Unidad 3 / Escenario 5

Lectura fundamental

Conceptos fundamentales de computación de alto desempeño

Contenido

1 Fundamentos de organizaciones computacionales en clúster
2 Computación de alto desempeño
3 Manejadores locales de recursos
4 Computación en malla (*grid computing* o *cluster* remoto)

Palabras clave: e-ciencia, computación de alto desempeño, manejador local de recursos, programación concurrente

Introducción

El crecimiento de la población mundial, según las Naciones Unidas, en 1950 era de 2600 millones, pero en 1990 alcanzó los 5300 millones. Es decir, en 40 años se duplicó y para el 2100 está previsto que la población llegue a los 11.200 millones de personas. Este crecimiento tan exagerado crea grandes problemas, grandes necesidades y, en consecuencia, se necesitan grandes soluciones.

Después de la Segunda Guerra Mundial fue claro que hacer cálculos manuales para alistar ángulos de disparo de los cañones no era funcional, así que el diseño de la máquina de Alan Turing dio pie al desarrollo de los computadores de cálculo para así generar tablas con ángulos de tiro de acuerdo con la distancia metro a metro.

Visto desde ese punto de vista, cuando la población mundial no superaba los 2500 millones, ya había procesos como el código enigma que requerían de una máquina para analizar cadenas de datos a gran escala y para descifrar los mensajes de la armada alemana de 1940.

Hoy el mismo proceso se desarrolla en la *National Security Agency* – SNA de los Estados Unidos, donde se analiza toda la información que cruza por Internet y que entra por EE.UU. Para desarrollar esa enorme tarea, se requiere de un mega super organización computacional en *cluster*, con muchos procesadores, de muchos cores y una gran cantidad de programadores de sistemas que sean capaces de programar en paralelo.

Así pues, la computación o las ciencias computacionales han sido de vital importancia en el desarrollo de la ciencia en el mundo. El uso de máquinas para correr simulaciones a gran escala es cada vez más popular y cada día se necesitan para más proyectos científicos en todo el mundo. Debido a esto las tecnologías han crecido de manera exponencial en las últimas décadas, dadas las condiciones del crecimiento mundial de la población, la globalización del mercado y las necesidades de tecnologías computacionales avanzadas distribuidas.

Los aspectos anteriores han llevado a la comunidad científica a crear e-ciencia, dentro de la cual se creó la rama de las tecnologías informáticas. La e-Ciencia de las tecnologías se refiere al estudio e investigación en tecnología informática para generar recursos de procesamiento de datos y almacenamiento de estos, al igual que todo lo relacionado con el trabajo colaborativo.

En términos coloquiales, e-ciencia es la transformación de la activad científica tradicional por la integración al quehacer de herramientas de comunicaciones y manejo de información, que en resumen se pueden llamar *cluster*.

1. Fundamentos de organizaciones computacionales en *clúster*

1.1. ¿Qué es un clúster?

El concepto de *clustring* fue forjado con base en la técnica que permite combinar múltiples sistemas organizados, de tal manera que puedan trabajar en paralelo, compartiendo recursos como la memoria, la red, los discos, y así poder desarrollar un conjunto enorme de tareas, siendo resistentes a los fallos y proveyendo disponibilidad continua.

Un *clúster* es una organización de computadores en racimo también conocido como granja de servidores y se define como un sistema computacional de procesamiento paralelo distribuido.

