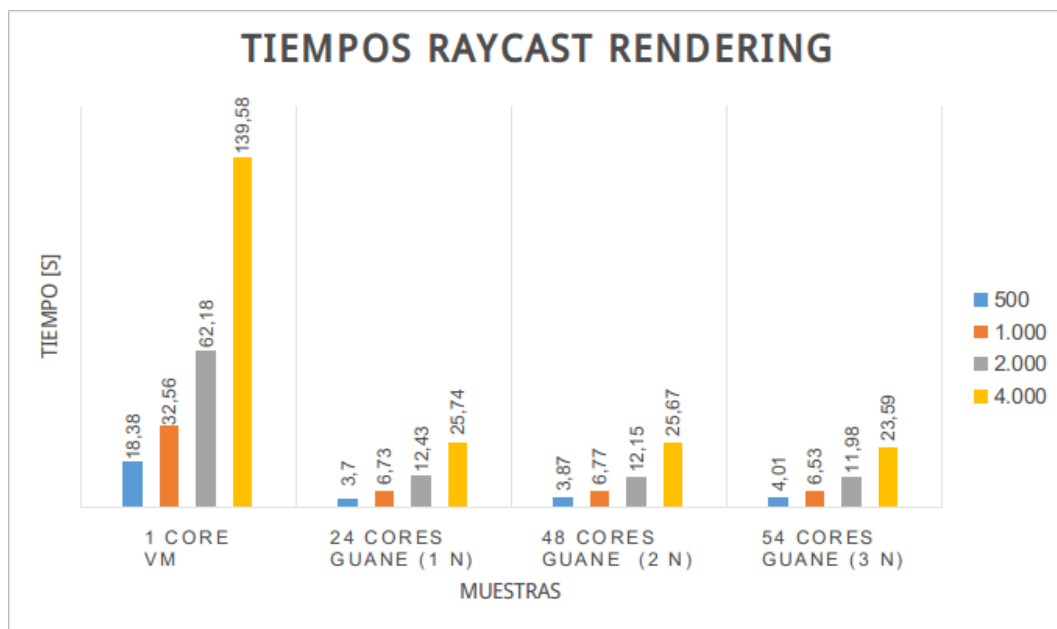


Figura 7.10: Tiempos Raycast Rendering. Fuente: Autor

Como se puede observar en la figura 7.10 los tiempos de generación del volumen en GUANE disminuyen considerablemente respecto a la máquina virtual. Aunque se observa

un comportamiento similar cuando se usan 1,2 o 3 nodos de GUANE. A continuación se muestra el uso de recursos en un nodo GUANE en el cual se envía el proceso de rendering del modelo, sin afectar el rendimiento de la maquina virtual.

