DESARROLLO DE UN PROTOTIPO VIRTUALIZADO DE COMPUTACIÓN DE ALTO DESEMPEÑO (HPC) PARA LA ANALÍTICA DE DATOS

JUAN MANUEL DELGADO RAMÍREZ

INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS
BOGOTA, D.C., COLOMBIA
2014

DESARROLLO DE UN PROTOTIPO VIRTUALIZADO DE COMPUTACIÓN DE ALTO DESEMPEÑO (HPC) PARA LA ANALÍTICA DE DATOS

JUAN MANUEL DELGADO RAMÍREZ

Tesis presentada como requisito parcial para optar al título de:
INGENIERO DE SISTEMAS

Director:
Ingeniero Alexis Rojas Cordero

Línea de Investigación:
Sistemas Distribuidos, Ciencia de datos

INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS
BOGOTA, D.C., COLOMBIA
2014

Esta disertación ha sido aprobada
por el Departamento de Sistemas y la Facultad de Ingeniería
y el Politécnico Grancolombiano – Institución Universitaria por

_____
Dissertation Committee Chairperson

_____
Department/Date

_____
Committee Member's Name

_____
Department/Date

_____
Committee Member's Name

_____
Department/Date

_____
Committee Member's Name

_____
Department/Date

_____
Committee Member's Name

_____
Department/Date

A mi esposa y mis hijos

CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

GLOSARIO

ADDONS: También conocidos como extensiones, plugins, snap-ins, etc, son programas que sólo funcionan anexados a otro y que sirven para incrementar o complementar sus funcionalidades.

API: Interfaz de Programación de Aplicaciones.

BIOS: Sistema básico de entrada/salida. El BIOS del equipo se almacena en un chip de memoria flash. El BIOS controla lo siguiente: comunicaciones entre el microprocesador y los dispositivos periféricos, tales como el teclado y el adaptador de vídeo, y funciones varias como mensajes del sistema.

CACHÉ: Memoria rápida que contiene los datos a los que se haya accedido recientemente. El uso de la memoria caché acelera el acceso posterior a los mismos datos. Cuando se leen datos de la memoria principal o se graban en ella, se guarda también una copia en la memoria caché con la dirección de memoria principal asociada. El software de la memoria caché supervisa las direcciones de las lecturas posteriores para ver si los datos necesarios ya están almacenados en la memoria caché.

DIRECCIÓN IP: Dirección de identificación para cada equipo en red.

DISCO VIRTUAL (VD): Unidad de almacenamiento creada por una controladora RAID a partir de uno o más discos físicos. Aunque un disco virtual puede crearse a partir de varios discos físicos, se ve en el sistema operativo como un solo disco. Según el nivel de RAID utilizado, el disco virtual puede conservar datos redundantes en el caso de que falle un disco.

DNS: Domain Name System asigna un nombre al dominio de red.

DUPLICACIÓN: Proceso de obtener redundancia de datos completa con dos discos físicos al mantener una copia exacta de los datos de un disco en el segundo disco físico. Si uno de los discos físicos falla, el contenido del otro disco físico se puede utilizar para mantener la integridad del sistema y reconstruir los discos físicos que han fallado.

FRAMEWORK: En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

FIREWIRE: Es un estándar multiplataforma para entrada/salida de datos en serie a gran velocidad.

FIRMWARE: Software almacenado en la memoria de sólo lectura (ROM) o en la memoria ROM programable (PROM). A menudo, el firmware es responsable del comportamiento de un sistema cuando se enciende por primera vez. Un ejemplo típico sería un programa de supervisión en un sistema que carga la totalidad del sistema operativo del disco o de una red y después pasa el control al sistema operativo.

GUEST: Equipo Invitado

HOST: Equipo Anfitrión

HPCC: High Performance Computing Cluster, cluster de alto rendimiento

Hw: Hardware

JSDK: Java ServletDevelopment Kit

JVM: Máquina virtual Java

KERNEL: En informática, un núcleo o kernel (de la raíz germánica Kern) es un software que actúa de sistema operativo.

LAN: local área network, red de área local

MAINFRAME: Una computadora central o mainframe es una computadora grande, potente y costosa usada principalmente por una gran compañía para el procesamiento de una gran cantidad de datos; por ejemplo, para el procesamiento de transacciones bancarias.

NIVEL DE RAID: Propiedad de un disco virtual que indica el nivel de RAID del disco virtual. En controladoras Dell PERC 5/i, se admiten niveles de RAID 0, 1, 5 y 10. En controladoras Dell SAS 5/iR, se admiten los niveles de RAID 0 y 1.

NODO: Cada uno de los componentes de cada Clúster

QoS: Quality of Service, calidad de servicio

RACK: Armario para guardar los elementos físicos del clúster

RIP: Routing Information Protocol, Protocolo de encaminamiento de información

ROI: Return On Investment, Retorno de la Inversión.

SAS: SCSI conectado al puerto serie. SAS es una interfaz de dispositivo punto a punto a nivel empresarial que nivela el conjunto de protocolos de

SCSI: Interfaz de ordenador pequeño (SCSI). La interfaz SAS proporciona un desempeño mejorado, cableado simplificado, conectores menores, conteo de patas menor y requisitos de alimentación reducidos en comparación con el SCSI paralelo. La controladora Dell SAS 5/iR admite la interfaz de SAS.

SMP: Es la sigla de Symmetric Multi- Processing, multiproceso simétrico. Se trata de un tipo de arquitectura de ordenadores en que dos o más procesadores comparten una única memoria central.

SSD: (Solid State Drive), Disco de estado Sólido.

Sw: Software

TASK: Tareas

TCO:Total Cost of Ownership, Costo Total de Propiedad.

TOLERANCIA A FALLOS: La tolerancia a fallos es la capacidad del subsistema de disco de experimentar un fallo en una unidad por grupo de discos sin comprometer el procesamiento y la integridad de los datos. La controladora PERC 5/i ofrece tolerancia a fallos a través de grupos de discos redundantes en los niveles de RAID 1, 5 y 10.
También admite discos de repuesto dinámico y la función de reconstrucción automática. Las controladoras SAS 5/iR admiten grupos de discos redundantes RAID 1.

UNIX: Es un sistema operativo portable, multitarea y multiusuario

VLAN: Virtual LAN, Red de Área Local Virtual

VM: Virtual Machine, Máquina Virtual.

RESUMEN

El presente documento se enfoca en la definición de un modelo de Computación de Alto Desempeño (High Performance Computing, HPC) que permita la integración de servicios de Analítica de Datos para su uso en la academia y las empresas en general. Se definen y describen conceptos como clúster y virtualización, que se usan en el modelo. Se presenta la estructura actual de las soluciones de Analítica de Datos y se relacionan un conjunto de herramientas de HPC que soportan los elementos de la Analítica de Datos.

El documento tiene una sección que describe los criterios de selección de la solución de software que apoya la construcción del modelo para brindar servicios de HPC y Analítica de Datos y luego describe el proceso requerido para la construcción del modelo en un ambiente de servidores virtuales con que cuenta el Politécnico Grancolombiano – Institución Universitaria.

Palabras clave: HPC, Clúster, Virtualización, Big Data, Analítica, Linux

INTRODUCCION

Dentro del marco de trabajo propuesto por la asignatura de Sistemas Distribuidos de la carrera de Ingeniería de Sistemas de la Institución Universitaria Politécnico Grancolombiano y basados en el trabajo de estudio de un modelo de Computación de Alto Desempeño con herramientas como Pelican HPC y Sun HPC, nació la idea de realizar un estudio de las alternativas disponibles para la academia y las organizaciones para la implementación de los modelos de computación de alto desempeño utilizando la virtualización de servidores como primera alternativa asequible para recrear la infraestructura requerida para la computación de alto desempeño y procurar garantizar una adecuada utilización de la tecnología dispuesta.

La Computación de Alto Desempeño ha sido tradicionalmente circunscrita a los centros de investigación, la academia y las instituciones de ciencia y tecnología por lo que también es un objetivo de este estudio el presentar un escenario actual en el que las empresas de cualquier envergadura pueden estar interesadas y en donde pueden encontrar en el modelo sugerido por el estudio una alternativa adecuada para sus necesidades de crecimiento.

Sin embargo, los requerimientos de capacidad de cómputo van de la mano con problemas de costos de equipos y energía eléctrica, requerimientos mayores de espacio y sistemas de refrigeración de las grandes cantidades de equipos y poder de procesamiento. Frente a estos inconvenientes se observa la conveniencia del uso del paradigma de virtualización del hardware para facilitar la creación del modelo y aprovechar sus capacidades de portabilidad y gestionabilidad.

La virtualización resuelve el problema de asignación de recursos y gestión balanceo de carga, así como los conflictos entre aplicaciones en situaciones de consolidación, al generar múltiples instancias o particiones de los sistemas operativos “virtuales” alojados dentro del mismo sistema de host físico. Esta estrategia además ofrece otras ventajas muy importantes, como el aprovisionamiento dinámico de recursos y buena capacidad de recuperación ante desastres. Se constituye entonces una herramienta poderosa para garantizar, dentro del prototipo, una emulación adecuada del comportamiento de un clúster de alto rendimiento y así posibilitar la creación de una metodología de administración que permita la utilización completa de su capacidad de procesamiento en la creación de conocimiento científico.

DESCRIPCION DEL PROBLEMA

Por décadas, la computación de alto desempeño ha permitido a investigadores y científicos la utilización de modelos matemáticos y simulaciones para el

descubrimiento de nuevos hallazgos científicos o el desarrollo de nuevos productos, mejores y más amables con el ambiente.

Por esta razón, la primera percepción de las organizaciones que no contaban en su ADN con equipos de investigadores era que la computación de alto desempeño era un espacio reservado para universidades y centros de investigación.

Sin embargo, el advenimiento de equipos de cómputo de alto rendimiento más asequibles para las organizaciones, la proliferación plataformas de computación más abiertas y colaborativas y la explosión de los datos en el entorno de la empresarial han generado un cambio en la apreciación de la computación de alto desempeño como una herramienta necesaria en un mayor número de organizaciones. Y esta tendencia va en aumento.

De igual manera, el entorno empresarial es cada vez más competitivo y exigente, consecuencia esto de la globalizado y la ausencia de las barreras tradicionales en el acceso a la información y de las comunicaciones, demandando de las organizaciones decisiones más acertadas que mejoren su ejecución y les den los resultados esperados con una mayor tasa de retorno. Estas condiciones de mercado han facilitado el camino de la analítica de datos (Data Analytics) como herramienta para sustentar la toma de decisiones, las acciones a emprender, evaluar su impacto o predecir comportamientos.

FORMULACIÓN DEL PROBLEMA

La Institución Universitaria Politécnico Grancolombiano debe ofrecer a sus estudiantes y al mercado laboral en Colombia y la región, la posibilidad de preparar profesionales y tecnólogos formados en tecnologías de computación de alto desempeño y analítica de datos. Sin embargo, estas tecnologías requieren de un alto número de recursos para brindar una experiencia efectiva y enriquecedora.

¿Cómo crear un prototipo que facilite la administración de la computación de alto desempeño y permita el procesamiento adecuado de la información que surge a través del desarrollo investigativo?

OBJETIVOS

OBJETIVO GENERAL

Desarrollar un prototipo de computación de alto desempeño, como modelo para su utilización en la analítica de datos usando componentes de máquinas virtuales, para el caso específico de este proyecto, y software abierto que haga viable la puesta en marcha el prototipo con datos de prueba

OBJETIVOS ESPECIFICOS

Evaluar las alternativas disponibles en el mercado para llevar a cabo la implementación tanto de la infraestructura de computación de alto desempeño como de la analítica de datos definiendo los criterios de asequibilidad que se piensan para las empresas.

Seleccionar los elementos que harán parte del prototipo de computación de alto desempeño y la analítica de datos acorde con los criterios definidos en la evaluación de alternativas a la vez que se define un caso de uso que será viable de presentar en este prototipo.

Construir un prototipo de computación de alto desempeño y analítica de datos con los componentes seleccionados para el caso de uso predefinido que sirva como prueba de concepto de algunos de los hallazgos de esta investigación

Elaborar un documento de grado y extraer de este un artículo de resultados y publicarlo en una revista científica.

JUSTIFICACIÓN DE LA INVESTIGACIÓN

El desarrollo de un prototipo de computación de alto desempeño para su utilización en el área de la analítica de datos permite establecer un escenario renovado de la aplicación de los modelos de computación distribuida y procesamiento colaborativo que se presenta en la asignatura de sistemas distribuidos.

La investigación conducente a la creación de este prototipo también permite identificar otras áreas de conocimiento relacionadas con los sistemas distribuidos y el procesamiento colaborativo, de manera que los estudiantes de pregrado de Ingeniería de Sistemas de la Institución tendrán un referente adicional sobre los campos en los que se puede desarrollar un egresado de la Institución con un interés profesional no sólo en el área de los sistemas distribuidos sino también en programación, bases de datos, análisis de datos, entre otros.

Como se advirtió anteriormente, las organizaciones están haciendo frente a una coyuntura del mercado que les exige actuar más rápido y responder más adecuadamente a las necesidades de sus clientes. Esta situación les demanda incrementar el uso de técnicas analíticas de datos que les permitan la toma de decisiones claras y oportunas. La universidad no puede estar ausente de esta necesidad de las organizaciones y por el contrario, por la naturaleza del

requerimiento, es la primera llamada a facilitar profesionales preparados para enfrentar estos retos de la ingeniería de sistemas.

1. COMPUTACIÓN DE ALTO DESEMPEÑO

1.1 ¿QUÉ ES COMPUTACIÓN DE ALTO DESEMPEÑO?

Generalmente, se refiere la Computación de Alto Desempeño (en adelante HPC, por sus siglas en inglés) como la práctica de agregar poder de computo de manera que se pueda obtener un mayor desempeño, con el propósito de resolver grandes problemas en la ciencia, la ingeniería o los negocios.

Como se indica, HPC se utiliza para resolver una serie de preguntas complejas en ciencias computacionales e intensivas en datos. Estas preguntas incluyen la simulación y modelado de fenómenos físicos, como el cambio climático, la producción de energía, diseño de fármacos, la seguridad mundial; el diseño de materiales; el análisis de grandes conjuntos de datos, como los de la secuenciación del genoma, la observación astronómica y la ciber-seguridad y el intrincado diseño de productos de ingeniería, tales como aviones.

Las simulaciones numéricas en todos los campos de la ciencia son ahora un ingrediente necesario en la investigación y el desarrollo, que ofrece la posibilidad de experimentos virtuales que se vuelven tan importantes como los experimentos reales. Por otro lado, los modelos teóricos de la mayoría de los sistemas complejos estudiados hoy en día por lo general requieren simulaciones numéricas sofisticadas con el fin de comparar sus predicciones con los datos experimentales. Por lo tanto, la ciencia computacional, como se designa a la investigación relacionada con las simulaciones numéricas y experimentos virtuales, es ahora un enfoque bien establecido para la investigación, que pone en igualdad de condiciones a la teoría y la experimentación.

Es claro y bien documentado que HPC se puede utilizar para generar conocimiento que de otra manera no sería posible. Las simulaciones pueden aumentar o sustituir los experimentos costosos, peligrosos o imposibles. Además, en el ámbito de la simulación, HPC tiene el potencial para sugerir nuevos experimentos que se escapan de los parámetros de lo observable.

1.2 EVOLUCIÓN DE LA COMPUTACIÓN DE ALTO DESEMPEÑO

Tras la aparición de las computadoras en la década de 1940, dos importantes compañías como International Business Machines Corporation (IBM) y Control Data Corporation (CDC) se involucraron en varios proyectos de investigación de diversas universidades de los Estados Unidos para crear la computadora con mayor “velocidad” de procesamiento, en los términos de la década de 1960.

Ya la computadora Atlas de la Universidad de Manchester se establecía en 1962 como la primera supercomputadora del mundo cuando Seymour Cray fue

comisionado por CDC para utilizar el paralelismo para alcanzar picos de desempeño computacional superiores.

El resultado del trabajo de Cray fue la computadora CDC 6600, puesta en el mercado en 1964, que sobrepasaba la capacidad de cómputo de sus rivales en más de 10 veces y pasaba de usar transistores de germanio a transistores de silicio, que adicionalmente ayudaban a sobrellevar los problemas de sobre calentamiento con apoyo de las primeras unidades de refrigeración.

La computadora CDC 6600 marca un hito muy relevante en el desarrollo del mercado de HPC al comercializar cientos de estas unidades y sus sucesoras y acuñar definitivamente el término “supercomputadora” como sinónimo de computación de alto desempeño.

Seymour Cray dejó CDC en 1972 para fundar su propia compañía, Cray Research Inc., empresa con la que lideró el mercado de HPC y las supercomputadoras hasta finales de la década de 1980. Sin embargo, durante las últimas tres décadas, la definición de lo que se llama computación de alto desempeño ha cambiado drásticamente.

De manera premonitoria, en 1988, apareció un artículo en el Wall Street Journal titulado "Attack of the Killer Micros" que describía cómo los sistemas de computación compuestos de muchos pequeños procesadores de bajo costo pronto harían que grandes supercomputadoras fueran obsoletas. Los autores de esta teoría conducían un análisis puramente económico que se apoyaba en la dinámica del naciente mercado de los computadores personales y las estaciones de trabajo científicas, cuyos ciclos de entrega de nuevas generaciones de productos venía dándose cada 12 o 18 meses frente a los 3 o 4 años que tenían las supercomputadoras.

Esta visión se ha hecho realidad en algunos aspectos, pero no en la forma en que los proponentes originales de la teoría del "Killer Micro" previeron. En cambio, el rendimiento del microprocesador ha ganado sin descanso sobre el rendimiento del supercomputador. Esto ha ocurrido por dos razones. En primer lugar, hubo mucha más "margen de maniobra" en la tecnología para mejorar el desempeño en el área de la computadora personal. Además, una vez que las empresas de supercomputación rompieron algunas barreras técnicas, las empresas de microprocesadores pudieron adoptar rápidamente los elementos de diseño de la supercomputadoras con éxito unos pocos años más tarde.

El segundo y quizás más importante factor fue la aparición de un mercado y próspero negocio de las computadoras personales con cada vez mayores demandas de rendimiento. Con un mercado tan grande, los dólares de

investigación disponibles se vertieron en el desarrollo de procesadores de alto rendimiento de bajo costo.

Como resultado de ello casi todas las personas con acceso a una computadora tienen algún tipo de procesamiento "de alto desempeño" a su alcance. A medida que las velocidades máximas de estos nuevos ordenadores personales aumentan, estos enfrentan los mismos retos de rendimiento que encuentran típicamente los supercomputadores.

1.3 COMPUTADORAS DE ALTO DESEMPEÑO

Por definición, las supercomputadoras son las computadoras más rápidas y más poderosas disponibles, y en la actualidad el término se refiere a máquinas con cientos de miles de procesadores. Las computadoras personales (PC) lo suficientemente pequeñas en tamaño y costo para ser utilizados por un individuo, pero lo suficientemente potentes para aplicaciones científicas y de ingeniería avanzada, también pueden ser computadoras de alto desempeño.

El estudio tradicional de HPC define computadoras de alto desempeño como máquinas con un buen equilibrio entre los siguientes elementos:

- Unidades funcionales de múltiples etapas (Pipeline).
- Múltiples unidades de procesamiento central (CPU) (máquinas paralelas).
- Múltiples núcleos.
- Registros centrales rápidos.
- Memorias muy grandes y rápidas.
- Comunicación muy rápida entre las unidades funcionales.
- Procesadores vectoriales, de vídeo o matriz.
- Software que integra lo anterior con eficacia.

Sin embargo, de acuerdo con la arquitectura, disposición y construcción, de los sistemas de cómputo y los elementos asociados con ellos, actualmente se suelen clasificar estos sistemas de alto desempeño en:

- Supercomputadores de procesamiento paralelo masivo (MPP): un supercomputador MPP requiere del uso de interconexiones, normalmente propietarias del fabricante, para soportar ya sea memoria compartida distribuida o un sistema de imagen única.
- Clúster: agrupaciones de computadores independientes interconectados por redes de alta velocidad, cada uno corriendo su propia copia del sistema operativo y sirviendo a un único propósito común.
- Sistemas de multiprocesamiento simétrico (SMP): arquitectura del hardware un computador de múltiples procesadores en el que dos o más

procesadores idénticos están conectados a una única memoria principal compartida y están controlados por una única instancia de sistema operativo.

- Constelación: clúster de grandes nodos SMP, donde el número de procesadores por nodo es mayor que el número de nodos.

En general, el panorama actual de los sistemas de computación de alto desempeño se encuentra dominado por los sistemas con arquitecturas en clúster con un segundo lugar de los supercomputadores de arquitectura MPP.

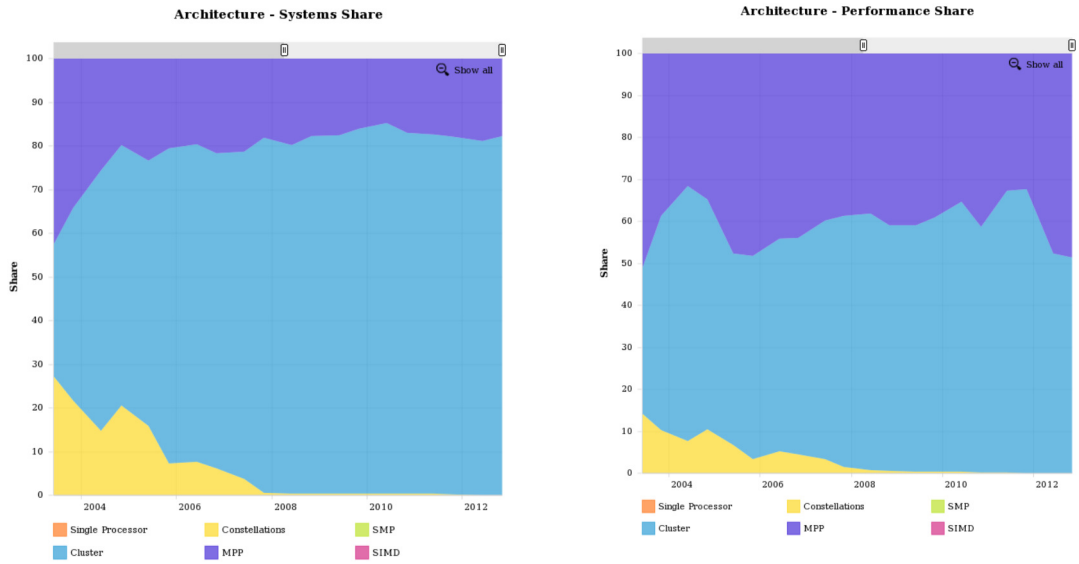

Figura 1 - Distribución de sistemas y desempeño por arquitectura. Fuente www.Top500.org

De la misma manera, el uso del sistema operativo Unix y otros sistemas propietarios en el mercado de los supercomputadores han decrecido notablemente desde el 2003 y en la actualidad, el universo de los supercomputadores y de los sistemas HPC, se encuentra dominado por el sistema operativo Linux. Aunque es importante recalcar que la dispersión de las distribuciones dentro de este segmento no permite mencionar un claro vencedor.

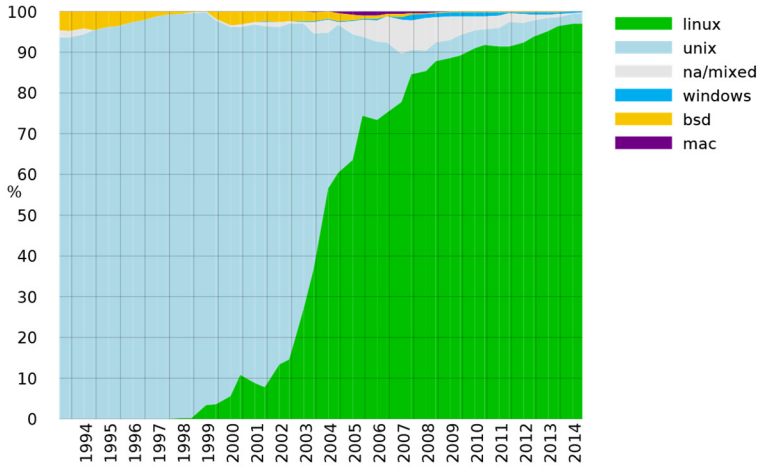

Figura 2 - Distribución de sistema operativo en HPC. Fuente www.Top500.org

1.4 RESTRICCIONES EN LA CONSTRUCCIÓN DE SISTEMAS HPC

Ya sea que se trate de un sistema de arquitectura de Procesamiento Paralelo Masivo o de un Clúster de computadores individuales, en la construcción de los sistemas HPC se imponen una serie de restricciones inherentes a la agregación de poder de cómputo:

- Limitaciones por el tamaño físico
- Consumo de energía
- Excesivo calor generado por la densidad del empaquetado y la alta frecuencia de intercambio
- Necesidad de refrigeración para alojar y correr el equipamiento agregado
- Disparidad entre la velocidad de reloj del procesador y los dispositivos periféricos cercanos (memoria, E/S, buses)
- Ampliación de la brecha entre relojes de procesador y de memoria DRAM
- Rendimiento de la red

1.4.1 MANEJO DEL ESPACIO FÍSICO

La disponibilidad del espacio físico con que se cuenta para construir el sistema HPC es la primera de las restricciones inherentes a la agregación de mayor poder de cómputo.

Desde la década de 1970 los desarrolladores de micro computadores ya estaban poniéndolos en una tarjeta y empaquetándolos en un gabinete estándar de 19 pulgadas. Luego, en la década de 1980, tras la aparición de la arquitectura VMEbus se define una interfaz de computadora que incluía la aplicación de un computador a nivel de tarjeta instalada en un plano posterior del chasis con múltiples ranuras para tarjetas acoplables para proporcionar E/S, memoria o capacidad de cómputo adicional.

Más tarde se hace la definición del estándar CompactPCI en la que se desarrolló una estructura de chasis/cuchilla. Común entre estos equipos basados chasis fue el hecho de que todo el chasis era un solo sistema. Mientras que un chasis puede incluir múltiples elementos de cómputo para proporcionar el nivel deseado de rendimiento y redundancia, siempre había una placa a cargo, una tarjeta maestra coordinando el funcionamiento de todo el sistema.

El primer sistema de cómputo que hacía uso de una estructura chasis/cuchillas fue comercializado en 2001 con el apelativo comercial de servidor blade y ya para el 2002 se hacia la primera presentación de un supercomputador de arquitectura de clúster con 240 servidores blade.

Esta construcción de sistemas HPC que hace uso de definiciones como CompactPCI y sus posteriores mejoras y sistemas blade se ha extendido entre los fabricantes de supercomputadores y los diseñadores de clústeres para HPC. Una representación de esto se puede ver en la siguiente figura en la que se representa la construcción de un supercomputador IBM BlueGene.

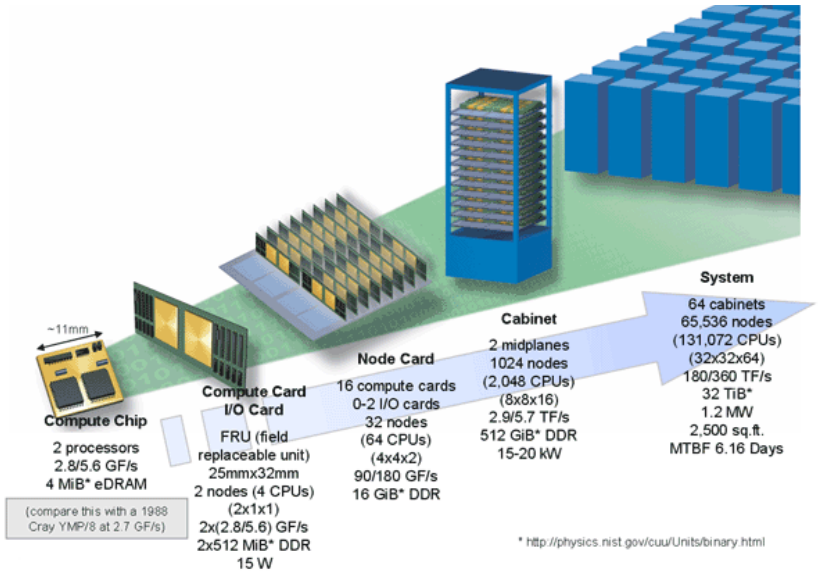

Figura 3 - Jerarquía de las unidades de procesamiento de Blue Gene. Fuente: http://es.wikipedia.org/wiki/Blue_Gene

Si bien un sistema paralelo masivo centralizado incluye las eficiencias de agrupamiento diseñadas por el fabricante de estos grandes supercomputadores pueden fácilmente alcanzar dimensiones del orden de los 400 m^2, por lo que en general la construcción de tales sistemas viene de la mano con la construcción de las instalaciones que lo albergarán una vez entre en operación.

Por otra parte, el uso de un clúster de computadores en un sitio cerrado plantea un reto mayor en cuanto al espacio necesario para su alojamiento, pues si bien prácticamente cualquier computador puede ser agregado al agrupamiento esta flexibilidad en la agregación del nuevo poder de computo pasa una factura mayor en el espacio físico que se requiere para su inclusión.

1.4.2 CONSUMO DE ENERGIA Y EFICIENCIA DEL CONSUMO

Otro de los grandes costos que trae la agregación de mayor poder de cómputo tiene que ver directamente con la cantidad de energía eléctrica que se requiere para su funcionamiento.

Mucho tiempo ha pasado desde el célebre Cray-1 que consumía 115 kilovatios, los niveles de consumo del top 10 de los supercomputadores ha sobrepasado ya con facilidad el megavatio de potencia por lo que a partir del 2005 se ha venido siguiendo con detenimiento la eficiencia de consumo de los supercomputadores.

La introducción de Blue Gene por parte de IBM estableció otra aproximación a las necesidades de eficiencia computacional y consumo de energía. Negociando la velocidad de los procesadores por bajo consumo de energía, Blue Gene utilizaba núcleos PowerPC integrados de baja frecuencia y baja potencia con aceleradores de punto flotante. Mientras que el rendimiento de cada chip fue relativamente bajo, el sistema podría lograr un mejor rendimiento de relación de energía, para aplicaciones que podrían utilizar un mayor número de nodos.

El manejo del calor es un problema importante en los dispositivos electrónicos complejos que afecta a los sistemas de computadoras en varias formas. Los problemas de diseño de potencia y disipación térmica de potencia en la CPU para la supercomputación superan los de tecnologías de refrigeración del ordenador tradicional. La eficiencia energética de los sistemas de cómputo es generalmente medida en términos de operaciones de punto flotante por vatio (flops/W).

Ahora los principales supercomputadores tiene promedios de eficiencia de consumo de 750 Mflops/W, lo que significa que un incremento superior al 300% en un periodo 3 años.

1.4.3 JERARQUÍA DE MEMORIA PARA ATACAR LA BRECHA ENTRE RELOJES

Un modelo idealizado de la arquitectura de computadora es una CPU ejecutando secuencialmente un chorro (stream) de instrucciones y leyendo de un bloque continuo de memoria. El mundo real es más complicado que esto. En primer lugar, las matrices no se almacenan en bloques, sino más bien en orden lineal. Por ejemplo, en Fortran esto es ordenado por columnas:

M (1, 1) M (2, 1) M (3, 1) M (1, 2) M (2, 2) M (3, 2) M (1, 3) M (2, 3) M (3, 3),

mientras que en Python, Java y C está en orden por las filas:

M (0, 0) M (0, 1) M (0, 2) M (1, 0) M (1, 1) M (1, 2) M (2, 0) M (2, 1) M (2, 2).

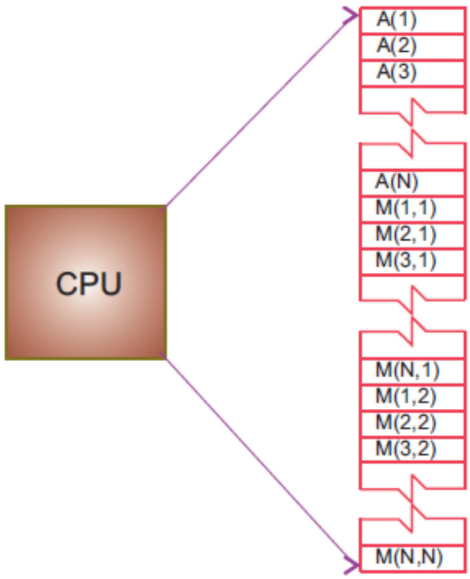

Figura 4 - La disposición lógica de la CPU y la memoria que muestra una matriz de Fortran A (N) y la matriz M (N, N) cargado en la memoria. Fuente: Computer Science Oregon State University.

