UCF DEPARTMENT OF COMPUTER SCIENCE

Fall 2015 Seminar Series

OPTIMIZING DYNAMIC TASK-PARALLEL EXECUTION ON NUMA SYSTEMS

TUESDAY DECEMBER 1, 2015

11:00 AM – HEC 438

Dynamic task-dependent parallelism is an increasingly popular programming model on shared-memory systems. Compared to data parallel loop-based concurrency, it promises enhanced scalability, load balancing and locality. These promises, however, are undermined by non-uniform memory access (NUMA). We show that it is possible to preserve the uniform hardware abstraction of contemporary task-parallel programming models, for both computing and memory resources, while achieving near-optimal data locality. Our run-time algorithms for NUMA-aware task and data placement are fully automatic, application-independent, performance-portable across NUMA machines, and adapt to dynamic changes. We take advantage of data-flow style task parallelism, where the privatization of task data enhances scalability through the elimination of false dependences, and enable fine-grained dynamic control over the placement of application data. In a second part, we present Aftermath, an interactive visualization tool for post-mortem trace analysis of cross-layer performance anomalies in dynamic task-parallel applications. We focus on the detection of anomalies inaccessible to state-of-the-art performance analysis techniques, including anomalies deriving from the interaction of multiple levels of software abstractions, anomalies associated with the hardware, and anomalies resulting from interferences between optimizations in the application and run-time system. Aftermath supports filtering, aggregation and joint visualization of key metrics and performance indicators, such as task duration, run-time state, hardware performance counters and data transfers. While not being specifically designed for NUMA architectures, Aftermath takes advantage of the explicit memory regions and dependence information in dependent task models to precisely capture long-distance and inter-core effects on complex, dynamic applications.

DR. ANTONIU POP
University of Manchester

Antoniu Pop holds a Royal Academy of Engineering University Research Fellowship and is a Lecturer in the Advanced Processor Technologies group at the University of Manchester. He conducts research in multiple aspects of parallel software including programming languages, program analysis, data-flow models of computation and compiler and runtime optimisations for parallel and high-performance computer systems. He is the designer of the OpenStream language and has contributed to the GCC compiler. Antoniu is involved in multiple EU and UK projects investigating data-flow programming models for many-core architectures and in the EuroExa projects, co-designing the European hardware and software architecture for exa-scale computing.

Hosted by: Dr. Damian Dechev