



Informe Técnico CESGA-2007-001

Benchmarks para el análisis de nuevas arquitecturas:

Xeon Single, Dual y Quadcore

J.Carlos Mouriño, Aurelio Rodríguez
CESGA. Centro de Supercomputación de Galicia
e-mail: aplicacions@cesga.es

Resumen: En este documento se resumen los resultados de una serie de benchmarks sintéticos y de aplicaciones reales ejecutados en las nuevas arquitecturas multicore de intel presentes en los servidores del CESGA.

Palabras Clave: Benchmarks, Arquitectura Multicore, Rendimiento.

Este trabajo ha sido cofinanciado por el FSE (Fondo Social Europeo)

Identificador documento: DOC_APL_Benchmarks-Multicore_V0

Fecha: 10/04/07

Autores: José Carlos Mouriño Gallego,
Manuel Aurelio Rodríguez López

Responsable: Andrés Gómez

Status:

Historia de Revisiones

Versión	Autor	Descripción
0	Aurelio Rodríguez	Desarrollo Informe Técnico

Tabla de contenidos

1	Introducción	4
2	Benchmarks Sintéticos	5
2.1	HPC Challenge Benchmark (http://icl.cs.utk.edu/hpcc/)	5
2.2	Linpack de Intel.....	5
2.3	SparseBench (http://www.netlib.org/benchmark/sparsebench/)	5
3	No sintéticos.....	6
3.1	Montecarlo.....	6
3.2	Gaussian 03.....	7
3.3	GROMACS.....	8
3.4	AMBER.....	9
4	Resultados.....	10
4.1	Linpack de Intel.....	10
4.2	HPCC	11
4.3	SparsenBench	12
4.4	Monte Carlo	13
4.5	Gaussian 03	14
4.5.1	Test 1:	14
4.5.2	Test 2:	15
4.5.3	Test 3:	16
4.5.4	Prefetch por hardware:	18
4.5.5	Test 4:	20
4.6	Gromacs y Amber	23
5	Resumen de resultados.....	25

1 Introducción

El benchmarking es una técnica utilizada para medir el rendimiento de un sistema o componente de un sistema, generalmente para ser comparado con algún parámetro de referencia. En el ámbito de la informática, puede entenderse un benchmark como el resultado de la ejecución de un programa o un conjunto de programas en una máquina, con el objetivo de estimar el rendimiento de un componente concreto o la totalidad de la misma, y poder comparar los resultados con otras máquinas.

Originalmente, el benchmarking consistía en estimar el tiempo de ejecución de un programa (medido en FLOPs). Pero actualmente con la existente variedad de arquitecturas y compiladores, este único parámetro resulta insuficiente para una comparación real.

Existen básicamente dos tipos de benchmarks: sintéticos y no sintéticos (aplicaciones). Los sintéticos suelen estar diseñados para medir el rendimiento de un componente individual de un ordenador, normalmente llevando el componente escogido a su máxima capacidad. Los no sintéticos están basados en aplicaciones reales, y simulan una carga de trabajo para medir el comportamiento global del equipo.

Con la idea de evaluar las nuevas arquitecturas / procesadores existentes en el mercado, y así asesorar en futuras incorporaciones de equipos por parte del CESGA, se escogerá un conjunto benchmarks, tanto sintéticos como no sintéticos, cuyas características describiremos a continuación.

2 Benchmarks Sintéticos

Con la idea de medir el rendimiento individual de distintos parámetros de cada maquina, se han escogido diferentes benchmarks, cada uno de ellos ideado para medir una característica concreta.

2.1 *HPC Challenge Benchmark*

Se compone básicamente de 7 tests¹:

- **HPL**: Es el benchmark Linpack TPP que mide la capacidad de ejecución en punto flotante resolviendo un sistema de ecuaciones lineales.
- **DGEMM**: Mide la capacidad de ejecución en punto flotante de una multiplicación de matrices en doble precisión.
- **STREAM**: Sencillo benchmark que mide el ancho de banda sostenido de memoria y la correspondiente capacidad de ejecución para un simple vector.
- **PTRANS** (transposición paralela de una matriz): Mide la capacidad de comunicación simultánea entre pares de procesadores. Es también útil para evaluar la capacidad total de comunicaciones que tiene la red.
- **RandomAccess**: Mide el rendimiento de los accesos aleatorios a memoria (GUPS).
- **FFT**: Mide la capacidad de ejecución en punto flotante de la Transformada de Fourier Discreta (DFT) en una dimensión con números complejos en doble precisión.
- **Communication bandwidth and latency**: conjunto de tests que miden la latencia y el ancho de banda de un conjunto de patrones de comunicación simultáneos.

2.2 *Linpack de Intel*

Este Benchmark es el test Linpack del paquete anterior pero preparado y compilado por Intel para sus procesadores Xeon, tanto a 32 como a 64 bits².

2.3 *SparseBench*

Este benchmark esta compuesto por un conjunto de tests de métodos iterativos sobre datos “sparse”³. Las matrices sparse constituyen un problema importante en el campo del análisis numérico. Al contrario que en el caso de las matrices densas, su manejo no conlleva reaprovechamiento de datos. Por lo tanto, los algoritmos para matrices sparse dependerán más de la velocidad de memoria, que de la velocidad del procesador.

Este benchmark utiliza métodos iterativos comunes, “preconditioners” y distintos esquemas de almacenamiento para evaluar el rendimiento de la máquina con operaciones típicas sobre datos sparse. Concretamente se compone de:

- Métodos iterativos GMRES y Conjugate Gradient.

¹ <http://icl.cs.utk.edu/hpcc/>

² <http://www.intel.com/ed/software/products/asmo-na/eng/266857.html>

³ <http://www.netlib.org/benchmark/sparsebench/>

- Jacobi and ILU preconditioners.
- Diagonal storage and compressed row storage matrices.

3 No sintéticos

Los benchmarks vistos hasta ahora, y los benchmarks sintéticos en general, están ideados para medir el rendimiento de un componente de manera individual, y suelen estar diseñados para extraer el máximo rendimiento de ese componente. Pero las conclusiones que se puedan extraer no siempre se cumplen a la hora de ejecutar aplicaciones reales, que ejecutan un sinfín de distintas operaciones sobre conjuntos muy variados y heterogéneos de datos. Es más no todas las máquinas obtienen mejores resultados con las mismas aplicaciones.

Es por ello que se ha escogido un conjunto de aplicaciones representativas para el CESGA, midiendo el tiempo de ejecución de cada una de ellas sobre las distintas arquitecturas y/o procesadores.

3.1 *Montecarlo*

El método Montecarlo es un método numérico que permite resolver problemas físicos y matemáticos mediante la simulación de variables aleatorias. Su importancia se basa en la existencia de problemas que tienen difícil solución por métodos exclusivamente analíticos o numéricos, pero que dependen de factores aleatorios o se pueden asociar a un modelo probabilística artificial.

Gracias al avance de los ordenadores, cálculos Montecarlo que en otro tiempo hubieran sido inconcebibles, hoy en día se presentan como asequibles para la resolución de ciertos problemas. En estos métodos el error depende del número de pruebas, por lo que para ganar una cifra decimal en la precisión, el número de pruebas debe elevarse exponencialmente.

Sus aplicaciones actuales se extienden a campos científicos y técnicos tan variados como son los de la Física estadística, biología molecular, genética, redes de información, telecomunicaciones o finanzas.

Muchas de las aplicaciones desarrolladas recientemente por los usuarios del CESGA utilizan esta técnica, por lo que es fundamental medir el rendimiento de un método de este tipo.

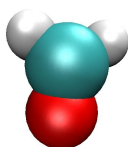
El código elegido para este benchmark está desarrollado por el usuario Manuel Martínez, de la Universidad de Vigo, que aplica el método a cadenas moleculares con el fin de calcular propiedades derivadas de hidrocarburos de alta presión. Este código ha sido revisado y optimizado por los técnicos del CESGA, y creemos que es suficientemente representativo para este tipo de aplicaciones.

3.2 Gaussian 03

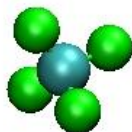
Gaussian 03 es una de las aplicaciones más utilizadas en el CESGA. Este paquete de programas permite el cálculo de propiedades (energías, frecuencias de vibración, geometría óptima, etc.) de moléculas e intermedios de reacción, fundamentalmente mediante métodos *ab initio* aunque también incorpora métodos semiempíricos⁴.

Se han elegido 4 casos que intentan abarcar los distintos requisitos computacionales demandados por los distintos algoritmos involucrados en la ejecución:

- 1) **test139 48Mb RAM 26Mb disk**: test número 139 de Gaussian 03, este test presenta una baja demanda de memoria y disco.
- 2) **incore 665Mb RAM**: test número 23 de Gaussian 03 modificado para el cálculo incore con un uso de 665Mb de RAM y baja demanda de disco.

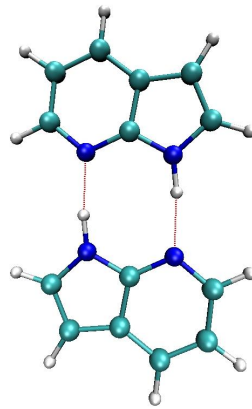


- 3) **512Mb RAM 1.5Gb**: Cálculo de Coupled Clusters semiconvencional con requerimientos de 512Mb de RAM y 1.5Gb de disco.



