CS301: HIGH PERFORMANCE COMPUTING
ASSIGNMENT THREE

Submitted By:
Shaleen Kumar Gupta (201301429)
Visharad Bansal (201301438)

PROBLEM STATEMENT

To perform the prefix scan operation on a given array, and subsequently filter the values that satisfy a given filter criteria.

METHOD

The prefix scan for an array a[0....(n-1)] can be given as

```
scan_array[i]=a[0] + a[1] +....a[i-1]
```

Filter is done based on a given filter value.

```
if(a[i] satisfies filter)
        select a[i]
```

COMPLEXITY

The serial code for both prefix scan and filter runs in O(n) time, where n is the size of the array.

Scan requires no extra space, and filter needs O(n) auxilay space.

OPTIMIZATION STRATEGY

In prefix scan, the number of iterations are reduced from n to log(n), and in each iteration consequtive 2i values are summed.

In the filter problem, a binary vector array is made which assign a value 1 to index i if a[i] satisfies the filter condition, 0 otherwise. Then, performing a prefix scan on the binary vector array gives us the indexes of the filtered values, which can then be put into the resultant array. Each of these steps take O(n) time serially, thus giving a time of O(3n/p) on p threads.

OBSERVATIONS AND PROBLEMS FACED

As you increase the number of parallel regions, the overhead cost of launching the threads also increases. This produces a roadblock in the possible speedup that can be achieved. Also, the parallel codes have an overhead in terms of the memory used, which is much more than the serial code.

Another limitation was on the system itself, and it failed to support array sizes larger than 100000.

We failed to achieve a good speedup for the prefix scan, and since the algorithm for prefix scan is directly used in the filter, it gave a speedup of less than 1 in the given range of n.

HARDWARE DETAILS

No. of Processors	: 3
vendor_id	: Genuine Intel
CPU family	: 6
model	: 60
model name	: Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz
stepping	: 3
CPU MHz	: 800.000
cache size	: 6144 KB
siblings	: 4
CPU cores	: 4