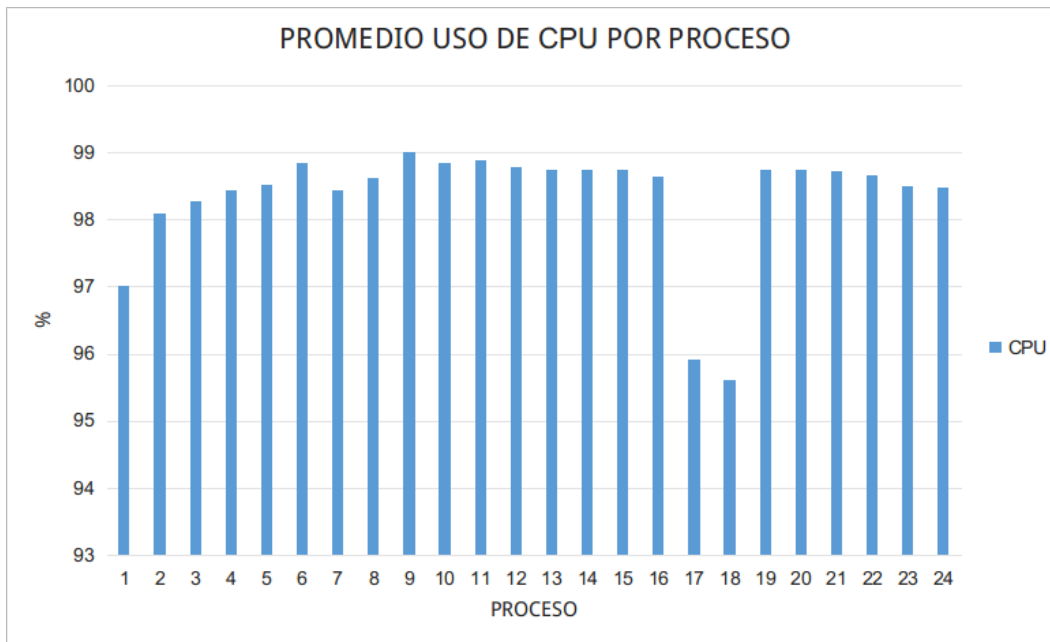


Figura 7.11: Promedio uso de CPU Raycast Rendering. Fuente: Autor

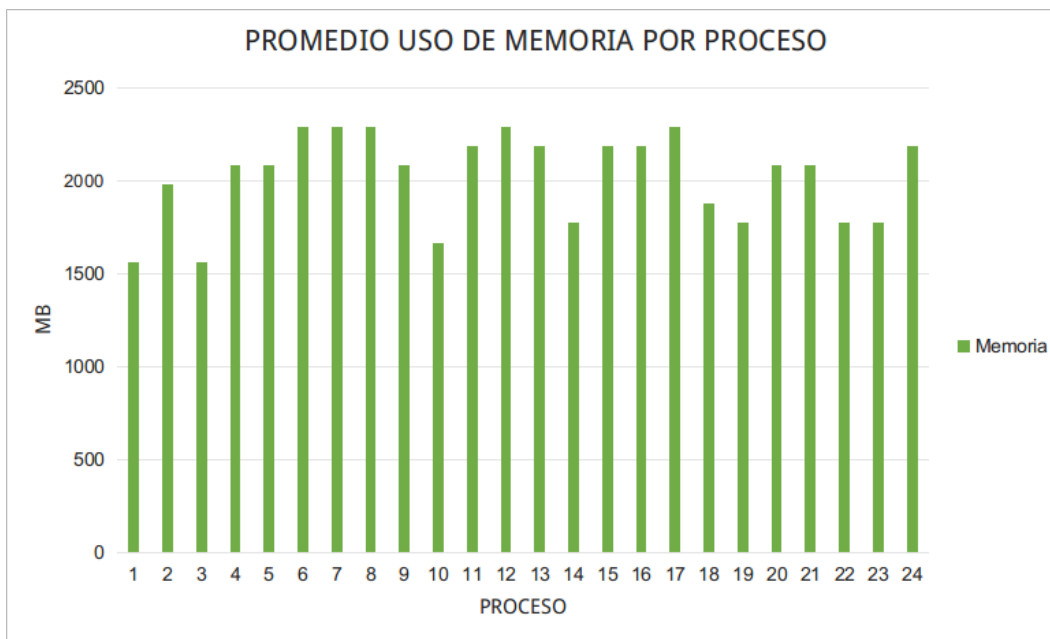


Figura 7.12: Promedio uso de memoria Raycast Rendering. Fuente: Autor

La figura 7.11 muestra el promedio de uso de cada uno de los hilos de ejecución en el procesador (más del 95 % de uso en cada hilo del procesador) , para este caso se utilizaron

24 procesos los cuales realizaron el procesamiento en una quinta parte del tiempo usado por la maquina virtual. En la figura 7.12 se presenta el uso de memoria RAM donde se pudo observar un promedio de uso por proceso de hasta 2.3 GB, generando hasta 46 GB de uso en un segundo para la generación del volumen. Este tipo de recursos no son comunes en estaciones de trabajo, por lo cual se demostraba la importancia no solo de tener una herramienta de visualización remota, sino también del acceso a recursos de cómputo de alto rendimiento como es el clúster GUANE.

Prueba de ancho de banda con vídeo de alta resolución

Esta prueba se realizó para comprobar el ancho de banda requerido por un vídeo donde la visualización cambia constantemente. La prueba arroja resultados similares a los que se obtuvieron con la interacción del clustering de tractos. Se mantiene un ancho de banda estable en el tráfico enviado por la máquina virtual y recibido por el cliente, obteniendo una buena experiencia de usuario sin retardos o delay considerables que afecten la interacción en la visualización. En la figura 7.13 se presentan los resultados de estas pruebas.

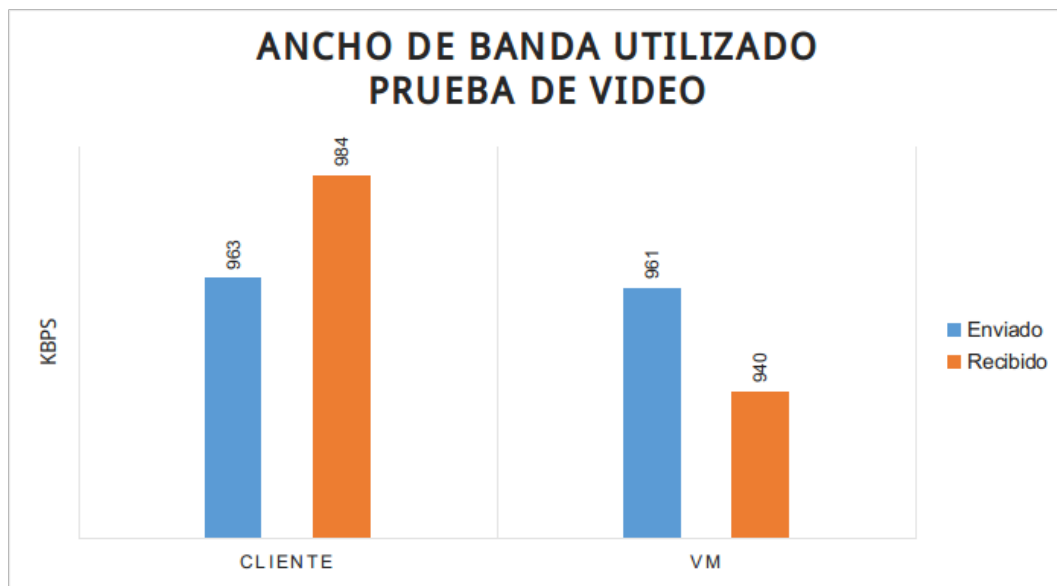


Figura 7.13: Ancho de banda utilizado en prueba de video. Fuente: Autor

Este capítulo presentó la ejecución de las pruebas de rendimiento, que fueron desarrolladas a partir de dos casos de estudio con la colaboración de la Universidad de los Andes y la Universidad Tecnológica de Kaiserslautern. Durante este proceso se realizó el monitoreo de los recursos, obteniendo conclusiones y apoyando el proceso de creación o modificación de lineamientos que fueron escritos anteriormente. A continuación se presentan las conclusiones obtenidas durante este trabajo de investigación.