En segundo lugar, los valores para los elementos de matriz pueden incluso no estar en el mismo lugar físico. Algunos pueden estar en la memoria RAM, algunos en el disco, algunos en caché, y algunos en la CPU. Para darle a algunas de estas palabras más significado, a continuación se muestran los modelos simples de la arquitectura de memoria de una computadora de alto rendimiento.

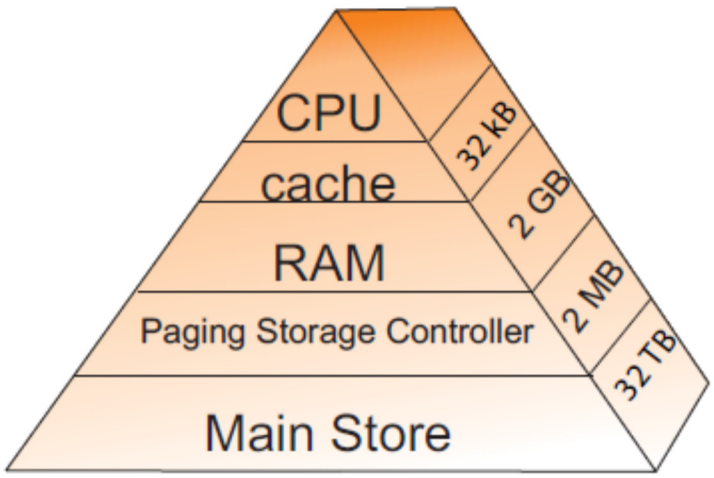

Figura 5 - Jerarquía de memoria típica para un único procesador, ordenador de alto rendimiento (B = bytes, K, M, G, T = kilo, mega, giga, tera). Fuente: Computer Science Oregon State University.

Esta disposición jerárquica surge de un esfuerzo por equilibrar la velocidad y el costo de memoria rápida y cara, complementada con memoria lenta, menos costosa. La arquitectura de memoria puede incluir los siguientes elementos:

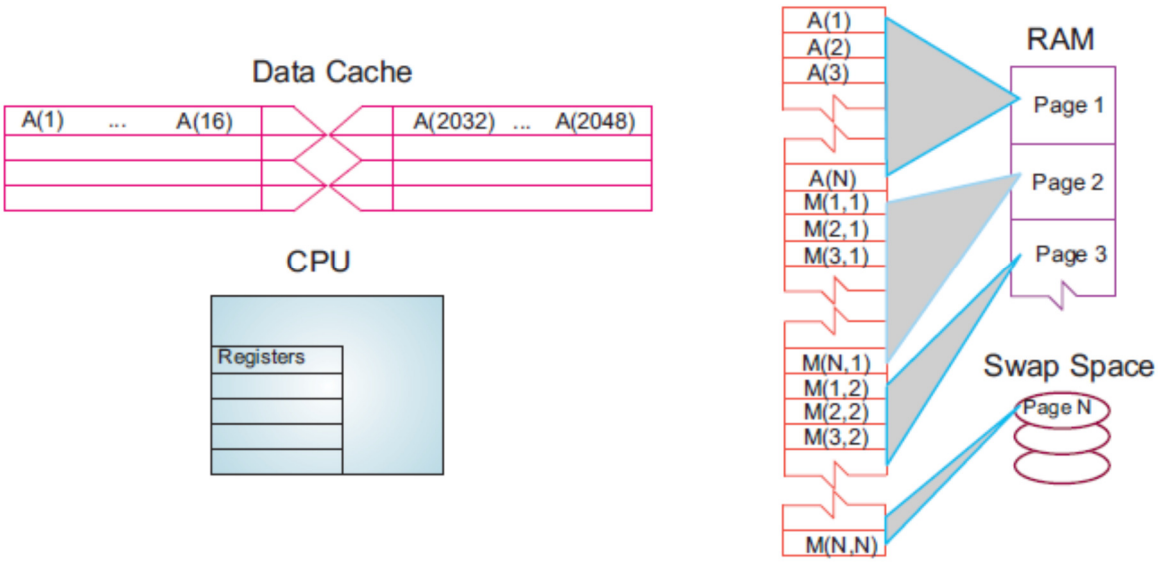

Figura 6 - Los elementos de la arquitectura de memoria de una computadora en el proceso de manipulación del almacenamiento de una matriz. Fuente: Computer Science Oregon State University.

CPU: Unidad central de proceso, la parte más rápida del equipo. La CPU se compone de un número de unidades de memoria de muy alta velocidad llamados registros que contienen las instrucciones enviadas al hardware para hacer cosas como ir a buscar, almacenar y operar sobre los datos. Por lo general hay registros independientes para instrucciones, direcciones, y operando (datos actuales). En muchos casos, la CPU también contiene algunas partes especializadas para acelerar el procesamiento de los números de punto flotante.

Caché: Una pequeña sección de memoria que contiene instrucciones, direcciones y datos en su paso entre los registros muy rápidos de la CPU y la memoria RAM más lenta. La memoria principal también se llama RAM dinámica (DRAM), mientras que la memoria caché se llama RAM estática (SRAM). Si el caché se utiliza correctamente, puede reducir en gran medida el tiempo que la CPU espera que los datos sean traídos de la memoria.

Líneas de caché: Los datos transferidos desde y hacia la memoria caché o CPU son agrupados en caché o líneas de datos. El tiempo que se necesita para llevar los datos de la memoria a la memoria caché se llama latencia.

RAM: memoria de acceso aleatorio o memoria central se encuentra en el medio de la jerarquía de memoria. La RAM es rápida porque sus direcciones se puede acceder directamente en orden aleatorio, y porque no se necesitan dispositivos mecánicos para leerlo.

Páginas: la memoria central está organizada en páginas, que son bloques de memoria de longitud fija. El sistema operativo etiqueta y organiza sus páginas de memoria al igual que hacemos con las páginas de un libro; éstas están numeradas y no se pierden de vista con una tabla de contenidos. Los tamaños de página típicos son de 4 a 16 kB, pero en supercomputadoras pueden estar en el rango de MB.

Disco duro: Por último, en la parte inferior de la pirámide de la memoria está el almacenamiento permanente en discos magnéticos o dispositivos ópticos. Aunque los discos son muy lentos, en comparación con la memoria RAM, pueden almacenar grandes cantidades de datos y, a veces, compensan sus velocidades más lentas utilizando una caché propia, paginación en el controlador de almacenamiento.

La memoria virtual: Fiel a su nombre, se trata de una parte de la memoria que no encontrará en nuestras cifras, ya que es virtual. Actúa como RAM, pero reside en el disco.

Cuando hablamos de memoria rápida y lenta estamos utilizando una escala de tiempo establecido por el reloj de la CPU. Para ser específicos, si el equipo tiene

una velocidad de reloj o tiempo de ciclo de 1 ns, esto significa que se podría realizar un mil millones de operaciones por segundo si pudiera tener los datos necesarios con la suficiente rapidez. Mientras que por lo general toma 1 ciclo transferir datos desde la memoria caché a la CPU, los otros tipos de memorias son mucho más lentos. En consecuencia, puede acelerar su programa teniendo todos los datos necesarios para la CPU cuando trata de ejecutar sus instrucciones; de lo contrario la CPU puede dejar caer su cálculo y pasar a otras tareas mientras sus datos se transfieren de la memoria inferior.

La memoria virtual permite a un programa utilizar más páginas de memoria de las que pueden caber físicamente en la memoria RAM de una sola vez. Una combinación de sistema operativo y hardware mapea esta memoria virtual en páginas con longitudes típicas de 4 a 16 kB. Las páginas que no se usan actualmente se almacenan en la memoria más lenta en el disco duro y se ponen en memoria rápida sólo cuando sea necesario. La ubicación de memoria separada para esta conmutación se conoce como espacio de intercambio (Swap).

Gracias a la memoria virtual, es posible ejecutar programas en ordenadores pequeños que de otro modo requerirían máquinas más grandes (o extensa reprogramación). El precio que paga por la memoria virtual es una desaceleración de la velocidad de su programa de orden de magnitud cuando se invoca en realidad la memoria virtual. Pero esto puede ser barato en comparación con el tiempo que tendría que pasar para volver a escribir el programa para que se ajuste en la memoria RAM o el dinero que tendría que gastar para comprar una computadora con suficiente memoria RAM para su problema.

La memoria virtual también permite la multitarea, la carga simultánea en la memoria de más programas de los que caben físicamente en la memoria RAM. Aunque la conmutación subsiguiente entre las aplicaciones utiliza ciclos de computación, al evitar largas esperas, mientras que una aplicación se carga en memoria, la multitarea aumenta el rendimiento total y permite un entorno informático mejorado para los usuarios. Por ejemplo, es la multitarea la que permite a un sistema de ventanas, como Linux, OS X o Windows, proporcionarnos múltiples ventanas. A pesar de que cada aplicación de ventana utiliza una cantidad razonable de memoria, sólo la única aplicación que actualmente recibe la entrada en realidad debe residir en la memoria; el resto están paginadas en el disco.

Esto explica por qué es posible que note un ligero retraso al cambiar a una ventana inactiva; las páginas para el programa activo ahora están siendo colocadas en la memoria RAM y la aplicación menos utilizada todavía en la memoria al mismo tiempo se está paginando.

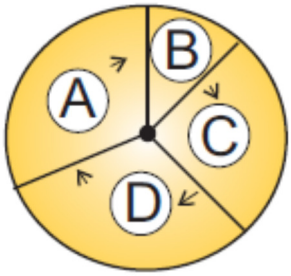

Figura 7 - La multitarea de cuatro programas en la memoria en un momento en el que los programas se ejecuta en orden round-robin. Fuente: Computer Science Oregon State University.

1.4.4 VENTAJAS DE LOS PROCESADORES DE MÚLTIPLES NÚCLEOS

El tiempo presente está experimentando un rápido aumento en la inclusión de chips de doble núcleo, cuatro núcleos, o incluso de dieciséis núcleos como el motor de cálculo de los ordenadores. Como se ve en la figura, un chip de doble núcleo tiene dos CPUs en un circuito integrado con una interconexión compartida y un caché de nivel-2 compartida. Este tipo de configuración con dos o más procesadores idénticos conectados a una sola memoria principal compartida está asociada con el multiprocesamiento simétrico o SMP.

Aunque los chips de múltiples núcleos fueron diseñados para los juegos de video y la precisión simple, están encontrando uso en la computación científica ya que se emplean nuevas herramientas, algoritmos y métodos de programación. Estos chips alcanzan velocidades integrales con menos calor y más eficiencia energética que los chips de un solo núcleo, cuya generación de calor les limita a velocidades de reloj de menos de 4 GHz. A diferencia de varios chips de un solo núcleo, los chips de varios núcleos usan menos transistores por CPU y por tanto son más fáciles de hacer y más fríos para funcionar.

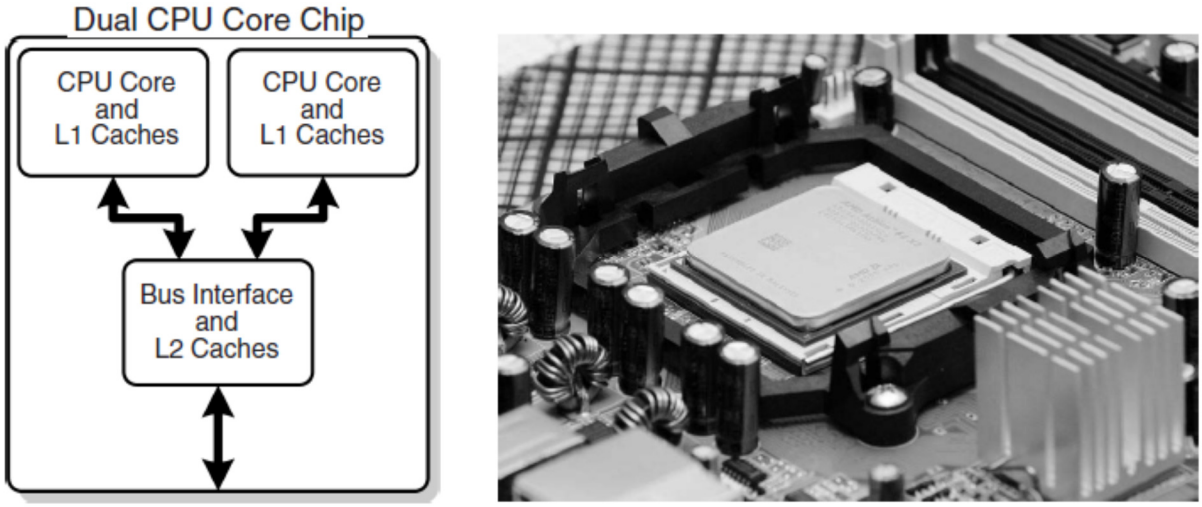

Figura 8 - Izquierda: Una visión genérica del procesador de doble núcleo Intel core-2, con caché local de nivel 1 en la CPU y una caché compartida de nivel 2 en el chip. Derecha: El procesador AMD Athlon 64 X2 3600 CPU de doble núcleo. Fuente: http://en.wikipedia.org/wiki/Multi-core_processor

El paralelismo está integrado en un chip múltiples núcleos ya que cada núcleo puede ejecutar una tarea diferente. Sin embargo, ya que los núcleos generalmente comparten el mismo canal de comunicación y el caché de nivel-2, hay la posibilidad de un cuello de botella de comunicación si ambas CPU utilizan el bus al mismo tiempo. Por lo general, el usuario no tiene que preocuparse por esto, pero los desarrolladores de los compiladores y software si deben, para que su código se ejecute en paralelo. Los compiladores Intel modernos utilizan automáticamente los múltiples núcleos, con MPI, incluso tratando cada núcleo como un procesador independiente.

1.5 LA RELEVANCIA DE LA COMPUTACIÓN EN PARALELO

No hay duda de que los avances en el hardware para la computación paralela son impresionantes. Desafortunadamente, el software que acompaña al hardware a menudo parece atrapado en la década de 1960. El paso de mensajes tiene demasiados detalles técnicos de que preocuparse para los científicos que hacen uso de la aplicación y (por desgracia) requiere codificación en un nivel elemental que recuerda a los primeros días de la computación.

Sin embargo, la creciente aparición de agrupaciones en las que los nodos son multiprocesadores simétricos ha llevado al desarrollo de compiladores sofisticados que siguen modelos de programación más simples; por ejemplo, compiladores de particionado de espacio de direcciones globales como Co-Array Fortran, Unified Parallel C, y Titanium. En estos enfoques el programador ve una matriz global de datos y luego manipula estos datos como si fueran contiguos. Por supuesto, los

datos realmente se distribuyen, pero el software se encarga de esto fuera de la vista del programador. Aunque un programa de este tipo puede hacer uso de los procesadores con menos eficiencia que lo haría un programa codificado a mano, es mucho más fácil que el rediseño de su programa.

La pregunta de si vale la pena su tiempo para hacer un programa más eficiente depende del problema en cuestión, el número de veces que el programa se llevará a cabo, y los recursos disponibles para la tarea. En cualquier caso, si cada nodo del equipo tiene varios procesadores con una memoria compartida y hay un número de nodos, entonces será necesario algún tipo de un modelo de programación híbrida.

Aunque un pequeño problema no vale la pena el esfuerzo con el fin de obtener un tiempo de ejecución más corto, vale la pena invertir el tiempo para ganar algo de experiencia en computación paralela. En general, la computación paralela mantiene la promesa de que permite obtener resultados más rápidos, para resolver los problemas más grandes, para ejecutar simulaciones en resoluciones más finas, o para modelar los fenómenos físicos de manera más realista; pero se necesita un poco de trabajo para lograr esto.

Los procesadores en un computador paralelo se colocan en los nodos de una red de comunicación. Cada nodo puede contener una CPU o un pequeño número de CPU, y la red de comunicación puede ser interna o externa a la computadora. Una forma de clasificar las computadoras paralelas es por el enfoque que emplean en el manejo de datos e instrucciones. Desde este punto de vista hay tres tipos de máquinas:

Instrucción individual, datos individuales (SISD): Estos son los clásicos computadores de serie (von Neumann) ejecutando una única instrucción en un único flujo de datos antes de la siguiente instrucción y la siguiente secuencia de datos sea encontrada.

Instrucción individual, múltiples datos (SIMD): Aquí las instrucciones se procesan a partir de una sola corriente, pero las instrucciones actúan simultáneamente en varios elementos de datos. Generalmente los nodos son simples y relativamente lentos, pero son grandes en número.

Múltiples instrucciones, múltiples datos (MIMD): En esta categoría cada procesador funciona independientemente de los otros con instrucciones y datos independientes. Estos son los tipos de máquinas que emplean paquetes de paso de mensajes, como MPI, para comunicarse entre los procesadores. Pueden ser una colección de estaciones de trabajo conectadas a través de una red, o máquinas más integradas con miles de procesadores en las tarjetas internas. Estos equipos, que no disponen de un espacio de memoria compartida, también

se llaman multi-computadoras. Aunque este tipo de computadoras son algunos de los más difíciles de programar, su bajo costo y la eficacia de ciertas clases de problemas han llevado a que son el tipo dominante de ordenador paralelo en la actualidad.

El funcionamiento de los programas independientes en un computador paralelo es similar a la función multitarea. En la multitarea varios programas independientes residen en la memoria de la computadora al mismo tiempo y comparten el tiempo de procesamiento en un round-robin u orden de prioridad. En un equipo de SISD, sólo un programa se ejecuta a la vez, pero si otros programas están en la memoria, entonces no se necesita mucho tiempo para cambiar a ellos. En multiprocesamiento estos trabajos pueden todos correr al mismo tiempo, ya sea en diferentes partes de la memoria o en la memoria de los distintos equipos. Claramente, el multiprocesamiento se complica si procesadores separados están operando en diferentes partes del mismo programa porque entonces la sincronización y el equilibrio de carga (manteniendo todos los procesadores igualmente ocupados) son preocupaciones reales.

1.6 PROGRAMACIÓN DE MEMORIA DISTRIBUIDA

Una aproximación al procesamiento concurrente que, porque se construye desde PCs básicos, ha ganado aceptación dominante son los de memoria distribuida. En este, cada procesador tiene su propia memoria y los datos se intercambian entre procesadores a través de un conmutador de alta velocidad y la red. Los datos intercambiados o pasados entre los procesadores se han codificado desde y hacia direcciones y se llaman mensajes.

Los grupos de PCs o estaciones de trabajo que constituyen este arreglo o clúster son ejemplos de los ordenadores de memoria distribuida. La característica unificadora de un clúster es la integración de componentes informáticos y de comunicaciones altamente replicados en un solo sistema, con cada nodo todavía siendo capaz de funcionar de forma independiente. En un clúster, los componentes son productos básicos destinados a un mercado general, al igual que la red de comunicación y su interruptor de alta velocidad (interconexiones especiales son utilizados por los principales fabricantes comerciales, pero no son baratas). Nota: Un grupo de ordenadores conectados por una red también puede ser llamado un clúster pero a menos que estén diseñados para el procesamiento paralelo, con el mismo tipo de procesador utilizado en varias ocasiones y con sólo un número limitado de procesadores (el extremo frontal) sobre el cual los usuarios pueden ingresar, no se suele llamar un clúster.

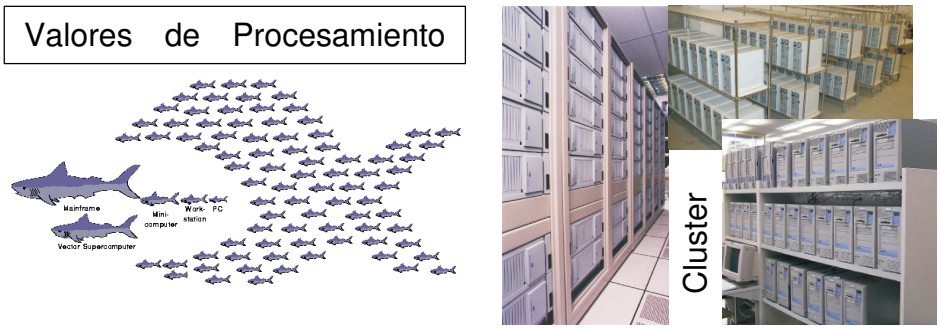

Figura 9 - Dos vistas de la computación paralela moderna (cortesía de Yuefan Deng).

Para que un programa de paso de mensajes pueda tener éxito, los datos deben ser divididos entre los nodos de modo que, al menos por un tiempo, cada nodo tenga todos los datos que necesita para ejecutar una subtarea independiente. Cuando un programa inicia la ejecución, los datos se envían a todos los nodos. Cuando todos los nodos han completado sus subtareas, intercambian datos de nuevo a fin de que cada nodo tenga un nuevo conjunto completo de datos para realizar la siguiente subtarea. Este ciclo repetido de intercambio de datos seguido por el procesamiento continúa hasta que se complete la tarea completa.

Los programas de paso de mensajes de tipo MIMD también son programas de único-programa, múltiples-datos, lo que significa que el programador escribe un único programa que se ejecuta en todos los nodos. A menudo un programa host independiente comienza los programas en los nodos, lee los archivos de entrada y organiza la salida.

1.7 EL PROBLEMA DEL DESEMPEÑO PARALELO

Imagine una fila de la cafetería en la que todos los servidores parecen estar trabajando duro y sin embargo rápidamente el dispensador de la salsa de tomate tiene cierto problema bloqueando parcialmente su producción y así todo el mundo en línea deben esperar a que los amantes de la salsa de tomate en la delantera aderecen su comida antes de continuar. Este es un ejemplo de la etapa más lenta en un proceso complejo de la determinación de la tasa global. Una situación análoga se mantiene para el procesamiento en paralelo, en el que el dispensador de la salsa de tomate puede ser una parte relativamente pequeña del programa que se puede ejecutar solamente como una serie de pasos en serie. Debido a que el cálculo no puede avanzar hasta que se completen estos pasos en serie, esta pequeña parte del programa puede terminar siendo el cuello de botella del programa.

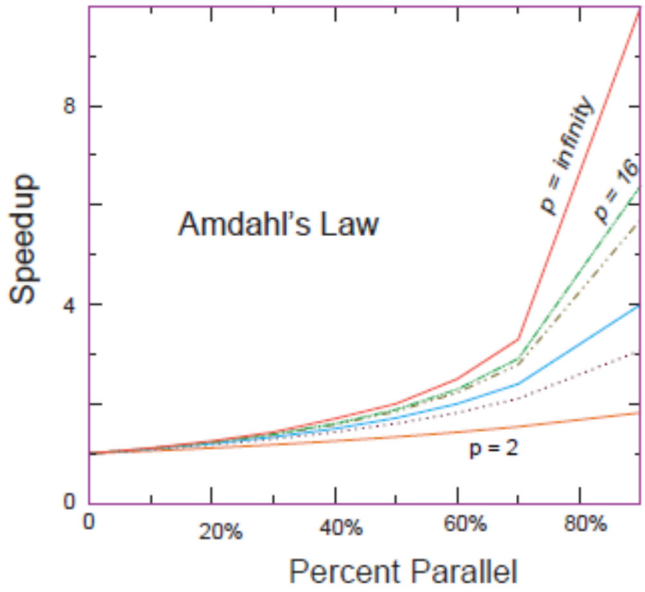

Figura 10 - El aumento de velocidad máxima teórica de un programa como una función de la fracción del programa que potencialmente se puede ejecutar en paralelo. Las diferentes curvas corresponden a diferentes números de los procesadores. Fuente: Computer Science Oregon State University

La aceleración de un programa no será significativo a menos que se pueda conseguir que ~ 90% de este se ejecute en paralelo, e incluso entonces la mayor parte del aumento de velocidad probablemente se obtendrá con sólo un pequeño número de procesadores. Esto significa que se necesita tener un problema intenso computacionalmente para hacer que la paralelización valga la pena, y que es una de las razones por las que algunos defensores de las computadoras paralelas con miles de procesadores sugieren que no se aplican las nuevas máquinas a viejos problemas, sino más bien buscar nuevos problemas que son a la vez lo suficientemente grandes y muy adecuados para el procesamiento masivamente paralelo para que el esfuerzo valga la pena.

La ecuación que describe el efecto sobre el aumento de velocidad del equilibrio entre partes serie y paralelo de un programa se conoce como la ley de Amdahl. Sea p = el número de CPUs, T1 = tiempo para correr en 1 CPU, Tp = tiempo para correr en las p CPU.

La máxima aceleración S_p posible con el procesamiento en paralelo es, pues,

$$S_p = \frac{T_1}{T_p} \rightarrow p.$$

En la práctica, este límite no se cumple por un número de razones: alguna parte del programa es serial, conflictos de datos y de memoria se producen, la comunicación y la sincronización de los procesadores llevan tiempo, y es raro que se pueda alcanzar un equilibrio de carga perfecta entre todos los procesadores. Por el momento ignoramos estas complicaciones y nos concentramos en cómo la parte de serie del código afecta a la aceleración. Sea f la fracción del programa que potencialmente pueden ejecutarse en múltiples procesadores. La fracción 1 - f del código que no se puede ejecutar en paralelo, se debe ejecutar a través de procesamiento en serie y por lo tanto necesita tiempo:

Ts = (1 – f)T1 (tiempo serial).

El tiempo Tp gastado en los procesadores paralelos p se relaciona con Ts por

$$T_p = f\,\frac{T_1}{p}.$$

Siendo esto así, la aceleración máxima como una función de f y el número de procesadores es (Ley de Amdahl):

$$S_p = \frac{T_1}{T_s + T_p} = \frac{1}{1 - f + f/p}$$

Es evidente que el aumento de velocidad no será lo suficientemente importante como para que merezca la pena a menos que la mayor parte del código se ejecuta en paralelo (aquí es donde el 90% de la cifra en paralelo viene). Incluso un número infinito de procesadores no puede aumentar la velocidad de funcionamiento de las piezas de serie del código, por lo que corre a una velocidad del procesador. En la práctica esto significa que muchos problemas se limitan a un pequeño número de procesadores, y que sólo el 10%-20% del máximo rendimiento del ordenador es a menudo todo lo que se obtiene para aplicaciones realistas.

1.8 ESTRATEGIAS DE PARALELIZACIÓN

Una organización típica de un programa que contiene las tareas tanto de serie y en paralelo se da en el siguiente diagrama. El usuario organiza el trabajo en unidades llamadas tareas, con cada tarea asignando trabajos (hilos) a un procesador. La tarea principal controla la ejecución general, así como las subtareas que ejecutan partes independientes del programa (llamadas subrutinas paralelas, esclavos, invitados o subtareas). Estas subrutinas paralelas pueden ser subprogramas distintivos, múltiples copias del mismo subprograma, o incluso bucles for.

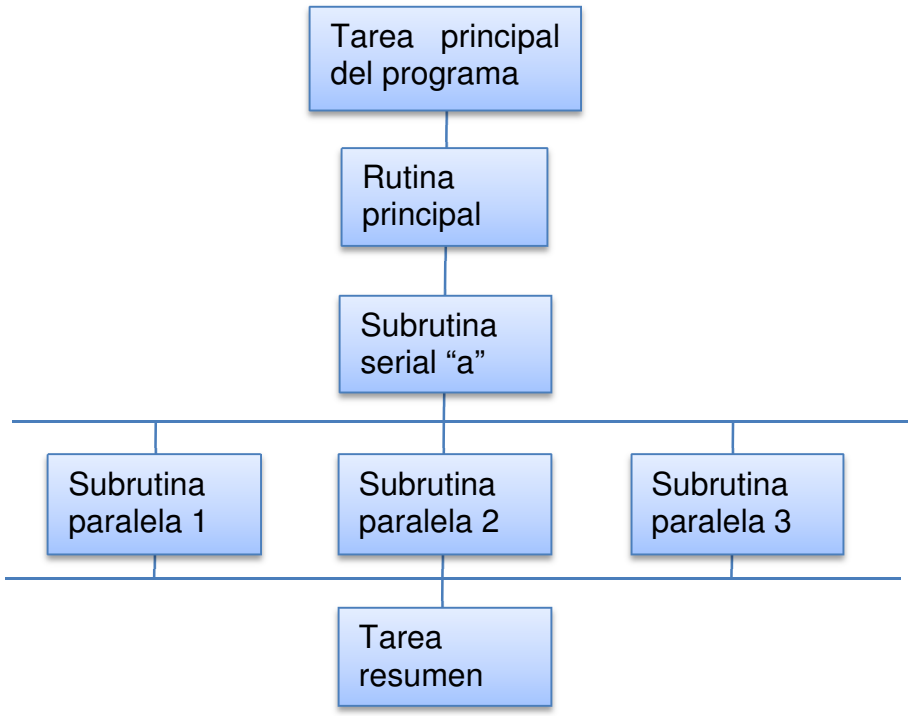

Figura 11 - Una organización típica de un programa que contiene ambas tareas serie y paralelo. Fuente: CSC Lecture Notes Louisiana State University

Es responsabilidad del programador asegurar que la ruptura de un código en subrutinas paralelas es matemáticamente y científicamente válida y es una formulación equivalente del programa original. Como ejemplo de ello, si la parte más intensiva de un programa es la evaluación de una gran matriz hamiltoniana, es posible que desee evaluar cada fila en un procesador diferente. En consecuencia, la clave para la programación paralela es identificar las partes del programa que pueden beneficiarse de la ejecución en paralelo. Para hacerlo, el programador debe entender las estructuras de datos del programa, saber en qué

orden se deben realizar los pasos de la computación, y saber coordinar los resultados generados por diferentes procesadores.

El programador ayuda a acelerar la ejecución manteniendo muchos procesadores simultáneamente ocupados y evitando conflictos de almacenamiento entre los distintos subprogramas paralelos. Esto se hace con el balanceo de carga al dividir su programa en subtareas de intensidad numérica aproximadamente igual que se ejecutará simultáneamente en diferentes procesadores. La regla de oro es: hacer la tarea con la mayor granularidad (carga de trabajo) dominante obligándola a ejecutarse en primer lugar y mantener todos los procesadores ocupados haciendo que el número dé tareas de un múltiplo entero del número de procesadores. Esto no siempre es posible.

Los hilos paralelos individuales pueden ser compartidos o datos locales. Los datos compartidos pueden ser utilizados por todas las máquinas, mientras que los datos locales son privados a sólo un hilo. Para evitar conflictos de almacenamiento, diseñe su programa para que subtareas paralelas utilicen datos que son independientes de los datos de la tarea principal y en otras tareas paralelas. Esto significa que estos datos no deben ser modificados o incluso examinados por diferentes tareas al mismo tiempo. En la organización de estas múltiples tareas, reducir los costos de la sobrecarga de comunicación mediante la limitación de la comunicación y la sincronización. Estos costos tienden a ser altos para la programación de grano fino donde se necesita mucha coordinación. Sin embargo, no elimine las comunicaciones que sean necesarias para garantizar la validez científica o matemática de los resultados.

1.9 ASPECTOS PRÁCTICOS DE PASO DE MENSAJES MIMD

Tiene sentido ejecutar sólo los códigos numéricamente más intensivos en máquinas paralelas. Frecuentemente, estos son programas muy grandes ensamblados durante varios años o décadas por un número de personas. No debería ser ninguna sorpresa, entonces, que los lenguajes de programación para máquinas paralelas son principalmente Fortran, que tiene estructuras explícitas para el compilador para paralelizar y C.

La programación paralela efectiva se hace más difícil, a medida que el número de procesadores aumenta. Los científicos en computación sugieren que lo mejor es no intentar modificar un código serial sino que es mejor volver a escribirlo desde cero utilizando algoritmos y bibliotecas de subrutinas que mejor se adapten a la arquitectura paralela. Sin embargo, esto puede implicar meses o años de trabajo, y los estudios encuentran que ~ 70% de los científicos computacionales revisan los códigos existentes en su lugar.

