Assignment 2: Modeling memory resources distribution on multicore processors using games on cellular automata lattices

Yu Wang, University of Ottawa, NO. 6852088

Part1 Introduction

Nowadays, multi-processors are the main stream microprocessor methodology. And the future integrated process (e.g., the three dimensional integrated process) will integrate more cores and memory in one chip, which makes the future chip more powerful ever before. In the multi-core design, though every core has a private L1 cache, which is not enough to mitigate the memory accessing latency, a much bigger shared cache (e.g., L2 cache, L3 cache) will be used to meet the computing needs of the cores. So those shared resources need carefully be divided among the cores to maximize the performance of the multi-core processor. This paper combines the game theory and cellular automata to test various memory distribution strategies.

In this paper, each core will be represented as a Cellular Automata Cell with a fixed number of CA neighbor cores, who will compete each other for the occupation of shared cache. In the game theory, the optimal choice not only depends on what I do but also on what others do. In this scenario, a core could cooperate if it doesn't use the predefined cache and abandons them while a core could defect if it needs more predefined cache. Because the resource is limited, there is a dilemma. This paper uses cellular automata to research these conflict-of-interest scenarios in multi-core shared cache division.

The cell lattice is shown in figure 1. There are 25 cells，which represent 25 cores. Every core has a state variable recording what the payoff it has now. The payoff table is shown in Table1. Every core has an output, which indicates it choice: cooperate or defect. Also the core will output its accumulated payoff to its neighbors. Every round the core collects the inputs， and use the rules(strategies) to decide it cooperates or defects in this round. There are seven rules (strategies) in this paper, though not formally defined. We could implement a few simple ones.

Table 1 core dilemma payoffs

	B		
A		Cooperate	Defect
	Cooperate	3/3	0/5
	Defect	5/0	1/1

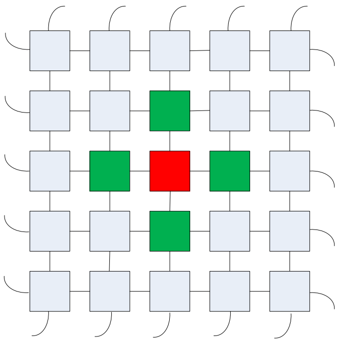

Figure 1 the cell lattices. The cell matrix is wrapped. Every cell has four neighbors. As shown, the red one has 4 green neighbors

Part2 Formal Specifications

There are three defect-cooperate strategies implemented in this cellular automata, which means three local computing rules exist. When every round begins (assume every 100 microseconds), a core can choose one of the strategies below.

- Radom strategy: Each core randomly choose to defect or cooperate.
- Pavlov strategy: Each core keep its former choice whenever it gets a high payoff, otherwise it will switch its choice. Because every core has four neighbors, we use the table 1 to calculate its total pay-offs, which is summarized in table 2.

Table 2 the payoffs under different core state (defect or cooperate) with different No. of cooperative neighbors.

Total payoffs		The cooperative neighbor No.				
		0	1	2	3	4
Core state	defect	4	8	12	16	20
	cooperate	0	3	6	9	12

If the core is in defect state and get a score less than 10, the core will switch its strategy. Otherwise, it will keep its choice.
If the core in cooperate state and get a score less than 5, the core will switch its strategy. Otherwise, it will keep its choice.

- Tit-Tat strategy: Each core will follow the state most of its neighbors have.
If the No. of the cooperative neighbors is below 2, the core will choose to defect.
If the No. of the cooperative neighbors is equal 2, the core will choose to keep its choice.
If the No. of the cooperative neighbors is above 2, the core will choose to cooperate.

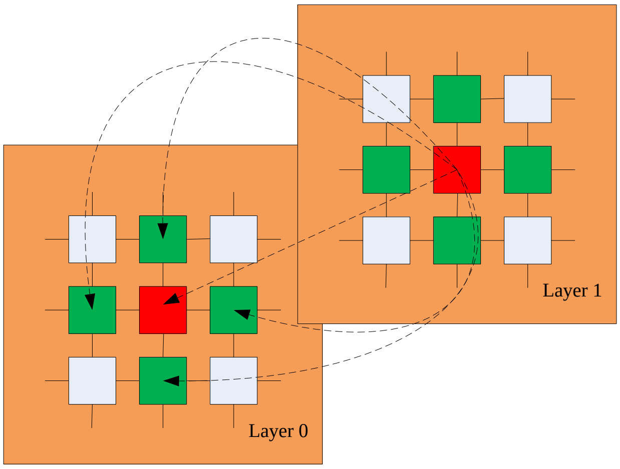

Figure 2 cellular automata

There are two layers in this cellular automata (Figure 2). One layer show every core's state while another layer shows its score last round. The local computing rule of the score layer is to sum the payoffs according to its neighbors 'choices using table 1. The following are the formal specifications.

