Introduction

Since the advent of the computer in 1940, computing power needs have not ceased to increase. Today, great scientific fields such as high-energy physics, astrophysics, climatology, biology and medical imagery rely on new mutualization technologies and worldwide sharing of computer potential across international grids to meet the huge demand for data processing. Every day, researchers submit hundreds of computations to large-scale distributed infrastructures such as the European Enabling Grids for E-sciencE grid (EGEE) [EGE 04], which gathers more than 100,000 processors. Soon European Grid Infrastructure (EGI) and TeraGrid [PRO 11] in the United States will each be able to aggregate more than double this number of processors. In the near future many industrial domains, such as automobile, energy and transport, which are increasingly relying on digital simulation, will be able to benefit from large shared reservoirs of computer resources. This approach will shortly be extended to e-commerce, finance and the leisure industry.

Over the past 15 years, three key technologies have followed each other in response to this growing computing power demand. These technologies embody the revolution of network computing: computer clusters, computing grids and computing clouds. A quick definition of these is as follows:

– a computing cluster is a collection of PCs interconnected via local-area, very-low-latency, high-speed networks;

– a computing grid is the aggregation of a very large number of distributed computing and storage resources, interconnected via wide-area networks. There are computing grids dedicated to intensive computations of data grids that store, process and give access to massive amounts of data in the order of hundreds of gigabytes or even several terabytes;

– a computing cloud provides access services to resources via the Internet. The underlying infrastructure is totally concealed from users. The available resources are generally virtual machines housed in resource centers, also called data centers.

Originally, it was the spectacular advances in transmission and communication technologies that enabled researchers to imagine these distributed architectures. These technologies made the aggregation and mutualization of computer equipment possible, which led to the rise in power of global computing. The hardware and software of interconnection networks, which are transparent in appearance, play a complex role that is difficult to grasp and not often studied. Yet the place of the network is central and its evolution will certainly be a key to ubiquitous computer systems to come.

Indeed, to make full use of a mutualized communication network, sharing policies implemented by robust and scalable arbitration and orchestration mechanisms are necessary. Today these mechanisms are included in distributed software called communication protocols. These protocols mask the complexity of the hardware and the organization of exchanges. Services of information transfer over a network rely on communication protocols and software that are built according to a layered model and the end-to-end principle. These architectural principles offer an interesting and robust compromise between the need for reliability and that for performance. They are well adapted for low-to-average speeds and unreliable network infrastructures, both when transport needs are relatively homogeneous and when security constraints are rather low. In the context of high-speed networks and computing grid environments, the orders of magnitude and ratios of the constants in use are quite far from the hypotheses initially made for communication software protocols and architecture design. For example, the size of an Ethernet frame (between 64 and 1,500 bytes) – a parameter that indirectly conditions the maximum size of transfer units sent over an IP network – was defined to satisfy propagation constraints on a 200 m coaxial cable and a throughput of 10 Mbit/s. Today optical links are used and throughputs can be greater than 10 Gbit/s. At the time when the Internet Protocol (IP) was being designed, access rates were in the order of 64 kbit/s in wide-area networks. Today, optical fibers are deployed with access rates from 100 Mbit/s to 1 Gbit/s. There are links of over 100 Gbit/s in network cores.

In the Internet, since the workload is not controlled by the network itself, it is traditionally the transport layer – the first end-to-end layer – that carries out the adaptation to fluctuations in performance linked to load changes. The complexity of the transport layer depends on the quality of service offered by the underlying network in terms of strict delay or loss-ratio service guarantees. In the IP model, which offers a best-effort network service, two main transport protocols are classically used:

– a rudimentary protocol, the *User Datagram Protocol* or UDP, which only carries out stream multiplexing; and

– a very sophisticated reliable protocol, *Transmission Control Protocol* or TCP, which carries out the adaptation to packet losses as well as congestion control by send-rate control. TCP was designed for a network layer with no guaranteed quality of service (IP), for local-area networks and low-speed wide-area networks, and for a limited number of application classes.

The transport protocols are not really well adapted to very-high-speed infrastructures. Let us take the example of a simple TCP connection over a link between Lyon (France) and Montreal (Canada), with a round trip delay in the order of 100 ms and a 10 Gbit end-to-end throughput. Due to the design of the TCP congestion-avoidance algorithm, if one single packet is lost, it will take one hour and 40 minutes to repair and regain maximum speed. The TCP protocol is designed to react dynamically (i.e. in an interval of a few milliseconds) to congestion phenomena. It is not very reactive, however, in such conditions!

