Modularity and Protection are Orthogonal, or "Why μ-kernel Architectures are Flawed"

Peter Druschel
University of Arizona
druschel@cs.arizona.edu

We argue that a modular operating system architecture should provide transparent access for modules *independent* of protection domains. Given such support, modules and interfaces can be designed according to sound software engineering principles, without concern for cross-domain communication costs. The distribution of modules across domains (and machines) becomes a matter of configuration, rather than design.

On current hardware, the fastest cross-domain invocation mechanism exceeds the cost of a local procedure call by an order of magnitude. Consequently, distribution among domains on the same machine is as much an issue as distribution among machines. The *policy* of distributing modules among domains should be separated from the *mechanism* of accessing modules. Current micro-kernel based architectures do not address this issue since they provide transparent access *only* between protection domains. As a result, the modular decomposition of these systems is limited due to concern for cross-domain communication costs. Moreover, the assignment of functionality to domains is static—a serious drawback, particularly in the case of the micro-kernel itself.

We propose an architecture that allows the definition of encapsulated modules with well-defined, procedural interfaces and permits late binding and location-transparent invocation among these modules, whether they are in the same domain, in different domains on the same machine, or on different machines. The key is an *optimistic* location-transparent invocation mechanism that is tailored to the local (intra-domain) case, and relies on specialized communications services such as RPC, LRPC, URPC for the remote cases.