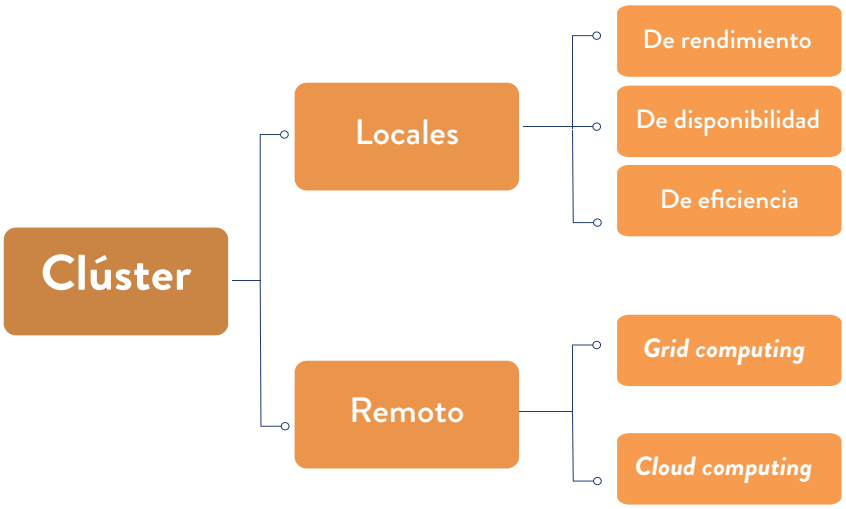

Figura 1. Tipos de clústeres
Fuente: elaboración propia

Un *clúster* local está conformado por un servidor máster y su *backup*, un conjunto de *switches* (concentradores redundantes), una red local privada, un arreglo de discos con NFS (*Network File System*), PFS (*Parallel File System*), Lustre, Orage, etc. Es un conjunto de servidores que se denominan nodos, los cuales son los que llevan a cabo el trabajo real, usando programación y procesamiento paralelo.

- **Lustre.** Es un *software* manejador de sistemas de archivos distribuidos en paralelos por red. Es un software libre desarrollado con base en PFS, es decir, es una versión avanzada de PFS. Se utiliza para manejar los arreglos de discos externos, permitiendo que un nodo cualquiera puede crear, escribir y leer sus archivos o trabajar con un archivo compartido por todos sus compañeros. Para instalar y hacer funcionar Lustre hay que crear primero los arreglos de discos en el servidor de discos, configurándolos desde el BIOS. luego se instala el Lustre server en los servidores maestros y por último se instala el Lustre cliente en los servidores de trabajo o nodos.
- **Orange.** Es otra versión diferente de manejo de archivos distribuidos utilizando la red. Pero se diferencia del Lustre porque no requiere de configuraciones especificas en el BIOS, sino que se configura en carpetas del sistema operacional. Las características fundamentales de un clúster es que el máster se fortalece con la memoria, procesadores y discos de sus nodos esclavos, llegando a verse como un solo súper computador con una memora RAM enorme, muchos procesadores y una enorme potencia de cálculo. El máster es quien administra la ejecución de procesos y accesos a los discos del arreglo.

- **Sistema Operacional.** El sistema operacional es conocido como sistema operacional distribuido porque se instala en el computador maestro pero cada nodo lo tiene sin habérselo instalado. La Universidad Autónoma de Barcelona creó un prototipo que será usado como taller en este Módulo y que se llama PelicanHPC. Hay muchas maneras de instalar el sistema operacional, pero por lo general se hace con Linux (Centos, Red Hat, Scientific Linux, etc.) o montando un sistema como Rocks HPC.

Clústeres locales

- **Clústeres locales de alto rendimiento.**
  Son organizaciones computacionales que consumen altas cantidades de memoria, de disco, de procesadores y pueden demorar largos periodos de tiempo haciendo una tarea como calcular y listar los números primos de un número de 18 dígitos. Sin embargo, son muy usados a nivel científico.

- **Clústeres locales de alta disponibilidad.**
  Estos tipos de clústeres se caracteriza por ofrecer disponibilidad permanente de uso de los recursos. Son diseñados para ofrecer confiabilidad y disponibilidad. La disponibilidad es dada por un *hardware* altamente redundante (discos espejos, doble tarjeta, doble ups, doble planta eléctrica, doble entrada de energía, etc.) y la confiablidad es dada mediante un *software* que detecta fallos y se recupera ante los mismos. Este tipo de clústeres son usados por el sistema financiero, las clouds y, en general, todo aquel que necesite ofrecer servicios las 24 horas del día, 7 días a la semana, durante los 365 días del año.

- **Clústeres locales de alta eficiencia.**
  Son organizaciones computacionales locales, que buscan hacer la mayor cantidad de tareas en el menor tiempo. Son muy usados por los sistemas de inteligencia, donde se busca obtener resultados rápidos, los usan también los sistemas de aerolíneas y muchas aplicaciones más.

- **Clústeres mixtos.**
  Son combinaciones de los tres tipos de clústeres anteriores.

En resumen, los clústeres son organizaciones de cómputo muy poderosas, para los que se requiere saber y conocer sistemas operacionales distribuidos, procesamiento paralelo, manejo de sistemas de E/S paralelos, programación paralela y conocimientos de administración de centros de cómputo distribuidos.

El procesamiento paralelo consiste en ejecutar en muchos procesadores y a su vez en diferentes cores, las instrucciones de un programa, trabajando de manera independiente.

La programación paralela es la misma programación clásica de computadores, pero organizada con base en estrategias de programación paralela por tareas o por datos para lograr que una tarea se desarrolle en el menor tiempo posible utilizando *hardware* paralelo como los HPC.

