

Paralelización del proceso de calcular Pi en el lenguaje C utilizando el Clúster Kabré y la plataforma OpenMP

Fabrizio Ceciliano Navarro
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
fabrizio.ceciliano.n@gmail.com

Alonso Jiménez Loaiza
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
aloji28@gmail.com

Abstract - In the present document, is going to be shown the conclusions obtained and the different tests that were done using the OpenMP API to parallelize a program written in C, which calculates Pi by iterations and was executed in the Cluster Kabré, which is property of the CeNAT (Centro Nacional de Alta Tecnología).

Keywords—cluster, openmp, arquitectura, algoritmos, paralelizar, computación paralela.

I. Introducción

La computación paralela es conocida como una técnica de la programación en la que muchas instrucciones o procesos se pueden ejecutar simultáneamente [1]. Se basa en el principio de que los problemas de mayor complejidad se pueden dividir en problemas menos complejos, los cuales a su vez se pueden dividir en partes más pequeñas que pueden resolverse de forma concurrente o en paralelo.

Por otra parte, el término cluster se refiere al conglomerado de computadoras unidos entre sí que se comportan como si fuesen una computadora.

Los clústeres han ido evolucionando hasta en el punto de convertirse en máquinas tan poderosas, que son capaces de mejorar el tiempo de ejecución de ciertos procesos que tardarían incluso minutos en una computadora simple. Esto es gracias a la computación paralela, la cual, basándose en diferentes leyes computacionales como la de Ley de Amdahl, Ley de Moore y Ley de Gustafson, ha hecho capaz la mejora del rendimiento en la ejecución de programas paralelizados, los cuales se dividen en pequeñas partes que se ejecutan al mismo tiempo en las diferentes partes de un clúster, conocidas como nodos.

II. Programa ejecutado

El programa que se seleccionó para hacer la presente investigación fue uno que se encargaba de calcular la constante matemática π , la cual se calculaba mediante una serie de pasos que se ejecutaban dentro de un ciclo for y se iban sumando en una variable. La cantidad de pasos seleccionada fue de 100 000 000 000, pues con una cantidad

menor la aproximación de π no era muy exacta y el programa duraba muy poco tiempo ejecutándose, lo cual no funcionaba para el propósito del proyecto.

III. Leyes

La Ley de Amdahl establece que la mejora en el rendimiento de un sistema debido a la alteración de uno de sus componentes está limitada por la cantidad de tiempo que se utiliza ese componente [2], dicho de otra forma, si se quiere mejorar el rendimiento de sistema, se debe mejorar el rendimiento de cada componente del sistema para poder tener una mejora considerable en el sistema completo, pues si se aumenta la capacidad de algún componente en específico que no se utilice mucho, el sistema no va a mejorar porque sus demás componentes no han aumentado su capacidad o potencia.

Por otra parte, la Ley de Moore establece que aproximadamente cada 2 años, se duplica el número de transistores en un microprocesador [3].

Finalmente, la Ley de Gustafson establece que cualquier problema suficientemente grande puede ser eficientemente paralelizado [4]. Esta ley se relaciona a Ley de Amdahl pero en un sentido opuesto, pues asegura que el tipo y la potencia de un procesador juega un papel muy importante en la paralelización de algún proceso y que no depende únicamente del componente paralizado [5].

IV. Relación entre las leyes y las pruebas de ejecución

Conforme lo obtenido en las pruebas de ejecución, se puede asegurar que, en cuanto a la Ley de Amdahl, se cumple en cierta parte y en cierta parte no. Se cumple en cierta parte porque, por ejemplo, al utilizar 2, 3, 4 y 5 nodos con 5 procesos por nodos se ve una mejora en el tiempo de ejecución conforme más nodos se tengan, por lo tanto, como se están utilizando más nodos para ejecutar el programa, se está teniendo una mejora en los diferentes procesos del programa y no en sólo uno, por lo tanto, al mejorar varias partes de la ejecución del programa, se obtiene una mejor ejecución en general. Por otra parte, si se utilizan 2, 3, 4 y 5

nodos con 1 proceso por nodo, el tiempo de ejecución del programa no varía de forma notable, incluso conforme más nodos se utilicen, el programa puede durar más. Esto es debido a que, al sólo utilizar 1 proceso por nodo, el programa se está optimizando en solo una parte y no en muchas, por lo tanto, la mejora del programa no es notoria a pesar de que se utilicen más nodos por ejecución.

