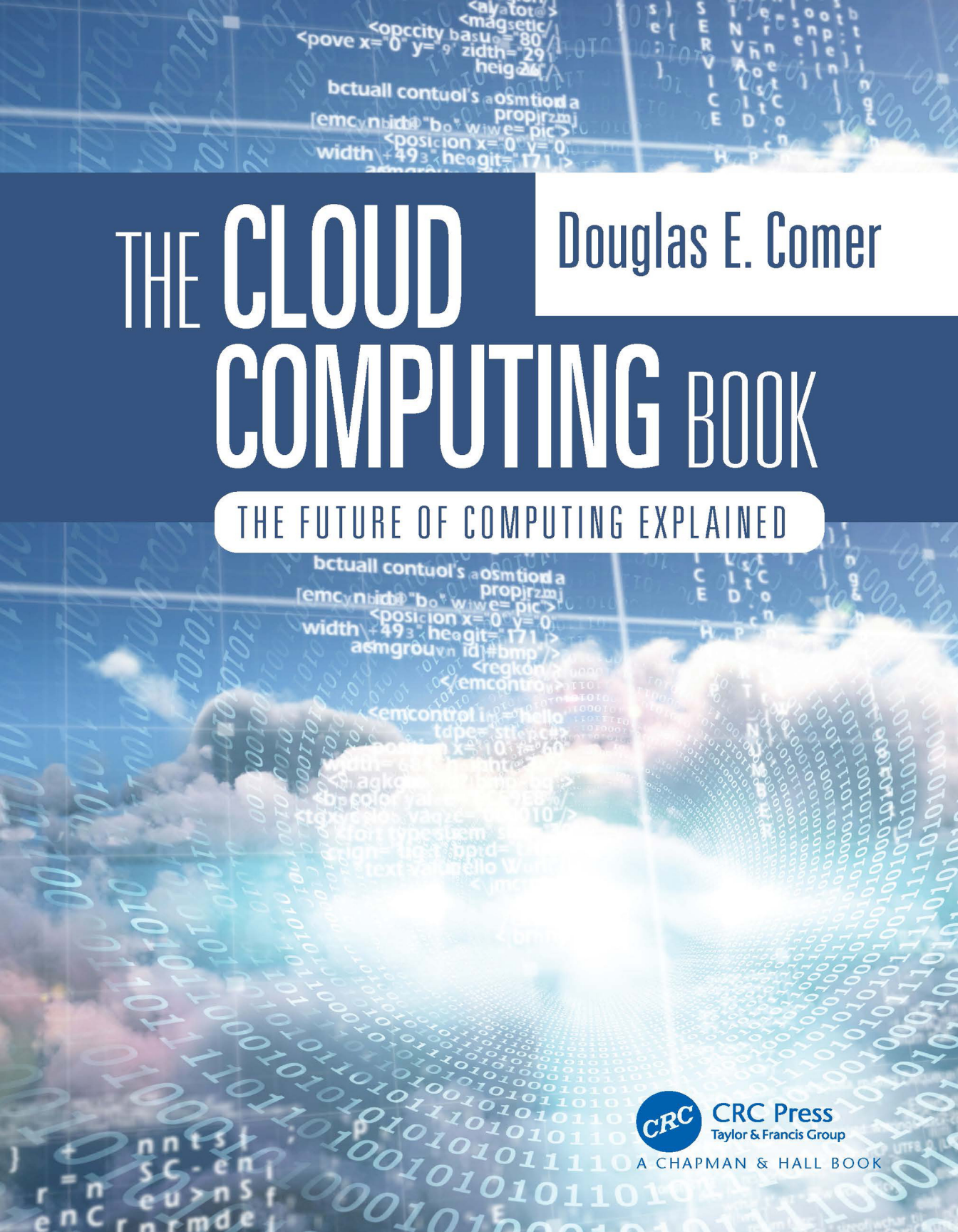

THE CLOUD COMPUTING BOOK

THE FUTURE OF COMPUTING EXPLAINED

Douglas E. Comer

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

The Cloud Computing Book

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

The Cloud Computing Book

The Future Of Computing Explained

DOUGLAS E. COMER

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

First edition published 2021
by CRC Press
6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742

and by CRC Press
2 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

Library of Congress Cataloging-in-Publication Data

Names: Comer, Douglas, author.
Title: The cloud computing book : the future of computing explained / Douglas Comer.
Description: First edition. | Boca Raton : CRC Press, 2021. | Includes bibliographical references and index.
Identifiers: LCCN 2020052980 | ISBN 9780367706807 (hbk) | ISBN 9781003147503 (ebk)
Subjects: LCSH: Cloud computing.
Classification: LCC QA76.585 .C636 2021 | DDC 004.67/82--dc23
LC record available at https://lccn.loc.gov/202005298

ISBN: 978-0-367-70680-7 (hbk)
ISBN: 978-1-003-14750-3 (ebk)

Typeset in Times font
by KnowledgeWorks Global Ltd.

