The QuickSilver Properties Framework (abstract for an OSDI 2006 poster)

Krzysztof Ostrowski[†] (student), Ken Birman[†], and Danny Dolev[§]
[†]Cornell University and [§]The Hebrew University of Jerusalem

for the poster corresponding to this abstract, visit http://www.cs.cornell.edu/~krzys/poster-osdi2006/

Publish-subscribe [1] and group communication are powerful paradigms that simplify the construction of distributed systems. As argued in [3], [4] and [5], reliable publish-subscribe could serve as component integration technology that enables new types of collaborative applications that do not rely on centralized infrastructure, and yet that can maintain and consistently update shared state. Examples of such applications range from collaborative document editing to massively multiplayer gaming platforms. The key enabling technology is the ability to create reliable publish-subscribe topics casually, the individual topics representing e.g. documents, media streams, files, rooms in a castle, stocks in a trading system, categories of products in a data center etc.

To realize this vision, we need scalability in many dimensions. These include the total number of nodes in the system or in each individual topic, the number of topics, the rate of multicast, churn, and failures, or the frequency and duration of losses, scheduling and other disturbances. Also needed is a clean way to express "types", both of topics and of their clients (publishers or subscribers), e.g. in the sense of the reliability properties guaranteed, security etc. The latter would allow us to decouple the applications using publish-subscribe or group communication from the specific runtime environment multicasting their messages and enforcing the reliability properties of the individual topics. The existing systems lack scalability in important dimensions, often can't sustain high throughput, or respond poorly to churn or failures. Also, they lack the strong typing aspect and clean language embedding.

We propose[1] a new, principled approach to building scalable and highly extensible group communication systems that addresses the vision laid out above. The QuickSilver Properties Framework [5] is a new platform that allows protocols with strong reliability, and possibly security or other types of properties, to be expressed using a simple rule-based language, and automatically translated into hierarchical protocols. Our work is somewhat related to declarative networking [2], although both our goals and our approach differ significantly from [2]. Our framework is flexible enough to express a wide range of properties, including virtual synchrony, consensus and transactional semantics, in an efficient manner.

The idea is based on an observation that state and progress of most reliable protocols can be described in terms of "properties" of individual nodes and groups of nodes, and rules that determine how values of such properties are derived, and how different properties are related to one another. An example of such property is **Received**(**x**). For an individual node **x**, it represents the set of identifiers of messages that **x** received. If **x** is a set of nodes, the value of the property is aggregated across its members, that is, **Received**(**x**) = sum of **Received**(**y**), for **y** ∈ **x**. Similarly, one can define **Cleaned**(**x**) for a node as a set of identifiers of the received messages that **x** is no longer locally buffering, and for a group of nodes via a set intersection. Actions of most protocol can be modeled as calculating values of certain distributed properties (e.g. an exchange of ACKs falls into this category), or relating the values of different properties via simple rules. For example, **Cached**(**x**) ∧ **Missing**(**y**) → **Forward**(**x**, **y**) represents a rule for peer-to-peer loss recovery, **Stable**(**topic**) → **Clean**(**topic**) controls cleanup in an atomic multicast protocol etc. Some of the properties require special handling. For example, **Stable** is monotonic: if **x** is a distributed entity, each new value for **Stable**(**x**) is aggregated over values of **Stable**(**y**) for **y** ∈ **x** at least as fresh as those used in any of the earlier aggregations.

We've made significant progress towards realizing our vision. We've built a prototype system [3] that offers simple ACK-based reliability and scales in multiple dimensions. We tested it with up to 200 nodes and 8192 groups. Within this range, performance falls by only about 5%, the system incurs little CPU overhead and responds well to crashes, bursts of losses, churn and other types of perturbances. In [4], we proposed a novel architecture, inspired by [3], in which dissemination and recovery for each topic are hierarchically decomposed into layers. In the resulting hierarchy, a different local dissemination or local recovery protocol can be used in different parts of the network, e.g. SRM in one data center and RMTP in another, and yet the system as a whole can provide a global reliability property for the topic, such as last-copy recall. The ideas prototyped in [3] and generalized in [4] are based on an object-oriented perspective on protocols. Properties Framework is based on the same approach, and it can be implemented on top of the architecture described in [4], but it goes much further, in that it allows protocols to be expressed in a purely declarative manner. We expect to have a working prototype of the framework, evaluate its performance, and make it publicly available by early 2007.

[1] B. Oki, M. Pfluegl, A. Siegel, D. Skeen. The Information Bus: An Architecture for Extensible Distributed Systems. In the Proceedings of ACM SOSP'94. Asheville, NC, 1994.

[2] B. Loo, T. Condie, J. Hellerstein, P. Maniatis, T. Roscoe, I. Stoica. Implementing Declarative Overlays. In Proceedings of ACK SOSP'05. Brighton, UK, October 2005.

[3] K. Ostrowski, K. Birman, and A. Phanishayee. QuickSilver Scalable Multicast. Cornell Univ. Tech. Report, April 2006.

[4] K. Ostrowski, K. Birman. Extensible Web Services Architecture for Notification in Large-Scale Systems. To appear in the Proceedings of 2006 IEEE International Conference on Web Services (ICWS'06). Chicago, IL, September 2006.

[5] K. Ostrowski, K. Birman, D. Dolev. Properties Framework and Typed Endpoints for Scalable Group Communication. Cornell University Technical Report (July 2006).

[1] We're supported by grants from AFRL/IFSE, AFOSR, Intel. Contact email: {krzys,ken}@cs.cornell.edu, dolev@cs.huji.ac.il