6

Technological Drivers for Cloud Computing

Learning Objectives

The major learning objectives of this chapter are the following:

- To identify the various technological drivers of cloud computing paradigm
- To analyze each underlying technology in detail in order to understand the characteristic features and advantages
- To familiarize with the latest developments in each of these enabling technologies
- To understand how each of these technological components contributes to the success of cloud computing
- To introduce the readers to various case studies in these areas
- To understand how cloud service providers and cloud service consumers are benefitted from these technological advancements in the cloud scenario

Preamble

Cloud computing is an emerging computing paradigm where various users access the resources and services offered by service providers. From a technological perspective, cloud computing is the culmination of many components that enable the present-day cloud computing paradigm to offer the services efficiently to the consumers. Advancements in each of these technological areas have significantly contributed to the widespread adoption of cloud computing. This chapter focuses on identifying those technological drivers of cloud computing and also discusses each technological component in detail. The recent advancements in each of these technologies are highlighted with their advantages and characteristic features. The readers are expected to get the understanding of each of these technological components, their strengths and weaknesses, and also about the benefits provided by these technological drivers to various stakeholders such as service providers and service consumers.

6.1 Introduction

Cloud computing enables service providers to offer various resources such as infrastructure, platform, and software as services to the requesting users on a pay-as-you-go model. The cloud service consumers (CSCs) are benefitted from the cost reduction in procuring the resources and the quality of service (QoS) this promising computing paradigm offers. Nowadays, more companies and enterprises are entering the cloud scenario either as service providers or as service consumers. There has been a considerable increase in the adoption rate of cloud computing among service providers and users across the globe.

The success of cloud computing can be closely associated with the technological enhancements in various areas such as service-oriented architecture (SOA), virtualization, multicore technology, memory and storage technologies, networking technologies, Web 2.0, and Web 3.0. Also, the advancements in programming models, software development models, pervasive computing, operating systems (OSs), and application environment have contributed to the successful deployment of various clouds.

This chapter focuses on various technological drivers for cloud computing and also shows how the latest advancements in each of these enabling technologies have made an impact on the widespread adoption of cloud computing. This chapter also discusses each technological component in detail. The recent advancements in each of these technologies are highlighted with their advantages and characteristic features. The various benefits provided by these technological drivers to different stakeholders such as service providers and service consumers are emphasized.

6.2 SOA and Cloud

Many people are confused thinking that SOA and cloud computing are the same, or cloud computing is another name for SOA. This is not true. SOA is a flexible set of design principles and standards used for systems development and integration. A properly implemented SOA-based system provides a loosely coupled set of services that can be used by the service consumers for meeting their service requirements within various business domains. Cloud computing is a service delivery model in which shared services and resources are consumed by the users across the Internet just like a public utility on an on-demand basis. Generally, SOA is used by enterprise applications, and cloud computing is used for availing the various Internet-based services. Different companies or service providers may offer various services such as financial services, health-care services, manufacturing services, and HR services. Various users can acquire and leverage the offered services through the Internet. In such an environment, cloud is about services and service composition. Cloud offers various infrastructure components such as central processing unit (CPU), memory, and storage. It also provides various development platforms for developing softwares that offer their programmed features to various cloud consumers through service-oriented application programming interfaces (APIs). The programs running on cloud could be implemented using SOA-related technologies. A cloud user can combine the services offered by a cloud service provider (CSP) with other in-house and public cloud services to create SOA-based composite applications.

6.2.1 SOA and SOC

The service-oriented computing (SOC) paradigm utilizes the services for the rapid and low-cost development of interoperable distributed applications in heterogeneous environments. In this paradigm, services are autonomous and platform-independent entities that can be described, published, discovered, and loosely coupled using various protocols and specifications. SOC makes it possible to create cooperating services that are loosely coupled, and these services can be used to create dynamic business processes and agile applications in heterogeneous computing platforms. SOC uses the services architectural model of SOA as shown in Figure 6.1. This model consists of entities such as

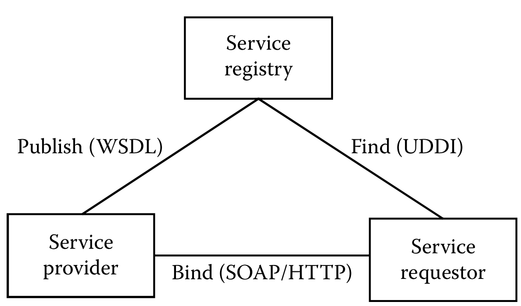

FIGURE 6.1
Services architectural model of SOA.

service provider and service requestor. Service providers publish the details of their services in the service registry using an Extensible Markup Language (XML) called Web Services Description Language (WSDL). Service requestors find the suitable services from the service registry using specifications such as Universal Description, Discovery, and Integration (UDDI). Service providers and service requestors communicate with each other using protocols such as Simple Object Access Protocol (SOAP). SOAP allows a program or service running on one platform to communicate with another program or service running on a different platform, using the Hypertext Transfer Protocol (HTTP) and its XML as the mechanisms for information exchange.

6.2.2 Benefits of SOA

SOA enables mutual data exchange between programs of different vendors without the need for additional programming or changes to the services. The services should be independent, and they should have standard interfaces that can be called to perform their tasks in a standard way. Also, a service need not have prior knowledge of the calling application, and the application does not need to have knowledge about how the tasks are performed by a service. Therefore, the various benefits of SOA are as follows:

1. *Reuse of services*: Various services can be reused by different applications, which results in lower development and maintenance costs. Having reusable services readily available also results in quicker time to market.
2. *Agility*: SOA can bring the architectural agility in an enterprise through the wide use of standards such as web services. This is the ability to change the business processes quickly when needed to support the change in the business activities. This agility aspect helps to deal with system changes using the configuration layer instead of redeveloping the system constantly.
3. *Monitoring*: It helps to monitor the performance of various services to make the required changes.
4. *Extended reach*: In the collaboration between enterprises or in the case of shared processes, it is the ability to get the service of various other processes for completing a particular task.

Hence, SOA can be used as an enabling technology to leverage cloud computing. Cloud computing offers on-demand information technology (IT) resources that could be utilized by extending the SOA outside of the enterprise firewall to the CSPs' domain. This is the process of SOA using cloud computing.

6.2.3 Technologies Used by SOA

There are many technologies and standards used by SOA, which could also be utilized in the cloud computing domain for delivering services efficiently to cloud customers. Some of the standards or protocols are given in the following:

1. *Web services*: Web services can implement an SOA. Web services make functional components or services available for access over the Internet, independent of the platforms and the programming language used. UDDI specification defines a way to publish and discover information about web services.
2. *SOAP*: The SOAP protocol is used to describe the communications protocols.
3. *RPC*: Remote procedure call (RPC) is a protocol that helps a program to request a service from another program located in another computer in a network, without the need to understand network details.
4. *RMI-IIOP*: This denotes the Java remote method invocation (RMI) interface over the Internet Inter-ORB Protocol (IIOP). This protocol is used to deliver Common Object Request Broker Architecture (CORBA) distributed computing capabilities to the Java platform. It supports multiple platforms and programming languages and can be used to execute RPCs on another computer as defined by RMI.
5. *REST*: REpresentational State Transfer (REST) is a stateless architecture that runs over HTTP. It is used for effective interactions between clients and services.
6. *DCOM*: Distributed Component Object Model (DCOM) is a set of Microsoft concepts and program interfaces in which client program can request the services from a server program running on other computers in a network. DCOM is based on the Component Object Model (COM).
7. *WCF* (*Microsoft implementation of web service forms a part of WCF*): Windows Communication Foundation (WCF) provides a set of APIs in the .NET Framework for building connected, service-oriented applications.

6.2.4 Similarities and Differences between SOA and Cloud Computing

There are certain common features that SOA and cloud computing share while being different from one another in certain other areas.

