J Internet Serv Appl (2010) 1: 7–18
DOI 10.1007/s13174-010-0007-6

ORIGINAL PAPERS

# Cloud computing: state-of-the-art and research challenges

**Qi Zhang · Lu Cheng · Raouf Boutaba**

Received: 8 January 2010 / Accepted: 25 February 2010 / Published online: 20 April 2010

**Abstract** Cloud computing has recently emerged as a new paradigm for hosting and delivering services over the Internet. Cloud computing is attractive to business owners as it eliminates the requirement for users to plan ahead for provisioning, and allows enterprises to start from the small and increase resources only when there is a rise in service demand. However, despite the fact that cloud computing offers huge opportunities to the IT industry, the development of cloud computing technology is currently at its infancy, with many issues still to be addressed. In this paper, we present a survey of cloud computing, highlighting its key concepts, architectural principles, state-of-the-art implementation as well as research challenges. The aim of this paper is to provide a better understanding of the design challenges of cloud computing and identify important research directions in this increasingly important area.

**Keywords** Cloud computing · Data centers · Virtualization

## 1 Introduction

With the rapid development of processing and storage technologies and the success of the Internet, computing resources have become cheaper, more powerful and more ubiquitously available than ever before. This technological trend has enabled the realization of a new computing model called cloud computing, in which resources (e.g., CPU and storage) are provided as general utilities that can be leased and released by users through the Internet in an on-demand fashion. In a cloud computing environment, the traditional role of service provider is divided into two: the *infrastructure providers* who manage cloud platforms and lease resources according to a usage-based pricing model, and *service providers*, who rent resources from one or many infrastructure providers to serve the end users. The emergence of cloud computing has made a tremendous impact on the Information Technology (IT) industry over the past few years, where large companies such as Google, Amazon and Microsoft strive to provide more powerful, reliable and cost-efficient cloud platforms, and business enterprises seek to reshape their business models to gain benefit from this new paradigm. Indeed, cloud computing provides several compelling features that make it attractive to business owners, as shown below.

*No up-front investment*: Cloud computing uses a pay-as-you-go pricing model. A service provider does not need to invest in the infrastructure to start gaining benefit from cloud computing. It simply rents resources from the cloud according to its own needs and pay for the usage.

*Lowering operating cost*: Resources in a cloud environment can be rapidly allocated and de-allocated on demand. Hence, a service provider no longer needs to provision capacities according to the peak load. This provides huge savings since resources can be released to save on operating costs when service demand is low.

*Highly scalable*: Infrastructure providers pool large amount of resources from data centers and make them easily accessible. A service provider can easily expand its service to large scales in order to handle rapid increase in service demands (e.g., flash-crowd effect). This model is sometimes called surge computing [5].

Q. Zhang · L. Cheng · R. Boutaba (✉)
University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1
e-mail: rboutaba@cs.uwaterloo.ca

Q. Zhang
e-mail: q8zhang@cs.uwaterloo.ca

L. Cheng
e-mail: l32cheng@cs.uwaterloo.ca

*Easy access*: Services hosted in the cloud are generally web-based. Therefore, they are easily accessible through a variety of devices with Internet connections. These devices not only include desktop and laptop computers, but also cell phones and PDAs.

*Reducing business risks and maintenance expenses*: By outsourcing the service infrastructure to the clouds, a service provider shifts its business risks (such as hardware failures) to infrastructure providers, who often have better expertise and are better equipped for managing these risks. In addition, a service provider can cut down the hardware maintenance and the staff training costs.

However, although cloud computing has shown considerable opportunities to the IT industry, it also brings many unique challenges that need to be carefully addressed. In this paper, we present a survey of cloud computing, highlighting its key concepts, architectural principles, state-of-the-art implementations as well as research challenges. Our aim is to provide a better understanding of the design challenges of cloud computing and identify important research directions in this fascinating topic.

The remainder of this paper is organized as follows. In Sect. 2 we provide an overview of cloud computing and compare it with other related technologies. In Sect. 3, we describe the architecture of cloud computing and present its design principles. The key features and characteristics of cloud computing are detailed in Sect. 4. Section 5 surveys the commercial products as well as the current technologies used for cloud computing. In Sect. 6, we summarize the current research topics in cloud computing. Finally, the paper concludes in Sect. 7.

## 2 Overview of cloud computing

This section presents a general overview of cloud computing, including its definition and a comparison with related concepts.

### 2.1 Definitions

The main idea behind cloud computing is not a new one. John McCarthy in the 1960s already envisioned that computing facilities will be provided to the general public like a utility [39]. The term “cloud” has also been used in various contexts such as describing large ATM networks in the 1990s. However, it was after Google’s CEO Eric Schmidt used the word to describe the business model of providing services across the Internet in 2006, that the term really started to gain popularity. Since then, the term cloud computing has been used mainly as a marketing term in a variety of contexts to represent many different ideas. Certainly, the lack of a standard definition of cloud computing has generated not only market hypes, but also a fair amount of skepticism and confusion. For this reason, recently there has been work on standardizing the definition of cloud computing. As an example, the work in [49] compared over 20 different definitions from a variety of sources to confirm a standard definition. In this paper, we adopt the definition of cloud computing provided by The National Institute of Standards and Technology (NIST) [36], as it covers, in our opinion, all the essential aspects of cloud computing:

**NIST definition of cloud computing** *Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*

The main reason for the existence of different perceptions of cloud computing is that cloud computing, unlike other technical terms, is not a new technology, but rather a new operations model that brings together a set of existing technologies to run business in a different way. Indeed, most of the technologies used by cloud computing, such as virtualization and utility-based pricing, are not new. Instead, cloud computing leverages these existing technologies to meet the technological and economic requirements of today’s demand for information technology.

