Implementation of a Computational Grid Infraestructure Based in the gLite Middleware Using the Paravirtualization Technique

Implementación de una Infraestructura Computacional Grid Basada en el Middleware gLite Utilizando la Técnica de Paravirtualización

Julián Mauricio Nobsa Vargas
Universidad Industrial de Santander, Colombia
jnobsa@gmail.com

Abstract

This article explains the fundamental concepts for understanding a grid infrastructure and discuss about the general aspects of implementation and operation with the EGEE middleware, gLite. Additionally, explain the methodology of implementation of the virtualization technique known as paravirtualization and the benefits it provides in the construction, configuration, implementation and administration of the platform.

1. Introducción

La computación en grid surge como una respuesta a la necesidad de altas capacidades de procesamiento y almacenamiento principalmente por parte de la comunidad investigadora, estas necesidades se han visto cubiertas a través de una infraestructura que permite el uso de recursos compartidos por diferentes organizaciones que se encuentran geográficamente distribuidas y ha originado una oportunidad de mejora en los procesos de investigación debido a que posibilita el trabajo colaborativo y la transferencia y construcción colaborativa de conocimiento. El uso de infraestructuras grid es de gran importancia para el desarrollo e impulso de la investigación en el contexto colombiano, ya que nos permite acceder a recursos con los que no siempre cuentan las instituciones de nuestro país a la vez que posibilita concentrar y compartir los recursos que estas posean. Sin embargo la complejidad que conlleva la implementación de este tipo de infraestructuras es alta, siendo esto un factor que incide negativamente en la masificación y difusión de esta tecnología. Adicionalmente en el caso de este proyecto la limitante de recursos exigió la aplicación de mecanismos que permitieran implementar toda la infraestructura en una cantidad de recursos inferior a la requerida, esto se logró a partir de la técnica de paravirtualización, la cual no solo sirvió para solucionar este problema sino que también aportó diversas ventajas en la logística de implementacion de la infraestructura.

2. Infraestructura Grid

Es una infraestructura de hardware y software que permite compartir y utilizar de forma coordinada diferentes tipos de recursos (cómputo, almacenamiento y aplicaciones específicas) que pueden ser heterogéneos (diferentes arquitecturas, supercomputadores, clusters) y que se encuentran conectados mediante redes de área extensa (por ejemplo Internet), sin la preocupación de ubicación física, facilitando la comunicación entre organizaciones geográficamente distribuidas.

3. Arquitectura general de una infraestructura grid

La arquitectura general de un grid establece 4 capas, cada una de ellas agrupa un conjunto de recursos o servicios que se encargan de realizar labores enfocadas a cumplir un objetivo en común, las capas son las siguientes:

- Capa de red *Network Layer*
- Capa de recursos *Resource Layer*
- Capa de middleware *Middleware Layer*
- Capa de aplicaciones y servicios *Software Application and Serviceware Layer*

Capa de red:

Las redes son una pieza fundamental en una infraestructura grid, estas se pueden caracterizar según su cobertura y su desempeño, entendido éste como la cantidad de datos que puede transmitir en determinado intervalo de tiempo. El desempeño generalmente se mide en Kilo, Mega o Giga Bits por segundo. Debido a que en un grid se utilizan recursos geográficamente dispersos, son necesarias

redes de área extensa y de alto desempeño que cubran grandes distancias y puedan transmitir a altas velocidades. En esta capa se encuentran protocolos de seguridad y comunicación estándarizados para transacciones de red. Los protocolos de comunicación permiten el intercambio de datos en la capa de recursos mientras que los protocolos de seguridad brindan mecanismos de criptografía para mantener la seguridad en la identificación de usuarios y recursos.

Capa de recursos:

Esta conformada por los recursos computacionales físicos que serán compartidos por las diferentes organizaciones virtuales, estos pueden ser servidores de almacenamiento masivo, supercomputadores, clusters de alto rendimiento y hardware especializado como telescopios, microscopios etc.

Capa de middleware:

El Middleware es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre la capa de aplicaciones y la capa de recursos abstrayendo de la complejidad y heterogeneidad de estos y proporcionando una API para el fácil manejo de aplicaciones distribuidas. El middleware organiza e integra los recursos en un grid y está conformado por un conjunto de servicios software que implementan el acceso uniforme a los datos y recursos, el manejo de su información de estado, la planificación de la asignación de estos y la autenticación y autorización. En esta capa también se encuentran los servicios que permiten obtener la información de un recurso en particular y gestionarlo controlando el acceso, arranque de procesos, monitorización y auditoria.

