Scalable Fault Tolerance

Shay Kutten
Faculty of Industrial Engineering, Technion, Haifa 32000, Israel, and IBM T.J. Watson Research Center, kutten@ie.technion.ac.il.

Abstract. As communication networks grow, existing fault handling tools become increasingly unaffordable. In many cases the reason is that they involve global measures such as global time-outs or reset procedures, and their cost grows with the size of the network. Rather, for a fault handling mechanism to scale to large networks, it should involve local measures, or, at worse, fault local measures, i.e. measures the cost of which depends only on the number of failed nodes (which, thanks to today's technology, grows much slower than the networks). This decreases the recovery time and, moreover, often allows the non-faulty regions of the networks to continue their operation even during the recovery of the faulty parts. We describe several research ideas that lead in this direction.

1 Introduction

Computer networks and distributed systems are growing very fast. The Internet [C91] is estimated to double in size every 6 to 12 months. Currently it connects (according to some estimates) about 16,000,000 hosts. Most of them belong to end users, whose involvement in control functions is limited (although they do participate in many distributed computing activities). However, at the rate of 100 to 1000 nodes per router, the number of control nodes is estimated in the tens of thousands. Moreover, the Internet is not the only huge and fast growing network; working hard to realize the celebrated "information super-highway" are all the main telecommunication companies in the world. Phone companies, for example, are planning to invest heavily in the next several years in updating their networks to carry hundreds of video channels and computer information to homes, as well as in wireless networks, to provide computer connection and information (e.g. stock prices, weather reports, etc) to every person on the go.

Another characteristic of the emerging networks is their diversity. Numerous manufacturers are involved in producing the hosts, routers and cables for these networks. Even when equipment of the same source is used, there are significant differences in the ages, sizes, speeds, reliability, and many other attributes of the equipment and software used. Moreover, the Internet is not managed by
one single authority, but rather by thousands of organizations, governed by very different policies and considerations. For example, an organization that does not impose strong security measures may face an increase in the number of faults, since its machines are exposed to intruders. Indeed, the parties concerned strive for compatibility, but many differences exist nevertheless.

In such a diverse environment, faults of many different types, leading to information inconsistency, are unavoidable. Indeed, coping with faults, and devising fault-tolerant solutions for various problems, is one of the most active research areas in networking and distributed systems.

However, the speed of growth of the networks out-paces the speed of the developments of the fault tolerance mechanisms, and they do not scale well. It is interesting to note that faults are often called *exceptions*, and their treatment is called *exception handling*. This seems to reflect the approach that faults rarely happen, so a treatment of a fault is really an exceptional case. Thus the treatment did not have to be very efficient, since its amortized cost was very small.

For example, one striking characteristic of this area is that in many of the solutions proposed in the literature, faults are fixed *globally*, i.e., by algorithms involving the entire system. Clearly, using global measures to detect and correct errors is becoming more and more unrealistic as networks grow, and it is essential to develop *scalable* fault-handling tools, that is, solutions that can be applied even in large networks. In particular, the cost of such tools is required to grow slower than the system size. It is also required that the non-faulty parts of the networks will be able to continue operating even while the faulty parts are recovering. Otherwise no meaningful work can be done.

Luckily, technology manages to keep the increase in the number of faults smaller than the increase in the size of the network. Indeed, the damage each fault may cause is amplified by the growing size of the network, but the faults are still very often of extremely *local* nature, and involve only a small number of hosts. For example, the famous ARPANET crash [MRR80] was caused by a single node giving all other nodes wrong routing information. (This node "told" other nodes that it had distance zero from every node.)

In this paper we describe several efforts that were made toward the goal of making fault handling tools more scalable, mostly by emphasizing locality in fault handling.

2 Fault Models

Different distributed systems may be designed to withstand faults of different level of severity. The number of models is huge, and we shall only mention a few