6.2.4.1 Similarities

Both cloud computing and SOA share some core principles. First, both rely on the service concept to achieve the objectives. Service is a functionality or a feature offered by one entity and used by another. For example, a service could be retrieving the details of the online bank account of a user. SOA and cloud computing use service delegation in that the required task is delegated either to service provider (in the case of cloud computing) or to other application or business components in the enterprise (in the case of SOA). Service delegation helps the people to use the services without being concerned about the implementation and maintenance details. Services could be shared by multiple applications and users, thereby achieving optimized resource utilization. Second, both cloud computing and SOA promote loose coupling among the components or services, which ensures the minimum dependencies among different parts of the system. This feature reduces the impact that any single change on one part of the system makes on the performance of the overall system. Loose coupling helps the implemented services to be separated and unaware of the underlying technology, topology, life cycle, and organization. The various formats and protocols used in distributed computing, such as XML, WSDL, Interface Description Language (IDL), and Common Data Representation (CDR), help to achieve the encapsulation of technology differences and heterogeneity among the various components used for combining a business solution for solving the computing problems. Various services should be location and technology independent in cloud computing, and SOA can be used for achieving this transparency in the cloud domain.

6.2.4.2 Differences

There are also some differences between the SOA and the cloud computing paradigm. The services in SOA mainly focus on business. Each service in SOA may represent one aspect of the business process. The services could be combined together to provide the required complete business application or business solution. Hence, in this sense, the services are horizontal. At the same time, various services in cloud computing are usually layered such as infrastructure, platform, or software, and the lower layer services support the upper services to deliver applications. Hence, the services in this case are vertical. SOA is used for defining the application architecture. The various components or services of the application are divided based on their roles in the SOA applications. That means the solution for a business problem could be achieved by combining the various abstract services performing the required functions. The services in the SOA can be reused by other applications. Cloud computing is a mechanism for delivering IT services. The various services can be divided or grouped based on their roles such as infrastructure, platform, or software. In this case, for utilizing the cloud services, the consumer does not require a problem before defining the cloud services. The services in this case could also be reused by other applications.

6.2.5 How SOA Meets Cloud Computing

SOA is widely considered to be an enabling technology for cloud computing. In the case of cloud computing, it requires high degree of encapsulation. There should not be any hard dependencies on resource location in order to achieve the true virtualization and elasticity in cloud. Also, threads of execution of various users should be properly isolated in cloud, as any vulnerability will result in the information or data of one user being leaked into another consumer. The web services standards (WS*) used in SOA are also used in the cloud computing domain for solving various issues, such as asynchronous messaging, metadata exchange, and event handling. SOA is an architectural style that is really agnostic of the technology standards adopted in the assembly of composite applications. The service orientation provided by SOA helps in the software design using different pieces of software, each providing separate application functionalities as services to other applications. This feature is independent of any platform, vendor, or technology. Services can be combined by other software applications to provide the complete functionality of a large software application. SOA makes the cooperation of computers connected over a network easy. An arbitrary number of services could be run on a computer, and each service can communicate with any other service in the network without human interaction and also without the need to make any modification to the underlying program itself. Within an SOA, services use defined protocols for transferring and interpreting the messages. WSDL is used to describe the services. The SOAP protocol is used to describe the communications protocols.

SOA is an architecture, and cloud computing is an instance of architecture or an architectural option, not an architecture by itself. When used with cloud computing, SOA helps to deliver IT resources as a service over the Internet, and to mix and match the resources to meet the business requirements. In an enterprise, the database could be hosted with one CSP, process server with another CSP, application development platform with another CSP, and web server with yet another CSP. That means the SOA can be extended to the cloud computing providers to provide a cost-effective solution in such a way that the cloud-based resources and on-premise resources work in tandem. SOA using cloud computing architecture provides the agility in such a way that it could easily be changed to incorporate the business needs since it uses services that are configured through a configuration or process layer.

Cloud and SOA are considered to actually complement each other. SOA is an architectural style for building loosely coupled applications and allows their further composition. It also helps in creating the services that are shared and reused. Cloud computing provides repeatability and standardized easy access to the various shared hardware and software at low cost. In fact, it offers a number of *X as a Service* capabilities. SOA and cloud together provide the required complete services-based solution. Hence, cloud and SOA are required to work together to provide service visibility and service management. Service visibility and governance provide the users the functionality of service discovery within a cloud, and the SOA service management helps in managing the life cycle of services available in cloud. Thus, through the integration of cloud and SOA, cloud can take advantage of the SOA governance approach without the necessity for creating new governance overhead. Having SOA and service orientation in place, the companies or organizations can make adopting cloud services easier and less complex, because a cloud computing environment is also based on services. Both cloud and SOA are focused on delivering services to the business with increased agility, speed, and cost effectiveness.

Multitenancy is the characteristic feature of cloud computing systems. It should be noted that this is a feature possessed by the SOA-based systems. In the multitenant application, a CSP has one instance of a program or the application running on the server, and more than one customer at a time is using the copies of the application. An example is Gmail or Hotmail program. Multitenancy improves the efficiency of the cloud system. In single-tenant applications, only one user at a time is using the application provided by the service provider. An example could be a text editor application used by a single user at a time. In cloud, multitenant applications are preferred as it helps in effective resource utilization. The CSC has to trust the CSP that it will perform its intended tasks or functions without fail.

SOA is considered to be the ideal architecture for cloud computing. Moving to the cloud environment requires having a solid SOA to provide the infrastructure required for successful cloud implementation. Cloud computing is a deployment architecture, and SOA is an architectural approach for how to architect the enterprise IT. Hence, cloud requires the service orientation provided by SOA [1]. With SOA already deployed and executed successfully, taking the full advantage of cloud computing becomes reliable, faster, and more secure. Figure 6.2 shows how SOA can be extended to use the cloud services. The perceived benefits of this integration could be improved collaboration, customer satisfaction, and business growth. By utilizing the SOA for deploying the business capabilities to the cloud environment, businesses can considerably improve the interactions with their business partners and existing customers, thereby increasing their revenues.

6.2.6 CCOA

Cloud computing open architecture (CCOA) [2] is an architecture for the cloud environment that incorporates the SOA. The goals of the CCOA are as follows:

1. To develop an architecture that is reusable and scalable. That means, in future, the architecture should incorporate any further changes without the need for replacing the entire architecture.

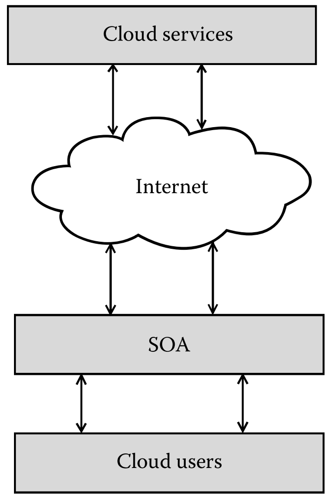

FIGURE 6.2
Convergence of SOA and cloud.

2. To develop a uniform platform for the cloud application development. This will allow the cloud users to switch between the CSPs without the need to make significant changes in the application.
3. To enable the businesses to run efficiently. This goal helps the CSPs to make more money by delivering quality services successfully.

Hence, the CCOA provides the required general guidelines and instructions for the design and development of scalable, reusable, and interoperable cloud applications incorporating SOA principles into them.

6.3 Virtualization

Virtualization is the underlying core technology of cloud computing. It helps in creating a multitenant model for the cloud environment by optimizing the resource usage through sharing (for more details on virtualization you may refer Chapter 7). Benefits of virtualization include the lower costs and extended life of the technology, which has made it a popular option with small- to medium-sized businesses [3]. Using virtualization, the physical infrastructure owned by the service provider is shared among many users, increasing the resource utilization. Virtualization provides efficient resource utilization and increased return on investment (ROI). Ultimately, it results in low capital expenditures (CapEx) and operational expenditures (OpEx).

Some of the benefits of virtualization include better utilization rate of the resources of the service providers, increased ROI for both the service providers and the consumers, and promotes the green IT by reducing energy wastage. Virtualization technology has the drawbacks of the chance of a single point of failure of the software achieving the virtualization and the performance overhead of the entire system due to virtualization.

