



**Tecnológico  
de Monterrey**

# **Multiplicación de matrices usando Open MP**

Implementación de sistemas embebidos avanzados  
Grupo 501

05.11.2022

## **Alumno**

Fernando Cerriteño Magaña - A01702790

## **Profesor**

Víctor Rodríguez Bahena

## Introducción

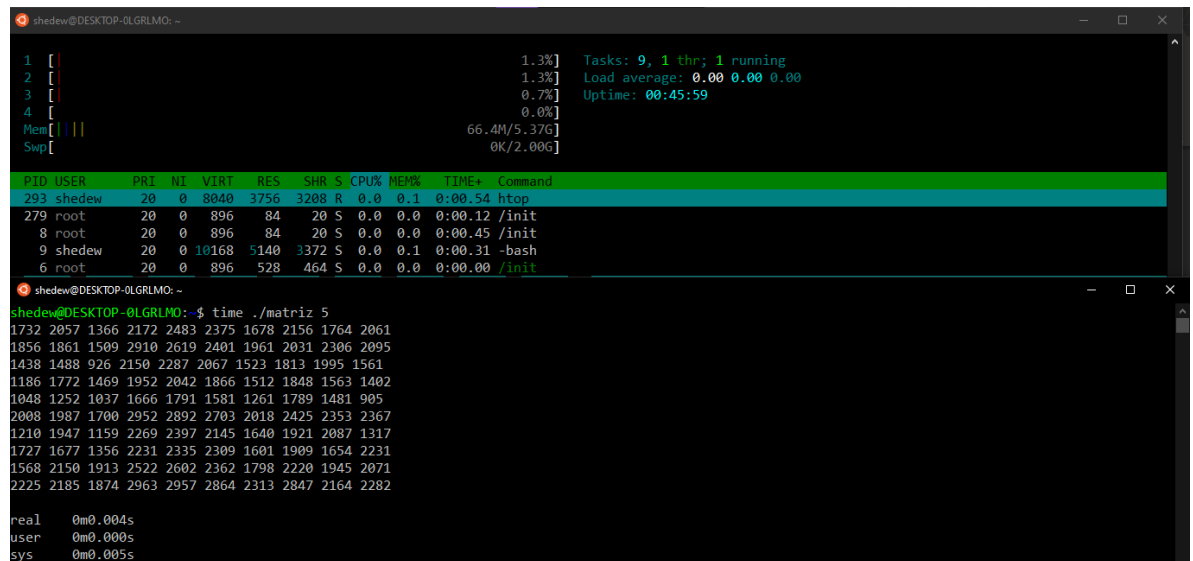
Esta práctica consistió en el calculo de la multiplicación de dos matrices de tamaño NxN usando Open MP, esto con el propósito de entender mejor el como funciona Open MP y el como se implementa de forma práctica.

Enlace del repositorio en GitHub:

<https://github.com/Shedew/parallel-programming-ITESM/tree/main/Matrix%20multiplication%20in%20open%20MP>

## Uso del CPU obtenido con el comando htop

- Usando 5 threads



The screenshot shows a terminal window with two panels. The top panel displays the output of the `htop` command, showing system statistics and a list of running processes. The bottom panel shows the output of the `time ./matriz 5` command, displaying a 5x5 matrix of random numbers and the execution time.

```
shedew@DESKTOP-OLGRIMO: ~  
1  [ ] 1.3% Tasks: 9, 1 thr; 1 running  
2  [ ] 1.3% Load average: 0.00 0.00 0.00  
3  [ ] 0.7% Uptime: 00:45:59  
4  [ ] 0.0%  
Mem[||||] 66.4M/5.37G  
Swp[ ] 0K/2.00G  
  
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command  
293 shedew 20 0 8040 3756 3208 R 0.0 0.1 0:00.54 htop  
279 root 20 0 896 84 20 S 0.0 0.0 0:00.12 /init  
8 root 20 0 896 84 20 S 0.0 0.0 0:00.45 /init  
9 shedew 20 0 10168 5140 3372 S 0.0 0.1 0:00.31 -bash  
6 root 20 0 896 528 464 S 0.0 0.0 0:00.00 /init  
  
shedew@DESKTOP-OLGRIMO: ~  
shedew@DESKTOP-OLGRIMO:~$ time ./matriz 5  
1732 2057 1366 2172 2483 2375 1678 2156 1764 2061  
1856 1861 1509 2910 2619 2401 1961 2031 2306 2095  
1438 1488 926 2150 2287 2067 1523 1813 1995 1561  
1186 1772 1469 1952 2042 1866 1512 1848 1563 1402  
1048 1252 1037 1666 1791 1581 1261 1789 1481 905  
2008 1987 1700 2952 2892 2703 2018 2425 2353 2367  
1210 1947 1159 2269 2397 2145 1640 1921 2087 1317  
1727 1677 1356 2231 2335 2309 1601 1909 1654 2231  
1568 2150 1913 2522 2602 2362 1798 2220 1945 2071  
2225 2185 1874 2963 2957 2864 2313 2847 2164 2282  
  
real 0m0.004s  
user 0m0.000s  
sys 0m0.005s
```

