Evaluación de costos de comunicación en arquitecturas para computación heterogénea aplicadas a computación científica

Nelson Rodríguez, Marìa Murazzo, Diego Medel, Maximiliano Fernández, Facundo González
Departamento e Instituto de Informática - F.C.E.F. y N. - U.N.S.J.
Complejo Islas Malvinas. Cereceto y Meglioli. 5400. Rivadavia. San Juan.
Tel:02644234129
nelson@iinfo.unsj.edu.ar marite@unsj-cuim.edu.ar mdiego88@gmail.com mascy086@gmail.com facu_jgg@hotmail.com

Resumen

Los cluster de commodity permiten alcanzar procesamiento de alta perforance a costos muy convenientes, son escalables y extensibles, permitiendo incrementar su capacidad gradualmente agregando equipos. Las arquitecturas paralelas homogéneas con elementos de proceso de características similares están siendo utilizadas con éxito en distintos ámbitos de la ciencia y de la industria, sin embargo no presentan toda la potencia computacional que podría obtenerse si se ampliara a un sistema de naturaleza heterogénea en el que los nuevos componentes presentan características diferenciales con los anteriores.
Este modelo de computación presenta nuevos desafíos fundamentalmente para integrarse con tecnologías diversas, las cuales necesitan comunicarse entre sí, pero este costo de comunicación (fundamentalmente en performance) puede resultar muy caro si no es convenientemente realizado. Por lo tanto al tener diversas estrategias y métodos que utilizan las distintas arquitecturas que componen un cluster, resulta sumamente importante evaluar y encontrar estrategias que minimicen estos costos.
El objetivo de este trabajo es la evaluación del costo de la comunicación entre unidades de computación en un Cluster de forma tal de poder analizar los resultados para generar estrategias más adecuadas para incrementar la performance.

Palabras clave: HPC, heterogeneous computing, Cluster, GPGPU

Contexto

El presente trabajo se encuadra dentro del área de I/D Procesamiento Distribuido y Paralelo, y se enmarca dentro del proyecto de investigación Cloud Computing con herramientas libres para evaluación de modelos de despliegue híbrido, presentado en diciembre de 2013 y que tiene como unidades ejecutoras al Departamento e Instituto de Informática de la FCEFyN de la UNSJ. Esta propuesta se ha presentado en una nueva convocatoria, pero los resultados de evaluación la misma se espera que estén para abril del presente año

Introducción

La computación heterogénea se refiere a sistemas que utilizan una variedad de diferentes tipos de unidades de computación. Una unidad de cálculo puede ser un procesador de propósito general (GPP) y un procesador de propósito especial (como DSP, GPU o

FPGA). En general, una plataforma de computación heterogénea consiste en procesadores con diferentes arquitecturas de conjuntos de instrucciones [1].

Una de las primeras definiciones, aparecieron cuando solo existían supercomputadoras y Main Frame, y la define como: la Computación heterogénea es el uso coordinado efectivo y bien orquestado de un conjunto de máquinas de ejecución alta diversas (incluyendo las máquinas paralelas) para proporcionar super velocidad de procesamiento para las tareas computacionales que demandan diversas necesidades informáticas [2].

Las arquitecturas heterogéneas a nivel de nodo han resultado atractivas durante la última década por varias razones comparadas con la CPU tradicional, estas ofrecen alto niveles de performance y son eficientes en consumo de energía y costos [3]. Así, es frecuente encontrar arquitecturas paralelas donde las características de los elementos que componen el sistema (procesadores, memoria, red, etc.) pueden ser diferentes.

Los primeros intentos de lograr paralelismo con equipos de commodity fueron los multi-chip, donde el paralelismo se logra teniendo varios procesadores físicos. Luego, al mejorar la escala de integración de los microprocesadores, se alcanzó el paralelismo a nivel de Multi-core.

