Reducing Consistency Traffic and Cache Misses in the Avalanche Multiprocessor

John B. Carter, Ravindra Kuramkote, Chen-Chi Kuo

UUCS-95-023

Computer Systems Laboratory
University of Utah

Abstract

For a parallel architecture to scale effectively, communication latency between processors must be avoided. We have found that the source of a large number of avoidable cache misses is the use of hardwired write-invalidate coherency protocols, which often exhibit high cache miss rates due to excessive invalidations and subsequent reloading of shared data. In the Avalanche project at the University of Utah, we are building a 64-node multiprocessor designed to reduce the end-to-end communication latency of both shared memory and message passing programs. As part of our design efforts, we are evaluating the potential performance benefits and implementation complexity of providing hardware support for *multiple coherency protocols*. Using a detailed architecture simulation of Avalanche, we have found that support for multiple consistency protocols can reduce the time parallel applications spend stalled on memory operations by up to 66% and overall execution time by up to 31%. Most of this reduction in memory stall time is due to a novel release-consistent multiple-writer write-update protocol implemented using a *write state buffer*.