

A View at Desktop Clouds

Abdulelah Alwabel, Robert Walters and Gary Wills

Electronics and Computer Science School, University of Southampton, Southampton, U.K.

Abstract. Cloud has emerged as a new computing paradigm that promises to move into computing-as-utility era. Desktop Cloud is a new type of Cloud computing. It merges two computing models: Cloud computing and volunteer computing. The aim of Desktop Cloud is to provide Cloud services out of infrastructure that is not made for this purpose in order to reduce running and maintenance costs. This paper discusses this new type of Cloud by comparing it with current Cloud and Desktop Grid models. It, also, presents several research challenges in Desktop Cloud that require further attention.

1 Introduction

Desktop Clouds represent a new direction of providing Cloud services based on non-dedicated resources. The resources can be any form of computing devices such as PCs, laptops ...etc. The new type attempts to combine two computing models, namely Cloud computing and Volunteer computing in order to form a Cloud that provides services for less or no cost. Throughout this paper, Traditional Cloud (TC) refers to a Cloud that relies on dedicated resources to provide services, whereas Desktop Cloud (DC) refers to a Cloud that relies on non-dedicated resources. Amazon Cloud, for instance, is a Traditional Cloud.

The remaining of this paper is organised as follows. First, the paper starts by giving an overview about the meaning of Desktop Clouds. In addition, the advantages of Desktop Clouds are presented. A brief comparison study between Desktop Clouds, Traditional Clouds and Desktop Grids is presented in order to study similarities and differences. Finally, the paper finishes by discussing research issues in Desktop Cloud.

2 Desktop Clouds

The success of Desktop Grids motivates the idea of harnessing idle resources to build Desktop Clouds. Hence, the term Desktop comes from Desktop Grids because both of Desktop Clouds and Desktop Grids are based on Desktop PCs and laptops etc. Similarly, the term Cloud comes from Cloud as Desktop Cloud aims to provide services based on the Cloud business model. Several synonyms for Desktop Cloud have been

used, such as Ad-hoc Cloud, Volunteer Clouds and Non-Dedicated Clouds. The literature shows that very little work has been undertaken in this direction.

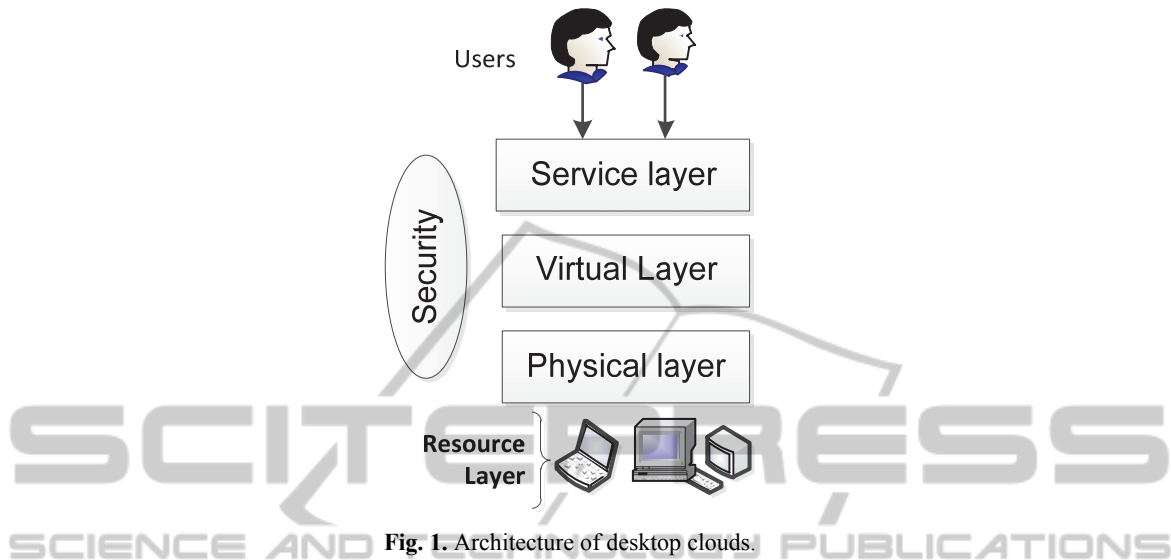


Fig. 1. Architecture of desktop clouds.