Layer 0 are cores whose state is defect or cooperate.

Atomic model

CD=<X, Y, I, S, θ, N, d, δint, δext,τ, λ, D>
X=Y=S={0,1}; // 0 stands for cooperate while 1 stands for defect
d = transport delay, 100msec
τ = (the strategy described above includes random strategy, Pavlov strategy, and tit-tat strategy)
I, θ, δint, δext, λ, D is implemented in accordance with Cell DEVS definitions.

Coupled model

GCC = < Xlist, Ylist, I, X, Y, η, N, {r, c}, C, B, Z, select >
Xlist=Φ
Ylist=Φ
I=<PX, PY>, with PX={Φ},PY={Φ};
X=Φ
Y=Φ
η=5
N={ (-1,0,0),(0,-1,0),(0,1,0),(1,0,0),(0,0,0)}
r=5; c=5;
B={Φ};
C={Cij | i∈[0,4], j∈[0,4]}
Z: defined in accordance with coupled Cell DEVS
Select ={(-1,0,0),(0,-1,0),(0,1,0),(1,0,0),(0,0,0) }

Plane 1 is to record the underlying core's score last round. Its neighbors could be seen in the figure 2. Every cell in this plane has five neighbors, all of whom are located in the below plane.

It will calculate the total payoffs of the core below using the table 1. Table 1 analyzes several scenarios. Under different scenario, a payoff will be calculated. The state of cell in this plane only have discontinuous integers for the calculations under different scenarios.

Atomic model

CD=<X, Y, I, S, θ, N, d, δint, δext,τ, λ, D>
X=Y=S={0,3,4,6,8,9,12,16,20}; // 0 stands for cooperate while 1 stands for defect
d = transport delay, 100msec
τ = (the described strategy above)
I, θ, δint, δext, λ, D is implemented in accordance with Cell DEVS definitions.

Coupled model

GCC = < Xlist, Ylist, I, X, Y, η, N, {r, c}, C, B, Z, select >
Xlist=Φ
Ylist=Φ
I=<PX, PY>, with PX={Φ},PY={Φ};
X=Φ
Y=Φ
η=6
N={ (-1,0,-1),(0,-1,-1),(0,1,-1),(1,0,-1),(0,0,-1) ,(0,0,0)}
r=5; c=5;
B={Φ};
C={Cij | i∈[0,4], j∈[0,4]}
Z: defined in accordance with coupled Cell DEVS
Select ={(-1,0,-1),(0,-1,-1),(0,1,-1),(1,0,-1),(0,0,-1) ,(0,0,0) }

Part3 Test and Final Result

Test the random strategy. We use the following layout. In the layout below, every core randomly defect or cooperate.

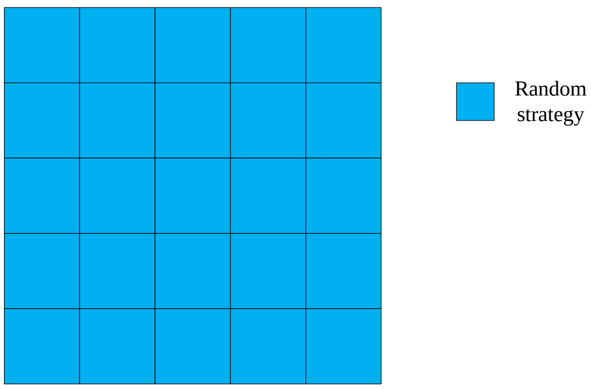

Figure 3 random strategy test layout.

