Cloud Computing Growth of Micro Services and Future of Application Development

*Deepa Puranik, Asst Professor, BCA Department, New Horizon College, Bangalore.

Abstract

This paper attempts to study **cloud computing** the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence. The impact of cloud computing on industry and end users would be difficult to overstate: everyday life has been completely transformed by the omnipresence of software that runs on cloud networks. By leveraging cloud computing, startups and businesses are able to optimize costs and increase their offerings without purchasing and managing all the hardware and software. Independent developers are empowered to launch globally-available apps and online services. Researchers can share and analyze data at scales once reserved only for highly-funded projects. And internet users can quickly access software and storage to create, share, and store digital media in quantities that extend far beyond the computing capacity of their personal devices. Despite the growing presence of cloud computing, its details remain obscure to many. What exactly is the cloud, how does one use it, and what are its benefits for businesses, developers, researchers, government, healthcare practitioners, and students? In this conceptual article, we'll provide a general overview of cloud computing, its history, delivery models, offerings, and risks. By the end of this article, you should have an understanding of how the cloud can help support business, research, education, and community infrastructure and how to get started using the cloud for your own projects.

Cloud computing is the delivery of computing resources as a service, meaning that the resources are owned and managed by the cloud provider rather than the end user. Those resources may include anything from browser-based software applications (such as Tik Tok or Netflix), third party data storage for photos and other digital media (such as iCloud or Dropbox), or third-party servers used to support the computing infrastructure of a business, research, or personal project. Before the broad proliferation of cloud computing, businesses and general computer users typically had to buy and maintain the software and hardware that they wished to use. With the growing availability of cloud-based applications, storage, services, and machines, businesses and

consumers now have access to a wealth of on-demand computing resources as internet-accessed services. Shifting from on-premise software and hardware to networked remote and distributed resources means cloud users no longer have to invest the labor, capital, or expertise required for buying and maintaining these computing resources themselves.

Key words: Cloud computing, resources, software, networks, servers, storage, applications, and services

Introduction

The National Institute of Standards and Technology (NIST), a non-regulatory agency of the United States Department of Commerce with a mission to advance innovation, defines cloud computing as:

a model for enabling ubiquitous, 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.

NIST lists the following as the five essential characteristics of cloud computing:

- On-demand self-service: Cloud resources can be accessed or provisioned without human interaction. With this model, consumers can gain immediate access to cloud services upon signup. Organizations can also create mechanisms for allowing employees, customers, or partners to access internal cloud services on demand according to predetermined logics without needing to go through IT services.
- Broad network access: Users can access cloud services and resources through any device and in any networked location provided that they have permission.
- Resource pooling: Cloud provider resources are shared by multiple tenants while keeping the data of individual clients hidden from other clients.

- Rapid elasticity: Unlike on-premise hardware and software, cloud computing resources can be rapidly increased, decreased, or otherwise modified based on the cloud user's changing needs.
- Measured service: Usage of cloud resources is metered so that businesses and other cloud users need only pay for the resources they use in any given billing cycle.

These characteristics offer a wide variety of transformative opportunities for businesses and individuals alike, which we'll discuss later in the section Benefits of Cloud Computing. To gain some additional context, let's briefly review the emergence of cloud computing. This unprecedented access to computing resources has given rise to a new wave of cloud-based businesses, changed IT practices across industries, and transformed many everyday computer-assisted practices. With the cloud, individuals can now work with colleagues over video meetings and other collaborative platforms, access entertainment and educational content on demand, communicate with household appliances, hail a cab with a mobile device, and rent a vacation room in someone's house.

Objective:

This paper intends to explore and analyze **cloud computing** means storing and accessing data and programs over the internet instead of computer's hard drive. Leading to delivery of on-demand computing resources—everything from applications to services

Cloud computing, monoliths, microservices

Many aspects of cloud computing can be traced as far back as the 1950s, when universities and companies rented out computation time on mainframe computers. At the time, renting was one of the only ways to access computing resources as computing technology was too large and expensive to be owned or managed by individuals. By the 1960s, computer scientists like John McCarthy of Stanford University and J.C.R Licklider of The U.S. Department of Defense Advanced Research Projects Agency (ARPA) began proposing ideas that anticipated some of the major features of cloud computing today, such as the conceptualization of computing as a public utility and the possibility of a network of computers that would allow people to access data and programs from anywhere in the world.