6.3.1 Approaches in Virtualization

There have been many approaches adopted in the implementation of virtualization technology. Some of the important approaches are discussed in the following subsections.

6.3.1.1 Full Virtualization

Full virtualization uses a special kind of software called a hypervisor. The hypervisor interacts directly with the physical server's hardware resources, such as the CPU and storage space, and acts as a platform for the virtual server's OSs. It helps to keep each virtual server completely independent and unaware of the other virtual servers running on the physical machine. Each guest server or the virtual machine (VM) is able to run its own OS. That means one virtual server could be running on Linux and the other one could be running on Windows. Examples include VMWare ESX and VirtualBox. In the full virtualization, the guest OS is unaware of the underlying hardware infrastructure. That means the guest OS is not aware of the fact that it is running on a virtualized platform and of the feeling that it is running on the real hardware. In this case, the guest OS cannot communicate directly to the underlying physical infrastructure. The OS needs the help of virtualization software hypervisors to communicate with the underlying infrastructure. The advantages of the full virtualization include isolation among the various VMs, isolation between the VMs and the hypervisor, concurrent execution of multiple OSs, and no change required in the guest OS. A disadvantage is that the overall system performance may be affected due to binary translation.

6.3.1.2 Paravirtualization

In this case, VMs do not simulate the underlying hardware, and this uses a special API that a modified guest OS must use. Examples include Xen and VMWare ESX server. In this type of virtualization, partial simulation of the underlying hardware infrastructure is achieved. This is also known as *partial virtualization* or *OS-assisted virtualization*. This virtualization is different from the full virtualization in that, here, the guest OS is aware of the fact that it is running in a virtualized environment. In this case, hypercalls are used for the direct communication between the guest OS and the hypervisor. In paravirtualization, a modified or paravirtualized guest OS is required.

An advantage of this approach is that it improves the overall system performance by eliminating the overhead of binary translation. A disadvantage could be that a modification of the guest OS is required.

6.3.1.3 Hardware-Assisted Virtualization

In this type of virtualization, hardware products supporting the virtualization are used. Hardware vendors like Intel and AMD have developed processors supporting the virtualization through the hardware extension. Intel has released its processor with its virtualization technology VT-x, and AMD have released its processor with its virtualization technology AMD-v to support the virtualization. An advantage of this approach could be that it eliminates the overhead of binary translation and paravirtualization. A disadvantage includes the lack of support from all vendors.

6.3.2 Hypervisor and Its Role

The concept of using VMs increases the resource utilization in a cloud computing environment. Hypervisors are software tools used to create the VMs, and they produce the virtualization of various hardware resources such as CPU, storage, and networking devices. They are also called virtual machine monitor (VMM) or virtualization managers. They help in the virtualization of cloud data centers (DCs). The various hypervisors used are VMware, Xen, Hyper-V, KVM, etc. Hypervisors help to run multiple OSs concurrently on a physical system sharing its hardware. Thus, a hypervisor allows multiple OSs to share a single hardware host. In this case, every OS appears to have the host's processor, memory, and other resources allocated solely to it. However, the hypervisor is actually controlling the host processor and resources and in turn allocates what is needed to each OS. The hypervisor also makes sure that the guest OSs (called VMs) do not interrupt each other. In virtualization technology, hypervisor manages multiple OSs or multiple instances of the same OS on a single physical computer system. Hypervisors are designed to suit a specific processor, and they are also called virtualization managers.

Hypervisors are of mainly two types:

1. *Type 1 hypervisor*: This type of hypervisor runs directly on the host computer's hardware in order to control the hardware resources and also to manage the guest OSs. This is also known as native or bare-metal hypervisors. Examples include VMware ESXi, Citrix XenServer, and Microsoft Hyper-V hypervisor.
2. *Type 2 hypervisor*: This type of hypervisor runs within a formal OS environment. In this type, the hypervisor runs as a distinct second layer while the guest OS runs as a third layer above the hardware. This is also known as the hosted hypervisors. Examples include VMware Workstation and VirtualBox.

6.3.3 Types of Virtualization

Depending on the resources virtualized, the process of virtualization can be classified into the following types.

6.3.3.1 OS Virtualization

In OS virtualization, a desktop's main OS is moved into a virtual environment. The computer that is used by the service consumers remains on their desk, but the OS is hosted on a server elsewhere. Usually, there is one version of the OS on the server, and copies of that individual OS are given to the individual user. Various users can then modify the OS as they wish, without affecting the other users.

6.3.3.2 Server Virtualization

In server virtualization, existing physical servers are moved into a virtual environment, which is then hosted on a physical server. Modern servers can host more than one server simultaneously, which allows the users to reduce the number of servers to be reserved for various purposes. Hence, IT and administrative expenditures are reduced. Server virtualization can use the virtual processors created from the real hardware processor present in the host system. The physical processor can be abstracted into a collection of virtual processors that could be shared by the VMs created.

6.3.3.3 Memory Virtualization

In main memory virtualization, the virtual main memory that is abstracted from the physical memory is allocated to various VMs to meet their memory requirements. The mapping of physical to virtual memory is performed by the hypervisor software. The main memory virtualization support is provided with the modern x86 processors. Also, the main memory consolidation in the virtualized cloud DCs could be performed by the hypervisor by aggregating the free memory segments of various servers to create a virtual memory pool that could be utilized by the VMs.

6.3.3.4 Storage Virtualization

In storage virtualization, multiple physical hard drives are combined into a single virtualized storage environment. To various users, this is simply called cloud storage, and it could be a private storage, such that it is hosted by a company, or a public storage, such that it is hosted outside of a company like DropBox, or a mixed approach of the two. In the case of storage virtualization, physical storage disks are abstracted to a virtual storage media. In cloud DCs, high availability and backup of the user's data are achieved through storage virtualization technology. Modern hypervisors help in achieving storage virtualization. The concept of storage virtualization is implemented in advance storage techniques such as storage area networks (SANs) and network-attached storage (NAS).

6.3.3.5 Network Virtualization

In network virtualization (NV), logical virtual networks are created from the underlying physical network. The physical networking components such as the router, switch, or network interface card could be virtualized by the hypervisor to create logical equivalent components. Multiple virtual networks can be created by using the same physical network components that could be used for various purposes. NV can also be achieved by combining various network components from multiple networks.

6.3.3.6 Application Virtualization

In application virtualization, the single application installed on the central server is virtualized and the various virtualized components of the application will be given to the users requesting the services. In this case, the application is given its own copy of components such as own registry files and global objects that are not shared with others. The virtual environment prevents conflicts in the resource usage. An example is the Java Virtual Machine (JVM). In the cloud computing environment, the CSPs deliver the SaaS model through the application virtualization technology. In the case of application virtualization, the cloud users are not required to install the required applications on their individual systems. They can, in turn, get the virtualized copy of the application, and customize and use it for their own purposes.

6.4 Multicore Technology

In multicore technology, two or more CPUs are working together on the same chip. In this type of architecture, a single physical processor contains the core logic of two or more processors. These processors are packaged into a single integrated circuit (IC). These single ICs are called a die. Multicore technology can also refer to multiple dies packaged together. This technology enables the system to perform more tasks with a greater overall system performance. It also helps in reducing the power consumption and achieving more efficient, simultaneous processing of multiple tasks. Multicore technology can be used in desktops, mobile personal computers (PCs), servers, and workstations. Hence, this technology is used to speed up the processing in a multitenant cloud environment. Multicore architecture has become the recent trend of high-performance processors, and various theoretical and case study results illustrate that multicore architecture is scalable with the number of cores.

6.4.1 Multicore Processors and VM Scalability

In the multicore processor–based system for cloud, state-of-the-art computer architectures are used to allow multiple VMs to scale as long as the cache, memory, bus, and network bandwidth limits are not reached. Thus, in the cloud computing domain, the CPU and memory-intensive virtualized workloads should scale up to the maximum limits imposed by the memory architecture [4].

