



Next Generation Cloud Architectures

*Konstantinos M. Giannoutakis,
Minas Spanopoulos-Karalexidis,
Christos K. Filelis Papadopoulos, and Dimitrios Tzovaras*

Abstract This chapter discusses the evolution of cloud infrastructures and the impact of IoT. We discuss the need for heterogeneous resources integration in resource provisioning and the necessity to find the golden ratio between the cloud, fog and edge for optimal user experience. Complexity across the cloud-to-thing continuum is escalating rapidly. One way that cloud computing can adapt is to incorporate self-management and self-organization techniques and methods to preserve service availability and avoid user Quality of Service and Quality of Experience violations.

Keywords Cloud computing • Cloud architecture • Resource provisioning • Self-management • Self-organization

K. M. Giannoutakis (✉) • M. Spanopoulos-Karalexidis • D. Tzovaras
Information Technologies Institute, Centre for Research and Technology Hellas,
Thessaloniki, Greece
e-mail: kgiannou@iti.gr; mspanopoulos@iti.gr; dimitrios.tzovaras@iti.gr

C. K. Filelis Papadopoulos
Department of Computer Science, University College Cork, Cork, Ireland
e-mail: christos.papadopoulos-filelis@cs.ucc.ie

© The Author(s) 2020

T. Lynn et al. (eds.), *The Cloud-to-Thing Continuum*, Palgrave
Studies in Digital Business & Enabling Technologies,
https://doi.org/10.1007/978-3-030-41110-7_2

2.1 INTRODUCTION

Cloud computing provides users with the potential to perform computing tasks utilizing resource physically distant to them. It offers virtually unlimited capabilities regarding storage, memory and processing units that augment IoT devices and components with limited computation capabilities due to form factors. However, this convergence of the cloud and the edge, provides significant issues, not least complexity at several orders of magnitude higher than the past. Forecast increases in the numbers of devices lead to a humongous escalation in generated data exacerbates this complexity. While Cloud-IoT cooperation seems perfect, managing the complexity across the a continuum can induce potential violations in Quality-of-Service (QoS) and Quality-of-Experience (QoE) user requirements. Thus, an evolution in conventional cloud infrastructure is necessary along with other infrastructure approaches such as fog and edge computing.

The remainder of this chapter is organized as follows. Section 2.2 provides a brief overview of the centralized cloud model and the significant impact of IoT on conventional cloud computing. Section 2.3 illustrates the changes in the centralized cloud paradigm. Next, decentralized cloud models are presented followed by a discussion of research opportunities and directions specifically focusing on the need for improved resource provisioning, support for resource heterogeneity, and self-management and self-organization patterns that cloud can adopt to address complexity. Concluding remarks are summarized in Sect. 2.6.

2.2 CENTRALIZED CLOUD COMPUTING MODEL

In the cloud computing era, the substance of a computer can no longer be limited to its physical “box” shape. Thus, a computer in the cloud has to introduce the capability of dynamically adjusting the required physical resources (i.e. processors, memory, storage, network bandwidth) based on any potential occasion. These resources can be distributed across physical servers and virtual machines creating a pool of available resources (Bhavani and Guruprasad 2014). This “cloud computer” offers increased performance levels, while dramatically reducing response time and operational costs.

2.2.1 Defining Cloud Computing

As defined in Chap. 1, cloud computing is

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. (Mell and Grance 2011, p. 2)

Conventional single provider infrastructures hosting cloud services on data centers offer a lot of benefits but also hide many challenges (Trilochan and Verma 2017). A large data center's energy consumption is high just to keep it operational and like any other centralized computing model, in case of a failure (single-point failure) the resulting issues would be adverse. Another issue is that required data may have to be transferred and stored to separate places, rather than the source, because data centers are often geographically distant from the application users. Thus, exchange of sensitive or personal data is considered critical for applications. For these reasons, alternate cloud infrastructure models which handle failures and use back up data centers have been introduced in recent years. Example models using cloud infrastructure rather than data centers are multi-cloud, micro cloud and cloudlet, ad hoc and heterogeneous clouds. The main cloud computing actors, characteristics and models are depicted in (Table 2.1).

A multi-cloud utilizes resources from multiple providers, thus making it possible to host large number of applications. Another form of multi-cloud is a federated cloud. This model is a federation of clouds from different cloud providers providing the potential to make applications portable, meaning that data from it or even the whole application can migrate from one cloud to another with the federation.