### 2.2 Related technologies

Cloud computing is often compared to the following technologies, each of which shares certain aspects with cloud computing:

*Grid Computing*: Grid computing is a distributed computing paradigm that coordinates networked resources to achieve a common computational objective. The development of Grid computing was originally driven by scientific applications which are usually computation-intensive. Cloud computing is similar to Grid computing in that it also employs distributed resources to achieve application-level objectives. However, cloud computing takes one step further by leveraging virtualization technologies at multiple levels (hardware and application platform) to realize resource sharing and dynamic resource provisioning.

*Utility Computing*: Utility computing represents the model of providing resources on-demand and charging customers based on usage rather than a flat rate. Cloud computing can be perceived as a realization of utility computing. It adopts a utility-based pricing scheme entirely for economic reasons. With on-demand resource provisioning and utility-based pricing, service providers can truly maximize resource utilization and minimize their operating costs.

*Virtualization*: Virtualization is a technology that abstracts away the details of physical hardware and provides virtualized resources for high-level applications. A virtualized server is commonly called a virtual machine (VM). Virtualization forms the foundation of cloud computing, as it provides the capability of pooling computing resources from clusters of servers and dynamically assigning or reassigning virtual resources to applications on-demand.

*Autonomic Computing*: Originally coined by IBM in 2001, autonomic computing aims at building computing systems capable of self-management, i.e. reacting to internal and external observations without human intervention. The goal of autonomic computing is to overcome the management complexity of today's computer systems. Although cloud computing exhibits certain autonomic features such as automatic resource provisioning, its objective is to lower the resource cost rather than to reduce system complexity.

In summary, cloud computing leverages virtualization technology to achieve the goal of providing computing resources as a utility. It shares certain aspects with grid computing and autonomic computing but differs from them in other aspects. Therefore, it offers unique benefits and imposes distinctive challenges to meet its requirements.

## 3 Cloud computing architecture

This section describes the architectural, business and various operation models of cloud computing.

### 3.1 A layered model of cloud computing

Generally speaking, the architecture of a cloud computing environment can be divided into 4 layers: the hardware/datacenter layer, the infrastructure layer, the platform layer and the application layer, as shown in Fig. 1. We describe each of them in detail:

*The hardware layer*: This layer is responsible for managing the physical resources of the cloud, including physical servers, routers, switches, power and cooling systems. In practice, the hardware layer is typically implemented in data centers. A data center usually contains thousands of servers that are organized in racks and interconnected through switches, routers or other fabrics. Typical issues at hardware layer include hardware configuration, fault-tolerance, traffic management, power and cooling resource management.

*The infrastructure layer*: Also known as the virtualization layer, the infrastructure layer creates a pool of storage and computing resources by partitioning the physical resources using virtualization technologies such as Xen [55], KVM [30] and VMware [52]. The infrastructure layer is an essential component of cloud computing, since many key features, such as dynamic resource assignment, are only made available through virtualization technologies.

*The platform layer*: Built on top of the infrastructure layer, the platform layer consists of operating systems and application frameworks. The purpose of the platform layer is to minimize the burden of deploying applications directly into VM containers. For example, Google App Engine operates at the platform layer to provide API support for implementing storage, database and business logic of typical web applications.

*The application layer*: At the highest level of the hierarchy, the application layer consists of the actual cloud applications. Different from traditional applications, cloud applications can leverage the automatic-scaling feature to achieve better performance, availability and lower operating cost.

Compared to traditional service hosting environments such as dedicated server farms, the architecture of cloud computing is more modular. Each layer is loosely coupled with the layers above and below, allowing each layer to evolve separately. This is similar to the design of the OSI

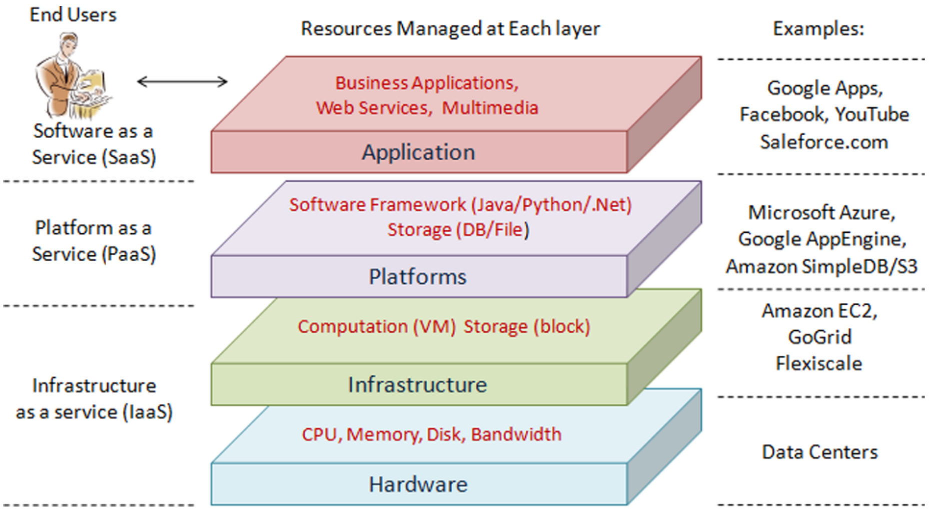
End Users
Software as a Service (SaaS)
Platform as a Service (PaaS)
Infrastructure as a service (IaaS)
Resources Managed at Each layer
Business Applications, Web Services, Multimedia
Application
Software Framework (Java/Python/.Net) Storage (DB/File)
Platforms
Computation (VM) Storage (block)
Infrastructure
CPU, Memory, Disk, Bandwidth
Hardware
Examples:
Google Apps, Facebook, YouTube Saleforce.com
Microsoft Azure, Google AppEngine, Amazon SimpleDB/S3
Amazon EC2, GoGrid Flexiscale
Data Centers

**Fig. 1** Cloud computing architecture

model for network protocols. The architectural modularity allows cloud computing to support a wide range of application requirements while reducing management and maintenance overhead.

### 3.2 Business model

Cloud computing employs a service-driven business model. In other words, hardware and platform-level resources are provided as services on an on-demand basis. Conceptually, every layer of the architecture described in the previous section can be implemented as a service to the layer above. Conversely, every layer can be perceived as a customer of the layer below. However, in practice, clouds offer services that can be grouped into three categories: software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS).