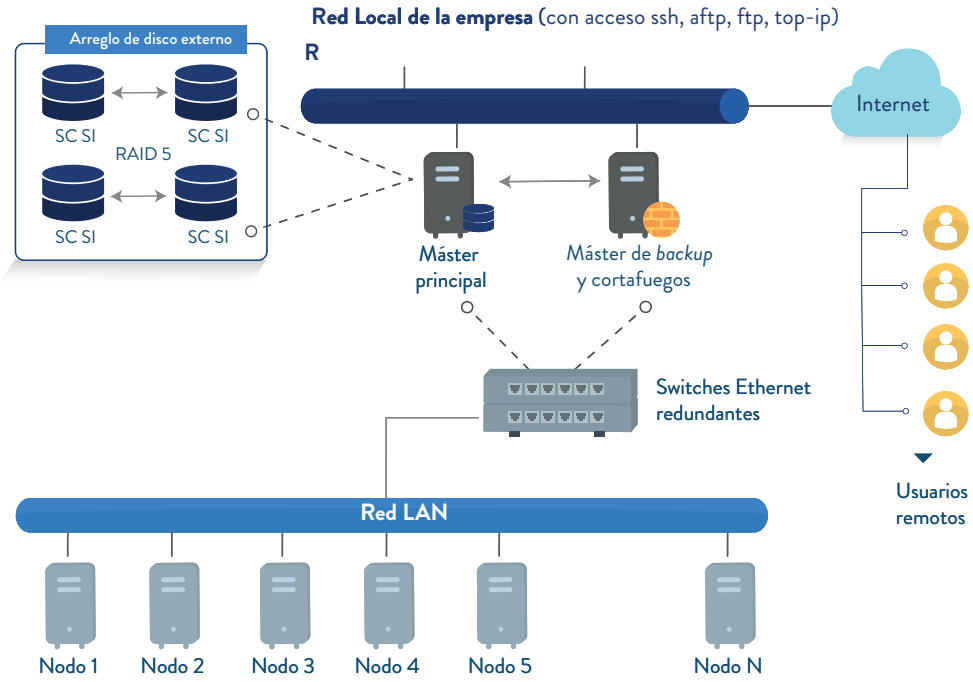

Figura 2. Modelo computacional de un clúster HPC local

Fuente: elaboración propia

2. Computación de alto desempeño

La Computación de Alto Desempeño (CAD) o *High Performance Computing* (HPC) es una herramienta relevante para el desarrollo de la ciencia, debido a la necesidad de procesar, manipular y almacenar datos a gran escala, que sin estos sistemas no se podrían realizar y tampoco sería posible probar las teorías propuestas en las diferentes ramas del conocimiento en el mundo.

La computación en clúster hace referencia a un grupo de computadores unidos entre sí, que le permite a todo el grupo comportarse como si fuera una misma entidad.

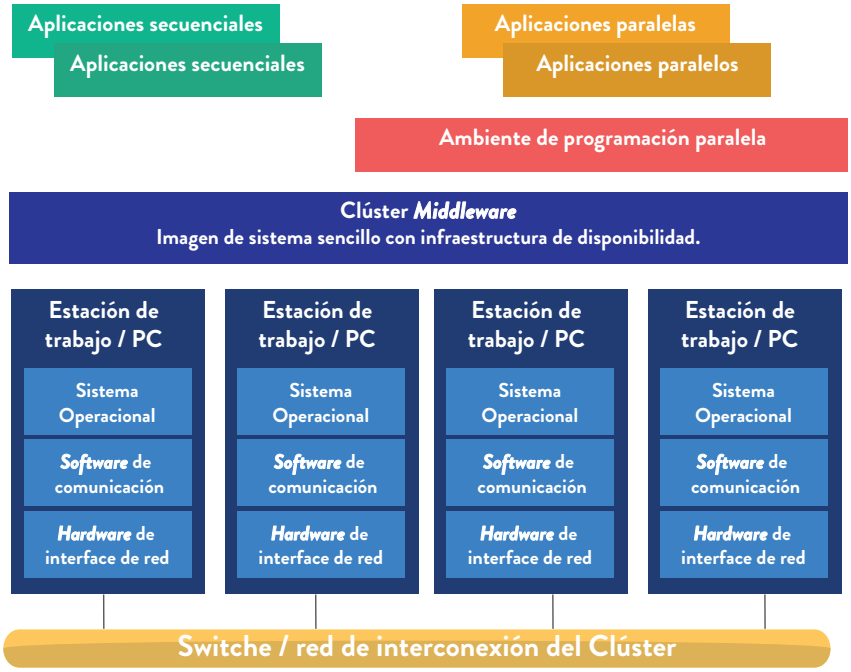

Figura 3. Arquitectura básica de componentes de *clúster*

Fuente: elaboración propia

Existe una gran cantidad de diferentes razones por la cual las personas pueden usar un clúster de computadores para varias tareas. Por ejemplo, un clúster de computadores combina la capacidad de todos los computadores involucrados, ayudando a las personas que no pueden costear un computador con esas capacidades a aprovechar las ventajas de ese nivel de computación.