Conforme a la Ley de Moore se puede asegurar que, dentro de aproximadamente 24 meses, el tiempo de ejecución que tomó el programa por ejemplo en 2 nodos con 5 procesos por nodo, el cual es 2 minutos, 44.641 segundos, se va a reducir a 1 minuto, 22.320 segundos, pues, al duplicar la cantidad de transistores en los microprocesadores, el tiempo que toma ejecutar el programa se reducirá a la mitad.

Y finalmente, en relación con la Ley de Gustafson se puede asegurar que es verdadera, pues, el procesador en el que se hicieron las pruebas fue un Xeon Phi KNL, el cual es extremadamente poderoso e incluso aceptaba realizar las pruebas con un número de iteraciones igual a 100 000 000 000, número que no fue aceptado por las computadoras de los integrantes del grupo por ser demasiado grande y que, el Xeon Phi ejecutó sin problema alguno.

V. Características de las computadoras de los integrantes del grupo.

Seguidamente, se presentarán dos tablas con la información respectiva de cada una de las computadoras de los autores del presente documento, esto para mostrar la capacidad que tiene cada máquina y poder compararla con la capacidad de los nodos del clúster. La computadora de Fabricio Ceciliano, posee las siguientes características [6]:

Tabla 1: Características de la computadora de Fabricio

Marca	Toshiba
Modelo	C55-B5296
CPU	Intel Celeron N2830 / 2.16 GHz
Número de núcleos	Dual-Core
Caché	1 MB
Tipo de Sistema	64 bits
Tecnología de memoria	DDR3L SDRAM
Tamaño de RAM	4 GB
Tipo de disco duro	HDD

Capacidad de disco duro	500 GB
-------------------------	--------

Por otra parte, la computadora de Alonso Jiménez posee las siguientes características [7]:

Tabla II: Características de la computadora de Alonso

Marca	Acer
Modelo	V5-472-6419
Procesador	Intel i3 3217U / 1.80 GHz
Número de núcleos	Dual-core
Caché	3 MB
Tipo de sistema	64 bits
Tecnología de memoria	DDR3 SDRAM
Tamaño de RAM	4 GB
Tipo de disco duro	HDD
Capacidad de disco duro	500 GB

VI. Información obtenida en las pruebas de ejecución

Para las pruebas de ejecución se realizaron varias ejecuciones de programa en el clúster, donde primeramente se ejecutó el programa en 1, 2, 3, 4 y 5 nodos, utilizando 1 proceso por nodo y se llegó a la Tabla III la cual contiene los promedios de haber ejecutado el programa 10 veces en cada nodo, los cuales se deducen de las Fig. del 1 al 5.

Tabla III. Datos promedio sobre el tiempo de ejecución del programa con diferente configuración de nodos, 1 proceso por nodo

Cantidad de nodos	Tiempo de ejecución promedio
1	2 minutos, 43.656 segundos
2	2 minutos, 43.523 segundos
3	2 minutos, 44.289 segundos
4	2 minutos, 42.983 segundos
5	2 minutos, 44.886 segundos