1. *Infrastructure as a Service*: IaaS refers to on-demand provisioning of infrastructural resources, usually in terms of VMs. The cloud owner who offers IaaS is called an IaaS provider. Examples of IaaS providers include Amazon EC2 [2], GoGrid [15] and Flexiscale [18].
2. *Platform as a Service*: PaaS refers to providing platform layer resources, including operating system support and software development frameworks. Examples of PaaS providers include Google App Engine [20], Microsoft Windows Azure [53] and Force.com [41].
3. *Software as a Service*: SaaS refers to providing on-demand applications over the Internet. Examples of SaaS providers include Salesforce.com [41], Rackspace [17] and SAP Business ByDesign [44].

The business model of cloud computing is depicted by Fig. 2. According to the layered architecture of cloud computing, it is entirely possible that a PaaS provider runs its cloud on top of an IaaS provider's cloud. However, in the current practice, IaaS and PaaS providers are often parts of the same organization (e.g., Google and Salesforce). This is why PaaS and IaaS providers are often called the *infrastructure providers* or *cloud providers* [5].

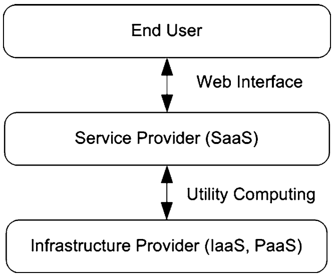

**Fig. 2** Business model of cloud computing

### 3.3 Types of clouds

There are many issues to consider when moving an enterprise application to the cloud environment. For example, some service providers are mostly interested in lowering operation cost, while others may prefer high reliability and security. Accordingly, there are different types of clouds, each with its own benefits and drawbacks:

*Public clouds*: A cloud in which service providers offer their resources as services to the general public. Public clouds offer several key benefits to service providers, including no initial capital investment on infrastructure and shifting of risks to infrastructure providers. However, public clouds lack fine-grained control over data, network and security settings, which hampers their effectiveness in many business scenarios.

*Private clouds*: Also known as internal clouds, private clouds are designed for exclusive use by a single organization. A private cloud may be built and managed by the organization or by external providers. A private cloud offers the highest degree of control over performance, reliability and security. However, they are often criticized for being similar to traditional proprietary server farms and do not provide benefits such as no up-front capital costs.

*Hybrid clouds*: A hybrid cloud is a combination of public and private cloud models that tries to address the limitations of each approach. In a hybrid cloud, part of the service infrastructure runs in private clouds while the remaining part runs in public clouds. Hybrid clouds offer more flexibility than both public and private clouds. Specifically, they provide tighter control and security over application data compared to public clouds, while still facilitating on-demand service expansion and contraction. On the down side, designing a hybrid cloud requires carefully determining the best split between public and private cloud components.

*Virtual Private Cloud*: An alternative solution to addressing the limitations of both public and private clouds is called Virtual Private Cloud (VPC). A VPC is essentially a platform running on top of public clouds. The main difference is that a VPC leverages virtual private network (VPN) technology that allows service providers to design their own topology and security settings such as firewall rules. VPC is essentially a more holistic design since it not only virtualizes servers and applications, but also the underlying communication network as well. Additionally, for most companies, VPC provides seamless transition from a proprietary service infrastructure to a cloud-based infrastructure, owing to the virtualized network layer.

For most service providers, selecting the right cloud model is dependent on the business scenario. For example, computation-intensive scientific applications are best deployed on public clouds for cost-effectiveness. Arguably, certain types of clouds will be more popular than others.

In particular, it was predicted that hybrid clouds will be the dominant type for most organizations [14]. However, virtual private clouds have started to gain more popularity since their inception in 2009.

## 4 Cloud computing characteristics

Cloud computing provides several salient features that are different from traditional service computing, which we summarize below:

*Multi-tenancy*: In a cloud environment, services owned by multiple providers are co-located in a single data center. The performance and management issues of these services are shared among service providers and the infrastructure provider. The layered architecture of cloud computing provides a natural division of responsibilities: the owner of each layer only needs to focus on the specific objectives associated with this layer. However, multi-tenancy also introduces difficulties in understanding and managing the interactions among various stakeholders.

*Shared resource pooling*: The infrastructure provider offers a pool of computing resources that can be dynamically assigned to multiple resource consumers. Such dynamic resource assignment capability provides much flexibility to infrastructure providers for managing their own resource usage and operating costs. For instance, an IaaS provider can leverage VM migration technology to attain a high degree of server consolidation, hence maximizing resource utilization while minimizing cost such as power consumption and cooling.

*Geo-distribution and ubiquitous network access*: Clouds are generally accessible through the Internet and use the Internet as a service delivery network. Hence any device with Internet connectivity, be it a mobile phone, a PDA or a laptop, is able to access cloud services. Additionally, to achieve high network performance and localization, many of today's clouds consist of data centers located at many locations around the globe. A service provider can easily leverage geo-diversity to achieve maximum service utility.

*Service oriented*: As mentioned previously, cloud computing adopts a service-driven operating model. Hence it places a strong emphasis on service management. In a cloud, each IaaS, PaaS and SaaS provider offers its service according to the Service Level Agreement (SLA) negotiated with its customers. SLA assurance is therefore a critical objective of every provider.

*Dynamic resource provisioning*: One of the key features of cloud computing is that computing resources can be obtained and released on the fly. Compared to the traditional model that provisions resources according to peak demand, dynamic resource provisioning allows service providers to acquire resources based on the current demand, which can considerably lower the operating cost.

*Self-organizing*: Since resources can be allocated or de-allocated on-demand, service providers are empowered to manage their resource consumption according to their own needs. Furthermore, the automated resource management feature yields high agility that enables service providers to respond quickly to rapid changes in service demand such as the flash crowd effect.