Using the simu tool, we can generate the *.log and then drawlog the result (Figure4). Every round (100 microseconds), every core will randomly defect or cooperate. Initially, every core cooperate, which show zero in all cells at time 0. Every round the layer 1 will give the score last round. It is easy to conclude that the model executes correctly.

```
Line : 146 - Time: 00:00:00:000
            0         1         2         3         4                0         1         2         3         4
    +--------------------------------------------------+     +--------------------------------------------------+
   0|                                                  |    0|                                                  |
   1|                                                  |    1|                                                  |
   2|                                                  |    2|                                                  |
   3|                                                  |    3|                                                  |
   4|                                                  |    4|                                                  |
    +--------------------------------------------------+     +--------------------------------------------------+

Line : 356 - Time: 00:00:00:100
            0         1         2         3         4                0         1         2         3         4
    +--------------------------------------------------+     +--------------------------------------------------+
   0|     1.000     1.000     1.000     1.000     1.000|    0|    12.000    12.000    12.000    12.000    12.000|
   1|     1.000               1.000                    |    1|    12.000    12.000    12.000    12.000    12.000|
   2|                         1.000     1.000          |    2|    12.000    12.000    12.000    12.000    12.000|
   3|     1.000                                   1.000|    3|    12.000    12.000    12.000    12.000    12.000|
   4|     1.000                                   1.000|    4|    12.000    12.000    12.000    12.000    12.000|
    +--------------------------------------------------+     +--------------------------------------------------+

Line : 552 - Time: 00:00:00:200
            0         1         2         3         4                0         1         2         3         4
    +--------------------------------------------------+     +--------------------------------------------------+
   0|     1.000     1.000               1.000          |    0|     4.000    12.000     8.000    12.000     8.000|
   1|               1.000                              |    1|    16.000     3.000    12.000     3.000     6.000|
   2|     1.000     1.000     1.000     1.000          |    2|     6.000     9.000    12.000    16.000     6.000|
   3|                         1.000     1.000     1.000|    3|    12.000     9.000     9.000     6.000    12.000|
   4|     1.000               1.000               1.000|    4|     8.000     6.000     9.000     6.000     8.000|
    +--------------------------------------------------+     +--------------------------------------------------+
```

Figure 4 random strategy test result.

Test the Pavlov strategy. We use the following layout to test the Pavlov strategy.

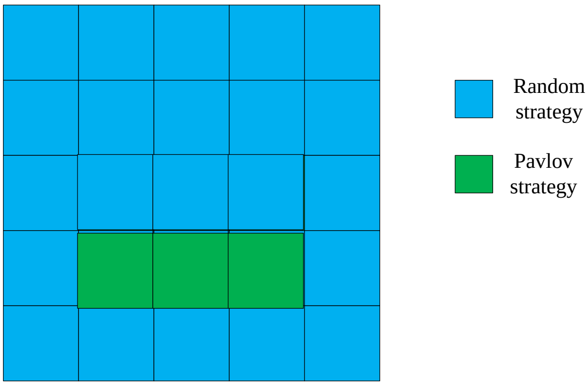

Figure 5 Test Pavlov strategy Layout

We get the following result, the result (Figure6). Every round (100 microseconds), the sky-blue core will randomly defect or cooperate. The Pavlov core in the shaded box will act in accordance with its strategy.

```
Line : 145 - Time: 00:00:00:000
            0         1         2         3         4                0         1         2         3         4
    +--------------------------------------------------+     +--------------------------------------------------+
   0|                                                  |    0|                                                  |
   1|                                                  |    1|                                                  |
   2|                                                  |    2|                                                  |
   3|                                                  |    3|                                                  |
   4|                                                  |    4|                                                  |
    +--------------------------------------------------+     +--------------------------------------------------+

Line : 354 - Time: 00:00:00:100
            0         1         2         3         4                0         1         2         3         4
    +--------------------------------------------------+     +--------------------------------------------------+
   0|     1.000     1.000     1.000     1.000     1.000|    0|    12.000    12.000    12.000    12.000    12.000|
   1|     1.000               1.000                    |    1|    12.000    12.000    12.000    12.000    12.000|
   2|                         1.000     1.000          |    2|    12.000    12.000    12.000    12.000    12.000|
   3|     1.000                                        |    3|    12.000    12.000    12.000    12.000    12.000|
   4|                         1.000     1.000          |    4|    12.000    12.000    12.000    12.000    12.000|
    +--------------------------------------------------+     +--------------------------------------------------+
```

