

PRIMER PARCIAL HPC (HIGH PERFORMANCE COMPUTING)

Ingeniería De Sistemas y Computación

Luis Fernando Salazar Escamilla

1088309888

Universidad Tecnológica de Pereira

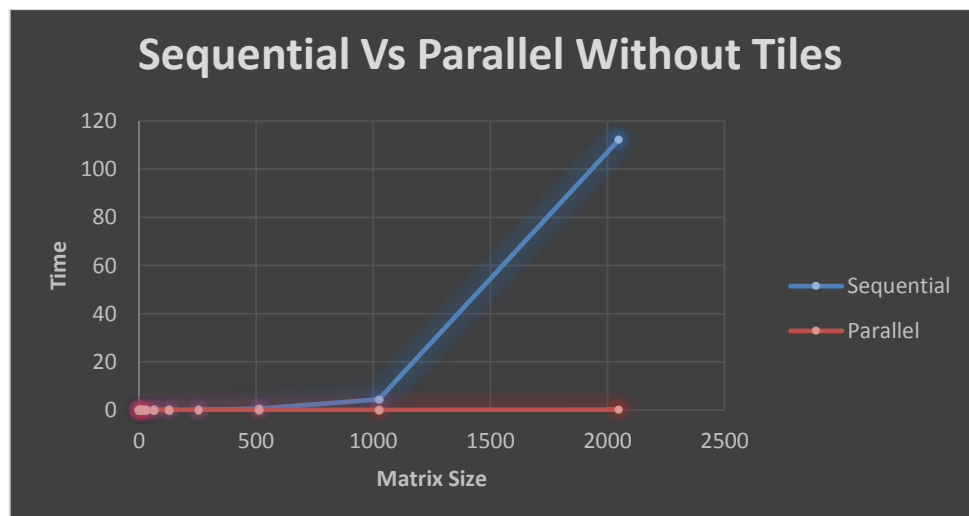
Marzo 2015

1. MATRICES CUADRADAS CON DATOS TIPO ENTERO

Tabla 1.1: Datos algoritmo secuencial sin tiles

| WITHOUT TILES | | | |
|---------------|------------|----------|-------------|
| Matrix Size | Sequential | Parallel | Aceleración |
| 2 | 0 | 0,000066 | 0 |
| 4 | 0,000001 | 0,000068 | 0,014705882 |
| 8 | 0,000002 | 0,000066 | 0,03030303 |
| 16 | 0,000027 | 0,000073 | 0,369863014 |
| 32 | 0,000092 | 0,000081 | 1,135802469 |
| 64 | 0,000698 | 0,00103 | 0,677669903 |
| 128 | 0,012067 | 0,000215 | 56,1255814 |
| 256 | 0,060314 | 0,000705 | 85,55177305 |
| 512 | 0,495802 | 0,004244 | 116,8242224 |
| 1024 | 4,406558 | 0,025185 | 174,9675601 |
| 2048 | 112,244747 | 0,188377 | 595,8516539 |

De los datos anteriores se obtiene la siguiente gráfica donde se compara el tiempo de ejecución de un algoritmo de multiplicación de matrices secuencial y paralelo.



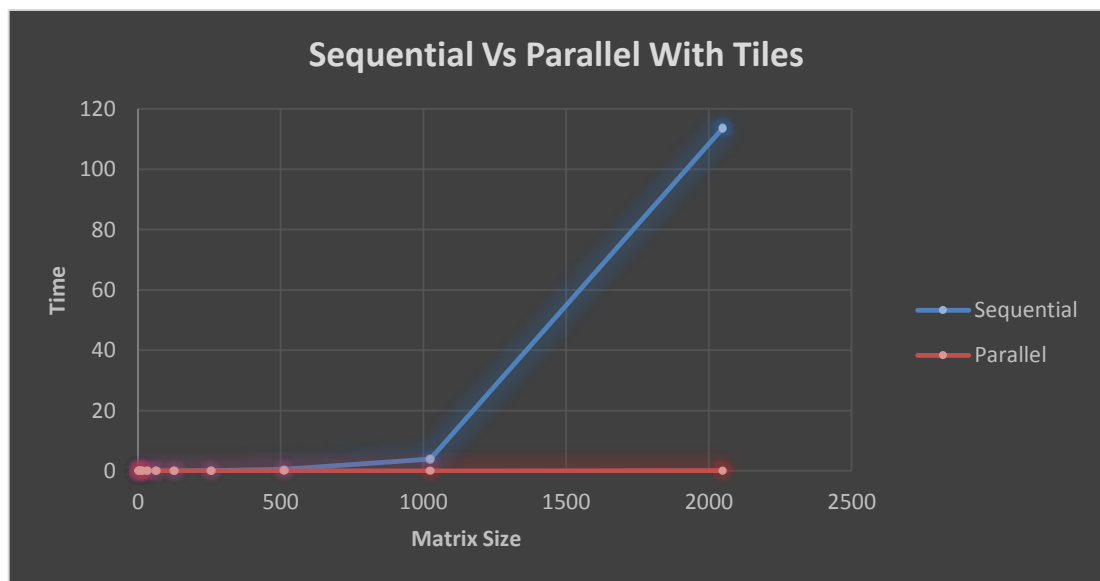
Gráfica 1: Secuencial vs Paralelo

Como se puede observar en la tabla 1.2, se pueden observar los datos que se obtuvieron en la ejecución del algoritmo de multiplicación de matrices en secuencial y el algoritmo en paralelo con el uso de tiles.

Tabla 1.2: Datos algoritmos secuencial con tiles

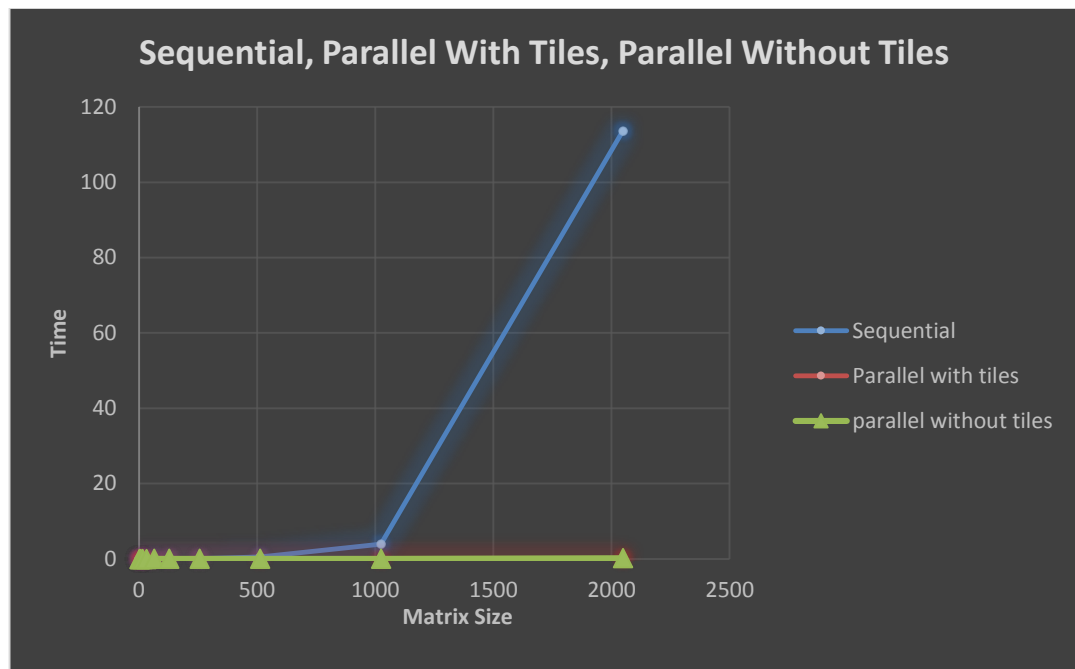
| WITHOUT TILES | | | |
|---------------|------------|----------|-------------|
| Matrix Size | Sequential | Parallel | Aceleración |
| 2 | 0 | 0,000066 | 0 |
| 4 | 0,000001 | 0,000068 | 0,014705882 |
| 8 | 0,000002 | 0,000066 | 0,03030303 |
| 16 | 0,000027 | 0,000073 | 0,369863014 |
| 32 | 0,000092 | 0,000081 | 1,135802469 |
| 64 | 0,000698 | 0,00103 | 0,677669903 |
| 128 | 0,012067 | 0,000215 | 56,1255814 |
| 256 | 0,060314 | 0,000705 | 85,55177305 |
| 512 | 0,495802 | 0,004244 | 116,8242224 |
| 1024 | 4,406558 | 0,025185 | 174,9675601 |
| 2048 | 112,244747 | 0,188377 | 595,8516539 |

A partir de los datos anteriores se realizó la siguiente gráfica.