6.4.2 Multicore Technology and the Parallelism in Cloud

In multicore chips, multiple simpler processors are deployed instead of a single large one, and the parallelism becomes exposed to programmers [5,6]. In the processor design, the development of multicore processors has been a significant recent architectural development. In order to exploit this architectural design fully, the software running on this hardware needs to exhibit concurrent behavior. This fact places greater emphasis on the principles, techniques, and technologies of concurrency. Multicore architecture dominates the server cloud today as it improves the speed and efficiency of processing in cloud. In this case, the main issue is how efficiently the multicore chips are used in the server clouds today in order to gain the real parallelism in terms of performance gain in a multitenant cloud environment through efficient programming.

6.4.3 Case Study

Chip-maker Intel has launched a second-generation family of system on chip (SoC) for microservers [7]. A 64-bit Intel Atom C2000 SoC product is designed for microservers and storage and networking platforms. This is suitable for software-defined networking (SDN). Microservers are tiny, power-efficient machines designed to handle light workloads such as entry-level web hosting. They are suited for small or temporary jobs without provisioning resources from contemporary high-end servers. The Atom C2000 is based on the Silvermont microarchitecture and is aimed at improving performance and energy efficiency. It features up to eight cores, up to 20 W TDP (Thermal Design Power), integrated Ethernet and supports up to 64 GB of memory. It addresses the specialized needs for securing and routing Internet traffic more efficiently. The web-hosting service company 1&1 has tested Atom C2000 as a pilot and is planning to deploy the chip in its entry-level dedicated hosting service. Telecommunications provider Ericsson will also add Atom C2000 SoC to its cloud service platform.

6.5 Memory and Storage Technologies

In the storage domain, it is seen that file-based data growth is faster than block-based data. Also, unstructured data growth is faster than structured data. Most of the organization's data are unstructured, and more than 50% of new storage requirements are consumed by unstructured data such as e-mail, instant messaging, radio frequency identification (RFID) MP3 players, photos, medical imaging, satellite images, and GPS. Hence, the memory or storage solutions used in the cloud environment should support the cloud requirements. The cloud storage has to deal with various kinds of data such as medical images, MP3, photos, 3D high-definition imaging, video streaming, surveillance camera captures, and film animations.

6.5.1 Cloud Storage Requirements

The storage technology or solutions used in the cloud environment should meet the following requirements.

1. *Scalability*: The storage system should support the scalability of the user's data.
2. *High availability*: The degree of availability of the storage solutions deployed in cloud should be very high.
3. *High bandwidth*: The cloud storage system should support the required fast data transfer rate.
4. *Constant performance*: There should not be any performance issues associated with the cloud storage system, and the performance should be consistent throughout the contract period.
5. *Load balancing (LB)*: In order to achieve effective resource usage, the storage systems deployed in cloud should be intelligent enough to support automatic LB of the users' data.

6.5.2 Virtualization Support

In storage virtualization, multiple network storage devices are amalgamated into a single storage unit. This type of virtualization is often used in SAN, which is a high-speed network of shared storage devices, and the SAN technology makes tasks such as archiving, backup, and recovery processes easier and faster. Software applications are used to implement the storage virtualization. Storage virtualization involves the pooling of physical storage from various network storage devices into a single logical storage device, and that is managed from a centralized console. Storage virtualization helps in achieving the easy and efficient backup, archiving, and recovery processes.

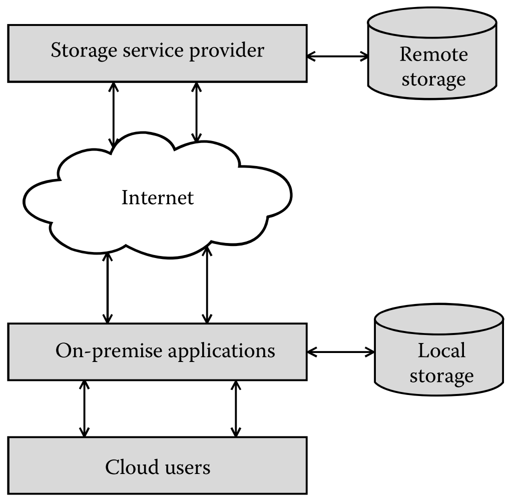

FIGURE 6.3
Storage as a Service.

6.5.3 Storage as a Service (STaaS)

Cloud storage can be internal to the organization or it could be an external storage where the storage is provided by a CSP located outside the organization's DC, which is also known as Storage as a Service (STaaS). STaaS is a cloud business model in which a service provider rents space in its storage infrastructure to various cloud users. Figure 6.3 shows the SaaS cloud model.

The subscribers of the STaaS can have significant cost savings in hardware, maintenance, etc. The STaaS provider agrees in the service-level agreement (SLA) to rent storage space on a cost-per-gigabyte-stored and cost-per-data-transfer basis. STaaS also helps in backup, disaster recovery, business continuity, and availability. Another advantage of STaaS is the capability to access the data stored in cloud from anywhere. Here, the storage is delivered on demand.

6.5.4 Emerging Trends and Technologies in Cloud Storage

The memory and storage technologies have been developing at a rapid pace, and the emerging technologies enable cloud to have a reliable, secure, and scalable storage system. The following are some of the developments in the memory and storage technologies that help the cloud system achieve its efficiency [8,9]:

- Hybrid HDDs with magnetic and flash memory having second-level cache
- Developments in the RAID technology such as RAID 6, RAID triple parity, erasure coding + RAIN

- Converging of block, file, and content data in a single storage subsystem
- Embedded deduplication, primary storage reductions in the storage units
- Usage of object-based storage device (OSD)
- D-RAM SSDs and flash HDDs having the features of improved server utilization, much less energy consumption, being less sensitive to vibration, etc.
- File virtualization or clustered NAS, which supports single namespace to view all files, scaling near linearly by adding nodes, better availability and performance, and LB. These approaches are best suited for seismic processing, video rendering, simulations, auto/aero/electronics design, etc.

The latest models of cloud storage appliances such as Whitewater cloud storage appliance can support up to 14.4 petabytes of logical data. These provide greater capacity, faster speeds, and more replication options. Disk-to-disk backup architectures have become extremely popular in the recent years. Adding the ability to integrate public cloud storage into this architecture offers an immediate return to operations at a disaster recovery site while capturing the cost advantages of very aggressive cloud storage services such as Amazon Glacier.

6.6 Networking Technologies

In cloud, the networking features should support the effective interaction between the CSPs and the CSCs [10].

6.6.1 Network Requirements for Cloud

The various networking requirements in the cloud environment are the following:

1. *Consolidate workloads and provide Infrastructure as a Service* (*IaaS*) *to various tenants*: The network technology should enable enterprise workload consolidation, which reduces the management overhead and provides more flexibility and scalability for the management of VMs.
2. *Provide VM connectivity to physical and virtual networks*: The cloud network system should support programmatically managed and extensible features to connect the various VMs to both virtual networks and the physical network. It should also enforce policy enforcement for security, isolation, and service levels.

3. *Ensure connectivity and manage network bandwidth*: The cloud network system should have the load balancing and failover (LBFO) features that allow server bandwidth aggregation and traffic failover.
4. *Speed application and server performance*: The network and OS performance should be improved with high-speed networking features. Various features and technologies such as low-latency technologies, Data Center Bridging, and Data Center Transmission Control Protocol (DCTCP) could be utilized to improve the performance of the cloud networking infrastructure.

6.6.2 Virtualization Support

In the NV network, resources are used through a logical segmentation of a single physical network. NV could be achieved by using software and services to manage the sharing of storage, computing cycles, and the various applications. It considers all the servers and services in the network as a single logical collection of resources that can be accessed without considering the physical components. Hence, NV creates logical, virtual networks that are decoupled or separated from the underlying physical network hardware to ensure that the network is better integrated with the virtual environments. In this case, physical networking devices are simply responsible for the forwarding of the data packets, while the software-controlled virtual network provides the required abstraction that makes it easy to deploy and manage network services and the underlying network resources. Figure 6.4 shows the virtualization support in the networking.