To incorporate resources located outside the cloud data centers. Micro cloud and cloudlet approaches offer a more decentralized computing infrastructure, as they are located closer to the edge of the network, thus minimizing transfer cost and communication latency as they are closer, sometimes a mere single hop, to the users and the generated data. Nevertheless, their computing power is much less than a conventional cloud infrastructure as they are composed of less powerful processors and are significantly smaller in size. An ad hoc cloud is an elastic infrastructure

Table 2.1 Cloud computing actors, essential characteristics, service models and deployment models. (Adapted from Liu et al. (2011) and Mell and Grance (2011))

<i>Actors in cloud computing</i>	
Consumer	A person or organization that maintains a business relationship with, and uses service from, cloud providers.
Cloud provider	A person, organization, or entity responsible for making a service available to interested parties. Cloud auditor A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.
Cloud auditor	A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.
Cloud broker	An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between cloud providers and cloud consumers.
Cloud carrier	An intermediary that provides connectivity and transport of cloud services from cloud providers to cloud consumers.
<i>Essential characteristics</i>	
On-demand self-service	Consumers can unilaterally provision computing capabilities as needed automatically without requiring human interaction with the cloud provider.
Broad network access	Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms and interfaces (e.g. devices).
Resource pooling	The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.
Rapid elasticity	Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outwards and inwards to meet demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
Measured service	Cloud systems automatically control and optimise resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service. Resource usage can be monitored, controlled, and reported, providing transparency to the service provider and the consumer.
<i>Service models</i>	
Software as a service	The capability provided to a consumer to use a provider's applications running on a cloud infrastructure and accessible by client interface.

(continued)

Table 2.1 (continued)

Platform as a service	The capability provided to a consumer to deploy onto the cloud infrastructure consumer-created or acquired applications created using development technologies provided by the provider.
Infrastructure as a service	The capability provided to a consumer to provision computing resources to deploy and run arbitrary software such as operating systems and applications.
<i>Deployment models</i>	
Private cloud	The cloud infrastructure is provisioned for exclusive use by a single organisation comprising multiple consumers. Ownership, management, and operation of the infrastructure may be done by one or more of the organisations in the community, by a third party, or a combination of both, and it may exist on or off premise.
Community cloud	The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organisations that have shared concerns. Ownership, management, and operation of the infrastructure may be done by one or more of the organisations in the community, by a third party, or a combination of both, and it may exist on or off premise.
Public cloud	The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organisation, or some combination of them. It exists on the premises of the cloud provider.
Hybrid cloud	The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability.

which deploys underutilized or spare resources of data centers, servers and any other mobile edge device in contribution to ad hoc computing (McGilvary et al. 2015). Finally, an heterogeneous cloud consists of different types of processing units in the infrastructure, mostly composed of accelerators such as Graphical Processing Units (GPUs), Intel Xeon Phi, Field-Programmable Gate Arrays (FPGAs) and others, offered in the form of VMs or containers. The problem is that, building a framework that utilizes and provides provisioning of these resources is still a challenging and difficult task (see e.g. the Horizon 2020 CloudLightning project (Filelis-Papadopoulos et al. 2018; Lynn et al. 2016; Xiong et al. 2017)).

2.2.2 *The Impact of IoT on Conventional Cloud Architectures*

The evolution of the Internet of Things has had a significant impact on cloud computing generally and stretching the limitations of conventional cloud architecture. The number of connected devices is increasing exponentially with estimations of dozens of billions of “things” going live in coming years (Bittencourt et al. 2018; Gubbi et al. 2013).

The connected devices (things) are extremely heterogeneous in form and function. The trend to combine and integrate more and more sensors to each and every device escalates the complexity of resource management. As a consequence of connecting these sensors to the Internet, large volumes of data are being generated in unprecedented volumes, variety and velocity, the so-called Big Data (El-Seoud et al. 2017). This data is currently transferred and stored in the cloud in a centralized manner. Data transfer, especially in these volumes, is extremely expensive and retards computational performance. Thus, a more decentralized solution is required where data analysis could take place before transfer and storage. Emerging computing paradigms to support this decentralized or distributed cloud are discussed in detail in Sect. 2.4. Furthermore, most cloud infrastructures scale horizontally across multiple nodes in a data center or more thus making it necessary to develop cloud models that can scale vertically from low end processors to data center nodes.

