HW2 Butterfly Barrier

MSIM 406 – Distributed Simulation

Thomas Laverghetta,
Old Dominion University – Computation Modeling and Simulation Engineering

Design

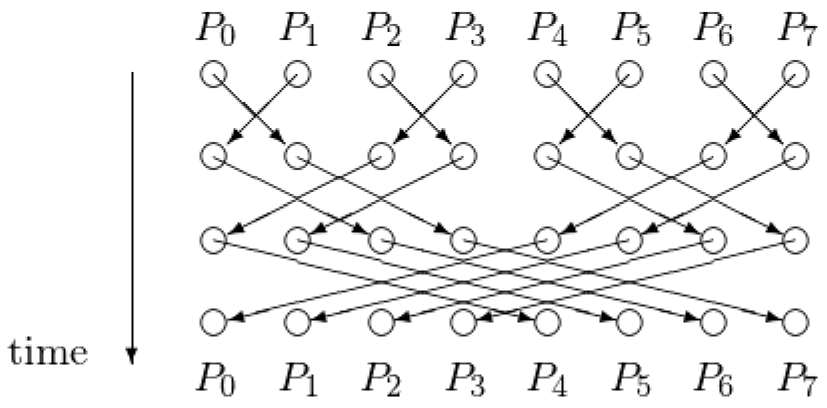


Figure 1. Butterfly Barrier

http://homepages.math.uic.edu/~jan/mcs572/mcs572notes/lec18.html

For this homework, I am to implement a butterfly barrier. As seen in figure 1, a butterfly barrier is a multi-layer synchronization method which groups processor information to determine when it is safe to do another time-step. The algorithm is shown below:

```
Butterfly Barrier ()
1. Get number of processors ($N_p$)
2. Get this process id ($P_{id}$)
3. for bitflip ($b_f$) from 0 to $\log_2 N_p$:
4.        calculate next process ($P_n$) using $P_{id} \oplus (1 \ll b_f)$
5.        send done msg to $P_n$
6.        wait for response msg from $P_n$
7.end for
```

Testing

To test the butterfly barrier, I designed and implemented a test-harness. The test-harness places a random work delay on each processor to see if butterfly barrier will not allow processors to advance until all processors are done with work.

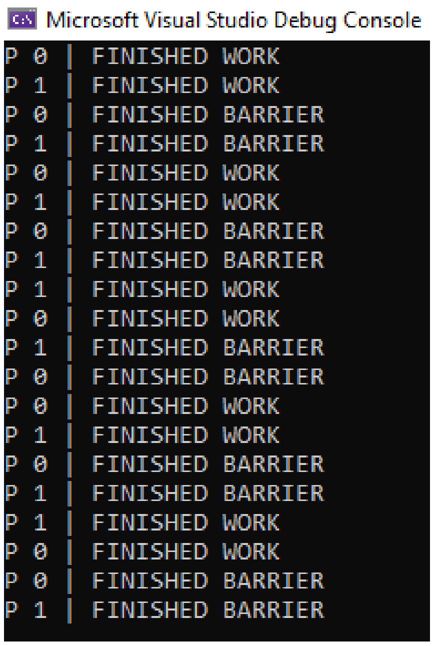
```
Test Harness ()
1. Init MPI
2. Get this process id
3. Get number of processors in system
4. Set random seed to this process ID
5. for number of work iteration:
6.        calculate random time delay
7.        delay thread for random time delay
8.        Butterfly Barrier
9. end for
```

Test Result

I ran serval tests using 2, 4, 8, and 16 processors. Each test I did 5-terations work and butterfly barrier with random work delay between 2 and 8. Within each iteration, I display when the processor has finished work and finished barrier. If the barrier is properly implemented, then you should see all finish work messages then all finish barrier messages.

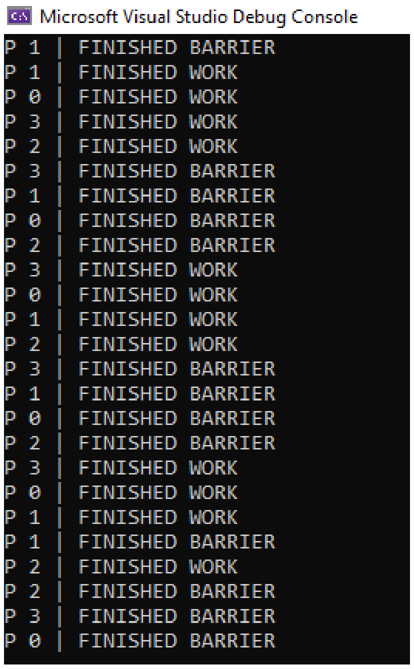
The following are the test results. Note, in the output, P-values is the processor ID.