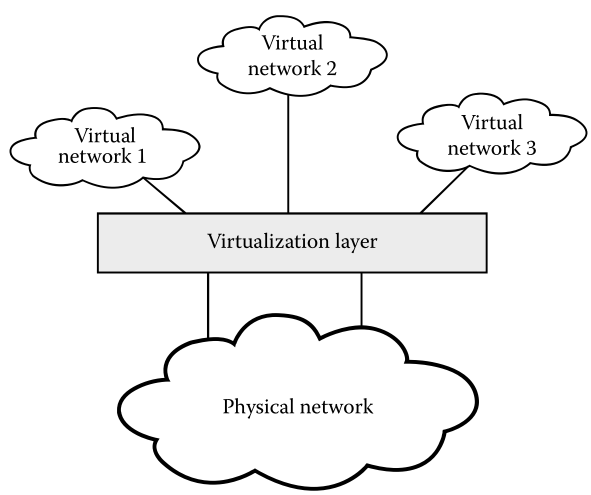

FIGURE 6.4
Virtualization in networking.

NV helps in the efficient utilization of network resources through logical segmentation of a single physical network. This type of virtualization can be used by different organizational units or departments in a company to share the company's physical network as separate logical networks. Here, the total cost of capital expenditures and operating expenses is reduced by sharing the various network resources. In NV, secure logical separation between organizations or groups is maintained. The logical segmentation of the physical network into secure virtual networks could be performed by overlaying virtual private network (VPN) mechanisms such as 802.1x, network admission control (NAC), generic routing encapsulation (GRE) tunnels, virtual routing and forwarding (VRF)-lite, and multiprotocol label switching (MPLS) VPNs onto the existing local area network (LAN).

6.6.3 Usage of Virtual Networks

NV can be used to create virtual networks within a virtualized infrastructure. This helps the NV support the complex resource requirements in multitenant cloud environments. NV can facilitate a virtual network within a virtualized environment that is decoupled from the other network resources. In these virtual environments, NV can separate the data traffic into various zones or containers in order to ensure that the traffic is not mixed with other resources or the transfer of other data.

6.6.4 DCs and VPLS

Truly virtualized DCs should support effective VM migration. In a virtualized DC environment, the VMs can be migrated from one server to another in order to improve the server utilization rate, and also to achieve effective LB. Virtual private LAN service (VPLS) provides the flat network and the long reach that could be used to connect the servers that are geographically apart, thereby achieving VM migration and LB.

6.6.5 SDN

SDN is an approach to networking in which control is decoupled from networking hardware and given to a software application called a controller. In a conventional network, when a data packet arrives at a switch, rules built into the switch's firmware guide the switch in forwarding the data packet. In a software-defined network, a network administrator can control traffic from a centralized control console without having to deal with individual switches. The administrator can change any network switch's rules when necessary such as blocking specific types of packets. This is extremely helpful in cloud computing environment that has a multitenant architecture as it gives the required flexibility and efficiency for the administrator to manage traffic loads. Nowadays, the software-defined network could be created using open standard specifications such as OpenFlow, which allows the network administrators to control routing tables remotely. This could be used in cloud environment for applications that require bandwidth and delay guarantees. Intel has launched Ethernet Switch FM5224 silicon designed to facilitate SDN and to improve compute density and power efficiency.

6.6.6 MPLS

MPLS is a switching mechanism in high-performance telecommunication networks that directs data from one network node to another based on short path labels rather than long network addresses. MPLS sets up a specific path for a given data packet that is identified by a label put in each packet. The various labels identify virtual links or paths between distant nodes rather than end points. This reduces the time needed for a router to look up the address to the next node in data forwarding. MPLS is called multiprotocol because it can be used to encapsulate data packets of different network protocols and technologies such as the Internet Protocol (IP), Asynchronous Transport Mode (ATM), and frame relay network protocols. Thus, MPLS increases the network speed and its manageability.

6.6.7 Other Emerging Networking Trends and Technologies in Cloud

For high bandwidth and ultralow latency in cloud computing, dense wavelength division multiplexing (DWDM) appears to be very promising as a future high-performance WAN transport technology. DWDM is protocol and bit-rate independent, and also it can multiplex multiple optical signals. Researchers are working in new wide area network (WAN) networking technologies such as Lambda networking that promises low-cost, high-capacity circuits in networking. Inside a cloud DC, various network servers, storage systems, network nodes, and other elements are interconnected. Three LAN networking technologies, namely Ethernet, FiberChannel, and InifiBand, are used in the DCs [11]. Ten GB Ethernet (10GBE) equipment and networks are widely used. FiberChannel is suitable for scientific computing and SANs, and InifiBand is almost exclusively deployed in scientific and engineering simulation networks, for example, using clustered servers.

Large cloud DCs support tens of thousands of servers, exabytes of storage, terabits per second of traffic, and tens of thousands of tenants [12]. In a DC, server and storage resources are interconnected with packet switches and routers that provide for the bandwidth and multitenant virtual networking needs. DCs are interconnected across the WAN via routing and transport technologies to provide a pool of resources, known as cloud. High-speed optical interfaces and DWDM optical transport are used to provide for high-capacity transport intra- and inter-DCs.

Private DCs are interconnected through an enterprise-dedicated private network or VPN. Public DCs are connected through the Internet, and they offer multitenant Internet-based services. Virtual private DCs could be built using a common DC infrastructure provided by an IaaS provider. The underlying network for a virtual private DC provides the tenant network isolation and privacy features. Cloud services could be built by interconnecting DCs and utilizing the collective set of resources in these DCs as a resource pool.

Virtualization technology is used to create a VM on a server with dedicated CPU cycles, memory, storage, and input/output (I/O) bandwidth. Multiple VMs for various tenants can be created on the same physical server. A tenant in the cloud environment may also be provided a set of VMs residing on servers distributed throughout a DC or even across various DCs. NV is used as an evolving technology to create intra- and inter-DC networks from the basic virtual local area network (VLAN) and IP routing architecture. This helps to support a large number of tenants and enables networking among their virtualized resources. In the case of large bandwidth requirements such as dealing with large blocks of data and video among DCs, a DC interconnection may utilize an optical transport mechanism. Various network services such as firewalling (FW), service LB, and network address translation (NAT) are provided as part of a tenant virtual network.

VPN technologies are used to carry both Ethernet and IP customer traffic across service provider IP/MPLS networks using tunneling technologies such as IP or MPLS. This helps to achieve the isolation among the customers, and these technologies can be utilized in the cloud domain. In a shared IP/MPLS packet-switched network (PSN), VPLS could be utilized to provide a transparent LAN service. Border Gateway Protocol (BGP) and MPLS IP VPNs [13] are used for providing private customer IP routing over a shared IP/MPLS PSN. Ethernet VPN (EVPN) [14] is an emerging technology aiming to provide Ethernet LAN service over an IP/MPLS PSN. In cloud computing environments, larger switches, higher-speed Ethernet interfaces, and DWDM transport address the bandwidth requirements. Also, evolving packet technologies such as VXLAN as well as Ethernet and IP VPNs are used to create the next-generation DC networking paradigm. Various networking protocols, standards, and technologies are rapidly advancing to meet the requirements of the inter- and intra-DC networks in the cloud domain.

6.7 Web 2.0

Web 2.0 (or Web 2) is the popular term given to the advanced Internet technology and applications that include blogs, wikis, really simple syndication (RSS), and social bookmarking. The two major contributors of Web 2.0 are the technological advances enabled by Ajax and other applications such as

RSS and Eclipse that support the user interaction and their empowerment in dealing with the web. The term was coined by Tim O'Reilly, following a conference dealing with next-generation web concepts and issues held by O'Reilly Media and MediaLive International in 2004. One of the most significant differences between Web 2.0 and the traditional World Wide Web (referred to as Web 1.0) is that Web 2.0 facilitates greater collaboration and information sharing among Internet users, content providers, and enterprises. Hence, in that sense, this can be considered as a migration from the *read-only web* to a *read/write web*.

