CS622A

ADVANCED COMPUTER ARCHITECTURE

ASSIGNMENT 2

Memory Reuse and Sharing Profile Analysis

GROUP 16

Aditya Rohan
160053

Aniket Pandey
160113

Instructor:
Dr. Mainak Chaudhury

September 24, 2019

1 Introduction

In this assignment, we use PIN tool to instrument a set of parallel programs and collect thread-wise memory access trace and break it down to x86 machine accesses. Then with the resulting trace, we analyze the sharing profile and memory reuse for the given parallel programs.

2 Analysis Results

PART 1: Collection of machine-access traces

The results were varying across individual runs. Hence, we have collected 5 results for a particular program and picked *addrtrace.out* corresponding to the middle value (highlighted).

Programs	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
prog1.c	**128988038**	128988149	128987956	128988046	128987901
prog2.c	2528955	2513452	2521172	**2524574**	2532314
prog3.c	**9508261**	9510696	9501049	9497081	9521463
prog4.c	1061544	1061507	1061492	1061525	**1061515**

Table 1: Machine accesses count across 5 runs

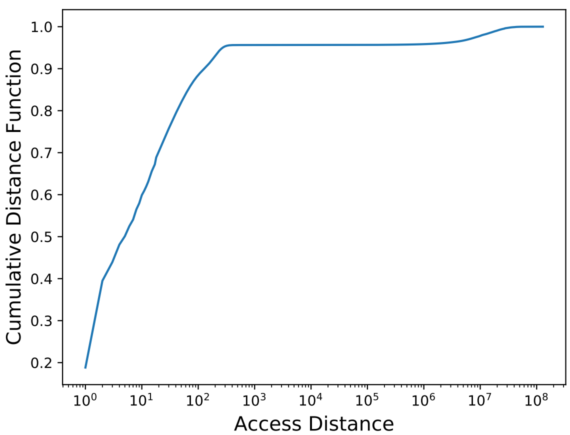
PART 4: Sharing profile analysis

The sharing profile for each of the 4 target programs is given below. The trace corresponding to the highlighted values in part1 were selected for the result analysis.

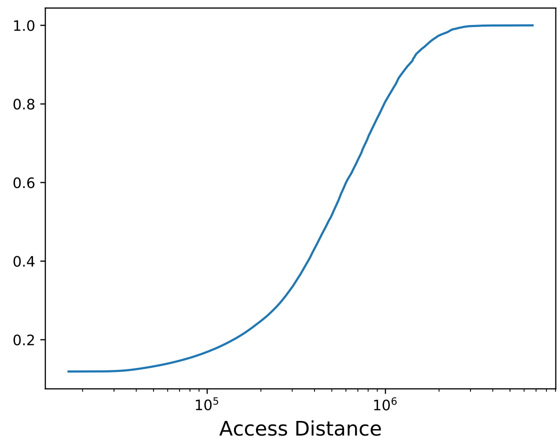
	prog1.c	**prog2.c**	**prog3.c**	**prog4.c**
Private	388	384	386	8573
2-Shared	63	8255	56	57403
3-Shared	1872	16384	0	6
4-Shared	32456	40958	1	0
5-Shared	143251	5	1	0
6-Shared	244970	0	0	0
7-Shared	173831	0	0	1
8-Shared	124527	9	65545	10

Table 2: Sharing profile analyis for 8 threads

PART 2 and 3: Access distance analysis (Normal vs Cache)

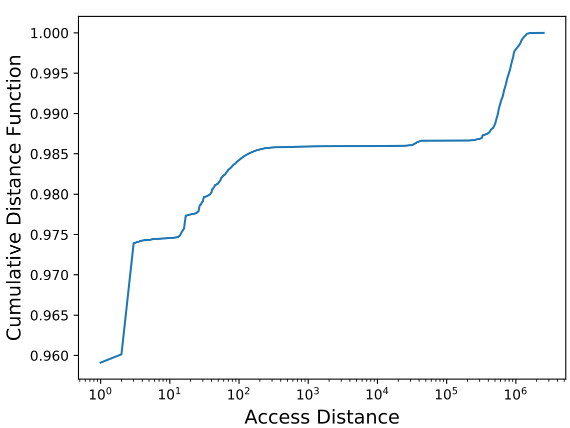


(a) Part2: CDF of access distance for prog1.c

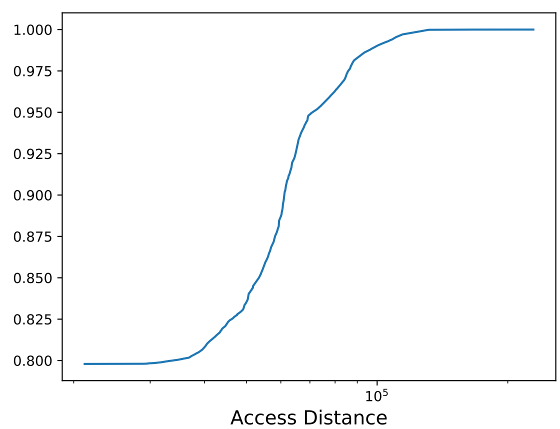


(b) Part3: CDF of access distance for misses of prog1.c

Figure 1: A comparison between the complete machine access trace and missed machine access trace for **prog1.c**