As seen in the following figures (2, 3, 4, 5), the barrier preformed as expected. Therefore, butterfly barrier was implemented correctly.



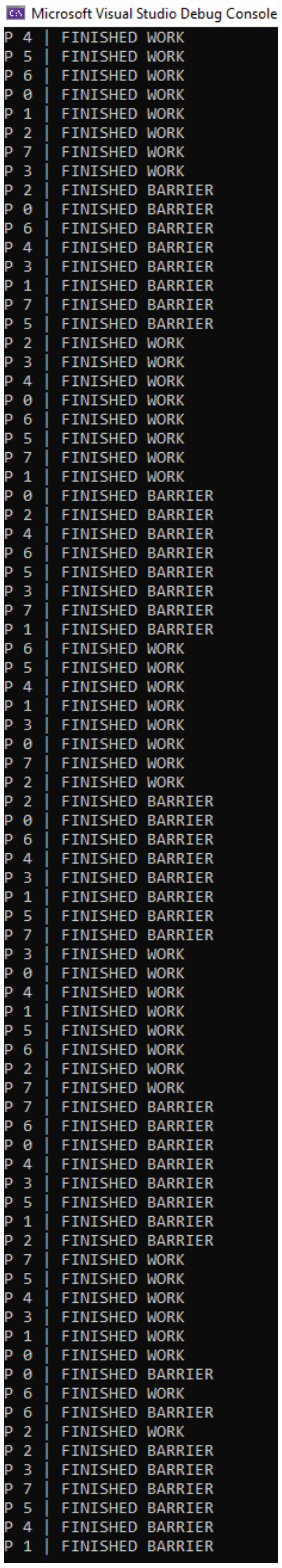
```
P 0 | FINISHED WORK
P 1 | FINISHED WORK
P 0 | FINISHED BARRIER
P 1 | FINISHED BARRIER
P 0 | FINISHED WORK
P 1 | FINISHED WORK
P 0 | FINISHED BARRIER
P 1 | FINISHED BARRIER
P 1 | FINISHED WORK
P 0 | FINISHED WORK
P 1 | FINISHED BARRIER
P 0 | FINISHED BARRIER
P 0 | FINISHED WORK
P 1 | FINISHED WORK
P 0 | FINISHED BARRIER
P 1 | FINISHED BARRIER
P 1 | FINISHED WORK
P 0 | FINISHED WORK
P 0 | FINISHED BARRIER
P 1 | FINISHED BARRIER
```

Figure 2. 2-Processors


Microsoft Visual Studio Debug Console

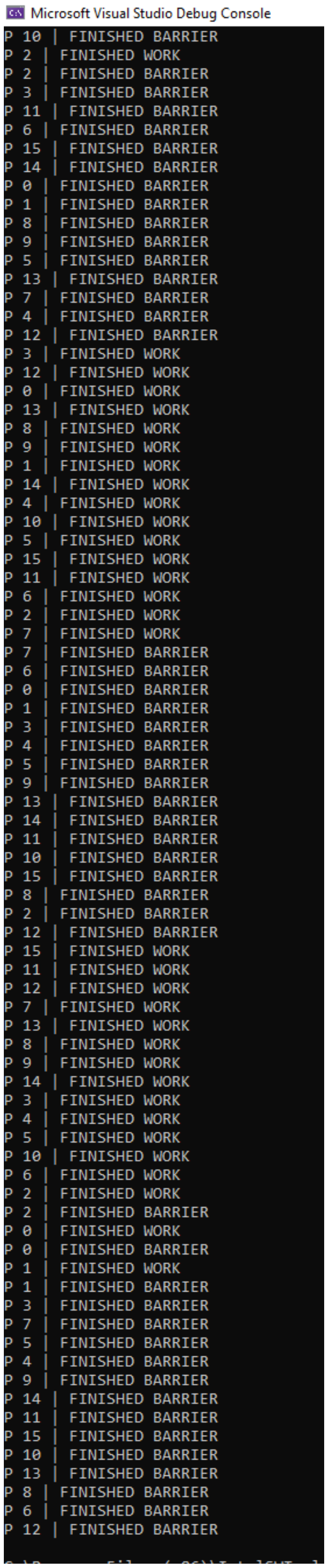
```
P 1 | FINISHED BARRIER
P 1 | FINISHED WORK
P 0 | FINISHED WORK
P 3 | FINISHED WORK
P 2 | FINISHED WORK
P 3 | FINISHED BARRIER
P 1 | FINISHED BARRIER
P 0 | FINISHED BARRIER
P 2 | FINISHED BARRIER
P 3 | FINISHED WORK
P 0 | FINISHED WORK
P 1 | FINISHED WORK
P 2 | FINISHED WORK
P 3 | FINISHED BARRIER
P 1 | FINISHED BARRIER
P 0 | FINISHED BARRIER
P 2 | FINISHED BARRIER
P 3 | FINISHED WORK
P 0 | FINISHED WORK
P 1 | FINISHED WORK
P 1 | FINISHED BARRIER
P 2 | FINISHED WORK
P 2 | FINISHED BARRIER
P 3 | FINISHED BARRIER
P 0 | FINISHED BARRIER
```