Fig. 1 Gráfico de tiempos de ejecución con 1 nodo, 1 proceso por nodo

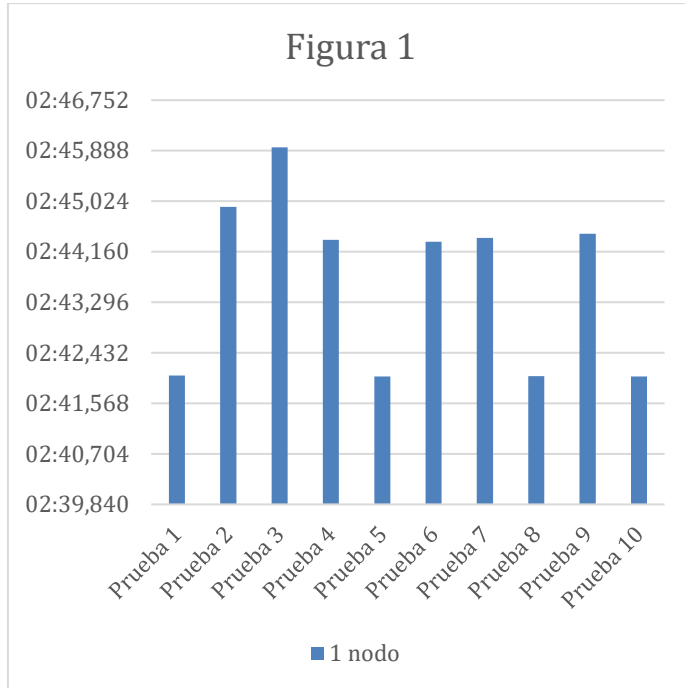


Fig. 3 Gráfico de tiempos de ejecución con 3 nodos, 1 proceso por nodo

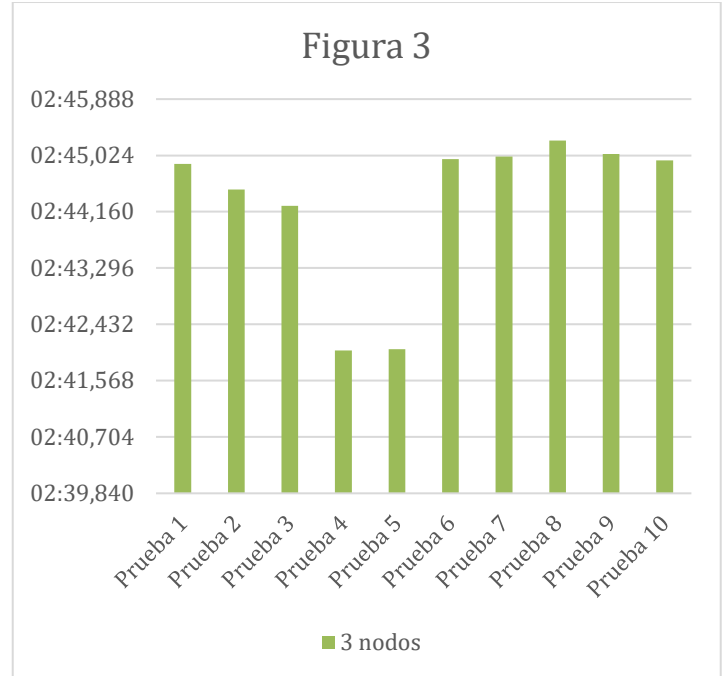


Fig. 2 Gráfico de tiempos de ejecución con 2 nodos, 1 proceso por nodo