La mayoría de los cálculos paralelos en la actualidad se llevan a cabo en computadoras de múltiple instrucción, múltiples datos a través de paso de mensajes usando MPI.

El paralelismo tiene un precio, hay una curva de aprendizaje que requiere esfuerzo intensivo. Las fallas pueden ocurrir por una variedad de razones, sobre todo porque los entornos paralelos tienden a cambiar a menudo y consiguen quedar "encerrados" por un error de programación. Además, con varios equipos y sistemas operativos múltiples involucrados, las técnicas que ya conoce para la depuración pueden no ser eficaces.

Existe pre-condiciones para el paralelismo cuando el programa se ejecuta miles de veces entre cambios, con el tiempo de ejecución en días, y se debe aumentar significativamente la resolución de la salida o estudiar sistemas más complejos a continuación, el paralelismo es digno de consideración. De lo contrario, y en la medida de la diferencia, la paralelización de un código puede no valer la pena la inversión de tiempo.

Se debe tener en cuenta que el problema afecta el paralelismo; se debe analizar el problema en términos de cómo y cuándo se utilizan los datos, la cantidad de cálculos que requiere para cada uso, y el tipo de arquitectura del problema.

También se puede encontrar una situación perfectamente paralela cuando la misma aplicación se ejecuta de forma simultánea en diferentes conjuntos de datos, con el cálculo para cada conjunto independiente de datos (por ejemplo, la ejecución de múltiples versiones de una simulación de Monte Carlo, cada uno con diferentes semillas, o analizar los datos de los detectores independientes). En este caso sería sencillo para paralelizar esperando un rendimiento respetable.

Si la misma operación que se aplica en paralelo a múltiples partes del mismo conjunto de datos, con algunas esperas necesarias (por ejemplo, determinar las posiciones y velocidades de las partículas simultáneamente en una simulación de dinámica molecular). Se requiere un esfuerzo significativo, y a menos que logre balancear la intensidad de cálculo, el aumento de velocidad puede no valer la pena el esfuerzo pues se está ante un escenario totalmente síncrono.

Sin embargo, puede tratarse de un escenario vagamente síncrono si los diferentes procesadores hacen pedazos pequeños de la computación, pero con el intercambio de datos intermitente (por ejemplo, la difusión de las aguas subterráneas de un lugar a otro). En este caso, sería difícil para paralelizar y probablemente no vale la pena el esfuerzo.

Finalmente, cuando los datos de los pasos anteriores son procesados por los pasos posteriores, con cierta superposición de procesamiento posible (por

ejemplo, el procesamiento de datos en imágenes y en las animaciones). Gran parte del trabajo puede ser complicado, y si no equilibra la intensidad de cálculo, el aumento de velocidad puede no valer la pena el esfuerzo.

1.10 ESCALABILIDAD

En el sentido más general, la escalabilidad se define como la capacidad de manejar más trabajo a medida que el tamaño de la computadora o la aplicación crecen.

Como ya se ha indicado, el principal reto de la computación paralela es decidir la mejor manera de romper un problema en piezas individuales que puede hacer un cómputo cada uno, por separado. En un mundo ideal sería un problema escalar de manera lineal, es decir, el programa podría acelerar por un factor de N cuando se ejecuta en una máquina que tiene N nodos. (Por supuesto, como $N \rightarrow \infty$ la proporcionalidad no se puede sostener porque el tiempo de la comunicación debe entonces dominar). En la terminología presente, este tipo de escala se llama escalamiento fuerte, y se refiere a una situación en la que el tamaño del problema permanece fijo mientras que el número de número de nodos (la escala de la máquina) aumenta. Claramente entonces, el objetivo cuando se resuelve un problema que escala fuertemente es disminuir la cantidad de tiempo que se necesita para resolver el problema mediante el uso de un ordenador más potente. Estos suelen ser los problemas vinculados a la CPU y son los más difíciles de rendir en algo parecido a una aceleración lineal.

En contraste con escalamiento fuerte en el que el tamaño del problema permanece fijo, en escalamiento débil tenemos aplicaciones del tipo antes mencionados; es decir, aquellos en los que nos hacen el problema más grande y más grande que el número de procesadores aumenta. Así que aquí, tendríamos un escalamiento lineal o perfecto si pudiéramos aumentar el tamaño del problema resuelto en proporción con el número N de nodos.

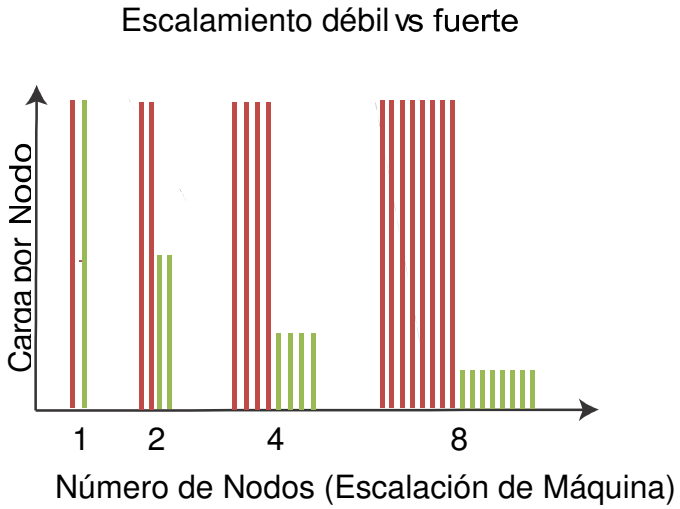

Figura 12 - Una representación gráfica de escalamiento débil vs fuerte. Fuente: CSC Lecture Notes Louisiana State University

Para ilustrar la diferencia entre escalamiento fuerte y débil, considere la anterior figura. Vemos que para una aplicación que se adapta perfectamente fuertemente el trabajo realizado en cada nodo disminuye a medida que la escala de la máquina aumenta, lo que por supuesto significa que el tiempo que tarda en completar el problema disminuye linealmente. Por el contrario, vemos que para una aplicación que se escala perfectamente débilmente, el trabajo realizado por cada nodo sigue siendo la misma que la escala de la máquina aumenta, lo que significa que estamos resolviendo problemas cada vez mayores en el mismo tiempo que se tarda en resolver los más pequeños que están en una máquina más pequeña.

Los conceptos de escalamientos débiles y fuertes son ideales que no tienden a alcanzarse en la práctica, con las aplicaciones del mundo real se tiene algo de cada uno presente. Además, es la combinación de aplicación y arquitectura del computador las que determinan el tipo de escalamiento que se produce. Por ejemplo, los sistemas de memoria compartida y de memoria distribuida, los sistemas de paso de mensajes escalan diferente. Por otra parte, una aplicación de datos en paralelo (una en la que cada nodo puede trabajar en su propio conjunto de datos independiente) por su naturaleza escalará muy débilmente.

Aplicaciones realistas tienden a tener diferentes niveles de complejidad, así que puede no ser obvia simplemente la forma de medir el aumento de "tamaño" de un problema. Como un ejemplo, se sabe que la solución de un conjunto de N ecuaciones lineales a través de la eliminación de Gauss requiere $O(N^3)$ operaciones de punto flotante (flop). Esto significa que la duplicación del número de ecuaciones no hace que el "problema" sea el doble de grande sino más bien

ocho veces más grande! Del mismo modo, si estamos resolviendo ecuaciones diferenciales parciales en una cuadrícula espacial tridimensional y una rejilla de tiempo 1-D, entonces el tamaño del problema escalaría como N^4. En este caso, duplicando el tamaño del problema significaría el aumento de N por sólo $2^{1/4}$ ~ 1.19.

1.11 PARALELISMO DE DATOS Y DESCOMPOSICIÓN DE DOMINIO

Existen dos enfoques básicos, pero muy diferentes, para la creación de un programa que se ejecuta en paralelo. En el paralelismo de tarea se descompone el programa por tareas, con diferentes tareas asignadas a diferentes procesadores, y con gran cuidado para mantener el equilibrio de cargas, es decir, mantener todos los procesadores igualmente ocupados. Es evidente que hay que entender el funcionamiento interno del programa con el fin de hacer esto, y también se debe haber hecho un perfil exacto del programa para que se sepa cuánto tiempo se gasta en cada una de sus partes.

En el paralelismo de datos se descompone el programa basándose en los datos que están siendo creados o sobre los que se actúa, con diferentes espacios de datos (dominios) asignados a diferentes procesadores. En el paralelismo de datos, con frecuencia se deben tener los datos compartidos en los límites de los espacios de datos y la sincronización entre estos espacios de datos. El paralelismo de datos es el enfoque más común y es muy adecuado para máquinas de paso de mensajes en el que cada nodo tiene su propio espacio de datos privado, aunque esto puede conducir a veces a una gran cantidad de transferencia de datos.

Al planear cómo descomponer los datos globales en sub espacios adecuados para el procesamiento paralelo, es importante dividir los datos en bloques contiguos con el fin de minimizar el tiempo empleado en el movimiento de datos a través de las diferentes etapas de la memoria (fallos de página). Algunos compiladores y sistemas operativos ayudan en este sentido mediante la explotación de la localidad espacial, es decir, suponiendo que si se está utilizando un elemento de datos de un solo lugar en el espacio de datos, entonces es probable que se pueda requerir utilizar algunos datos más cercanos también para que ellos también se estén a disposición. Algunos compiladores y sistemas operativos también explotan la localidad temporal, es decir, suponiendo que si está utilizando un elemento de datos a la vez, entonces hay una mayor probabilidad de que se pueda utilizar de nuevo en un futuro próximo, por lo que también se mantiene a mano.

1.12 COMPUTADORAS MULTINODO-MULTINUCLEO-GPGPU

La arquitectura actual de superordenadores de gama superior utiliza un gran número de nodos, con cada nodo conteniendo un conjunto de chips que incluye múltiples núcleos (hasta 32 en la actualidad), así como una unidad de

procesamiento gráfico de propósito general (GP/GPU) unido al conjunto de chips. En un futuro próximo esperamos ver ordenadores portátiles capaces de llegar a teraflops (10^{12} operaciones de punto flotante por segundo), los ordenadores secundarios de escritorio capaces de llegar a petaflops (10^{15} operaciones de punto flotante por segundo), y supercomputadoras en la exaescala (exa: 10^{18}) en términos tanto de flops como de memoria.

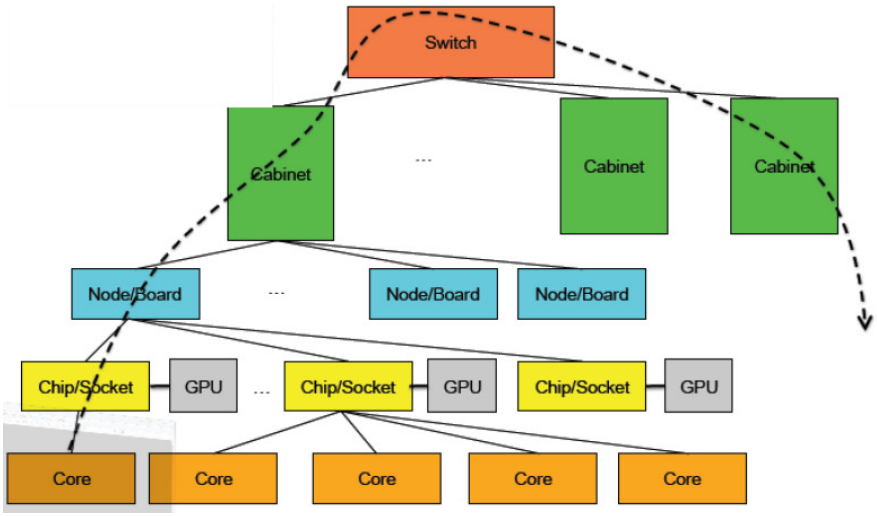

Figura 13 - Un diagrama esquemático de una computadora de exaescala en el que, además de que cada chip tiene múltiples núcleos, una unidad de procesamiento gráfico está unido a cada chip. Fuente: Jack Dongarra

Mirando de nuevo el esquema de la figura anterior, realmente hay un gran número de placas de chip y un gran número de gabinetes. Aquí mostramos un solo nodo y un gabinete y no el número total de núcleos. La línea discontinua en la figura representa las comunicaciones, y se ve permear todos los componentes del ordenador. De hecho, las comunicaciones se han convertido en una parte tan esencial de supercomputadoras modernas, que pueden contener cientos de miles de CPUs, que la "tarjeta" de la interfaz de red puede estar directamente en la placa del chip. Debido a que una computadora de este tipo contiene memoria compartida a nivel de nodo y la memoria distribuida a nivel de gabinete o a nivel superior, la programación para la transferencia de datos necesaria entre los múltiples elementos es el reto esencial.

El componente de GPU en la figura extiende este tipo de arquitectura de supercomputadora más allá de lo discutido previamente. La GPU es un dispositivo electrónico diseñado para acelerar la creación de imágenes visuales a medida que se muestran en un dispositivo de salida gráfica. La eficiencia de una GPU surge de su capacidad de crear diferentes partes de una imagen en paralelo, una

habilidad importante, ya que hay millones de píxeles para mostrar simultáneamente. De hecho, estas unidades pueden procesar cientos de millones de polígonos en un segundo.

Debido a que las GPU están diseñadas para ayudar al procesamiento de vídeo en los dispositivos básicos, tales como ordenadores personales, máquinas de juego, y los teléfonos móviles, se han vuelto de bajo costo, alto rendimiento, computadoras paralelas en su propio derecho. Sin embargo, debido a que las GPU están diseñadas para ayudar en el procesamiento de vídeo, su arquitectura y su programación son diferentes a la de los CPU de propósito general que habitualmente se utilizan para los algoritmos científicos, y se necesita algo de trabajo para poder utilizarlos para la computación científica.

La programación de GPU requiere herramientas especializadas específicas a la arquitectura de la GPU que se utiliza, y están más allá de la discusión de este trabajo. Por ejemplo, la programación CUDA se refiere a la programación para la arquitectura desarrollada por Nvidia, y es probablemente el enfoque más popular en la actualidad. Aunque puede programar en el nivel básico, ahora hay extensiones y “wrappers” desarrollados para lenguajes de programación como C, Fortran, Python, Java y Perl.

2. CLÚSTER

2.1 DEFINICIÓN DE CLUSTERING

En los términos más simples un clúster es un grupo de servidores independientes que funcionan como un solo sistema. También se puede definir como un grupo de sistemas que realizan una sola función (una caja negra). En una forma más específica clústering es tomar un grupo de servidores en donde y ejecutar de 1 a n aplicaciones.

De un clúster se espera que presente alto rendimiento, alta disponibilidad, balanceo de carga, alta fiabilidad, escalabilidad. La escalabilidad es la capacidad de un equipo de trabajar en volúmenes de trabajo cada vez mayores sin, por ello, dejar de prestar un nivel de rendimiento aceptable, la disponibilidad es la calidad de estar presente, listo para su uso, a mano, accesible; mientras que la fiabilidad es la probabilidad de un funcionamiento correcto. En definitiva, un clúster es un conjunto de computadoras interconectadas con dispositivos de alta velocidad que actúan en conjunto usando el poder cómputo de varios CPU en combinación para resolver ciertos problemas dados.

Para crear un clúster se necesitan al menos dos nodos. Una de las características principales de estas arquitecturas es que exista un medio de comunicación (red) donde los procesos puedan girar para computarse en diferentes estaciones paralelamente. Un solo nodo no cumple este requerimiento por su condición de aislamiento para poder compartir información. Las arquitecturas con varios procesadores en placa tampoco son consideradas clúster, bien sean maquinas SMP o mainframes, debido a que el bus de comunicación no suele ser de red, sino interno.

En resumen se puede decir que las características de un clúster son:

- Debe constar de 2 o más nodos.
- Los nodos de un clúster están conectados entre sí por al menos un canal de comunicación.
- Los clúster necesitan software de control especializado.

2.2 VENTAJAS DEL CLÚSTERING

Los clúster mejoran el rendimiento en general y el tiempo de respuesta del sistema en general lo cual se ve representado en las siguientes ventajas:

- Los clúster permite aumentar la escalabilidad, disponibilidad y fiabilidad de múltiples niveles de red.
- Cuando un equipo se encuentra dentro de un clúster y falla solamente basta con reemplazarlo sin que esto implique dejar de prestar el servicio.
- Se pueden resolver problemas de alta complejidad ya que se cuenta con la capacidad de rendimiento de más de un equipo.
- Facilita la creación de supercomputadores.
- Redundancia de equipos.

2.3 DESVENTAJAS DEL CLÚSTERING

Así como genera unos beneficios considerables en la producción del sistema en general, al implementarse un clúster en un ambiente computacional se deben tener en cuenta algunos aspectos, los cuales pueden generar adversidades en el desarrollo y expectativas que tiene el usuario con respecto a su sistema, los principales inconvenientes y dificultades que se pueden presentar al utilizar un clúster son:

- Ocupa un mayor espacio físico.
- El consumo de energía es considerable.
- Si hay problemas dentro de la LAN el clúster refleja estos inconvenientes.
- La escalabilidad de un clúster es mala al momento de abarcar aplicaciones transaccionales.

2.4 HERRAMIENTAS PARA EL DESARROLLO DE CLUSTERING

El comienzo del término y del uso de este tipo de tecnología es desconocido pero se puede considerar que comenzó a finales de los años 50 y principios de los años 60. El concepto de clúster viene muy unido a la computación multiprocesador, la diferencia entre las dos es que la segunda se basa en 1 sola maquina mientras la primera es el conjunto unificado de computadoras trabajando como una sola.

La historia de los primeros grupos de computadoras está más o menos directamente ligada a la historia de principios de las redes, como una de las principales motivaciones para el desarrollo de una red para enlazar los recursos de computación, utilizando el concepto de una red de conmutación de paquetes, el proyecto ARPANET logró crear lo que fue posiblemente la primera red de computadoras básico basadas en el clúster de computadoras por cuatro tipos de centros informáticos, este proyecto se fue desarrollando a tal grado que se convirtió en uno de los clúster más importantes de la historia (internet).

El primer producto comercial de tipo clúster fue ARCnet, desarrollada en 1977 por Datapoint pero no obtuvo un éxito comercial, pero se siguió con la investigación en

el área de clústering buscando promover la programación en paralelo y al mismo tiempo mantener la fiabilidad de los datos. Después de que nuevas tecnologías salieran al mercado se llegó a la invención de la Beowulf, una granja de computación diseñada según un producto básico de la red con el objetivo específico de ser un superordenador capaz de realizar firmemente y cálculos paralelos HPC.

En la actualidad se cuenta con múltiples vías para manejar el procesamiento en paralelo los cuales pueden ser clasificados como comerciales (necesitan licenciamiento) y libres (o de código abierto).

2.4.1 DE CÓDIGO ABIERTO

Las rutinas en la biblioteca de la agrupación C pueden ser incluidos o vinculados a otros programas en C. Para utilizar la biblioteca de la agrupación C, simplemente recopilar los archivos fuente relevante de la distribución de código fuente. Desde la versión 1.04, la biblioteca de la agrupación C cumple con el estándar ANSI C.

- AutoClass C, un sistema bayesiano de clasificación no supervisada desarrollado por la NASA, disponibles para Unix y Windows.
- CLUTO, proporciona un conjunto de algoritmos de clústering particional que tratan el problema de la agrupación como un proceso de optimización.
- Databionic Esom Tools, un conjunto de programas para la agrupación, la visualización y clasificación con Emergentes Self-Organizing Maps (Esom).
- MCLUST / EMCLUST, grupo basado en modelos y análisis discriminante, como agrupación jerárquica. En Fortran con el interfaz de S-PLUS.
- PermutMatrix, software gráfico para el agrupamiento y socialización, con varios tipos de análisis de conglomerados jerárquicos y varios métodos para encontrar una reorganización óptima de filas y columnas.
- StarProbe, servidor multiusuario basado en web disponible para las instituciones académicas.

2.4.2 SOFTWARE COMERCIAL

- BayesiaLab, incluye algoritmos bayesianos de clasificación para la segmentación de datos y utiliza redes bayesianas de forma automática para las variables del clúster.
- CViz clúster de visualización, para el análisis de grandes conjuntos de datos de alta dimensión, permite la visualización de movimiento completo del clúster.

- IBM Intelligent Miner de datos, incluye dos algoritmos de agrupamiento aXi.Kohonen Neusciences, ActiveX Control para Kohonen Clústering, incluye una interfaz de Delphi.
- perSimplex, software de clústering basado en lógica difusa.
- PolyAnalyst, ofrece la agrupación basada en la localización de anomalías (LA) en el algoritmo.
- StarProbe, multiplataforma, muy rápido en los datos grandes, el apoyo esquema de la estrella, herramientas especiales y características de los datos con una rica información dimensional categórica.
- Viscovery módulos de exploración de minería de datos, con análisis de agrupamiento visual, segmentación, y la asignación de medidas operativas para los segmentos definidos.

3. VIRTUALIZACIÓN

3.1 ¿QUÉ ES Y QUE OFRECE LA VIRTUALIZACIÓN?

La virtualización, en un sentido global es la emulación de una o más estaciones de trabajo y servidores en un equipo. En otras palabras, es la emulación del hardware, dentro de un software; este tipo de virtualización se refiere a la virtualización completa, permitiendo a los equipos emulados compartir recursos a través de muchos ambientes. Esto implica que una computadora puede tomar el papel de varios equipos.

La virtualización no solo está acotada a la simulación de equipos de hecho, existen varios tipos de virtualización. Uno de estos tipos está en la mayoría de máquinas de la actualidad, y es conocida como memoria virtual. Aunque la ubicación en memoria de los datos puede estar dispersa a través de la RAM y el disco duro de los ordenadores, el proceso de memoria virtual hace parecer que los datos están almacenados de forma ordenada y contigua.

Los dos conceptos más importantes para entender qué es la virtualización son los de anfitrión e invitado. Ambos conceptos se refieren a nuestro sistema operativo, y por lo tanto deberíamos hablar de sistema operativo anfitrión y sistema invitado:

- El anfitrión (host) es el ordenador en el cual instalamos nuestro programa de virtualización y que asignará o prestará determinados recursos de hardware a la máquina virtual que creemos.
- El invitado (guest) es el ordenador virtual que hemos creado, mediante nuestro programa de virtualización y al cual hemos asignado determinados recursos para funcionar.

A pesar de que la tecnología de virtualización ha existido de durante muchos años, sólo hasta ahora comienza a ser utilizada, una de las razones para que esto ocurra se debe al aumento en el procesamiento y los avances en tecnología de HW. La Virtualización puede beneficiar a una gran cantidad de usuarios, desde profesionales de Tic´s hasta grandes empresas y organizaciones gubernamentales. Los beneficios sólo ahora están comenzando a hacerse realidad.

3.2 VENTAJAS DE LA VIRTUALIZACIÓN

La virtualización permite gestionar de forma centralizada los sistemas virtualizados, así como sus recursos de almacenamiento y de red proporcionando:

- Rápida incorporación de nuevos recursos para los servidores virtualizados.

- Reducción de los costes de espacio y consumo necesario del Hardware.
- Reducción de los costes de TI gracias al aumento de la eficiencia y la flexibilidad en el uso de recursos.
- Administración global centralizada y simplificada.
- Permite gestionar el CPD como un pool de recursos o agrupación de toda la capacidad de procesamiento, memoria, red y almacenamiento disponible en la infraestructura
- Mejora en los procesos de clonación y copia de sistemas.
- Un fallo general de sistema de una máquina virtual no afecta al resto de máquinas virtuales
- Mejora de TCO y ROI
- Reduce los tiempos de interrupción.
- Migración servidor físico a otro.
- Balanceo dinámico de máquinas virtuales entre los servidores físicos que componen el pool de recursos, garantizando

3.3 DESVENTAJAS DE LA VIRTUALIZACIÓN

A medida que las máquinas virtuales se propagan por nuestros escritorios y servidores corporativos, se ponen de manifiesto las limitaciones de esta nueva técnica:

- Rendimiento inferior. Un sistema operativo virtualizado nunca alcanzará las mismas cotas de rendimiento que si estuviera directamente instalado en el HW.
- No todas las soluciones de virtualización obtienen el mismo rendimiento en las mismas operaciones.
- No es posible utilizar hardware que no esté gestionado o soportado por el hipervisor.
- El software de virtualización impone una serie de dispositivos virtuales como tarjetas de vídeo y red que a veces no se pueden satisfacer.
- Hardware virtual obsoleto. USB 1.0, Firewire 400, Ethernet 100 son algunos de los dispositivos a los que nos veremos sometidos.
- No dispondremos de aceleración de vídeo por hardware, por lo que aplicaciones con efectos 3D como compiz-fussion y juegos que utilizan las librerías OpenGL o DirectX no funcionarán en la máquina virtual.
- Como no hay que comprar hardware, el número de máquinas y servidores virtuales se dispara en todos los ámbitos. Los efectos colaterales se perciben después: aumenta el trabajo de administración, gestión de licencias, riesgos de seguridad, etc.
- Crear máquinas virtuales innecesarias tiene un coste en ocupación de recursos, principalmente en espacio en disco, RAM y capacidad de proceso.

- La avería del servidor anfitrión de virtualización afecta a todas las máquinas virtuales alojadas en él.
- La portabilidad entre plataformas está condicionada a la solución de virtualización adoptada.

3.4 HERRAMIENTAS PARA EL DESARROLLO DE VIRTUALIZACIÓN

En sus inicios, la virtualización era mejor conocida como 'time sharing', este término usado con frecuencia en la época día a pie la implementación la técnica de 'multiprogramming', que permite a un programador escribir el código fuente de un programa mientras otro programador compila otro programa.

Con el desarrollo de uno de los primeros sistemas operativos de tiempo compartido, CTSS ('Compatible Time-Sharing System') viene también la implementación de una de las primeras supercomputadoras mundiales, 'The Atlas Computer', y la más rápida de su tiempo. El Atlas aprovecha los conceptos de 'time sharing', 'multiprogramming' 'virtual memory' y control compartido de periféricos. Los 'extracodes' (nuevas instrucciones que pueden añadirse por software) son la única forma en la que un programa puede comunicarse con el 'Atlas Supervisor'. El 'Atlas Supervisor' es un programa que gestiona el tiempo de procesamiento; en terminología moderna, un 'job scheduler' avanzado o un sistema operativo simple.

El Centro Científico de Cambridge de IBM, liderado por Robert empieza el desarrollo del CP-40 y el CMS ('Cambridge Monitor System'). El CP-40 es el primer sistema operativo que implementa 'full virtualization', que permite emular simultáneamente hasta 14 'pseudo machines' (múltiples instancias del CMS), más tarde llamadas máquinas virtuales, ejecutándose en 'problem state'. Cuando una máquina virtual ejecuta una instrucción privilegiada (por ejemplo, una operación de E/S) o utiliza una dirección de memoria inválida, se produce una excepción que captura el 'Control Program', que se ejecuta en 'supervisor state', para simular el comportamiento adecuado.

Debido al desarrollo de ordenadores personales los cuales eran más pequeños en tamaño pero que en relación entre rendimiento/espacio ocupado rivalizaban con los grandes servidores de alto rendimiento en los cuales se desarrollaba la ‘full virtualization’ se perdió cierto interés en la investigación en el área de virtualización generando un estándar en la industria basado en microcomputadoras, aplicaciones cliente-servidor y computación distribuida.

Después de un largo tiempo y desarrollo en el campo de hardware surge de nuevo la pregunta “¿se están desaprovechando los recursos de las maquinas disponibles en la actualidad distribuyendo una actividad/función por unidad de trabajo?”, al realizar un estudio basados en ese predicamento se concluyó que utilizar cada

“caja” para una sola aplicación sería un desperdicio de recursos, espacio, energía y dinero; y tampoco es conveniente asignarle múltiples usos o instalar varias aplicaciones en un solo servidor convencional, por más de una razón (ej. estas aplicaciones podrían ser conflictivas entre sí, o podrían requerir diferentes configuraciones e inclusive diferentes sistemas operativos, o tener diferentes requerimientos de seguridad, entre otras variables que podrían causar problemas al ejecutar estas funciones simultáneamente). De allí vuelve a nacer la necesidad de dividir el hardware existente en una máquina de manera tal que funcione como múltiples servidores independientes pero compartiendo los recursos de un mismo servidor físico. Y es de aquí que nace lo que hoy todos conocemos como “Virtualización”.

Actualmente se cuentan con diversos medios que facilitan la creación, manejo, supervisión y desarrollo de sistemas virtualizados entre los cuales tenemos:

- Oracle VM VirtualBox es un software de virtualización para arquitecturas x86, creado originalmente por la empresa alemana innotek GmbH. Actualmente es desarrollado por Oracle Corporation como parte de su familia de productos de virtualización. Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como «sistemas invitados», dentro de otro sistema operativo «anfitrión», cada uno con su propio ambiente virtual.
- KVM apareció en febrero del 2007 con el kernel Linux 2.6.20, su principal programador es Avi Kivity y recibe apoyo económico de la start up Qumranet (en la actualidad parte de Red Hat) y la podemos identificar más con el paradigma de desarrollo “Consensus-based development” o desarrollo basado en consenso, de hecho si nos fijamos en la web de KVM hay una lista de cosas por hacer que cualquier colaborador puede abordar, además de contar con una comunidad más amplia que la concerniente a Xen.
- VMware (virtualización completa) es una solución comercial para la virtualización completa. Entre los sistemas operativos alojados y el hardware existe un hipervisor funcionando como capa de abstracción. Esta capa de abstracción permite que cualquier sistema operativo se ejecute sobre el hardware sin ningún conocimiento de cualquier otro sistema operativo alojado. VMware también virtualiza el hardware de entrada/salida disponible y ubica drivers para dispositivos de alto rendimiento en el hipervisor.

4. ANALÍTICA DE DATOS

4.1 ¿QUÉ ES LA ANALÍTICA DE DATOS?

La definición más simple de la analítica de datos seria decir que es la ciencia del análisis de los datos. Sin embargo, esta simple definición conduce a preguntas tales como ¿es la analítica de datos lo mismo que el análisis de datos? ¿Si la analítica de datos es una ciencia, entonces es lo mismo que la ciencia de datos?

Desafortunadamente, gran cantidad de textos y artículos relacionados con estos temas no contribuyen para aclarar estas interrogantes pues suelen hacer uso de cada una de estas frases de manera indiscriminada y en muchas oportunidades yendo y viniendo entre ellas sin guardar rigurosidad en su significado.

Tal vez, el significado más simple de esclarecer sea el de “ciencia de datos” (Data Science, de acuerdo con la literatura en inglés), pues sobre este término existe un mayor consenso en que se trata de una disciplina que incorpora varios grados de Ingeniería de Datos, Método Científico, Matemáticas, Estadística, Computación Avanzada, Visualización, mentalidad de Hacker y experiencia en el campo. A un practicante de la Ciencia de Datos se le llama Científico de Datos.

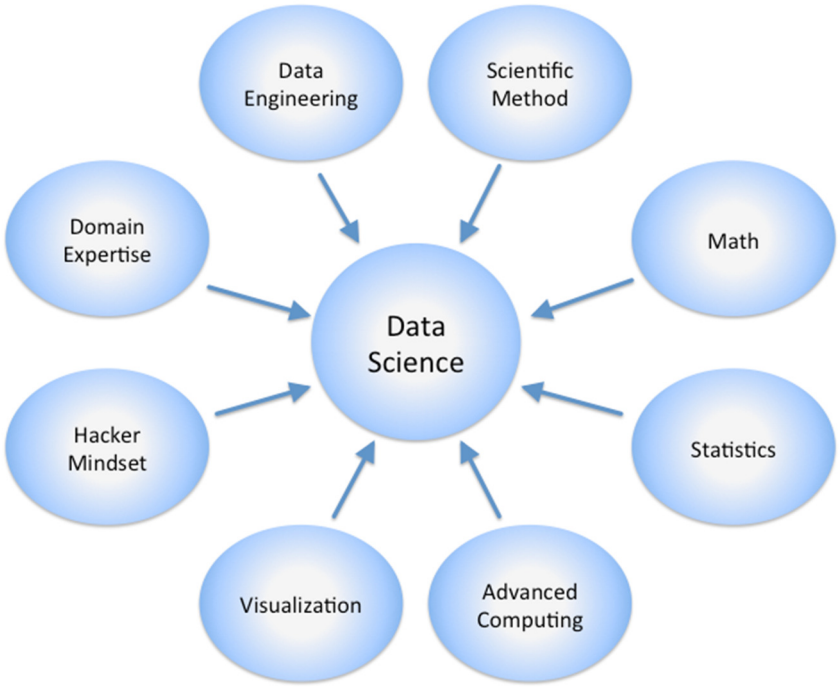

Figura 14 - Disciplinas que forman la Ciencia de Datos. Fuente: http://en.wikibooks.org/wiki/Data_Science:_An_Introduction/A_Mash-up_of_Disciplines

El término "Ciencia de Datos" fue acuñado en los comienzos del siglo 21 y es atribuido a William S. Cleveland quien en 2001, escribió un artículo para el International Statistical Review titulado "Data Science: An Action Plan for Expanding the Technical Areas of the Field of Statistics."