To Chris, who takes
me to cloud nine

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Contents

Preface

In late 2020, a headline in a tech industry newsletter announced:

Cloud Is Changing Everything

Although it may seem like hype, the headline captures a fundamental change in computing. Organizations, both large and small, are migrating computing and data storage to the cloud, which has grown an entirely new ecosystem. Services ranging from Internet search, AI, and big data analysis to individual document processing use cloud facilities.

Curricula in Computer Science and Computer Engineering must shift to prepare students for cloud computing. Instead of simply learning how to design and use software and hardware systems for the traditional mobile and desktop world, courses must prepare students to design and build hardware and software systems for use in a cloud data center. Instead of merely building a conventional computer program and relying on faster hardware to increase performance, cloud employs a parallel approach that requires one to deploy and manage multiple copies of each software system.

What should one learn that will help them navigate constant change? A colleague quipped that courses on cloud will quickly sink into teaching technology because commercial technologies dominate the cloud industry. Indeed, dozens of technologies exist, and new technologies seem to appear every few months, forcing constant churn. The answer is that colleges and universities should teach basic concepts and principles that will remain valid despite the ever-changing commercial world. For example, to master new orchestration technologies, one needs to understand the big picture of why orchestration is needed and how orchestration systems work. Similarly, to make sense of Virtual Machine and Container technologies, one needs to understand the fundamentals of each. To understand why a cloud provider may scatter a tenant's virtual servers across a set of physical servers, one must understand the underlying infrastructure and surprising topics such as power and cooling.

This text provides a broad overview of cloud computing, covering all aspects from basic data center facilities to the ways cloud-native software differs from traditional software. Instead of describing the services offered by a particular public cloud company or attempting to cover the wide range of third-party offerings, the text concentrates on general concepts that span many providers and services. To help keep the discussion focused on reality, the text uses concrete examples of technologies that currently dominate the industry, including Docker Containers and Kubernetes Orchestration technology. However, rather than attempting to present all details, the text gives examples that show the essence of each.

Designed as both a textbook and professional reference, the text is suitable for a one-semester course in computer science and engineering and for professional software engineers and IT staff who need to understand cloud. The material has been divided into five parts. The first part describes the motivation, advantages, and growth of cloud computing. The second part describes cloud infrastructure and virtualization, including virtual computing, networking, and storage mechanisms. The third part describes high-level automation and orchestration systems that manage the virtualized infrastructure. The fourth part describes cloud software, including the programming paradigms used and how cloud software deployments scale to large numbers of users. The final part describes remaining topics, including the concept of edge computing and its relationship to the Industrial Internet of Things, security problems that arise in a cloud environment, and approaches that help designers control the complexity of cloud deployments.

One cannot appreciate cloud computing without hands-on experience. Fortunately, major cloud providers offer "free tier" services that allow individuals to try deploying cloud services at no cost; some providers also offer special educational accounts and curricula materials†. Although free accounts have limited computational and storage resources, they allow one to learn about the technologies and interfaces available from various cloud providers. I strongly encourage everyone who wants to understand cloud to obtain one or more free accounts and deploy a VM (e.g., a web server), use Docker to deploy a container, and (for more advanced readers) use Kubernetes to orchestrate replicated containers. Examples of free tier accounts include:

Amazon Web Services free tier	**https://aws.amazon.com/free/**
Google Cloud Platform free tier	**https://cloud.google.com/free/**
Azure Cloud free tier	**https://azure.microsoft.com/free/**
IBM Cloud free tier	**https://www.ibm.com/cloud/free**
Oracle Cloud	**https://www.oracle.com/cloud/free/**

I thank many individuals and groups who contributed to the book. Daniel J. Krolopp, Adib Rastegarnia, Paul Schmitt, Phil Van Every, and John Lin proofread chapters and provided suggestions about the content. Adib developed the TLA+ example in Chapter 18. Ted Turner provided insights on trends in the industry. Nick Lippis from the Open Network User's Group graciously invited me to participate in meetings, where I learned how Fortune 100 firms are moving to a hybrid multi-cloud infrastructure. Aryo Kresnadi helped me understand the scale of Fed Ex facilities and their use of cloud. Ernest Leffler introduced me to how the banking and financial services industries are making use of cloud. Scott Comer, Sharon Comer, and Mark Kunschke provided support and suggested cover designs.

