Programming Assignment 4

Comparison of MRMW Obstruction Free

And

MRMW wait Free Snapshot

Name : Rohit Kapoor
Roll No: cs21mtech12011

Name : Rohit Kapoor Roll No: cs21mtech12011

Program Design

The given program takes input from the inp-params.txt file and creates n_s snapshot and n_w writer threads. Each writer thread attempts to perform a write operation on the shared set of registers while each snapshot thread, scans the entire set of registers and checks whether a clean collect is obtained or not.

There are two files in the program and each produces an output file in order to log the timestamps at which the values were written to the registers and the timestamps at which the snapshots were completed.

Algorithms

MRMW Obstruction Free

- In obstruction free algorithm, the snapshot is obtained by checking both the timestamp and the value at each position of the register array.

- The thread keeps on making repeated calls to the scan function until a clean collect is obtained (i.e. all the values are matched.)

- Once a clean collect is obtained it is returned as the result.

- In this case the snapshot threads can starve, if the writer thread does not allow them to complete the snapshot.

MRMW Wait Free

- In the wait Free Implementation of the snapshot algorithm, the scan is obtained by both the writer and the snapshot threads and the corresponding id is stored in the Helpsnap vector.

- If at any point the id associated with the snapshot threads is found in the Helpsnap, then we can directly return the value associated with the Helpsnap vector, this helps the thread in obtaining a snapshot taken by the writer threads and hence saves time.

- Since each thread performs a scan operation, hence the program terminates in a finite no of steps.

Name : Rohit Kapoor Roll No: cs21mtech12011

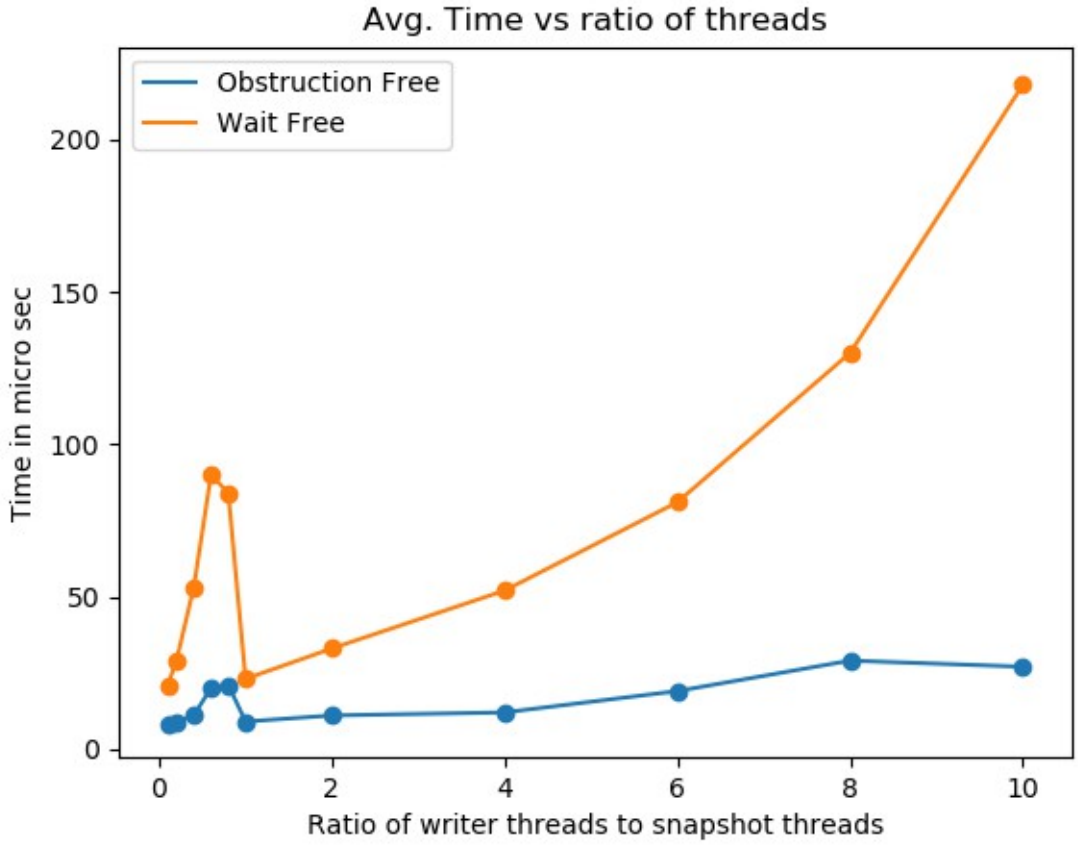
Comparison of performance

The average delays and the worst case delays are evaluated by varying the values of the no of writer threads and the number of snapshot threads while keeping the other parameters constant. For obtaining the results in this algorithm, the values of us and uw are set as 0.5 each. K is set as 3, the no of locations in the register M are set to 20. The data has been tabulated in the given table.

