VIKRAM SRI NITESH TANTRAVAHI
13247361
DREXEL UNIVERSITY

Improving Direct-Mapped Cache Performance by the Addition of a Small Fully-Associative Cache and Prefetch Buffers

Norman P.Jouppi

Paper Review

1. The Problem

This paper is a new idea that focuses mainly on improving the performance of caches and investigates on new hardware techniques for increasing the performance of the memory hierarchy and also the cache. The cache discussed here in this paper is majorly the direct mapped cache on which some of the techniques like comprehending an additional hardware that resides out the CPU core or adding small fully associative cache and Prefetch buffer. It also depicts the relationship between the cost of miss rates, main memory access time and machine cycles per instruction in the baseline design aspect of caching techniques.

Also this paper constitutes of other various procedures like *Miss caches*, *Victim caches*, *Stream buffers*, and *Multi-way stream buffers* that can dramatically reduce cache miss rates and improve system performance. The reduction of reducing capacity and compulsory misses with prefetch techniques such as longer cache line sizes or prefetched methods are also discussed. These mapping conflicts among various techniques are basically because of lack of associativity. Other key topics are the first level and second level caches in terms of their individual performances and the effect of the direct-mapped cache of victim cache performance or line cache performance.

2. Evaluation & Analysis

The cache performance and optimization has always been the major concern for the advanced processors and also the cache miss times that affect the machine performance. One of the major factors disrupting the performance, mentioned here are the misses in the cache, which are categorized into four types namely; *Conflict, Compulsory, Capacity and Coherence.* The miss cache and victim cache play an important role in reducing the conflict misses. This is showed in the paper with some examples like "an instruction reference stream that calls a small procedure in its inner loop that conflicts with the loop body". But this has a kind of drawback of data duplication. The workloads that are used in the paper like *ccom*, *grr*, *yacc*, *met*, *linpack* and *liver* are like used for minute and small programs and the analysis made in the paper cannot be extended to a broader view and not applicable for all the cases.

The statistics and the studies made in the paper are pertained to few and small programs and we cannot make assumptions and relate them to the real time processors. Also the results in the paper do not show the actual numbers and represents percentages. The major influencing factors that we depend on, from the results are the execution time and the number of cycles but no where this shows us in the final improved performance as shown below.

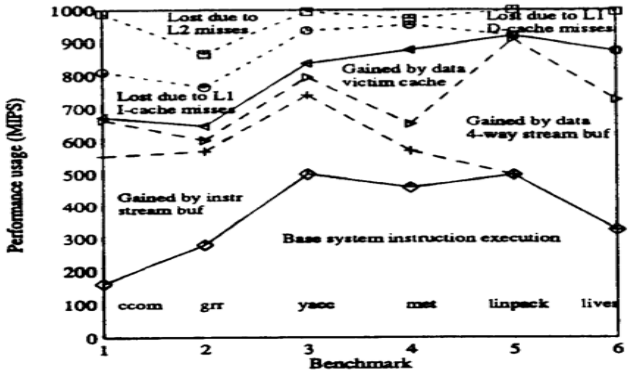

Fig: Improved system performance in paper.

3. Results

This proposed solution might be implemented in terms of cost because if the associativity is less, it is simpler and cheaper. But there can be further amendments made to this in terms of including heavy workloads that can be considered, also more about the second level cache and with results more precise compared to the existing ones and performance of a system cache or memory hierarchy in terms of execution time and number of cycles in implementation.