8.CONCLUSIONES

- La arquitectura desarrollada en este trabajo de investigación YAJE (Arquitectura ofrecida como servicio para la Computación y Visualización de Alto Rendimiento), se constituye como un esquema que permite a los usuarios de la comunidad científica acceder a recursos de cómputo y su correspondiente visualización de resultados, sin la necesidad de adquirir hardware de altas prestaciones, disminuyendo de esta forma gastos asociados a adquisición de equipos y de tarjetas gráficas.
- El uso de escritorios virtuales con GPU dedicada permite la personalización de ambientes de visualización de los investigadores, permitiendo que distintos tipos de sistemas operativos y aplicaciones puedan ser instalados sin necesidad de realizar configuraciones e instalaciones en el cluster de procesamiento. Esta característica amplía la demanda de usuarios que pueden usar la arquitectura.
- La virtualización de GPU permite optimizar los dispositivos de procesamiento gráfico, balanceando los recursos de cada escritorio virtual dependiendo de las necesidades de visualización de cada usuario.
- Durante la realización de pruebas de interacción se obtiene que el ancho de banda requerido debe ser superior a 8 Mbps, esto para garantizar una buena experiencia de usuario.
- Con la realización de este trabajo de investigación se logro proponer lineamientos para implementar una arquitectura para el procesamiento, visualización e interacción en aplicaciones científicas, ofrecida como servicio a la comunidad. Estos lineamientos se basan en cuatro pasos fundamentales: Definición del alcance, Planificación, Diagramación e Implementación de la arquitectura.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Infraestructura sc3. [Online]. Available: <http://www.sc3.uis.edu.co/servicios/hardware/>
- [2] Nvidia tesla. [Online]. Available: <http://www.nvidia.com/object/tesla-supercomputing-solutions.html>
- [3] J. Allard, C. Ménier, E. Boyer, and B. Raffin, “Running large vr applications on a pc cluster: the flowvr experience,” in *9th international Workshop on Immersive Projection Technology*. Eurographics Association, 2005, pp. 59–68.
- [4] (2016) Vmware. [Online]. Available: <http://www.vmware.com/>
- [5] (2016) Xenserver. [Online]. Available: <http://xenserver.org/>
- [6] (2016) Kvm. [Online]. Available: http://www.linux-kvm.org/page/Main_Page
- [7] (2016) Qemu. [Online]. Available: http://wiki.qemu.org/Main_Page
- [8] S. Iserte, A. Castelló, R. Mayo, E. S. Quintana-Ortí, F. Silla, J. Duato, C. Reaño, and J. Prades, “Slurm support for remote gpu virtualization: Implementation and performance study,” in *Computer Architecture and High Performance Computing (SBAC-PAD), 2014 IEEE 26th International Symposium on*. IEEE, 2014, pp. 318–325.
- [9] J. Duato, A. J. Pena, F. Silla, J. C. Fernandez, R. Mayo, and E. S. Quintana-Orti, “Enabling cuda acceleration within virtual machines using rcuda,” in *2011 18th International Conference on High Performance Computing*. IEEE, 2011, pp. 1–10.
- [10] VMware. (2016) VMware horizon 6 3d engineering workloads reference architecture. [Online]. Available: <http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-horizon-6-view-virtual-desktop-3d-graphics-reference-architecture-whitepaper.pdf>
- [11] C. Barrios. (2014) Programación paralela y programación científica grid uis desde la e- ciencia y el cómputo avanzado. [Online]. Available: <http://grid.uis.edu.co/images/9/92/E-Ciencia.pdf>
- [12] B. Barney, “What is parallel computing,” *Introduction to Parallel Computing*, 2012.
- [13] D. P. Rodgers, “Improvements in multiprocessor system design,” in *ACM SIGARCH Computer Architecture News*, vol. 13, no. 3. IEEE Computer Society Press, 1985, pp. 225–231.

- [14] G. Wallace, O. J. Anshus, P. Bi, H. Chen, Y. Chen, D. Clark, P. Cook, A. Finkelstein, T. Funkhouser, A. Gupta *et al.*, “Tools and applications for large-scale display walls,” *Computer Graphics and Applications, IEEE*, vol. 25, no. 4, pp. 24–33, 2005.
- [15] (2016, 4) Visualization clusters using commodity hardware. DELL. [Online]. Available: http://i.dell.com/sites/content/business/solutions/whitepapers/en/Documents/hpc_vis_wall_011209.pdf
- [16] J. Kohlhammer, D. Keim, M. Pohl, G. Santucci, and G. Andrienko, “Solving problems with visual analytics,” *Procedia Computer Science*, vol. 7, pp. 117–120, 2011.
- [17] J. Jomier, S. Jourdain, U. Ayachit, and C. Marion, “Remote visualization of large datasets with midas and paraviewweb,” in *Proceedings of the 16th International Conference on 3D Web Technology*, ser. Web3D ’11. New York, NY, USA: ACM, 2011, pp. 147–150. [Online]. Available: <http://doi.acm.org/10.1145/2010425.2010450>
- [18] Q. Hou, C. Qiu, K. Mu, Q. Qi, and Y. Lu, “A cloud gaming system based on nvidia grid gpu,” in *Distributed Computing and Applications to Business, Engineering and Science (DCABES), 2014 13th International Symposium on.* IEEE, 2014, pp. 73–77.
- [19] NVIDIA. (2013) Nvidia grid k2 datasheet. [Online]. Available: <http://www.nvidia.com/content/cloud-computing/pdf/nvidia-grid-datasheet-k1-k2.pdf>
- [20] VMware. (2014) View architecture planning. [Online]. Available: <https://pubs.vmware.com/horizon-view-60/topic/com.vmware.ICbase/PDF/horizon-view-60-architecture-planning.pdf>
- [21] P. Cook, Y. Bai, S. Nedjati-Gilani, K. Seunarine, M. Hall, G. Parker, and D. Alexander, “Camino: open-source diffusion-mri reconstruction and processing,” in *14th scientific meeting of the international society for magnetic resonance in medicine*, vol. 2759. Seattle WA, USA, 2006.
- [22] E. Garyfallidis, M. Brett, B. Amirbekian, A. Rokem, S. Van Der Walt, M. Descoteaux, and I. Nimmo-Smith, “Dipy, a library for the analysis of diffusion mri data,” *Frontiers in neuroinformatics*, vol. 8, p. 8, 2014.
- [23] L. L. N. Laboratory. (2016) Visit. [Online]. Available: <https://wci.llnl.gov/simulation/computer-codes/visit>
- [24] (2016) Top. [Online]. Available: <http://man7.org/linux/man-pages/man1/top.1.html>
- [25] E. Kuehn, M. Fischer, M. Giffels, C. Jung, and A. Petzold, “Analyzing data flows of wlcg jobs at batch job level,” in *Journal of Physics: Conference Series*, vol. 608, no. 1. IOP Publishing, 2015, p. 012017.
- [26] D. A. Angulo, C. Schneider, J. H. Oliver, N. Charpak, and J. T. Hernandez, “A multi-facetted visual analytics tool for exploratory analysis of human brain and

