

# Improving the performance of Last Level Cache

Sanjana Garg, 13617

## Motivation

A cache miss in the last-level cache is very expensive because of a very high latency. Therefore, it is important to improve the cache hit rate in last-level cache especially for memory intensive applications. But if the working set of an application (the set of most frequently accessed memory locations) is greater than even the size of the last level cache, it can result in a very poor cache performance. This is termed as thrashing. Also there might be a set of data references which are referenced not in the immediate future. Many real world applications have this set of data references in their cache patterns. One set of such data references is termed as a scan. Therefore, it is also important to make the last level cache scan resistant.

## Aim

Through this project I intend to compare the performance of LRU (most commonly used cache policy) and cache insertion and replacement policies that are designed to make the cache thrash-resistant and scan-resistant. First, analyse the cache performance using thrash resistant policies for applications with a large working size. Then using scan-resistant policies for applications with scans having re-reference in distant future. Finally, design a policy that is a mix of both scan-resistant and thrash-resistant and analyse its performance.

## Tool and benchmark applications

I would be using pintool to analyse the applications. The applications I would be using are mcf, sphinx3, hmmer, bzip2, cactus, PC games and multimedia applications.