- Usando 10 threads

```
shedew@DESKTOP-OLGRLMO: ~  
1 [ 0.0% Tasks: 9, 1 thr; 1 running  
2 [ 0.0% Load average: 0.00 0.00 0.00  
3 [ 0.7% Uptime: 00:46:43  
4 [ 0.0%  
Mem[||||] 66.1M/5.37G  
Swp[ ] 0K/2.00G  
  
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command  
8 root 20 0 896 84 20 S 0.0 0.0 0:00.46 /init  
293 shedew 20 0 8040 3756 3208 R 0.0 0.1 0:00.60 htop  
9 shedew 20 0 10168 5140 3372 S 0.0 0.1 0:00.32 -bash  
279 root 20 0 896 84 20 S 0.0 0.0 0:00.13 /init  
6 root 20 0 896 528 464 S 0.0 0.0 0:00.00 /init  
1 root 20 0 896 528 464 S 0.0 0.0 0:00.05 /init  
7 root 20 0 896 84 20 S 0.0 0.0 0:00.00 /init  
259 shedew 20 0 9500 4848 3088 T 0.0 0.1 0:00.51 nano matriz.c  
  
shedew@DESKTOP-OLGRLMO: ~  
shedew@DESKTOP-OLGRLMO:~$ time ./matriz 10  
1907 2238 1831 2356 1911 2049 1853 2803 2274 2627  
1078 1340 1273 1380 1299 1277 1283 1705 1335 2048  
1590 1886 1931 1857 2036 1708 1230 2220 1870 1879  
2072 1811 1588 2369 2062 1697 2007 2451 2364 2338  
1577 1610 1598 1825 1667 1511 1971 2197 1804 2623  
1691 1533 1438 2095 1661 1574 1867 2207 2146 2370  
1306 1829 1880 1708 2118 1473 1617 2111 1634 2297  
1116 1314 1217 1547 1273 1162 924 1794 890 1702  
1214 1401 1503 1446 1806 1188 1017 1621 1381 1654  
1874 1964 1904 2088 1756 2140 1480 2586 2046 2773  
  
real 0m0.004s  
user 0m0.005s  
sys 0m0.001s  
shedew@DESKTOP-OLGRLMO:~$
```