- function datasets,” *Frontiers in Neuroinformatics*, vol. 10, p. 36, 2016. [Online]. Available: <http://journal.frontiersin.org/article/10.3389/fninf.2016.00036>
- [27] NifTi. (2016) Archivos en formato nifti. [Online]. Available: <https://nifti.nih.gov/nifti-1>
- [28] Camino. (2016) Camino track. [Online]. Available: <http://camino.cs.ucl.ac.uk/index.php?n=Man.Track>
- [29] A. Cook, Zhang and J. Gee. (2008) Probabilistic diffusion-tensor fiber tractography in a bayesian framework with an atlas prior. [Online]. Available: <http://cds.ismrm.org/ismrm-2008/files/00426.pdf>
- [30] E. Garyfallidis, M. Brett, M. M. Correia, G. B. Williams, and I. Nimmo-Smith, “Quickbundles, a method for tractography simplification,” *Frontiers in neuroscience*, vol. 6, p. 175, 2012.

BIBLIOGRAFÍA

- A. Cook, Zhang and J. Gee. (2008) Probabilistic diffusion-tensor fiber tractography in a bayesian framework with an atlas prior.
[Online]. Disponible:<http://cds.ismrm.org/ismrm-2008/files/00426.pdf>
- B. Barney, 'What is parallel computing,' Introduction to Parallel Computing, 2012.
- C. Barrios. (2014) Programación paralela y programación científica griduis desde la ciencia y el cómputo avanzado.
[Online]. Disponible: <http://grid.uis.edu.co/images/9/92/E-Ciencia.pdf>
- Camino. (2016) Camino track. [Online].
Disponible: <http://camino.cs.ucl.ac.uk/index.php?n=Man.Track>
- D. A. Angulo, C. Schneider, J. H. Oliver, N. Charpak, and J. T. Hernandez, 'A multi-facetted visual analytics tool for exploratory analysis of human brain and 70 function datasets,' *Frontiers in Neuroinformatics*, vol. 10, p. 36, 2016. [Online]. Disponible: <http://journal.frontiersin.org/article/10.3389/fninf.2016.00036>
- D. P. Rodgers, 'Improvements in multiprocessor system design,' in *ACM SIGARCH Computer Architecture News*, vol. 13, no. 3. IEEE Computer Society Press, 1985, pp. 225–231.69
- E. Garyfallidis, M. Brett, B. Amirbekian, A. Rokem, S. Van Der Walt, M. Descoteaux, and I. Nimmo-Smith, 'Dipy, a library for the analysis of diffusion mri data,' *Frontiers in neuroinformatics*, vol. 8, p. 8, 2014.
- E. Garyfallidis, M. Brett, M. M. Correia, G. B. Williams, and I. Nimmo-Smith, 'Quick-bundles, a method for tractography simplification,' *Frontiers in neuroscience*, vol. 6, p. 175, 2012.
- E. Kuehn, M. Fischer, M. Giffels, C. Jung, and A. Petzold, 'Analyzing data flows of wlcg jobs at batch job level,' in *Journal of Physics: Conference Series*, vol. 608, no. 1. IOP Publishing, 2015, p. 012017.
- G. Wallace, O. J. Anshus, P. Bi, H. Chen, Y. Chen, D. Clark, P. Cook, A. Finkelstein, T. Funkhouser, A. Gupta et al., 'Tools and applications for large-scale display walls,' *Computer Graphics and Applications, IEEE*, vol. 25, no. 4, pp. 24–33, 2005.
- Infraestructura sc3. [Online]. Disponible: <http://www.sc3.uis.edu.co/servicios/hardware>
- J. Allard, C. Ménier, E. Boyer, and B. Raffin, 'Running large vr applications on a pc cluster: the flowvr experience,' in *9th international Workshop on Immersive Projection Technology*. Eurographics Association, 2005, pp. 59–68.

J. Duato, A. J. Pena, F. Silla, J. C. Fernandez, R. Mayo, and E. S. Quintana-Orti, 'Enabling cuda acceleration within virtual machines using rcuda,' in 2011 18th International Conference on High Performance Computing. IEEE, 2011, pp. 1–10.

J. Jomier, S. Jourdain, U. Ayachit, and C. Marion, 'Remote visualization of large datasets with midas and paraviewweb,' in Proceedings of the 16th International Conference on 3D Web Technology, ser. Web3D '11. New York, NY, USA: ACM, 2011, pp. 147–150. [Online]. Disponible: <http://doi.acm.org/10.1145/2010425.2010450>

J. Kohlhammer, D. Keim, M. Pohl, G. Santucci, and G. Andrienko, 'Solving problems with visual analytics,' *Procedia Computer Science*, vol. 7, pp. 117–120, 2011.

L. L. N. Laboratory. (2016) Visit.

[Online]. Disponible: <https://wci.llnl.gov/simulation/computer-codes/visit> (2016)

Top. [Online]. Disponible: <http://man7.org/linux/man-pages/man1/top.1.html>

NifTi. (2016) Archivos en formato nifti.