Over the past 10 years, a certain number of alternatives to TCP have been put forward and introduced in modern exploitation systems.

The protocol aspect is not the sole parameter to take into consideration for evaluating and improving end-to-end performance. Actually, in the very core of the communication nodes used, delays due to different data movement and control operations within a machine are significant compared to the delays encountered on the network itself (cables and routers). The heterogeneity of performance needs must also be taken into consideration.

The protocols used in the context of distributed computing have gradually became increasingly diverse because of the heterogeneity of the underlying physical technologies and applications needs. When the end-user of a cluster or a grid struggles to obtain the performance, however, he or she could expect delays with regard to the theoretical performance of the hardware used. He or she often has difficulties understanding where the problems with performance come from.

For this reason, this book invites the reader to concentrate more specifically on the core of distributed multi-machine architectures: the interconnection network and its communication protocols. The objective is to present, synthesize and articulate the different network technologies used by current and future distributed computing infrastructures. As these technologies are very heterogeneous in their physical characteristics and software, our aim is to propose the correct level of abstraction to help the reader structure and understand the main problems. It distinguishes the guidelines that, on the one hand, have oriented the technological evolution at the hardware and software levels, and on the other hand can guide programmers and users of distributed computing applications to adopt a programming model and an infrastructure adapted to their specific needs.

This book therefore has two objectives:

– to enable the reader who is familiar with communication networks to better understand the stakes and challenges that the new distributed computing revolution poses to networks and to their communication software;

– to enable the reader who is familiar with distributed computing to better understand the limits of current hardware and software tools, and how he or she can best adapt his or her application to the computing and communication infrastructure that is at his or her disposal to obtain the best possible performance.

To achieve these two objectives, we alternately move from one point of view to the other, introducing the core principles of distributed computing and networks and progressively detailing the most innovative approaches in these two fields.

In Chapter 1, we identify the needs, motivations and forces pushing the computer sector, over the years, towards distributed computing and the massive use of computing networks. We go into the details of the different network computing technologies that have evolved and show the technological and conceptual differences between them.

In Chapter 2 we classify distributed computing applications and analyze the communication specificities and constraints of each one of these classes of applications. In particular, we introduce the Message-Passing Interface communication library, or MPI, which is frequently used by distributed parallel application programmers.

In Chapter 3 we review the core principles of traditional communication networks and their protocols. We make an inventory of their limits compared to distributed computing constraints, which are introduced in the previous chapter. We then analyze the path of communications in a TCP/IP context.

The next two chapters are devoted to a detailed analysis of two major challenges that distributed computing poses to the network: latency and throughput. Two types of characteristic applications serve to illustrate their aim:

- – delay-sensitive parallel computing applications; and
- – communication-intensive, throughput-sensitive applications

In these chapters, we also discuss the direct interaction between the hardware level and the software level – a characteristic element of distributed computing.

Chapter 4 studies how the challenge of latency was overcome in computer cluster infrastructures to address the needs of applications that are very sensitive to information-routing delay between computing units.

Chapter 5 focuses on the needs of applications transferring significant masses of data in order to take them from their acquisition point to the computing centers where they are processed as well as to move them between storage spaces to conserve them and make them available to large and very scattered communities. We therefore study how the TCP protocol reacts in high bandwidth-delay product environments and detail the different approaches put forward to enable high-speed transport of information over very long distances.

Chapter 6 deals with performance measurement and prediction. It enables the reader, coming from the field of distributed computing, to understand the contributions of network performance measurement, prediction infrastructures and tools.

Chapter 7 shows how new optical switching technologies make it possible to provide a protected access to a communication capability adapted to the needs of each application.

Chapter 8 presents new dynamic bandwidth-management services, such as those currently proposed in the Open Grid Forum that suggest solutions for applications with sporadic needs relating to speeds that are not very high.

Chapter 9 introduces the issue of security and its principles in computing networks. This chapter presents the main solutions currently deployed as well as a few keys capable of increasing user confidence in distributed computing infrastructures.

Chapter 10 proposes a few protocol- and system-parameterization examples and exercises for obtaining high performance in a very-high-speed network with tools currently available in the Linux system.

To conclude, we summarize the different network technologies and protocols used in network computing, and provide a few perspectives for future networks that will integrate, among other things, our future worldwide computing power reserve.