- 4) **Parallel**:
 - a. Cálculo de optimización MP2 con requerimientos de 1700Mb de RAM y baja demanda de disco.
 - b. Cálculo de optimización DFT con requerimientos de 1700Mb de RAM y baja demanda de disco.
 - c. Cálculo paralelo DFT incore con requerimientos de 1700Mb de RAM.

⁴ Gaussian 03, Revision C.03, M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, J. A. Montgomery, Jr., T. Vreven, K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, C. Gonzalez, and J. A. Pople, Gaussian, Inc., Wallingford CT, 2004. <http://www.gaussian.com>



Para los 3 primeros tests se medirán tiempos ejecutando 1,2,4 y hasta 8 copias simultáneas en el caso del procesador quadcore concurrentemente. De esta manera se intenta medir la capacidad de throughput en la arquitectura concreta para esta aplicación.

A su vez estos tres primeros casos se ejecutarán con las siguientes opciones de la BIOS deshabilitadas relacionadas con la optimización del rendimiento mediante técnicas de prefetch por hardware en el nodo quadcore⁵:

- **Hardware Prefetcher (Data Prefetch Logic)**
- **Adjacent Cache Line (L2 Streaming Prefetch)**

El 4º test (caso a y b) se ejecutará demandando 1,2 y 4 procesadores mediante ejecución paralela de la implementación de memoria compartida (OpenMP) de Gaussian 03. Con este test se pretende probar el escalado paralelo del algoritmo concreto dentro de esta aplicación para cada arquitectura. El caso c, centrado únicamente en el quadcore, pretende medir la capacidad de throughput con aplicaciones paralelas y se ejecutará concurrentemente varios procesos demandando diferente número de hilos.

3.3 GROMACS

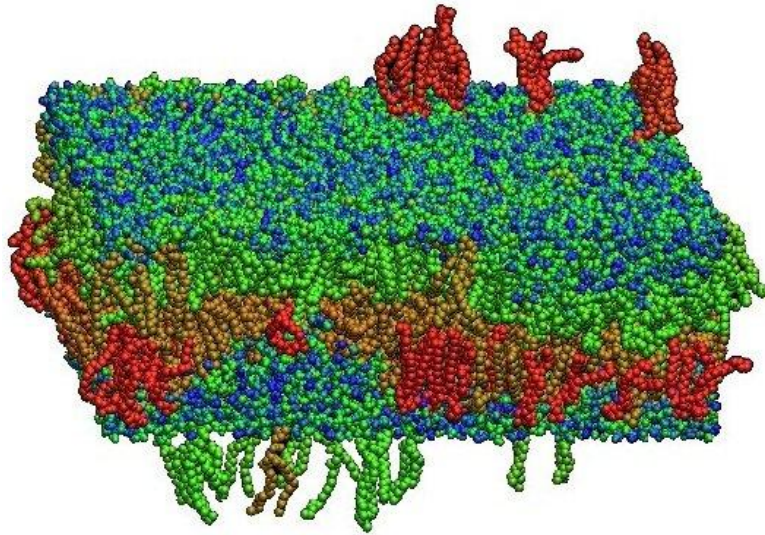
GROMACS es un software versátil para cálculos de dinámica molecular, es decir, simular las ecuaciones de Newton del movimiento para sistemas de cientos a millones de partículas.⁶

Se ejecutará en paralelo usando la versión compilada con OpenMPI el benchmark DPPC de la aplicación⁷. Con este test se pretende probar el escalado paralelo del algoritmo concreto dentro de esta aplicación para cada arquitectura.

⁵ <http://www.intel.com/cd/ids/developer/asmo-na/eng/dc/xeon/298229.htm?page=2>

⁶ Versión 3.3.1 <http://www.gromacs.org>

⁷ <http://www.gromacs.org/gromacs/benchmark/benchmarks.html>



DPPC

A phospholipid membrane, a total of 121,856 atoms.

3.4 *AMBER*

AMBER es un conjunto de programas que permiten al usuario realizar simulaciones de dinámica molecular, fundamentalmente en biomoléculas, basadas en teorías de campos de fuerza⁸.

Se ejecutarán en serie los benchmarks estandar de AMBER 8.0⁹. Esta ejecución complementa al test con GROMACS probando los algoritmos involucrados en programas de modelización molecular clásica.

⁸ <http://amber.scripps.edu/amber.html>

⁹ <http://amber.scripps.edu/amber8.bench1.html>

4 Resultados

Se han elegido 4 procesadores Xeon diferentes presentes en el CESGA para ser evaluados mediante estos Benchmarks:

- **Nodo Eimrt:** contiene 2 procesadores Intel Xeon a 3.60GHz con 2048KB de caché, 4Gb de RAM y SO 32bits CentOS release 4.3 (Final) con kernel 2.6.9.
- **Nodo DualCore:** 2 procesadores duales Intel Xeon 5160 a 3.00GHz con 4096KB de caché, 4Gb de RAM y SO 32bits Scientific Linux Release 3.0.5 con kernel 2.4.21.
- **Nodo QuadCore:** 2 procesadores QuadCore Intel Xeon E5310 a 1.6 GHz con 4096KB de caché, 4Gb y 8Gb de RAM y SO 64bits Scientific Linux SL release 4.4 (Beryllium) con kernel 2.6.9 .
- **Nodo QuadCore2:** 2 procesadores QuadCore Intel Xeon X5355 a 2.66GHz con 409 KB de caché , 8Gb de RAM y SO 64bits Scientific Linux SL release 4.4 (Beryllium) con kernel 2.6.9 .

	Frecuencia/Relación	Nº cores	cache	RAM	disco	SO
Nodo Eimrt	3.60GHz / 2.25	2	2048KB	4Gb	SCSI	32bits
Nodo DualCore	3.00GHz / 1.875	4	4096KB	4Gb	SATA	32bits
Nodo QuadCore	1.60GHz / 1	8	4096KB	4Gb/8Gb	SCSI	64bits
Nodo QuadCore2	2.66 GHz / 1.66	8	4096KB	8Gb	SCSI	64bits

Para todos los tests sintéticos, excepto el de Intel que viene precompilado se ha utilizado el compilador de Intel v. 9.1, con las mejores opciones de compilación encontradas para cada caso.

En todas las tablas y gráficas aparte de las medidas sin modificar aparece un dato extra en azul en el caso de las tablas y una gráfica extra en el caso de las graficas donde se expresan los resultados normalizados y considerando la relación de frecuencia presente entre los distintos cores.

4.1 Linpack de Intel

En la siguiente tabla se muestran los resultados en GFLOPS de ejecutar el test Linpack precompilado por Intel para cada uno de sus procesadores. Entre paréntesis se indica en miles el tamaño máximo de datos con el que se ha podido ejecutar el test. Puede observarse que este tamaño es de 22000 (unos 4Gb de memoria) en el caso de 64 bits y de 16000 (unos 2 Gb de memoria) para el caso de SO de 32 bits.

	QuadCore	QuadCore2	DualCore	Eimrt
1	5.68/1 ₍₂₂₎	8.55/0.91 ₍₃₂₎	7.17/0.67 ₍₁₆₎	5.73/0.45 ₍₁₆₎
2	11.51/1 ₍₂₂₎	17.48/0.91 ₍₃₂₎	14.15/0.66 ₍₁₆₎	11.05/0.43 ₍₁₆₎
4	21.97/1 ₍₂₂₎	35.77/0.98 ₍₃₂₎	26.51/0.64 ₍₁₆₎	-
8	38.06/1 ₍₂₂₎	56.66/0.90 ₍₃₂₎	-	-

 Tabla 1.- *Resultados del test Linpack de Intel*

4.2 HPCC

Esta suite de test se compone de varios benchmarks individuales. Presentamos el HPL por separado y después una tabla comparativa del resto de los tests.

	QuadCore	QuadCore2	DualCore	Eimrt
1	3.85/1 ₍₈₎	5.92/0.93 ₍₂₂₎	7.09/0.98 ₍₁₃₎	5.68/0.65 ₍₁₅₎
2	7.58/1 ₍₁₁₎	11.75/0.93 ₍₂₉₎	13.81/0.97 ₍₁₇₎	10.82/0.63 ₍₁₉₎
4	14.68/1 ₍₂₁₎	23.94/0.98 ₍₃₁₎	25.24/0.92 ₍₂₂₎	
8	25.71/1 ₍₂₂₎	40.73/0.95 ₍₃₁₎		

 Tabla 2.- *Resultados del test HPL de HPCC*

En la tabla se muestran de nuevo los resultados del test de linpack, pero en este caso compilado y parametrizado por nosotros. Se han probado distintas compilaciones e infinidad de valores en sus parámetros. Aquí se muestran los resultados que se han conseguido en el mejor de los casos. En el caso de los Procesadores Xeon y Dual, los resultados están muy próximos a los conseguidos con el propio test de Intel, lo que indica la buena elección de opciones de compilación y parámetros que se ha hecho.

Sin embargo, en el QuadCore, los resultados obtenidos distan bastante de los conseguidos con el propio test de Intel. Esto puede ser debido principalmente al compilador, ya que la versión que tenemos instalada, no contempla este tipo de procesadores. Se ha compilado utilizando las mismas opciones que si fueran Dual core. Además, mientras que en los otros dos casos se ha compilado en la misma máquina en la que se iba a ejecutar, en este caso no ha sido posible, ya que no se disponía de licencia para estas máquinas.

