

Operating Systems - II (CS3523)

Mar 2, 2025

Programming Assignment 2: Dynamic Validation of Sudoku

Test and Set (TAS)-

→The purpose of this assignment is to implement a multithreaded sudoku validator using Cpp with pthread library. The threads are handled using the Test-and-Set(TAS) lock mechanism.

→The program reads the Sudoku grid from an input file (`input.txt`) and checks with multiple threads to check the correctness of the Sudoku. A shared atomic counter (`C`) ensures task allocation among threads, while an atomic flag is used for locking mechanisms.

→The program spawns `3*K` threads to check rows and columns in parallel.

→Atomic variables are used to ensure synchronization of the shared counter and Sudoku validity status.

→Dynamic task assignment based on `taskInc` for dividing the work properly.

About IMPLEMENTATION:

→Each thread increments the atomic counter (`C.fetch_add(taskInc)`) to claim a task.

→If a thread ID is less than `K`, it verifies rows. Otherwise, it checks columns. The validity of a row/column is determined using a `visited` array to track numbers encountered.

→Synchronization Using TAS

1) `TAS_lock(int id)`: A thread requests the critical section (CS) and records entry time.

2) `TAS_unlock(int id)`: The thread releases the CS and records exit time.

→If an invalid row/column is found, the `sudoku_valid` atomic flag is set to `false`, and threads terminate early.

Compare-and-swap(CAS)

The main features and components are almost the same as in the TAS algorithm.

except, Each thread increments the atomic counter (`C.fetch_add(taskInc)`) to claim a task.

→ **Synchronization Using CAS:**

1) `CAS_lock(int id)`: A thread requests the critical section (CS) and records entry time.

2) `CAS_unlock(int id)`: The thread releases the CS and records exit time.

Bounded CAS

The main features and components are almost the same as in the TAS algorithm.

About Programs

→ The programs records various performance metrics:

1) **Total execution time.**

2) **Average time to enter CS.**

3) **Average time to exit CS.**

4) **Worst-case time for entering CS.**

5) **Worst-case time for exiting CS.**

→ If any row or column contains duplicate numbers, the output gives `The sudoku is invalid.`

→ If the Sudoku grid is valid, the output gives `The sudoku is valid.`

→This multithreaded Sudoku checking is done efficiently among threads while ensuring mutual exclusion through all the **Bounded CAS ,CAS,TAS**operations.

TABLES(Experiment 1)-

Average Case Entry Tables:

S.No	X axis (Sudo ku Size)	TAS Avg CS Entry time (in micro sec)	CAS Avg CS Entry time (in micro sec)	Bound ed CAS Avg CS Entry time (in micro sec)	TAS Avg CS Exit time (in micro sec)	CAS Avg CS Exit time (in micro sec)	Bound ed CAS Avg CS Exit time (in micro sec)	Total time taken (in micro sec)
1	400	8160	11051	2692	2	0.71	0.5	35066, 50173, 12003
2	900	7910	15398. 7	1808	1.87	0.4402 52	0.34	61213, 11229 1,1491 9
3	1600	13165	12074	875	0.93	0.56	0.28	15819 3,1464 82,123 29

4	2500	10011	17719	818	1.32	0.30	0.19	18271 5,3175 30,177 03
5	3600	11927	16973	872	1.31	0.42	0.15	31063 5,4295 48,279 69

Worst Case Entry Tables:

S.No	X axis (Sudo ku Size)	TAS worst CS Entry time	CAS worst CS Entry time	Bound ed worst CS Entry	TAS worst CS Exit time	CAS worst CS Exit time	Bound ed CAS worst CS	Total time taken (in micro
------	---------------------------------	-------------------------------------	-------------------------------------	-------------------------------------	------------------------------------	------------------------------------	-----------------------------------	--

		(in micro sec)	(in micro sec)	time (in micro sec)	(in micro sec)	(in micro sec)	Exit time (in micro sec)	sec)
1	400	30623	48142	9869	11	8	0.90	35066, 50173, 12003
2	900	57277	10728 1	9243	10	4	0.49	61213, 11229 1,1491 9
3	1600	10068 3	11846 2	7431	7	5	0.40	15819 3,1464 82,123 29
4	2500	90680	16708 5	6905	9	6	0.37	18271 5,3175 30,177 03
5	3600	10208 6	20094 0	7947	8	5	0.31	31063 5,4295 48,279 69