*Utility-based pricing*: Cloud computing employs a pay-per-use pricing model. The exact pricing scheme may vary from service to service. For example, a SaaS provider may rent a virtual machine from an IaaS provider on a per-hour basis. On the other hand, a SaaS provider that provides on-demand customer relationship management (CRM) may charge its customers based on the number of clients it serves (e.g., Salesforce). Utility-based pricing lowers service operating cost as it charges customers on a per-use basis. However, it also introduces complexities in controlling the operating cost. In this perspective, companies like VKernel [51] provide software to help cloud customers understand, analyze and cut down the unnecessary cost on resource consumption.

## 5 State-of-the-art

In this section, we present the state-of-the-art implementations of cloud computing. We first describe the key technologies currently used for cloud computing. Then, we survey the popular cloud computing products.

### 5.1 Cloud computing technologies

This section provides a review of technologies used in cloud computing environments.

### *5.1.1 Architectural design of data centers*

A data center, which is home to the computation power and storage, is central to cloud computing and contains thousands of devices like servers, switches and routers. Proper planning of this network architecture is critical, as it will heavily influence applications performance and throughput in such a distributed computing environment. Further, scalability and resiliency features need to be carefully considered.

Currently, a layered approach is the basic foundation of the network architecture design, which has been tested in some of the largest deployed data centers. The basic layers of a data center consist of the core, aggregation, and access layers, as shown in Fig. 3. The access layer is where the servers in racks physically connect to the network. There are typically 20 to 40 servers per rack, each connected to an access switch with a 1 Gbps link. Access switches usually connect to two aggregation switches for redundancy with

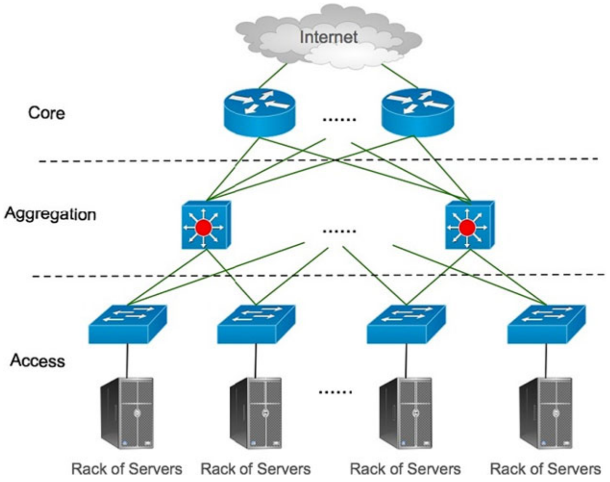

**Fig. 3** Basic layered design of data center network infrastructure

10 Gbps links. The aggregation layer usually provides important functions, such as domain service, location service, server load balancing, and more. The core layer provides connectivity to multiple aggregation switches and provides a resilient routed fabric with no single point of failure. The core routers manage traffic into and out of the data center.

A popular practice is to leverage commodity Ethernet switches and routers to build the network infrastructure. In different business solutions, the layered network infrastructure can be elaborated to meet specific business challenges. Basically, the design of a data center network architecture should meet the following objectives [1, 21–23, 35]:

*Uniform high capacity*: The maximum rate of a server-to-server traffic flow should be limited only by the available capacity on the network-interface cards of the sending and receiving servers, and assigning servers to a service should be independent of the network topology. It should be possible for an arbitrary host in the data center to communicate with any other host in the network at the full bandwidth of its local network interface.

*Free VM migration*: Virtualization allows the entire VM state to be transmitted across the network to migrate a VM from one physical machine to another. A cloud computing hosting service may migrate VMs for statistical multiplexing or dynamically changing communication patterns to achieve high bandwidth for tightly coupled hosts or to achieve variable heat distribution and power availability in the data center. The communication topology should be designed so as to support rapid virtual machine migration.

*Resiliency*: Failures will be common at scale. The network infrastructure must be fault-tolerant against various types of server failures, link outages, or server-rack failures. Existing unicast and multicast communications should not be affected to the extent allowed by the underlying physical connectivity.

*Scalability*: The network infrastructure must be able to scale to a large number of servers and allow for incremental expansion.

*Backward compatibility*: The network infrastructure should be backward compatible with switches and routers running Ethernet and IP. Because existing data centers have commonly leveraged commodity Ethernet and IP based devices, they should also be used in the new architecture without major modifications.

Another area of rapid innovation in the industry is the design and deployment of shipping-container based, modular data center (MDC). In an MDC, normally up to a few thousands of servers, are interconnected via switches to form the network infrastructure. Highly interactive applications, which are sensitive to response time, are suitable for geo-diverse MDC placed close to major population areas. The MDC also helps with redundancy because not all areas are likely to lose power, experience an earthquake, or suffer riots at the same time. Rather than the three-layered approach discussed above, Guo et al. [22, 23] proposed server-centric, recursively defined network structures of MDC.

### *5.1.2 Distributed file system over clouds*

Google File System (GFS) [19] is a proprietary distributed file system developed by Google and specially designed to provide efficient, reliable access to data using large clusters of commodity servers. Files are divided into chunks of 64 megabytes, and are usually appended to or read and only extremely rarely overwritten or shrunk. Compared with traditional file systems, GFS is designed and optimized to run on data centers to provide extremely high data throughputs, low latency and survive individual server failures.

Inspired by GFS, the open source Hadoop Distributed File System (HDFS) [24] stores large files across multiple machines. It achieves reliability by replicating the data across multiple servers. Similarly to GFS, data is stored on multiple geo-diverse nodes. The file system is built from a cluster of data nodes, each of which serves blocks of data over the network using a block protocol specific to HDFS. Data is also provided over HTTP, allowing access to all content from a web browser or other types of clients. Data nodes can talk to each other to rebalance data distribution, to move copies around, and to keep the replication of data high.

### *5.1.3 Distributed application framework over clouds*

HTTP-based applications usually conform to some web application framework such as Java EE. In modern data center environments, clusters of servers are also used for computation and data-intensive jobs such as financial trend analysis, or film animation.