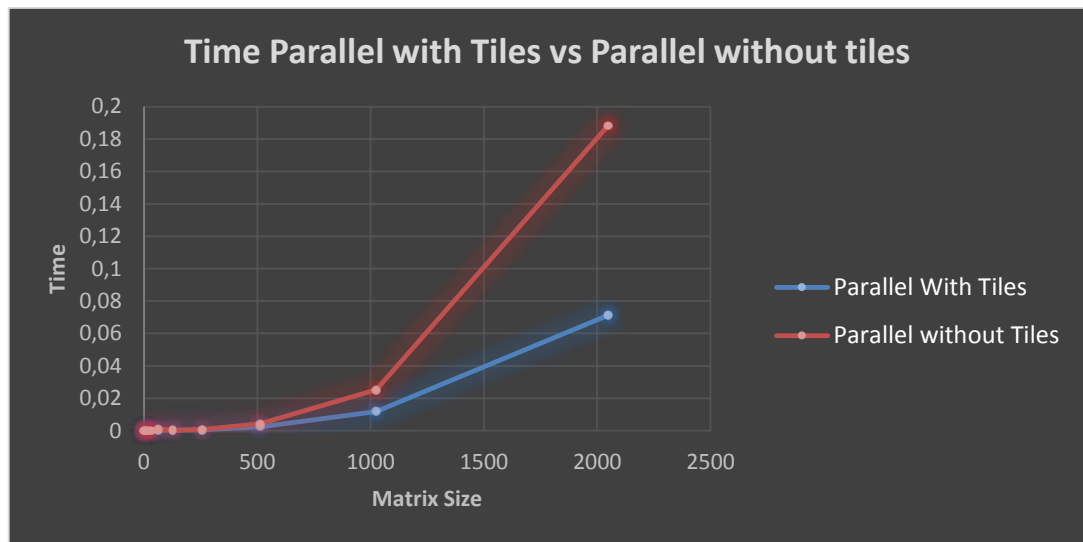
Gráfica 2: Secuencial Vs Paralelo con tiles

Después de tener gráficas del tiempo de ejecución del algoritmo secuencial, paralelo con tiles y paralelo sin tiles, en la gráfica que viene a continuación se presenta una comparación de las tres:



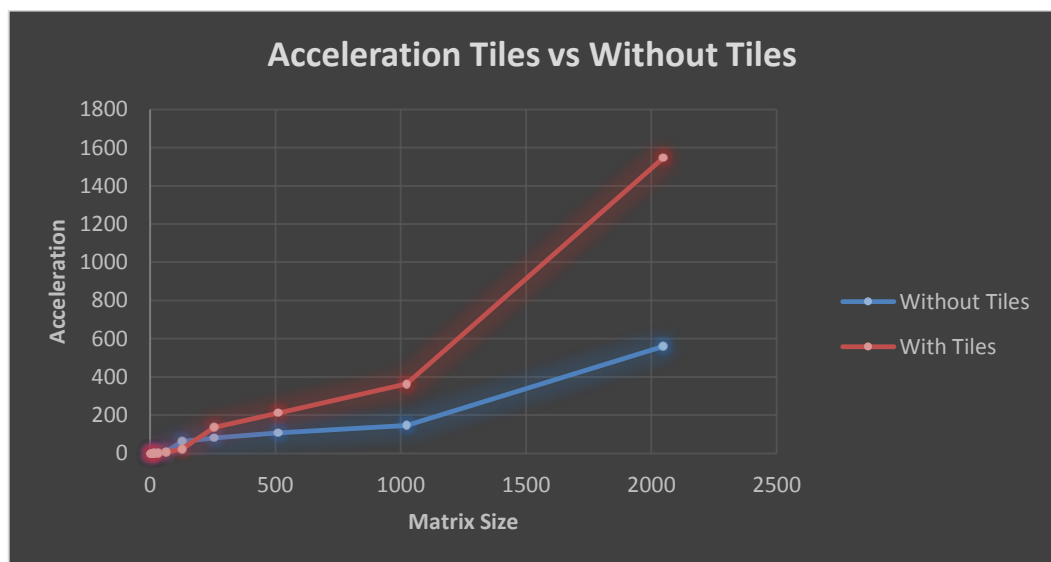
Gráfica 3: Secuencial / Paralelo con tiles / Paralelo sin tiles

En comparación con el algoritmo secuencial, los algoritmos en paralelo (con tiles y sin tiles) tienen mejores resultados en tiempo. Para poder diferenciar un poco los tiempos de los mismos a continuación se presenta una gráfica donde se muestre la diferencia que hay entre ellos, puesto que en la gráfica anterior esta diferencia no es notable debido a que el tiempo del algoritmo secuencial es muy alto.



Gráfica 4: Paralelo sin tiles Vs Paralelo con tiles

Basándose en los tiempos que se tienen de las diferentes formas de ejecutar el algoritmo. Se puede decir que existe una aceleración del algoritmo en paralelo sin tiles en cuanto al secuencial, y así mismo ocurre con el algoritmo con tiles. A continuación se presenta una gráfica que compara las dos aceleraciones ya mencionadas.



Gráfica 5: Aceleración con tiles vs sin tiles

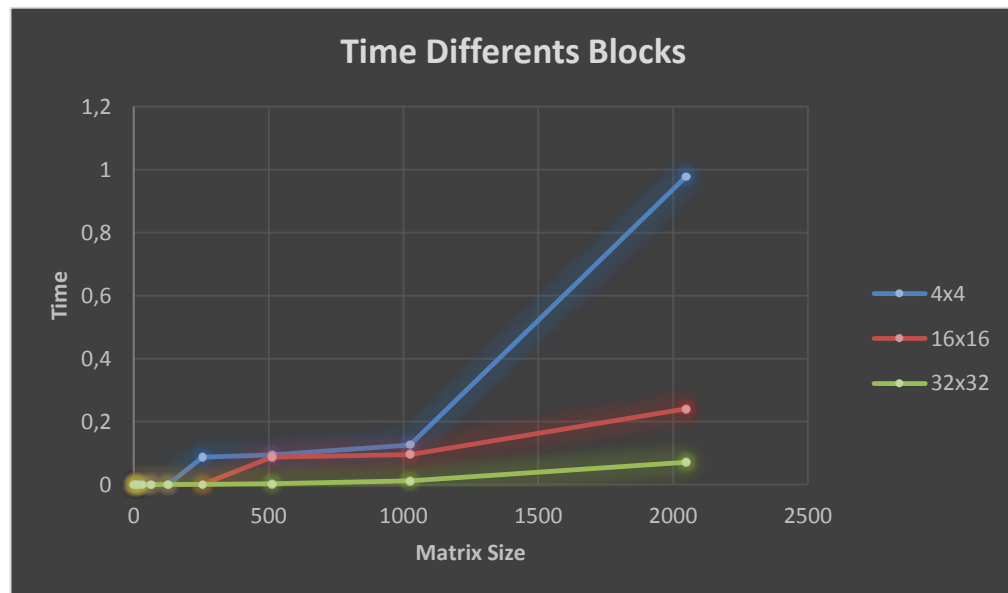
Diferente Tamaño de Bloques

Para el mismo desarrollo que se hace en el apartado anterior se tienen en cuenta diferentes tamaño de bloques para el análisis: **4x4**, **16x16**, **32x32**.

Tabla 1.3: Datos Diferentes tamaños de bloques

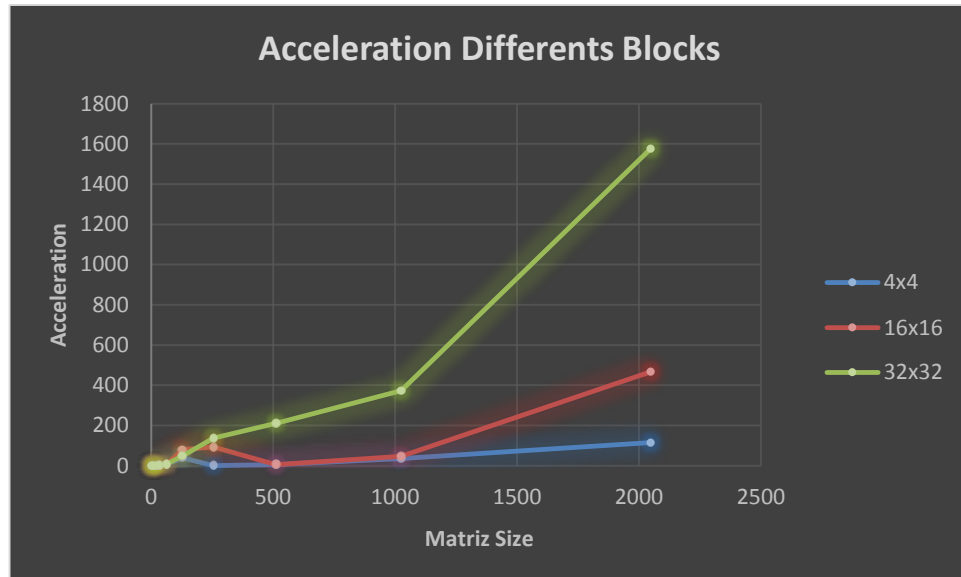
| CHANGE BLOCK SIZES | | | |
|--------------------|----------|----------|----------|
| Matrix Size | 4x4 | 16x16 | 32x32 |
| 2 | 0,000065 | 0,000066 | 0,000067 |
| 4 | 0,000065 | 0,000065 | 0,000112 |
| 8 | 0,000066 | 0,000084 | 0,000109 |
| 16 | 0,000088 | 0,000065 | 0,000066 |
| 32 | 0,000073 | 0,000068 | 0,000071 |
| 64 | 0,000102 | 0,000112 | 0,000084 |
| 128 | 0,000309 | 0,000152 | 0,00025 |
| 256 | 0,087051 | 0,000652 | 0,00044 |
| 512 | 0,094096 | 0,086775 | 0,002349 |
| 1024 | 0,126045 | 0,095365 | 0,011804 |
| 2048 | 0,977454 | 0,240271 | 0,071214 |

De los datos de la tabla anterior se obtuvo la siguiente gráfica:



Gráfica 7: Tiempo de ejecución con cada tamaño de bloques

Con cada tamaño diferente de bloques se obtiene una aceleración diferente, ahora se muestra la aceleración para cada uno de ellos:



Gráfica 8: Aceleración con diferente tamaño de bloques

Como se puede observar en la gráfica 8 la aceleración que tiene el algoritmo en paralelo con tiles con un tamaño de bloques de 32x32 con respecto al algoritmo secuencial es el más óptimo, el que tiene un mejor desempeño en la solución de este problema.

