TIPOS DE SISTEMAS DISTRIBUIDOS

Freddy L. Abad Leon

—————————— ◆ ——————————

**1- ¿QUÉ ES Y CÓMO FUNCIONA EL CLÚSTER BEOWULF?**

El Clúster Beowulf es una tecnología que agrupa computadores basados en el sistema Linux con el fin de formar una máquina poderosa que emula el comportamiento de un supercomputador de multiprocesamiento. El clúster se resume como un sistema de cómputo paralelo basado en clusters de ordenadores personales conectados a través de redes informáticas estándar, sin el uso de equipos desarrollados específicamente para la computación paralela. Este c**lúster fue** creado en la NASA en 1994 con hardware de tercera clase, casi de desecho.

El Proyecto Beowulf desarrollado por Thomas Sterling y Donald Becker, se realice bajo el siguiente procedimiento: Se agrupo 16 procesadores Intel DX4 de unos 100 MHz, los interconectaron con tecnología Ethernet a 10 Mbps, en equipo viejos con Linux instalado, para ejecutar tareas científicas en paralelo a un precio incomparablemente bajo para su tiempo. De hecho, el kernel Linux de aquel entonces no tenía tanto soporte para placas Ethernet como hoy en día, por eso Donald Becker se dio a la tarea de escribir drivers propios que por supuesto compartió a la comunidad.

**2- ¿QUÉ ES Y PARA QUE SIRVE MOSIX?**

MOSIX es un sistema de gestión dirigido a la computación distribuida en clústeres de Linux y nubes de varios clústeres. A su vez, MOSIX es un sistema operativo distribuido propietario. Sus primeras versiones se basaron en sistemas UNIX más antiguos, desde 1999 se centra en clusters y redes de Linux. En una cuadrícula de MOSIX no hay necesidad de modificar o vincular aplicaciones con ninguna biblioteca, copiar archivos o iniciar sesión en nodos remotos, o incluso asignar procesos a diferentes nodos - todo se hace automáticamente, como en un SMP. MOSIX es una extensión del kernel de Linux que permite ejecutar aplicaciones "normales" (no paralelizadas) en un Clúster. Una de las posibilidades de MOSIX es la "migración de procesos", que permite migrar procesos de nodo en nodo. Si, por ejemplo, cierto proceso está dominando la carga de un nodo, este será movido a otro que tiene más recursos. A diferencia de otros tipos de Clusters, MOSIX puede ser usado en PCs de distintas características. Se podría "planificar" a las estaciones de trabajo que tiene MOSIX, para que estos se activen periódicamente a cierta hora. Por ejemplo, en una empresa donde las estaciones de trabajo se dejan de usar. Los nodos pueden ser usados por los servidores que realizan algún "proceso nocturno" (procesamiento, carga de datos, re indexación, etc.).

Una de las características de MOSIX es que, a diferencia de otros clusters, no es necesario modificar las aplicaciones ni tampoco utilizar librerías especiales. De hecho, tampoco es necesario asignar "a mano" los procesos a los diferentes nodos que componen el clúster.

La idea es que después de la creación de un nuevo proceso (fork), MOSIX intenta asignarlo al mejor nodo disponible en ese entonces. MOSIX estará constantemente monitoreando los procesos, y si fuera necesario, migrará un proceso entre los nodos para maximizar el rendimiento promedio.

MOSIX realiza todo esto automáticamente, bajo el concepto de "fork and forget" al igual que en un sistema SMP. Esto significa que sólo algunas aplicaciones se beneficiarán de un clúster MOSIX, básicamente:

* Procesos que requieren de harta CPU, aplicaciones científicas, de ingeniería, etc.
* Procesos paralelos, especialmente los que tienen tiempos de ejecución impredecibles.
* Clusters con nodos de diferentes velocidades y/o distintas cantidades de memoria.
* Entornos multi-usuario y de tiempo compartido.
* Servidores WEB escalables.

