Análisis de consumo energético en Cluster de GPU y MultiGPU en un problema de Alta Demanda Computacional

Erica Montes de Oca[1], Laura De Giusti[1,3], Armando De Giusti[1,2] ,Marcelo Naiouf[1]

[1] Instituto de Investigación en Informática LIDI (III LIDI)
Facultad de Informática, Universidad Nacional de La Plata
Comisión de Investigaciones Científicas de la Pcia. de Buenos Aires (CIC
La Plata, Buenos Aires, Argentina
[2] CONICET
[1] Investigador Asociado CIC
{emontesdeoca,ldgiusti,degiusti,mnaiouf}@lidi.info.unlp.edu.ar

Resumen. En este trabajo se realiza un análisis de consumo energético en dos Cluster de GPU y una MultiGPU utilizando como caso de estudio el problema de los N Cuerpos. Se describen las soluciones implementadas con MPI+CUDA para las arquitecturas usadas. Se muestran los resultados y un análisis de performance y consumo energético.

Palabras claves: GPU, Cluster de GPU, MultiGPU, N Cuerpos, Green Computing, Consumo Energético.

1 Introducción

Los profundos cambios generados por el avance tecnológico han modificado las formas de comunicación. En muchos casos, el resultado de este desarrollo ha traído aparejado un impacto ambiental. En las últimas décadas, Green Computing, ha emergido como un nuevo concepto para hacer frente a las preocupaciones sobre el cuidado del medio ambiente cuando las Tecnologías de la Información y la Comunicación están presentes [1].
En la actualidad, la generación de grandes cantidades de datos requiere reducir los tiempos de ejecución de las aplicaciones que los procesan. Sin embargo, acelerar el procesamiento ocasiona un mayor consumo de energía consumida, por lo que es necesario también minimizar esta variable [2]. Una opción de aceleramiento de cómputo son las GPU (Unidad de Procesamiento Gráfico) [3]. En los últimos años, se ha comenzado a tomar consciencia en el ámbito del software en desarrollar algoritmos más eficientes, no sólo desde el punto de vista de la performance, sino también del consumo energético [4][5]. Green Computing ya es un concepto instalado y adoptado. y las Ciencias de la Computación son las responsables de la investigación y promoción de sus prácticas [6][7].
En este trabajo, la Sección 2 introduce los conceptos básicos de Green Computing; la Sección 3, describe la simulación del problema de Atracción Gravitacional de los N

Cuerpos. La Sección 4 muestra los resultados obtenidos en el trabajo experimental. Por último, la Sección 5, presenta las conclusiones y trabajos futuros.

2 Green Computing

Desde la Revolución Industrial, el avance tecnológico alcanzado por el hombre inició la degradación del medio ambiente. El desequilibrio en los Gases de Efecto Invernadero (GEI) ocasiona los cambios climáticos actuales [8]. Los sectores generadores de las mayores emisiones de Dióxido de Carbono (el más conocido de los GEI) son: la generación de energía eléctrica, la industria, el transporte, el comercio y las residencias [9].
Green Computing comprende el uso eficiente de los recursos, la reducción del uso de materiales peligrosos, maximizar la eficiencia en la manufactura de productos, promover el reciclaje o biodegrabilidad de los productos y desechos fabriles [10].
El precio de la energía ha ido en aumento, por lo que la reducción del consumo energético es un punto clave. En este sentido, un sistema de cómputo no sólo debe ser escalable y reducir los tiempos de procesamiento, sino que debe ser eficiente en cuanto a consumo de energía, reduciendo los costos del gasto de consumo eléctrico y la disminución del impacto ambiental.

