CSC367 Assignment 3 report

Name: XinHao Hou
Utorid: houxinha

Part 1 - Join.c

This part of this assignment consists of implementing the pseudo code given into working c code. Implementing nested-loop join and sort-merge join did not take much time because these two algorithms are very straightforward and easy to implement. Hash join gave me a lot of trouble because I didn't have a working hashtable implementation from lab 3 and implementing and debugging this hashtable took a very long time. In the hash-join implementation I chose to have all the students' gpa being inserted into the hashmap using their sid as the key. I chose to do this because every student and their student number is unique. I chose to not use tas' contract number as key because there will be no way to look it up while looping through the students and I chose not to use tas' student number as key because it will cause issues when one student number has multiple contract numbers.

Part 2 - OpenMP Implementation

For fragment and replicate I simply split the tas array using the tas_count. Each thread will get an equal portion and if tas_count is not divisible by thread count the last thread will have a little more work than the other threads. The omp command I chose to use is "#pragma omp parallel for reduction(+:count)" which would have each thread run one interaction of the loop and add up the integer count at the end.

For symmetric partitioning I have split up the work based on student numbers. Each thread would take on an equal amount of student number and the students and tas array would be partitioned accordingly. I also used the command "#pragma omp parallel for reduction(+:count)" for similar reasons as above. and this method should run faster than fragment and replicate because it would have less students to search through

Part 3 - OpenMP Performance

To understand performance we need to understand the strength and weakness of each join method.

- Nested loop
    - Just slow
    - Simple to write
    - n^2
- Sort and merge
    - Strength
        - Almost no setup time
        - Fast runtime, since it only compares students with the same student number
    - Weakness
        - Need both array to be sorted by the same attribute
- Hash join
    - Strength
        - No assumption need to make with input data
        - Fast assess time
    - Weakness
        - Slow setup time

From the table above assuming the input is already sorted, sort and merge will have an advantage for data with a lot of students and less tas. Hash join will be faster when there are not a lot of students but many tas. With fragment and replicate we have the situation where there are potentially many students and less tas which would give sort and merge join the advantage. Symmetric partitioning would have less students when running more multiple threads compared to fragment and replicate which would mean hash join will potentially run faster in this situation.

Part 4 - MPI Implementation

Fragment and replicate is kind very slow in this implementation because every thread needs to send their part of students to every other thread. Sending data through the network is using io and using io is slow. With my implementation I used allgatherv because every thread can send a different amount of data and I treated the student array as binary so I used MPI_BYTE to send them. After all the data has been sent I run the join function and call gather on thread 0 to return my results.

Symmetric partitioning is considerably fast than fragment and replicate because it doesn't call allgatherv on every thread and therefore avoid io operation. My implementation calls the join function on each thread. After the join function returns it calls gather on thread 0 to return the results.