Es interesante comprobar en este caso el tamaño máximo de datos permitido, que parece que crece a medida que se utilizan más cores. No encontramos una respuesta convincente a este fenómeno, por lo que es un caso pendiente de tratar y estudiar.

		QuadCore	QuadCore2	DualCore	Eimrt
DGEMM (Gflop/s)		5.36/1	6.40/0.72	7.27/0.73	5.91/0.49
STREAM (GB/s)		2.50	2.85	4.11	3.57
PTRANS (GB/s)		0.699	0.633	0.682	0.689
Random Access	Latency (msec)	0.000985	0.000936	0.000912	0.000939
	Bandwidth (MB/s)	335.80	362.59	508.80	372.97
FFT - single (Gflop/s)		0.74/1	0.98/0.80	1.01/0.73	0.63/0.38
Communications	Latency (msec)	0.017300	0.000339	0.000331	0.000745
	Bandwidth (MB/s)	1216.181	5896.620	3799.188	824.069

 Tabla 3.- *Resumen de resultados del test HPCC*

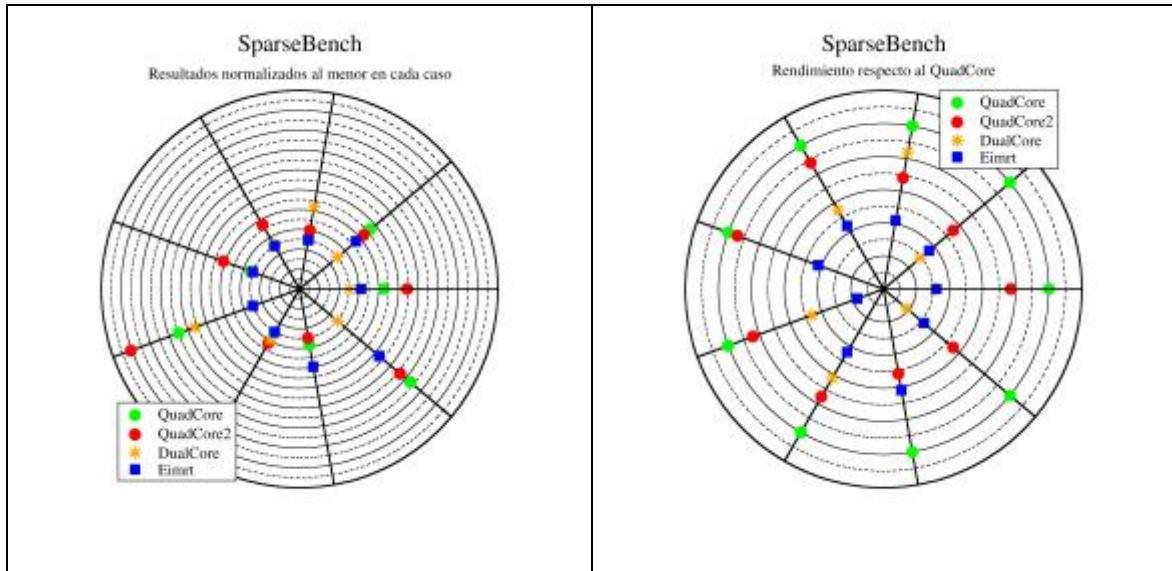
En esta segunda tabla se muestra un resumen de resultados de los otros benchmarks incluidos en la suite HPCC. Tal y cómo se han medido los tests, estos resultados no tienen una gran relevancia, ya que las ejecuciones se han ido ajustando para conseguir un valor máximo en el test HPL, y no en cualquiera de los otros tests. Además estos resultados son los mejores en cualquiera de las ejecuciones que se han realizado, no teniendo porque corresponder a la misma ejecución ni a la misma combinación de parámetros.

4.3 *SparserBench*

En la tabla se muestra el rendimiento máximo obtenido en MFLOPs tras la ejecución de este test, y el rendimiento asintótico de diversas operaciones con datos sparse.

	QuadCore	QuadCore2	DualCore	Eimrt
Highest attained	3777/1	3464/0.55	2034/0.29	3035/0.36
Diagonal storage Matrix-vector product	395/1	444/0.68	624/0.84	371/0.42
Symmetrically stored CRS Matrix-vector product	235/1	345/0.88	241/0.55	232/0.44
CRS Matrix-vector product	293/1	457/0.94	-	280/0.42
Diagonal storage ILU solve	267/1	373/0.84	232/0.46	103/0.17
Symmetrically stored CRS ILU solve	209/1	260/0.75	243/0.62	206/0.44
CRS ILU solve	144/1	126/0.52	-	200/0.62
Vector operations in CG	851/1	771/0.55	291/0.18	612/0.32
Diagonal storage Block Jacobi solve	370/1	473/0.77	218/0.31	271/0.32

 Tabla 4.- *Resultados del test SparseBench*



4.4 Monte Carlo

De este código se han hecho dos ejecuciones en cada uno de los procesadores, una con 100 ciclos y otra con 200 ciclos. Como la aplicación no es paralela, sólo se han realizado ejecuciones sobre un único core.

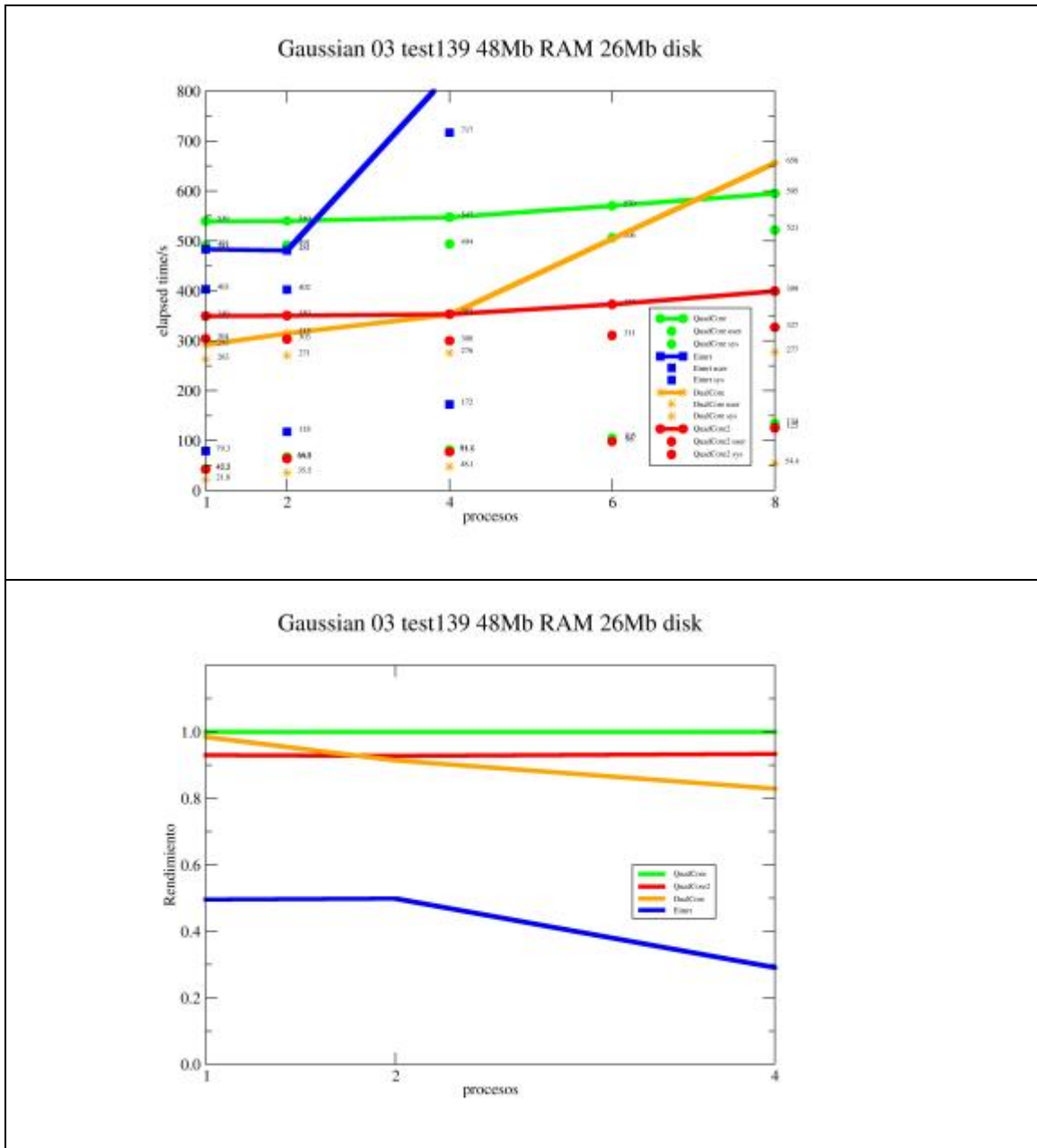
Elapsed time/s	QuadCore	QuadCore2	DualCore	Eimrt
100 ciclos	190/1	118/1.03	103/0.98	150/0.56
200 ciclos	381/1	229/1.00	205/0.99	300/0.56

Tabla 5.- Resultados del test de Monte Carlo

En la tabla se muestran los tiempos de estas ejecuciones en segundos. Puede observarse que una vez más los cores del Dual son los que obtienen un mejor resultado, debido, seguramente a su combinación de frecuencia caché, y los Quad son los que obtienen un peor rendimiento (aunque sería en la máquina que mas casos podríamos ejecutar simultáneamente gracias a su mayor número de cores).