MOSIX funciona silenciosamente. Sus operaciones son transparentes para las aplicaciones. Los usuarios no necesitan saber dónde se están ejecutando los procesos, tampoco necesitan preocuparse de lo que están haciendo otros usuarios.

Como MOSIX está implementado en el kernel de Linux, sus operaciones son totalmente transparentes para las aplicaciones. Esto permite definir distintos tipos de clusters, incluso un clúster con diferentes CPU's o velocidades LAN.

Otra característica de MOSIX, es que sus algoritmos son descentralizados, esto significa que cada nodo puede ser el maestro de los procesos creados localmente, y un servidor de los procesos remotos que migraron desde otros nodos. Esto permite agregar o remover nodos desde el clúster en cualquier momento.

**3- ¿QUÉ GRIDS COMPUTING EXITEN EN ECUADOR/LATINOAMÉRICA Y CUÁL ES SU OBJETIVO?**

El objetivo de los “Grid Computing” o “malla de ordenadores” es el de compartir potencia computacional. Se denomina Grid Computing al sistema de computación distribuido que permite compartir recursos no centrados geográficamente para resolver problemas de gran escala. Estos recursos pueden ser hardware, software, datos o información, medios, etc. Se basa en el aprovechamiento de los ciclos de procesamiento no utilizados por los millones de ordenadores conectados a la Red. De esta forma se consigue que puedan resolver tareas que son demasiadointensivas para ser resueltas por una única máquina.

Brevemente los beneficios de los Grid Computing son: Ofrecer flexibilidad para llenar las necesidades cambiantes del negocio. Brindar alta calidad a menor costo. No necesitar de toda una nueva infraestructura para que funcione. Facilitar poder de computación / precio muy barato. Brindar el poder de un supercomputador virtual. Utilizar software gratuito y usar código fuente abierto. No precisar hardware adicional, para posibilitar el incremento de la potencia de cómputo, esto ha derivado en el cloud computing ofrecido por ISP y empresas de hosting. Como servidores escalables a la carta. Brindar transparencia para el usuario que participa en el Grid. Aprovechar tiempos de ciclo de procesador no utilizados, ahorro energético.

***GRID COMPUTING en Latinoamérica:***

El proyecto “CLARA” - Cooperación Latinoamericana de Redes Avanzadas es una organización no gubernamental sin fines de lucro, que estimula la cooperación regional en actividades educativas, científicas y culturales, además promueve la integración directa con las comunidades científicas de Europa [CLARA 2006], esta organización trata de integrar una red regional de telecomunicaciones de la más avanzada tecnología para interconectar a las redes académicas nacionales de la región, hasta el momento tiene los siguientes miembros:

* Argentina – Red Teleinformática Académica-RETINA
* Bolivia – Agencia para el Desarrollo de la Sociedad de la Información
* en Bolivia-ADSIB
* Brasil – Red Nacional de Enseñanza e Investigación – RNP
* Colombia – Red Nacional Académica de Tecnología Avanzada – RENATA
* Costa Rica – Red Nacional de Investigación – CR2Net
* Cuba – RedUniv
* Chile – Red Universitaria Nacional – REUNA - GREUNA
* Ecuador – Consorcio Ecuatoriano para el Desarrollo de Internet
* Avanzado – CEDIA
* El Salvador – Red Avanzada de Investigación, Ciencia y Educación Salvadoreña – RAICES
* Guatemala – Red Avanzada Guatemalteca para la Investigación y Educación – RAGIE
* Honduras – Universidad Tecnológica Centroamericana – UNITEC
* México – Corporación Universitaria para el Desarrollo de Internet –
* CUDI
* Nicaragua – Red Nicaragüense de Internet Avanzada –RENIA
* Panamá – Red Científica y Tecnológica – RedCyT
* Paraguay – Arandu
* Perú – Red Académica Peruana – RAAP
* Uruguay – Red Académica Uruguaya – RAU
* Venezuela – Red Académica de Centros de Investigación y Universidades Nacionales – REACCIUN

Otro proyecto de trabajo en GRID Computing en Latinoamérica es “EELA” (E-infrastructure Shared between Europe and Latin America), en el cual participan 10 países con sus redes académicas y diferentes instituciones entre universidades y centros de investigación. Su objetivo es llevar las e-Infraestructuras de los países latinoamericanos al nivel de explotación de los países europeos. La EELA se beneficia del estado maduro del proyecto ALICE (América Latina Interconectada Con Europa) y de la red CLARA.