Para lograr el funcionamiento de todos los equipos de la red actuando como uno solo, es necesario delegar tareas a demonios los cuales están encargados de tareas como programación, recolección, planificación, trabajo, lanzamiento y negociación.

La sincronización de todas estas operaciones se hace mediante aplicaciones creadas específicamente para estos propósitos denominadas como *middleware*. Hay muchos tipos de *middleware*: el SOA, que es un *middleware* comercial y los de propósito científico como: Condor Grid., gLite, Teragrid, Globus, etc.

La mayoría de los usuarios de los sistemas HPC son científicos, investigadores, ingenieros e instituciones académicas. Algunas agencias, particularmente la milicia, también confían en HPC para aplicaciones complejas. Los sistemas de alto desempeño generalmente son usados con componentes hechos a la medida llamados ***commodity components***.

2.1. Supercomputación

Los supercomputadores son usados para tareas de cálculos altamente intensivos tales como resolución de tareas de física cuántica, pronóstico del clima, moldeamiento molecular, simulaciones físicas (Randell, 1980).

En la actualidad, estos computadores ofrecen arquitecturas de niveles altos consistentes en un clúster de procesadores tipo MIMD, cada procesador de ellos es SIMD, y con cada multiprocesador se controlan múltiples coprocesadores (Grama, 2003) (Tanenbaum & Steen, s.f.).

Los supercomputadores varían radicalmente con respecto a el número de procesadores por clúster, el número de procesadores por multiprocesador, el número de instrucciones simultáneas por procesador SIMD y el tipo y el número de coprocesadores.

2.2. High-throughput Computing

High-throughput Computing (HTC) es un término en las ciencias de la computación que se utiliza para describir el uso de recursos computacionales sobre largos periodos de tiempo, para lograr que una tarea computacional sea completada exitosamente.

La comunidad de HTC está algo preocupada con la robustez y la fiabilidad de los trabajos sobre largas escalas de tiempo, aunque algunos sistemas HTC tales como Cóndor® y PBS® puedan correr tareas en recursos oportunistas. Este es un problema difícil para manejar en este ambiente. En una mano los sistemas necesitan proveer fiabilidad de un ambiente operativo para sus procesos, pero al mismo tiempo el sistema no debe comprometer la integridad de ejecución en un nodo y permitir al dueño tener siempre control completo de sus recursos (Beck, 1997).

2.3. Computación cuántica

Los computadores cuánticos son diferentes de los tradicionales computadores basados en transistores. La computación cuántica se basa en principios del fenómeno de la mecánica cuántica, tales como superposición y entrelazamiento cuántico, para realizar operaciones en datos.

El fundamento de la computación cuántica son las propiedades del *quantum*, que pueden representar datos y realizar operaciones en esos datos. Los transistores, aunque sean cada vez más pequeños, jamás podrán ser manométricamente infinitesimales porque son estados físicos, mientras que un *quantum* es una definición de la física mecánica que se denomina cúbit.

El Cúbit se le conoce como el bit cuántico y solo se puede describir correctamente desde la mecánica cuántica. Tiene, al igual que el bit informático, dos estados (0 y 1), solo que el bit informático puede ser (0 o 1) mientras que el cúbit puede ser 0, 1, o 0 y 1 a la vez.

Los computadores cuánticos a gran escala podrían ser capaces de resolver ciertos problemas mucho más rápido que cualquier computador clásico, usando el mejor algoritmo concurrente conocido como factorización de enteros usando algoritmos de Shore o la simulación de *quantums* en sistemas complejos. Existen algunos algoritmos cuánticos, tales como el algoritmo de Simón, el cual corre más rápido que cualquier algoritmo clásico de probabilística.

3. Manejadores locales de recursos

Un sistema manejador local de recursos controla la carga de procesamiento previniendo que los trabajos compitan por recursos de cómputo limitado. Típicamente, un sistema manejador de recursos comprende un manejador de recursos, un programador y un *scheduler* (Iqbal y GUPTA, 2005).