Part 5 - MPI Performance

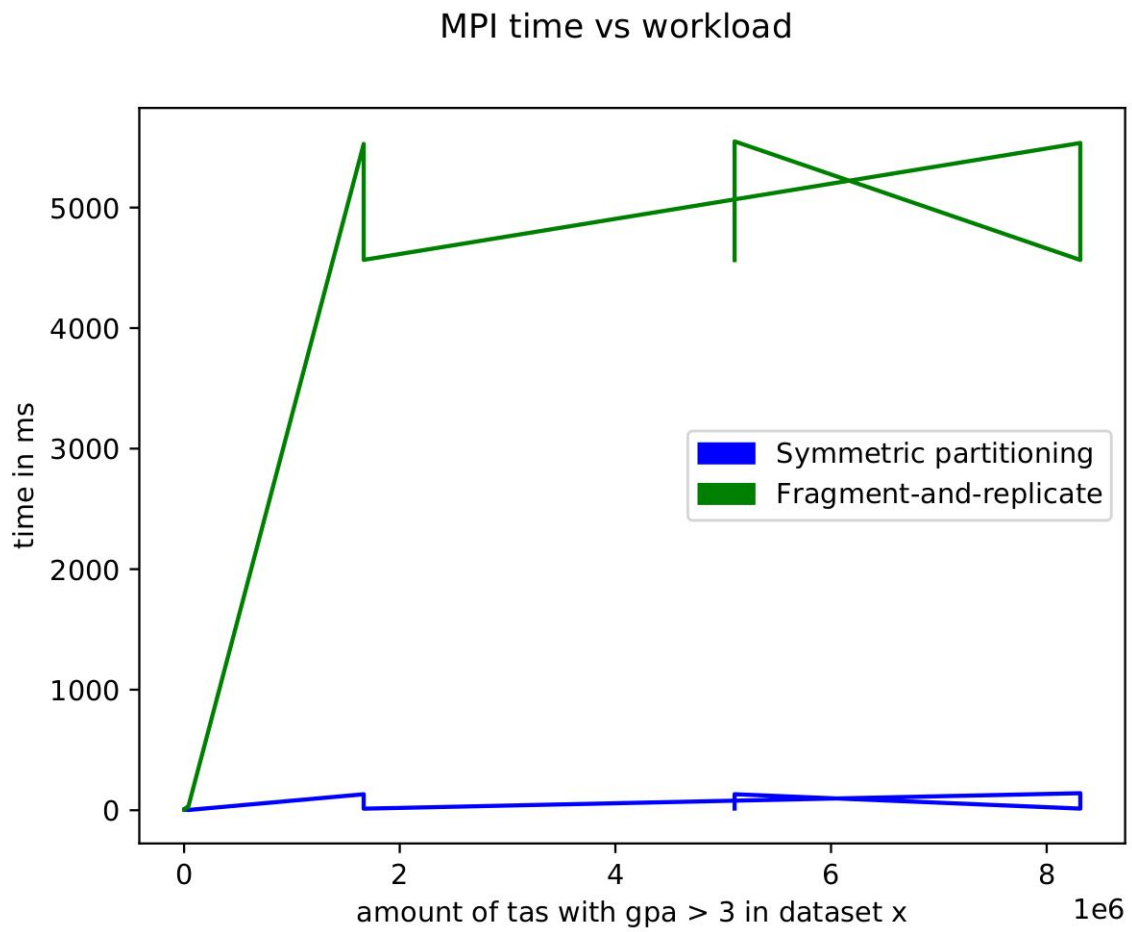
MPI performs worse than OMP because the work is split onto nodes on the network and sending data through the network is slow. Symmetric partitioning performs significantly better compared to fragment and replicate because it avoids io operation as mentioned above and symmetric partitioning also benefits from the way that the data is split. The advantage of MPI is we can use multiple nodes on the same network to achieve much more computing power. There is no reason that we can not run 4 threads per compute node and MPI would also benefit from having more work. In the real world an MPI implementation would mean we can use 4 servers each with 32 cores instead of finding some server with 128 cores.

Part 5 - My experience with this project

This assignment wasn’t as difficult as assignment 2. I think this is mainly because we had experience splitting work for assignment 2 and that experience carried over to this assignment. I had difficulty getting the hashtable to work properly but didn’t have much trouble implementing anything else. All of the experiments and conclusions were expected since we talked about them in lecture. I wish our assignment 4 would be similar to this assignment since I feel like this assignment we were prepared with all the knowledge needed to complete the assignment.