- Usando 50 threads

```
shedew@DESKTOP-OLGRLMO: ~  
1 [ 0.7% Tasks: 9, 1 thr; 1 running  
2 [ 0.0% Load average: 0.00 0.00 0.00  
3 [ 0.7% Uptime: 00:47:34  
4 [ 0.0%  
Mem[||||] 68.0M/5.37G  
Swp[ ] 0K/2.00G  
  
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command  
293 shedew 20 0 8040 3756 3208 R 0.0 0.1 0:00.69 htop  
9 shedew 20 0 10168 5140 3372 S 0.0 0.1 0:00.34 -bash  
8 root 20 0 896 84 20 S 0.0 0.0 0:00.47 /init  
279 root 20 0 896 84 20 S 0.0 0.0 0:00.14 /init  
6 root 20 0 896 528 464 S 0.0 0.0 0:00.00 /init  
1 root 20 0 896 528 464 S 0.0 0.0 0:00.05 /init  
7 root 20 0 896 84 20 S 0.0 0.0 0:00.00 /init  
259 shedew 20 0 9500 4848 3088 T 0.0 0.1 0:00.51 nano matriz.c  
  
shedew@DESKTOP-OLGRLMO: ~  
shedew@DESKTOP-OLGRLMO:~$ time ./matriz 50  
1252 1705 1139 1599 1603 1467 1569 1265 1395 1579  
1489 1632 1086 1651 2072 2052 1173 1226 1487 1777  
2086 2087 1080 1727 1971 2418 1357 1375 1695 1819  
1042 1070 908 1061 1334 1506 1086 1228 1133 1031  
1417 1896 1372 1757 2143 1882 1587 1692 1729 1891  
1470 1558 833 1454 1273 1682 1001 973 1177 1363  
1170 1269 1012 1426 1886 1660 1146 1205 1283 1432  
1356 1790 947 1552 1125 1402 1201 1250 1370 1438  
848 1180 713 927 1315 1300 839 795 1021 1235  
2133 2230 1355 2006 2375 2322 1716 1676 1960 2090  
  
real 0m0.014s  
user 0m0.000s  
sys 0m0.020s
```

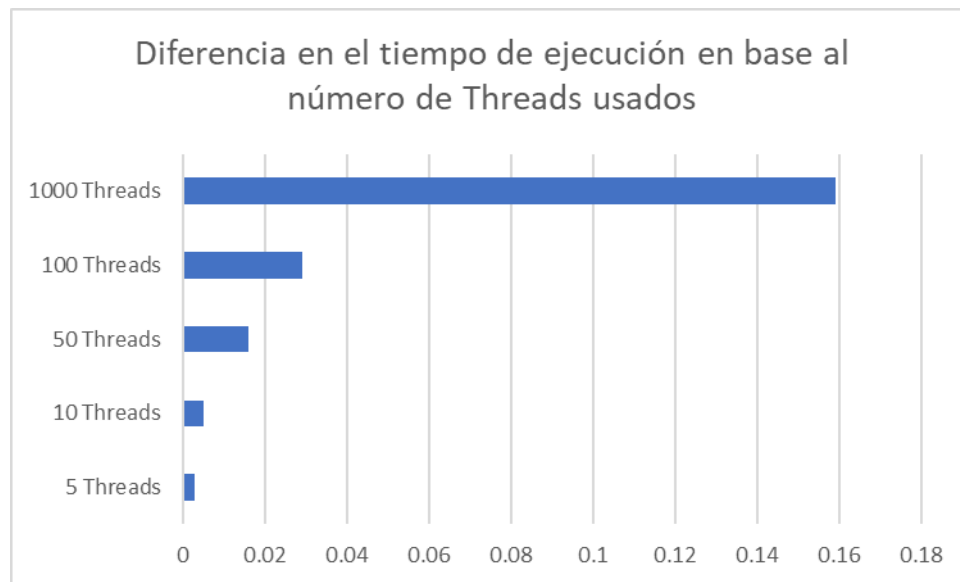
- Usando 100 threads

```
shedew@DESKTOP-OLGRIMO: ~  
1  [ ] 2.0% Tasks: 9, 1 thr; 1 running  
2  [ ] 0.7% Load average: 0.00 0.00 0.00  
3  [ ] 0.7% Uptime: 00:47:59  
4  [ ] 0.0%  
Mem[ ] 68.4M/5.37G  
Swp[ ] 0K/2.00G  
  
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command  
293 shedew 20 0 8040 3756 3208 R 0.0 0.1 0:00.71 htop  
9 shedew 20 0 10168 5140 3372 S 0.0 0.1 0:00.34 -bash  
8 root 20 0 896 84 20 S 0.0 0.0 0:00.47 /init  
279 root 20 0 896 84 20 S 0.0 0.0 0:00.14 /init  
6 root 20 0 896 528 464 S 0.0 0.0 0:00.00 /init  
1 root 20 0 896 528 464 S 0.0 0.0 0:00.05 /init  
7 root 20 0 896 84 20 S 0.0 0.0 0:00.00 /init  
259 shedew 20 0 9500 4848 3088 T 0.0 0.1 0:00.51 nano matriz.c  
  
shedew@DESKTOP-OLGRIMO: ~  
shedew@DESKTOP-OLGRIMO:~$ time ./matriz 100  
1250 2453 1463 1367 1709 2728 1450 2022 1834 1803  
1043 2219 1338 1806 1744 2327 1449 1439 1747 1596  
733 2033 1189 1708 1364 2279 1544 1820 1551 1680  
1245 2952 1841 1932 1872 3218 1802 2674 2337 2241  
1094 2657 1632 1667 2050 2660 1365 1910 1878 1883  
1073 2103 1182 1567 1649 2121 1094 1287 1547 1428  
1027 2127 1258 1691 1605 2702 1571 2076 2037 1918  
978 2173 1353 2053 1730 2376 1439 1467 1930 1697  
1089 2387 1550 2011 1607 2694 1545 1667 2048 1823  
1379 2343 1756 1290 1791 2661 1369 1935 1950 1655  
  
real 0m0.029s  
user 0m0.000s  
sys 0m0.036s  
shedew@DESKTOP-OLGRIMO:~$
```