[Online]. Disponible: <https://nifti.nimh.nih.gov/nifti-1>

P. Cook, Y. Bai, S. Nedjati-Gilani, K. Seunarine, M. Hall, G. Parker, and D. Alexander, 'Camino: open-source diffusion-mri reconstruction and processing,' in 14th scientific meeting of the international society for magnetic resonance in medicine, vol. 2759. Seattle WA, USA, 2006.

Q. Hou, C. Qiu, K. Mu, Q. Qi, and Y. Lu, 'A cloud gaming system based on nvidia grid gpu,' in Distributed Computing and Applications to Business, Engineering and Science (DCABES), 2014 13th International Symposium on. IEEE, 2014, pp. 73–77.

S. Iserte, A. Castelló, R. Mayo, E. S. Quintana-Ortí, F. Silla, J. Duato, C. Reaño, and J. Prades, 'Slurm support for remote gpu virtualization: Implementation and performance study,' in Computer Architecture and High Performance Computing (SBAC-PAD), 2014 IEEE 26th International Symposium on. IEEE, 2014, pp. 318–325.

Vmware. [Online]. Disponible: <http://www.vmware.com/>

Xenserver. [Online]. Disponible: <http://xenserver.org/>

ANEXOS

A.Anexo A: Generación de imágenes de tractografía con Camino

- Grupo de datos inicial: Para generar las imágenes de tractografía se inicia el procesamiento con imágenes de resonancia magnética. Estas imágenes obtenidas mediante la técnica de difusión por resonancia magnética (DWI)¹, están representadas en archivos con formato NifTi[27].

Adicional a la imagen DWI el escáner de resonancia magnética genera la dirección de los vectores, que es la dirección que recorre un tracto. Por último se utiliza una máscara binaria de imagen DWI para separar el cerebro del fondo. La tabla A.1 presenta un ejemplo de los datos iniciales.

Archivo	Descripción
4Ddwi_b1000.nii.gz	RAW DWI en formato NifTI-1
grad_dirs.txt	Lista de direcciones de los gradientes
brain_mask.nii.gz	Máscara binaria que separa el cerebro del fondo

Tabla A.1: Ejemplo grupo de datos iniciales. Fuente: Autor.

- Generar archivo de parámetros: Este archivo describe los parámetros de la imagen que se utilizan para el procesamiento de la imagen. El archivo es generado a partir del listado de los gradientes. Para ello se utiliza una función de Camino llamada pointset2scheme.

```
$ pointset2scheme -inputfile grad_dirs.txt - bvalue 1E9  
  ↪ -outputfile 4Ddwi_b1000_bvector.scheme
```

- Convertir los datos de origen: En este paso se debe convertir el archivo de datos DWI generado por el scanner en un formato de voxeles.

```
$ image2voxel -4dimage 4Ddwi_b1000.nii.gz -outputfile  
  ↪ dwi.Bfloat
```

- Ajustar el tensor de difusión: Usando modelfit se ajusta el tensor de difusión a un conjunto de medidas para generar el archivo con la orientación de los tensores.

```
$ modelfit -inputfile dwi.Bfloat -schemefile 4  
  ↪ Ddwi_b1000_bvector.scheme -model ldt -bgmask  
  ↪ brain_mask.nii.gz -outputfile dt.Bdouble
```

¹Diffusion Weighted Imaging

Ahora se debe calcular los mapas anisotropía y difusividad media.

```
$ for PROG in fa md; do
    cat dt.Bdouble | ${PROG} | voxel2image -outputroot $
    ↪ {PROG} -header 4Ddwi_b1000.nii.gz
done
```

Finalmente se puede visualizar la orientación de los tensores en cada uno de las vistas de la imagen NifTI.

```
$ cat dt.Bdouble | dteig > dteig.Bdouble
$ pdview -inputfile dteig.Bdouble -scalarfile fa.nii.gz
```

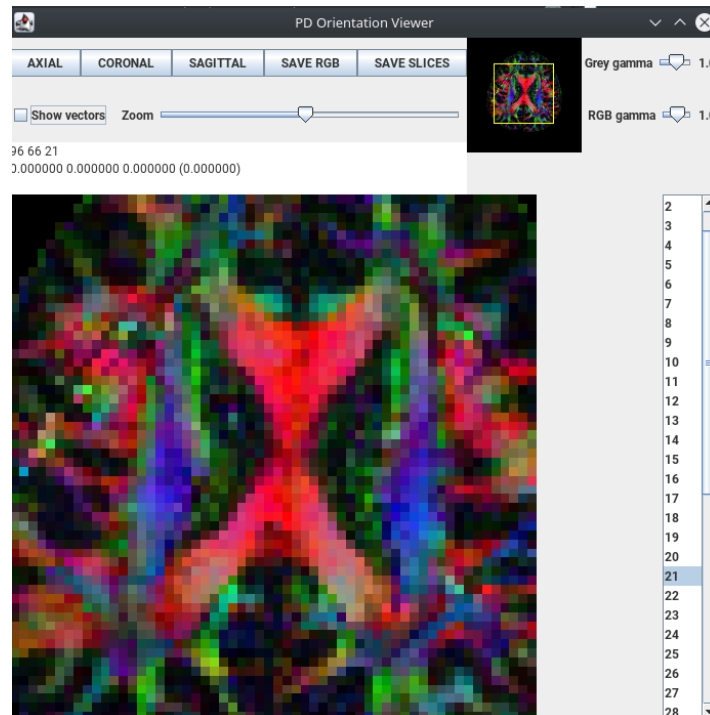


Figura A.1: Anisotropía fraccional sin vectores de orientación. Fuente: Autor.