4.5 Gaussian 03

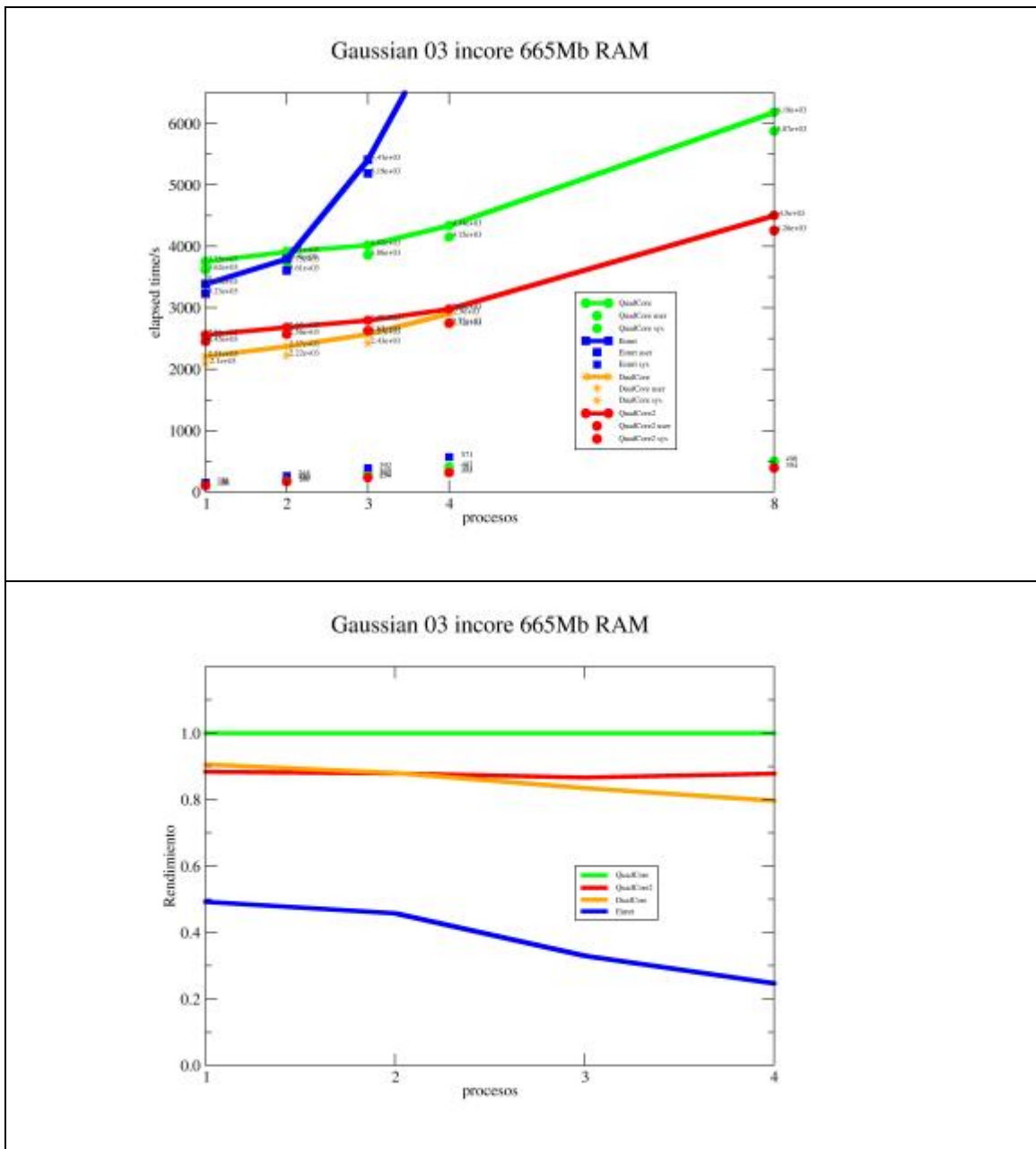
4.5.1 Test 1:



Este test ejecuta procesos concurrentemente. Se aprecia un pequeño degradado en el rendimiento según aumenta el número de procesos. En el caso del procesador quadcore el degradado es sólo de un 10% en la ejecución de 8 procesos (número de cores presente en el nodo).

Si tenemos en cuenta el factor frecuencia el procesador quadcore presenta un apreciable mejor rendimiento según aumenta el número de procesos.

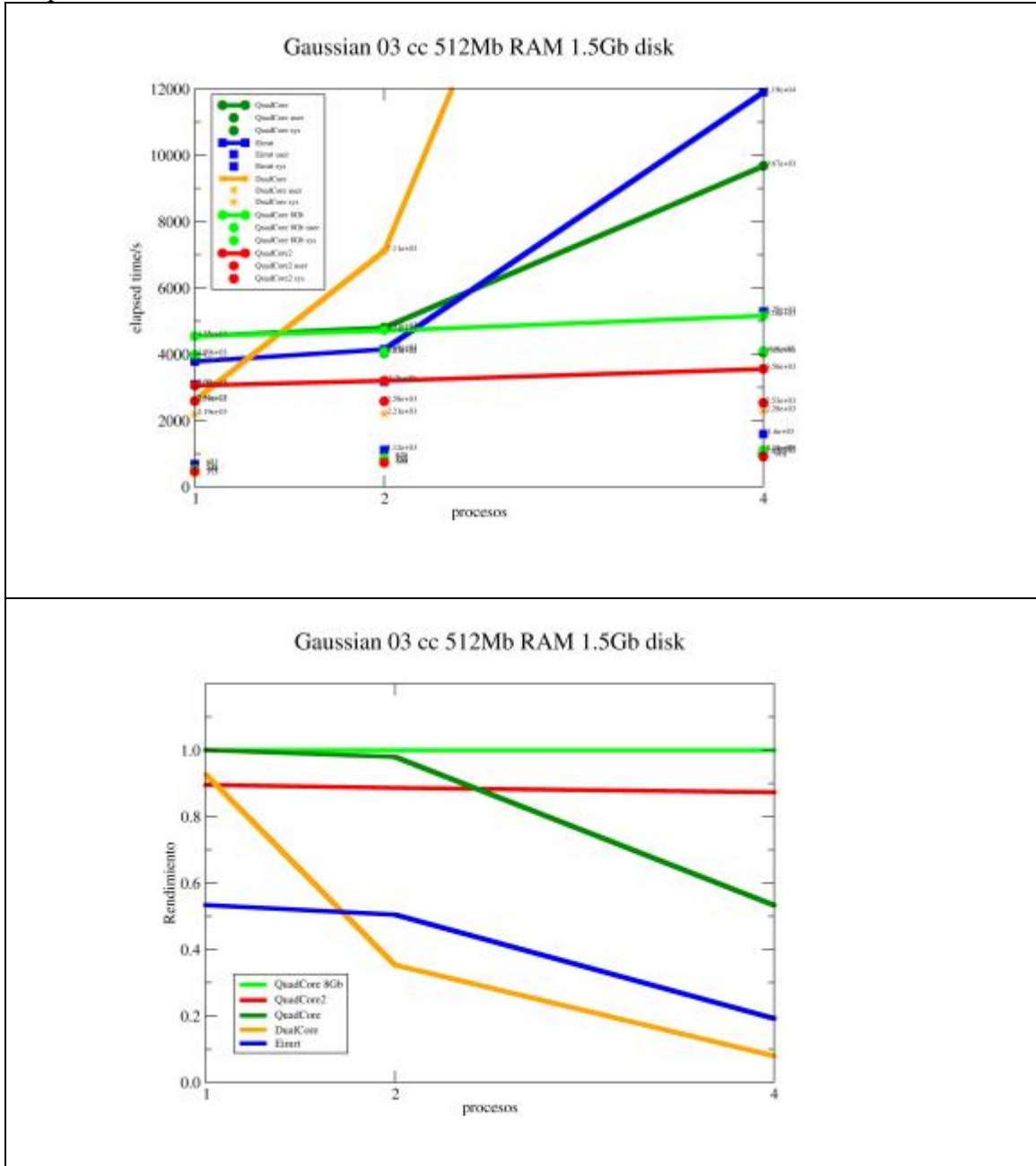
4.5.2 Test 2:

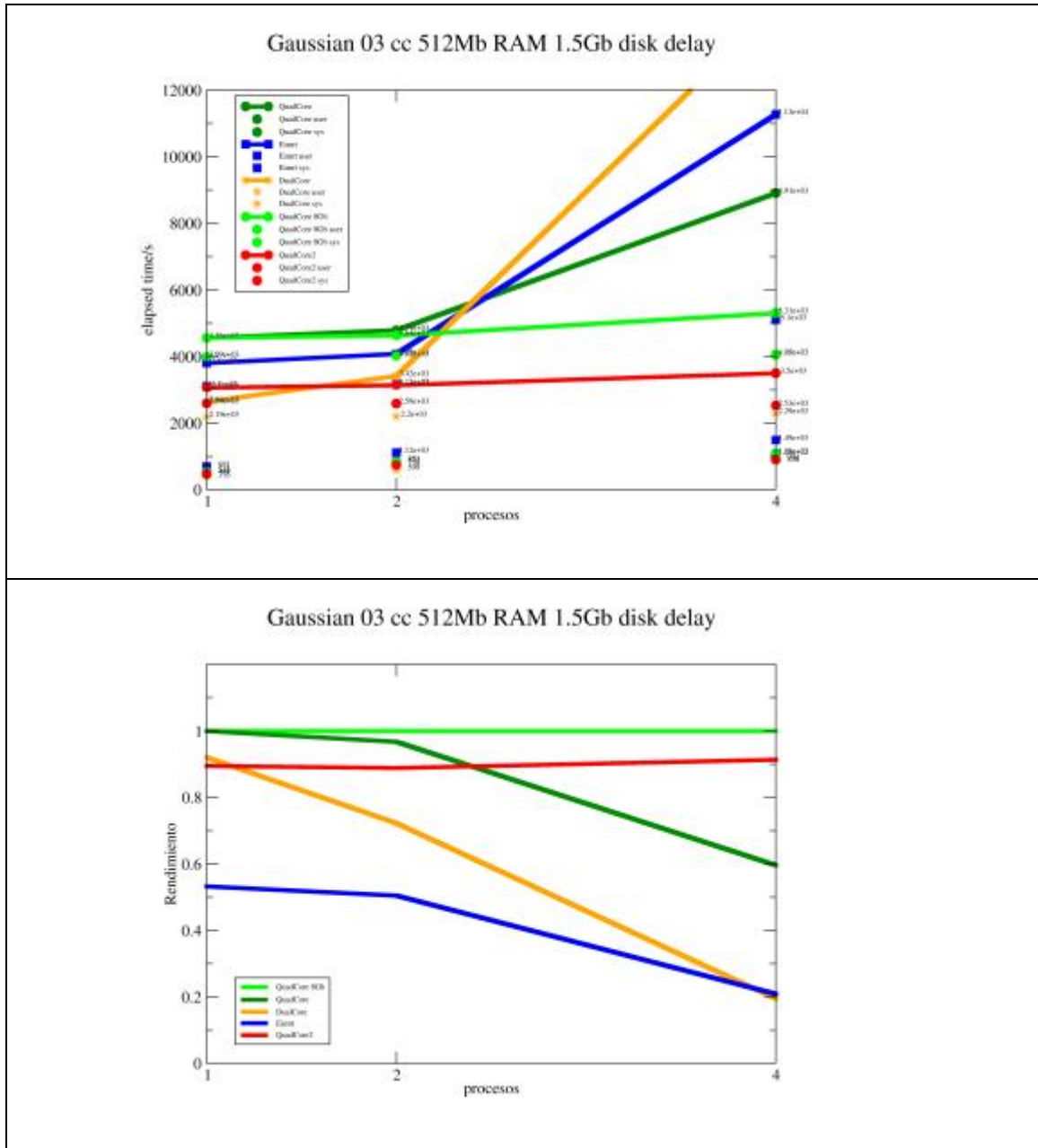


Este test también ejecuta procesos concurrentemente aunque estos procesos presentan un consumo de memoria RAM elevado. Se aprecia un degradado mucho más importante en el rendimiento según aumenta el número de procesos. En el caso del procesador quadcore el degradado es de un 60% en la ejecución de 8 procesos (número de cores presente en el nodo, este caso ha sido ejecutado en un nodo con 8Gb de RAM). Teniendo en cuenta el factor de frecuencia se observa un rendimiento apreciablemente superior en el procesador quadcore.

4.5.3 Test 3:

Este test también ejecuta procesos concurrentemente aunque estos procesos presentan un consumo de Disco elevado. Se han realizado 2 ejecuciones, en la primera los procesos son ejecutados concurrentemente mientras que en la segunda se introduce un retraso aleatorio entre 1 y 2 minutos en la ejecución. A su vez esta segunda ejecución se ha probado también en un nodo con 8Gb de RAM.

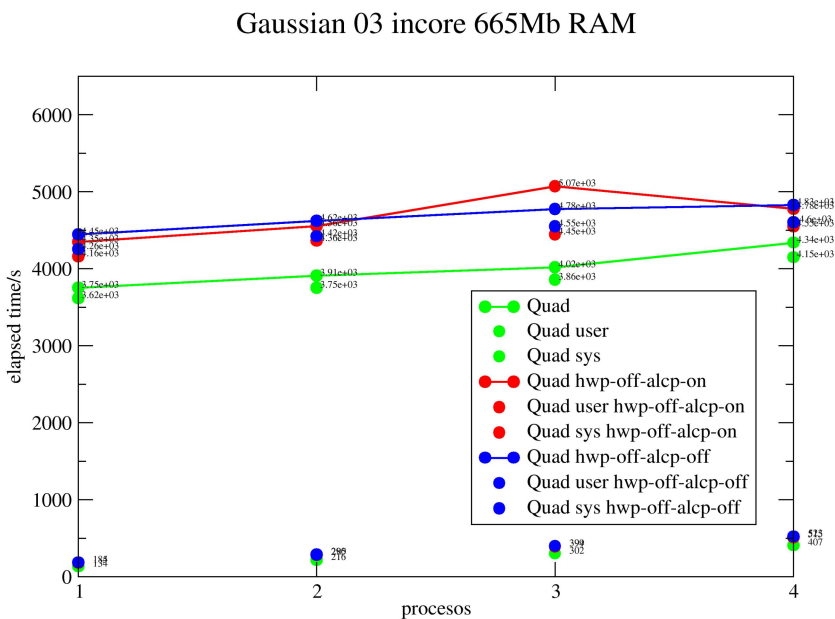
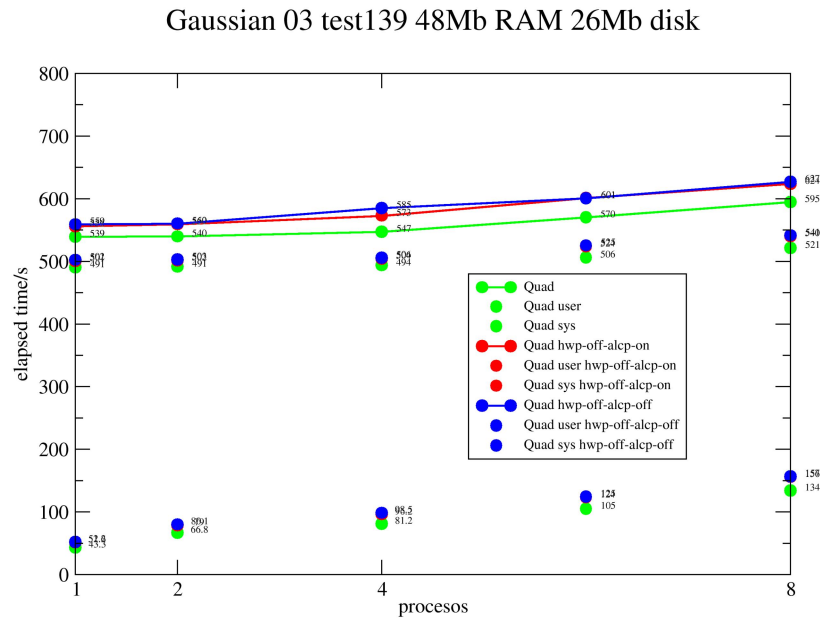




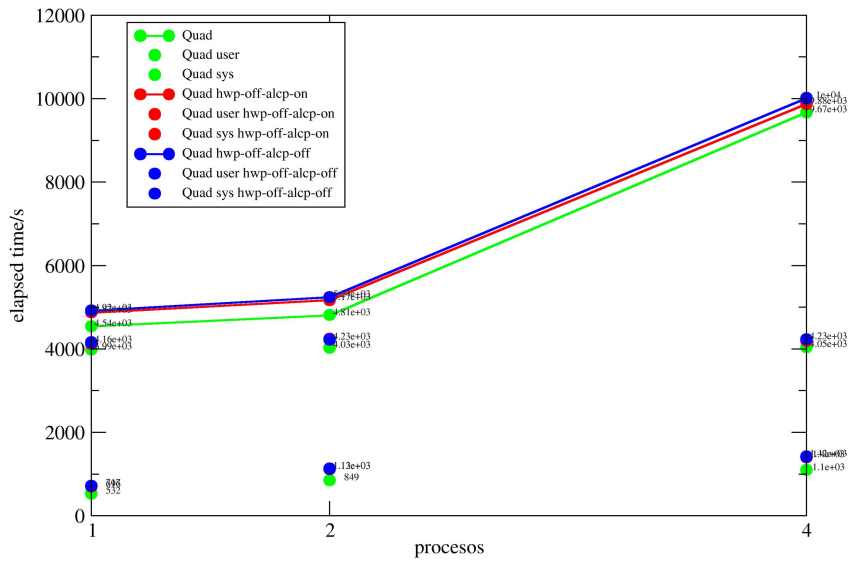
En general se aprecia un degradado muy importante en el rendimiento en cuanto se superan 2 procesos concurrentes. En el caso ejecutado con un retraso el degradado es menos importante aunque aún muy marcado pero especialmente trascendente para el caso del nodo con dualcore probablemente debido al disco SATA de este frente a los SCSI del resto. Cabe destacar la importancia del cache de disco, en el caso de la ejecución en el nodo quadcore con 8Gb de RAM el degradado ha disminuido radicalmente en el caso con 4 procesadores concurrentes.

