

Programming Models for Concurrency and Real-Time

Jan Vitek

Computer Science Department
Purdue University
Jv@cs.purdue.edu

Abstract. Modern real-time applications are increasingly large, complex and concurrent systems which must meet stringent performance and predictability requirements. Programming those systems require fundamental advances in programming languages and runtime systems. This talk presents our work on Flexotasks, a programming model for concurrent, real-time systems inspired by stream-processing and concurrent active objects. Some of the key innovations in Flexotasks are that it support both real-time garbage collection and region-based memory with an ownership type system for static safety. Communication between tasks is performed by channels with a linear type discipline to avoid copying messages, and by a non-blocking transactional memory facility. We have evaluated our model empirically within two distinct implementations, one based on Purdue's Ovm research virtual machine framework and the other on Websphere, IBM's production real-time virtual machine. We have written a number of small programs, as well as a 30 KLOC avionics collision detector application. We show that Flexotasks are capable of executing periodic threads at 10 KHz with a standard deviation of 1.2us and have performance competitive with hand coded C programs.