TABLES(Experiment 2)-

Taking 50 square and doing because for 90 square my laptop is crashing isn't supporting. Num of threads = 8

Average Case Entry Tables:

S.No	X axis (taskIn c)	TAS Avg CS Entry time (in micro sec)	CAS Avg CS Entry time (in micro sec)	Bound ed CAS Avg CS Entry time (in micro sec)	TAS Avg CS Exit time (in micro sec)	CAS Avg CS Exit time (in micro sec)	Bound ed CAS Avg CS Exit time (in micro sec)	Total time taken (in micro sec)
1	10	7662	9819	444	0.64	0.27	0.10	27295 0,3433 5,1841 2
2	20	10011	17719	818	1.32	0.30	0.19	18271 5,3175 30,177 03
3	30	15038	13571	511	0.489	0.19	0.11	18847 9,1718 72,787 9
4	40	4106	11955	477	0.70	0.15	0.07	43753, 11691 8,6156
5	50	11935	15524	555	0.49	0.09	0.10	96583, 12175 4,5714

Worst Case Entry Tables:

S.No	X axis (taskInc)	TAS worst CS Entry time (in micro sec)	CAS worst CS Entry time (in micro sec)	Bound ed CAS worst CS Entry time (in micro sec)	TAS worst CS Exit time (in micro sec)	CAS worst CS Exit time (in micro sec)	Bound ed CAS worst CS Exit time (in micro sec)	Total time taken (in micro sec)
1	10	73906	81586	3588	5	3	0.80	27295 0,3433 5,1841 2
2	20	90680	16708 5	6905	9	6	0.37	18271 5,3175 30,177 03
3	30	14196 5	12688 8	3554	4	2	0.29	18847 9,1718 72,787 9

4	40	12820 1	94673	3376	5	2	0.24	43753, 11691 8,6156
5	50	91288	12089 2	4391	3	2	0.19	96583, 12175 4,5714

TABLES(Experiment 3)-

Taking 50 square and doing because for 90 square my laptop is crashing isn't supporting.taskInc = 50

Average Case Entry Tables:

S.No	X axis (threads)	TAS Avg CS Entry time (in micro sec)	CAS Avg CS Entry time (in micro sec)	Bound ed CAS Avg CS Entry time (in micro sec)	TAS Avg CS Exit time (in micro sec)	CAS Avg CS Exit time (in micro sec)	Bound ed CAS Avg CS Exit time (in micro sec)	Total time taken (in micro sec)
1	1	5.98	5.73	7.26	0.0	0.0	0.0	4901,4

								724,42 09
2	2	25	20.859	39	0.013	0	0	3249,3 363,38 98
3	4	206	197	176	0.419	0.09	0	5048,4 909,57 5
4	8	11935	15524	555	0.49	0.09	0	96583, 12175 4,5714

