Assignment 2

Aditya Jain (20111004) | Mahi Agrawal (20111029)
adityaj20@iitk.ac.in | maahi20@iitk.ac.in
Group Id: 23

Part 1

Program	Total Machine Access
prog1:	140525433
prog2:	2718547
prog3:	10621695
prog4:	1064755

Table 1. Total Machine Access

Note: Total Machine Accesses will differ from one run to another, these won't be exactly same.

Part 2

Graphs for prog1, prog2, prog3, and prog4 are as follows:

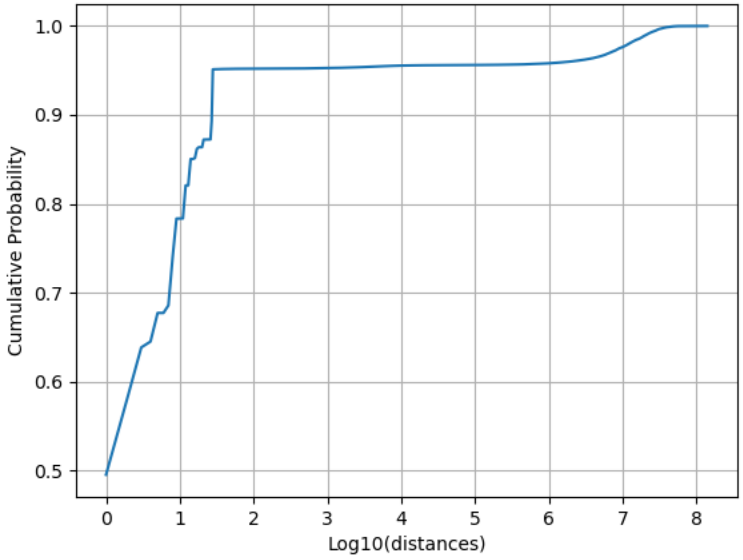

Figure 1. prog1

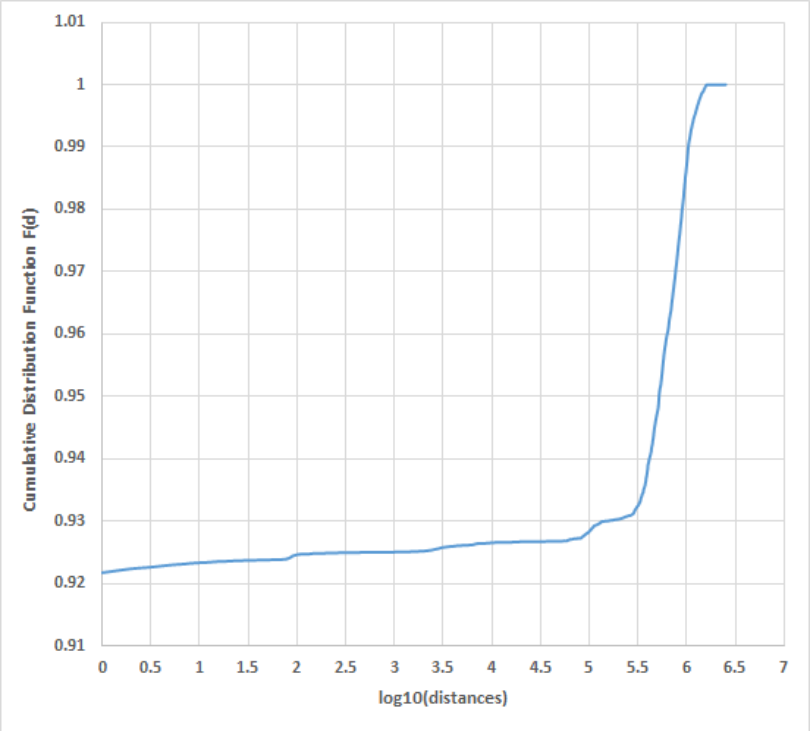

FIGURE 2. prog2

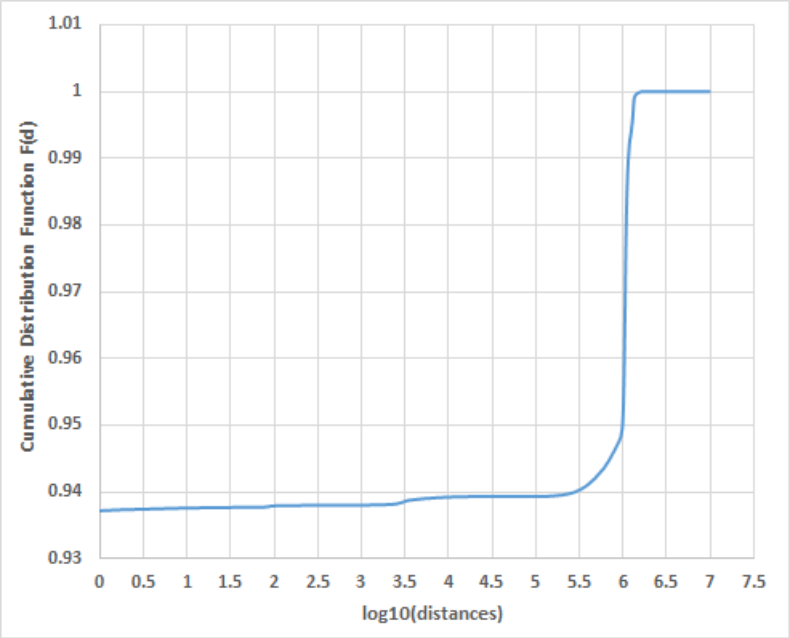

FIGURE 3. prog3

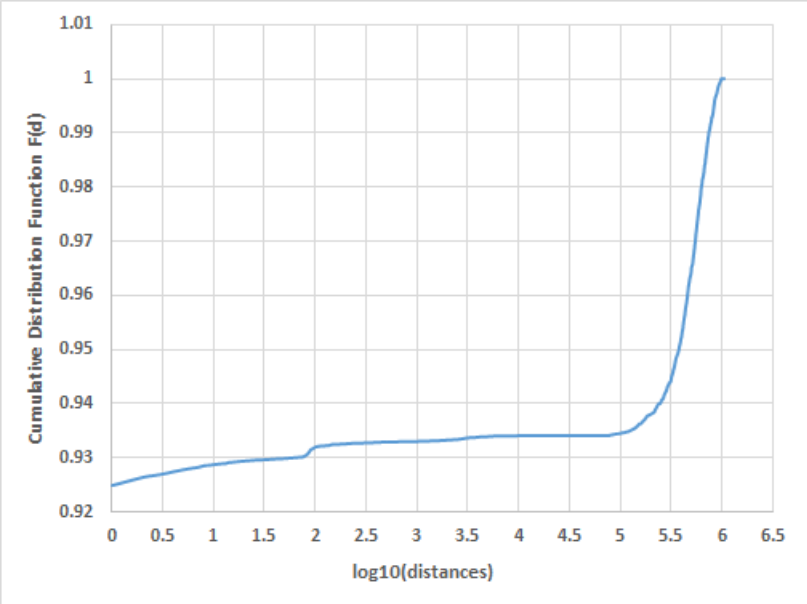

FIGURE 4. prog4

By the above-given graphs, we can say that 92% - 93% of blocks enjoy access distances of 1 in prog2, prog3 and prog4. It essentially means that a block is consecutively accessed one or more times and implies that there is a high chance of them enjoying locality of reference, i.e. a large number of cache hits. Still, these can only be confirmed when we use a cache to do the same analysis.

We can also not say that the program will enjoy a good locality of reference as for the case where block requests are as follows: 2,2,3,3,4,4,5,5,10,17,17... In this, the blocks have access distance as one, but still, the number of cache hits will only be 50% which is not good enough.

In prog1 approximately 95% of blocks have access distance of equal or smaller than 20, so in-fact prog1 might enjoy a good locality of reference and a large number of cache hits, which is again verified in part 3, but the prog1 will have lesser locality as compared to other 3 programs.

As the prog1 has the knee around log(1.3), which means the access distance is approximately 20, and F(20) $\approx$ 95% hence the fully associative cache with the crude upper bound of **20*64 = 1280 byte $\approx$ 2048 byte cache** should be able to capture all the reuses for all the four programs

Note :- For finding crude upper bound we have considered prog1 as the prog1 has the 0.95 at the greater access distance as compared to other programs.