Cloud computing, however, didn't become a mainstream reality and a popular term until the first decade of the 21st century. This decade saw the launch of cloud services like Amazon's Elastic Compute (EC2) and Simple Storage Service (S3) in 2006, Heroku in 2007, Google Cloud Platform in 2008, Alibaba Cloud in 2009, Windows Azure (now Microsoft Azure) in 2010, IBM's SmartCloud in 2011, and DigitalOcean in 2011. These services allowed existing businesses to optimize costs by migrating their in-house IT infrastructure to cloud-based resources and provided independent developers and small developer teams resources for creating and deploying apps. Cloud-based applications, known as *Software as a Service (SasS)* — which we'll discuss in greater detail in the **Cloud Delivery Models** section — also became popular during this time period. Unlike on-premise software, or software that users need to physically install and maintain on their machines, SaaS increased the availability of applications by allowing users to access them from a variety of devices on demand.

Some of these cloud-based applications — such as Google's productivity apps (Gmail, Drive, and Docs) and Microsoft 365 (a cloud-based version of the Microsoft Office Suite) — were offered by the same companies that launched cloud infrastructure services, while other pre-existing software products, such as Adobe Creative Cloud, were launched as cloud-based applications using the services of cloud providers. New SaaS products and businesses also emerged based on the novel opportunities of these cloud providers, such as Netflix's streaming services in 2007, the music platform Spotify in 2008, the file-hosting service Dropbox in 2009, the video conferencing service Zoom in 2012, and the communication tool Slack in 2013. Today, cloud-based IT infrastructure and cloud-based applications have become a popular choice for both businesses and individual users and their market share is expected to grow.

Cloud Delivery Models

Cloud resources are provided in a variety of different delivery models that offer customers different levels of support and flexibility.

Infrastructure as a Service (IaaS)

IaaS is the on-demand delivery of computing infrastructure, including operating systems, networking, storage, and other infrastructural components. Acting much like a virtual equivalent to physical servers, IaaS relieves cloud users of the need to buy and maintain physical servers while also providing the flexibility to scale and pay for resources as needed. IaaS is a popular option for businesses that wish to leverage the advantages of the cloud and have system administrators who can oversee the installation, configuration, and management of operating systems, development tools, and other underlying infrastructure that they wish to use. However, IaaS is also used by developers, researchers, and others who wish to customize the underlying infrastructure of their computing environment. Given its flexibility, IaaS can support everything from a company's computing infrastructure to web hosting to big data analysis.

Platform as a Service (PaaS)

PaaS provides a computing platform where the underlying infrastructure (such as the operating system and other software) is installed, configured, and maintained by the provider, allowing users to focus their efforts on developing and deploying apps in a tested and standardized environment. PaaS is commonly used by software developers and developer teams as it cuts down on the complexity of setting up and maintaining computer infrastructure, while also supporting collaboration among distributed teams. PaaS can be a good choice for developers who don't have the need to customize their underlying infrastructure, or those who want to focus their attention on development rather than DevOps and system administration.

Software as a Service (SaaS)

SaaS providers are cloud-based applications that users access on demand from the internet without needing to install or maintain the software. Examples include GitHub, Google Docs, Slack, and Adobe Creative Cloud. SaaS applications are popular among businesses and general users given that they're often easy to adopt, accessible from any device, and have free, premium, and enterprise versions of their applications. Like PaaS, SaaS abstracts away the underlying infrastructure of the software application so that users are only exposed to the interface they interact with. **Cloud Environments**

Cloud services are available as public or private resources, each of which serves different needs.

Public Cloud

The *public cloud* refers to cloud services (such as virtual machines, storage, or applications) offered publicly by a commercial provider to businesses and individuals. Public cloud resources are hosted on the commercial provider's hardware, which users access through the internet. They are not always suitable for organizations in highly-regulated industries, such as healthcare or finance, as public cloud environments may not comply with industry regulations regarding customer data.

Private Cloud

The *private cloud* refers to cloud services that are owned and managed by the organization that uses them and available only to the organization's employees and customers. Private clouds allow organizations to exert greater control over their computing environment and their stored data, which can be necessary for organizations in highly-regulated industries. Private clouds are sometimes seen as more secure than public clouds as they are accessed through private networks and enable the organization to directly oversee their cloud security. Public cloud providers sometimes provide their services as applications that can be installed on private clouds, allowing organizations to keep their infrastructure and data on premise while taking advantage of the public cloud's latest innovations.