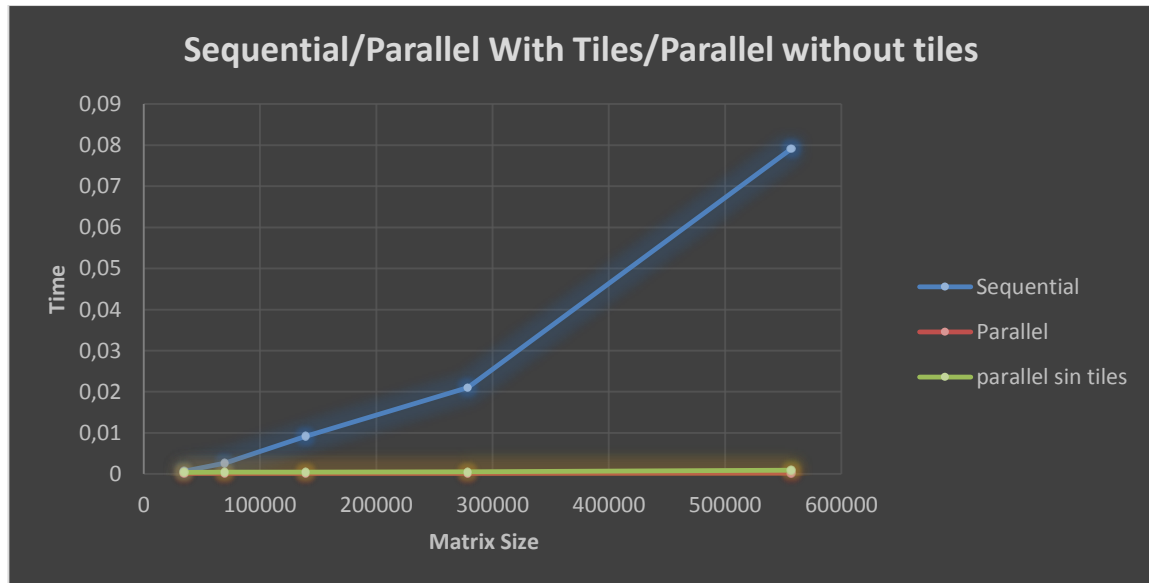
2. MATRICES NO CUADRADAS CON DATOS TIPO ENTEROS

Se llevó un proceso de análisis de los algoritmos de multiplicación de matrices secuencial, paralelo y paralelo con tiles (optimizado), todo esto con diferentes tamaños de bloques. A continuación se presenta la tabla de datos obtenidos para una cantidad de **32x32** bloques:

Tabla 2.1: Datos recolectado bloques 32x32 matrices no cuadradas

| INFORMATION | | | | | | | | |
|-------------|------------------|----------|--------|------------|---------------------|-------------------------|------------------------|----------------------------|
| Row A | Column A & Row B | Column B | Data | Sequential | Parallel with tiles | Acceleration with tiles | Parallel without tiles | Acceleration without tiles |
| 2 | 1024 | 32 | 34816 | 0,000657 | 0,000077 | 8,532467532 | 0,00042 | 1,564285714 |
| 4 | 1024 | 64 | 69632 | 0,002648 | 0,000093 | 28,47311828 | 0,000421 | 6,289786223 |
| 8 | 1024 | 128 | 139264 | 0,009169 | 0,000113 | 81,14159292 | 0,000454 | 20,19603524 |
| 16 | 1024 | 256 | 278528 | 0,021046 | 0,000186 | 113,1505376 | 0,000879 | 23,94311718 |
| 32 | 1024 | 512 | 557056 | 0,079148 | 0,000583 | 135,7598628 | 0,002553 | 31,00195848 |

De la tabla anterior se obtuvieron las siguientes gráficas donde se relaciona la cantidad de datos que hay con el tiempo de cada uno de los algoritmos.



Gráfica 9: Tiempo para algoritmo secuencial, paralelo con tiles y paralelo sin tiles

Como ocurría en el apartado anterior al establecer estos tres tiempos en una sola gráfica, la diferencia que hay entre el algoritmo con tile y el algoritmo sin tile no se puede ver, por esto se realiza la siguiente gráfica donde la diferencia entre estos algoritmos se pueda percibir.

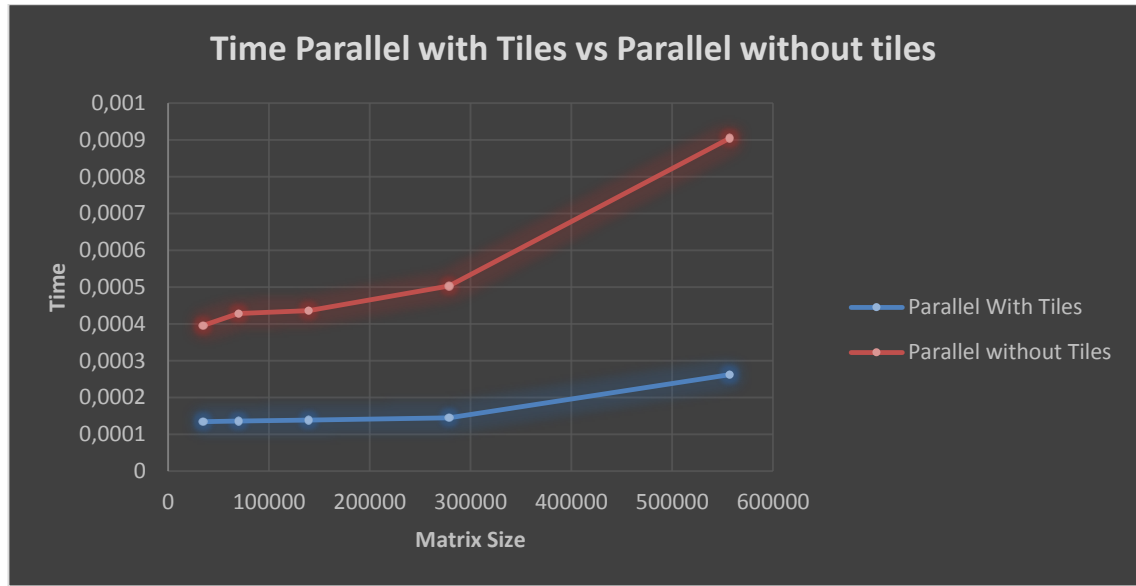
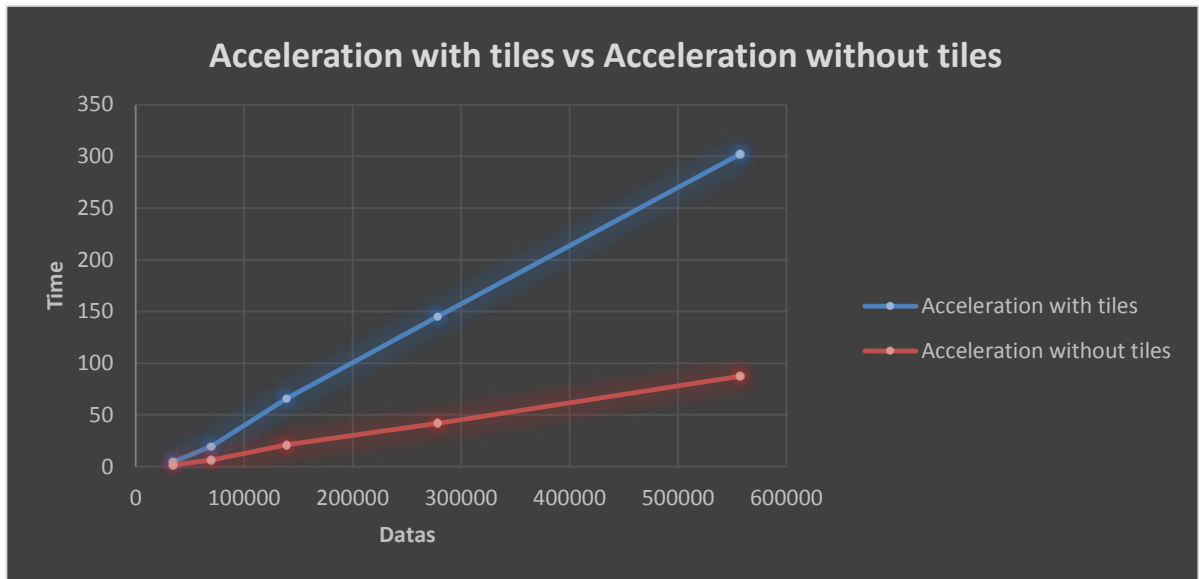


Gráfico 10: Tiempo para algoritmo con tiles y sin tiles.

Ahora se procede a representar la aceleración que tiene los algoritmos en paralelo con respecto al algoritmo secuencial. Se podrá observar en la gráfica 11 que la aceleración que presenta un algoritmo en paralelo con tiles es mejor que la que tiene el mismo sin tiles. El desempeño es mejor y es más eficaz para solucionar este tipo de problemas.



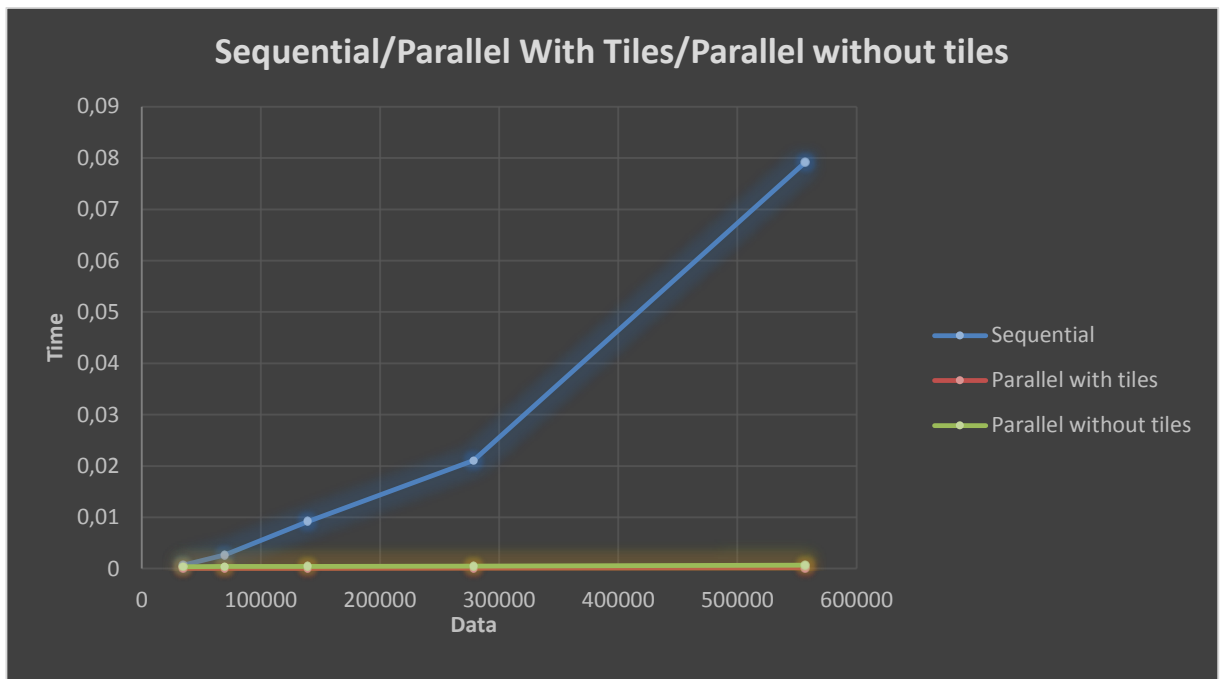
Gráfica 11: Aceleración del algoritmo en paralelo con tiles y sin tiles con respecto al algoritmo secuencial

