

**ANÁLISIS Y DISEÑO DE UNA SOLUCIÓN COMPUTACIONAL EN FORMA DE
RACIMO O CLUSTER APTA PARA INTEGRARSE A GRIDS O MALLAS
COMPUTACIONALES**

SAMIR AUGUSTO ARIAS GARTNER

**UNIVERSIDAD CATÓLICA POPULAR DEL RISARALDA
PROGRAMA DE INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES
PROYECTO DE GRADO
PEREIRA
2009**

**ANÁLISIS Y DISEÑO DE UNA SOLUCIÓN COMPUTACIONAL EN FORMA DE
RACIMO O CLUSTER APTA PARA INTEGRARSE A GRIDS O MALLAS
COMPUTACIONALES**

SAMIR AUGUSTO ARIAS GARTNER

Asesor

**LUÍS ALEJANDRO FLETSCHER BOCANEGRA
INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES
DE LA UNIVERSIDAD DEL CAUCA
ESPECIALISTA EN GERENCIA DE PROYECTOS DE TELECOMUNICACIONES
DE LA UNIVERSIDAD DEL ROSARIO**

**UNIVERSIDAD CATÓLICA POPULAR DEL RISARALDA
PROGRAMA DE INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES
PROYECTO DE GRADO
PEREIRA
2009**

DECLARACIÓN DE DERECHOS DE AUTOR

Yo, Samir Augusto Arias Gärtner , manifiesto mi voluntad de ceder a la Universidad Católica Popular del Risaralda (UCPR) los derechos patrimoniales, consagrados en el artículo 72 de la Ley 23 de 1982, del proyecto de grado de mi autoría titulado “ANÁLISIS Y DISEÑO DE UNA SOLUCIÓN COMPUTACIONAL DE CLUSTER APTA PARA INTEGRARSE A GRIDS O MALLAS COMPUTACIONALES” el cual fue producto de mi investigación y es requisito para optar por el título de Ingeniero de Sistemas y Telecomunicaciones de la Universidad Católica Popular del Risaralda. La UCPR, queda facultada para ejercer los derechos patrimoniales en sus actividades de investigación, docencia y publicación. La cesión de estos derechos se ajusta a lo establecido en la Ley 23 de 1982. En mi condición de autor conservo los derechos morales de la obra los cuales son irrenunciables de acuerdo a lo establecido con el artículo 30 de la Ley 23 de 1982. Suscribo este documento en el momento que haga entrega del trabajo final a la Biblioteca Cardenal Darío Castrillón Hoyos de la UCPR.

Samir Augusto Arias Gärtner 9.872.113

Pereira, a los 21 días del mes de Agosto del año 2009.

CLÁUSULA DE MARCAS REGISTRADAS

Las marcas, logotipos, nombres de compañías y líneas de productos mencionados a lo largo de este documento son propiedad de sus respectivos dueños los cuales no aprueban su uso en este documento, tampoco están afiliados con la Universidad Católica Popular del Risaralda, o con el autor de este documento como tampoco están involucrados en el desarrollo o patrocinio de este proyecto.

AGRADECIMIENTOS

Muchas personas, instituciones, corporaciones y situaciones me llevaron por el camino inevitable del desarrollo de este proyecto ya sea por haber inducido en mí la curiosidad o haber impuesto retos y obstáculos en mis procesos académicos y mi vida personal. A todos ellos muchas gracias.

A Dago Hernando Bedoya ex Decano de la Facultad de Ciencias Básicas e Ingeniería de la Universidad Católica Popular del Risaralda por haber creído en mí al haberme vinculado en una pequeña parte de su proyecto sobre redes académicas de alta velocidad, lo que me dio un vistazo general del concepto que se tiene sobre este tema en las distintas universidades del país y finalmente me llevó por el camino de la investigación sobre mallas computacionales en el semillero de investigación que sobre este tema se estructuró en la Universidad.

A Luis Alejandro Fletscher Bocanegra por sus clases de telecomunicaciones y su habilidad para despertar y mantener el interés, hacer las clases agradables, como también por haber tenido la disposición y el denuedo para ser asesor de mi proyecto.

A Alexander Quintero por enseñarme a no temerle a las exposiciones ni a las figuras de autoridad, como también por tratar a los alumnos sin condescendencia.

A Luis Eduardo Peláez Valencia Decano de la Facultad de Ciencias Básicas e Ingeniería de la Universidad Católica Popular del Risaralda, por haber dudado de mí y haberme retado académicamente una vez tras otra.

A los malos profesores que alguna vez pasaron por la Universidad y que probaron que esta generación de profesionales superara a la anterior.

A los buenos profesores que alguna vez pasaron por la Universidad y que fueron lo suficientemente rectos y estrictos como para exigirnos más de lo que podíamos dar pero que también fueron lo suficientemente compasivos para tendernos la mano, levantarnos y hacernos caer nuevamente.

A David “Dave” Ramírez por su sacrificio lúdico tecnológico.

A Luis Andrés Londoño por su motivación.

A la “venusina” por haberme demostrado que nada importa realmente.

A “pitico” por no haber tenido la paciencia para estar conmigo estos últimos meses.

A Dios en caso de que le sea relevante nuestra frágil y desamparada humanidad o sea tan vanidoso como para importarle un agradecimiento.

A mi madre y a mi padre por haber sido.

DEDICATORIA

A mi madre por su apoyo y confianza permanente.

TABLA DE CONTENIDO

DECLARACIÓN DE DERECHOS DE AUTOR.....III

CLÁUSULA DE MARCAS REGISTRADAS IV

AGRADECIMIENTOS..... V

DEDICATORIA VII

TABLA DE CONTENIDO..... 1

ÍNDICE DE FIGURAS 5

GLOSARIO 6

RESUMEN12

ABSTRACT13

1 INTRODUCCIÓN14

2 ANTECEDENTES17

3 SITUACIÓN PROBLEMA18

4 OBJETO DE ESTUDIO19

5 CAMPO DE ACCIÓN20

6	<u>OBJETIVO GENERAL</u>	<u>21</u>
7	<u>OBJETIVOS ESPECÍFICOS</u>	<u>22</u>
8	<u>DELIMITACIÓN DEL PROYECTO.....</u>	<u>23</u>
9	<u>APORTE PRÁCTICO Y TEÓRICO</u>	<u>24</u>
10	<u>FACTIBILIDAD TÉCNICA.....</u>	<u>25</u>
11	<u>FACTIBILIDAD ECONÓMICA</u>	<u>26</u>
12	<u>MÉTODO DE INVESTIGACIÓN.....</u>	<u>27</u>
13	<u>MARCO CONTEXTUAL.....</u>	<u>28</u>
13.1	MALLAS Y RACIMOS COMPUTACIONALES (CLUSTERS Y GRIDS)	28
13.1.1	HISTORIA DE LOS CLUSTERS Y LOS GRIDS DE COMPUTADORES.....	28
13.1.2	CARACTERÍSTICAS DE LAS MALLAS Y RACIMOS DE COMPUTADORES.....	31
13.2	MALLAS Y RACIMOS DE COMPONENTES EXÓTICOS.....	32
13.3	E-SCIENCE LA NUEVA FORMA DE HACER CIENCIA	33
13.4	LA INFLUENCIA DE LA INDUSTRIA DEL ENTRETENIMIENTO EN LA COMPUTACIÓN Y EL PROCESAMIENTO	35
13.5	EL LEVANTAMIENTO DEL SOFTWARE.....	37
13.6	LA EVOLUCIÓN EN EL HARDWARE DE LAS CONSOLAS DE VIDEOJUEGOS	38
13.7	LA CRISIS DEL PROCESAMIENTO, LA LEY DE MOORE TAMBALEA	40
13.8	EL PORQUE DEL PROCESADOR CELL BEA.....	44
14	<u>MARCO TEÓRICO.....</u>	<u>48</u>
14.1	DIFERENCIAS ENTRE RISC/POWERPC Y CISC/X86	48
14.2	LA COMPLEJIDAD EN EL MICROCHIP RISC VS. CISC	49
14.3	TIPOS DE MULTIPROCESAMIENTO	52
14.3.1	MULTIPROCESAMIENTO SIMÉTRICO O SMP.....	53

14.3.2	MULTIPROCESAMIENTO ASIMÉTRICO O ASMP.....	55
14.4	TIPOS DE PARALELISMO SIMD Y MIMD.....	57
14.5	PROCESADOR CELL BEA.....	59
14.5.1	ARQUITECTURA.....	59
14.5.2	MODELOS DE PROGRAMACIÓN DEL CELL BEA.....	62
14.6	MALLAS Y RACIMOS DE COMPUTADORES (CLUSTERS Y GRIDS)	64
14.6.1	ADMINISTRADORES DE CARGA Y RECURSOS.....	67
14.7	COMPONENTES DE UN RACIMO DE COMPUTADORES SIMPLE.....	70
14.7.1	AUTENTICACIÓN Y SESIÓN REMOTA POR SSH.	70
14.7.2	INTERFACES DE PROGRAMACIÓN DE APLICACIONES MPI Y PVM.....	71
14.7.3	OBJETOS DISTRIBUIDOS, CORBA, COM, PDO.	72
14.7.4	COMPARTIR ARCHIVOS MEDIANTE NFS.	73
14.7.5	SISTEMAS OPERATIVOS.	74
14.7.6	RED Y SEGURIDAD.	78
14.8	MALLAS COMPUTACIONALES O GRIDS	79
14.8.1	GLOBUS TOOLKIT.....	81
14.8.2	ESTÁNDARES DE LA OGF.....	82
14.8.3	ARQUITECTURA DE UNA MALLA COMPUTACIONAL Y SUS COMPONENTES.....	87
15	<u>MODELO TEÓRICO.....</u>	<u>94</u>
15.1	DESPLIEGUE DE CLUSTER COMPUESTO POR PLAYSTATION 3.....	94
15.1.1	GENERALIDADES.....	94
15.1.2	SISTEMA OPERATIVO.....	94
15.1.3	RECOMENDACIONES PREVIAS.	97
15.1.4	PREPARACIÓN DEL SISTEMA OPERATIVO.....	103
15.1.5	HERRAMIENTAS PARA EL DESPLIEGUE DEL CLUSTER.	108
15.1.6	SEGURIDAD BÁSICA.....	116
15.1.7	EJECUCIÓN PARALELA DE APLICACIONES.	116
16	<u>CONCRECIÓN DEL MODELO</u>	<u>119</u>
16.1	DIAGRAMA DE CAPAS DE IMPLEMENTACIÓN DE CLUSTER.....	119

<u>17</u>	<u>CONCLUSIONES.....</u>	<u>123</u>
<u>18</u>	<u>RECOMENDACIONES</u>	<u>127</u>
<u>19</u>	<u>BIBLIOGRAFÍA.....</u>	<u>129</u>
<u>20</u>	<u>ANEXOS</u>	<u>139</u>
20.1	DESPLIEGUE DE UN CLUSTER E INTEGRACIÓN A UN GRID.....	139

ÍNDICE DE FIGURAS

Figura 1. Arquitectura Cell BEA.....	59
Figura 2. Modelo Reloj de Arena GLOBUS.....	90
Figura 3. Interfaz de configuración del PS3	95
Figura 4. Diagrama de capas nodo maestro	120
Figura 5. Diagrama de capas nodos esclavos.....	121
Figura 6. Diagrama de interconexión del CLUSTER	122
Figura 7. Prototipo.....	139
Figura 8. Diagrama Prototipo Cluster	141
Figura 9. Comportamiento del Cluster.....	142

GLOSARIO

ALU (ARITHMETIC LOGIC UNIT): acrónimo en inglés de Unidad Aritmético lógica. Es la sección del circuito digital de un microprocesador encargada de ejecutar las operaciones lógicas y aritméticas con números enteros.

API (APPLICATION PROGRAMMING INTERFACE): acrónimo en inglés de Interfaces de Programación de Aplicaciones, es un conjunto de interfaces abstractas provistas por librerías de desarrollo o sistemas operativos para abstraer la complejidad de los subsistemas o dar acceso a facilidades de programación. Están compuestas por clases, métodos o funciones, protocolos, estructuras y tipos de datos.

ARM (ADVANCED RISC MACHINE o ACORN RISC MACHINE): acrónimo actual en inglés de Máquina Avanzada RISC, anteriormente era acrónimo de Máquina RISC de ACORN. Es la arquitectura RISC más difundida, diseñada por ARM LIMITED.

Es usada ampliamente en dispositivos móviles como teléfonos celulares, PDA, Smartphones, DAP y DVP como el Ipod, consolas portátiles de videojuegos como PSP y Nintendo DS, calculadoras enrutadores y gran variedad de dispositivos embebidos. Al igual que otros procesadores RISC más conocidos se caracterizan por su bajo consumo de electricidad y baja generación de calor.

Barrera de Latencia en Memoria : es una disparidad entre la velocidad de CPU y la memoria que está por fuera de ella. Esta disparidad ha sido el principal cuello de botella en el avance de la capacidad de procesamiento, por cuanto los accesos frecuentes a memoria principal debido a fallos en la predicción de ramificaciones y dependencias implican accesos más frecuentes a la misma. Una de las soluciones a este problema ha sido el reordenamiento de instrucciones, mayores espacios de cache y el uso de memoria local en arquitecturas ASMP. (Mahapatra & Venkatrao)(Novakovic, DRAM latency explained, 2007)

CISC (COMPLEX INSTRUCTION SET COMPUTER): acrónimo en inglés de Computador de Conjunto de Instrucciones Complejas. En contraposición a RISC los procesadores de este tipo fueron diseñados con mas cantidad de instrucciones de esta forma se facilitaba la programación, se requería menos código y menos accesos a memoria.

CLUSTER Beowulf: es un tipo de CLUSTER o racimo de computadores en los que sus nodos son PCS u otro tipo de "computadores" genéricos y de bajo costo. Su nombre se desprende del primer CLUSTER de este tipo desplegado en la NASA y cuya dirección estuvo a manos de Donald Becker. Usualmente se hace uso de herramientas y software libre o de código abierto y su interconexión se hace mediante conexiones LAN TCP/IP. (What makes a cluster a Beowulf?)

EIB (ELEMENT INTERCONNECTION BUS): acrónimo en inglés de Bus de Interconexión de Elementos. Es el sistema de interconexión en BUS que une los SPE entre sí con el o los PPU. Es una vía de conexión redundante de cuatro canales.

Fuertemente Acoplado: el concepto de acoplamiento es usado a través de múltiples disciplinas. En el caso de los procesadores se refiere a aquellos sistemas con múltiples unidades de procesamiento ya sean núcleos o procesadores que se encuentran interconectados a nivel de bus dentro de una sola máquina.

FPU (FLOATING POINT UNIT): acrónimo en inglés de Unidad de Punto Flotante. Es la sección del circuito digital de un microchip encargada de realizar operaciones aritméticas pero en números decimales o de representación de punto flotante. Un FPU puede contener sus propios ALU internos.

Hilo de Procesamiento (THREAD): es una tarea de ejecución independiente que puede ser una bifurcación de un proceso principal y puede ejecutarse paralelamente junto con otras. Normalmente un hilo ejecuta un determinado método o función dependiendo del tipo de lenguaje bajo el que se ejecute.

NUMA (NON UNIFORM MEMORY ACCESS): es un diseño de memoria computacional usado en arquitecturas de múltiples procesadores en los que cada unidad de procesamiento tiene su propia memoria local interna, lo cual significa que al estar más cerca de esta permite aumentar la velocidad de acceso a memoria. Adicionalmente un sistema NUMA puede hacer uso simultáneo de tipos de memoria convencionales. (Definition of: NUMA) (NUMA FAQ)

Oblea (WAFER): en el mundo de la informática un oblea se refiere al sustrato sobre el que se fabrican los microprocesadores, son comúnmente hechos a base de materiales semiconductores como los cristales silicio.

Paralelismo: es una forma de computación en la que múltiples cálculos pueden ser realizados simultáneamente. El paralelismo no es inherente al multiprocesamiento, incluso en monoprocesadores es posible ejecutar ciertas cálculos de forma simultánea mediante el uso de vectores de 64 o 128 bits que pueden almacenar y ejecutar 2 o hasta 4 instrucciones simultáneamente.

POWERPC o PPC (POWER PERFORMANCE COMPUTING): acrónimo en inglés de Computación de Rendimiento Poderoso. Es la segunda arquitectura RISC de mas uso desarrollada en conjunto entre Apple IBM y Motorola tradicionalmente han sido chips RISC y aunque han gozado de mejor desempeño que los chips x86 no fueron un éxito comercial y han caído un poco en desuso principalmente por el hecho de que su mayor comprador Apple recientemente mudo su sistema operativo a la mas ubicua arquitectura x86.

Actualmente son usados en todas las consolas de videojuegos desde el Wii de Nintendo pasando por el PlayStation 3 de Sony y el Xbox 360 de Microsoft. Adicionalmente son usados fuertemente en enrutadores y conmutadores de red y otros dispositivos como televisores e incluso algunos supercomputadores como el Roadrunner el cual es actualmente el computador más rápido del mundo.

PPU / PPE (POWER PROCESSING UNIT, POWER PROCESSING ELEMENT, PHYSICS PROCESSING UNIT): es el procesador principal y de más alta jerarquía de la arquitectura CELL de IBM TOSHIBA y SONY, cada procesador CELL puede en teoría contener uno o más de estos que son procesadores CISC POWERPC de Doble núcleo. Su función natural es despachar tareas al SPE o a los SPU aunque puede ser usado como un procesador de propósitos generales. (Kahle, Day, Hofstee, Johns, Maeurer, & Shippy, 2005)

PPU es simultáneamente el acrónimo en inglés de Unidad de Procesamiento de Física el cual es un tipo de tarjeta similar a las tarjetas aceleradoras de gráficos o GPU pero se encarga de acelerar cálculos de ciertos comportamientos físicos como el comportamiento de objetos en la gravedad, deformación de materiales, o comportamiento del fuego, el viento entre otros tantos. (Kahle, Day, Hofstee, Johns, Maeurer, & Shippy, 2005)

Programación Distribuida: es el tipo de programación usada en sistemas distribuidos como lo pueden ser CLUSTERS GRIDS o incluso computadores con procesadores múltiples SMP o ASMP. La ejecución puede ser lograda al nivel de bits, instrucciones, datos o tareas. En estos sistemas el paralelismo puede ser logrado por el compilador o por el programador.

Renderización: es un proceso en el que a partir de simulaciones computacionales de objetos detalladas y precisas desde el punto de vista de la física se generan imágenes secuenciales que muestran el comportamiento de dichos objetos en forma de video. Este proceso es usado usualmente en las películas de animación en 3D modernas como también en ciertos campos científicos debido a que la visualización en tiempo real de dichos objetos y comportamientos es imposible debido a la gran cantidad de cálculos que deben ser llevados a cabo para generar tan solo un cuadro de animación.

RISC (REDUCED INSTRUCTION SET COMPUTER): acrónimo en inglés de computador de conjunto de instrucciones reducidas. Es una estrategia de diseño de

microchips que hace énfasis en la simplicidad para lograr alto rendimiento, bajo consumo de electricidad, baja generación de calor y bajas frecuencias de operación.

SDK (SOFTWARE DEVELOPER KIT): acrónimo en inglés de Kit de desarrollo de software. Es un conjunto de herramientas integradas para el desarrollo de software. Lo componen lenguajes librerías de programación y sus API, compiladores, herramientas de diagnóstico y documentación.

SPU (SYNERGISTIC PROCESSING UNIT): acrónimo en inglés de Unidad de Procesamiento Sinérgico. Es un procesador de un solo hilo de ejecución y una de las unidades de procesamiento que contiene un SPE o Synergistic Processing Elements o elementos de procesamiento sinérgicos presentes en la arquitectura CELL BEA. (Kahle, Day, Hofstee, Johns, Maeurer, & Shippy, 2005)

SPE (SYNERGISTIC PROCESSING ELEMENT): son un conjunto de procesadores asimétricos llamados SPU que trabajan en conjunto y que conforman la arquitectura CELL de IBM SONY y TOSHIBA. Cada SPU comparte un bus o EIB de cuatro vías de comunicación con el resto de SPU y cada uno contiene un espacio de memoria local independiente de la memoria principal del sistema. (Kahle, Day, Hofstee, Johns, Maeurer, & Shippy, 2005)

Sueltamente Acoplado: se refiere a sistemas de múltiples procesadores en los que las unidades de procesamiento son independientes pero se encuentran interconectadas mediante algún tipo de interfaz de red. Los ejemplos más comunes de estos sistemas son los CLUSTERS Beowulf.

UMA (UNIFORM MEMORY ACCESS): es un diseño de memoria computacional usado en arquitecturas de múltiples procesadores en los que todas las unidades de procesamiento comparten la misma memoria. Este diseño es normalmente usado en arquitecturas SMP en conjunto con memorias cache para disminuir un poco la latencia y la ineficacia debido el acceso simultáneo a memoria. (Barney)

X86: es el término genérico para referir los chips con arquitectura compatible con los chips Intel 8086 entre los que se encuentran la mayoría de chips fabricados por AMD e Intel y que es la arquitectura más ampliamente difundida en parte a buenas técnicas de comercialización y publicidad.

ZII: es una nueva arquitectura de multiprocesamiento asimétrico desarrollada por CREATIVE en la que múltiples unidades de procesamiento o de transistores se reprograman en la marcha para procesar datos conjuntamente copando totalmente de trabajo todos los bloques de transistores de un microchip que en arquitecturas tradicionales pueden quedar sin carga o en tiempo de ocio.

Por otro lado facilita el ahorro de electricidad al estar en capacidad de usar solo los bloques de transistores necesarios para realizar determinado cálculo. Esta arquitectura supera en órdenes de magnitud las arquitecturas actuales aunque aun no hay productos comerciales que hagan uso de esta arquitectura.(Zii.com).

RESUMEN

El multiprocesamiento fue por muchos años una tecnología restringida a ciertos campos exclusivos, por cuanto el avance casi exponencial de la capacidad de procesamiento del monoprocesador suplía la mayor parte de las necesidades emergentes en este campo.

Estos avances se vieron interrumpidos por las barreras impuestas por la física, por lo que la industria optó por el uso del multiprocesamiento en un intento por continuar el ritmo establecido por la "ley" de Moore, la cual en vista de estos acontecimientos, ha perdido validez.

Al mismo tiempo, los nuevos paradigmas científicos demandan capacidades de procesamiento colosales que solo pueden ser logradas mediante la colaboración interinstitucional. Infortunadamente estas tecnologías han sido poco estudiadas por lo cual existe un desconocimiento en la academia.

Cualquier esfuerzo de despliegue debe partir de una demanda puntual por grandes capacidades de procesamiento, de lo contrario dichas soluciones se convierten en tan solo una novedad sin usos claros.

DESCRIPTORES: multiprocesamiento, paralelismo, globus toolkit, grids, clusters, mallas computacionales, racimos computacionales, calendarizadores, playstation, Yellow Dog Linux, pvm, mpi, openmpi, cell bea, broad band architecture, sun grid engine, powerpc, ppc, nfs, ssh, ogf. open globus forum, fstab. spu. ppu, coprocesadores, sinergisticos, amdalh, gustafson, moore, procesamiento, crisis, nucleos, simetrico, asimetrico, fuertemente acoplado, sueltamente acoplado, acoplamiento, arpanet, sage, semi automatic ground environment, ian foster, carl kesselman, steve tuecke, computación en la nube, Jhon Mccarty, I-WAY

ABSTRACT

Distributed Computing and multiprocessing were a restricted technology that was only found in some specific fields, thanks to the fact that the speed of mono-processors were increasing exponentially.

The obstacles imposed by thermodynamics interrupted the exponential increases in processor speed and the industry decided to use multiprocessing technology in every field and platform in an attempt to keep up the pace established by Moore's law.

At the same time, the new scientific paradigms demand colossal processing capacity that can only be achieved through inter-institutional collaboration. Unfortunately multiprocessing hasn't been studied deeply outside some very restricted fields and there is a lack of knowledge in the academy.

Any effort targeted at the design and deployment of multiprocessing solutions requires a punctual and clear demand, application or necessity and interdisciplinary groups that can help in every aspect of it. Otherwise the solution will become just a novelty without any practical uses.

DESCRIPTORS: multiprocessing, parallelism, globus toolkit, grids, clusters, , schedulers, playstation, Yellow Dog Linux, pvm, mpi, openmpi, cell bea, broad band architecture, sun grid engine, powerpc, ppc, nfs, ssh, ogf. open globus forum, fstab. spu. ppu, coprocessors, synergistic, Amdahl, Gustafson, Moore, processing crisis, core, symmetric, asymmetric, strongly coupled, loosely coupled, processor coupling, arpanet, sage, semi automatic ground environment, ian foster, carl kesselman, steve tuecke, cloud computing, Jhon Mccarty, I-WAY, folding at home, seti at home, distributed.net, lhc, large hadron collider, in order, out of order, smp asmp, soa, service oriented architecture

1 INTRODUCCIÓN

La evolución de los computadores de simples máquinas para hacer cálculos a medios de comunicación, información, control y entretenimiento, ha estado marcada por logros, que aunque pasan inadvertidos para el público general, han sido determinantes para su transformación en un elemento imprescindible para lograr toda clase de objetivos. Es sorprendente ver cómo tecnologías y aplicaciones que en su momento aparentaban ser frívolas, han influido en todos los aspectos de la vida humana, desde la forma como se enfrentan situaciones cotidianas hasta la manera en que se pretende evolucionar como sociedad y especie.

La manera de hacer ciencia ha cambiado radicalmente, los logros científicos heroicos unipersonales ya hacen parte de la historia, por cuanto la complejidad de la tecnología y de los problemas que se intentan resolver sobrepasan las capacidades naturales de los científicos y profesionales de todas las áreas, incluso sobrepasan la capacidad de los computadores personales, a pesar del avance aparentemente exponencial en la capacidad de procesamiento que venían experimentando los microchips.

Para lograr resolver los nuevos enigmas científicos se requiere de un poder de procesamiento inimaginable, que solo se puede obtener a partir de la colaboración tecnológica interinstitucional e internacional. Esta necesidad dio lugar al nacimiento de la computación distribuida y los conceptos de racimo de computadores o CLUSTER, como también a la computación en malla o GRIDS, junto con las técnicas de programación distribuida diseñadas para aprovechar estos sistemas, los cuales buscan compartir y usar recursos de procesamiento dispersos geográficamente.

Si bien es cierto estos conceptos en un principio estaban restringidos únicamente a los campos científicos, la necesidad de grandes capacidades de procesamiento ha llegado también a los escritorios y centros de entretenimiento de los hogares, debido

entre otras cosas a que infortunadamente la industria se ha topado con una barrera que impide el avance de la capacidad de procesamiento del monoprocesador, el cual hasta hace unos pocos años avanzaba exponencialmente, tendencia que parecía no tener un fin aparente.

Dicha barrera apareció debido a que los fabricantes de microchips se vieron imposibilitados en seguir incrementando la capacidad de procesamiento de estos dispositivos solo por medio de la reducción de tamaño y el incremento de frecuencias, siendo necesario recurrir a la idea de usar más de un procesador o núcleo en los chips de computadores. Actualmente se pueden encontrar microprocesadores de hasta ocho núcleos o procesadores independientes y se espera en un futuro no muy lejano tener decenas de estos en los computadores de escritorio y dispositivos móviles (ST-Ericsson, 2009), por ello la computación distribuida ya no pertenece únicamente a los ámbitos científicos sino que incluso ahora hace parte de todos los ámbitos de la computación.

El uso de sistemas de múltiples procesadores ya sean fuertemente acoplados en una sola placa de silicio o sueltamente acoplados en forma de CLUSTERS o GRIDS, no es un tema trivial e impone una carga inmensa sobre los hombros de los ingenieros de software, telecomunicaciones, hardware y los programadores. Desafortunadamente a pesar de que este concepto, las herramientas y técnicas de desarrollo existen desde hace muchos años, los profesionales del ámbito de la computación y las empresas han sido muy lentos a la hora de hacer amplio uso de ellas y aceptar esta nueva tendencia en su campo de acción.

De esta manera, los racimos y las mallas computacionales permiten el acceso compartido a recursos de procesamiento avanzados. Definen una arquitectura genérica de construcción de aplicaciones junto con una estructura distribuida no solo de infraestructura, sino también en desarrollo de MIDDLEWARE y aplicaciones.

Estos entornos, ya sean tipo CLUSTER o GRIDS computacionales son complejos y herramientas como el GLOBUS TOOLKIT (diseñada específicamente para su desarrollo) por si solas son vastas y meritorias de un estudio académico prolongado e interdisciplinario, aun si su instalación resulta relativamente trivial, es decir, la verdadera complejidad de estos sistemas subyace en la programación y el desarrollo sobre estas herramientas.