Hybrid Cloud and Multicloud

Many organizations use a *hybrid cloud* environment which combines public and private cloud resources to support the organization's computing needs while maintaining compliance with industry regulation. *Multicloud* environments are also common, which entail the use of more than one public cloud provider (for example, combining Amazon Web Services and DigitalOcean).

Benefits of Cloud Computing

Cloud computing offers a variety of benefits to individuals, businesses, developers, and other organizations. These benefits vary according to the cloud users goals and activities.

For Business and Industry

Prior to the proliferation of cloud computing, most businesses and organizations needed to purchase and maintain the software and hardware that supported their computing activities. As cloud computing resources became available, many businesses began using them to store data, provide enterprise software, and deploy online products and services. Some of these cloud-based adoptions and innovations are industry-specific. In healthcare, many providers use cloud services that are specifically designed to store and share patient data or communicate with patients. In academia, educators and researchers use cloud-based teaching and research apps. But there are also a large number of general cloud-based tools that have been adopted across industries, such as apps for productivity, messaging, expense management, video conferencing, project management, newsletters, surveys, customer relations management, identity management, and scheduling. The rapid growth of cloud-based business apps and infrastructure shows that the cloud isn't just changing business IT strategy: it's a booming business in its own right.

Cloud-based technologies offer businesses several key advantages. First, they can help optimize IT costs. As businesses shift towards renting computing resources, they no longer have to invest as much in purchasing and maintaining on-premise IT infrastructure. Cloud computing is also enormously flexible, allowing businesses to rapidly scale (and only pay for) the computing resources they actually use. Cost, however, is not the only consideration that drives cloud adoption in business. Cloud-based technologies can help make internal IT processes more efficient as they can be accessed on demand by employees without needing to go through IT approval processes. Cloud-based apps can improve collaboration across a business as they allow for real-time communication and data sharing.

For Independent Developers

Computing resources that were once only affordable to large companies and organizations are now available on demand through an internet connection and at a fraction of their previous cost. In effect, independent developers can rapidly deploy and experiment with cloud-based apps. Cloud-based apps for sharing code (such as GitHub) have also made it easier for developers to build upon and collaborate on open source software projects. Additionally, cloud-based educational platforms and interactive coding tutorials have expanded access to developer education, enabling individuals without formal technical training to learn to code in their own time.

Altogether, these cloud-based computing and educational resources have helped lower the barriers to learning developer skills and deploying cloud-based apps. Formal training, company support, and massive amounts of startup capital are no longer necessary for individuals to experiment with creating and deploying apps, allowing for more individuals to participate in cloud development, compete with established industry players, and create and share apps as side projects.

For Researchers

As machine learning methods become increasingly important in scientific research, cloud computing has become essential to many scientific fields, including astronomy, physics, genomics, and artificial intelligence. The massive amount of data collected and analyzed in machine learning and other data-intensive research projects often require computing resources that scale beyond the capacity of hardware owned by an individual researcher or provisioned by the university. Cloud computing allows researchers to access (and only pay for) computing resources as their workloads require and allows for real-time collaboration with research partners across the globe. Without commercial cloud providers, a majority of academic machine learning research would be limited to individuals with access to university-provisioned, high-powered computing resources.

For Educators and Students

Cloud computing has also provided students with tools for supplementing their education and opportunities to put their technical skills into practice as they learn. Cloud-based apps for sharing, teaching, and collaborating on code and data (such as GitHub and Jupyter Notebooks) enable students to learn technical skills in a hands-on manner by studying, deploying, and contributing to open source software and research projects relevant to their field or professional aspirations. And just like independent developers, students are able to use cloud computing resources to share their code and apps with the public and reap the satisfaction of understanding the real-world application of their skills.

Students, researchers, and educators can also take advantage of cloud computing resources to support personalized academic infrastructure and practice greater control over their computing environments. Some academics prefer this approach as it lets them pick which applications they use, customize the functionality and design of these tools, and limit or prohibit the collection of data. There are also a growing number of cloud-based applications developed specifically for academic purposes that supplement or provide alternatives to traditional academic IT offerings. Voyant Tools offers students and researchers a code-free method for providing textual analysis on documents of their choosing and The HathiTrust provides access to its digital collection of millions of volumes. Reclaim Hosting, Commons in a Box, the Modern Language Humanities Commons, and Manifold offer educational, publishing, and networking tools designed specifically for academic communities.

Conclusion

Cloud technologies offer a variety of opportunities to businesses, independent developers, researchers, educators, and students. By understanding the different services, models, benefits and risks offered by the cloud, users can make informed decisions about how to best take advantage of its offerings. Migrating an organization's computing resources to the cloud can be an extremely complex endeavor, requiring in-depth planning, governance structures, and continuous oversight to avoid incompatibilities, data loss, and cost optimization.

Though the cloud can help organizations cut costs on computing infrastructure, they will still need IT experts to direct and manage infrastructure. Given the vast power some cloud service providers have over world affairs, cloud users may want to consider the ethics of the company that their business is supporting. Reviewing company practices with regard to topics such as data collection, advertising, hate speech, politics, misinformation, the environment, and labor may help a cloud user choose a provider that best reflects their personal values.

References

1. Wiegers, K. (2013). Creating a Software Engineering Culture. Addison-Wesley. pp. 211–2. ISBN 9780133489293.
2. Lewis, W.E. (2016). Software Testing and Continuous Quality Improvement (3rd ed.). CRC Press. pp. 92–6. ISBN 9781439834367.
3. Machado, P.; Vincenzi, A.; Maldonado, J.C. (2010). "Chapter 1: Software Testing: An Overview". In Borba, P.; Cavalcanti, A.; Sampaio, A.; Woodcook, J. (eds.). Testing Techniques in Software Engineering. Springer Science & Business Media. pp. 13–14. ISBN 9783642143342.
4. Clapp, J.A.; Stanten, S.F.; Peng, W.W.; et al. (1995). Software Quality Control, Error Analysis, and Testing. Nova Data Corporation. p. 254. ISBN 978-0815513636.

5. "ISTQB CTFL Syllabus 2018". ISTQB - International Software Testing Qualifications Board.
6. Binder, Robert V. (1999). Testing Object-Oriented Systems: Objects, Patterns, and Tools. Addison-Wesley Professional. p. 45. ISBN 978-0-201-80938-1.
7. Beizer, Boris (1990). Software Testing Techniques (Second ed.). New York: Van Nostrand Reinhold. pp. 21, 430. ISBN 978-0-442-20672-7.
8. Xuan, Jifeng; Monperrus, Martin (2014). "Test case purification for improving fault localization". Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering - FSE 2014: 52–63. arXiv:1409.3176. Bibcode:2014arXiv1409.3176X. doi:10.1145/2635868.2635906. ISBN 9781450330565. S2CID 14540841.
9. Woods, Anthony J. (June 5, 2015). "Operational Acceptance – an application of the ISO 29119 Software Testing standard" (Whitepaper). Capgemini Australia. Retrieved January 9, 2018.
10. Kaner, Cem; Bach, James; Pettichord, Bret (2001). Lessons Learned in Software Testing: A Context-Driven Approach. Wiley. pp. 31–43. ISBN 9780471081128.
11. Ammann, Paul; Offutt, Jeff (January 28, 2008). Introduction to Software Testing. Cambridge University Press. p. 215. ISBN 978-0-521-88038-1. Retrieved November 29, 2017.
12. Danglot, Benjamin; Vera-Pérez, Oscar Luis; Baudry, Benoit; Monperrus, Martin (2019). "Automatic test improvement with DSpot: a study with ten mature open-source projects". Empirical Software Engineering. 24 (4): 2603–2635. arXiv:1811.08330. doi:10.1007/s10664-019-09692-y. S2CID 53748524.
13. Danglot, Benjamin; Vera-Perez, Oscar; Yu, Zhongxing; Zaidman, Andy; Monperrus, Martin; Baudry, Benoit (November 2019). "A snowballing literature study on test amplification" (PDF). Journal of Systems and Software. 157: 110398. arXiv:1705.10692. doi:10.1016/j.jss.2019.110398. S2CID 20959692.
14. "Standard Glossary of Terms used in Software Testing" (PDF). Version 3.1. International Software Testing Qualifications Board. Retrieved January 9, 2018.
15. O'Reilly, Tim (September 30, 2005). "What is Web 2.0". O'Reilly Media. Section 4. End of the Software Release Cycle. Retrieved January 11, 2018.

16. Auerbach, Adam (August 3, 2015). "Part of the Pipeline: Why Continuous Testing Is Essential". TechWell Insights. TechWell Corp. Retrieved January 12, 2018.
17. Philipp-Edmonds, Cameron (December 5, 2014). "The Relationship between Risk and Continuous Testing: An Interview with Wayne Ariola". Stickyminds. Retrieved January 16, 2018.
18. Ariola, Wayne; Dunlop, Cynthia (October 2015). DevOps: Are You Pushing Bugs to Clients Faster? (PDF). Pacific Northwest Software Quality Conference. Retrieved January 16, 2018.
19. Auerbach, Adam (October 2, 2014). "Shift Left and Put Quality First". TechWell Insights. TechWell Corp. Retrieved January 16, 2018.
20. "Section 4.38". ISO/IEC/IEEE 29119-1:2013 – Software and Systems Engineering – Software Testing – Part 1 – Concepts and Definitions. International Organization for Standardization. Retrieved January 17, 2018.
21. "Globalization Step-by-Step: The World-Ready Approach to Testing. Microsoft Developer Network". Msdn.microsoft.com. Retrieved January 13, 2012.
22. ""QuickCheck: a lightweight tool for random testing of Haskell programs"". Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming. 2000. doi:10.1145/351240.351266. S2CID 5668071.
23. "Software Testing Lifecycle". etestinghub. Testing Phase in Software Testing. Retrieved January 13, 2012.
24. Dustin, Elfriede (2002). Effective Software Testing. Addison-Wesley Professional. p. 3. ISBN 978-0-201-79429-8.
25. Brown, Chris; Cobb, Gary; Culbertson, Robert (April 12, 2002). Introduction to Rapid Software Testing.
26. "What is Test Driven Development (TDD)?". Agile Alliance. December 5, 2015. Retrieved March 17, 2018.
27. "Test-Driven Development and Continuous Integration for Mobile Applications". msdn.microsoft.com. Retrieved March 17, 2018.
28. Rodríguez, Ismael; Llana, Luis; Rabanal, Pablo (2014). "A General Testability Theory: Classes, properties, complexity, and testing reductions". IEEE

Transactions on Software Engineering. 40 (9): 862–894. doi:10.1109/TSE.2014.2331690. ISSN 0098-5589. S2CID 6015996.

29. Rodríguez, Ismael (2009). "A General Testability Theory". CONCUR 2009 - Concurrency Theory, 20th International Conference, CONCUR 2009, Bologna, Italy, September 1–4, 2009. Proceedings. pp. 572–586. doi:10.1007/978-3-642-04081-8_38. ISBN 978-3-642-04080-1.
30. IEEE (1998). IEEE standard for software test documentation. New York: IEEE. ISBN 978-0-7381-1443-9.
31. Strom, David (July 1, 2009). "We're All Part of the Story". Software Test & Performance Collaborative. Archived from the original on August 31, 2009.
32. Griffiths, M. (2005). "Teaching agile project management to the PMI". Agile Development Conference (ADC'05). ieee.org. pp. 318–322. doi:10.1109/ADC.2005.45. ISBN 0-7695-2487-7. S2CID 30322339.
33. Willison, John S. (April 2004). "Agile Software Development for an Agile Force". CrossTalk. STSC (April 2004). Archived from the original on October 29, 2005.
34. An example is Mark Fewster, Dorothy Graham: Software Test Automation. Addison Wesley, 1999, ISBN 0-201-33140-3.
35. "stop29119". commonsensetesting.org. Archived from the original on October 2, 2014.
36. Paul Krill (August 22, 2014). "Software testers balk at ISO 29119 standards proposal". InfoWorld.
37. Kaner, Cem (2001). "NSF grant proposal to 'lay a foundation for significant improvements in the quality of academic and commercial courses in software testing'" (PDF).
38. Kaner, Cem (2003). Measuring the Effectiveness of Software Testers (PDF). STAR East. Retrieved January 18, 2018.
39. McConnell, Steve (2004). Code Complete (2nd ed.). Microsoft Press. p. 29. ISBN 978-0735619678.

40. Bossavit, Laurent (November 20, 2013). The Leprechauns of Software Engineering: How folklore turns into fact and what to do about it. Chapter 10: leanpub.

41. Tran, Eushiuan (1999). "Verification/Validation/Certification" (coursework). Carnegie Mellon University. Retrieved August 13, 2008.