PDC ASSIGNMENT 3 REPORT

SHANG-IAN TAN
A1818695

Objective

The objective of this report is to compare the performance of two parallel solvers for the N body problem using OpenMP: the basic solver and the reduced solver. This comparison aims to generate performance figures akin to those in Table 6.3 of the textbook by executing the programs on USS.cs.adelaide.edu.au, considering the system's four hardware processors. Modifications have been made to the reduced cyclic program, provided by the textbook, to match the scheduling modes used in the table.

Methodology

1. **Program Modification:**
   The reduced solver program was modified to produce results for column2 by removing any scheduling for OpenMP loops. Additionally, scheduling for all OpenMP loops was added to generate results for column4. These modifications allowed for the comparison of performance with and without specific scheduling modes.

2. **Data Collection:**
   A bash script was created to execute each program 10 times, collecting timing data for different core counts and programs. The fastest runtime from each core count and program iteration was recorded for further analysis. This approach ensured robustness in timing measurements and minimized the impact of external factors on program execution.

3. **Results Presentation:**
   The collected timing data was input into Microsoft Excel for formatting and presentation. A table was constructed to display the fastest runtime for each core count and program, allowing for easy comparison of performance between different scheduling modes and core counts. Additionally, efficiency metrics were calculated and included in the table to provide insights into the scalability of the solvers. This was calculated based on the runtimes of the basic solver for each thread count. A graph of the efficiencies has also been provided for better visualisation.

Table 1: Best Runtimes (s)

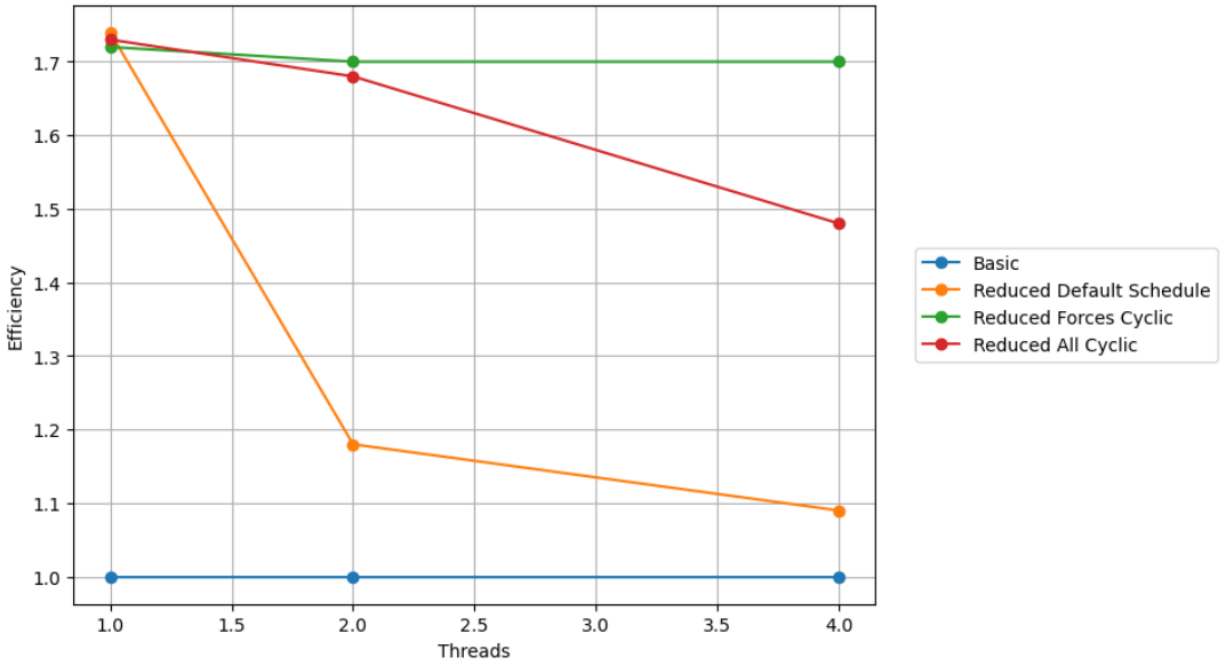
Threads	Basic	Reduced Default Sched	Reduced Forces Cyclic	Reduced All Cyclic
1	2.55	1.47	1.48	1.47
2	1.31	1.11	0.77	0.78
4	0.71	0.65	0.42	0.48

Table 2: Efficiencies (E)

Threads	Basic	Reduced Default Sched	Reduced Forces Cyclic	Reduced All Cyclic
1	1.00	1.74	1.72	1.73
2	1.00	1.18	1.70	1.68
4	1.00	1.09	1.70	1.48

Note: Runtimes rounded to 2 decimal points

Graph 1: Efficiency (E)



Results

1. At 1 thread, runtimes seem to be consistent through the three tested variants of the reduced method. All of them taking **1.47-1.48s** to run.
2. At 2 threads, the **Reduced Default Schedule** took the longest, whilst both **Reduced Forces Cyclic** & **Reduced All Cyclic** had very similar times of **0.77s & 0.78s** respectively.
3. At 4 threads, the **reduced default schedule** remained the slowest, taking **0.65s**, whilst **Reduced All Cyclic** was faster with **0.48s**, and with **Reduced Forces Cyclic** being the fastest at **0.42s**.

Conclusion

In conclusion, the analysis demonstrates that the **Reduced Forces Cyclic** method is the fastest and most efficient among the tested scheduling methods, maintaining high efficiency across varying thread counts. This method significantly outperforms the **Basic**, **Reduced Default Schedule**, and **Reduced All Cyclic** methods, showcasing superior scalability and robustness. Consequently, adopting the **Reduced Forces Cyclic** scheduling method is recommended for applications requiring high concurrency and performance.