

El enfoque del paralelismo en la programación del día a día

Abstract

In the following article, we are going to show how do we investigate and develop a program that could be parallelized and executed in the CeNAT Cluster, and also examine some important laws about parallel processing and the importance of processors.

Introducción

Para el desarrollo de este artículo se llevará a cabo una serie de pruebas con código previamente trabajado, con el cuál se desarrollará e implementará el paralelismo, y además se estudiarán una serie de leyes que nos ayudarán en cuanto a demostrar el beneficio de aprender a paralelizar y el desarrollo de código empleado con el mejor rendimiento que pueda dar la computadora. Para el desarrollo de este proyecto se utilizará también un el Cluster Kabré y en las pruebas del proyecto se mostrará el uso de este.

Especificaciones del computador de Paul Villafuerte:

Principal Memory	8192MBytes (x2)
Type Of Memory	DDR3
Number of cores	4 (max 4)
Number of threads	8 (max 8)
Name	Intel Core i7 6700HQ
Codename	Skylake

Specification Intel(R) Core(TM) i7-6700HQ
CPU @ 2.60GHz

Technology	14 nm
TDP Limit	45.0 Watts
Tjmax	100.0 °C
Core Speed	3091.7 MHz
Base frequency (cores)	99.7 MHz
Base frequency (ext.)	99.7 MHz
Stock frequency	2600 MHz
Max frequency	3500 MHz

L1 Data cache 4 x 32 KBytes, 8-way set
associative, 64-byte line size

L1 Instruction cache 4 x 32 KBytes, 8-way
set associative, 64-byte line size

L2 cache 4 x 256 KBytes, 4-way set
associative, 64-byte line size

L3 cache 6 MBytes, 12-way set
associative, 64-byte line size

Cache descriptor Level 1, D, 32 KB, 2
thread(s)

Cache descriptor Level 1, I, 32 KB, 2
thread(s)

Cache descriptor Level 2, U, 256 KB, 2
thread(s)

Cache descriptor Level 3, U, 6 MB, 16
thread(s)

Display adapter 0

Name Intel(R) HD Graphics 530
Core clock 448.8 MHz

Display adapter 1

Name NVIDIA GeForce GTX 960M
Technology 28 nm

Memory size 4 GB

Memory type GDDR5

Drive 0

Name SanDisk Z400s M.2 2280 128GB

Capacity 119.2 GB

Type SSD

Bus Type SATA (11)

Features SMART, TRIM

Volume c:\, 117.9 GBytes (24.1 percent available)

Drive 1

Name WDC WD10JPVX-75JC3T0

Capacity 931.5 GB

Bus Type SATA (11)

Rotation speed 5400 RPM

Features SMART

Volume d:\, 931.5 GBytes (51.4 percent available)

Pixel Rate: 18.82 GPixel/s

Texture Rate: 47.04 GTexel/s

FP32 (float) performance: 1,505 GFLOPS

FP64 (double) performance: 47.04 GFLOPS (1:32)

Especificaciones del computador de Daniel Camacho:

Principal Memory 8192MBytes

Type Of Memory DDR3

Number of cores 4 (max 4)

Number of threads 4 (max 4)

Name Intel Core i5 3470

Codename Ivy Bridge

Specification Intel(R) Core(TM) i5-3470 CPU @ 3.20GHz

Technology 22 nm

TDP Limit 77.0 Watts

Tjmax 105.0 °C

Core Speed 1596.2 MHz

Multiplier x Bus Speed 16.0 x 99.8 MHz

Base frequency (cores) 99.8 MHz

Base frequency (ext.) 99.8 MHz

Stock frequency 3200 MHz

L1 Data cache 4 x 32 KBytes, 8-way set associative, 64-byte line size

L1 Instruction cache 4 x 32 KBytes, 8-way set associative, 64-byte line size

L2 cache 4 x 256 KBytes, 8-way set associative, 64-byte line size

L3 cache 6 MBytes, 12-way set associative, 64-byte line size

Cache descriptor Level 1, D, 32 KB, 2 thread(s)

Cache descriptor Level 1, I, 32 KB, 2 thread(s)

Cache descriptor Level 2, U, 256 KB, 2 thread(s)

Cache descriptor Level 3, U, 6 MB, 16 thread(s)

Display adapter 0

Name NVIDIA GeForce GTX 960

Board Manufacturer EVGA Corp.

