Parallel Programming Exercise 8 – 10

Author:	許秉倫 (b05901011@ntu.edu.tw)
Student ID	B05901011
Department	電機系

(If you and your team member contribute equally, you can use (co-first author), after each name.)

1 Problem and Proposed Approach

8.10 Write a program that implements matrix-vector multiplication based on a checkerboard block decomposition of the matrix. The program should read the matrix and the vector from an input file and print the answer to standard output. The names of the files containing the matrix and the

I use the "checkerboard" method described in the textbook.

(Brief your problem, and give your idea or concept of how you design your program.)

2 Theoretical Analysis Model

Complexity Analysis (continued)

- Each process does its share of computation:
  $\Theta(n^2/p)$
- Redistribute **b**:
  $\Theta(n / \sqrt{p} + \log \sqrt{p}(n / \sqrt{p})) = \Theta(n \log p / \sqrt{p})$
- Reduction of partial results vectors:
  $\Theta(n \log p / \sqrt{p})$
- Overall parallel complexity:
  $\Theta(n^2/p + n \log p / \sqrt{p})$

Isoefficiency Analysis

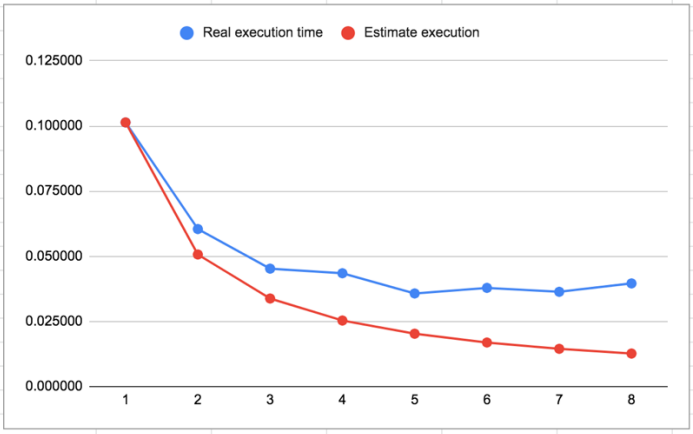
- Sequential complexity: $\Theta(n^2)$
- Parallel communication complexity: $\Theta(n \log p / \sqrt{p})$
- Isoefficiency function:
  $n^2 \geq Cn\sqrt{p}\log p \Rightarrow n \geq C\sqrt{p}\log p$
  $M\left(C\sqrt{p}\log p\right)/p = C^2 p \log^2 p / p = C^2 \log^2 p$
- This system is much more scalable than the previous two implementations

(Try to give the time complexity of the algorithm, and analyze your program with iso-efficiency metrics)

3 Performance Benchmark

(Give your idea or concept of how you design your program.)

	A	B	C	D	E	F	G	H	I
1	Processors	1	2	3	4	5	6	7	8
2	Real execution time	0.101360	0.060445	0.045247	0.043484	0.035732	0.037863	0.036356	0.039579
3	Estimate execution time	0.101360	0.050680	0.033787	0.025340	0.020272	0.016894	0.014480	0.012670
4	Speedup	X	1.676889	2.240129	2.330978	2.836663	2.676993	2.787968	2.560923
5	Karp-flatt metrics	X	0.192685	0.169604	0.238673	0.190659	0.248264	0.251798	0.303411



4 Conclusion and Discussion

1. The speedup mostly increases with the number of processors.
2. Reduce the number of computation.
3. According to the blue line, overhead counts for a large part for this program.
4. K reveals that overhead counts for a large part for this program. I reveals that this program has great scalability.

(Discuss the following issues of your program

1. What is the speedup respect to the number of processors used?
2. How can you improve your program further more
3. How does the communication and cache affect the performance of your program?
4. How does the Karp-Flatt metrics and Iso-efficiency metrics reveal?

)

Appendix(optional):

(If something else you want to append in this file, like picture of life game)