La mayoría de los manejadores de recursos tienen un programador interno, pero los administradores de sistemas pueden substituir un programador externo por uno interno para mejorar la funcionalidad. En tal caso el programador se comunica con el manejador de recursos para obtener información acerca de las colas de trabajo o de impresión, cargas, nodos de trabajo, y la disponibilidad de recursos para tomar las decisiones de programación.

Usualmente, el manejador de recursos controla varios demonios (programas que no terminan) en los nodos maestros y en los nodos de trabajo, incluyendo un demonio programador, que típicamente corre en los nodos maestros.

El manejador de recursos también fija un sistema de colas para los usuarios que envían trabajos, los usuarios pueden consultar el manejador de recursos para determinar el estado de sus trabajos. En adición un manejador de recursos mantiene una lista de recursos disponibles y reporta el estado de los trabajos previamente enviados por el usuario. El manejador de recursos ayuda a organizar los trabajos enviados basados en prioridades, recursos solicitados y disponibilidad de los nodos trabajadores.

A medida que la demanda por procesamiento y velocidad crezca, HPC será un área de interés para los negocios de todos los tamaños, particularmente para los de transacciones de procesos y bodegas de datos.

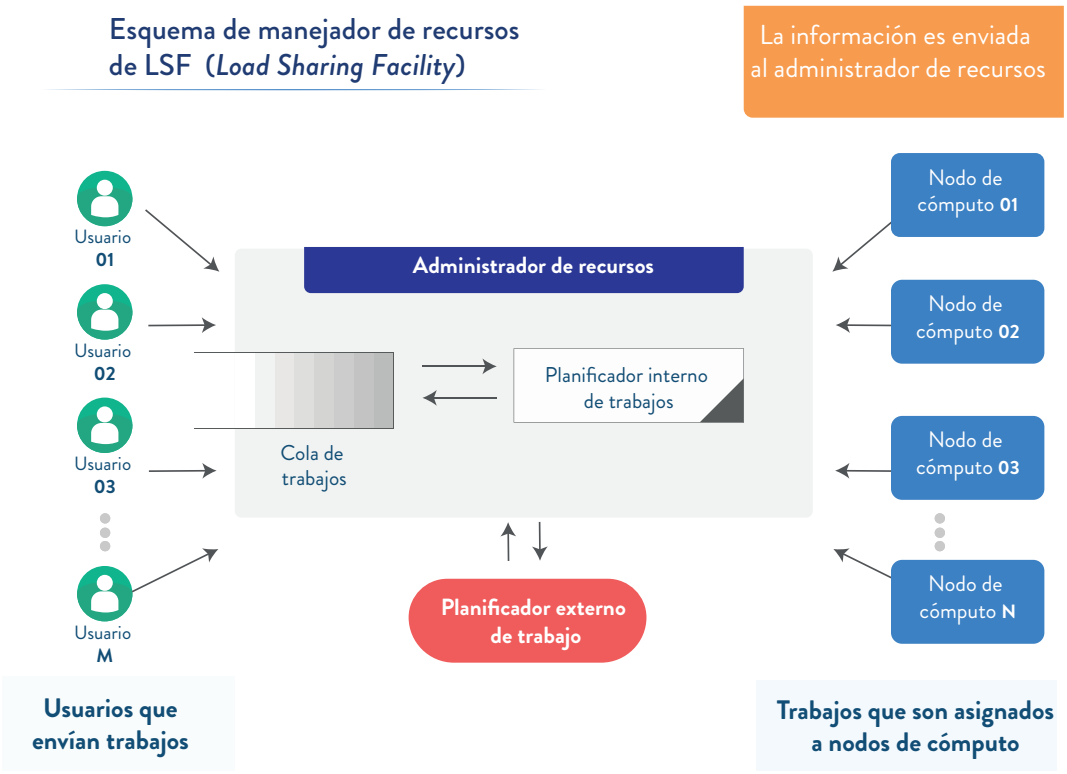

Figura 4. Esquema de manejador de recursos de LSF (*Load Sharing Facility*)

Fuente: elaboración propia adaptado de Iqbal y Gupta (2005)

3.1. Programadores y colas de trabajos

Cuando un trabajo es enviado al manejador de recursos, el trabajo espera en cola hasta que es programado y ejecutado. El tiempo gastado en la cola o tiempo de espera depende de varios factores, incluyendo la prioridad del trabajo, la carga del sistema y la disponibilidad de los recursos solicitados. La utilización de los recursos: durante la línea de tiempo del trabajo representa la carga útil que el trabajo ha ejecutado.

