Increasing and Detecting Memory Address Congruence

Samuel Larsen, Emmett Witchel, and Saman Amarasinghe

***Abstract*—A static memory reference exhibits a unique property when its dynamic memory addresses are congruent with respect to some non-trivial modulus. Extraction of this congruence information at compile-time enables new classes of program optimization. In this paper, we present methods for forcing congruence among the dynamic addresses of a memory reference. We also introduce a compiler algorithm for detecting this property. Our transformations do not require interprocedural analysis and introduce almost no overhead. As a result, they can be incorporated into real compilation systems.**

On average, our transformations are able to achieve a five-fold increase in the number of congruent memory operations. We are then able to detect 95% of these references. This success is invaluable in providing performance gains in a variety of areas. When congruence information is incorporated into a vectorizing compiler, we can increase the performance of a G4 AltiVec processor up to a factor of two. Using the same methods, we are able to reduce energy consumption in a data cache by as much as 35%.

[Full Text Not Available]