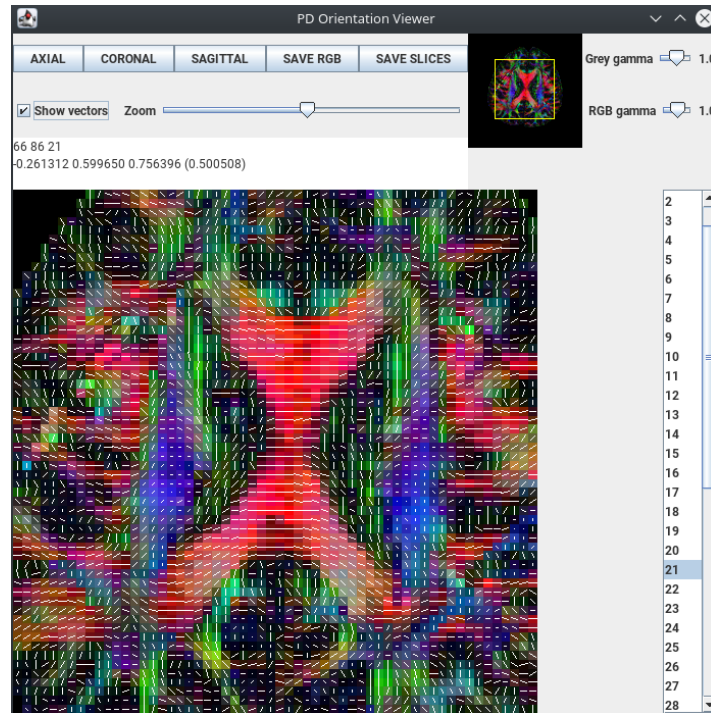


Figura A.2: Anisotropía fraccional con vectores de orientación. Fuente: Autor.

Estos gráficos presentan una visualización bi-dimensional de las imágenes de tensor de difusión junto con la orientación de los tensores. Ahora se debe generar la visualización 3D de esta imagen.

A partir de este paso se puede continuar con el calculo de tractos en un área específica o de todo el cerebro, la cual da una visión global al investigador.

- Generar visualización 3D de todo el cerebro: Finalmente se genera la visualización 3D de la imagen de tractografías de todo el cerebro. Para esto se trabaja con librerías de visualización científica como VTK ² cuyo formato de archivo puede ser visualizado en VisIt, Paraview o software especializado para imágenes de tractografías, en este caso se usa la librería para visualización científica de imágenes medicas (SMILI) ³. Para generar la imagen de tractografías se debe establecer una semilla (ROI ⁴), que calcula los tractos que están en el área donde la semilla se encuentra. En este caso se toma la imagen de todo el cerebro como la semilla, para obtener todos los tractos.

```
$ track -inputmodel dt -seedfile 4Ddwi_b1000.nii.gz -
  ➔ anisthresh 0.5 -curvethresh 60 -inputfile dt.
  ➔ Bdouble > full.Bfloat
```

²Visualization Toolkit

³Simple Medical Imaging Library Interface

⁴Region of interest/Región de interés

```
$ vtkstreamlines -colourorient < full.Bfloat >  
↪ full_brain.vtk
```

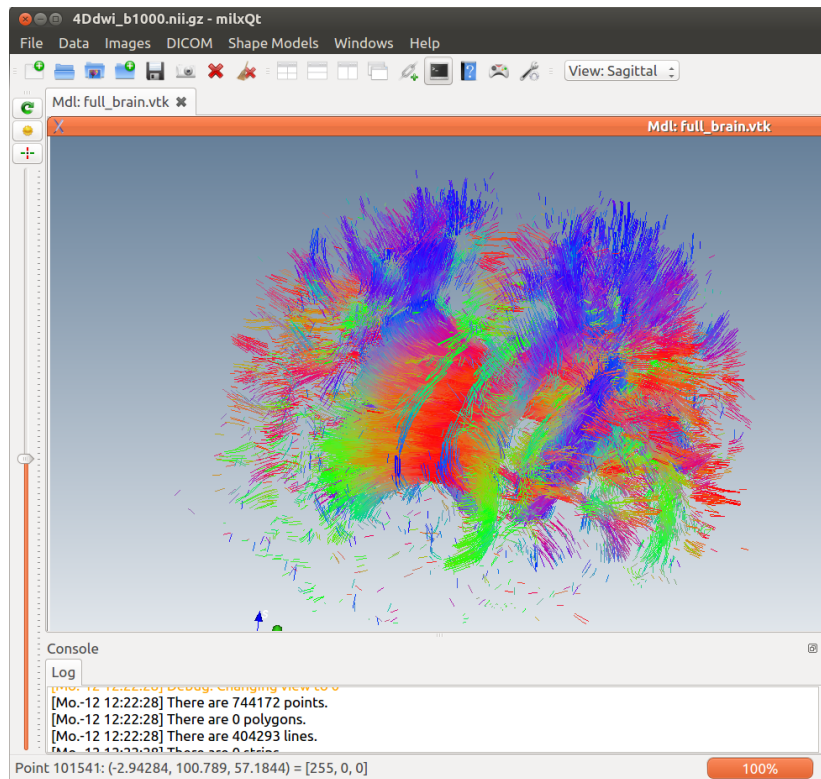


Figura A.3: Imagen de tractografías del cerebro. Fuente: Autor.

Los pasos mencionados anteriormente, fueron paso a paso generando la imagen de tractografías de todo el cerebro. A continuación se genera la imagen de tractografía en una región de interés.

El primer paso es establecer una región de interés en la que se van a buscar tractos. En este paso el investigador debe interactuar con la imagen, seleccionando la semilla o ROI ⁵. La región de interés es seleccionada de la imagen MRI.