As an example for this paradigm, multiple-vendor online book outlets such as BookFinder4U allow the users to upload book reviews to the site and also help the users find rare and out-of-print books at a reasonable price. In another example, dynamic encyclopedias such as Wikipedia permit users not only to read the stored information but also to create and edit the contents of the information database in multiple languages. Internet forums such as blogging have become more popular and extensive and have led to the proliferation and sharing of information and views. Also, RSS feeds have been used for the dissemination of news across users and websites.

The main focus of Web 2.0 is to provide the web users the ability to share and distribute information online with other users and sites. It refers to the transition from the static HTML web pages to a more dynamic web for serving web applications to users effectively. A Web 2.0 site such as a social networking site allows its users to interact with each other in a social media dialogue, in contrast to websites where people are restricted to the passive viewing of information [15]. Common examples of Web 2.0 include social networking sites, blogs, wikis, video-sharing sites, and any other hosted services or web applications that allow dynamic sharing of information among users. Also, Web 2.0 technologies can be used as interactive tools to provide feedback on contents or information provided in the web page such as the best practices and recent updates. This feedback will help the service providers to improve the quality of their services and thereby the business values.

Cloud computing is closely related to the SOA of Web 2.0 and virtualization [16] of hardware and software resources. Cloud computing makes it possible to build applications and services that can run/execute utilizing the resources (hardware and software) provided by the service providers, without restricting the application developers or consumers to the resources available on premise.

6.7.1 Characteristics of Web 2.0

In Web 2.0 websites, as already mentioned, instead of merely *reading* the contents from a web page, a user is allowed to *write* or contribute to the content available to everyone in an effective and user-friendly manner. Web 2.0 is also called *network as a platform computing* as it provides software, computing, and storage facilities to the user all through the browser. The major applications of Web 2.0 include social networking sites, self-publishing platforms, tagging, and social bookmarking.

The key features of Web 2.0 include the following:

- Folksonomy
- Rich user experience
- User as a contributor
- User participation
- Dispersion

Folksonomy allows the free classification of information available on the web, which helps the users to collectively classify and find information using approaches such as tagging. Rich user experience is provided because of the dynamic content offered on the web that is responsive to user input in a user-friendly manner. Web 2.0 allows a web user to assume the role of information contributor as information flows in two ways, that is, between site owner and site user by means of evaluation, review, and feedback. This paradigm also facilitates user participation as site users are allowed to add content for others to see (e.g., crowdsourcing). The contributions made by the individual users are available for other users to use and reuse as the web contents. Also, multiple channels are used for content delivery among the users.

Some of the characteristic features of Web 2.0 are as follows [17]:

1. *Blogging*: Blogging allows a user to make a post to a web log or a blog. A blog is a journal, diary, or a personal website that is maintained on the Internet, and it is updated frequently by the user. Blogs increase user interactivity by including features such as comments and links.
2. *Usage of Ajax and other new technologies*: Ajax is a way of developing web applications that combines XHTML and CSS standards–based presentation. It allows the interaction with the web page through the DOM and data interchange with XML and XSLT.
3. *RSS-generated syndication*: RSS is a format for syndicating web content. It allows to *feed* the freshly published web content to the users through the RSS reader/aggregator.
4. *Social bookmarking*: Social bookmarking is a user-defined taxonomy system for storing tags to web contents. The taxonomy is also called *folksonomy,* and the bookmarks are referred to as tags. Instead of storing bookmarks in a folder on the user's computer, tagged pages are stored on the web increasing the accessibility from any computer connected to the Internet.

TABLE 6.1

Differences between Web 1.0 and Web 2.0

Feature	Web 1.0	Web 2.0
Authoring mechanism	Personal websites	Blogging
Information sources	Britannica	Online Wikipedia
Content creation and maintenance	Via CMS	Via wikis
Data storage	Local disk	Online disk
Online advertising	Banners	Google AdSense
Online payment	Bank account	PayPal

5. *Mash-ups*: A mash-up is a web page or an application that can integrate information from two or more sources. Development methodologies using Ajax can be used to create mash-ups. It helps to create a more interactive and participatory web with user-defined contents and services integrated.

6.7.2 Difference between Web 1.0 and Web 2.0

Some of the differences between Web 1.0 and Web 2.0 are shown in Table 6.1.

6.7.3 Applications of Web 2.0

Web 2.0 finds applications in different fields. Some of the applications of Web 2.0 are discussed in the following subsections.

6.7.3.1 Social Media

Social web is an important application of Web 2.0 as it provides a fundamental shift in the way people communicate and share information. The social web offers a number of online tools and platforms that could be used by the users to share their data, perspectives, and opinions among other user communities.

6.7.3.2 Marketing

Web 2.0 offers excellent opportunities for marketing by engaging customers in various stages of the product development cycle. It allows the marketers to collaborate with consumers on various aspects such as product development, service enhancement, and promotion. Collaboration with the business partners and consumers can be improved by the companies by utilizing the tools provided by the Web 2.0 paradigm. Consumer-oriented companies use networks such as Twitter, Yelp, and Facebook as common elements of multichannel promotion of their products. Social networks have become more intuitive and user friendly and can be utilized to disseminate the product information so as to reach the maximum number of prospective product consumers in an efficient manner.

6.7.3.3 Education

Web 2.0 technologies can help the education scenario by providing students and faculty with more opportunities to interact and collaborate with their peers. Effective *knowledge discovery* is possible with the features offered by the Web 2.0 such as greater customization and choice of topics, and less distraction from their peers. By utilizing the tools of Web 2.0, the students get the opportunity to share what they learn with other peers by collaborating with them.

6.7.4 Web 2.0 and Cloud Computing

In Web 2.0, the metadata describing the web content is written in languages such as XML, which can be read and processed by the computers automatically. Various XML-based web protocols such as SOAP, WSDL, and UDDI help to integrate applications developed using different programming languages utilizing heterogeneous computing platforms and OSs. Relying on this capability of data integration and data exchange across heterogeneous applications, new business models of application development, deployment, and delivery over the Internet have been conceptualized and implemented. That means the applications can be hosted on the web and accessed by geographically separated clients over the Internet. Web services are such interoperable applications or services hosted on the web for remote use by multiple clients with heterogeneous platforms, and they can even be discovered dynamically on the fly with no prior knowledge of their existence.

In the business model of cloud computing, the application development infrastructures such as processors, storage, memory, OS, and application development tools and software can be accessed by the clients as services over the Internet in a pay-per-use model. In this model of service delivery, a huge pool of physical resources hosted on the web by the service providers will be shared by multiple clients as and when required. Cloud computing is based on the SOA of Web 2.0 and virtualization [16,18] of hardware and software resources stored hosted by the service providers. Hence, cloud computing is considered as the future of Internet computing because of the advantages offered by this business model such as no capital expenditure, speed of application deployment, shorter time to market, lower cost of operation, and easier maintenance of resources for the clients.

The success of online social networks and other Web 2.0 functionalities encouraged many SaaS applications to offer features that let its users work together, and distribute and share data and information. Cloud computing is a platform that helps individuals and enterprises to access hardware, software, and data resources using the Internet for most of their computing needs [19].

6.8 Web 3.0

The name *Web 3.0* was given by John Markoff of *The New York Times* to this third-generation of the web. The first two generations of the web were called Web 1.0 and Web 2.0 [20]. The three technologies could be briefly described as follows:

Web 1.0: Web 1.0 was the first generation of the Web in which the main focus was building the web, making it accessible, and also commercializing it. The key areas of interest in Web 1.0 included protocols such as HTTP, open standard markup languages such as HTML and XML, Internet access through ISPs, the first web browsers, platforms and tools for web development, web-development software languages such as Java and Javascript, and the commercialization of the web.

Web 2.0: The phrase Web 2.0 was coined by O'Reilly and it refers to the second generation of Internet-based services, such as social networking sites, wikis, and communication tools, that facilitate online collaboration and sharing among various users.