The overview architecture of Desktop Clouds is depicted in fig. 1. The architecture is consisted of several layers. The users contact the service layer in order to submit their demands. The physical layer is responsible of managing physical nodes that are aggregated the resource layer. The virtual layer plays a curtail role in terms of isolating Clients request from the physical nodes via virtualisation. Users are assigned virtual machines that are located in physical machines. Physical machines can be connected by LAN or WLAN.

Ad-hoc Cloud is the idea of harvesting distributed resources within an organisation to form a Cloud [1]. Nebula [2, 3] is a project aiming to exploit distributed resources in order to create a volunteer Cloud which offers services free of charge. Cloud@home [4, 5] is a project representing the @home philosophy in Cloud computing. The goal of Cloud@home is to form a new model of Cloud computing contributed to by individual users over the Internet. In addition to that, Cern¹ has announced an initiative to move their Desktop Grid project, which is called LHC@home, toward the Cloud [6]. It is suggested that non-dedicated resources can be exploited by Cloud providers in case their local infrastructure cannot meet requests by consumers at peak times [7].

2.1 Scenario

Suppose a group of universities wishes to benefit from its computing resources to form a Cloud. The resources range from PCs to servers etc, each of them is called a

¹ the European Organization for Nuclear Research

Cloud node. A node can join the Cloud when it becomes idle. This scenario is motivated by Condor [8]. Users in Desktop Cloud submit their request to acquire services with requirements as it is stated in the service level agreement between a client and the Cloud interface. The requests are processed in the virtualisation layer on top of Cloud physical nodes. The virtualisation isolates the guest operating system from the host physical machine. The isolation improves security and prevents unauthorised access between two parties.

Another scenario that can be considered is a universal Desktop Cloud which allows people to contribute their own computing resources to be used by Cloud clients [9]. This example can be considered as public Desktop Cloud. The people are asked to contribute their machines in order to form a Desktop Cloud. People can be stimulated to participate in DC to serve science within research communities.

Table 1. Desktop Cloud vs. Traditional Cloud.

Feature	Desktop Cloud	Traditional Cloud
Elasticity	√	√
Virtualisation	√	√
Idle Resources	√	X
Ease of Use	√	√

3 Desktop Clouds vs. Traditional Clouds

This section clarifies Desktop Clouds further by comparing them with related areas: Traditional Clouds, Grids and Desktop Grids. There are basic differences between Desktop Cloud and Traditional Clouds as it is shown in Table 1. Elasticity means that a consumer can acquire computing services. Then the user can scale up or down according to their needs. Both Traditional Cloud and Desktop Cloud rely heavily on virtualisation. The ease of use principle means that users can use a specific service without making a lot of changes to their work. Both Traditional Clouds and Desktop Clouds let their users harness services without making significant changes to their code. However, there are several differences between Desktop and Traditional Clouds. Firstly, the infrastructure of DC is made of resources that are non-dedicated, i.e. not dedicated to be part of Cloud infrastructure. In the contrary, the infrastructure of TC is made of a huge number of dedicated computing resources. Secondly, the resources in DC are highly distributed across the globe, whereas they are limited in TC to several locations in data centres. Furthermore, nodes in DC are highly volatile due to the fact that they can be down unexpectedly without prior notice. Resource high volatility has negative impact on availability and performance [10]. Although, resources in both TC and DC are heterogeneous, they are even more heterogeneous and dispersed in DC. Virtualisation plays a key role in Desktop Clouds which makes it different from other large scale systems.

Desktop Clouds have some advantages over Traditional Clouds. Firstly, Traditional Clouds have a negative impact on the environment since their data centres consume massive amounts of electricity [11]. The second advantage is cost effectiveness of Desktop Clouds for both Cloud services providers and their consumers. For

service provider, there is no need to build new data centres to meet the increasing demands of the future. Consumers will get their services at lower prices, if not free compared to Traditional Clouds. Also, Desktop Cloud helps in reducing energy consumption since it utilises already-running undedicated resources which would otherwise remain idle. Some studies show that the average percentage of local resources being idle within an organisation is about 80% [12]. Furthermore, Traditional Clouds are formed from a limited number of data centres located around the globe. Therefore, they are inefficient in terms of data mobility and pay little attention to the location of clients [3]. Finally, Traditional Clouds are centralised, which leads to the potential that there could be a single point of failure issue if a Cloud service provider goes out of the business. In contrast, Desktop Clouds manage and offer services in a decentralised manner.