Una discusión un tanto más extensa se puede dar al tratar de precisar las definiciones y diferencias entre “Análisis de Datos” y “Analítica de Datos”. Pues en estos dos términos se encuentra la mayor parte de las ambigüedades de la literatura disponible en estas materias.

El “Análisis de Datos” es el proceso de la aplicación sistemática de técnicas estadísticas y/o lógicas para describir e ilustrar, condensar y resumir y evaluar los datos. El análisis de datos involucra un proceso de inspección, limpieza, transformación y modelado de datos con el objetivo de descubrir información útil, lo que sugiere conclusiones, y el apoyo a la toma de decisiones. El análisis de datos es el acto de transformación de los datos con el fin de extraer información útil y facilitar conclusiones. Dependiendo del tipo de datos y de la cuestión, esto podría incluir la aplicación de métodos estadísticos, de ajuste de curva, seleccionar o descartar ciertos subconjuntos en base a criterios específicos, u otras técnicas.

La “Analítica de Datos” busca brindar observaciones operacionales en cuestiones que, o bien sabemos que sabemos o sabemos que no sabemos. La analítica va en pos del descubrimiento y la comunicación de patrones significativos en los datos. Es especialmente valiosa en áreas ricas en información ya registrada, la analítica se basa en la aplicación simultánea de estadística, programación informática y la investigación de operaciones para cuantificar el rendimiento. La analítica con frecuencia favorece la visualización de datos para comunicar la visión. La analítica de datos realiza investigaciones para descubrir métodos computacionales escalables para la búsqueda de modelos de utilidad a partir de cantidades masivas de datos. Estos modelos pueden ser utilizados en una variedad de tareas, incluyendo la agrupación/descubrimiento, la regresión/predicción y clasificación/puntuación.

La analítica de datos construye modelos de predicción y descubren patrones de datos. Mientras que el análisis de datos convertir los datos en inteligencia y construye mejores relaciones con los clientes.

En pocas palabras podríamos decir que si típicamente la Inteligencia de Negocios (BI) puede decir lo que ha sucedido, el Análisis de Datos dirá por qué ha sucedido y la Analítica de Datos dirá lo que va a suceder.

4.2 ARQUITECTURA DE UNA SOLUCIÓN DE ANALÍTICA DE DATOS

La figura representa una arquitectura técnica típica para una solución analítica construida sobre un almacén de datos.

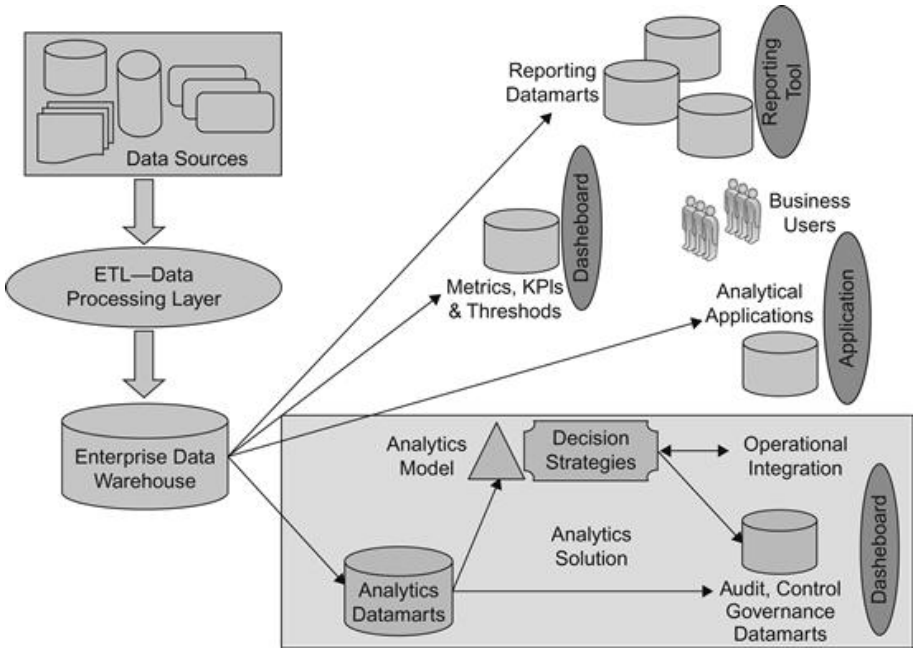

Figura 15 - Arquitectura técnica de una solución de Analítica de Datos.

Esta arquitectura representa el modelo generalizado de una solución de analítica de datos en donde priman las despensas de datos o "Datamarts". El término despensa de datos se convirtió en parte convencional del almacenamiento de datos y la industria de informes después de Ralph Kimball presentó al mundo el modelado tridimensional. Se define como un tema en un área específica de recolección de dimensiones y hechos, como una despensa de datos de ventas orden, una despensa de datos de facturación, una despensa de beneficios para empleados, una despensa de datos de préstamos, etc. Lo que solía implicar una tabla de hechos y las diferentes dimensiones relevantes.

Con la madurez del diseño en la industria, las despensas de datos podían manejar varias tablas de hechos como tablas agregadas, además de tablas de hechos en el detalle más bajo. Un ejemplo de varias tablas de hechos dentro de una despensa de datos es una despensa de datos de entradas servicio de asistencia

donde una tabla de hechos está en el detalle del boleto (un registro representa un boleto) y otro puede estar en el detalle del estado de los boletos.

La definición de una despensa de datos evolucionó con los paquetes de software más especializados y centrados en datos, como el lavado de dinero, precios, o la gestión de campañas. Generalmente hay una necesidad de crear una base de datos independiente totalmente cargada con los datos relevantes necesarios para que el software funcione. Se necesita un proceso de ETL para extraer los datos ya sea desde el almacén de datos o de los sistemas de origen y cargar en esta base de datos especializada estructurada para apoyar el paquete de software. Esto también se conoce como una despensa de datos, ya que es una colección especializada de datos que pueden abarcar varias áreas temáticas, pero no está necesariamente construida como un modelo dimensional y se utiliza para algo más que informar. Esta definición de una despensa de datos tiene datos con ambas funciones de lectura-escritura y diferentes detalles de datos en la misma despensa de datos.

Una despensa de datos analítica es básicamente una colección de todos los datos necesarios para que una solución analítica funcione. Esto incluiría el grado más bajo de datos detallados, datos de resumen y de instantáneas, variables de rendimiento, las características, los resultados del modelo y los datos necesarios para la auditoría y el control – cualquier tipo de dato relevante para la solución tiene que ser diseñado y almacenado en la despensa de datos analítica.

La despensa de datos analítica tiene cuatro áreas distintas dentro de su construcción lógica. Una implementación de solución específica puede tener una manifestación física diferente de estas cuatro construcciones lógicas. Así que un arquitecto de la solución puede decidir la creación de dos bases de datos separadas para alojar dos construcciones lógicas cada una. Las construcciones son:

1. Datos de base analítica
2. Las variables de desempeño
3. Modelo y características
4. Modelo de auditoría y control de ejecución

4.3 ELEMENTOS PARA LA IMPLEMENTACIÓN DE LA ANALÍTICA DE DATOS

4.3.1 HADOOP

La creación de Hadoop se remonta a 2005, como una iniciativa en Yahoo por Doug Cutting impulsado o inspirado en la tecnología MapReduce de Google. Para el año 2009, Hadoop dominaba búsquedas en la web y el funcionamiento interno

de grandes páginas Web, para organizar, indizar y buscar tesoros de datos y anuncios de servicio en empresas como Yahoo, Google y Facebook.

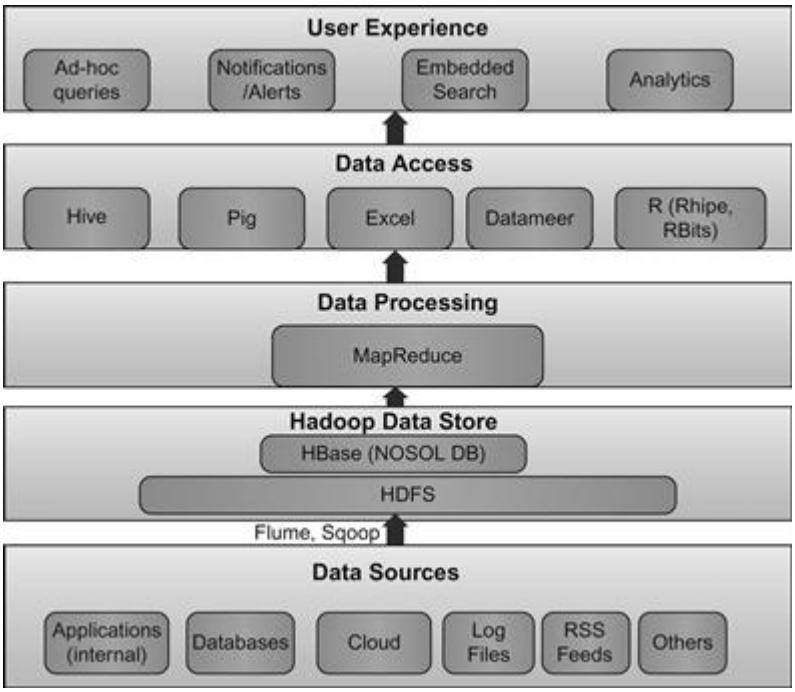

Figura 16 - Conjunto tecnológico de Hadoop.

Hadoop es un sistema de archivos capaz de almacenar y procesar una cantidad sin precedentes de los datos presentados en cualquier formato de archivo. Utiliza una tecnología llamada MapReduce para buscar, acceder y manipular los datos mientras se ejecuta en máquinas computadoras interconectadas baratas (incluso PCs viejos).

La figura muestra donde un conjunto de tecnología Hadoop encaja en la solución global de procesamiento de datos y qué tipo de herramientas están disponibles en cada capa de la pila para funciones específicas.

4.3.1.1 FUENTES DE DATOS

Las fuentes de datos se refieren a los datos, o más bien los grandes datos, que debe introducirse en el sistema de archivos Hadoop (HDFS). Existen varias herramientas y paquetes de software que permite mover los datos de almacenamiento convencionales como los sistemas de archivos de UNIX o

sistemas de bases de datos relacionales o incluso de registros o varias otras formas de almacenamiento en Hadoop. Hadoop toma los datos entrantes y los carga en su propio sistema de archivos, mientras que la estructuración de los datos a través de los distintos grupos (grupos de equipos o nodos para ejecutar Hadoop) la distribución de los datos de entrada a través de los nodos del clúster.

4.3.1.2 ALMACEN DE DATOS DE HADOOP

El almacén de datos Hadoop tiene HDFS como sistema de archivos y un catálogo que rastrea dónde se ha almacenado los datos. Los archivos de las fuentes de datos (por ejemplo, registros, archivos PDF, imágenes, etc.) no retienen esa estructura nativa, sino que se convierten en el formato HDFS. A diferencia de Explorer, que muestra todos los archivos en el sistema de archivos de Windows, como documentos, hojas de cálculo, etc. de archivos de Windows, no se puede abrir el HDFS y mirar los archivos originales fácilmente.

4.3.1.3 PROCESAMIENTO DE DATOS

Una vez que los datos están dentro de la HDFS, la única manera de acceder a ella es a través de la interfaz de comandos MapReduce. La interfaz de comandos MapReduce permite que toda la lógica de procesamiento sea escrita en MapReduce. Sin embargo, la programación MapReduce no es trivial, ya que requiere romper la lógica de procesamiento por una en paralelo en lugar de la generación de código de programación secuencial. Romper un problema de negocio en forma de funciones Map y Reduce es todo un reto de programación.

4.3.1.4 ACCESO A DATOS

Para hacer frente a este reto de procesamiento de datos de MapReduce, que no es lo suficientemente rico como un entorno de programación DotNet, toda una capa de acceso de datos se ha construido con el tiempo. Esta capa de acceso a datos se compone de una amplia variedad de código abierto y herramientas propietarias y bibliotecas desarrolladas para diferentes necesidades de programación. Si MapReduce era como la programación del lenguaje ensamblador, la capa de acceso a datos es más como C / C ++, SQL, y el tipo de programación Java, que es más negocios.

4.3.1.5 APLICACIONES DE USUARIO

La capa de aplicación de usuario o la experiencia del usuario es como la capa de aplicación en la lógica compleja de negocio que se combina para ofrecer valor. Este es el espacio de mayor crecimiento dentro de la pila de tecnología Hadoop donde las bibliotecas, herramientas, paquetes de software y suites están

empezando a estar disponibles en una amplia variedad de aplicaciones de negocio específicas.

4.3.2 USO DE HADOOP EN LA ANALÍTICA DE DATOS

El uso de Hadoop para resolver problemas analíticos tiene dos variaciones:
1. Hadoop que actúa como una capa de ETL para procesar grandes volúmenes de datos y carga en una despensa de datos analítica basada en una RDBMS tradicional.
2. Hadoop actuando como motor de minería de los datos procesando los datos para construir un modelo.

4.3.2.1 HADOOP COMO UN MOTOR ETL

La figura muestra cómo Hadoop encajaría en una solución analítica. La idea de este enfoque es agregar o construir variables de rendimiento partir de los datos utilizando Hadoop, mientras que los datos tradicionales que necesita ser entremezclados toma una ruta ETL convencional. Una vez que el Big Data se reduce a tamaños más manejables (mediante la eliminación de uno o más de V a partir de sus características), se puede tratar como datos estructurados convencionales que pueden ser almacenados y procesados en un sistema de base de datos relacional. En este escenario, Hadoop está actuando como un componente ETL ágil y eficiente. Los principales proveedores de ETL han añadido soporte Hadoop en su suite de herramientas para hacerlo desde una suite de integración de datos.

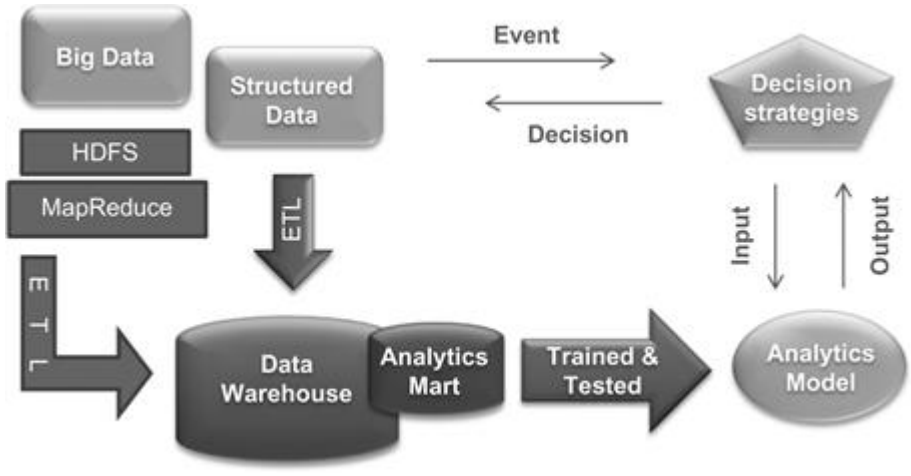

Figura 17 - Hadoop como motor ETL

Se requiere esta forma avanzada de ETL para convertir datos desestructurados como tweets, vídeos, llamadas telefónicas, registros de los sensores de la máquina, etc. en datos estructurados. Esto es importante porque los datos no estructurados pueden no ser muy útiles para ser utilizados en los modelos analíticas a menos que se integre el conocimiento de esos datos con otros datos de negocio y operaciones estructuradas con el fin de construir mejores variables de rendimiento y modelos analíticos.

Por lo tanto, el tratamiento de los datos no estructurados se debe considerar una capa separada de ETL especializada (en esteroides) que puede leer y descifrar las estructuras digitales detrás de los datos no estructurados y obtener valor de la misma en forma estructurada y alimentar a una capa más convencional de ETL utilizando Hadoop, que es entonces capaz de integrar este con otras fuentes Big Data.

Si después de aplicar Hadoop para datos no estructurados, las variables y los datos estructurados que extraemos son manejables en tamaño, a continuación, se pueden ejecutar a través de una capa de ETL convencional así siguiendo métodos de almacenamiento de datos tradicionales.

4.3.2.2 HADOOP COMO MOTOR ANALÍTICO

La otra opción para Hadoop dentro de una solución analítica es donde todo el algoritmo de minería de datos se lleva a cabo en el entorno de programación Hadoop y actúa como el motor de la minería de datos, mostrado en la siguiente figura. Esto se utiliza cuando no hay opción de reducción, agregación, o muestreo de los datos para eliminar los V de las características Big Data. Esto se hace muy complejo a medida que las variables de rendimiento, el corazón de la innovación en el modelado analítica, no pueden ser fácilmente añadidas a la serie de datos sin crear una capa de almacenamiento adicional. Sin embargo, diversos problemas realmente requieren correr todo el Big Data set en busca de tendencias específicas o realizar búsqueda difusa o correlaciones entre las variables a través de Hadoop y sus programas de acceso a datos.

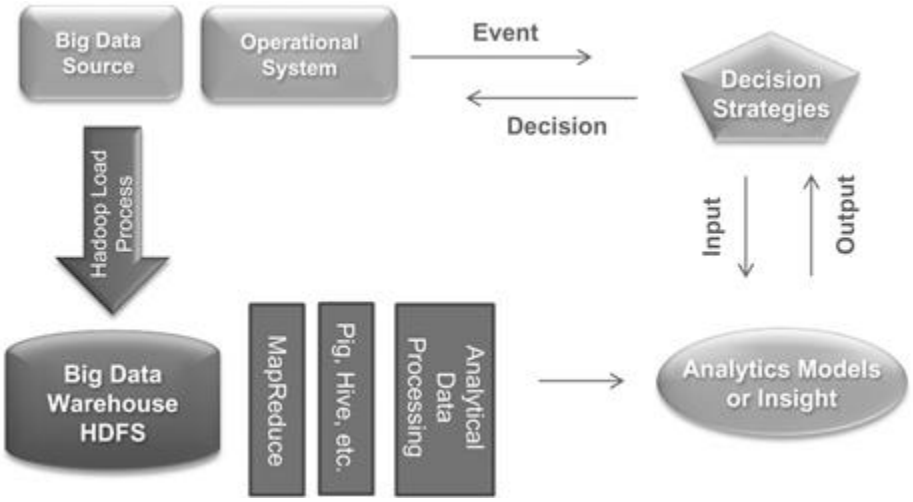

Figura 18 - Hadoop como motor analítico

5. INTEGRACIÓN DE LA COMPUTACIÓN DE ALTO DESEMPEÑO Y LA ANALÍTICA DE DATOS

La computación de alto desempeño (HPC) ya ha contribuido enormemente a la innovación científica, la competitividad industrial y económica, la seguridad nacional y regional, y la calidad de la vida humana. Este papel crucial se ha enfatizado en los últimos años por los presidentes de los Estados Unidos y Rusia, así como por funcionarios de alto nivel en Europa y Asia.

Hasta la fecha, la mayoría de los trabajos HPC intensivos en datos para los sectores gubernamental, académico e industrial han implicado el modelado y simulación de sistemas físicos y cuasi-físicos complejos. Estos sistemas van desde diseños de productos para automóviles, aviones, palos de golf y productos farmacéuticos, a las partículas subatómicas, los patrones meteorológicos y climáticos globales y el cosmos mismo. Sin embargo, desde el comienzo de la era supercomputadora en la década de 1960 - e incluso antes - un subconjunto importante de los trabajos HPC ha involucrado la analítica, intentando descubrir información útil y patrones en los datos en sí. Por ejemplo, la criptografía, una de las aplicaciones originales de cómputo científico-técnico, cae principalmente en esta categoría.

La industria de servicios financieros fue el primer mercado comercial en adoptar las supercomputadoras para analítica de datos avanzados. Los grandes bancos de inversión comenzaron a contratar los físicos de partículas del Laboratorio Nacional de Los Álamos (LANL) y el Instituto de Santa Fe en la década de 1980 con el fin de emplear los sistemas HPC para tareas analíticas de enormes proporciones, como la optimización de carteras de valores respaldados por hipotecas, precios a los instrumentos financieros exóticos, y la gestión del riesgo global de toda la empresa. Esta práctica ha continuado: en 2013, Goldman Sachs atrajo un físico de partículas lejos del Gran Colisionador de Hadrones del CERN.

5.1 ¿QUÉ ESTÁ DIRIGIENDO LA DEMANDA?

El análisis de alto rendimiento de datos – lo que se refiere como "HPDA" – es a la vez una evolución y una historia revolucionaria. La explosión de datos está alimentando el crecimiento del análisis de datos de alto rendimiento y se debe a una combinación de larga data y factores nuevos:

- La capacidad de los sistemas HPC cada vez más potentes para ejecutar problemas intensivo de datos de modelado y simulación a gran escala, en una resolución más alta, y con más elementos (por ejemplo, la inclusión del ciclo del carbono en los modelos de conjunto climático)

- La proliferación de grandes instrumentos científicos más complejos y redes de sensores, de las redes de energía "inteligentes" al Gran Colisionador de Hadrones
- La creciente transformación de ciertas disciplinas en ciencias basadas en datos - la biología es un ejemplo notable - pero esta transformación se extiende incluso a disciplinas humanísticas como la arqueología y la lingüística
- El crecimiento de la modelización estocástica (servicios financieros), el modelado paramétrico (fabricación) y otros métodos de resolución de problemas iterativos, cuyos resultados acumulativos producen grandes volúmenes de datos
- La disponibilidad de nuevos métodos de análisis avanzado y herramientas: MapReduce / Hadoop, análisis gráfico, análisis semántico, algoritmos de descubrimiento de conocimiento y otros
- La creciente necesidad de realizar análisis avanzados en tiempo casi real - una necesidad que está provocando una nueva ola de empresas comerciales en adoptar HPC, por primera vez

5.2 DISCIPLINAS HPC EXISTENTES SE AMPLIAN A LA ANALÍTICA

Algunos miembros de la comunidad de investigación del clima han comenzado a incrementar métodos existentes con los algoritmos de descubrimiento de conocimientos basado en analítica para promover nuevas ideas, y tal vez ningún campo tiene un potencial tan fuerte para beneficiarse de la analítica basada en HPC que la biociencia. Las aplicaciones de uso intensivo de datos que ya están en marcha en el campo de las ciencias biológicas variadas que van desde la investigación avanzada - especialmente en la genómica, la proteómica, la epidemiología y la biología de sistemas - a las iniciativas comerciales para desarrollar nuevos medicamentos y tratamientos médicos, pesticidas agrícolas y otros productos biológicos.

Uno de los empujes más importantes en lo social y económicamente para el HPDA del mundo es casi seguro que será la transición dentro de varios años de la medicina de hoy basada en procedimientos a una personalizada, cuidado de la salud basada en resultados. La identificación de tratamientos altamente efectivos casi en tiempo, real mediante la comparación de la composición genética de un individuo, la historia clínica y la sintomatología en contra de decenas de millones de registros de pacientes archivados, plantea enormes desafíos que HPDA puede tardar una década más para dominar. Cuando esta capacidad madure, se cree que es probable que sirva como herramienta de apoyo a las decisiones de utilidad sin precedentes para la comunidad mundial de la salud.

Los nuevos métodos y herramientas de análisis son susceptibles de beneficiarse de todos los segmentos verticales HPC existentes, al menos hasta cierto punto. Estos segmentos también incluyen ingeniería asistida por ordenador, la ingeniería química, la creación de contenido digital y la distribución, automatización electrónica de datos, servicios financieros, ciencias de la tierra y la geoingeniería (petróleo y gas), la defensa, los laboratorios del gobierno y la academia. Pero la historia no termina allí.

Las aplicaciones de alto potencial de análisis horizontal también están comenzando a tener un impacto importante en el mundo de la computación de alto desempeño.

La detección de fraude, seguridad cibernética y las amenazas de la información privilegiada son, cada vez más, desafíos que deben cumplir los usuarios de HPC establecidos en el gobierno, la academia y la industria, y que están causando una nueva ola de organizaciones comerciales que buscan integrar HPC, por primera vez.

El desafío que presentan estos problemas es descubrir patrones y relaciones ocultas - cosas que usted no sabía que estaban allí - y luego rastrear los patrones de forma dinámica a medida que se forman y evolucionan.

A medida que el escenario de proveedores de HPDA es cada vez más creciente y heterogéneo, la parte analítica del mercado HPDA en formación es donde los usuarios HPC tradicionales y adoptantes por primera vez convergen más rápidamente.

5.3 INTEGRACIÓN PARA ENFRENTAR LOS DESAFÍOS DEL USO INTENSIVO DE DATOS

El creciente mercado de análisis de datos de alto rendimiento - con ayuda de HPC para los desafíos de datos intensivos - ya está ampliando las contribuciones de HPC a la ciencia, el comercio y la sociedad, y HPDA promete desempeñar un papel importante en ayudar a abordar los principales retos y oportunidades del siglo 21. El crecimiento espectacular de los datos científicos, comercial y social se traduce en una mayor base de organizaciones que está pidiendo un análisis mucho más complejo y simulación. Hay un circuito de retroalimentación entre más y más grandes conjuntos de datos y modelos de simulación más complejos.

La tecnología tiene un legado dentro de la comunidad científica y de investigación, pero Big Data sirvió como catalizador para la adopción HPC en las empresas tradicionales. Mientras que Big Data a largo plazo podría haberlas sumergido en la desilusión, HPC mantiene la promesa de guiar Big Data hacia el crecimiento empresarial y la productividad.

A medida que las capacidades técnicas continúan expandiéndose en las formas en que se puede recoger y almacenar datos, el problema de la forma en que se acceden y utilizan los datos sólo está creciendo. Aunque se pueden encontrar varias alternativas en torno a este problema, existe la tendencia fuerte que la capacidad de orquestar fácilmente y acceder a grandes volúmenes de datos en la nube ofrece soluciones claras.

La computación en la nube permite la posibilidad de acceder a cantidades muy grandes de datos - algunas de las cuales se están recopilando casi en tiempo real - y permite aprovechar el poder de computación prácticamente ilimitado. Aquí hay algunos temas que lo ponen de relieve en lo siguiente:

- Preguntar la pregunta correcta: Hay un cambio en marcha, donde investigadores, ingenieros y analistas pueden cambiar el modo de pensar acerca de los problemas. Anteriormente, se había estado limitado por los recursos de computación que se tenían - los grupos que se tienen en las instalaciones. Hoy, se puede cambiar el modo mismo en que se hacen las preguntas. Se hacen las preguntas correctas - y con el uso de la nube se crea el tamaño del sistema necesario para responder a esas preguntas.
- Colaboración sin precedentes: Hoy en día, los avances científicos provienen de equipos de personas, en lugar del científico solitario. A menudo, los equipos están, de hecho, colaborando en diferentes continentes. Es increíble que la tecnología ha permitido este tipo de colaboración en todo el mundo. Pero es aún más emocionante considerar cómo HPC en la nube está tomando esta capacidad a un nivel completamente nuevo.

5.4 UN ENFOQUE INTEGRAL

¿Cómo pueden las organizaciones adoptar el rápido incremento de la colisión de nubes públicas y privadas, entornos HPC y big data? La actual solución para muchas organizaciones es ejecutar estos activos de tecnología en silos, en ambientes especializados. Sin embargo, este enfoque se queda corto, por lo general gravando un área del centro de datos, mientras que otras permanecen subutilizadas, funcionando como poco más que espacio de almacenamiento costoso. Como los conjuntos de datos emergen más grandes y más complejos, se hace cada vez más difícil de procesar grandes volúmenes de datos utilizando herramientas de gestión de base de datos a disposición o aplicaciones tradicionales de procesamiento de datos. Para maximizar sus inversiones significativas en estos recursos de centros de datos, las empresas deben hacer frente a grandes volúmenes de datos con un enfoque integral que maximiza los recursos de centros de datos y agiliza el proceso de análisis de simulación y datos.

Este enfoque utiliza todos los recursos disponibles en el centro de datos, incluidos los entornos HPC, así como otros recursos de centros de datos como la nube privada y pública, los grandes datos, las máquinas virtuales y físicas. Bajo este paraguas, todos los recursos del centro de datos se han optimizado, lo que elimina el estancamiento y lo convierte en un flujo de trabajo organizado que aumenta en gran medida el rendimiento y la productividad.

5.5 ESTADO DEL ARTE ACTUAL

En la industria, se está viendo actualmente en el estado del arte herramientas como OpenStack y Hadoop siendo utilizadas para el procesamiento de Big Data. Desde 2013, algunos proveedores se han unido a la comunidad OpenStack y han anunciado su integración. Además, han integrado la Distribución de Intel HPC para el software Apache Hadoop, un hito en el gran ecosistema de datos que permite a las cargas de trabajo de Hadoop ejecutarse en sistemas HPC. Entonces, las organizaciones tienen la capacidad de expandirse más allá de un enfoque de silos y aprovechar tanto su HPC como las inversiones de Big Data juntas.

El estado del arte para la supercomputación intensiva en datos tiene que traer paralelización productiva. Ya sea que se trate de hardware y software potente de procesamiento paralelo, sistemas de archivos paralelos tremendamente rápidos, o de analítica de gráficos paralelos, el paralelismo es un componente clave del estado del arte. Los sistemas abiertos, donde los datos no están ligados permanentemente a una empresa o tecnología en particular también son pieza clave.

5.6 PERSPECTIVAS DE FUTURO

Este creciente vínculo entre analítica de datos, gestión de datos y la computación es una tendencia a largo plazo. Este estrecho acoplamiento dará lugar a descubrimientos, tanto científicos como sociales, lo que dejará su sello en la conciencia de tanto del usuario de computación de alto desempeño y de la sociedad en su conjunto. HPC siempre ha tenido dificultades para mostrar al público en general sobre cómo sus productos impactan sus vidas, y esta industria tiene que mejorar en esto. Estos impactos serán más generalizados y, si los usamos sabiamente, harán todas nuestras vidas más productivas y más significativas.

En 2014, se espera que las capacidades clave HPC mejoren el nivel de inteligencia y velocidad que la analítica de datos puede proporcionar a la empresa, incluyendo:

- Representación gráfica y cartografía: mapeo de datos y gráficos potenciadas por HPC llevarán a una mayor precisión en la predicción del negocio
- Visualización de patrones: emergerán herramientas potenciadas por HPC que puede proporcionar una visión intuitiva de conjuntos de datos complejos, lo que permite la rápida identificación de las relaciones para análisis simples
- Escalamiento de bases de datos en memoria: HPC permitirá que sistemas empresariales en memoria manejen cargas de trabajo de datos más grandes - que permite acercarse a los conjuntos de datos completos (sobre conjuntos parciales) para beneficiarse de análisis en tiempo real en movimiento
- Meta-datos: La importancia de los metadatos saltará dramáticamente - veremos las empresas darse cuenta que el aprovechamiento de la analítica de meta-datos para la virtualización y el mapeo relacional puede producir una mayor precisión, nuevas ideas de negocios e incluso revelar las amenazas de seguridad

El mayor problema hoy en día es la falta de reconocimiento del verdadero alcance de los datos relevantes disponibles. La analítica de Big Data es más eficaz cuando se combina no sólo datos estructurados y no estructurados internos, sino cuando estos se emparejan con datos externos disponibles de todas las fuentes de información tales como sociales, de mercado, Web y datos de los sensores. Al vincular HPC con grandes volúmenes de datos, las empresas maximizarán la inteligencia de todas estas fuentes, procesando datos a volúmenes muy elevados, con la velocidad y precisión que las empresas necesitan para seguir prosperando.