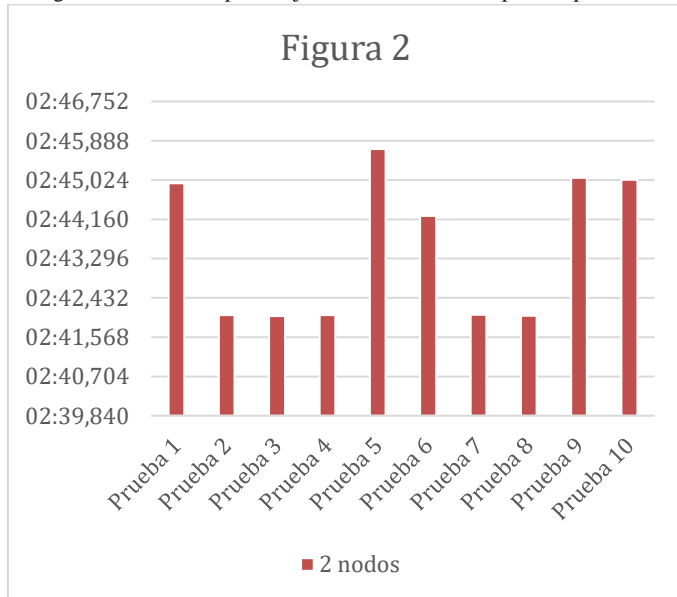


Fig. 4 Gráfico de tiempos de ejecución con 4 nodos, 1 proceso por nodo

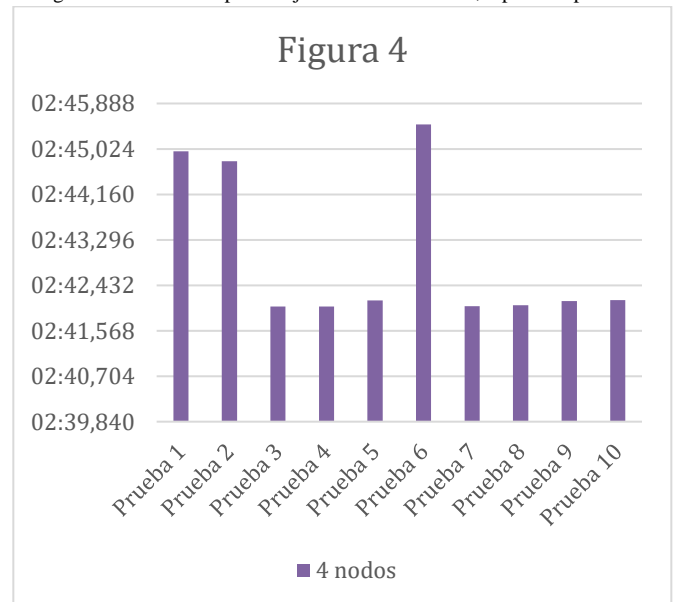
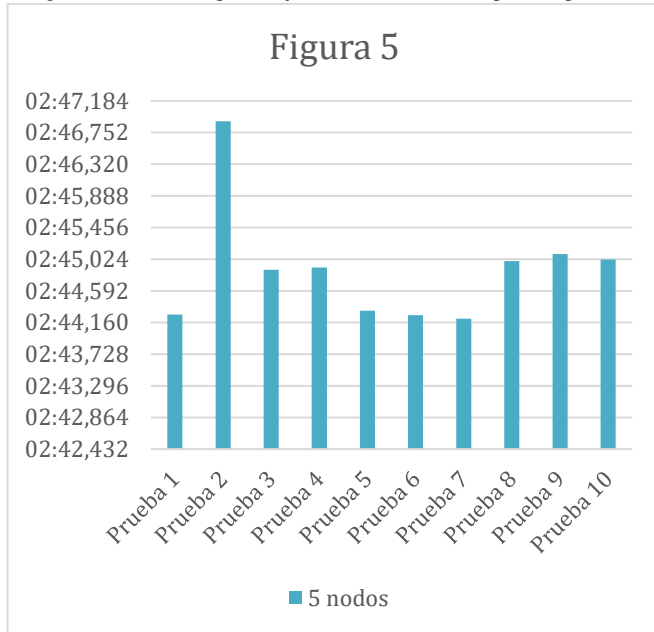
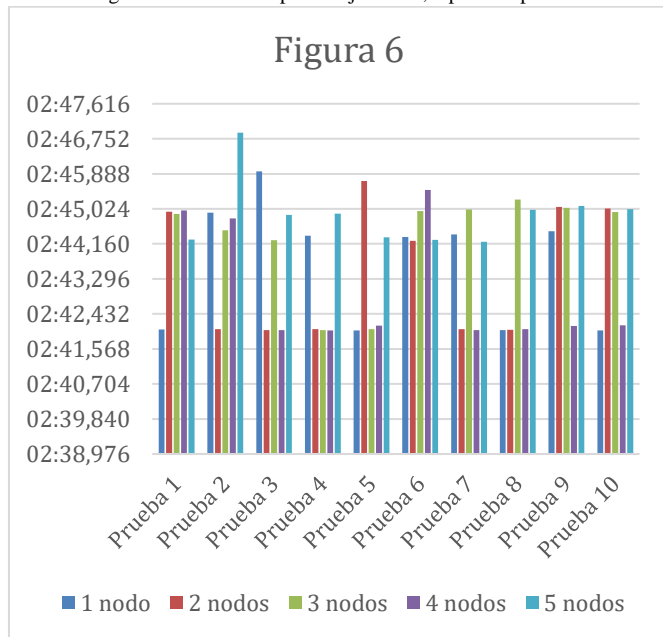