MapReduce [16] is a software framework introduced by Google to support distributed computing on large data sets on clusters of computers. MapReduce consists of one Master, to which client applications submit MapReduce jobs. The Master pushes work out to available task nodes in the data center, striving to keep the tasks as close to the data as possible. The Master knows which node contains the data, and which other hosts are nearby. If the task cannot be hosted on the node where the data is stored, priority is given to nodes in the same rack. In this way, network traffic on the main backbone is reduced, which also helps to improve throughput, as the backbone is usually the bottleneck. If a task fails or times out, it is rescheduled. If the Master fails, all ongoing tasks are lost. The Master records what it is up to in the filesystem. When it starts up, it looks for any such data, so that it can restart work from where it left off.

The open source Hadoop MapReduce project [25] is inspired by Google's work. Currently, many organizations are using Hadoop MapReduce to run large data-intensive computations.

## 5.2 Commercial products

In this section, we provide a survey of some of the dominant cloud computing products.

### *5.2.1 Amazon EC2*

Amazon Web Services (AWS) [3] is a set of cloud services, providing cloud-based computation, storage and other functionality that enable organizations and individuals to deploy applications and services on an on-demand basis and at commodity prices. Amazon Web Services' offerings are accessible over HTTP, using REST and SOAP protocols.

Amazon Elastic Compute Cloud (Amazon EC2) enables cloud users to launch and manage server instances in data centers using APIs or available tools and utilities. EC2 instances are virtual machines running on top of the Xen virtualization engine [55]. After creating and starting an instance, users can upload software and make changes to it. When changes are finished, they can be bundled as a new machine image. An identical copy can then be launched at any time. Users have nearly full control of the entire software stack on the EC2 instances that look like hardware to them. On the other hand, this feature makes it inherently difficult for Amazon to offer automatic scaling of resources.

EC2 provides the ability to place instances in multiple locations. EC2 locations are composed of Regions and Availability Zones. Regions consist of one or more Availability Zones, are geographically dispersed. Availability Zones are distinct locations that are engineered to be insulated from failures in other Availability Zones and provide inexpensive, low latency network connectivity to other Availability Zones in the same Region.

EC2 machine images are stored in and retrieved from Amazon Simple Storage Service (Amazon S3). S3 stores data as "objects" that are grouped in "buckets." Each object contains from 1 byte to 5 gigabytes of data. Object names are essentially URI [6] pathnames. Buckets must be explicitly created before they can be used. A bucket can be stored in one of several Regions. Users can choose a Region to optimize latency, minimize costs, or address regulatory requirements.

Amazon Virtual Private Cloud (VPC) is a secure and seamless bridge between a company's existing IT infrastructure and the AWS cloud. Amazon VPC enables enterprises to connect their existing infrastructure to a set of isolated AWS compute resources via a Virtual Private Network (VPN) connection, and to extend their existing management capabilities such as security services, firewalls, and intrusion detection systems to include their AWS resources.

For cloud users, Amazon CloudWatch is a useful management tool which collects raw data from partnered AWS services such as Amazon EC2 and then processes the information into readable, near real-time metrics. The metrics about EC2 include, for example, CPU utilization, network in/out bytes, disk read/write operations, etc.

### *5.2.2 Microsoft Windows Azure platform*

Microsoft's Windows Azure platform [53] consists of three components and each of them provides a specific set of services to cloud users. Windows Azure provides a Windows-based environment for running applications and storing data on servers in data centers; SQL Azure provides data services in the cloud based on SQL Server; and .NET Services offer distributed infrastructure services to cloud-based and local applications. Windows Azure platform can be used both by applications running in the cloud and by applications running on local systems.

Windows Azure also supports applications built on the .NET Framework and other ordinary languages supported in Windows systems, like C#, Visual Basic, C++, and others. Windows Azure supports general-purpose programs, rather than a single class of computing. Developers can create web applications using technologies such as ASP.NET and Windows Communication Foundation (WCF), applications that run as independent background processes, or applications that combine the two. Windows Azure allows storing data in blobs, tables, and queues, all accessed in a RESTful style via HTTP or HTTPS.

SQL Azure components are SQL Azure Database and "Huron" Data Sync. SQL Azure Database is built on Microsoft SQL Server, providing a database management system (DBMS) in the cloud. The data can be accessed using ADO.NET and other Windows data access interfaces. Users can also use on-premises software to work with this cloud-based information. "Huron" Data Sync synchronizes relational data across various on-premises DBMSs.

**Table 1** A comparison of representative commercial products

| Cloud Provider | Amazon EC2 | Windows Azure | Google App Engine |
|---|---|---|---|
| Classes of Utility Computing | Infrastructure service | Platform service | Platform service |
| Target Applications | General-purpose applications | General-purpose Windows applications | Traditional web applications with supported framework |
| Computation | OS Level on a Xen Virtual Machine | Microsoft Common Language Runtime (CLR) VM; Predefined roles of app. instances | Predefined web application frameworks |
| Storage | Elastic Block Store; Amazon Simple Storage Service (S3); Amazon SimpleDB | Azure storage service and SQL Data Services | BigTable and MegaStore |
| Auto Scaling | Automatically changing the number of instances based on parameters that users specify | Automatic scaling based on application roles and a configuration file specified by users | Automatic Scaling which is transparent to users |

The .NET Services facilitate the creation of distributed applications. The Access Control component provides a cloud-based implementation of single identity verification across applications and companies. The Service Bus helps an application expose web services endpoints that can be accessed by other applications, whether on-premises or in the cloud. Each exposed endpoint is assigned a URI, which clients can use to locate and access a service.

All of the physical resources, VMs and applications in the data center are monitored by software called the fabric controller. With each application, the users upload a configuration file that provides an XML-based description of what the application needs. Based on this file, the fabric controller decides where new applications should run, choosing physical servers to optimize hardware utilization.