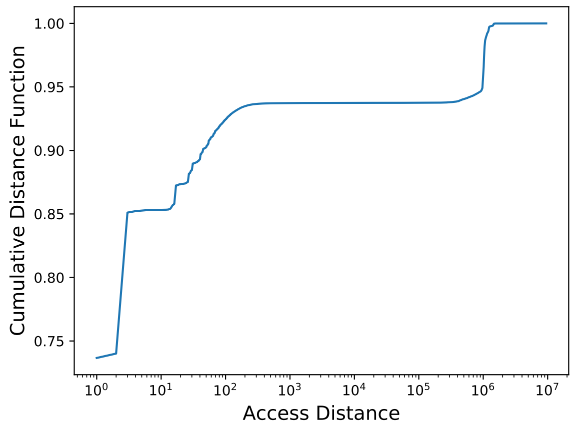


(a) Part2: CDF of access distance for prog2.c

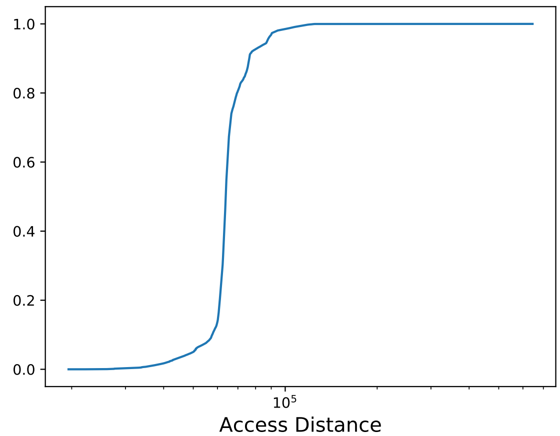


(b) Part3: CDF of access distance for misses of prog2.c

Figure 2: A comparison between the complete machine access trace and missed machine access trace for **prog2.c**

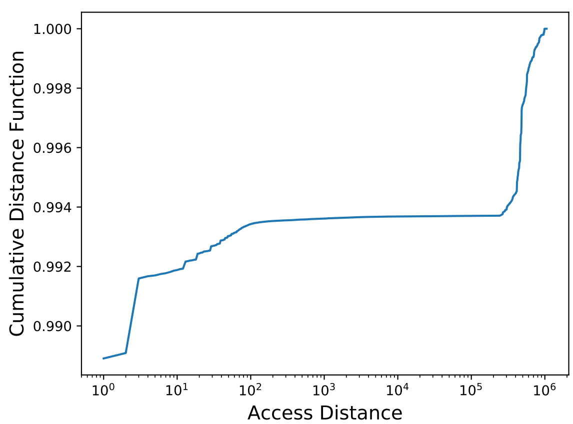


(a) Part2: CDF of access distance for prog3.c

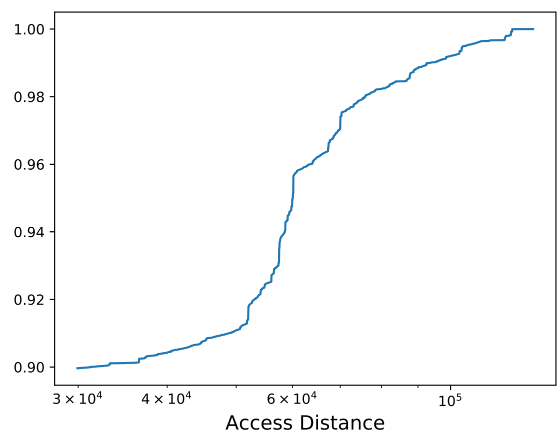


(b) Part3: CDF of access distance for misses of prog3.c

Figure 3: A comparison between the complete machine access trace and missed machine access trace for **prog3.c**



(a) Part2: CDF of access distance for prog4.c



(b) Part3: CDF of access distance for misses of prog4.c

Figure 4: A comparison between the complete machine access trace and missed machine access trace for **prog4.c**

PART 3: Hits and Misses

The cumulative distribution function for part-3 is counted specifically against the misses after modelling the single-level cache to the obtained traces.

	prog1.c	**prog2.c**	**prog3.c**	**prog4.c**
Hits	122297455	2295505	8862896	930888
Misses	6690582	229068	645364	130626

Table 3: Result of modelling a 2MB 16-Way Cache on the traces

3 Comparison of the plots

The obtained plots of cumulative distribution function of access distances for *prog{1, 2, 3, 4}.c* show subtle yet noticeable differences when taken the original trace and after modelling through 2MB 16-way single level cache and focussing only on miss traces.

1. The cdf-plots for original trace have uneven slope while the cdf-plots for miss traces shows a proper *sigmoidal* curve, which is generally seen for *(Gaussian) Normal distribution.* This implies even if the program has uneven access pattern, the hit/miss pattern solely depends on cache specifications and is more-or-less similar for any program, on a given trace.

2. The initial slope of cdf-plot for miss trace is *zero* and increase gradually. While for the original plots, there is a sudden increase initially which evens out in the end. This can be explained due to *temporal locality* of accesses, where programs access nearby elements quickly in a short duration which results in high number of low-access-distance patterns.

3. The slope for cdf-plots of miss traces starts out at *zero* because it is expected that for a reasonable cache specification, low-access-distance for a block will generally result in a hit. That is, misses will be observed for blocks with high-access-distance as the block might get evicted, following the eviction policies like LRU.

4. The cdf-plot for miss traces show a sudden jump midway (observed in Gaussian curves). This can be attributed to the fact that there is a certain limit (e.g 6×10^4) beyond which the cache capacity is exceeded and we see a large number of misses. This lets us know about the critical access distance of the blocks and that accesses below it have a greater chance of registering a hit.

5. A comparison between plots of Part 2 and Part 3 also shows that a similar percentage of reuse, say 90%, requires a larger sized next-level cache. The cumulative distribution function for prog1.c is shown in **??** & **??**. Part 3 requires 10^4 times more blocks than Part 2, to accomodate 90% reuse.