Figure 3. 4-Processor


Microsoft Visual Studio Debug Console

```
P 4 | FINISHED WORK
P 5 | FINISHED WORK
P 6 | FINISHED WORK
P 0 | FINISHED WORK
P 1 | FINISHED WORK
P 2 | FINISHED WORK
P 7 | FINISHED WORK
P 3 | FINISHED WORK
P 2 | FINISHED BARRIER
P 0 | FINISHED BARRIER
P 6 | FINISHED BARRIER
P 4 | FINISHED BARRIER
P 3 | FINISHED BARRIER
P 1 | FINISHED BARRIER
P 7 | FINISHED BARRIER
P 5 | FINISHED BARRIER
P 2 | FINISHED WORK
P 3 | FINISHED WORK
P 4 | FINISHED WORK
P 0 | FINISHED WORK
P 6 | FINISHED WORK
P 5 | FINISHED WORK
P 7 | FINISHED WORK
P 1 | FINISHED WORK
P 0 | FINISHED BARRIER
P 2 | FINISHED BARRIER
P 4 | FINISHED BARRIER
P 6 | FINISHED BARRIER
P 5 | FINISHED BARRIER
P 3 | FINISHED BARRIER
P 7 | FINISHED BARRIER
P 1 | FINISHED BARRIER
P 6 | FINISHED WORK
P 5 | FINISHED WORK
P 4 | FINISHED WORK
P 1 | FINISHED WORK
P 3 | FINISHED WORK
P 0 | FINISHED WORK
P 7 | FINISHED WORK
P 2 | FINISHED WORK
P 2 | FINISHED BARRIER
P 0 | FINISHED BARRIER
P 6 | FINISHED BARRIER
P 4 | FINISHED BARRIER
P 3 | FINISHED BARRIER
P 1 | FINISHED BARRIER
P 5 | FINISHED BARRIER
P 7 | FINISHED BARRIER
P 3 | FINISHED WORK
P 0 | FINISHED WORK
P 4 | FINISHED WORK
P 1 | FINISHED WORK
P 5 | FINISHED WORK
P 6 | FINISHED WORK
P 2 | FINISHED WORK
P 7 | FINISHED WORK
P 7 | FINISHED BARRIER
P 6 | FINISHED BARRIER
P 0 | FINISHED BARRIER
P 4 | FINISHED BARRIER
P 3 | FINISHED BARRIER
P 5 | FINISHED BARRIER
P 1 | FINISHED BARRIER
P 2 | FINISHED BARRIER
P 7 | FINISHED WORK
P 5 | FINISHED WORK
P 4 | FINISHED WORK
P 3 | FINISHED WORK
P 1 | FINISHED WORK
P 0 | FINISHED WORK
P 0 | FINISHED BARRIER
P 6 | FINISHED WORK
P 6 | FINISHED BARRIER
P 2 | FINISHED WORK
P 2 | FINISHED BARRIER
P 3 | FINISHED BARRIER
P 7 | FINISHED BARRIER
P 5 | FINISHED BARRIER
P 4 | FINISHED BARRIER
P 1 | FINISHED BARRIER
```