Este trabajo no pretende metas inalcanzables, ni ser un tratado sobre hardware o sobre una arquitectura específica. Los objetivos son dar un vistazo del estado del arte del procesamiento distribuido y determinar cuál puede ser su evolución y cómo esta ha sido influenciada por sectores que normalmente son desestimados e ignorados. Por otro lado se discutirán de forma general algunos conceptos y herramientas que pueden ser el punto de partida de un estudio más profundo sobre el procesamiento distribuido o una investigación aplicada por parte de las próximas promociones de ingenieros. Finalmente se plantearan algunas sugerencias e hipótesis propuestas por el autor.

2 ANTECEDENTES

Aunque en el país hay muchas instituciones que en mayor o menor grado usan ciertos tipos de CLUSTERS discretos en su infraestructura y procesos, aun no hay implementaciones ni integración en GRIDS nacionales o internacionales. Sin embargo, recientemente nació una iniciativa conformada principalmente por universidades y centros de investigación de Bogotá y Medellín que pretende integrar mediante la red RENATA en un GRID nacional de cómputo a todas ellas, en lo que se llamaría “GRID COLOMBIA”.

Su misión es, de acuerdo con su sitio Web: “En dos años completar el total de universidades conectadas en RENATA a nivel nacional y ser reconocida como la iniciativa nacional para la implantación, desarrollo e investigación en entornos de Grid. En cuatro años ser reconocida como una alternativa de Grid fuerte a nivel internacional.” (GRID COLOMBIA)

A nivel local los dos jugadores más grandes en este campo son la Universidad Tecnológica de Pereira y CENICAFE, quienes han implementado CLUSTERS para llevar a cabo ciertos proyectos al interior de sus instituciones, pero aun no están integradas ni están compartiendo recursos de procesamiento en GRIDS.

3 SITUACIÓN PROBLEMA

Los nuevos paradigmas científicos, las nuevas aplicaciones multimedia y de entretenimiento exigen cantidades bastas de procesamiento que el microchip por si solo ya no esta en capacidad de ofrecer. La industria se topo con una barrera que no permite mas el avance exponencial en las capacidades de procesamiento del monoprocesador por lo que recurrió a los paradigmas del multiprocesamiento y el procesamiento distribuido. Estos paradigmas que hasta hace poco estaban restringidos a los campos científicos mas exclusivos se han vuelto un requerimiento en todos los ámbitos de la computación y se han de volver la norma de la industria.

La nueva visión de país que implica el libre comercio y la globalización contempla la generación de conocimiento propio, infortunadamente la falta de inversión e investigación histórica en ciencia y tecnología por parte de los gobiernos ha causado que se cierren las puertas a la integración y colaboración con grupos de investigación internacionales y universidades extranjeras para lograr avances científicos propios que traerían consigo un mejor desarrollo socioeconómico al país.

Ya sea para fundamentar los conocimientos sobre el multiprocesamiento o la computación distribuida se desea investigar si es posible, teniendo en cuenta la infraestructura de telecomunicaciones y las carencias presupuestales habituales, implementar una solución computacional en CLUSTER de bajo costo que al mismo tiempo sea lo suficientemente interesante desde el punto de vista de la capacidad de procesamiento como para ser tenida en cuenta a nivel nacional e internacional por otras universidades y centros de investigación de tal forma que este sea participe de proyectos de computación distribuida.

4 OBJETO DE ESTUDIO

El objeto de estudio del proyecto es la teoría que comprende los sistemas de computación distribuida en CLUSTERS y GRIDS como también su metodología de implementación.

5 CAMPO DE ACCIÓN

La computación distribuida en CLUSTERS o GRIDS no está ligada a un campo de acción específico, siendo esta una herramienta de uso científico general con gran variedad de aplicaciones, desde “renderización” de animaciones en 3D, pasando por la planeación vial y urbana, hasta aplicaciones científicas como simulación de fluidos, choques de masas, plegamiento de proteínas y secuenciación genética.

En este entorno, el mayor beneficio con el desarrollo del proyecto, lo obtendría la educación local por cuanto no existe mucha documentación ni experiencia sobre esta disciplina en la región.

6 OBJETIVO GENERAL

Determinar una solución de computación en clúster que se ajuste al contexto local y al mismo tiempo que se integre exitosamente a GRIDS nacionales e internacionales, con un buen balance entre el costo y la capacidad de procesamiento.

7 OBJETIVOS ESPECÍFICOS

Estudiar la terminología y conceptos básicos sobre el multiprocesamiento y la computación distribuida ya sea en forma de CLUSTERS o GRIDS para que de esta forma se cimiente un lenguaje común de entendimiento.

Estudiar proyectos similares de otros países hayan o no hayan sido implementados, para aprender de sus aciertos y errores determinando así cuales de estos se ajustan mejor a nuestro medio.

Estudiar las herramientas y sistemas necesarios para implementar un sistema de computación en racimo o CLUSTER y proponer un diseño que se pueda integrar a un GRID o malla computacional.

8 DELIMITACIÓN DEL PROYECTO

Dado el alto costo que acarrea desplegar una implementación práctica de un CLUSTER y considerando el hecho de que la Universidad Católica Popular del Risaralda no posee dichos recursos, el proyecto se limitará a una investigación teórica que dará como resultado una propuesta que prepare el camino para futuros proyectos de implementación y aplicaciones sobre dicha plataforma.

9 APORTE PRÁCTICO Y TEÓRICO

El aporte teórico consiste en el compendio de información, documentación y experiencias referentes al objeto de estudio, el cual puede servir como base misma de una implementación práctica que podría ser llevada a cabo por la Universidad. Todo esto con el fin de cerrar la brecha cognoscitiva que sobre dicho tema existe actualmente en la región.

10 FACTIBILIDAD TÉCNICA

Las pruebas necesarias para elegir determinada plataforma, configuraciones de software y topología de red para un clúster que se integre a GRIDS pueden ser realizada mediante el uso de infraestructura hogareña tal y como se mostrará a lo largo del desarrollo del proyecto.

Por otro lado aunque sería ideal realizar pruebas de marcas y arquitecturas de hardware sobre equipos de cómputo específicos, es posible llegar a una conclusión basándose en implementaciones y experiencias previas de terceros sin incurrir en el costo que significaría la compra de equipos única y exclusivamente para la realización de dichas pruebas.

11 FACTIBILIDAD ECONÓMICA

El costo de implementar un CLUSTER está sujeto a una gran variedad de factores, principalmente a la proyección a largo plazo y los requerimientos técnicos los cuales se desprenden de las necesidades o problemas que se quieran abordar.

A pesar de esto, juegan a favor las mismas particularidades de los mismos en los cuales las diferentes unidades de procesamiento que los conforman son “sueltamente acopladas”, lo que quiere decir que aunque trabajan en conjunto, cada una es independiente y puede ser reemplazada sin que esto afecte permanentemente a las demás. Esto al mismo tiempo ofrece escalabilidad del sistema al poder añadir elementos y de esta forma expandirse según las necesidades.

Para llevar a cabo una futura investigación aplicada sobre CLUSTERS y GRIDS, no se requieren grandes inversiones de dinero en equipos de cómputo. La Universidad ya cuenta con una infraestructura de red y computadores, que aunque no son ideales para la puesta en marcha de un CLUSTER que cumpla con altos estándares de calidad, sirven para realizar un proceso de pruebas de aprendizaje y selección de los elementos y herramientas necesarios para implementar un CLUSTER que se integre eficientemente a GRIDS o mallas computacionales nacionales e internacionales. El mayor costo que acarrearía la investigación sería la adquisición de ciertos libros relacionados con el objeto de la misma, los cuales no tienen precios prohibitivos para una institución educativa.

12 MÉTODO DE INVESTIGACIÓN

Como se mencionó anteriormente el método de investigación que se usara será el de investigación teórica por cuanto se busca emplear conocimientos que ya han sido demostrados y puestos en práctica.

Para ello, se realizará una recopilación de información bibliográfica, la que será debidamente organizada y clasificada, de tal forma que a partir de los datos obtenidos se logre estructurar un modelo de desarrollo que se pueda implementar en un futuro en la Universidad.

13 MARCO CONTEXTUAL

13.1 MALLAS Y RACIMOS COMPUTACIONALES (CLUSTERS Y GRIDS)

13.1.1 Historia de los CLUSTERS y los GRIDS de computadores.

Aunque algunas compañías como DEC se adjudican la creación de los CLUSTERS de computadores entre 1977 y 1980, el concepto venía siendo usado desde mucho antes de los primeros estudios de Gene Amdahl en 1967, incluso antes de la creación de los microchips.

Uno de los primeros sistemas en hacer uso de estos conceptos fue el sistema SAGE Semi-Automatic Ground Environment construido conjuntamente para el NORAD (North American Aerospace Defense Command o comando de defensa aeroespacial norteamericano) por IBM y el MIT para detectar aviones bombarderos durante la guerra fría, pero posteriormente quedó obsoleto cuando los misiles y no los aviones bombarderos se convirtieron en la principal amenaza contra la seguridad de los Estados Unidos.

El SAGE fue construido incluso antes del desarrollo de los microchips y funcionaba gracias a tubos de vacío. Todo el sistema estaba constituido por subsistemas que se encontraban interconectados entre si y en tiempo real colaboraban para una detección temprana de aviones bombarderos.

A pesar de este hecho histórico la industria está de acuerdo en que no hay una fecha específica ni un evento particular que marque la invención de los CLUSTERS o racimos de computadores modernos. El consenso general ha sido que los usuarios fueron los que inventaron los CLUSTERS tan pronto se vieron incapacitados o limitados por sus computadores para procesar todos sus cálculos en tiempos

sensatos, requerían sistemas de alta disponibilidad o resistentes a fallos por cuanto se necesitaban respaldos de computadores. Aunque si hay que dar una fecha, tentativamente podría ser alrededor de los 60s. (Pfister, 1997)

Los primeros conceptos, estudios y teorías sobre la computación distribuida fueron desarrollados en 1967 a manos de Gene Amdahl en medio de su trabajo en IBM. Sus estudios dieron como resultado la ley de Amdahl apodada así en honor a su trabajo (Amdahl, 1967).

La ley de Amdahl permite medir matemáticamente qué velocidad de procesamiento se puede obtener al paralelizar determinado algoritmo o cálculo. Esta ley ha sido usada por algunas personas como argumento a favor del procesamiento paralelo masivo, mientras que la ley de Gustafson ha sido usada para establecer límites en cuanto a qué rendimiento se puede obtener al paralelizar el procesamiento (GUSTAFSON, 1988).

Recientes estudios han demostrado que ambas leyes son de hecho la misma ley, pero su uso no ha sido correcto y por ello se han postulado para demostrar teorías opuestas. Uno de los estudios sobre esta materia (Shi, 1996), concluye que la obtención de resultados no depende enteramente del procesador, por cuanto la discusión intelectual de múltiples procesadores contra un solo procesador poderoso es irrelevante y para medir el rendimiento esperado al paralelizar una tarea se está condicionado, además del mencionado procesador, a muchos más factores técnicos, como la latencia de la memoria, velocidad de la red, el sistema de entrada y salida, así como de la forma en la que se plantea el problema.

A pesar de las críticas contra estas leyes o contra esta ley, esta continúa siendo válida desde el punto de vista matemático. El problema recae en la forma en que se usa dicho algoritmo para obtener las medidas de desempeño esperado. Una sugerencia que se plantea es el uso de la variable de tiempo en los cálculos, esto permite evaluar en el mismo problema el impacto de otros factores como los ya mencionados. (Shi, 1996)

Posterior a los estudios de Amdahl el uso de CLUSTERS experimentó un impulso gracias a la creación de las primeras redes de computadores, pero fue la implementación por parte de la compañía RAND del concepto propuesto por Paul Baran (Baran, 1962) de las redes de conmutación de paquetes, lo que dio nacimiento al primer CLUSTER basado en computadores genéricos no especializadas, este CLUSTER se llamó la ARPANET que posteriormente evolucionaría en la INTERNET.

Pero las herramientas que facilitaron el procesamiento distribuido y el compartir archivos no fueron desarrolladas sino hasta 1983 con la creación de la rama libre BSD de Unix, la cual fue pionera en el despliegue de una versión abierta de Unix mucho antes de que se popularizara GNU/LINUX. El sistema BSD al ser abierto y estar enfocado a individuos e instituciones educativas pequeñas tuvo como resultado inevitable el uso masivo de computadores genéricos en la elaboración de CLUSTERS.

Los primeros CLUSTERS comerciales fueron el ARCnet fabricado por Datapoint en 1977 pero no fue un gran éxito, hasta que en 1984 DEC sacó su primer producto el VAXCLUSTER (Vax por el Sistema operativo que usaba). También hay que nombrar el Tandem Himalaya en 1994 y por supuesto el IBM S/390 Parallel Sysplex que, aunque no fueron los primeros si fueron muy exitosos.

No fue sino hasta 1989 que gracias a la apertura al público de la ARPANET como Internet y a la creación del sistema PVM, que los CLUSTERS basados en componentes genéricos despegaron y lograron exceder en capacidad de procesamiento a los CLUSTERS comerciales. Esta tecnología fue estudiada y usada fuertemente por parte de la NASA en 1993 y se definieron finalmente los lineamientos para la construcción de computadores de alto desempeño, basados en componentes genéricos interconectados mediante tecnologías TCP/IP, a los que se les dio el nombre de CLUSTERS Beowulf. (Stutz, 1998) (Merkey) (The History of Custer Computing)

Con la popularización de Internet se creó un ambiente de computadores muy heterogéneo, en él que participaban compañías e instituciones no solo con distintos tipos de recursos computacionales y de información, sino también con políticas organizacionales independientes. Esta particularidad creó un concepto nuevo el cual se conoce como GRID.

13.1.2 Características de las mallas y racimos de Computadores.

El término GRID, los algoritmos, técnicas y herramientas fueron creados gracias al planteamiento de Ian Foster, Carl Kesselman, Steve Tuecke, en 1995 en su trabajo "el GRID: diseño de una nueva infraestructura de computación"(The GRID: Blueprint for a new computing infrastructure).

Al estudiar las características de un GRID en trabajos como el de Ian Foster, Carl Kesselman y Steve Tuecke, las diferencias entre un GRID y un CLUSTER son claras. En un CLUSTER se tienen interconectados equipos homogéneos o heterogéneos es decir con arquitecturas, procesadores sistemas operativos y hardware igual o desigual. En él, todos los recursos están administrados por un nodo central, trabajan como si fueran un solo sistema, normalmente se encuentran bajo un dominio institucional específico y geográficamente cerca unos de otros (What is a Computer Cluster?, 2006). En el caso de los GRIDS se trabaja usando sistemas independientes distribuidos en regiones geográficamente separadas y bajo el dominio de instituciones con políticas organizacionales independientes y únicas.

El despliegue y la gestión de un GRID por lo tanto implica una dificultad mayor al tener que sincronizar esfuerzos entre máquinas e instituciones significativamente diferentes y que no siempre pueden estar disponibles, lo cual lleva nuevamente a plantear el interrogante sobre qué tan correcto es estudiar un tema tan amplio y complejo sin una colaboración interdisciplinaria e interinstitucional.

La popularización de los GRIDS se hizo evidente gracias a iniciativas como distributed.net en 1997, seti@home en 1999 y folding@home en el año 2000 los

cuales lograron llevar la colaboración a estos proyectos a los computadores personales de millones de personas en todo el mundo, sin que estuvieran enteramente dedicados a estos propósitos.

Ejemplos más recientes de GRIDS son el GRID usado para procesar los datos del LHC (Large Hadron Collider) el cual es el experimento mas costoso y gigantesco hasta la fecha, que busca encontrar la partícula más elemental que compone la materia, la llamada “partícula de Dios” o el Boson de Higgs y probar de esta forma el modelo estándar de la física de partículas.

El LHC es el colisionador de partículas más grande jamás construido y ocupa una longitud de 27 Kilómetros y toca cuatro puntos de la frontera entre Suiza y Francia. Este experimento se venía preparando desde 1991 (RELEASE, 1994) y para poder procesar los datos del experimento se creó un GRID en el que aproximadamente 100 países colaboran con recursos computacionales de procesamiento y almacenamiento.

13.2 MALLAS Y RACIMOS DE COMPONENTES EXÓTICOS

Recientemente el mundo de la supercomputación, antes restringido para fines científicos y educativos, ha venido entrando en el campo del entretenimiento, por lo que los CLUSTERS o incluso los GRIDS ya no solo están compuestos de computadores genéricos de bajo costo, sino también de plataformas de entretenimiento y multimedia, como los CLUSTERS basados en consolas de videojuegos PlayStation 3, así como también en tarjetas aceleradoras de video en 3D (Cohen, 2008).

Ambas plataformas han dado resultados formidables en capacidad de procesamiento para fines científicos y en el caso del PlayStation 3 esto solo ha sido posible gracias a la inclusión de un procesador creado para revolucionar el procesamiento, en una época en la que la ley de Moore chocó estrepitosamente con los límites impuestos por la física y la matemática (Khanna, 2007) (Huang, 2003) (Folding@home Client statistics by OS, 2009) (Pande, 2008).

Sin embargo, es necesario aclarar que el chip del PlayStation 3 no es un chip restringido a los videojuegos sino que también hace parte del ROADRUNNER de IBM el que es al día de hoy el computador más potente de mundo.

En el caso de las tarjetas aceleradoras de video, éstas han venido explotando el procesamiento paralelo desde el día en que fueron creadas y son particularmente eficientes en ciertas tareas que no solo se encuentran en el mundo de la animación en 3D y la multimedia, sino también en ciertos campos científicos.

Algunos expertos incluso afirman que los procesadores del futuro serán más parecidos a las tarjetas aceleradoras de video (Evans) (AMD Focuses on Fusion CPU/GPU Concept, 2006) (Hodgin, 2008) (Maesschalck, 2008) (Antwerp), la prueba de ello ha sido no solo el CELL BEA presente en el PlayStation 3 y el ROADRUNNER, sino también el chip del PlayStation 2, el cual aun hoy día, diez años después de su lanzamiento, continua sorprendiendo.

13.3 E-SCIENCE LA NUEVA FORMA DE HACER CIENCIA

En 1999 John Taylor, director del Concejo Nacional de Investigación del Reino Unido acuñó el término para referirse a esta nueva tendencia de la ciencia moderna a larga escala, la cual solo podría ser llevada a cabo mediante la colaboración global distribuida por intermedio de la Internet. Permitiendo de esta forma, a los esfuerzos de colaboración científica y los mismos científicos, acceder a recursos computacionales de datos y visualización a gran escala y de alto desempeño. (Thorpe, 2009)(Taylor, 2008)(Getov, 2008)

Este cambio de paradigma en la investigación solo ha sido posible gracias a las nuevas tecnologías disruptivas como la computación y la Internet, impulsada por los grandes volúmenes de datos y toda clase de dispositivos avanzados que han nacido de una gran variedad de disciplinas.

De esta manera, la forma de hacer ciencia ha cambiado radicalmente, ya no se tiene que esperar pacientemente un fenómeno natural para estudiarlo sino que se puede

modelar virtualmente y de esta forma hacer mejor uso del tiempo y los recursos, brindando a los científicos posibilidades casi infinitas.

Es posible simular computacionalmente una ciudad y probar cambios en la malla vial sin afectar el tránsito real, simular mercados y las reacciones de estos a ciertas medidas económicas, planear estructuras y someterlas a toda clase de pruebas sin invertir frecuentemente en prototipos. En fin las posibilidades son infinitas.

Los ejemplos más conocidos sobre esta nueva tendencia en la ciencia han sido proyectos como SETI@HOME, el cual busca posibles señales de origen extraterrestre o FOLDING@HOME cuya finalidad es el entendimiento del plegamiento de proteínas y las enfermedades relacionadas con los males de Alzheimer, Parkinson, Huntington y muchos tipos de cáncer.

De igual manera, el GRID creado para procesar los datos arrojados por el colisionador de hadrones LHC (LARGE HADRON COLLIDER), definido como el proyecto científico más grande y costoso de la historia hasta el momento.

En Colombia, iniciativas como la red RENATA pretenden interconectar universidades y otras instituciones educativas para llevar a cabo toda clase de proyectos, desde el despliegue y puesta en marcha de laboratorios remotos y videoconferencias, hasta la colaboración en proyectos como LHC e investigaciones locales como el estudio de los efectos de campos electromagnéticos externos y la presión hidrostática en las propiedades optoelectrónicas de pozos, hilos y puntos cuánticos o colaboraciones con otras universidades de la región como el estudio de aplicaciones de la teoría generalizada de formas.

A nivel regional uno de los proyectos más llamativos es el estudio del genoma del café por parte de CENICAFE, la broca y el hongo controlador *Beauveria bassiana*, con el propósito de desarrollar nuevas variedades resistentes a plagas y enfermedades y buenos atributos de calidad en taza, así como el desarrollo de

mejores estrategias para el control de enfermedades y plagas que permitan la disminución de los costos de producción y la sostenibilidad de la caficultura colombiana. (Cenicafe)

En el centro de todos estos estudios e investigaciones se encuentra el poder computacional que solo el multiprocesamiento o el procesamiento distribuido pueden brindar, ya sea en supercomputadores, CLUSTERS o GRIDS. Pero este poder computacional por si solo es inútil si no se tienen profesionales formados con las habilidades necesarias para hacer uso de él y servir de puente entre los científicos y estas tecnologías.

13.4 LA INFLUENCIA DE LA INDUSTRIA DEL ENTRETENIMIENTO EN LA COMPUTACIÓN Y EL PROCESAMIENTO

Durante la investigación sobre el procesamiento distribuido un hecho particular saltó a la vista, aunque los conceptos del multiprocesamiento fueron desarrollados hace 45 años, estos fueron olvidados rápidamente debido a las nuevas técnicas de fabricación de microchips que permitieron el éxito del monoprocesador y sus incrementos casi exponenciales en la capacidad de procesamiento.

Pese a esta situación, las consolas de videojuegos y el entretenimiento siempre han empujado los límites del hardware y el software, influyendo de gran manera en la evolución no solo de la computación en general sino del procesamiento, hecho que normalmente pasa desapercibido en la academia, puesto que las consolas de videojuegos son usualmente desestimadas técnicamente por algunos profesionales al estar enfocadas al entretenimiento y ser divertidas, llegando al punto de afirmar que no es computación seria.

La computación en épocas pasadas, e incluso hoy día, no se concibe como algo divertido, pero empresas como Atari, Nintendo, Sega, Sony, Apple e incluso Microsoft con sus consolas de videojuegos han innovado en este campo y han hecho del mundo de la computación lo que es hoy.

En el año 2001 Apple lanzó al mercado su PowerMac G4 con procesador dual de IBM, era el primer computador comercial en incluir dos procesadores. En su momento fue visto como un gran avance, algo que normalmente solo sería usado por diseñadores gráficos o cualquier otra profesión o tarea que requiriera de grandes capacidades de procesamiento, sin embargo, el PowerMac G4 no fue el primer producto en hacer uso simultáneo de dos CPU en equipos comerciales.

Sega, con su Sega Saturn lanzado en 1995 fue el primer producto comercial en incluir dos procesadores centrales, dos Hitachi SH-2 RISC de 32 bits a 28.6 MHz. Su decisión fue muy controvertida ya que no fue la mejor implementación desde el punto de vista técnico, ya que ambos procesadores compartían el mismo BUS, el cual solo tenía una dirección esto quiere decir que solamente un procesador podía usar el BUS en determinado momento, por otro lado el acceso a la memoria era ineficiente ya que igualmente solo uno podía acceder a la vez, un problema común en las arquitecturas de multiprocesamiento simétrico o SMP de lo cual se hablará más adelante.

El Sega Saturn no solo tenía dos procesadores centrales (CPU) sino también dos procesadores gráficos (GPU) el VDP1 y VDP2 un concepto que solo se volvió a ver hasta hace unos pocos meses (Novakovic, 2008).

Aunque el Sega Saturn fue el primer producto con un éxito comercial que incluía múltiples CPU y GPU el concepto de usar dos procesadores simultáneamente venía siendo usado de forma práctica desde la mitad de los años 90 cuando se comenzaron a incluir tarjetas aceleradoras de video en 3D para computadores personales, era la época de oro de los juegos de computador.

A pesar de que ya existían tarjetas aceleradoras en 2D para computadores tanto AMIGA como PC en los 80's y principios de los 90, estas eran poco comunes y el uso del procesador principal bastaba, pero a medida que los videojuegos avanzaron se requirió de mayor potencia y se comenzaron a popularizar las tarjetas aceleradoras 3D tanto en PC's como en consolas de videojuegos. En un principio los

programadores seguramente no están muy felices, era la primera vez que debían programar intensivamente para lo que en la práctica son dos procesadores, aun sí uno fuera CPU y el otro GPU. No era para menos, programar de esta forma siempre ha sido demasiado complejo incluso hoy en día.

13.5 EL LEVANTAMIENTO DEL SOFTWARE

En un principio el software era visto como algo secundario, como algo estático que se incluía junto con el hardware, pero los roles se invirtieron y el software comenzó a tomar el papel protagónico en la industria. Las consolas de videojuegos pero en especial Nintendo establecieron un modelo de hacer negocio en el que el dinero está en el contenido y no en la plataforma sobre la que este se ejecuta. Esta estrategia no solo continúa siendo la norma en la industria de los videojuegos, sino que fue tomada por Microsoft y fue clave para establecer su imperio y al mismo tiempo ponerle fin a la posición histórica de los fabricantes de hardware como los líderes de la industria.

Las consolas de videojuegos normalmente son lanzadas al mercado con especificaciones técnicas sorprendentes y tecnologías radicales, pero los fabricantes venden sus consolas “a pérdida” y obtienen sus ganancias por las ventas de juegos (contenido) y la reducción de costos de fabricación.

Si se analiza el costo de fabricación de la mayoría de consolas durante sus primeros años de vida este supera ampliamente el valor que el consumidor paga por él (Goldstein, 2006) (Hayes, 2005). En pocas palabras con cada consola que se vende los fabricantes de hecho pierden dinero, pero todo jugador de videojuegos sabe que en el transcurso de la vida de la consola se pueden llegar a comprar tantos juegos, que el costo de estos duplica o triplica el costo inicial de la consola.

Adicionalmente y en vista de que el hardware de las consolas es irrevocablemente el mismo durante toda su vida útil, que puede llegar a ser de 6 o hasta 10 años, los costos de fabricación se reducen al final de la misma y en el peor de los casos las

pérdidas cesan, en el mejor bajan tanto que obtienen ganancias con las que subsidian las pérdidas iniciales.

Es una de las razones por las que las consolas de videojuegos han tenido tanto éxito, garantizan a su dueño un flujo constante de juegos por lo menos por 6 o incluso 10 años, totalmente compatibles con su hardware sin tener que invertir en tarjetas de video, hardware especializado o actualizar su computador como ocurre en muchos casos con los juegos para PC, los cuales salen al mercado para el máximo común denominador, dejando por fuera algunos dueños y proporcionando una experiencia desigual a través de todos los usuarios.

13.6 LA EVOLUCIÓN EN EL HARDWARE DE LAS CONSOLAS DE VIDEOJUEGOS

En la época de su lanzamiento, el hardware del PlayStation 2 era bastante potente, tanto así que algunas naciones como Japón y Estados Unidos declararon el chip del PlayStation 2 como potencial arma de guerra y prohibieron su exportación a países conflictivos como Libia e Irak (Farah, 2000) (Smith, 2001) (Martin, 2000) (Wearden, 2000) (Bowman, 2001).

Por otra parte, el Xbox original, aunque aparentemente era muchísimo más potente que el PlayStation 2, ya que el incluía un procesador Pentium 3 de 733 Mhz mientras el PlayStation 2 un chip de 294 MHz, tenía como principal inconveniente que su microchip era un procesador de 32bits diseñado para propósitos generales y PCS de escritorio, más no para aplicaciones multimedia y juegos en los que priman las operaciones escalares, las transformaciones aritméticas y matemáticas en 3D las cuales eran la fortaleza del chip de 128 bits Emotion Engine de la PlayStation 2.

Para sus posteriores versiones, Microsoft no solo comenzó a usar procesadores más cercanos al mundo de la computación de escritorio, sino que decidió implementar puertos USB en vez de puertos propietarios para los periféricos, discos duros ATA, reproductores de DVD e incluso tarjetas inalámbricas WIFI y Bluetooth. Todo esto

puso al Xbox y al Xbox 360 más cerca de ser un computador que de ser una consola tradicional de videojuegos, hecho que no se dio por casualidad, ni como una forma adicional de reducir costos, sino que por el contrario evidenció que la multimedia y el entretenimiento han influido en la forma en la que se diseñan los microchips hasta el punto que se podría pensar que ocurrió el proceso inverso, donde los microchips de los computadores de escritorio comenzaron en un principio a tomar elementos de los microchips de las consolas de videojuegos.

Posteriormente, Sony con su PlayStation 3 subió las apuestas al usar un procesador revolucionario de IBM llamado el CELL BEA (Cell Broadband Engine Architecture) y un tipo de procesamiento que normalmente no está a disposición de las personas como es el multiprocesamiento asimétrico o ASMP el cual es más común en el campo de la supercomputación.