3 Arquitectura GPU

Las Unidades de Procesamiento Gráfico (GPU), comenzaron a ser utilizadas en las últimas décadas para cómputo de propósito general (GPGPU). Es una arquitectura paralela desde su nacimiento, y cada nueva generación doblan el número de cores. e incrementan la cantidad de operaciones de punto flotante por segundo. [11] [12].
La programación en GPU se facilitó a través del surgimiento de CUDA (Compute Unified Device Architecture) en 2007, por parte NVIDIA. CUDA es una plataforma hardware-software que le permite al desarrollador crear código en lenguaje C.
La arquitectura de una GPU-Nvidia está organizada en Streams Multiprocessors (SMs), los cuales tienen un determinado número de Streams Processors (SPs). Los SPs comparten la lógica de control y la caché de instrucciones. Nvidia [12] varía la cantidad de SMs en cada nueva generación logrando que las mismas sean escalables en performance, además de variar el número de DRAM para lograr escalar el ancho de banda de memoria y la capacidad. Cada SM provee los suficientes threads, cores y memoria shared para ejecutar uno o más bloques de threads CUDA. Los que realizan verdaderamente la ejecución son los SPs, que ejecutan múltiples threads concurrentemente. Los threads se organizan en bloques de hilos, y existen uno o más bloques ejecutándose concurrentemente en SM. La organización de los threads puede verse por nivel o jerárquicamente: Grid, Bloques y Threads (Fig. 1).

3.1 Cluster de GPU

Un Cluster GPU es una arquitectura heterogénea en la cual, por un lado, se tiene un subsistema conformado por varios cores CPUs y su sistema de memoria y E/S, mientras, por otro lado, se encuentra el subsistema GPU con sus memorias on y off chip. Estos dos sistemas se comunican a través de PCI-E con una velocidad de ancho

de banda baja comparada con las velocidades de comunicación de los sistemas de memoria. Por lo que, el cuello de botella es la comunicación que existe entre dichos subsistemas [14].

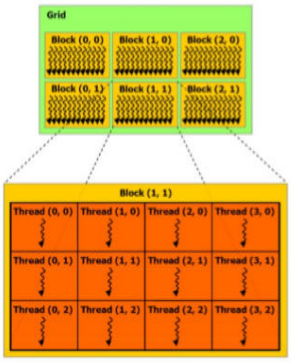

Fig. 1 Jerarquía de Threads de una GPU [13]

3.2 MultiGPU

La arquitectura MultiGPU se refiere al uso de una o más tarjetas gráficas en una misma PC. En este caso las GPUs son fácilmente programables a través de CUDA utilizando SLI (Scalable Link Interface) provista por Nvidia. En el caso de los Cluster de GPU, es necesario utilizar MPI+CUDA [15].

4 N Cuerpos: Problema de Alta Demanda Computacional

El problema de los N Cuerpos es clásico en el ámbito científico, y ha sido estudiado por su adaptabilidad a distintas aplicaciones del mundo real. Carente de una solución analítica, lo convierte en uno de los primeros problemas que se buscó resolver con una computadora [16]. Consiste en la simulación del comportamiento de N Cuerpos en un espacio de trabajo. En particular, el presente estudio se ha basado en el cálculo de la Atracción Gravitacional [17].
Newton planteó la *Ley de Gravitación Universal*, sentando las bases de la *Teoría de la Atracción Gravitacional*: cada cuerpo cuenta con una masa, una posición inicial y una velocidad. La gravedad hace que los cuerpos se aceleren y se muevan provocando la atracción de unos con otros. La magnitud de la fuerza de gravedad entre dos cuerpos se encuentra expresada en la fórmula (1):

$$F = \frac{G * m_i * m_j}{r^2} \quad (1)$$

Siendo m la masa, r la distancia, y G constante gravitacional (6,67 x 10-11)

3.1 Solución del problema en una GPU

Para realizar la simulación se utilizan vectores de tamaño N, que representan las fuerzas, posiciones, velocidades y masas. El algoritmo realiza las siguientes acciones:

1. Calcular la fuerza de atracción del cuerpo i, la cual se determina con los N-i cuerpos. A su vez, los N-i cuerpos se verán influenciados por la fuerza de gravedad de i.
2. Se modifica la posición del cuerpo i, dependiendo de su fuerza de gravedad.
3. Se repite 1 y 2, tantas veces como pasos de simulación se quieran realizar.

La independencia del cómputo del cálculo de la fuerza de atracción de cada cuerpo permite que se pueda ejecutar en una GPU. Básicamente, el algoritmo para una GPU realiza los siguientes pasos:

1. Desde la CPU se reserva memoria en la GPU;
2. Se transfieren los datos desde la CPU a la GPU;
3. Se ejecuta el *kernel*, que realiza el procesamiento del cálculo de las fuerzas de atracción gravitacional;
4. Se transfieren los datos procesados desde la GPU a la CPU.