4.5.4 Prefetch por hardware:

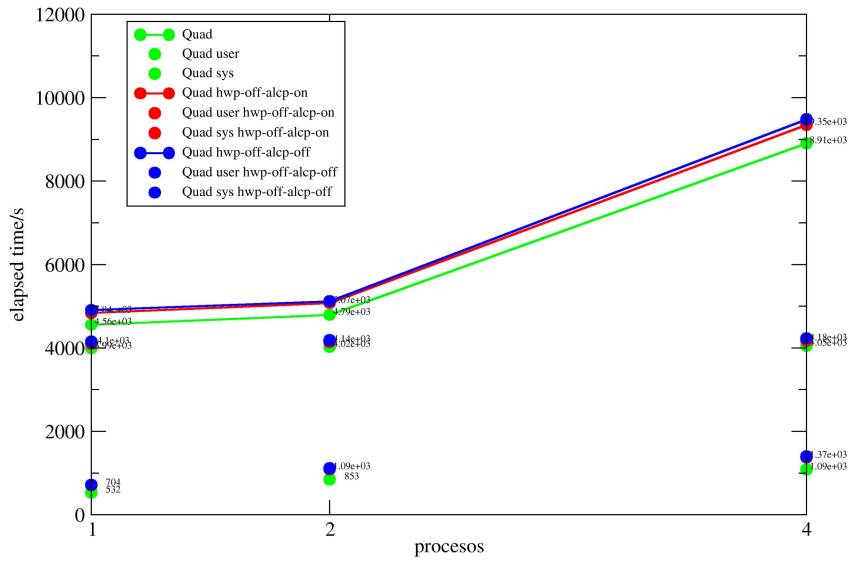
A continuación en el nodo quadcore se deshabilitó en un caso únicamente la opción de la BIOS Hardware Prefetcher (Data Prefetch Logic) (**caso hwp-off-alcp-on**) y en otro tanto la opción Hardware Prefetcher (Data Prefetch Logic) como la opción Adjacent Cache Line (L2 Streaming Prefetch) (**caso hwp-off-alcp-off**) y se procedió a ejecutar los 3 tests anteriores. Los resultados se exponen a continuación:



Gaussian 03 cc 512Mb RAM 1.5Gb disk



Gaussian 03 cc 512Mb RAM 1.5Gb disk delay

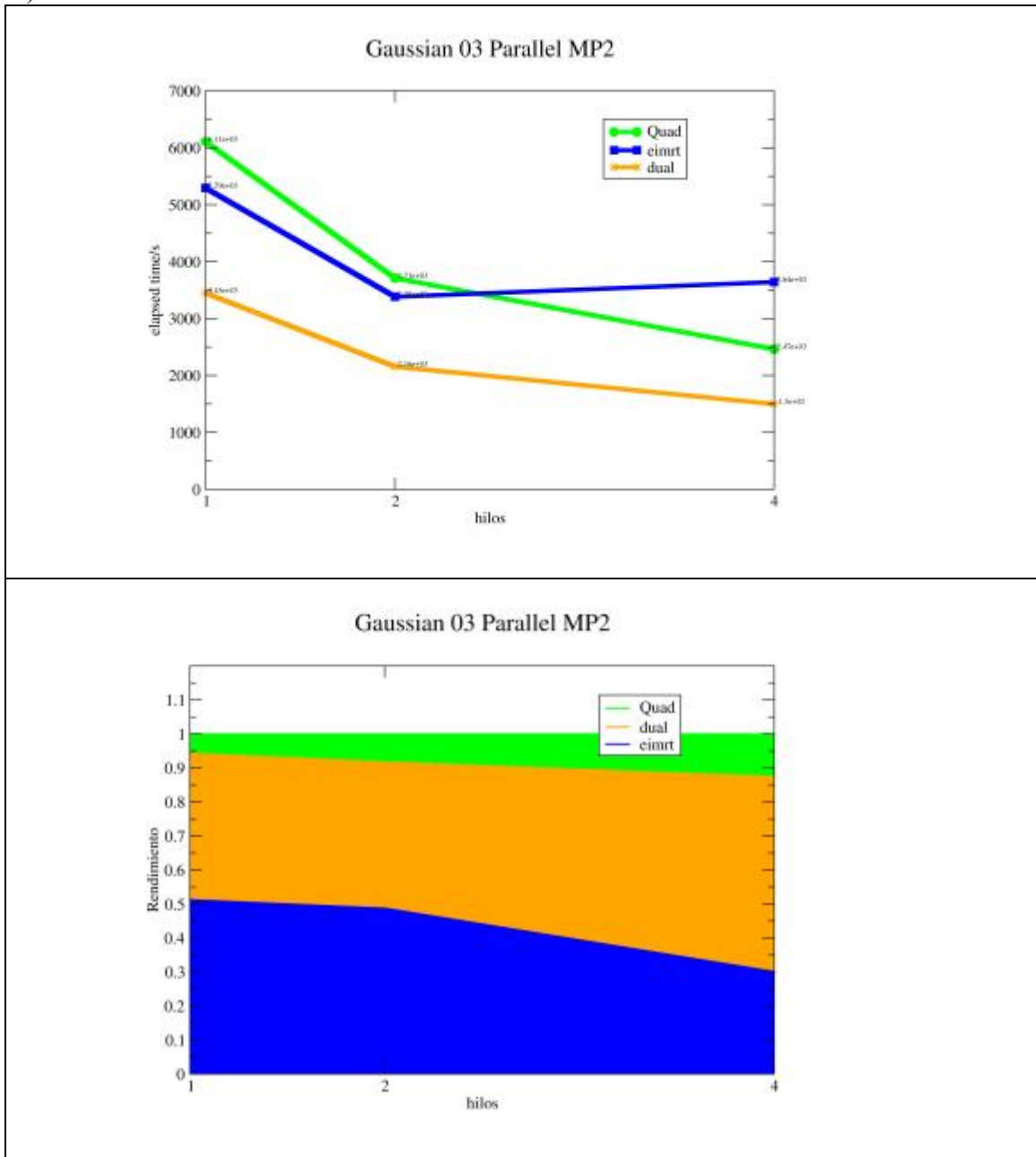


Se aprecia como el “Hardware Prefetcher” tiene una influencia apreciable en el cálculo. Incrementos de hasta un 10% en el tiempo de ejecución hacen que esta opción sea recomendable en nodos de cálculo utilizados para la ejecución de este tipo de software.

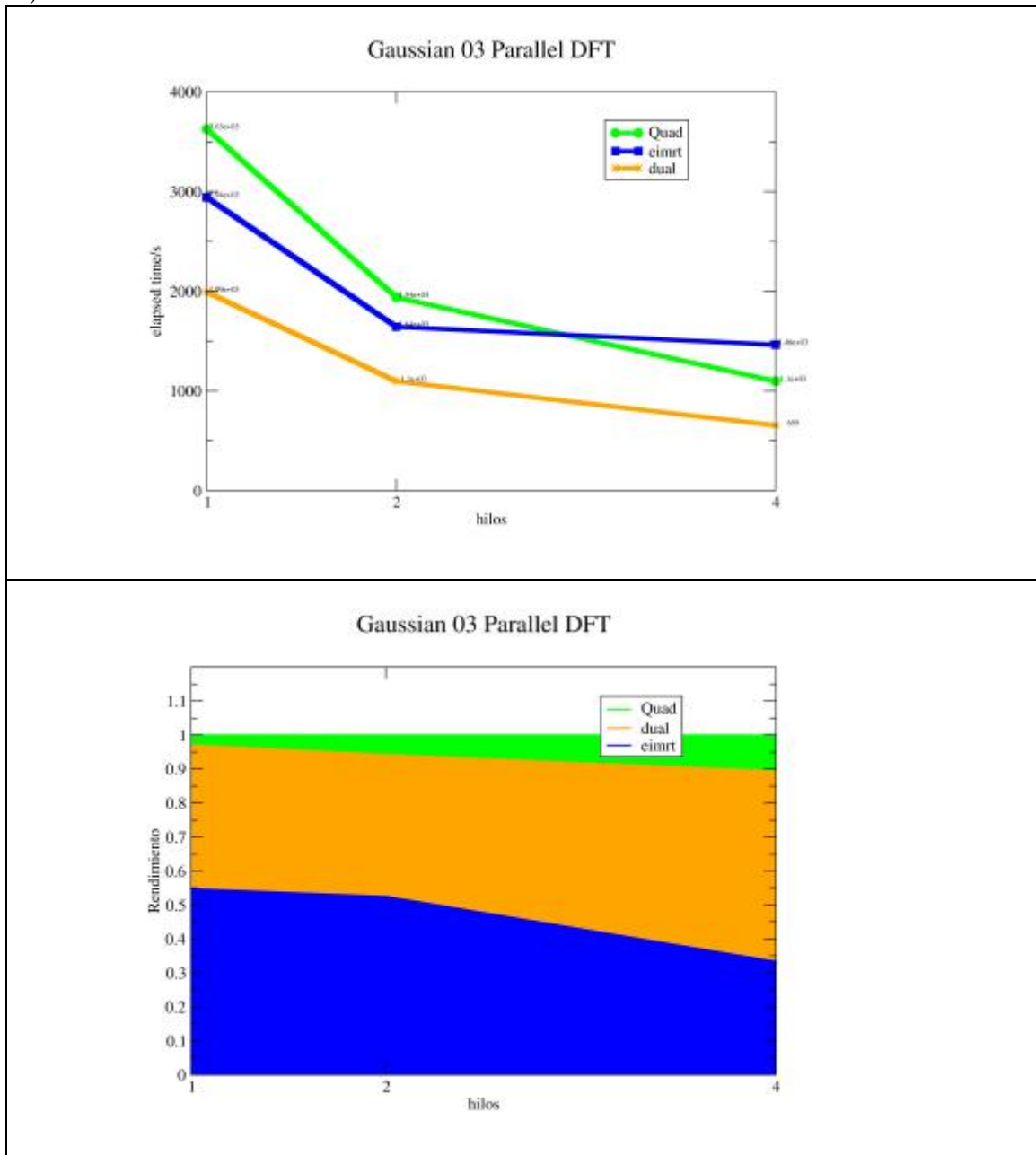
4.5.5 Test 4:

En los casos a) y b) esta vez no se ejecutan procesos concurrentemente sino que se busca medir el escalado de la paralelización OpenMP presente en el programa. Para ello se han elegido 2 métodos ampliamente usados: MP2 y DFT y se ha diseñado un test con un consumo de memoria de 1700Mb.

a)



b)

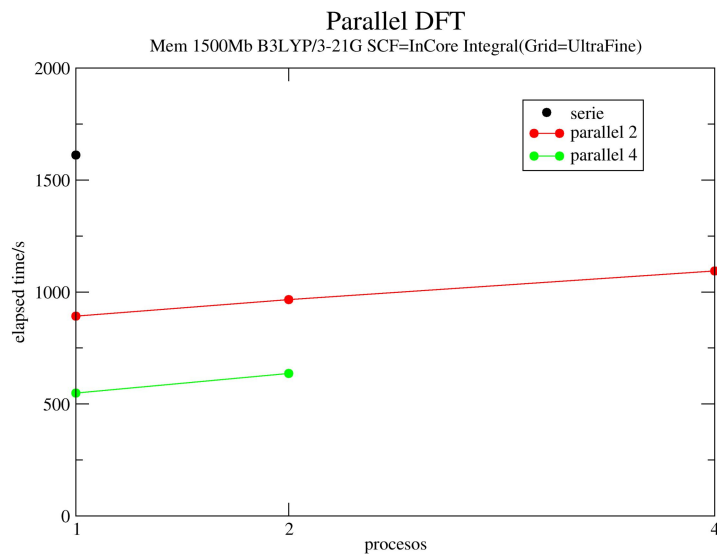


Se aprecia un mejor escalado en el procesador quadcore pudiendose llegar a intuir la consecución de un rendimiento similar a los duales según aumenta el número de procesadores. Sería interesante tener una confirmación de esta suposición.

A la vez se aprecia un muy pequeño escalado en el nodo eimrt en el paso de 2 a 4 hilos en el cálculo DFT que se puede atribuir al hyperthreading presente en esta arquitectura. Si tenemos en cuenta el factor de frecuencia entre los procesadores se aprecia un rendimiento similar entre el dual y el quadcore.

c) En este caso se intenta discernir el modelo de ejecución más óptima que combine menor tiempo de ejecución del proceso y mayor aprovechamiento del nodo quadcore.

Para ello se han ejecutado 4 procesos concurrentes que usan 2 hilos para su ejecución y 2 procesos concurrentes que usan 4 hilos en su ejecución.

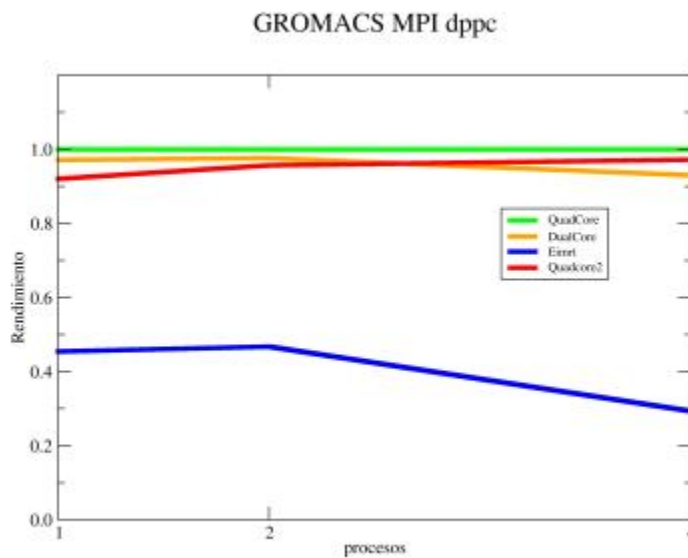
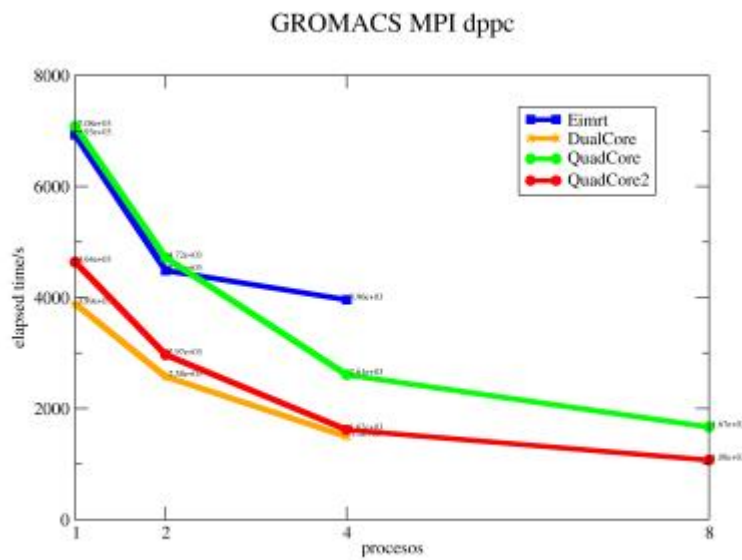


Se aprecia que ambas alternativas, 4 procesos x 2 hilos y 2 procesos x 4 hilos, son adecuadas para la ejecución de estos procesos no presentando una diferencia en degradado excesiva. Por supuesto esta conclusión es altamente dependiente del escalado paralelo del trabajo a ejecutar.

4.6 Gromacs y Amber

Con estos tests se intenta probar los algoritmos involucrados en modelización molecular clásica. Se han elegido estas aplicaciones por ser ampliamente utilizadas en este campo y por los usuarios del CESGA.

En el caso de Gromacs se intenta ver el escalado paralelo de la implementación MPI de Gromacs en las distintas arquitecturas. La ejecución paralela involucra únicamente procesos intranodo.

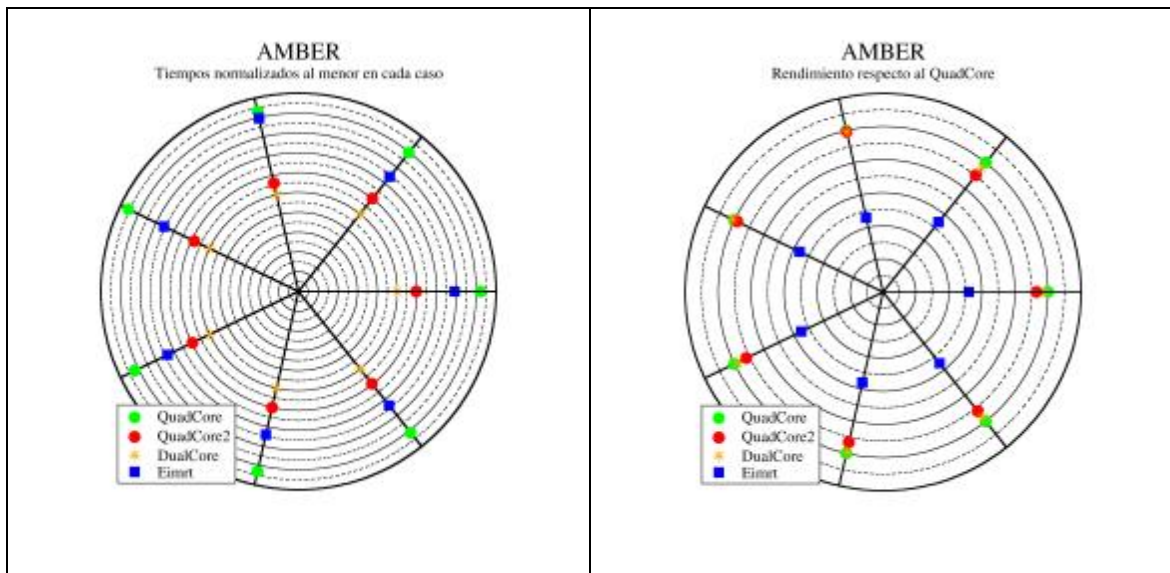


Esta aplicación presenta un buen escalado hasta 8 procesos lo que ha implicado la obtención de un rendimiento similar con el nodo quadcore comparado con el dualcore a pesar de su más baja frecuencia de reloj. A su vez cabe destacar una ganancia significativa en el caso del eimrt al pasar de 2 a 4 procesos que confirma la ventaja del hyperthreading presente en este nodo.

A continuación se exponen los tiempos de ejecución de los benchmarks estandar de Amber 8.0 en las distintas arquitecturas.