El Cell BEA cuenta con nueve procesadores con una frecuencia de 3.2 GigaHertz, un chip de red dedicado que quita la carga del manejo de las tramas de red de los procesadores principales, chip Bluetooth y Wi-Fi incluidos de fábrica, al igual que un disco duro SATA desde 60GB, memoria RAM XDR de 256MB, una tarjeta de video NVIDIA RSX de 256MB desarrollada conjuntamente entre Sony y Nvidia de la cual hasta el día de hoy se desconocen muchas características y especificaciones secretas. También cuenta una unidad óptica Blu-Ray la cual es el nuevo estándar de almacenamiento y reproducción óptica, con capacidades de hasta 200GB, aunque con la especificación actual estandarizada se optó solo por un máximo de 50GB para reducir los costos de fabricación ya de por sí altos.

El chip incluido en el PlayStation 3 no está solo presente en esta consola. Este procesador hace parte de una estrategia conjunta de IBM, SONY y TOSHIBA para crear un nuevo tipo de procesador que solucione la crisis de procesamiento por la que está atravesando la industria, mediante el uso del procesamiento paralelo masivo, no solo para consolas de videojuegos sino también para servidores de alto rendimiento, centros de entretenimiento y televisores de alta definición, llevando de esta forma el procesamiento distribuido al interior de los hogares, en donde todos los dispositivos comparten sus recursos de procesamiento. (Spooner, 2002)

13.7 LA CRISIS DEL PROCESAMIENTO, LA LEY DE MOORE TAMBALEA

Una de las estrategias de Apple siempre fue la de sacar un portátil de gama alta con el procesador de mayor rendimiento disponible en el momento, lo que solo fue posible durante el auge del PowerPC G4, el cual era incluido en la línea de portátiles Powerbook. Luego del PowerPC G4 salió a la luz el G5, que no solo era de doble procesador sino que se constituía como el primer producto comercial en usar la nueva arquitectura de 64 bits. Apple la comercializó como el computador de escritorio más rápido del mundo, ciertamente era rápido pero no era el más rápido del mundo y tampoco era el primer procesador de 64 bits.

El uso de procesadores de 64 bits data de los años 70 y 80, pero el uso de procesadores de 32 bits tuvo mayor acogida debido a que la principal utilidad de los registros de 64 bits era poder direccionar más cantidades de memoria RAM y en ese entonces no se veían aplicaciones comerciales que pudieran requerir toda la memoria que se puede direccionar con 64 bits.

Hasta finales de los años 90's, Intel e IBM/Motorola tenían dos estrategias tecnológicas muy distintas. Intel manejaba una arquitectura de instrucciones complejas CISC e IBM/Motorola una arquitectura de instrucciones reducidas RISC (Bishop & House, 1998) . Los procesadores RISC eran más sencillos y se evitaba en lo posible incluir instrucciones extras que facilitarían el trabajo a la hora de programarlo. Esto mantenía al procesador más simple con menos transistores y por tanto requería menos frecuencias por lo que consumía menos energía, generaba menos calor y era más eficiente. De hecho, los computadores Apple solo hasta principios del año 2000 requirieron enfriamiento con ventiladores.

La estrategia de INTEL era distinta. Incluir mas instrucciones para facilitar el desarrollo pero esto significaba incrementar la frecuencia y la memoria caché para mejorar el rendimiento (Sprangle & Carmean, 2002) lo que tenía un efecto negativo y era que se requería mayor longitud de su vector de instrucciones o "pipeline" lo cual incrementa el número de ciclos necesarios para realizar determinada instrucción o cálculo al incrementar la cantidad de pasos que se requieren para procesar determinada instrucción.

Al mismo tiempo el vector de instrucciones al ser más profundo implicaba que las ramificaciones (branches) lo obligaban purgar todo el vector y por lo tanto descartar muchas instrucciones a medio terminar, generando caídas en la capacidad de procesamiento real (Stokes, 2002) (Hodgin, What is a processor's pipeline?, 2006), por ello INTEL comenzó a incrementar la frecuencia de los procesadores buscando compensar este fenómeno y los Hertz comenzaron a ser la medida de potencia de facto de un procesador, a pesar de que técnicamente esta no es del todo una verdadera medida de capacidad de procesamiento.

Los procesadores de INTEL en un principio eran risibles, había incluso bromas publicitarias dentro del sector técnico sobre lo ineficiente que eran los procesadores INTEL, en especial el Pentium 4. Pero el número de Hertz tenía un impacto contrario en la percepción de los consumidores, después de todo las personas suelen usar analogías cotidianas para entender lo no cotidiano y frecuentemente caen en la analogía del automóvil, en donde más kilómetros por hora significan más velocidad. ¿Porque no habría de serlo entonces los Hertz en los procesadores?

De esta manera, los chistes comenzaron a ir en contra de IBM/Motorola y Apple, ya que se les comenzó a tildar de tener procesadores inferiores. A los ojos de un consumidor un INTEL Pentium de 1.4GHz era más rápido que un PowerPC G4 de 700Mhz, tiene el doble de números después de todo.

Los tiempos cambiaron y la complejidad de los procesadores RISC de IBM fue aumentado gradualmente hasta el punto que RISC ya no significaba realmente nada (Stokes J. , RISC vs. CISC: the Post-RISC Era, 1999). IBM decidió entonces usar la misma estrategia de INTEL en un intento de mejorar el rendimiento y mejorar la percepción pública de sus procesadores y comenzó a incrementar la frecuencia de los procesadores hasta más no poder.

Gracias a estos incrementos en la frecuencia, los procesadores estaban consumiendo demasiada energía y produciendo demasiado calor, hasta el punto que si continuaban los aumentos de frecuencia, estos se fundirían o requerirían costosos

y voluminosos sistemas de enfriamiento por nitrógeno y todo esto sin necesariamente aumentar la capacidad de procesamiento real.

La ley de Moore entonces comenzó a tambalearse y este hecho se hacía cada vez más evidente, la antes absoluta ley comenzaba a verse artificial. Los aumentos en capacidad de procesamiento hasta ese entonces estaban ligados a aumentar las instrucciones, los transistores, la frecuencia y a reducir la escala en la que se fabrican, pero ningún fabricante estaba en capacidad de continuar el ritmo que estableció la mencionada "ley". (Broersma, 2004) (Agarwal, Hrishikesh, Keckler, & Burger)

Al final las dos arquitecturas llegaron al mismo obstáculo, ya no importaba quien había llegado primero, todos estaban estancados en el mismo lugar. La última versión del Pentium 4, a pesar de haber incrementado la frecuencia del procesador, no había logrado aumentar la eficiencia y la capacidad de procesamiento, lo que era un golpe tremendo, pero se guardó silencio hasta que las pruebas de rendimiento comenzaron a aparecer en todas partes y la industria decidió aceptar la realidad (Mainelli, 2002) (King, 2007) (Times, 2000) (Kanellos, 2004) (Shilov, 2004).

En ese momento la industria decidió retomar el olvidado concepto de usar más de dos núcleos o procesadores fuertemente acoplados, de esta forma se estaba aumentando aparentemente la capacidad de procesamiento. Después de esto, toda la industria comenzaría a ocultar o ignorar la frecuencia de los procesadores al momento de comercializarlos, ya que muchos de los procesadores venideros tenían frecuencias inferiores.

El primero fue INTEL, sus futuros productos ya no promocionaban orgullosamente en letra negra capital de media página los 3.8 GHZ que antes enseñaban con orgullo durante el auge del Pentium 4, en vez de esto comercializarían sus procesadores por nombres claves y por su cantidad de núcleos y la frecuencia quedo relegada a una esquina de las especificaciones.

IBM siguió la misma ruta, pero a diferencia de INTEL apostó por dos implementaciones diferentes, una de procesamiento simétrico o SMP usada en el Xbox 360 similar al PowerPC G5 que usaba Apple y también similar a los nuevos procesadores INTEL y una arquitectura asimétrica ASMP usada en sus servidores de alto rendimiento, televisores de alta definición y en la consola de videojuegos de SONY.

La nueva hoja de ruta de INTEL mostraba procesadores de múltiples núcleos en una arquitectura basada en Pentium 3, más eficiente en cuanto a consumo de electricidad y generación de calor y más importante aún, multiprocesamiento simétrico (SMP), el cual a pesar de tener unos límites claros es una tecnología que evita la necesidad de reescribir totalmente el software para hacer uso de múltiples procesadores.

IBM, ante la noticia de Apple de cambiarse a procesadores INTEL, se apresuró en ofrecerle su procesador CELL BEA el cual trabaja en una arquitectura ASMP, considerada como el tipo de procesamiento del futuro, sin embargo Apple y Steve Jobs declinaron la oferta. El CELL BEA después de todo hubiera implicado hacer modificaciones de fondo al sistema operativo debido a la peculiaridad de ser ASMP y Apple tomó la mejor decisión estratégica, la cual era estar en la misma canasta que el resto de la industria.

De esta forma, la crisis en el procesamiento encontró una solución a mediano plazo con el uso del procesamiento distribuido, desafortunadamente este es un paradigma estudiado y usado casi exclusivamente en sectores científicos que no ha sido muy explotado ni enseñado fuertemente en la mayoría de Universidades. El procesamiento distribuido debe volverse algo común no solo por las implicaciones positivas que puede tener a nivel científico sino también porque es un hecho que este paradigma del procesamiento estará presente en toda la industria de la computación.

13.8 EL PORQUE DEL PROCESADOR CELL BEA

La idea de usar una consola de videojuegos como herramienta computacional no es una invención nueva, aunque a nivel internacional aun continúa siendo una idea innovadora, aunque a pesar de que hay varias iniciativas e instituciones renombradas investigando en este sentido, su acogida no ha sido tan amplia.

Una de las razones de la situación mencionada anteriormente es el prejuicio que se tiene hacia los videojuegos y las consolas de videojuegos, pero el principal obstáculo ha sido lo poco usual que es esta plataforma, lo que implica tener la disposición y el interés para probar algo diferente y también en muchos casos poseer ciertos conocimientos que no han sido fuertemente enseñados en los cursos de las carreras afines, principalmente la programación multihilo, distribuida y el lenguaje ensamblador.

Es importante mencionar que el objetivo del presente proyecto no es el uso de una consola de videojuegos como plataforma computacional, sino el estudio de los conceptos del procesamiento distribuido y paralelo en forma de CLUSTERS o GRIDS. La propuesta del uso de un procesador como es el CELL BEA en su estudio, fue el resultado inevitable de la dirección que tomó la investigación por ser este un ejemplo a escala de estos conceptos y la consola de videojuegos PlayStation 3 se presenta como una alternativa de muy bajo costo para acceder a este procesador, el cual sea dicho de paso, no solo se encuentra en la consola PlayStation 3 sino también en los servidores IBM.

Por otro lado muchas instituciones educativas internacionales y centros científicos reconocidos decidieron hacer uso de esta consola de videojuegos para el despliegue de supercomputadores en forma de CLUSTERS y han obtenido resultados sorprendentes, por lo que no se podía pasar por alto el proponer esta plataforma en un diseño de CLUSTER. De hecho, el supercomputador más potente al día de hoy es el Roadrunner de IBM y cuenta con el mismo procesador.

El Roadrunner está ubicado en el laboratorio nacional de los Álamos en Nuevo México Estados Unidos y está siendo usado principalmente para calcular la descomposición del arsenal nuclear de Estados Unidos. (Fildes, 2008) (Gaudin,

2008) (IBM, 2007) El hecho de proponer el uso de una consola de videojuegos como posible plataforma es meramente incidental ya que no se sabe con certeza si en un futuro se pretenda incluir nuevamente el procesador CELL BEA en un dispositivo de este tipo o si se va a permitir el acceso a desarrolladores ajenos a los videojuegos al interior del mismo, las cuales son las principales razones para proponer el uso de esta plataforma de entretenimiento en la investigación sobre GRIDS y CLUSTERS.

De todas las instituciones que se encuentran investigando sobre esta plataforma resaltan dentro de la búsqueda realizada el Centro Nacional de Supercomputación de Barcelona, en donde se encuentran trabajando no solo en desarrollar los conocimientos sobre la computación paralela sino también en aplicaciones para este modelo de procesamiento y en el “CELL Superescalar (CELLs) Programming Model” o modelo de programación CELL Superescalar.

El chip CELL BEA al ser un multiprocesador asimétrico no es fácil de programar o por lo menos no se puede programar de la misma forma que se ha hecho tradicionalmente y esperar que todo el poder de este procesador esté disponible inmediatamente. Se requieren conocimientos peculiares y habilidades cuya enseñanza no es muy común hoy día, por este motivo, el CELLS busca desarrollar un modelo de programación y compiladores que faciliten la programación.

La meta es que los programas que fueron desarrollados sin tener en cuenta el paradigma del multiprocesamiento puedan hacer uso de casi todo el poder de este procesador simplemente añadiendo ciertas “banderas” antes de las funciones o métodos. Posteriormente el compilador se encarga de crear un mapa de dependencias y genera el código listo para ser ejecutado en todos los procesadores disponibles (Pérez, Bellens, Badia, & Labarta, 2007).

La Universidad de Dartmouth en Massachusetts tiene su propio centro de investigación sobre gravedad y agujeros negros y actualmente lleva a cabo sus cálculos y modelos en un CLUSTER compuesto por 12 PlayStation 3 (PS3s help astrophysicists solve mystery of black hole vibrations, 2008) (PlayStation3 Gravity

Grid). Dos de sus investigadores el Dr. Gaurav Khanna y el Dr. Chris Poulin se encuentran particularmente interesados en los desarrollos en CELL y el PlayStation 3 y han dispuesto una página Web en la que esperan recopilar las experiencias con esta plataforma (ps3cluster.org).

De igual forma, el Dr. Frank Mueller, Ph.D de la Universidad del Estado de Norte de Carolina en Estados Unidos se encuentra investigando y enseñando sobre simulaciones de alto desempeño y una de sus herramientas es un CLUSTER creado a base de consolas PlayStation 3 (Mueller, 2009).

Rene Muller, estudiante de postgrado del instituto federal suizo de tecnología en Zúrich desarrolló una simulación de un péndulo y la ejecutó a través de varios CLUSTERS y computadores individuales evaluando el desempeño de cada uno de ellos. En sus resultados en el primer lugar lo ocupó un CLUSTER de treinta y dos nodos, cada uno con procesador AMD Opheron de doble núcleo a 2.2 GHz, el cual obtuvo un tiempo de ejecución de 10.54 segundos. El segundo lugar lo ocupó un solo un PlayStation 3 con un tiempo de ejecución de 18.39 segundos (Müller, 2005). Nada mal para un solo computador contra treinta y dos. Claro está que el procesador CELL del PlayStation 3 cuenta con 8 procesadores fuertemente acoplados, aunque un programador ajeno a los videojuegos solo tiene acceso a 6 de los 8 SPU y a al PPU o procesador principal de doble núcleo.

Lo ejemplos no paran solamente en investigación científica, Warhawk uno de los primeros juegos para PlayStation 3 usa un CLUSTER inmenso formado por consolas PlayStation 3 para prestar los servicios en línea del juego (Jobe, 2007) (Murph, 2007).

Tanto es el poder del CELL BEA y del PlayStation 3 que la Universidad de Stanford se asoció con SONY para instalar oficialmente la aplicación `foldings@HOME` en el PlayStation 3, permitiendo que la consola en sus tiempos de reposo sea usada para colaborar con el proyecto. Sus aportes fueron tantos que en los primeros días la consola de videojuegos alcanzó el primer puesto en colaboración compitiendo en

tiempo de computadora contra los PCS tradicionales (Stokes J. , Why the PlayStation 3 owns the PC in F@H, 2007).

De esta manera es posible observar cómo se va perfilando el CELL BEA como una de las arquitecturas o modelos de multiprocesamiento del futuro y también cómo su uso se ha ido colando en toda clase de aplicaciones computacionales. El uso del paralelismo, el procesamiento distribuido, CLUSTERS y GRIDS va más allá de cuestiones enteramente científicas y se vislumbra como una tecnología que podría estar presente en toda la industria de la computación, el entretenimiento, la multimedia y las comunicaciones.

De hecho, Sony e IBM no son los únicos detrás de esta tecnología, Toshiba también estuvo involucrada en el desarrollo de la misma y al igual que Sony tiene como objetivo llevar el procesador CELL BEA a muchas instancias, desde consolas de entretenimiento y televisores pasando por computadores de escritorio, hasta servidores de alto rendimiento (Hofstee, 2005) (Miller, 2008).

La estrategia de IBM, Sony y Toshiba es llevar la computación distribuida al hogar, en donde todos los dispositivos electrónicos estarán interconectados y compartirán recursos computacionales en forma de GRIDS y CLUSTERS caseros. El principal obstáculo es la dependencia de sistemas operativos comerciales como Windows y Mac OS que se han convertido en un lastre para la industria del hardware.

14 MARCO TEÓRICO

14.1 DIFERENCIAS ENTRE RISC/POWERPC Y CISC/X86

Como se mencionó anteriormente existen dos vertientes en el mundo del procesamiento, la vertiente RISC común en la arquitectura PowerPC y la vertiente CISC propia de Intel y AMD. La vertiente RISC trata de mantener el chip lo más simple posible, sin incluir instrucciones adicionales o facilidades de programación, esto siempre ha hecho que el chip se desempeñe mejor, consuma menos electricidad y genere menos calor.

La vertiente CISC trata de añadir la mayor cantidad de instrucciones reduciendo la complejidad del código con la desventaja de requerir más electricidad, generar más calor y requerir más ciclos para realizar determinado cálculo o procesar cierta instrucción.

Otra gran diferencia de la arquitectura RISC /PowerPC en comparación con la CISC es que, la arquitectura PowerPC es una arquitectura de carga/almacenamiento (load/store). En esta los cálculos no son realizados en la memoria principal sino en registros a diferencia de la arquitectura x86, donde los cálculos se pueden realizar tanto sobre la memoria principal como sobre los registros. La arquitectura PowerPC busca claramente mantenerse simple y de esta forma mejorar el desempeño, lo cual en vista de la crisis de procesamiento por la que se está atravesando rinde sus frutos (Bartlett, 2006).

La diferencia más notoria desde el punto de vista del programador entre PowerPC y x86 por supuesto son las instrucciones, las cuales son específicas para cada arquitectura, pero estas no forman parte del propósito de este documento.

En el mundo del procesamiento paralelo existen dos tipos de implementaciones, las fuertemente acopladas y las sueltamente acopladas. Los ejemplos de las fuertemente acopladas abundan, en ellos encajan el Cell BEA, el Xenon, el Intel Core 2, los AMD Ophtheron, en fin todos aquellos procesadores que incluyen dos o más unidades de procesamiento, llámense núcleos o coprocesadores en la misma máquina. Todo esto independientemente de si son o no SMP, ASMP, UMA o NUMA.

La implementación sueltamente acoplada es aquella propia de los CLUSTER Beowulf o los GRIDS, en los que dos o más unidades de procesamiento, se acoplan mediante interfaces de red, llámense inalámbricas, Ethernet o fibra óptica.

Esta última implementación tiene una gran ventaja y es que si alguna de las unidades de procesamiento falla, puede ser reemplazada fácilmente. Un CLUSTER Beowulf puede ser fácilmente ampliado tan solo añadiendo computadores genéricos adicionales a la red.

En contraposición hay algunos inconvenientes, como su alto uso de electricidad, espacio, ventilación y enfriamiento. Por otro lado hay un límite muy claro en escalabilidad, ya que las interfaces de red tienen grandes desventajas en cuanto a latencia si se comparan con uniones en BUS dentro de la misma máquina o en el mismo chip, por lo tanto la escalabilidad es limitada.

Las implementaciones fuertemente acopladas tienen por supuesto la ventaja en capacidad de procesamiento, pero usualmente son mucho más costosas y un daño puede significar altas inversiones en reparación.

14.2 LA COMPLEJIDAD EN EL MICROCHIP RISC VS. CISC

El dilema de la navaja suiza es muy conocido, una navaja suiza es muy útil en situaciones de emergencia, pero si se desea realizar algún trabajo profesional se requiere de herramientas profesionales especializadas para tal fin. Los procesadores

se rigen por el mismo principio, pero hasta hace poco los programadores e ingenieros ignoraban esta realidad. Los procesadores se hacían más rápidos y parecía que no iba a tener fin esta tendencia, existía comodidad en su programación ya que no era necesario recurrir a la planeación ni a la optimización de código, labores que realizaban los compiladores.

Como ya se mencionó, esta tendencia en el avance del procesamiento se topó con una imposibilidad práctica para seguir aumentando la eficiencia de los procesadores de la forma en que se venía haciendo.

Aunque parecía que los fabricantes ignoraban este hecho, algunas compañías ya tenían sospechas. David R. Ditzel y TRANSMETA fueron unos de los primeros y decidieron lanzar un procesador muy radical llamado Crusoe. El objetivo de Transmeta y Ditzel era reescribir el significado de RISC. Para ese entonces la diferencia en la fabricación de chips RISC y CISC prácticamente había desaparecido, ambos tenían tantos transistores e instrucciones que ya era irrelevante hablar de RISC o CISC, pero Transmeta apostó fuertemente a volver el concepto RISC a su esencia y dejar la complejidad en el software para mejorar significativamente el rendimiento de los procesadores, reducir su consumo de electricidad y la generación de calor.

El resultado fue un chip muy sencillo que evitaba los principales problemas presentados en ese entonces debido a los mecanismos utilizados para ocultar la latencia, técnicas presentes en la mayoría de los procesadores, lo que incrementaba la complejidad de los integrados, ocurriendo algo muy similar a las primeras redes de computadores en las que la mayor parte del ancho de banda era consumido por información de control para mantener la red en funcionamiento.

En el caso de los chips, esto significa más transistores y por lo tanto mayor área del chip dedicada a cuestiones de control y no de ejecución. Por otro lado significa más generación de calor, aumento en el consumo de electricidad y de los ciclos de procesamiento dedicados a reordenar instrucciones y no a ejecutarlas.

La implementación de Transmeta no fue la mejor en ese entonces, incluso hoy en día la latencia entre la memoria y el procesador es muy alta, por lo que la idea de sacar la complejidad del chip y pasarla al software o a la memoria principal no es tan eficaz después de todo. A pesar de ello la idea de Ditzel y Transmeta no deja de ser correcta, prueba de ello es el procesador Cell BEA de IBM el cual tomó el mismo concepto, pero creó una implementación muy distinta que finalmente dio los resultados esperados (Stokes J. , Introducing the IBM/Sony/Toshiba Cell Processor, 2005).

Claramente incrementar la complejidad dentro del chip no ha dado los resultados esperados a largo plazo y reducir la complejidad por si sola tampoco es efectiva. La solución de hecho se dio inconscientemente desde hace unos años con las tarjetas aceleradores en 3D, las cuales sin ser su objetivo, fueron la primera implementación práctica de una arquitectura de procesamiento asimétrico (ASMP), en la que un chip especializado es dedicado a tareas especializadas, liberando así la carga en el procesador principal. Esto no solo mejora el rendimiento aparente al dedicar la CPU a otras cuestiones, sino que la GPU al ser especializada le permite realizar dichos cálculos mucho más rápido que una CPU.

Hace unos años se lanzó también el primer procesador dedicado a cálculos físicos o PPU, enfocado principalmente a juegos y su objetivo era acelerar los cálculos de partículas, simular de forma más precisa el viento, el fuego y la deformación de materiales. Esta tarjeta aceleradora en un principio no tuvo éxito, pero los diferentes fabricantes de GPU y CPU decidieron tomar un rumbo similar y añadir dicho chip a sus aceleradoras de video o ciertas características a las CPU como el Cell BEA.

La CPU de propósito general seguramente no va a desaparecer, pero los hechos mencionados anteriormente muestran que en el futuro se verán más procesadores especializados en tareas como decodificación y codificación de audio y video, encriptación, inteligencia artificial procesamiento de señales para captura de voz y reconocimiento óptico, incluso para cuestiones matemáticas avanzadas, acelerando cálculos científicos y se hará más evidente la arquitectura ASMP.

En el escenario planteado anteriormente es donde entran a jugar los CLUSTERS y GRIDS. La implementación ASMP y su modelo de programación es muy similar a ellos, ya que se requieren las habilidades para hacer un uso colaborativo paralelo de todas las unidades de procesamiento disponibles a través de una red de computadores.

De hecho el procesador Cell BEA del PlayStation 3 ha sido catalogado como una "red en un chip" ya que cada SPE o elemento de procesamiento sinérgico puede ser considerado como un procesador vectorial independiente (Stokes J. , *Introducing the IBM/Sony/Toshiba Cell Processor*, 2005) con su propia memoria local, o como se le llama "almacenamiento local", todos interconectados a través de un bus circular de cuatro canales en alguna especie de red en anillos redundante.

14.3 TIPOS DE MULTIPROCESAMIENTO

Contrario a la lógica, los sistemas de procesamiento asimétricos ASMP no fueron los primeros en existir. Si bien se podría pensar que si no se tiene suficiente poder con un procesador, se podría simplemente partir el trabajo y usar otro sin tener que desarrollar un microchip con dos procesadores inmersos, aún así, los sistemas ASMP fueron posteriores a los sistemas SMP.

El primer indicio de procesamiento paralelo simétrico ocurrió en 1961 a manos de Burroughs en su B5500, el cual usaba una arquitectura de multiprocesamiento simétrico o SMP (Wilson, 1994), previo a los estudios en procesamiento asimétrico (ASMP) a manos del MIT y DEC.

Pero, ¿de qué trata específicamente cada uno de estos estilos de procesamiento en paralelo? ¿Que los hace distintos? ¿Cual es mejor?

SMP es el acrónimo en inglés de Multiprocesamiento Simétrico (Symetric Multi Processing), de otro lado, ASMP es acrónimo de Multiprocesamiento Asimétrico (Asymetric Multi Processing).

Su definición puede engañar un poco, sin embargo poseen ciertas similitudes. Desde el punto de vista del desarrollador, ambos no son muy distintos de programar si se requiere aprovechar todo su rendimiento, pero cada uno tiene unas ventajas y desventajas que son mutuamente excluyentes.

Es por eso que no es posible afirmar que una arquitectura es superior a la otra, ya que las ventajas de cada uno pueden atraer distintos tipos de desarrollos y limitar otros, así que todo se resume en una decisión subjetiva en la mayoría de los casos.

14.3.1 Multiprocesamiento simétrico o SMP.

El modelo SMP al que de ahora en adelante se llamará "simétrico", pone a disposición de los programadores múltiples "núcleos" de procesamiento o "ALU" (Arithmetical Logical Unit) inmersos en un solo procesador.

Dicho de forma simple, son por lo menos dos procesadores en uno, en los que a diferencia de los ASMP ("asimétricos" de ahora en adelante) cada unidad lógica (núcleo), no cuenta con memoria propia. Esto significa que todos los núcleos comparten tanto la memoria principal como la memoria cache.

Por otro lado, las tareas son repartidas entre los diferentes núcleos por el sistema operativo, más específicamente por su kernel y su gestor de tareas y calendarización. El sistema operativo puede perfectamente pasar la ejecución de una tarea (o hilo para usar el término técnico apropiado) de un núcleo a otro, pero pueden ocurrir circunstancias de pérdida de memoria ya que el espacio, tanto en la cache como en la memoria principal, es el mismo, trayendo esto sus ventajas y desventajas.

Desde el punto de vista del programador tradicional es un alivio el no tener que lidiar con diferentes espacios de memoria al mismo tiempo, dejando en manos del sistema operativo el manejo y asignación de los hilos.

Desde el punto de vista del usuario es una mejora significativa, ya que aún si el software no está diseñado para ejecutarse de forma paralela, el sistema operativo puede perfectamente asignar un núcleo a determinado programa, como por ejemplo una aplicación ofimática y usar el otro núcleo para otra tarea, tal como codificar un CD de música en MP3 o MP4. Esto da la sensación de que la computadora es más rápida aún si cada procesador no ha evolucionado o mejorado demasiado con respecto a su generación anterior.

Desde el punto de vista del fabricante comienzan a verse las desventajas, ya que durante el proceso de elaboración o impresión de chips, siempre ocurren defectos en las unidades de fabricación, estos chips al contener más de una unidad lógica aritmética tienen mayor probabilidad de fallar, así que con una sola unidad o núcleo defectuoso se descarta el integrado completo.

Los defectos o inconvenientes desde el punto de vista del rendimiento son más significativos. Un chip simétrico puede ser considerado como un monstruo de dos o más cabezas independientes. Pero ¿Qué ocurriría si uno de ellos quiere ir en una dirección y otro en una diferente?

Los chips SMP al compartir tanto la memoria principal como la cache y el bus pueden entrar en conflicto o caer en retardos al intentar acceder simultáneamente al mismo recurso y esto normalmente ocurre o se hace más notorio en los picos de procesamiento.