The net result is that an enormous amount of data needs to be transmitted over the network, stored and/or processed by the receivers in an efficient way. The heterogeneity of the connected devices is immense and can be discrete in many levels, such as computing performance, storage and network requirements, communication protocols, energy consumption amongst others. This heterogeneity is adapted to many applications running on the IoT connected devices and as their numbers increase, so do their requirements accordingly, making it more complex and far more difficult to cope with the extensive needs that the computing system has to be able to accommodate.

2.3 CHANGES TO CENTRALIZED CLOUD COMPUTING MODEL

More recently, application deployment in the cloud has been a challenge for the various providers in the network. To address this challenge, cloud architectures, infrastructure and deployment has evolved. This subsection

provides a brief summary of a number of changes applied to these fields, including the evolution of service provisioning, the timeline of service models and the addition of new resources and workloads over time.

Monolithic Architectures are those architectures where the application is composed of a single program or platform, typically providing a user interface and data access through a database. Rationally, building and deploying an application of this model type is easy, especially when the application requirements are simple or the project is small. Nevertheless, the real challenge and the difficulties start to grow exponentially, when the application needs to scale up. Any adjustments, or any development and testing that needs to be made, initiates rebuilding of the whole code/project as it is a single unit or component. This could be extremely time consuming and potentially harmful, especially in large projects.

Service Oriented Architectures (SOA) are based on services to be provided in order to develop software product. Services are built to work in an orchestrated manner to modularize the system and provide a total service as a group. It is more challenging to divide the application in to multiple services, but it enables greater flexibility, extensibility and reusability of existing services for multiple use cases. The grouped services can communicate with each other by exchanging messages or events through APIs, which trigger the reaction of the corresponding services. The benefits of this model are application modularity, service reusability and enhanced security in the (re)building process and development of the application. A major disadvantage is the complexity in orchestrating all the services from a centralized component, especially when the project is complex and the components are huge.

Microservices introduced a solution for the gaps in the SOA approach. This approach divides applications in to more granular components by distributing them into small independent services. Each service implements and attends to separate business functions and capabilities to maintain independency from other services. They are the mainly deployed in an automated manner, through a container and communicating through RestAPIs, thus making the impact of programming language and data management techniques insignificant. This allows microservices to be easily deployed in the cloud, offering great reusability and minimal or no centralized management and orchestration. Essentially, microservices offer even more modularity than SOA and are more conducive in complex and large projects while, at the same time, providing independency in service

development and maintenance, thus enhancing security in business products.

In addition to architecture modeling, different approaches have emerged regarding service modeling. In addition to conventional SaaS, IaaS, PaaS models, a new approach in virtualization is *bare metal or native virtualization* (Scarfone et al. 2011). Here, the hypervisor can run directly on the underlying hardware of the provider without a host operating system. Bare metal offers more security, providing that the hypervisor itself is well-secured. As the hypervisor is placed directly over the hardware, there is no host OS thus it cannot be harmed. This model is mostly used for servers in the cloud. Nevertheless, the hardware provided is limited so that a hypervisor does not consume the total available resources.

Hosted Virtualization is where the hypervisor runs over the host OS. The difference between hosted virtualization and bare metal is that the hypervisor is provided with more virtual resources but, on the other hand, the potential to harm the host OS is significantly increased.

Function as a Service (FaaS) is synonymous with serverless computing. Basically, FaaS enhances the microservices model development. During the development process, server operations are not taken into account, as services are hosted externally. Compared to conventional cloud models, where at least one server is utilized, FaaS triggers a server only when a function is conducted, executes the expected operations and then terminates. The major advantages of this model are increased scalability and independency of the applications and lower costs. As costs are based only on per used functionality, expenses from inactive resources are eliminated. A major disadvantage is the reduced transparency as FaaS is managed externally.

Composable architecture and Infrastructure is an approach used for provisioning both physical and virtual resources. It is an application-centric approach which greatly enhances operational performance with dynamic and flexible on-demand provision of resources. With the ability to manage a great variety of resources, it can easily scale up at an even greater extent than a traditional infrastructure. The flexibility composable infrastructure offers with the on-demand provision of resources, both hardware and software ones. Different resources can be provisioned independently (aggregated, disaggregated) and generally adjusted, based on their type (compute, storage, fabric) which is a major advantage of this approach.

