

**ANALISIS Y DISEÑO DE UNA ESTRATEGIA DE INTERACCIÓN ENTRE
RECURSOS DISTRIBUIDOS DE UNA INFRAESTRUCTURA DE CÓMPUTO EN
REDES HETEROGENEAS.**

AUTORES

**MIREYA CAROLINA MANTILLA SERRANO
SERGIO ANDRES OROSTEGUI PRADA**

ESCUELA DE INGENERIA DE SISTEMA E INFORMATICA

UNIVERSIDAD INDUSTRIAL DE SANTANDER

BUCARAMANGA

2011

**ANALISIS Y DISEÑO DE UNA ESTRATEGIA DE INTERACCIÓN ENTRE
RECURSOS DISTRIBUIDOS DE UNA INFRAESTRUCTURA DE CÓMPUTO EN
REDES HETEROGENEAS.**

AUTORES

**MIREYA CAROLINA MANTILLA SERRANO
SERGIO ANDRES OROSTEGUI PRADA**

DIRECTOR

ING. JUAN CARLOS ESCOBAR RAMIREZ

CODIRECTOR

Ph.D CARLOS JAIME BARRIOS HERNANDEZ

ASESORES

**ING. ANTONIO JOSE LOBO FIGUEROA
ING. CRISTIAN CAMILO RUIZ SANABRIA**

ESCUELA DE INGENERIA DE SISTEMA E INFORMATICA

UNIVERSIDAD INDUSTRIAL DE SANTANDER

BUCARAMANGA

2011

DEDICATORIA

A Dios por permitirme culminar con esta meta, por darme la salud y las ganas para darla toda.

A mi mamá Irene Serrano Parra por todo su apoyo, su comprensión, su fuerza para que saliera adelante y por darme la valentía para cumplir con esta meta.

A mis amigos de siempre, mis compañeros de carrera y cada uno de los que hacen parte del grupo de Supercomputación y calculo científico de la UIS; por su compañía, su apoyo y su ayuda para lograr que el poder cumplir esta meta fuera algo muy agradable.

Mireya C. Mantilla Serrano

A mi mamá Gloria Prada por el apoyo incondicional y la confianza que me brindo a lo largo de la carrera

A grupo de Supercomputación y Calculo Científico de la UIS por guiarnos a través de todo este proceso y permitirnos desarrollar esta tesis al lado de estas grandes personas que lo conforman.

A mis amigos por los consejos y apoyo que dieron en todo momento.

Sergio Andrés Orostegui Prada

AGRADECIMIENTOS

- Al **Ing. Juan Carlos Escobar Ramírez**, director del proyecto por su labor, su comprensión y por todo su apoyo
- Al **Ph.D. Carlos Jaime Barrios Hernández**, codirector del proyecto por toda su gestión, sus enseñanzas y por su tiempo para la realización del proyecto.
- Al **Ing. Antonio José Lobo Figueroa**, asesor del proyecto, por toda su apoyo, conocimientos y por toda su ayuda a lo largo de este proyecto.
- Al **Ing. Cristian Camilo Ruiz Sanabria**, asesor del proyecto, por todas sus enseñanzas y colaboración en el proyecto.
- Grupo de Supercomputación y Calculo Científico de la UIS, (**SC3-UIS**), por su colaboración y apoyo
- A la **Universidad Industrial de Santander y al CENTIC**, por proveer los recursos necesarios para el desarrollo del proyecto.
- Equipo MESCAL del Laboratorio de Informática de Grenoble, sitio de Montbonnot-Srt Martin, Francia, principalmente a los proyecto OAR+KaaTools y Computemode.

TABLA DE CONTENIDO

1. INTRODUCCION.....	16
2. GLOSARIO.....	18
3. OBJETIVOS.....	20
3.1 OBJETIVO GENERAL.....	20
3.2 OBJETIVOS ESPECÍFICOS.....	20
4. PLANTEAMIENTO DEL PROBLEMA.....	21
5. JUSTIFICACION.....	23
6. MARCO TEORICO.....	24
6.1 TOPOLOGIAS DE RED.....	24
6.1.1 TIPOS DE TOPOLOGIAS.....	25
6.2 COMPUTACION DISTRIBUIDA.....	26
6.2.1 CARACTERÍSTICAS DE LA COMPUTACION DISTRIBUIDA.....	28
6.3 ARQUITECTURAS DISTRIBUIDAS.....	29
6.3.1 COMPUTACIÓN VOLUNTARIA O DE CICLOS REDUNDANTES.....	29
6.3.2 CLUSTER COMPUTING.....	31
6.3.2.1 CLASIFICACION DE LOS CLUSTER.....	33
6.3.2.2 CLUSTER DEDICADO.....	35
6.3.2.3 CLUSTER LIGERO.....	35
6.3.3 GRID COMPUTING.....	36
6.3.3.1 GRID LIGERA.....	38
6.3.3.2 DESARROLLO HISTORICO DE GRID COMPUTING.....	38
6.3.3.3 CARACTERISTICAS DE GRID COMPUTING.....	39
6.3.3.4 VENTAJAS DE GRID COMPUTING.....	40
6.3.3.5 DESVENTAJAS DE GRID COMPUTING.....	41
6.3.3.6 APLICACIONES DE GRID COMPUTING.....	41
6.3.3.7 SEGURIDAD.....	42
6.3.3.8 ALGUNOS PROYECTOS DE GRID COMPUTING.....	42
6.4 COMPUTEMODE.....	44
6.4.1 DESCRIPCION COMPUTEMODE.....	47
6.5 OAR.....	49
6.5.1 CAPACIDADES DE OAR.....	51

6.5.2 VENTAJAS DE OAR.....	51
7. ESTADO DEL ARTE	53
8. ARQUITECTURA PROPUESTA.....	56
8.1 DISEÑO.....	57
8.1.1 DESCRIPCION DE CADA CLUSTER Y SUS NODOS.....	60
8.2 INTERACCION DEL USUARIO CON EL SISTEMA DE DESPLIEGUE DE APLICACIONES.....	64
9. IMPLEMENTACION Y RESULTADOS.....	66
9.1 IMPLEMENTACION.....	66
9.1.1 MODIFICACIONES A COMPUTEMODE.....	66
9.1.2 CENTRALIZACION DE LOS USUARIOS Y DE LA BASE DE DATOS.....	69
9.1.3 MODIFICACION DE LIBRERIAS.....	73
9.2 PRUEBAS Y RESULTADOS.....	76
9.2.1 POVRAY.....	76
9.2.2 IPERF.....	79
9.2.3 USO DE LA PLATAFORMA.....	82
10. LIMITACIONES DEL PROYECTO.....	84
11. CONCLUSIONES.....	85
12. RECOMENDACIONES.....	87
13. ANEXOS.....	88
14. BIBLIOGRAFIA.....	97
14.1 ARTICULOS Y PROYECTOS.....	97
14.2 ENLACES.....	98

LISTA DE TABLAS

TABLA 1 DESCRIPCIÓN DE LOS CLÚSTER USADOS.....	59
TABLA 2 DESCRIPCIÓN DEL CLÚSTER LEAC.....	61
TABLA 3 DESCRIPCIÓN DEL CLÚSTER CMCENTIC.....	62
TABLA 4 DESCRIPCIÓN DEL CLÚSTER CMJAVS.....	62
TABLA 5 DESCRIPCIÓN DEL CLÚSTER GIIB.....	63
TABLA 6 MEDIA E INTERVALO DE VARIABILIDAD DE LA PRUEBA POVRAY.....	79
TABLA 7 MEDIA E INTERVALO DE VARIABILIDAD DE LA PRUEBA IPERF.....	80

LISTA DE FIGURAS

ILUSTRACIÓN 1 TOPOLOGÍAS DE RED.....	26
ILUSTRACIÓN 2 ARQUITECTURA COMPUTEMODE.....	45
ILUSTRACIÓN 3 DISEÑO DE LA ARQUITECTURA PROPUESTA.....	58
ILUSTRACIÓN 4 DIAGRAMA DE CASOS DE USO.....	64
ILUSTRACIÓN 5 DISEÑO DE LOS 4 CLÚSTER UBICADOS EN EL CAMPUS UNIVERSITARIO.....	68
ILUSTRACIÓN 6 REPRESENTACIÓN DE LOS PROTOCOLOS NFS Y LDAP.....	69
ILUSTRACIÓN 7 REPRESENTACIÓN DEL SISTEMA DE MONITOREO (MONIKA) POR NODOS.....	71
ILUSTRACIÓN 8 REPRESENTACIÓN DEL SISTEMA DE MONITOREO (MONIKA) POR CLÚSTER.....	72
ILUSTRACIÓN 9 IMAGEN OBTENIDA POR MEDIO DE POVRAY, PEBBLES.....	77
ILUSTRACIÓN 10 DESCRIPCIÓN DEL RENDERIZADO DE LA IMAGEN PEBBLES.....	78
ILUSTRACIÓN 11 REPRESENTACIÓN DE LA HERRAMIENTA IPERF.....	80
ILUSTRACIÓN 12 RESULTADOS OBTENIDOS CON LA PRUEBA IPERF.....	81
ILUSTRACIÓN 13 REPRESENTACIÓN DE LA RESERVA DE NODOS.....	82
ILUSTRACION 14 REPRESENTACIÓN DE UNA GRAN CANTIDAD DE NODOS RESERVADOS.....	83

LISTA DE ANEXOS

A. SHORT PAPER PRESENTADO EN EL CLCAR 2010.....	88
B. POSTER PRESENTADO EN EL CLCAR 2010.....	95
C. ASESORES LIGADOS AL PROYECTO.....	96

RESUMEN

TITULO: ANALISIS Y DISEÑO DE UNA ESTRATEGIA DE INTERACCIÓN ENTRE RECURSOS DISTRIBUIDOS DE UNA INFRAESTRUCTURA DE CÓMPUTO EN REDES HETEROGENEAS¹

AUTORES: MIREYA CAROLINA MANTILLA SERRANO
SERGIO ANDRES OROSTEGUI PRADA²

PALABRAS CLAVE: Clúster, Computación de alto rendimiento (HPC), computación distribuida, ComputeMode, recursos ociosos.

DESCRIPCION: El buen uso de los recursos para la realización de cálculos que necesiten gran capacidad de procesamiento y tiempo de cómputo, es cada vez más importante en los campus Universitarios. El contar con maquinas que permitan crear diferentes clústers y que a su vez estos clústers puedan interactuar con otros dentro del campus universitario, es fundamental para el aprovechamiento de los diferentes recursos que no son usados en tiempos no laborales (horarios nocturnos, vacaciones y fines de semana). Además, tener la posibilidad de contar cada vez más con diferentes recursos que puedan ayudar en el campo de la computación distribuida, resulta de gran importancia para darle un valor agregado a la Universidad que lo está desarrollando.

Actualmente se puede contar con diferentes herramientas que puedan ayudar a la creación de los clúster. Entre estas, una de la más completa y sencillas es la herramienta ComputeMode. El uso de esta herramienta permite crear un clúster ligero, agregando recursos de cómputo no usados, pero se cuenta con el inconveniente al momento de interactuar con otros nodos que no estén en la misma jerarquía de red, es decir, imposibilitando el uso de otros recursos que no estén dentro de la misma subred del servidor de cada clúster.

Es por ello que el presente trabajo pretende analizar como primera instancia la manera en la que se podría dar la interacción entre recursos distribuidos de una infraestructura de cómputo en redes heterogéneas y en base a este análisis poder diseñar y desarrollar una estrategia que pueda resolver la limitación antes mencionada y que además se aprovechen de la mejor manera los diferentes recursos con los que cuenta el campus Universitario.

¹ Trabajo de grado en la Modalidad de Investigación

² Facultad de ingenierías Físico-mecánicas. Escuela de ingeniería de sistemas e informática.
Director: Ing Juan Carlos Escobar Ramírez. Codirector: PhD: Carlos Jaime Barrios Hernández.
Asesores: Ing Antonio Jose Lobo, Figueroa, Ing Cristian Camilo Ruiz Sanabria

ABSTRACT

TITLE: ANALYSIS AND DESIGN OF AN INTERACTIVE STRATEGY BETWEEN DISTRIBUTED RESOURCES OF A COMPUTING INFRASTRUCTURE HETEROGENEOUS NETWORK³

AUTHORS: MIREYA CAROLINA MANTILLA SERRANO
SERGIO ANDRES OROSTEGUI PRADA⁴

KEYWORD: Cluster, ComputeMode, distributed computing, High Performance Computing (HPC), idle resources.

DESCRIPTION: Nowadays the efficient use of resources for research and scientific computing is a big issue in different universities. For this reason it is important to find and implement new tools to improve in this area, a possible way to accomplish this is through the use of non-dedicated cluster. The opportunistic computation is the ability to create cluster of different computational resources which usually are power off or idle in certain periods of time as: nights, holidays and weekends. Also it is important to increase the interest and use in this kind of distributed computing infrastructure.

At the moment, different tools can be found on the market that can help in setting up non-dedicated clusters, one of them is Computemode. Computemode is the most useful and simple free software. Computemode allows the user to create a HPC non-dedicated cluster taking advantage of unused resources, but it is limited to resources belonging to a network hierarchy making impossible the communication between resources in different networks hierarchies.

Therefore, in this engineer report, we present the design and developing of a strategic which makes possible the interaction among the distributed resources in a heterogeneity infrastructure in such a way that the university's resources are well-spent.

³Undergraduate Final Project, Research modality

⁴ Systems Engineering and Computer Science School.

Director: Ing: Juan Carlos Escobar Ramírez. Codirector: PhD. Carlos Jaime Barrios Hernández
Advisers: Ing Antonio Jose Lobo Figueroa, Ing: Cristian Camilo Ruiz Sanabria

1. INTRODUCCION

Cada vez más los diferentes proyectos de investigación que se llevan a cabo en grandes universidades del mundo, están requiriendo de mayores capacidades de cómputo para llevarlos a cabo. La necesidad de poder contar con infraestructuras computacionales que permitan realizar grandes cálculos, que cuenten con una buena capacidad de procesamiento y que permitan que el tiempo requerido para los cálculos no sea tan absurdo al momento de presentar los resultados; es lo que ha llevado últimamente a que los campus Universitarios se estén desarrollando y creando clúster que ayuden a minimizar estos inconvenientes. La implementación de los diferentes tipos de clúster que ayuden a realizar grandes cálculos ha posibilitado que se desarrollen proyectos dentro del campo de la computación distribuida.

Actualmente la Universidad Industrial de Santander cuenta con varios clúster para el procesamiento y desarrollo de estos proyectos. Entre estos, se tienen los clúster dedicados, los cuales están disponibles todo el tiempo para ser utilizados en la realización de cálculos y están también los clúster ligeros, que son aquellos que sólo se usan en horarios fuera de clases, fines de semana y vacaciones; en pocas palabras son clúster que emplean recursos que pasan la mayor parte del tiempo ociosamente. Sin embargo, y queriendo ser más ambiciosos en la posibilidad de mejorar la capacidad de computo que dan estos clúster, se desea que entre ellos exista una interacción que pueda permitir un mayor nivel de computación, de capacidad de procesamiento y que disminuya notoriamente el tiempo de calculo que se necesita para el desarrollo de las diferentes aplicaciones.

Es por esto que el presente trabajo tiene como fin posibilitar esta interacción, para así poder contar con una plataforma en donde exista una mayor cantidad de recursos al servicio de los diferentes usuarios dentro y fuera del campus Universitario. El desarrollo del trabajo realizado se encuentra distribuido de la siguiente manera: en la sección 2 esta la especificación referente al proyecto, en

la sección 3 encontramos el glosario para hacer más fácil el entendimiento de los diferentes términos usados a lo largo del documento, en la sección 4 se encuentran los objetivos de proyecto, en la sección 5 se encuentra el planteamiento del problema al cual está dirigido el desarrollo del proyecto, en la sección 6 se especifica la justificación del porque del proyecto, en la sección 7 se expone el marco teórico en donde se establecen temas como las topologías de red, la computación distribuida, arquitecturas distribuidas, Herramientas como ComputeMode y Oar, entre otros temas, en la sección 8 se expone el estado del arte, en las siguientes secciones se presenta la implementación del proyecto, los resultados obtenidos y finalmente las conclusiones a las cuales se llegaron.

2. GLOSARIO

CLÚSTER COMPUTACIONAL: conjunto o conglomerado de computadores que interconectados aprovechan el conjunto de sus características individuales para soportar el procesamiento en paralelo.

COMPUTEMODE: Herramienta software que permite crear clúster ligero por agregación de recursos en forma no intrusiva

DoS: (Denegación de servicio), es el ataque a un sistema de computadores o red, que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la conectividad de la red por el consumo del ancho de banda de la red de la víctima o sobrecarga de los recursos computacionales del sistema de la víctima

FAIL OVER: modo de operación de backup en el cual las funciones de un componente del sistema son asumidas por un segundo componente del sistema cuando el primero no se encuentra disponible debido a un fallo ó un tiempo de parada preestablecido.

GRID COMPUTACIONAL: Tecnología que permite el uso de plataformas de computo geográficamente distribuidas para fines comunes y específicos.

IPTABLES: es un sistema de firewall vinculado al kernel de Linux que se ha extendido enormemente a partir del kernel 2.4 de este sistema operativo.

NFS: Network File System o Sistema de archivos de red, es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales.

LDAP: (Lightweight Directory Access Protocol, Protocolo Ligero de Acceso a Directorios), es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

SSH: (Secure Shell, intérprete de órdenes segura), protocolo y programa que sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo un computador mediante un intérprete de comandos.

WALLTIME: Es el tiempo de reserva que es asignado a unos nodos para poder desplegarse una aplicación.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

- Analizar y proponer un diseño de una estrategia que garantice interconectividad y gestión de recursos en una infraestructura para el cómputo distribuido en redes heterogéneas.

3.2 OBJETIVOS ESPECÍFICOS

- Analizar y caracterizar diferentes tipos de interacción entre redes que soportan infraestructura de cálculo distribuido.
- Estudiar diferentes casos de interconectividad, basado en la infraestructura existente de cómputo distribuido.
- Diseñar una estrategia para interconectar y gestionar los recursos de una infraestructura para el cómputo distribuido en redes heterogéneas.
- Desarrollar un prototipo que incluya procesos de interconexión y gestión de recursos de una infraestructura de cómputo distribuido en redes heterogéneas.
- Evaluar el desempeño y la pertinencia del prototipo para validar la estrategia propuesta.

4. PLANTEAMIENTO DEL PROBLEMA

Actualmente en diferentes campus universitarios se cuentan con gran cantidad de equipos de cómputo interconectados por una red que son usados normalmente por los estudiantes, docentes y parte del personal administrativo de la institución. Estos equipos de cómputo cuentan con diversos sistemas operativos predefinidos y con políticas administrativas que han sido precisadas previamente por la universidad las cuales no permiten modificar el software o su estado, siendo así una limitante en el momento de hacer de estos recursos un uso diferente al que se le da. Estos equipos normalmente se encuentran distribuidos en aulas de clase para su uso cotidiano por los estudiantes, pero podrían utilizarse más allá; por ejemplo, en horarios nocturnos y fines de semana siendo posible la construcción de Clúster que permita hacer de estos una mejor herramienta para la solucionar diferentes problemas de cálculo distribuido.

Un clúster computacional (o simplemente clúster) es un tipo de sistema de procesamiento distribuido que consiste en un conjunto de computadores interconectados mediante un sistema de red de alta velocidad y un software que es el encargado de la distribución de los trabajos en cada computador, con el objetivo de agilizar procesos que requieran gran capacidad de cómputo y almacenamiento. Actualmente desempeñan un valor importante en el apoyo de actividades que van desde aplicaciones de súper cómputo, servidores web, comercio electrónico, base de datos de alto rendimiento hasta soluciones de problemas de las ciencias, las ingenierías entre otros. Esto debido a la posibilidad de realizar cálculo de alto rendimiento a un relativo bajo costo.