El BUS por su parte se convierte en un cuello de botella a la hora de tratar de extraer hasta la última instrucción por segundo, por otro lado la coherencia de la memoria puede verse afectada si algún hilo cambia su ejecución de núcleo y hace uso de un espacio de memoria diferente. Algunos procesadores tratan de corregir esto al segmentar la memoria, usualmente la cache pero esto reduce el tamaño útil de cada segmento.

El programar cada núcleo manualmente para mantener la coherencia de los datos y extraer hasta la última instrucción por segundo, se ha hecho notorio en el desarrollo de juegos para las consolas actuales, donde los desarrolladores se han dado cuenta que es más eficiente desarrollar primero los juegos para PlayStation 3 cuyo chip es asimétrico y posteriormente portarlo a Xbox 360 con su chip simétrico, haciendo inmediatamente eficiente el código para la consola con chip simétrico y no al contrario como ocurría en un principio, lo cual daba como resultados experiencias y calidades desiguales entre ambas consolas y costos de desarrollo elevados.

14.3.2 Multiprocesamiento asimétrico o ASMP.

Los chips asimétricos por su parte se parecen más a un CLUSTER con interconexión LAN, con la excepción de ser "fuertemente acoplados" o lo que es lo mismo, que residen en una sola maquina. Cada chip de un procesador asimétrico tiene su espacio de memoria propia y la diferencia más notable es que hay jerarquías de procesadores.

Aunque este estilo de multiprocesamiento nació en los 70, fue olvidado casi de inmediato junto con en el SMP, debido a los avances en fabricación de microchips que permitieron elevar la frecuencia y la densidad de los transistores.

Sin embargo, la industria ha venido usando inconscientemente un estilo asimétrico desde hace unos años con los chips aceleradores de video en 2D y 3D o GPU. Cuando se comenzaron a incluir estos en los computadores, los programadores tenían básicamente que programar paralelamente lo cual resultaba incómodo. No es una forma fácil de programar, pero en vista del aumento tan significativo que esto trajo al procesamiento de gráficos al usar de un chip enteramente dedicado a ellos, se volvió una norma y los programadores tuvieron que adaptarse.

En este estilo de procesamiento paralelo hay un procesador principal de propósito general que está en la cima de la pirámide jerárquica y se encuentra rodeado de coprocesadores, en algunos casos especializados para tareas específicas. En ellos

el cuello de botella es el procesador principal y cuántas instrucciones puede despachar a sus coprocesadores.

El hecho de que cada procesador tenga su espacio de memoria propio evita las incoherencias de memoria, ya que le permite dedicar sus ciclos completos al cálculo de su instrucción sin preocuparse por competir por recursos. Por otro lado cada coprocesador al ser especializado en determinada tarea, está en capacidad de realizar dicho cálculo más rápido que un procesador de propósito general.

Por otra parte, un chip asimétrico puede ser en teoría más eficiente eléctricamente al solo usar los coprocesadores necesarios para determinada tarea, apagando aquellos que no están en uso.

Los chips asimétricos igualmente tienen su cuota de problemas, al igual que los SMP tienen un cuello de botella y es el procesador de jerarquía más alta. La programación por defecto es compleja, al tener el programador que hacer uso en la mayoría de los casos de los coprocesadores de forma directa. Al mismo tiempo sufren de los inconvenientes de fabricación de los SMP al ser procesadores fuertemente acoplados, si falla la impresión de uno se descarta todo el integrado.

Los chips asimétricos traen grandes ventajas en cuanto a desempeño a diferencia de los chips simétricos, pero su carga recae en los desarrolladores y programadores. A largo plazo parecen ser la solución a la crisis en que se encuentra la industria. Tanto es así que hay tres plataformas que usan este concepto para sus productos, una es el Cell BEA de IBM usado por Sony en su PlayStation 3 y por IBM en sus servidores.

Actualmente el record de procesamiento lo tiene el ya mencionado Roadrunner el cual usa el mismo chip de la consola de videojuegos con algunas diferencias. La otra plataforma es el Zii de Creative, una plataforma de computación apodada de "células madres" (Stemcell) la cual recién salió a la luz en el 2009 y promete dar un salto exponencial en el procesamiento, al usar pequeñas unidades de procesamiento

programables (similares a las GPUs) las cuales se agrupan y trabajan en conjunto (como células vivientes) para realizar determinada tarea. Finalmente ARM planea una arquitectura asimétrica para su uso en dispositivos móviles.

14.4 TIPOS DE PARALELISMO SIMD Y MIMD

Resulta muy difícil hablar enteramente de procesadores y procesamiento sin hablar de software. Un fabricante podría hablar de innumerables características que pueden no tener relevancia respecto a lo que se quiere ejecutar sobre él.

La supercomputación normalmente se encuentra caracterizada por manejar grandes volúmenes de datos y una cantidad limitada de instrucciones que operan dichos datos, encajando naturalmente en la técnica SIMD (SIMPLE INSTRUCTION MULTIPLE DATA) de paralelismo, a diferencia de otros tipos de computación, principalmente de escritorio en el que priman un gran número de instrucciones que operan sobre una cantidad limitada de datos.

Aplicaciones como un editor de texto por ejemplo, tienen gran cantidad de opciones de formato (instrucciones) que operan sobre una cantidad casi estática de texto (datos). La técnica SIMD fue muy común durante los años 80 en el campo de la supercomputación, pero el interés se desvaneció con el avance de los procesadores de propósito general MIMD (MULTIPLE INSTRUCTION MULTIPLE DATA) de bajo costo.

La computación desde ese entonces ha evolucionado, los computadores ya no son máquinas enteramente de escritorio, ofimáticas, aisladas y como ya se ha mencionado, la multimedia y el entretenimiento se han ido colando en ella. Este tipo de aplicaciones cambiaría la forma en que se diseñan y fabrican los procesadores (Diefendorff & Dubey, 1997).

La MIMD no es necesariamente mejor que la SIMD solo por el hecho aparente de que esta maneja múltiples instrucciones a diferencia de la SIMD. Toda decisión que se toma sobre un procesador afecta notablemente la eficiencia de ciertos tipos de aplicaciones que se ejecutan sobre él, lo que en algunos casos aparenta ser negativo, pero todo se resume nuevamente a lo que se quiera realizar.

Los fabricantes de procesadores conocen esta realidad y decidieron añadir ciertas extensiones en sus procesadores para resolver este dilema y mantenerse al día.

IBM con su ALTIVEC, Intel con su MMX y AMD con su 3Dnow, trataron precisamente de añadir facilidades SIMD a sus procesadores MIMD, dándoles de esta forma a los programadores acceso a un rendimiento ligeramente cercano a procesadores SIMD, en sus procesadores MIMD para determinadas aplicaciones. Otro intento de resolver este dilema técnico fue el nacimiento de las tarjetas aceleradoras de video las cuales son procesadores SIMD por naturaleza. De hecho las tarjetas aceleradoras de video son tan potentes para cálculos vectoriales y otras aplicaciones SIMD que paralelamente al estudio de CLUSTERS con PlayStation 3 se plantean CLUSTERS con solo tarjetas aceleradoras de video en 3D para ciertos cálculos científicos.(Stokes J. , Arstechnica, 2000) (Stewart, 2005)

Como se mencionó anteriormente el interés por chips SIMD cayó un tanto en desuso, pero con el advenimiento de los videojuegos en tres dimensiones y otras aplicaciones de tiempo real y científica, se ha retomado el interés en este tipo de chips, principalmente en el campo de entretenimiento, el cual se ha convertido en una mina de oro para las aplicaciones científicas.

Hay una discusión que todavía perdura sobre cual procesador es más potente, el Xenon del Xbox 360 o el Cell BEA del PlayStation 3. Aunque ambos son sistemas diferentes tienen bastantes similitudes, ambos son fabricados por IBM, ambos son de arquitectura RISC PowerPC, pero de hecho el Xenon es básicamente un derivado del PPU (procesador principal) de CELL BEA y el CELL BEA además de el PPU tiene 8 unidades de coprocesadores sinérgicos o SPU, como los llama IBM.

Aunque el Xenon brinda mayor capacidad de procesamiento general MIMD, el CELL BEA con su PPU de doble núcleo a 3.2 GHz y 8 coprocesadores SIMD también a la misma frecuencia, brinda una avalancha de procesamiento dedicado a aplicaciones de multimedia y científica.

El Xenon por supuesto no interesa en este documento por cuanto la arquitectura del Xbox 360 no permite el acceso a su programación sin autorización plena de Microsoft y sus herramientas oficiales, pero es inevitable al discutir sobre una consola y no entrar en comparaciones con otra sobre todo siendo de la misma generación y siendo implementaciones diferentes de conceptos similares. (Brown, 2005)

14.5 PROCESADOR CELL BEA

14.5.1 Arquitectura.

Al chip Cell BEA se le ha apodado como “red en un chip”(Kahle, Day, Hofstee, Johns, Maeurer, & Shippy, 2005). En este siete de ocho unidades coprocesamiento o SPU se encuentran interconectadas en un bus de cuatro direcciones al procesador principal o PPU, tal como se observa en la figura 1.

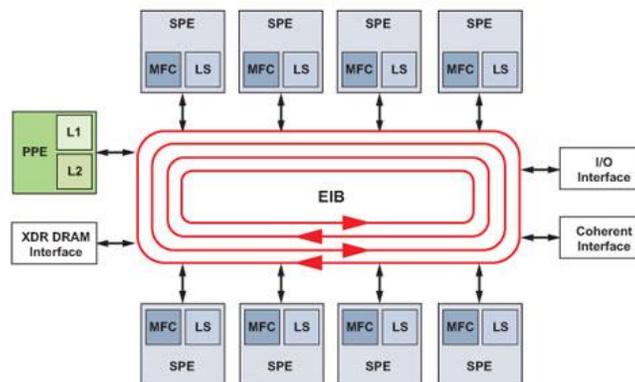


Figura 1. Arquitectura Cell BEA

Cada SPU está en capacidad de ejecutar un hilo en determinado momento, mientras que el PPU de doble núcleo puede soportar dos hilos de hardware simultáneamente. Aunque el PPU puede ser usado como un procesador de propósito general su

verdadera función es sincronizar y asignar tareas a cada SPU. El verdadero poder del Cell BEA claramente está en los ocho SPU.

El Cell BEA en cuanto a acceso a memoria es de cierta forma híbrido, tiene una memoria principal, en el caso de la versión para PlayStation 3 es una memoria de tecnología XDR de 256MB y posee una memoria de video de 256MB adicionales. Aunque están separados, el PlayStation 3 puede usar la memoria de video como memoria RAM adicional.

En el acceso UMA se cuenta solamente con una memoria principal que todos los núcleos comparten y en la NUMA cada núcleo o procesador posee su propio espacio de memoria. En el caso de los SPU estos no pueden acceder a la memoria principal, pero cada SPU tiene su propio espacio de almacenamiento inmerso en su área de ejecución, el cual usan para guardar sus datos temporalmente y procesarlos para que luego el procesador principal los pase a la memoria principal. Por esto el Cell BEA usa simultáneamente UMA y NUMA.

Esta característica precisamente ha llevado a plantear que los chips UMA/SMP tendrán serias limitaciones en corto tiempo. En el caso de los chips de múltiples núcleos SMP, todos los núcleos comparten la memoria caché y a medida que pase el tiempo y se incrementen los núcleos, la cache supone un cuello de botella significativo no solo por el espacio sino por el acceso a esta.

En el caso del Xenon del Xbox 360 se implementó una medida muy peculiar y fue la de dividir la memoria caché en segmentos, de esta forma se le da cierta independencia a cada núcleo. Sin embargo, esto tiene un efecto contraproducente que es la reducción de la cantidad de memoria caché disponible para cada núcleo. Los procesadores ASMP por supuesto no cuentan con este problema ya sean NUMA o híbridos como el Cell, con la desventaja de incrementar la dificultad para su programación. (Brown, 2005)

El hecho de usar memoria local para cada unidad de procesamiento o SPU tiene un doble propósito. La velocidad en la que avanza el procesamiento no es la misma que la velocidad de acceso a la memoria, aun hoy con los nuevos tipos de RAM, este continua siendo el principal cuello de botella.

Los fabricantes viene usando el concepto de la memoria caché principalmente para reducir u ocultar un poco la latencia que implica el acceder a la memoria principal, en vez de ello, usan un espacio de almacenamiento local o caché que aunque es pequeño es bastante rápido al estar tan cerca del procesador y en él se almacenan datos basados en predicciones estadísticas y lógicas que buscan encontrar dependencias y mantener el procesador ocupado.

Por supuesto, las predicciones no siempre son certeras, lo que implica no solo vaciar la memoria caché sino buscar nuevamente instrucciones para solucionar la dependencia lo que al mismo tiempo significa acceder a la memoria principal causando caídas en la eficiencia de procesamiento (Stokes J. , Introducing the IBM/Sony/Toshiba Cell Processor, 2005).

En el caso del Cell BEA se decidió eliminar esto para simplificar enormemente el procesador, ya que el área dedicada al reordenamiento es bastante significativa, de esta forma se mejorará su rendimiento. La responsabilidad del ordenamiento de las instrucciones recae en los programadores, los calendarizadores y compiladores especializados que reordenan el código antes de ejecutarlo; dicho de forma simple “limpian el desastre de los programadores” el cual se hace más evidente debido a la implementación asimétrica del Cell BEA.

Proyectos como el CELL Superescalar del instituto de supercomputación de Barcelona buscan desarrollar compiladores que eliminen la necesidad de reordenar las instrucciones por parte de los programadores y de hecho este proyecto busca que programar el Cell BEA como si se tratase de un procesador de un solo núcleo.

El bus que interconecta las unidades de ejecución se encuentra en la misma oblea del chip a diferencia de otros sistemas ASMP, esto incrementa la eficiencia al estar más cerca los unos de los otros.

A diferencia de otros sistemas el Cell posee un chip de red dedicado, por lo que la transmisión de grandes volúmenes de datos por la red no consume los recursos de procesamiento, haciendo este sistema más responsivo al reducir la latencia asociada con el manejo de las tramas de red.

Su diseño es modular por lo que es muy adaptable y se pueden hacer fácilmente modificaciones para adaptarlo a diferentes tipos de aplicaciones y sistemas, ya sean consolas de videojuegos, televisores, centros de entretenimiento o supercomputadores.

El chip Cell BEA es capaz de auto verificarse y sus SPU son configurables, lo que ha resultado muy útil para establecer protocolos de encriptación para protección de derechos de autor, lo cual es obligatorio para los usos en entretenimiento como música y video.

El Cell BEA por otra parte usa registros de 128 bits, aunque por supuesto no existen instrucciones de 128 bits hasta ahora, pero esto significa que en un solo ciclo de procesador se pueden ejecutar simultáneamente desde dos instrucciones de 64 bits, hasta cuatro instrucciones de 32 bits.

14.5.2 Modelos de Programación del Cell BEA.

En el Cell BEA cada hilo es ejecutado en un SPE y los hilos se pueden comunicar entre sí de la misma forma que se hace en sistemas SMP. La programación multihilo es la forma más tradicional y de programar el Cell BEA pero no es la única y puede ser usada en conjunto con otras técnicas como las que se mencionan a continuación.

14.5.2.1 Modelo de delegación.

La forma más rápida de programar el Cell BEA es con el modelo de delegación de funciones. En este modelo se puede hacer uso de un programa tradicional diseñado para un procesador de un solo núcleo, sin requerir grandes modificaciones.

Este se ejecuta en el PPE pero se delegan ciertos procesos, funciones o métodos a uno o más SPE. Este modelo es el objetivo de "CELL Superescalar" el cual busca que el compilador genere el código libre de dependencias y ramificaciones, usando un solo código y quitando de la mano del programador la necesidad de definir qué procesos se deben correr en el PPU y cuáles en los SPU al usar código fuente y compilación separada para el PPU y los SPU.

14.5.2.2 Modelo de extensión del dispositivo.

El modelo de extensión de dispositivos es muy similar al de delegación, solo que cada SPE se usa como interfaz con un dispositivo externo y se usan las facilidades del motor DMA para mapear la memoria.

Este modelo es usado principalmente para el manejo de protección de propiedad intelectual de contenido multimedia. En estos usos una porción del almacenamiento local de cada SPU es usada para la comunicación entre los SPE y el PPU (o los PPU, el CELL BEA puede en teoría usar más de un PPU), el resto de la memoria es aislada totalmente del sistema mediante hardware y solo el código que se diseñó para leerlo y que está también aislado en el SPU puede hacerlo, creando una especie de bóveda.

La única forma para que el resto del sistema acceda nuevamente a esta porción de memoria, es enviando un mensaje de terminación con el que no solo se finaliza el proceso, sino que también la memoria es totalmente

borrada, esto deja sin utilidad las técnicas tradicionales para acceder de forma ilegal al contenido intelectual protegido, como lo es el volcado de memoria.

14.5.2.3 Modelo de aceleración computacional.

Es una forma de explotar mas “granularmente” el poder de procesamiento, esto se logra ejecutando las porciones de código computacionalmente más intensivas en los SPU y el PPU simplemente actúa como un dispositivo de control y despacho.

La paralelización se puede lograr mediante compiladores como también por parte del programador al calendarizar las operaciones DMA y el movimiento de datos desde y hacia los SPU.

14.5.2.4 Modelo de flujo o “streaming”.

El modelo de flujo o “streaming” puede ser usado con un conjunto de instrucciones que se ejecutan en paralelo en las que los SPU aplican cierto proceso computacional a los flujos de datos.

En este, la sincronización se logra mediante el uso del intercambio de mensajes estándar entre procesadores lo cual se encuentra totalmente soportado por el CELL BEA (Kahle, Day, Hofstee, Johns, Maeurer, & Shippy, 2005) (Hofstee, Cell Broadband Engine Architecture from 20,000 feet, 2005) .

14.6 MALLAS Y RACIMOS DE COMPUTADORES (CLUSTERS Y GRIDS)

Aunque el concepto de CLUSTER está claramente definido y su existencia no es reciente, el concepto de computación en malla o GRID COMPUTING por el contrario más nuevo, siendo su definición más simple la de un CLUSTER a escala geográfica o global,

Un CLUSTER es como ya se mencionó un conjunto de computadores locales interconectados entre sí que trabajan en conjunto, cuya función es suplir la falta de poder computacional o poder resolver determinados problemas computacionales más rápido y eficientemente.

Estos problemas pueden ir desde el manejo de grandes cantidades de carga mediante el balanceo de la misma, proveer gran disponibilidad resistente a fallos o ser una solución de procesamiento paralelo. Claro que para que un CLUSTER sea "exitoso" se debe contar, no solo con computadores sino con interfaces de alta velocidad para reducir la latencia a términos aceptables y protocolos e interfaces comunes entre todos los nodos que lo conforman.

Un GRID aunque comparte ciertas características de un CLUSTER y puede buscar resolver los mismos problemas, es un concepto más ambicioso y por lo tanto más complejo, siendo su alcance mayor, por esto puede resolver otros problemas que se salen del alcance de un CLUSTER.

Ian Foster y Karl Kesselman definen un GRID como un sistema que coordina recursos distribuidos, mediante el uso de protocolos e interfaces estandarizadas abiertas y de propósitos generales, para entregar calidad de servicio no trivial (Foster & Kesselman, THE GRID: BLUEPRINT FOR A NEW COMPUTING INFRASTRUCTURE, 2003). Este concepto se desprende un poco de la filosofía del software libre, en cuanto a que se parte de que el conocimiento es de toda la humanidad desestimando cualquier herramienta o protocolo propietario de código fuente cerrado, que busque lograr los propósitos del GRID pero de forma privada o restringida.

No debe extrañar entonces el uso de herramientas como GNU/LINUX, GLOBUS, CONDOR entre otras, en la mayoría de organizaciones en las que se haga uso de GRIDS o CLUSTERS BEOULF.

Ian Foster y Karl Kesselman reconocen esta realidad y de hecho lideraron la creación de la OGF la cual busca precisamente definir estándares mediante un consenso, mas no mediante imposiciones de un sector o un grupo particular y desarrollaron con base en estos estándares la tercera versión de su Globus Toolkit, una serie de herramientas para la implementación de GRIDS que de acuerdo a ellos mismos, se había convertido en un estándar de facto que a pesar de ser abierto no fue definido mediante un consenso general.

Aunque se podría inferir instantáneamente que los componentes obvios de un CLUSTER y un GRID son PCs, de hecho como se ha tratado de dejar claro a lo largo del proyecto, existen sistemas compuestos por elementos exóticos como consolas de videojuegos e incluso tarjetas aceleradoras de video en 3D o GPUS, que han demostrado sobrepasar en varios órdenes de magnitud el rendimiento de un procesador típico.

Los componentes de un CLUSTER y un GRID computacional fuera de los obvios, son principalmente elementos de software, ya que la forma en la que estos se interconectan no difieren de cualquier otro tipo de red de computadores, de hecho se podría usar tecnología WIFI, incluso existen pruebas de concepto de GRIDS en redes celulares (Wesner, Jähnert, & Escudero) pero por supuesto las particularidades de estos restringirían el tipo de aplicaciones principalmente por la latencia inherente a este tipo de enlaces.

Una de las características principales de un CLUSTER es su jerarquía maestro/esclavo. Un CLUSTER visto desde el exterior es identificado como una sola máquina independientemente de que cuente con un nodo o cien nodos.

Las tareas son iniciadas en el nodo principal o maestro, pero los "componentes" de software y el código del mismo programa permiten la ejecución en todos o algunos nodos de acuerdo al diseño del algoritmo y a la asignación de recursos.

Aunque uno de los requerimientos de un CLUSTER ideal es que esté conformado exactamente por los mismos componentes a través de todos sus nodos, esto no siempre es la realidad. El concepto del CLUSTER Beowulf es precisamente el poder usar nodos compuestos por PCs genéricos que se pueden ir agregando de acuerdo al crecimiento de las necesidades y aunque se pueden usar PCs con variaciones en sus características, se deben tener en cuenta las consideraciones de memoria e interfaces de red puesto que un solo nodo podría retrasar la ejecución de determinado algoritmo o cálculo.

14.6.1 Administradores de carga y recursos.

Como se mencionó previamente, una de las tareas principales al realizar procesamiento en paralelo es la distribución de cargas y recursos, contando actualmente con ciertas herramientas que facilitan el desarrollo de estas tareas. A continuación se relacionan algunas de ellas.

14.6.1.1 XGrid.

Existen sistemas como XGrid de Apple que permite la creación instantánea y automática de un CLUSTER o un GRID tan solo activando una casilla en el panel de preferencias (Mac OS X) sin necesidad de instalar software adicional.

XGrid, aunque es un sistema propietario, está basado parcialmente en estándares y de hecho la aplicación cliente de XGrid puede ser instalada en sistemas Unix y GNU/Linux, aunque sin las facilidades de configuración de un solo clic de Mac OS X, pero sí con las facilidades que brinda el protocolo de autodescubrimiento "Bonjour" de cero configuración.

XGrid maneja principalmente procesamiento en lotes o "batch" o como se le conoce popular y despectivamente, tareas embarzosamente paralelizables (Apple) por cuanto dichas tareas no son ejecutadas en tiempo real y pueden ser consideradas como tareas "ligeras" al no depender de otros resultados como la codificación de audio y video, renderización entre otros. Pero esto no

es debido a XGrid ni a Apple ni a Mac OS X, es una característica propia de estos tipos de herramientas al igual que Cándor o Sun Grid Engine, con la diferencia que en estas herramientas Apple busca llevar el procesamiento distribuido a las manos de casi cualquier persona y facilitar su uso.

No sorprende entonces ver con tanta frecuencia computadores Apple en muchos centros de investigación alrededor del mundo, de hecho hay un GRID global de computadores Macintosh (dgohara, 2007). XGrid cuenta con una API propia accesible solo mediante las API de Cocoa y el lenguaje C-Objetivo para el desarrollo de programas que puedan hacer uso del procesamiento distribuido pero al mismo tiempo soporta API estándar en el mundo Unix como lo son MPI y sus derivados como OpenMPI MPICH. (Cote)

El fuerte principal de XGrid es la automatización y la facilidad de uso. No se requieren grandes conocimientos para poner en marcha un CLUSTER o un GRID ligero, ni para su administración ya que la facilidad de uso de Apple se imprime en las herramientas administrativas de XGRID.

XGrid también cuenta, al igual que Cándor, con el tan aclamado hurgamiento de ciclos (CYCLE SCAVENGING), mediante un DAEMON que permanentemente está verificando el estado de la máquina para integrarlo en el GRID o CLUSTER en sus tiempos de ocio.

14.6.1.2 Cándor.

Condor es una herramienta de administración de carga computacional que implementa mecanismos y políticas para soportar la computación de alto desempeño sobre amplias colecciones de recursos distribuidos (CONDOR Team).

Es como ya se mencionó, una herramienta similar a XGrid pero de carácter abierto desarrollada en la Universidad del Estado de Wisconsin en Madison.

Al igual que XGrid su fuerte es el procesamiento en lotes y también usa un DAEMON que se encarga de verificar el uso de determinados computadores o nodos, para de esta forma determinar cuándo este se encuentra desocupado para asignar determinados trabajos.

Maneja mecanismos que permiten la resistencia y aislamiento de fallos, los cuales sólo afectarían el proceso que se estaba ejecutando al momento del fallo y puede reasignar automáticamente la tarea a otro nodo para su ejecución. Al igual que la gran mayoría de estas herramientas soporta el uso de PVM y MPI para su uso en CLUSTERS y trabajos de ejecución en tiempo real.

Respecto a su uso en GRIDS, el objetivo principal de Cóndor siempre fue los CLUSTERS de computadores, pero se puede reemplazar el administrador de trabajos estándar por el Cóndor-G el cual se encuentra basado en los protocolos definidos por el Globus Toolkit. De esta forma, un CLUSTER bajo Cóndor pueden iniciar la ejecución de trabajos en un GRID GLOBUS por fuera del dominio de este.

Por otro lado, si determinado usuario en un GRID quiere ejecutar tareas en un CLUSTER Cóndor, debe hacerlo mediante las interfaces que provee Globus Toolkit. Cóndor también puede hacer uso de una técnica llamada “flocking” la cual permite ejecutar trabajos en más de una piscina (Pool) de procesamiento pero solo se puede hacer uso de esta técnica si no se puede ejecutar determinado trabajo en la piscina desde donde este se ejecutó.

Cóndor también tiene su propia implementación de la especificación DRMAA la cual es una API creada por la OGF para la ejecución y control de trabajos

en sistemas de recursos distribuidos en una arquitectura GRID (CONDOR Team) (CONDOR Team) (CONDOR Team).

14.7 COMPONENTES DE UN RACIMO DE COMPUTADORES SIMPLE

Aunque herramientas como XGrid y Cónдор son bastante útiles para CLUSTERS en entornos corporativos o científicos, donde se quiere hacer uso de equipos con alto tiempo de ocio, son herramientas que pueden complicar el estudio de la computación y programación distribuída ya que añaden nuevos niveles de complejidad y no aportan conocimientos nuevos sobre cómo dividir tareas de tal forma que puedan ser ejecutadas de forma distribuída ya sea en lotes o paralelamente.

Como se ha visto, estas herramientas al más bajo nivel todas soportan la especificación MPI, la cual se ha vuelto el común denominador de los sistemas distribuidos y debe ser de hecho el punto de partida para un estudio más profundo, junto con los hilos de ejecución y otras técnicas de procesamiento distribuido.

Un CLUSTER puede ser creado de forma rápida y en poco tiempo mediante el uso de ciertas herramientas simples y abiertas como las que se describen a continuación.

14.7.1 Autenticación y sesión remota por SSH.

SSH es un protocolo de autenticación y de sesión remota, totalmente necesario para los sistemas de computación distribuida, llámense CLUSTER o GRIDS.

Hay que recordar que en estos casos se está trabajando con sistemas que están separados físicamente y dispersos geográficamente, por ello se requiere de un mecanismo que permita el acceso instantáneo y automático, pero solo a aquellos a los que se les pretende dar tal. SSH no es el único mecanismo, pero la tendencia es convertirse en este por cuanto se desarrolló precisamente como reemplazo de otros programas o protocolos usados para el mismo fin, pero que no eran seguros ni eficientes como RLOGIN, RSH y TELNET.

SSH u OpenSSH requieren de dos componentes, un componente cliente y otro servidor, la autenticación mediante llaves públicas es usada para encriptar los datos y la información de sesión la cual puede ser enviada por red o puede ser almacenada en archivos e intercambiada de forma privada.

El intercambio de estas llaves permite el inicio de sesión remota sin la necesidad del uso de contraseñas privadas al almacenar dichas llaves en uno de los archivos de configuración de llaves autorizadas, permitiendo el acceso inmediato en la marcha. Aunque también se puede hacer uso del SSH mediante usuarios y contraseñas en sistemas CLUSTER es necesario el poder establecer sesiones remotas automáticas y transparentes.

