# Exercise 3.4

|  |  |  |  |
| --- | --- | --- | --- |
| **Process** | **Time** | **Speed-Up** | **Efficiency** |
| **1** | 37,56 | 1 | 100 |
| **2** | 19,95 | 1,88270677 | 94,1353383 |
| **4** | 10,06 | 3,73359841 | 93,3399602 |
| **6** | 6,75 | 5,56444444 | 92,7407407 |
| **8** | 5,09 | 7,37917485 | 92,2396857 |
| **10** | 4,09 | 9,18337408 | 91,8337408 |
| **12** | 3,4 | 11,0470588 | 92,0588235 |

As shown in the table there is a significant speed up to be achieved by parallelizing the matrix multiplication. However, it has to be noted that as the number of processes increases, the efficiency of parallelization tends to decrease. The fast increase of speed up is also illustrated in the diagram above (blue line).

# Exercise 3.5

|  |  |  |
| --- | --- | --- |
| **Size** | **Time (s)** | **GFLOPS/s** |
| **128** | 0,018594 | 0,225578 |
| **256** | 0,026827 | 1,25 |
| **512** | 0,094914 | 2,8282 |
| **1024** | 0,619426 | 3,466891 |
| **2048** | 4,348465 | 3,95079 |
| **4096** | 34,584087 | 3,974052 |
| **8192** | 277,430282 | 3,9632 |

As the matrix size increases, the time taken for the multiplication operation also increases exponentially, as expected due to the computational complexity. The performance, measured in GFLOPS/s increases for larger matrix sizes until it reaches a stagnation at 4 GFLOPS/s.