Finally, until recently workloads and tasks on the cloud were largely loosely coupled. As such, the use of distributed memory was enough for the execution of any task, while the need for data rate and bandwidth was low. Thus, in a loosely coupled multiprocessor system, a simple message transfer system was enough to connect all the modules in the network. As the need for larger data rates and bandwidths are becoming more and more demanding especially with the addition of the heterogeneous resources in everyday tasks and applications, *tightly coupled workloads* in the essence of High Performance Computing (HPC) have been introduced in to cloud computing. These tasks are continuously more dependent on each other, utilize a common shared memory and the need for data rate and transfer is huge. A tightly coupled workload requires inter-process communication patterns that rely on high bandwidth with low latency in order to maintain optimal performance. This could lead to significantly reduced number of executed tasks, considering the demanding requirements.

2.4 DECENTRALIZED CLOUD COMPUTING MODEL

As the number of connected devices escalates rapidly, the generated data reaches unprecedented levels, and complexity increases as more and more sensors are integrated into user devices located at the edge of the network, conventional centralized data centers in the cloud can no longer provide an efficient and sustainable solution. It is necessary to provide resources and computational power closer to the edge users.

2.4.1 *Fog Computing*

Fog computing is a decentralized computing infrastructure which is used particularly as a complement to cloud computing. It leverages the compute resources at the edge network and brings the computational processing closer to the data source by offloading workload to edge nodes from cloud data centers. The network nodes near the edge providing these resources are called fog nodes. Overall, any device with computing, storage and network connectivity can constitute a fog node, for example switches and routers, industrial controllers, embedded servers and video surveillance cameras. A major benefit of fog computing is the reduction in application latency and as a result the improvement in QoS and QoE for users. Its first level usually lays a single hop away from the edge and is an extension of edge computing. Fog nodes can be organized in clusters,

either vertically or horizontally depending on requirements, and can be either physical or virtual components that are tightly coupled with the end-devices. This introduces the need to be geographically aware in order to enhance performance and minimize end-user latency. Fog computing offers, similarly to conventional clouds, the aforementioned architectural service models (SaaS, PaaS and IaaS) and the deployment models (private, community, public, hybrid).

2.4.2 *Mobile Edge Computing*

Mobile Edge Computing, also referred to as just edge computing, takes place only on the edge of the network. Processing is executed closer to the data source and eliminates the costly data transfer to a remote data center or cloud (Ahmed and Ahmed 2016). This significantly improves user QoE as similar to fog computing, there is considerable network latency reduction and bandwidth consumption by the mobile subscribers.

2.4.3 *Volunteer Computing*

Volunteer Computing refers to a form of ad hoc cloud and cloudlets composed mostly of spare resources from users' computers or devices generally (Durrani and Shamsi 2014). The most common case where a VM is utilized in this manner is for social networks where users share their heterogeneous resources in the form of the aforementioned ad hoc cloud. The most reliable ones are rewarded and occasionally rewarded by payment for their contribution. Two significant challenges have to be overcome in order to fully benefit from volunteer computing. The first one is the difficulty to set up a reliable and functional virtualized environment considering the obstacles the heterogeneous resources and the ad hoc nature of this model. The second one is the privacy and security concerns users might have and need to be overcome in order to make them offer their spare resources and become volunteers.

2.4.4 *Serverless Computing*

Serverless Computing involves building, running and providing applications and services without taking into consideration the server side. "Serverless" does not mean that there is no server usage but rather the main focus on the application itself and the virtual resource provisioning

in the hosting VM, rather than what happens on the physical infrastructure (Hellerstein et al. 2018). Serverless Computing is synonymous with FaaS and event-based programming as the execution of an application will be executed only when necessary and not all the time, thus meaning that an event can trigger the execution of a function or more than one function concurrently.

2.4.5 *Software-Defined Computing*

Software-defined computing separates the control plane from data plane and the utilized hardware in the network from the data control traffic components (Badotra and Singh 2017). This approach can also be adapted to other QoS metrics, rather than networking, such as storage and compute and resources located outside the data centers. It allows network managers to create a flexible, scalable pool of resources that are controlled by a software-driven cloud platform.