System throughput: está definido por el número de trabajos (*jobs*) completados por unidad de tiempo.

Mean response: es una medida de desempeño para los usuarios, quienes esperan mínimos tiempos de respuesta.

En ambientes típicos de producción, trabajos muy diferentes son enviados a los clústeres, esos trabajos pueden ser caracterizados por factores tales como el número de procesadores solicitados, el tiempo estimado de ejecución y los requerimientos E/S (Grama, 2003). Durante la ejecución larga, trabajos (*jobs*) específicos pueden ocupar porciones significativas del clúster procesando y compartiendo memoria.

4. Computación en malla (*grid computing* o clúster remoto)

Con la aparición de NGS (secuenciación genómica de próxima generación), varios ecosistemas informáticos del genoma se enfrentan ahora a un posible tsunami de datos genómicos que inundarán sus sistemas de almacenamiento y aplastarán sus grupos de equipos informáticos.

El ADN humano consta de aproximadamente 3 mil millones de pares de bases con un genoma personal que representa aproximadamente 100 gigabytes (GB) de datos. A finales de 2011, la capacidad de secuenciación anual global se estimaba en 13 cuatrillones de bases y contando.

El próximo diluvio de datos obliga a los investigadores a encontrar métodos confiables y convenientes para el almacenamiento de los mismos. En la comunidad de bioinformática, la adquisición de datos de secuencia siempre es seguido de un análisis computacional a gran escala para procesar los datos, validar los resultados del experimento y extraer conocimientos científicos.

La Universidad de Illinois es la universidad que estudia las ciencias de la vida y para correr sus modelos matemáticos en conjunto con otras universidades y el Fernilab (Laboratorio Ferni) utiliza una malla computacional, pero su nodo (centro de cómputo) tiene un millón de *cores*.

Es importante tener en cuenta que cada nodo de una malla computacional es un centro de cómputo y puede estar en cualquier lugar del mundo o de un país. El Ministerio de Educación de Colombia montó una malla computacional en el 2008 y funcionó por más de 4 años. Los nodos estaban conformados por centros de cómputo de Uniandes (CA-Certificate Authority), Udistrital, Poli, Uninorte, Univalle y 20 Universidades más. Unidandes fue la Autoridad Certificadora.

Una malla computacional funciona como un clúster local, pero la diferencia radica en que en un clúster local, un nodo es un servidor y los nodos están conectados por red LAN al nodo master, mientras que en una malla computacional, la red es una red WAN, los nodos son centros de cómputo que le rinden informe a la entidad certificadora, y esta, a su vez, hace de nodo maestro y da los certificados o permisos de uso para ejecutar un proceso en la malla computacional.

En la Figura 6 se muestra un modelo de lo que es una malla computacional a gran escala, en donde los círculos rojos corresponden a la puerta de entrada al nodo de la red (UI – *User Interface*) y los círculos naranja corresponde a los servidores maestros del nodo (CE – *Computer Element*). Los nodos verdes son nos servidores de trabajo conocidos como *Workerds Nodes* - WN. El conjunto de la UI, CE y WN agrupa los elementos principales de un nodo de una malla computacional y para verlo completo se puede ver la Figura 7, en la cual se muestra un diseño completo de un nodo de una malla computacional.

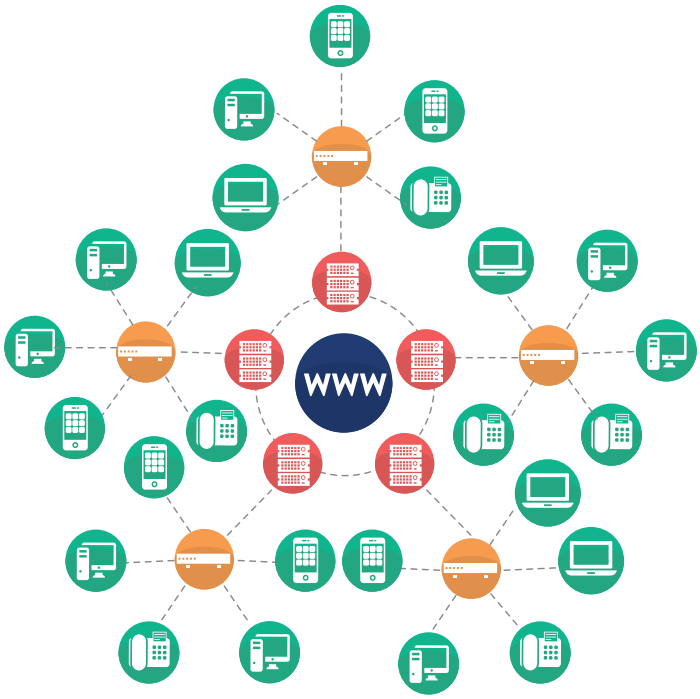

Figura 5. Diagrama de una malla computacional a gran escala
Fuente: elaboración propia

4.1. Descripción de los componentes de una malla computacional

Un nodo normal de una malla computacional está compuesto por:

UI o interfase de usuario: por donde llega y sale todo lo que debe ejecutarse en el nodo, o por donde salen los resultados. También se le conoce como grid portal porque es un portal.