⁵Region of interest

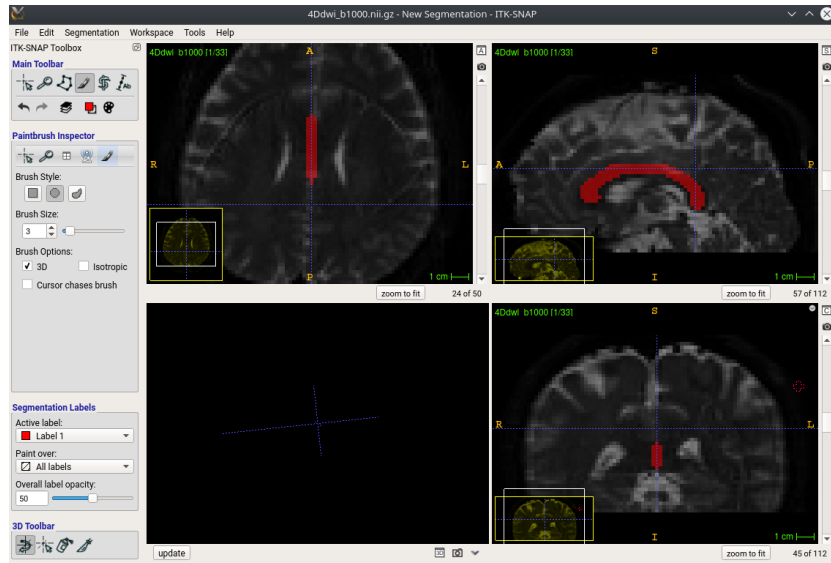


Figura A.4: Región de interés seleccionada para la búsqueda de tractos. Fuente: Autor.

En la imagen A.4 se selecciona el cuerpo calloso, que es la estructura que une los dos hemisferios del cerebro. Al realizar el calculo de fibras con esta región de interés se espera una gran cantidad de fibras o tractos.

- Calculo de fibras utilizando un método probabilista para una región de interés [28]: este método probabilístico es una implementación del método de Bayesiano propuesto por Friman et al [29].

```
track -inputfile dwi.Bfloat -inputmodel bayesdirac -
  ➔ schemefile 4Ddwi_b1000_bvector.scheme -iterations 50
  ➔ -seedfile roi.nii > bayesianTractsAll.Bfloat
```

```
dtlutgen -schemefile 4Ddwi_b1000_bvector.scheme -snr 20 >
  ➔ picoTableAll.dat
```

```
picopdfs -inputmodel dt -luts picoTableAll.dat < dt.
  ➔ Bdouble > pdfsAll.Bdouble
```

```
track -inputmodel pico -seedfile roi3.nii -iterations 50 <
  ➔ pdfsAll.Bdouble > picoTractsAll.Bfloat
```

```
vtkstreamlines -colourorient < picoTractsAll.Bfloat >
  ➔ picoCCsmall.vtk
```

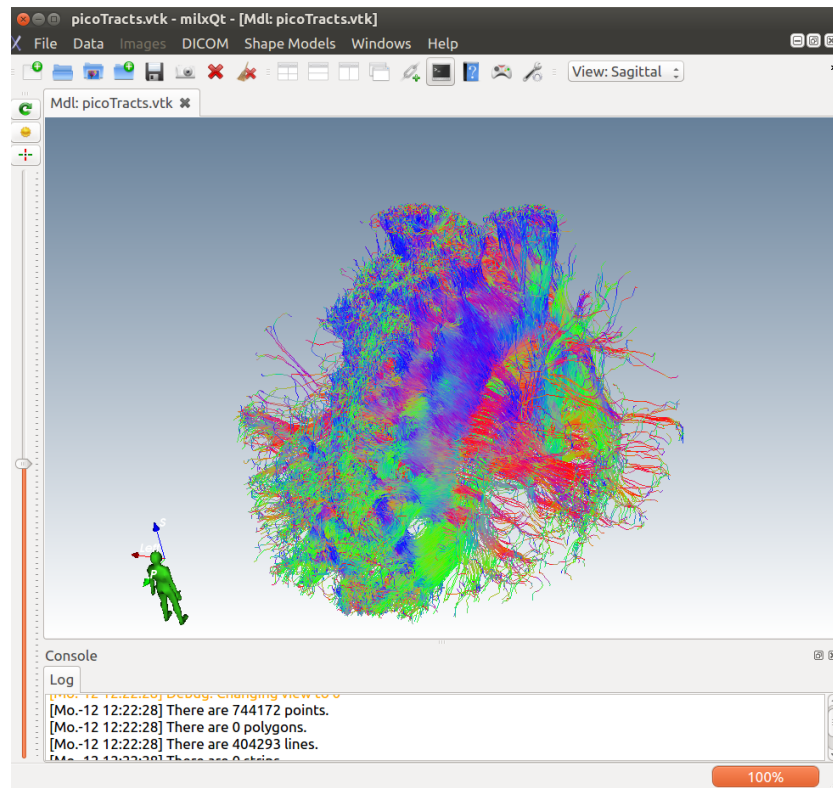


Figura A.5: Tractos encontrados en la región de interés usando método probabilístico.
Fuente: Autor

HPC for Remote Visualization and Interaction with Scientific Applications



D. Riaño Nuñez¹, C.J. Barrios Hernández¹, J. T. Hernández Peñañoza²

1. High Performance and Scientific Computing Center (SC3-UIS) www.sc3.uis.edu.co
2. IMAGINE (Unia)ndes imagine.uniaandes.edu.co

Universidad Industrial de Santander www.uis.edu.co

Bucaramanga, Colombia

Abstract

Most of the scientific applications required visualization tools to evaluate their results. Also the scientific data are increasing exponentially, therefore the computational resources and visualization hardware must be more complex and expensive. NVIDIA GRID(TM) [1] is one of the latest remote graphics technology solution, designed for cloud gaming, allows hardware virtualization and improves the delay problems with remote applications. This technology can be used to develop scientific applications, integrating high performance computing and taking advantage of remote graphics.