Entre los tipos de clúster, existen aquellos que se construyen por agregación de recursos de manera no dedicada llamados clúster ligeros; los cuales son implementados usando salas de prácticas de cómputo. Igualmente existe la posibilidad de integrar equipos a plataformas de cálculo distribuido – siguiendo las políticas y técnicas normales de un clúster de pc's- de manera no intrusiva y no

permanente. Es por esto que la creación de plataformas de cálculo distribuido son muy interesantes dentro de los campus universitarios, pues existe la posibilidad de usar equipos de cómputo en tiempos ociosos para sacar el mayor provecho y para las posibles soluciones a problemas que se presentan cuando hay una alta demanda de procesamiento, sin importar cuando termine.

Hoy en día, existen varias herramientas que permiten el uso de recursos eventualmente dentro de lo que se ha nombrado un “clúster”, entre esas se puede mencionar ComputeMode⁵. ComputeMode es una herramienta software que permite crear clúster por agregación de recursos en forma no intrusiva y es basado en una arquitectura maestro-esclavo, utilizando un servidor central como maestro. A pesar de las ventajas ofrecidas por ComputeMode, presenta una limitación a nivel de red que se debe a que no permite la interconectividad de equipos de cómputo ajenos a la subred del servidor, llevando a que se pierda importante cantidad de recursos que se encuentran distribuidos alrededor del campus universitario pero por fuera de la subred del servidor.

Expuesto esto, se hace necesaria la creación de una estrategia que permita interconectar y gestionar recursos de una infraestructura de red para el cómputo distribuido en diferentes redes logrando una capacidad de cómputo que supla las necesidades de cálculo de los investigadores.

⁵ ComputeMode Project: <http://computemode.imag.fr>

5. JUSTIFICACION

En una sociedad tecnológica como en la que nos encontramos actualmente en donde el manejo de la información y los diferentes recursos que se comparten juegan un papel muy importante, la gran cantidad de operaciones necesarias para la solución de problemas de cálculo y el tiempo que lleva cada una de estas operaciones, son factores que a lo largo del tiempo han tomado importancia para el desarrollo de la computación distribuida, como una nueva forma de lograr el mayor rendimiento para la solución de problemas que requieren gran complejidad de cómputo. La creación de Clústers ligeros da la posibilidad de resolver parte de estas necesidades pero cuando nos encontramos con diversidad en los recursos de cómputo y la red que los interconecta es heterogénea y de diferentes niveles se hace imprescindible que estos recursos se entiendan entre sí, como si estuvieran en un mismo nivel de red. Actualmente es posible crear una GRID computacional ligera usando recursos que se encuentren dentro de una misma jerarquía de red. Existen diversas plataformas diseñadas para crear GRID ligeras entre ellas ComputeMode.

Esta plataforma esta implementada dentro del campus universitario, con la limitación de que los recursos interconectados deben estar en la misma jerarquía de red. Para poder proponer una solución es necesario analizar y caracterizar diferentes tipos de interacción entre redes que soportan infraestructura de cálculo distribuido a través de GRID ligeras de tal manera que se expanda la capacidad de procesamiento actual en la universidad. Además de poder contar con la mayor disponibilidad en el uso de los diferentes recursos dentro de la plataforma, con el fin de que pueda ser empleada por una gran cantidad de usuarios.

6. MARCO TEORICO

6.1 TOPOLOGIAS DE RED

Para empezar a mencionar las diferentes topologías de red, es necesario decir que las redes de computadoras surgieron como una necesidad de interconectar los diferentes host de una organización para poder así compartir recursos y equipos específicos. Pero los diferentes componentes que van a formar una red se pueden interconectar o unir de diferentes formas, siendo la forma elegida un factor fundamental que va a determinar el rendimiento y la funcionalidad de la red. La disposición de los diferentes componentes de una red se conoce con el nombre de topología de la red. La topología idónea para una red concreta va a depender de diferentes factores, como el número de máquinas a interconectar, el tipo de acceso al medio físico que deseemos etc.

La topología de red se define como la cadena de comunicación usada por los nodos que conforman una red para comunicarse y la determina únicamente la configuración de las conexiones entre nodos. Cuando hablamos de topología de una red, hablamos de su configuración. Esta configuración recoge tres campos: físico, eléctrico y lógico

- **Campo Físico:** La topología física, que es la disposición real de las máquinas, dispositivos de red y cableado (los medios) en la red.
- **Campo Eléctrico:** El nivel físico y eléctrico se puede entender como la configuración del cableado entre máquinas o dispositivos de control o conmutación.
- **Campo Lógico:** La topología lógica, que es la forma en que las máquinas se comunican a través del medio físico. Los dos tipos más comunes de topologías lógicas son broadcast (Ethernet) y transmisión de tokens (Token Ring).

6.1.1 TIPOS DE TOPOLOGIAS

La distancia entre los nodos, las interconexiones físicas, las tasas de transmisión y los tipos de señales no pertenecen a la topología de la red, aunque pueden verse afectados por la misma, a lo largo de este proyecto serán necesarias algunas de ellas y por ello es de gran importancia nombrarlas:

1. **Red centralizada:** todos los nodos, menos uno, son periféricos y sólo pueden comunicarse a través del nodo central. La caída del nodo central priva del flujo a todos los demás nodos. En redes centralizadas se pueden nombrar la topología en estrella y la topología en árbol.
2. **Red descentralizada:** es aquella en donde aparece por interconexión los nodos centrales de varias redes centralizadas. Como resultado no existe un único nodo central sino un centro colectivo de conectores. La caída de uno de los nodos centralizadores, conlleva la desconexión de uno o más nodos del conjunto de la red mientras que la caída del clúster centralizador produciría necesariamente la ruptura o desaparición de la red. En una red descentralizada se puede mencionar la topología en malla, en la cual cada servidor tiene sus propias conexiones con todos los demás servidores.
3. **Red distribuida:** Es aquella en donde todos los nodos se conectan entre sí, sin que tengan que pasar necesariamente por uno o varios centros. Desaparece la división centro/periferia y por tanto el poder de filtro sobre la información que fluye por ella. La red es robusta ante la caída de nodos: ningún nodo al ser extraído genera la desconexión de otro. [24]

La siguiente grafica representa las tres topologías de red mencionadas anteriormente, por Paul Baran.

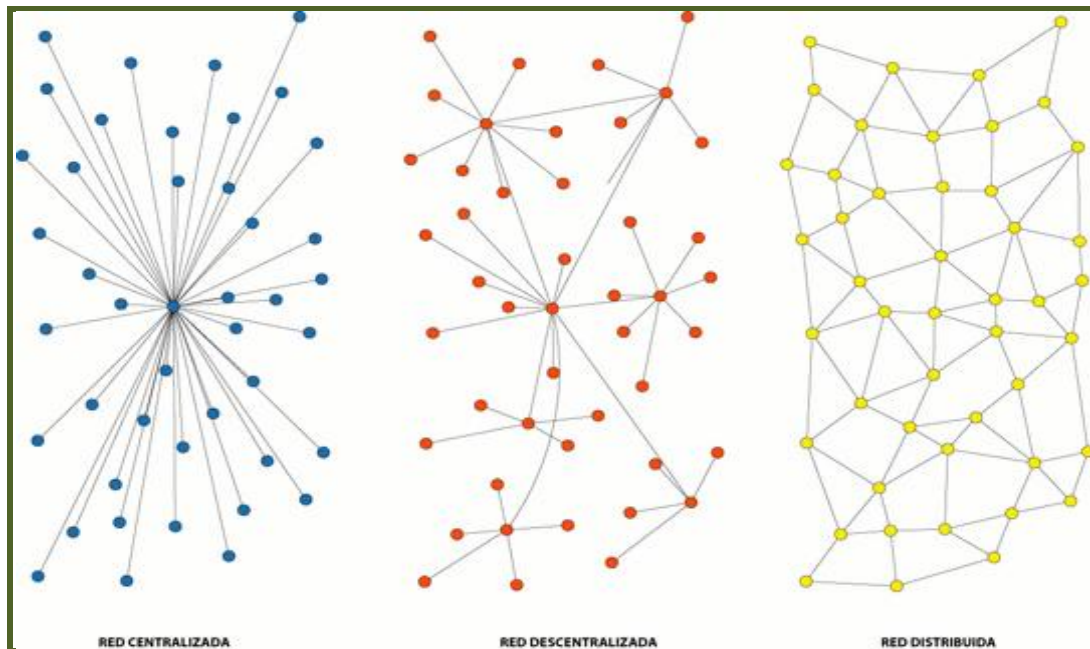


Ilustración 1: Topologías de red

Fuente: Imagen realizada por Rodrigo Araya Dujisin a partir de un grafo original de Paul Baran⁶

6.2 COMPUTACION DISTRIBUIDA

La computación distribuida es un nuevo modelo para resolver problemas de computación masiva utilizando un gran número de computadores organizados en una infraestructura distribuida.⁷ Estos recursos son compartidos con todos los otros computadores en el sistema. La potencia de procesamiento, la memoria y el almacenamiento de datos, son recursos de la comunidad donde los usuarios autorizados pueden entrar y realizar ciertas tareas. Una red distribuida puede ser tan simple como una colección de equipos similares funcionando con el mismo sistema operativo, o tan complejo como una red interconectada de sistemas, formada por cualquier plataforma informática.⁸

⁶ David de Ugarte, *El poder de las redes*, 2007, ISBN 978-84-611-8873-4

⁷ Marielisa Peralta Valarezo, COMPUTACION DISTRIBUIDA, Artículo, Universidad técnica particular de Loja. (Disponible en: <http://sig.utpl.edu.ec/download/data/computacion%20distribuida.PDF>)

⁸ Computación distribuida, Ordenadores y portátiles: <http://www.ordenadores-y-portatiles.com/computacion-distribuida.html>

Con esta tecnología, diferentes computadores dentro de una red comparten uno o más recursos. En uno de estos sistemas considerado ideal, todos los recursos son compartidos, convirtiendo una red de computadores en un potente supercomputador. Con la interfaz adecuada, acceder a uno de estos sistemas no es muy diferente a acceder a los recursos de una máquina local. Todo computador autorizado tendrá acceso a una potencia de procesamiento enorme, y una gran capacidad de almacenamiento. Normalmente, un computador solo puede operar dentro de los límites de sus propios recursos. Hay un límite máximo para que pueda completar una operación o cuanta información puede almacenar. La mayoría de los computadores son actualizables, lo cual significa que se puede añadir más potencia y capacidad a un solo equipo, pero eso sigue siendo un aumento incremental de rendimiento. [11]

Los sistemas de computación distribuida enlazan los recursos de red todos juntos, de una forma tal, que permite a un solo computador heredar la potencia del resto de computadores en el sistema. Para un solo usuario, esto le supone un gran avance en cualquier trabajo que esté haciendo, sea del tipo que sea. Es por esto que la computación distribuida ha sido diseñada para resolver problemas demasiado grandes para cualquier supercomputador y mainframe, mientras se mantiene la flexibilidad de trabajar en múltiples problemas más pequeños. Por lo tanto, la computación en grid es naturalmente un entorno multiusuario; por ello, las técnicas de autorización segura son esenciales antes de permitir que los recursos informáticos sean controlados por usuarios remotos.

Los sistemas distribuidos deben ser muy confiables, ya que si un componente del sistema se descompone otro componente debe de ser capaz de reemplazarlo, por otro lado el tamaño de un sistema distribuido puede ser muy variado, ya sean decenas de hosts (red de área local), centenas de hosts (red de área metropolitana), o miles o millones de hosts (Internet).

Un sistema distribuido se define entonces como una colección de computadoras separados físicamente y conectados entre sí por una red de comunicaciones distribuida; cada máquina posee sus componentes de hardware y software que el usuario percibe como un solo sistema (no necesita saber qué cosas están en qué máquinas). El usuario accede a los recursos remotos de la misma manera que accede a recursos locales.

A continuación se mencionaran algunas características que hacen parte de la computación distribuida y las cuales permiten que sea tan importante para el desarrollo científico. Entre ellas tenemos:

6.2.1 CARACTERÍSTICAS DE LA COMPUTACION DISTRIBUIDA

1. Para cada uno de los usuarios debe ser similar al trabajo en el Sistema Centralizado.
2. Seguridad interna en el sistema distribuido.
3. Se ejecuta en múltiples Computadores.
4. Tiene varias copias del mismo Sistema Operativo o de diferentes Sistemas Operativos que proveen los mismos servicios.
5. Entorno de trabajo cómodo
6. Dependiente de redes (LAN, MAN, WAN, etc.)
7. Compatibilidad entre los dispositivos conectados
8. Transparencia (El uso de múltiples procesadores y el acceso remoto debe de ser invisible).

La aplicación de estos conceptos y de las características con las que cuenta la computación distribuida busca resolver problemas existentes en la computación como:

Tiempo: La computación distribuida permite que investigaciones o procesos que requieren manejo y procesamiento de gran cantidad de información se haga de forma más rápida, lo que permite obtener resultados en menos tiempo.

Almacenamiento: Al reducir tiempo en el procesamiento de la información se incrementa la cantidad de datos que se obtienen, estos datos no necesariamente tiene que ser almacenados en una sola máquina, debido a que su tamaño obligaría a tener una gran infraestructura de almacenamiento aumentando los costos y tiempos de acceso.

Dinero: Debido a los altos costos, no todas las organizaciones tienen la capacidad de adquirir maquinas robustas con configuraciones de alto desempeño que les permitan asumir proyectos que requieran este tipo de computadoras. Mediante la computación distribuida las empresas tienen acceso a muchos computadores que son subutilizados y que al interconectarse pueden llegar a tener características superiores a las de un supercomputador, ahorrando costos en la adquisición de equipos, mantenimiento, infraestructura y administración.

6.3 ARQUITECTURAS DISTRIBUIDAS

6.3.1 COMPUTACIÓN VOLUNTARIA O DE CICLOS REDUNDANTES

La computación de ciclos redundantes o también llamada computación zombie “está compuesta por un servidor o grupo de servidores que distribuyen tareas para ser procesadas entre los diferentes sistemas voluntarios que se han adjuntado para colaborar en la ejecución del proyecto. Los computadores voluntarios ejecutan estas tareas cuando se encuentran ociosos, no interfiriendo en la ejecución y uso del computador por parte del usuario. De esta manera, se dona a la ejecución de los proyectos los ciclos redundantes, aprovechando al máximo la capacidad de procesamiento del computador.”⁹ Básicamente, cuando se deja el computador encendido, pero sin utilizarlo, la capacidad de procesamiento se desperdicia por lo general en algún protector de pantalla, este tipo de procesamiento distribuido utiliza un computador que se ha dado como voluntario cuando no se está utilizando, aprovechando al máximo la capacidad de procesamiento.

⁹ Carlos Moreno Losada, PLATAFORMA COMPUTING@HOME, 2008: (Disponible en : <http://www.recercat.net/bitstream/2072/13680/1/PFC+Carlos+Moreno+Losada.pdf>)

La Computación Voluntaria es un acuerdo en el que personas o entidades donan sus recursos computacionales a proyectos de investigación de su interés; los cuales utilizan dichos recursos para realizar computación y/o almacenamiento distribuido. El uso de una gran red de computadores trabajando juntos es capaz de desarrollar una gran capacidad de cálculo, que es la que usan los diferentes proyectos científicos para estudiar teorías, pruebas, experimentos, etc.

El primer proyecto de computación voluntaria conocido fue el Great Internet Mersenne Prime Search¹⁰, que se inició en enero de 1996. En años sucesivos surgieron otros proyectos como Superweb¹¹, Popcorn¹², Charlotte¹³ y Bayanihan¹⁴. El desarrollador de este último proyecto, Bayanihan, fue quien acuñó el término "computación voluntaria". En 1999 fueron lanzados al público proyectos como SETI@home¹⁵ y Folding@home¹⁶, Recibiendo buena acogida por varios de cientos de miles de voluntarios alrededor del mundo.

El software cliente de los primeros proyectos de computación voluntaria consistía en un simple programa que combinaba la computación distribuida y la infraestructura del sistema. Esta arquitectura monolítica no era flexible, y por ejemplo era difícil actualizar las versiones sin modificar la infraestructura. Recientemente se han desarrollado sistemas middleware que proveen una infraestructura de computación distribuida independiente de la computación

¹⁰ Great Internet Mersenne Prime Search, Dc Central Real World. (Disponible en: <http://library.thinkquest.org/C007645/english/2-mersenne-0.htm>)

¹¹ Alexandrov, A.D.; M. Ibel, K.E. Schauser, K.E. Scheiman (1996). "SuperWeb: Research issues in Java-Based Global Computing". Proceedings of the Workshop on Java for High performance Scientific and Engineering Computing Simulation and Modelling. New York: Syracuse University.

¹² Regev, O; Nisan, N (October 25 - 28, 1998). "The POPCORN market~Wan online market for computational resources". Proceedings of the First international Conference on information and Computation Economies. Charleston, South Carolina, United States: ACM Press, New York, NY. pp. 148–157.

¹³ A.; Karaul, M., Kedem, Z., Wyckoff, P. (Sept 1996). "Charlotte: Metacomputing on the Web". Proceedings of the 9th International Conference on Parallel and Distributed Computing Systems. <http://citeseer.ist.psu.edu/article/baratloo96charlotte.html>.

¹⁴ Sarmanta, L.F.G. "Bayanihan: Web-Based Volunteer Computing Using Java". Lecture Notes in Computer Science 1368, Springer-Verlag, 1998. pp. 444-461. Proc. of the 2nd International Conference on World-Wide Computing and its Applications (WWCA'98), Tsukuba, Japan, March 3-4, 1998

¹⁵ Seti@Home, Detección de señales de otras civilizaciones, <http://seti.astroseti.org/setiathome/>

¹⁶ Foldin@home, Understaning de protein folding, <http://folding.stanford.edu/>

científica. Por ejemplo entre ellos el más utilizado es Berkely Open Infrastructure for Network Computing (BOINC), de la Universidad de Berkely, XtremWeb desarrollado por la Universidad Paris-South, Xgrid desarrollado por Apple para Mac OS X, GridMP desarrollado por United Devices para uso comercial.

6.3.2 CLUSTER COMPUTING

Un clúster es un tipo de sistema de procesamiento paralelo o distribuido, el cual consiste en un grupo de computadores de relativo bajo costo conectados entre sí mediante un sistema de red de alta velocidad (gigabit de fibra óptica por lo general) y un software que realiza la distribución de la carga de trabajo entre los equipos. Por lo general, este tipo de sistemas cuentan con un centro de almacenamiento de datos único. La característica inherente de un clúster es la de compartir recursos: ciclos de CPU, memoria, datos y servicios. Los clúster están conformados por computadores genéricos con uno o más procesadores, denominados nodos. Además tienen la ventaja de ser sistemas redundantes, al estar fuera de servicio el procesador principal el segundo se dispara y actúa como un Fail Over.

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

Para empezar a hablar acerca de la clasificación de los clúster es bueno empezar por nombrar que existe una taxonomía que clasifica las arquitecturas de computadores en base en el flujo que siguen los datos dentro de la máquina y de

las instrucciones sobre esos datos. Esta taxonomía fue propuesta por Michael J. Flynn en 1972, al cual se le debe el nombre de dicha clasificación.¹⁷

Flynn clasifica los sistemas en cuatro categorías:

- **SISD (Una instrucción, un dato)**

Los sistemas de este tipo se caracterizan por tener un único flujo de instrucciones sobre un único flujo de datos, es decir, se ejecuta una instrucción detrás de otra. Un ejemplo de estos sistemas son las máquinas secuenciales convencionales.

- **MISD (Múltiples instrucciones un dato)**

Poco común debido al hecho de que la efectividad de los múltiples flujos de instrucciones suele precisar de múltiples flujos de datos. Sin embargo, este tipo se usa en situaciones de paralelismo redundante, como por ejemplo en navegación aérea, donde se necesitan varios sistemas de respaldo en caso de que uno falle. También se han propuesto algunas arquitecturas teóricas que hacen uso de MISD, pero ninguna llegó a producirse en masa.