Fig. 5 Gráfico de tiempos de ejecución con 5 nodos, 1 proceso por nodo



Por otra parte, en la Fig. 6, se tienen los tiempos de ejecución de todas las pruebas en todos los nodos.

Fig. 6 Gráfico de tiempos de ejecución, 1 proceso por nodo



Por lo tanto, de las 6 figuras anteriores se puede deducir que, al ejecutar 1 proceso por nodo, la forma de obtener el mejor rendimiento del programa es utilizando 4 nodos, pues, aunque se pensaría que entre más nodos mejor resultado se obtendrá se observa que esto es falso y hay varios factores que pueden afectar como la duración del proceso, el uso que se le

esté dando a cada nodo en un instante dado, la cantidad de iteraciones que se utilicen, entre varias más.

Seguidamente, se hicieron pruebas con 1, 2, 3, 4 y 5 nodos, pero esta vez utilizando 5 procesos por nodo, y se llegó a la Tabla IV, la cual igualmente contiene los promedios de haber ejecutado el programa un total de 10 veces en cada configuración de nodos y se deduce de las Fig. del 7 al 11.

Tabla IV. Datos promedio sobre el tiempo de ejecución del programa con diferente configuración de nodos, 5 procesos por nodo

Cantidad de nodos	Tiempo de ejecución promedio
1	2 minutos, 44.910 segundos
2	2 minutos, 44.624 segundos
3	2 minutos, 44.199 segundos
4	2 minutos, 43.530 segundos
5	2 minutos, 43.520 segundos