	Quadcore Total time/s	Quadcore2 Total time/s	Dualcore Total time/s	Eimrt Total time/s
factor_ix	1241/1	831/0.90	693/0.96	1027/0.54
gb_cox2	585/1	354/0.99	315/0.99	565/0.46
gb_mb	67/1	41/0.98	35/1.01	53/0.56
hb	561/1	364/0.92	305/0.98	449/0.55
jac	748/1	484/0.93	403/0.99	595/0.56
rt	1792/1	1172/0.92	982/0.97	1445/0.55
trx	57/1	37/0.93	31/0.98	49/0.52

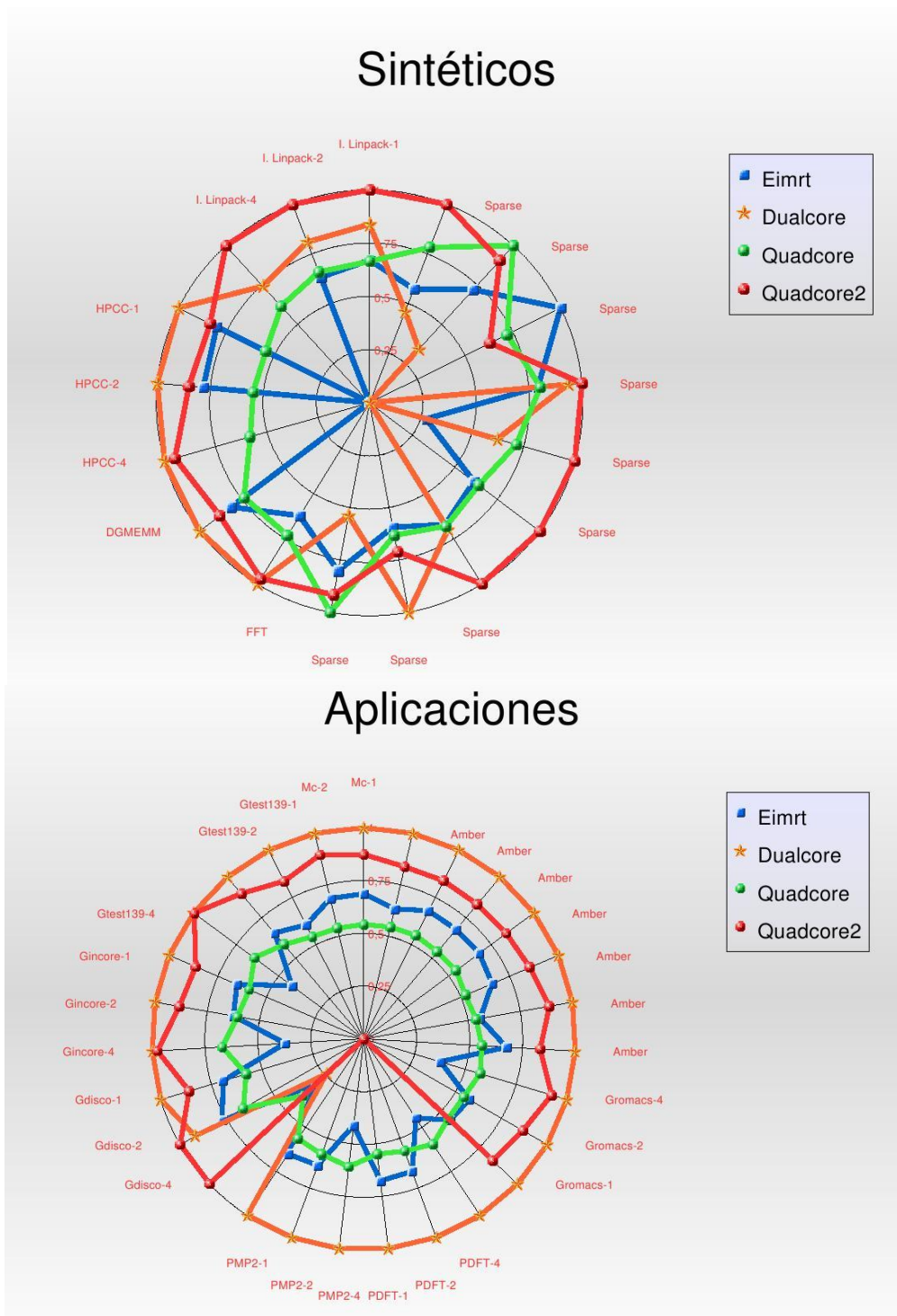
Tabla 6.- *Benchmarks estandar de Amber 8.0*



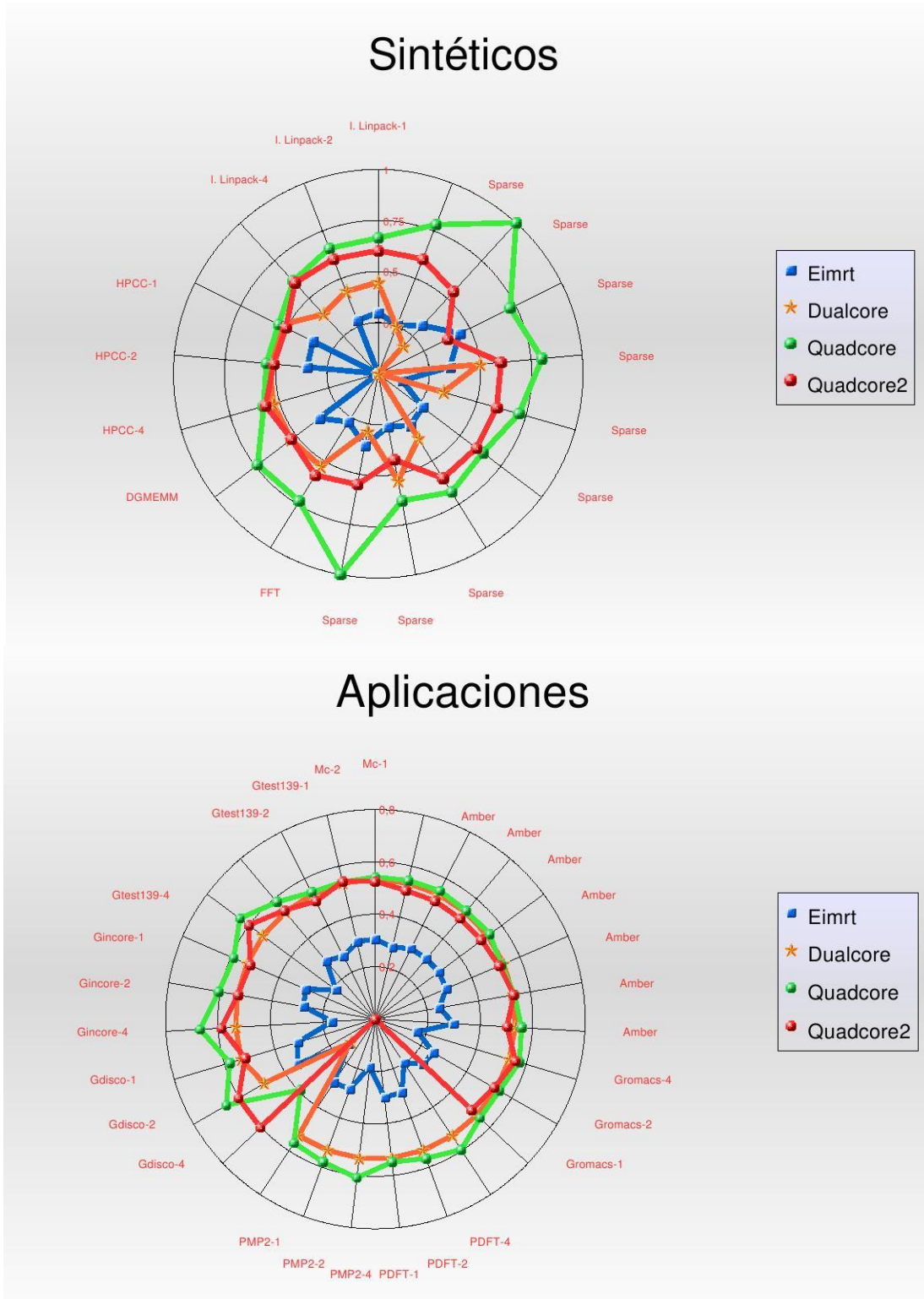
Cabe destacar el similar rendimiento entre el procesador Quadcore y el Eimrt a pesar de enorme diferencia de frecuencia de reloj presente entre los 2 procesadores. Si comparamos teniendo en cuenta el factor de frecuencia el rendimiento del quadcore y el dualcore es el mismo.

5 Resumen de resultados.

A continuación se exponen los resultados normalizados al rendimiento máximo alcanzado entre las distintas arquitecturas: (nótese que de algunos casos no han sido tomados datos por lo el resultado es fijado a cero en las figuras)



Resultados de rendimiento normalizados al rendimiento máximo divididos por el factor de frecuencia de reloj: (nótese que de algunos casos no han sido tomados datos por lo el resultado es fijado a cero en las figuras)



De estos gráficos se puede apreciar como la arquitectura presente en el QuadCore es la más equilibrada:

- En el caso de los benchmarks sintéticos un aumento de frecuencia de reloj hace disminuir únicamente el rendimiento en los casos con accesos a memoria dispersos (sparse, FFT).
- En el caso de los benchmarks de aplicaciones el rendimiento es casi independiente de la frecuencia y las pérdidas de rendimiento en los casos con acceso a disco elevado puede ser atribuible al menor buffer de disco.