3.2 Solución del problema en un Cluster de GPU

La ejecución en un Cluster de GPU requiere la utilización de una combinación de MPI con CUDA. Por cada proceso MPI creado se tiene asociada una GPU. Utilizando un esquema master/slave, el proceso master es el encargado de inicializar la información de los cuerpos a distribuir entre los procesos esclavos. Los *N* cuerpos se distribuyen en *P* procesos, por lo que cada proceso debe computar un total de *N/P* elementos. A su vez, cada proceso envia por medio de PCI-E dichos datos a sus respectiva GPU, siendo éstas las que realizan cómputo, ya que los procesos MPI se limitan a comunicar entre ellos los datos. Una vez que la GPU ha calculado la fuerza de atracción de los cuerpos que le corresponden, envía dichos datos a la CPU. Luego, el proceso MPI, comunica dichos datos al resto de los procesos [6] [18].

3.3 Solución del problema en una MultiGPU

En la solución con MultiGPU, se reparte en partes iguales la cantidad de datos a procesar por cada GPU. CUDA provee la función `cudaSetDevice(i)` (donde $i \leq T$, con *T* la cantidad de GPUs en una PC) para setear la afinidad de las GPUs de una PC [17]. El cálculo de la fuerza de atracción gravitacional de cada uno de los cuerpos que componen el espacio de trabajo se realiza del mismo modo que en la solución explicada para una GPU [19].

4 Trabajo Experimental

El entorno de prueba está compuesto por las siguientes máquinas:

- Cluster de multicore con procesadores Quad Core Intel i5-2310 de 2,9 GHz, caché de 6 MB: cada nodo cuenta con una GPU GeForce TX 560TI. CUDA 8.0
- Cluster de multicore con procesadores Quad Core Intel i5-4460 de 3.2 GHz, caché de 6 MB: cada nodo cuenta con una GPU GeForce GTX 960. CUDA 8.0
- Una PC con un procesador Quad Core Intel i5-2310 de 2,9 GHz, con dos GPU Tesla C2075. CUDA 4.2

Todas las pruebas de la simulación se hicieron para uno y tres pasos. La cantidad de cuerpos (N) se varió para 128000, 256.000 y 512.000. En las pruebas se utilizaron bloques de threads de 256 hilos para todas las GPUs. Los datos obtenidos son el resultado de un promedio de diez ejecuciones del algoritmo y mediciones de consumo energético. Se plantó un lote de escenarios conformados por los tres tamaños de entrada, repitiendo la simulación diez veces.
Para realizar las mediciones se utilizaron una pinza amperimétrica (para la medición de corriente, con una sensibilidad 1A/100mv, 1A/10mv y 1A/1mv) y un transformador (que censa la tensión existente en la entrada del suministro de energía). La tensión se midió directamente de la línea eléctrica a la cual se encuentra conectado el Cluster de multicore y la MultiGPU. Ambos dispositivos se conectan a los canales de entrada del Osciloscopio (Rigol DS1074Z, que tiene una tasa de muestreo de un 1GSa/s [20]). Para la medición de consumo realizada se utilizaron 2 canales (uno para la corriente y otro para la tensión), almacenando 12Mpts en memoria.
Para analizar las muestras obtenidas para cada escenario planteado, se modificó una aplicación de análisis de consumo energético denominada energyAnalyser [21].

A partir de los datos de la corriente y la tensión extraídos del osciloscopio, se realiza el cálculo de la potencia instantánea (2).

$$P(t) = V(t) * I(t) \quad (2)$$

siendo V el voltaje e I la corriente
Una vez obtenida la potencia instantánea, se calcula la potencia promedio, y la cantidad de Joules consumidos por la ejecución del algoritmo de los N Cuerpos.

En la Tabla 1, se presentan los tiempos de ejecución para la simulación con tamaño de entrada N: 128000, 256000 y 512000, para uno y tres pasos de la simulación, para una GPU GeForce GTX 960, una Tesla C2075, una GeForce GTX 560, Cluster de GPU y la MultiGPU.

Tabla 1. Tiempos de ejecución de la simulación de los N Cuerpos medidos en segundos utilizando una GPU, Cluster con dos GeForce GTX 960, una MultiGPU con dos Tesla C2075 y un Cluster de dos GeForce GTX 560.

