In this section we discuss another class of application that could benefit from cloud computing. In

Chapter 4 we discussed cloud applications in computational science and engineering. The benchmarks

presented in Section 4.9 compared the performance of several codes running on a cloud with runs

on supercomputers; as expected, the results showed that a cloud is not an optimal environment for

applications exhibiting fine- or medium-grained parallelism. Indeed, the communication latency is

considerably larger on a cloud than on a supercomputer with a more expensive, custom interconnect.

This means that we have to identify cloud applications that do not involve extensive communication or

applications exhibiting coarse-grained parallelism.

A cloud is an ideal running environment for scientific applications that involve model development.

In this case, multiple cloud instances could concurrently run slightly different models of the system.

When the model is described by a set of parameters, the application can be based on the SPMD paradigm

combined with an analysis phase when the results from the multiple instances are ranked based on a

well-defined metric. In this case there is no communication during the first phase of the application,

when partial results are produced and then written to the storage server. Then individual instances

signal the completion and a new instance to carry out the analysis and display the results is started. A

similar strategy can be used by engineering applications of mechanical, civil, electrical, electronic, or

any other system design area. In this case, the multiple instances run concurrent design for different

sets of parameters of the system.

A cloud application for optimal design of Field-Programmable Gate Arrays (FPGAs) is discussed

next. As the name suggests, an FPGA is an integrated circuit designed to be configured, adapted, or

programmed in the field to perform a well-defined function [311]. Such a circuit consists of logic blocks

and interconnects that can be “programmed” to carry out logical and/or combinatorial functions (see

Figure 11.17).

The first commercially viable FPGA, the XC2064, was produced in 1985 by Xilinx. Today FPGAs are

used in many areas, including digital signal processing, CRNs, aerospace, medical imaging, computer

vision, speech recognition, cryptography, and computer hardware emulation. FPGAs are less energy

efficient and slower than application-specific integrated circuits (ASICs). The widespread use of FPGAs

is due to their flexibility and the ability to reprogram them.

Hardware description languages (HDLs) such as VHDL and Verilog are used to program FPGAs.

HDLs are used to specify a register-transfer level (RTL) description of the circuit. Multiple stages are

used to synthesize FPGAs.

A cloud-based system was designed to optimize the routing and placement of components. The basic

structure of the tool is shown in Figure 11.18. The system uses the PlanAhead tool from Xilinx (see

www.xilinx.com) to place system components and route chips on the FPGA logical fabric. The

computations involved are fairly complex and take a considerable amount of time; for example, a fairly

simple system consisting of a software core processor (Microblaze), a block random access memory

(BRAM), and a couple of peripherals can take up to 40 minutes to synthesize on a powerful workstation.

Running N design options in parallel on a cloud speeds up the optimization process by a factor close toN.