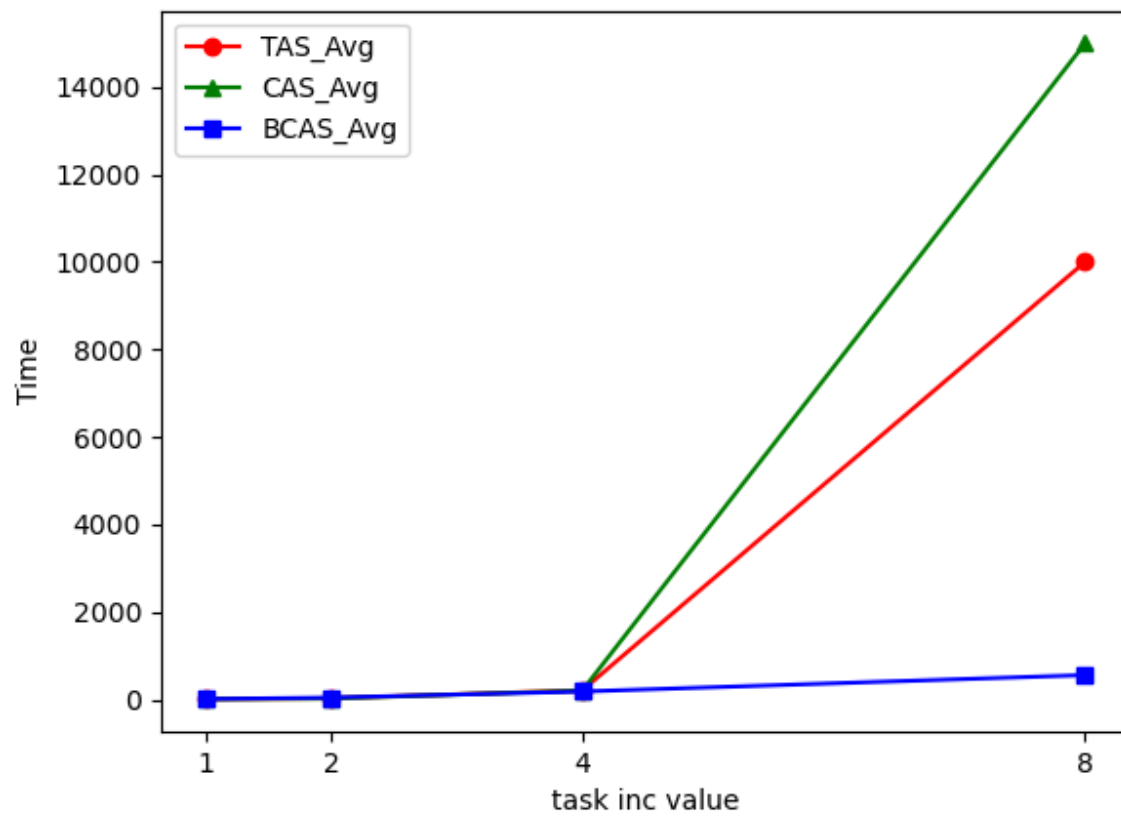
Worst Case Entry Tables:

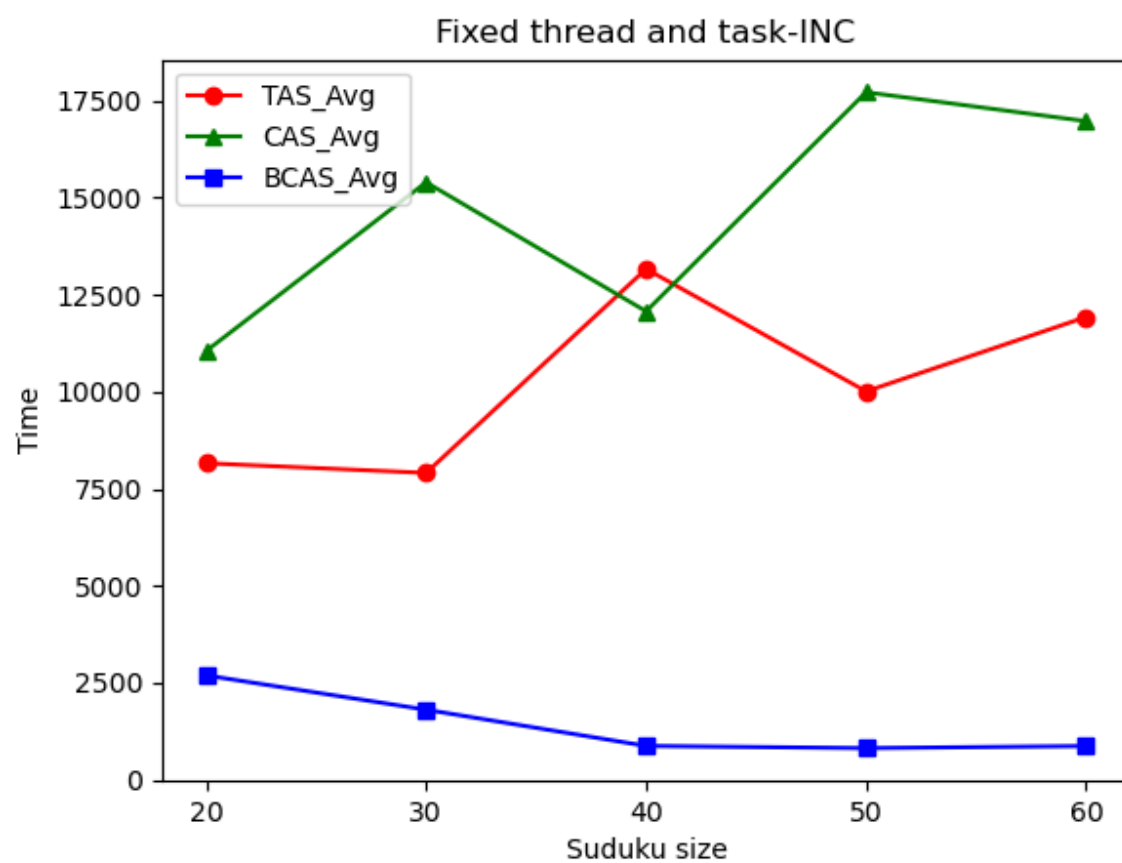
S.No	X axis (threads)	TAS worst CS Entry time (in micro sec)	CAS worst CS Entry time (in micro sec)	Bound ed CAS worst CS Entry time (in micro sec)	TAS worst CS Exit time (in micro sec)	CAS worst CS Exit time (in micro sec)	Bound ed CAS worst CS Exit time (in micro sec)	Total time taken (in micro sec)
------	----------------------	---	---	--	--	--	---	--