### *5.2.3 Google App Engine*

Google App Engine [20] is a platform for traditional web applications in Google-managed data centers. Currently, the supported programming languages are Python and Java. Web frameworks that run on the Google App Engine include Django, CherryPy, Pylons, and web2py, as well as a custom Google-written web application framework similar to JSP or ASP.NET. Google handles deploying code to a cluster, monitoring, failover, and launching application instances as necessary. Current APIs support features such as storing and retrieving data from a BigTable [10] non-relational database, making HTTP requests and caching. Developers have read-only access to the filesystem on App Engine.

Table 1 summarizes the three examples of popular cloud offerings in terms of the classes of utility computing, target types of application, and more importantly their models of computation, storage and auto-scaling. Apparently, these cloud offerings are based on different levels of abstraction and management of the resources. Users can choose one type or combinations of several types of cloud offerings to satisfy specific business requirements.

## 6 Research challenges

Although cloud computing has been widely adopted by the industry, the research on cloud computing is still at an early stage. Many existing issues have not been fully addressed, while new challenges keep emerging from industry applications. In this section, we summarize some of the challenging research issues in cloud computing.

### 6.1 Automated service provisioning

One of the key features of cloud computing is the capability of acquiring and releasing resources on-demand. The objective of a service provider in this case is to allocate and de-allocate resources from the cloud to satisfy its service level objectives (SLOs), while minimizing its operational cost. However, it is not obvious how a service provider can achieve this objective. In particular, it is not easy to determine how to map SLOs such as QoS requirements to low-level resource requirement such as CPU and memory requirements. Furthermore, to achieve high agility and respond to rapid demand fluctuations such as in flash crowd effect, the resource provisioning decisions must be made online.

Automated service provisioning is not a new problem. Dynamic resource provisioning for Internet applications has been studied extensively in the past [47, 57]. These approaches typically involve: (1) Constructing an application performance model that predicts the number of application instances required to handle demand at each particular level, in order to satisfy QoS requirements; (2) Periodically predicting future demand and determining resource requirements using the performance model; and (3) Automatically allocating resources using the predicted resource requirements. Application performance model can be constructed using various techniques, including Queuing theory [47], Control theory [28] and Statistical Machine Learning [7].

Additionally, there is a distinction between proactive and reactive resource control. The proactive approach uses predicted demand to periodically allocate resources before they are needed. The reactive approach reacts to immediate demand fluctuations before periodic demand prediction is available. Both approaches are important and necessary for effective resource control in dynamic operating environments.

### 6.2 Virtual machine migration

Virtualization can provide significant benefits in cloud computing by enabling virtual machine migration to balance load across the data center. In addition, virtual machine migration enables robust and highly responsive provisioning in data centers.

Virtual machine migration has evolved from process migration techniques [37]. More recently, Xen [55] and VMWare [52] have implemented "live" migration of VMs that involves extremely short downtimes ranging from tens of milliseconds to a second. Clark et al. [13] pointed out that migrating an entire OS and all of its applications as one unit allows to avoid many of the difficulties faced by process-level migration approaches, and analyzed the benefits of live migration of VMs.

The major benefits of VM migration is to avoid hotspots; however, this is not straightforward. Currently, detecting workload hotspots and initiating a migration lacks the agility to respond to sudden workload changes. Moreover, the in-memory state should be transferred consistently and efficiently, with integrated consideration of resources for applications and physical servers.

### 6.3 Server consolidation

Server consolidation is an effective approach to maximize resource utilization while minimizing energy consumption in a cloud computing environment. Live VM migration technology is often used to consolidate VMs residing on multiple under-utilized servers onto a single server, so that the remaining servers can be set to an energy-saving state. The problem of optimally consolidating servers in a data center is often formulated as a variant of the vector bin-packing problem [11], which is an NP-hard optimization problem. Various heuristics have been proposed for this problem [33, 46]. Additionally, dependencies among VMs, such as communication requirements, have also been considered recently [34].

However, server consolidation activities should not hurt application performance. It is known that the resource usage (also known as the footprint [45]) of individual VMs may vary over time [54]. For server resources that are shared among VMs, such as bandwidth, memory cache and disk I/O, maximally consolidating a server may result in resource congestion when a VM changes its footprint on the server [38]. Hence, it is sometimes important to observe the fluctuations of VM footprints and use this information for effective server consolidation. Finally, the system must quickly react to resource congestions when they occur [54].

### 6.4 Energy management

Improving energy efficiency is another major issue in cloud computing. It has been estimated that the cost of powering and cooling accounts for 53% of the total operational expenditure of data centers [26]. In 2006, data centers in the US consumed more than 1.5% of the total energy generated in that year, and the percentage is projected to grow 18% annually [33]. Hence infrastructure providers are under enormous pressure to reduce energy consumption. The goal is not only to cut down energy cost in data centers, but also to meet government regulations and environmental standards.

Designing energy-efficient data centers has recently received considerable attention. This problem can be approached from several directions. For example, energy-efficient hardware architecture that enables slowing down CPU speeds and turning off partial hardware components [8] has become commonplace. Energy-aware job scheduling [50] and server consolidation [46] are two other ways to reduce power consumption by turning off unused machines. Recent research has also begun to study energy-efficient network protocols and infrastructures [27]. A key challenge in all the above methods is to achieve a good trade-off between energy savings and application performance. In this respect, few researchers have recently started to investigate coordinated solutions for performance and power management in a dynamic cloud environment [32].

### 6.5 Traffic management and analysis

Analysis of data traffic is important for today's data centers. For example, many web applications rely on analysis of traffic data to optimize customer experiences. Network operators also need to know how traffic flows through the network in order to make many of the management and planning decisions.

However, there are several challenges for existing traffic measurement and analysis methods in Internet Service Providers (ISPs) networks and enterprise to extend to data centers. Firstly, the density of links is much higher than that in ISPs or enterprise networks, which makes the worst-case scenario for existing methods. Secondly, most existing methods can compute traffic matrices between a few hundreds end hosts, but even a modular data center can have several thousand servers. Finally, existing methods usually assume some flow patterns that are reasonable in Internet and enterprises networks, but the applications deployed on data centers, such as MapReduce jobs, significantly change the traffic pattern. Further, there is tighter coupling in application's use of network, computing, and storage resources, than what is seen in other settings.