- **SIMD (Una instrucción, múltiples datos)**

Un computador que explota varios flujos de datos dentro de un único flujo de instrucciones para realizar operaciones que pueden ser paralelizadas de manera natural. Por ejemplo, un procesador vectorial.

- **MIMD (Múltiples instrucciones, múltiples datos)**

Varios procesadores autónomos que ejecutan simultáneamente instrucciones diferentes sobre datos diferentes. Los sistemas distribuidos suelen clasificarse como arquitecturas MIMD; bien sea explotando un único espacio compartido de memoria, o uno distribuido. [8]

¹⁷ Flynn, M., Some Computer Organizations and Their Effectiveness, IEEE Trans. Comput., Vol. C-21, pp. 948, 1972.

Partiendo de esta clasificación tenemos que los sistemas distribuidos se pueden clasificar como arquitecturas MIMD, y que esta arquitectura se puede clasificar a su vez dependiendo de su acoplamiento en Sistemas multiprocesador (fuertemente acoplados) y sistemas multicomputador (débilmente acoplados). Los sistemas fuertemente acoplados son aquellos en los que los procesadores dependen unos de otros y los sistemas débilmente acoplados son aquellos en los que existe poca interacción entre los diferentes procesadores que forman el sistema.

Teniendo en cuenta las clasificaciones nombradas anteriormente se puede decir que un ejemplo de una maquina paralela multicomputador es un clúster y que en estos multicomputadores la memoria se encuentra distribuida y no compartida como en los sistemas multiprocesador. Los computadores se comunican a través de paso de mensajes, ya que éstos sólo tienen acceso directo a su memoria local y no a las memorias del resto de procesadores.

6.3.2.1 CLASIFICACION DE LOS CLUSTER

El termino clúster tiene diferentes connotaciones para diferentes grupos de personas. Los tipos de clúster, según el uso que se les dé y los servicios que ofrezcan, determinan el significado del término para el grupo que lo utiliza. Tres distinciones básicas en los clúster son la base para su clasificación, el rendimiento, el balanceo de carga y la disponibilidad; si bien todas se pueden lograr con un clúster al optimizar una se pierden características de las otras, por lo tanto, según la aplicabilidad de los clúster se han desarrollado diferentes líneas tecnológicas. [3]

Alto rendimiento: Esta línea surge frente a la necesidad de súper computación para determinadas aplicaciones, lo que se persigue es conseguir que un gran número de maquinas individuales actúen como una sola maquina muy potente, este tipo de clúster se aplica mejor en problemas grandes y complejos que requieren una cantidad enorme de potencia computacional. Este tipo de clúster en

general está enfocado hacia las tareas que requieren gran poder computacional, grandes cantidades de memoria, o ambos a la vez, teniendo en cuenta que las tareas podrían comprometer los recursos por largos periodos de tiempo.

Balanceo de carga: Este tipo de tecnología de clúster, es el destinado al balanceo de carga. Surge el concepto de clúster de servidores virtuales, lo cual es un clúster que permite que un conjunto de servidores de red compartan la carga de trabajo y de tráfico de sus clientes, aunque aparezcan para estos clientes como un único servidor. Al balancear la carga de trabajo en un conjunto de servidores, se mejora el tiempo de acceso y la confiabilidad. Además, como es un conjunto de servidores el que atiende el trabajo, la caída de uno de ellos no ocasiona la caída total del sistema. Este tipo de servicio es de gran valor para las compañías que trabajan con grandes volúmenes de tráfico y trabajo en sus web.

Alta disponibilidad: En un clúster de alta disponibilidad lo que se pretende es lograr la máxima disponibilidad de servicios y el rendimiento sostenido lo cual se puede lograr con el mantenimiento de servidores que actúen entre ellos como respaldos de la información que sirven. La flexibilidad y robustez que proporcionan este tipo de clúster, los hace necesarios en ambientes de intercambio masivo de información, almacenamiento de datos sensibles y allí donde sea necesaria una disponibilidad continua del servicio ofrecido.

Alta eficiencia: Son clúster cuyo objetivo de diseño es el ejecutar la mayor cantidad de tareas en el menor tiempo posible. Existe independencia de datos entre las tareas individuales. El retardo entre los nodos del clúster no es considerado un gran problema.

Por otro lado, se pueden clasificar los clúster dependiendo de cómo serán usados cotidianamente, es decir para que sean usados normalmente, es por esta característica que pueden ser clúster dedicados y no dedicados, a continuación se mostrara una explicación acerca de cada uno.

6.3.2.2 CLUSTER DEDICADO

Un clúster dedicado es un clúster en el que sus nodos no disponen de teclado, ratón ni monitor. Son clúster en donde su uso está exclusivamente dedicado a realizar tareas relacionadas con el clúster. Este tipo de clúster se enfoca necesariamente en los diferentes cálculos o trabajos para los cuales fueron establecidos, los computadores conectados no cuentan con más sistemas operativos que el que se está usando.

6.3.2.3 CLUSTER LIGERO

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

Los clúster actualmente son de gran utilidad en diferentes aplicaciones tanto científicas, como empresariales. En el contexto en que se están usando los clúster, tenemos las aplicaciones científicas, en donde vemos que son de gran ayuda debido a que permite realizar ciertas aplicaciones que computacionalmente son muy intensivas, da recursos importantes como lo son de almacenamiento y de memoria. Además da la posibilidad de usar nodos y sistemas dedicados, en entornos HPC Y HTC. Ejemplos de estas aplicaciones son Simulaciones, genómica computacional, predicción meteorológica (MM5), simulación de corrientes, vertidos en el mar y aplicaciones en química computacional.

Los clústers ligeros son una alternativa muy empleada actualmente para el uso de recursos que pasan gran parte de su tiempo de manera ociosa, esto ayuda y facilita el desarrollo de diferentes actividades que requieren gran capacidad de almacenamiento y procesamiento. Teniendo en cuenta estos aspectos es importante recalcar la importancia que tienen este tipo de clústers dentro de los campus universitarios y como su implementación ha permitido el desarrollo de

grandes proyectos que generan a su vez el querer cada vez mas que se creen y se empleen este tipo de alternativas para sacar el mejor provecho a tantos recursos que normalmente no son usados de la mejor manera.

Como se mencionará más adelante, para el desarrollo de este proyecto fue necesaria la creación de clústers ligeros, y que a su vez darán paso al concepto de Grid Ligera y a poder concluir con los objetivos planteados.

6.3.3 GRID COMPUTING

Varios conceptos se han dado acerca de lo que es una grid y entre ellos podemos hablar de el concepto elaborado por el Grid Computing Information Centre, una de las asociaciones dedicada exclusivamente al desarrollo de esta tecnología, la cual llama *grid* a un “*tipo de sistema paralelo y distribuido que permite compartir, seleccionar y reunir recursos autónomos geográficamente distribuidos en forma dinámica y en tiempo de ejecución, dependiendo de su disponibilidad, capacidad, desempeño, costo y calidad de servicio requerida por sus usuarios*”.¹⁸ Según esta definición, se busca aprovechar la sinergia que surge de la cooperación entre recursos computacionales y proveerlos como servicios.

Otra definición más estructurada expuesta por Foster, Kesselman y Tuecke, precursores de la computación grid y a los cuales se les debe el término grid, plantea la existencia de *organizaciones virtuales (OV)* como puntos de partida de este enfoque. Una *organización virtual* es “un conjunto de individuos y/o instituciones definida por reglas que controlan el modo en que comparten sus recursos”. Básicamente, son organizaciones unidas para lograr objetivos comunes. Ejemplos de OVs podrían ser proveedores de servicios de aplicaciones o almacenamiento, equipos de trabajo empresarial realizando análisis y planeamiento estratégico, universidades involucradas en un proyecto de investigación conjunto, etc. Las OVs varían enormemente en cuanto a sus objetivos, alcance, tamaño, duración, estructura, comunidad y sociología. Sin

¹⁸ Computación Grid, Textos científicos, 2007: <http://www.textoscientificos.com/redes/computacion-grid/>

embargo, existen varios requerimientos y problemas subyacentes tales como la necesidad de relaciones flexibles para compartir recursos, niveles de control complejos y precisos, variedad de recursos compartidos, modos de funcionamiento, calidad de servicio, etc. Las tecnologías actuales o bien no proveen espacio para la variedad de recursos involucrados o no aportan la flexibilidad y control de las relaciones cooperativas necesarias para establecer las OV. [12]

Como solución, se propone el *grid* como un modelo de trabajo para “compartir recursos en forma coordinada y resolver problemas en organizaciones virtuales multi-institucionales de forma dinámica”. De esta manera, varias instituciones pueden formar distintas OV e incluso formar parte de más de una al mismo tiempo, realizando diferentes roles e integrando distintos recursos.

La grid permite el intercambio, selección y agregación de una amplia variedad de recursos entre los que se pueden nombrar: computadores, supercomputadores, sistemas de almacenamiento, dispositivos especializados, etc. Que están generalmente distribuidos geográficamente y pertenecen a diferentes organizaciones, con el fin de resolver problemas que impliquen cómputo a gran escala y uso intensivo de datos en la ciencia, industria y el comercio. La computación grid fue desarrollada en ámbitos científicos a principios de los años 1990 y su entrada al mercado comercial siguiendo la idea de la llamada Utility computing supone una revolución que dará mucho que hablar.

Los recursos que se distribuyen en la red por medio de la grid, se distribuyen de forma transparente pero guardando unas pautas de seguridad y políticas de gestión de carácter tanto técnico como económico. Así pues, su objetivo será el de compartir una serie de recursos en la red de manera uniforme, segura, transparente, eficiente y fiable, ofreciendo un único punto de acceso a un conjunto de recursos distribuidos geográficamente en diferentes dominios de administración. Estos recursos englobados no están conectados o enlazados firmemente, es decir no tienen por qué estar en el mismo lugar geográfico. Se

puede tomar como ejemplo el proyecto SETI@Home, en el cual trabajan computadores alrededor de todo el planeta para buscar vida extraterrestre.

Una de las principales diferencias entre la computación grid y la computación clúster es la forma en la que los recursos son administrados, en el caso de los clúster la asignación de los recursos es llevada a cabo centralizadamente por un administrador de recursos, y todos estos recursos trabajan juntos cooperativamente como un solo recurso unificado en cambio, en el caso de la grid la administración tiende a ser más descentralizada y no intenta proveer una sola imagen del sistema.

6.3.3.1 GRID LIGERA

Una grid ligera es un tipo de grid que además de utilizar recursos heterogéneos utiliza recursos que no son dedicados, que solo se utilizan en determinados momentos. La disponibilidad de recursos en este tipo de grid es dinámica. En pocas palabras, una grid ligera usa los recursos cuando es posible que estos estén disponibles para trabajar en cierto intervalo de tiempo. Al emplear recursos no dedicados hace que sea de carácter oportunista su desempeño de una grid como tal.

6.3.3.2 DESARROLLO HISTORICO DE GRID COMPUTING

Con el paso del tiempo, el desarrollo del hardware llevó a la mejora y abaratamiento de este, dando paso a tecnologías que antes era impensable emplear, debido a costos o el bajo rendimiento que ofrecían, como lo eran los clúster. Los clúster surgieron como una nueva alternativa muy recomendable en el campo de la computación, tanto paralela como distribuida. Consiste en unir un conjunto de computadores personales o estaciones mediante una red (LAN, SAN, etc.). Su principal ventaja es la mejor relación de costo a rendimiento, sin embargo la gran demanda tanto de computación como de espacio y gestión de almacenamiento requeridos por un gran número de aplicaciones que gestionan grandes cantidades de datos y han de hacerlo de forma eficiente y sencilla, exige

el uso de nuevas tecnologías, como es el caso de la computación grid o grid computing.

Debido a esto es que los sistemas grid se han convertido en un paradigma de la computación distribuida aparecido en las últimas décadas. Estos sistemas disponen de un número indeterminado de computadores dedicados, que funcionan como si se trataran de uno único. Para llevar a cabo estas funciones se suele utilizar un software determinado que permite las comunicaciones entre computadores, la gestión de almacenamiento de datos y el envío de trabajos, entre otras tareas.

6.3.3.3 CARACTERISTICAS DE GRID COMPUTING

Entre las características más importantes de grid computing podemos encontrar:

- **Capacidad de balanceo de sistemas:** no habría necesidad de calcular la capacidad de los sistemas en función de los picos de trabajo, ya que la capacidad se puede reasignar desde la granja de recursos a donde se necesite.
- **Alta disponibilidad:** si un servidor falla, se reasignan los servicios en los servidores restantes, dando la posibilidad de disponer de recursos para seguir con las aplicaciones que se estén ejecutando.
- **Reducción de costos:** con esta arquitectura los servicios son gestionados por "granjas de recursos". Ya no es necesario disponer de "grandes servidores" y se puede hacer uso de componentes de bajo costo. Cada sistema puede ser configurado siguiendo el mismo patrón.

Es importante mencionar que se relaciona el concepto de grid con la nueva generación del protocolo IP. El nuevo protocolo de Internet IPv6 permitirá trabajar con una Internet más rápida y accesible. Una de las ideas clave en la superación de las limitaciones actuales de Internet IPv4 es la aparición de nuevos niveles de servicio que harán uso de la nueva capacidad de la red para intercomunicar los computadores.

Este avance en la comunicación permitirá el avance de las ideas de grid computing al utilizar como soporte la altísima conectividad de Internet. Es por ello que uno de los campos de mayor innovación en el uso del grid computing, fuera de los conceptos de supercomputación, es el desarrollo de un estándar para definir los Grid Services frente a los actuales Web Services.

6.3.3.4 VENTAJAS DE GRID COMPUTING

La Computación Grid está creada con el fin de brindar una solución a determinadas cuestiones, como problemas que requieren de un gran número de ciclos de procesamiento o acceso a una gran cantidad de datos. Se trata de una solución altamente escalable, potente y flexible, ya que evitarán problemas de falta de recursos (cuellos de botella). Esta tecnología brinda a las empresas el beneficio de la velocidad, lo que supone una ventaja competitiva, con lo cual se provee una mejora de los tiempos para la producción de nuevos productos y servicios.

Además facilita la posibilidad de compartir, acceder y gestionar información, mediante la colaboración y la flexibilidad operacional, asociando no sólo recursos tecnológicos dispares, sino también personas y aptitudes diversas. Otro de los aspectos al que se tiende es a incrementar la productividad otorgando a los usuarios finales acceso a los recursos de computación, datos y almacenamiento que necesiten, cuando los necesiten.¹⁹

La computación Grid ofrece beneficios como:

- Proporcionar un mecanismo de colaboración transparente entre grupos dispersos, tanto científicos como comerciales.
- Posibilita el funcionamiento de aplicaciones a gran escala.
- Facilita el acceso a recursos distribuidos desde nuestros computadores.
- Permite colaborar con la idea de "e-Ciencia".

¹⁹ Computación Grid, Ventajas y desventajas, Aplicaciones, Textos científicos, 2007:
<http://www.textoscientificos.com/redes/computacion-grid/ventajas-desventajas-aplicaciones>

La tecnología derivada del grid abre un enorme abanico de posibilidades para el desarrollo de aplicaciones en muchos sectores. Por ejemplo: desarrollo científico y tecnológico, educación, sanidad, y administración pública entre otros. [12]

6.3.3.5 DESVENTAJAS DE GRID COMPUTING

Algunas desventajas con las que cuenta la computación grid y que se deben tratar de solucionar son:

- **El paralelismo:** ya que puede estar visto como un problema, debido a que una máquina paralela es muy costosa. Pero si en cambio se tiene la disponibilidad de un conjunto de máquinas heterogéneas de pequeño o mediano porte, cuya potencia computacional sumada sea considerable, eso permitiría generar sistemas distribuidos de muy bajo costo y gran potencia computacional.
- **Recursos heterogéneos:** la computación grid debe ser capaz de poder manejar cualquier tipo de recurso que maneje el sistema, si no resultará totalmente inútil.
- Necesidad de desarrollo de aplicaciones para manejar el grid, así como desarrollo de modelos eficientes de uso.
- Grid computing necesita, para mantener su estructura, de diferentes servicios como Internet, conexiones de 24 horas, los 365 días, con banda ancha, servidores de capacidad, seguridad informática, VPN, firewalls, encriptación, comunicaciones seguras, políticas de seguridad, normas ISO, y algunas características más. Sin todas estas funciones y características no es posible hablar de Grid Computing. [12]

6.3.3.6 APLICACIONES DE GRID COMPUTING

Como se mencionó anteriormente la computación grid puede ser útil en varios sectores y ser usada en diferentes aplicaciones dependiendo del sector al cual este dirigida, ejemplos de estas aplicaciones tenemos:

- **Computación distribuida:** ayuda en la Infraestructura y aplicaciones para el cálculo de alto rendimiento.
- **Servicios puntuales:** Se tiene en cuenta los recursos que una organización puede considerar como no necesarios. Grid presenta a la organización esos recursos.
- **Proceso intensivo de datos:** Son aquellas aplicaciones que hacen un gran uso del espacio de almacenamiento. Este tipo de aplicaciones desbordan la capacidad de almacenamiento de un único nodo y los datos son distribuidos por todo el grid. Además de los beneficios por el incremento de espacio, la distribución de los datos a lo largo del grid permite el acceso a los mismos de forma distribuida.
- **Entornos virtuales de colaboración:** Área asociada al concepto de Tele inmersión, de manera que se utilizan los enormes recursos computacionales del grid y su naturaleza distribuida para generar entornos virtuales 3D distribuidos.[12]

6.3.3.7 SEGURIDAD

Hablar de seguridad es un poco delicado en este tipo de computación distribuida pues las conexiones se hacen de forma remota y no local, entonces suelen surgir problemas para controlar el acceso a los otros nodos. Esto puede aprovecharse para un ataque de DoS, aunque la red no va a dejar de funcionar porque uno falle. Esa es una ventaja de este sistema grid. La seguridad en la grid, está sustentada con las “intergrids”, donde esa seguridad es la misma que ofrece la red Lan sobre la cual se utiliza tecnología grid.

6.3.3.8 ALGUNOS PROYECTOS DE GRID COMPUTING

- **Globus Project:** es una iniciativa multi-institucional para la investigación y el desarrollo de tecnologías fundamentales para Grids, con la activa participación de la empresa IBM, cuya intención principal es crear una

plataforma completa donde compartir aplicaciones y recursos informáticos en Internet.

- **TeraGrid:** Dedicado a la investigación científica abierta, tiene el objetivo de interconectar instalaciones y centros de investigación académica en puntos distantes geográficamente, está considerado como una de las infraestructuras más grandes y más rápidas del mundo. Mediante este proyecto, los científicos tendrán la capacidad para simular actividades sísmicas en estructuras a fin de diseñar edificios y puentes más seguros, los astrónomos podrán compartir datos desde sus telescopios y los investigadores médicos tendrán la posibilidad de compartir ideas y datos para quizá curar una enfermedad, lo que asegura que las oportunidades que se ofrecen a los científicos son incalculables.
- **CrossGrid:** tiene como fin el desarrollo, de aplicaciones interactivas en entornos Grid y la extensión del banco de ensayo (testbed) de DataGrid.
- **OpenMolGrid:** utiliza un set de aplicación con herramientas orientadas a establecer servicios centrales Grid y funciones provistas por la infraestructura Eurogrid, y se desarrollan herramientas para aumentar el acceso a las bases de datos heterogéneas y de distribución y para adaptar las herramientas de software existentes.
- **UK e-Science:** proyecto que facilitará a los investigadores aprovechar enormes recursos informáticos de todo el mundo para afrontar los desafíos científicos clave en campos como el genoma humano y la física de partículas. Este proyecto permitirá a los científicos que utilicen Grid accedan a las bases de datos de los resultados de investigación procedentes de sistemas que se encuentren en cualquier parte del mundo.
- **EGEE (Enabling Grids for e-Science in Europe):** este proyecto utilizará la tecnología Grid para interconectar recursos computacionales de veintisiete países europeos, con el objeto primordial de unir los recursos de los equipos informáticos de las instituciones participantes y crear de este modo un supercomputador virtual, aprovechando la infraestructura de

comunicación de banda ancha proporcionados por la Red Europea de Investigación Géant.[7]

6.4 COMPUTEMODE

ComputeMode (CM) es una infraestructura software basada en Debian Linux que permite extender o crear un Clúster HPC agregando recursos de cómputo no usados. Esto debido a que la mayoría de los computadores en las grandes compañías y grandes campus universitarios están ociosos durante la noche, los fines de semana, las vacaciones, por entrenamiento de personal o viajes de negocios.²⁰ Fue desarrollado por el equipo Mescal²¹ del Laboratorio de Informática de Grenoble ²² en Francia.