Ahora se pasa analizar lo que sucede cuando se cambia el tamaño de los bloques a **16x16**, la siguiente tabla representa los datos que se obtuvieron al ejecutar los algoritmos con esta cantidad de bloques:

Tabla 2.2: Datos recolectado bloques 16x16 matrices no cuadradas

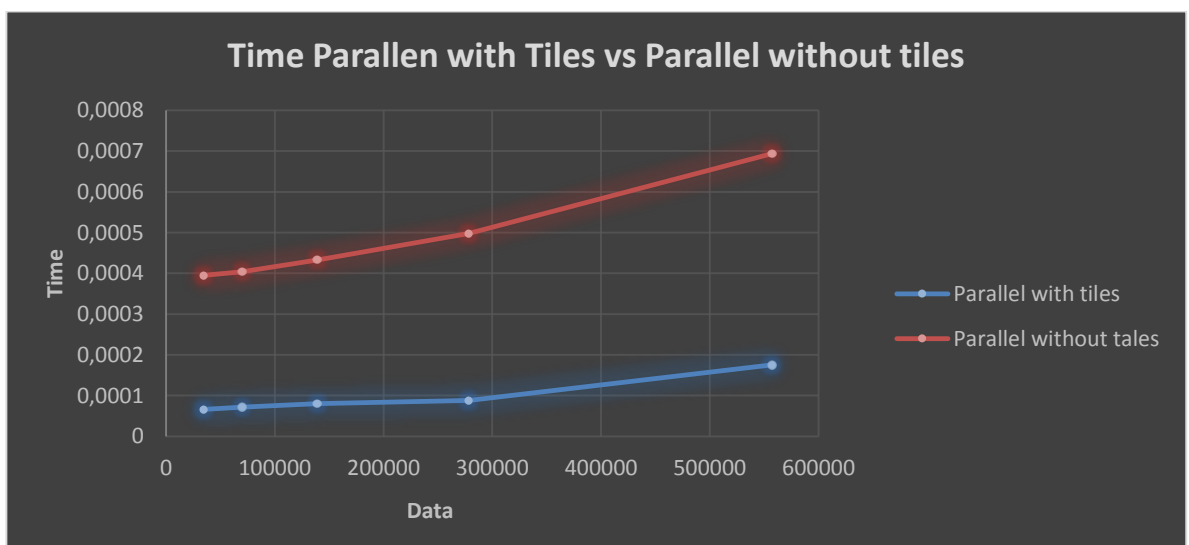
| INFORMATION | | | | | | | | |
|-------------|------------------|----------|--------|------------|---------------------|-------------------------|------------------------|----------------------------|
| Row A | Column A & Row B | Column B | Data | Sequential | Parallel with tiles | Acceleration with tiles | Parallel without tiles | Acceleration without tiles |
| 2 | 1024 | 32 | 34816 | 0,000657 | 0,000066 | 9,954545455 | 0,000395 | 1,663291139 |
| 4 | 1024 | 64 | 69632 | 0,002648 | 0,000072 | 36,77777778 | 0,000404 | 6,554455446 |
| 8 | 1024 | 128 | 139264 | 0,009169 | 0,00008 | 114,6125 | 0,000433 | 21,17551963 |
| 16 | 1024 | 256 | 278528 | 0,021046 | 0,000088 | 239,1590909 | 0,000498 | 42,26104418 |
| 32 | 1024 | 512 | 557056 | 0,079148 | 0,000175 | 452,2742857 | 0,000694 | 114,0461095 |

Las siguientes gráficas se basan en la tabla anterior para representar el tiempo de ejecución de los algoritmos de multiplicación de matrices:



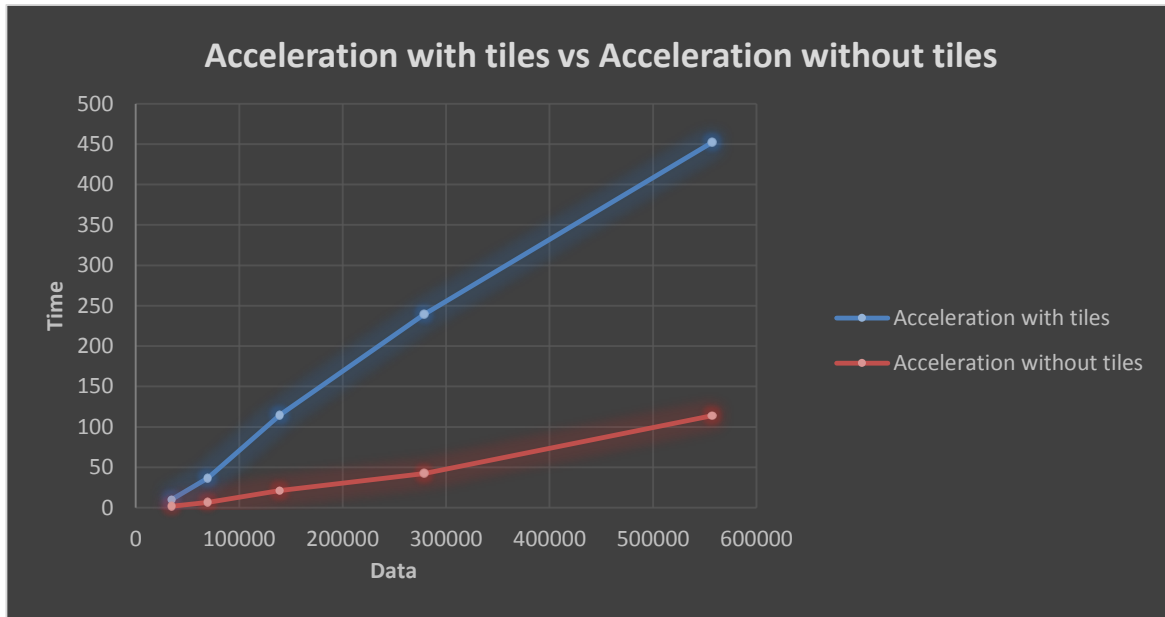
Gráfica 12: Tiempo del algoritmo en paralelo con tiles y sin tiles, y el algoritmo secuencial

Para poder observar mejor el comportamiento de los algoritmos en paralelo se realiza una gráfica en contraste de los mismos:



Gráfica 13: Tiempo del algoritmo en paralelo con tiles y sin tiles con respecto al algoritmo secuencial

Teniendo en cuenta los tiempos de ejecución de los algoritmos se representa la aceleración que tienen los paralelos frente al secuencial en la siguiente gráfica:



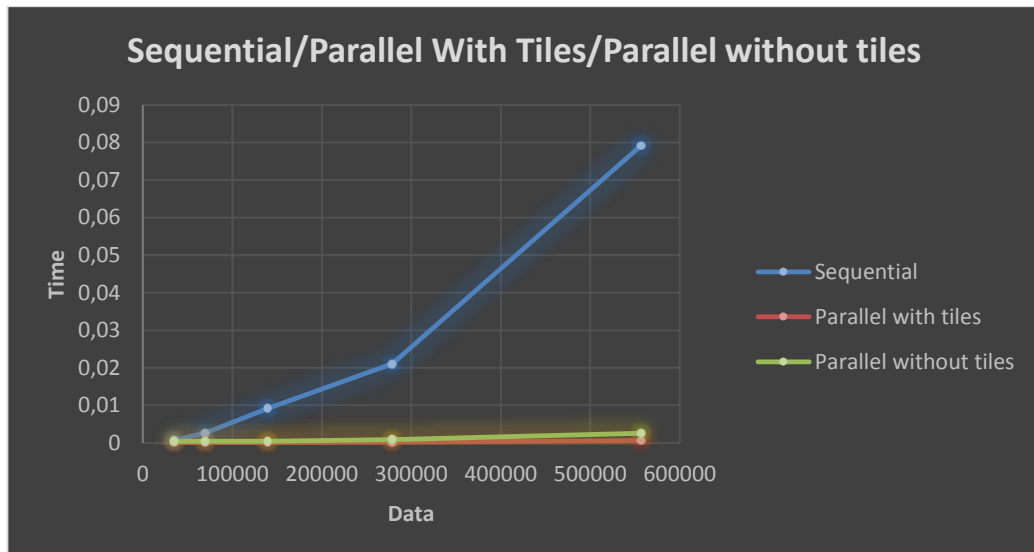
Gráfica 14: Aceleración del algoritmo en paralelo con tiles y sin tiles con respecto al algoritmo secuencial

Por último, se pasa a analizar los datos obtenidos con matrices no cuadradas usando tamaño de bloques de **4x4**, para esto se usará la tabla de datos presentada a continuación:

Tabla 2.3: Datos recolectado bloques 4x4 matrices no cuadradas

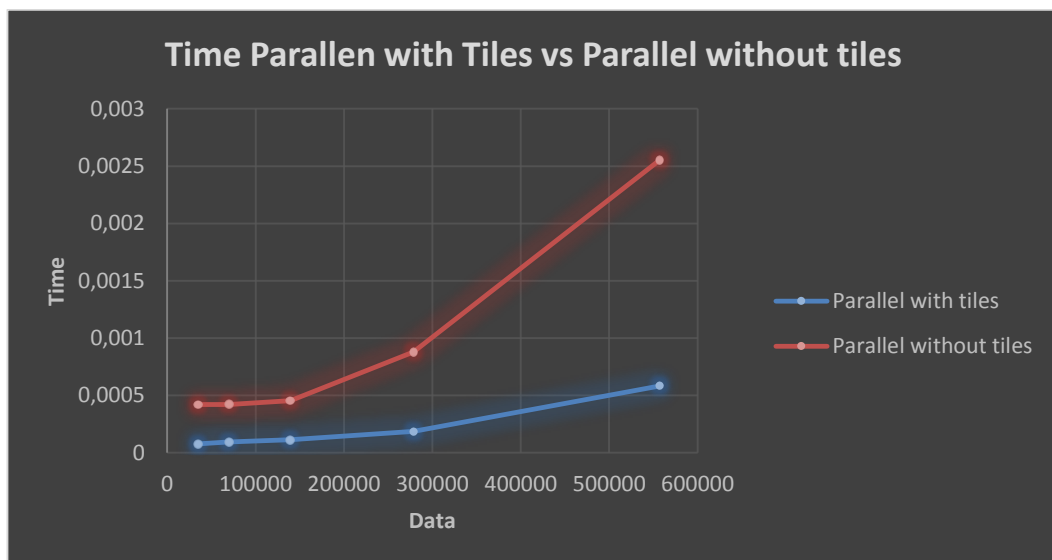
| INFORMATION | | | | | | | | |
|-------------|-----------------|---------|--------|------------|---------------------|-------------------------|------------------------|----------------------------|
| Row A | Colum A & Row B | Colum B | Data | Sequential | Parallel with tiles | Acceleration with tiles | Parallel without tiles | Acceleration without tiles |
| 2 | 1024 | 32 | 34816 | 0,000657 | 0,000077 | 8,532467532 | 0,00042 | 1,564285714 |
| 4 | 1024 | 64 | 69632 | 0,002648 | 0,000093 | 28,47311828 | 0,000421 | 6,289786223 |
| 8 | 1024 | 128 | 139264 | 0,009169 | 0,000113 | 81,14159292 | 0,000454 | 20,19603524 |
| 16 | 1024 | 256 | 278528 | 0,021046 | 0,000186 | 113,1505376 | 0,000879 | 23,94311718 |
| 32 | 1024 | 512 | 557056 | 0,079148 | 0,000583 | 135,7598628 | 0,002553 | 31,00195848 |

De los datos otorgados en la tabla anterior se realiza la siguiente gráfica donde se representa el tiempo de ejecución de los algoritmos de multiplicación de matrices con un tamaño de bloques de 4x4.



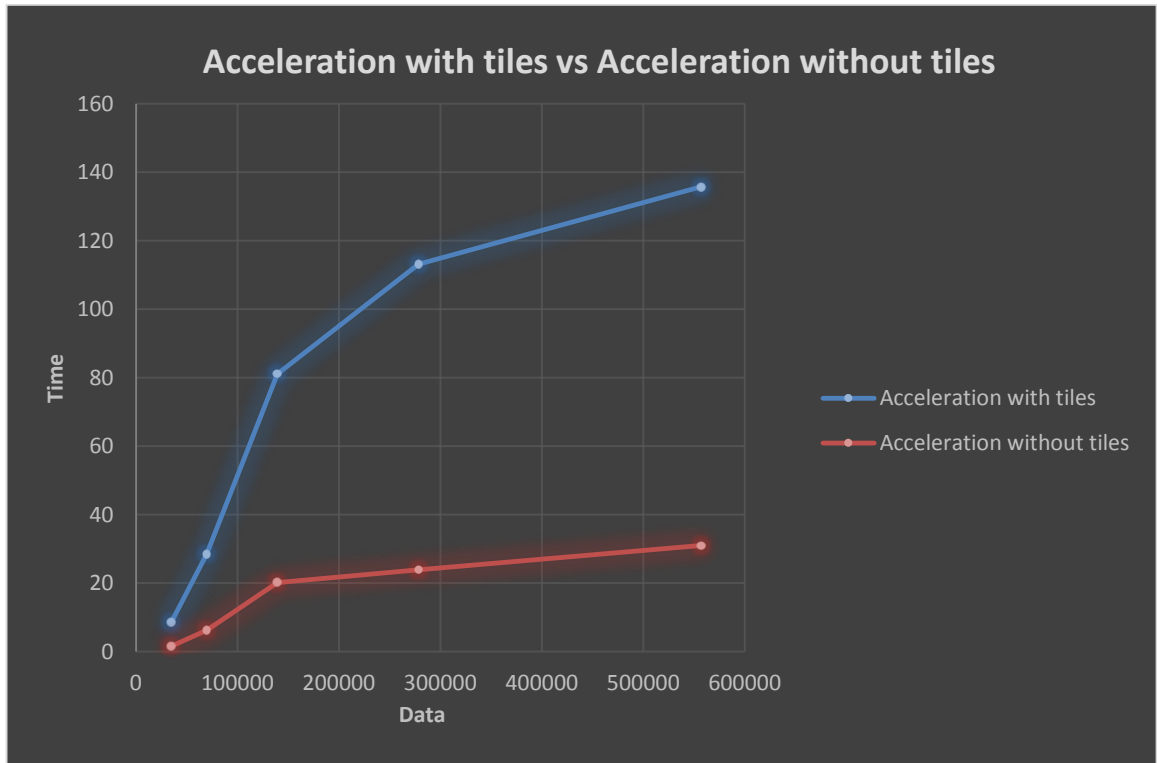
Gráfica 15: Tiempo del algoritmo en paralelo con tiles y sin tiles, y el algoritmo secuencial

Para diferencial el tiempo del algoritmo con tiles y sin tiles se muestra la siguiente gráfica:



Gráfica 16: Tiempo del algoritmo en paralelo con tiles y sin tiles

Ahora se pasa a obtener la aceleración, que es representada por la siguiente gráfica:



Gráfica 17: Aceleración del algoritmo en paralelo con tiles y sin tiles con respecto al algoritmo secuencial

3. MATRICES NO CUADRADAS CON DATOS TIPO FLOTANTES

Como se realizó en el punto anterior se hará un análisis de desempeño de algoritmo de multiplicación de matrices teniendo en cuenta el tiempo de ejecución y la aceleración del algoritmo secuencial con respecto a los algoritmos en paralelo. También se tomaran en cuenta diferentes tamaños de bloque y se determinara que algoritmo es más eficaz.

Se comenzará con bloques de tamaño de **32x32**. Se obtuvieron los siguientes datos con este tamaño de bloques:

Tabla 3.1: Datos recolectado bloques 32x32

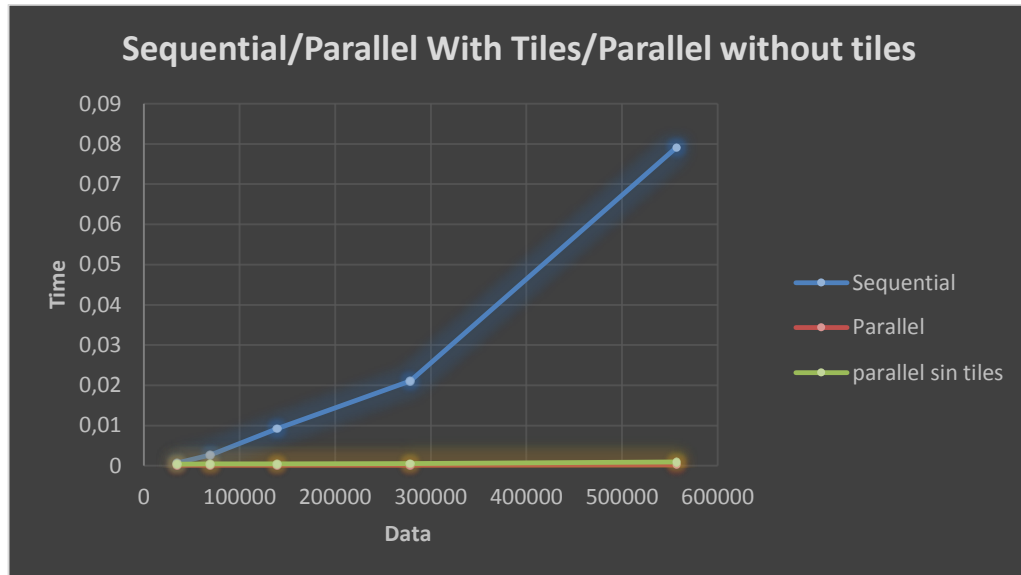
| INFORMATION | | | | | | | | |
|-------------|-----------------|---------|--------|------------|---------------------|-------------------------|------------------------|----------------------------|
| Row A | Colum A & Row B | Colum B | Data | Sequential | Parallel with tiles | Acceleration with tiles | Parallel without tiles | Acceleration without tiles |
| 2 | 1024 | 32 | 34816 | 0,000657 | 0,000124 | 5,298387097 | 0,000401 | 1,63840399 |
| 4 | 1024 | 64 | 69632 | 0,002648 | 0,000126 | 21,01587302 | 0,000425 | 6,230588235 |
| 8 | 1024 | 128 | 139264 | 0,009169 | 0,000138 | 66,44202899 | 0,000426 | 21,52347418 |
| 16 | 1024 | 256 | 278528 | 0,021046 | 0,000154 | 136,6623377 | 0,000505 | 41,67524752 |
| 32 | 1024 | 512 | 557056 | 0,079148 | 0,000227 | 348,6696035 | 0,000922 | 85,84381779 |

Con ayuda de la tabla anterior se van a realizaron tres gráficas donde se muestran:

- El tiempo de los tres algoritmos.
- El tiempo de los algoritmos en paralelo.
- La aceleración de los algoritmos en paralelo