Technology 28 nm

Memory size 2 GB

Memory type GDDR5

Core clock 135.0 MHz

Memory clock 405.0 MHz

Drive 0

Name TOSHIBA DT01ACA050

Revision MS10A750

Capacity 465.8 GB

Bus Type SATA (11)

Rotation speed 7200 RPM

Volume c:\, 465.2 GBytes (65.2 percent available)

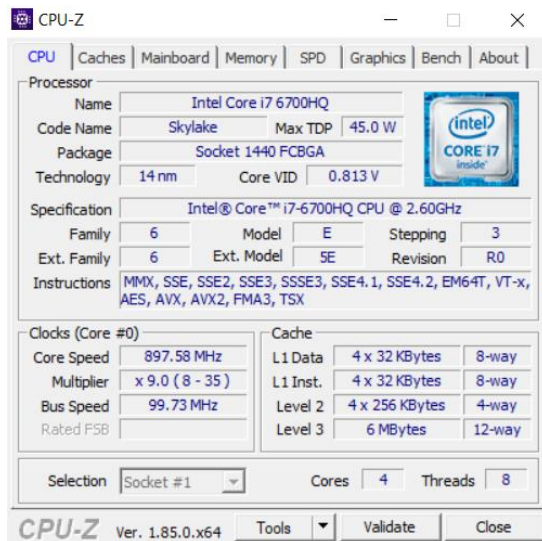
Pixel Rate: 37.70 GPixel/s

Texture Rate: 75.39 GTexel/s

FP32 (float) performance: 2,413 GFLOPS

FP64 (double) performance: 75.39 GFLOPS (1:32)

Programa para determinar aspectos de las computadoras



Tablas de Tiempos de Ejecución.

	2 Nodos	3 Nodos	4 Nodos	5 Nodos
Prueba 1	1:42	1:27	1:09	1:01
Prueba 2	1:40	1:35	1:11	1:01
Prueba 3	1:40	1:32	1:23	0:57
Prueba 4	1:44	1:32	1:19	0:59
Prueba 5	1:43	1:29	1:23	0:51
Prueba 6	1:39	1:30	1:16	1:03
Prueba 7	1:41	1:42	1:12	0:52
Prueba 8	1:45	1:34	1:13	0:53
Prueba 9	1:46	1:28	1:13	1:00
Prueba10	1:38	1:30	1:11	0:59
Total	16:58	15:21	12:30	7:36
Promedio	1:42	1:32	1:15	0:58