Por otro lado para la programación en Clusters, se consolidaron de estándares como PVM y MPI,

Aparecen nuevas situaciones que no pueden ser resueltas mediante la aplicación directa de los modelos y las técnicas conocidas para el caso homogéneo. Se hace necesario adaptar los métodos conocidos y en muchos casos diseñar nuevas estrategias comenzando desde cero. Aparece un conjunto importante de problemas abiertos que están siendo intensamente estudiados y analizados [4]. Dentro del ámbito de la investigación científica, siempre ha existido la necesidad de tener computadoras con grandes capacidades de procesamiento (del orden de los Teraflops) y grandes capacidades de almacenamiento (del orden de los Petabytes). El aporte que ofrecen estos tipos de computadoras es de crucial importancia debido a que permiten realizar números de cálculos imponentes que se desarrollan a nivel de investigación como son las simulaciones en producción industrial, computología examinando bugs en grandes programas, explorando propiedades ferromagnéticas en la física, analizar dinámica molecular a nivel químico, etc. El área informática que lleva a cabo este tipo de procesamiento se llama HPC (High Performance Computing).

Cabe resaltar que existen trabajos previos en el área pero ninguno integra las arquitecturas multi-core, placa de video multinùcleo en un Cluster.[5]

La dificultad en esta área de conocimiento es la formación de recursos humanos que manejen una variedad de arquitecturas y herramientas de programación. Cada paradigma está caracterizado por un conjunto de atributos de recursos hasta la infraestructura y de las aplicaciones ejecutando en esa infraestructura.

Estos sistemas heterogéneos escalables pueden manejar diferentes demandas que los nodos de los sistemas gráficos stand-alone. En particular el diseño de cada nodo soporta comunicación inter e intra nodo por medio de MPI o IO a disco. Además del benchmark de Linpack en TOP500, existen pocos suites de benchmark aceptados que permiten que la comunidad pueda caracterizar y comparar las arquitecturas y entornos de programación para estos sistemas escalables heterogéneo [6].

La computación Heterogénea es una solución sumamente potente, debido a que la computadora que lidera el TOP 500, la Tianhe-2, desarrollada por La Universidad Nacional de tecnología de defensa de China, utiliza esta tecnología [7]. Combinando la escalabilidad de los clúster con MPI y la gran capacidad de procesamiento de las placas GPGPU [8] con CUDA y el procesamiento multinùcleo se pueden alcanzar niveles colosales de procesamiento.
Existen diversos problemas a resolver como aspectos de disponibilidad, eficiencia, extensibilidad, reusabilidad y personalización. Además por la naturaleza distribuida de los datos, resulta dificultoso mantener la consistencia y coherencia de datos, contar con una representación homogénea de los mismos (en una plataforma heterogénea), eficiencia en la sincronización entre las plataformas (distribuida y paralela). Por otro lado si se quiere obtener código fiable se debería realizar la verificación y validación de programas de este tipo, algo que parece por ahora imposible de lograr.

GPGPU

Las GPUs son un componente común en los clúster de alto rendimiento, por todas sus ventajas conocidas, como su velocidad de procesamiento en paralelo, bajo consumo etc. En la HPC, el número de GPU en cada nodo, cada vez aumenta más, por lo tanto debemos tener una comunicación eficiente, ya sea por las GPU que se encuentran en el mismo nodo, como así también las que se encuentran en distintos nodos, resultando MPI la opción más utilizada.
La combinación de MPI con CUDA son variadas, por ejemplo en aplicaciones donde el tamaño de los datos es muy grande para almacenar en la memoria de una sola GPU, no demandará un tiempo mucho mayor de ejecución. Con MPI se puede acelerar, dividiendo y enviando los datos a otras GPUs, e ir escalando. Con CUDA y MPI se puede alcanzar estos objetivos en forma eficiente y sencilla.
Como el espacio de memoria de la CPU y la GPU son distintos, debe realizarse una secuencia de pasos para trabajar con ambas.
CUDA 4.0 introdujo un nuevo concepto que es Unified Virtual Addressing (UVA), significa que la memoria del host y de todas las GPU (que se encuentran en un nodo) es combinada dentro de un espacio virtual de direcciones.
Las tecnologías de aceleración como GPUDirect puede ser utilizado por la librería MPI transparentemente al usuario. Las tecnologías GPUDirect proporcionan gran ancho de banda, comunicaciones de baja latencia con las GPUs de Nvidia. Proporciona comunicación directa entre las GPU y otros dispositivos PCI-E, y acceso directo a la memoria entre las tarjetas de red y la GPU. También reduce considerablemente la latencia de MPISendRecv entre los nodos de GPU de un cluster y mejora el rendimiento general de las aplicaciones. Por lo tanto en la propuesta de investigación todas estas estrategias deben ser comparadas con los métodos tradicionales.