Fig. 7 Gráfico de tiempos de ejecución con 1 nodo, 5 procesos por nodo

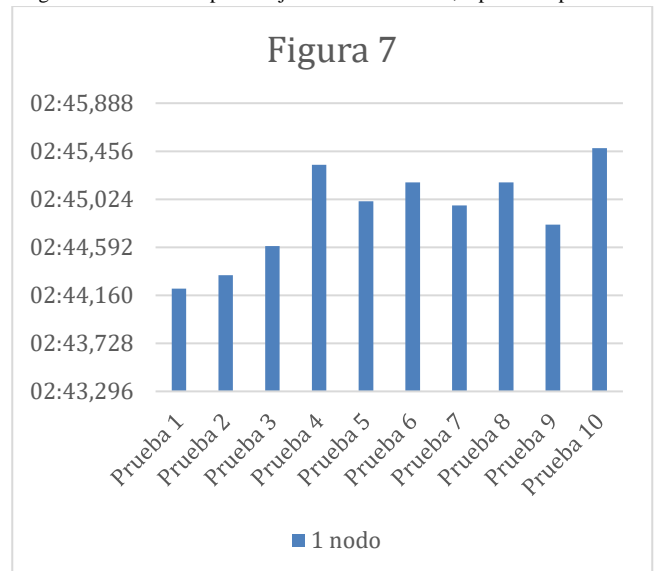


Fig. 8 Gráfico de tiempos de ejecución con 2 nodo, 5 procesos por nodo

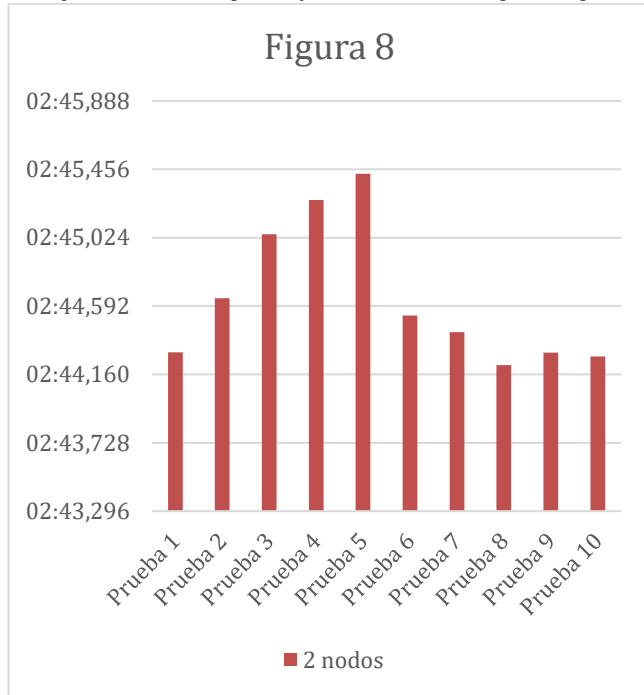


Fig. 9 Gráfico de tiempos de ejecución con 3 nodo, 5 procesos por nodo

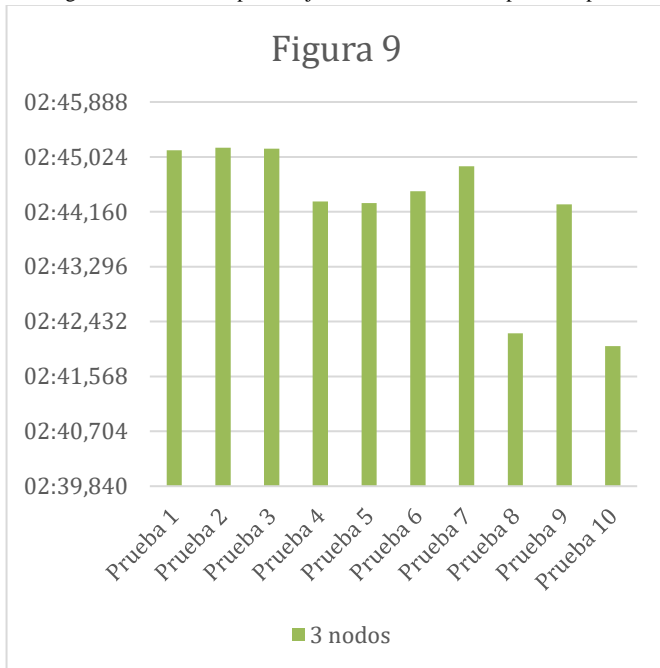


Fig. 10 Gráfico de tiempos de ejecución con 4 nodo, 5 procesos por nodo

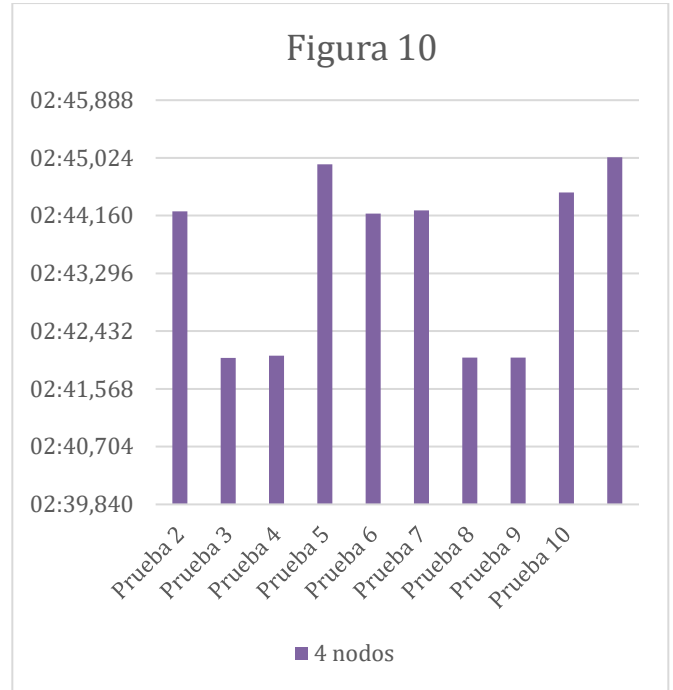
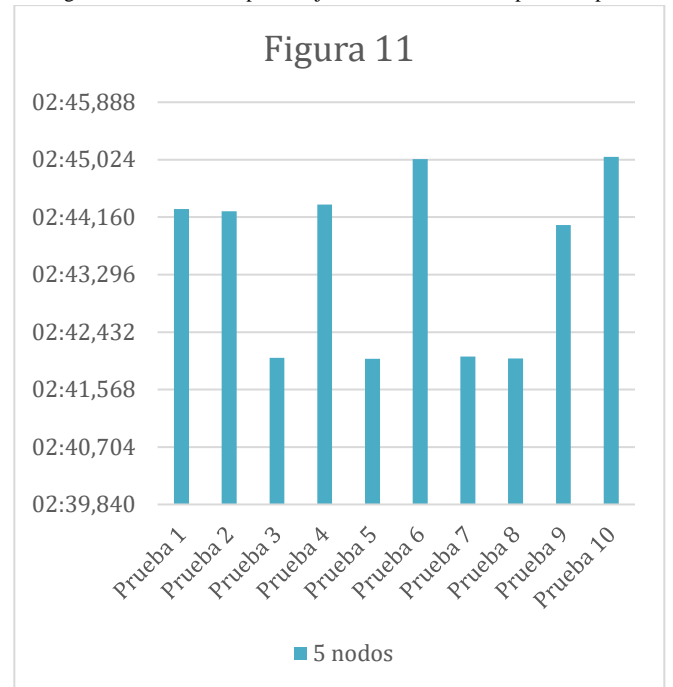
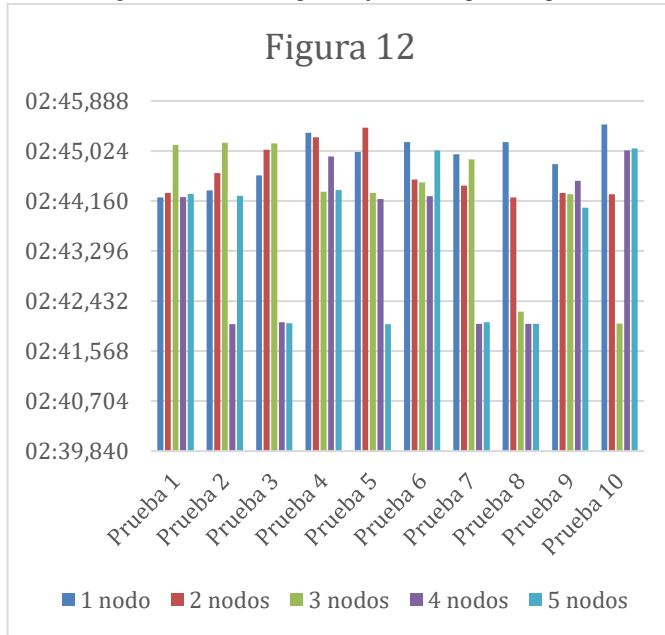


Fig. 11 Gráfico de tiempos de ejecución con 5 nodo, 5 procesos por nodo



Por otra parte, en la Fig. 12, se tienen los tiempos de ejecución de todas las pruebas en todos los nodos.

Fig. 12 Gráfico de tiempos de ejecución, 5 procesos por nodo



Por lo tanto, de las 6 figuras anteriores, se puede deducir que se cumple que conforme más procesadores se tengan para hacer una misma tarea, mejor será su eficiencia, pues se puede observar que los tiempos de ejecución se reducen conforme con más nodos se esté trabajando ya que se utilizan 5 procesos por nodo, a diferencia de si se usara solamente 1 proceso por nodo a como ya se mostró anteriormente. Esto es debido a que, al asignar más procesos por nodo, cada nodo va a ejecutar más tareas en la misma cantidad de tiempo, por ende, si se tiene más de 1 nodo, el proceso va a durar menos a que si se tuviera solamente uno, lo que da lugar a asegurar la Ley de Gustafson.