6. ALTERNATIVAS DE SOFTWARE PARA LA CONSTRUCCIÓN DE UN MODELO DE COMPUTACIÓN DE ALTO DESEMPEÑO

6.1 CONSIDERACIONES INICIALES

Esta investigación centra sus esfuerzos para la implementación del prototipo de HPC en los sistemas clúster con sistema operativo Linux, pues en este sentido ha presentado y documentado en su capítulo primero, numeral 1.3 sobre COMPUTADORAS DE ALTO DESEMPEÑO, la predominancia que tienen los sistemas de clúster en la implementación de supercomputadores y sistemas de computación de alto desempeño. Y de igual manera, en la misma sección del documento, se dejó establecido con la evidencia presentada, que desde hace más de 10 años el sistema operativo Linux es también el sistema operativo dominante en la arena de los supercomputadores y que, actualmente, sobrepasa el 90% de la distribución de este mercado.

Por otra parte, cuando se realizó el análisis sobre RESTRICCIONES EN LA COSTRUCCIÓN DE SISTEMAS HPC, literal 1.4 del capítulo primero, se identificaron los beneficios que los sistemas con estructura de chasis/cuchilla tienen para reducir los problemas de espacio físico, consumo de energía y generación de calor causados por la condensación y agrupamiento de los recursos de cómputo.

Los puntos acá expuestos permiten generalizar que son los sistemas construidos con equipos de cómputo de "formato" chasis/cuchilla aquellos que aventajan cualquier otra arquitectura de servidores para los clústeres Linux en la implementación de HPC. Sin embargo, para la implementación del prototipo de HPC que se desea construir como resultado de esta investigación se hará uso de los servicios de virtualización, presentados en el capítulo tercero, en lugar del uso de servidores de cuchilla que permitan representar la condensación y agrupamiento de recursos de cómputo y establecer su validez operativa en modelos para la academia y otras organizaciones.

6.2 COMPONENTES DE SOFTWARE MÁS COMUNMENTE UTILIZADOS EN HPC

Aunque los sistemas HPC comparten muchos componentes de hardware con los servidores en las empresas y centros de datos, la pila de software HPC es dramáticamente diferente de la pila de software empresarial o de la nube y es único para HPC. Hablando generalmente, una pila de software HPC tiene múltiples niveles: software de sistema, ambientes de desarrollo, software de gestión del sistema y software de sistemas de gestión y visualización de datos científicos. Hacia el nivel del hardware, el software del sistema típicamente incluye sistemas operativos, sistemas de ejecución y software de bajo nivel de E/S, como

sistemas de archivos. Contiguo a este, el ambiente de desarrollo es una amplia área que facilita el diseño y desarrollo de aplicaciones. En el marco de referencia que se trabaja, este incluye modelos de programación, compiladores, librerías y marcos de referencia científicos y herramientas de exactitud y desempeño. Entonces, el software de gestión del sistema coordina, programa y monitorea el sistema y las aplicaciones ejecutándose en ese sistema. Finalmente, el software de gestión y visualización de datos científicos le ofrece a los usuarios herramientas de dominio específico para generar, gestionar y explorar los datos para su ciencia. Estos datos pueden incluir datos empíricos medidos desde sensores en el mundo real que son usados para calibrar y validar modelos de simulación o salidas de simulaciones per se. Como lo muestra la tabla 1, el sistema descrito en esta investigación tiene una gran cantidad de software común, aún cuando algunos de los sistemas son muy diversos en términos de hardware. Más aun, una considerable cantidad de este software es código abierto (open-source) y está respaldado por un amplio abanico de patrocinadores.

Tabla 1 - HPC software summary.

Categoría	Elemento
Sistemas Operativos	Linux (Múltiples versiones)
Lenguajes	C, C++, FORTRAN
Compiladores	CAPS, Cray, GNU, IBM, Intel, Pathscale, PGI
Lenguajes de Scripting	Java, Perl Python, Ruby, Tcl/Tk
Modelos de programación de memoria distribuida	Charm++, Co-Array Fortran, Global Arrays, Hadoop/MapReduce, MPC, MPI (OpenMPI, MVAIPICH, Intel MPI, Cray MPI, MPICH), MPT, SHMEM, Unified Parallel C, XMP
Modelos de programación de memoria compartida	OpenMP, Pthreads, TBB
Modelos de programación hetereogénea	CAPS HMPP, CUDA, OpenACC, OpenCL, PGI Accelerate,
Herramienta de Desempeño	BPMON, Cray CPMAT, Extrae/Paraver, HPCToolkit, HWLOC, IHPCT, IPM, Intel Trace Analyzer, MPIP, MPInside, NVIDIA Visual Profiler, Ocelot, oprofile, PAPI, PDToolkit, PerfSuite, SCALASCA, TAU, VampirTrace/Vampir, Vtune
Herramientas de Exactitud	DDT, GNU GDB, STAT, Threadchecker, Threadspotter, Totalview, Valgrind
Librerías científicas	ACML, ARPACK, BLAS, Boost, CASE, CRAFFT, cuBLAS,

	cuFFT, cuLA, cuRAND, cuSP, cuSPARSE, ESSL, FFTW, GNU GSL, Gridgen, hypre, LAPACK, MAGMA, MASS, MKL, MUMPS, PARPACK, ParMetis, SPRNG, SUNDIALS, ScaLAPACK, Scotch, SuperLU, Thrust
Marcos de referencia científica	Arcane, GraphLab, JASMIN, PETSc, Trilinos
Sistemas de archivos y almacenamiento paralelo	GPFS, GridFtp, HPSS, Lustre, Panasas, StorNext
Programadores de tareas y manejadores de recursos	ALPS, GangliaMole, LoadLeveler, Moab, PBSPro, SLURM, Sun Grid Engine, Torque
Gestión del sistema	ACE, ClusterShell, Ganglia, Inca, NFS-Ganesh, NHC, Netlogger, NodeKARE, Robinhood, Rocks, SEC, Shine, TEAL, THRMS, xCAT
Librerías y Software de E/S	HDF5, Hercule, pnetCDF
Visualización	AVS/Express, EnSight, FieldView, Grace, IDL, POV-Ray, ParaView, Tecplot360, VTK, VisIt
Ambientes integrados de desarrollo	Eclipse+PTP
Ambientes integrados de solución de problemas	MATLAB, Octave, R
Virtualización	Eucalyptus, HPUC, Shadowfax, vSMP, Xen

En los últimos 15 años, el software HPC ha tenido que adaptarse y responder a varios desafíos. En primer lugar, la concurrencia en aplicaciones y sistemas ha crecido más de tres órdenes de magnitud. El modelo de programación primaria, MPI, ha tenido que crecer y cambiar para permitir esta escala. En segundo lugar, el aumento de la concurrencia tiene, en una base por núcleo, un arrastre para que la capacidad de memoria y E/S sea más baja, y el ancho de banda de la memoria, la E/S y la interconexión. En tercer lugar, en los últimos cinco años, la heterogeneidad y la diversidad arquitectónica han puesto un nuevo énfasis en la aplicación y la portabilidad del software.

6.3 LEXIS NEXSYS HPCC

LexisNexis es una empresa líder en el contenido de los datos, la agregación de datos y servicios de información, que de forma independiente desarrolló e

implementó una solución para la informática de uso intensivo de datos llamada HPCC (High-Performance Computing Cluster), que también se conoce como el superordenador Data Analytics (DAS). La visión LexisNexis para esta plataforma informática se representa en la figura.

El enfoque de LexisNexis también utiliza clústeres de productos básicos de hardware que ejecutan el sistema operativo Linux. Software de sistema personalizado y componentes de middleware fueron desarrollados y puestos en capas en la base del sistema operativo Linux para proporcionar el entorno de ejecución y el apoyo del sistema de archivos distribuido requerido para la computación de datos intensivos. Debido a que LexisNexis reconoció la necesidad de un nuevo paradigma de computación para hacer frente a sus crecientes volúmenes de datos, el enfoque de diseño incluyó la definición de un nuevo lenguaje de alto nivel para el procesamiento de datos en paralelo llamado ECL (Enterprise Data Control Language). El poder, la flexibilidad, capacidades avanzadas, velocidad de desarrollo, madurez, y la facilidad de uso del lenguaje de programación ECL es un factor distintivo principal entre la plataforma LexisNexis HPCC y otras soluciones de computación de datos intensivos.

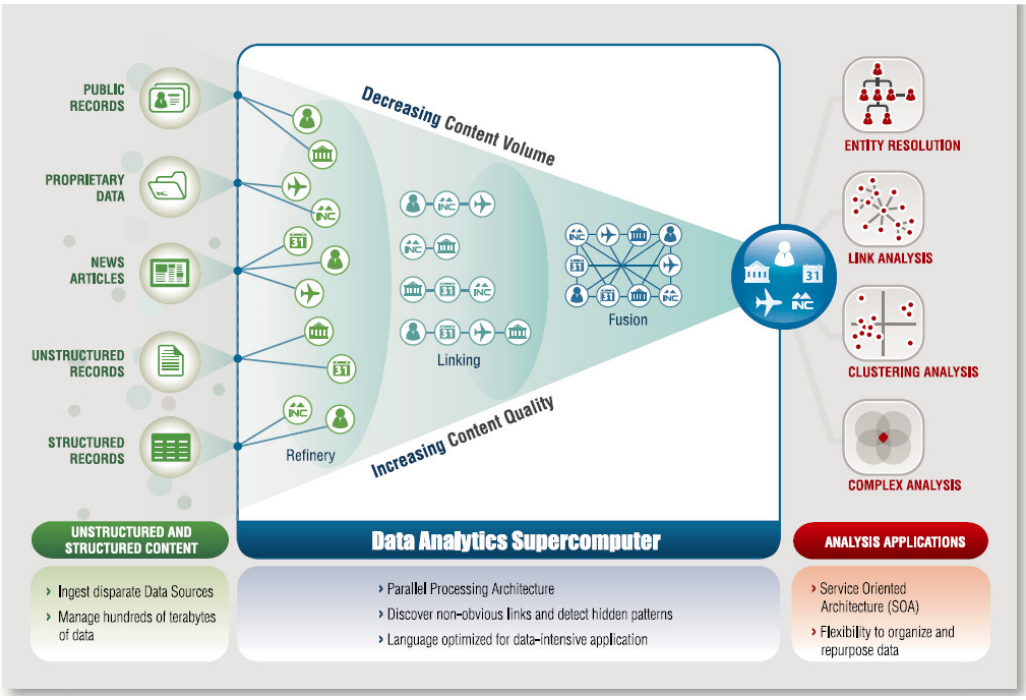

Figura 19 - Visión de LexisNexis de HPC para Analítica de Datos. Fuente LexisNexis

Los desarrolladores de LexisNexis reconocieron que para cumplir todos los requisitos de las aplicaciones informáticas intensivas de datos de una manera óptima requerían del diseño e implementación de dos entornos de procesamiento de clúster distintos, cada uno de las cuales se podrían optimizar de forma independiente para su propósito de procesamiento de datos en paralelo. La primera de estas plataformas se llama una refinería de datos cuyo objetivo general es el procesamiento general de grandes volúmenes de datos en bruto de cualquier tipo para cualquier propósito, pero normalmente se utiliza para la limpieza de datos y la higiene, el procesamiento ETL de los datos en bruto, la vinculación de registros y resolución de entidades, analítica compleja ad-hoc a gran escala, y la creación de datos introducidos y los índices de apoyo a las consultas estructuradas de alto rendimiento y aplicaciones de bodegas de datos. La Refinería de datos también se conoce como Thor, una referencia al mítico dios nórdico del trueno con el gran martillo, simbolizando la trituración de grandes cantidades de datos en información útil. Un clúster de Thor es similar en su función, entorno de ejecución, sistema de archivos, y las capacidades a las plataformas de Google y Hadoop MapReduce.

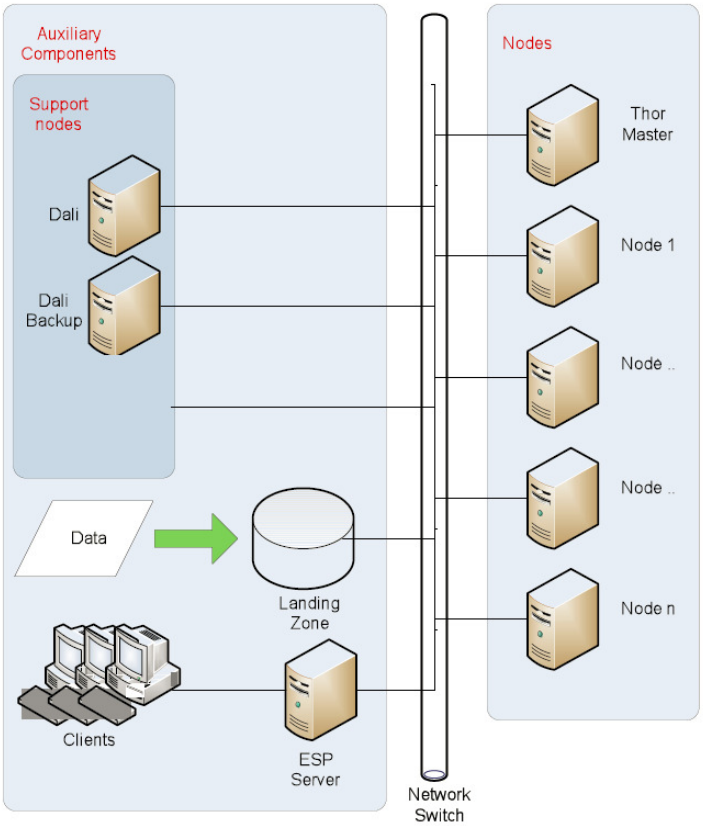

Figura 20 - Clúster de procesamiento Thor. Fuente: LexisNexis

La segunda de las plataformas de procesamiento de datos paralelos diseñada e implementada por LexisNexis se denomina Motor de Entrega Rápida de Datos. Esta plataforma se ha diseñado como una plataforma en línea de alto rendimiento de consulta y análisis estructurado o almacén de datos que entrega los requerimientos de procesamiento de acceso a datos en paralelo de las aplicaciones en línea a través de interfaces de servicios web de apoyo para miles de consultas y usuarios simultáneos con tiempos de respuesta inferiores a un segundo. El motor entrega rápida de datos también se conoce como Roxie, que es un acrónimo para Rapid Online XML Motor de mensaje. Roxie utiliza un sistema de archivos distribuido indexado especial para proporcionar procesamiento paralelo de consultas. Un clúster de Roxie es similar en su función y capacidades a Hadoop con las capacidades añadidas de HBase y Hive, pero proporciona significativamente mayor rendimiento ya que utiliza un entorno de ejecución y sistema de archivos más optimizado para el procesamiento en línea de alto rendimiento. Lo más importante, tanto los clústeres de Thor como de Roxie utilizan el mismo lenguaje de programación ECL para la implementación de aplicaciones, aumentando la continuidad y la productividad del programador.

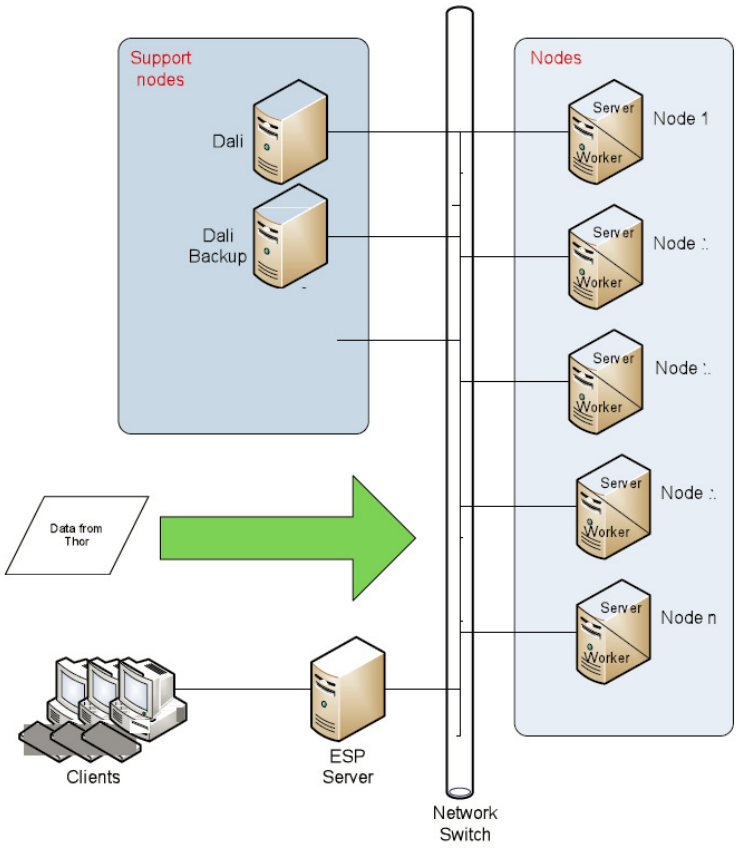

Figura 21 - Clúster de procesamiento Roxie. Fuente: LexisNexis

Un clúster de Roxie incluye múltiples nodos con servidores y trabajador de procesos para procesar consultas; un componente auxiliar adicional llamado un servidor ESP que proporciona interfaces para el acceso de clientes externos a la agrupación; y los componentes adicionales comunes que se comparten con un clúster Thor en un entorno HPCC. Aunque un clúster de procesamiento de Thor puede ser implementado y usado sin un clúster Roxie, un entorno HPCC que incluye un clúster Roxie también debe incluir un clúster Thor. Se requiere el clúster de Thor para construir los archivos de índice distribuidos utilizados por el clúster Roxie y desarrollar consultas en línea que se implementarán con los archivos de índice en el clúster Roxie.

La aplicación de dos tipos de plataformas de procesamiento de datos paralelos (Thor y Roxie) en el entorno de procesamiento HPCC sirviendo diferentes necesidades de procesamiento de datos permite a estas plataformas ser optimizados y afinados para sus fines específicos para proporcionar el más alto nivel de rendimiento posible del sistema para los usuarios. Esto es una clara ventaja en comparación con Hadoop donde la arquitectura MapReduce debe cubrirse con sistemas adicionales tales como HBase, Hive, y Pig que tienen diferentes objetivos y requisitos de procesamiento, y no siempre se correlacionan fácilmente en el paradigma MapReduce. Además, el enfoque LexisNexis HPCC incorpora la noción de un entorno de procesamiento que puede integrar clústeres Thor y Roxie, según sea necesario para satisfacer las necesidades de procesamiento completos de una organización. Como resultado, la escalabilidad se puede definir no sólo en términos del número de nodos en un clúster, sino en función de la cantidad de clusters y de qué tipo son necesarios para cumplir con los objetivos de rendimiento del sistema y las necesidades de los usuarios. Esto proporciona una flexibilidad significativa en comparación con los clústeres de Hadoop que tienden a ser islas independientes de procesamiento.

6.4 ROCKS HPC

Rocks es considerada una distribución de Linux creada con la intención de facilitar la creación de clústeres de computación de alto desempeño. Fue iniciada en el año 2000 como un trabajo contributivo entre la Asociación Nacional para la Infraestructura de Computación Avanzada (NPACI, por sus sigla en inglés) y el Centro de Supercomputación de San Diego (SDSC, pos sus sigla en inglés).

Rocks fue inicialmente basado en la distribución de RedHat Linux, pero las versiones modernas de Rocks están basadas en CentOS con una modificación del instalador Anaconda que simplifica la instalación masiva en muchos computadores.

Rocks busca simplificar la creación de clústeres para que la comunidad científica en general (no solo está dirigido a los participantes en la ciencias de la

computación) esté en capacidad de crear clústeres de alto desempeño y pueda desarrollar sus aplicaciones con el soporte a un ambiente distribuido y de ejecución en paralelo.

Algunos de los logros más importantes de la distribución Rocks han sido:

- Todos los nodos se configuran automáticamente al 100% sin intervención manual y con facilidad de personalización
- Permite que se instale y ejecute en cualquier tipo de equipo estándar y en una ambiente de equipos heterogéneos. Los clústeres 100% homogéneos no existen
- Ya que está optimizado en su instalación, permite que el sistema este operativo rápidamente facilitando la construcción de un supercomputador en horas en lugar de meses
- Para evitar las discrepancias entre el software de los nodos del sistema, que puede causar la parálisis del sistema, no se ofrece soporte a las actualizaciones al vuelo del sistema operativo, por lo que se trabaja orientado a la reinstalación en lugar de la reconfiguración

La pila de software que compone la solución de clúster de Rocks se representa a continuación:

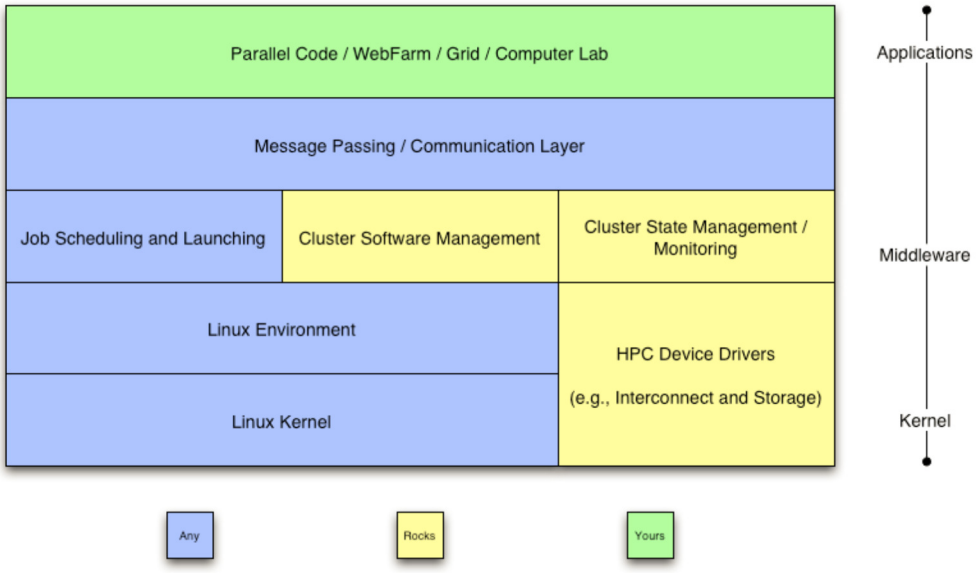

Figura 22 - Pila de software de Rocks Clúster. Fuente: www.rocksclusters.org

La personalización de la instalación con paquetes adicionales, y que se representan en la anterior figura con la franja superior, se hace al momento de la instalación a través de paquetes especiales llamados Roles. Los roles extienden el

sistema integrándose perfectamente y de forma automática en los mecanismos de gestión y de envasado utilizados por el software de base, en gran medida simplifica la instalación y configuración de un gran número de ordenadores.

Durante el proceso de instalación se hará uso del repositorio de software recolectando todos los posibles paquetes de software. Para construir una versión (distribución) personalizada del software se trabaja con el archivo kickstart en el que se condensarán todas las instrucciones de instalación con información descriptiva y la configuración de los nodos.

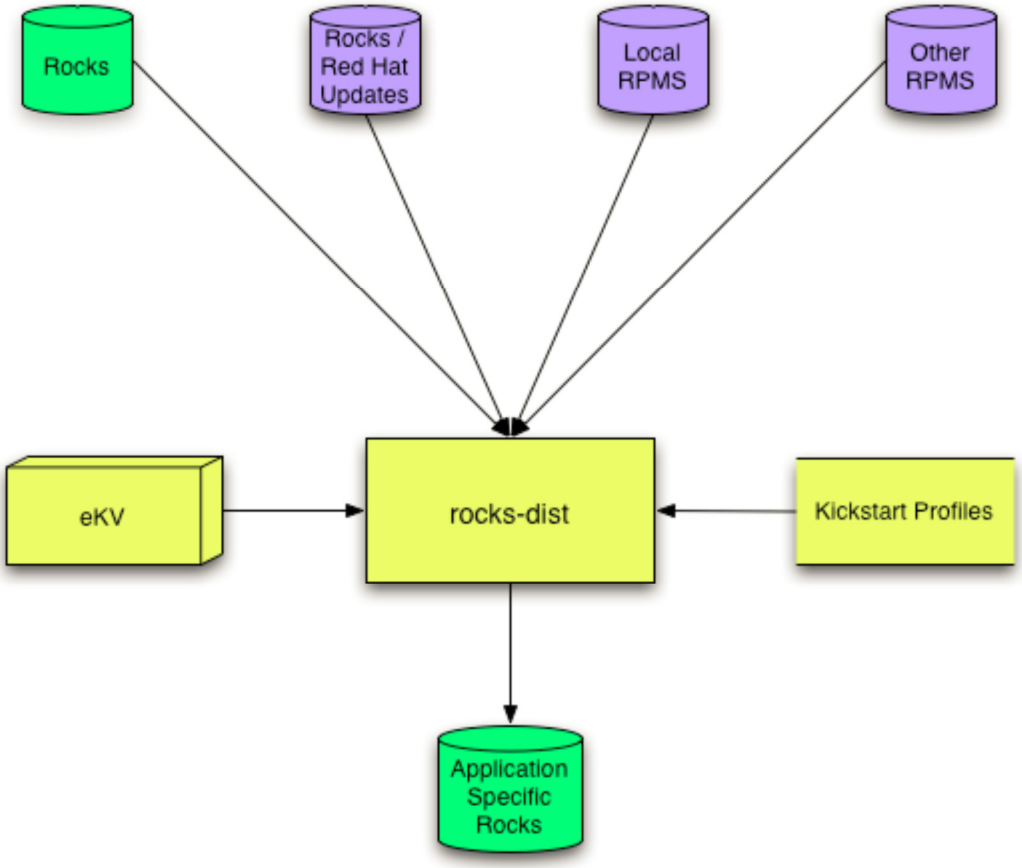

Figura 23 - Construcción de una versión personalizada de Rocks. Fuente: www.rocksclusters.org

Cada instalación requiere los roles Rocks Base y HPC. El núcleo de instalación ofrece varios sabores de MPICH, Ganglia y PVFS. Si desea que el software adicional que no es parte de la instalación de Rocks básica, se tendrá que descargar roles adicionales.

Actualmente los roles disponible incluyen los siguientes:

Rol Sun Grid Engine (SGE): Este rol incluye el Sun Grid Engine, un sistema de trabajo en cola para las redes. Esta es una alternativa, que tiene en cuenta la red en cuadrícula (grid-aware), para OpenPBS. Este es un software de código abierto de gestión distribuida.

Rol Grid: El rol de la Iniciativa grid Middleware NSF (NMI) contiene un complemento completo de software de grid, incluyendo el kit de herramientas Globus, Condor-G, Servicio de Red de Tiempo, y MPICH-G2, por nombrar sólo unos pocos.

Rol Intel: Este rol instala y configura el compilador Intel C y el compilador Intel Fortran. (Todavía necesitará licencias de Intel). También incluye los entornos MPICH construidos para estos compiladores.

Rol Area 51: Este rol actualmente incluye Tripwire y chkrootkit. Tripwire es un paquete de auditoría de seguridad. chrootkit examina un sistema para cualquier indicación de que un rootkit se ha instalado.

Rol Scalable Cluster Environment (SCE): Este rol incluye el software OpenSCE que se originó en la Universidad Kasetsart, Tailandia.

Rol Java: Este rol contiene la máquina virtual de Java.

Rol PBS: El rol Portable Batch System incluye el software de programación y colas OpenPBS y Maui.

Rol Condor: Este rol incluye el software de gestión de carga de trabajo Cóndor. Condor ofrece colas de trabajo, programación y gestión de prioridades junto con el seguimiento y la gestión de recursos.

Algunos roles no están disponibles para todas las arquitecturas. Está bien instalar más de un rol, a fin de obtener lo que se piensa que puede necesitar a futuro. En general, no se podrá añadir un rol una vez instalado el clúster. (Esto debe cambiar en el futuro.)

6.5 OSCAR

OSCAR (Clúster Open Source de recursos de aplicaciones) es un paquete de software que está diseñado para simplificar la instalación del clúster. Una colección de software de clúster de código abierto, OSCAR incluye todo lo que es probable que se necesite para un clúster de alto rendimiento dedicado. OSCAR lo lleva completamente a través de la instalación del clúster. Al descargar, instalar y

ejecutar OSCAR, se tendrá un clúster funcional por completo cuando haya finalizado.

Los objetivos de diseño de OSCAR incluyen el uso del software de la mejor clase, la eliminación de la descarga, instalación y configuración de los componentes individuales, y avanzar hacia la estandarización de los clústeres. OSCAR, se dice, reduce la necesidad de expertos en la creación de un clúster. En la práctica, tal vez sea más apropiado decir que OSCAR retrasa la necesidad de expertos y permite crear un clúster en pleno funcionamiento antes de dominar todas las habilidades que eventualmente va a necesitar. OSCAR hace que sea muy fácil de experimentar con paquetes y reduce drásticamente la barrera para empezar.

OSCAR fue creado y es mantenido por el Grupo de Clúster Abierto (http://www.openclustergroup.org), un grupo informal dedicado a simplificar la instalación y el uso de los clusters y la ampliación de su uso. Con los años, una serie de organizaciones y empresas han apoyado el Grupo de Clúster Abierto, incluyendo Dell, IBM, Intel, NCSA, y ORNL, por mencionar sólo algunos.

OSCAR está diseñado con la informática de alto rendimiento en mente. Básicamente, está diseñado para ser utilizado con un clúster asimétrico. A menos que se personalice la instalación, los nodos informáticos están destinados a ser dedicados al clúster. Por lo general, no se hace conexión directamente a los nodos cliente sino que se trabaja desde el nodo principal. (Aunque OSCAR configura SSH para que pueda conectarse a los clientes sin una contraseña, esto se hace principalmente para simplificar el uso del software de clúster.)

Mientras que un hardware idéntico no es un requisito absoluto, instalar y administrar un clúster OSCAR es mucho más simple cuando se utiliza un hardware idéntico.

Es probable que todo lo que realmente necesita para empezar a trabajar con un clúster de alto rendimiento se incluye ya sea en el tar-ball de OSCAR o como parte del sistema operativo base en el que OSCAR se instala. No obstante, OSCAR proporciona un script, el Paquete de Descargas de Oscar (OPD, por su sigla en inglés) que simplifica la descarga e instalación de paquetes adicionales que están disponibles en los repositorios de OSCAR en un formato compatible con OSCAR. OPD es tan fácil de usar que a efectos prácticos cualquier paquete disponible a través de OPD se puede considerar parte de OSCAR. OPD puede invocarse como un programa independiente o desde el asistente de instalación de OSCAR, el instalador basado en GUI de OSCAR. Paquetes adicionales disponibles para OPD incluyen cosas como controladores Myrinet y apoyo a clientes OSCAR livianos, así como paquetes de gestión como Ganglia.

Los paquetes de OSCAR se dividen en tres categorías. Se deben instalar los paquetes principales. Paquetes incluidos se distribuyen como parte de OSCAR, pero usted puede optar por salir de la instalación de estos paquetes. Los paquetes de terceros son paquetes adicionales que están disponibles para su descarga y que son compatibles con OSCAR, pero no son necesarios. Hay seis paquetes principales en el corazón de OSCAR que debe instalar:

Núcleo (Core): Este es el paquete núcleo de OSCAR.

C3: El conjunto de herramientas Cluster, Comando y Control proporciona una interfaz de administración de línea de comandos.

Conmutador de Ambiente: Esto se basa en Modules, un script en Perl que permite al usuario realizar cambios en el entorno de los shells futuros. Por ejemplo, Switcher permite a un usuario cambiar entre MPICH y LAM / MPI.

Oda: La aplicación de base de datos OSCAR ofrece una base de datos central para OSCAR.

perl-qt: Esta es la interfaz Perl orientada a objetos para el kit de herramientas Qt GUI.

SIS: La suite de instalación del sistema se utiliza para instalar los sistemas operativos en los clientes.

