Real time, Non-Intrusive Instrumentation & Monitoring of Standards-based Event-based Middleware

Geetha R. Satyanarayana, Liren Tu, James H. Hill
Department of Computer & Information Science
Indiana University – Purdue University, Indianapolis

Standards-based middleware, such as the Common Object Request Broker Architecture (CORBA) Component Model and the Data Distribution Service, support event-based services for decoupled, asynchronous messaging between software components in a distributed system. The messaging models use the publisher-subscriber paradigm where one or more subscribers can subscribe to events from one or more publishers. The advantage of this paradigm is neither the publisher nor subscriber needs to be aware of the other. Instead, either entity is only concerned with the publication or receipt of an event. A critical aspect of these systems, however, is their instrumentation for analysis purposes, like monitoring its performance, state, and behavior to ensure the system is executing as expected.

Traditionally, instrumenting such systems relied on intrusive instrumentation approaches, where developer inserted code snippets into the source code to collect the information needed. This means that developers must understand the original code, run the risk of inserting malicious code, and intermix code related to instrumentation with the normal business logic. Moreover, as the normal business logic evolves, the instrumentation code must also evolve. This can become a burden on developers until the business logic become less volatile.

To overcome this complexity, we present Dynamic Event Monitor, a tool that can non-intrusively instrument and monitor events in a large scale distributed system at run time, using dynamic binary instrumentation. It operates in contexts without any *a priori* knowledge of the concrete events in the system, or how the system is composed. We have applied Dynamic Event Monitor to applications implemented in CORBA. Our results show that once the application is completely instrumented, the performance impact of actually monitoring events is minimal. For the applications we have tested, the instrumentation time is about 30-45 seconds and the time for real-time monitoring of events is about 2 milliseconds.