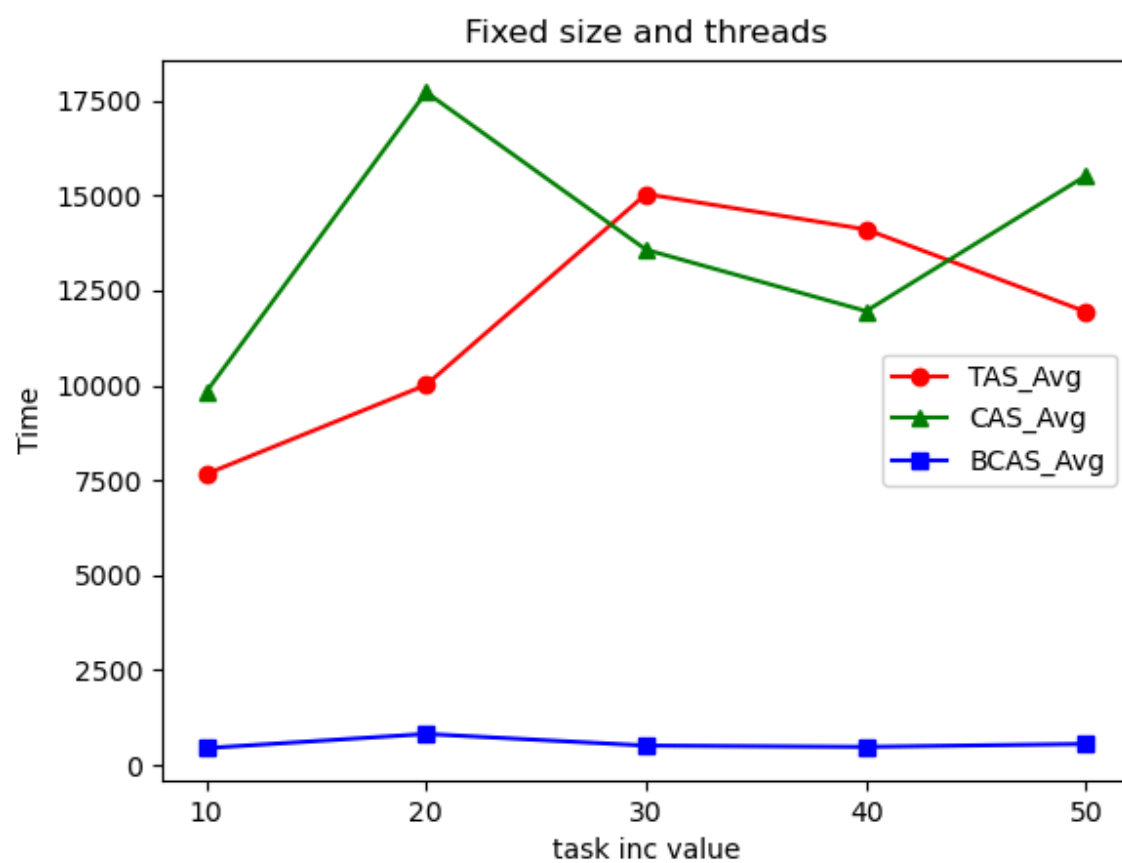
1	1	193	186	268	0	0	0	4901,4 724,42 09
2	2	327	308	624	1	0	0	3249,3 363,38 98
3	4	1204	1773	1458	3	1	0	5048,4 909,57 5
4	8	91288	12089 2	4391	3	2	0	96583, 12175 4,5714