Web 3.0: John Markoff of *The New York Times* coined the term Web 3.0 and it refers to the third generation of Internet-based services that is collectively called *the intelligent web*. Web 3.0 includes services on the Internet that use technologies such as semantic web, natural language search, machine learning, recommendation agents, and artificial intelligence to achieve machine-facilitated understanding of information in order to provide a more productive and intuitive experience to the web users.

Web 2.0 technology allows the use of read/write web, blogs, interactive web applications, rich media, tagging or folksonomy while sharing content, and also social networking sites focusing on communities [21]. At the same time, the Web 3.0 standard uses semantic web technology, drag and drop mash-ups, widgets, user behavior, user engagement, and consolidation of dynamic web contents depending on the interest of the individual users. Web 3.0 uses the *Data Web* technology, which features the data records that are publishable and reusable on the web through query-able formats such as Resource Description Framework (RDF), XML, and microformats. It is an important component facilitating the semantic web, which enables new levels of application interoperability and data integration among various application and services, and also makes data dynamically linkable and accessible in the form of web pages. The complete semantic web stage expands the scope of both structured and unstructured contents through the use of Web Ontology Language (OWL) and RDF semantics.

The Web 3.0 standard also incorporates the latest researches in the field of artificial intelligence. The wide usage of the technology is promptly visible in the case of an application that makes hit-song predictions based on user feedback from music websites hosted by various colleges on the Internet. Web 3.0 achieves the intelligence in an organic fashion through the interaction of the web users. Thus, Web 3.0 makes it possible for an application to think on its own with the data available to make certain decisions, and it also allows to connect one application to another dynamically depending on the context of usage. An example of a typical Web 3.0 application is the one that uses content management systems along with artificial intelligence. These systems are capable of answering the questions posed by the users, because the application is able to think on its own and find the most probable answer, depending on the context, to the query submitted by the user. In this way, Web 3.0 can also be described as a *machine to user* standard in the Internet.

6.8.1 Components of Web 3.0

The term Web 3.0, also known as the *semantic web*, describes sites wherein the computers will be generating raw data on their own without direct user interaction. Web 3.0 is considered as the next logical step in the evolution of the Internet and web technologies. For Web 1.0 and Web 2.0, the Internet is confined within the physical walls of the computer, but as more and more devices such as smartphones, cars, and other household appliances become connected to the web, the Internet will be omnipresent and could be utilized in the most efficient manner. In this case, various devices will be able to exchange data among one another and they will even generate new information from raw data (e.g., a music site, Last.fm, will be able to anticipate the type of music the user likes depending on his previous song selections). Hence, the Internet will be able to perform the user tasks in a faster and more efficient way, such as the case of search engines being able to search for the actual interests of the individual users and not just based on the keyword typed into the search engines.

Web 3.0 embeds intelligence in the entire web domain. It deploys web robots that are smart enough of taking decisions in the absence of any user interference. If Web 2.0 can be called a *read/write* web, Web 3.0 will surely be called a *read/write/execute* web. The two major components forming the basis of Web 3.0 are the following:

1. Semantic web
2. Web services

6.8.1.1 Semantic Web

The semantic web provides the web user a common framework that could be used to share and reuse the data across various applications, enterprises, and community boundaries [22]. The semantic web is a vision of IT that allows the data and information to be readily interpreted by machines, so that the machines are able to take contextual decisions on their own by finding, combining, and acting upon relevant information on the web. The semantic web, as originally envisioned, is a system enabling the machines to *understand* the context and meaning of complex human requests and respond to them appropriately. Also, the semantic web is considered as an integrator of contents or information across different applications and systems.

Web 1.0 represents the first implementation of the web, which, according to Berners-Lee, could be considered as the *read-only web*. This means that the early web allowed the users to search for the required information and read from it. There was very little user interaction or content contribution in this case. The website owners achieved their goal of establishing the online presence and making their information available to all at any time [23].

One of the biggest challenges of presenting information on the web is that web applications are not able to associate the context information to the data, and as a result, they cannot really understand what is relevant and what is not. Through the use of some sort of semantic markup, or data interchange formats, data could be represented in a form that is not only accessible to humans via natural language but also able to be understood and interpreted by software applications. Formatting the data to be understood by software agents is emphasized by the *execute* portion of the *read/write/execute* definition of Web 3.0.

6.8.1.2 Web Services

A web service is a software system that supports computer-to-computer interaction over the Internet. Web services are usually represented as APIs. For example, the popular photography-sharing website Flickr provides a web service that could be utilized by the developers to programmatically interface with Flickr in order to search for images. Currently, thousands of web services are available for users, and they form an important component in the context of Web 3.0. By the combination of semantic markup and web services, the Web 3.0 paradigm promises the potential for applications that can communicate to each other directly and also facilitates broader searches for information through simpler interfaces.

6.8.2 Characteristics of Web 3.0

Web 3.0 could be considered as the third generation of the web and it is enabled by the convergence of several key emerging technology trends as discussed earlier. The major characteristics of this paradigm are the following [24]:

- Ubiquitous connectivity
- Network computing
- Open technologies
- Open identity
- The intelligent web

The Web 3.0 technology enables the continuous connectivity of the user requesting services with various services available through the usage of mobile devices and mobile Internet access with broadband adoption. The Web 3.0 technology enables the Software-as-a-Service (SaaS) business model in cloud computing. Web 3.0 also makes the various web services interoperable by providing open standards. Web 3.0 helps in the creation and usage of open APIs and protocols for service composition and communication. Open data formats and open-source software platforms are supported for the development and use of various applications and services. Web 3.0 also enables the use of open-identity protocols such as OpenID, which helps to port the user account from one service to another effectively.

Web 3.0 supports semantic web technologies (RDF, OWL, SWRL, SPARQL, semantic application platforms, and statement-based datastores such as triplestores, tuplestores, and associative databases) and distributed databases (wide-area distributed database interoperability enabled by semantic web technologies). Also, intelligent applications (using the concepts of natural language processing, machine learning, machine reasoning, and autonomous agents) are created, and their effective communication is made possible. Hence, Web 3.0 helps to achieve a more connected, open, and intelligent web utilizing the aforementioned technologies.

6.8.3 Convergence of Cloud and Web 3.0

The concepts of Web 2.0 and Web 3.0 could be utilized to implement the web as a platform, which forms the basis for the effective delivery and utilization of cloud services [25]. The evolution of Web 3.0 with its enriched capabilities such as personalization, data portability, and user-centric identity provided enough opportunities for enterprises and individuals to generate and use the web contents in a more effective manner, which in turn increased the performance of cloud computing services. Thus, the Web 3.0 paradigm has changed the mode of computing by enabling the service consumers to tap into software and services located in the DCs of service providers rather than confining to the services available on a user's PC. Hence, with the Web 3.0 features, the users could store their data and applications in *cloud* and subscribe to the cloud services as and when needed. Also, the cloud services could be accessed through various devices other than the PC, and the information or the services are made available to various users while moving from one device to another.

With the introduction of Web 2.0, the Internet has been utilized much more than making an online purchase. It has become a gathering place where social communities are formed, personal information is shared, and people dialogue and reconnect with old acquaintances. The launch of social networking sites such as Twitter and Facebook has been a huge success among end users. Because of their ever-growing customer base, these highly interactive social networking sites are used as popular e-marketing tools by business vendors to reach consumers and promote their products through inexpensive and creative ways.

6.8.4 Case Studies in Cloud and Web 3.0

Web 3.0 represents the next generation of web technology that helps achieve an unprecedented level of intelligence and interaction in computing systems and applications. Smarter computing capabilities will be introduced into the web applications in order to perform complex tasks that previously required human interaction for the purposes of understanding and reasoning. With this technology, the various computing tasks could be completed at an enormous scale and efficiency. Today, service providers such as Facebook use these technologies in effective ways. They introduce a variety of innovative capabilities including faster access to the information stored in their DC utilizing the highly intelligent decision-making applications. Some examples are given in the following subsections [26].