Table 2. Desktop Cloud vs. Traditional Cloud.

Feature	Desktop Cloud	Desktop Grid
Elasticity	√	X
Virtualisation	√	X
Idle Resources	√	√
Ease of Use	√	X

4 Desktop Cloud vs. Desktop Grid

Desktop Clouds can be confused with other distributed systems, especially Desktop Grids. Table 2 shows a comparison between DC and Desktop Grid. Both Desktop Clouds and desktop Grids serve the same goal that is exploiting idle computing resources denoted by the public. However, Desktop Grid cannot offer services to clients in elasticity way as in Desktop Clouds. Elasticity means that users can require more computing resources in short term. Users, in Desktop Grids, are expected to know in-depth details about the middleware used in order to be able to harness the offered services [13]. Desktop Grids do not employ virtualisation to isolate users from the actual machines. People who wish to contribute their computing machines need to install a specific software in order to join a Desktop Grid.

5 Research Challenges

This section discusses several research issues that need further attention. Some of these challenges are inherited from Cloud computing while others are driven by the nature of used resources being highly volatile.

5.1 Security

Security is one of the major concerns that prevent organisations from moving to the cloud [14]. Ristenpart et al. show that an attacker can uncover the actual location of a

particular virtual machine (VM) [15]. Then, a cross-VM side channel attack can reveal critical information about the targeted VM by placing a malicious VM on the same physical machine.

More worries arise in DCs where both consumers and contributors are from the public. Therefore, security can be a major issue in this context. In addition to the previous threats presented in the cloud, both consumers and contributors take on risk themselves when they join a DC. A contributor can put his own data at risk by allowing access to a virtual image located in his machine. Likewise, consumers are vulnerable to malicious contributors. Nodes in DC are more likely to be vulnerable to outside attacks due to weaknesses in local antivirus software and firewalls.

Virtualisation can be vital in order to isolate the host completely from guest operating systems and, thus, prevent any unwanted access from either party. Trust mechanisms can be employed in this matter. For example, a DC can maintain a behaviour table which contains information about both consumers and contributors. The table can be used to decide which parties are trustworthy enough to join the cloud. Furthermore, VCCs should rely on autonomous mechanisms such as sandbox or certification in order to prevent various attacks from participants [16].

5.2 Resource Management

Resources in volunteer clouds are highly heterogeneous, therefore, managing them can be considered problematic. Virtualisation plays a key role in Desktop Clouds because it virtualises contributed resources and delivers them to users as VMs. DCs face a challenge of developing a resource allocation mechanism that is able to: a) manage non-dedicated, heterogeneous resources, b) deliver a virtualized machine to upper level in DCs and c) work closely with users' tasks in order to find most suitable nodes for each request.

It has been pointed out that lacking central management in DCs cause a major issue in terms of reliability and state maintenance in case of failures [17]. The infrastructure of DC is consisted of nodes that are highly volatile. Therefore, fault recovery mechanisms are crucial in order to improve reliability in this environment [10]. In addition, volunteer clouds require means to interact with other clouds for data migration or to gain extra computing resources [4].

5.3 Quality of Service

VCCs are expected to offer services at a low level of reliability and availability due to the fact that they depend on unreliable volunteered resources which can join or leave the cloud without prior knowledge for various reasons [7]. Availability of individual nodes is considered a primary issue in VCCs [10]. For example, it is estimated that resource unavailability can reach up to 50% in volunteer projects [18]. Availability of each individual node can affect the service quality of VCC. Andrzejak et al. [19] propose a technique to predict the availability of a group of high volatility resources.

6 Conclusion

This paper has introduced Desktop Clouds as being a new type of Cloud computing. Desktop Cloud aims at providing services based on Cloud business model on top on infrastructure that is not made for this purpose. The success of Desktop Grids projects has simulated the idea of applying the same concept within Cloud computing. However, the paper has presented several research issues that need further attention.