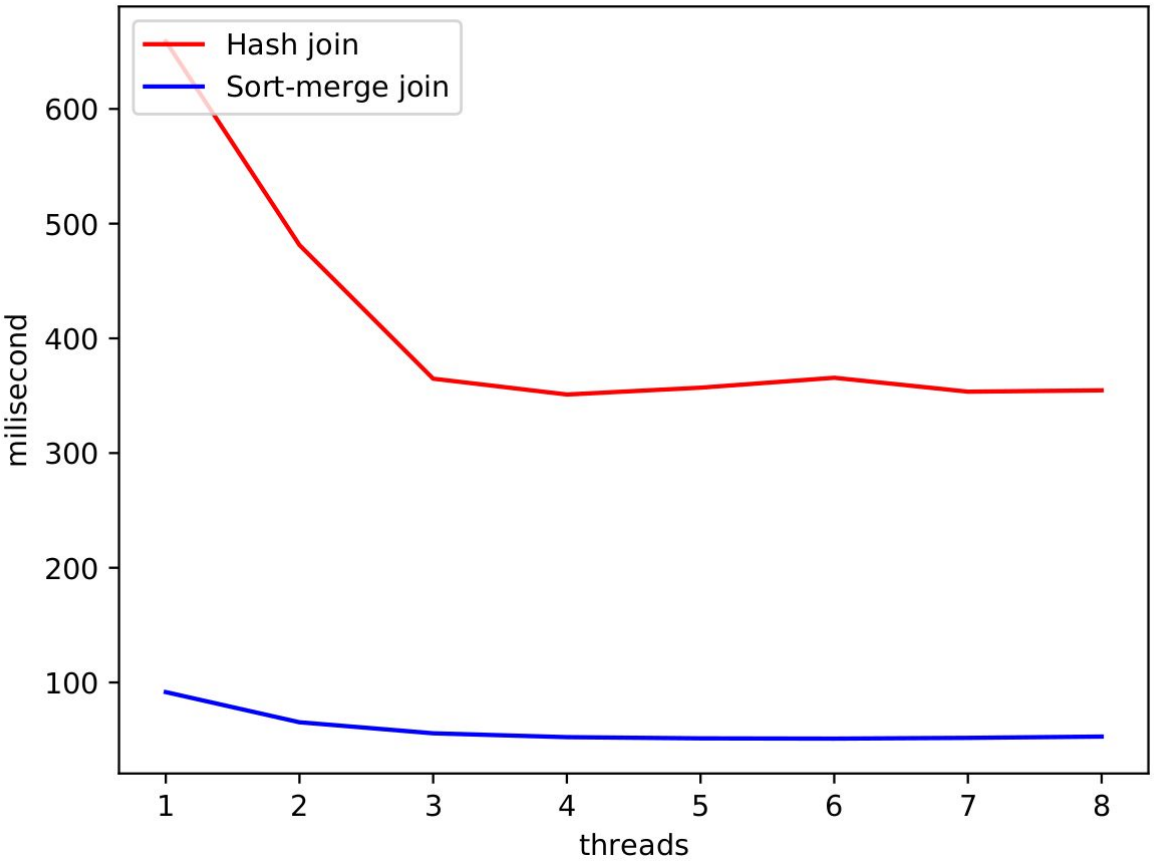
Part 6 - Plots

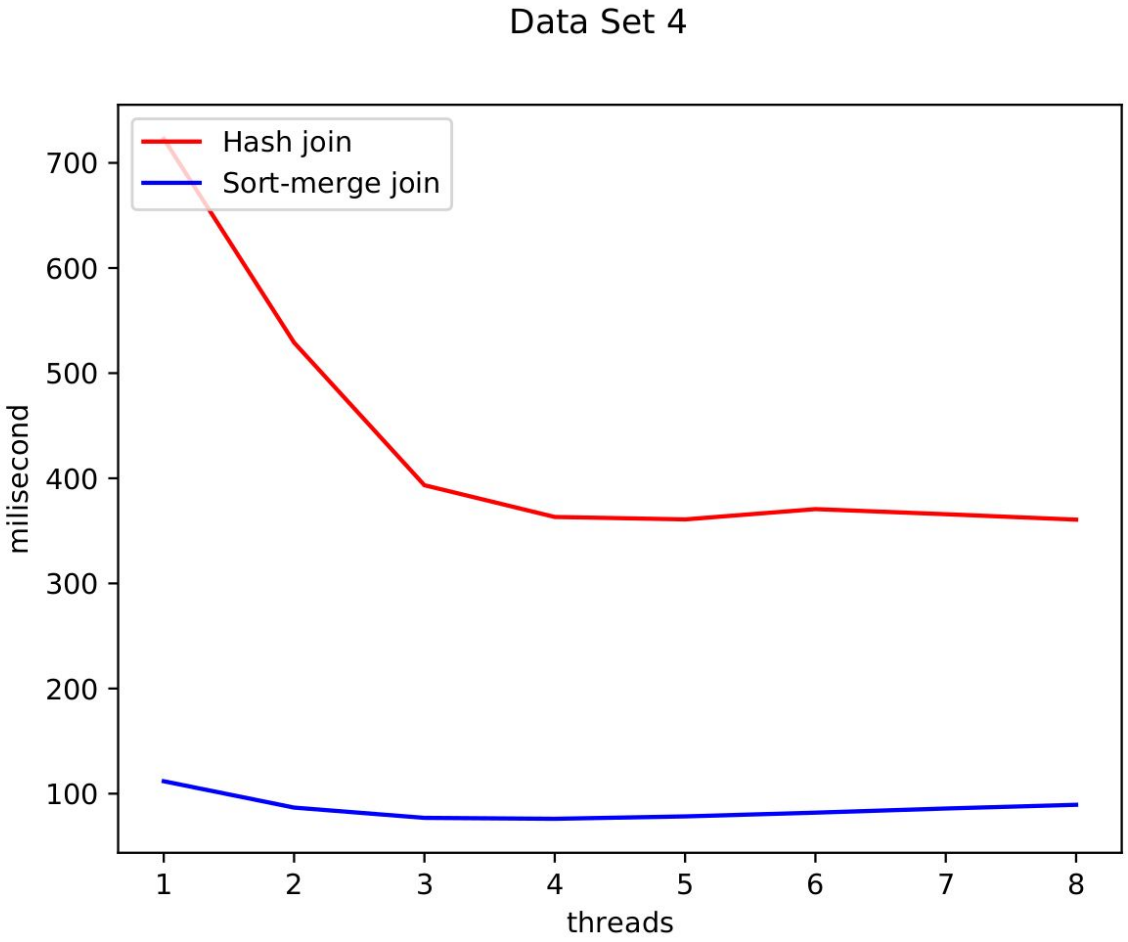
MPI plots

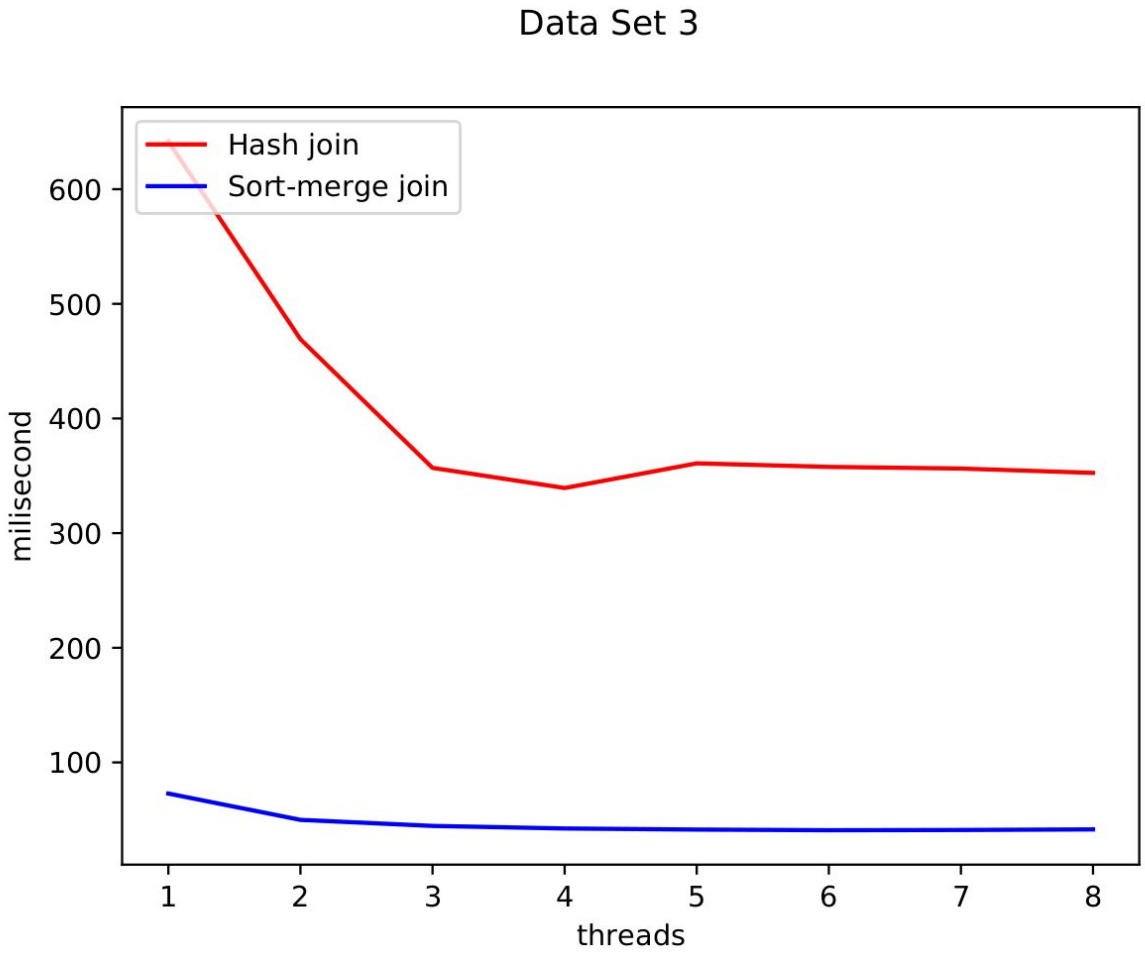


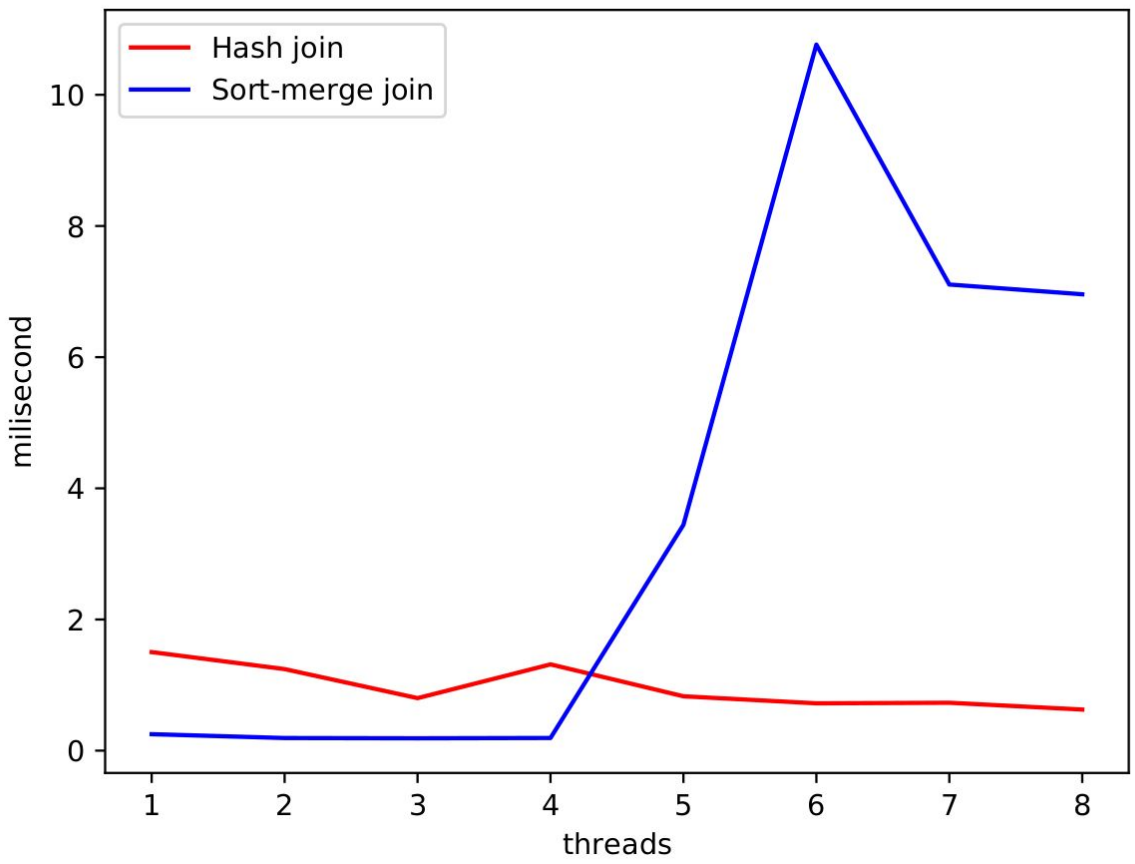
OMP plots

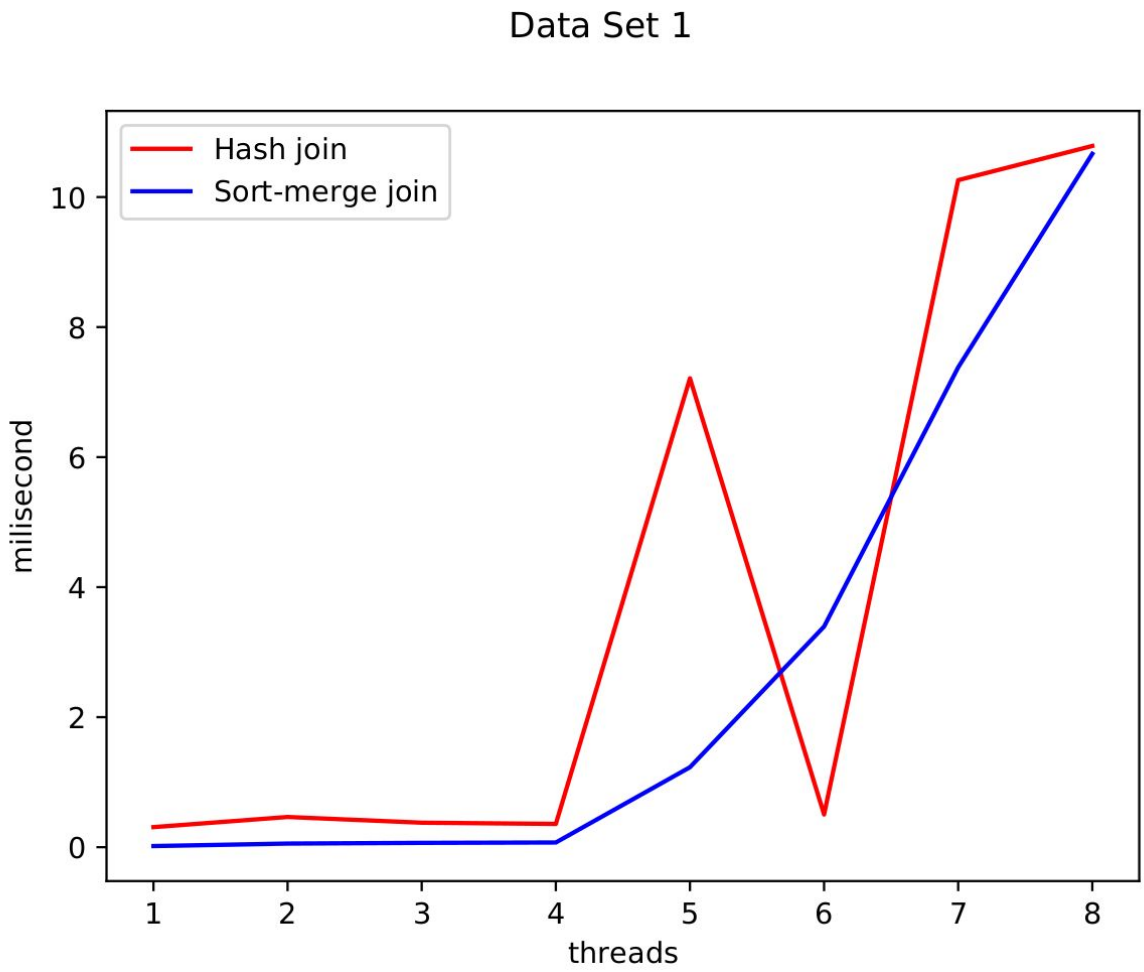
Data Set 5


Hash join
Sort-merge join
600
500
400
300
200
100
milisecond
1
2
3
4
5
6
7
8
threads


Data Set 4
Hash join
Sort-merge join
700
600
500
400
300
200
100
milisecond
1
2
3
4
5
6
7
8
threads


Data Set 3
Hash join
Sort-merge join
600
500
400
300
200
100
milisecond
1
2
3
4
5
6
7
8
threads


Data Set 2
Hash join
Sort-merge join
milisecond
threads
0
2
4
6
8
10
1
2
3
4
5
6
7
8