Finally, I thank my wife, Christine, for her patient and careful editing and valuable suggestions that improved and polished the text.

Douglas E. Comer

About The Author

Douglas E. Comer is a Distinguished Professor at Purdue University in the departments of Computer Science and Electrical and Computer Engineering (courtesy). He has an extensive background in computer systems, and has worked with both hardware and software.

One of the researchers who contributed to the Internet as it was being formed in the late 1970s and 1980s, he has served as a member of the Internet Architecture Board, the group responsible for guiding the Internet's development. Prof. Comer is an internationally recognized expert on computer networking, the TCP/IP protocols, and the Internet.

In addition to research articles, he has written a series of widely acclaimed textbooks on the topics of computer architecture, operating system design, computer networking, and the Internet. Prof. Comer's books have been translated into many languages, and are used in industry as well as Computer Science, Engineering, and Business departments around the world. He continues to consult and lecture at universities, industries, and conferences.

While on leave from Purdue, Prof. Comer served as the inaugural VP of Research at Cisco Systems. For twenty years, Prof. Comer served as the editor-in-chief of the journal *Software —Practice and Experience* . He is a Fellow of the Association for Computing Machinery (ACM), a Fellow of the Purdue Teaching Academy, and a recipient of numerous awards, including a USENIX Lifetime Achievement Award. Prof. Comer is a member of the Internet Hall of Fame.

Additional information can be found at:

www.cs.purdue.edu/people/comer

and information about Comer's books can be found at:

www.comerbooks.com

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Part I

Why Cloud?

The Motivation, Advantages, And Growth Of Cloud Computing

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Chapter Contents

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

1

The Motivations For Cloud

1.1 Cloud Computing Everywhere

In a short time, cloud computing has become significant. Consider the following examples of situations that involve cloud computing.

- A startup leases cloud facilities for its web site; the company can pay for additional facilities as web traffic grows.
- An individual uses a smart phone to check *Internet of Things* (*IoT*) devices in their residence.
- An enterprise company leases facilities and software for business functions, such as payroll, accounting, and billing.
- Students working on a team project use a browser to edit a shared document.
- A patient wears a medical device that periodically uploads readings for analysis; their doctor is alerted if a medical problem is detected.
- A seasonal company leases computing facilities during four peak months each year; the company doesn't pay for facilities at other times.
- A teenager logs into a social media site and uploads photos.
- A retail company leases facilities at the end of each fiscal year to run data analytics software that analyzes sales for the year.
- An individual uses a streaming service to watch a movie; a copy of the movie is kept in a facility near the family's residence.
- The recipient of a package uses a tracking number to learn about the current location of the package and the expected delivery time.

Most enterprises — not just high tech firms and social media companies — are moving to the cloud. In the early 2000s, business functions, such as payroll, accounting, billing, and supply chain management were implemented with local facilities operated by an organization's IT staff. Now, such functions are being migrated to cloud computing. Enterprises that do decide to retain some local computing are shifting from a paradigm of having services run on individual computers located in various departments throughout the organization to a paradigm where the facilities are consolidated into a local cloud data center.

In addition to providing communication to consumer IoT devices, the cloud provides processing. For example, battery-operated IoT sensors are used to monitor civil infrastructure, such as bridges. A set of sensors on a bridge periodically measures vibrations or stress and uploads the data to the cloud. Software running in the cloud data center combines measurements from dozens of sensors and assesses the safety of the bridge. The amount of processing needed for such computations far exceeds the capability of battery-operated sensors.

1.2 A Facility For Flexible Computing

Many of the examples illustrate a key aspect of cloud computing: flexibility to accommodate both incremental growth and cyclic demand. A cloud offers flexible computing facilities (servers and software), storage facilities, and communication facilities (Internet connections).

Incremental growth. The startup scenario shows why incremental growth is important. A small startup can begin by leasing minimal cloud facilities (e.g., enough to support a basic web site), and then increase its lease as the business grows. Similarly, the startup can begin by leasing minimal software (e.g., basic web and payment processing software), and then add database and accounting software later. The cloud provider will be able to satisfy computing needs even if the startup grows into a substantial enterprise business.