References

1. Kirby, G., Dearle, A., Macdonald, A., Fernandes, A.: An Approach to Ad hoc Cloud Computing. Arxiv Prepr. arXiv1002.4738. (2010).
2. Chandra, A., Weissman, J.: Nebulas: Using distributed voluntary resources to build clouds. Proceedings of the 2009 conference on Hot topics in cloud computing. pp. 2–2. USENIX Association (2009).
3. Weissman, J.B., Sundarrajan, P., Gupta, A., Ryden, M., Nair, R., Chandra, A.: Early experience with the distributed nebula cloud. Proceedings of the fourth international workshop on Data-intensive distributed computing. pp. 17–26. ACM (2011).
4. Cunsolo, V.D., Distefano, S., Puliafito, A., Scarpa, M.: Volunteer computing and desktop cloud: The cloud@ home paradigm. Network Computing and Applications, 2009. NCA 2009. Eighth IEEE International Symposium on. pp. 134–139. IEEE (2009).
5. Cunsolo, V., Distefano, S.: From volunteer to cloud computing: cloud@ home. Conf. Comput. Front. 103–104 (2010).
6. Harutyunyan, a, Blomer, J., Buncic, P., Charalampidis, I., Grey, F., Karneyeu, a, Larsen, D., Lombraña González, D., Liseć, J., Segal, B., Skands, P.: CernVM Co-Pilot: an Extensible Framework for Building Scalable Computing Infrastructures on the Cloud. J. Phys. Conf. Ser. 396, 032054 (2012).
7. Andrzejak, A., Kondo, D., Anderson, D.P.: Exploiting non-dedicated resources for cloud computing. 2010 IEEE Netw. Oper. Manag. Symp. - NOMS 2010. 341–348 (2010).
8. Litzkow, M.J., Livny, M., Mutka, M.W.: Condor-a hunter of idle workstations. [1988] Proceedings. 8th Int. Conf. Distrib. 104–111 (1988).
9. Cunsolo, V., Distefano, S., Puliafito, A., Scarp, M.: Cloud@ home: Bridging the gap between volunteer and cloud computing. ICIC'09 Proc. 5th Int. Conf. Emerg. Intell. Comput. Technol. Appl. 2009. (2009).
10. Marosi, A., Kovács, J., Kacsuk, P.: Towards a volunteer cloud system. Futur. Gener. Comput. Syst. (2012).
11. Gupta, A., Awasthi, L.K.L.: Peer enterprises: A viable alternative to Cloud computing? Internet Multimedia Services Architecture and Applications (IMSAA), 2009 IEEE International Conference on. pp. 1–6. IEEE (2009).
12. Arpaci, R.H., Dusseau, A.C., Vahdat, A.M., Liu, L.T., Anderson, T.E., Patterson, D.A.: The Interaction of Parallel and Sequential Workloads on a Network of Workstations. ACM (1995).
13. Foster, I., Zhao, Y., Raicu, I., Lu, S.: Cloud computing and grid computing 360-degree compared. Grid Computing Environments Workshop, 2008. GCE'08. pp. 1–10. IEEE (2008).
14. Dillon, T., Wu, C., Chang, E.: Cloud computing: Issues and challenges. 2010 24th IEEE International Conference on Advanced Information Networking and Applications. pp. 27–33. IEEE (2010).

15. Ristenpart, T., Tromer, E., Savage, S., Shacham, H.: Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. Proceedings of the 16th ACM conference on Computer and communications security. pp. 199–212. ACM (2009).
16. Cao, B.Q., Li, B., Xia, Q.M.: A Service-Oriented Qos-Assured and Multi-Agent Cloud Computing Architecture. *Cloud Comput.* 644–649 (2009).
17. Endo, P., Palhares, A. de A., Pereira, N.N., Goncalves, G.E., Sadok, D., Kelner, J., Melander, B., Mangs, J.-E.: Resource allocation for distributed cloud: concepts and research challenges. *Network, IEEE.* 25, 42–46 (2011).
18. Kondo, D., Taufer, M., Brooks, C.: Characterizing and evaluating desktop grids: An empirical study. *Int. Parallel Distrib. Process. Symp.* 2004. 00, (2004).
19. Andrzejak, A., Kondo, D., Anderson, D.P.: Ensuring collective availability in volatile resource pools via forecasting. *Manag. Large-Scale Serv.* 149–161 (2008).



SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS