In Proc. 8th Workshop on Hot Topics in Operating Systems (HotOs-VIII)

Active Streams: An approach to adaptive distributed systems

Fabián E. Bustamante, Greg Eisenhauer, Patrick Widener, Karsten Schwan, and Calton Pu*
College of Computing, Georgia Institute of Technology
{fabianb, eisen, pmw, schwan, calton}@cc.gatech.edu

An increasing number of distributed applications aim to provide services to users by interacting with a correspondingly growing set of data-intensive network services. Such applications, as well as the services they utilize, are generally expected to handle dynamically varying demands on resources and to run in large, heterogeneous, and dynamic environments, where the availability of resources cannot be guaranteed *a priori* — all of this while providing acceptable levels of performance.

To support such requirements, we believe that new services need to be customizable, applications need to be dynamically extensible, and both applications and services need to be able to adapt to variations in resource availability and demand. A comprehensive approach to building new distributed applications can facilitate this by considering the contents of the information flowing across the application and its services and by adopting a component-based model to application/service programming. It should provide for dynamic adaptation at multiple levels and points in the underlying platform; and, since the mapping of components to resources in dynamic environment is too complicated, it should relieve programmers of this task. We propose *Active Streams* [1], a middleware approach and its associated framework for building distributed applications and services that exhibit these characteristics.

With Active Streams, distributed systems are modeled as being composed of *applications*, *services*, and *data streams*. Services define collections of operations that servers can perform on behalf of their clients. Data streams are sequences of self-describing application data units flowing between applications' components and services. They are made *active* by attaching application- or service-specific location-independent functional units, called *streamlets*. Streamlets can be obtained from a number of locations; they can be downloaded from clients or retrieved from a streamlet repository. The Active Streams C-based framework employs dynamic code generation in order to insure that streamlets can be dynamically deployed and efficiently executed across heterogeneous environments. Application evolution and/or a relatively coarse form of adaptation is obtained by the attachment/detachment of streamlets that operate on and change data streams' properties. Finer grain adaptation involves tuning an individual streamlet's behavior through parameters remotely updated via a push-type operation, and by re-deploying streamlets to best leverage the available resources over the datapath.

Active Streams are realized by mapping streamlets and streams onto the resources of the underlying distributed platform, seen as a collection of loosely coupled, interconnected computational units. These units make themselves available by running as Active Streams Nodes (ASNs), where each ASN provides a well-defined environment for streamlet execution. Active Streams applications rely on a push-based customizable resource monitoring service (ARMS) to collect resource information and trigger adaptation. Through ARMS, applications can select a subset of the data made available by distributed monitors. These data streams can be integrated to produce application-specific views of system state and decide on possible adaptations.

As is common in distributed systems, a directory service provides the "glue" that holds the Active Streams framework together. The dynamic nature of most relevant objects in Active Streams makes the passive client interfaces of classical directory services inappropriate. Thus, the Active Streams framework includes a *proactive* directory service with a publish/subscribe interface through which clients can register for notification on changes to objects currently of interest to them. The levels of detail and granularity of these notifications can be dynamically tuned by the clients.

The implementation of Active Streams is mostly complete, and we plan on making it available by December 2001.

References

[1] F. E. Bustamante and K. Schwan. Active Streams: An approach to adaptive distributed systems. Tech. report, College of Computing, Georgia Institute of Technology, Atlanta, GA, June 1999.

* Active Streams is part of the Infosphere Project, partially supported by DARPA/ITO under the Information Technology Expeditions, Ubiquitous Computing, Quorum, and PCES programs, by NSF/CISE under the CCR and ANIR programs, and by Intel.