Reducing Data Movement using Cache-Oblivious Layouts

Distributed and Multiresolution Streaming Analysis of Petascale Data

Peter Lindstrom
Lawrence Livermore National Laboratory 7000 East Ave, Livermore, CA 94550

Abstract

As the high-performance computing community pushes toward exascale computing, one of the key roadblocks to scalable performance is how to limit data movement. On massively multicore, distributed architectures, such data movement occurs between NUMA memory banks, across distributed compute nodes, and between main memory and disk, and is expected to dominate power consumption and severely limit processing throughput. To partially alleviate this bottleneck, it is widely believed that future computers will employ deeper and wider cache hierarchies, possibly by making use of new technologies such as solid state storage.

In order to make effective use of such caches, both in simulation codes and in subsequent data analysis, applications will have to reorder computations—e.g. using sequential, streaming access—and data layouts to improve spatial and temporal locality of reference. However, developing optimal data layouts for unstructured data, possibly with multiple mesh centerings, is a challenging problem. In addition, because of the diversity of caches in terms of capacity, line size, associativity, and replacement policy, it becomes impractical to optimize data layouts for any one cache. A layout explicitly optimized for a particular cache usually results in highly suboptimal performance with respect to a different-size cache, in effect canceling any potential performance gains.

To address this challenge, we have developed *cache-oblivious layouts* for unstructured data that provide good and reliable average-case performance across the whole memory hierarchy, without optimizing for any particular cache. We model the data using an undirected *affinity graph*, which connects data elements with edges to indicate a desire to store those data elements close together on linear storage. The nodes of this graph are then permuted so as to minimize a locality functional that models the memory hierarchy as a sequence of nested caches of geometrically increasing size. We solve this linear ordering problem effectively using a multilevel procedure inspired by algebraic multigrid. The resulting layouts are "fractal" in nature, and are in a sense algebraic generalizations of space-filling curves to unstructured graphs that require no geometric embedding.

We show that our layouts result in dramatic reductions in cache misses over bandwidth-reducing layouts in sparse matrix-vector multiplication and other kernels, and provide recipes for ordering unstructured data within and across subdomains in domain-decomposed meshes for distributed streaming analysis. Moreover, our layouts find utility in graph partitioning for reducing inter-node communication volume, and allow graphs to be partitioned "instantly" by dividing linear storage into equal-sized blocks of nodes, each of which intrinsically exhibits good locality. Finally, we propose a new space-filling curve that minimizes our locality functional, and whose properties with respect to several locality measures improves on the best-known orderings of Cartesian grids.