ANÁLISIS DE RENDIMIENTO DE LAS TRANSFERENCIAS DE MEMORIA Y SUBRUTINAS GEMM EN CLÚSTER DE GPU NVIDIA TESLA

Los clúster comerciales aumentados con aceleradores de aplicaciones están evolucionando hacia sistemas de cómputo de alto rendimiento.
Las unidades de procesamiento gráfico (GPU) con un alto costo son una buena plataforma para la aceleración de la aplicación científica.

INTRODUCCIÓN

- Entre las GPUs CUDA de NVIDIA disponibles, la serie Tesla esta diseñada específicamente para el campo de la computación científica. En este trabajo, se estudió el rendimiento de las copias de memoria y subrutinas GEMM, Para ello, un punto de referencia FRAMEWORK NETPIPE [1], que se ha desarrollado para evaluar la latencia y ancho de banda en las copias de memoria entre el host y el dispositivo GPU.
- Un problema importante que aparece en el modelo de coprocesador es la sobrecarga involucrada en la transferencia de datos al espacio de memoria del dispositivo contra el tiempo de calculo actual.
- Uno de los principales objetivos del entorno de programación CUDA es el desarrollo de programas paralelos escalables y eficientes [4].En este modelo, la GPU es vista como un dispositivo de computo multiprocesos que es capaz de ejecutar hilos en paralelo.

PALABRAS CLAVES

- **GPU NVIDIA TESLA** [5]
- **CPU** [6]
- **CLÚSTER** [7]
- **GEMM** [1]
- **NETPIPE** [8]
- **SUBRUTINA** [9]
- **HOST** [10]
- **CUDA** [11]
- **HPC** [12]

¿QUE ES?

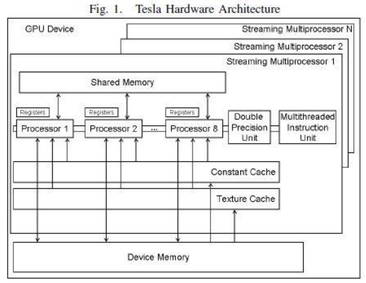
NVIDIA ha posicionado sus GPUs al lado de las actuales tecnologías de computación de alto rendimiento (HPC), entre las cuales están las GPUs CUDA de NVIDIA en la serie Tesla.

Las GPU NVIDIA Tesla, son tarjetas de “video” como es común escuchar en el mercado, pero esta serie está específicamente diseñada para el campo de la computación científica. Se diferencia de sus contrapartes graficas como Quadro y GeForce en términos de frecuencia en reloj del procesador y configuración de memoria.

Las ultimas características en la serie tesla, es su doble precisión de punto flotante en el hardware, lo que las hace ideales para aplicaciones científicas que dependen de alta precisión numérica.

Se muestra en la figura la arquitectura de hardware Tesla. Cada procesador de computo Tesla T10 tiene 4 GB de memoria dedicada, cada uno de estos procesadores se puede programar usando el framework CUDA.

Fig. 1. Tesla Hardware Architecture



Buscan optimizar estos dispositivos por medio de frameworks como CUDA los cuales tienen un gran interés en desarrollar algoritmos no gráficos para hardware GPU y permiten programar cada uno de los procesadores de la arquitectura Tesla.

Para medir el rendimiento que tienen el host y el dispositivo GPU, se basan en un framework llamado NetPIPE para evaluar la latencia y ancho de banda de las copias de la memoria entre el host y el dispositivo de la GPU.

Se compara el funcionamiento de las subrutinas GEMM de precisión single y double de la biblioteca de NVIDIA CUBLAS 2.0 y se comparan con los resultados de las rutinas BLAS desde el procesador Intel Math Kernel Library (MKL), para entender ventajas y desventajas. Se realiza la prueba con un Cluster Intel Xeon equipado con una NVIDIA Tesla GPU.

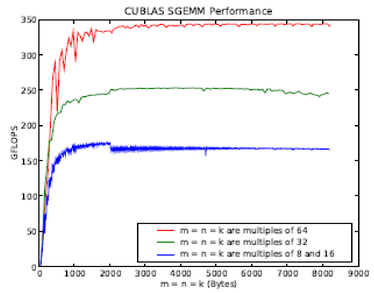
RESULTADOS

NETPIPE CUDAMEMCPY

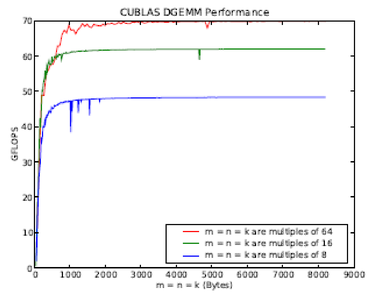
- En el módulo NetPIPE cudaMemcpy la salida archivo contiene el tamaño del búfer, el rendimiento y el tiempo de transferencia.
- Los resultados están en GB / s para comparar con las velocidades de enlace físicas que son por lo general reportado en GB / s. La gráfica de rendimiento NetPIPE el tamaño del paquete en Bytes se muestra en una escala logarítmica en el eje X y la gráfica en segundos en el eje y. Ambos ejes están representados en la escala logarítmica.
- Los resultados proporcionados por el SDK de CUDA, las mediciones de tiempo por el índice de referencia se realizan mediante el temporizador CUDA funciones. La función de temporización que se utiliza para NetPIPE es basado en la función gettimeofday (). En todos los resultados, la prueba se ejecuta NetPIPE hasta tamaños de búfer de 192 MB.

CONCLUSIONES

- Se estudió el rendimiento de las subrutinas SGEMM / DGEMM de la biblioteca CUBLAS 2.0 en la última versión del sistema de cómputo NVIDIA Tesla. Durante el desarrollo del módulo NetPIPE cudaMemcpy los resultados fueron validados con referencia a los resultados del benchmark dados por el CUDA SDK.
- Basado en los resultados del estudio se llega como conclusión que el módulo cudaMemcpy de NetPIPE se puede utilizar para medir los datos de rendimiento de movimiento entre el host y el dispositivo GPU.
- Como trabajo futuro, se tiene como idea principal integrar este módulo con el módulo NetPIPE Infiniband para incorporar copias de memoria a dispositivos remotos utilizando el GPU directo a memoria remota, se pretende también ampliar el framework NetPIPE para incorporar el rendimiento de las copias de memoria asíncronos.



(a)



(b)

Referencias

- [1] Q. O. Snell, A. R. Mikler, and J. L. Gustafson, “Netpipe: A network protocol independent performace evaluator,” in In Proceedings of the IASTED International Conference on Intelligent Information Management and Systems, 1996.
- [4] J. Nickolls, I. Buck, M. Garland, and K. Skadron, “Scalable parallel programming with cuda,” Queue, vol. 6, no. 2, pp. 40–53, 2008.
- [5] http://www.nvidia.es/object/tesla-server-gpus-es.html
- [6] https://es.wikipedia.org/wiki/Unidad_central_de_procesamiento
- [7] https://es.wikipedia.org/wiki/Cl%C3%BAster_(inform%C3%A1tica)
- [8] Framework para HPC – Performance Comparasion
- [9] https://es.wikipedia.org/wiki/Subrutina
- [10] https://es.wikipedia.org/wiki/Host
- [11] http://www.nvidia.es/object/cuda-parallel-computing-es.html
- [12] https://es.wikipedia.org/wiki/Computaci%C3%B3n_de_alto_rendimiento