Report on Cache Simulation Activity

This report compiles the observations made on the cache simulation activity performed under three scenarios.

Scenario A:

Conditions:

Cache size should be a constant (1KB preferred).

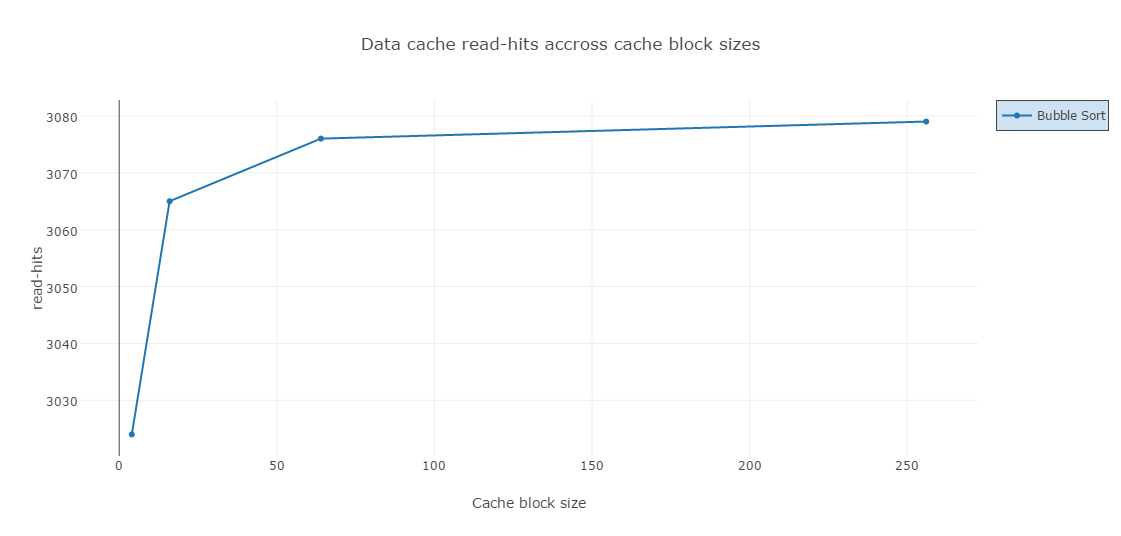
Associativity = 1

Verify Bubble sort algorithm.

Input string:

55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0;

Graph:



Observation and Inference:

* For constant cache size and associativity, expected behavior is that the successful reads from cache should increase with increase in the size of cache block.
* The graph depicts the same behavior. This implies that consecutive memory locations were accessed for data and reads from cache were more successful when the size of the cache block increased.
* Another inference is that for constant cache size, the size of individual block of cache is dominant in getting successful cache reads when compared to number of cache blocks.

Scenario B:

Conditions:

Cache size should be a constant (1KB preferred).

Cache block size= 64 bytes

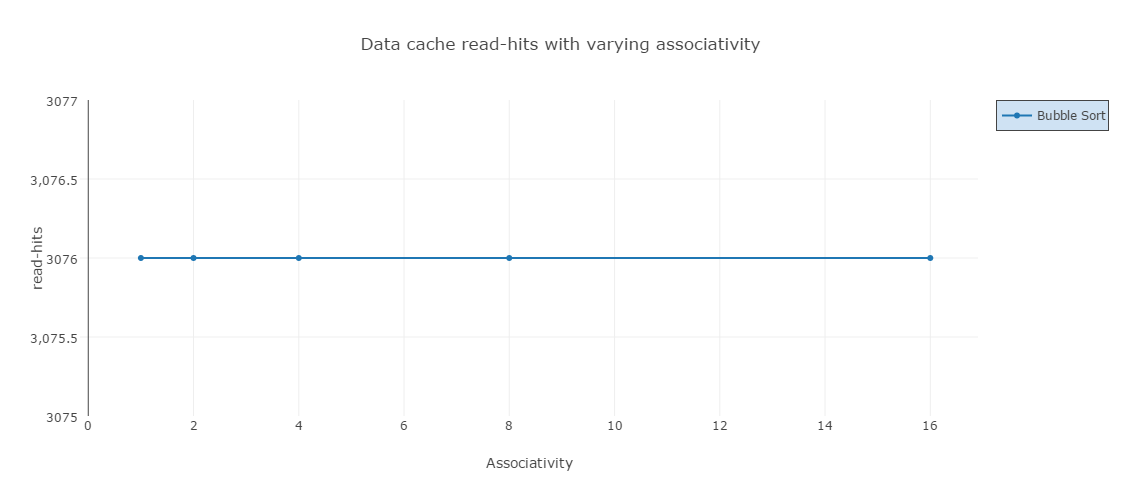
Number of cache blocks = 16

Verify Bubble sort algorithm.