Currently, there is not much work on measurement and analysis of data center traffic. Greenberg et al. [21] report data center traffic characteristics on flow sizes and concurrent flows, and use these to guide network infrastructure design. Benson et al. [16] perform a complementary study of traffic at the edges of a data center by examining SNMP traces from routers.

### 6.6 Data security

Data security is another important research topic in cloud computing. Since service providers typically do not have access to the physical security system of data centers, they must rely on the infrastructure provider to achieve full data security. Even for a virtual private cloud, the service provider can only specify the security setting remotely, without knowing whether it is fully implemented. The infrastructure provider, in this context, must achieve the following objectives: (1) *confidentiality*, for secure data access and transfer, and (2) *auditability*, for attesting whether security setting of applications has been tampered or not. Confidentiality is usually achieved using cryptographic protocols, whereas auditability can be achieved using remote attestation techniques. Remote attestation typically requires a trusted platform module (TPM) to generate non-forgeable system summary (i.e. system state encrypted using TPM's private key) as the proof of system security. However, in a virtualized environment like the clouds, VMs can dynamically migrate from one location to another, hence directly using remote attestation is not sufficient. In this case, it is critical to build trust mechanisms at every architectural layer of the cloud. Firstly, the hardware layer must be trusted using hardware TPM. Secondly, the virtualization platform must be trusted using secure virtual machine monitors [43]. VM migration should only be allowed if both source and destination servers are trusted. Recent work has been devoted to designing efficient protocols for trust establishment and management [31, 43].

### 6.7 Software frameworks

Cloud computing provides a compelling platform for hosting large-scale data-intensive applications. Typically, these applications leverage MapReduce frameworks such as Hadoop for scalable and fault-tolerant data processing. Recent work has shown that the performance and resource consumption of a MapReduce job is highly dependent on the type of the application [29, 42, 56]. For instance, Hadoop tasks such as `sort` is I/O intensive, whereas `grep` requires significant CPU resources. Furthermore, the VM allocated to each Hadoop node may have heterogeneous characteristics. For example, the bandwidth available to a VM is dependent on other VMs collocated on the same server. Hence, it is possible to optimize the performance and cost of a MapReduce application by carefully selecting its configuration parameter values [29] and designing more efficient scheduling algorithms [42, 56]. By mitigating the bottleneck resources, execution time of applications can be significantly improved. The key challenges include performance modeling of Hadoop jobs (either online or offline), and adaptive scheduling in dynamic conditions.

Another related approach argues for making MapReduce frameworks energy-aware [50]. The essential idea of this approach is to turn Hadoop node into sleep mode when it has finished its job while waiting for new assignments. To do so, both Hadoop and HDFS must be made energy-aware. Furthermore, there is often a trade-off between performance and energy-awareness. Depending on the objective, finding a desirable trade-off point is still an unexplored research topic.

### 6.8 Storage technologies and data management

Software frameworks such as MapReduce and its various implementations such as Hadoop and Dryad are designed for distributed processing of data-intensive tasks. As mentioned previously, these frameworks typically operate on Internet-scale file systems such as GFS and HDFS. These file systems are different from traditional distributed file systems in their storage structure, access pattern and application programming interface. In particular, they do not implement the standard POSIX interface, and therefore introduce compatibility issues with legacy file systems and applications. Several research efforts have studied this problem [4, 40]. For instance, the work in [4] proposed a method for supporting the MapReduce framework using cluster file systems such as IBM's GPFS. Patil et al. [40] proposed new API primitives for scalable and concurrent data access.

### 6.9 Novel cloud architectures

Currently, most of the commercial clouds are implemented in large data centers and operated in a centralized fashion. Although this design achieves economy-of-scale and high manageability, it also comes with its limitations such high energy expense and high initial investment for constructing data centers. Recent work [12, 48] suggests that small-size data centers can be more advantageous than big data centers in many cases: a small data center does not consume so much power, hence it does not require a powerful and yet expensive cooling system; small data centers are cheaper to build and better geographically distributed than large data centers. Geo-diversity is often desirable for response time-critical services such as content delivery and interactive gaming. For example, Valancius et al. [48] studied the feasibility of hosting video-streaming services using application gateways (a.k.a. nano-data centers).

Another related research trend is on using voluntary resources (i.e. resources donated by end-users) for hosting cloud applications [9]. Clouds built using voluntary resources, or a mixture of voluntary and dedicated resources are much cheaper to operate and more suitable for non-profit applications such as scientific computing. However, this architecture also imposes challenges such managing heterogeneous resources and frequent churn events. Also, devising incentive schemes for such architectures is an open research problem.

## 7 Conclusion

Cloud computing has recently emerged as a compelling paradigm for managing and delivering services over the Internet. The rise of cloud computing is rapidly changing the landscape of information technology, and ultimately turning the long-held promise of utility computing into a reality.

However, despite the significant benefits offered by cloud computing, the current technologies are not matured enough to realize its full potential. Many key challenges in this domain, including automatic resource provisioning, power management and security management, are only starting to receive attention from the research community. Therefore, we believe there is still tremendous opportunity for researchers to make groundbreaking contributions in this field, and bring significant impact to their development in the industry.

In this paper, we have surveyed the state-of-the-art of cloud computing, covering its essential concepts, architectural designs, prominent characteristics, key technologies as well as research directions. As the development of cloud computing technology is still at an early stage, we hope our work will provide a better understanding of the design challenges of cloud computing, and pave the way for further research in this area.