14.7.2 Interfaces de programación de aplicaciones MPI y PVM.

Algunas personas podrían no estar de acuerdo con el hecho de presentar estas dos API en una misma categoría ya que se ha argumentado y discutido bastante sobre el hecho de que cada una de hecho trata de resolver problemas distintos (Gropp & Lusk). Pero lo cierto es que ambas están enfocadas en lograr un mismo objetivo que es el diseño y ejecución de programas paralelos “portables”.

A pesar de tener similitudes y de que ambas pueden abordar algunos problemas similares, cada uno se especializa en diferentes problemas inherentes a este tipo de programación. PVM fue desarrollado en 1989 en laboratorio nacional Oak Ridge en conjunto entre la Universidad de Tennessee en Knoxville y la Emory de Atlanta Georgia Estados Unidos como respuesta a la necesidad de crear CLUSTERS compuestos por sistemas heterogéneos y rápidamente se perfiló como un estándar de facto debido a que no había otro sistema similar.

Los problemas de PVM se hicieron notables posteriormente ya que los programas paralelos diseñados bajo este sistema no eran tan rápidos y eficientes como otras implementaciones propietarias de determinados fabricantes de hardware. Debido a esto la industria se vio en la necesidad de crear un estándar para evitar la

proliferación de diferentes sistemas incompatibles entre sí, lo cual iría en perjuicio de los desarrollos sobre estos sistemas. Gracias a esto nació la especificación MPI de la cual se han desprendido varias implementaciones pero que de hecho comparten la misma especificación.

A pesar del desarrollo de MPI, PVM no cayó en desuso no solo por su base de usuarios fieles sino porque que ambas tomaban enfoques diferentes. En la actualidad existen proyectos para unificar ambos sistemas en lo que se llamaría PVMPI y sería una solución más completa pero todavía no hay resultados concretos.

Las ventajas más notables de PVM son su simplicidad al tener tan solo 38 funciones y el hecho de que un programa escrito mediante PVM puede ser ejecutado en un CLUSTER con nodos de diferentes arquitecturas.

MPI por otro lado tiene 130 funciones y se requiere que todos los nodos de un CLUSTER sean homogéneos lo cual es más cercano al concepto de CLUSTERS. MPI por otro lado provee mejor desempeño ya que este fue diseñado teniendo en cuenta su ejecución en verdaderos supercomputadores más no en CLUSTERS Beowulf y migrar el código de estos a los primeros sería casi transparente.

MPI a pesar de contar con mas funciones (130) ha mostrado que de hecho facilita el desarrollo de aplicaciones con menos código que un equivalente en PVM (Manrique, 2002).

14.7.3 Objetos distribuidos, CORBA, COM, PDO.

Sistemas como PVM y MPI no son la única forma de desarrollar programas que interactúen y colaboren remotamente. Casi paralelamente al desarrollo de PVM se crearon especificaciones para el desarrollo de objetos distribuidos capaces de interactuar remotamente a través de la red como la especificación estándar "multilenguaje" CORBA, COM y DCOM de Microsoft, el mecanismo PDO de C-Objetivo (lenguaje usado por NextSTEP Apple GNUSTEP) y mas recientemente JavaSpaces y RMI para Java, también cabe mencionar Pyro para Python, DDObjets

para Delphi y Distributed Ruby para el lenguaje Ruby. En el caso de COM y DCOM han caído en desuso y han sido reemplazados por las facilidades que ofrece Punto NET.

Es de especial mención PDO del lenguaje C-Objetivo el cual permite la creación casi transparente de objetos distribuidos en red casi de la misma forma que se crean objetos locales reduciendo enormemente la cantidad de código que debe ser escrito en algunos casos hasta un décimo (1/10) del código necesario en otros sistemas de objetos distribuidos como los que se mencionó anteriormente (Singh, 2001) (Prabhakar, 1995)(Valdés, 1994).

Por supuesto los mecanismos de objetos distribuidos no están enfocados a la computación de alto desempeño y computación distribuida pero podrían proveer una solución mucho mas simple para ciertos problemas que podrían llevar a pensar que requieren del despliegue de un CLUSTER o un GRID.

14.7.4 Compartir archivos mediante NFS.

NFS es un protocolo de sistemas de archivos por red desarrollado originalmente por Sun Microsystems en 1984 y estandarizado por la IETF similar a otros sistemas cerrados más populares como SMB o CIFS de Microsoft, AFP de Apple entre otros no tan conocidos como AFS, CODA, DCE, FAL, SSF, NCP de Netware incluso SSH tiene su propio protocolo llamado SSHFS.

Estos mecanismos fueron desarrollados para acceder archivos remotos de la misma forma que se pueden acceder a archivos en discos locales.

NFS es el protocolo para compartir archivos por red mas difundido en entornos tipo Unix (incluyendo a GNU/Linux aunque sea un clon) y su uso es obligatorio para el montaje de un CLUSTER MPI ya que el nodo principal debe acceder de forma transparente a los archivos remotos de los demás nodos, adicionalmente facilita la

copia de las aplicaciones que se quiere ejecutar en todos los nodos lo cual es una tarea tediosa cuando se cuentan con muchos de ellos.

14.7.5 Sistemas Operativos.

Normalmente se asume el uso de GNU/LINUX cuando se trata de CLUSTERS Beowulf pero se puede desplegar un CLUSTER Beowulf bajo cualquier sistema operativo incluso Microsoft Windows. Este prejuicio ha perjudicado el desarrollo en otros sistemas operativos que en algunos casos se podrían adaptar mejor a ciertos grupos u organizaciones. La escogencia de determinado sistema operativo o distribución se desprende de muchos factores, entre ellos los obvios como seguridad, rendimiento, hardware disponible e inversiones previas en licencias hasta otros más subjetivos como la comodidad de manejo o la experiencia de los administradores de los sistemas.

Hasta el año 2003 FreeBSD ofrecía mejor rendimiento en red (Leitner, 2003), por supuesto actualmente esto no es necesariamente igual, ambos sistemas continúan avanzando y se encuentran casi a la par en rendimiento, pero BSD y sus ramas han mostrado ser mucho mas estables ya que no se han fragmentado en la misma forma que lo ha hecho GNU/Linux con cientos de distribuciones. Es por esto que ha sido más fácil mantener el código y probarlo exhaustivamente. Por otro lado el hardware soportado es menor que en GNU/Linux a pesar de que la licencia BSD es mas cómoda para las empresas. Al final todo se resume al soporte y a la base de usuarios. BSD por otro lado es compatible a nivel binario con GNU/Linux por ello es posible correr programas diseñados para estos en BSD.

En cuanto a facilidad de uso e instalación por supuesto los sistemas Windows y Mac OS X llevan la delantera por ordenes de magnitud aunque en el caso de GNU/Linux y FreeBSD su instalación ya no es tan compleja y se cuenta con distribuciones como Ubuntu en el mundo GNU/Linux y PCBSD en BSD y los instaladores de las principales distribuciones de GNU/Linux que facilitan enormemente los procesos de instalación.

En el caso de los CLUSTERS GNU/Linux lleva la delantera en muchos sentidos. La mayoría de aplicaciones y herramientas son probadas mas exhaustivamente en GNU/Linux debido a la gran popularidad que estos sistemas han tenido y distribuciones como Rocks Cluster son especializadas en CLUSTERS y contienen tanto herramientas por defecto como especializadas.

La preferencia por BSD o mas específicamente FreeBSD debe partir del hecho de que estos son sistemas Unix y no clones de este a diferencia de GNU/Linux cuya estrategia ha sido copiar el funcionamiento de otros sistemas y que a largo plazo podría implicar llegar a estar tan lleno de errores como Windows, de hecho actualmente se ha aceptado entre la comunidad que ya esta sucediendo según Andrew Morton jefe de mantenimiento de GNU/Linux. En todo caso aunque GNU/Linux parecería ser la opción mas acertada es posible hacer uso del sistema con el que los usuarios se sientan mas cómodos por lo menos para propósitos de aprendizaje y posteriormente migrar a plataformas distintas.

Aproximadamente en el año 2002 hubo un momento en el que GNU/Linux estaba de moda, se podía encontrar en casi todas las revistas de tecnología algún CD que contenía una u otra distribución de GNU/Linux. En ese entonces la banda ancha no estaba difundida en Colombia, solo era posible para la mayoría de usuarios el usar MODEM Telefónico a 56kbps, una miseria comparada con las velocidades actuales.

Era inconcebible pensar que en ese entonces se podía descargar una distribución de Linux por Internet mucho menos cuando se cobraban por tiempo en línea. La cuenta podía salir muy costosa.

En ese entonces había dos distribuciones muy comunes, Redhat, Suse y Mandrake. Existía Debian pero Redhat y Mandrake eran las más usadas por su aparente facilidad para el usuario. En ese entonces muchos usuarios se encontraban emocionados y se suscribían a las revistas y muchas veces trataban de instalar GNU/Linux en su computador pero siempre ocurría algún problema, no funcionaba el MODEM se tenían que montar manualmente los disquetes o los CDs, las tarjetas de

video no eran soportadas. Todos estos problemas obligaban a muchos usuarios a volver a Windows debido a estas frustraciones. A algunas personas les gusta modificar y personalizar sus vehículos y sistemas operativos, lo encuentran divertido, pero ese no es el común denominador. Muchos prefieren que el sistema funcione desde el inicio sin tener que hacer ningún esfuerzo para ponerlo en funcionamiento o para hacerle mantenimiento, simplemente quieren que la maquina trabaje para ellos y no al revés.

Hoy casi 8 años después GNU/Linux continua mas fragmentado que nunca, con cientos de distribuciones, una variedad de sistemas de distribución de software, interfaces graficas y estructuras de archivos diferentes y no hay una forma unificada de distribuir binarios. Algunos afirman que es el fuerte de GNU/Linux y es cierto, pero su fuerte como sistema operativo para servidores es al mismo tiempo su debilidad como sistema operativo de escritorio.

Si alguien no esta de acuerdo con el rumbo que toma X o Y distribución simplemente crea una ramificación con los elementos que considera deben ser los correctos. Esto seria válido para un sistema operativo que NO intenta colarse al escritorio y quitarle cuota de mercado a Windows. Pero aun así GNU/Linux y algunos usuarios están empeñados en comprarse con Windows y Mac OS X, pero no alcanzan a raspar la superficie. La experiencia de usuario esta muy fragmentada y ni siquiera Ubuntu que es la distribución mas enfocada al escritorio logra hacerlo todo de la forma correcta. La razón es que a pesar de sus esfuerzos siguen ligados al mismo concepto de otras distribuciones enfocadas a otros mercados.

Si los promotores de GNU/Linux de verdad quieren llegar fuertemente a los escritorios necesitan definir una base o una especificación, desafortunadamente ya hubo un intento infructuoso con United Linux en el 2005.

Con este movimiento se pretendía precisamente crear una base única de Linux pero al mismo tiempo garantizar la libertad y las ramificaciones pero que cualquier programa hecho para una distribución funcionara transparentemente para otra,

sentando las bases para la distribución de software de forma tradicional como es costumbre en Mac y Windows, algo que sí lograron sistemas operativos que no se han fragmentado innecesariamente como BSD y sus diferentes distribuciones con su sistema de paquetes .PBI y su base estandarizada, los cuales son también libres u OPENSOURCE y existen desde antes que GNU/Linux. Desafortunadamente el esfuerzo pereció prematuramente porque Redhat al ser el rey de las distribuciones decidió darles la espalda.

Durante una entrevista con un exmantenedor del kernel Linux este afirmo estos puntos. Él quería implementar ciertos cambios en el kernel para hacerlo mas amigable en el escritorio, haciéndolo mas responsivo para las tareas mas comunes, infortunadamente no le permitieron seguir trabajando en estos cambios ya que las "compañías patrocinadoras de GNU/Linux" requerían mas rendimiento en cuestiones no orientadas al escritorio, como las consultas SQL o el acceso al disco duro.

Para el propósito de este trabajo por fortuna o infortunadamente, es necesario recurrir a GNU/Linux. Sony decidió abrir casi totalmente su consola a este sistema a diferencia de otras consolas y otros fabricantes principalmente porque la arquitectura del Cell BEA es muy radical, diferente y desconocida, incluso para programadores de juegos tradicionales y seria muy útil que hayan más programadores listos para programar en ella. Desafortunadamente su preferencia por Yellow Dog Linux es un poco lamentable.

Yellow Dog Linux fue famosa debido a que, por razones desconocidas, Apple confió en ellos y les dio acceso a valiosa información secreta sobre su hardware, controladores y sistema de arranque. Durante mucho tiempo fueron también los únicos autorizados para vender computadores Apple con Linux preinstalado. Toda esta confianza los transformo en la distribución estrella para PowerPC. Claro que, posteriormente con el paso de Apple a Intel y x86, la cuota de mercado de la arquitectura PowerPC quedo extremadamente reducida, hasta el punto que su negocio estaba a punto de la quiebra. Por fortuna encontraron refugio en IBM y SONY, el primero con sus servidores Blade basados en Cell BEA y el segundo por

supuesto por su consola de videojuegos PlayStation 3, ambos basados en la arquitectura Cell BEA.

Desafortunadamente Yellow Dog esta basado en Redhat, aunque técnicamente esta basado en RHEL y se ve plagado de los mismos problemas que estos últimos con su sistema de paquetes basado en RPM, el cual a vista de muchos se puede convertir en un infierno por las dependencias de paquetes, aunque se ha desarrollado un sistema de instalación conocido como YUM que se encarga de resolver estas dependencias.

Por supuesto en teoría se puede instalar casi cualquier distribución de GNU/Linux para PowerPC en el PlayStation 3 de hecho hay dos distribuciones adicionales que buscan entrar fuertemente en este segmento de mercado que son OpenSUSE y Ubuntu. Esta última sería la más amigable por su sistema de paquetes APTGET, el cual no sufre de los problemas de los RPM. Infortunadamente las herramientas oficiales están distribuidas en forma de RPM y aunque hay forma de usarlas en un sistema DEBIAN o Ubuntu es mejor no tomar esa molestia, después de todo el PlayStation 3 será usado para solo un objetivo y si las herramientas de desarrollo del CELL BEA funcionan correctamente en Yellow Dog lo mejor es usarlas tal como están.

14.7.6 Red y Seguridad.

La elección de determinada topología de red, protocolo o interfaz debe responder a las necesidades, requerimientos y tecnología ya desplegada de cada grupo u organización. No hay ninguna tecnología única y exclusiva para el despliegue de un CLUSTER y aunque son preferibles las interfaces de alta velocidad como Infiniband, fibra óptica o GigaBit Ethernet en teoría se podría desplegar un CLUSTER a través de cualquier protocolo de red o interfaz, claro que con las limitaciones que puede traer cada una.

Una consideración importante a la hora de desplegar CLUSTERS que se integren a GRIDS es la seguridad en la red ya que se esta expuesto a todos los peligros que se pueden esperar de cualquier computador con acceso a Internet. Es una de las razones por la que el uso de sistemas como Cóndor o XGrid son preferibles para sistemas aislados y restringidos ya que al tener que compartir su uso con usuarios de escritorio y sobre sistemas vulnerables o inseguros se pone en riesgo la integridad no solo del CLUSTER si no de otros sistemas que accedan a los recursos de procesamiento ofrecidos pues se es mas propenso a virus, troyanos, gusanos, ataques de HACKERS, CRACKERS y LAMERS. Por otro lado los nodos son vulnerables al mal uso que le pueden dar los usuarios o fallos no intencionales del software que se ejecuta en ellos.

14.8 MALLAS COMPUTACIONALES O GRIDS

Si regresamos hasta 1960 cuando John Mccarty pronostico por primera vez que el procesamiento se convertiría en un servicio publico y retomamos las razones por las que en 1990 Ian Foster y Karl Kesselman decidieron acuñar este concepto como GRID o malla, haciendo un símil con las malla de electricidad, es probable que aun no estemos en presencia de un GRID computacional como tal. Una prueba de ello es que todos los autores "están de acuerdo en que no hay acuerdo" en una definición de que es un GRID computacional y si analizamos la "computación en la nube" probablemente el concepto de GRID, por lo menos comercial, evolucione hacia este, el cual es un concepto que actualmente se considera que se desprende del primero. Si el procesamiento en verdad tiende a convertirse en un servicio al que se accede casi de la misma forma que se accede al servicio de electricidad, hay que tener ciertas consideraciones tecnológicas y practicas.

Aun si se contara con redes enteramente compuestas por conexiones de fibra óptica y lográramos desarrollar una tecnología de conmutación capaz de manejar toda la velocidad y ancho de banda que permiten las conexiones de fibra óptica (lo cual es el principal cuello de botella actualmente) la latencia seria aun muy alta como para pensar que todos los tipos de procesamiento y aplicaciones puedan hacer uso de recursos de procesamiento dispersos geográficamente. Si se analiza la tecnología de los procesadores y computadores actuales vemos que de hecho, cada vez se trata

de acercar más la memoria al procesador precisamente para reducir la latencia por lo que tener recursos dispersos por cientos de kilómetros cuadrados no mejora necesariamente la velocidad de procesamiento para ciertas aplicaciones en las que la latencia es un obstáculo.

Todo esto conduce al concepto de la computación en la nube donde de hecho las aplicaciones se ejecutan remotamente y los computadores personales solo se encargan de recibir las entradas (input) y visualizar el resultado de estas y la carga de procesamiento es realizada en CLUSTERS cuyos nodos se encuentran mas cerca unos de otros. Por otro lado estos mismos CLUSTERS podrían proveer servicios de procesamiento en lotes para otro tipo de aplicaciones y de esta forma se haría un mejor uso de estos al reducir los tiempos de ocio.

La diferencia clave entre un CLUSTER y un GRID es un poco esquivada, en un principio parecía que esta fuese la homogeneidad o heterogeneidad de sus nodos, pero como se ha visto durante el estudio de las herramientas y tecnologías para el despliegue de los CLUSTERS estos de hecho pueden estar compuestos por distintas arquitecturas o sistemas operativos. Es posible desplegar supercomputadores con nodos homogéneos sin que necesariamente sea considerado un CLUSTER. Tampoco pareciera ser su forma de manejar las tareas ya que de la misma manera que un CLUSTER puede realizar procesamiento en tiempo real, puede hacerlo en BATCH que es la forma de ejecución de tareas más común en los GRIDS por la latencia en la transmisión de datos o los volúmenes de información.

La distancia es un concepto muy relativo ¿que tan alejados deben estar los nodos para ser considerados GRIDS? Probablemente no haya una diferencia única y clave que determine cual es cual y sean varias características que confluyen simultáneamente, pero dos características se han mantenido constantes a lo largo de los esfuerzos para definir el concepto y podrían ser las candidatas para una definición final son el hecho trascender las barreras corporativas y organizacionales, configurándose en un esfuerzo conjunto entre diferentes organizaciones claramente definidas, como también el que no haya un control centralizado como si lo hay en el concepto de CLUSTERS. Pero ¿que ocurriría si dos organizaciones separadas

geográficamente comparten recursos computacionales homogéneos a través de una red VPN conjunta creada solo para ese propósito? ¿Podría ser considerada un CLUSTER o un GRID? Claramente todavía hay muchas cuestiones por definir.

Si bien el concepto de GRID aun se esta forjando y los primeros protocolos estandarizados son recientes, el concepto de GRID como una infraestructura de computación distribuida a escala global ya esta en marcha y redes de distribución como la LHCCG o Malla Computacional del Gran Colisionador de Hadrones (LARGE HADRON COLLIDER COMPUTING GRID) del CERN entre otras, se encuentran en funcionamiento y tras estos esfuerzos se encuentra una herramienta que como ya se menciono se convirtió en un estándar de facto para el despliegue de GRIDS y es el GLOBUS TOOLKIT.

14.8.1 Globus Toolkit.

Antes de la OGF y la estandarización de protocolos para el despliegue de GRIDS existió la I-WAY producto de la propuesta del matemático Rick Stevens del laboratorio nacional de Argonne y Tom Defati director del laboratorio de visualización electrónica de la universidad de Illinois en Chicago. Ambos propusieron la creación de un enlace entre once (11) redes de investigación de alta velocidad. A raíz de esta necesidad Ian Foster y su equipo desarrollaron los primeros protocolos que permitieron a los usuarios de la I-WAY ejecutar aplicaciones en computadores distribuidos alrededor de todo Estados Unidos. Esta iniciativa fue financiada por la DARPHA y de ella nació el grupo Globus y su Globus Toolkit.

El Globus Toolkit es un conjunto de aplicaciones protocolos y herramientas de monitoreo, seguridad, administración y descubrimiento de recursos que permiten la colaboración interinstitucional al compartir recursos computacionales, bases de datos y otras herramientas de forma segura, atravesando barreras corporativas y geográficas sin sacrificar la autonomía local.

Globus Toolkit no nació de un proceso de estandarización más si de una necesidad y se convirtió en un estándar de facto para el despliegue de GRIDS. Ambos Globus Toolkit y el concepto de GRID han evolucionado conjuntamente.

A pesar de que el Globus Toolkit es una herramienta libre y soporta algunos protocolos estandarizados, la mayor parte de sus protocolos y subsistemas fueron impuestos dejando por fuera de su definición y desarrollo a muchos centros de investigación y usuarios tanto de Estados Unidos como de otros países. Ya que el objetivo de los GRIDS siempre ha sido traspasar los límites de las bóvedas corporativas y de la heterogeneidad de sus sistemas, era imperiosa la necesidad de crear protocolos estándares y esto solo es posible mediante la creación de grupos de estandarización interdisciplinarios, con representación de diferentes instituciones académicas, grupos investigación y equipos de desarrollo de otras herramientas similares e incluso corporaciones privadas como Avaki, DataSynapse, Entropia, Fujitsu, Hewlett-Packard, IBM, NEC, Oracle, Platform, Sun y United Devices. A partir de esta necesidad nace la Open GRID Forum u OGF en el año 2006 cuyo propósito es precisamente lograr consenso y estandarizar los protocolos necesarios para el despliegue de GRIDS.

Gracias a sus esfuerzos se han puesto a consideración varios borradores estandarizando múltiples APIS lenguajes descriptivos, arquitecturas y ciclos de vida administrativos. A partir de la 3era versión el GLOBUS TOOLKIT y sus herramientas se rigen por estos estándares.

14.8.2 Estándares de la OGF

14.8.2.1 SAGA.

Es una especificación de API de programación orientada a objetos independiente de lenguaje que provee facilidades para el desarrollo de aplicaciones en entornos GRID. Entre las facilidades que esta provee están:

- Administración de Trabajos. Provee facilidades para el envío de trabajos para su ejecución en GRIDS y su administración posterior ya sea en lotes o de forma interactiva mediante el uso de un modelo consistente con la OGSA-BES.
- Espacios de Nombre o “Name Spaces”. Provee métodos para la administración de espacios de nombre jerárquicos.
- Administración de Archivos. En conjunto con los espacios de nombre provee métodos para operaciones sobre archivos como leer escribir y encontrar.
- Administración de Réplicas. Permite el manejo de archivos lógicos y réplicas al igual que búsquedas basadas en metadatos.
- Corrientes de Datos o Streams. Proveen una abstracción persistente de flujos byte para objetos de comunicación.
- Llamada Remota de Procedimientos. Provee un estilo de llamadas remotas de procedimientos de interacción consistente con el estándar GRIDRPC de la OGF.
- Modelo de Tareas. Provee administración de cualquier tarea asíncrona ya sea remota o local.
- Modelo de Monitoreo. Provee mecanismos de respuesta donde una aplicación puede ser notificada de los cambios en el estado de los objetos.

- Sesiones. Permite que un grupo de objetos puedan ser aislados y administrados independientemente.
- Contextos. Provee un contenedor de información de seguridad adherido a la sesión.
- Objetos Base. Provee métodos esenciales para los objetos SAGA como identificaciones únicas.
- Interfaces de Atributos. Provee una interfaz común para obtener y establecer los atributos de los objetos.
- Manejo de Errores. Provee un manejo uniforme de los errores para los objetos SAGA.(Goodale, et al., 2007)

14.8.2.2 OGSA (OPEN GRID SERVICE ARCHITECTURE).

Acrónimo en ingles de Arquitectura Abierta de Servicios GRID el cual es un modelo estándar que define el grupo de comportamientos y características que se encargan de solucionar los principales inconvenientes que surgen al desplegar sistemas GRID entre los que se encuentran el cómo se establecen las identidades y se negocian las autenticaciones, cómo son expresadas y negociadas las políticas, cómo se descubren monitorean y administran los servicios, como se integran los recursos de datos en los procesos computacionales, como se organizan las colecciones de servicios de forma jerárquica para de esta forma proveer servicios escalables y confiables, como se negocian y se monitorean los acuerdos de niveles servicios. La OGSA se basa en la OGSF la cual define las interfaces los comportamientos y convenciones que controlan como los servicios GRID son creados nombrados monitreados y finalmente destruidos.

Los objetivos de la OGSA fueron desde un inicio el ser una arquitectura orientada al servicio (SOA), promover los estándares, plataformas de aplicación, herramientas de desarrollo, estándares de la Web y la modularidad para permitir el reuso de subsistemas y funciones de tal forma que puedan ser usados en diferentes comportamientos.

Para la definición de la arquitectura se tienen en cuenta toda las clases de servicios que se espera pueden operar sobre un GRID tanto científicos como comerciales. (Foster I. , Kesselman, Nick, & Tuecke, 2002).

Algunos de los componentes más importantes propuestos por la OGSA se describen a continuación:

14.8.2.3 DRMAA (DISTRIBUTED RESOURCE MANAGEMENT APPLICATION API).

Es el acrónimo en ingles de API de aplicaciones administrativas de recursos distribuidos. Es una especificación para el envío y control de trabajos en uno o más sistemas DRM o sistemas de administración de recursos distribuidos. Su función es proveer funcionalidad a alto nivel necesaria para que determinada aplicación envíe trabajos a sistemas DRM entre los que se incluyen operaciones de terminación o suspensión. Su objetivo es facilitar la interfase directa en sistemas DRM de aplicaciones que no fueron diseñadas para su ejecución en estos. (Chan, et al., 2004)

14.8.2.4 JSDL (JOB SUBMISSION DESCRIPTION LANGUAGE).

Acrónimo en ingles de lenguaje descriptivo de envío de trabajos. Es una especificación XML extensible, usada para la descripción de tareas simples a sistemas computacionales de ejecución no interactiva. Entre las descripciones que se pueden proveer cuentan el nombre del trabajo y descripción requerimientos de recursos que determinada computadora debe tener para ser elegible como la memoria RAM, espacio de intercambio o SWAP, velocidad de

CPU numero de CPUS, sistemas operativos, limites de tiempo de ejecución y memoria, variables de entorno por definir argumentos de línea de comando entre otros. (Anjomshoaa, et al., 2005)

14.8.2.5 CDDL (CONFIGURATION DESCRIPTION DEPLOYMENT AND LIFECYCLE MANAGMENT).

Acrónimo en ingles de Administración de ciclo de vida, configuración, descripción y despliegue el cual es una especificación de lenguaje descriptivo basado en parte en XML y SmartFrog o armazón inteligente para grupos de objetos (Smart Framework for Object Groups) que puede contener enlace con múltiples lenguajes. Ambos son lenguajes descriptivos, en el caso de SmartFrog este es usado para describir, activar y administrar sistemas de software distribuido como componentes que colaboran entre si. CDDL al ser basado en este, usa el mismo concepto pero enfocado a la integración de sistemas distribuidos virtualización, administración de servicios y manejo de ciclo de vida del mismo GRID. Al mismo tiempo cuenta con una API de despliegue basada en SOAP. (Patrick Goldsack)

14.8.2.6 GRIDRPC.

Es una especificación estándar de API para la ejecución de procedimientos remotos (RPC) similar a CORBA, PDO, JavaRMI, COM, DCOM, entre otros sistemas de objetos distribuidos que se mencionaron anteriormente, pero con un enfoque hacia los GRIDS buscando simplicidad, ligereza y rendimiento para facilitar el desarrollo de aplicaciones sobre estos entornos. (Nakada, Matsuoka, Seymour, Dongarra, Lee, & Casanova, 2002)

14.8.2.7 RSL (RESOUCE SPECIFICATION LANGUAGE).

Acrónimo en ingles de lenguaje de especificación de recursos. Es un lenguaje común usado para describir recursos usados por los diferentes componentes de GRMA para manipular cadenas hechas en base a este, para realizar funciones administrativas en cooperación con el resto de componentes del

sistema. RSL provee una un esqueleto sintáctico usado para componer descripciones de recursos complejos. (RSL Especification)

14.8.3 Arquitectura de una malla computacional y sus componentes.

Parte de los objetivos iniciales de este estudio y una de las exigencias de la Universidad fue la de "hacer una propuesta de implementación o diseño de arquitectura donde se especifiquen los componentes de un GRID" pero en vista de los hallazgos resultado del estudio de este tema, se llego a la conclusión de que proponer una arquitectura o un diseño de un GRID que se ajuste a todas las necesidades es un objetivo pretencioso e ilusorio debido precisamente a las particularidades inherentes al concepto de GRID y a la magnitud de los retos que este intenta afrontar, principalmente a el hecho de que semejante "organización virtual" (Foster, The Anatomy of the Grid) como lo es la de un GRID o la participación en uno involucra a diversas organizaciones, disciplinas, estudios de necesidades y objetivos muy variados que resultarían imposibles de abordar por una sola persona pero sobre todo por una sola organización por cuanto no es el objetivo de un GRID. Por su puesto se podría hacer una propuesta para cumplir con los requerimientos pero esta será inocua y superficial de hecho ni siquiera Ian Foster Y Karl Kesselman se atreven a proponer una arquitectura que se ajuste a todas las necesidades y se limitan a analizar los componentes mas comunes y esenciales.