Cantidad de cuerpos	GeForce GTX 960		Tesla C2075		GeForce GTX 560	
	1 Paso	3 Pasos	1 Paso	3 Pasos	1 Paso	3 Pasos
128000	5,23	13,54	5,36	16,1	6,42	18,91
256000	16,16	49,33	19,66	63,9	24,87	75,68
512000	64,64	194,58	78,64	252,87	99,08	297,45
Cantidad de cuerpos	Cluster de dos GeForce GTX 960		MultiGPU con dos Tesla C2075		Cluster de dos GeForce GTX 560	
	1 Paso	3 Pasos	1 Paso	3 Pasos	1 Paso	3 Pasos
128000	3,69	7,52	2,69	8,08	3,46	9,93
256000	9,42	26,83	10,74	32,21	12,48	37,44
512000	35,18	130,59	42,59	85,18	49,75	149,44

Puede notarse que la GeForce GTX 960 es la que ejecuta en un menor tiempo el problema. Sin embargo, cualquiera de las tres GPU, consiguen acelerar el cómputo

del algoritmo desarrollado. En trabajos previos, puede observarse la aceleración significativa que se obtiene utilizando GPU comparada con las versiones en memoria compartida, memoria distribuida y memoria híbrida [3][6].
Los algoritmos paralelos utilizan como medida de performance el *speedup* para medir el comportamiento del aumento de la cantidad de procesadores comparado con utilizar sólo uno. Las arquitecturas empleadas en este trabajo son inherentemente paralelas [22], por lo tanto, esta métrica no es la más adecuada para medir su rendimiento. Por esto se define el concepto de *Aceleración* (3), como métrica para medir performance al utilizar más de una GPU [6].

$$Aceleración = \frac{Tiempo\_1\_GPU(N)}{Tiempo\_ + \_GPU(N)} \quad (3)$$

Para el caso de la MultiGPU con dos Tesla C2075, se puede observar en la Fig. 2 que se obtiene una aceleración superlineal. Esto se debe generalmente a la división de los datos: teniendo N datos a procesar con una GPU, cuando se trabaja con dos, la división de los datos es N/2, es decir se tiene una menor cantidad de datos a procesar en cada GPU, obteniéndose un resultado superlineal.
Actualmente, acelerar el cómputo, teniendo en cuenta sólo performance en la escalabilidad de los sistemas no es suficiente. Un sistema que brinde performance pero que consume demasiada energía, no es sostenible en el tiempo ni rentable económicamente. Por esto, eEl estudio del consumo energético de las aplicaciones se ha tornado un punto importante.
Se define la potencia promedio de energía como la cantidad de energía promedio entregada o absorbida por un elemento. Lo que implica que la cantidad de energía necesaria para procesar un dato es la misma independientemente del dato que sea y la cantidad de datos a procesar. La Tabla 2, presenta las potencias promedio de energía medidas en Watts por segundo para el uso de una GPU, los Cluster de GPU y MultiGPU al ejecutar el algoritmo de los N Cuerpos.

Tabla 2. Potencia promedio de energía medida en Watts por segundo para la simulación de los N Cuerpos utilizando una GPU, un Cluster con dos GeForce GTX 960, una MultiGPU con dos Tesla C2075 y un Cluster de dos GeForce GTX 560.

Cantidad de cuerpos	GeForce GTX 960		Tesla C2075		GeForce GTX 560	
	1 Paso	3 Pasos	1 Paso	3 Pasos	1Paso	3 Pasos
128000	160,48	163,75	276,74	279,26	191,87	196,49
256000	160,91	166,07	276,96	279,26	193,01	203,28
512000	162,36	164,49	278,51	278,17	202,77	208,97
Promedio	163,01		278,18		199,39	
Cantidad de cuerpos	Cluster de dos GeForce GTX 960		MultiGPU con dos Tesla C2075		Cluster de dos GeForce GTX 560	
	1 Paso	3 Pasos	1 Paso	3 Pasos	1 Paso	3 Pasos
128000	306,85	307,6	376,4	385,71	382,29	383,78
256000	315,62	320,42	375,94	381,4	390,88	392,33
512000	321,13	322,89	378,39	388,37	394,64	406,63
Promedio	315,76		381,08		391,75	