Tabla 1. Tiempos de Ejecución Cluster

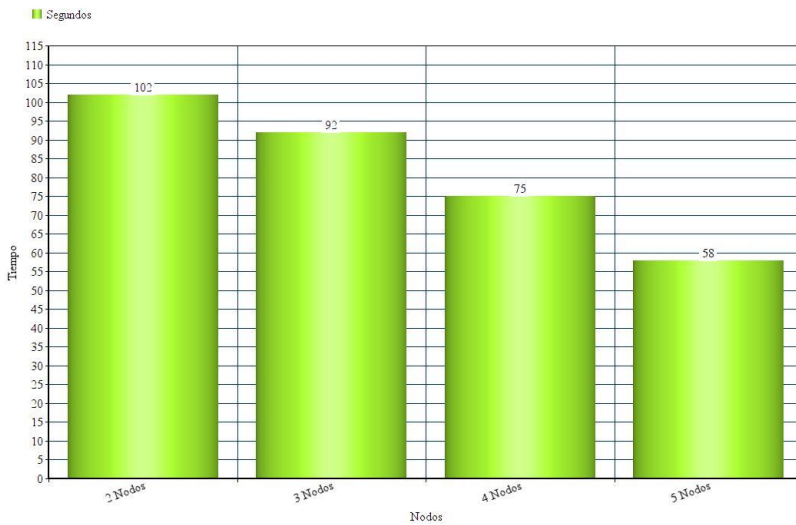
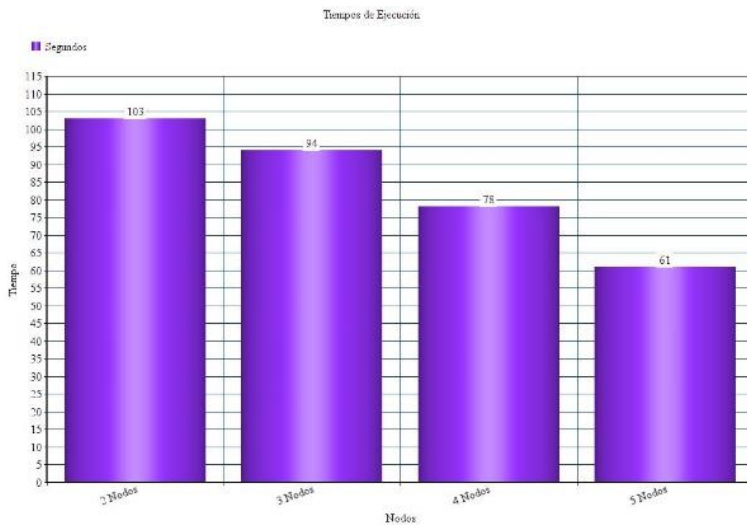
Usuario: Curso-098, Paul Villafuerte Beita.

	2 Nodos	3 Nodos	4 Nodos	5 Nodos
Prueba 1	1:45	1:27	1:09	1:02
Prueba 2	1:37	1:40	1:11	1:05
Prueba 3	1:50	1:32	1:23	0:57
Prueba 4	1:47	1:32	1:19	0:59
Prueba 5	1:45	1:29	1:23	0:46
Prueba 6	1:39	1:38	1:16	1:08
Prueba 7	1:41	1:42	1:33	1:02
Prueba 8	1:47	1:34	1:17	0:53
Prueba 9	1:40	1:29	1:13	1:14
Prueba 10	1:45	1:33	1:11	0:55
Promedio	1:43	1:34	1:18	1:01

Tabla2. Tiempos de Ejecución Cluster

Usuario: Curso-099, Jose Daniel Camacho.

Gráficos de Tiempo de Ejecución



Ley de Gustafson

Esta ley, también conocida como la Ley de Gustafson-Barsis, es una ley de ciencias de la computación la cual establece que cualquier problema suficientemente grande puede ser de la forma más eficiente paralelizado, esta está muy ligada a la Ley de Amhdal, la cual será seguidamente explicada también en el artículo.

El enfoque de esta ley es dar a entender que se puede obtener un beneficio y mejora de rendimiento considerable gracias a la paralelización.

Comparada a la Ley de Amdahl, este ofrece otro punto de vista, ya que posee una visión positiva del procesamiento en paralelo.

$$S(P) = P - \alpha \cdot (P - 1)$$

P: Número de procesadores.

S: Speedup.

a: Parte no paralelizable.

Como se puede ver en lo anterior esa es la fórmula principal de Gustafson, y lo que permite que esta ley aborde las limitaciones que la Ley de Amdahl, las cuales se dan debido a que no escala la disponibilidad del poder de las computadoras a medida que el número de estas aumentan.

La idea principal de esta ley es que un aumento cuádruple del poder de paralelización conllevaría a un incremento similar en las capacidades del sistema.

Ley de Amdahl

$$T_m = T_a \cdot \left((1 - F_m) + \frac{F_m}{A_m} \right)$$

Fm = fracción de tiempo que el sistema utiliza el subsistema mejorado.

Am = factor de mejora que se ha introducido en el subsistema mejorado.

Ta = tiempo de ejecución antiguo.

Tm = tiempo de ejecución mejorado

Puede ser sobrescrita por la definición del incremento de velocidad $A = T_a/T_m$

como:

$$A = \frac{1}{(1 - F_m) + \frac{F_m}{A_m}}$$

siendo:

A: La aceleración o ganancia en velocidad conseguida en el sistema completo debido a la mejora de uno de sus subsistemas.