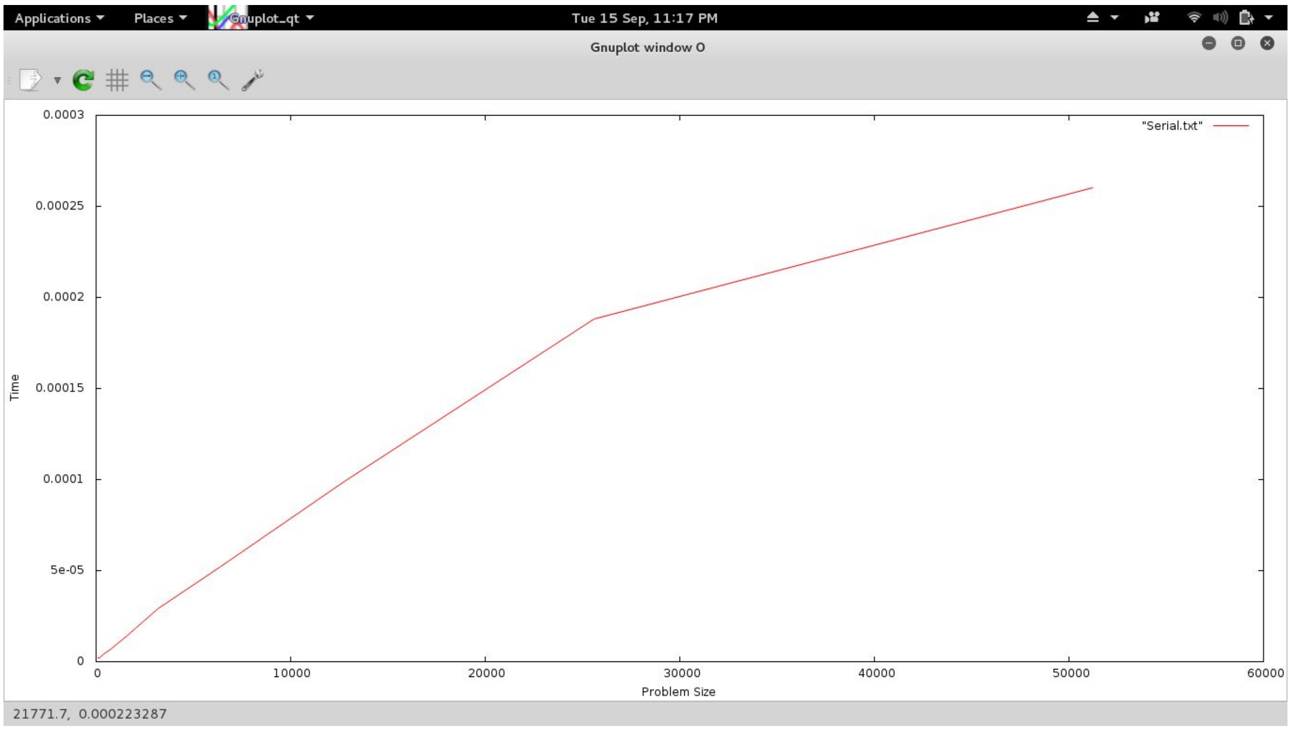
INPUT PARAMETERS

Size of the array (n).
Range of n has been taken from 1000 to 100000.

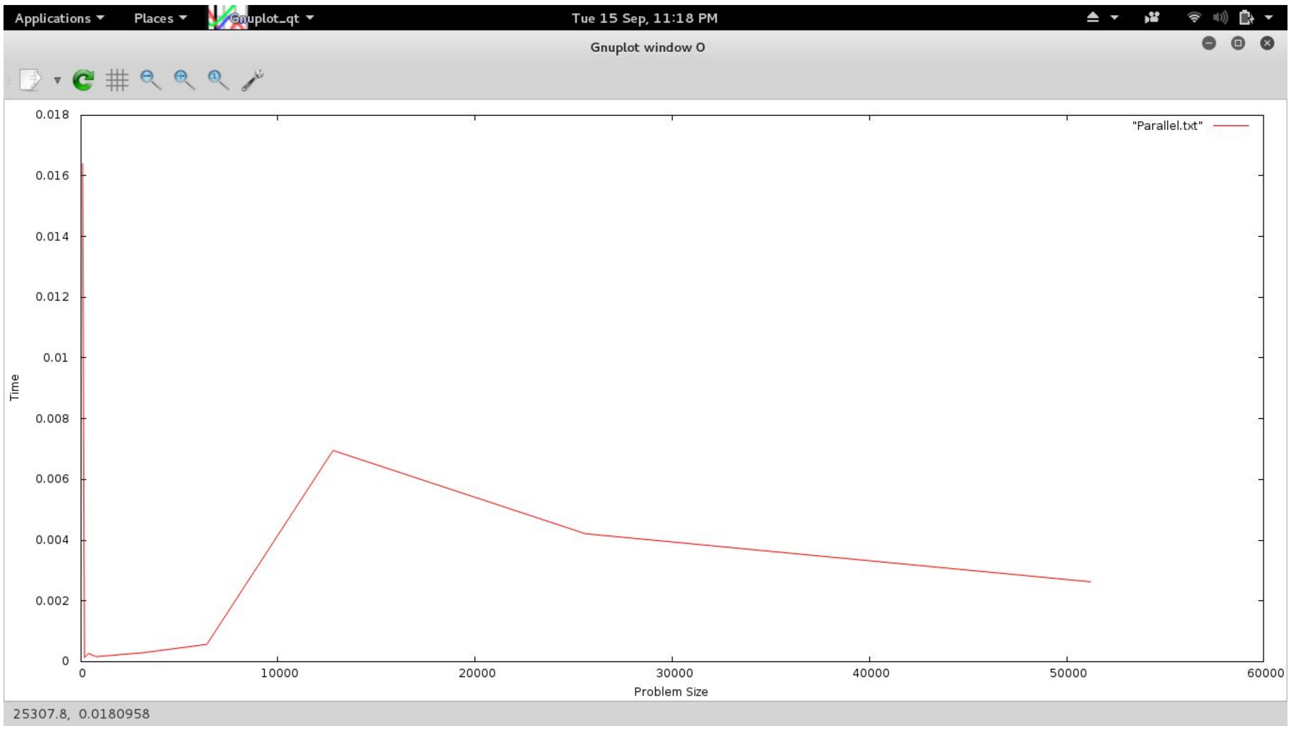
OBSERVATIONS

The serial codes gave a linear speedup, while the parallel code was erratic.