Aunque en un país como Colombia la inversión en infraestructura es vital para su desarrollo el invertir en el despliegue de CLUSTERS o GRIDS sin propósitos claros u objetivos específicos se configura en un desperdicio de tiempo esfuerzo y dinero. Un CLUSTER o un GRID no es el tipo de inversión que rinde frutos a largo plazo como lo pueden ser las vías, puertos marítimas, aeropuertos, sistemas de acueducto, o redes eléctricas y de telecomunicaciones que son inversiones que pueden prestar servicios con la misma eficiencia durante muchos años, son incrementales ya que para aumentar su capacidad se puede reusar la infraestructura existente, no se devalúan tan rápidamente y aunque pueden sufrir deterioro el costo de mantenimiento es bajo y las mantiene actuales y en servicio por mucho tiempo. En el caso de la computación y el procesamiento no ocurre lo mismo, los equipos son propensos a daños, su deterioro es inevitable e insoluble y mas importante no es un

tipo de inversión incremental por cuanto la capacidad instalada se devalúa tecnológicamente de forma rápida y en un lapso de tiempo corto resulta poco rentable su mantenimiento.

Si bien es posible desplegar las herramientas propias de un GRID como lo son el Globus Toolkit en unos cuantos equipos sería una tarea que aunque es relativamente trivial y estructurada resulta exagerada, prolongada y no lograría un propósito más allá de simplemente saciar la curiosidad, por cuanto estas herramientas sobrepasan enormemente las necesidades de una organización pequeña como lo es una universidad pequeña y complicarían el aprendizaje de los conceptos que deberían ser objeto de un estudio más profundo como lo son las técnicas de programación distribuida y paralelismo de algoritmos. En su defecto se podría proponer un diseño de CLUSTER solo para propósitos educativos, que podría contribuir a formar profesionales capacitados en el desarrollo de aplicaciones distribuidas y el despliegue de dichos sistemas. De esta forma se crea una base de profesionales que estarían preparados al momento de la entrada en funcionamiento de redes de esfuerzos y proyectos de colaboración científica y tecnológica nacional.

La arquitectura de cualquier GRID esta basada en el concepto o paradigma SOA (Service Oriented Architecture) o arquitectura orientada al servicio. En el paradigma SOA se logra prestar un servicio, mediante un lenguaje común y la unión de múltiples capacidades distintivas, prestadas ya sea por personas o máquinas independientes. Se refiere también a personas porque el concepto SOA no es solo computacional, sino que es un concepto que de se ha venido usando incluso antes de la creación de las computadores (Erl, 2007).

Dicho servicio en el campo de la computación es logrado gracias al ofrecimiento de ciertas capacidades o recursos computacionales con el uso del intercambio de mensajes mediante interfaces, protocolos, lenguajes, políticas y sistemas comunes entre la diferentes entidades que conforman dicho servicio, los cuales pueden ser de almacenamiento, transferencia de datos, administración y gestión de errores. En el campo de los GRIDS computacionales dichos elementos comunes son definidos por la OGSA (OPEN GRID SERVICE ARCHITECTURE) y son implementados de forma

autónoma por sus entidades, de esta forma se logra la abstracción entre el cómo se accede a determinado servicio y cómo es este implementado.

Las interfaces definidas para estos propósitos se deben ocupar de ciertas cuestiones tales como el descubrimiento y composición de servicios para anunciar que capacidades están disponibles por cada entidad y de esta forma crear servicios compuestos por ellas, especialización con las que cada capacidad puede ser usada tanto por arquitecturas similares como por otras distintas y extensión de interfaces con las que se puede ampliar el abanico de funcionalidades de ciertas implementaciones manteniendo las interfaces comunes, para de esta forma no tener que enfocar las aplicaciones a un mínimo común denominador e ignorar ciertas capacidades únicas.

Los GRIDS presentan un reto que va mas allá de los sistemas SOA comunes debido a que estos requieren de un gran nivel de cooperación que involucra a múltiples, corporaciones organizaciones e individuos todos con políticas, objetivos y recursos diferentes. Es por esto que nace la necesidad de la coordinación de recursos, el acceso no solo a computadores y recursos de procesamiento sino también a datos, software y otros tipos de recursos tanto para aplicaciones científicas como comerciales, en lo que se denomina una organización virtual. Estas organizaciones pueden variar enormemente, tanto en tamaño como en propósito duración, estructura y sociología, pero usualmente comparten requerimientos comunes en términos como seguridad, aislamiento de problemas, administración de recursos y flujos de trabajo, pero usualmente los recursos son condicionales por cuanto cada individuo organización o corporación determinan las restricciones de como cuando y donde se puede hacer uso de sus recursos.

La arquitectura propuesta por Ian Foster y Karl Kesselman no debe ser considerada como una respuesta única, sino como un punto de partida sobre el que se pueden añadir capas para enfrentar las necesidades propias de cada organización.

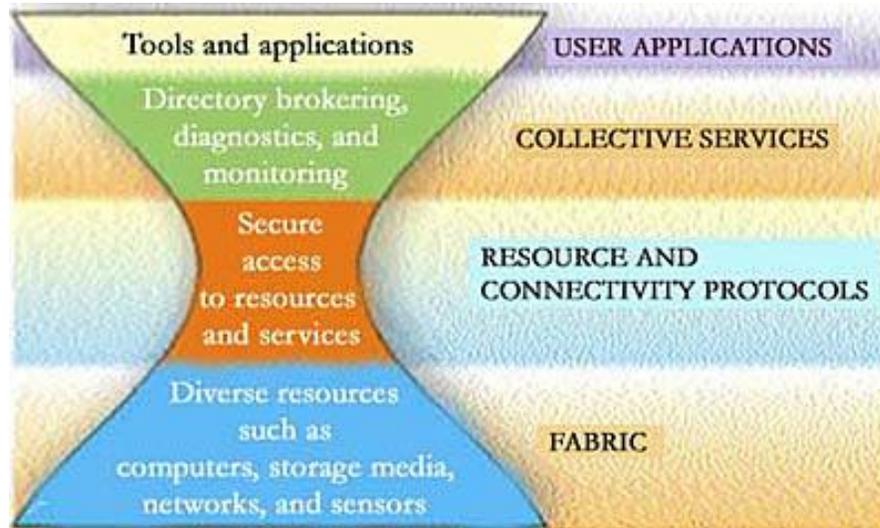


Figura 2. Modelo Reloj de Arena GLOBUS

La arquitectura es un modelo de cuatro capas basada en el modelo reloj de arena como se muestra en la figura 2, en el que la parte mas angosta corresponda a un conjunto pequeño de abstracciones, protocolos de conectividad y recursos sobre los que se pueden construir comportamientos de mas alto nivel y que al mismo tiempo están contruidos sobre un amplio rango de recursos. Las cuatro capas que compone la arquitectura son:

14.8.3.1 Entretejido o Fabric.

La capa de entretejido es aquella que contiene los recursos sobre los que los protocolos GRID median. Estos pueden ser recursos computacionales de almacenamiento catálogos recursos de red e incluso sensores. Al mismo tiempo un recurso puede ser una entidad lógica como un sistema de archivos distribuidos, un CLUSTER de computadores o una piscina o “pool” de computadores distribuidos. En tales casos cada recurso puede manejar protocolos adicionales internos, pero estos no son del interés de una arquitectura GRID precisamente por la abstracción.

14.8.3.2 Capa de conectividad y recursos.

Esta capa define el núcleo de protocolos de autenticación y comunicación demandados por los tipos de transacciones específicas requeridas para GRIDS los cuales permiten el intercambio de datos entre recursos de la capa de entretelido. Aunque existen alternativas la mayoría de estos protocolos son tomados de la pila de protocolos TCP/IP y en un futuro se podría requerir de protocolos diferentes a medida que las redes de telecomunicaciones evolucionen.

Entre los requerimientos que trata de proveer esta capa están:

- Autenticación única para no requerir de múltiples autenticaciones para acceder a los distintos recursos.
- Delegación por medio de la cual un programa puede actuar en nombre de un usuario automáticamente y acceder transparentemente a los recursos de forma segura.
- Integración con soluciones de seguridad locales para evitar el tener que reemplazar soluciones y esquemas de seguridad ya establecidas en las diferentes organizaciones que hace parte del GRID.
- Relaciones de confianza, basadas en usuario, que buscan que este pueda acceder a múltiples recursos sin que estos deban interactuar o cooperar entre ellos.

Una vez se ha establecido y verificado la autenticidad de los usuarios la capa de recursos la cual funciona sobre la capa de conectividad y los protocolos de autenticación, se encarga de interactuar entre los diferentes recursos y servicios, definiendo protocolos para la negociación segura, inicio, monitoreo,

control, contabilidad y pago de operaciones de colaboración sobre recursos individuales. Los protocolos de información se encargan de obtener información sobre la estructura y el estado de determinados recursos como su configuración, carga y políticas de uso. Los protocolos de administración se encargan de negociar el acceso a un recurso compartido especificando los requerimientos de recursos, las operaciones a ser realizadas y la consistencia entre las políticas requeridas y las ofrecidas. (Foster & Kesselman, THE GRID: BLUEPRINT FOR A NEW COMPUTING INFRASTRUCTURE, 2003)

14.8.3.3 Servicios Colectivos.

La capa de servicios colectivos contiene los protocolos y servicios que no están asociados con recursos específicos, pero con base a los protocolos de la capa de recursos y conectividad los cuales son reducidos se implementan una gran variedad de comportamientos que se construyen con base a colecciones de recursos sin requerir de protocolos adicionales.

Entre los servicios que esta capa presta están los servicios de directorio, los cuales permiten que los participantes de una Organización Virtual descubran la existencia o propiedades de determinados recursos permitiendo búsquedas por nombre atributos tipos disponibilidad carga.

El coalojamiento y calendarización permite a los participantes requerir el alojamiento de uno o más recursos para un propósito específicos y la calendarización de tareas en los recursos apropiados.

El monitoreo y diagnostico de servicios permiten el monitoreo de recursos para detectar fallos ataques o irrupción de intrusos, sobrecarga entre otros.

La replicación de datos permite la administración de recursos de almacenamiento para maximizar el desempeño del acceso a datos con base a métricas como tiempo confiabilidad y costo.

La capa de servicios colectivos también contempla modelos de programación y herramientas para la programación en sistemas GRID, sistemas para describir, usar y administrar flujos de trabajo asincrónicos de múltiples componentes y etapas, servicios de descubrimiento de software para seleccionar las mejores implementaciones de software y plataformas sobre las cuales están desplegadas, sistemas colaborativos para el intercambio de información síncrona o asíncrona entre comunidades de usuarios potencialmente amplias.

14.8.3.4 Aplicaciones de Usuarios.

La capa de aplicaciones la conforman las aplicaciones de los usuarios que operan en una organización virtual (GRID) los cuales son construidos con base a o mediante los servicios definidos en cualquiera de las capas inferiores.

15.1 DESPLIEGUE DE CLUSTER COMPUESTO POR PLAYSTATION 3

La siguiente guía, se estructura como una compilación de los trabajos publicados en sus sitios Web por el Dr. Gaurav Khanna y el Dr. Chris Poulin respecto a implementaciones de sistemas de procesamiento distribuido.

Durante su desarrollo se han tratado de suplir las falencias detectadas en los originales en cuanto al nivel de detalle en la descripción de los procesos asociados a la configuración e instalación sobre sistemas YellowDog Linux. Buscando así generar una propuesta global y unificada que se pueda publicar de forma conjunta con los investigadores antes mencionados.

15.1.1 Generalidades.

Aunque las pruebas relacionadas a continuación se llevaron a cabo en nodos compuestos por consolas PlayStation 3, la mayor parte de las instrucciones presentadas son válidas para el montaje de un Cluster sobre computadores de escritorio genéricos.

15.1.2 Sistema Operativo.

El PlayStation 3 permite la instalación de sistemas operativos, con el requerimiento de estar diseñados para correr sobre plataformas PowerPC y tener ciertas modificaciones para comunicarse con el hardware a través de llamadas a la capa de maquina virtual o HYPERVISOR. Por lo tanto, es bueno aclarar que su instalación está totalmente permitida y respaldada por Sony y no implica daños al sistema.

Para poder instalar GNU/Linux sobre el PlayStation 3 se requiere la descarga de una de las versiones y distribuciones diseñadas para este sistema, como pueden ser

OpenSUSE, UBUNTU, FEDORA 9 o YELLOW DOG LINUX todas en su versión específica para PS3.

Una vez descargado y quemado en un CD o DVD la versión del sistema operativo, este es insertado en la unidad óptica del PS3, luego mediante el XMB (Interfaz del PS3) se debe acceder a la opción /ajustes/ajustes del Sistema /herramienta de formateo, donde se acepta el formateo del disco y se elige la opción de ajustes personalizados, asignando 10GB al PlayStation 3 y el resto quedará para la instalación de GNU/Linux, interfaz mostrada en la figura 3.



Figura 3. Interfaz de configuración del PS3

Solo existen 3 opciones de formateo: todo el espacio para el PlayStation, 10 GB para el PlayStation, ó 10 GB para otro sistema operativo. Posteriormente se puede elegir la opción de formateo rápido o completo. Esto elimina todo archivo del disco duro por lo tanto borra cualquier información que se tenga en él, incluyendo los archivos de instalación de juegos y el avance de las partidas.

Una vez terminado el proceso, se retorna al menú ajustes/instalar otro sistema operativo, donde el PlayStation 3 escaneará los discos y los puertos USB en busca de archivos de arranque (BOOT LOADER) .bld. Las distribuciones actuales para PS3

incluyen por defecto el archivo .bld en su estructura de ficheros por lo que no es necesario descargarlo por aparte.

Aunque Yellowdog incluye su propio cargador de arranque (BOOT LOADER) éste cuenta con una interfaz muy burda a diferencia del cargador de arranque de OpenSUSE, por lo cual se recomienda descargar el DVD de instalación de OpenSUSE para PS3 e instalar su cargador de arranque.

Posteriormente y antes de reiniciar, es necesario expulsar el disco e insertar el disco de instalación de Yellowdog, el que será reconocido por el cargador de arranque. Aunque es posible instalar OpenSUSE que es más amigable e integrado, este es demasiado pesado debido a los efectos de sus escritorios incluidos, por lo que YellowDog fue más adecuado para los propósitos de esta guía.

En caso de querer usar un archivo .bld diferente para propósitos de recuperación del sistema, para que pueda ser reconocido este debe estar en una memoria USB (SD MEMORY STICK o COMPACT FLASH si el modelo cuenta con estos puertos) en la ruta /PS3/otros/.

Una vez el sistema identifique el archivo .bld este se instalará. Concluida la instalación es necesario acceder a /ajustes/sistema predeterminado y elegir la opción otro sistema operativo. Paso seguido, la plataforma preguntará si se desea reiniciar el sistema, a lo que se responderá afirmativamente.

Culminado el reinicio, el sistema cargará automáticamente el archivo .bld con el que se procederá a realizar la instalación del sistema desde el CD o DVD de instalación, tal y como si fuese una instalación común de GNU/Linux en un computador genérico. Terminado este proceso el equipo se comportara como cualquier instalación GNU/Linux.

15.1.3 Recomendaciones Previas.

Las recomendaciones para tener en cuenta son las que conciernen a los usuarios, contraseñas y permisos, los cuales aunque son simples, pueden resultar en complicaciones que aparentemente no tienen explicación y pueden retrasar cualquier despliegue.

Se recomienda el uso de contraseñas y usuarios idénticos a través de todos los nodos y si se trabaja sobre sistemas heterogéneos la asignación idéntica de permisos, identificaciones de usuarios y grupos. Esto permite que los servicios para compartir archivos como NFS se sincronicen entre los nodos esclavos y el nodo maestro y sean totalmente transparentes ya que incluso en sistemas tipo Unix no son asignados de la misma forma.

15.1.3.1 Modificación de Permisos y Usuarios.

La mejor manera de llevar a cabo este proceso es hacerlo inmediatamente después de una instalación fresca y sin haber accedido al sistema con otro usuario diferente a root, como siempre, se motiva a no hacer uso del usuario root frecuentemente.

Si los nodos dentro del clúster usan exactamente el mismo sistema operativo, es solo cuestión de crear el mismo usuario con las mismas contraseñas a través de todos ellos, pero si se tienen sistemas mixtos muy diferentes, como por ejemplo máquinas GNU/Linux, FreeBSD y Mac OS X, es necesario lograr un consenso no solo sobre los nombres de usuario y contraseñas sino sobre los nombres e identificaciones de los grupos.

En los sistemas GNU/Linux el primer usuario diferente a root se le asigna una UID (Identificación de Usuario o User ID) de 500 y a los usuarios números subsecuentes a partir de este, pero en los sistemas BSD y Mac OS X esto ocurre a partir del UID 501. Por otro lado, en los sistemas GNU/Linux todo usuario pertenece a un grupo principal llamado de la misma forma que el

usuario y con un GID (Identificación de Grupo o Group ID) que también tiene correspondencia con el UID, pero en sistemas BSD y Mac OS X esto no ocurre de esta forma y todos los usuarios pertenecen al grupo principal Staff con un GID de 20.

En el prototipo realizado se decidió llevar a cabo los cambios en GNU/Linux, porque aunque solo hay un computador con Mac OS X, este está siendo usado para otros propósitos y modificar los permisos es una actividad que no es trivial y puede implicar daños en el sistema.

Por ejemplo, se quiere crear el grupo staff con un GID de 20, el grupo 20 corresponde al grupo "games", por lo que para usar el GID se debe cambiar el GID del grupo "games" así:

groupmod -g 23 games

Seguidamente se crea el grupo staff:

groupadd -gid 20 staff

Una vez realizado esto se modifica el grupo usuario1:

Groupmod -g 501 usuario1

Posteriormente se cambia el UID y los grupos del usuario que se quieren usar (otro diferente a root):

usermod -u 501 usuario1

Luego se cambia el grupo principal:

usermod -g 20 usuario1

Para mantener el grupo tradicional usamos la G mayúscula, la cual asigna grupos secundarios:

usermod -G 501 usuario1

Una vez llevado esto a cabo, es necesario modificar los permisos de los archivos para que reflejen dichos cambios, esto se debe realizar como root y sin haber accedido aún como un usuario diferente. Si esto no es posible, mientras el usuario no esté en el sistema se ejecuta:

chown -R usuario1:staff /home/usuario1

Se recomienda hacerlo sin haber accedido al sistema con este usuario por primera vez, para que el proceso sea más limpio ya que de lo contrario habrá archivos y procesos bajo la pertenencia de este y se requerirá ejecutar como root lo siguiente:

find / -uid 500 | xargs ls -al >> /root/archivosviejos.txt

find / -gid 500 | xargs ls -al >> /root/archivosgviejos.txt

Luego se revisa el contenido de estos y si hay archivos que pertenezcan a estos usuarios o grupos se ejecuta:

find / -uid 500 | xargs chown usuario1 find / -gid 500 | xargs chgrp staff

15.1.3.2 Modificación de Variables de Entorno de Consola.

Es necesario igualmente modificar las variables de entorno del Shell para simplificar la ejecución de comandos y el acceso. Una vez se ha accedido como el usuario1, se deben modificar los archivos *~/.bash_profile* y *~/.bashrc*. El primero es usado para terminales de autenticación (SSH) y el otro para terminales interactivas sin autenticación (ventanas de consolas de cada GUI). A continuación se muestra un ejemplo de este proceso.

Nano ~/.bash_profile

nano ~/.bashrc

Y en cada uno se agrega lo siguiente:

***PATH=/usr/lib/openmpi/1.2.5-
gcc/bin:/usr/sbin:/usr/bin:/sbin:/bin:/mpiexec:\$HOME/bin:\$PATH
LD_LIBRARY_PATH=/usr/lib:/usr/lib/openmpi/1.2.5-
gcc/lib:\$LD_LIBRARY_PATH***

***export PATH
export LD_LIBRARY_PATH
unset USERNAME***

Es posible también añadir las rutas de forma individual, así:

***PATH=\$PATH:HOME/bin
PATH=/usr/lib/openmpi/1.2.5-gcc/bin:\$PATH
PATH=/usr/sbin:\$PATH
PATH=/sbin:\$PATH
PATH=/bin:\$PATH
PATH=/mpiexec:\$PATH***

```
LD_LIBRARY_PATH=/usr/lib/openmpi/1.2.5-gcc/lib:$LD_LIBRARY_PATH  
LD_LIBRARY_PATH=/usr/lib:$LD_LIBRARY_PATH
```

```
export PATH  
export LD_LIBRARY_PATH  
unset USERNAME
```

15.1.3.3 Configuración General de Nombres de Huésped o Hostname.

El archivo **/etc/hosts** contiene la relación de direcciones IP con los nombres de los huéspedes o hostname de la siguiente manera:

```
# Do not remove the following line, or various programs  
# that require network functionality will fail.  
127.0.0.1 localhost.localdomain localhost  
::1 localhost6.localdomain6 localhost6  
192.168.2.4 pelusa.perrera.local pelusa  
192.168.2.6 auyin.perrera.local auyin  
192.168.2.5 rufian.perrera.local rufian  
192.168.2.2 lamparita.perrera.local lamparita
```

Las primeras cuatro líneas vienen por defecto en el archivo de configuración y no deben ser modificadas.

La 5ta línea es necesaria por cuanto al modificar el nombre del huésped por defecto en el archivo **/etc/sysconfig/network**, algunos servicios como NFS, STATD, CUPS, SENDMAIL y SM-CLIENT requieren este parámetro y cuando está ausente el arranque del sistema puede tomar hasta 30 minutos y puede llevar a pensar que el sistema está corrupto sin razón aparente.

Las siguientes líneas muestran la relación ya mencionada, junto con el nombre del dominio, que en este caso es local (**perrera.local**) y finalmente el nombre corto de cada huésped.

15.1.3.4 Configuración del Nombre de Huésped en Cada Máquina.

El archivo `/etc/sysconfig/network` especifica los parámetros anteriores pero de forma interna en cada máquina.

```
NETWORKING=yes  
HOSTNAME=pelusa.perrera.local  
DOMAINNAME=perrera.local
```

15.1.3.5 Configuración de la Red Durante el Arranque.

La configuración de la red no es muy compleja, por conveniencia se pueden configurar los archivos **ifcfg-eth0**, ó, **ifcfg-wlan0** ubicados en `/etc/sysconfig/network-scripts/`. No se recomienda el uso de DHCP por cuanto este sistema puede asignar arbitrariamente direcciones IP lo cual rompería la coherencia de los archivos de configuración y afectaría la ejecución de programas en el clúster.

En este caso el montaje del CLUSTER se llevó a cabo bajo WIFI, por lo tanto se deshabilito la interfaz eth0 en **ifcfg-eth0**.

```
DEVICE=eth0  
BOOTPROTO=dhcp  
HWADDR=XX:XX:XX:XX:XX:XX  
ONBOOT=no
```

XX:XX:XX:XX:XX:XX corresponde a la dirección de hardware propia en cada máquina, estos valores son asignados automáticamente. En este caso no se especifican en el documento por razones de seguridad.

El archivo ***ifcfg-wlan0*** se modificaría de la siguiente forma:

```
DEVICE=wlan0  
BOOTPROTO=static  
BROADCAST=192.168.2.255  
IPADDR=192.168.2.4  
NETMASK=255.255.255.0  
HWADDR=XX:XX:XX:XX:XX:XX  
ONBOOT=yes
```

15.1.4 Preparación del sistema operativo.

Antes de instalar y configurar los programas y servicios propios del clúster es necesario preparar el sistema. La versión 6.1 de YellowDog está lejos de ser perfecta y aunque como hubiera sido mejor una instalación tipo DEBIAN como UBUNTU por su facilidad de uso, YellowDog es el sistema oficialmente soportado por SONY e IBM y aunque de hecho es posible instalar UBUNTU en la PlayStation 3, para ello hay que realizar demasiados procedimientos si se desea hacer uso de las librerías de desarrollo para el CELL BEA, por cuanto si el objetivo es el desarrollo sobre CELL BEA, es mejor trabajar sobre YellowDog o FEDORA a pesar de sus imperfecciones.

Conforme a esto, es necesario realizar 4 procedimientos para corregir las imperfecciones de Yellow Dog:

- Corregir los errores del WICD Manager para el manejo de redes inalámbricas 802.11g con encriptación WPA2/PSK.

- Corregir el controlador de sonido.
- Habilitar la memoria de video como espacio de intercambio o SWAP para mejorar el desempeño del sistema debido a la poca memoria RAM.
- Finalmente actualizar todo el sistema.

15.1.4.1 Corrección de la Aplicación de Conexión Inalámbrica WICD.

Para corregir los errores del WICD es necesario descargar los siguientes archivos:

wicd-1.5.9.tar.gz

wnettools.py.tar.gz para la versión wicd-1.5.9

Los sitios Web de en los que están disponibles se relacionan a continuación:

<http://downloads.sourceforge.net/wicd/wicd-1.5.9.tar.gz>

<http://www.yellowdog-board.com/download/file.php?id=62>

Una vez se tengan estos archivos fuente, se debe copiar el archivo ***wicd-1.5.9.tar.gz*** en ***/usr/local/src***, para posteriormente, a través de línea de comandos, ejecutar:

tar -xf wicd-1.5.9.tar.gz

cd wicd-1.5.9

python setup.py configure

python setup.py install

Posteriormente se debe descomprimir **wnettools.py.tar.gz** ya sea vía GUI o vía línea de comandos y copiar el contenido (**wnettools.py**) en **/usr/lib/python2.4/site-packages/wicd/**, confirmando la sobre escritura del archivo.

Una vez ejecutadas estas instrucciones se puede reiniciar el sistema o continuar directamente con los pasos descritos a continuación si ya se tienen todos los archivos.

15.1.4.2 Corrección del Error del Controlador de Sonido.

Para corregir el error con el controlador de sonido es necesario descargar el archivo **PS3.conf.alsa**, renombrar el archivo a **PS3.conf** y copiar el archivo en **/etc/alsa/cards/** y en **/usr/share/alsa/cards/**, si la carpeta cards no existe es necesario crearla.

Desafortunadamente al momento de elaborar este proyecto el enlace de descarga ha dejado de funcionar, pero es solo cuestión de copiar el código siguiente en un archivo de texto plano, llamar a este **PS3.conf** y copiarlo en las ubicaciones antes mencionadas.

```
# Alsa configuration for PS3  
# 08.04.09  
# Place in your system's alsa cards directory:  
# /etc/alsa/cards/PS3.conf  
# /usr/share/alsa/cards/PS3.conf  
#
```

```
<confdir:pcm/front.conf>
```

```
PS3.pcm.front.0 {  
  @args [ CARD ]
```

```

@args.CARD {
type string
}
type softvol
slave.pcm {
type hw
card $CARD
device 0
}
control {
name "PCM Playback Volume"
card $CARD
}
}

```

Default to dmix + softvol

```

PS3.pcm.default {
@args [ CARD ]
@args.CARD {
type string
}
type asym
playback.pcm {
type plug
slave.pcm {
type softvol
slave.pcm {
@func concat
string [ "dmix:CARD=" $CARD ",FORMAT=S16" ]
}
control {
name "PCM Playback Volume"
card $CARD
}
}
}

```

15.1.4.3 Habilitación de la Memoria de Video Como Espacio de Intercambio.

Para habilitar la memoria de video como espacio de intercambio es recomendable reiniciar en este punto y posteriormente configurar las conexiones a Internet, una vez se tenga acceso a la red se debe hacer lo siguiente:

Ya sea como usuario root, o mediante el comando su desde la consola de cualquier usuario, ejecutar:

su - (no olvidar el -)

Se realiza la autenticación con la contraseña de root y luego:

```
cd /etc/107ell.d  
wget http://us.fixstars.com/support/solutions/ydl\_6.x/ps3-vram-swap  
chmod 755 ps3-vram-swap  
chkconfig --add ps3-vram-swap  
chkconfig ps3-vram-swap on  
service ps3-vram-swap start
```

Para verificar el éxito del trabajo realizado se debe ejecutar el siguiente comando:

swapon -s

La información que debe aparecer es:

<i>Filename</i>	<i>Type</i>	<i>Size</i>	<i>Used</i>	<i>Priority</i>
<i>/dev/ps3da3</i>	<i>partition</i>	<i>522104</i>	<i>12196</i>	<i>-1</i>
<i>/dev/mtdblock0</i>	<i>partition</i>	<i>241656</i>	<i>0</i>	<i>1</i>

15.1.5 Herramientas Para el Despliegue del Cluster.

Para llevar a cabo el montaje y puesta en funcionamiento de un Cluster básico no se requiere demasiado esfuerzo, en primer lugar es necesaria la instalación y configuración de SSH y alguna implementación de la especificación MPI y NFS tanto en los servidores como en los clientes o nodos.

