CS 5341

Oscar Galindo

**Predicting Whole-Program Locality through Reuse Distance Analysis**

This paper mainly develops the idea of utilizing the concept of *Reuse Distance*, or the amount of distinct data elements between two accesses to the same data element. As mentioned, the intention of this work is not to make a prediction of how long a program will take to execute, rather the focus of the work is on the prediction of individual program execution behavior. The behavior observed in programs can be used to analyze how different situations (i.e., inputs, sequences, etc.) can modify the execution of a program.

The bulk of the work show in the paper is found in three different contributions. First is the *approximate reuse distance analysis* which is a methodology that provides with a bounded relative error “*e*”, or alternatively with a bounded absolute error “*B*” an approximation to the actual reuse distance. The implementation outlined in the paper for the *approximate reuse distance analysis* is possible using tree-data structures which make possible the assessment which otherwise would be to expensive memory-wise with long data access sequences/reuse distances. The paper indicates asymptotically speaking that the variant with a bounded relative error is superior in terms of space and time cost for any error “that is greater than zero”. The second methodology presented in this paper is the *pattern recognition* mechanism, which aims at “[detecting] whether the recurrence pattern is predictable across different data inputs”. In other words, the methodology detects foreseeable patterns of memory used based on input size. To perform the pattern recognition, this methodology proposes the use of histograms that quantify the percentage of memory accesses within a specific range of reuse distance. Given the constructed histograms the methodology proposes to solve a system of linear equations to solve for specific coefficients for each histogram bin across all “training” samples, so that then the calculated coefficients are then used for the prediction on reuse-distance for new programs. This methodology, of course comes with the limitations of higher error with more deviation of data “shape” between training and testing samples, and of size to offset the non-recurring data accesses. Lastly, the third contribution on this paper is called *Distance-based sampling*. This last methodology proposes to utilize a sampler that calculates reuse-distances among several data entries at the beginning of the execution. The “peaks” or higher-than-a-threshold reuse distance are used to provide upper bounds on the size of the entire data and hence the upper bound of reuse-distance.

Among the results shown in the paper there is the observation that “compared to accurate methods, approximate analysis is faster and more scalable with data size and distance length.” Which is an expected result since less memory is used than in accurate methods and the error is controlled to have a very accurate prediction. The paper also offers as a result that by performing pattern prediction the programs are very well predicted in terms of reuse distance for their data accesses regardless of their data size and difference in histograms. In addition, the experiments presented in this paper show backward and forward reuse-distance prediction are both accurate. Finally, the paper suggests although in some cases utilizing the *Distance-based methodology* approximation-result can be used as means to approximate the average reuse-distance through a simple multiplication time a known fraction, the approximation is still limited to only providing an estimate and lacks the robustness of knowing the distribution of reuse-distances.

Lastly, the paper offers some uses for the methodologies proposed. Among the uses are File Caching -to adjust distribution policies of the cache memories- and in Compilers Design – to explore the limits of register reuse and cache reuse, etc.

My opinion of this paper is that it presents interesting proposals for more comprehensive methodologies for memory use. I believe if anything of this paper is debatable is the inclusion of a technique like distance-based sampling, since as it is mentioned, we could obtain an estimate of the data size but not of reuse distance as the paper seems to be focused on. Also, I believe the compression algorithm presented in the paper may be a little bit too big, and the proofs a little too difficult to follow. Also, I which that there was the inclusion of a section that described further what to do specifically with the reuse distance in terms of cache management.