|  |  |  |  |
| --- | --- | --- | --- |
| **Explain the result** | **Execution time** | **Number of Slaves** | **Solution type** |
| One process takes time because it does all the work |  | 1 | Sequential Solution |
| The work is divided between two processes, so the time is shorter |  | 2 | Static Task Pool |
| The work is divided between 4 processes, so the time is shorter but not in a really big improvement than 2 processes because a lot of time is wasted on communication between the processes |  | 4 | Static Task Pool |
| There are two processes but in the dynamic method there is a process that divides the work between the processes so only one process does all the calculations and time is wasted on communication between the processes |  | 2 | Dynamic Task Pool |
| There are 4 processes, one process divides the work between the processes and the rest of the processes do calculations.  Every time a process finishes, if a job remains the main process gives it another task and no time is wasted that the process does nothing.  So, the time is very short |  | 4 | Dynamic Task Pool |
| There are 20 processes, one process divides the work between the processes and the rest of the processes do calculations.  Whenever a process ends, if a job remains the main process gives it a different task and does not waste time that the process does nothing.  The time is shorter than 4 processes but not at a significant level because a lot of time is wasted on communication between the processes. |  | 20 | Dynamic Task Pool |