Parallel Algorithms and their Performance on 64-core Tilera Chips

SPEAKERS

Bob Benner and Uzoma Onunkwo

Sandia National Laboratories

Emerging multi-core chips and GPUs can potentially be applied to a wide spectrum of numerical algorithms, methods and applications. In this talk, we will discuss our work on 64-core Tilera chips, including the implementation and performance of parallel hashing algorithms.

In the first part of our talk, we will go over the features of the Tilera chipsets. We will also discuss memory management and other tools necessary for profiling and achieving high-end solutions with Tilera. We then discuss some achievable parallel efficiency in the Tilera system using a coarsely parallel program, multi-processor scheduler benchmark.

In the other half, we will discuss Hash table implementation on Tilera systems. A Hash table is an important data structure with usefulness ranging from caching to database indexing. Parallel hash tables have a long, colorful, and often dismal history, dating back to the earliest commercial shared and distributed memory systems of the mid-1980's.

We have recently implemented several hash algorithms on the 64-core Tilera chip and have some very promising results. On the 2nd generation, Tile64Pro chip we see parallel efficiencies of 50-90% for various hash algorithms and hash table sizes on up to 57 tile processors. Why 57 tiles? The current default system configuration reserves 7 tiles for various system tasks. Several other architectural factors must be accounted for when considering potential applications for Tilera. For example, the 64 cores are integer cores – all floating point computations are done via emulation – making it useful for fixed-point based solutions. Most important, the chip supports both shared memory and private memory – but, since both kinds of memory are off-chip, shared memory implementations are preferable. The chip also has five independent inter-processor communication grid, some of which are reserved for system tasks. We will not even attempt to describe user-defined and system-controlled cache management strategies until the talk itself.