Como se puede observar en la Tabla 3, la potencia promedio de energía no varía significativamente. Las variaciones dadas pueden deberse a los movimientos de los datos en la jerarquía de memoria de la GPU o a la refrigeración, entre otras. Por lo cual, se puede calcular una media de las potencias promedio de energía medidas y obtener una potencia media de energía que caracterice a cada GPU y Cluster de GPU/MultiGPU.
En este punto, no es posible hacer una valoración precisa en cuanto al consumo energético de cuál arquitectura es conveniente para ejecutar la simulación propuesta. Esto se debe a la forma de medición disponible, que permite medir "desde afuera" de la máquina, por lo que no sólo se está obteniendo el consumo energético de la GPU, sino también del resto de los componentes de la PC. Para poder comparar las arquitecturas, se realizaron mediciones en modo de reposo, para obtener de esta manera la potencia estática de cada una. En la Tabla 3, se pueden observar las potencias estáticas en Watts por segundo para las arquitecturas utilizadas en la experimentación.

Tabla 3. Potencia estática medida en Watts por segundo para una PC o para un Cluster con dos GeForce GTX 960, una MultiGPU con dos Tesla C2075 y un Cluster de dos GeForce GTX 560.

GPU	Una PC	Cluster/MultiGPU
GeForce GTX 960	69,4	120,2
Tesla C2075	132	132
GeForce GTX 560	97,6	122,1

En la Tabla 4, se pueden observar las potencias promedio de energía sin la potencia estática en Watts por segundo para el uso de una GPU, el uso de los Cluster de GPU y MultiGPU.

Tabla 4. Potencia promedio de energía sin la potencia estática en Watts por segundo para la simulación de los N Cuerpos utilizando una GPU, un Cluster con dos GeForce GTX 960, una MultiGPU con dos Tesla C2075 y un Cluster de dos GeForce GTX 560.

Cantidad de cuerpos	GeForce GTX 960		Tesla C2075		GeForce GTX 560	
	1 Paso	3 Pasos	1 Paso	3 Pasos	1 Paso	3 Pasos
128000	91,01	94,28	144,74	147,45	94,22	98,84
256000	91,44	96,6	144,96	147,26	95,36	105,63
512000	92,89	95,02	146,51	146,17	105,12	111,32
Cantidad de cuerpos	Cluster de dos GeForce GTX 960		MultiGPU con dos Tesla C2075		Cluster de dos GeForce GTX 560	
	1 Paso	3 Pasos	1 Paso	3 Pasos	1 Paso	3 Pasos
128000	186,64	187,35	244,4	253,7	260,27	261,76
256000	178,97	180,44	243,94	249,4	268,86	270,31
512000	185,45	183,6	246,39	256,37	272,62	284,61

Finalmente, la Fig. 3 presenta la energía en Joules consumida por la simulación utilizando una GPU.

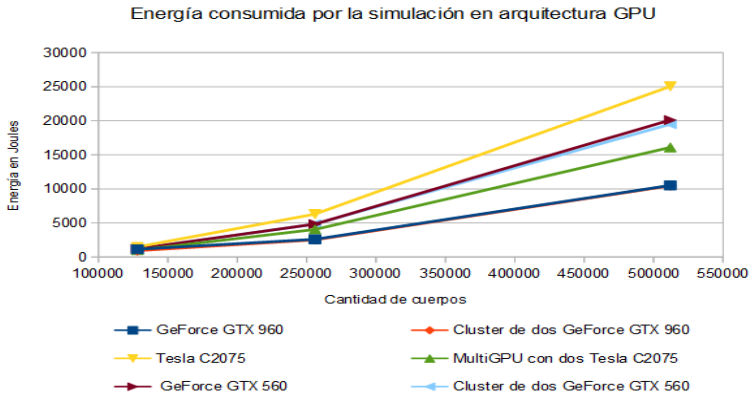

Fig. 3. Energía en Joules consumida por la simulación para cada arquitectura GPU

Puede observarse que para el caso de las GeForce la energía consumida es la misma, pero en menor tiempo de trabajo al agregar un nodo. En cambio en la MultiGPU, trabajar con dos tarjetas gráficas reduce el tiempo de cómputo, pero desde el punto de vista energético, el consumo es mayor.