2.4.6 *Mist Computing*

Mist computing is a lightweight and elementary form of fog computing which resides even closer to the edge network and devices thus minimizing even more end-user latency (Iorga et al. 2018). It is comprised of dedicated nodes, with specialized capabilities but with less computational resources than fog. It is usually implemented as an extra layer of fog computing, closer or even on same layer with end-devices, but the existence of it is not considered mandatory.

2.5 RESEARCH DIRECTIONS AND OPPORTUNITIES

Cloud computing and associated technical evolutions is a solution to many IoT challenges but there are still obstacles that need to be overcome. The IoT-Cloud combination has to be able to provide accurate real-time processing and resource and service provisioning (Biswas and Giaffreda 2014). As already mentioned, these resources can be highly heterogeneous and require dynamic provisioning thus escalating complexity.

We identify five significant challenges for IoT-Cloud:

1. *Interoperability*—applications on a platform should be able to amalgamate services and infrastructure from another Cloud-IoT platform.

2. *Security and Privacy*—personal and sensitive user data are subjected to high risk while many users access public and ad hoc clouds. In some instances, personal data may have to be stored closer to the users/devices in order to facilitate computing and processing on the edge or fog layer. Furthermore, there is also the challenge to develop suitable and reliable encryption-decryption mechanisms and algorithms which could scale among distributed clouds and at the same time reduce energy consumption. This is discussed further in Chap. 6.
3. *Portability*—efficient migration of each application and service has to be supported from platform to platform and follow the users' traces and paths in the network.
4. *Reliability*—establishing real-time communication between objects and applications with high connectivity and accessibility
5. *Virtualization*—the potential to provision resources and provide access to heterogeneous resources and hardware such as GPUs, FPGAs, etc.

2.5.1 Resource Provisioning and Heterogeneous Resources

A central objective of cloud computing is to provide availability and share essential resources to each user to fulfill the QoS demands in respective SLAs (Parikh et al. 2017). Resource provisioning can be categorized into three main types:

1. *Static provisioning*—workloads and resource demands of an application are predefined or easily predictable.
2. *Dynamic provisioning*—is applied to applications that can adjust their demands during service, thus further resource allocation, or deallocation, may be needed or even migrate the application to another VM on-the-fly.
3. *Self-provisioning*—the user/application purchases specific resources from the cloud provider and can utilize them at will.

Efficient provisioning techniques target QoS metric improvement and reduce response times, SLA violations and power consumption. Furthermore, the cloud has to be operational even in case of a failure, that is being able to service user requests without making the failure noticeable to the outer world. This can be accomplished with optimal and novel

remediation techniques in the cloud infrastructure (Ostberg et al. 2017). Similarly, resource provisioning in fog cells has to provide optimal utilization of fog resources and reduce offloading to cloud which dramatically increases costs (Skarlat et al. 2016). Additionally, data originated in the fog should be used in the particular fog and minimize data transfer between fog and cloud which significantly augments communication delays.

Evolving cloud infrastructures have introduced more heterogeneous resources, services and workloads while also augmenting complexity in resource provisioning. Heterogeneity can be discriminated in two types, machine heterogeneity and workload heterogeneity (Zhang et al. 2013). The addition of accelerators, such as GPUs and FPGAs, offer less powerful general purpose processors than a CPU but in distinctively greater numbers. These provide increased computational power and performance while reducing costs and improving energy efficiency. This is due to the potential of utilizing these many-core and multi-core systems for specific and specialized tasks like HPC. In this way, the overall computational and energy efficiency are noticeably increased. Workload analysis along with workload, machine and task heterogeneity can be characterized and classified (Zhang et al. 2014). A number of research studies have been undertaken in this domain including Kollenstart et al. (2018), Xiong et al. (2017), and Dai et al. (2015).

2.5.2 *Self-management, Self-organization Approach*

The ultimate goal of cloud computing as an *aaS (IaaS, PaaS, SaaS), is obviously to provide efficient services to users and meet their QoS requirements. Nevertheless, there are many occasions where failures occur and SLA violations occur. Additionally, massive changes in the cloud structure can also introduce significant performance challenges, for example flash crowd events, significant adjustment in the number of nodes. These challenges necessitate the development of adaptive patterns that can also address the increasing complexity of the cloud. Self-management techniques provide an approximate solution to the escalating complexity, as they tend to interact with both internal and external stimulus without any human intervention. They can be classified to four self-management aspects:

1. *Self-configuration*—the system manages the deployment of newly inserted nodes or disappearing ones by itself.
2. *Self-optimization*—a node or a network link hits its capacity limits, it has to be able to offload some of the tasks to another (optimal constituent) node/link,
3. *Self-protection*—the ability to protect itself against third party attacks, such as Distributed Denial-of-Service (DDoS), and
4. *Self-healing*—in the case of a failure, the active and executing applications have to be migrated and become available again elsewhere.

