



Carnegie Mellon University

Computer Science Department

Speaking Skills Talk

Hierarchical Memory Management for Parallel Programs

Ram Raghunathan

Friday, September 15, 2017

12:00 pm

GHC 6501

An important feature of functional programs is that they are parallel by default. Implementing an efficient parallel functional language, however, is a major challenge, in part because the high rate of allocation and freeing associated with functional programs requires an efficient and scalable memory manager.

In this talk, we present a technique for parallel memory management for strict functional languages with nested parallelism. At the highest level of abstraction, the approach consists of a technique to organize memory as a hierarchy of heaps, and an algorithm for performing automatic memory reclamation by taking advantage of a disentanglement property of parallel functional programs. More specifically, the idea is to assign to each parallel task its own heap in memory and organize the heaps in a hierarchy/tree that mirrors the hierarchy of tasks.

We present a nested-parallel calculus that specifies hierarchical heaps and prove in this calculus a "disentanglement" property, which prohibits a task from accessing objects allocated by another task that might execute in parallel. Leveraging the disentanglement property, we present a garbage collection technique that can operate on any subtree in the memory hierarchy concurrently as other tasks (and/or other collections) proceed in parallel. We prove the safety of this collector by formalizing it in the context of our parallel calculus. In addition, we describe how the proposed techniques can be implemented on modern shared-memory machines and present a prototype implementation as an extension to MLton, a high-performance compiler for the Standard ML language. Finally, we evaluate the performance of this implementation on a number of parallel benchmarks.