Building Secure and Reliable Network Applications

Professor Kenneth P. Birman

Dept. of Computer Science
Cornell University, Ithaca, NY

Abstract. Despite nearly 20 years of progress toward ubiquitous computer connectivity, distributed computing systems have only recently emerged to play a serious role in industry and society. Perhaps this explains why so few distributed systems are reliable in the sense of tolerating failures automatically, guaranteeing properties such as performance or response time, or offering security against intentional threats. In many ways the engineering discipline of reliable distributed computing is still in its infancy.

1. The Emerging Crisis in Distributed Systems Reliability

One might be tempted by a form of circular reasoning, concluding that reliability must not be all that important in distributed systems (otherwise, the pressure to make such systems reliable would long since have become overwhelming). Yet, it seems more likely that we have only recently begun to see the types of distributed computing systems in which reliability is critical. To the extent that existing mission- and even life-critical applications rely upon distributed software, the importance of reliability has perhaps been viewed as a narrow, domain-specific issue. On the other hand, as distributed software is placed into more and more critical applications, where safety or financial stability of large organizations depends upon the reliable operation of complex distributed applications, the inevitable result will be a growing demand for technology developers to demonstrate the reliability of their distributed architectures and solutions. It is time to tackle distributed system reliability in a serious manner. To fail to do so today is to invite catastrophic computer-system failures tomorrow.

2. Impact of the World Wide Web

At the time of this writing, the sudden emergence of the World Wide Web is bringing the issue of network reliability to the forefront. It is clear that some form of critical mass has recently been reached: Distributed computing is emerging from its specialized and very limited niche to become a mass-market commodity—something that literally everyone depends on, such as a telephone or an automobile. The Web paradigm brings together the key attributes of this new market in a single package: easily understandable graphical displays, substantial content, unlimited information to draw upon, and virtual worlds in which to wander and work. But the Web is also stimulating growth in other types of distributed applications. In some intangible way, the experience of the Web has caused modern society to suddenly notice the potential of distributed computing.

Consider the implications of a societal transition whereby distributed computing has suddenly become ubiquitous. In the past, a mass-market phenomenon normally involved a product that everyone "owned." With the Web, one suddenly sees a type of behavior that is transforming the manner in which everyone does things. Moreover, for the most part, the computers and networks on which this activity is taking place already exist.
What has changed is the way people see them and use them. The paradigm of the Web is to connect useful things (and many useless things) to the network. Communication and connectivity suddenly seem to be mandatory: No company can possibly risk arriving late for the Information Revolution. Increasingly, it makes sense to believe that if an application can be put on the network, someone is thinking about doing so, and soon.

Whereas reliability and indeed distributed computing were slow to emerge prior to the introduction of the Web, reliable distributed computing will be necessary if networked solutions are to be used safely for many of the applications that are suddenly being envisioned. In the past, researchers in the field wondered why the uptake of distributed computing had been so slow. Overnight, the question has become one of understanding how the types of computing systems that run on the Internet and the Web, or that will be accessed through them, can be made reliable enough for emerging critical uses.

Imagine computer systems that use Web-like interface to present medical status information and records to a doctor in a hospital, or to control a power plant from a remote console, or to guide the decision making of major corporations. The reliability and security of those interfaces and applications will be absolutely critical to the users. Some may have life-or-death implications: If a physician bases a split-second decision on invalid data, the patient might die. Other interfaces may be critical to the efficient function of the organization that uses them: If a bank mismanages risk because of an inaccurate picture of how its investments are allocated, the bank could incur huge losses or even fail. In still other settings, reliability may emerge as a key determinant in the marketplace: The more-reliable product, at a comparable price, may simply displace the less-reliable one. Reliable distributed computing suddenly has broad relevance.

3. Two Styles of Distributed Computing System

For those of us who work in this area, the term "distributed computing" is used to describe a type of computer system that differs from what could be called a "network computing" system. The distinction illuminates the basic issues with which I am primarily concerned.

A computer network is a communication technology supporting the exchange of messages among computer programs executing on computational nodes. Computer networks are data movers, providing capabilities for sending data from one location to another, dealing with mobility and changing topology, and automating the division of available bandwidth among contending users. Computer networks have evolved over a 20-year period, and during the mid-1990s network connectivity between computer systems became pervasive. Network bandwidth has also increased enormously, rising from hundreds of bytes per second in the early 1980s to millions of bytes per second in the mid-1990s, with gigabyte rates anticipated in the late 1990s and beyond.

It has been suggested that technology revolutions are often spurred by discontinuous, as opposed to evolutionary, improvement in a key aspect of a technology. The bandwidth improvements we are now experiencing are so disproportionate with respect to other performance changes (memory sizes, processor speeds) as to fall squarely into the discontinuous end of the spectrum. The sudden connectivity available to PC users is similarly disproportionate to anything in prior experience. The Web is perhaps just the first of a new generation of communication-oriented technologies enabled by these sudden developments.