Ciertos sistemas operativos pueden traer por defecto estos programas incluidos en su instalación, en otros sistemas como DEBIAN o REDHAT es solo cuestión de usar sus respectivos manejadores de paquetes como “apt-get” y “yum”, los cuales simplifican enormemente el proceso de instalación al automatizar los procesos de creación de entornos para al configuración de librerías y compilación.

En el caso de YellowDog el procedimiento se supedita a la ejecución de los siguientes comandos:

```
yum search nfs
yum search openmpi
yum search ssh
```

Si por alguna razón estos no se encuentran instalados, el proceso a realizar es ejecutar y confirmar cada una de las siguientes instrucciones:

```
yum install nfs-utils
yum install openmpi
yum install opensshssh-clients
```

yum install openssh-server

Una vez instalados se requiere la configuración de ciertos servicios como SSH y NFS. La configuración de SSH facilita muchas tareas ya que se puede realizar toda la configuración desde un solo nodo remoto sin necesidad de comprar tantas pantallas como nodos se poseen o estar permanentemente conectando y desconectando una sola pantalla para configurar los nodos.

15.1.5.1 Configuración de SSH para Sesiones Remotas de Usuario.

Una vez estén instalados los servicios de SSH o OpenSSH se debe generar en cada nodo una llave pública y luego exportarla al nodo principal para que este tenga la posibilidad de acceder remotamente a los entornos de configuración. Este procedimiento puede ser llevado a cabo manualmente en cada nodo o de forma remota mediante el uso de SSH.

En cada usuario de los nodos en que se pretendan usar, se debe ejecutar el siguiente comando:

ssh-keygen -t rsa

Este comando genera la llave pública y privada mediante el uso de criptografía RSA, existen otros sistemas como DSA pero estos están obsoletos y solo se deben usar cuando se tienen sistemas antiguos que soporten únicamente criptografía de ese tipo.

El sistema preguntará la ubicación de las llaves y propondrá una ubicación por defecto la cual es recomendable aceptar al presionar ENTER o RETURN. De igual forma, cuando se pregunte por una contraseña se dejará en blanco y solo se presionará ENTER o RETURN, esto con el fin de permitir el acceso

transparente e inmediato a los nodos. La siguiente es la salida que da como resultar la ejecución del comando anterior.

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id_rsa):

Enter passphrase (empty for no passphrase):

Enter passphrase again:

Your identification has been saved in /root/.ssh/id_rsa.

Your public key has been saved in /root/.ssh/id_rsa.pub.

Una vez se hayan creado las llaves, es necesario enviar la llave pública al nodo principal, lo que se puede hacer de forma física copiando el archivo ***id_rsa.pub*** en un medio de almacenamiento y luego concatenar este en el nodo maestro en el archivo ***~/.ssh/authorized_keys***. Es importante usar el comando “cat” para esto, ya que copiar y pegar mediante la interfaz grafica ha probado ignorar ciertos parámetros.

La otra forma de copiar las llaves es mediante el comando ***ssh-copy-id***. Para llevar a cabo este procedimiento desde el nodo en el que se generó la llave se ejecuta:

ssh-copy-id -i ~/.ssh/id_rsa.pub root@pelusa

Este proceso se tiene que llevar a cabo en todos y cada uno de los nodos y usuarios de los nodos que se quieran usar hacia él, o los usuarios del nodo maestro que se desee usar.

Si se espera hacer uso de SSH para evitar el desplazamiento nodo por nodo, es solo cuestión de ejecutar SSH usuario@servidor el cual solicitará la contraseña del usuario y advertirá que se desconoce la autenticidad del

servidor. En el siguiente ejemplo se hizo desde y hacia el usuario root, pero podría y se debería llevar a cabo desde un usuario diferente.

ssh root@pelusa

Luego de esto se procede de la misma forma que se haría si se estuviere físicamente frente a cada nodo, al finalizar es necesario reiniciar el sistema o reiniciar el servicio SSH, mediante las siguientes instrucciones:

para reiniciar:

shutdown -r now

luego para salir de la sesión SSH:

logout

Si se hizo desde un nodo diferente:

/etc/111ell.d/ssh restart, ó, service ssh restart

Finalmente para probar que todo salió como debía ser, se ejecuta desde el nodo principal ssh nombreDelNodo. En este ejemplo, desde el nodo pelusa:

ssh rufian

Si no se solicita contraseña y el “prompt” muestra el nombre del nodo, es indicación de que todo resultó correctamente.

15.1.5.2 Configuración de NFS para Montaje de Carpetas Remotas Compartidas.

En el caso de NFS se requiere modificar el archivo ***/etc/exports*** el que especifica los directorios que se pretenden compartir, los cuales contienen los recursos y programas que se pretenden ejecutar en el clúster.

```
/mpiexec rufian(rw, sync, no_root_squash)  
auyin(rw, sync, no_root_squash)
```

El primer parámetro especifica la ruta de la carpeta que se desea compartir, los siguientes parámetros especifican los huéspedes o HOSTS que pueden acceder al recurso y entre paréntesis los parámetros que especifican como se deben compartir como lo son RW para lectura y escritura, SYNC para esperar a que se realicen definitivamente los cambios antes de permitir otros (esto puede reducir un poco el desempeño pero incrementa la fiabilidad).

En el caso de los nombres rufian y auyin estos corresponden a direcciones IP en el archivo ***/etc/hosts***, es posible usar únicamente direcciones IP pero esta forma podría simplificar ciertos cambios futuros en la red. Es posible también establecer parámetros generales para compartir archivos a toda la red de la siguiente forma:

```
/mpiexec 192.168.2.0/255.255.255.0(rw, sync, no_root_squash)
```

también

/mpiexec 192.168.2.0/24(rw,sync,no_root_squash)

15.1.5.3 Configuración del Montaje Automático Durante el Arranque de Carpetas Remotas.

Si todo se deja de esta forma sería necesario ir nodo por nodo montado la carpeta `/mpiexec` o peor aún, compilar o copiar el archivo ejecutable nodo por nodo manualmente lo que sería un trabajo extremadamente tedioso. Es por ello que se requiere automatizar el montaje de estos recursos ya sea mediante FSTAB o AUTOFS.

Dos de las formas más conocidas de montar remotamente y de forma automática carpetas o discos sistemas de archivos, es mediante el uso de AUTOFS y FSTAB. En el caso de AUTOFS se configuran ciertos parámetros para que cuando se intente acceder a un recurso este se monte en la marcha, pero este método no se estudió por cuanto el uso de FSTAB es lo suficientemente claro y simple y el montaje de los recursos ocurre durante el arranque del sistema operativo.

Para usar este método se debe modificar el archivo ***/etc/fstab***. Es necesario ser muy cuidadoso al modificar este archivo para no cambiar accidentalmente parámetros que pueden arruinar el arranque del sistema, lo cual implicaría iniciar el sistema con el CD o DVD de instalación en modo de recuperación y corregir el archivo.

Para abrir el archivo se ejecuta lo siguiente dependiendo del editor de texto:

nano /etc/fstab, ó, ***vi /etc/fstab***

Al final se agregarán los parámetros para montar la carpeta */mpiexec* la que contendrá los ejecutables para correr en el clúster.

```
Pelusa:/mpiexec /mpiexec nfs rw,hard,intr 0 0
```

Pelusa es el nombre del nodo principal, que puede ser reemplazado al igual que otros parámetros antes mencionados por direcciones IP o por los nombres que se deseen.

Inicialmente se especifica la carpeta que el nodo principal pelusa está compartiendo, luego se determina el lugar en el nodo en que se desea montar la carpeta remota y seguidamente el tipo de sistema de archivos, en este caso NFS.

Más adelante se especifica el parámetro lectura escritura, el parámetro “hard” indica que el sistema se monta de tal forma que si el servidor no se encuentra disponible se espera que lo esté nuevamente, la opción contraria sería “soft”. El parámetro “intr” permite poder matar el proceso si es del caso.

15.1.5.4 Configuración del Orden de Inicio de los Servicios Durante el Arranque.

Si el montaje falla, puede ser debido al orden en el que se inician los servicios durante el arranque. En el caso de YellowDog, aunque los servicios de inicio de interfaces Ethernet e inalámbricas ocurren al inicio el soporte para redes 802.11g con encriptación WPA2 PSK, esto sucede porque el WICD manager se ejecuta posteriormente al montaje de sistemas de archivos NFS y debido a que la conexión de los nodos en este caso se llevo a cabo mediante una red inalámbrica es necesario modificar el orden de arranque de los servicios y ejecutar el servicio NETFS luego del servicio WICD.

Para llevar esto a cabo hay que determinar el nivel de ejecución de arranque, que en el caso de YellowDog es el 5.

El procedimiento consiste en dirigirse hacia */etc/rc5.d*, allí aparecen los enlaces simbólicos a los diferentes servicios que se inician durante un arranque en nivel 5. El formato es ***KXXnombreservicio*** o ***SXXnombreservicio*** donde XX indica el orden numérico en el que se inician.

Luego de verificar este orden, se procede en forma alfabética en caso de que dos o más servicios tengan el mismo número de orden. En este caso el servicio es el ***S25netfs*** y el ***S98wica***. Primero se nombrara el ***S25netfs*** a ***S98netfs*** procediendo así:

```
mv S25nfs S98netfs
```

Seguidamente se nombra el ***S98wica*** a ***S97wica*** de la siguiente forma:

```
mv S98wica S97wica
```

En teoría cualquier orden es correcto siempre y cuando el servicio WICA se ejecute antes que NETFS

15.1.5.5 Configuración OpenMPI.

Cuando se pretende desplegar un clúster de múltiples nodos mediante la implementación de MPI OpenMPI versión 1.2.5., es cuestión de modificar el archivo ***/usr/lib/openmpi/1.2.5-gcc/etc/openmpi-default-hostfile*** en el nodo principal, este archivo debe contener una lista de los nodos y la cantidad de unidades de procesamiento (núcleos o procesadores) disponibles. Dicha información se encuentra de esta forma:

auyin slots=2
rufian slots=2
pelusa slots=2

Aunque el Playstation 3 cuenta con 8 procesadores, se debe recordar que su arquitectura es asimétrica y aun no hay manera de hacer uso de los 7 coprocesadores como si estos fueran simétricos, es por esto que se definen 2 “spots” o ranuras por cada nodo, que equivalen a los dos núcleos del procesador principal o PPU. Para poder hacer uso de los coprocesadores es necesario estudiar a fondo la especificación MPI y los modelos de programación del CELL BEA.

15.1.6 Seguridad Básica.

En una implementación típica es necesaria la desactivación de Cortafuegos ó Firewall de los nodos para no interferir en los procesos que rodean la ejecución en paralelo y solo hacer uso del Firewall del nodo principal.

En estos casos esto no implica riesgos adicionales a los usuales por cuanto los nodos esclavos están aislados de del resto de la red. En el caso de clúster que hacen uso de terminales y computadores que realizan tareas adicionales y tienen tráfico de usuarios, es necesario mantener altos niveles de seguridad por cuanto estas herramientas exponen la red a más ataques al permitir el inicio de sesiones remotas, compartir de archivos y ejecución remota de programas.

15.1.7 Ejecución Paralela de Aplicaciones.

El código del programa que se pretende ejecutar es muy simple, hay otros ejemplos más complejos pero se consideró que este es bastante útil para los propósitos del proyecto, ya que muestra claramente los nodos sobre los que se ejecutó el programa. Este código es:

```

#include <stdio.h>
#include <mpi.h>

int main(int argc, char *argv[]) {
int numprocs, rank, namelen;
char processor_name[MPI_MAX_PROCESSOR_NAME];
MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
MPI_Get_processor_name(processor_name, &namelen);

printf("Process %d on %s out of %d\n", rank, processor_name,
numprocs);

MPI_Finalize();
}

```

Para llevar a cabo su ejecución, solo se debe copiar la fuente en un archivo .c y luego compilarlo tal como se muestra a continuación (si es necesario se debe ir a la carpeta **/usr/lib/openmpi/1.2.5-gcc/bin**):

```
mpicc -o hello hello.c
```

Una vez realizado esto, se copia el ejecutable en **/mpiexec**:

```
mv hello /mpiexec
```

Finalmente, cuando todos los servicios estén instalados y configurados, la ejecución de los programas se lleva a cabo, en el caso de OpenMPI, mediante el comando **mpirun** el cual especifica el ejecutable PRE-compilado, con uno de los compiladores

proprios de cada implementación MPI y la cantidad de núcleos o procesadores sobre los que se quiere ejecutar, tal como se enseña en la siguiente línea:

```
mpirun -np 8 /mpiexec/hello
```

16.1 DIAGRAMA DE CAPAS DE IMPLEMENTACIÓN DE CLUSTER

A continuación se muestran los diagramas de las capas de aplicaciones y software como también las capas físicas de interconexión tanto de los nodos esclavos como del nodo maestro de una propuesta de despliegue de CLUSTER. Las capas muestran los diferentes niveles de abstracción de un CLUSTER de procesamiento que esta en capacidad de integrarse a un GRID. Es necesario aclarar que este modelo es un modelo genérico que no necesariamente se tiene que adaptar a todas las necesidades y/o requerimientos de investigación de la Universidad, partiendo del hecho de la gran variedad de entornos en los que se puede usar el concepto de CLUSTER y GRID como lo pueden ser de procesamiento para entornos de computación de alto desempeño, alta disponibilidad para el manejo de concurrencia y grandes volúmenes de datos, almacenamiento, visualización para entornos como el de la astronomía y geología y sensores como los usados para la meteorología.

Las interconexiones para fines diferentes al educativo requieren como mínimo de conexiones Gigabit Ethernet ya que una conexión inferior se convertiría en un cuello de botella en la capacidad de procesamiento neto.



Figura 4. Diagrama de capas nodo maestro

En este diagrama se puede observar las capas que conformarían la implementación de un nodo maestro de un CLUSTER el cual es el rostro visible desde el exterior el cual abstrae la complejidad que hay detrás de él. Cada una de las capas representa los niveles de complejidad y abstracción los cuales van desde el mas bajo nivel representado por las interconexiones de red pasando por el sistema operativo y las API de programación hasta las aplicaciones de administración o que proveen facilidades de conexión desde el exterior.



Figura 5. Diagrama de capas nodos esclavos

Este diagrama guarda bastantes similitudes con el diagrama de capas del nodo maestro y muestra la parte que el primero oculta. La única capa adicional la representan las aplicaciones paralelas las cuales deben ser diseñadas para ejecutarse bajo alguna de las API que se muestran en el diagrama.

Aunque se puede ejecutar parte de la carga de dichas aplicaciones también en el nodo maestro este no sería un escenario ideal por cuanto la ejecución de estos podrían copar la capacidad de procesamiento del nodo maestro imposibilitándolo para recibir tareas o retrasar la obtención de los resultados .

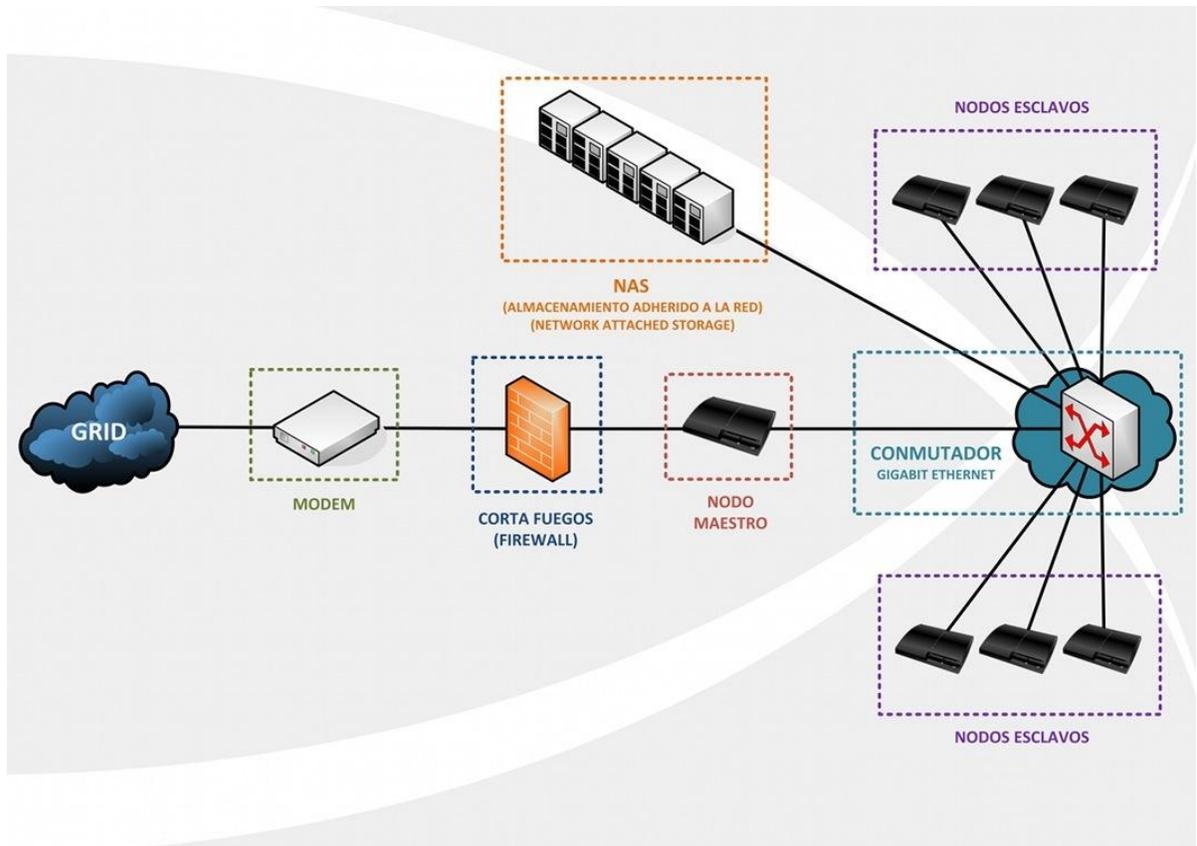


Figura 6. Diagrama de interconexión del CLUSTER

Este diagrama muestra detalladamente la nube de interconexión de los nodos y otros equipos de red que conforman el cluster. El uso de una red Gigabit Ethernet aunque no es obligatorio si es necesaria para obtener el mejor rendimiento y evitar que este sea el cuello de botella. El cortafuegos o firewall se encuentra por fuera del CLUSTER por cuanto este se supone es un entorno aislado dedicado, aunque es posible como ya se menciono instalar todas estas herramientas en computadores de escritorio y compartir su uso con usuarios, para aprovechar mejor la capacidad instalada al reducir el tiempo de ocio, usando el “hurgamiento de ciclos” pero en tal escenario se comprometería la seguridad del CLUSTER e incluso de un GRID al que este se conecte.

17 CONCLUSIONES

Hace aproximadamente un año cuando inició su trabajo el semillero de investigación sobre CLUSTERS y GRIDS, la idea de desplegar exitosamente un CLUSTER o racimo de computadores e integrarlo a un GRID o una malla computacional parecía ser una tarea extremadamente compleja hasta el punto de llegar a ser desmotivante, era un tema que parecía estar restringido a sectores científicos y militares, desviando de esta manera la atención de lo que debería ser el objetivo del mismo.

Durante los primeros días de investigación para el desarrollo de este proyecto se determinó casi instantáneamente que el despliegue de un CLUSTER o un GRID no es una tarea extremadamente compleja a pesar de los inconvenientes que se presenten y puede ser llevada a cabo en el transcurso de unos pocos días, ya que a pesar de que es un tema que aún se encuentra en exploración ha tenido bastantes progresos y tanto las herramientas como las guías para llevar a cabo los procesos de montaje se encuentran ampliamente disponibles.

De otro lado, se determinó que el mayor reto se encuentra en la comprensión de la arquitectura sobre la que se pretende desarrollar las aplicaciones, pero sobre todo el desarrollo de las mismas lo cual exige ciertas experticias que no son comunes, como también la identificación de necesidades claras que requieran el uso de amplias capacidades de procesamiento, lo que a nivel local pareciera ser el más grande inconveniente producto de la concepción errónea que se tiene de estos sistemas.

Algunas iniciativas y estudios pretenden el despliegue de infraestructuras de procesamiento distribuido sin tener clara una necesidad particular, haciendo un símil con infraestructuras como pueden ser las de telecomunicaciones o de transporte, esta visión errada se traduce generalmente en desperdicios de tiempo y dinero debido a la forma en que se plantean.

Por otra parte, si se fuera bastante riguroso en el rendimiento, claramente hay una desventaja al reutilizar componentes viejos por cuanto estos nodos pueden demorar la ejecución en paralelo de ciertas tareas, tal como lo afirman los estudios de Gene Amdalh, por cuanto el rendimiento se reduciría a la velocidad del nodo más lento. Sin embargo, esto también depende de qué tipo de CLUSTER se pretenda desplegar ya que aplicaciones de almacenamiento o que provean resistencia a fallos y alta disponibilidad pueden no ser tan afectadas como CLUSTER de procesamiento puro donde se ejecuten tareas o cálculos interrelacionados.

Para llevar a cabo el despliegue de un CLUSTER se requiere de distintos niveles de MIDDLEWARE que dependen del tipo de CLUSTER que se pretenda desplegar como también a las tareas a ejecutar, el nivel de acceso y los servicios que se quieren prestar.

No existe una plataforma o una arquitectura que se ajuste a todas las necesidades por lo que es fundamental definir las prioridades y con base en ellas establecer qué herramientas, plataformas o arquitectura se ajusta mejor a estas. Así mismo, toda solución debe estar adecuada al tipo de aplicaciones que se piensan ejecutar sobre el CLUSTER. Por ejemplo, en el caso del PlayStation 3 el límite más claro es la reducida cantidad de memoria RAM por lo que las aplicaciones deben ser diseñadas para este tipo de restricciones.

En su nivel más básico, solo se requiere de la instalación de MIDDLEWARE como MPI o PVM de la forma en que se llevó a cabo en este estudio. Dicho sistema puede servir desde propósitos meramente académicos, hasta investigaciones dedicadas sin tráfico adicional.

Si se quiere realizar una variedad de tareas que puedan concurrir, es necesario el uso de MIDDLEWARE de administración de recursos y carga como Condor, aunque el despliegue de Clusters basados en hurgamiento de ciclos con herramientas como estas, no resultan muy convenientes cuando son ejecutadas sobre plataformas inseguras que deben ser compartidas con usuarios que realizan tareas, ya que estos

pueden comprometer la integridad del sistema lo cual pondría en peligro no solo el CLUSTER sino cualquier GRID del que este haga parte, al igual que podría afectar los resultados de los procesos que se lleven a cabo cuestionando sus resultados.

Herramientas como Condor, Sun Grid Engine o Torque, tan solo componen la capa de administración de recursos, pero si se pretende desplegar CLUSTERS mas complejos, que manejen altos niveles de uso, se requiere también de administradores de carga como el Moab y calendarizadores avanzados como el Maui o el mismo Moab, que proveen mejores niveles de administración de carga y por la tanto un uso más eficiente del sistema, aumentando los niveles de ocupación.

Si se tiene previsto proveer el acceso desde el exterior es necesario el despliegue de servidores, portales y herramientas de visualización y envío de trabajos y si se pretende integrar el Cluster a redes de colaboración o GRIDS es necesario el uso de herramientas como Globus Toolkit, que aunque no son únicas, han liderado el desarrollo de estas tecnologías y son abiertas.

El PlayStation 3 provee grandes potencialidades de procesamiento llegando a competir con CLUSTERS de hasta 32 nodos de computadores genéricos y aunque para poder acceder a todas sus capacidades de procesamiento se requiere del manejo de ciertas técnicas de programación distribuida, estas no son distintas a las técnicas que se requerirían para desplegar aplicaciones sobre CLUSTERS de computadores genéricos. Por todo, esto es necesario que se estudien profundamente los conceptos, las técnicas y las herramientas de desarrollo sobre sistemas distribuidos y de procesamiento paralelo, ya que la mayor dificultad recae precisamente en el desarrollo sobre dichos sistemas más no en el despliegue de CLUSTERS o racimos de computadores y su integración en GRIDS o mallas de computación.

Finalmente, se corroboró que es viable realizar el despliegue de un CLUSTER usando el concepto de CLUSTER Beowulf mediante el uso de componentes genéricos, en este caso de consolas de videojuego PlayStation 3. Las pretensiones

de desempeño del mismo están ligadas a las necesidades que se busquen satisfacer, ya que el PlayStation 3 provee amplias capacidades de procesamiento y excelente desempeño en la red a un costo relativamente bajo.

Aunque hubiera sido ideal el desarrollo de una aplicación de prueba que haga uso de los coprocesadores disponibles en el sistema implementado, esto no pudo ser llevado a cabo debido a las limitaciones de tiempo y personal, como también al hecho de que sería una investigación que requeriría dedicación exclusiva ya que implica el manejo de ciertos conceptos de programación avanzada. A pesar de ello, las evidencias del desempeño y la viabilidad del desarrollo sobre el procesador Cell BEA y sobre el PlayStation 3, han sido demostradas durante el transcurso de la investigación por cuanto muchas instituciones internacionales se encuentran trabajando sobre esta plataforma.

Las pretensiones de desempeño que se busquen cumplir están ligadas a los requerimientos que impongan las necesidades que se pretendan satisfacer y a la capacidad de inversión. Para propósitos estrictamente educativos es viable la interconexión mediante una red inalámbrica sin la necesidad de incurrir en el costo de adquisición de conmutadores de alto desempeño y características de administración avanzadas, por cuanto la configuración de las interconexiones no es diferente a la que se haría para desplegar una red casera. Si dado el caso se requiere de mejor desempeño, cada consola cuenta con puertos Gigabit Ethernet y en tales circunstancias se podría perfectamente llevar a cabo la interconexión mediante conmutadores profesionales capaces de manejar amplios anchos de banda.

18 RECOMENDACIONES

Intel ya ha advertido que en el futuro cercano y lejano se verán cada vez mas y mas núcleos o unidades de procesamiento en sus integrados, (Bright, 2008) (Crothers, 2008). Aunque ciertos sectores de la industria no ven con buenos ojos esta tendencia y plantean que a pesar de que se incluyan mayor cantidad de núcleos, muchas aplicaciones no podrán hacer uso de forma práctica de ellos, por cuanto el objetivo debe ir enfocado en la optimización de las aplicaciones (Edwards, 2008) (West, 2008) (Bright, 2008) lo que puede implicar nuevos lenguajes con mejor desempeño a bajo nivel o incluso el hecho de volver a programar en lenguaje ensamblador.

A pesar de estas reservas la computación distribuida de alto desempeño será de hecho el único tipo de computación o por lo menos los paradigmas de programación inherentes a este. Desafortunadamente la educación tecnológica en la Universidad no ha hecho ningún énfasis en el desarrollo o la programación sobre sistemas distribuidos o paralelos, de hecho tampoco se ha enseñado la programación multihilo. Esto pone a los estudiantes en desventaja frente a los cambios que se avecinan en la industria y ponen en duda la relevancia del estudio de los GRIDS y CLUSTERS aisladamente, sin tener en cuenta estudios sobre las técnicas de programación afines. Este es un síntoma global (Fried, 2007) (Krazit, 2007) (Sutter, 2005) por lo que las habilidades para programar en sistemas distribuidos se hacen necesarias a corto plazo para seguir el ritmo de los desarrollos de la industria.

Con base a estos planteamientos se recomienda en primer lugar fortalecer la enseñanza no solo de las técnicas de programación paralela sino la enseñanza general de programación, la que ha sido en muchos casos pobre, ya que esta es el lenguaje común de entendimiento de todas las carreras afines a la computación e incluso los desarrollos en telecomunicaciones se pueden beneficiar del fortalecimiento de las habilidades de programación.

Por otro lado, se recomienda que no solo se exijan horas de participación en los semilleros de investigación sino el cumplimiento de los objetivos que se definan para cada uno de ellos. Al igual que la apertura de semilleros de investigación variados que satisfagan las ambiciones y aptitudes de los diferentes perfiles estudiantiles donde se pueda profundizar sobre las materias y temas definidos en el pensum.

Para las instituciones que se encuentran en desventaja económica frente a los requerimientos que puede traer la profundización de determinados temas y el estudio de ciertas tecnologías, existen alternativas como la mostrada con este proyecto, que permiten llevar a cabo dichos estudios mediante la adquisición de tecnologías alternativas y de bajo costo que aunque pueden no ser aptas para tareas de investigación aplicada, si pueden servir para procesos de aprendizaje.