OSCAR incluye una serie de paquetes y scripts que se utilizan para crear el clúster. El asistente de instalación le dará la opción de decidir qué incluir:

disable-services: Este script deshabilita los servicios innecesarios en los clientes, tales como kudzu, slocate, y servicios de correo como sendmail.

Networking: Este script configura el servidor de clúster como un servidor de nombres de caché para los clientes.

Ntpconfig: Este script configura NTP. OSCAR utiliza NTP para sincronizar los relojes dentro del clúster.

kernel_picker: Esto se utiliza para cambiar el kernel utilizado en su imagen SIS antes de la construcción de los nodos del clúster.

Loghost: Esto configura los ajustes de registro del sistema, por ejemplo, se configura nodos para reenviar mensajes Syslog para el nodo principal.

OSCAR proporciona herramientas adicionales del sistema, ya sea como parte de la distribución OSCAR o a través de OPD, que se utilizan para administrar el clúster:

Autoupdate: Este es un script de Perl utilizado para actualizar los clientes y el servidor (similar a up2date o autorpm).

Clumon (opd): Clumon es un sistema de monitorización del rendimiento basado en la web de la NCSA.

Ganglia (opd): Ganglia es una herramienta de monitoreo de sistema y entorno de ejecución en tiempo real.

MAUI: Este planificador de tareas se utiliza con OpenPBS.

Controladores Myrnet (opd): Si se usa Myrnet hardware, necesita cargar los controladores correspondientes.

openPBS: El sistema de lotes portátil es un sistema de gestión de carga de trabajo.

Pfilter: Este paquete se utiliza para generar conjuntos de normas utilizadas para el filtrado de paquetes.

PVFS (opd): Parallel Virtual File System es un sistema de archivos virtual paralelo de alto rendimiento, escalable.

OPIUM:

Este es el conjunto de herramientas instalador de contraseña y de gestión de usuarios de OSCAR.

thin client (opd): este paquete provee soporte para nodos OSCAR sin disco.

Torque (opd): El gestor de escala Tera de recursos y la cola de los recursos de código abierto se basa en OpenPBS.

VMI (opd): La interfaz de máquina virtual proporciona una capa de middleware de comunicaciones para SAN a través de las redes.

Por supuesto, cualquier cluster de alto rendimiento no estaría completo sin las herramientas de programación. La distribución OSCAR incluye cuatro paquetes, mientras que dos más (como se ha señalado) están disponibles a través opd:

HDF5: Esta es una biblioteca de formato de datos jerárquica para el mantenimiento de los datos científicos.

LAM/MPI: Esta es una implementación de las bibliotecas de interfaz de paso de mensajes (MPI).

MPICH: Esta es otra aplicación de las bibliotecas de interfaz de paso de mensajes (MPI).

MPICH-GM (opd): Este paquete proporciona MPICH con soporte para mensajes de bajo nivel que pasa por las redes Myrnet.

MPICH-VMI (opd): Esta versión de MPICH utiliza VMI.

PVM: Este paquete proporciona el sistema de máquina virtual paralelo, otra librería de paso de mensajes.

Si se instalan los cuatro paquetes incluidos, por defecto, se debe cubrir todas las necesidades de programación.

Los elementos presentados harán parte de la composición del sistema OSCAR y que se ve reflejado a continuación.

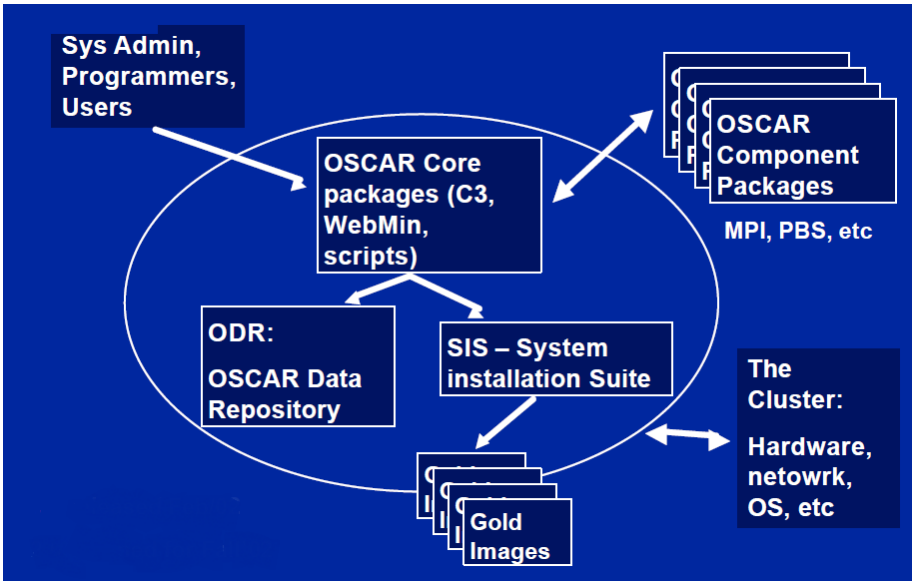

Figura 24 - Arquitectura OSCAR Clúster. Fuente: Intel

Además, OSCAR instalará y configurará (o reconfigurar) una serie de servicios y paquetes suministrados como parte de su versión de Linux. Estos potencialmente incluyen Apache, DHCP, NFS, mySQL, openSSL, OpenSSH, rrdtool, pcp, PHP, Python, rsync, tftp, etc. Exactamente cuál de ellos es realmente instalado o configurado dependerá de qué otro software se decide instalar. En el improbable caso de que no se esté satisfecho con la forma en que OSCAR configura cualquiera de estos, tendrá que volver a reconfigurar después de que se complete la instalación.

6.6 openMosix

openMosix es un software que extiende el núcleo de Linux para que los procesos puedan migrar de forma transparente entre las diferentes máquinas dentro de un clúster con el fin de distribuir más uniformemente la carga de trabajo.

Básicamente, el software openMosix incluye tanto un conjunto de parches del kernel y herramientas de apoyo. Los parches se extienden el núcleo para proporcionar apoyo a los procesos de movimiento entre las máquinas del clúster. Típicamente, la migración del proceso es totalmente transparente para el usuario. Sin embargo, mediante el uso de las herramientas proporcionadas con openMosix, así como las herramientas de terceros, puede controlar la migración de los procesos entre las máquinas.

Si se quiere acelerar un conjunto de tareas computacionalmente costosas, a continuación se describe como openMosix podría ser utilizado. En este escenario se tiene una docena de archivos para comprimir mediante un programa intensivo de la CPU en una máquina que no es parte de un clúster openMosix. Se podría comprimir cada archivo uno a la vez, esperando que uno termine antes de iniciar la siguiente. O se pueden ejecutar todas las compresiones simultáneamente iniciando cada compresión en una ventana aparte o mediante la ejecución de cada compresión en el fondo (que finaliza cada línea de comandos con un &). Por supuesto, de cualquier manera tendrá aproximadamente la misma cantidad de tiempo y se carga el ordenador, mientras que los programas se están ejecutando.

Sin embargo, si el equipo forma parte de un clúster openMosix, esto es lo que va a pasar: En primer lugar, se dará comienzo a todos los procesos que se ejecutan en el equipo. Con un clúster openMosix, después de unos segundos, procesos comenzarán a migrar de este equipo con mucha carga a otros equipos ociosos o menos cargados en los clústeres. Si se tiene una docena de máquinas ociosas en el clúster, cada compresión debe ejecutarse en un equipo diferente. La máquina tendrá una sola compresión que se ejecuta en él (junto con una pequeña sobrecarga en el original) por lo que aún puede ser capaz de usarlo. Y la docena

de compresiones tendrán sólo un poco más de lo que normalmente sería necesario para hacer una sola compresión.

Si no se tiene una docena de computadoras, o algunos de los equipos son más lentos que los demás, o de alguna otra manera está cargado, openMosix moverá los trabajos en torno a la mejor manera posible para equilibrar la carga. Una vez que el clúster está configurado, todo esto se hace de forma transparente por el sistema. Normalmente, solo hay que empezar los trabajos. openMosix hace el resto. Por otro lado, si se quiere controlar la migración de puestos de trabajo de un equipo a otro, openMosix le proporciona las herramientas para hacer precisamente eso.

openMosix se originó como una derivación del proyecto anterior MOSIX (sistema operativo multicomputador para Unix). El proyecto openMosix comenzó cuando la estructura de licencias para MOSIX se alejó de la Licencia Pública General. Hoy, se ha convertido en un proyecto en sí mismo. El proyecto MOSIX original es todavía muy activo bajo la dirección de Amnón Barak (http://www.mosix.org). openMosix es obra de Moshe Bar, originalmente un miembro del equipo de MOSIX, y un número de voluntarios.

Como se ha señalado antes, un enfoque para compartir un cálculo entre los procesadores en un equipo de caja única con varias CPU es la computación de multiprocesamiento simétrico (SMP). openMosix ha descrito, con precisión, como convertir un conjunto de ordenadores en una máquina virtual SMP, con cada nodo que proporciona una CPU. openMosix es potencialmente mucho más barato y escala mucho mejor que SMP, pero la sobrecarga de comunicación es mayor. (openMosix funciona con ambos sistemas, de un solo procesador y sistemas SMP.) openMosix es un ejemplo de lo que a veces se llama una sola agrupación imagen del sistema (SSI) dado cada nodo del clúster tiene una copia del kernel del sistema operativo.

La granularidad para openMosix es el proceso. Los programas individuales, como en el ejemplo de compresión, pueden crear los procesos o los procesos pueden ser el resultado de diferentes bifurcaciones de un solo programa. Sin embargo, si tiene una tarea de cómputo intensivo que hace todo en un solo proceso (e incluso si se utilizan múltiples hilos), entonces, ya que es sólo un proceso, no puede ser compartida entre procesadores. Lo mejor que puede esperar es que se migre a la máquina más rápida disponible en el clúster.

No todos los procesos migran. Por ejemplo, si un proceso dura sólo unos pocos segundos (muy aproximadamente, menos de 5 segundos dependiendo de una serie de factores), él no tendrán tiempo para migrar. Actualmente, openMosix no funciona con múltiples procesos que utilizan memoria grabable compartida, como servidores web. Del mismo modo, procesos que hacen la manipulación directa de

dispositivos de E / S no migrarán. Y los procesos que utilizan la programación en tiempo real no se migrarán. Si un proceso que ya ha migrado a otro procesador e intenta hacer alguna de estas cosas, el proceso va a migrar de nuevo a su único nodo casa (UHN), el nodo donde se creó inicialmente el proceso, antes de continuar.

Para apoyar el proceso de migración, openMosix divide procesos en dos partes o contextos. El contexto de usuario contiene el código del programa, pila, datos, etc., y es la parte que puede migrar. El contexto del sistema, que contiene una descripción de los recursos que el proceso se adjunta y la pila del núcleo, no migra, sino que permanece en la UHN.

openMosix utiliza una política de asignación de recursos de adaptación. Es decir, cada nodo monitorea y compara su propia carga con la carga de una porción de los otros equipos dentro del clúster. Cuando un ordenador encuentra un ordenador con la carga más ligera (basado en la capacidad global de la computadora), intentará migrar un proceso para la computadora de la carga más ligera, creando así una carga más equilibrada entre los dos. A medida que las cargas sobre las computadoras individuales cambia, por ejemplo, cuando comienzan los trabajos o acaban, los procesos migran entre los equipos para reequilibrar las cargas a través del clúster, adaptándose dinámicamente a los cambios en las cargas.

Los nodos individuales, que actúan como sistemas autónomos, deciden qué procesos migran. Las comunicaciones entre los pequeños conjuntos de nodos dentro del clúster se utilizan para comparar las cargas que se asignaron al azar. En consecuencia, los clústeres se escalan bien debido a este elemento aleatorio. Como las comunicaciones están dentro de subconjuntos del clúster, los nodos tienen información limitada pero reciente sobre el estado de todo el clúster. Este enfoque reduce la sobrecarga y la comunicación.

Si bien la comparación de carga y migración son procesos generalmente automáticos dentro de un clúster, openMosix proporciona herramientas para controlar la migración. Es posible alterar la percepción del clúster de cómo se ha cargado en gran medida un equipo individual, para atar los procesos a un equipo específico, o para bloquear la migración de los procesos a un ordenador. Sin embargo, un control preciso para la migración de un grupo de procesos no es práctico con openMosix en este momento.

El API openMosix utiliza los valores de los archivos planos en /proc/hpc para registrar y controlar el estado del clúster. Si necesita información sobre la configuración actual, querer hacer realmente la gestión de bajo nivel, o escribir scripts de gestión, puede mirar o escribir en estos archivos.

6.7 OTROS KITS DE HPC DISPONIBLES

Uno de los aspectos que no se enfatiza, pero es común a todas las soluciones de software presentadas en este capítulo, para la computación de alto desempeño es que se han escogido solo aquellas que se pueden calificar como suites o distribuciones que incluyen todas las herramientas necesarias para la implementación de soluciones de computación de alto desempeño. Con excepción quizá de openMosix, pero no se dedicará tiempo a discutir si debe hacer parte de la lista o no.

Otro aspecto importante es que solo se han detallado aquellas soluciones que se ofrecen de manera gratuita al público y cuentan con una organización y comunidades de usuarios activas que aseguren el respaldo que cualquier organización comercial requiere al momento de buscar implementar la solución.
De esta manera, se ha omitido intencionalmente un conjunto de soluciones de HPC que no se ajustan a estos criterios de manera simultánea y que se listan a continuación:

- IBM Platform Computing: http://www-03.ibm.com/systems/platformcomputing/
- DIET de SysFera-DS: http://graal.ens-lyon.fr/diet/
- ProACTIVE: http://proactive.activeeon.com//
- Univa UniCloud: http://www.univa.com/products/unicloud.php
- Red Hat Enterprise MRG Grid: http://www.redhat.com/hpc-grid/
- SuSE Linux Enterprise Server: https://www.suse.com/products/server/hpc.html

7. IMPLEMENTACION DEL PROTOTIPO VIRTUALIZADO DE HPC PARA LA ANALITICA DE DATOS

Como se detalla en el capítulo anterior, hay una gran cantidad de elementos de software que pueden usarse para constituir el sistema HPC y estos se traducen en un reto más que deben manejar las organizaciones interesadas en las capacidades de estos sistemas.

Una situación típica de las distribuciones de Linux, que se suele potenciar cuando se hace uso de distribuciones que son de libre acceso y no cuentan con un respaldo corporativo, es que los componentes instalados en una máquina pueden diferir de los que se han instalado en su "gemela" por origen de aplicaciones y de librerías o por procesos de actualización realizados en momentos diferentes.

Esto significa que se debe ampliar el alcance de los controles de versión y de instalación de software en cada nodo del sistema, tarea que puede volverse imposible de sostener a medida que se va incrementando el número de nodos y las versiones de los paquetes instalados en cada nodo varia.

Esta situación típica de las distribuciones de Linux obliga a buscar alternativas para manejar un conjunto de software de OS + HPC que se mantenga constante y se pueda fácilmente distribuir y actualizar en cada uno de los nodos sin poner en riesgo la integridad de todo el sistema ni su capacidad de renovación y actualización.

En este capítulo se presentan las definiciones y consideraciones tenidas en cuenta para la conceptualización del prototipo de HPC y Analítica de Datos.

7.1 SELECCIÓN DE LOS COMPONENTES DE SOFTWARE

A lo largo del documento se han mencionado algunos elementos a considerar al momento de hacer la selección de entre las diversas alternativas presentadas del software que se puede utilizar para construir un sistema de cómputo HPC.

Si bien no es objetivo de esta investigación conducir un estudio de dichas alternativas y realizar una evaluación exhaustiva de sus condiciones de mercado y técnicas, si se mencionan a continuación algunos aspectos tenidos en cuenta para la selección de la herramienta que se incluirá como parte de las recomendaciones al final de este capítulo.

1. Por tratarse de un modelo pensado para la academia y como alternativa a las soluciones propietarias y con costos de licenciamiento, se consideran solamente soluciones sin costos de licenciamiento y, preferiblemente, de código abierto.

2. Para no perder de vista las necesidades de las empresas que puedan estar interesadas en implementar este modelo, se privilegian aquellas soluciones que ofrezcan soporte corporativo y servicios consultivos.
3. Aunque se busca modelar la solución con el "estado del arte" en HPC, se considera más importante la madurez y estabilidad sobre la última versión y la nueva tendencia.
4. Se requiere propiciar la homogeneidad y la estandarización de los componentes de la solución por lo que se prefiere el uso de una suite completa en lugar de componentes aislados.
5. Finalmente, se considera la implementación solamente sobre sistema operativo Linux, en una distribución que se ajuste a las condiciones antes descritas para la solución HPC.

Dejando de lado las suites de HPC que se ofrecen comercialmente y requieren de la adquisición de licencias para su utilización, se identificaron tres alternativas de software que facilitan el aprovisionamiento de los nodos del sistema HPC mientras que al mismo tiempo se preocupan de la consistencia y control necesarios para el mantenimiento de todo el software que lo compone. Estos son ROCKS HPC, LexisNexis HPCC y OSCAR.

La primera validación de estas alternativas arrojó que ninguna de ellas cumplia totalmente con las condiciones establecidas:

- LexisNexis HPCC es una herramienta que aunque no requiere de licenciamiento, tiene soporte corporativo, es una herramienta madura y estable y provee una instalación estándar se trata de software propietario sin gran divulgación en el mercado. Es una alternativa atractiva para las empresas pero no tanto para la academia.
- OSCAR es una herramienta que ofrece madurez, estabilidad y consistencia en la instalación pero no es homogéneo a nivel de sistema operativo y no ofrece soporte corporativo. Adicionalmente, la implementación de las herramientas analíticas no está incluida en la suite. Es adecuado para la academia pero no así para las empresas.
- ROCKS HPC parece la alternativa más sólida, no tiene costos de licenciamiento, tiene la madurez y estabilidad requerida, ofrece una muy buena homogeneidad en su despliegue y aunque la instalación de las herramientas analíticas no está incluida en la suite, su funcionalidad de personalización de la distribución facilitan homogeneizar las herramientas analíticas. Sin embargo, no ofrece un soporte corporativo aunque tiene mucho respaldo de la academia. Se puede decir que es ideal para la academia pero no así para las empresas.

Teniendo ROCKS HPC como una alternativa más viable para el objetivo primario de la investigación se quiso estudiar las alternativas de soporte corporativo que

estarían disponibles encontrándose con una organización que ha hecho una variación de ROCKS HPC creando una versión especializada que recibe soporte corporativo.

STACKIQ CLUSTER MANAGER, es el producto de StackIQ (http://www.stackiq.com/product/) que se presenta como una variación de Rocks especializada en Big Data, Cloud y HPC. Este producto mantiene las funcionalidades de Rocks al trabajar con roles especializados al mismo tiempo que permite crear nuevos roles o modificar los existentes. Esta situación hace ideal esta suite para la implementación de HPC y Analítica de Datos en la academia y las empresas.

7.2 EVALUACIÓN DE LA INFRAESTRUCTURA DE HARDWARE DISPONIBLE

El Politécnico Grancolombiano – Institución Universitaria ha puesto a disposición de este trabajo de investigación dos equipos de cómputo de tipo servidor, de marca Dell y modelo PowerEdge 2950.

Estos servidores cuentan cada uno con dos procesadores de doble núcleo de 1.6 GHz, 6 o 4 GB de memoria RAM, 3 unidades de disco duro de 146 GB cada uno, doble tarjeta de red de 1Gb y doble fuente de poder.

Los requisitos de hardware de StackIQ Cluster Manager son:

- Nodo Frontal
  - Capacidad disco: 100 GB
  - Capacidad Memoria: 2 GB
  - Ethernet: 2 puertos físicos
  - Orden Boot BIOS: CD, Hard Disk
- Nodo de Cómputo
  - Capacidad disco: 100 GB
  - Capacidad memoria: 1 GB
  - Ethernet: 1 puerto físico
  - Orden Boot BIOS: PXE (Boot red), Hard Disk

Las condiciones físicas del hardware disponible permiten que se modele el prototipo con las siguientes características:

- Host 1
  - Nodo frontal
  - 2 Nodos de cómputo
- Host 2
  - 3 Nodos de cómputo

En estas condiciones cada Host tendrá acceso a un procesador dedicado, memoria de 2 GB para el Host 1 y memoria de 1 GB para el Host 2 y 200 GB de espacio en disco para los servidores virtuales en cada Host. En el Host 1 la interfaz de red 0, eth0, será la red privada del clúster y la interfaz 1, eth1, será la red de acceso público. En el Host 2 solo se hará uso de la interfaz eth0 para conectarse a la red interna del clúster.

7.3 CONSTRUCCIÓN DEL PROTOTIPO

Para la construcción del prototipo se considera que los componentes de hardware se encuentran dispuestos de acuerdo con el siguiente diagrama:

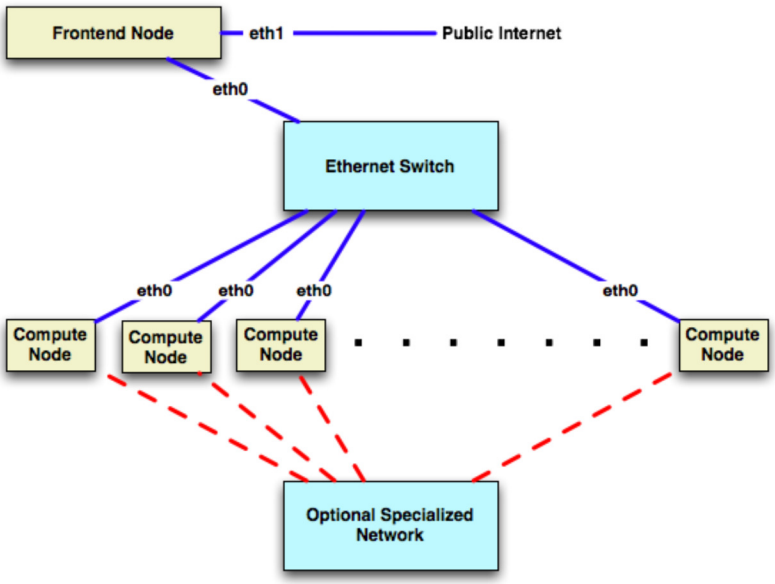

Figura 25 - Diagrama de interconexión para StackIQ. Fuente: StackIQ.com

En el escenario que se está contemplando en esta investigación los nodos Frontal y de Cómputo serán máquinas virtuales preparadas para ejecutarse según la distribución dada en el numeral anterior.

Se considera que la instalación y configuración de los host Linux así como la preparación de los nodos virtuales no es materia de esta investigación y, por lo tanto, no se detalla en el presente documento. El uso de máquinas virtuales está sujeto a las condiciones particulares de la implementación que es viable hacer y no se trata de componentes normalmente presentes en toda implementación de HPC y Analítica de datos.

7.3.1 INSTALACIÓN Y CONFIGURACIÓN DEL NODO FRONTAL

Para iniciar la instalación, se necesita uno de los siguientes paquetes de software:

- StackIQ Enterprise HPC
- StackIQ Enterprise Data
- StackIQ Enterprise Cloud

Todos los paquetes de software contienen los siguientes roles:

- Cluster-Core – Este contiene el software requerido para instalar, configurar, monitorear y gestionar el clúster
- OS – Este contiene el subconjuto de paquetes de CentOS-6.5 y las actualizaciones de CentOS-6.5 hasta Feb 17 2014

1. Se inserta uno de los DVD de StackIQ Enterprise y se reinicia la máquina

2. Después de que el nodo Frontal hace inicia desde el DVD, se ve:

Figura 26 - Instalación StackIQ, bienvenida. Fuente: StackIQ.com

3. Las siguientes pantallas no aparecen durante la instalación si se cuenta con un servidor DHCP en su red pública que atienda la solicitud del nodo Frontal

Si se ve la siguiente pantalla:

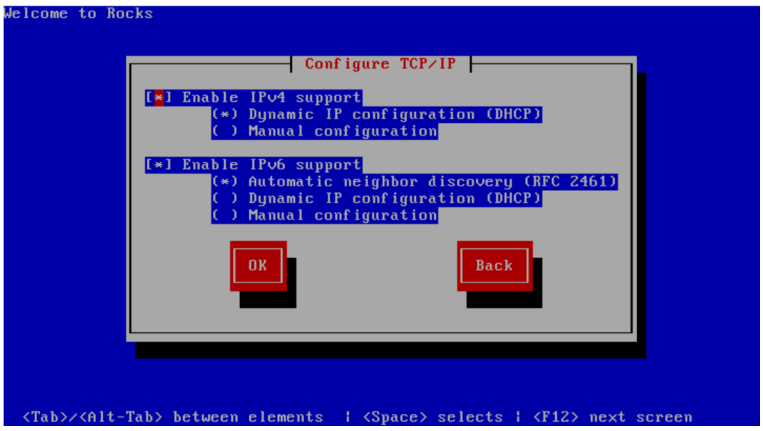

Figura 27 - Instalación StackIQ, configuración TCPIP paso 1. Fuente: StackIQ.com

Se debe: 1) habilitar soporte IPv4, 2) seleccionar configuración manual para el soporte de IPv4 (no DHCP) y, 3) deshabilitar el soporte IPv6. La pantalla lucirá como:

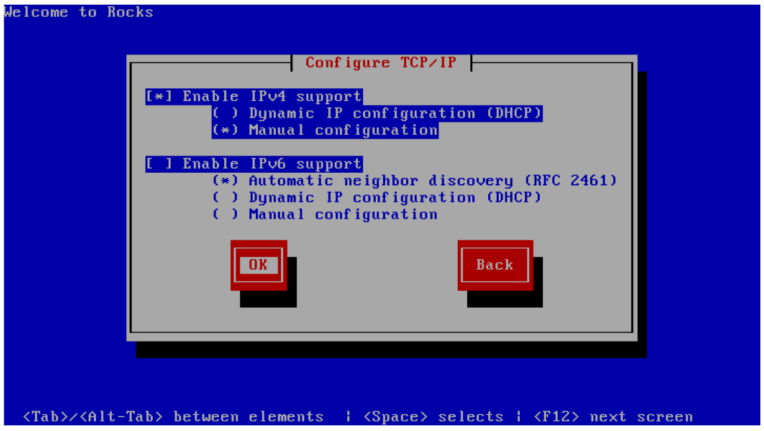

Figura 28 - Instalación StackIQ, configuración TCPIP paso 2. Fuente: StackIQ.com

Después se selecciona "OK". Entonces se verá la pantalla de configuración manual de TCP/IP:

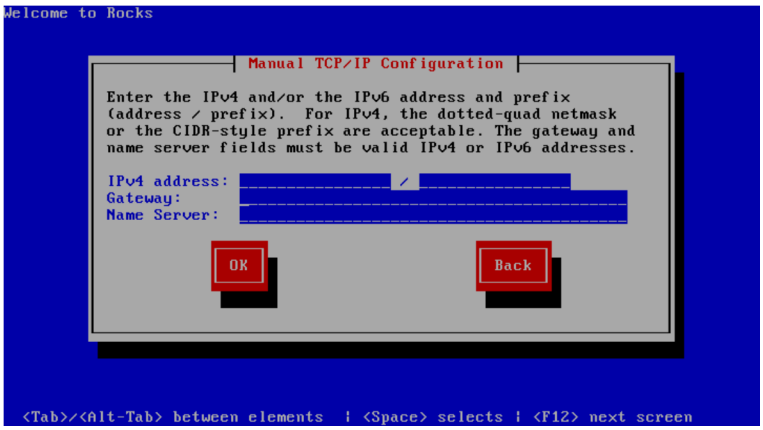

Figura 29 - Instalación StackIQ, configuración TCPIP paso 3. Fuente: StackIQ.com

En esta pantalla, se ingresa la configuración IP pública. Este es un ejemplo de la dirección IP pública ingresada para un nodo Frontal:

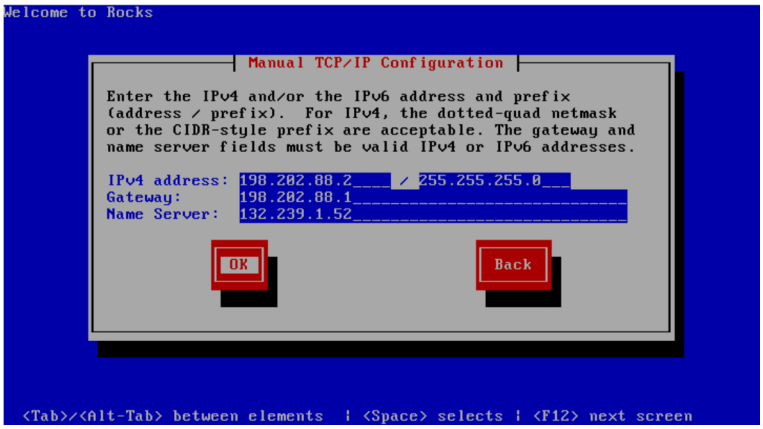

Figura 30 - Instalación StackIQ, configuración TCPIP paso 4. Fuente: StackIQ.com

Después de llenar la información de IP pública se presiona "OK".

4. Pronto, aparece una pantalla que luce como la siguiente:

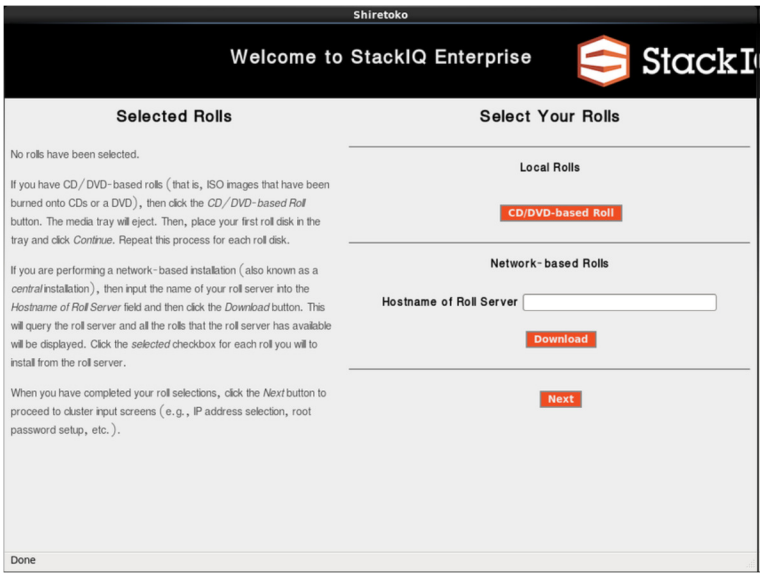

Figura 31 - Instalación StackIQ, seleccion de roles 1. Fuente: StackIQ.com

Desde esta pantalla se seleccionan los roles

En este procedimiento, se usará solamente un medio en DVD, así que solo se hará “click2 en el botón ’CD/DVD-based Roll’.

5. Al seleccionar el botón ’CD/DVD-based Roll’. Se observa la siguiente pantalla:

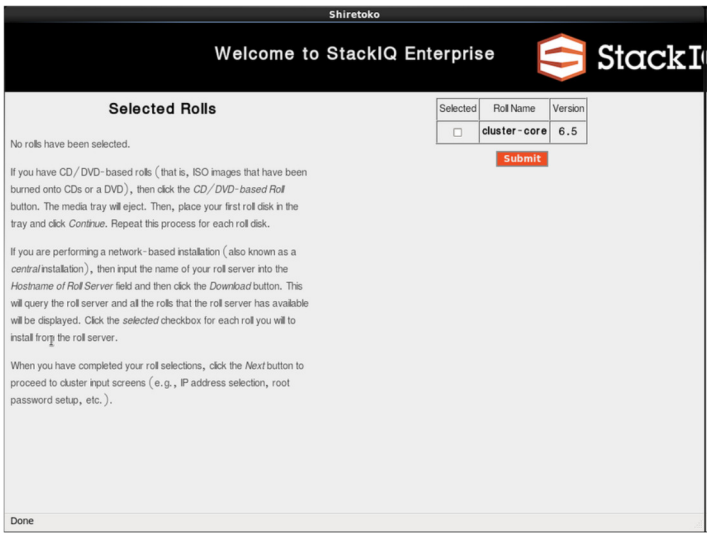

Figura 32 - Instalación StackIQ, seleccion de roles 2. Fuente: StackIQ.com

Se selecciona el rol "Cluster-Core" y se hace “click” en "Submit"

Se hace “Click” en el botón ’Continue’.