Cyclic demand. Even if a company does not engage in seasonal business, demand for computing changes throughout the year. Reports may be generated at the end of each month or each quarter as well as at the end of the year. Sales activity and order processing often spike at the end of the month as sales staff work to meet their monthly quotas. Cloud allows companies to lease additional facilities to accommodate such demand.

Although the ability to lease resources as needed is attractive, the most significant aspect arises from the pricing model: a cloud provider only charges the customer for the facilities they actually use.

> *Cloud computing allows each customer to increase or decrease their use of cloud facilities at any time; a customer only pays for the facilities they use.*

1.3 The Start Of Cloud: The Power Wall And Multiple Cores

What has motivated the shift to cloud computing? This chapter considers how cloud arose. The next chapters provide an overview of cloud, and describe the rise of cloud providers. Later chapters explore cloud infrastructure and technologies.

Two intertwined factors contributed to the start of the cloud paradigm.

- Technological: limits on speed forced a move to parallelism
- Economic: changes in technology changed IT costs

We begin by looking at the technological factor, and then examine how the new technology changed the cost of acquiring and running equipment and software.

Throughout the 1980s and early 1990s, chip manufacturers produced a series of processors that had more functionality and higher speed than the previous model. As a consequence, individual computers became more powerful each year while costs remained approximately the same. The availability of powerful, low-cost computational facilities encouraged both individuals and organizations to expand their use of computers. At any time, if the processing power of a computer became insufficient for the workload, a computer could easily be upgraded to a newer, more powerful model. In particular, to offer internal and external services, such as the World Wide Web, an organization could run the software on a powerful computer known as a *server*. When demand for a particular service became high, the organization could replace the server being used with a model that had a faster processor and more memory.

By the late 1990s, the chip industry faced a serious limitation. Moore's Law — the prediction that the number of transistors would double every eighteen months — meant that the size of a given transistor was shrinking. More transistors were squeezed together on a chip each year. Each transistor consumes a small amount of power and emits a small amount of heat. When billions of transistors are squeezed together in a small space, the temperature can climb high. More important, the amount of power consumed — and therefore, the temperature of the chip — is proportional to the square of the clock speed. Consequently, even a small increase in clock speed increases the temperature significantly. Manufacturers eventually reached a critical point, and processor speeds could not be increased beyond a few Gigahertz without generating so much heat that the chip would burn out. Industry uses the term *power wall* to characterize the limit on processor speed.

How can additional computational power be achieved without increasing the speed of a processor? The answer lies in parallelism — using multiple processors that each operate at a speed below the power wall instead of one processor that operates at a super high speed. To handle the situation, chip manufacturers devised chips that contain multiple copies of a processor, known as *cores*. Each core consists of a complete processor that operates at a safe speed, and software must devise a way to use multiple cores to perform computation.

Industry uses the term *multicore processor* to describe a chip with multiple processors. A *dual core* chip contains two processors, a *quad core* chip contains four, and so

on. Multicore processors form one of the fundamental building blocks for cloud computing. Unlike the processors used in consumer products, however, the multicore processors used in cloud systems have many cores (e.g., 64 or 128). A later chapter examines virtualization software and explains how cloud systems use computers with multiple cores.

1.4 From Multiple Cores To Multiple Machines

Although they offer increased processing power on a chip, multiple cores do not solve the problem of arbitrary scale. The cores on a chip all share underlying memory and I/O access. Unfortunately, as the number of cores increases, I/O and memory accesses become a bottleneck.

How can more powerful computers be constructed? The science research community was among the first groups to explore a design that provided the basis for cloud. As scientific instruments, such as colliders and space telescopes, moved to digital technologies, the amount of data grew beyond the capabilities of even the most powerful supercomputers. Furthermore, a supercomputer was an extremely expensive machine; few universities and laboratories could afford to purchase multiple supercomputers. However, personal computers had become commodity items as reflected by their low price. Despite the drop in price, personal computers had also become more powerful. Scientists wondered if instead of using expensive supercomputers, a new form of supercomputing could be achieved by interconnecting a large set of inexpensive personal computers. The resulting configuration, which became known as a *cluster architecture*, has a key advantage: processing power can be increased incrementally by adding additional inexpensive commodity computers.

Using multiple computers for scientific computations poses a software challenge: a calculation must be divided into pieces so that each piece can be handled by one of the smaller computers in the cluster. Although the approach does not work well in some cases, the science community found ways to use a cluster for many of its important computational problems. Thus, the cluster architecture became accepted as the best way to build affordable, incrementally expandable supercomputers.