En un futuro se podría plantear como alternativa a adquirir computadores para las salas de cómputo, que en la mayoría de los casos terminan siendo subutilizados, llegar a acuerdos con fabricantes para realizar compras de computadores en altos volúmenes a precio de mayoreo y financiar su adquisición a los estudiantes durante el transcurso de la carrera. De esta forma se puede invertir en elementos que sean de mayor utilidad para fines educativos como lo pueden ser licencias de software, conmutadores y enrutadores para prácticas de telecomunicaciones y laboratorios de electrónica digital más completos, entre otros.

Uno de los principales inconvenientes durante la investigación de este proyecto fue el acceso a libros, tesis y publicaciones tecnológicas las cuales en muchos casos estaban disponibles a altos costos, lo que implicó la obtención de estos mediante formas diferentes a la adquisición directa. Los libros de tecnología tienden a volverse obsoletos en cortos periodos de tiempo y sus costos de adquisición y transporte son bastante altos por lo que alternativas como Kindle de Amazon se presentan como una solución muy conveniente, al eliminar los intermediarios y los costos de envío para la adquisición de libros. Al mismo tiempo la Universidad debe mantener suscripciones a sitios conocidos de publicaciones científicas y tesis de tal forma que los estudiantes puedan acceder transparentemente a ellos conforme surja la necesidad.

19 BIBLIOGRAFÍA

(n.d.). From http://www.historyofthecomputer.net/the_first_pda.html

(n.d.). From ps3cluster.org: <http://www.ps3cluster.org>

(n.d.). From Red Barn High Performance Computing:
<http://www.redbarnhpc.com/v2/index.php/cluster-computing/the-history-of-cluster-computing>

(n.d.). From Zii: <http://www.zii.com>

(n.d.). From Linux Scalability Effort: <http://lse.sourceforge.net/numa/faq/>

(n.d.). From <http://www.beowulf.org>: <http://www.beowulf.org/overview/index.html>

(n.d.). From GRID COLOMBIA: <http://urania.udea.edu.co/sites/grid-colombia/>

(18 de Julio de 2008). From PC Magazine:
<http://www.pcmag.com/article2/0,1895,2325943,00.asp>

(22 de Diciembre de 2008). From <http://www.physorg.com/news149192580.html>

Agarwal, V., Hrishikesh, M., Keckler, S. W., & Burger, D. (n.d.). Clock Rate versus IPC:

The End of the Road for Conventional Microarchitectures.

Amdahl, G. M. (1967). *Validity of the single processor approach to achieving large scale computing capabilities*. Sunnyvale, California, USA.

Anjomshoa, A., Brisard, F., Drescher, M., Fellows, D., Ly, A., McGough, S., et al. (7 de
de
Noviembre de 2005). Job Submission Description Language (JSDL) Specification,
Version 1.0. (A. Savva, Ed.)

Antwerp, A. T. (n.d.). <http://fastra.ua.ac.be/en/about.html>. From <http://fastra.ua.ac.be/en/index.html>

Apple. (n.d.). From <http://developer.apple.com/hardwaredrivers/hpc/mpionmacosx.html>

Apple-Inc. (n.d.). From http://images.apple.com/server/macosx/docs/L355779B_Xgrid_tb.pdf

Baran, P. (Septiembre de 1962). On Distributed Communications Networks.

Barney, B. (n.d.). From Lawrence Livermore National Laboratory: https://computing.llnl.gov/tutorials/parallel_comp/#SharedMemory

Bartlett, J. (3 de 10 de 2006). *Developer Works*. From IBM: <http://www.ibm.com/developerworks/linux/library/l-powasm1.html>

Bishop, K., & House, M. (1998). *RISCy Business: Intel's New IA-64 Architecture*. From <http://www.cs.umd.edu/>:

<http://www.cs.umd.edu/class/fall2001/cmsc411/projects/IA64-2/411project.htm>

Bowman, L. M. (11 de 1 de 2001). *ZDnet.co.uk*. Retrieved 22 de 04 de 2009 from Zdnet: <http://news.zdnet.co.uk/itmanagement/0,1000000308,2083632,00.htm>

Bright, P. (2 de Julio de 2008). From Arstechnica: <http://arstechnica.com/software/news/2008/07/intel-an-expensive-many-core-future-is-ahead-of-us.ars>

Broersma, M. (2 de 2 de 2004). *zdnet.co.uk*. From Zdnet UK: <http://news.zdnet.co.uk/hardware/0,1000000091,39145271,00.htm>

Brown, J. (6 de Diciembre de 2005). Just like being there: Papers from the Fall Processor Forum 2005: Application-customized CPU design The Microsoft Xbox 360 CPU story.

CDR Info. (15 de 12 de 2006). From <http://www.cdrinfo.com/>: <http://www.cdrinfo.com/sections/news/Details.aspx?NewsId=19298>

Cenicafe. (n.d.). *Genoma del Cafe*. From Cenicafe:

http://www.cenicafe.org/modules.php?name=Genoma_del_Cafe&lite=0

Chan, W., Ferstl, F., Gardiner, J., Haas, A., Nitzberg, B., Rajic, H., et al. (Junio de 2004). Distributed Resource Management Application API Specification 1.0 .

Cohen, J. (24 de Julio de 2008). <http://www.hec.nasa.gov/>. From nasa.gov:

<http://www.hec.nasa.gov/news/features/2008/cell.074208.html>

CONDOR Team. (n.d.). From <http://www.cs.wisc.edu/condor/>

CONDOR Team. (n.d.). From <http://www.cs.wisc.edu/condor/description.html>

CONDOR Team. (n.d.). From <http://www.cs.wisc.edu/condor/overview/>

CONDOR Team. (n.d.). From <http://www.cs.wisc.edu/condor/condorg/versusG.html>

Cote, D. (n.d.). From <http://unu.novajo.ca/simple/archives/000026.html>

Crothers, B. (1 de Julio de 2008). *Cnet News*. From Cnet: http://news.cnet.com/8301-13924_3-9981760-64.html

cse.stanford.edu. (n.d.). From Standford University:

<http://cse.stanford.edu/class/sophomore-college/projects-00/risc/riscisc/>

dgohara. (11 de Febrero de 2007). From <http://www.macresearch.org/openmacgrid>

Diefendorff, K., & Dubey, P. K. (1997). How Multimedia Workloads Will Change Processor Design. *Computer* , 30 (9), 43-45.

Edwards, J. (24 de Agosto de 2008). From alarmingdevelopment:

<http://alarmingdevelopment.org/?p=117>

Erl, T. (28 de Julio de 2007). From <http://www.soapprinciples.com/p1.asp>

Evans, D. (n.d.). <http://www.techradar.com>. From TechRadar:

http://www.techradar.com/news/computing/pc/why-your-next-cpu-could-be-a-gpu-585876?artc_pg=1

Farah, J. (19 de 12 de 2000). *Why Iraq's buying up Sony PlayStation 2s Intelligence experts fear games bundled for military applications*. From http://www.worldnetdaily.com/news/article.asp?ARTICLE_ID=21118

Fildes, J. (9 de Junio de 2008). *BBC News*. From BBC: <http://news.bbc.co.uk/1/hi/technology/7443557.stm>

foldings.stanford.edu. (22 de 04 de 2009). From Stanfor University: <http://fah-web.stanford.edu/cgi-bin/main.py?qtype=osstats>

Foster, I. (n.d.). *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*.

Foster, I., & Kesselman, K. (2003). *THE GRID: BLUEPRINT FOR A NEW COMPUTING INFRASTRUCTURE* (2 ed.). Morgan Kaufmann.

Foster, I., Kesselman, C., Nick, J. M., & Tuecke, S. (22 de Junio de 2002). *The Physiology of the Grid An Open Grid Services Architecture for Distributed Systems Integration* .

Fried, I. (25 de Mayo de 2007). *Cnet*. From <http://news.cnet.com>: http://news.cnet.com/Intel-Software-needs-to-heed--Moore's-Law/2100-1012_3-6186765.html?tag=mncol;txt

Gaudin, S. (9 de Junio de 2008). From http://www.computerworld.com/action/article.do?command=viewArticleBasic&taxonomyName=hardware&articleId=9095318&taxonomyId=12&intsrc=kc_top

Getov, V. (Noviembre de 2008). *e-Science: The Added Value for Modern Discovery*. *Computer.org* .

Goldstein, H. (16 de Noviembre de 2006). *Sony Losing Almost \$250 per Console*. From <http://ps3.ign.com/articles/746/746482p1.html>

Goodale, T., Jha, S., Kaiser, H., Kielmann, T., Kleijer, P., Merzky, A., et al. (9 de Mayo de 2007). A Simple API for Grid Applications (SAGA).

gravity.phy.umassd.edu. (n.d.). From umassd.edu:
<http://gravity.phy.umassd.edu/ps3.html>

Gropp, W., & Lusk, E. (n.d.). PVM and MPI Are Completely Diferent.

GT 2.4: The Globus Resource Specification Language RSL v1.0. (n.d.).

GUSTAFSON, J. L. (1988). *Communications of the ACM* , 31 (25).

Hayes, S. (26 de Noviembre de 2005). *Xbox 360 initial production costs may be steep*. From <http://www.psu.com/Xbox-360-initial-production-costs-may-be-steep-News--a000518-p0.php>

Hodgin, R. (10 de 2 de 2006). From <http://www.geek.com/articles/chips/what-is-a-processors-pipeline-20060210/>

Hodgin, R. (29 de 4 de 2008). <http://www.geek.com>. From Geek:
<http://www.geek.com/articles/chips/nvidia-believes-gpu-will-overtake-cpu-as-future-heart-of-computer-amd-hopes-so-too-20080429/>

Hof, R. D. (13 de Noviembre de 2006). From businessweek:
http://www.businessweek.com/magazine/content/06_46/b4009001.htm

Hofstee, D. H. (24 de Agosto de 2005). Cell Broadband Engine Architecture from 20,000 feet.

Hofstee, D. H. (25 de 8 de 2005). *IBM*. From ibm.com:
<http://www.ibm.com/developerworks/power/library/pa-cbea.html>

Huang, E. (23 de 3 de 2003). <http://www.gamepro.com>. From GAMEPRO:
<http://www.gamepro.com/article/news/107328/ps3-doubles-folding-homes-processing-power/>

IBM. (n.d.). From IBM: <http://www-03.ibm.com/systems/bladecenter/hardware/servers/qs21/index.html>

IBM. (12 de Noviembre de 2007). *IBM's Blue Gene Pulls Away from the Pack Fastest Computer in the World Laps the Field As IBM Guns for Petaflop Barrier*. From <http://www-03.ibm.com/press/us/en/pressrelease/22579.wss>

Jobe, D. (8 de Agosto de 2007). From <http://blog.us.playstation.com/>:
<http://blog.us.playstation.com/2007/08/08/behind-the-curtain-the-warhawk-servers/>

Kahle, J. A., Day, M. N., Hofstee, H. P., Johns, C. R., Maeurer, T. R., & Shippy, D. (2005). Introduction to the Cell multiprocessor. *IBM Journal of Research and Development* , 49 (4/5).

Kanellos, M. (15 de 10 de 2004). From Zdnet.co.uk:
<http://news.zdnet.co.uk/hardware/0,1000000091,39170358,00.htm>

Kanellos, M. (1 de Diciembre de 2003). *Cnet News*. From Cnet:
<http://news.cnet.com/2100-1008-5112061.html>

Khanna, D. G. (19 de 2 de 2007). *North Carolina State University*. From <http://ncsu.edu/>: <http://moss.csc.ncsu.edu/~mueller/cluster/ps3/coe.html>

King, I. (9 de 4 de 2007). *seattletimes.nwsourc.com*. From Seattle Times:
http://seattletimes.nwsourc.com/html/business/technology/2003658346_intelisrael09.html

Krazit, T. (13 de Agosto de 2007). <http://news.cnet.com/>. From Cnet:
http://news.cnet.com/8301-13579_3-9759153-37.html

Leitner, F. v. (2003). From <http://bulk.fefe.de/scalability/>

Müller, R. (07 de 2005). <http://www.inf.ethz.ch>. From ETH Zürich (Eidgenössische Technische Hochschule Zürich):
<http://www.inf.ethz.ch/personal/muellren/pendulum/index.html>

Maesschalck, T. D. (29 de 5 de 2008). *dvhardware*. From <http://www.dvhardware.net:>
<http://www.dvhardware.net/article27538.html>

Mahapatra, N. R., & Venkatrao, B. (n.d.). From <http://www.acm.org/crossroads/xrds5-3/pmgap.html>

Mainelli, T. (16 de 8 de 2002). *PCWorld.com*. From PCWorld:
http://www.pcworld.com/article/104075/intel_pc_makers_sued_over_p4_performance.html

Manrique, D. (15 de Mayo de 2002). Construcción y evaluación de desempeño de un cluster tipo Beowulf para cómputo de alto rendimiento.

Martin, P. (18 de 4 de 2000). <http://www.abc.net.au>. From <http://www.abc.net.au:>
<http://www.abc.net.au/pm/stories/s119754.htm>

McCarthy, J. (1992). " 1959 Memorandum," *IEEE Annals of the History of Computing*. 14 (1).

Merkey, P. (n.d.). From Beowulf.org: <http://www.beowulf.org/overview/history.html>

Miller, P. (28 de 8 de 2008). <http://www.engadget.com/>. From Engadget:
<http://www.engadget.com/2008/08/28/toshibas-new-zf-lclds-include-cell-based-resolution-upscaling/>

Mueller, P. F. (21 de Febrero de 2009). From moss.csc.ncsu.edu:
<http://moss.csc.ncsu.edu/~mueller/cluster/ps3/>

Murph, D. (11 de Agosto de 2007). From Engadget:
<http://www.engadget.com/2007/08/11/sony-erects-massive-ps3-server-cluster-for-warhawk-mayhem/>

Nakada, H., Matsuoka, S., Seymour, K., Dongarra, J., Lee, C., & Casanova, H. (Julio de 2002). GridRPC: A Remote Procedure Call API for Grid Computing.

Novakovic, N. (7 de Agosto de 2007). From
<http://www.theinquirer.net/inquirer/news/687/1008687/dram-latency-explained>

Novakovic, N. (28 de Enero de 2008). From The Inquirer:
<http://www.theinquirer.net/inquirer/review/045/1022045/first-amd-dual-gpus-asus>

Pande, V. (17 de 9 de 2008). *foldings.stanford.edu*. From Stanford University:
<http://folding.stanford.edu/English/FAQ-NVIDIA>

Patrick Goldsack, H. P. (Ed.). (n.d.). CDDL Configuration Description, Deployment and Lifecycle Management . 2005, Septiembre, 9.

PC Magazine Encyclopedia. (n.d.). From PC Magazine:
http://www.pcmag.com/encyclopedia_term/0,2542,t=NUMA&i=48164,00.asp

Perez, J. M., Bellens, P., Badia, R. M., & Labarta, J. (Enero de 2007). CellSs: Making it easier to Program the Cell Broadband Engine processor. *51* , 5. (I. J. R&D, Ed.)

Pfister, G. (1997). *In Search of Clusters* (2 ed.). Austin, Texas, USA: Prentice Hall PTR.

Prabhakar, E. N. (1 de Agosto de 1995). From
<http://www.ddj.com/architect/184409611>

RELEASE, C. P. (8 de 9 de 1994). *CERN celebrates 40th Anniversary*. From CERN:
<http://press.web.cern.ch/press/PressReleases/Releases1994/PR11.94E40thAnni.html>

Shi, Y. (Octubre de 1996). From
<http://www.cis.temple.edu/~shi/docs/amdahl/amdahl.html>

Shilov, A. (21 de 01 de 2004). <http://www.xbitlabs.com>. From Xbit Labs:
<http://www.xbitlabs.com/news/cpu/display/20040121203014.html>

Singh, H. L. (5 de Diciembre de 2001). From
<http://cocoadevcentral.com/articles/000062.php>

Smith, T. (1 de 7 de 2001). <http://www.theregister.co.uk/>. From The Register:
http://www.theregister.co.uk/2001/07/03/sony_pulls_playstation_2_production/

Spooner, J. (6 de Agosto de 2002). *Zdnet News*. From Zdnet:
http://news.zdnet.com/2100-9584_22-195279.html

Sprangle, E., & Carmean, D. (Mayo de 2002). Increasing Processor Performance by Implementing Deeper Pipelines. *ACM SIGARCH Computer Architecture News* .

ST-Ericsson. (16 de Febrero de 2009). From
<http://news.prnewswire.com/DisplayReleaseContent.aspx?ACCT=104&STORY=/www/story/02-16-2009/0004972700&EDATE=>

Stewart, J. (27 de Noviembre de 2005). An Investigation of SIMD instruction sets. (U. o. Sciences, Ed.)

Stokes, J. (21 de Marzo de 2000). From Arstechnica:
<http://arstechnica.com/old/content/2000/03/simd.ars>

Stokes, J. (12 de 2002). From Arstechnica: <http://arstechnica.com/paedia/c/cpu/part-2/cpu2-3.html>

Stokes, J. (27 de Marzo de 2007). From Arstechnica:
<http://arstechnica.com/hardware/news/2007/03/why-the-playstation-3-owns-the-pc-in-fh.ars>

Stokes, J. (7 de Febrero de 2005). *Introducing the IBM/Sony/Toshiba Cell Processor ? Part I: the SIMD processing units*. From arstechnica:
<http://arstechnica.com/old/content/2005/02/cell-1.ars>

Stokes, J. (Octubre de 1999). *RISC vs. CISC: the Post-RISC Era*. From
<http://arstechnica.com: http://arstechnica.com/cpu/4q99/risc-cisc/rvc-1.html>

Stutz, M. (17 de Agosto de 1998). From Wired:
<http://www.wired.com/science/discoveries/news/1998/08/14450>

Sutter, H. (1 de Marzo de 2005). *Dr. Dobb's Portal*. From <http://www.ddj.com>:
<http://www.ddj.com/web->

development/184405990;jsessionid=VLJ1O0GC41XDAQSNDLOSKHSCJUNN2JVN?
_requestid=4633612

Taylor, J. (2008). *Defining e-Science*. From <http://www.nesc.ac.uk/nesc/define.html>

Thorpe, A. (Marzo de 13 de 2009). *Environmental eScience*. From Royal Society Publishing: <http://rsta.royalsocietypublishing.org/content/367/1890/801.full>

Times, T. N. (18 de 12 de 2000). *NYTIMES.COM*. From The New York Times: <http://search.nytimes.com/2000/12/28/technology/28STAT.html>

Valdés, R. (1 de Enero de 1994). From <http://www.ddj.com/184409416?pgno=3>

Wearden, G. (7 de 9 de 2000). *Zdnet.co.uk*. From ZDnet: <http://news.zdnet.co.uk/emergingtech/0,1000000183,2081275,00.htm>

Wesner, S., Jähnert, J. M., & Escudero, M. A. (n.d.). Mobile Collaborative Business Grids – A short overview of the Akogrimo Project.

West, J. (25 de Agosto de 2008). From inside hpc: <http://insidehpc.com/2008/08/25/performance-is-not-the-problem-a-vote-against-multicore/>

What is a Computer Cluster? (14 de Febreo de 2006). From UK National Grid Service

(NGS): Carl Kesselman, Steve Tuecke

Wilson, G. V. (28 de 10 de 1994). *Virginia Tech University*. From <http://www.vt.edu/~ei.cs.vt.edu/~history/Parallel.html>

ZDNET Nweus. (10 de Marzo de 2006). From ZDNET: <http://news.zdnet.co.uk/hardware/0,1000000091,39256676,00.htm>

Zhirnov, V., III, R. C., Hutchby, J., & Bourianoff, G. (2003). Limits to binary logic switch scaling - a gedanken model. *Proceedings of the IEEE*, 91 (11).

20.1 DESPLIEGUE DE UN CLUSTER E INTEGRACIÓN A UN GRID

Aunque en los objetivos iniciales no se había considerado la puesta en marcha de un CLUSTER o un GRID, se consideró realizar una implementación a pequeña escala o prototipo para poder llevar a cabo un estudio más a fondo y aclarar ciertos aspectos relacionados con su despliegue y comportamiento.



Figura 7. Prototipo

En la figura 7 es posible observar la estructura general del Cluster, donde el nodo maestro está compuesto por una consola PlayStation 3, modelo de introducción de 60 GB con lector de tarjetas SD MemoryStick y compact flash. El procesador fue elaborado con técnicas de producción de 90nm. Los otros dos nodos esclavos están compuestos por PlayStation 3 de 80 GB más recientes con chips elaborados bajo técnicas de 65nm por lo que generan menos calor y por tanto sus ventiladores son menos ruidosos.

Estos últimos a diferencia del nodo principal no cuentan con lectores de tarjetas de memoria como tampoco con el chip del PlayStation 1 y 2 para mantener la

compatibilidad con los juegos de estos, sin embargo, esto es irrelevante para el montaje del clúster.

Todas las consolas cuentan con tarjetas inalámbricas 802.11g, así como con Bluetooth y a pesar de tener puertos Gigabit Ethernet la interconexión se llevo a cabo inalámbricamente mediante un enrutador Belkin FD8230-4 Pre-N MIMO.

El PlayStation 3 cuenta con dos salidas de video una HDMI y AV/OUT propietaria de Sony. De igual manera, al ser también un reproductor de películas en Blu-Ray tiene sistemas de protección de contenido por lo que es necesario usar pantallas que soporten protección contenido digital de alto ancho de banda o HDCP (High-bandwidth Digital Content Protección), de lo contrario no será posible la visualización. La interconexión puede ser hecha ya sea mediante HDMI de extremo a extremo o HDMI-DVI. En este caso se uso una pantalla Hewlett Packard W2207h.



Figura 8. Diagrama Prototipo Cluster

Es claro que las herramientas para el montaje de un CLUSTER y la integración a GRIDS están lo suficientemente maduras y las guías sobre su aplicación son comunes y fáciles de encontrar. Sin embargo, durante las pruebas sobre los sistemas PlayStation 3 se presentaron algunos inconvenientes debidos a que estas no se han probado exhaustivamente sobre otras arquitecturas diferentes a X86, pero sobre todo por las deficiencias de la distribución de GNU/Linux Yellow Dog utilizada.

A pesar de esto, el montaje del clúster se llevó a cabo exitosamente y todos sus nodos ejecutaron alguna parte de la carga de procesamiento tal como se observa en la figura 9.

```
root@pelusa:/mpirexec
Archivo Editar Ver Terminal Solapas Ayuda
lower performance.
-----
[0,1,3]: OpenIB on host rufian was unable to find any HCAs.
Another transport will be used instead, although this may result in
lower performance.
-----
[0,1,3]: uDAPL on host rufian was unable to find any NICs.
Another transport will be used instead, although this may result in
lower performance.
-----
[0,1,2]: uDAPL on host rufian was unable to find any NICs.
Another transport will be used instead, although this may result in
lower performance.
-----
Process 4 on pelusa.perrera.local out of 8
Process 5 on pelusa.perrera.local out of 8
Process 2 on rufian.perrera.local out of 8
Process 0 on auyin.perrera.local out of 8
Process 1 on auyin.perrera.local out of 8
Process 6 on auyin.perrera.local out of 8
Process 7 on auyin.perrera.local out of 8
Process 3 on rufian.perrera.local out of 8
[root@pelusa mpirexec]#
```

Figura 9. Comportamiento del Cluster

Durante la preparación del CLUSTER se trató de integrar un nodo compuesto por un sistema Mac OS X Leopard corriendo sobre Imac G4 de Apple, buscando crear un clúster pseudo heterogéneo. A pesar de que todas las herramientas se instalaron con éxito, incluso OpenMPI, durante las pruebas no se logró ejecutar carga de procesamiento sobre este nodo a pesar de ser también PowerPC.

Es muy prematuro sacar más conclusiones sobre la naturaleza del problema, pero una de las posibles razones es el hecho de que Mac OS X Leopard dejó de soportar el sistema FSTAB para el montaje automático de carpetas remotas y fue necesario recurrir a copiar el ejecutable compilado en la misma carpeta que se pretendía montar, pero compilada sobre la misma máquina.

Por otro lado se contaba con una versión más antigua de OpenMPI que viene instalada por defecto y esto pudo haber causado alguna incompatibilidad. No fue posible entonces continuar con las pruebas sobre esta máquina debido a que era el computador principal sobre el que se estaba elaborando el presente documento y se prefirió no realizar modificaciones que pudieran causar comportamientos extraños.

Aunque puede parecer obvio, no sobra decir que los programas que se pretenden ejecutar deben ser diseñados teniendo en cuenta los paradigmas de paralelismo y mediante el uso de las librerías y funciones de MPI, ya que algunas veces se puede tener la idea errónea sobre los requerimientos para ejecutar un programa de forma paralela o hacer uso de un clúster, y aunque sería muy interesante el estudio de este tema específico no hace parte de los objetivos iniciales de este trabajo ya que por sí solo es meritorio de un desarrollo profundo y exclusivo.

A pesar de que durante el desarrollo de este documento se expresó y se sustentó la inconveniencia de proponer o llevar a cabo el diseño de una arquitectura o el despliegue de un GRID por una sola persona, se decidió desplegar un GRID de prueba siguiendo las instrucciones publicadas por el grupo de trabajo de Globus Toolkit, únicamente para propósitos educativos y de familiarización con las herramientas presentes en este sistema.

Durante este proceso se presentaron algunos problemas de los que se pueden extraer conclusiones que no dejan de ser útiles, ya que reflejan inconvenientes que se podrán tener en cuenta en futuros desarrollos.

Inicialmente es necesario decir que la guía propuesta por Globus Toolkit no se encuentra totalmente actualizada, ya que algunas de las herramientas han cambiado y con ellas sus configuraciones y opciones. De igual forma, al depender la guía de archivos binarios precompilados, es limitada respecto a la cantidad de plataformas y sistemas operativos sobre los que se pueden instalar. A pesar de esto, se pueden instalar luego de solventar algunos problemas, ya que cuentan con la ventaja de ser herramientas libres cuyo código fuente está disponible y además poseen listas de correo con gente dispuesta a ayudar en la resolución de inconvenientes como los mencionados.

La guía pretendía simular en dos computadores los nodos maestros de dos CLUSTERS diferentes, por cuanto el concepto de CLUSTER esconde los nodos esclavos y se presenta como un solo computador. Uno de los nodos haría uso del

administrador de carga Torque PBS y el otro de Sun Grid Engine y todos contendrían la instalación de las herramientas del GLOBUS TOOLKIT.

La instalación de Torque presentó varios inconvenientes que fueron solucionados gracias a la ayuda de los miembros de la lista de correo. Finalmente la instalación se llevó a cabo con éxito pero en el caso de Sun Grid Engine se presentaron más inconvenientes, a pesar de que aparentemente ya había soporte para instalaciones sobre PowerPC tanto de 32 como de 64 bits.

Desafortunadamente no existían binarios para esta arquitectura y sistema operativo, excepto aquellos distribuidos por Fixstars y YDL.NET en paquetes de herramientas y soporte que tiene un costo de 6000 dólares. Debido a esto se decidió descargar el código fuente y realizar la compilación para el sistema pero se presentaron una serie de errores durante la configuración de instaladores y su posterior compilación causados por ciertas librerías que modificaron componentes y rompieron la instalación de herramientas que dependían de ellas.

Dichos problemas afectaron otras plataformas pero las más perjudicadas fueron aquellas en las que no se han realizado pruebas a gran escala. Finalmente se pudieron solucionar cierta cantidad de errores al descargar las fuentes mediante CVS ya que solo de esta forma se logró acceder al código fuente mas reciente.

A pesar de esto era necesario llevar a cabo modificaciones al código fuente de los instaladores y archivos de configuración, tarea que no fue posible realizar por los errores presentados en FLEX y Linux PAM y al depender el uno del otro no se pudieron compilar exitosamente.

Respecto a la instalación de las herramientas de Globus Toolkit es bastante simple. Es solo cuestión de descargar los binarios o las fuentes. En este caso fue necesario descargar las fuentes debido a que como ya se ha mencionado es una plataforma un tanto diferente y no hay binarios precompilados.

Aún así el proceso de compilación es bastante simple. Es solo cuestión de ejecutar los comandos **`./configure --with-flavor=gcc64dbg`** para especificar la plataforma y posteriormente ejecutar el comando **`make`** y **`make install`**.

A pesar de esto las herramientas de Globus Toolkit dependen fuertemente de Java pero Sun Microsystems no distribuye binarios de Java para Yellow Dog Linux u otros sistemas operativos para arquitectura PowerPC (Excepto Mac OS X) y el código fuente no está disponible por lo que fue necesario acudir a la distribución de IBM lo cual requirió crear una cuenta de usuario en IBM para poder acceder a dicha información.

Lamentablemente no hubo suficiente tiempo para continuar con las pruebas de despliegue del GRID, pero se corroboró que es posible instalar las aplicaciones requeridas para ello sobre una arquitectura PowerPC y sobre una plataforma como el PlayStation 3.