Computemode se basa en una arquitectura maestro-esclavo usando el servidor central como maestro. Dispone de una interfaz de administración web para un manejo fácil, manteniendo la disponibilidad de los computadores registrados en la red de área local.

El servidor CM mantiene la disponibilidad de los computadores personales registrados en la red local. Cada propietario de un computador, de acuerdo con su política de la empresa, puede optar por dejar que su computador se utilice en las noches, los fines de semana o vacaciones. La mayoría de estaciones de trabajo se utilizan de forma interactiva 50 horas a la semana.

²⁰ Computemode Project http://computemode.imag.fr/mediawiki/index.php/ComputeMode_Grid_Manager

²¹ <http://mescal.imag.fr>

²² <http://liglab.imag.fr>

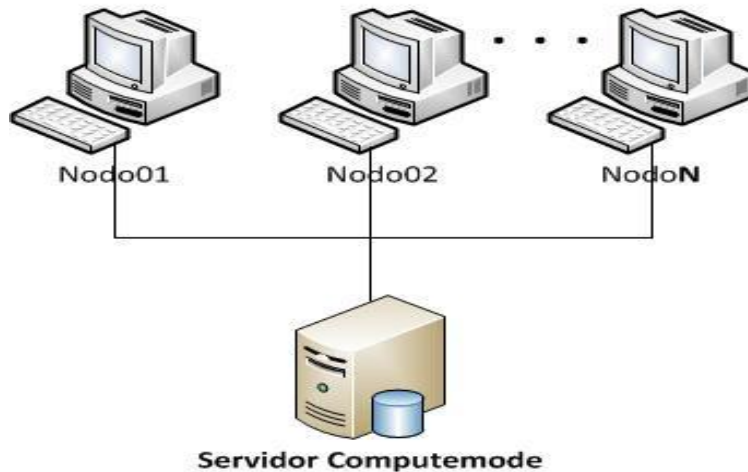


Ilustración 2: Arquitectura ComputeMode

Se conocen dos estados, los cuales pueden ser adquiridos por los computadores que se utilizan en el sistema:

- **Usermode:** En modo usuario, es cuando la maquina está corriendo en el sistema que habitualmente usa, que en la mayoría de las organizaciones es Windows. El propietario de la máquina ni siquiera notará que su PC se gestiona como un nodo de procesamiento ComputeMode.²³
- **Computemode:** El modo de la computación que es de donde viene su nombre, es cuando se realizan cálculos para el Clúster. Se activa cuando la maquina esta en un periodo de tiempo que se haya estimado como inactivo. Si el servidor ComputeMode detecta un pico de cómputo, el PC puede ser conectado de forma remota a la computación modo. El cambio de modo de usuario se realiza a través de un reinicio automático de la máquina y procede a un arranque remoto. El arranque remoto es manejado por el servidor ComputeMode con el "entorno de pre-ejecución" (PXE) de protocolo, que está disponible de forma nativa desde la BIOS de los PC

²³ Computemode Project <http://computemode.imag.fr/mediawiki/index.php/Overview>

desde 1999. Mientras que en el modo de la Computación, la máquina se está ejecutando bajo el sistema operativo Linux, y no tiene acceso a cualquier unidad local. ¹⁴

El sistema ofrece un conjunto de servicios que hacen posible que los recursos reciban por red su sistema operativo y monten su sistema de archivos raíz por medio de la red. Se basa en la implementación de disco RAM para la configuración preliminar que luego transfiere el control al sistema de archivos raíz que se implementa totalmente en red a través del protocolo NFS, permite tanto la adicción de computadoras, como maquinas virtuales.

Un usuario puede enviar trabajos de cómputo al sistema a través del uso de un clásico manejador de colas ingresando en el servidor (a través de ssh por ejemplo); Computemode emplea OAR para realizar el manejo de colas.

El manejador de colas tiene en cuenta lo siguiente:

- Reservar recursos adecuados para los cálculos.
- Planificar la ejecución de los trabajos.

El equilibrio de carga con un gestor de trabajo que ya está instalado también es algo que se puede lograr: por ejemplo, el clúster dedicado normalmente maneja la carga de trabajo pero para las horas punta, un poder extra-CPU podría ser muy útil.

Cuando el servidor ComputeMode detecta que un nodo cliente no está disponible, éste regresa al anterior modo de usuario para que el dueño ni siquiera se dé cuenta de que su PC ha sido utilizado por ComputeMode.

Más adelante se explicara detalladamente acerca del manejador de colas OAR.

6.4.1 DESCRIPCION COMPUTEMODE

ComputeMode se compone de tres partes principales:

Interface de Administración de recursos:

Esta interface permite manejar los recursos a utilizar, se basa en una interface web de fácil manejo que interactúa con una base de datos. Aquí se deciden el nombre del nodo, el horario en que será utilizado y la imagen que se desplegara en el. Se puede definir un horario para varios recursos, también permite agregar y eliminar nodos. El procedimiento para agregar nodos permite hacerlo manual o automáticamente lo cual lo hace fácil de administrar.

Herramientas de manejo de colas y monitorización de recursos:

Para el manejo de colas se cuenta con OAR (Resource Management System for High Performance Computing), este es un administrador de recursos de computo, el cual es flexible y simple, maneja recursos en un clúster, permite la ejecución de trabajos interactivos y de reserva. En su forma más simple un trabajo se define por un programa a ejecutar, el número de nodos a utilizar y el tiempo por el cual se necesitan los recursos. El trabajo interactivo permite que el usuario le sea asignado los recursos inmediatamente y este interactúe directamente con ellos. En un trabajo de reserva, el usuario reserva los recursos para utilizarlos a determinada fecha por un espacio de tiempo definido, y es el usuario el que se encarga de conectarse al trabajo. Para monitorizar los recursos se cuenta con Monika el cual permite visualizar el estado de los recursos mediante una interfaz web, esta se comunica constantemente con el servidor OAR.

Imágenes de sistema operativo y servicios para el despliegue:

Los sistemas operativos que se despliegan en los computadores se dividen en 2 partes: el kernel Linux y el sistema de ficheros. Ya que Computemode es un

ambiente configurable lo cual Linux es transferido al nodo de cómputo mientras que el sistema de ficheros es compartido por medio de NFS. Los servicios para realizar el despliegue de los sistemas son los siguientes: TFTP, DHCP, NFS.

La funcionalidad más importante que agrega el mecanismo de generación automática de archivos, es la posibilidad de mantener el servidor funcionando en cualquier momento sin obstaculizar las actividades que las máquinas tienen por defecto, como trabajar con un sistema Windows. Una máquina integrada al sistema Computemode debe configurarse para establecer la tarjeta de red como primer dispositivo de inicio. Cuando una maquina inicia por red, pregunta al servidor si es tiempo de unirse al sistema de computo, si se ha programado así, el servidor le envía un archivo para que inicie por la imagen del sistema de computo; de lo contrario envía otro archivo el cual le indica que inicie por el sistema local, que en este caso será Windows.

En el servidor se emplean dos interfaces de red virtuales, una de estas interfaces es para la comunicación del servidor con el exterior y es la que normalmente utilizan los equipos que se encuentran conectados a la red. La otra de las interfaces se utiliza para la comunicación con los nodos de cómputo, las direcciones IP utilizadas en esta interfaz son privadas y completamente diferentes a las utilizadas por otros equipos. Lo anterior brinda dos ventajas, la primera relacionada con la imposibilidad que tiene un usuario de conectarse a estos equipos desde otra red, en otras palabras estos equipos quedan aislados de las redes externas existentes y la otra ventaja tiene que ver con la no creación de conflictos de direcciones IP provocadas por la utilización de una IP que ya se halla asignado a otro equipo. [7]

Los principales beneficios de Computemode son los siguientes:

- **Fácil despliegue:** la integración dentro de una infraestructura existente es muy fácil. Ninguna modificación es necesaria en los computadores. Computemode viene como una solución única de software. Puede llevarse

a cabo la integración con los mayores sistemas de manejo de lotes como Sun Grid engine, plataforma LSF y PBS.

- **Integración transparente para el Investigador:** No se tiene que modificar los trabajos, se trabaja con una interfaz como cualquier Clúster Beowulf.
- **Integración transparente para el dueño de la computadora:** Computemode corre cuando el PC esta ocioso (noche, fines de semana) molestando en lo más mínimo al usuario del PC.
- Cuenta con una interfaz que permite la administración de los recursos para establecer horarios de cómputo.
- Esta herramienta es la única solución que cuenta con un completo conjunto de herramientas que permite la fácil administración de recursos tanto a nivel de incorporación y su manejo en horarios definidos, como a nivel de monitorización y la planificación de recursos en un clúster. [3]

6.5 OAR

Oar es un sistema de proceso por lotes que se basa en una base de datos (MySQL²⁴), un lenguaje de script (en Perl²⁵) y una opcional herramienta escalable de administración (componente del marco Taktuk²⁶). Se compone de módulos que interactúan sólo con base de datos y se ejecutan como programas independientes. Se puede decir que formalmente no hay API, el sistema está completamente definido por el esquema de base de datos. Este enfoque facilita el desarrollo de módulos específicos y cada módulo (como programadores) puede desarrollarse en cualquier idioma que tiene una biblioteca de acceso a bases de datos.²⁷ A

²⁴ <http://www.mysql.com/>

²⁵ <http://www.perl.org/>

²⁶ <http://taktuk.gforge.inria.fr/>

²⁷ OAR project: <http://oar.imag.fr/>

continuación se presentan algunas características principales de esta herramienta.

Características principales:

Entre las características principales con las que cuenta esta herramienta tenemos que:

- Permite trabajos interactivos y por lotes.
- Cuenta con Walltime.
- Multi-programadores de apoyo y multi-colas con prioridad.
- Programador de primer ajuste.
- Se pueden realizar reservas.
- Permite el soporte de tareas moldeable.
- Revisa los nodos de cómputo.
- Apoyo a la dinámica de los nodos.
- Permite el registro y la contabilidad.
- Suspendede y reanuda los trabajos.
- Adecuación de los recursos (trabajo / propiedades del nodo)
- Mejor esfuerzo-colas (para la explotación de recursos ociosos)
- Revisa los nodos de cómputo antes de lanzar los trabajos
- Cuenta con la herramientas de visualización de la actividad (Monika)
- No hay demonio en los nodos de cómputo
- ssh como los protocolos de ejecución remota (gestionado por Taktuk)
- Inserción dinámica y supresión de nodo de cómputo
- Permite el Inicio de sesión.
- Medio ambiente de apoyo a la demanda
- Red de integración con el sistema Cigri.

6.5.1 CAPACIDADES DE OAR

Oar es un programador de código abierto por lotes que proporciona una explotación simple y flexible de un clúster. Se administran los recursos de las agrupaciones como un planificador de proceso por lotes tradicionales (como PBS / Par / LSF / SGE), es decir, no se ejecuta su trabajo sobre los recursos, pero los administra (reserva, la concesión de acceso) a fin de permitir la conexión de estos recursos y utilizarlos. [18]

Su diseño se basa en herramientas de alto nivel:

- Motor de bases de datos relacionales MySQL o PostgreSQL,
- Lenguaje de script Perl,
- Sistema de confinamiento cpuset
- Herramienta escalable de administración: Taktuk.

Es lo suficientemente flexible para ser adecuado para grupos de producción y experimentos de investigación. En la actualidad gestiona más de 5.000 nodos y ha ejecutado a más de 5 millones de puestos de trabajo.

6.5.2 VENTAJAS DE OAR

Oar cuenta con ciertas ventajas que permiten que sea una herramienta de gran uso para la administración de recursos en una infraestructura de cómputo.

- No hay demonio específicos en los nodos.
- No hay dependencia de las bibliotecas de computación específicos como MPI.
- Las actualizaciones se realizan en los servidores, no en los nodos de computación.
- Todas las tareas de administración se realizan con el comando taktuk

- Permite hacer peticiones jerárquicas de recursos (manejar grupos heterogéneos).
- Gantt de programación (para que pueda visualizar las decisiones del planificador interno).
- Permite total o parcial del tiempo compartido.
- Los mejores trabajos esfuerzo: si quiere otro trabajo con los mismos recursos, entonces es eliminado automáticamente (útil para ejecutar programas como SETI @ home).
- Medio Ambiente apoyar el despliegue (Kadeploy) [18]

7. ESTADO DEL ARTE

El inicio de este proyecto y en el cual ha sido basado para su realización, fue en el proyecto de grado del ingeniero Cristian Camilo Ruiz Sanabria de título ANALISIS E IMPLEMENTACION DE UNA INFRAESTRUCTURA DE CÁLCULO DISTRIBUIDO EN LA RED UNIVERSITARIA²⁸, el cual desarrollo e implementó una arquitectura distribuida que usa los recursos de cómputo existentes en la red universitaria durante su tiempo ocioso. Esto con el fin de obtener una infraestructura que proporcione alto poder de cálculo. Para su implementación fue necesario que los equipos de cómputo contaran con el sistema operativo Linux para crear los clústers necesarios para llevar a cabo el desarrollo del proyecto. Para lograr que la creación de estos clústers se realizara de una manera sencilla y con una herramienta completa se elaboró un estudio de varias de ellas. Computemode y PelicanHPC permiten encontrar solución a la creación de estas estructuras basadas en la tecnología diskless. Para lograr esto fueron necesarias una serie de modificaciones y mejoras a la herramienta ComputeMode después de haber sido elegida entre otras herramientas para ser la que permitiera la implantación de dicha solución. En base a estas mejoras se consiguió que ComputeMode permitiera crear un clúster usando los recursos ociosos y de esta manera se concluyera con el objetivo primordial del proyecto.

Establecida e implementada la solución dentro del campus universitario en dos salas del CENTIC (Centro de Tecnologías de Información y Comunicación), las cuales se encuentran en una misma red, surgió la expectativa de ir un poco más allá de lo que se había logrado en busca de mayor capacidad de almacenamiento y de procesamiento por medio de la interacción con otros clúster dentro del campus universitario, pero la herramienta empleada contaba con la limitación de que no permitía la integración de recursos que no se encuentren dentro de la misma subred del servidor. Debido a esta limitación se dio paso al desarrollo de

²⁸ Cristian Camilo Ruiz Sanabria, ANALISIS E IMPLEMENTACION DE UNA INFRAESTRUCTURA DE CÁLCULO DISTRIBUIDO EN LA RED UNIVERSITARIA, Trabajo de grado , Universidad Industrial de Santander 2009

este proyecto, queriendo lograr que haya una interacción entre recursos que se encuentren en otras redes.

En base a la limitación nombrada anteriormente existen algunos mecanismos de manejo de redes que trabajando en conjunto pueden ser una alternativa para mejorar en el aprovechamiento de los equipos de computo. [4]

Entre estas alternativas podemos encontrar algunas como:

- **B-OAR:**

Esta herramienta presenta un manejo de recursos distribuidos basado en OAR [19], como se menciona anteriormente el cual es un manejador de colas desarrollado con herramientas de alto nivel, las cuales ofrecen un buen rendimiento, además de ser liviano y de fácil uso por parte del usuario. Además de OAR, esta herramienta se integra con BOINC, el cual es usado como middleware para la gestión de recursos en computación voluntaria, esto con el fin de utilizar las mejores características que ofrece cada una de estas plataformas. B-OAR busca integrar dos sistemas de gestión de recursos diferentes, implementando una interfaz que comprenda las arquitecturas subyacentes de cada sistema (OAR, BOINC) y que traduzca los formatos de almacenamiento de datos, a fin de que se puedan comunicar exitosamente.²⁹

Otra herramienta que se centra en el manejo de recursos distribuidos basada en OAR es CiGri.

- **CiGri:**

Este proyecto busca reunir los recursos no utilizados de las infraestructuras de cómputo de alto rendimiento locales para que estén disponibles para un gran conjunto de tareas. El software del servidor CiGri se basa en una base de datos y ofrece una interfaz de usuario para el lanzamiento de cálculos en red

²⁹ Combining OAR with the power of volunteer computing through BOINC
<http://oar.imag.fr/works/BOAR.html>

(secuencias de comandos y herramientas de la web. Los usuarios pueden supervisar y controlar desde su puesto de trabajo. Además el sistema proporciona mecanismos de detección de fallas, permitiendo la ejecución completa de tareas de larga duración.³⁰

Estas alternativas no fueron posibles de implementar en el desarrollo del proyecto debido a que por un lado B-OAR, es una herramienta usada para el desarrollo de la computación voluntaria y hasta el momento no ha sido implementada para permitir la gestión de recursos en una infraestructura de computo en redes heterogéneas. Por otro lado CiGri cuenta con ciertos problemas al momento de ser implementado, como lo son la no pre asignación de recursos, que algunos datos lleguen desordenados o se pierda la información y que cuenta con una prioridad mínima para el envío de los trabajos.

Teniendo en cuenta estas alternativas y los inconvenientes con los que cuenta cada una de ellas, se planteo una nueva propuesta para poder resolver las limitaciones del proyecto. En la siguiente sección se presentara la propuesta planteada.

³⁰ Cigri web site, <http://cigri.imag.fr/>

8. ARQUITECTURA PROPUESTA

Para el desarrollo de este trabajo y su culminación fue necesario realizar un análisis de las diferentes posibilidades que se tenían para poder cumplir con los objetivos planteados y en base a este análisis llegar a realizar el diseño de la estrategia que diera paso a la implementación del proyecto como tal. Es por ello que en esta sección se explicara detalladamente la arquitectura propuesta.

El presente trabajo consistió en el análisis y diseño de una estrategia que permitiera la interacción entre recursos distribuidos de una infraestructura de cómputo en redes heterogéneas. Al inicio del proyecto se contaba con una cantidad de computadores, distribuidos en diferentes lugares del campus universitario los cuales no eran aprovechados efectivamente además del uso que se le daba para las actividades académicas. Contaban también con el sistema operativo Windows y en horas fuera de clase, fines de semana y vacaciones no eran usados, desperdiciando tiempo de cómputo. Debido a esto se inicio con la instalación del sistema operativo Linux, en dos salas dentro del campus universitario; una en el CENTIC y otra en el laboratorio de redes de la Escuela de Ingeniería de Sistemas. Después se prosiguió a crear un clúster en cada sala por medio de la herramienta ComputeMode, al mismo tiempo que se contaba con estos dos clúster que se crearon, existían a su vez otros dos clúster los cuales hacían parte de la plataforma del centro de computación y cálculo científico UIS - SC3.³¹

Al contar con los 4 clúster creados en el campus universitario, estos se encontraban de manera independiente y se podían desplegar aplicaciones en cada uno de ellos pero no existía una interacción entre ellos mismos. En términos más generales cada clúster contaba con su propia base de datos, sus usuarios y su respectivo sistema de monitoreo (Monika) en donde se podían observar los cambios en los respectivos nodos de cada clúster. Los diferentes usuarios que se conectaban a la plataforma a través de ssh, podían hacer uso de uno de los

³¹ Supercomputación y Calculo Científico (SC3) UIS, <http://sc3.uis.edu.co> SC3

clúster dependiendo de las características y de la cantidad de nodos que necesitara para correr su aplicación. Gracias a la infraestructura que se presenta, se puede decir que nos encontramos ante una red heterogénea.