Note that dataset1 and dataset2 is too small to conclude with any meaningful data.

Part 7 - Test scripts and results

Test script 1

```
./join-omp -h -s -t 1 inputs/dataset5
./join-omp -h -s -t 2 inputs/dataset5
./join-omp -h -s -t 3 inputs/dataset5
./join-omp -h -s -t 4 inputs/dataset5
./join-omp -h -s -t 5 inputs/dataset5
./join-omp -h -s -t 6 inputs/dataset5
./join-omp -h -s -t 7 inputs/dataset5
./join-omp -h -s -t 8 inputs/dataset5

./join-omp -m -s -t 1 inputs/dataset5
./join-omp -m -s -t 2 inputs/dataset5
./join-omp -m -s -t 3 inputs/dataset5
./join-omp -m -s -t 4 inputs/dataset5
./join-omp -m -s -t 5 inputs/dataset5
./join-omp -m -s -t 6 inputs/dataset5
./join-omp -m -s -t 7 inputs/dataset5
./join-omp -m -s -t 8 inputs/dataset5
echo
echo dataset4

./join-omp -h -s -t 1 inputs/dataset4
./join-omp -h -s -t 2 inputs/dataset4
./join-omp -h -s -t 3 inputs/dataset4
./join-omp -h -s -t 4 inputs/dataset4
./join-omp -h -s -t 5 inputs/dataset4
./join-omp -h -s -t 6 inputs/dataset4
./join-omp -h -s -t 7 inputs/dataset4
./join-omp -h -s -t 8 inputs/dataset4

./join-omp -m -s -t 1 inputs/dataset4
./join-omp -m -s -t 2 inputs/dataset4
./join-omp -m -s -t 3 inputs/dataset4
./join-omp -m -s -t 4 inputs/dataset4
./join-omp -m -s -t 5 inputs/dataset4
```

./join-omp -m -s -t 6 inputs/dataset4
./join-omp -m -s -t 7 inputs/dataset4
./join-omp -m -s -t 8 inputs/dataset4
echo
echo dataset3

./join-omp -h -s -t 1 inputs/dataset3
./join-omp -h -s -t 2 inputs/dataset3
./join-omp -h -s -t 3 inputs/dataset3
./join-omp -h -s -t 4 inputs/dataset3
./join-omp -h -s -t 5 inputs/dataset3
./join-omp -h -s -t 6 inputs/dataset3
./join-omp -h -s -t 7 inputs/dataset3
./join-omp -h -s -t 8 inputs/dataset3

./join-omp -m -s -t 1 inputs/dataset3
./join-omp -m -s -t 2 inputs/dataset3
./join-omp -m -s -t 3 inputs/dataset3
./join-omp -m -s -t 4 inputs/dataset3
./join-omp -m -s -t 5 inputs/dataset3
./join-omp -m -s -t 6 inputs/dataset3
./join-omp -m -s -t 7 inputs/dataset3
./join-omp -m -s -t 8 inputs/dataset3