Capa de aplicaciones y servicios software: Es la capa del mas alto nivel de la estructura siendo visible a los usuarios e interactuando con ellos de varias maneras brindándoles facilidades para trabajar con aplicaciones en diferentes áreas como ciencia, ingeniería, finanzas y simulación entre otras. Las aplicaciones grid interactuan con otras capas como la de Middleware debido a que una aplicación necesita obtener las credenciales necesarias para autenticación con el fin de acceder a recursos y archivos, consultar su estado, determinar su ubicación y características, enviar solicitudes a la capa de recursos para extraer datos, iniciar cálculos y finalmente monitorizar el progreso de estos y las transferencias de datos así como notificar al usuario cuando lo análisis hayan finalizado suministrando los resultados, adicionalmente detectar y responder a fallos. La capa de aplicaciones también incluye algunos servicios que implementan funciones generales de manejo como seguimiento de quien provee los recursos y quien los está usando.

4. Arquitectura del middleware gLite

El middleware gLite esta estructurado en una serie de módulos que se encargan de aspectos específicos del grid como el esquema de seguridad, los servicios de información y monitoreo, gestión de trabajos y servicios de datos. A su vez estos módulos están compuestos por una serie de elementos que estan relacionados fuertemente entre si y que configurados y sincronizados, permiten que los módulos de cada tipo de servicio funcionen correctamente.

Los elementos mas importantes que constituyen los módulos de gLite son:

User Interface (UI): La interfaz de usuario es el punto de acceso al Grid, esta puede ser cualquier máquina donde los usuarios tienen una cuenta personal y en la cual el certificado de usuario está instalado. Desde una interfaz, el usuario puede ser autenticado y autorizado para utilizar los recursos, e inmediatamente acceder a las funcionalidades que ofrecen los sistemas de Información, Workload y Data Manager. La interfaz de usuario proporciona herramientas para realizar algunas operaciones básicas en el Grid sobre los recursos y los trabajos.

Computing Element (CE):

El Computing Element es el servicio que representa un recurso de cómputo. Su principal funcionalidad es la gestión de trabajos (envío y control de trabajos, etc). Tiene bajo su cargo los nodods que ejecutan realmente los trabajos conocidos como worker nodes, el CE actúa como nodo maestro de un cluster donde los nodos esclavos son los worker nodes.

Storage Element (SE):

Es el servicio que permite a un usuario o una aplicación guardar datos para la futura recuperación y uso de los mismos. Es consultado por las aplicaciones que requieran estos datos. Un Storage Element proporciona acceso uniforme a los recursos de almacenamiento de datos. El SE puede controlar servidores de disco simple, grandes arrays de disco o cinta, basados en los sistemas de almacenamiento masivo (MSS).

Workload Management (WM):

Su propósito es el de aceptar y satisfacer las solicitudes de gestión de trabajos procedentes los clientes. Para un cálculo de trabajos, hay dos tipos principales de petición: presentación y cancelación. En particular el significado de la petición de presentación es pasar la responsabilidad del trabajo al WM.

El WM luego pasará el trabajo a un "CE" apropiado para su ejecución, teniendo en cuenta las necesidades y preferencias expresadas en la

descripción de trabajo. La decisión de cual recurso debe utilizarse es el resultado de un proceso de vinculación entre la presentación solicitudes y los recursos disponibles.

BDII (Berkeley Database Information Index): El componente principal del Information Service es el Berkeley Database Information Index (BDII) que tiene la función de proveer la información del estado de los recursos de un grid, es a este elemento que el WM consulta por el estado de los Computing Elements y otros nodos. El BDII consiste en dos o mas bases de datos LDAP que son alimentadas mediante un proceso de actualización. El redireccionamiento de puertos es usado para habilitar una base de datos para ser servidor de datos mientras la otra se actualiza.

File Catalog (LFC): Este componente se encarga de manejar la información acerca de los archivos de datos presentes en todo el grid, hace las veces de servidor de páginas amarillas para los archivos que pueden ser usados como entradas para ejecutar trabajos o que los usuarios han almacenado. Es consultado por el WM para saber en que Storage Elements se encuentra determinado archivo.