Graphs for the experiments:

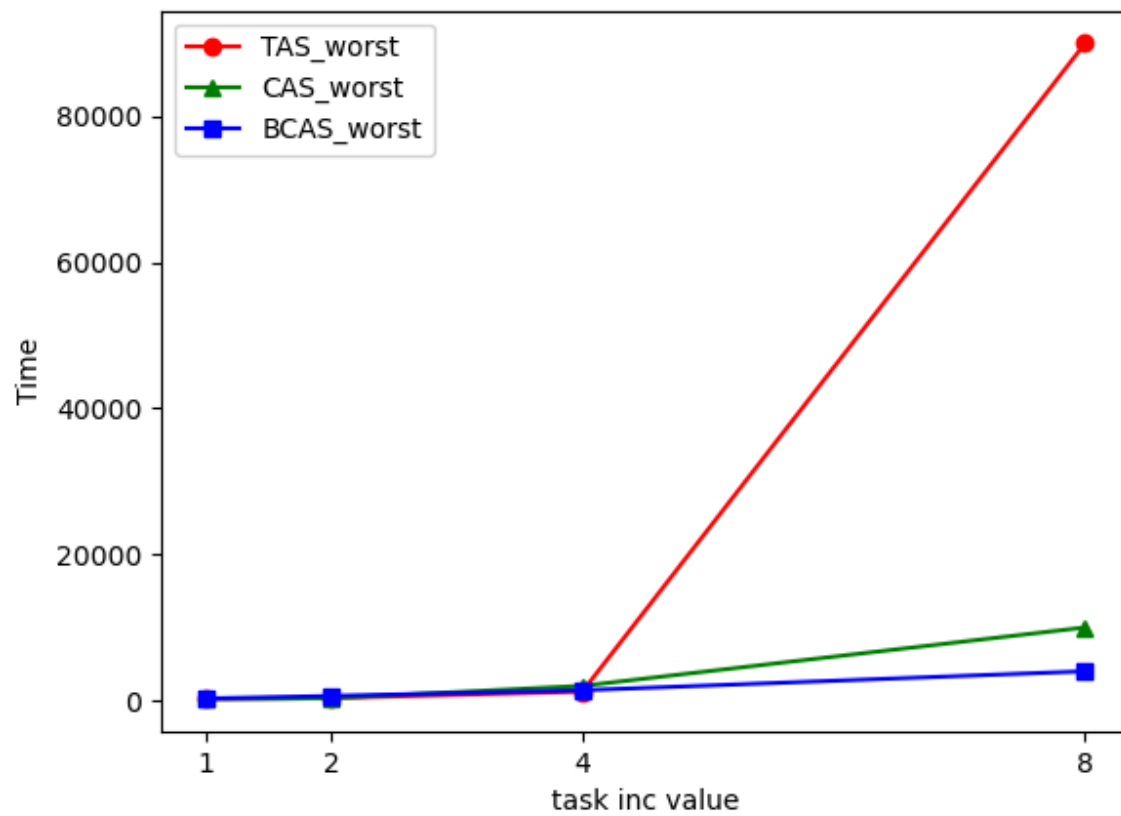
Fixed size and task-INC







Fixed size and task-INC



Fixed thread and task-INC