Generally, any cloud system has to guarantee its capacity to adapt and address the aforementioned challenges, that is continuous operation under any circumstances, load balancing, security, interoperability and energy efficiency. In order to achieve this, several patterns may be used based on service components and autonomic managers including self-organization and self-management (Xiong et al. 2017), P2P negotiation (Puviani et al. 2013) and centralized autonomic managers (Puviani and Frei 2013).

2.5.3 Separation of Concerns

Another major challenge derived from cloud computing utilization is the distinction between consumer-aware and provider-aware services respectively, along with a service interface establishment between them. This issue was first introduced by the CloudLightning project arising many considerable questions about services' origin and handling (Lynn et al. 2016; Xiong et al. 2017). Separation of Concerns the need to distinguish the exact services that users and providers should be concerned with respectively. Consumers should only be concerned with *what* they want to do and accomplish, and providers with *how* that could be done and provided to the user. A successful interface establishment between those two actors can lead to minimal direct consumer interaction with provider's infrastructure, thus allowing full control to the provider. Furthermore, this results in the assumption that various service implementations should already exist and the consumer himself does not have to be an expert and develop them. Additionally, physical resources provided by the infrastructure should not be consumer-aware and yet there may be several diverse implementations to meet specific service demands. These implementations can differ in hardware type and could be characterized by different price and performance attributes. Thus, consumers should be able to

differentiate and discriminate between these implementations and choose the appropriate one that meets their service delivery attributes. In address such a difficult challenge, and to find the golden ratio between consumer and provider services, additional research is necessitated.

2.6 CONCLUSION

As a result of the Internet of Things and related technologies, cloud computing is experiencing a phase of rapid evolution. This chapter described how the Cloud-IoT convergence is moving cloud computing from a centralized model to a more distributed one and from a commoditized homogenous cloud to a specialized, heterogeneous one. New techniques and approaches are needed to exploit these new evolutions in cloud computing and to support the Internet of Things.

REFERENCES

- Ahmed, A., and E. Ahmed. 2016. *A Survey on Mobile Edge Computing*. 2016 10th International Conference on Intelligent Systems and Control (ISCO).
- Badotra, S., and J. Singh. 2017. A Review Paper on Software Defined Networking. *International Journal of Advanced Research in Computer Science* 8 (2): 17.
- Bhavani, B.H., and H.S. Guruprasad. 2014. Resource Provisioning Techniques in Cloud Computing Environment: A Survey. *International Journal of Research in Computer and Communication Technology* 3 (3): 395–401.
- Biswas, A.R., and R. Giaffreda. 2014. *IoT and Cloud Convergence: Opportunities and Challenges*. 2014 IEEE World Forum on Internet of Things (WF-IoT). IEEE.
- Bittencourt, L., R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. DaSilva, C. Lee, and O. Rama. 2018. The Internet of Things, Fog and Cloud Continuum: Integration and Challenges. *Internet of Things* 3–4: 134–155.
- Dai, W., H. Chen, and W. Wang. 2015. *RaHeC: A Mechanism of Resource Management for Heterogeneous Clouds*. 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, 2015 IEEE 12th International Conference on Embedded Software and Systems, 40–45. IEEE.
- Durrani, M.N., and J.A. Shamsi. 2014. Volunteer Computing: Requirements, Challenges, and Solutions. *Journal of Network and Computer Applications* 39: 369–380.