Results 1

5105918
499.151271
5105918
377.536692
5105918
284.028814
5105918
242.801447
5105918
223.784605
5105918

221.257767
5105918
225.950902
5105918
228.984731
5105918
83.790834
5105918
60.948949
5105918
48.075842
5105918
52.396982
5105918
45.330842
5105918
49.311884
5105918
46.735502
5105918
49.386813

dataset4
8312045
575.823086
8312045
424.301732
8312045
308.743010
8312045
256.408236
8312045
246.439271
8312045
239.469961
8312045
238.340752

8312045
244.950966
8312045
91.780524
8312045
73.254645
8312045
62.667242
8312045
60.411509
8312045
62.759391
8312045
67.267189
8312045
70.796253
8312045
69.080587

dataset3
1665827
483.515585
1665827
370.445061
1665827
265.063856
1665827
242.478446
1665827
216.080996
1665827
213.378334
1665827
214.895534
1665827
229.349414
1665827

81.067076
1665827
55.311950
1665827
44.948002
1665827
40.873150
1665827
39.417171
1665827
44.052218
1665827
44.206277
1665827
39.654000

Test script 2

```
echo dataset0
mpirun -np 8 --hostfile hostfile ./join-mpi -s -h
/virtual/csc367/datasets-a3/mpi8/dataset0
mpirun -np 8 --hostfile hostfile ./join-mpi -s -h
/virtual/csc367/datasets-a3/mpi8/dataset0
mpirun -np 8 --hostfile hostfile ./join-mpi -r -h
/virtual/csc367/datasets-a3/mpi8/dataset0
mpirun -np 8 --hostfile hostfile ./join-mpi -r -h
/virtual/csc367/datasets-a3/mpi8/dataset0
mpirun -np 8 --hostfile hostfile ./join-mpi -s -m
/virtual/csc367/datasets-a3/mpi8/dataset0
mpirun -np 8 --hostfile hostfile ./join-mpi -s -m
/virtual/csc367/datasets-a3/mpi8/dataset0
mpirun -np 8 --hostfile hostfile ./join-mpi -r -m
/virtual/csc367/datasets-a3/mpi8/dataset0
mpirun -np 8 --hostfile hostfile ./join-mpi -r -m
/virtual/csc367/datasets-a3/mpi8/dataset0
echo dataset1
```

```
mpirun -np 8 --hostfile hostfile ./join-mpi -s -h
/virtual/csc367/datasets-a3/mpi8/dataset1
mpirun -np 8 --hostfile hostfile ./join-mpi -s -h
/virtual/csc367/datasets-a3/mpi8/dataset1
mpirun -np 8 --hostfile hostfile ./join-mpi -r -h
/virtual/csc367/datasets-a3/mpi8/dataset1
mpirun -np 8 --hostfile hostfile ./join-mpi -r -h
/virtual/csc367/datasets-a3/mpi8/dataset1
mpirun -np 8 --hostfile hostfile ./join-mpi -s -m
/virtual/csc367/datasets-a3/mpi8/dataset1
mpirun -np 8 --hostfile hostfile ./join-mpi -s -m
/virtual/csc367/datasets-a3/mpi8/dataset1
mpirun -np 8 --hostfile hostfile ./join-mpi -r -m
/virtual/csc367/datasets-a3/mpi8/dataset1
mpirun -np 8 --hostfile hostfile ./join-mpi -r -m
/virtual/csc367/datasets-a3/mpi8/dataset1
echo dataset2
mpirun -np 8 --hostfile hostfile ./join-mpi -s -h
/virtual/csc367/datasets-a3/mpi8/dataset2
mpirun -np 8 --hostfile hostfile ./join-mpi -s -h
/virtual/csc367/datasets-a3/mpi8/dataset2
mpirun -np 8 --hostfile hostfile ./join-mpi -r -h
/virtual/csc367/datasets-a3/mpi8/dataset2
mpirun -np 8 --hostfile hostfile ./join-mpi -r -h
/virtual/csc367/datasets-a3/mpi8/dataset2
mpirun -np 8 --hostfile hostfile ./join-mpi -s -m
/virtual/csc367/datasets-a3/mpi8/dataset2
mpirun -np 8 --hostfile hostfile ./join-mpi -s -m
/virtual/csc367/datasets-a3/mpi8/dataset2
mpirun -np 8 --hostfile hostfile ./join-mpi -r -m
/virtual/csc367/datasets-a3/mpi8/dataset2
mpirun -np 8 --hostfile hostfile ./join-mpi -r -m
/virtual/csc367/datasets-a3/mpi8/dataset2
echo dataset3
mpirun -np 8 --hostfile hostfile ./join-mpi -s -h
/virtual/csc367/datasets-a3/mpi8/dataset3
```

```
mpirun -np 8 --hostfile hostfile ./join-mpi -s -h
/virtual/csc367/datasets-a3/mpi8/dataset3
mpirun -np 8 --hostfile hostfile ./join-mpi -r -h
/virtual/csc367/datasets-a3/mpi8/dataset3
mpirun -np 8 --hostfile hostfile ./join-mpi -r -h
/virtual/csc367/datasets-a3/mpi8/dataset3
mpirun -np 8 --hostfile hostfile ./join-mpi -s -m
/virtual/csc367/datasets-a3/mpi8/dataset3
mpirun -np 8 --hostfile hostfile ./join-mpi -s -m
/virtual/csc367/datasets-a3/mpi8/dataset3
mpirun -np 8 --hostfile hostfile ./join-mpi -r -m
/virtual/csc367/datasets-a3/mpi8/dataset3
mpirun -np 8 --hostfile hostfile ./join-mpi -r -m
/virtual/csc367/datasets-a3/mpi8/dataset3
echo dataset4
mpirun -np 8 --hostfile hostfile ./join-mpi -s -h
/virtual/csc367/datasets-a3/mpi8/dataset4
mpirun -np 8 --hostfile hostfile ./join-mpi -s -h
/virtual/csc367/datasets-a3/mpi8/dataset4
mpirun -np 8 --hostfile hostfile ./join-mpi -r -h
/virtual/csc367/datasets-a3/mpi8/dataset4
mpirun -np 8 --hostfile hostfile ./join-mpi -r -h
/virtual/csc367/datasets-a3/mpi8/dataset4
mpirun -np 8 --hostfile hostfile ./join-mpi -s -m
/virtual/csc367/datasets-a3/mpi8/dataset4
mpirun -np 8 --hostfile hostfile ./join-mpi -s -m
/virtual/csc367/datasets-a3/mpi8/dataset4
mpirun -np 8 --hostfile hostfile ./join-mpi -r -m
/virtual/csc367/datasets-a3/mpi8/dataset4
mpirun -np 8 --hostfile hostfile ./join-mpi -r -m
/virtual/csc367/datasets-a3/mpi8/dataset4
echo dataset5
mpirun -np 8 --hostfile hostfile ./join-mpi -s -h
/virtual/csc367/datasets-a3/mpi8/dataset5
mpirun -np 8 --hostfile hostfile ./join-mpi -s -h
/virtual/csc367/datasets-a3/mpi8/dataset5
```