La implementación de la infraestructura propuesta en este proyecto pretende resolver la limitación que se tiene al no existir interconectividad ni gestión de recursos, en la infraestructura antes mencionada. Es por ello que a lo largo de la realización del proyecto se desea resolver esta limitación y unos pequeños inconvenientes que se dieron al momento de crear los dos clúster con la herramienta ComputeMode.

Principales limitaciones

- No existía una comunicación externa con los nodos que se encontraban en una VLAN.
- Para realizar la reserva de diferentes recursos, era necesario hacerla en el frontend de cada nodo que se fuera a utilizar.
- La descentralización de las cuentas de usuario y sus respectivos directorios de trabajo.
- La descentralización de la Base de Datos OAR, debido a que cada clúster contaba con su respectivo sistema de monitoreo (Monika).

8.1 DISEÑO

La arquitectura propuesta pretende resolver los inconvenientes antes mencionados y adaptarse a la arquitectura de red presente en la universidad, cumpliendo a su vez con los objetivos planteados. El diseño de la arquitectura propuesta se presenta a continuación con la siguiente grafica.

Mapa de red del centro de Super Computacion y Calculo Cientifico UIS – SC3

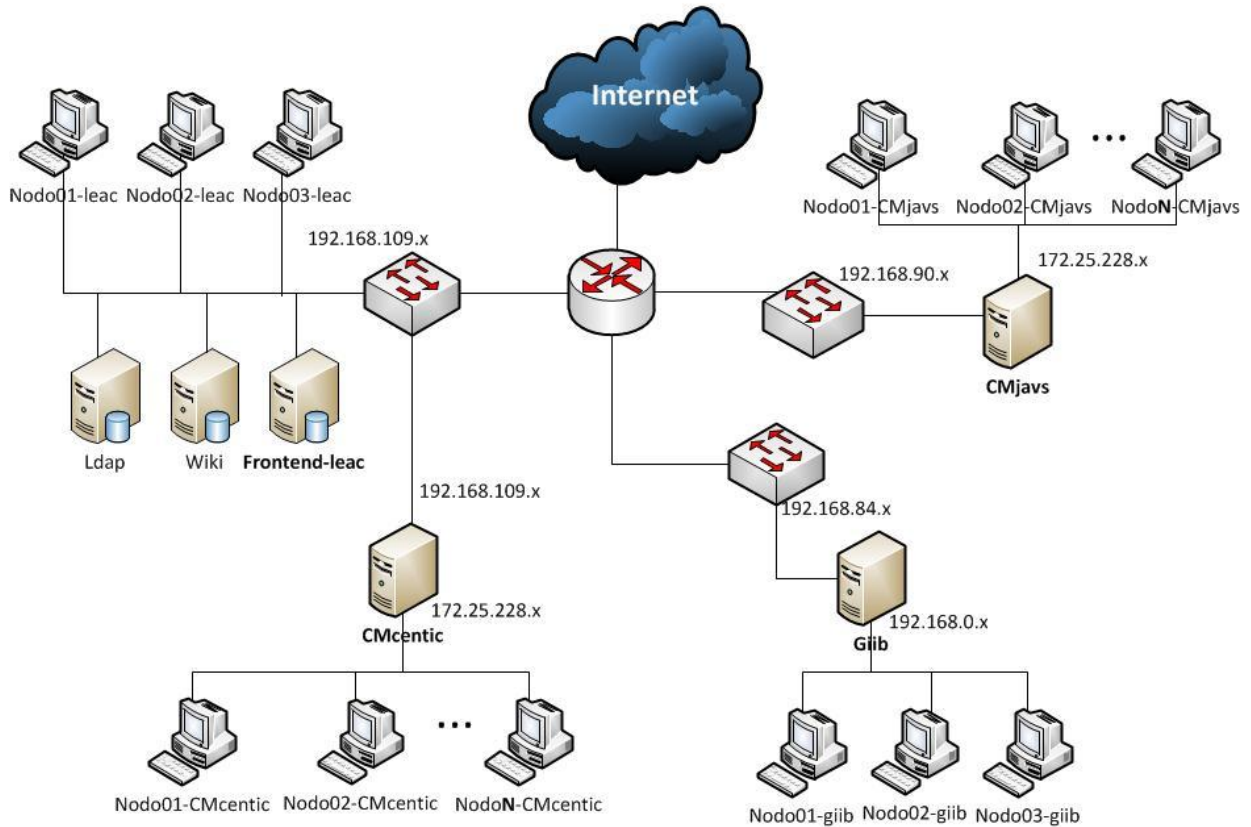


Ilustración 3: Diseño de la arquitectura propuesta

Esta arquitectura consta de 4 clúster ubicados en 4 sitios de la universidad, uno en la sala de redes de la Escuela de ingeniería de sistemas, otro en el grupo de investigación en ingeniería Biomédica (GIIB), y los otros dos en el CENTIC distribuidos en dos salas. Estos clústers se encuentran distribuidos en diferentes niveles de red dentro del campus universitario, 3 de ellos cuentan con una IP privada, lo cual no permite su interacción con otros clústers. Por otro lado el clúster Leac, el cual es el único que cuenta con una IP pública, permite que se pueda ingresar a el por fuera del campus universitario y que por dentro pueda interactuar y gestionar recursos con otros clústers. Observando estas características con las que cuenta cada uno de los clúster implementados hayan sido estos creados con ComputeMode o siendo clústers dedicados, es importante

resaltar que al no contar estos con una ip pública, no les es posible la comunicación con los demás y hay menos posibilidad de que se pueda lograr gestionar recursos que ayuden en el campo de la computación distribuida. Cada clúster cuenta a su vez con sus respectivos nodos y estos nodos unas respectivas características, las cuales serán presentadas mas adelante con el fin de que se tenga en cuenta la homogeneidad y heterogeneidad para las posibles interacciones que se lleven cuando este implementada la solución del proyecto.

A continuación se presenta una tabla con la distribución de cada clúster.

Nombre del Clúster	Numero de Nodos	Ubicación
CMcentic	32 Nodos	Centic
CMjavs	24 Nodos	EISI - Laboratorio Villabona
Giib	3 Nodos	EISI - GIIB
LEAC	3 Nodos	Centic Supercomputación

Tabla 1: Descripción de los clúster usados

Para más información acerca de la distribución de los clúster en el campus universitario, se puede ingresar a la página del centro de supercomputación y calculo científico UIS-SC3: ³²

El clúster CMcentic y el CMjavs, son clúster que fueron creados usando la herramienta ComputeMode, son clúster ligeros y que además son usados en horas fuera de clases (De 8 pm a 6 am), vacaciones y fines de semana. Por otro lado el clúster Leac y el Giib son clúster dedicados, los cuales están todo el tiempo disponibles para realizar cálculos y desplegar las diferentes aplicaciones que se les sean asignadas.

³² Supercomputación y cálculo científico UIS-SC3: <http://sc3.uis.edu.co/index.php/Infraestructura>

Cada servidor cuenta con servicios de administración y configuración de los nodos de cómputo, además de los servicios necesarios para el despliegue y monitoreo. Estos servidores cuentan a su vez con un conjunto de herramientas que permiten la interacción entre los diferentes servidores en la red, para proveer un servicio de cómputo distribuido.

Los nodos que se encuentran conectados a cada servidor poseen características diferentes, las cuales permiten que el usuario tenga la posibilidad de escoger entre lo que le sea más conveniente para desplegar su aplicación. A continuación se presentara una tabla con las características de los nodos en cada clúster

8.1.1 DESCRIPCION DE CADA CLUSTER Y SUS NODOS

En las siguientes tablas se especificaran las características de cada uno de los nodos pertenecientes a los diferentes clúster para mostrar el poder computacional con el que cuenta la plataforma. Además de presentar de una manera clara la homogeneidad y la heterogeneidad con la que cuentan cada uno de los clúster dependiendo de las características con las que cuenta cada nodo. Para la implementación del proyecto fue necesario el uso, la combinación y la interacción de cada uno de estos clúster, sean estos dedicados o ligeros.

CLUSTER LEAC

Referencia Nodos	Arquitectura	RAM Por Nodo	Tarjeta de Video	Cores por Nodo	GPUs por Nodo	Interfaz de Red	Cantidad de Nodos
DELL Poweredge 2850	3.60 GHz Intel Xeon	2 GB	ATI Radeon 7000	4	0	Gigabit Ethernet	1
DELL Precision T3400	2.66 GHz Intel Core 2 Quad	2 GB	nVidia Corporation Quadro FX 570	4	32	Gigabit Ethernet	1
DELL Poweredge 2950	2.50GHz Intel Xeon	2 GB	ATI ES1000	4	0	Gigabit Ethernet	1

Tabla 2: Descripción del clúster LEAC

Fuente:

Supercomputación y cálculo científico UIS-SC3:

<http://sc3.uis.edu.co/index.php/Infraestructura>

CLUSTER CMCENTIC

Referencia Nodos	Arquitectura	RAM por Nodo	Tarjeta de Video	Cores por Nodo	GPUs por Nodo	Interfaz de Red	Cantidad de Nodos
DELL OPTIPLEX GX620	3.8 GHz Intel pentium 4	2.0 GB DDR3	ATI RADEON X600	2	0	Gigabit Ethernet	32

Tabla 3 Descripción del clúster CMCENTIC

Fuente:

Supercomputación y cálculo científico UIS-SC3:

<http://sc3.uis.edu.co/index.php/Infraestructura>

CLUSTER CMJAVS

Referencia Nodos	Arquitectura	RAM Por Nodo	Tarjeta de Video	Cores por Nodo	GPUs por Nodo	Interfaz de Red	Cantidad de Nodos
DELL OPTIPLEX 740	2.4 GHz AMD Athlon 64X24600	2.0 GB DDR3	ATI RADEON XT2400	2	0	2 x Broadcom Gigabit Ethernet	24

Tabla 4: Descripción del clúster CMJAVS

Fuente:

Supercomputación y cálculo científico UIS-SC3:

<http://sc3.uis.edu.co/index.php/Infraestructura>

CLUSTER GIIB

Referencia Nodos	Arquitectura	RAM por Nodo	Tarjeta de Video	Cores por Nodo	GPUs Por Nodo	Interfaz de Red	Cantidad de Nodos
DELL	3.2 GHz Intel Pentium 4	512MB	-	2	0	Intel 82545GM Gigabit Ethernet	1
DELL	3.0 GHz Intel Pentium 4	512MB	-	2	0	Broadcom NetXtreme Gigabit Ethernet	1
DELL	1.8 GHz Intel Pentium 4	256MB	-	1	0	Intel 82540EM Gigabit Ethernet	1

Tabla 5: Descripción del clúster GIIB

Fuente:

Supercomputación y cálculo científico UIS-SC3:

<http://sc3.uis.edu.co/index.php/Infraestructura>

8.2 INTERACCION DEL USUARIO CON EL SISTEMA DE DESPLIEGUE DE APLICACIONES

En esta sección se mostrara de una manera sencilla pero detallada, la interacción que tiene el usuario con el sistema de despliegue de aplicaciones y como es también la interacción del administrador en el desarrollo e implementación de la plataforma. A continuación se presentara un diagrama de casos de uso con el fin de representar las diferentes funciones con las que cuenta cada uno.

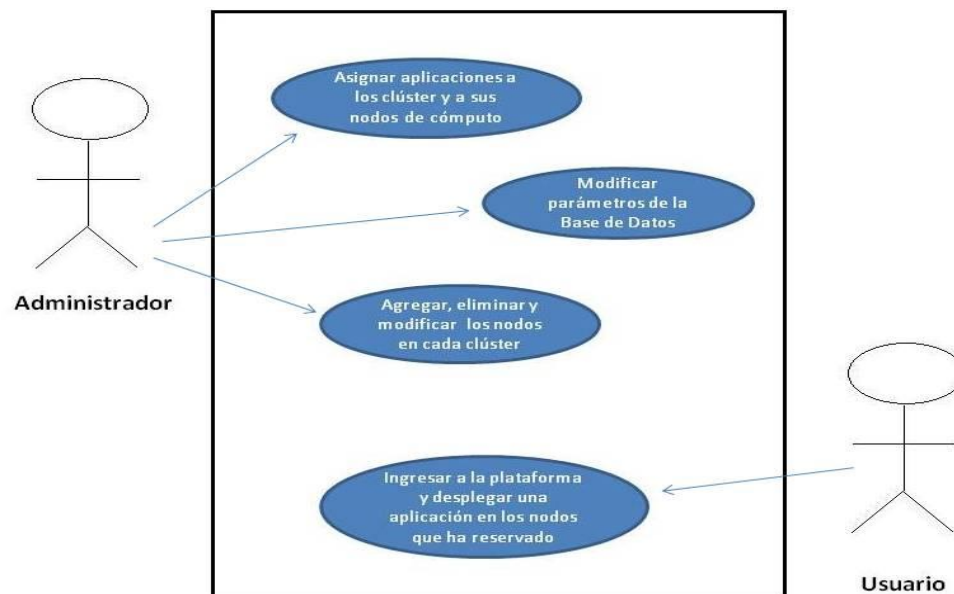


Ilustración 4: Diagrama de casos de Uso

El diagrama de casos de uso mostrado en la ilustración describe el comportamiento y la interacción del usuario y el administrador con la plataforma de cálculo distribuido presente en la universidad. El administrador y el usuario cuentan con diferentes privilegios al momento de hacer uso de la plataforma. A continuación se hará una descripción de los componentes con las cuales interactúa el usuario y con cuáles no.

COMPONENTES QUE INTERACTUAN CON EL USUARIO

Los componentes que interactúan con el usuario, son aquellos módulos a los cuales tiene acceso y con los cuales puede realizar operaciones para poder reservar nodos y desplegar aplicaciones. Estos son:

- Módulo de administración de aplicaciones: Permite agregar, editar y eliminar una aplicación de la plataforma.
- Modulo de administración de aplicaciones en los nodos: permite asignar, actualizar, eliminar aplicaciones asignadas a los nodos de cómputo.
- Modulo de despliegue de aplicaciones: es el que permite el despliegue de la aplicación en los nodos reservados.

COMPONENTES QUE NO INTERACTÚAN CON EL USUARIO

Los componentes que no interactúan con el usuario y los cuales parte de ellos son ejecutados por el administrador son aquellos como la asignación de las diferentes aplicaciones a cada uno de los nodos que se encuentren disponibles, modificar los diferentes parámetros en la base de datos, como su actualización y monitoreo. También es el encargado de agregar y eliminar los nodos en cada uno de los clúster si se encuentran con alguna falla. Algunos de los módulos con los cuales el usuario no interactúa son:

- Modulo de Consulta para aplicaciones de despliegue al momento del arranque: Este modulo recibe las peticiones del modulo cliente, realiza las respectivas consultas requeridas y envía información al módulo cliente.
- Modulo de integración con OAR: Permite restablecer el entorno de los recursos reservados, una vez la reservación termina.

- Modulo Cliente para despliegue de aplicaciones en caliente: permite la ejecución de los respectivos scripts de instalación.
- Modulo Cliente para despliegue de aplicaciones en el momento de arranque: permite la ejecución de los respectivos scripts de instalación al momento del arranque.

9. IMPLEMENTACION Y RESULTADOS

En esta sección se presentara de manera detallada como se realizo la implementación de la solución planteada, las pruebas que se hicieron y los resultados obtenidos de lo que fue el desarrollo del proyecto.

9.1 IMPLEMENTACION

Para la implementación de lo que sería la solución al problema expuesto y como se menciona anteriormente, se inicio con la creación de los dos clúster (CMCentic y el CMjavs) usando la herramienta ComputeMode. Durante la instalación y hasta su buen funcionamiento, fueron necesarias ciertas modificaciones y mejoras a esta herramienta:

9.1.1 MODIFICACIONES A COMPUTEMODE

- Debido a que los clúster ligeros creados con ComputeMode tienen la característica de crear una subred con número de ip 172.28.255.X, esto conlleva a que los nodos de este clúster estén "aislados" de las otras redes, lo cual presento un gran problema para la implementación del proyecto. Esto debido a que impedía la comunicación de este tipo de clúster con un servidor central. Para solucionar este inconveniente fue necesario crear reglas iptables como:

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Esta regla permite el paso de paquetes a través del servidor de ComputeMode hacia afuera de dicha subred. Otra regla iptables que se utilizó como parte de la solución del proyecto fue:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 2001 -j DNAT --to 172.28.255.10:22
```

Su propósito es permitir conexiones ssh desde fuera de la subred enrutando los paquetes que vengan por un puerto predefinido a un nodo específico logrando la comunicación del frontend central de OAR con los nodos que antes eran inalcanzable o aislados.

- **Se modificó el script de inicio por red:**

ComputeMode tiene la opción de reiniciar los nodos que se encuentren en el sistema operativo Windows a través de la utilización de un script. Este script consiste en hacer una petición al servidor de computeMode, logrando obtener el estado del nodo que se encuentra en Windows, si este nodo esta en hora de computo el script reiniciara el nodo permitiendo el inicio por red de dicho nodo. El inconveniente que se tuvo acerca de este script es que no estaba planteado para computadores que contengan más de una tarjeta de red, esto llevaba a que el script fallara. La solución sobre este pequeño inconveniente fue modificar dicho script de tal forma que haga las peticiones hacia el servidor por las dos tarjetas de red con sus respectivas Mac. A continuación se mostrara la parte del código que fue necesario agregar al script para que sirviera cuando los computadores tuvieron 2 o más tarjetas de red.