- El-Seoud, S.A., H.F. El-Sofany, M. Abdelfattah, and R. Mohamed. 2017. Big Data and Cloud Computing: Trends and Challenges. *International Journal of Interactive Mobile Technologies (ijIM)* 11 (2): 34–52.
- Filelis-Papadopoulos, Christos K., Konstantinos M. Giannoutakis, George A. Gravvanis, and Dimitrios Tzovaras. 2018. Large-scale Simulation of a Self-organizing Self-Management Cloud Computing Framework. *The Journal of Supercomputing* 74 (2): 530–550.
- Gubbi, J., R. Buyya, S. Marusic, and M. Palaniswami. 2013. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems* 29 (7): 1645–1660.
- Hellerstein, J.M., J. Falciro, J.E. Gonzalez, J. Schlier-Smith, V. Sreekanti, A. Tumanov, and C. Wu. 2018. Serverless Computing: One Step Forward, Two Steps Back. *arXiv preprint arXiv:1812.03651*.
- Iorga, M., L. Feldman, R. Barton, M.J. Martin, N. Goren, and C. Mahmoudi. 2018. *Fog Computing Conceptual Model*. U.S. Department of Commerce, Information Technology Laboratory. Gaithersburg: National Institute of Standards and Technology Special Publication 500-325.
- Kollenstart, M., E. Harmsma, E. Langius, V. Andrikopoulos, and A. Lazovik. 2018. Adaptive Provisioning of Heterogeneous Cloud Resources for Big Data Processing. *Big Data and Cognitive Computing* 2 (3): 15.
- Liu, F., J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf. 2011. *NIST Cloud Computing Reference Architecture*. U.S. Department of Commerce, Information Technology Laboratory. Gaithersburg: National Institute of Standards and Technology Special Publication 500-292.
- Lynn, T., H. Xiong, D. Dong, B. Momani, G. Gravvanis, C. Filelis-Papadopoulos, A. Elster, et al. 2016. Cloudlightning: A Framework for a Self-organising and Self-managing Heterogeneous Cloud. *CLOSER*: 1 (2): 333–338. <https://dl.acm.org/doi/10.5220/0005921503330338>
- McGilvary, G.A., A. Barker, and M. Atkinson. 2015. *Ad hoc Cloud Computing*. 2015 IEEE 8th International Conference on Cloud Computing (CLOUD). New York: IEEE.
- Mell, P., and Grance, T. 2011. The NIST definition of cloud computing, Recommendations of the National Institute of Standards and Technology, NIST Special Publication 800-145.
- Ostberg, P.-O., J. Byrne, P. Casari, P. Eardley, A.F. Anta, J. Forsman, J. Kennedy et al. 2017. *Reliable Capacity Provisioning for Distributed Cloud/Edge/Fog Computing Applications*. 2017 European Conference on Networks and Communications (EuCNC), 1–6. IEEE.
- Parikh, S.M., N.M. Patel, and H.B. Prajapati. 2017. Resource Management in Cloud Computing: Classification and Taxonomy. *arXiv preprint arXiv:1703.00374*.

- Puviani, M., and R. Frei. 2013. *Self-management for Cloud Computing*. Science and Information Conference, London.
- Puviani, M., Cabri, G., and Zambonelli, F. (2013, July). *A Taxonomy of Architectural Patterns for Self-adaptive Systems*. In Proceedings of the International C* Conference on Computer Science and Software Engineering, 77–85.
- Scarfone, K., M. Souppaya, and P. Hoffman. 2011. *Guide to Security for Full Virtualization Technologies*. U.S. Department of Commerce, Computer Security. Gaithersburg: National Institute of Standards and Technology Special Publication 800–125.
- Skarlat, O., S. Schulte, M. Borkowski, and P. Leitner. 2016. *Resource Provisioning for IoT Services in the Fog*. 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA), 32–39. IEEE.
- Trilochan, and A. Verma. 2017. Cloud Computing: Evolution and Challenges. *International Journal of Engineering Science and Computing* 7 (4): 10197–10200.
- Xiong, H., D. Dong, C. Filelis-Papadopoulos, G.G. Castane, T. Lynn, D.C. Marinescu, and J.P. Morrison. 2017. *CloudLightning: A Self-organized Self-managed Heterogeneous Cloud*. Proceedings of the Federated Conference on Computer Science and Information Systems, 749–758. ACSIS.
- Zhang, Q., M.F. Zhani, R. Boutaba, and J.L. Hellerstein. 2013. *HARMONY: Dynamic Heterogeneity-Aware Resource Provisioning in the Cloud*. 2013 IEEE 33rd International Conference on Distributed Computing Systems, 510–519. IEEE.
- . 2014. Dynamic Heterogeneity-Aware Resource Provisioning in the Cloud. *IEEE Transactions on Cloud Computing* 2 (1): 14–28.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copy-right holder.