MPI

El pase de mensajes es un modelo de comunicación ampliamente usado en computación paralela. En años recientes se han logrado desarrollar aplicaciones importantes basadas en este paradigma.
El crecimiento en el volumen y diversidad de tales aplicaciones originaron la necesidad de crear un estándar, es así como surge MPI o Message Passing Interface
MPI es un estándar que funciona en una amplia variedad de computadoras paralelas y de forma tal que los códigos

sean portables. Su diseño está inspirado en máquinas con una arquitectura de memoria distribuida, sin embargo, también se encuentran implementaciones en máquinas con memoria compartida.
La ventaja de MPI sobre otras bibliotecas de paso de mensajes, es que los programas que utilizan la biblioteca son portables y rápidos, (porque cada implementación de la librería ha sido optimizada para cada hardware).
Sin embargo, el acceso remoto a memoria es lento. El ancho de banda de red y el rendimiento es uno de los factores más cruciales en el rendimiento de implementación de MPI.
El estándar es extenso en cuanto al número de rutinas que se especifican y contiene alrededor de 129 funciones muchas de la cuales tienen numerosas variantes y parámetros, aunque muchos programas paralelos pueden ser escritos usando sólo 6 funciones básicas, aunque el resultado puede no ser eficiente.
MPI es totalmente compatible con CUDA, que está diseñado para la computación en paralelo en una sola computadora o nodo. Hay muchas razones para querer combinar MPI con CUDA. Una razón es para resolver problemas con un tamaño de datos demasiado grande para caber en la memoria de una sola GPU, o eso requeriría un tiempo de cálculo excesivamente largo en un solo nodo. Otra razón es acelerar una aplicación MPI existente con GPUs o para permitir una aplicación existente multi-GPU de un solo nodo de escalar a múltiples nodos.

Otras opciones a evaluar
Los mecanismos para desarrollar programas para las distintas arquitecturas expuestos son los más populares, sin embargo en una segunda fase se espera evaluar a PVM, OpenGL (para tarjeta gráfica), pthreads y SILC (dado que se trabajará con OpenMP para multinùcleo) entre otras opciones.

Líneas de Investigación, Desarrollo e Innovación

Debido a que el espacio de direcciones de memorias entre las distintas computaciones que realiza un Cluster son diferentes (host, multinùcleo, GPU), es necesario evaluar parámetros de la comunicación entre los distintos componentes y analizar cómo afecta a la performance final.
Hay que tener en cuenta que las aplicaciones necesitan esta comunicación porque no son cien por ciento paralelas, o sea tienen una porción secuencial y otra paralela, y a su vez algunas computaciones son más convenientes ejecutarlas en una parte de la arquitectura, generando comunicación entre las distintas computaciones.

Resultados y Objetivos

Se han publicados varios trabajos en el área de Cloud Computing, sin embargo para computación heterogénea se puede citar [10,11]. También se llevaron a cabo trabajos de divulgación.
En la actualidad se encuentras 3 tesinas de licenciatura en desarrollo dos de las cuales están próximas a finalizar.