```
:SETMAC
```

```
rem if "%MAC%"==" " set MAC=%1
```

```
cm_wget -q -O-
```

```
"http://%CMSERVER%/cm/engine/automaton.php?mac=%1&bmid=1&event=Local  
Idle&os=windows" > %TEMP%\pxetemp
```

```
type %TEMP%\pxetemp | findstr "<action>BootCM</action>" > NUL
```

```
if not ERRORLEVEL 1 set MAC=%1
```

```
goto:EOF
```

Al contar con estos 4 clúster montados dentro del campus universitario se tenía una infraestructura como la mostrada en la siguiente grafica.

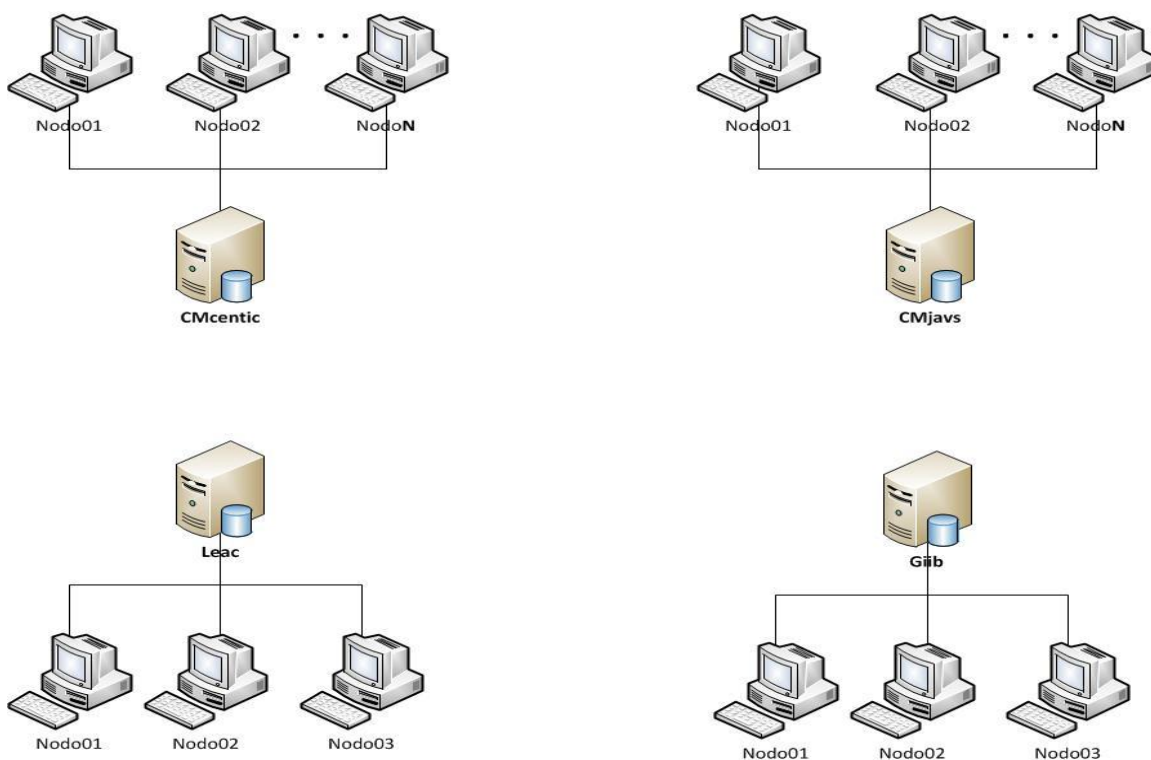


Ilustración 5: Diseño de los 4 clúster ubicados en el campus Universitario

Debido a que cada clúster contaba con su propia base de datos y su sistema de monitoreo, fue necesario centralizar la base de datos en una sola y a su vez centralizar los respectivos usuarios, sus cuentas y sus directorios de trabajo.

9.1.2 CENTRALIZACION DE LOS USUARIOS Y DE LA BASE DE DATOS

Para la Centralización de la Base de Datos y los diferentes usuarios, se utilizaron dos protocolos muy importantes que permitieron que se centralizara todo en una sola base de datos. Estos protocolos son el NFS (Sistema de Archivos de Red) y el LDAP (Protocolo Ligero de Acceso a Directorios). El primero es un protocolo de nivel de aplicación, según el Modelo OSI, es también un servicio de red que permite a un ordenador cliente montar y acceder a un sistema de archivos (en concreto, un *directorio*) remoto, exportado por otro computador servidor. Por otro lado el protocolo LDAP es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red, organiza la información en un modo jerárquico usando directorios. Estos directorios pueden almacenar una gran variedad de información y se pueden incluso usar de forma similar al Servicio de información de red (NIS), permitiendo que cualquiera pueda acceder a su cuenta desde cualquier máquina en la red acreditada con LDAP. La siguiente ilustración representa el funcionamiento de estos protocolos.

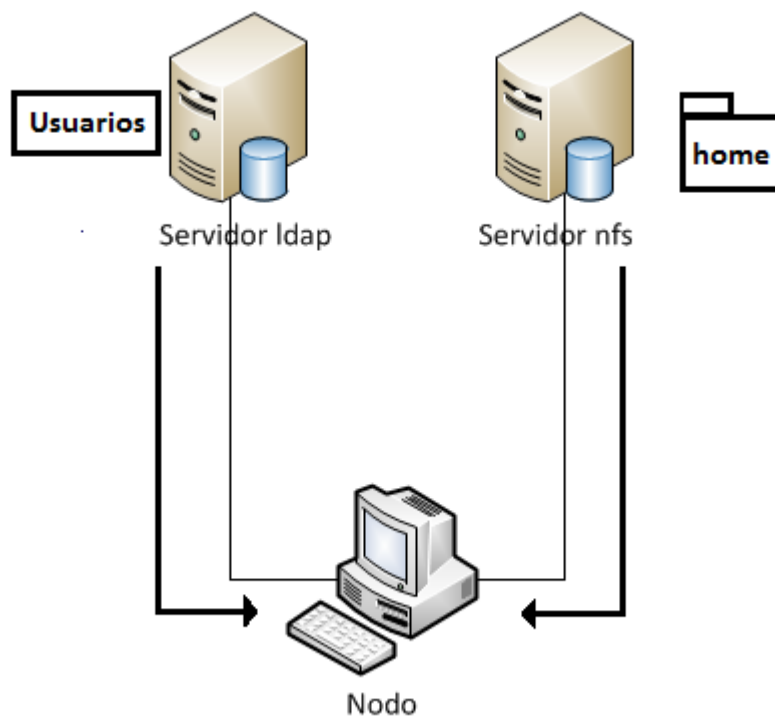


Ilustración 6: Representación del Protocolo NFS y LDAP

En base y con ayuda de estos protocolos se realizó lo que sería la centralización de los usuarios y de sus directorios, con esto resuelto se tenía en una sola base de datos toda la información necesaria para poder correr aplicaciones en los diferentes clúster. Por medio del sistema de monitoreo (Monika), se podía observar de manera gráfica, por medio de la página web, cuáles recursos estaban disponibles, cuáles estaban siendo usados y cuáles estaban usando en ese momento cada usuario. Además de esto se mostraban los 4 clúster que se encontraban dentro de la infraestructura, los nodos correspondientes a cada uno y las diferentes características con las que contaban. Tener esta centralización no solucionaba el problema que se tenía al no existir una interacción entre los clúster, lo cual no permitía que los usuarios pudieran escoger sus recursos sin importar a que red estuvieran conectados.

La ilustración 7, muestra un pantallazo del sistema de monitoreo (Monika), con los respectivos nodos y muestra además cuáles están siendo usados, cuáles están libres y cuáles están caídos.

OAR SC3-Cluster nodes

<i>default summary</i>			
	Free	Busy	Total
network_address	2	7	62
resource_id	7	17	129

Reservations:

nodo01-CMcentic.uis.edu.co	Absent	Absent	nodo01-CMiavs.uis.edu.co	Absent	Absent	nodo01-qiib.uis.edu.co	Free	Free	nodo01-jeac.uis.edu.co	8402	8402	8402	8402
nodo02-CMcentic.uis.edu.co	Absent	Absent	nodo02-CMiavs.uis.edu.co	Absent	Absent	nodo02-qiib.uis.edu.co	Free	Free	nodo02-jeac.uis.edu.co	8221	8221	8221	8221
nodo03-CMcentic.uis.edu.co	Absent	Absent	nodo03-CMiavs.uis.edu.co	Absent	Absent	nodo03-qiib.uis.edu.co	Absent		nodo03-jeac.uis.edu.co	8433	Free	Free	Free
nodo04-CMcentic.uis.edu.co	Absent	Absent	nodo04-CMiavs.uis.edu.co	Absent	Absent	nodo05-CMcentic.uis.edu.co	Absent	Absent	nodo05-CMiavs.uis.edu.co	Absent	Absent		
nodo06-CMcentic.uis.edu.co	Absent	Absent	nodo06-CMiavs.uis.edu.co	Absent	Absent	nodo07-CMcentic.uis.edu.co	Absent	Absent	nodo07-CMiavs.uis.edu.co	Absent	Absent		
nodo08-CMcentic.uis.edu.co	Absent	Absent	nodo08-CMiavs.uis.edu.co	Absent	Absent	nodo09-CMcentic.uis.edu.co	Absent	Absent	nodo09-CMiavs.uis.edu.co	Absent	Absent		
nodo10-CMcentic.uis.edu.co	Absent	Absent	nodo10-CMiavs.uis.edu.co	Absent	Absent	nodo11-CMcentic.uis.edu.co	Absent	Absent	nodo11-CMiavs.uis.edu.co	Absent	Absent		
nodo12-CMcentic.uis.edu.co	Absent	Absent	nodo12-CMiavs.uis.edu.co	Absent	Absent	nodo13-CMcentic.uis.edu.co	Absent	Absent	nodo13-CMiavs.uis.edu.co	Absent	Absent		
nodo14-CMcentic.uis.edu.co	Absent	Absent	nodo14-CMiavs.uis.edu.co	Absent	Absent	nodo15-CMcentic.uis.edu.co	Absent	Absent	nodo15-CMiavs.uis.edu.co	Absent	Absent		
nodo16-CMcentic.uis.edu.co	Absent	Absent	nodo16-CMiavs.uis.edu.co	Absent	Absent	nodo17-CMcentic.uis.edu.co	Absent	Absent	nodo17-CMiavs.uis.edu.co	Absent	Absent		
nodo18-CMcentic.uis.edu.co	Absent	Absent	nodo18-CMiavs.uis.edu.co	Absent	Absent	nodo19-CMcentic.uis.edu.co	Absent	Absent	nodo19-CMiavs.uis.edu.co	Absent	Absent		
nodo20-CMcentic.uis.edu.co	Absent	Absent	nodo20-CMiavs.uis.edu.co	Absent	Absent	nodo21-CMcentic.uis.edu.co	Absent	Absent	nodo21-CMiavs.uis.edu.co	Absent	Absent		
nodo22-CMcentic.uis.edu.co	Absent	Absent	nodo22-CMiavs.uis.edu.co	Absent	Absent	nodo23-CMcentic.uis.edu.co	Absent	Absent	nodo23-CMiavs.uis.edu.co	Absent	Absent		
nodo24-CMcentic.uis.edu.co	Absent	Absent	nodo24-CMiavs.uis.edu.co	Absent	Absent	nodo25-CMcentic.uis.edu.co	Absent	Absent	nodo26-CMcentic.uis.edu.co	Absent	Absent		
nodo27-CMcentic.uis.edu.co	8448	8449	nodo28-CMcentic.uis.edu.co	8446	8447	nodo29-CMcentic.uis.edu.co	Absent	Absent	nodo30-CMcentic.uis.edu.co	Absent	Absent		
nodo31-CMcentic.uis.edu.co	8444	8445	nodo32-CMcentic.uis.edu.co	8442	8443								

?: Running job but suspected resources.

Ilustración 7: Representación del sistema de Monitoreo (Monika) Por nodos

La siguiente ilustración representa de una manera más detallada por medio del Monika, los 4 clúster con sus respectivos nodos, y el monitoreo respectivo.

OAR SC3-Cluster nodes

default summary			
	Free	Busy	Total
network_address	2	7	62
resource_id	7	17	129

Reservations:

Reservations for property cluster=CMcentic:

nodo01-CMcentic.uis.edu.co	Absent	Absent	nodo02-CMcentic.uis.edu.co	Absent	Absent	nodo03-CMcentic.uis.edu.co	Absent	Absent	nodo04-CMcentic.uis.edu.co	Absent	Absent
nodo05-CMcentic.uis.edu.co	Absent	Absent	nodo06-CMcentic.uis.edu.co	Absent	Absent	nodo07-CMcentic.uis.edu.co	Absent	Absent	nodo08-CMcentic.uis.edu.co	Absent	Absent
nodo09-CMcentic.uis.edu.co	Absent	Absent	nodo10-CMcentic.uis.edu.co	Absent	Absent	nodo11-CMcentic.uis.edu.co	Absent	Absent	nodo12-CMcentic.uis.edu.co	Absent	Absent
nodo13-CMcentic.uis.edu.co	Absent	Absent	nodo14-CMcentic.uis.edu.co	Absent	Absent	nodo15-CMcentic.uis.edu.co	Absent	Absent	nodo16-CMcentic.uis.edu.co	Absent	Absent
nodo17-CMcentic.uis.edu.co	Absent	Absent	nodo18-CMcentic.uis.edu.co	Absent	Absent	nodo19-CMcentic.uis.edu.co	Absent	Absent	nodo20-CMcentic.uis.edu.co	Absent	Absent
nodo21-CMcentic.uis.edu.co	Absent	Absent	nodo22-CMcentic.uis.edu.co	Absent	Absent	nodo23-CMcentic.uis.edu.co	Absent	Absent	nodo24-CMcentic.uis.edu.co	Absent	Absent
nodo25-CMcentic.uis.edu.co	Absent	Absent	nodo26-CMcentic.uis.edu.co	Absent	Absent	nodo27-CMcentic.uis.edu.co	8448	8449	nodo28-CMcentic.uis.edu.co	8446	8447
nodo29-CMcentic.uis.edu.co	Absent	Absent	nodo30-CMcentic.uis.edu.co	Absent	Absent	nodo31-CMcentic.uis.edu.co	8444	8445	nodo32-CMcentic.uis.edu.co	8442	8443

Reservations for property cluster=CMjavs:

nodo01-CMjavs.uis.edu.co	Absent	Absent	nodo02-CMjavs.uis.edu.co	Absent	Absent	nodo03-CMjavs.uis.edu.co	Absent	Absent	nodo04-CMjavs.uis.edu.co	Absent	Absent
nodo05-CMjavs.uis.edu.co	Absent	Absent	nodo06-CMjavs.uis.edu.co	Absent	Absent	nodo07-CMjavs.uis.edu.co	Absent	Absent	nodo08-CMjavs.uis.edu.co	Absent	Absent
nodo09-CMjavs.uis.edu.co	Absent	Absent	nodo10-CMjavs.uis.edu.co	Absent	Absent	nodo11-CMjavs.uis.edu.co	Absent	Absent	nodo12-CMjavs.uis.edu.co	Absent	Absent
nodo13-CMjavs.uis.edu.co	Absent	Absent	nodo14-CMjavs.uis.edu.co	Absent	Absent	nodo15-CMjavs.uis.edu.co	Absent	Absent	nodo16-CMjavs.uis.edu.co	Absent	Absent
nodo17-CMjavs.uis.edu.co	Absent	Absent	nodo18-CMjavs.uis.edu.co	Absent	Absent	nodo19-CMjavs.uis.edu.co	Absent	Absent	nodo20-CMjavs.uis.edu.co	Absent	Absent
nodo21-CMjavs.uis.edu.co	Absent	Absent	nodo22-CMjavs.uis.edu.co	Absent	Absent	nodo23-CMjavs.uis.edu.co	Absent	Absent	nodo24-CMjavs.uis.edu.co	Absent	Absent

Reservations for property cluster=glib:

nodo01-glib.uis.edu.co	Free	Free	nodo02-glib.uis.edu.co	Free	Free	nodo03-glib.uis.edu.co	Absent
------------------------	------	------	------------------------	------	------	------------------------	--------

Reservations for property cluster=leac:

nodo01-leac.uis.edu.co	8402	8402	8402	8402	nodo02-leac.uis.edu.co	8221	8221	8221	8221	nodo03-leac.uis.edu.co	8433	Free	Free	Free
------------------------	------	------	------	------	------------------------	------	------	------	------	------------------------	------	------	------	------

?: Running job but suspected resources.

Ilustración 8: Representación del sistema de Monitoreo (Monika), por cada clúster

Fue necesario modificar cuatro librerías de OAR, para poder saltar las redes a través de SSH y que con ayuda de las modificaciones de iptables se lograra la interacción entre los recursos disponibles en las diferentes redes.

9.1.3 MODIFICACION DE LIBRERIAS

/usr/lib/oar/oar_iolib.pm

```
sub get_node_property($$$){
    my $dbh = shift;
    my $hostname = shift;
    my $property = shift;
    my $sth = $dbh->prepare(" SELECT $property
FROM resources
WHERE
network_address = '$hostname' AND cpuset=0
");
    $sth->execute();
    my @res = ();
    while (my $results = $sth->fetchrow_hashref){
        push(@res, $results->{$property});
    }
    $sth->finish();
    return(@res);
}
```

Descripción:

Esta librería de oar, contiene todas las funciones que interactúan con las base de datos. Dicha función antes nombrada fue creada con el fin de obtener una propiedad de un nodo específico. Además se creó un parámetro Port: # puerto (2022); para asignarle un puerto a cada nodo de tal forma que la función creada en el archivo oar_iolib leyera el puerto de la base de datos.

/usr/lib/oar/bipbip

```
my $dbh = iolib::connect_ro();
$host_to_connect_via_ssh=@hosts[0];
my @port2 = iolib::get_node_property($dbh,$host_to_connect_via_ssh,"port");
if(@port2[0] ne ""){
  my($ssh_cmd) = split(" ",$Openssh_cmd);
  $ssh_cmd = $ssh_cmd . " -p ";
  foreach my $p (@port2){
    $ssh_cmd = $ssh_cmd . $p;
  }
  $Openssh_cmd=$ssh_cmd;
}
```

Descripción:

Es una de las librerías más importantes de oar, la cual se encarga de manejar los procesos de conexión entre los nodos y el servidor verificando la existencia de carpetas, llaves y todos los requisitos previos para que estas conexiones sean exitosas. La función creada logra cambiar el puerto por defecto de conexión por uno definido en la base de datos logrando establecer la conexión por puertos predefinidos por el administrador.

/usr/lib/oar/oarsub

```
my $dbh = iolib::connect_ro();
my @port =iolib::get_node_property($dbh,$host_to_connect_via_ssh,"port");
$cmd[$i] = $cmd_name;$i++;
if(@port[0]=="" ){
  foreach my $p (@cmd_opts){
    $cmd[$i] = $p;$i++;
  }
}
```

```

}
else{
    $openssh_cmd=$cmd_name." -p ";
    $cmd[$i] = "-p";$i++;
    foreach my $p (@port){
        $cmd[$i] = $p;
    }
}
}
}

```

Descripción:

Oarsub es una instrucción por medio de la cual se hace la reserva de los nodos deseados. El código creado anterior permitió hacer dicha reserva por un puerto predefinido por el administrador.

/usr/lib/oar/finaud