Part 3

TABLE 2. Number of Cache Hits and Misses

	prog1	prog2	prog3	prog4
Total Accesses	140525433	25,18,630	97,90,965	10,64,726
Cache Hits	133733928	22,81,363	91,35,892	9,34,364
Cache Misses	6791505	2,37,267	6,55,073	1,30,362
Hit Rate	95.17%	90.58%	93.31%	87.76%

Graphs of prog1, prog2, prog3 and prog4 after adding cache in the system are as follows:

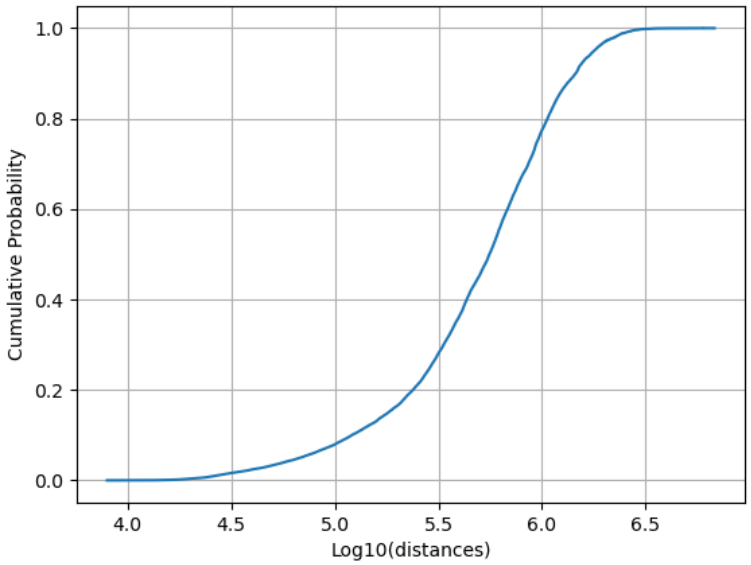

FIGURE 5. prog1

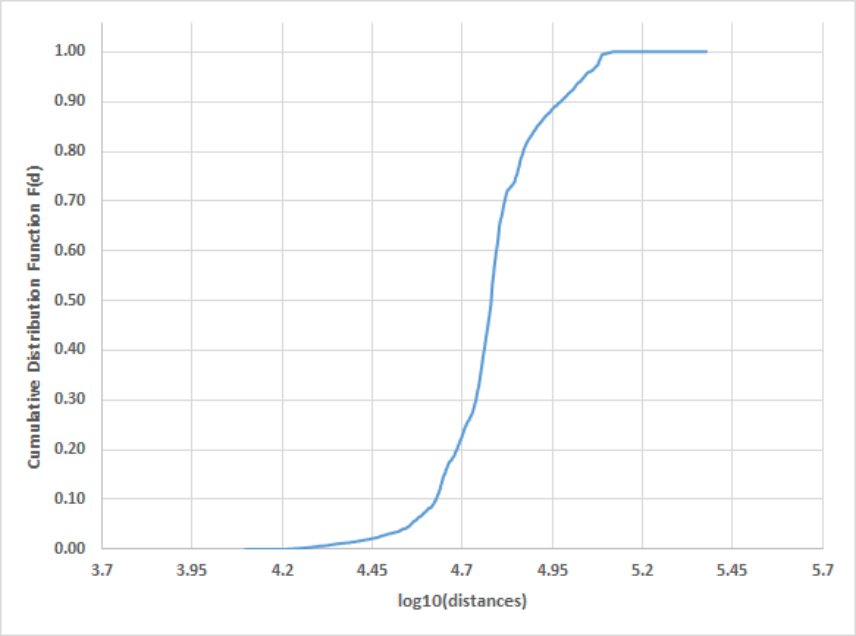

FIGURE 6. prog2

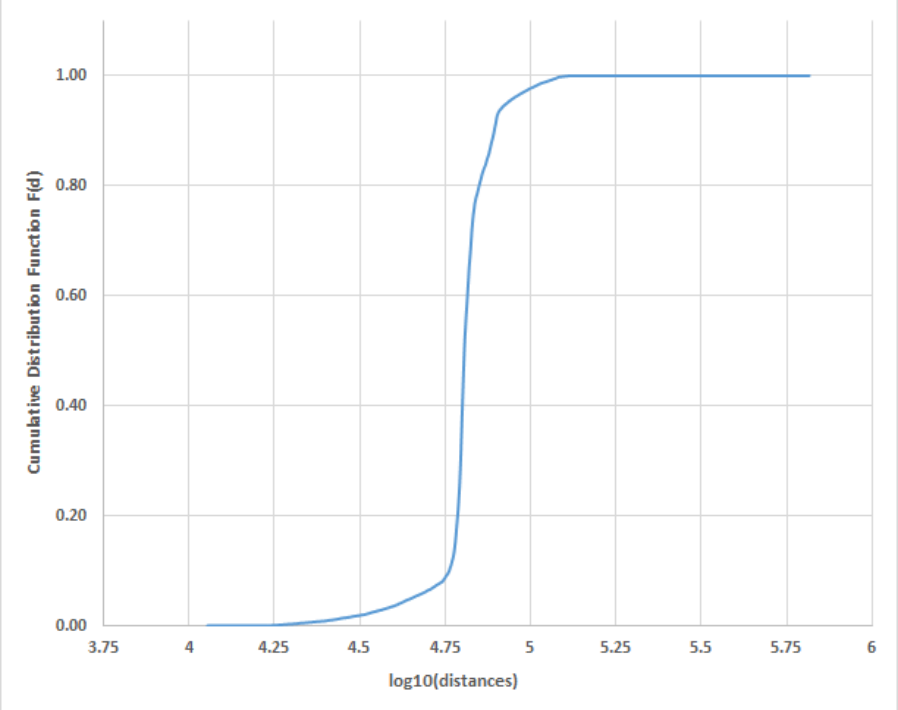

FIGURE 7. prog3

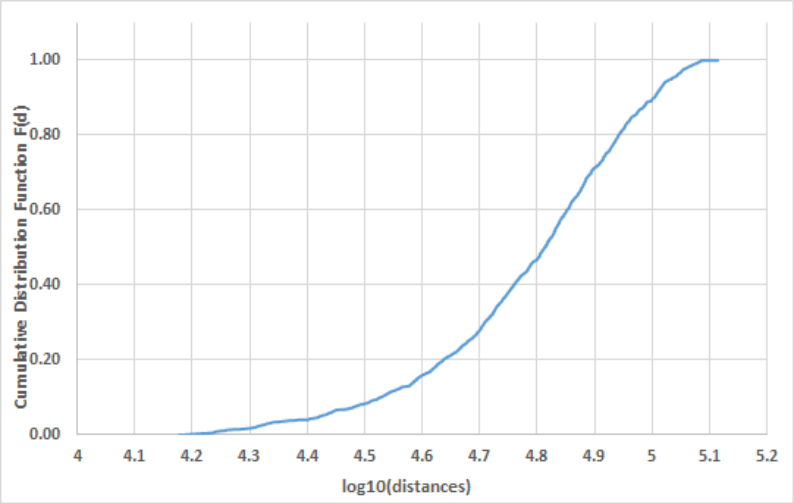

FIGURE 8. prog4

Note: F(d) represents the number access smaller than or equal to d divided by total number of access distance

The curves before and after the cache vary The reason for this is as follows:-

We observed in part 2 that the large number of blocks have access distance equals to 1 for prog2, prog3,prog4 and lesser than or equal to 20 for prog1.

In this part we observed that the F(d) before 4.5 (or access distance 31622) is approximately zero (0). This is because access distance of majority of access is equals to 1,(for prog2 , prog3 and prog4) and less than or equal to 20(for prog1) and all these accesses hit in cache.

If we observe the above two points we can say that the programs enjoy locality of reference `i.e spatial locality and temporal locality` for the large number of accesses due to the smaller access distance which we observed in part 2. This essentially means that large number of accesses hits in the cache and hence do not turn up in the miss trace and hence we can observe F(d) approximately equals to zero before 4.5.