**4- ¿QUE CLOUDS EXISTEN Y CUAL ES SU FUNCIÓN?**

Los clouds se pueden clasificar por:

* *El grado de propiedad que guarde la infraestructura (servidores, almacenamiento, etc.) con la compañía que la utiliza.*

**Asi su puede dividir por servicios cloud privados**, que son de acceso exclusivo por parte de una empresa, normalmente incluso desplegados sobre sus propias infraestructuras internas.

Los clouds también se dividen como **servicios cloud públicos**, en los que la empresa contrata a un tercero para conseguir un determinado servicio, despreocupándose por completo de la parte técnica y operativa, la cual corresponde al partner. En estos casos, la empresa no tiene control sobre qué otros clientes están alojados en las mismas infraestructuras que sus sistemas, con lo que confía (siempre según lo contratado) en las buenas prácticas, calidad y seguridad del partner con el que trabaja.

Asi también existen **los servicios de nube híbrida**, en los cuales la empresa crea un servicio de nube privada para su día a día, pero en función de los picos de carga o para determinadas tareas, ésta se conecta a servicios públicos de terceros con los que completar su ecosistema tecnológico. De esta forma se combina la seguridad y máximo control del cloud privado con la flexibilidad y escalabilidad inmediata de la nube pública, paradigma que está en pleno apogeo.

* *Otra forma de clasificar los* ***distintos tipos de cloud computing*** *que existen es por el nivel en el que operan.*

Así, los servicios informáticos pueden descomponerse en capas, de las que la infraestructura es el nivel más profundo, seguido por la plataforma y, en su parte superior, encontramos el software que utilizan los usuarios finales.

Asi se entiende por **Software-as-a-Service (SaaS)** todos aquellos **servicios en la nube en los que el usuario acceden directamente al programa o aplicativo de negocio, sin control ni conocimiento de la infraestructura que subyace bajo ellos.** Toda la parte técnica queda en manos del partner que la ha desplegado, asumiendo éste toda la complejidad de su instalación, mantenimiento y seguridad. Es la opción más sencilla de contratar, ya que no requiere conocimientos técnicos, pero en muchas empresas y organismos públicos se recela de esta alternativa ya que implica necesariamente perder el control de toda la cadena de vida de la información, con lo que puede que los requisitos y protocolos de seguridad y privacidad del SaaS no se ajusten a determinados datos de especial sensibilidad para las empresas. Por ello el SaaS suele usarse para servicios de correo electrónico, plataformas colaborativas y, poco a poco, también en algunas aplicaciones de negocio como el CRM (Customer Relationship Management). Ejemplos SaaS: Google Docs, Salesforce, Dropbox, Gmail

Un nivel por debajo del SaaS existe el **Platform-as-a-Service (PaaS)**: Se trata de un modelo intermedio en el que el equipo técnico de la empresa puede desplegar sus propias aplicaciones o desarrollos sin tener que preocuparse de los recursos (como servidores o almacenamientos) que requieren, ya que éstos corren a cargo del proveedor de servicios que tengan contratado. Asi los desarrolladores pueden construir sus aplicaciones a su gusto, con sus requisitos específicos de funcionamiento y seguridad, pero desligándose por completo de la instalación y configuración del hardware que las sostiene. Ejemplo: Google App Engine, Heroku.