Proxy: el proxy es el cortafuegos del nodo. Viene habilitado por un certificado x509 que emite la CA para el nodo. Este certificado solo tiene vida por 12 horas, y si el trabajo no se concluye en ese lapso, el proceso es cancelado. Para evitar este inconveniente, se hace un alias que se llama Myproxy. Myproxy tiene que ser instalado en el nodo y es habilitado luego por la CA y extiende el tiempo de vida del certificado hasta que el proceso termine. Si el proceso cancela por cualquier causa, el certificado queda caducado y debe pedirse otro certificado a la CA, para la reejecución del proceso.

CE – *Computer Element* (computador maestro o contenedor): es el nodo coordinador de los trabajos. Todos los WN *(worker node)* deben recibir y reportar los trabajos al nodo maestro. El CE es quien reparte cargas y asigna trabajo a los WN.

WN – *Worker Node* o nodo de trabajo: este nombre es dado a los servidores que desarrollan los trabajos, los verdaderos peones del ajedrez. Deben estar sincronizados por el NTP (*Network Time Prolocol*) con el CE y la CA.

NTP: es el servidor dedicado a la sincronización del tiempo. El reloj de los nodos de trabajo nunca debe estar por encima del tiempo del reloj de la CA porque el sistema *(middleware)* lo saca del sistema.

Grid FTP: es el servidor dedicado al recibo y transmisión de datos. Es capaz de transferir cientos de gigas o teras a través de la red WAN. Es quien debidamente autorizado por la CA envía o recibe datos para la ejecución de trabajos.

DNS: es el servidor de nombres. Se requiere para que el CE, la CA, el LDAP, en NTP, etc. conozcan cuál IP tiene cada uno de los integrantes del nodo.

LDAP: es un servidor activo, que se encarga de ejecutar los certificados de seguridad, crear el acceso a los usuarios y administrar las claves de los usuarios.

SE – *Storage Element:* es el servidor que administra las unidades de almacenamiento (Entrada/Salida) y las unidades de cinta magnética.

Hasta aquí es la arquitectura normal de un nodo.

CA – Autoridad Certificadora: si el nodo es el administrador de la malla computacional, entonces tiene otro servidor que se llama CA (*Certificate Authority*) que se encarga de generar los certificados de usuario, certificados de trabajo, certificados de host, para los diferentes nodos de la malla computacional y los usuarios de cada nodo, al igual que los certificados para cada trabajo que se mande ejecutar en la malla por un usuario “Y” de un nodo “X”.