Adicionalmente a estos nodos descritos existen algunos otros para diferentes funciones como el monitoreo de trabajos a partir del Logging and Bookeeping, los cuales se pueden implementar en una misma máquina con otro elemento mas complejo.

5. Virtualización

La necesidad de virtualizar parte de la disponibilidad de un número de recursos inferior al necesitado para implementar los nodos necesarios para el envío completo de un trabajo, esto plantea la necesidad de poder ejecutar diferentes sistemas operativos de manera simultánea en una máquina.

Para esto se realizó un estudio de las diferentes técnicas de virtualización que permitían esto y se optó por la implementación de la técnica llamada paravirtualización por ventajas como las que se mencionan a continuación.

6. Paravirtualización

Esta técnica se basa en la arquitectura de niveles de los actuales procesadores

que consiste en que cada procesador tiene 4 niveles de privilegios representados por 4 anillos enumerados del 0 al 3.

El anillo 0 es quien permite mayores privilegios de ejecución, permitiendo acceso sin restricciones al hardware, es en este anillo donde se ejecuta el kernel o núcleo del sistema operativo y normalmente los controladores de dispositivos.

Los anillos 1 y 2, tienen menos privilegios que el 0 y en estos se ejecutan servicios básicos del sistema, los cuales pueden ser llamados por las aplicaciones, por ejemplo protocolos de red y gestores gráficos, esto permite que los servicios acceso a los datos de las aplicaciones pero a su vez estén protegidos de estas, ya que las aplicaciones se ejecutan en el anillo 3.

El anillo 3 es el de menor privilegio, y en este se ejecutan las aplicaciones, manteniendo así un esquema de seguridad para mantener la integridad de los recursos. La siguiente figura explica esta estructura:

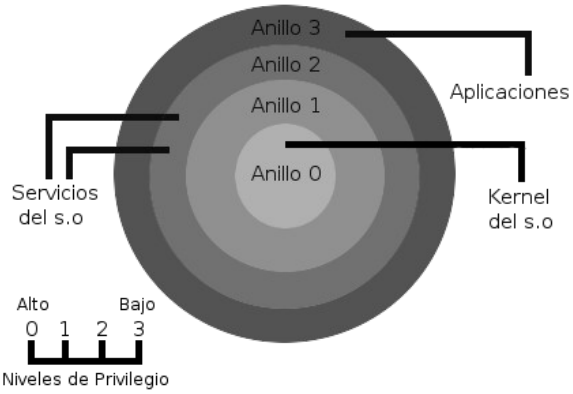

Figura 1. Estructura de anillos

La técnica de paravirtualización consiste en la ubicación de un Hypervisor en el anillo 0, este Hypervisor es un software que tiene la funcionalidad de comunicarse directamente con el hardware, el kernel del sistema anfitrión pasa a ejecutarse en un nivel superior, dejando el anillo 0 asociado al hypervisor.

Para poder realizar esto, se deben modificar tanto el kernel del sistema operativo anfitrión, como el de los sistemas operativos a virtualizar, esto podría parecer una limitante para poder paravirtualizar sistemas operativos no modificables como es el caso de los S.O's privativos, sin embargo es posible virtualizar este tipo de sistemas por medio de la paravirtualización haciendo uso de las tecnologías de virtualización VT-X de Intel y Pacifica de AMD presentes en los procesadores modernos, las cuales permiten ejecutar sistemas operativos en el nivel 0 sin tener que modificarlos, esto gracias a un nivel de privilegio especial en el anillo 0 para el hypervisor llamado root-mode también conocido como “nivel -1”, los demás componentes se ejecutan en otro nivel llamado non-root-mode.

De esta manera quedan redistribuidos los componentes del sistema, en el anillo 0 se encuentra el hypervisor y en los anillos superiores se encuentran tanto el S.O anfitrión como de los sistemas operativos invitados y sus kernel que se comunican con el Hypervisor y este a su vez con el

hardware. La estructura de la técnica de Paravirtualización se detalla en la siguiente figura:

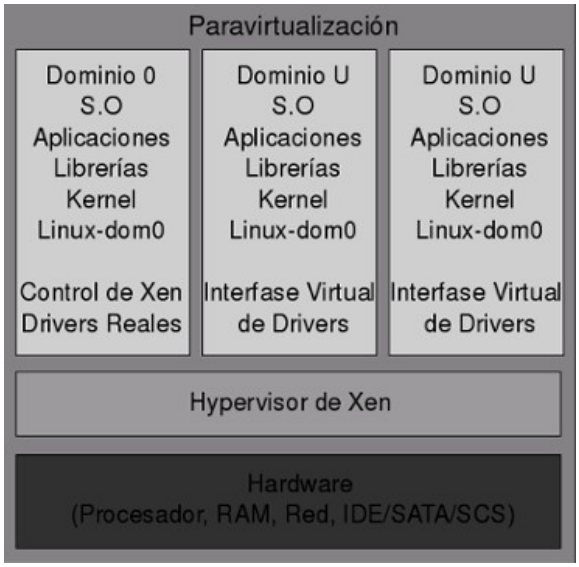

Figura 2. Estructura de la Paravirtualización

Entre las ventajas de esta técnica se encuentran:

- Facilidad para la migración en caliente sin perdida de desempeño en las máquinas que están siendo virtualizadas.
- Heterogeneidad en los S.O's a virtualizar ya que los sistemas operativos invitados no tienen que compartir un mismo kernel.
- Gran rapidez de ejecución al no tener que realizar emulación alguna.
- Portabilidad, ya que las imágenes de los S.O invitados se pueden extraer con facilidad y es posible convertir imágenes de sistemas operativos virtualizados con otras técnicas en imágenes de paravirtualización.
- Escalabilidad, debido a que inicializar una imagen puede hacerse con mucha facilidad y son totalmente aisladas de tal forma que no se afectan en caso de problemas.

7. Metodología de implementación

Para el montaje y configuración de las máquinas paravirtualizadas es necesario primero configurar el arranque del computador permitiendo que el Hypervisor se ubique en el modo privilegiado (anillo 0) y posteriormente el sistema operativo que va acompañado del hypervisor, llamado de otra forma el dom0.

Con el fin de iniciar el hypervisor en el anillo 0 es necesario modificar el sistema operativo GNU/Linux Scientific Linux Cern 4.4 instalándole un kernel modificado que permita ejecutarse en un anillo superior diferente del anillo 0 otra forma es obteniendo una imagen modificada del sistema operativo que incluya dicho kernel instalado a partir del arranque del dom0 los S.O paravirtualizados serán dom1, dom2 o mejor conocidos como domU.

Scientific Linux Cern 4.4 distribución construida con base en Scientific Linux la cual a su vez está basada en RedHat Enterprise Linux 4, incluye paquetes del CERN el cual desarrolla gLite, tiene gran soporte en la comunidad que hace uso de ella y tiene una gama amplia de paquetes precompilados, entre ellos los paquetes necesarios para iniciar el S.O como dom0.

Cada nodo de los que conforman gLite y que son fundamentales para el envío completo de un trabajo fué implementado en un domU, de forma tal que cada nodo es independiente del otro y la cantidad de máquinas físicas a utilizar disminuye, logrando implementar un grid funcional con varios nodos por cada máquina física.

8. Conclusiones

La implementación de los nodos que componen un grid utilizando el middleware gLite se puede realizar paravirtualizando cada nodo, de manera que se logra tener para cada elemento de estos las ventajas que con lleva el uso de esta técnica.

La técnica de paravirtualización provee de un rendimiento aceptable para la mayoría de nodos, sin embargo para nodos que realicen operaciones complejas como los worker nodes es recomendable estar instalados en una máquina física.

La labor de instalación, configuración y sincronización de los nodos se facilita con el uso de la paravirtualización ya que probar una nueva configuración es sencillo utilizando máquinas virtuales que se activan y desactivan con facilidad, ventaja que no se tiene con la instalación de los nodos en máquinas físicas.

10. Referencias

[1] Programming the Grid with gLite Laure, E.(CERN, Geneva, Switzerland) et al 22 March 2006 EGEE. http://cdsweb.cern.ch/search.py?p=EGEE-TR-2006-01.

[2] Introduction to High Performance and Grid Computing, Architecture and Services of the gLite Middleware, Dusan Vudragovic, Faculty of Sciences University of Novi Sad, Scientific Computing Laboratory, Institute of Physics Belgrade, Serbia, Feb 6 of 2009.

[3] The gLite Middleware distribution, OSG Consortium Meeting, Seattle, 21-23 August 2006.

[4] J. Nobsa, I. Rodriguez, G. Díaz. Estudio Comparativo de las Técnicas de Paravirtualización y Virtualización a nivel de Sistema Operativo. Bucaramanga, Colombia. Agosto 2008.