Figure 4. 8-Processors


Microsoft Visual Studio Debug Console

```
P 10 | FINISHED BARRIER
P 2 | FINISHED WORK
P 2 | FINISHED BARRIER
P 3 | FINISHED BARRIER
P 11 | FINISHED BARRIER
P 6 | FINISHED BARRIER
P 15 | FINISHED BARRIER
P 14 | FINISHED BARRIER
P 0 | FINISHED BARRIER
P 1 | FINISHED BARRIER
P 8 | FINISHED BARRIER
P 9 | FINISHED BARRIER
P 5 | FINISHED BARRIER
P 13 | FINISHED BARRIER
P 7 | FINISHED BARRIER
P 4 | FINISHED BARRIER
P 12 | FINISHED BARRIER
P 3 | FINISHED WORK
P 12 | FINISHED WORK
P 0 | FINISHED WORK
P 13 | FINISHED WORK
P 8 | FINISHED WORK
P 9 | FINISHED WORK
P 1 | FINISHED WORK
P 14 | FINISHED WORK
P 4 | FINISHED WORK
P 10 | FINISHED WORK
P 5 | FINISHED WORK
P 15 | FINISHED WORK
P 11 | FINISHED WORK
P 6 | FINISHED WORK
P 2 | FINISHED WORK
P 7 | FINISHED WORK
P 7 | FINISHED BARRIER
P 6 | FINISHED BARRIER
P 0 | FINISHED BARRIER
P 1 | FINISHED BARRIER
P 3 | FINISHED BARRIER
P 4 | FINISHED BARRIER
P 5 | FINISHED BARRIER
P 9 | FINISHED BARRIER
P 13 | FINISHED BARRIER
P 14 | FINISHED BARRIER
P 11 | FINISHED BARRIER
P 10 | FINISHED BARRIER
P 15 | FINISHED BARRIER
P 8 | FINISHED BARRIER
P 2 | FINISHED BARRIER
P 12 | FINISHED BARRIER
P 15 | FINISHED WORK
P 11 | FINISHED WORK
P 12 | FINISHED WORK
P 7 | FINISHED WORK
P 13 | FINISHED WORK
P 8 | FINISHED WORK
P 9 | FINISHED WORK
P 14 | FINISHED WORK
P 3 | FINISHED WORK
P 4 | FINISHED WORK
P 5 | FINISHED WORK
P 10 | FINISHED WORK
P 6 | FINISHED WORK
P 2 | FINISHED WORK
P 2 | FINISHED BARRIER
P 0 | FINISHED WORK
P 0 | FINISHED BARRIER
P 1 | FINISHED WORK
P 1 | FINISHED BARRIER
P 3 | FINISHED BARRIER
P 7 | FINISHED BARRIER
P 5 | FINISHED BARRIER
P 4 | FINISHED BARRIER
P 9 | FINISHED BARRIER
P 14 | FINISHED BARRIER
P 11 | FINISHED BARRIER
P 15 | FINISHED BARRIER
P 10 | FINISHED BARRIER
P 13 | FINISHED BARRIER
P 8 | FINISHED BARRIER
P 6 | FINISHED BARRIER
P 12 | FINISHED BARRIER
```

Figure 5. 16-Processors

Application

To demonstrate the butterfly barrier, a 2D gravitational simulation will be applied. A 2D gravitational simulation application is supplied by Dr. Jim Leathrum. The application supplied was implemented with a ring-barrier. So, I replaced the ring-barrier with butterfly-barrier for my testing. If everything is correct, all barriers will produce the same outputs.

During the demonstration, process time will be collected to compare between ring-, MPI-, and butterfly-barriers. One of the goals of doing this demonstration is determine the performance between a butterfly-barrier compared to a ring-barrier and MPI-barrier. To do this, I will be collecting processing time for different number of processors used. Starting with one processor to 16-processors by factors of 2 (2, 4, 8, 16-processors). Also, to conduct further analysis of process times, I will be conducting batch runs of 30-sampels for each number of processors. This way I can calculate averages and standard deviations.

Results

Using a python script to run batch and collect process times and Microsoft PowerShell diff command to compare outputs between barrier techniques, I was able to determine (1) all barrier implementations produced the same output data from application and (2) performance of barriers. As seen in table 1 below, the ring-barrier was the slowest across the board; butterfly was the second best overall only being first using 8-processors; and MPI was the best overall being first in all processors except for 8-processors.

Table 1. Process Time Data

#Processors	MPI (s)		CI (α=0.90)	Ring (s)		CI (α=0.90)	Butterfly (s)		CI (α=0.90)
1	0.772606	±	0.005335469	0.785791	±	0.014063872	0.813857913	±	0.011610002
2	0.915523	±	0.004087498	0.888817	±	0.005318429	0.859311732	±	0.004018969
4	1.377977	±	0.006905183	1.879055	±	0.017384714	1.327896682	±	0.006788737
8	2.238902	±	0.00973874	3.61291	±	0.014902388	2.200712196	±	0.011809743
16	4.823631	±	0.021871117	7.39415	±	0.027801372	5.010173011	±	0.029320037

Data can be found in BarrierProcessTimeDataCollected.xlsx