Input string:

55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0;

Graph:



Observation and Inference:

* For constant cache size and cache block size, expected behavior is that the successful reads from cache should increase with increase in the associativity.
* The graph depicts a different behavior in this case.
* We observe that the successful reads do not change with change in the associativity.
* Based on the observation we can infer that change in associativity does not have any impact on the successful cache reads for a medium unsorted array.

Scenario C:

Conditions:

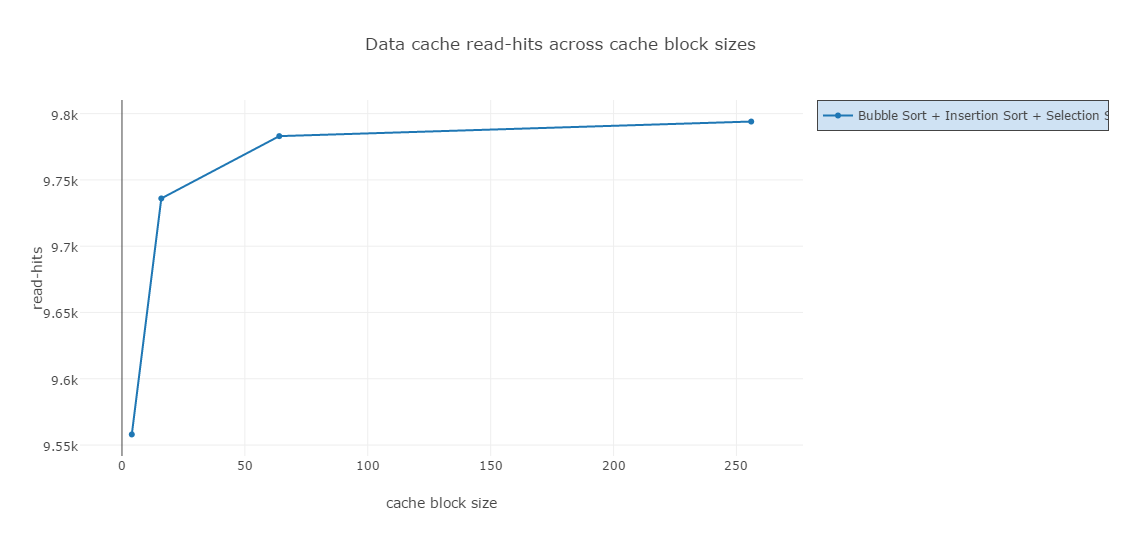
Cache size should be a constant (1KB preferred).

Verify Bubble sort, insertion sort and wild card sort algorithms together.

Input string:

43 36 44 50 59 17 22 30 57 51;55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0;

Graph:



Observation and Inference:

* For constant cache size, expected behavior is that the successful reads from cache should increase with increase in the associativity and in the size of cache block.
* The graph above showcases the change in successful reads with respect to change in size of cache block. All the graphs plotted for different associativity values are same as the one above.
* Based on the observation we can infer that change in associativity does not have any impact on the successful cache reads for medium unsorted array and short unsorted array.
* We can also infer that consecutive memory locations were accessed for data and were available in cache when the size of the cache block increased.
* Another inference is that for constant cache size, the size of individual block of cache is dominant in getting successful cache reads when compared to number of cache blocks.
* NOTE: The graph plotted above shows cache read-hits of Bubble sort and Selection sort. For Insertion sort, the cache read-hits were observed to be ZERO (0), the code is successfully sorting the given inputs though.

Conclusion:

In conclusion, it is evident that associativity did not really have any impact when reading cached data. The size of cache block, on the other hand, did influence the success of cache reads. Thus, highlighting the concept of special locality. This influence decreased with increase in the size of cache block as observed in the graphs A and C. So, we can infer that to get more success in cache reads, the size of the cache block should be inclined towards the size of the cache itself.