Figure 6 Test Pavlov strategy result: keep its choice

We can see initially the Pavlov cores keep its choice because it get a relative high score. But under certain scenarios, it will change (Figure 7).

```
Line : 1113 - Time: 00:00:00:500
             0         1         2         3         4                 0         1         2         3         4
    +--------------------------------------------------+    +--------------------------------------------------+
   0|                                                  |   0|     9.000     6.000    12.000    16.000     6.000|
   1|     1.000     1.000                         1.000|   1|     8.000     8.000     8.000              12.000|
   2|     1.000     1.000                         1.000|   2|     4.000     8.000     8.000    12.000     4.000|
   3|     1.000                                   1.000|   3|    12.000     6.000     9.000     6.000    12.000|
   4|               1.000     1.000     1.000          |   4|     9.000    12.000     9.000     9.000     9.000|
    +--------------------------------------------------+    +--------------------------------------------------+

Line : 1310 - Time: 00:00:00:600
             0         1         2         3         4                 0         1         2         3         4
    +--------------------------------------------------+    +--------------------------------------------------+
   0|               1.000     1.000     1.000     1.000|   0|     9.000     6.000     9.000     9.000     9.000|
   1|     1.000     1.000                              |   1|     8.000    12.000     9.000     9.000    12.000|
   2|                         1.000     1.000     1.000|   2|     4.000    12.000     9.000     9.000     8.000|
   3|     1.000     1.000                              |   3|    12.000     3.000     9.000     6.000    12.000|
   4|                                   1.000          |   4|     6.000    16.000    12.000    16.000     6.000|
    +--------------------------------------------------+    +--------------------------------------------------+
```

Figure 7 Test Pavlov strategy result: change its choice

Test the Tit-Tat strategy and also the Final test. We use the following layout.

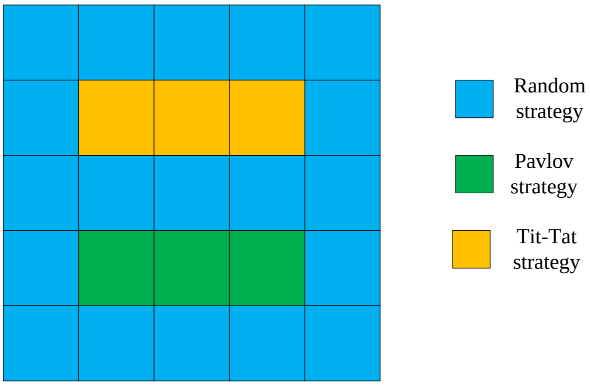

Figure 8 final test layout

The result is shown in Figure 9. We can see that the core choose to defect or cooperate following most neighbor's choices.

```
Line : 1297 - Time: 00:00:00:600
             0         1         2         3         4                 0         1         2         3         4
    +--------------------------------------------------+    +--------------------------------------------------+
   0|     1.000               1.000     1.000     1.000|   0|     3.000    16.000     6.000    16.000    12.000|
   1|                                             1.000|   1|     6.000     6.000     9.000     6.000    16.000|
   2|     1.000               1.000     1.000          |   2|    12.000    12.000    16.000     9.000     6.000|
   3|     1.000     1.000                         1.000|   3|    12.000     3.000     9.000    12.000     9.000|
   4|     1.000               1.000                    |   4|    12.000    12.000     9.000     9.000     6.000|
    +--------------------------------------------------+    +--------------------------------------------------+

Line : 1490 - Time: 00:00:00:700
             0         1         2         3         4                 0         1         2         3         4
    +--------------------------------------------------+    +--------------------------------------------------+
   0|               1.000                              |   0|    12.000     6.000    12.000    12.000     8.000|
   1|                                   1.000     1.000|   1|     3.000    12.000     6.000     3.000    16.000|
   2|     1.000               1.000     1.000          |   2|    16.000     3.000    16.000    16.000          |
   3|     1.000     1.000     1.000               1.000|   3|     4.000    16.000     3.000     6.000    16.000|
   4|     1.000                                   1.000|   4|    12.000     3.000    16.000     6.000     3.000|
    +--------------------------------------------------+    +--------------------------------------------------+
```

Figure 9 final test result

In conclusion, in this assignment, I implement three simple strategies described in the paper, and finish doing some simulations.