A_m: El factor de mejora que se ha introducido en el subsistema mejorado.

F_m: La fracción de tiempo que el sistema utiliza el subsistema mejorado.

Analizando la fórmula de Amdahl al tener una fracción como lo es la segunda fórmula lo que se busca es obtener un porcentaje de mejora y así poder calcular un aproximado del tiempo de ejecución esperado al hacer dicha mejora, la limitante que parece tener dicha fórmula es que no toma en cuenta la cantidad de procesos en los que se va a dividir para paralelizar, lo anterior se puede tomar como el factor de mejora pero no hay una relación tan directa como tomar la cantidad de procesos en los que se va a dividir dicho programa como lo hace la ley de Gustafson.

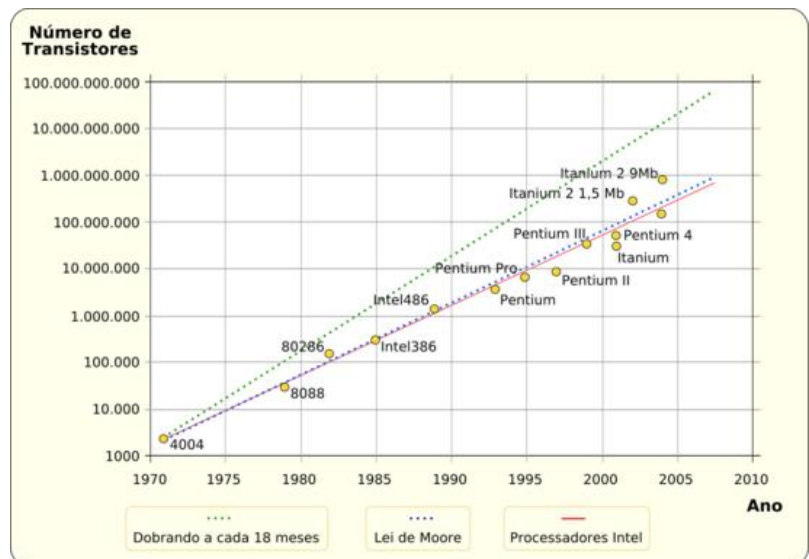
Ley de Moore

Esta ley expresa que aproximadamente cada 2 años se duplica el número de transistores en un microprocesador, y a pesar de que la ley originalmente fue formulada para establecer que la duplicación se realizaría cada año, Moore tuvo que redefinir su ley y ampliar el periodo de esta a 2 años.

Esta es una ley empírica, formulada por el cofundador de Intel Gordon Moore, esta la

desarrolló el 19 de Abril de 1965 y la cual su cumplimiento se ha podido constatar hasta hoy.

En general se podría decir que la Ley de Moore es una forma en la que Moore anticipaba que la complejidad de los circuitos integrados se duplicaría cada año con una reducción de costo en ellos. Esta ley se ha convertido en el motor del rápido cambio tecnológico que ha surgido hasta ahora.



Conclusiones

Finalmente podemos determinar que en cuanto a lo aprendido con respecto a las leyes y los casos de prueba y error que hemos tenido durante el desarrollo de este proyecto, el paralelismo es una forma importante de la mejora en cuanto al rendimiento y reducción de tiempo para los programas y todos los casos que se ocupen desarrollar. Además, que es una muy buena práctica el adquirir esta clase de conocimiento para aprender a desarrollar no solo para que lo pedido funcione, sino que lo haga de la mejor forma y podamos utilizar todas las herramientas que tenemos a nuestro alcance para lograrlo.

REFERENCES

Wikipedia. (03 de Marzo de 2016).

Wikipedia. Obtenido de Wikipedia:
https://es.wikipedia.org/wiki/Ley_de_Gustafson

Wikipedia. (01 de Diciembre de 2017).

Wikipedia. Obtenido de Wikipedia:
https://es.wikipedia.org/wiki/Ley_de_Moore

Wikipedia. (18 de Febrero de 2018).

Wikipedia. Obtenido de Wikipedia:
https://es.wikipedia.org/wiki/Ley_de_Amdahl