Asi en la capa más profunda, se encuentra el **Infraestructure-as-a-Service (IaaS).** En este caso, **el equipo técnico de la empresa tiene el control absoluto no sólo sobre las aplicaciones que construye en la nube sino también sobre la infraestructura que tiene contratada, de forma virtual.** Así pues, la empresa es la responsable de configurar las máquinas que soporten sus desarrollos, elegir incluso la capacidad de memoria o procesador que necesitan. De este modo se consigue la mayor personalización y adecuación a lo que la empresa necesita, incluso en entornos críticos, pero a cambio se incrementa notablemente la complejidad y la labor que deben desempeñar los técnicos. Es el caso de Amazon Web Services, el referente internacional de esta tecnología, sobre cuyas máquinas virtuales se mantienen centenares de proyectos y servicios digitales de alto nivel. Ejemplos: Amazon Web Service (AWS), Rackspace Cloud y vCloud.

**5 - ¿QUÉ ES Y CÓMO FUNCIONA EL MODELO DE PUBLICACIÓN‐SUSCRIPCIÓN EN SSDD?**

El modelo de comunicación de publicación/suscripción en Sistemas Distribuidos hace referencia a que las aplicaciones no están relacionadas con socios específicos. Esto se debe a que los sistemas de publicación-suscripción manejan datos y no presentan requisitos específicos para los destinatarios o los orígenes de los mensajes. La opción de publicación-suscripción desacopla el proveedor de información de los consumidores de dicha información. El proveedor de información es denominado publicador. Los publicadores proporcionan información sobre un asunto. El consumidor de la información se denomina suscriptor. Existe un intermediario entre ambos. El suscriptor debe registrar y des registrar con un intermediario para recibir las publicaciones. Asi La información se envía en un mensaje MQ y el asunto de la información se identifica mediante un tema. El publicador especifica el tema cuando se publica la información. El suscriptor especifica los temas sobre la información deseada. Al suscriptor sólo se le envía la información a la que se ha suscrito.

**BIBLIOGRAFIA:**

1. Alan Lazalde , “Historia de la Tecnología: Clúster Beowulf, la supercomputadora de los pobres”, <https://hipertextual.com/2011/11/historia-de-la-tecnologia-cluster-beowulf-la-supercomputadora-de-los-pobres>
2. Victorino, J., & R. Hernández, “Implementación de un Clúster Beowulf”, <http://ciencias.bogota.unal.edu.co/fileadmin/content/geociencias/revista_meteorologia_colombiana/numero03/03_05.pdf>
3. Colaboradores de Wikipedia, “Clúster Beowulf”, <https://es.wikipedia.org/w/index.php?title=Cluster_Beowulf&oldid=101725001>
4. Anónimo, “MOSIX”, [www.mosix.org/](http://www.mosix.org/).
5. Anónimo, “Sistemas distribuidos de alto rendimiento a bajo costo (clustering) en sistemas operativos de libre distribución.”, <http://sistemasoperativos.wikia.com/wiki/1.4._Sistemas_distribuidos_de_alto_rendimiento_a_bajo_costo_(clustering)_en_sistemas_operativos_de_libre_distribuci%C3%B3n>.
6. Colaboradores SI, “Grid Computing” <http://www.sociedadelainformacion.com/12/Paper_Grid.pdf>
7. Weimar Díaz Garzón, “REALITY OF COMPUTATION GRID IN LATIN AMERICA”
8. <http://www.palentino.es/blog/grid-computing-concepto-y-ejemplos/>

## Oscar De La Cuesta , “Grid Computing, concepto y ejemplos.”, <http://www.ticbeat.com/tecnologias/cuantos-tipos-de-cloud-computing-existen-y-en-que-se-diferencian/>

# Alberto Iglesias Fraga, ¿Cuántos tipos de ‘cloud computing’ existen y en qué se diferencian?

1. Anónimo, “Cloud Computing y Grid Computing”, <https://prezi.com/snnf2gdz5yku/cloud-computing-y-grid-computing/>
2. Colaboradores IBM Journal, “Publish/Suscribe Model”, <https://www.ibm.com/support/knowledgecenter/es/SSZJPZ_9.1.0/com.ibm.swg.im.iis.conn.wsmq.help.doc/topics/c_cwsmqcon_Publish_Subscribe_Comm_Model.html>