nw	ns	ratio	Obs. Free (avg. time) mico. sec.	Obs. Free (worst time) mico. sec.	Wait free (avg. time) mico. sec.	Wait free (worst time) mico. sec.
30	3	10	27	183	218	643
24	3	8	29	178	130	369
18	3	6	19	100	81	232
12	3	4	12	23	52	95
6	3	2	11	42	33	73
3	3	1	9	36	23	55
24	30	0.8	21	482	84	782
18	30	0.6	20	305	90	499
12	30	0.4	11	57	53	171
6	30	0.2	9	38	29	91
3	30	0.1	8	38	21	58

Name : Rohit Kapoor Roll No: cs21mtech12011

Graph for Avg delay vs. Ratio of threads



Observations:

1. It can be observed that the obstruction free algorithm is performing better than the wait free algorithm.

- The reason is that the obstruction free algorithm takes a collect if it matches with the new collect then it directly return the snapshot otherwise it simply takes a new copy, while in wait free algorithm, if a set of writes are performed in between the first and the second collect then the algorithm might have to wait for a particular no of writes to complete before it can make a new collect or steal some other thread's collect.

- In the wait free algorithm, a write step involves both write and scan steps, so a write takes much more time to complete and hence adding more delay two snapshot threads.

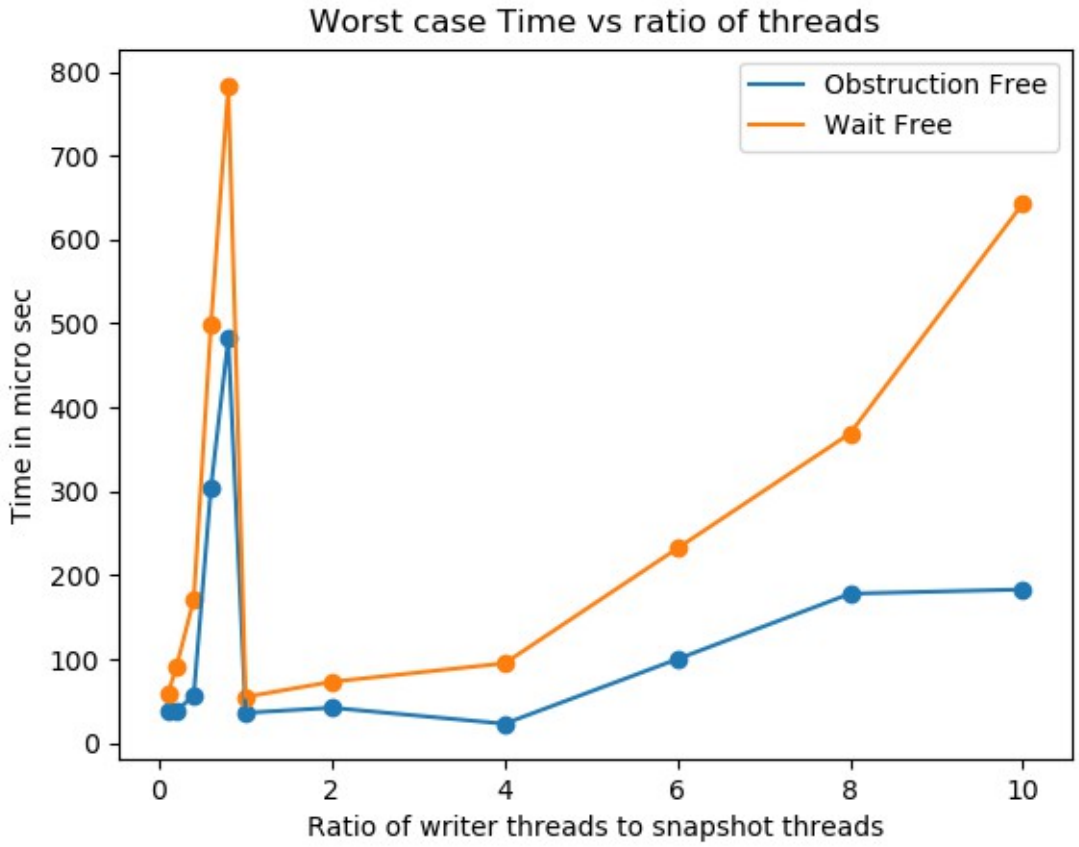
- Checking in the set of id's which have written in between two collects also takes time.

2. There is a slight peak at ratio 0.6 and 0.8.

3. When the ratio is positive the time increases.

The reason behind this observation is when the no of writer threads are significantly more than the no of snapshot threads then it will be more likely that the a writer thread performs a read between two snapshot threads.

Graph For Worst Case delay vs. Ratio of Threads



Observations

1. The values obtained in each case are significantly higher than the avg. times for the two algorithms.

2. A similar trend to that of avg. case is followed in terms of the delay for obstruction free and wait free for the reasons explained in previous section.

3. With the increase in the number of writer threads the time taken increases.

4. Values peak when the ratio is 0.8

Name : Rohit Kapoor Roll No: cs21mtech12011