mpirun -np 8 --hostfile hostfile ./join-mpi -r -h /virtual/csc367/datasets-a3/mpi8/dataset5
mpirun -np 8 --hostfile hostfile ./join-mpi -r -h /virtual/csc367/datasets-a3/mpi8/dataset5
mpirun -np 8 --hostfile hostfile ./join-mpi -s -m /virtual/csc367/datasets-a3/mpi8/dataset5
mpirun -np 8 --hostfile hostfile ./join-mpi -s -m /virtual/csc367/datasets-a3/mpi8/dataset5
mpirun -np 8 --hostfile hostfile ./join-mpi -r -m /virtual/csc367/datasets-a3/mpi8/dataset5
mpirun -np 8 --hostfile hostfile ./join-mpi -r -m /virtual/csc367/datasets-a3/mpi8/dataset5

Result 2
dataset0
232
4.518663
232
2.216899
232
5.644136
232
6.059552
232
1.956833
232
3.641682
232
7.484893
232
6.890309
dataset1
2091
2.560446
2091
2.145787
2091

10.590206
2091
9.470857
2091
2.882339
2091
2.801837
2091
9.318932
2091
7.164639
dataset2
37290
3.362364
37290
4.100498
37290
31.616461
37290
29.814596
37290
1.913444
37290
2.491817
37290
23.563886
37290
23.020832
dataset3
1665827
132.599818
1665827
135.480491
1665827
5527.696155
1665827
5522.323284

1665827
12.991135
1665827
12.109092
1665827
4566.015352
1665827
4564.832985
dataset4
8312045
141.259465
8312045
142.365631
8312045
5535.806229
8312045
5601.425513
8312045
13.521269
8312045
14.348708
8312045
4564.932173
8312045
4564.202929
dataset5
5105918
132.882170
5105918
135.026926
5105918
5548.914734
5105918
5608.547622
5105918
13.912582
5105918

13.890833
5105918
4564.230664
5105918
4566.791967