This can also be verified by the Table 2, which shows that hit rate for programs are high ($\approx$ 87% - 95%)

There is a spike in the curve at the log(access distance) $\approx$ 4.5, which correspond to the access distance of **31622**, which approximately equals to the number of blocks in cache `i.e 32768`, this signifies that the large number of cache misses are **capacity misses.**

Part 4

Blocks Shared by	Programs			
	Prog1	Prog2	Prog3	Prog4
Private	432	432	441	8621
Two Threads	70	8262	63	57410
Three Threads	1872	16384	0	6
Four Threads	32456	40958	1	0
Five Threads	143251	5	1	0
Six Threads	244970	0	0	0
Seven Threads	173832	1	1	2
Eight Threads	124528	11	65546	12
Total	721411	66053	66053	66051

TABLE 3. Sharing Profile

By looking at sharing profile we can say that the number of local or the private blocks used by each thread are very less, and number of shared blocks are high. The number of private blocks are almost same for prog1, prog2 and prog3, whereas for prog4 the number of private blocks are comparatively higher. This shows that prog4's execution involves large number of local variables relatively to other programs, moreover most of the blocks are shared by two threads in prog4.

In prog1, a large number of blocks are shared between five, six and seven, and eight threads. It shows that prog1's execution involves a large amount of shared data.