1.5 From Clusters To Web Sites And Load Balancing

As the World Wide Web grew in popularity in the 1990s, the traffic to each web site increased. As in the science community, the limitation on the speed of a given processor presented a challenge to the staff responsible for running web sites, and they also considered how to use multiple personal computers to solve the problem.

Web sites and scientific computing systems differ in a fundamental way. Supercomputer clusters intended for scientific calculations are designed so that small computers can work together on one computation at a time. In contrast, a web site must be designed to process many independent requests simultaneously. For example, consider

a retail web site. One user may be looking at tools while another shops for clothing, and so on, with little overlap among the web pages they visit.

The question arose: how can a web site scale to accommodate thousands of users? Part of the answer came from a technology that has become a fundamental component in cloud computing: a *load balancer*. Typically implemented as a special-purpose hardware device, a load balancer divides incoming traffic among a set of computers that run servers. Figure 1.1 illustrates the idea. As the figure shows, a server may need to access a company database (e.g., to validate a customer's account during checkout).

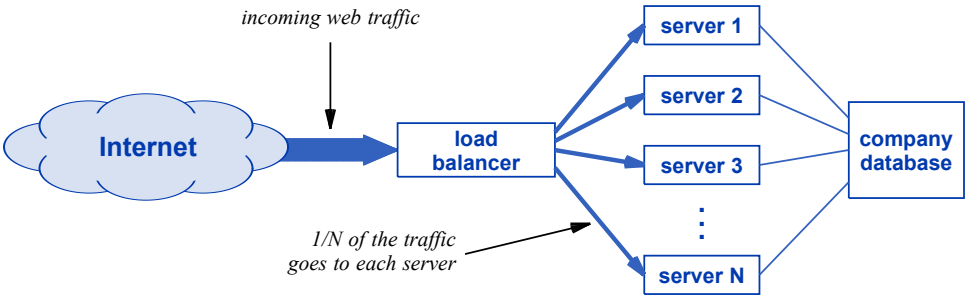

Figure 1.1 Illustration of a load balancer used to divide incoming network traffic among a set of computers.

Load balancing technology ensures that all communication from a given customer goes to the same server. The scheme has a key advantage: successive requests from a customer go back to the server that handled earlier requests, making it possible for the server to retain information and use it for a later request. From the web site owner's point of view, if the site has *N* servers, load balancing means that each server handles approximately *1/N* of the customers. At any time, a site could be expanded merely by adding additional servers; the load balancer automatically includes the new servers when dividing traffic.

1.6 Racks Of Server Computers

As demand increased for facilities composed of many smaller computers, computer vendors responded by changing the shape of computers to make it easier to store many computers in a small space. Instead of large enclosures that had significant amounts of empty space inside, designers focused on finding ways to reduce the size. Furthermore, they redesigned the enclosures to fit into tall metal equipment cabinets called *racks*. Before computers, racks had been used for telephone equipment, and were widely available. A full-size rack is approximately six and one-half feet tall, two feet wide, and three and one-half feet deep. The rack contains metal mounting bars called *rails* to which equipment can be attached.

Server computers can be stacked vertically in a rack. A full rack contains forty-two *Units* of space, where a Unit is 1.752 inches. A server is designed to be one unit tall, written *1U*. In principle, one could stack forty-two 1U servers in each rack, as Figure 1.2 illustrates.

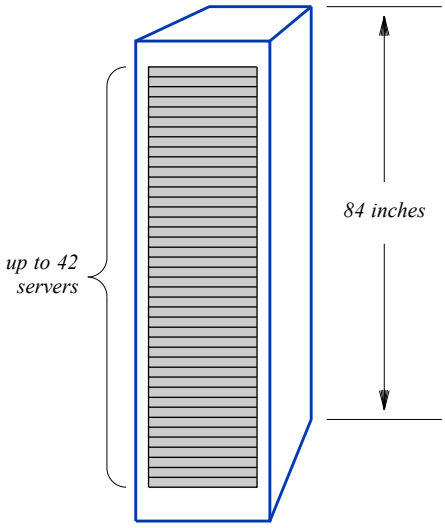

Figure 1.2 Illustration of a rack that holds multiple servers.

In practice, additional constraints usually mean that a rack is not full of servers. For example, at least one slot in a rack (usually the top of the rack) contains a network switch used to provide network connections for the servers in the rack. In addition, some slots may be left empty to allow air flow to keep the servers from overheating.