In this poster we present a work in progress, integrating remote visualization, interaction and high performance computing for the development of scientific applications, and also taking advantage of ultra high resolution display walls as a collaborative environment.

Background and motivation

Graphics as a Service (GaaS) is a new trend in computer science. This approach removes the need to purchase expensive GPU for rendering tasks. Working as a service, end-users can access to remote desktop environments with high-end configurations for a low cost. However, bandwidth is one of the common issues in cloud gaming, affecting the gaming experience with delay. GaaS requires high-end PC, this requirement is also for scientific applications, making this architecture ideal for researchers with high performance visualization needs. Therefore this visualization is the result of a long research process that generated a very large amount of data; the sheer size of data generated made high performance computing a necessity to obtain results.

When developing HPC applications visualization is often considered an independent step in the process, hence creating an opportunity to take advantage of this GaaS technology including HPC architectures like NVIDIA Tesla(TM) GPU.

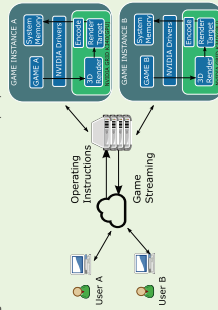


Fig 1. Architecture diagram of cloud gaming framework [2]



NVIDIA(r) GRID

NVIDIA GRID(TM) is a technology for desktop virtualization and video game streaming, that offers remote graphics processing using GPU virtualization, remote processing and session management libraries. Furthermore it allows access to GPU for multiple users running their applications simultaneously with high speed. NVIDIA GRID(TM) SDK provides two component software APIs [2]:

- NVBC: captures the entire desktop, including windows applications.
- NVFR: captures from a specific render target.

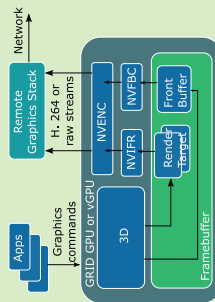


Fig 2. NVIDIA GRID(TM) Remote Graphics SDK [3]

Related Work

GGPU computing is part of high performance and scientific computing, but GGPU platforms are not available in all research sites. We proposed visualization as a service for scientific computing using GUANE, a hybrid supercomputing platform based on NVIDIA TESLA(TM) GPU. [4]

Software as Paraview [5] which provides a scientific visualization tool, was developed to analyse extremely large datasets. It runs in various modes of operation: Client-Server Mode, Distributed Server Mode and Tiled-Display Mode.

SAGE2 [6] is a graphics middleware that enables real-time streaming of ultra-high resolution images and supports collaboration.

HPC Architecture SC3-UIS

UIS High performance and scientific computing center (SC3-UIS) provides the fastest supercomputer in Colombia (GUANE) to researchers in many fields of science.

Our infrastructure has 16 computing nodes with 2 Xeon processors, 8 NVIDIA Tesla(TM) Fermi GPUs, 104 GB of RAM connecting with an Infiniband network. The NVIDIA GRID(TM) Server is a HPMV1350R with 6 Core Xeon processors, 16 GB of RAM and NVIDIA GRID(TM) K2. The NVIDIA GRID(TM) K2 has 2 NVIDIA Kepler(TM) GPUs with 8GB of DDR5 memory, 1536 cores per GPU and allows multiple user connections.

Further Work

Owing to the quantity of information that can be processed and visualized, the experts and developers must acquire high-end hardware and perform complex setup processes. Because of those processes, often cumbersome and time-consuming, the research itself gets less effective work time from the experts and developers. Besides, once the setup of the development environment is finished, bandwidth limitations are found.

There are similarities between hardware requirements for cloud gaming platforms and remote scientific application visualization. In this work we will propose a workflow for the development of applications that require HPC and visualization, using NVIDIA GRID(TM) to decrease delay that can affect the user experience while interacting with a remote scientific application. In addition, the proposed workflow will aim to eliminate the necessity of acquiring high-cost hardware for the visualization of scientific application output.

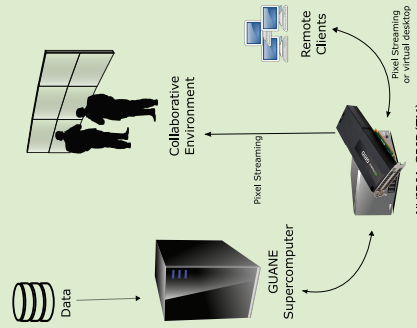


Fig 3. Proposed connection model of HPC-Visualization

References

1. NVIDIA GRID. Website: <http://www.nvidia.com/object/grid.html>
2. Zhou, Q., Chai, X., Gu, X. & Li, Y. (2014). A Cloud Gaming System Based on NVIDIA GRID GPU. 2014 13th International Symposium on Distributed Computing and Applications for Security and Privacy (DCASAP), 1-7. <http://ieeexplore.ieee.org/xpl/abstract/stamp.jsp?xap=1>
3. NVIDIA GRID. Website: <http://www.nvidia.com/object/grid.html>
4. Visualization of Scientific Data. Website: <http://www.visualizationofscientificdata.com/>
5. Shen, L., Wang, L., Johnson, A., Gahleitner, L. (2009). Remote visualization of large scientific data for ultra-high resolution display environments. Proceedings of the 2009 Workshop on High-Resolution Visualization (InfoVis), 1-4. <http://www.visualizationofscientificdata.com/>
6. Marmorek, T., Kuroki, T., Nakamura, A., Bhattacharya, S., Marmorek, L., Singh, J. (2013). SAGE2: A New Approach for Data Visualization. Proceedings of the 2013 ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), 1-7. <http://www.visualizationofscientificdata.com/>