5 Conclusiones y Trabajos Futuros

En la actualidad, el procesamiento de grandes cantidades de datos, el alcance de la máxima cantidad de transistores por microchip, y la demanda de la reducción de los tiempos de repuesta de las aplicaciones, han hecho que el procesamiento paralelo sea, en muchos casos, la única solución para tratar algunos problemas computables [18].
Las GPU permiten una reducción significativa de los tiempos de procesamiento, además de un menor costo y la facilidad de adaptar algunos problemas a estas arquitecturas, lo que ha hecho que sean una opción altamente viable a la hora de elegir una arquitectura paralela. Sin embargo, la aceleración del cómputo trae aparejada una problemática: la cantidad de energía consumida. El uso de la computadora, sus accesorios y sus recursos está catalogado en la actualidad como uno de los responsables del calentamiento global. El diseño de todos los sistemas de cómputo, desde dispositivos móviles hasta los sistemas de cómputo de altas prestaciones, está siendo impactado por el consumo energético.
En el presente trabajo se ha mostrado el consumo energético de tres diferentes tarjetas gráficas al ejecutar una solución del problema de Alta Demanda Computacional (N Cuerpos). Por un lado, utilizando una GPU de manera individual, y por otro, con dos Cluster de GPU y una MultiGPU.
A partir de las mediciones de performance y consumo energético presentadas en el apartado anterior, se concluye que:

- Referido al tiempo de ejecución, la GPU GeForce GTX 960 es la que ejecuta en un menor tiempo el problema. Sin embargo, cualquiera de las tres GPU, consiguen acelerar el cómputo del algoritmo desarrollado.
- La aceleración obtenida en cualquiera de las tres GPU utilizadas se encuentra cerca de la aceleración óptima.

- Además, se pudo comprobar lo demostrado en [23], que la cantidad de Watts promedios consumidos por una GPU, un Cluster de GPU o una MultiGPU, es independiente del tamaño de la entrada, y la variación que se observa en las mediciones es insignificante. Dicha variación puede deberse a los movimientos de datos y/o a la refrigeración de las placas.

  A partir de los resultados obtenidos, para el problema de los N Cuerpos, y con los escenarios experimentales llevados a cabo, se puede concluir que el uso de una GPU GeForce GTX 960 o de un Cluster de GPU GeForce GTX 960 consigue tanto en performance como en consumo energético las mejores prestaciones. Si se tienen en cuenta los costos de las distintas GPU utilizadas, la mejor opción sigue siendo la GeForce GTX 960. La placa que menos consume en líneas generales es la GeForce 960 para todos los tamaños del problema probados.

Como trabajos futuros se puede mencionar:

- Proponer un modelo de estimación de energía para el problema planteado, que permita predecir la cantidad de energía consumida al agregar GPU o al incrementar el tamaño de la entrada.
- Utilizando el mismo algoritmo de N cuerpos, medir consumo energético en más de dos nodos para los Cluster de GPU con GeForce GTX 560 y GeForce GTX 960.
- Estudiar y utilizar NVIDIA Management Library (NVML), que permite monitorear y administrar varios estados de los dispositivos NVIDIA GPU. Proporciona un acceso directo a las consultas y comandos expuestos a través de nvidia-smi. Entre las consultas que se pueden realizar, se encuentra la administración de la energía en la placa. La misma está disponible para la Tesla C2075 (no así para las otras placas: GeForce GTX 560 y 960). La consulta devuelve como resultado la potencia instantánea consumida por la tarjeta gráfica.
- Realizar análisis de consumo energético referidas a otro tipo de simulación, tales como simulaciones orientadas a individuos.