1.7 The Economic Motivation For A Centralized Data Center

The availability of low-cost servers and the ability to collect multiple servers into a rack may seem insignificant. From the point of view of IT management, however, collecting servers into a small place has an important advantage: lower cost. There are two aspects:

- Operating expenses (opex): lower recurring cost
- Capital expenses (capex): lower equipment cost

Lower operating expenses. To understand how placing servers in racks can reduce operating expenses, recall that the low cost of computer hardware made it easy for each group in a large organization to acquire its own computing facilities. While hardware costs were falling, the recurring cost of IT staff increased because demand for trained IT

professionals caused salaries to rise. Each group hired staff, and because technology changes rapidly, each group had to pay for training to keep their staff up-to-date.

Ironically, the availability of low-cost personal computers, which everyone assumed would lower overall expenses, created a situation in which an organization faced much higher costs because each department purchased many computers and then hired their own staff to manage them. By the 2000s, many organizations became concerned about rising IT costs. An executive at one company quipped,

> Cheap computers have turned into a major expense.

Organizations faced an IT staffing dilemma. On the one hand, allowing every department to hire and train its own IT staff to maintain skills results in unnecessary duplication. Some of the skills are only needed infrequently when problems arise. On the other hand, although it helps share expertise and reduce overall training costs, consolidating IT staff into a central department has the disadvantage that when a problem occurs, a staff member must travel to the department that owns the server to assess the problem and work on the equipment. For an organization with a large campus, such travel may require hiring additional staff.

The availability of high-speed computer networks allows an organization to optimize costs by consolidating server equipment into a single physical location. Instead of locating a server in each department, the organization places all servers in racks in a central facility, and hires a small, centralized staff to maintain the servers. Employees in departments can access the servers over a network. Such a centralized facility has become known as a *data center*†.

Lower capital expenses. The data center approach has an advantage of reducing overall equipment cost. If server computers are distributed throughout an organization, each individual department must choose when to upgrade their server. In addition, each department will consider the cost of a new server as well as the features and performance they need. As a result, a department will usually order one server at a time, and the configuration ordered by one department is likely to differ from the configuration ordered by another department.

If an organization consolidates servers into a data center, the organization can choose a uniform configuration for all servers. Furthermore, the organization can upgrade many servers at once (e.g., upgrade one-third of all servers every year). Consequently, when an upgrade occurs, the organization will purchase dozens or hundreds of servers at the same time, making it possible to negotiate a quantity discount.

We will learn that cloud providers exploit and extend the economic benefits described above. Not only do their data centers benefit from lower capex and lower opex, they also employ technologies that increase or decrease available resources dynamically, allowing them to achieve better overall utilization.

---

†Although the term *computing center* may seem more appropriate than *data center*, industry had used the term *computing center* decades earlier to describe a centralized mainframe facility; using a new name helped avoid the impression of moving backward.

1.8 Origin Of The Term "In The Cloud"

As Figure 1.1† illustrates, we use a cloud to depict the Internet in diagrams. The cloud represents a set of networks and the equipment that interconnects the networks. Saying a data center is "in the cloud" is technically inaccurate because the servers in a data center are not part of the Internet itself. Instead, servers are merely computers that attach to the Internet, and should be depicted outside the cloud, with network connections leading to the cloud.

Why did the industry start using the terms *cloud computing* and say that the computing is *in the cloud* if it is not? Early data centers that supplied large-scale web services needed high-capacity connections to major Internet backbone networks. Placing a data center near an Internet *peering point* where major Internet backbones interconnect minimizes cost. Some providers actually placed their data centers on the same floor in the same building as an Internet peering point. Because such a data center was located with networking equipment rather than in a separate building, engineers said the data center was *in the cloud*, and the terminology caught on. The point is:

> *Although technically inaccurate, the phrase* in the cloud *arose because early data centers were located close to networking equipment at Internet peering points rather than in separate buildings.*

1.9 Centralization Once Again

Surprisingly, after many decades of increasing decentralization, the move to data centers reverses the trend and moves back toward a more centralized model. The next chapter considers how public cloud providers extend centralization further by creating data centers with servers for multiple organizations. We can summarize:

> *For decades, the low cost of computers encouraged decentralization. The power wall and cost of IT staffing favor a return to a centralized model that consolidates computing facilities into data centers.*

†Figure 1.1 can be found on page 9.