```

my @node_list_tmp = iolib::get_finaud_nodes($base);
my %Nodes_hash;
foreach my $i (@node_list_tmp){
    if (($i->{cl})==4){
        $Nodes_hash{$i->{network_address}} = $i;
    }
}
}

```

Descripción:

Demonio de oar encargado de examinar el estado de los nodos. Agregando el código anterior se logro la preselección de los nodos a examinar, permitiendo que los diferentes servidores oar solo inspeccionen los nodos que se encuentran la misma red.

9.2 PRUEBAS Y RESULTADOS

9.2.1 POV-Ray

POV-Ray (Persistence of Vision Raytracer) es un programa para representar y crear imágenes foto-realistas mediante una técnica de renderizado llamada trazado de rayos. La escena se define en un archivo de texto con una sintaxis similar a C y en este archivo se definen la cámara, las luces y los objetos que se desean representar. [3][20]

El trazado de rayos (ray tracing) es una técnica para representar escenas simulando el recorrido que hacen los rayos que salen de las fuentes de luz, chocan con los objetos y estos reflejan o/y refractan el rayo. Los rayos que llegan a la cámara forman la imagen. En la práctica se realiza el trazado inverso, es decir, solo se tienen en cuenta los rayos que llegan a la cámara. Los algoritmos de trazado de rayos consiguen de esta forma representar los efectos de sombras, trasparencias, reflejos y refracción. [20]

A continuación se mostrara una imagen la cual ha sido renderizada, utilizando los nodos pertenecientes a los diferentes clúster dentro de la plataforma del grupo UIS-SC3. Para renderizar esta imagen se usó varios nodos escogidos de diferentes clúster y se fue variando para sacar conclusiones acerca de la implementación, del diseño y de la estrategia escogida para el proyecto

PEBBLES



Ilustración 9: Imagen obtenida por medio de POVRAY, Pebbles

Fuente:

PovRay: <http://hof.povray.org/>

La Imagen ilustra rocas de diversos colores y tamaños, creada por Jonathan Hunt (2008) la cual se encuentra entre las pocas imágenes que conforman el Hall of Fame (salón de la fama) oficial de povray.

La siguiente imagen representa el tiempo de cómputo que gasta el renderizado de la imagen anterior (Pebbles) de tamaño 640x480 pixeles, la cual es dividida en la cantidad de procesos como nodos a computar. La prueba se realizó cinco veces midiendo el tiempo promedio de cada combinación, es decir, al usar nodos heterogéneos en diferente red o nodos homogéneos en diferente red y en la misma. Observar la grafica y la tabla para mayor comprensión.

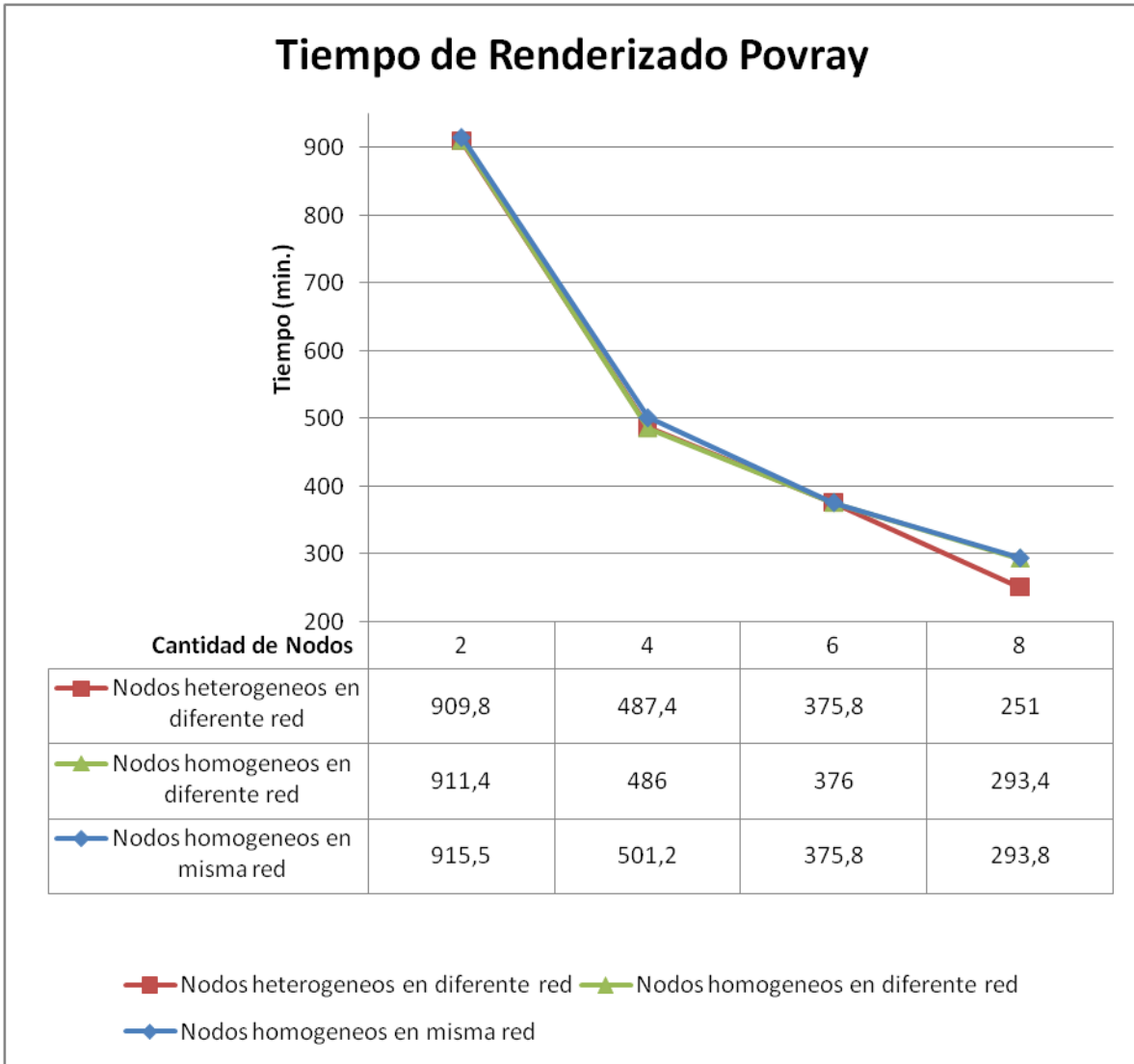


Ilustración 10: Descripción del Renderizado De la imagen Pebbles

Observando los resultados de la ilustración 10, podemos ver como el tiempo de cómputo va disminuyendo a medida que se aumenta el número de nodos usados en la renderización permitiendo que el procesamiento sea más rápido.

Con los resultados obtenidos, se prosiguió a hallar la varianza y la media de cada una de las combinaciones para así tener más certeza de los datos que se obtuvieron; la tabla siguiente representa estos valores:

Cantidad de Nodos	Media	Intervalo de variabilidad
Nodos heterogéneos en diferente red	506	258.28 < x < 753.74
Nodos homogéneos en diferente red	516.7	278.76 < x < 754.63
Nodos homogéneos en misma red	521.45	282.40 < x < 760.49

Tabla 6: Media e intervalo de variabilidad de la Prueba PovRay

Observando los valores arrojados por medio de la media hallada y del intervalo de variabilidad se puede concluir que así los nodos utilizados sean homogéneos y se encuentren en diferente red, los tiempos de renderizado son muy parecidos a los que se muestran aplicando el proyecto en la plataforma del SC3 sin afectar el desempeño del clúster. Por otro lado, los resultados obtenidos con nodos heterogéneos en diferente red muestra una mejoría usando 8 nodos, esto se debe a que por la misma heterogeneidad hay ciertos nodos en los cuales se presentan un mayor número de cores que otros. Estas características permiten que estos nodos sean más rápidos y que de esta manera se reduzca el tiempo de cómputo considerablemente respecto a las otras pruebas.

9.2.2 IPERF

IPerf es una herramienta de software para medir el ancho de banda disponible en un canal de datos que utilice el Protocolo de Internet (IP), proporciona información como la tasa de transferencia de datagramas en la red, el retardo (jitter) y la pérdida de paquetes. Esta herramienta resulta útil para todo tipo de aplicaciones de red independientemente del protocolo de comunicaciones usado, permitiendo especificar el host, puerto, protocolo TCP o UDP. Además de ser una herramienta

cliente-servidor, lo que permite medir el ancho de banda y el rendimiento entre dos host. [16][17]

Al tratarse de una herramienta cliente-servidor, se ejecutara Iperf en dos máquinas. Una hará de Servidor y la otra de Cliente.

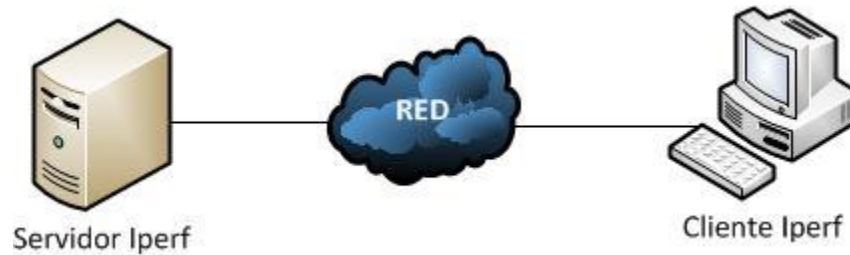


Ilustración 11: Representación de la herramienta Iperf

A continuación se mostraran los resultados obtenidos al aplicarle esta prueba a la arquitectura propuesta en el proyecto. Los estados usados en la prueba fueron:

Enviando Trabajo: Se mide en el momento que se hace la petición a través de OAR con el fin de enviar un trabajo a un nodo específico.

Ocupado: se mide cuando el nodo se encuentra ejecutando un trabajo

Libre: Se mide cuando el nodo esta sin ningún trabajo (job).

Para la realización de esta prueba se utilizó un intervalo de confianza de 95% el cual arrojó los siguientes resultados en los intervalos del promedio y la varianza.

Estado	Media	Intervalo de Variabilidad
Enviando trabajo	94.41	$91.61 < x < 97.21$
Ocupado	94,53	$91.34 < x < 97.71$
Libre	94.6	$91.31 < x < 97.88$

Tabla 7: Media e intervalo de variabilidad de la Prueba Iperf

Cada prueba se realizó 5 veces con el fin de conseguir un promedio acertado en los datos tomados. En base en los resultados obtenidos en cada uno de los estados, se puede observar que el promedio entre los mismos no difiere mucho uno del otro, lo cual es reafirmado con el intervalo de variabilidad siendo este no muy significativo. La siguiente ilustración muestra el ancho de banda disponible según los estados (enviando trabajo, ocupado, libre).

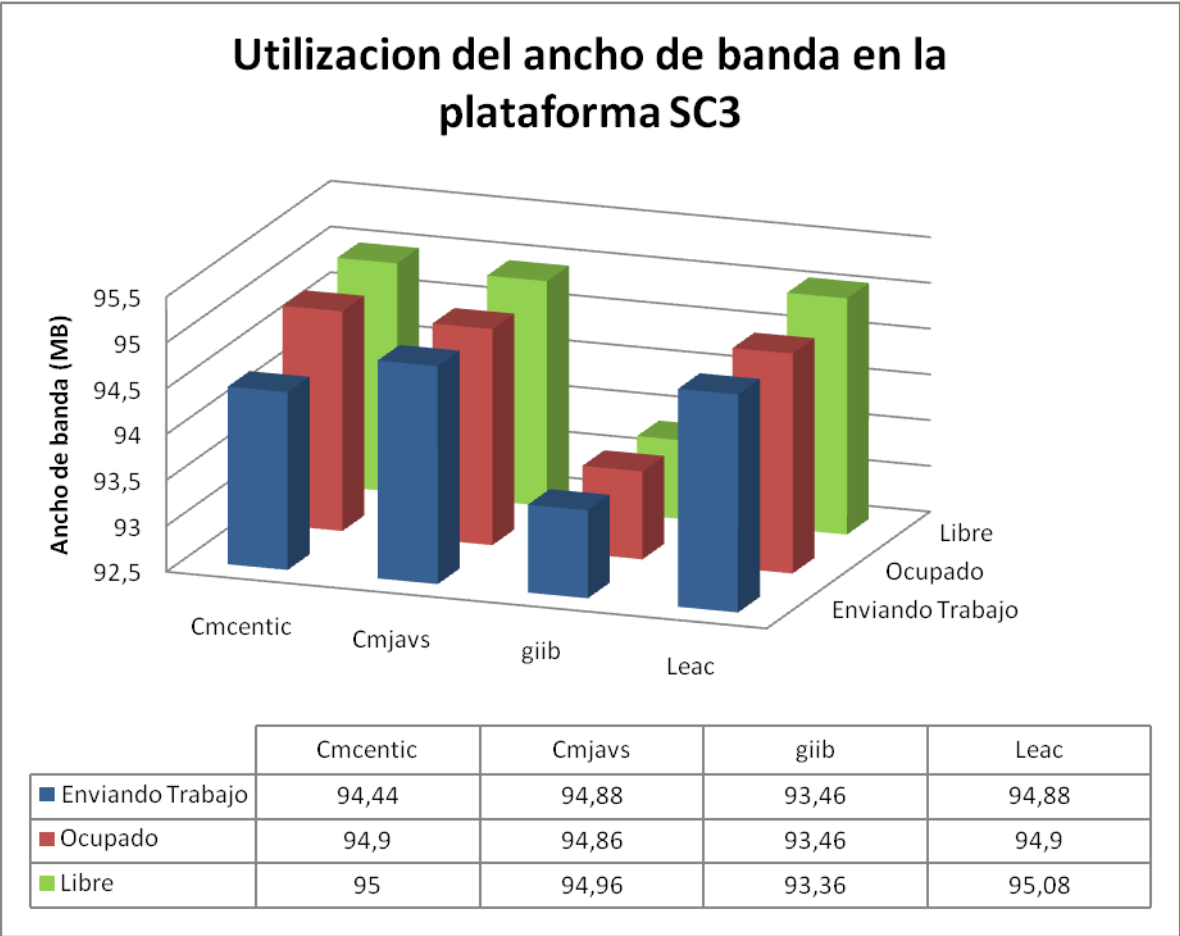


Ilustración 12: Resultados obtenidos con la prueba IPERF

Con esta prueba se puede observar que a pesar de la utilización de recursos de red en la plataforma esta no altera de manera significativa el proceso de las mismas, es decir, los cambios entre los estados Ocupado y Enviando son insignificantes con el ancho de banda que usa el estado Libre.

9.2.3 USO DE LA PLATAFORMA

La ilustración 13 muestra unos trabajos corriendo en los nodos 02 y 03 del clúster Leac, un trabajo en el nodo 02 del clúster Giib, 4 nodos trabajando en el clúster CMcentic y en el clúster Cmjavs no se están corriendo trabajos en este momento. Los nodos que aparecen en azul absent, se encuentran en mantenimiento en el instante en que fue tomada la imagen.



Ilustración 13: Representación de la reserva de Nodos

En la ilustración 14 se puede observar la reserva de una gran cantidad de nodos, que se encuentran desplegando una aplicación asignada previamente y demuestra como se ha desplegado dicha aplicación en nodos que se encuentran

en diferente red. Con esta imagen se pretende mostrar la implementación de la solución a la problemática planteada y su efectividad.

Reservations:

Reservations for property cluster=CMcentic:

nodo01-CMcentic.uls.edu.co	8956	8956	nodo02-CMcentic.uls.edu.co	8956	8956	nodo03-CMcentic.uls.edu.co	8956	8956	nodo04-CMcentic.uls.edu.co	8956	8956
nodo05-CMcentic.uls.edu.co	8956	8956	nodo06-CMcentic.uls.edu.co	8956	8956	nodo07-CMcentic.uls.edu.co	8956	8956	nodo08-CMcentic.uls.edu.co	8956	8956
nodo09-CMcentic.uls.edu.co	8956	8956	nodo10-CMcentic.uls.edu.co	8956	8956	nodo11-CMcentic.uls.edu.co	8956	8956	nodo12-CMcentic.uls.edu.co	8956	8956
nodo13-CMcentic.uls.edu.co	8956	8956	nodo14-CMcentic.uls.edu.co	8956	8956	nodo15-CMcentic.uls.edu.co	8956	8956	nodo16-CMcentic.uls.edu.co	8956	8956
nodo17-CMcentic.uls.edu.co	8956	8956	nodo18-CMcentic.uls.edu.co	8956	8956	nodo19-CMcentic.uls.edu.co	8956	8956	nodo20-CMcentic.uls.edu.co	8956	8956
nodo21-CMcentic.uls.edu.co	8956	8956	nodo22-CMcentic.uls.edu.co	8956	8956	nodo23-CMcentic.uls.edu.co	8956	8956	nodo24-CMcentic.uls.edu.co	8956	8956
nodo25-CMcentic.uls.edu.co	8956	8956	nodo26-CMcentic.uls.edu.co	8956	8956	nodo27-CMcentic.uls.edu.co	8942	8943	nodo28-CMcentic.uls.edu.co	8940	8941
nodo29-CMcentic.uls.edu.co	Absent	Absent	nodo30-CMcentic.uls.edu.co	Absent	Absent	nodo31-CMcentic.uls.edu.co	8938	8939	nodo32-CMcentic.uls.edu.co	8936	8937

Reservations for property cluster=CMjavs:

nodo01-CMjavs.uls.edu.co	8956	8956	nodo02-CMjavs.uls.edu.co	8956	8956	nodo03-CMjavs.uls.edu.co	8956	8956	nodo04-CMjavs.uls.edu.co	8956	8956
nodo05-CMjavs.uls.edu.co	8956	8956	nodo06-CMjavs.uls.edu.co	8956	8956	nodo07-CMjavs.uls.edu.co	Absent	Absent	nodo08-CMjavs.uls.edu.co	8956	8956
nodo09-CMjavs.uls.edu.co	8956	8956	nodo10-CMjavs.uls.edu.co	8956	8956	nodo11-CMjavs.uls.edu.co	8956	8956	nodo12-CMjavs.uls.edu.co	8956	8956
nodo13-CMjavs.uls.edu.co	8956	8956	nodo14-CMjavs.uls.edu.co	8956	8956	nodo15-CMjavs.uls.edu.co	8956	8956	nodo16-CMjavs.uls.edu.co	Absent	Absent
nodo17-CMjavs.uls.edu.co	8956	8956	nodo18-CMjavs.uls.edu.co	8956	8956	nodo19-CMjavs.uls.edu.co	8956	8956	nodo20-CMjavs.uls.edu.co	8956	8956
nodo21-CMjavs.uls.edu.co	8956	8956	nodo22-CMjavs.uls.edu.co	8956	8956	nodo23-CMjavs.uls.edu.co	8956	8956	nodo24-CMjavs.uls.edu.co	8956	8956

Reservations for property cluster=giib:

nodo01-giib.uls.edu.co	8950	8951	nodo02-giib.uls.edu.co	8956	8956	nodo03-giib.uls.edu.co	Absent
------------------------	------	------	------------------------	------	------	------------------------	--------

Reservations for property cluster=leac:

nodo01-leac.uls.edu.co	8956	8956	8956	8956	nodo02-leac.uls.edu.co	8515	8515	8515	8515	nodo03-leac.uls.edu.co	8891	8891	8891	8891
------------------------	------	------	------	------	------------------------	------	------	------	------	------------------------	------	------	------	------

*: Running job but suspected resources.

Ilustración 14: Representación de una gran cantidad de nodos reservados

10. LIMITACIONES DEL PROYECTO

- El no poder correr aplicaciones que estén realizadas con MPI limita el proyecto debido a que varias aplicaciones corren bajo esta plataforma. La estructura de MPI, está planteada para funcionar en clústers que se encuentren en el mismo segmento de red. Esto se debe a que MPI al no usar SSH es imposible que le puedan llegar trabajos a los nodos que se encuentran en una red privada.
- La no portabilidad al momento de implementar el proyecto, debido a que se requiere de conocimientos avanzados para su desarrollo y futuras modificaciones o mejoras.

11. CONCLUSIONES

- En base al análisis realizado para la solución a la limitación planteada en el proyecto, se logro diseñar e implementar una estrategia que implicó la modificación de varios archivos y librerías del OAR, además de la creación de reglas de Iptables que permitieron la interacción entre recursos que no se encuentran en la misma subred del servidor del clúster al cual pertenecen. Esto con el fin de hacer reservas y de poder desplegar diferentes aplicaciones por parte de los usuarios.
- Se estudiaron varios casos de interconectividad logrando caracterizar los diferentes tipos de solución y poder elegir de esta manera la estrategia más acertada para la implementación del proyecto.
- Se desarrollo un prototipo que permitió la interconexión y gestión de recursos de una infraestructura de cómputo distribuido en redes heterogéneas, el cual aprobó que se diera paso a la implementación de la estrategia planteada y que sirviera de base para realizar las diferentes modificaciones y pruebas para aprobar su efectividad.
- Se demostró en base a las pruebas realizadas a la estrategia implementada que el usar nodos heterogéneos que estén en diferentes redes, permite que se obtenga un mejor tiempo de computo, en las diferentes aplicaciones que se corran en la plataforma y que es eficiente la interacción entre estos recursos que en los recursos que se encuentran en una misma red y exista una homogeneidad entre ellos. El escoger que tantos nodos se deben usar y que características tengan dependerá de que tan grande es el trabajo que se desea correr, para así lograr el mayor rendimiento de cada nodo.
- La creación de clústers ligeros es una alternativa muy llamativa al momento de querer utilizar una gran cantidad de recursos que la mayor parte del

tiempo se encuentran ociosamente para realizar cálculo distribuido. Además, existen herramientas como ComputeMode que provee una solución integral, permitiendo una fácil y rápida integración de diferentes recursos.

12. RECOMENDACIONES

- La creación e implementación de una herramienta web, que permita que el usuario pueda monitorear el estado de los diferentes recursos, logre lanzar trabajos, seleccionar nodos, escoger por número de cores de una manera más sencilla.
- Se recomienda que la agregación de nuevos nodos y clústers a la plataforma sea de una manera más transparente para el administrador al igual que la agregación de los nodos a la base de datos. Que sea más fácil el realizar estas acciones para el administrador, buscando la integridad de cada uno de los factores que interviene en la mejora de la plataforma como tal y del proyecto.

13.ANEXOS

A. Participación en el **CLCAR 2010** (Conferencia Latinoamericana de computación de alto rendimiento), **con la presentación de un Short Paper llamado STRATEGIES TO FAULT TOLERANCE AND HIERARCHICAL NETWORKS INTERACTION FOR LIGHTWEIGHT CLUSTERING**, el cual fue elegido como parte del evento y consignado en las memorias del mismo.

Strategies to Fault Tolerance and hierarchical networks interaction for Lightweight Clustering

Estrategias para el manejo de Fallas e interacción entre redes jerárquicas en clústers ligeros

Mantilla Serrano, M., Orostegui Prada , S., Uribe Espinosa, R., Ruiz Sanabria, C.,
Escobar Ramírez, J., Barrios Hernández, C ;
Universidad Industrial de Santander
{mireyitak,86nano, rosterg85,camilo1729}@gmail.com,
carlosjaimebh@computer.org, juancaes@uis.edu.co

Abstract

Lightweight clusters are infrastructure composed by “light” resources such as desktop computers or workstations. Management and implementation of lightweight clusters is provided by different frameworks as Compute mode [1,2]. Compute mode creates a distributed computing

infrastructure through the aggregation of unused computing resources.

However, it exists two critical situations: non-support to fault tolerance and not provide a management of the interaction among elements of different network levels.

In this work, we propose strategies to treat these situations, making possible the execution of longtime

jobs and a management of different networks using Computemode.

Keywords: Fault tolerance; Cluster Computing; Distributed systems

Palabras claves:

Tolerancia a fallas; Computación clúster; Sistemas distribuidos.

1. INTRODUCCIÓN

Los campus universitarios cuentan con gran cantidad de equipos de cómputo interconectados, con diversos sistemas operativos predefinidos y con políticas administrativas que no permiten modificar software o el estado de los recursos. Normalmente estos se encuentran distribuidos en aulas de clase que podrían utilizarse más allá de su uso tradicional, por ejemplo, en horarios nocturnos y fines de semana siendo posible construir Clústers Ligeros (extensibles a Grid Ligeros), para soportar cálculo científico.

Existen herramientas para crear clústers ligeros por agregación de recursos en forma no intrusiva como

lo es Computemode.