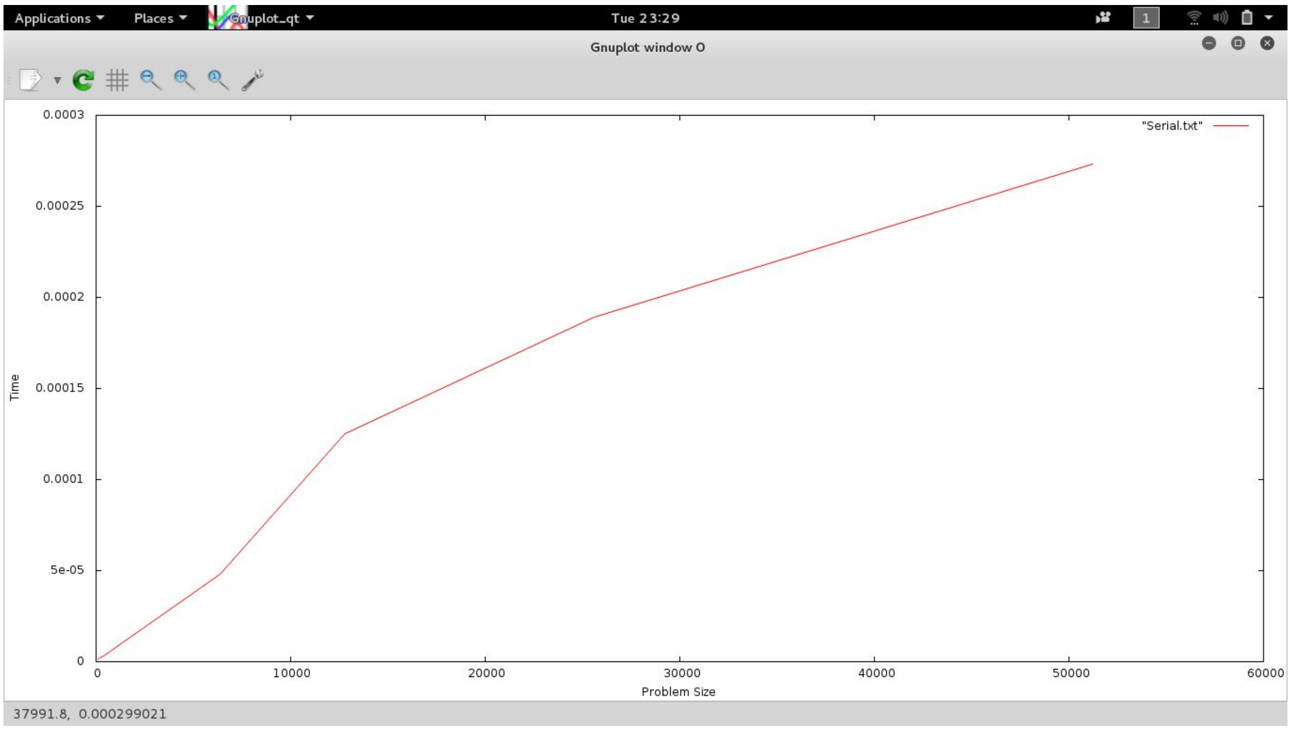
Time was calculated using omp_get_wtime() library function include in the “omp.h” header file.



