PROGRAMAÇÃO PARALELA E CONCORRENTE

COIN COUNTER

REPORT

Mestrado em Engenharia Informática
2023

Ana Teixeira - fc56336

The chosen parallelization strategy is the Fork-Join framework due to its effectiveness in dividing problems into smaller subtasks and executing them concurrently. It is particularly suitable for recursive algorithms, making it the ideal choice for parallelizing the Coin problem. In order to balance parallelism and overhead, three granularity control mechanisms were used: the surplus mechanism reduces overhead by switching to sequential execution when the task queue exceeds a threshold; the index strategy switches to sequential execution when the index reaches a specified value, optimizing the performance because of the recursive tree depth control; the max tasks strategy switches to sequential execution when the total number of tasks surpasses the threshold.

The CPU model of the machine used to execute the program is 1,6 GHz dual-core Intel Core i5 and has 2 cores and 4 threads, so each method was implemented using 2 and 4 threads. Fine-tuning involved adjusting values/thresholds for each granularity control mechanism, so each strategy was also implemented with different tuning values. Benchmarking involved evaluating the performance of each mechanism by the analysis of graphs with boxplots for each tuning value and the corresponding number of threads (see the results images attached). After that analysis, it was notable that all of the three mechanisms with different tuning values and number of threads had a better performance than the sequential method, excluding the max tasks strategy with 4 threads. This occurred because the overhead caused by thread and task creation and management surpassed the benefits of running the tasks concurrently, leading to a worse performance.

The best tuning value returned for each granularity control mechanism was the lower value of each boxplots' mean. After executing the program 40 times, the best fine-tuning value for surplus mechanism was 3 using 4 threads, for indexing was 13 with 4 threads and for max tasks was 2 with 2 threads. To evaluate which was the best strategy with its best tuning value and in order to compare it with the sequential approach, a graph with the boxplot of the final 4 methods (sequential, surplus, index and max tasks) was created. That graph is available in 'Strategies_Comparisons.png' and highlights that the index granularity control mechanism with a fine-tuning value of 13 and 4 threads was the most efficient strategy for parallel Coin problem solving compared to the other mechanisms and the sequential approach.