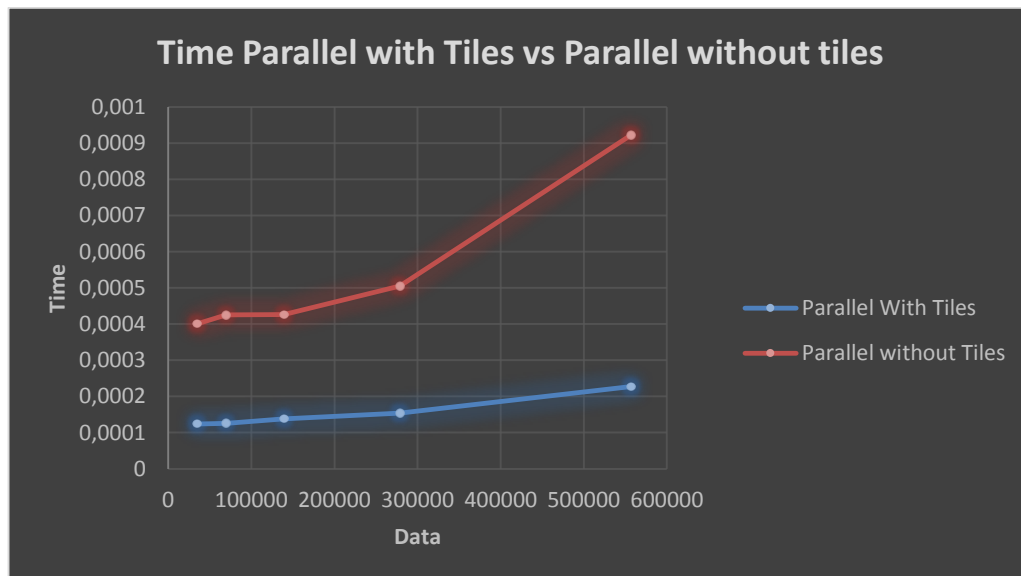
Esto con bloques de 32x32, y se puede observar en la gráfica 18, 19 y 20 respectivamente.

Esta primera gráfica representa los tiempos de ejecución de cada uno de los algoritmos:



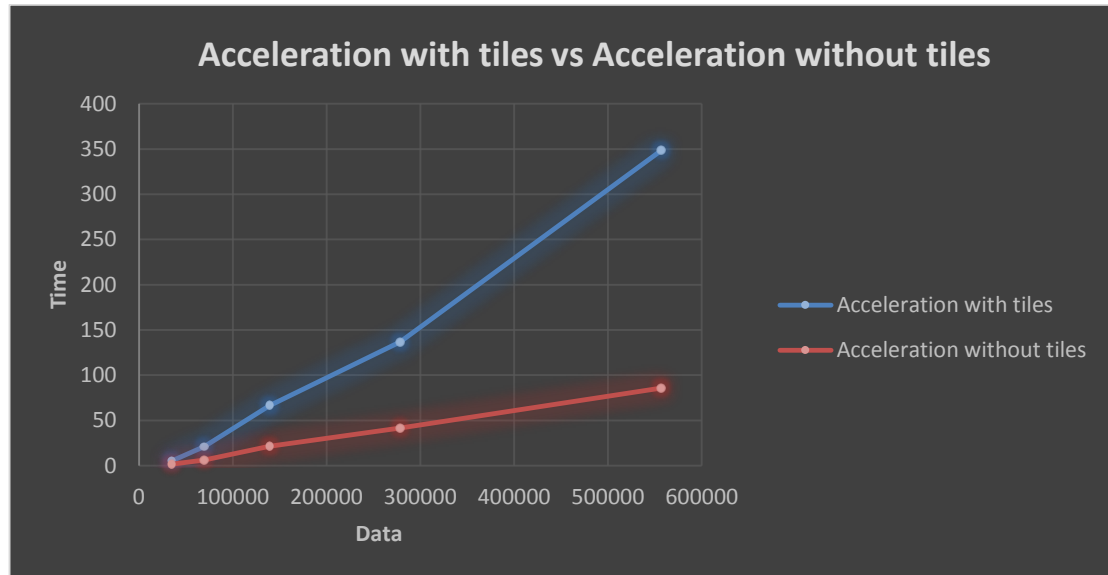
Gráfica 18: Tiempos de ejecución de los tres algoritmos

Para observar la diferencia de desempeño de los algoritmos en paralelo se realizó la siguiente gráfica:



Gráfica 19: Tiempos de ejecución de los algoritmos en paralelo

Se analiza la aceleración que tienen los algoritmos en paralelo con respecto al algoritmo secuencial:

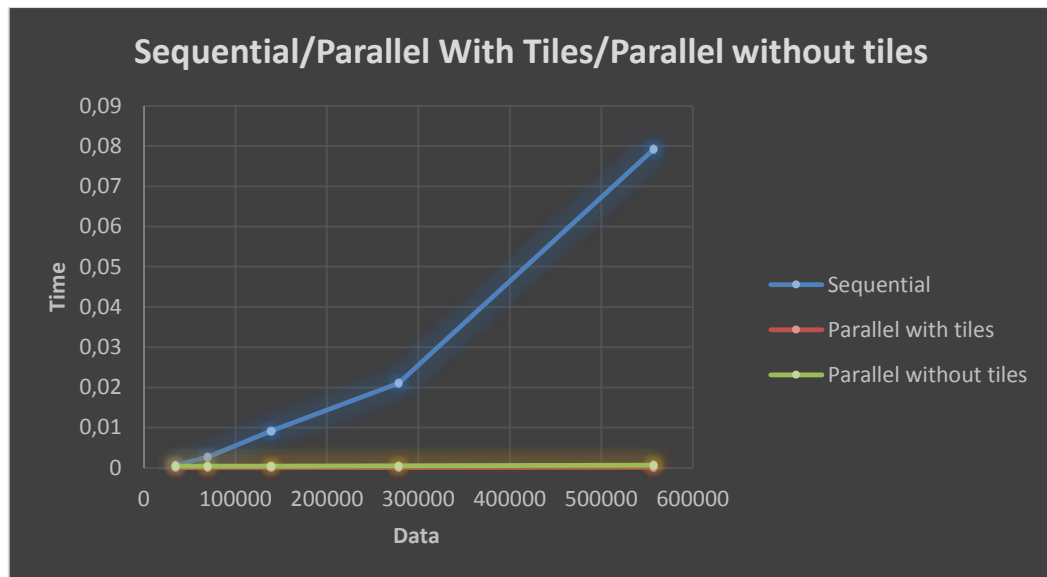


Gráfica 20: Aceleración del algoritmo en paralelo con tiles y sin tiles con respecto al algoritmo secuencial

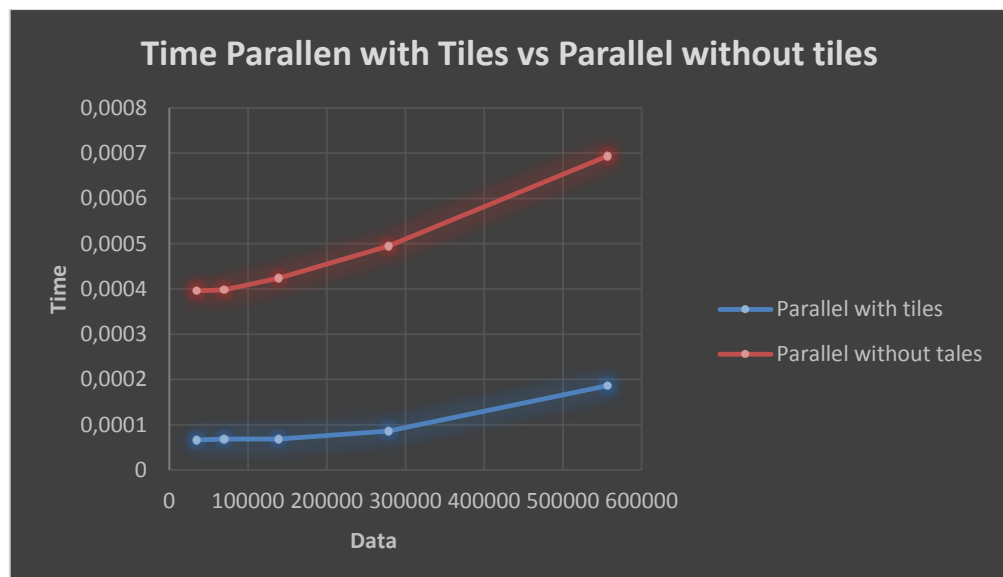
Se pasa analizar lo que sucede con bloques de 16x16, para esto se tienen en cuenta los siguientes datos:

Tabla 3.2: Datos recolectado bloques 16x16

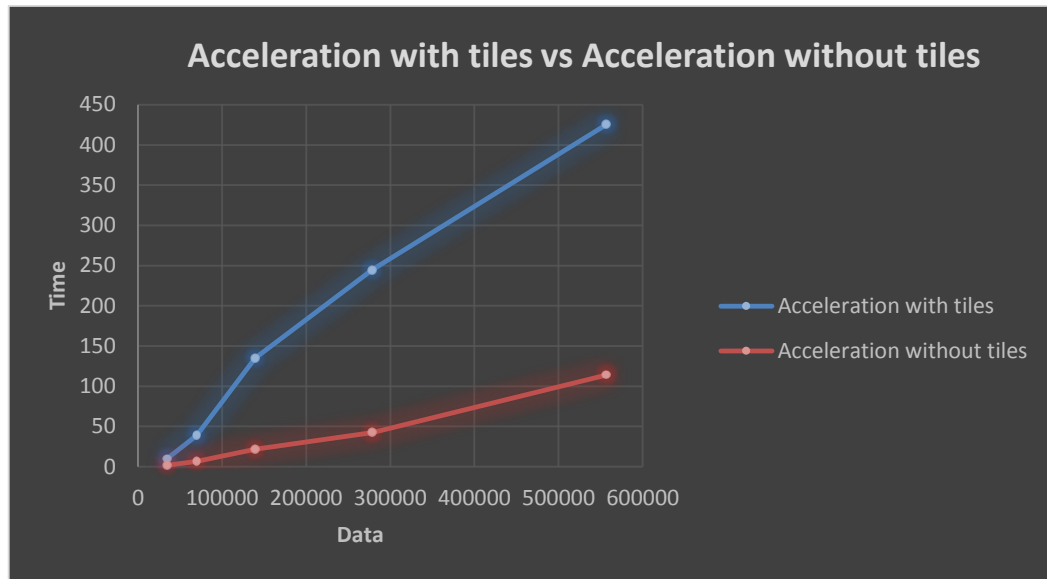
| INFORMATION | | | | | | | | |
|-------------|------------------|----------|--------|------------|---------------------|-------------------------|------------------------|----------------------------|
| Row A | Column A & Row B | Column B | Data | Sequential | Parallel with tiles | Acceleration with tiles | Parallel without tiles | Acceleration without tiles |
| 2 | 1024 | 32 | 34816 | 0,000657 | 0,000066 | 9,954545455 | 0,000396 | 1,659090909 |
| 4 | 1024 | 64 | 69632 | 0,002648 | 0,000068 | 38,94117647 | 0,000398 | 6,653266332 |
| 8 | 1024 | 128 | 139264 | 0,009169 | 0,000068 | 134,8382353 | 0,000424 | 21,625 |
| 16 | 1024 | 256 | 278528 | 0,021046 | 0,000086 | 244,7209302 | 0,000495 | 42,51717172 |
| 32 | 1024 | 512 | 557056 | 0,079148 | 0,000186 | 425,5268817 | 0,000694 | 114,0461095 |



