EPOS: Paving the Path for Parallel Applications

Antônio Augusto Fröhlich
GMD FIRST
kekuléstraße 7
12489 Berlin, Germany
guto@first.fhg.de
http://www.first.fhg.de/∼guto

Abstract

Every time more applications demand performance levels that can only be achieved by parallelization. In order to properly support them, new operating systems and tools are to be conceived. Our experiences developing runtime support systems for parallel applications convinced us that adjectives such as "generic" and "all purpose" do not fit together with "high performance", whereas different parallel applications have quite different requirements regarding the operating system. Even apparently flexible designs, like micro-kernel based operating systems, may imply in waste of resources that, otherwise, could be used by applications. The promotion of configurability has been properly addressed by the PURE operating system. PURE is designed as a collection of configurable classes that can be seen as building blocks to assemble application-oriented operating systems. Approaches like this, although doing much for performance, reusability and maintainability, usually are not enough to support application programmers, since the number and the complexity of available building blocks grows quickly with the system evolution. In such a context, selecting and configuring the proper building blocks becomes a nightmare and yields a gap between that what the operating system offers and that what the applications expect.

EPOS aims to deliver, whenever possible automatically, a customized runtime support system for each parallel application. In order to achieve this, EPOS introduces three main concepts: 1 - adaptable, scenario independent system abstractions that result from composing PURE building blocks into application-ready abstractions. These abstractions are designed to be as much independent from the execution scenario as possible. 2 - Scenario adapters that adapt existing system abstractions to a given execution scenario, for instance, by making an existing thread abstraction ready to run in a SMP configuration. 3 - Inflated interfaces that export the system abstraction repository by gathering several different implementations of each system abstraction in a single, well-known interface.

An application designed and implemented following the guidelines behind these concepts can be submitted to a tool that will proceed syntactical and data flow analysis to extract an operating system blueprint. This blueprint is then refined by dependency analysis against information about the execution scenario acquired from the user via visual tools; and then submitted to another tool that will generate the application-oriented operating system. With this approach, EPOS shall diminish the gap that usually separates operating systems from parallel applications.