

Dynamic CPU Management for Real-Time, Middleware-Based Systems

Eric Eide, Tim Stack, John Regehr, and Jay Lepreau

UUCS-04-004

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

January 30, 2004

Abstract

Many real-world distributed, real-time, embedded (DRE) systems, such as multi-agent military applications, are built using commercially available operating systems, middleware, and collections of pre-existing software. The complexity of these systems makes it difficult to ensure that they maintain high quality of service (QoS). At design time, the challenge is to introduce coordinated QoS controls into multiple software elements in a non-invasive manner. At run time, the system must adapt dynamically to maintain high QoS in the face of both expected events, such as application mode changes, and unexpected events, such as resource demands from other applications.

In this paper we describe the design and implementation of a *CPU Broker* for these types of DRE systems. The CPU Broker mediates between multiple real-time tasks and the facilities of a real-time operating system: using feedback and other inputs, it adjusts allocations over time to ensure that high application-level QoS is maintained. The broker connects to its monitored tasks in a non-invasive manner, is based on and integrated with industry-standard middleware, and implements an open architecture for new CPU management policies. Moreover, these features allow the broker to be easily combined with other QoS mechanisms and policies, as part of an overall end-to-end QoS management system. We describe our experience in applying the CPU Broker to a simulated DRE military system. Our results show that the broker connects to the system transparently and allows it to function in the face of run-time CPU resource contention.