- Usando 1000 threads

```
shedew@DESKTOP-OLGRIMO: ~  
1  [|||||] 10.7% Tasks: 9, 1 thr; 1 running  
2  [||] 1.3% Load average: 0.00 0.00 0.00  
3  [||] 1.3% Uptime: 00:48:13  
4  [||] 1.3%  
Mem[ ] 69.4M/5.37G  
Swp[ ] 0K/2.00G  
  
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command  
293 shedew 20 0 8040 3756 3208 R 0.0 0.1 0:00.73 htop  
9 shedew 20 0 10168 5140 3372 S 0.0 0.1 0:00.34 -bash  
8 root 20 0 896 84 20 S 0.0 0.0 0:00.47 /init  
279 root 20 0 896 84 20 S 0.0 0.0 0:00.14 /init  
6 root 20 0 896 528 464 S 0.0 0.0 0:00.00 /init  
1 root 20 0 896 528 464 S 0.0 0.0 0:00.05 /init  
7 root 20 0 896 84 20 S 0.0 0.0 0:00.00 /init  
259 shedew 20 0 9500 4848 3088 T 0.0 0.1 0:00.51 nano matriz.c  
  
shedew@DESKTOP-OLGRIMO: ~  
shedew@DESKTOP-OLGRIMO:~$ time ./matriz 1000  
1817 1470 2162 1822 2217 2008 1412 2562 2335 1628  
1463 1383 1825 1922 2052 1635 1375 2386 2272 1547  
2283 2427 2684 2472 3096 2403 2077 3534 3028 2561  
1583 1511 1894 1690 2139 1587 1300 2345 2271 1742  
1711 2070 2484 2346 2903 2175 1971 3007 2597 2121  
1295 1607 1620 1346 1824 1295 1425 1869 1760 1500  
1022 1308 1150 1168 1650 1165 1013 1647 1354 1353  
787 1169 1420 1434 1674 1176 1122 1768 1650 1316  
1398 1690 2182 1790 2216 1665 1470 2491 2233 1801  
1631 1688 1919 1618 2332 1766 1313 2453 2088 2027  
  
real 0m0.159s  
user 0m0.023s  
sys 0m0.232s  
shedew@DESKTOP-OLGRIMO:~$
```

## Gráfica comparativa del tiempo de ejecución con respecto a diferentes números de threads



## Conclusiones

Mientras que en teoría el tiempo de ejecución debería de ser menor en cuanto se usen mas threads, se tiene que aclarar que se uso la maquina virtual de Ubuntu para poder trabajar con threads, por lo que es muy probable que al usar una maquina virtual, esta se puede llegar a tardar más cuando se trabaja con diferentes threads. No obstante, observando la utilización del CPU con htop al momento de obtener los resultados, se puede observar como en cuanto se usan mas threads, más se alteran las gráficas, en especial si se comparan cuando se usaron 5, 100, y 1000 threads.

En cuanto al funcionamiento del código, se usó como referencia el pseudocódigo encontrado en la página de [tutorialpoints](#), con el cual nada más se le implemento la librería `omp.h`, con la cual se nos fue posible trabajar con threads usando la función `"#pragma omp parallel for private(i,j,k) shared( a,b,c )"`.