Report on “Increasing the Effectiveness of Directory Caches by Deactivating Coherence for Private Memory Blocks”

This paper is focused upon improving directory caching schemes by paring down the number of shared blocks that need to be managed. Generally, cache performance is degraded by two factors: (1) premature cache evictions, which reduces application performance by causing cache misses; and (2) access latency, which reduces application performance by degrading data acquisition rates. Obviously, one approach to reducing premature evictions is to increase the size of the cache. However, not only does this increase hardware costs, it also increases access latency. Thus, cache size and access latency are negatively linked; increasing cache size decreases evictions but increases latency; decreasing cache size has the opposite effect.

The approach that the authors have chosen to this problem starts with a few premises. First, application performance would improve with more available directory cache space, especially if this could be provided without an increase in latency. Second, directory caches, by default, assume that blocks used by parallel applications are to be shared. Third, most blocks used by parallel applications are used by only one processor, i.e. do not need to be shared amongst multiple processors.

Combined, the second and third premises lead inexorably to an important conclusion: many, if not most, blocks maintained by directory caches do not, in fact, need to be maintained by directory caches. When this conclusion is further combined with the first premise, an opportunity is recognized: parallel application performance would improve if directory caches did not maintain blocks, even for parallel applications, that will never be used by more than one processor.

The authors implement this realization by limiting the cached blocks that are treated as shared, i.e. the blocks for which coherence is a concern. The basic idea is that, initially, all blocks (actually, the blocks, and their status, are managed within pages) are marked as private. So, as long as a cached block B is only requested by a single processor, it remains private. However, once a different processor attempts to access B, it is marked as shared and coherence management begins for B.

Given that many future performance increases will come from the use of parallelism, efficiently maintaining shared caches for parallel processors is an important problem. The authors of this paper have indeed made some headway with this difficult problem by implementing an idea that is very intuitive: do not included private blocks within the directory cache. With this realization in hand, future directory caching schemes will likely use it as a starting point to preserve valuable cache space.

Reference