

# Cache Miss Pattern and Miss Predictability Analysis

The efficiency of a computer system in executing a program is highly dependent on the effectiveness of the memory hierarchy to supply requested data quickly. However, since the structure of the processor's memory hierarchy is almost always fixed, the overall efficiency depends on how well the program reference pattern matches the given cache structure. The memory reference pattern of the program itself depends on the particular data set used for input. Thus, it is necessary to understand the fundamental memory reference behavior of a program across all input data sets to determine how effective the memory hierarchy will be at efficiently supporting the program in general.

The effectiveness of memory cache hierarchies depends on the locality of data accesses in the programs. Past work mainly provides three ways of locality analysis: by a compiler, which models loop nests but is not as effective for dynamic control flow and data indirection; by frequency profiling, which analyzes a program for select inputs but does not predict the behavior change in other inputs; or by runtime analysis, which cannot afford to analyze every access to every data. None of these methods adequately provides the capability of predicting the memory reference patterns across a broad range of programs and data input sizes.

Improving cache performance requires understanding cache behavior. However, measuring cache performance for one or two data input sets provides little insight into how cache behavior varies across

all data input sets and all cache configurations. This project uses locality analysis to generate a parameterized model of program cache behavior. Given a cache size and associativity, this model predicts the miss rate for arbitrary data input set sizes. This model also identifies critical data input sizes where cache behavior exhibits marked changes.