Es importante mencionar que el programa, en las computadoras de los integrantes del grupo, tomaba aproximadamente 3 minutos o más ejecutándose, lo cual ayuda a mostrar que efectivamente si un programa, aunque esté paralelizado, se ejecuta en una máquina que posea mucho hardware como el clúster, va a ser mucho más eficiente a si se ejecutara en una computadora común, esto debido a que una computadora común no posee la misma cantidad de cores, por lo tanto no puede llevar a cabo tantas tareas a la vez.

VII. Flops de las computadoras utilizadas

La computadora de Fabricio Ceciliano, ejecuta 1.826 GFlops. La computadora de Alonso Jiménez, ejecuta 10.5685 GFlops [8]. Y la configuración del cluster utilizada fue siempre la misma, la cual fue Intel Xeon Phi KNL, y ésta ejecuta 33.86 petaFlops [9].

IX. Conclusiones

Por lo tanto, se concluye que, al utilizar más cantidad de nodos para ejecutar un proceso, éste se va a ejecutar más rápido siempre y cuando se utilicen más procesos por nodo.

Si no se utiliza solamente un proceso por nodo, el programa no necesariamente va a ejecutarse más rápido, pues puede que tome más tiempo enviado la parte del proceso al nodo que ejecutándolo, lo cual aumenta los tiempos de ejecución.

Si se ejecuta el mismo programa en una computadora menos potente, va a durar más, pues las computadoras comunes no poseen la misma cantidad de nodos ni de procesadores que un clúster, por lo tanto, el tiempo de ejecución va a ser mayor.

Si se quieren utilizar más cantidad de threads, se debe tener el programa paralelizado en cada una de sus partes y lo más que se pueda, pues el uso de hilos requiere tiempo de ejecución y tiende a subir si el programa es muy pequeño.

Referencias

- [1] Tino. (2012, Jan 4). ¿Qué es la computación paralela?. [Online] Available : <https://conceptosarquitecturadecomputadoras.wordpress.com/computacion-paralela/>
- [2] R. Cotera. (2015, Feb 12). La ley de Amdahl. [Online] Available: <https://instintobinario.com/la-ley-de-amdahl/>
- [3] Wikipedia. (2017, Dec 1). Ley de Moore. [Online] Available: https://es.wikipedia.org/wiki/Ley_de_Moore
- [4] Wikipedia. (2016, Mar 3). Ley de Gustafson. [Online] Available: https://es.wikipedia.org/wiki/Ley_de_Gustafson
- [5] A. Romero. (2016, Feb 24). Ley de Gustafson. [Online] Available: <http://angelromeroes.blogspot.com/2016/02/ley-de-gustafson.html>
- [6] CNET (n.d.). Toshiba Satellite C55-B5296 - 15.6" - Celeron N2830 - 4GB RAM - 500 GB - HDD - US. [Online] Available: <https://www.cnet.com/es/analisis/toshiba-satellite-c55-b5296-15-6-celeron-n2830-4-gb-ram-500-gb-hdd-us/>
- [7] SNLookUp. (n.d.). Acer Aspire V5-472-6419 (NX.MB2AL.011). [Online] Available: <https://snlookup.com/acer-aspire-v3-472-notebook-nx-mb2al-011-p25184#ffs-tabbed-14>
- [8] F. Gennady, Z. Shaojuan. (2016, Nov 1). Intel Math Kernel Library Benchmarks (Intel MKL Benchmarks). [Online] Available: <https://software.intel.com/en-us/articles/intel-mkl-benchmarks-suite>
- [9] Wikipedia. (2018, Jun 8). Xeon Phi. [Online] Available: https://en.wikipedia.org/wiki/Xeon_Phi