A pesar de las ventajas que ofrece Computemode se presenta dos limitaciones importantes: no existe un mecanismo de tolerancia a fallas que permita crear puntos de restauración (checkpoints) y en segundo lugar, esta plataforma limita los equipos de cómputo (PCs) a la subred del servidor.

De acuerdo a lo anterior y dada la naturaleza oportunista de estos tipos de infraestructura se hace necesario la creación de mecanismos de migración y checkpoints que permitan la ejecución de procesos de larga duración, así como el restablecimiento de las tareas o procesos que no llegaron a termino por fallas en la infraestructura.

En cuanto a la segunda limitación presentada, se hace oportuna la creación de una estrategia de interacción entre recursos distribuidos de una infraestructura de cómputo en redes heterogéneas.

En la siguiente sección se presenta el estado del arte. En la sección 3, se describe la propuesta general para tratar las dos problemáticas planteadas. Finalmente, presentamos en la sección 4 y 5 algunas conclusiones y los agradecimientos.

2. ESTADO DEL ARTE

La mayoría de los equipos de cómputo cuentan con una plataforma Windows como el sistema operativo residente para su uso diario. Para la implementación de clúster es común encontrar ambientes basados en distribuciones Linux, por esto su elección para la creación de (CLUSTER y GRID).

Computemode y PelicanHPC permiten encontrar solución a la creación de estas estructuras basadas en la tecnología diskless, pero no cuentan con un manejo de tolerancia a fallas ni permiten la integración de recursos que no se encuentren dentro de la misma subred del servidor. Para las limitaciones mencionadas anteriormente existen algunos

mecanismos de checkpoint y manejo de redes que trabajando en conjunto pueden ser una alternativa para mejorar el aprovechamiento de los equipos de cómputo.

Dentro de las herramientas de creación de checkpoint existen dos tipos, las que trabajan a nivel de usuario y las que lo hacen a nivel del kernel, a nivel de usuario se exige el cambio del programa y una recompilación del mismo, un ejemplo de este tipo es Condor³³ que trabaja a este nivel.

Berkeley Lab Checkpoint/Restart (BLCR)[3], trabaja a nivel de kernel y permite a los programas que se ejecutan en Linux realizar checkpoints que pueden ser usados durante el reinicio de la máquina e incluso en otra máquina diferente.

B-OAR[5] presenta un manejo de recursos distribuidos basado en OAR[4] que es un manejador de colas desarrollado con herramientas de alto nivel, las cuales ofrecen un buen rendimiento, además de ser

³³ <http://www.cs.wisc.edu/condor/overview>

liviano y de fácil uso por parte del usuario. Para gestión de recursos en computación voluntaria se usa BOINC como middleware, esto con el fin de utilizar las mejores características que ofrece cada una de estas plataformas.

Otra herramienta que se centra en el manejo de recursos distribuidos basada en OAR es CiGri [6] cuyo fin es reunir los recursos no utilizados de las infraestructuras de computo de alto rendimiento locales para que estén disponible para un gran conjunto de tareas. Los usuarios pueden supervisar y controlar su puesto de trabajo a través de un portal web. Además el sistema proporciona mecanismos de detección de fallas, permitiendo la ejecución completa de tareas de larga duración.

3. PROPUESTA

Para la implementación de una solución se escogió Computemode que a diferencia de PelicanHPC cuenta con un manejador de colas.

Se busca entonces abordar las

problemáticas que se muestran a continuación.

3.1 Mecanismo de tolerancia a fallas

En el momento de implementar Computemode se encuentran las fallas señaladas en la figura 1,

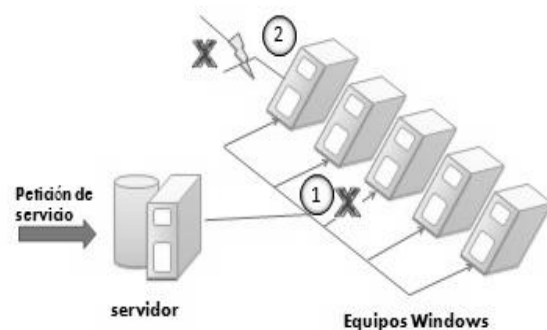


Figura 1 Fallas frecuentes infraestructura

Numeradas por 1 y 2, se puede observar la pérdida de un nodo por falla en la infraestructura de comunicación, en la red eléctrica o por el cambio de su estado para uso de Windows; es aquí que necesitamos la implementación de un mecanismo para la creación de checkpoints. Dentro de las alternativas para la implementación de dicho mecanismo podemos

encontrar Condor, el cual realiza un checkpoint a nivel de aplicación disminuyendo la sobrecarga, pero a costo de modificar el código de la aplicación. Por otro lado BLCR trabaja a nivel de kernel lo que da como resultado que se almacene completamente el estado del proceso, generando archivos de gran tamaño, lo cual ocasiona un aumento en la carga del sistema. Entre sus ventajas se encuentra permitir la creación de un mecanismo de restauración de trabajos transparente al usuario.

En el desarrollo de este trabajo se busca crear un módulo que se integre al manejador de colas, con el fin de que sea flexible y fácil de utilizar por parte del usuario. Al crear este módulo se tiene en cuenta que el usuario debe elegir el tiempo entre la creación de los checkpoints para que se acomoden al trabajo a ejecutar, ya que el tiempo de ejecución se puede ver seriamente degradado por la frecuencia de creación de estos. La integración de este mecanismo con el manejador de colas permitirá además crear una estrategia de migración de procesos, lo cual beneficiaría

aplicaciones paramétricas caracterizadas por un largo tiempo de ejecución

3.2 Interacción entre redes heterogéneas

Como se observa en la figura 2, no es posible la comunicación entre nodos de diferentes redes ya que estos se comunican a través de una red privada, que solo hace posible la comunicación de estos con su respectivo servidor. Debido a lo anterior se hace factible encontrar una forma en la cual se pueda utilizar recursos pertenecientes a diferentes servidores o subredes, un mecanismo que se podría utilizar sería el de cigri, el cual proporciona una manera transparente de aprovechamiento de varios clústers.

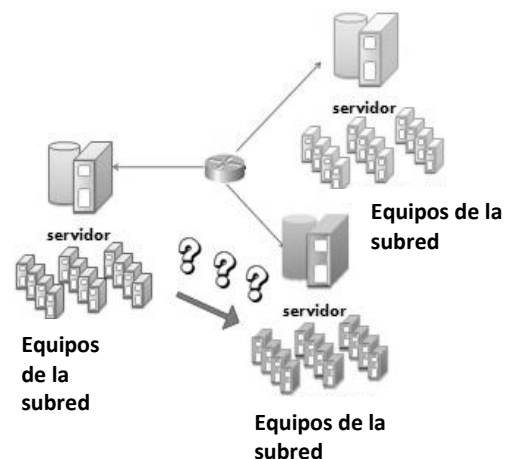


Figura 2 **Arquitectura distribuida**

Cigri está diseñado como un servidor central que se encarga de recibir y enviar trabajos a los diferentes clústers registrados en el sistema, en el cual el objetivo es utilizar los instantes de tiempo en los que los recursos del clúster están libres, creando una clase de trabajos llamados best-effort los cuales cuentan con una prioridad mínima y pueden ser cancelados por otros usuario del clúster. El objetivo de la propuesta es utilizar los recursos a plenitud y evitar que estos sean cancelados como se hace en Cigri, por lo tanto se propone una modificación para que estos sean tomados como trabajos propios del clúster, implementando estas funcionalidades dentro de un conjunto de scripts que extenderán las capacidades de Computemode para formar una infraestructura distribuida.

4. CONCLUSIONES

Teniendo en cuenta la cantidad de recursos que se pueden utilizar para hacer cálculo distribuido, la creación

de clúster ligeros es una alternativa altamente atractiva y se encuentra una amplia gama de plataformas que permiten la integración de recursos. Sin embargo, Computemode provee una solución integral que permite una fácil y rápida integración de recursos. Para lograr un mecanismo adecuado de tolerancia a fallas, es necesario identificar diferentes niveles de fallas y tener en cuenta la degradación en el desempeño. Esto tiene como consecuencia que ningún mecanismo de tolerancia a fallas es absoluto. Así mismo, cualquier mecanismo de tolerancia a fallas implementado, agrega costo computacional.

El mecanismo propuesto, tiene en cuenta fallas más comunes y el costo computacional es minimizado. Igualmente, la interacción con el usuario es transparente y provee información adecuada para desarrollar nuevos mecanismos externos.

En cuanto a la interacción de recursos distribuidos en redes de diferente nivel, existen diferentes soluciones que permiten esa

interacción, sin embargo ninguna ha sido implementada en Computemode.

Para aprovechar las oportunidades y ventajas de Computemode, observando las estrategias utilizadas en CiGri añadiendo la gestión de recursos distribuidos en redes de diferente nivel, es necesario tener en cuenta la prioridad de los trabajos. Por eso, nuestra propuesta busca igualar la prioridad de los trabajos.

5. AGRADECIMIENTOS

Los autores agradecen al equipo de COMPUTEMODE del LIG-Montbonnot-St Martin, Francia, dirigido por el profesor Olivier Richard.

6. REFERENCIAS

[1] Bruno Richard and Philippe Augerat, "Effective Aggregation of Idle Computing Resources for Cluster Computing", ERICM News No. 59, October 2004.

[2] Computemode Project:
<http://computemode.imag.fr>

[3] BLCR Frequently Asked Questions
<https://upc-bugs.lbl.gov/blcr/doc/html/FAQ.html#whatisblcr-Futur>

[4]Resource manager system for high performance computing.,
<http://oar.imag.fr/>

[5] Combining OAR with the power of volunteer computing through BOINC
<http://oar.imag.fr/works/BOAR.html>

[6] Cigri web site, <http://cigri.imag.fr/>

[7]Yiannis Georgiou, Olivier Richard and Nicolas Capit, "Evaluations of the light grid CIGRI upon the Grid5000 platform", Third IEEE International Conference on e-Science and Grid Computing, 2007

B. Participación igualmente al CLCAR 2010 por medio de un Poster alusivo al short paper elegido anteriormente, el cual se expuso en dicho evento.

Strategies to Fault Tolerance and Hierarchical Networks Interaction for Lightweight Clustering

M. Mantilla , S. Orostegui, R. Espinosa, C. Ruiz, J. Escobar and C. J. Barrios Hernandez
Universidad Industrial de Santander
Bucaramanga, Colombia
<http://sc3.uis.edu.co>

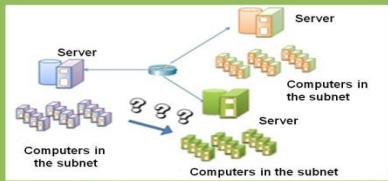
Abstract

Lightweight clusters are infrastructures composed by "light" resources such as desktop computers or workstations. Management and implementation of lightweight clusters is provided by different frameworks as Computemode(2). It creates a distributed computing infrastructure through the aggregation of unused computing resources.

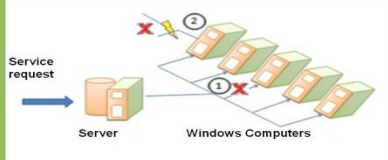
However, it exists two critical situations: nonsupport to fault tolerance and not provide a management of the interaction among elements of different network levels.

In this work, we propose strategies to treat these situations, making possible the execution of longtime jobs and a management of different networks using Computemode.


Description



Communication States



Failure State



Strategy

Conclusion and Further Work

- Implementation of Fault Tolerance Mechanism
 - Identification and Definition of Failures and Levels
 - Log's Management
- Implementation of hierarchical network interaction
 - Resources Identification
 - Network Levels Definitions
 - Resources' Scheduling (Based in CiGri)

References

(1) Bruno Richard and Philippe Augerat, "Effective Aggregation of Idle Computing Resources for Cluster Computing", ERICM News No. 59, October 2004.

(2) Computemode Project: <http://computemode.imag.fr>




(3) Resource manager system for high performance computing., <http://oar.imag.fr/>

(4) Cigri web site, <http://cigri.imag.fr/>

(5) Yiannis Georgiou, Olivier Richard and Nicolas Capit, "Evaluations of the light grid CIGRI upon the Grid5000 platform", Third IEEE International Conference on e-Science and Grid Computing, 2007

Aknowledgments

Mescal Team, ComputeMode and OAR project. Laboratoire d'Informatique de Grenoble, Grenoble Universités, France.

Anexo B: Poster Presentado en el CLCAR 2010

C. ASESORES LIGADOS AL PROYECTO

A continuación se hará una descripción de cada uno de los asesores que ayudaron en el desarrollo del proyecto y cuál fue su rol.

ANTONIO JOSE LOBO FIGUEROA

Ingeniero Electricista de la Universidad de los Andes, de Mérida en Venezuela, estudiante de la maestría informática e ingeniería en la Universidad Industrial de Santander (UIS) y es actualmente el ingeniero de soporte del centro de cálculo científico en el proyecto gridUIS-2. Su participación o rol en el desarrollo de este proyecto fue el del acoplamiento de la estrategia implementada con la plataforma del centro de computación y cálculo científico UIS - SC3.

CRISTIAN CAMILO RUIZ SANABRIA

Ingeniero de Sistemas de la Universidad Industrial de Santander y es actualmente estudiante de maestría en Ciencias de la Informática en la Universidad Joseph Fourier en Francia. Su rol en el progreso de este proyecto fue principalmente el empalme de lo que se había realizado anteriormente en el proyecto que desarrolló, presentándonos lo que implemento y ayudando en el entendimiento de la herramienta que usó.

14. BIBLIOGRAFIA

14.1 ARTICULOS Y PROYECTOS

1. Bruno Richard and Philippe Augerat, "Effective Aggregation of Idle Computing Resources for Cluster Computing", ERICM News No. 59, October 2004.
2. Carlos Moreno Losada, PLATAFORMA COMPUTING@HOME, 2008:
(Disponible en:
<http://www.recercat.net/bitstream/2072/13680/1/PFC+Carlos+Moreno+Losada.pdf>)
3. Cristian Camilo Ruiz Sanabria, ANALISIS E IMPLEMENTACION DE UNA INFRAESTRUCTURA DE CÁLCULO DISTRIBUIDO EN LA RED UNIVERSITARIA, Trabajo de grado , Universidad Industrial de Santander 2009
4. Mireya Carolina Mantilla Serrano, Sergio Andrés Orostegui Prada, Rosemberg José Uribe Espinosa, Cristian Camilo Ruiz Sanabria, Juan Carlos Escobar Ramírez, Carlos Jaime Barrios Hernández, STRATEGIES TO FAULT TOLERANCE AND HIERARCHICAL NETWORKS INTERACTION FOR LIGHTWEIGHT CLUSTERING, Short Paper, CLCAR (Conferencia Latinoamericana de computación de alto rendimiento) 2010, Página 333, isbn 978-85-7727-252-5
5. The Grid 2, Second Edition: Blueprint for a New Computing Infrastructure (The Elsevier Series in Grid Computing), Ian Foster (Editor), Carl Kesselman (Editor), Kaufmann Editors, USA 2004
6. Yiannis Georgiou, Olivier Richard and Nicolas Capit, "Evaluations of the light grid CIGRI upon the Grid5000 platform", Third IEEE International Conference on e-Science and Grid Computing, 2007

14.2 ENLACES

7. Algunos Proyectos en desarrollo bajo el sistema Data Grid, Textos científicos, 2007: (Disponible en: <http://www.textoscientificos.com/redes/computacion-grid/proyectos>)
8. Arquitecturas Paralelas, ITESCAM: (Disponible en: www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r21804.DOC)
9. Clúster informática, Buenas tareas, 2009: (Disponible en : <http://www.buenastareas.com/ensayos/Cluster-Informatica/48752.html>)
10. Combining OAR with the power of volunteer computing through BOINC: <http://oar.imag.fr/works/BOAR.html>
11. Computación distribuida, Ordenadores y portátiles. (Disponible en <http://www.ordenadores-y-portatiles.com/computacion-distribuida.html>)
12. Computación Grid, Textos científicos, 2007: <http://www.textoscientificos.com/redes/computacion-grid/>
13. Computación Grid, Ventajas y desventajas, Aplicaciones, Textos científicos, 2007: <http://www.textoscientificos.com/redes/computacion-grid/ventajas-desventajas-aplicaciones>
14. Computación Voluntaria, Extremadur at home, 2008: http://www.extremadurathome.org/extremadurathome/index.php?option=com_content&view=article&id=16&Itemid=16&lang=es
15. Computemode project: <http://computemode.imag.fr>

16. IPERF Midiendo ancho de banda entre dos hosts, Seguridad y Redes, 2008:
<http://seguridadyredes.nireblog.com/post/2008/06/18/iperf-midiendo-ancho-de-banda-entre-dos-hosts>

17. Medir la red con Iperf, Bytecoders, 2009: <http://bytecoders.net/content/medir-la-red-con-iperf.html>

18. OAR Documentation, Admin Guide:
http://oar.imag.fr/admins/admin_documentation.html

19. OAR project: <http://oar.imag.fr/>

20. POVRAY, Tutorial Introductorio de PovRay:
<http://www.tecnun.es/asignaturas/grafcomp/practicas/povray/povray1.pdf>

21. Redes informáticas, Medios físicos, Topologías de red, Wikilibros, 2010:
http://es.wikibooks.org/wiki/Redes_inform%C3%A1ticas/Medios_f%C3%ADsicos/Topolog%C3%ADas_de_red

22. Redes, Topologías de red, Slideshare, 2008 : (Disponible en
<http://www.slideshare.net/guest7bb5a1/redes-topologia-de-red>)

23. Página Oficial de Grupo SC3:
http://sc3.uis.edu.co/index.php/P%C3%A1gina_Principal

24. Topologías de red, Indianopedia: (Disponible en
http://lasindias.net/indianopedia/Topolog%C3%ADas_de_red)