Referencias

1. Porcelli A.M., Martinez A.N.: La nueva economía del siglo XXI: análisis del impacto de la informática en el ambiente. Tendencias actuales en tecnologías informáticas verdes, un compromiso con la sustentabilidad. Questio Iuris vol. 8 nro 4 pp. 2174-2208. (2015)
2. Francis K., Richardson P.: Green Maturity Model for Virtualization. The Architecture Journal. pp. 9-15 (2008)
3. Montes de Oca E., De Giusti L., De Giusti A., Naiuof M.: Comparación del uso de GPU y cluster de multicore en problemas con alta demanda computacional. XI Workshop de Procesamiento Distribuido y Paralelo. CACIC 2012. Bahía Blanca, Buenos Aires, Argentina. (2012)
4. Baladini J., Morán M., Rozas C., De Giusti A., Suppi R., Rexachs D., Luque E.: Consumo energético de sistemas de cómputo de alta prestaciones. (2016)

5. AlMusbahi, Nahhas, AlMuhammadi, Anderkairi, Hemalatha: Survey on Green Computing: Vision and Challenges. Interenational Journal of Computer Applicatios. Vol 167, nro 10. (2017)
6. Montes de Oca E., De Giusti L., Chichizola F., De Giusti A., Naiouf M., "Utilización de Cluster de GPU en HPC. Un caso de estudio". IVX Workshop de Procesamiento Distribuido y Paralelo. CACIC2014. ISBN 978-987-3806-05-6. La Matanza, Buenos Aires, Argentina (2014)
7. Díaz J., Ambrosi V., Castro N., Candia D., Vega E., Rodriguez A.: Experiencia de la enseñanza de Green IT en la currícula de carreras de Informática de la UNLP. XI Congreso de Educación en Tecnología y Tecnología en Educación (2016)
8. Valdés Castro E.: Tecnologías de información que contribuyen con las prácticas de Green IT. Ingenium, pp. 11-26 (2014)
9. González C., Pérez R., Vásquez Stanescu C., Araujo G.: Eficiencia energíetica. Uso racional de la energía eléctrica en el sector administrativo. Ministerio del Poder Popular para la Energía Eléctrica. Municipio Libertador, Distrito Capital República Bolivariana de Venezuela (2014)
10. Talebi M., WayT.: Methods, Metrics and Motivation for a Green Computer Science Program. SIGCSE'09, Chattanooga, Tennessee, USA. (2009)
11. Silvestein M., Kim S., Huh S., Zhang X., Hu Y., Wated A.,Witchel E., "GPUnet: Networking Abstractions for GPU Programs", Transactions on Computer Systems, Vol. 34, No. 3, Article 9, Publication date: September 2016, ACM 0734-2071/2016/09, 2016
12. www.nvidia.com/object/what-is-gpu-computing.html fecha de acceso febrero 2018
13. www.nvidia.es/object/cuda-parallel-computing-es.html fecha de acceso: febrero 2018.
14. Nvidia Development: https://devtalk.nvidia.com/default/topic/808106/question-abount-cudasetdevice-and-multiple-host-threads (2018)
15. la.nvidia.com/object/tesla-feactures-la.hmtl fecha de acceso febrero de 2018
16. Bruzzone S.: LFN10, LFN10-OMP y el Método de Leapfrog en el Problema de los N Cuerpos. Instituto de Física, Departamento de Astronomia, Universidad de la República y Observatorio Astronómico los Molinos, Uruguy (2011)
17. https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html fecha de acceso: febrero 2018
18. Tsuyoshi H., Keigo N.: 190 TFlops Astrophysical N-body Simulation on cluster of GPUs. Universidad de Nagasaki. IEEE 978-1-4244-7558-2 (2010)
19. Montes de Oca E., De Giusti L., Chichizola F., De Giusti A., Naiouf M.: Análisis de uso de un algoritmo de balanceo de carga estático en un Cluster Multi-GPU Heterogéneo. XV Workshop de Procesamiento Distribuido y Paralelo. CACIC2016. San Luis, San Luis, Argentina (2016)
20. "RIGOL User's Guide MSO1000Z/DS1000Z Series Digital Oscilloscope": http://beyondmeasure.rigoltech.com/acton/attachment/1579/f-050a/1/-/-/-/-/MSO1000Z%26DS1000Z_UserGuide.pdf (2018)
21. Herramienta desarrollada por el Dr. Balladini J. de la Universidad Nacional del Comahue
22. Nvidia GPU Computing: www.nvidia.com/object/what-is-gpu-computing.html (2018)
23. Adrian Pousa, Victoria Sanz, Armando De Giusti. Análisis de rendimiento de un algoritmo de criptografía simétrica sobre GPU y Cluster de GPU. Instituto de Investigación en Informática LIDI, Fac. de Informática, UNLP. HPC La TAM 2013. (2013)