**LAB4**

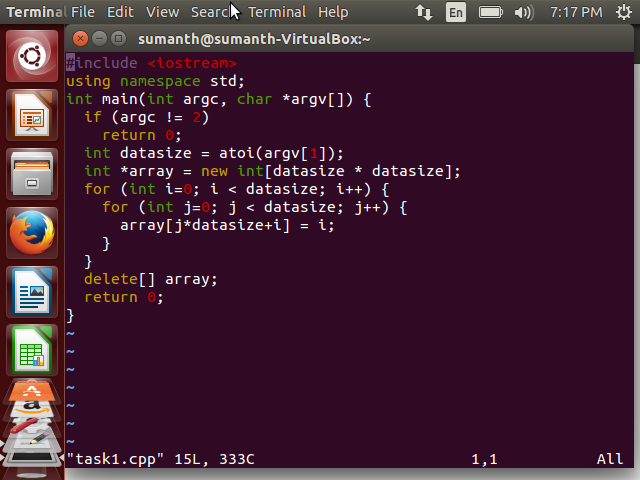
Saisumanth Gopisetty

#831608197

[sgopiset@syr.edu](mailto:sgopiset@syr.edu)

**TASK1**

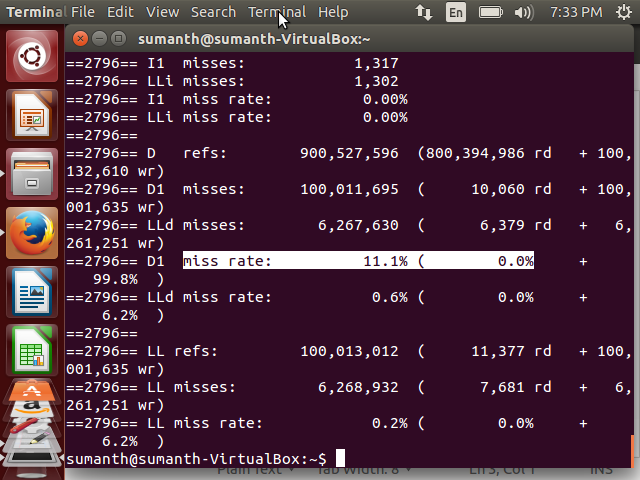
**Original Code:**



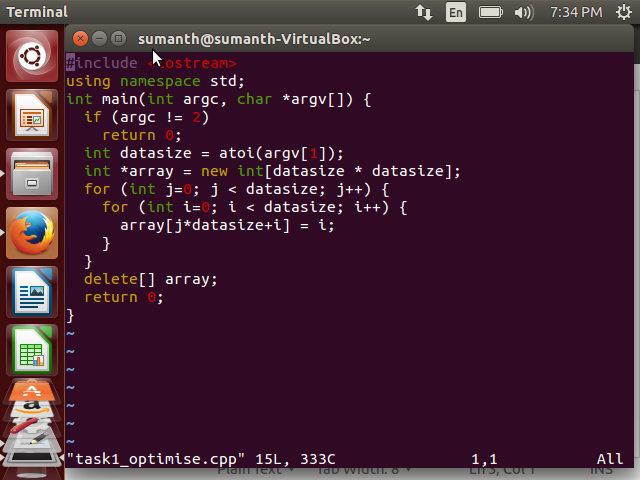
Now,Compiled the original code and ran the valgrind tool to get the D1 miss rate.

The D1 miss rate for original code is as follows:

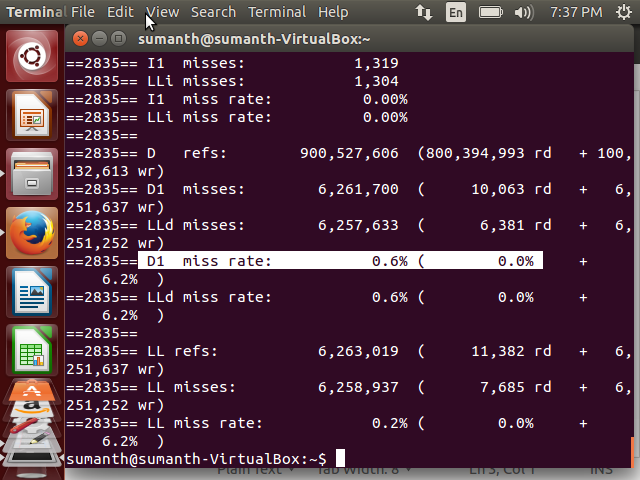
**D1 miss rate for original code:**

****

**Optimised code for Task1:**



**D1 miss rate for optimized code:**

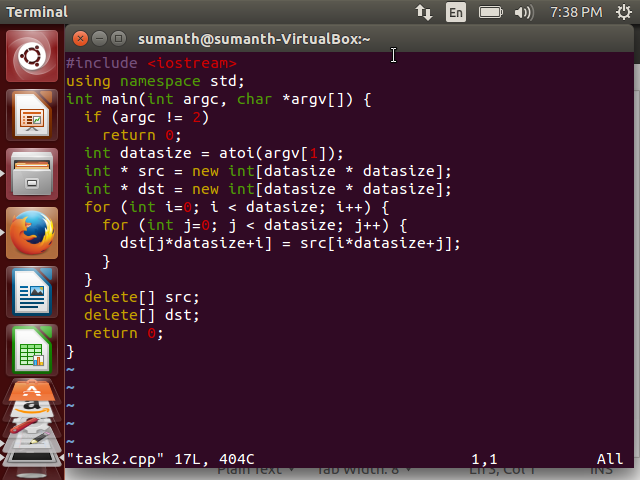
****

**Explanation:**

1. In the optimized code I have interchanged the indexes in the for loop.
2. In the 2-D array after Interchanging spatial locality is improved.
3. In the original code the elements accessed are not sequential.
4. Because the elements accessed are 10000 far away from each other.
5. I have optimized the code by accessing the values sequentially.
6. This technique is called **Loop Interchange** technique**.**
7. The miss rate has gome down because of this technique.

**TASK2:**

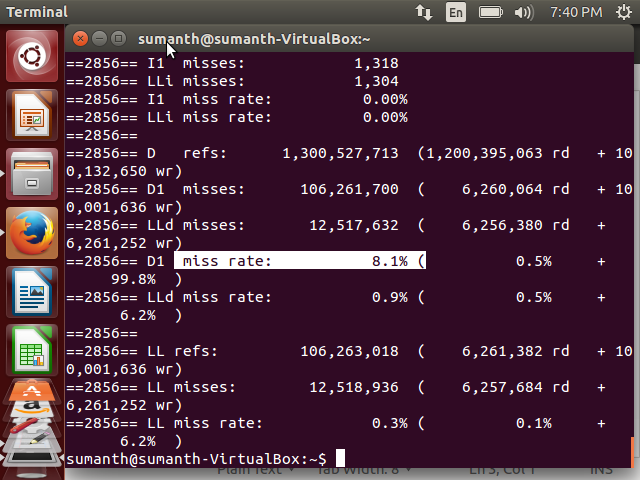
**Original Code:**



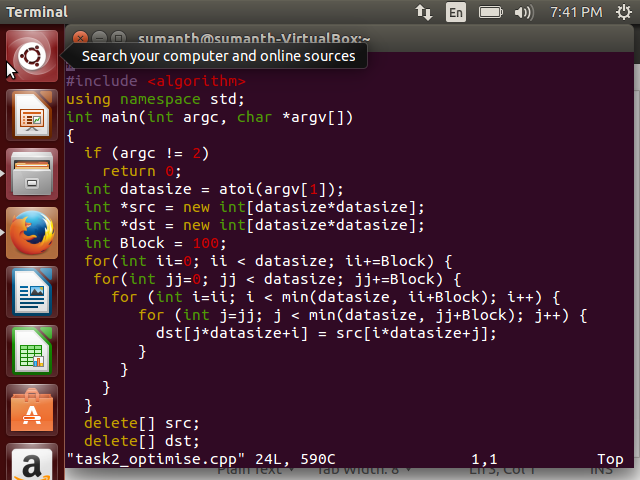
Now, Compiled the original code and ran the valgrind tool to get the D1 miss rate.

The D1 miss rate for original code is as follows:

**D1 miss rate for original code:**



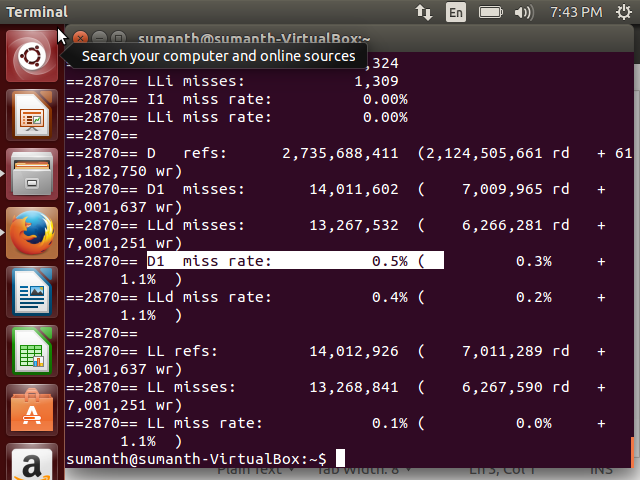
**Optimised code for Task2:**



Now,Compiled the original code and ran the valgrind tool to get the D1 miss rate.

The D1 miss rate for original code is as follows:

**D1 miss rate for original code:**

****

**Explanation:**

1. I used **Blocked algorithm** to optimize the code.
2. Instead of accessing entire rows or columns, I subdivided matrices into blocks
3. Computations are done on blocks which are in Cache.
4. For the destination matrix, We fetched the block of column and source matrix from the cache.
5. The destination matrix benefits from spatial locality and source matrix benefits from spatial locality.
6. Here, We are also reducing the memory access by factor of Block size.