

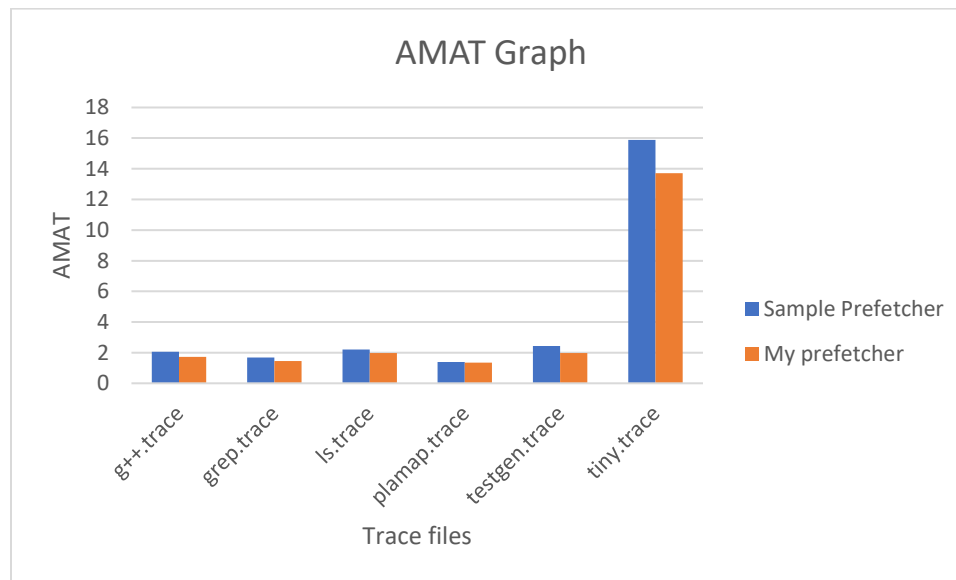
CS 203 Project Report

Index-based Stride Prefetcher

Description of prefetcher: The prefetcher technique I've used is based on **Stride Directed prefetching**. I've implemented a prefetcher that calculates stride between consecutive requests but instead of using the complete address for stride calculation, I'm using only the index bits of the addresses. The indexes are found by calculating the number of sets based on the D-cache and L2-cache configurations. This requires less state variable and space and is simple to implement than the regular stride prefetchers. The address parameter of `_nextReq` is fed with sum of the requested address and the recently calculated stride corresponding to indexed entry in the table. In order to check the confidence levels of the calculated stride, I've included a parameter 'state' which informs if the stride is steady or not. Also, in case of unsteady strides, prefetcher fetches the next block (current address + 32, where 32 is added since each L2 cache line is of length 32 bytes). Entire algorithm runs if `_ready` is false or if there is a miss in L2 cache. This helps reduce the long memory accesses to disc from L2.

State Accounting: The states used for the implementation are an instance of structure `Request: nextReq` which is used to feed the prefetch data into, a state variable `_ready` to check the request status in the prefetcher and another structure array 'table' containing address (`u_int32_t`), stride (`int`) and state (`int`). The length of the array 'table' is 1024 corresponding to the 10-bit index and is used to store the entries of incoming requests.

AMAT Graph: The graph below shows the average memory access times of the implemented prefetcher alongside the sample prefetcher for all the trace files.



As we can see in the graph above, the implemented prefetcher performs better than sample prefetcher in all the trace files.

References:

1. Stride Directed Prefetching: <https://dl.acm.org/doi/pdf/10.1145/144965.145006>
2. Reference Prediction Table: http://web.cecs.pdx.edu/~alaa/ece587/papers/chen_ieeetoc_1995.pdf
3. Prefetching lecture by Prf. Onur Mutlu: <https://youtu.be/xZmDyj0g3Pw>