6. La pantalla muestra que se ha seleccionado adecuadamente el rol Cluster-Core.

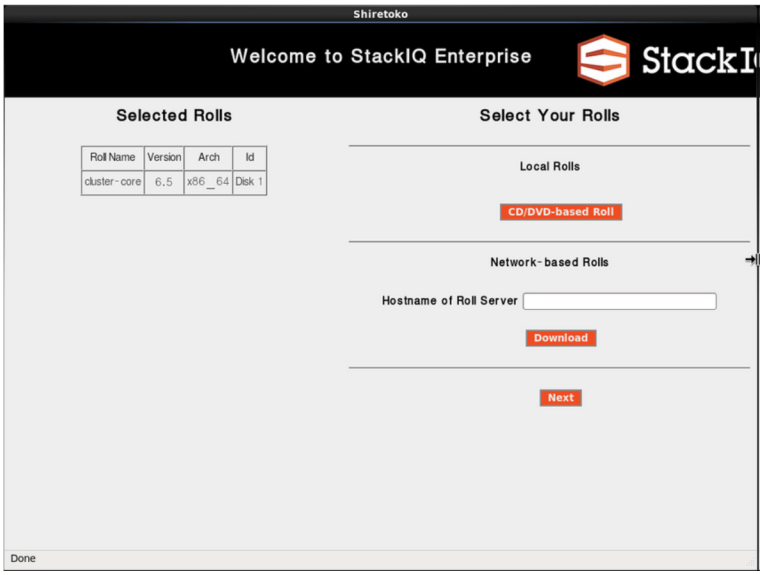

Figura 33 - Instalación StackIQ, seleccion de roles 3. Fuente: StackIQ.com

Se deben repetir los pasos 3 a 5 para cualquier otro rol que se quiera instalar.

7. Cuando se han seleccionados todos los roles asociados a un nodo frontal básico, la pantalla debe lucir como la siguiente:

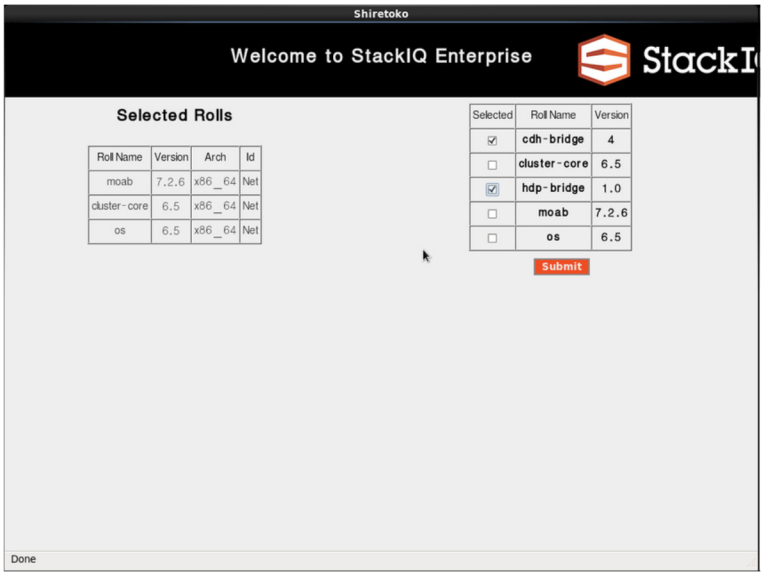

Figura 34 - Instalación StackIQ, seleccion de roles 4. Fuente: StackIQ.com

Cuando se termina la selección de roles se hace “click” en el botón ’Next’.

8. Entonces se muestra la pantalla de información del Clúster:

Figura 35 - Instalación StackIQ, Información clúster. Fuente: StackIQ.com

Nota: El único campo importante en esta pantalla es el campo Fully-Qualified Host Name (todos los otros son campos opcionales).

Se debe escoger el nombre de la máquina cuidadosamente. El nombre de máquina es escrito en docenas de archivos, tanto en el nodo frontal como en los nodos de cómputo. Si el nombre de máquina es cambiado después que el nodo frontal esté instalado, varios servicios del clúster no serán capaces de encontrar la máquina del nodo frontal. Alguno de estos servicios incluye: SGE, NFS, AutoFS y Apache.

Se llena la forma, entonces se hace click en el botón ’Next’.

9. La pantalla de configuración de la red privada del cluster permite definir los parámetros de conectividad para la red ethernet que conecta el nodo frontal con los nodos de cómputo.

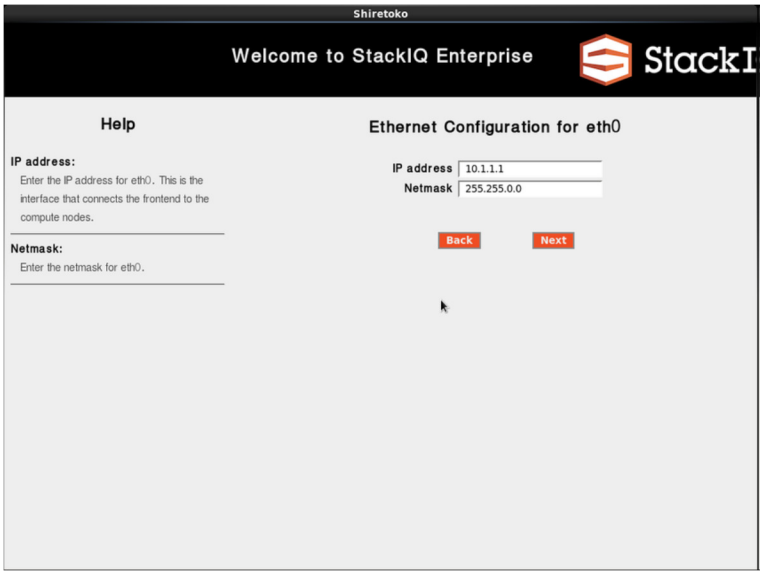

Figura 36 - Instalación StackIQ, configuración red eth0. Fuente: StackIQ.com

Nota: Se recomienda aceptar los valores predeterminados (hacienda click en el botón 'Next'). Pero para aquellas únicas circunstancias que requieren diferentes valores para la conexión interna ethernet, se dejan expuestos los parámetros de configuración de red para su modificación.

10. La pantalla de configuración de red pública del cluster permite definer los parámetros de conectividad de red ethernet que conecta el nodo frontal con el mundo exterior (p.ej., internet).

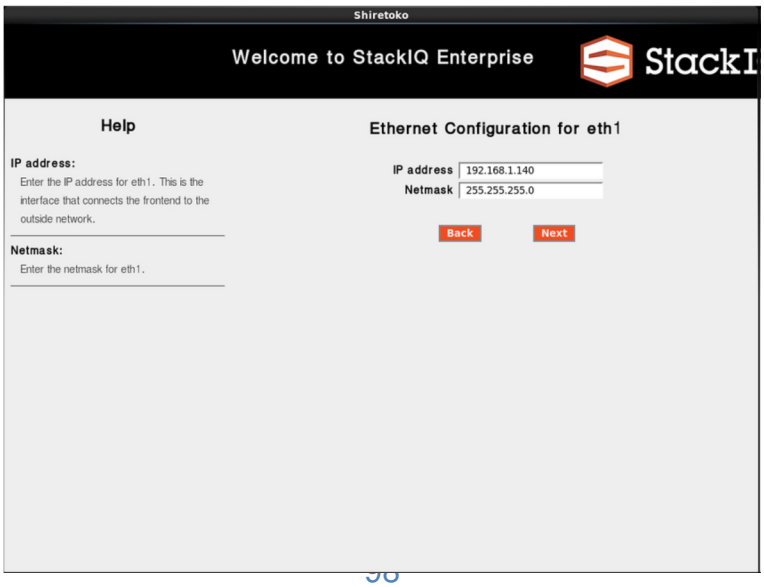

Figura 37 - Instalación StackIQ, configuración red eth1. Fuente: StackIQ.com

La figura anterior es un ejemplo de como se configura la red externa en uno de los nodos frontales.

11. Configure las entradas de Gateway y DNS:

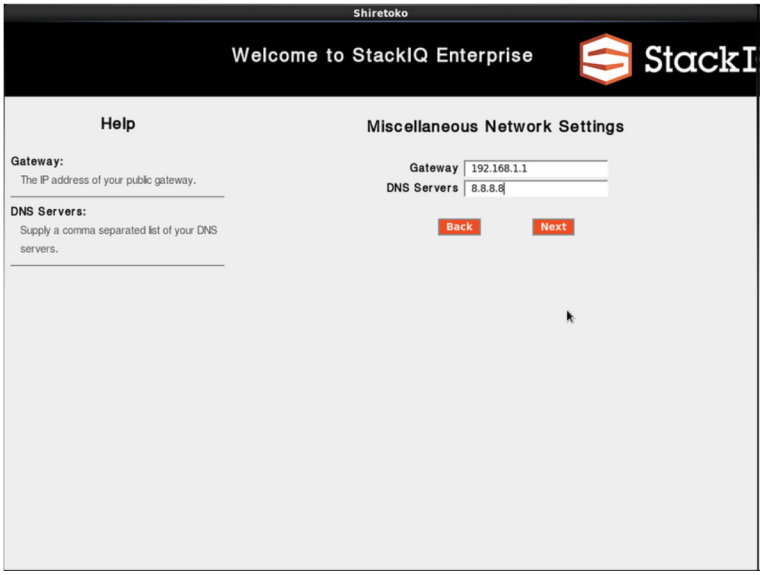

Figura 38 - Instalación StackIQ, configuración DNS. Fuente: StackIQ.com

12. Ingrese la contraseña para el root:

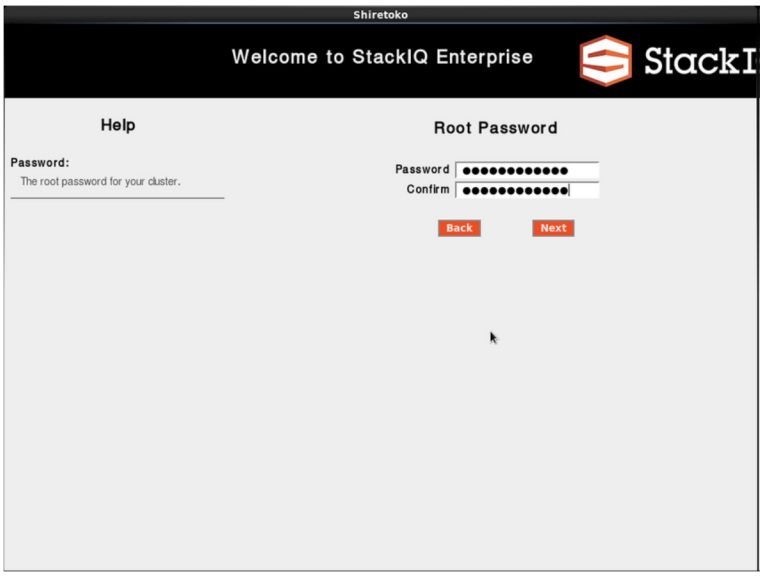

Figura 39 - Instalación StackIQ, contraseña de Root. Fuente: StackIQ.com

13. Configure la zona horaria y el servidor de tiempo:

Figura 40 - Instalación StackIQ, configuración de tiempo. Fuente: StackIQ.com

14. La pantalla de particiones del disco permite seleccionar particionado automático o manual.

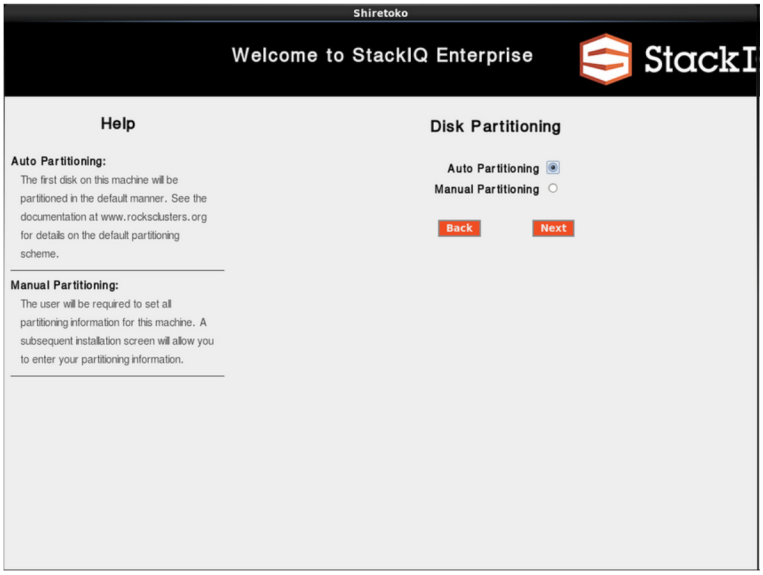

Figura 41 - Instalación StackIQ, Particionamiento. Fuente: StackIQ.com

Para seleccionar particionado automático, se hace click en el botón radial de Auto Particionado. Esto reparticiona y reformatea el primer disco duro descubierto que está conectado al nodo frontal. Todos los otros dispositivos conectados al nodo frontal se dejaran intactos.

La primera unidad descubierta será particionada como sigue:

Tabla 2 - Frontend -- Default Root Disk Partition

Nombre Partición	**Tamaño**
/	16 GB
/var	16 GB
swap	1 GB
/export (enlace simbólico a /state/partition1)	Remanente del disco raiz

Cuando se usa particionado automático, el instalador reparticiona y reformatea el primer disco duro que descubre el instalador. Todos los datos de la unidad serán borrados. Todas las otras unidades permanecen sin cambio.

El proceso de descubrimiento de las unidades de disco utiliza la salida de “cat /proc/partitions” para obtenerla lista de unidades.

15. Si se selecciona particionado manual, entonces se presenta la pantalla de paricionador manual de Red Hat:

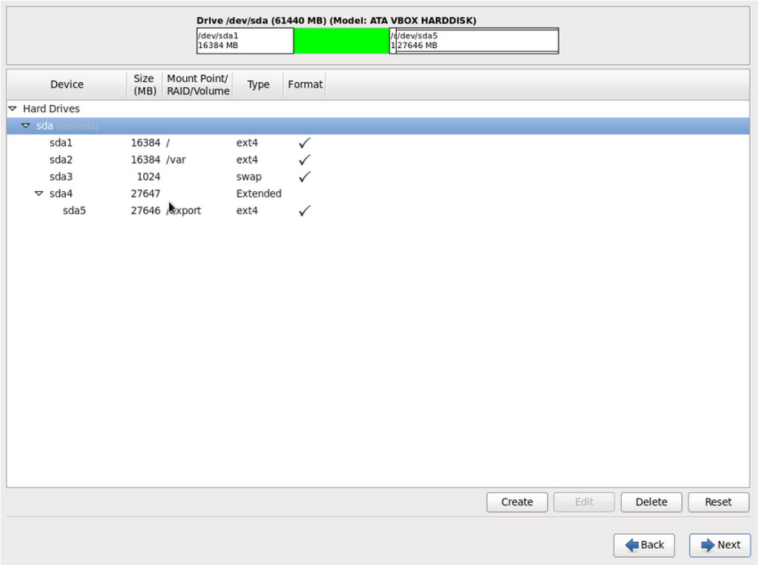

Figura 42 - Instalación StackIQ, particiones manuales. Fuente: StackIQ.com

El ejemplo anterior muestra la creación de particiones ’/’, ’/var’, swap y ’/export’.

Si se selecciona particionado manual, se debe especificar por lo menos 16 GB para la partición raiz y se debe crear una partición separada /export.

LVM no es soportado por StackIQ Enterprise.

Cuando se finaliza la definición de las particiones, se hace click en el botón ’Next’.

16. El nodo frontal formateará su sistema de archivos y luego solicitará cada uno de los DVD(s) que se adicionaron al comienzo de la instalación del nodo.

Figura 43 - Instalación StackIQ, inicio copia. Fuente: StackIQ.com

En el ejemplo arriba, se debe insertar el DVD del Rol Cluster-Core y presionar ’OK’.

El contenido del DVD será copiado al disco duro del nodo frontal.

Este paso se repite oara cada rol provisto en los pasos 3 a 5.

Nota: Después que todos los roles son copiados, no se requiere más interacción del usuario.

17. Después de que todos los DVD de los roles son copiados, se instalarán los paquetes:

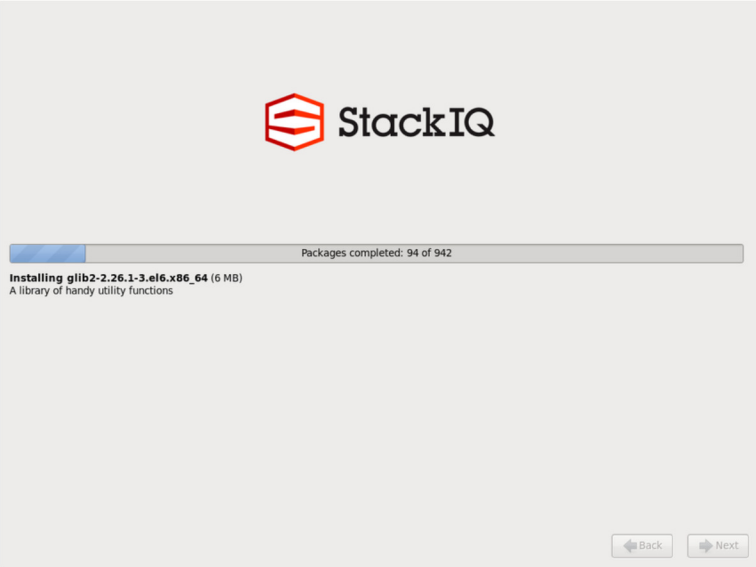

Figura 44 - Instalación StackIQ, finalización instalación. Fuente: StackIQ.com

18. Finalmente, el cargador de boot será instalado y los scripts de post configuración se ejecutarán en segundo plano. Cuando se completen los scripts el nodo frontal será reiniciado, quedando operativo.

7.3.2 INSTALACIÓN DE LOS NODOS DE CÓMPUTO

Los nodos de cómputo o nodos de segundo plano pueden ser instalados en un nodo de gestion de StackIQ Enterprise con uno de los siguientes métodos:

- Instalación de los nodos de cómputo usando Discovery
- Instalación de los nodos de cómputo usando CSV
- Instalación de los nodos de cómputo usando Insert-Ethers

7.3.2.1 Instalación de los nodos de cómputo usando Discovery Process

StackIQ soporta la adición e instalación de nodos de cómputo usando el proceso de descubrimiento (Discovery Process). Esto permite al administrador adicionar nodos al clúster sin conocer ningún detalle de los nodos mismos.

De manera predefinida, StackIQ deshabilita el uso del modo de descubrimiento para detectar e instalar nodos de computo. Al habilitar el modo de descubrimiento,

se permite entonces que el nodo frontal detecte y responda a cualquier petición DHCP que venga por la red privada del nodo frontal.

En grandes centros de datos empresariales, esta funcionalidad de responder a peticiones DHCP puede estar contra las políticas corporativas. Por esta razón, el software StackIQ Enterprise explícitamente requiere que el administrador habilite el soporte para el "Discovery Mode".

Para habilitar el soporte al "Discovery Mode", se debe ingresar al nodo frontal como usuario root y ejecutar el comando:

rocks set attr discover_start true

Una vez que el "Discovery Mode" es habilitado, los siguientes pasos deben seguirse para descubrir nuevos nodos.

1. Usando un navegador web, se accede a la intefaz gráfica de usuario web del nodo frontal:

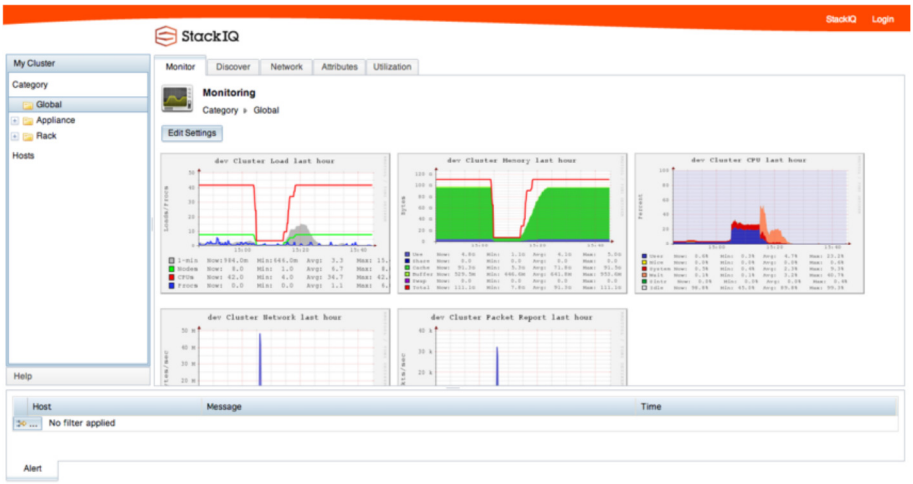

Figura 45 - Interfaz Web de nodo frontal. Fuente: StackIQ.com

2. Se hace click en el enlace de "Login", en la esquina superior derecha de la pantalla y se ingresa el nombre de usuario root y la contraseña de root para ingresar.

3. Después de haber ingresado, se hace click en la sección "Discover" para ver la siguiente pantalla.

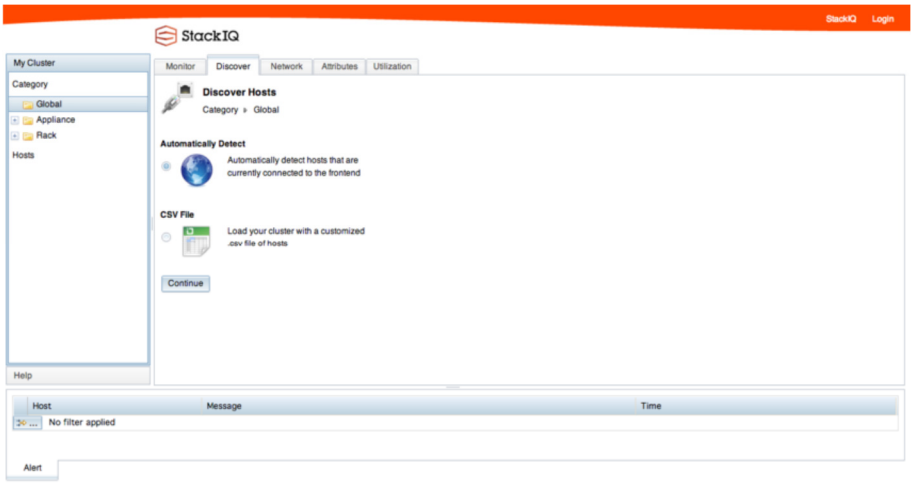

Figura 46 - Ingreso al TAB Discover. Fuente: StackIQ.com

Para ejecutar el descubrimiento, se deja el botón radial "Automatically Detect" seleccionado y se hace click en "Continue".

4. Una vez se está en el modo de descubrimiento, se hace click en el botón "Start" para ejecutar el proceso de descubrimiento.

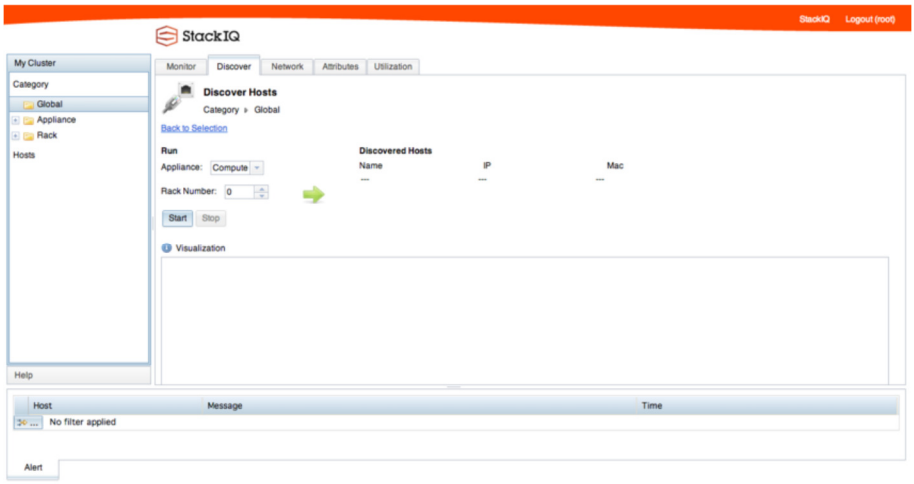

Figura 47 - Inicio de Discovery con botón Start. Fuente: StackIQ.com

5. Una vez que el proceso de descubrimiento ha iniciado, se deben encender los nodos de cómputo en el orden que se quiere que sean instalados. Una vez que los nodos comienzan a ser descubiertos, deben aparecer en la pantalla como se muestra a continuación.

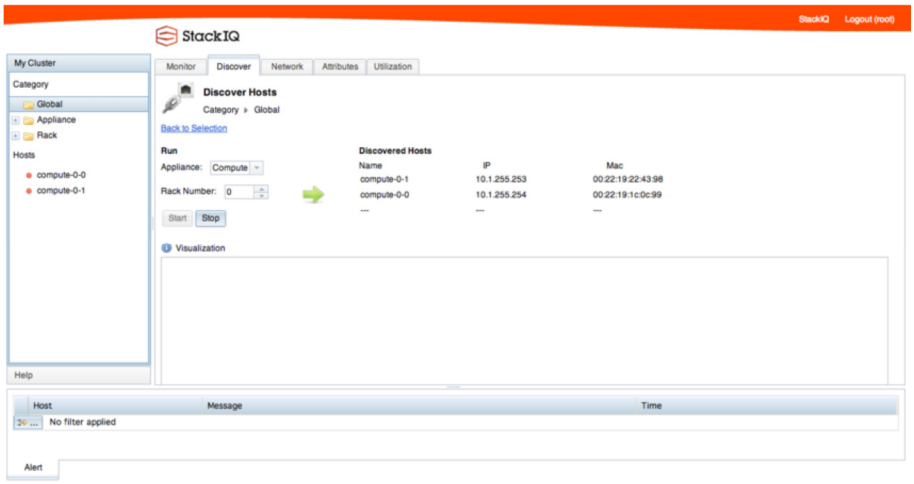

Figura 48 - Registro de nodos con Discovery. Fuente: StackIQ.com

6. Una vez que la instalación de los nodos de cómputo está en ejecución, se debe poder ver el proceso de transferencia de paquetes, como se muestra a continuación.

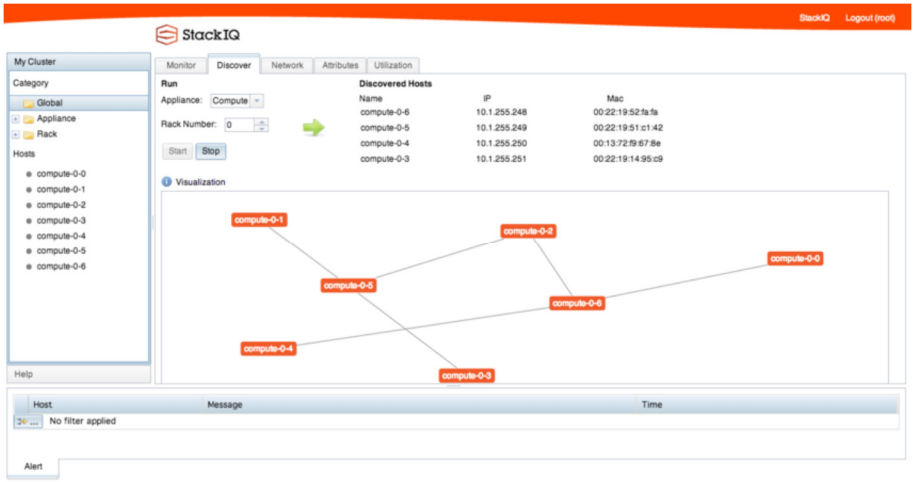

Figura 49 - Transferencia de paquetes. Fuente: StackIQ.com

7. La parte inferior de la pantalla contiene una ventana de "Alertas" que le indica al administrador acerca de los eventos en el sistema. Cuando un nodo de cómputo finaliza la instalación, y está disponible para su uso, la ventana de "Alertas" muestra un mensaje "Kickstart Completed" como se muestra a continuación.

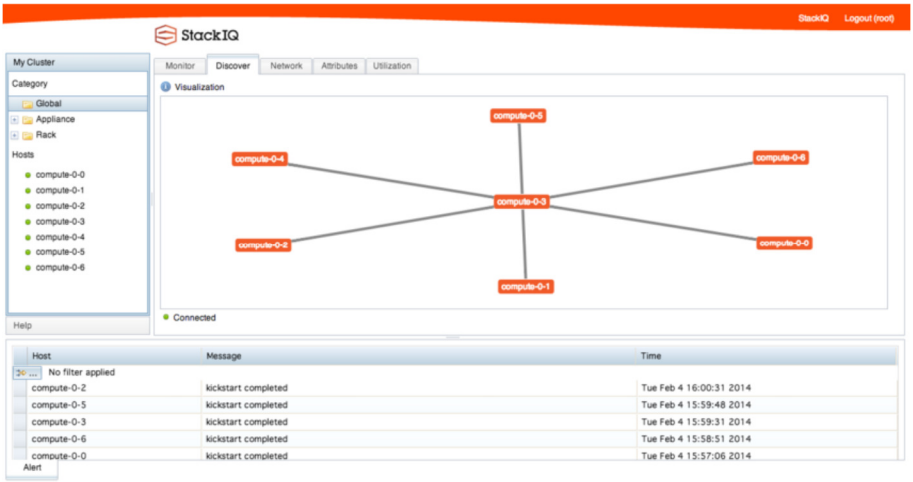

Figura 50 - Mensaje de KickStart Completed. Fuente: StackIQ.com

En el lado izquierdo de la página web está un listado de los nodos de cómputo en el clúster y su estado.

- Verde: Indica que el nodo está encendido y en funcionamiento.
- Gris: Indica que el nodo se está instalando.
- Rojo: Indica que el nodo está ya sea apagado o no se ha iniciado su instalación aún.

7.3.2.2 Instalación de los nodos de cómputo usando archivos CSV

Otra característica del software StackIQ Enterprise es la capacidad de adicionar nodos de cómputo al sistema usando archivos CSV (Comma Separated Value).

La ventaja de usar archivos CSV, es que da un control más detallado sobre la configuración del clúster. El archivo CSV de equipos necesita tener el siguiente formato:

Tabla 3 - Archivo CSV de equipos

NOMBRE	APPLIANCE	RACK	RANK	IP	MAC	INTERFAZ	SUBRED
compute-0-0	compute	0	0	10.1.255.250	40:22:19:1c:0c:99	eth0	private
compute-0-1	compute	0	1	10.1.255.250	30:22:19:22:43:98	eth0	private
compute-0-2	compute	0	2	10.1.255.250	20:22:19:54:b7:55	eth0	private
compute-0-3	compute	0	3	10.1.255.250	10:22:19:14:95:c9	eth0	private
compute-0-4	compute	0	4	10.1.255.250	0:13:72:f9:67:8e	eth0	private
compute-0-5	compute	0	5	10.1.255.250	90:22:19:51:c1:42	eth0	private
compute-0-6	compute	0	6	10.1.255.250	80:22:19:52:fa:fa	eth0	private

Una vezque el archivo CSV de equipos es creado, este puede ser adicionado al nodo de gestión de StackIQ en una de las siguientes formas.