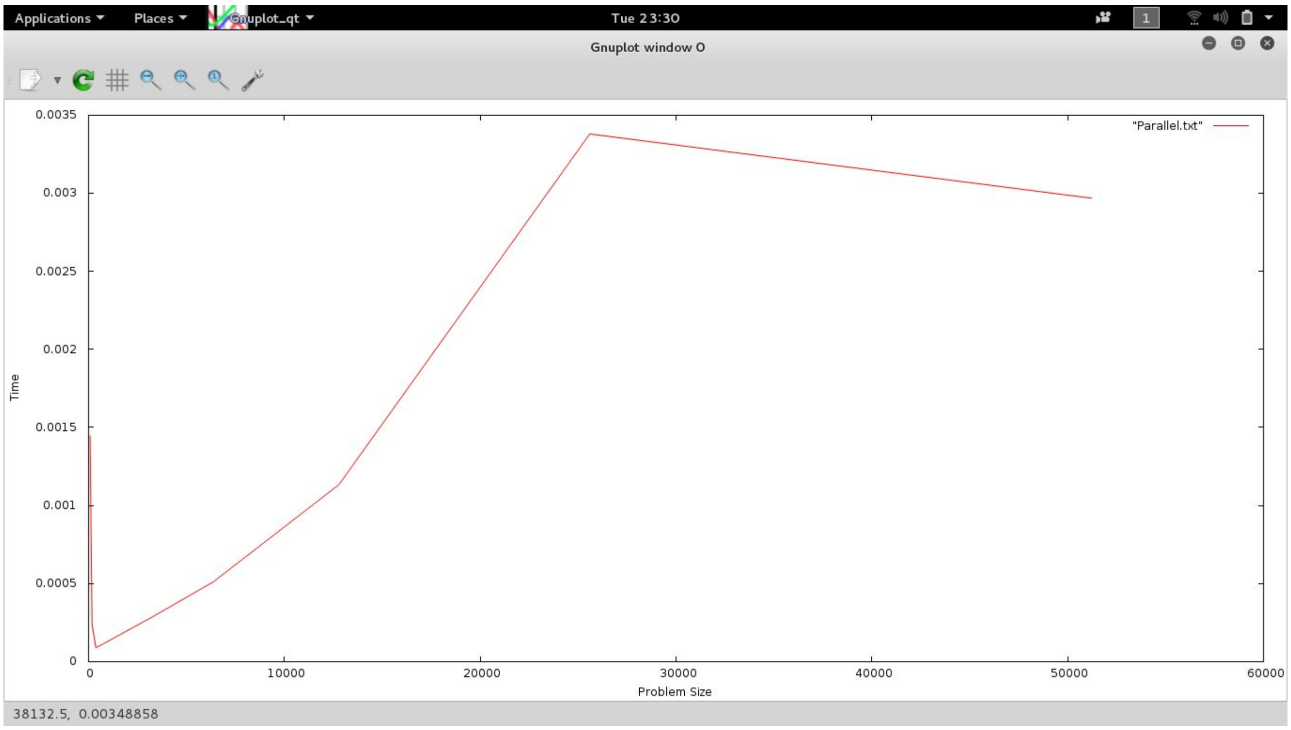
Serial Scaling for Scan



Parallel Scaling for Scan



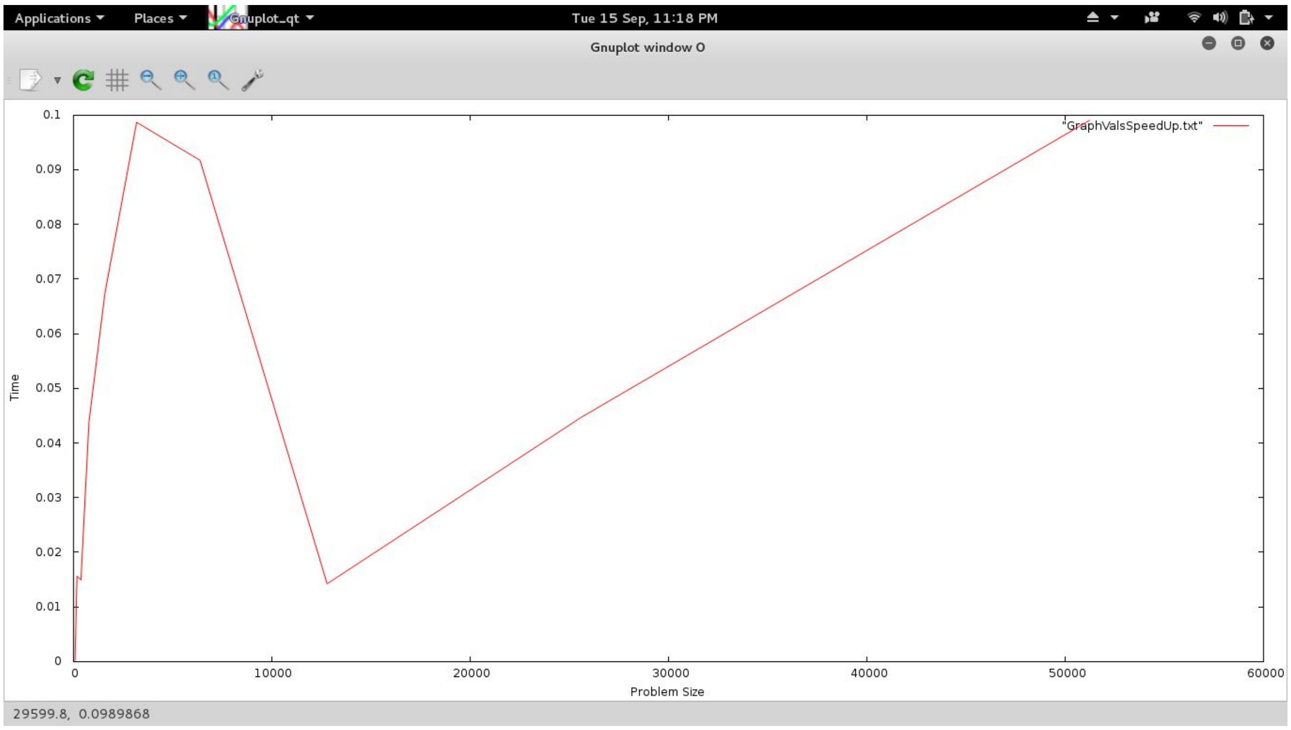
Serial Scan for Filter



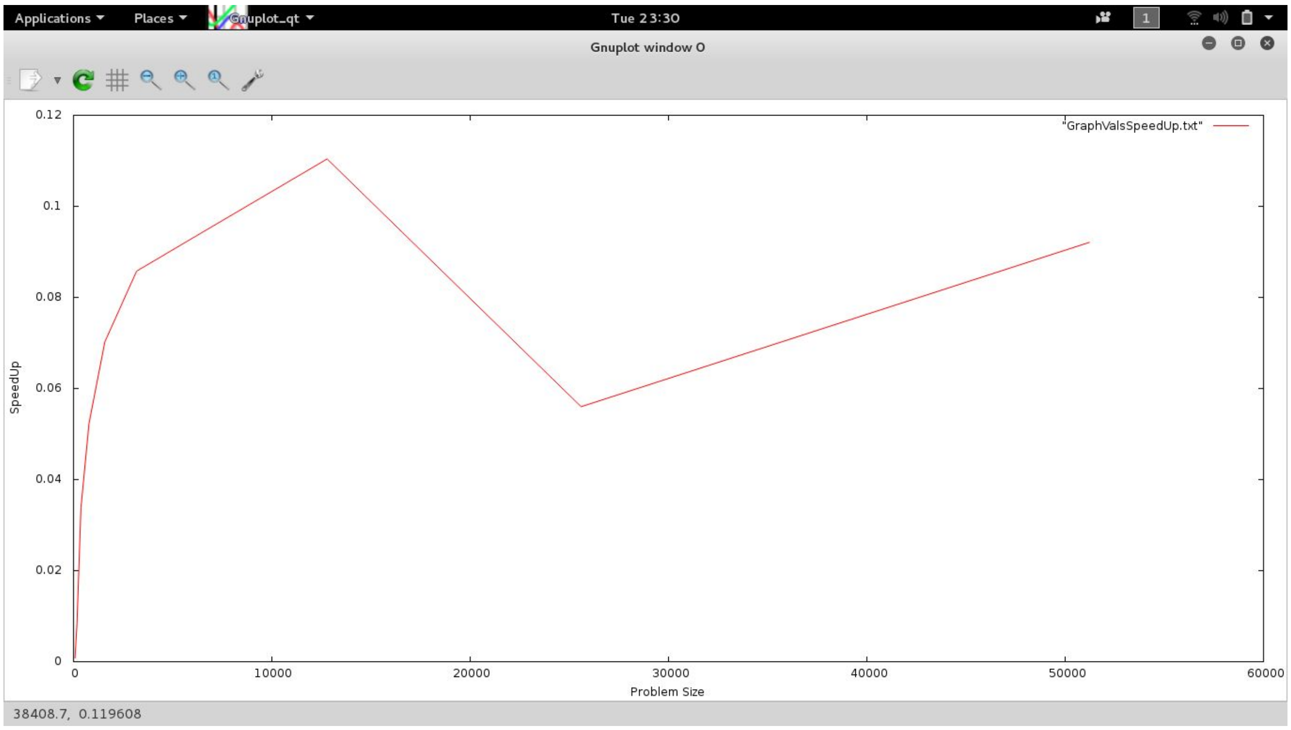
Parallel Filter

SPEED UP

The maximum speedup achieved was < 1
Speed Up was calculated as:
Speed Up= (Time taken by serial code) / (Time taken by parallel code)



Speed Up v/s Time Taken for Scan



SpeedUp vs Time for Filter