6.8.4.1 Connecting Information: Facebook

The Facebook Open Graph is a great example for the scalability feature offered by Web 3.0. Open Graph includes a format for marking up web pages based on the RDF and the semantic web data model so that anybody using a website can incorporate Facebook's markup to define what that site is all about. This goes beyond basic metadata processing because the descriptions enable the web users to find and connect with their friends who share similar interests. The *Like* button provided by Facebook could be considered as a simple manifestation of all these because a single click can offer the analysts an invaluable amount of information that could later be used for further communication with friends and also to make recommendations and discoveries.

6.8.4.2 Search Optimization and Web Commerce: Best Buy

One of the major benefits of Web 3.0 is more relevant search results. In the case of Web commerce, this means that additional information can be incorporated into product descriptions and online ads in order to make them easier for search engines to find. Best Buy is a frontrunner in using this technology to leverage its e-commerce efforts. It is using RDFa (RDF in attributes, which adds a set of XHTML attributes) markup and the GoodRelations vocabulary so that more targeted search results are produced for shoppers looking for various products. So far, some internal company audit measures suggest that this approach has increased the consumer traffic by 30%.

6.8.4.3 Understanding Text: Millward Brown

Web 3.0 is ideally suited for the management and analysis of various documents and information because it enables computing system to quickly process large amounts of text and extract the meaning from them. *Sentiment analysis* could be considered as a good example for this, which involves the measurement of how various customers feel about an organization, as expressed through surveys, blogs, online forums, and social networks. The global research agency Millward Brown, which works with Fortune 500 companies for developing their branding strategies, uses Web 3.0 technologies from OpenAmplify to identify the strategically meaningful information extracted from customer feedback. This information could then be used to drive marketing messages and also to improve public relations efforts, pricing strategies, and service responses of the companies.

6.9 Software Process Models for Cloud

The success or quality of a software project is measured by whether it is developed within time and budget and by its efficiency, usability, dependability, and maintainability [27,28]. The whole development process of software from its conceptualization to operation and retirement is called the software development life cycle (SDLC). SDLC goes through several framework activities like requirements gathering, planning, design, coding, testing, deployment, maintenance, and retirement. These activities are synchronized in accordance to the process model adopted for a particular software development.

There are many process models to choose from, depending on the size of the project, delivery time requirement, and type of the project. For example, the process model selected for the development of an avionic embedded system will be different from the one selected for the development of a web application.

6.9.1 Types of Software Models

There are various software development models or methodologies such as waterfall, V, incremental, RAD, agile, iterative, and spiral. They are discussed in the following.

6.9.1.1 Waterfall Model

This is the most common life cycle model and is also referred to as a linear-sequential life cycle model. In a waterfall model, each phase must be completed in its entirety before the next phase begins. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue the project.

6.9.1.2 V Model

V model means verification and validation model. Just like the waterfall model, the V model is a sequential path of execution of processes. Each phase must be completed before the next phase begins. Testing of the product is planned in parallel with a corresponding phase of development.

6.9.1.3 Incremental Model

The incremental model is an intuitive approach to the waterfall model. Multiple development cycles take place here, making the life cycle a *multi-waterfall* cycle. The cycles are divided into smaller, more easily managed iterations. The iterations pass through the requirements, design, implementation, and testing phases, and during the first iteration, a working version of the software is generated.

6.9.1.4 RAD Model

The rapid application development (RAD) model is a type of incremental model. In the RAD model, the functions or components are generated in parallel, and these generated outcomes are timeboxed, delivered, and then combined to a working prototype. This can quickly give the customer something to operate and to give feedback about the requirements.

6.9.1.5 Agile Model

The agile model is also an incremental model where the software is developed in rapid, incremental cycles. The development results in tiny incremental releases and is based on previously built functionality and is carefully tested to ensure software quality. In time-critical applications, this model is preferred more. Extreme programming (XP) is one popular example of this developmental life cycle model.

6.9.1.6 Iterative Model

It does not start with a full specification of requirements, but it begins by specifying and implementing just a part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model.

6.9.1.7 Spiral Model

The spiral model is similar to the incremental model, with more emphasis on the risk analysis. The spiral model has four phases: planning, risk analysis, engineering, and evaluation. In this model, a software project repeatedly passes through these phases in iterations called spirals. The baseline spiral starts at the planning phase, requirements are gathered, and risk is assessed. Each subsequent spiral builds on the baseline spiral.

6.9.2 Agile SDLC for Cloud Computing

In the rapidly changing computing environment in web services and cloud platforms, software development is going to be very challenging [29]. Software development process will involve heterogeneous platforms, distributed web services, and multiple enterprises geographically dispersed all over the world. Existing software process models and framework activities are not adequate unless interaction with cloud providers is included. Requirements gathering phase so far included customers, users, and software engineers. Now, it has to include the cloud providers as well, as they will be providing the computing infrastructure and its maintenance. As only the cloud providers will know the size, architectural details, virtualization strategy, and resource utilization of the infrastructure, they should also be included in the planning and design phases of software development. Coding and testing can be done on the cloud platform, which is a huge benefit as everybody will have easy access to the software being built. This will reduce the cost and time for testing and validation.

In the cloud environment, software developers can use the web services and open-source software freely available from the cloud instead of procuring them. Software developers build software from readily available components rather than writing it all and building a monolithic application. Refactoring of existing application is required to best utilize the cloud infrastructure architecture in a cost-effective way. In the latest hardware technology, the computers are multicore and networked and the software engineers should train themselves in parallel and distributed computing to complement these advances of hardware and network technology. Cloud providers will insist that software should be as modular as possible for occasional migration from one server to another for LB as required by the cloud provider [30].

SDLC is a framework that defines tasks to be performed at each step in the software development process [31]. Cloud computing provides an almost instant access to the software and development environments, by providing multitenancy of the virtualized servers and other IT infrastructures. Specifically, Platform as a Service (PaaS), the development platform environment in the cloud, encourages the use of agile methodologies. Agile and PaaS together add great value to the SDLC processes. They help in reducing costs for enterprises in the long run and help in increasing developer productivity at the same time.

6.9.2.1 Features of Cloud SDLC

SDLC for cloud computing is different from the traditional SDLC in the following ways:

1. *Inclination toward agile methodologies*: Cloud SDLC can utilize methodologies such as agile SDLC. These are designed for iterative approach to development and fast deployment life cycles.
2. *Customizable SDLC framework for different stages*: Cloud computing SDLC must have the capabilities to be customized according to the requirements of the project. In other words, the elasticity and robustness of cloud computing environment can be best utilized if the SDLCs for cloud are customizable.
3. *Installation and configuration guidelines*: SDLC for cloud must provide implementation approach and guidelines for installation and configuration of the cloud depending on its size. The guidelines must ensure that installation and configuration of infrastructure and application environment are completed appropriately for different stages of SDLC including operations and maintenance. These guidelines are the key to differentiating SDLC for cloud from traditional SDLC.

6.9.3 Agile Software Development Process

More than 50% of software projects fail due to various reasons like schedule and budget slippage, non-user-friendly interface of the software, and nonflexibility for maintenance and change of the software [29]. The reason for all these problems is lack of communication and coordination between all the parties involved. Requirement changes of a software are the major cause of increased complexity, schedule, and budget slippage. Incorporating changes at a later stage of SDLC increases cost of the project exponentially. Adding more programmers at a later stage does not solve the schedule problem as increased coordination requirement slows down the project further. It is very important that requirements gathering, planning, and design of the software are done involving all the concerned parties from the beginning of the project.

That is why several agile process models like XP, Scrum, Crystal, and Adaptive have been introduced in the mid-1990s to accommodate continuous changes in requirements during the development of the software. These agile process models have shorter development cycles where small pieces of work are *timeboxed*, developed, and released for customer feedback, verification, and validation iteratively. One timebox takes a few weeks up to a month. The agile process model is communication intensive. It eliminates the exponential increase in cost to incorporate changes as in the waterfall model, by keeping the customer involved throughout the development process and