Gráfica 21: Tiempos de ejecución de los tres algoritmos



Gráfica 22: Tiempos de ejecución de los algoritmos en paralelo



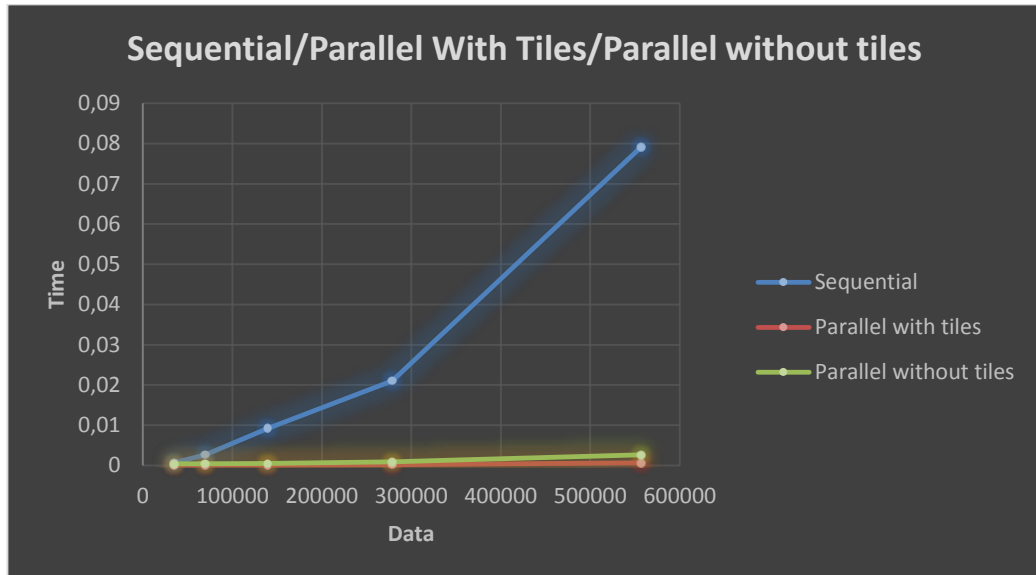
Gráfica 23: Aceleración del algoritmo en paralelo con tiles y sin tiles con respecto al algoritmo secuencial

Ahora con bloques de 4x4:

Tabla 3.3: Datos recolectado bloques 4x4

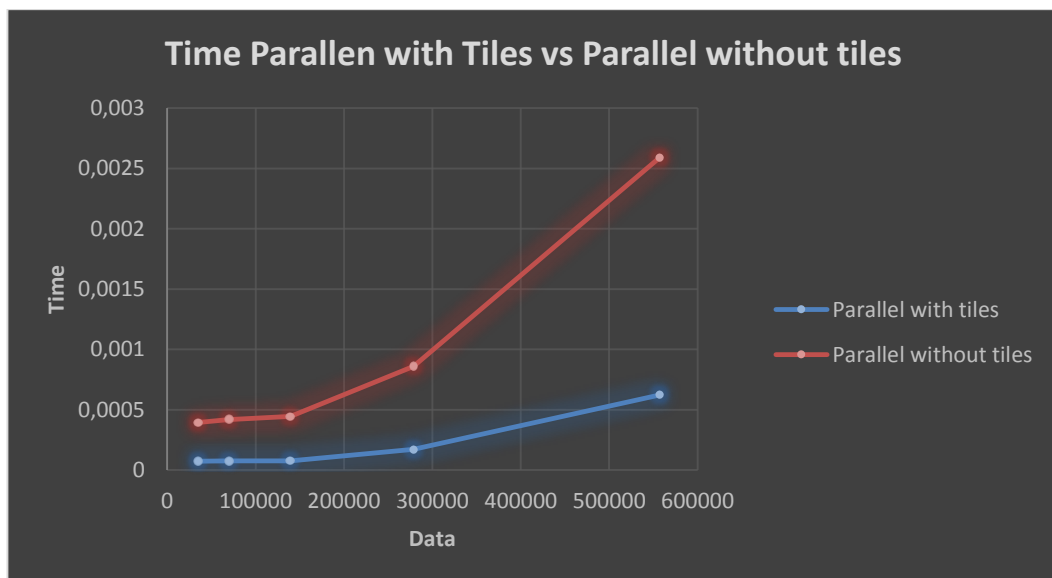
| INFORMATION | | | | | | | | |
|-------------|-----------------|---------|--------|------------|---------------------|-------------------------|------------------------|----------------------------|
| Row A | Colum A & Row B | Colum B | Data | Sequential | Parallel with tiles | Acceleration with tiles | Parallel without tiles | Acceleration without tiles |
| 2 | 1024 | 32 | 34816 | 0,000657 | 0,000075 | 8,76 | 0,000394 | 1,66751269 |
| 4 | 1024 | 64 | 69632 | 0,002648 | 0,000076 | 34,84210526 | 0,000419 | 6,319809069 |
| 8 | 1024 | 128 | 139264 | 0,009169 | 0,000078 | 117,5512821 | 0,000445 | 20,60449438 |
| 16 | 1024 | 256 | 278528 | 0,021046 | 0,000172 | 122,3604651 | 0,00086 | 24,47209302 |
| 32 | 1024 | 512 | 557056 | 0,079148 | 0,000624 | 126,8397436 | 0,00259 | 30,55907336 |

Teniendo en cuenta la tabla anterior se obtuvieron las siguientes gráficas:



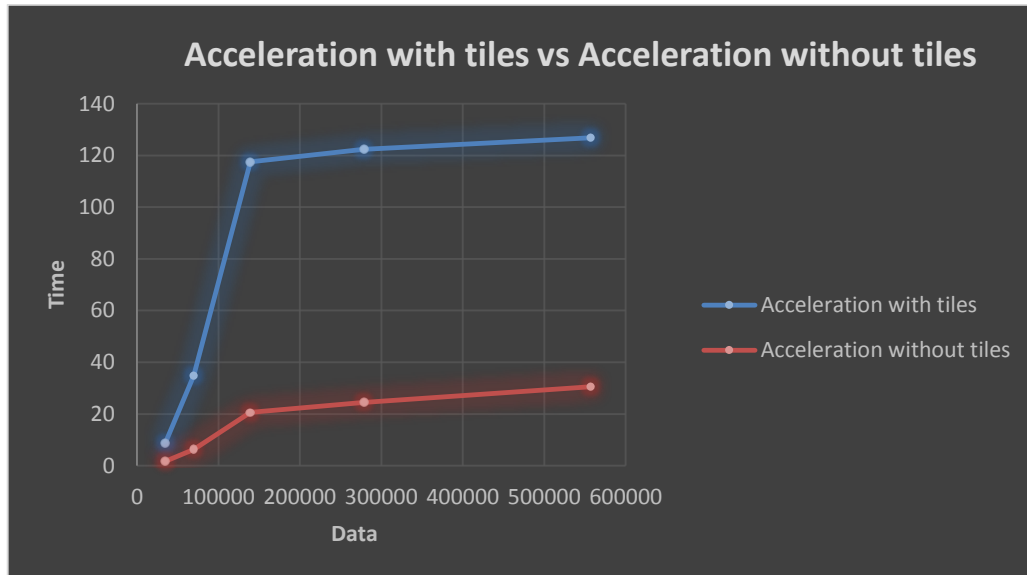
Gráfica 24: Tiempos de ejecución de los tres algoritmos

Los tiempos de los algoritmos en paralelo:



Gráfica 25: Tiempos de ejecución de los algoritmos en paralelo

La aceleración que se presenta entre los algoritmos en paralelo con respecto al algoritmo secuencial:

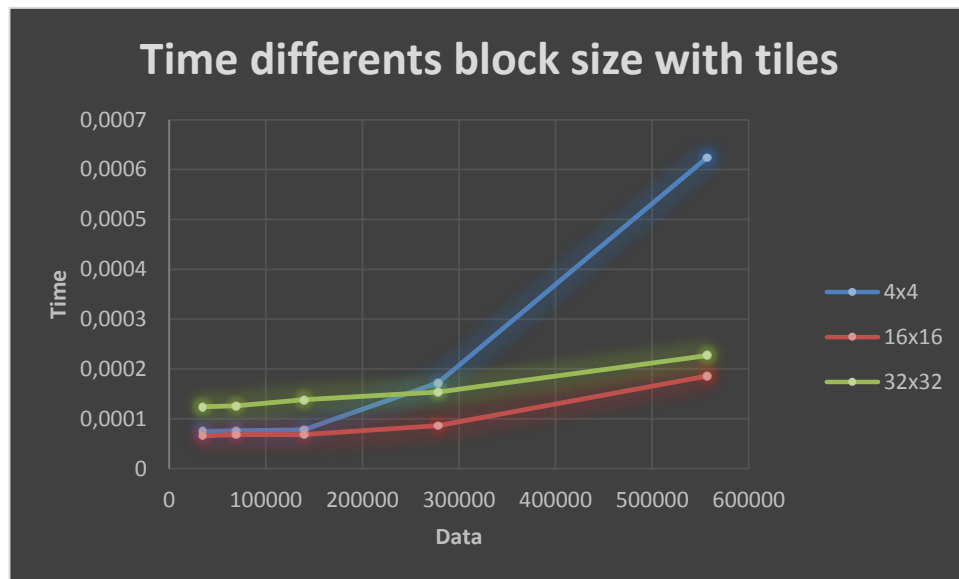


Gráfica 26: Aceleración del algoritmo en paralelo con tiles y sin tiles con respecto al algoritmo secuencial

A lo largo del documento se analizan gráficamente todos los tiempos de ejecución y las aceleraciones que tienen los algoritmos en paralelo (con tiles y sin tiles) para con respecto al tiempo de ejecución del algoritmo de multiplicación de matrices secuencial.