- Adicionar el archivo CSV usando interfaz gráfica web
- Adicionar el archivo CSV usando la interfaz de línea de comando (CLI)

Adicionar el archivo CSV usando interfaz gráfica web

Para adicionar el archivo CSV usando la interfaz gráfica web -

1. Usando un navegador web, se accede a la intefaz gráfica de usuario web del nodo frontal.

2. Se ingresa a la interfaz web usando el nombre de usuario root y la contraseña en el nodo frontal.

3. Se navega a la sección "Discover".

4. Se selecciona el botón radial junto a "CSV File" y se hace click en "Continue.

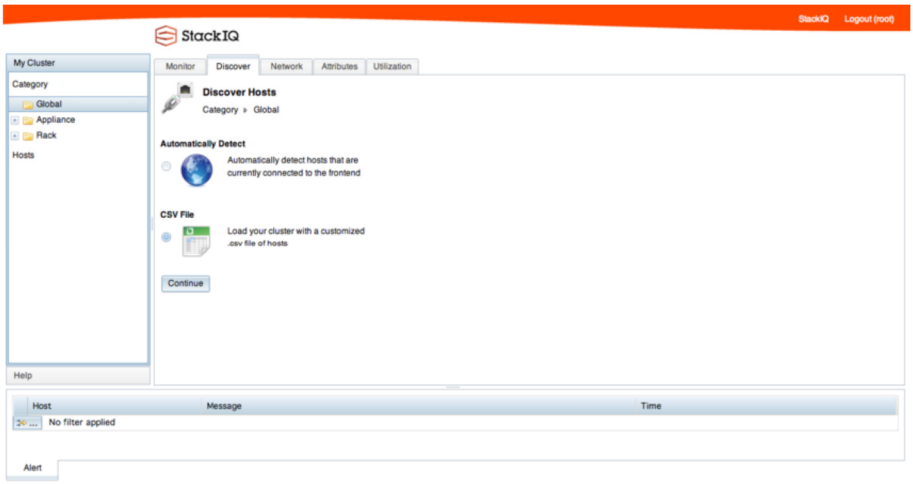

Figura 51 - Uso deCSV en interfaz WEB. Fuente: StackIQ.com

5. A continuación, se hace click en "Browse CSV File". Esto presenta una ventana para poder escoger el archivo CSV desde la máquina en la que se está ejecutando el navegador. Entonces se hace click en "Upload"

6. Después de cargar el archivo CSV, la interfaz muestra el contenido del archivo

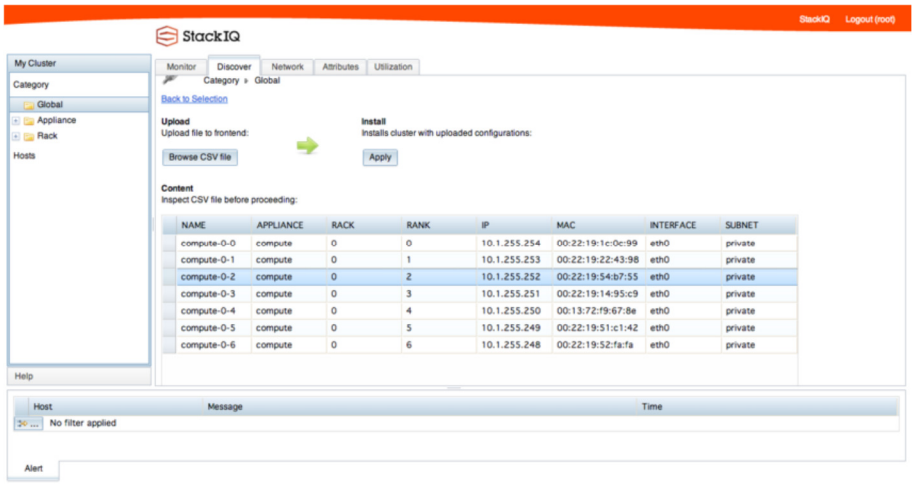

Figura 52 - Contenido archivo CSV cargado. Fuente: StackIQ.com

7. Se hace click en "Apply" para aplicar la configuración del archivo CSV al nodo frontal. Se debe entonces ver el mensajeen la interfaz gráfica de web que indicará si el proceso fue exitoso o no.

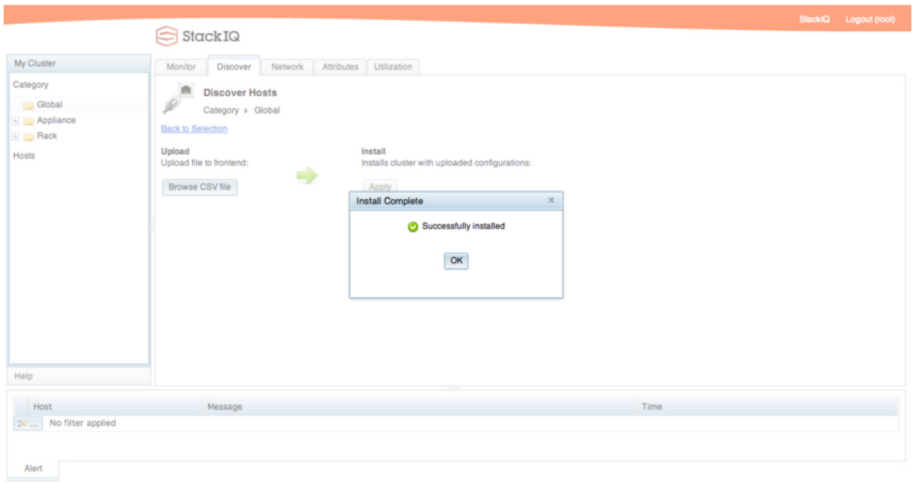

Figura 53 - Fin de proceso con archivo CSV. Fuente: StackIQ.com

Adicionar el archivo CSV usando la interfaz de línea de comandos (CLI)

Para adicionar el archivo CSV al nodo frontal de StackIQ usando la interfaz de linea de comando-

1. Se copia el archivo CSV al nodo de gestión de StackIQ.

2. Se ejecuta el comando:

rocks load hostfile file=hostfile.csv

7.3.2.3 Instalación de los nodos de cómputo usando Insert-Ethers

1. Se ingresa al nodo frontal como el usuario root.

2. Se invoca la ejecución del programa Insert-Ethers, que captura las solicitudes DHCP de los nodos de cómputo e ingresa su información en una base de datos soportada por MySQL en el nodo de gestión de StackIQ Management:

insert-ethers

Este programa presenta una pantalla como la siguiente:

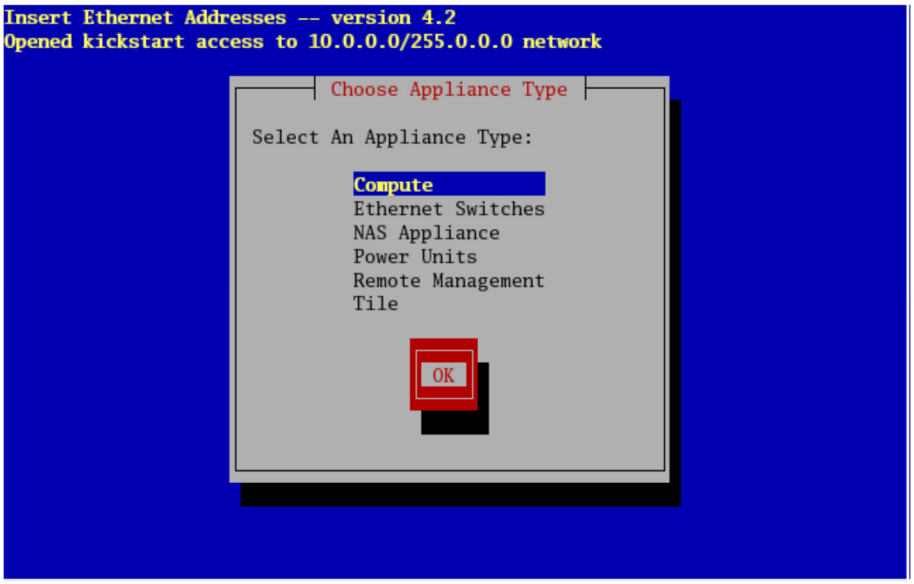

Figura 54 - Pantalla de ingreso a Insert-Ethers. Fuente: StackIQ.com

Si los nodos frontal y de cómputo están conectados via un switch ethernet gestionado, se deberá seleccionar 'Ethernet Switches' del listado presentado. Esto es debido a que el comportamiento de muchos switches gestionados es realizar la petición DHCP, con el objeto de recibir una dirección IP que el cliente pueda usar para configurar y monitorear el switch.

Cuando insert-ethers captura la solicitud DHCP para el switch gestionado, lo configurará como un switch ethernet y almacenará la información en la base de datos MySQL del nodo frontal.

Se toma la selección predefinida, Compute, y si presiona 'Ok'.

3. Entonces se verá:

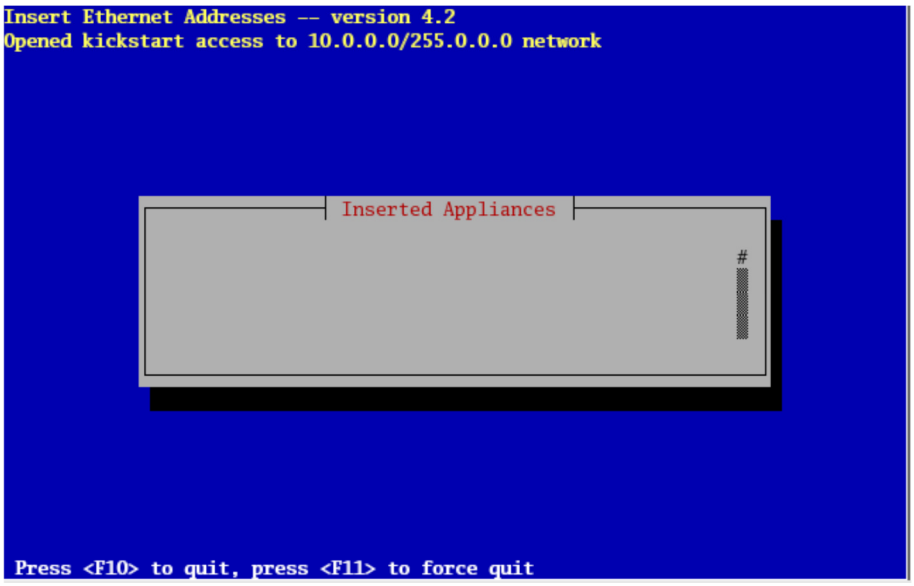

Figura 55 - Entrada opción Compute. Fuente: StackIQ.com

Esto indica que insert-ethers está esperando por los nodos de cómputo.

4. Se enciende el primer nodo de cómputo.

5. Cuando el nodo frontal recibe la petición DHCP del nodo de cómputo, se verá algo similar a:

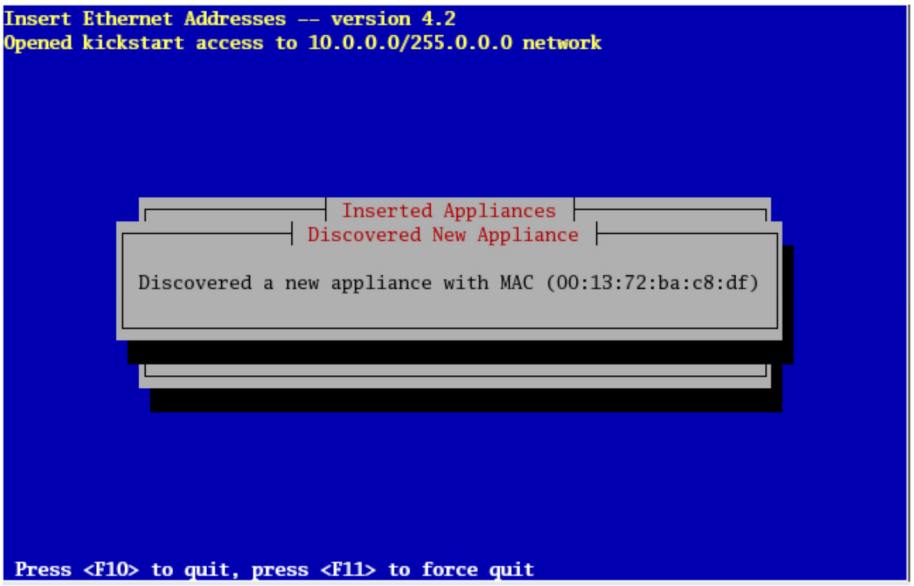

Figura 56 - Insert-ethers, solicitud DHCP. Fuente: StackIQ.com

Esto indica que insert-ethers recibió la solicitud DHCP delnodo de cómputo, lo insertó en la base de datos y actualizó todos los archivos de configuración (p.ej., /etc/hosts, /etc/dhcpd.conf y DNS).

La pantalla antes mostrada será presentada por unos segundos y entonces se presentará lo siguiente:

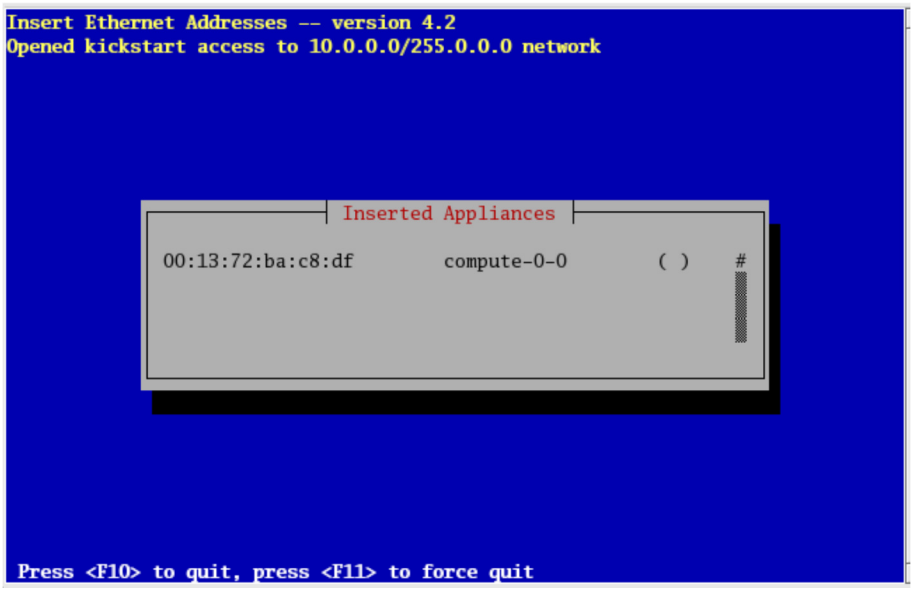

Figura 57 - Insert-ethers, nodo de computo registrado. Fuente: StackIQ.com

En la figura 57, insert-ethers ha descubierto un nodo de cómputo. La marca "( )" junto al compute-0-0 indica que el nodo no ha solicitado el archivo de inicio aún. Se presentará este tipo de salida para cada nodo de cómputo que sea identificado exitosamente por insert-ethers.

Luego de unos segundos deberá verse:

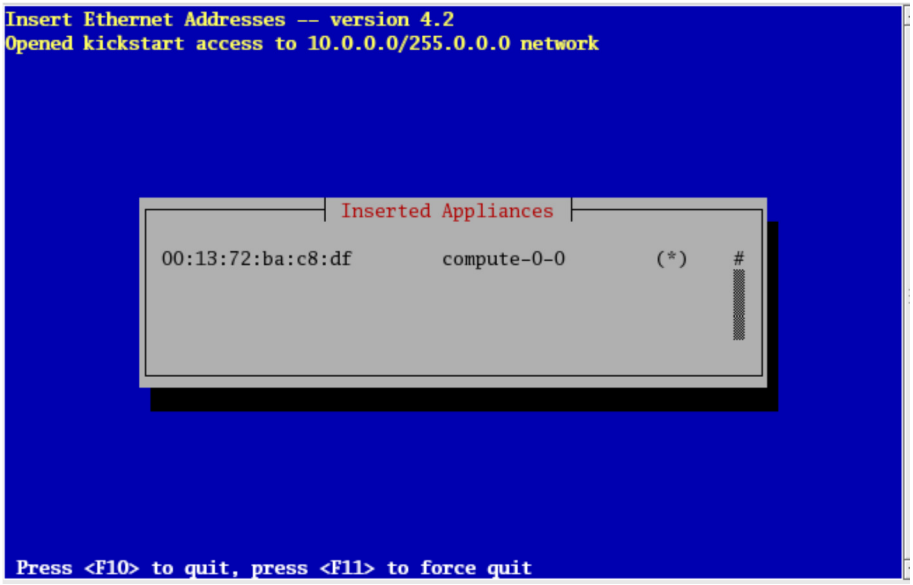

Figura 58 - Insert-ethers registró exitosamente el nodo de cómputo. Fuente: StackIQ.com

En esta figura se muestra que el nodo de cómputo ha solicitado exitosamente en alchivo de inicio (kickstart) al nodo frontal. Si no hay más nodos de cómputo se puede finalizar la ejecución de insert-ethers.

Los archivos kickstart son recuperados via HTTPS. Si hay un error durante la transmisión, el código de error será visualizado en lugar de "*".

6. En este punto, se puede monitorear la instalación usando rocks-console. Solo se debe ejecutar el comando:

rocks-console compute-0-0

7. Después de haber instalado todos los nodos de cómputo, se puede terminar la ejecución de insert-ethers presionando la tecla 'F8'.

8. Después de haber instalado todos los nodos de cómputo en el primer gabinete y si se desea instalar nodos de cómputo en el siguiente gabinete, se debe iniciar insert-ethers así:

insert-ethers --cabinet=1

Esto hará que el monbre de todos los nodos de computo figuren como compute-1-0, compute-1-1, ...

7.4 RESULTADO ESPERADO DE LA IMPLEMENTACIÓN DE STACKIQ ENTERPRISE DATA

StackIQ Enterprise Data es una solución completa e integrada de Hadoop para los clientes empresariales. Este incluye la plataforma de datos Hortonworks que está compuesta por HDFS, MapReduce, Pig, Hive, HBase, y Zookeeper, junto con tecnologías de código abierto que conforman la plataforma Hadoop más manejable, abierta y extensible. Estos incluyen HCatalog, un servicio de gestión de metadatos para simplificar el intercambio de datos entre Hadoop y otros sistemas de información de la empresa, y un conjunto completo de APIs abiertas como WebHDFS para que sea más fácil para los ISV a integrar y extender Hadoop.

La implementación de esta solución le permite a las organizaciones aproximarse a HPC y la Analítica con un conjunto de componentes como:

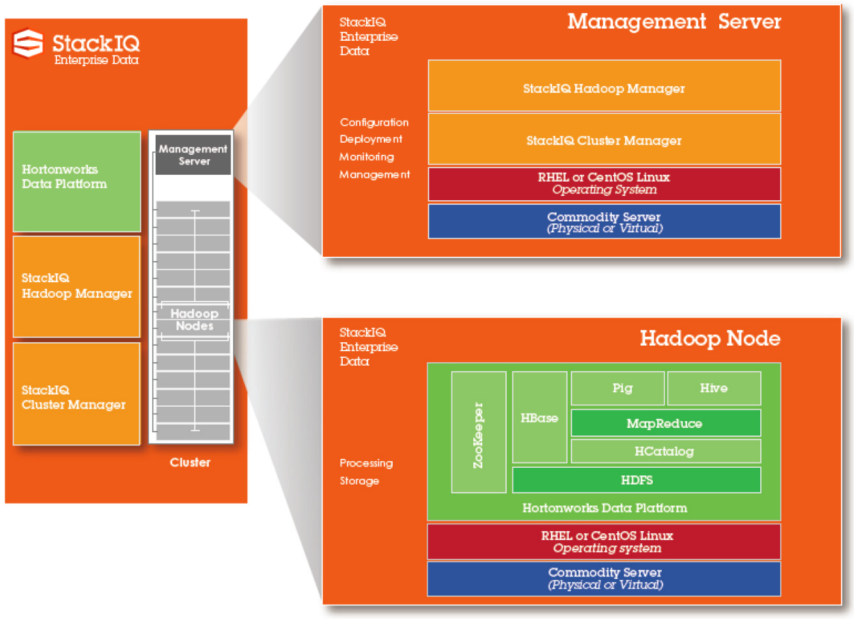

Figura 59 - Componentes de StackIQ Enterprise Data. Fuente: StackIQ.com

8. SUPER CLÚSTER CON RASPBERRY PI

8.1 ¿QUÉ ES RASPBERRY PI?

De acuerdo como raspberrypi.org presenta el Raspberry Pi, menciona que “es un ordenador de bajo costo, de tamaño de tarjeta de crédito que se conecta a un monitor de ordenador o un televisor, y utiliza un teclado y un ratón estándar. Es un dispositivo pequeño con capacidad que permite a las personas de todas las edades para explorar la computación, y aprender a programar en lenguajes como Scratch y Python. Es capaz de hacer todo lo que se espera que una computadora de escritorio haga, desde navegación por Internet y reproducción de vídeo de alta definición, crear hojas de cálculo, procesar textos, y jugar juegos”.

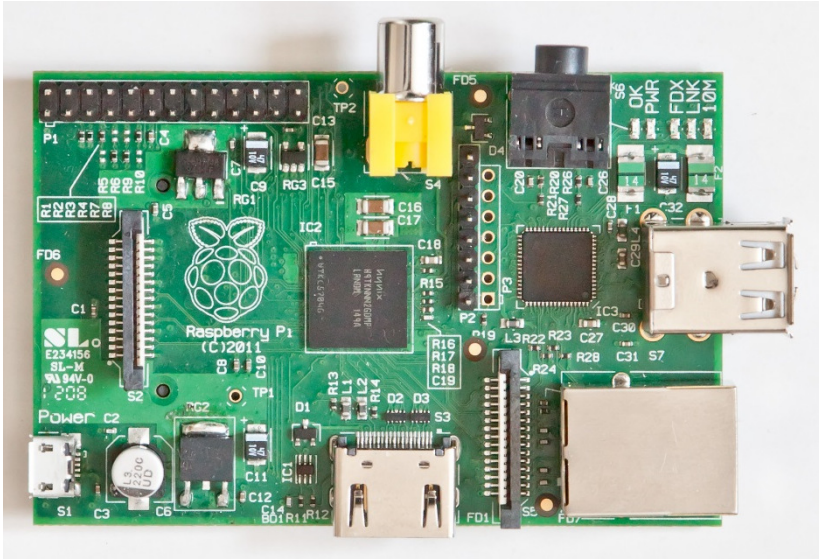

Figura 60 - Dispositivo básico Raspberry Pi. Fuente: raspberrypi.org

8.2 BONDADES DE RASPBERRY PI

Debido al pequeño tamaño y bajo costo del Raspberry Pi, se convierte en una buena alternativa para la construcción de un clúster en la nube, de Amazon o proveedores similares que pueden ser costosos, o usando PCs de escritorio.

El Raspberry Pi viene con un puerto Ethernet incorporado, que permite conectarlo a un switch, router, o un dispositivo similar. Múltiples dispositivos de Raspberry Pi conectados a un switch entonces pueden formar un clúster; esta funcionalidad será la base de la configuración propuesta en este capítulo.

A diferencia de un ordenador portátil o PC, que puede contener más de una CPU,

el Raspberry Pi contiene sólo un único procesador ARM; Sin embargo, varios Raspberry Pi combinados proporcionan más CPU para trabajar.

Otro de los beneficios del Raspberry Pi es que también utiliza tarjetas SD como almacenamiento secundario, que pueden ser fácilmente copiados, lo que permite crear una imagen del sistema operativo del Raspberry Pi y luego clonarlo para su reutilización en varias máquinas.

Desde la perspectiva del software de escritura, el Raspberry Pi puede ejecutar varias versiones del sistema operativo Linux, así como otros sistemas operativos, como FreeBSD y el software y utilidades relacionados con el desarrollo de la misma. Esto permite implementar los tipos de tecnología encontrados en clústeres Beowulf y otros sistemas paralelos.

Esta característica de los sistemas RaspBerry Pi que las hacen sistemas viables para el estudio de la programación y tecnologías de clúster, finalmente propicio para que entre 2012 y 2014 se publicaran diversos estudios, artículos y hasta textos sobre la construcción de sistemas de computación de alto desempeño utilizando agrupaciones de estos pequeños computadores.

8.3 COMPUTACIÓN PARALELA – MPI PARA RASPBERRY PI

Hay dos implementaciones de MPI prominentes que se pueden utilizar en la Raspberry Pi. Estos son: OpenMPI y MPICH.

OpenMPI es una implementación de código abierto de MPI mantenido por un grupo de socios industriales y académicos. Se ha implementado en varias de las TOP500 supercomputadoras del mundo, incluyendo el equipo japonés K.

Los orígenes de OpenMPI se pueden encontrar en varios otros proyectos, incluyendo el proyecto de la Universidad de Tennessee FT-MPI, LAM/MPI de la Universidad de Indiana, Universidad de Stuttgart PCX-MPI, y LA-MPI del Laboratorio Nacional de Los Álamos en el EE.UU.

Usted puede encontrar más información acerca de la tecnología en la web oficial:

http://www.open-mpi.org/

MPICH, que originalmente significaba para Message Passing Interface Chameleon, es una implementación del estándar MPI que soporta aplicaciones C, C ++ y Fortran. Fue desarrollado inicialmente a principios de los 90 para proporcionar retroalimentación sobre problemas de implementación al foro de MPI.

El MPICH Wiki proporcionar más antecedentes sobre la tecnología se puede encontrar en:

http://wiki.mpich.org/mpich/index.php/Frequently_Asked_Questions#General_Information

Cuando se trata de la velocidad de aplicación entre las dos bibliotecas hay algún debate. En última instancia, sin embargo, cómo se optimice el programa y el hardware que lo ejecuta hará una gran diferencia.

A diferencia de la OpenMPI más nuevo, MPICH tiene más tiempo en uso y es extremadamente portable entre sistemas. También hay amplias opciones de soporte y documentación disponible en línea.

Debido a que MPICH está en uso por largo tiempo también se pueden encontrar más fácilmente los binarios de aplicaciones que trabajan con MPICH (si el código fuente no está disponible) y no se incurren en tantos problemas de compatibilidad.

Por esta razón se recomienda el uso de MPICH para la construcción de aplicaciones para un supercomputador con Raspberry Pi.

9. CONCLUSIONES

En este documento de trabajo de grado se ha podido desarrollar un modelo viable para la academia y las organizaciones que integra la computación de alto desempeño con la analítica de datos.

Los resultados de la investigación demuestran que existen alternativas serias para las organizaciones que desean avanzar en sus soluciones analíticas adoptando tecnologías estándares y maduras que cuentan con el respaldo comercial necesario para las necesidades operativas de las organizaciones.

Igualmente, la academia puede encontrar en este trabajo las diferentes herramientas para el estudio de la analítica y la computación de alto desempeño y la formación de nuevos profesionales con conocimientos en estas áreas adquiridos de primera mano con la experimentación y desarrollo de sistemas en sus laboratorios.

Se ha mostrado como estas dos áreas de conocimiento convergen actualmente y como el mercado y los especialistas han entendido que lejos de ser dos soluciones apoyadas en la computación intensiva que llevaban caminos separados y aproximaciones al uso de recursos incompatibles, la computación de alto desempeño y la analítica de datos tienen en su haber años de experiencia y conocimiento que beneficia su convergencia.

El modelo que se ha desarrollado en este trabajo se ha servido de ello. Ha tomado de la historia de la computación de alto desempeño los mecanismos desarrollados para hacer frente a los problemas primarios de espacio, consumo de energía, manejo de la concentración del calor y, finalmente, los costos de construcción y mantenimiento de estas soluciones. Por otra parte, ha tomado de la evaluación de las soluciones analíticas disponibles el imperativo de la homogeneidad de los componentes y la repetitividad de su construcción.

Finalmente, con la inclusión de la plataforma ARM ha se encontrado un espacio propicio para la creatividad que debe surgir de la academia. Esta no tan novicia plataforma permite considerar ahora la factibilidad económica de soluciones de computación paralela e intensiva en las manos de los estudiantes de universidad por un costo similar al de un teclado de un computador.

El hallazgo de una solución como StackIQ permite vislumbrar la relevancia de los temas abordados en este trabajo y la seriedad con la que también la industria lo está abordando. Es un buen momento, como se mencionaba al formular el problema del estudio, para que la academia salga a la vanguardia de la necesidad

del mercado y presente a sus profesionales formados con las capacidades que se demandan para la organizaciones del siglo 21.

BIBLIOGRAFIA

1. JAMES REINDERS; JAMES JEFFERS; High Performance Parallelism Pearls - Morgan Kaufmann - November 4, 2014 - Print ISBN-13: 978-0-12-802118-7

2. DENNIS ABTS; JOHN KIM; High Performance Datacenter Networks - Morgan & Claypool Publishers - February 2, 2011 - Print ISBN-13: 978-1-60845-402-0

3. EMMANUEL UDOH; Cloud, Grid and High Performance Computing - IGI Global - June 30, 2011 - Print Isbn-10: 1-60960-603-5

4. EMMANUEL UDOH; Applications and Developments in Grid, Cloud, and High Performance Computing - IGI Global - September 30, 2012 - Print Isbn-10: 1-4666-2065-X

5. JEFFREY VETTER; Contemporary High Performance Computing: From Petascale toward Exascale - Chapman and Hall/CRC - April 23, 2013 - Print ISBN-13: 978-1-4665-6834-1

6. MARIJANA DESPOTOVIĆ-ZRAKIĆ; VELJKO MILUTINOVIĆ; ALEKSANDAR BELIĆ; Handbook of Research on High Performance and Cloud Computing in Scientific Research and Education - IGI Global - March 31, 2014 - Print Isbn-10: 1-4666-5784-7

7. PETHURU RAJ; GANESH DEKA; Handbook of Research on Cloud Infrastructures for Big Data Analytics - IGI Global - March 31, 2014 - Print Isbn-10: 1-4666-5864-9

8. NAUMAN SHEIKH; Implementing Analytics - Morgan Kaufmann - May 6, 2013 - Print ISBN-13: 978-0-12-401696-5

9. ENDA RIDGE; Guerrilla Analytics - Morgan Kaufmann - September 26, 2014 - Print ISBN-13: 978-0-12-800218-6

10. JACK DONGARRA; On the Future of High Performance Computing: How to Think for Peta and Exascale Computing – The Scientific Computing and Imaging Institute at the University of Utah – February 12, 2012

11. WIKIPEDIA; Supercomputer; http://en.wikipedia.org/wiki/Supercomputer

12. INSIDE HPC; What is high performance computing; http://insidehpc.com/hpc-basic-training/what-is-hpc/

13. WIKIPEDIA; History of Supercomputing; http://en.wikipedia.org/wiki/History_of_supercomputing

14. JOHN MARKOFF; The Attack of 'Killer Micros' – The New York Times - May 6, 1991 - http://www.nytimes.com/1991/05/06/business/the-attack-of-the-killer-micros.html

15. WIKIPEDIA; Blue Gene; http://en.wikipedia.org/wiki/Blue_Gene

16. WIKIPEDIA; Supercomputer architecture; http://en.wikipedia.org/wiki/Supercomputer_architecture

17. GIL PRESS; A very short history of Data Science – May 28 2013 - http://www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history-of-data-science/

18. PIYANKA JAIN; Data Science or Analytics - February 25 2013 - http://www.forbes.com/sites/piyankajain/2013/02/25/data-science-or-analytics/

19. DR. JERRY A. SMITH; Data Analytics vs Data Science: Two Separate, but Interconnected Disciplines – September 9 2013 - http://datascientistinsights.com/2013/09/09/data-analytics-vs-data-science-two-separate-but-interconnected-disciplines/

20. RAHUL NAWAB; History of Data Analytics - August 21 2012 - http://datascientistinsights.com/2013/09/09/data-analytics-vs-data-science-two-separate-but-interconnected-disciplines/

21. WIKIPEDIA; Data Analysis - http://en.wikipedia.org/wiki/Data_analysis

22. WIKIBOOKS; Data Science: An Introduction / A History of Data Science - http://en.wikibooks.org/wiki/Data_Science:_An_Introduction/A_History_of_Data_Science

23. TIM MATTSON; The OSCAR Solution Stack for Cluster Computing – Intel Corp.

24. CHARLES SEVERANCE; High Performace Computing – Connexions Rice University – October 29 2012

25. DOUGLAS EADLINE, PHD; High Performance Computing For Dummies®, Sun and AMD Special Edition - Wiley Publishing, Inc. – 2009 - ISBN: 978-0-470-49008-2