## References

1. Al-Fares M et al (2008) A scalable, commodity data center network architecture. In: Proc SIGCOMM
2. Amazon Elastic Computing Cloud, aws.amazon.com/ec2
3. Amazon Web Services, aws.amazon.com
4. Ananthanarayanan R, Gupta K et al (2009) Cloud analytics: do we really need to reinvent the storage stack? In: Proc of HotCloud
5. Armbrust M et al (2009) Above the clouds: a Berkeley view of cloud computing. UC Berkeley Technical Report
6. Berners-Lee T, Fielding R, Masinter L (2005) RFC 3986: uniform resource identifier (URI): generic syntax, January 2005
7. Bodik P et al (2009) Statistical machine learning makes automatic control practical for Internet datacenters. In: Proc HotCloud
8. Brooks D et al (2000) Power-aware microarchitecture: design and modeling challenges for the next-generation microprocessors, IEEE Micro
9. Chandra A et al (2009) Nebulas: using distributed voluntary resources to build clouds. In: Proc of HotCloud
10. Chang F, Dean J et al (2006) Bigtable: a distributed storage system for structured data. In: Proc of OSDI
11. Chekuri C, Khanna S (2004) On multi-dimensional packing problems. SIAM J Comput 33(4):837–851
12. Church K et al (2008) On delivering embarrassingly distributed cloud services. In: Proc of HotNets
13. Clark C, Fraser K, Hand S, Hansen JG, Jul E, Limpach C, Pratt I, Warfield A (2005) Live migration of virtual machines. In: Proc of NSDI
14. Cloud Computing on Wikipedia, en.wikipedia.org/wiki/Cloudcomputing, 20 Dec 2009
15. Cloud Hosting, CLoud Computing and Hybrid Infrastructure from GoGrid, http://www.gogrid.com
16. Dean J, Ghemawat S (2004) MapReduce: simplified data processing on large clusters. In: Proc of OSDI
17. Dedicated Server, Managed Hosting, Web Hosting by Rackspace Hosting, http://www.rackspace.com
18. FlexiScale Cloud Comp and Hosting, www.flexiscale.com
19. Ghemawat S, Gobioff H, Leung S-T (2003) The Google file system. In: Proc of SOSP, October 2003
20. Google App Engine, URL http://code.google.com/appengine
21. Greenberg A, Jain N et al (2009) VL2: a scalable and flexible data center network. In: Proc SIGCOMM
22. Guo C et al (2008) DCell: a scalable and fault-tolerant network structure for data centers. In: Proc SIGCOMM
23. Guo C, Lu G, Li D et al (2009) BCube: a high performance, server-centric network architecture for modular data centers. In: Proc SIGCOMM
24. Hadoop Distributed File System, hadoop.apache.org/hdfs
25. Hadoop MapReduce, hadoop.apache.org/mapreduce
26. Hamilton J (2009) Cooperative expendable micro-slice servers (CEMS): low cost, low power servers for Internet-scale services In: Proc of CIDR
27. IEEE P802.3az Energy Efficient Ethernet Task Force, www.ieee802.org/3/az
28. Kalyvianaki E et al (2009) Self-adaptive and self-configured CPU resource provisioning for virtualized servers using Kalman filters. In: Proc of international conference on autonomic computing
29. Kambatla K et al (2009) Towards optimizing Hadoop provisioning in the cloud. In: Proc of HotCloud
30. Kernal Based Virtual Machine, www.linux-kvm.org/page/MainPage
31. Krautheim FJ (2009) Private virtual infrastructure for cloud computing. In: Proc of HotCloud
32. Kumar S et al (2009) vManage: loosely coupled platform and virtualization management in data centers. In: Proc of international conference on cloud computing
33. Li B et al (2009) EnaCloud: an energy-saving application live placement approach for cloud computing environments. In: Proc of international conf on cloud computing
34. Meng X et al (2010) Improving the scalability of data center networks with traffic-aware virtual machine placement. In: Proc INFOCOM

35. Mysore R et al (2009) PortLand: a scalable fault-tolerant layer 2 data center network fabric. In: Proc SIGCOMM
36. NIST Definition of Cloud Computing v15, csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc
37. Osman S, Subhraveti D et al (2002) The design and implementation of zap: a system for migrating computing environments. In: Proc of OSDI
38. Padala P, Hou K-Y et al (2009) Automated control of multiple virtualized resources. In: Proc of EuroSys
39. Parkhill D (1966) The challenge of the computer utility. Addison-Wesley, Reading
40. Patil S et al (2009) In search of an API for scalable file systems: under the table or above it? HotCloud
41. Salesforce CRM, http://www.salesforce.com/platform
42. Sandholm T, Lai K (2009) MapReduce optimization using regulated dynamic prioritization. In: Proc of SIGMETRICS/Performance
43. Santos N, Gummadi K, Rodrigues R (2009) Towards trusted cloud computing. In: Proc of HotCloud
44. SAP Business ByDesign, www.sap.com/sme/solutions/businessmanagement/businessbydesign/index.epx
45. Sonnek J et al (2009) Virtual putty: reshaping the physical footprint of virtual machines. In: Proc of HotCloud
46. Srikantaiah S et al (2008) Energy aware consolidation for cloud computing. In: Proc of HotPower
47. Urgaonkar B et al (2005) Dynamic provisioning of multi-tier Internet applications. In: Proc of ICAC
48. Valancius V, Laoutaris N et al (2009) Greening the Internet with nano data centers. In: Proc of CoNext
49. Vaquero L, Rodero-Merino L, Caceres J, Lindner M (2009) A break in the clouds: towards a cloud definition. ACM SIGCOMM computer communications review
50. Vasic N et al (2009) Making cluster applications energy-aware. In: Proc of automated ctrl for datacenters and clouds
51. Virtualization Resource Chargeback, www.vkernel.com/products/EnterpriseChargebackVirtualAppliance
52. VMWare ESX Server, www.vmware.com/products/esx
53. Windows Azure, www.microsoft.com/azure
54. Wood T et al (2007) Black-box and gray-box strategies for virtual machine migration. In: Proc of NSDI
55. XenSource Inc, Xen, www.xensource.com
56. Zaharia M et al (2009) Improving MapReduce performance in heterogeneous environments. In: Proc of HotCloud
57. Zhang Q et al (2007) A regression-based analytic model for dynamic resource provisioning of multi-tier applications. In: Proc ICAC