Infrastructures for High-Performance Computing: Cloud Computing

Loris Belcastro
c/o DIMES – University of Calabria
Via P. Bucci 42c
87036 Rende (CS)
Italy
Email: lbelcastro@dimes.unical.it
Telephone: +39 0984 494782

Fabrizio Marozzo
c/o DIMES – University of Calabria
Via P. Bucci 42c
87036 Rende (CS)
Italy
Email: fmarozzo@dimes.unical.it
Telephone: +39 0984 494782

Paolo Trunfio (corresponding author)
c/o DIMES – University of Calabria
Via P. Bucci 41c
87036 Rende (CS)
Italy
Email: paolo.trunfio@unical.it
Telephone: +39 0984 494788

Keywords

Cloud computing definition, Cloud deployment models, Cloud federation, Cloud service models, Community cloud, Hybrid cloud, Infrastructure as a service (IaaS), Inter-cloud, Interconnected clouds, Platform as a service (PaaS), Private cloud, Public cloud and Software as a service (SaaS)

Abstract

This chapter introduces the basic concepts of cloud computing, which is a model for providing on-demand network access to a shared pool of remote and infinitely scalable computing resources, such as networks, servers, storage, applications, and services. Such resources can be rapidly provisioned and released by final users, in most cases without requiring any interaction with the

service provider. After providing a definition of cloud computing, this chapter discusses the main characteristics of cloud systems, and describes the main service models (Software as a Service, Platforms as a Service, Infrastructure as a Service) and deployment models (Private Cloud; Community Cloud; Public Cloud; Hybrid Cloud) adopted by cloud providers. This chapter also discusses the most important key research findings on cloud computing.

Key points

- The chapter introduces the basic concepts of cloud computing, including the main features and benefits of cloud systems.
- The content of this chapter discusses the service models (IaaS, PaaS, SaaS) and deployment models (Public Cloud; Private Cloud; Hybrid Cloud; Community Cloud) adopted by cloud service providers.
- Some examples of cloud-based applications used in scientific fields such as bioinformatics, social computing, urban mobility are also provided.

Introduction

Cloud Computing abstracts computing resources to a utility-based model. This model is based on the virtualization of networks, servers, storage and services that clients can allocate on a pay-per-use basis to implement their distributed applications. From a client perspective, the cloud is an abstraction for remote, infinitely scalable provisioning of computing resources. From an implementation point of view, cloud systems are based on large sets of computing resources, located somewhere "in the cloud", which are allocated to applications on demand (Barga, Gannon and Reed, 2011). In addition, cloud computing is a distributed computing paradigm in which all the resources are dynamically scalable and virtualized (Talia, Trunfio and Marozzo, 2015). Virtualization is software-based technique that implements the separation of physical computing infrastructures and allows creating various "virtual" computing resources on the same hardware. It is a basic technology that powers cloud computing by making it possible to concurrently run different operating environments and multiple applications on the same server. Differently from other distributed computing paradigms, cloud users are not required to have knowledge of, expertise in, or control over the technology infrastructure in the "cloud" that supports them. A number of features define cloud applications, services, data, and infrastructure:

- Remotely hosted: services and/or data are hosted on remote infrastructure.
- Ubiquitous: services or data are available from anywhere.
- Pay-per-use: The result is a utility computing model similar to that of traditional utilities, like gas and electricity, where you pay for what you use.

As defined by NIST (National Institute of Standards and Technology) (Mell and Grance, 2011), cloud computing can be described as: "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".

From the NIST definition, we can identify five essential characteristics of cloud computing systems:

1. *On-demand self-service*: the ability to allocate resources on-demand and through self-service interfaces.
2. *Broad network access*: cloud services are available over high speed networks and are accessed through standard mechanisms that promote use by heterogeneous client devices (e.g., mobile phones, tablets, laptops, and workstations).
3. *Resource pooling*: A common pool of resources is transparently allocated to multiple users.

4. *Rapid elasticity*: the ability to scale up or down in the shortest amount of time, which helps manage unexpected spikes in workload while also preventing waste of money due to overbuilding and overprovisioning of computing resources.
5. *Measured service*: the ability to control the use of resources by leveraging a resource metering system.

In addition, cloud computing systems have other advantages in terms of:

- *Resilience and disaster recovery:* the use of multiple redundant hosting sites makes well-designed cloud computing suitable for business continuity and disaster recovery.
- *Energy saving*: using virtualization and workload consolidation, modern data centers can achieve up to a 90% reduction in energy consumption compared to traditional data centers.
- *Better collaboration and productivity*: cloud service allows access to data and services anytime, from anywhere and from any device, as long as you have an Internet connection.

Cloud systems can be classified on the basis of their *service model* (Software as a Service, Platform as a Service, Infrastructure as a Service) and their *deployment model* (public cloud, private cloud, community cloud).

Service Models

Cloud computing vendors provide their services according to three main models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). These three models are not mutually exclusive, but their use depends very much on the skills of the users. In fact, in medium-large companies, where there are people with medium-high level IT and systems engineering skills, all three models are often used simultaneously. In the following we discuss the main characteristics of such service models, which can satisfy the requirements of developers and final users, in terms of flexibility, portability, security, maintenance and costs.

Software as a Service defines a delivery model in which software and data are provided through the Internet to customers as ready-to-use services. Specifically, software and associated data are hosted by providers, and customers access them without need to use any additional hardware or software. In some cases, customers could pay a fee for using such services, with no additional purchase of infrastructure or software licenses. Examples of common SaaS applications are email systems (e.g., Gmail, Outlook), collaboration and messaging (Slack), document management (Microsoft Office 365, Google Docs), file hosting (Dropbox), customer relationship management (Salesforce), graphics (Adobe Creative Cloud), and others. The entire SaaS application is fully managed by the service vendor, who is responsible for providing updates, fixing bugs, managing user access and security, storing and preserving data, ensuring continuity of service. SaaS services are generally considered solutions with minimal risk for end users, but may have some limitations in terms of customization possibilities. In most cases, in fact, users can customize the application interface and control its behavior, but they cannot introduce new features or decide which software and hardware components to use to support its execution. The underlying processing and storage resources typically scale automatically to meet growing application demand, so users don't have to worry about allocating resources manually (at most, an additional cost may be required by the customer). However, SaaS services suffer from the vendor lock problem, meaning that in some cases applications and data cannot be moved to other vendors. For example, application data may be in a format that cannot be exported or read by applications provided by other vendors.

In the *Platform as a Service* model, cloud vendors deliver a computing platform for developing and running applications, which typically include databases, development environments, frameworks and runtimes for building, testing and running custom applications. Developers can just focus on deploying applications since cloud providers are in charge of maintenance and optimization of the environment and underlying infrastructure, including operating system and software updates, bug fixes, backups. The security of code and additional libraries used to build custom applications is under the responsibility of the developer. Like the SaaS model, the underlying computing and storage resources normally scale automatically. Developers pay for the compute and storage resources, and for the licenses of libraries and tools used by their applications. In most cases, developers are provided with a web graphical user interface that helps them in the development of complex applications, as it provides a set of preconfigured, modular and easily integrated "environment" services. Furthermore, the possibility of sharing the workspace with all members of the DevOps team during the entire software lifecycle (e.g., coding, testing, deploying), permits to drastically reduce the time to market of the software. Typically, applications developed in a PaaS environment are distributed as an out-of-the-box SaaS. Google App Engine, Microsoft Azure and AWS Elastic Beanstalk are some examples of PaaS cloud environments. The vendor lock problem is theoretically more limited in the PaaS model, since a custom application can be moved to another provider, but only if the new provider shares the tools and services required by the application to function properly.

Finally, *Infrastructure as a Service* is an outsourcing model under which customers rent hardware, such as CPUs, disks, RAM, or more complex resources like virtual machines or operating systems to support their operations (e.g., Amazon EC2, Google Compute Engine, RackSpace Cloud). Differently from on-premise provisioning, such resources are managed and maintained in the vendor's data centers. IaaS customers can access and use these resources over the Internet, paying for their use with a subscription fee or pay-as-you-go basis. Compared to traditional IT, IaaS solutions are much more cost effective, as customers can rapidly scale allocated computing resources, scaling them up in response to spikes in workloads and decreasing them during periods of inactivity to avoid waste. IaaS solutions are also very powerful in terms of flexibility, as developers can configure the servers that will host their applications according to their needs, choosing the most appropriate software components (operating system, libraries, software modules) and configuring them in the most efficient way.

It is therefore easy to understand that IaaS solutions are not designed for low-skilled users. Users of an IaaS normally have skills on system and network administration as they must deal with configuration, operation and maintenance tasks. In addition, IaaS customers must address all security issues, from the operating system to the application layer, as well as business continuity and disaster recovery procedures (for example, by using high availability services with geographical redundancy and automatic data backup procedures). Compared to the PaaS approach, the IaaS model has a higher system administration costs for the user, although it allows a full customization of the execution environment. In terms of portability, the IaaS deployment model doesn't present any particular problems as most vendors allow you to download virtual machines in standard formats, which can easily be moved to a different provider.

Deployment Models

Cloud computing services are delivered according to four main deployment models: public, private, hybrid, and community cloud.

A *public cloud* provider delivers services to the general public through the Internet. The users of a public cloud have little or no control over the underlying technology infrastructure. In this model, services can be offered for free, or provided according to a pay-*per*-use policy. The main public providers, such as Google, Microsoft, Amazon, own and manage their proprietary data centers delivering services built on top of them.

A *private cloud* provider offers operations and functionalities “as a service”, which are deployed over a company intranet or hosted in a remote data center. Often, small and medium-sized IT companies prefer this deployment model as it offers advanced security and data control solutions that are not available in the public cloud model.

A *hybrid cloud* is the composition of at least one private and at least one public cloud, which remain different entities but are linked together to provide a flexible set of computing services. Companies can extend their private clouds using other private clouds from partner companies, or public clouds. In particular, by extending the private infrastructure with public cloud resources, it is possible to satisfy peaks of requests, better serve user requests, and implement high availability strategies. As an example, an organization may use a public cloud to archive older data or backups, while using in-house, private cloud storage for more recent data. Using a hybrid cloud model, organizations can improve data management, storing the most sensitive data in the internal private cloud and the less important ones in the public cloud (which is usually also less expensive). It is worth noting that the hybrid cloud model requires a form of federation among clouds. The term *federated cloud* was used for many years, but over time this term fell into disuse. In fact, the terms hybrid cloud and federated cloud are different terms that essentially describe the same way of use: both terms refer to merging internal and external clouds to achieve a common goal.

A *community cloud* is a model in which the infrastructure is provisioned for reserved use of a given community of organizations that have shared objectives and requirements. This kind of cloud may be owned and managed by one or more of the organizations in the community, or by an organization that is not part of the community, or some combination of them.

Users access cloud computing services using *client* devices, such as desktop computers, laptops, tablets and smartphones. Through these devices, users access and interact with cloud-based services using a Web browser or desktop/mobile app. The business software and user's data are executed and stored on servers hosted in cloud data centers that provide *storage and computing resources*. Resources include thousands of servers and storage devices connected to each other through an *intra-cloud network*. The transfer of data between data centers and users takes place on a wide-area *network*.

Several technologies and standards are used by the different components of the architecture. For example, users can interact with cloud services through SOAP-based or RESTful Web services (Richardson and Ruby, 2007). *HTML5* and *Ajax* technologies allow Web interfaces to cloud services to have look and interactivity equivalent to those of desktop applications. *Open Cloud Computing Interface* (OCCI, OCCI Working Group, see Relevant Websites section) specifies how cloud providers can deliver their compute, data, and network resources through a standardized interface. Another example is *Open Virtualization Format* (OVF, OVF Specification, see Relevant Websites section) for packaging and distributing virtual devices or software (e.g., virtual operating system) to be run on virtual machines.

Examples of Cloud-based Applications

Cloud computing has been widely utilized in scientific fields such as astronomy, meteorology, social computing, and bioinformatics, which require extensive analysis of large volumes of data. However, developing and configuring Cloud-based applications requires a high level of expertise, which can be a significant bottleneck in the adoption of such applications by researchers. Many solutions have been proposed for Big Data analysis on Clouds in the bioinformatics field. As examples: Langmead, Hansen and Leek (2010) proposed Myrna, a Cloud system that leverages MapReduce to calculate differential gene expression in large RNA-seq datasets; Wang, Li, Chen and al. (2014) proposed a heterogeneous Cloud framework that employs MapReduce and multiple hardware execution engines on FPGA to accelerate genome sequencing applications.

Moreover, cloud computing has been used for executing complex Big Data mining applications, such as association rule analysis between genome variations and clinical conditions of a large group of patients.

Cloud computing has a wide range of applications in the field of urban mobility, including: real-time traffic data analysis for reducing congestion and improving traffic flow (Ashokkumar, Baron and Arshadprabhu (2015)); optimizing public transportation systems, by collecting and analyzing real-time data on passenger demand and traffic conditions to improve schedules and routes (Ali, Jaber, Abd et al (2022)); analyzing data of vehicles and pedestrians in a wide urban scenario for discovering frequent patterns, trajectories and rules from user trajectory (Belcastro, Marozzo and Perrella (2021)); and predicting flight delay according to weather conditions (Belcastro, Marozzo, Talia et al. (2016)).

Cloud computing has also been employed for data analysis on vast amounts of data gathered from social media. As an example, cloud computing can be used to process large amounts of social media data, such as tweets or posts, to extract sentiment information about how users feel about certain topics, products, political factions (Belcastro, Branda, Cantini et al. (2022); Liang and Wang (2019)). In addition, Cloud computing can be used to analyze social networks and identify patterns of behavior, influencers, hot topics, clusters of users, and other important network structures (Jayaram, Jayachandran, Dileep et al. (2018)).

Open Challenges

In the following we discuss some of the most important research trends and issues to be addressed in Cloud systems, especially concerning managing and mining large-scale data repositories:

- Security and privacy: mainly due to the increasing risk of data breaches, cyber-attacks, and data loss.
- Performance and scalability: cloud computing systems must be capable of handling a massive amount of data and workload. Ensuring high performance and scalability is essential to meet the demands of users and applications.
- Interoperability and standardization: cloud computing is a highly distributed environment, and interoperability and standardization among different cloud providers are still significant challenges. This lack of interoperability can create vendor lock-in, limit user choice, and increase costs. In addition, workflow systems available on Cloud for intensive data processing should be extended to support interoperability and ease cooperation among teams using different data formats and tools.
- Programming models: several scalable programming models have been proposed (Belcastro, Cantini, Marozzo et al. (2022)), such as MapReduce (Dean and Ghemawat, 2008), Message Passing (Gropp, Lusk and Skjellum (1999)), and Bulk Synchronous Parallel (Valiant, 1990). Some development work must be done for extending existing Cloud-based applications to support different programming models, which could improve their capabilities in terms of efficiency, scalability and interoperability.
- Data storage: the increasing amount of data generated every day needs even more scalable

and distributed data storage systems. Cloud services and applications should improve their capabilities to access data stored on high-performance storage systems (e.g., NoSQL systems, Object based storage) by using different protocols.

Addressing these open issues is critical for the continued growth and success of cloud computing. Research and development efforts are underway to develop new technologies and solutions to address these challenges and improve the overall performance and reliability of cloud computing systems.

Closing Remarks

In this chapter we provided an overview of cloud computing concepts and models. Starting from the NIST definition of cloud computing, we discussed the main features of cloud systems, and analyzed the most important service models (Software as a Service, Platforms as a Service, Infrastructure as a Service) and deployment models (Private Cloud; Community Cloud; Public Cloud) currently adopted by cloud providers. We discussed the most important models for the interconnection and interoperability of cloud environments (Hybrid Cloud, Cloud Federation, and Inter-cloud). Finally, we presented some examples of Cloud-based applications and discussed the main current challenges and future research directions.

See also

Computing Languages for Bioinformatics: Java. Dedicated Bioinformatics Analysis Hardware. Models and Languages for High-Performance Computing. Parallel Architectures for Bioinformatics. Text Mining Applications.

References

Ali, M.H., Jaber, M.M., Abd, S.K., Alkhayyat, A. and Albaghdadi, M. F. (2022). Big data analysis and cloud computing for smart transportation system integration. *Multimedia Tools and Applications*, 1-18.

Ashokkumar, K., Baron, S. and Arshadprabhu, R. (2015). Cloud based intelligent transport system. *Procedia Computer Science* **50**, 58-63.

Barga, R., Gannon, D. and Reed, D. (2011). The client and the cloud: Democratizing research computing. *IEEE Internet Computing* **15**(1), 72–75

Belcastro, L., Marozzo, F., Talia, D. and Trunfio, P. (2016). Using scalable data mining for predicting flight delays. *ACM Transactions on Intelligent Systems and Technology (TIST)* **8**(1), 1-20.

Belcastro, L., Marozzo, F. and Perrella, E. (2021). Automatic detection of user trajectories from social media posts. *Expert Systems with Applications* **186**, 115733.

Belcastro, L., Branda, F., Cantini, R., Marozzo, F., Talia, D. and Trunfio, P. (2022). Analyzing voter behavior on social media during the 2020 US presidential election campaign. *Social Network Analysis and Mining* **12**(1), 83.

Belcastro, L., Cantini, R., Marozzo, F., Orsino, A., Talia, D. and Trunfio, P. (2022). Programming Big Data Analysis: Principles and Solutions. *Journal of Big Data* **9**(4).

Dean, J. and Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM* **51**(1), 107-113.

Gropp, W., Lusk, E. and Skjellum, A. (1999). Using MPI: portable parallel programming with the message-passing interface. Vol. 1. Cambridge: MIT press.

Jayaram, B., Jayachandran, A., Dileep, M. et al. (2018). A Survey On Social Media Data Analytics And Cloud Computing Tools. *International Journal of Mechanical and Production Engineering Research and Development* **8**(3), 243.

Langmead, B., Hansen, K. D. and Leek, J. T. (2010). Cloud-scale RNA-sequencing differential expression analysis with Myrna. *Genome biology* **11**, 1-11.

Liang, R. and Wang, J.Q. (2019). A linguistic intuitionistic cloud decision support model with sentiment analysis for product selection in E-commerce. *International Journal of Fuzzy Systems* **21**, 963-977.

Mell, P. and Grance, T. (2011). The NIST Definition of Cloud Computing. National Institute of Standards and Technology (NIST), Gaithersburg, MD: NIST.

Richardson, L. and Ruby, S. (2007). RESTful Web Services. Sebastopol, CA: O'Reilly & Associates.

Talia, D., Trunfio, P. and Marozzo, F. (2015). Data Analysis in the Cloud. The Netherlands: Elsevier.

Toosi, A.N., Calheiros, R.N. and Buyya, R. (2014). Interconnected cloud computing environments. *ACM Computing Surveys* **47**(1), 1-47.

Valiant, L.G. (1990). A bridging model for parallel computation. *Communications of the ACM* **33**(8), 103-111.

Wang, C., Li, X., Chen, P, Zhou, X. and Yu, H. (2014). Heterogeneous cloud framework for big data genome sequencing. *IEEE/ACM transactions on computational biology and bioinformatics* **12**(1), 166-178.

Relevant Websites

http://www.occi-wg.org – OCCI Working Group.

http://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.0.pdf – OVF Specification.