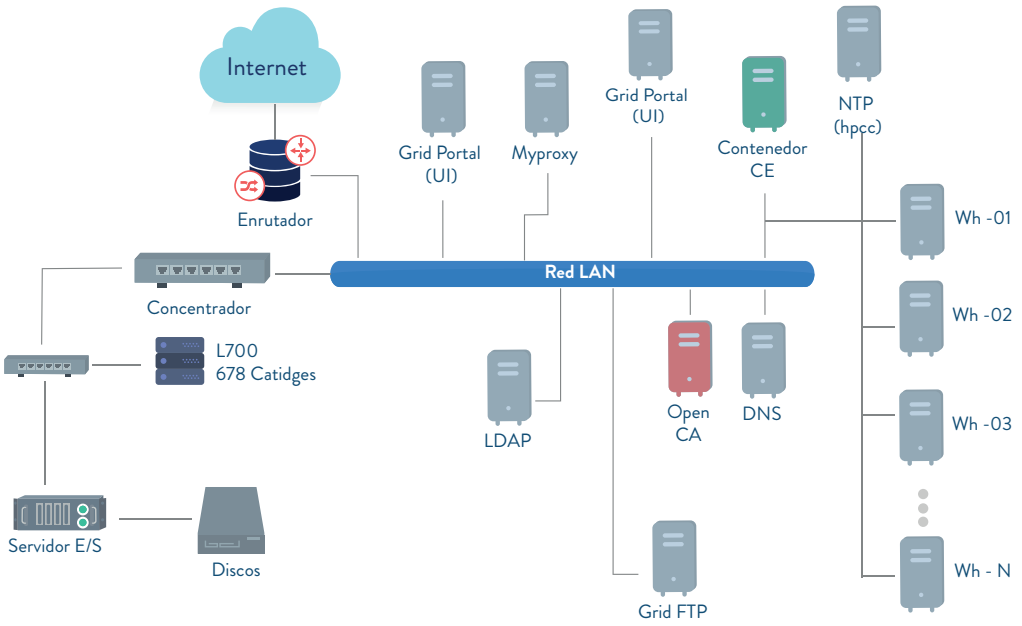

Figura 6. Arquitectura *grid computing*

Fuente: elaboración propia

4.2. Redes para las mallas computacionales

Las redes para las mallas computacionales son redes de banda ancha. En Colombia la red se llama RENATA (Red Nacional de Tecnología Avanzada) y su amplitud de banda es de más de 10GB. En lo hogares se cuenta a lo sumo 20 Megas, para que sirva como punto de referencia y esa es la razón por la cual un nodo de una malla computacional parece o da la sensación de ser un servidor más de la malla.

4.3. Middleware para mallas computacionales

Como se ha venido explicando, un *middleware* es un *software* que es capaz de interconectar diferentes tipos de servidores, con diferentes sistemas operacionales, que son interconectados por diferentes redes, que usan diferentes compiladores, que trabajan en una multitud de lenguajes de programación y que ejecutan aplicaciones de cualquier tipo.

Dentro de esos *middlewares* se tienen, entre otros:

SOA. Arquitectura abierta de servicios usada en sistemas comerciales y más específicamente en los buses de servicios de las empresas o bien conocidos como Enterprise Server Bus-ESB.

Condor Grid. *Middleware* usado por las universidades y centros de investigación norteamericanos. Ese proyecto es de índole académico y se puede encontrar en https://research.cs.wisc.edu/htcondor/

gLite. La distribución de gLite era un conjunto integrado de componentes diseñados para permitir el intercambio de recursos. En otras palabras, esto era un *middleware* para construir una grilla. El *middleware* gLite fue producido por el proyecto EGEE y posteriormente fue desarrollado por el proyecto EMI. Mas información en http://grid-deployment.web.cern.ch/grid-deployment/glite-web/license

Globus. Globus es una herramienta de código abierto desarrollado en java, fundamental para "Grid", que permite a las personas y a las organizaciones compartir su poder de cómputo, bases de datos y otras herramientas en línea de manera segura a través de límites corporativos, institucionales y geográficos sin sacrificar la autonomía local.

Más información en: http://toolkit.globus.org/toolkit/about.html

Referencias bibliográficas

Beck, A. (1997). *High throughput computing: an interview with miron livny. HPCwire*, 2.

Grama, A. (2003). *Introduction to parallel computing.* Addison-Wesley.

Iqbal, S y Gupta, R. (2005). *Planning Considerations for Job Scheduling in HPC Clusters.* Dell Power Solutions, 4.

Mead, C. (s.f.). *Excerpts from A Conversation with Gordon Moore: Moore's Law Moore's Law, 2.* Recuperado de: http://large.stanford.edu/courses/2012/ph250/lee1/docs Excepts_A_Conversation_with_Gordon_Moore.pdf

Randell, B. (1980). The COLOSSUS. *A History of Computing in the Twentieth Century*, 47–92. Recuperdo de: https://doi.org/10.1016/B978-0-12-491650-0.50013-7

Tanenbaum, A. S. y Steen, M. (s.f.). *Distributed systems: principles and paradigms*.

UN. *Población Mundial.* Recuperado de: http://www.un.org/es/sections/issues-depth/population/index.html

Yang, L. T. y Guo, M. (2006). *High-performance computing: paradigm and infrastructure.* Wiley-Interscience. Recuperado de: https://www.wiley.com/en-us/High+Performance+Computing%3A+Paradigm+and+Infrastructure-p-9780471654711

INFORMACIÓN TÉCNICA

FACULTAD DE
INGENIERÍA, DISEÑO
E INNOVACIÓN

Módulo: Sistema Distribuidos

Unidad 3: Computación en clúster y programación paralela

Escenario 5: Objetos distribuidos, sistemas de memoria

Autor: Alexis Rojas

Asesor Pedagógico: Jeimy Lorena Romero Perilla

Diseñador Gráfico: Brandon Steven Ramírez Carrero

Asistente: Maria Elizabeth Avilán Forero