Parallel Programming Exercise 6 – 13

Author:	許秉倫 (b05901011@ntu.edu.tw) (感謝張凱捷同學協助)
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

Problem: In 1970, Princeton mathematician John Conway invented the game of Life. Life is an example of a cellular automaton. It consists of a rectangular grid of cells. Each cell is in one of two states: alive or dead The game consists of a number of iterations. During each iteration a dead cell with exactly three neighbors becomes a live cell. A live cell with two or three neighbors stays alive. A live cell with less than two neighbors or more than three neighbors becomes a dead cell. All cells are updated simultaneously. Figure 6.12 illustrates three iterations of Life for a small grid of cells. Write a parallel program that reads from a file an m x n matrix containing the initial state of the game. It should play the game of Life for *j* iterations, printing the state of the game once every *k* iterations, where *j* and *k* are command-line arguments.

Proposed Approach: 我參照張凱捷同學的做法，按 row 去切 block，由 processor 0 負責讀檔，接著把 n, m broadcast 給每個 processor，再把 board scatter 到每個 processor。

每個 iteration 中，block 會將邊界的 row 傳給上下兩個 block，如此一來就可以獲得足夠的資訊來更新 cell。

當要印出 board 時，我們將資料都 gather 到 processor 0 再一併印出。

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

2 Theoretical Analysis Model

X: 更新一個 cell 的時間
n: # of row
m: # of col
p: # of processors

$$X * T * \left(\frac{n}{p}\right) * m + T * \frac{2m}{\beta} + \lambda logp + \frac{nm}{\beta} = O\left(\frac{Tnm}{p} + \frac{Tm}{\beta} + logp + \frac{nm}{\beta}\right)$$

Iso-efficiency:

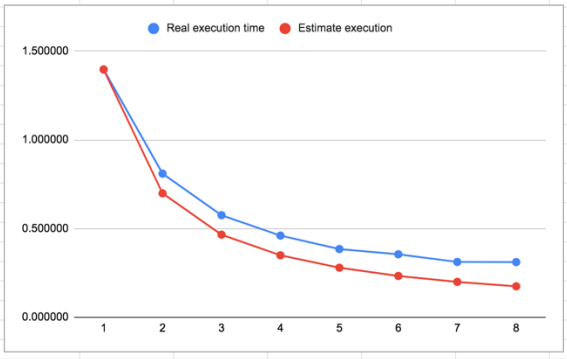
$$nmT > C * plogp$$
$$n^2 > C\ plogp$$
$$n > \sqrt{Cplogp}$$
$$\frac{M(f(p))}{p} = Clogp$$

(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	1.397027	0.809838	0.575414	0.460428	0.384603	0.355115	0.311992	0.311166
3	Estimate execution time	1.397027	0.698514	0.465676	0.349257	0.279406	0.232839	0.199576	0.174629
4	Speedup	X	1.725069	2.427865	3.034195	3.632389	3.934015	4.477760	4.489650
5	Karp-flatt metrics	X	0.159374	0.117827	0.106102	0.094126	0.105032	0.093880	0.111697



4 Conclusion and Discussion

1. Speedup 隨 processor 數而提高
2. 改成 block decomposition
3. Communication 佔的比例不少，因為每個 iteration 都要交換資料
4. 由 K 可看出 serial 佔了不少比例。Iso 可看出 scalability 還不錯。

(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)