Resultados Esperados (Objetivos)
El objetivo del grupo de investigación es la evaluación de parámetros de comunicación entre distintas arquitecturas instaladas en un Cluster.
Se espera que una vez realizados los estudios pertinentes puedan hacerse propuestas de mejoras para mejorar la mayor eficiencia, el balance de carga y la interoperabilidad.

Formación de Recursos Humanos

El equipo de trabajo está compuesto por los tres (3) docentes-investigadores (dentro de un grupo de investigación en Cloud Computing que está compuesto de 5 investigadores más), que figuran en este trabajo y 2 alumnos, a los cuales se espera que se incorporen otros alumnos en condición de hacer sus tesinas.
Durante 2013 se dirigió una beca de investigación sobre CUDA y otra sobre Hadoop aplicado a computación heterogénea. Ademàs cuatro alumnos que trabajan en el grupo de investigación han sido beneficiados con becas en la escuala de verano de la UNRC y en el HPCLatam. Por otro lado se está dirigiendo una tesina de grado sobre MPI y otra sobre Hybrid Cloud Computing. Se espera también aumentar el número de publicaciones. Por otro lado también se prevé la divulgación de varios temas investigados por medio de cursos de postgrado y actualización o publicaciones de divulgación

Referencias

[1] Nelson Rodríguez, María Murazzo, Daniela Villafañe, Maximiliano Alves, Diego Medel. “Integración de Computación Heterogénea con Hadoop para Cloud Computing”. XV WICC. Abril 2013. Paraná. Entre Ríos.
[2] Brodtkorba,, Dykena, Hagena, Hjelmervika, Storaas. “State-of-the-art in heterogeneous computing”. Scientific Programming 18 (2010) 1–33.
[3] Ashfaq A. Khokhar, Viktor K. Prasanna, Muhammad E. Shaaban,
and Cho-Li Wang. Heterogeneous Computing: Challenges and Opportunities. IEEE COMPUTER. 1993. http://meseec.ce.rit.edu/eecc722-fall2002/papers/hc/1/r6018.pdf
[4] Moreno de Antonio. “Computación paralela y entornos heterogéneos”. Soportes Audiovisuales e informáticos. Serie Tesis Doctorales. Servicio de Publicaciones. Universidad de la laguna. Curso 2004/05. Ciencias y tecnologías/23. ISBN.: 84-7756-662-3.
[5] Fabiana Leibovich, Armando De Giusti, Marcelo Naiouf, Laura De Giusti, Franco Chichizola. “Programación híbrida en arquitecturas cluster de multicore. Escalabilidad y comparación con memoria compartida y pasaje de mensajes”. CACIC 2010.
[6] Anthony Danaliszy Gabriel Mariny Collin McCurdyy Jeremy S. Meredithy Philip C. Rothy Kyle Spaffordy Vinod Tipparajuy Jeffrey S. Vetter. “The Scalable HeterOgeneous Computing (SHOC) Benchmark Suite”. GPGPU ’10 March 14, 2010.
[7] www.top500.org
[8] Olexandr Isayev. “Computación Heterogénea: Nuevo Paradigma Para La Era Exaescala”. Paralelizados.com. Comunidad de usuarios de HPC. GPUScience. IDC-Exascale-Executive-Brief_Nov2011. Noviembre 23, 2011 ·
[9] Programación paralela facilitada. NVIDIA Corporation .http://la.nvidia.com/object/cuda_home_new_la.html. 2013.
[10] Rodríguez, Murazzo, Villafañe, Alves, Medel. Integración de Computación Heterogénea con Hadoop para Cloud Computing. XV WICC. Abril 2013. Paraná. Entre Ríos
[11] Murazzo, Rodríguez, Villafañe, González. Perspectivas en el análisis de grandes volúmenes de datos en el Cloud. I JCC. UNLP. Jun. 2013. La Plata- Bs As.