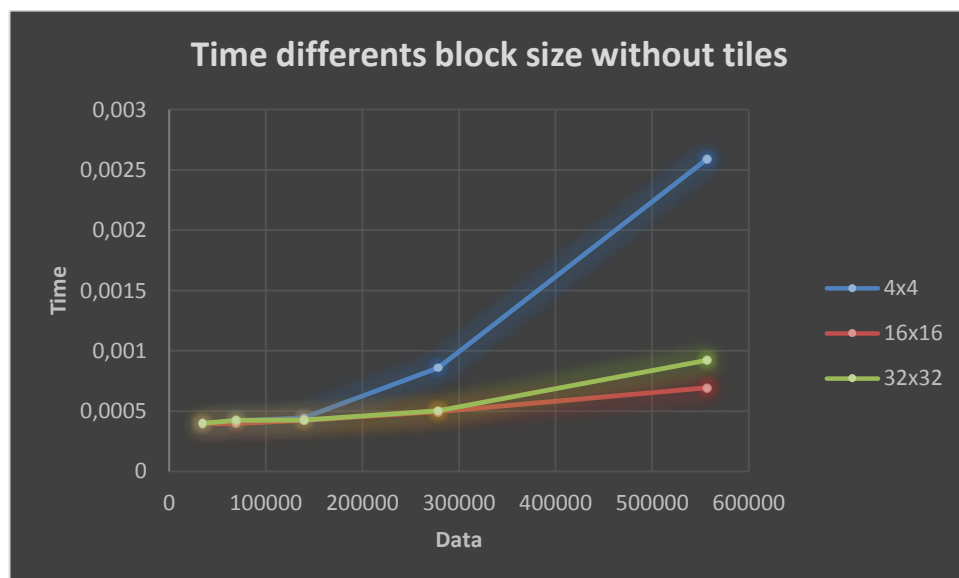
Hasta el momento se ha realizado un análisis de los tiempos y aceleraciones que tienen los algoritmos con diferentes tamaños de bloques (4x4, 16x16, 32x32), esto con el fin de poder concluir que tamaño de bloques funciona más para este tipo de problema. En las gráficas que se van a presentar a continuación se hace una comparación de los tiempos que toma con cada tamaño de bloque y las aceleraciones respectivas que se tienen en cada uno de ellos (teniendo en cuenta algoritmo con tiles y sin tiles).

La gráfica 27 representa los tiempos de ejecución del algoritmo para cada uno de los bloques propuestos para el análisis.



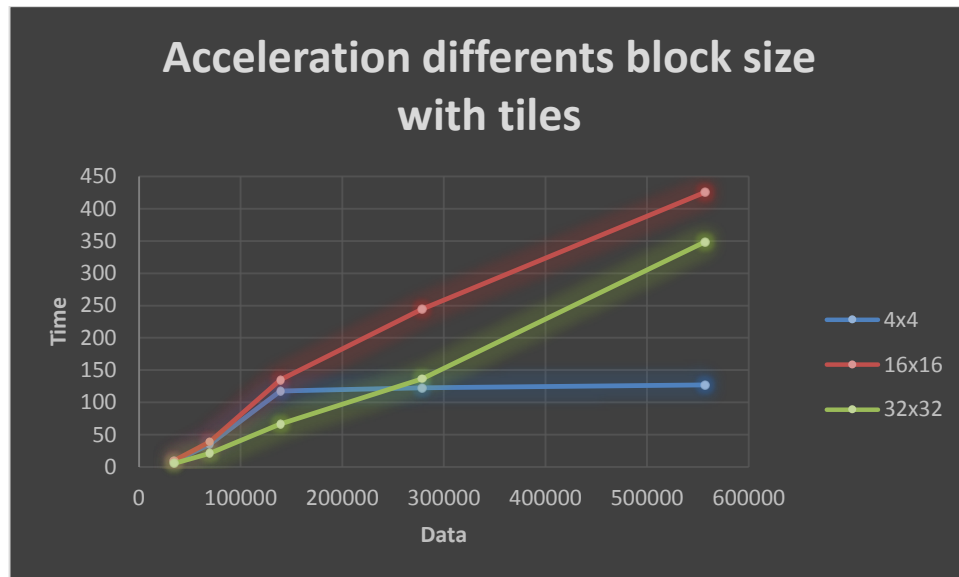
Gráfica 27: Comparación de tiempos de ejecución con tiles con diferentes bloques

En la gráfica anterior se puede ver claramente que el algoritmo paralelo con tiles no es óptimo usando bloques de 4x4, y que para este caso tampoco es mejor usar bloques de 32x32. Un tamaño de bloques de 16x16 mejora el desempeño del algoritmo reduciendo el tiempo de ejecución.



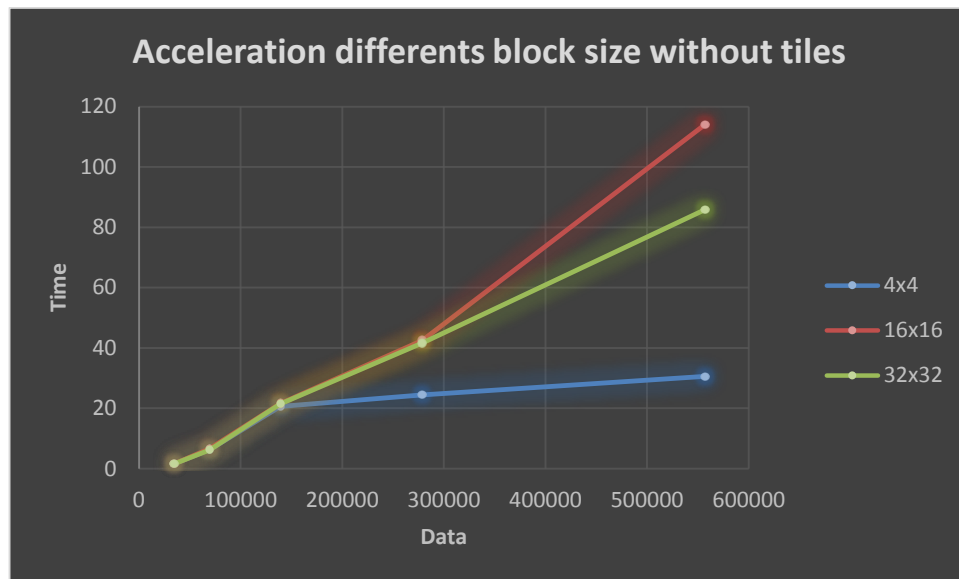
Gráfica 28: Comparación de tiempos de ejecución sin tiles con diferentes bloques

En el caso de no usar tiles ocurre que también es más eficaz usar un tamaño de bloques de 16x16 pero el desempeño general del algoritmo en comparación al usar tiles no es el más óptimo.



Gráfica 29: Comparación de aceleración con tiles con diferentes bloques

La aceleración que presentan los algoritmos en paralelo con respecto al algoritmo secuencial se muestran en las gráficas 29 y 30. En la gráfica 30 se representa la aceleración de los algoritmos sin usar la técnica de tiling, como se puede evidenciar la aceleración que alcanza el algoritmo con los datos que se presentaron va en aumento y alcanza una cifra aproximada de 115X usando bloques de 16x16, usando tiles con este mismo tamaño de bloques la aceleración alcanzada es de alrededor de 430X casi 4 veces mayor que la cifra sin tiling. Como se puede notar el desempeño es mucho mejor usando tiles.



Gráfica 30: Comparación de aceleración sin tiles con diferentes bloques

4. CONCLUSIONES

- Usando un algoritmo secuencial, paralelo con tiles y paralelo sin tiles para la multiplicación de matrices cuadradas de datos enteros se tiene que la mejor solución es el algoritmo paralelo con tiles.
- Al analizar el uso de diferentes tamaños de bloque para la multiplicación de matrices cuadradas con datos enteros, teniendo en cuenta las gráficas 7 y 8 se puede concluir que el mejor tamaño de bloques para usar es de 32x32, puesto que es el que tiene menor tiempo de ejecución y mayor aceleración, sin llegar a alcanzar un tiempo de 0.02s y alcanzar una aceleración aproximadamente de 1600X.
- En el caso de multiplicación de matrices no cuadradas con datos tipos enteros, se puede decir que el algoritmo paralelo con tiles es el más eficaz y que en este caso específico el tamaño de los bloques óptimo es de 16x16.
- Teniendo en cuenta las gráficas 27 y 29, para este caso de matrices no cuadradas la técnica de tiling es más eficaz y el tamaño de bloques óptimo es de 16x16.

- En general, la multiplicación de matrices tanto cuadradas como no cuadradas tienen un mejor desempeño usando algoritmos paralelos aplicando la técnica de tiling.
- Se puede concluir que la aceleración va en aumento a medida que la matriz aumenta de tamaño. Hay que tener en cuenta que, debido a la cantidad de datos obtenidos no se puede decir exactamente si la aceleración seguirá en aumento o en algún punto llegará a estabilizarse.
- En general tanto para matrices cuadradas o no cuadradas el uso de pocos bloques no es una buena opción para mejorar el desempeño de este algoritmo.
- El desempeño del algoritmo de multiplicación de matrices no cuadradas es mejor usando un tamaño de bloques de 16×16 a diferencia del de matrices cuadradas que es de 32×32 .