

## PEER TO PEER CLOUD COMPUTING

Suyash Medhavi, Ashima Sharma, Thirunavukkarasu K , \*Ajay S Singh

Department of Computer Science & Engineering , Galgotias University, Greater Noida, India

\*Address for correspondence: Dr. Ajay S. Singh, Professor, Department of Computer Science & Engineering , Galgotias University, Greater Noida, India  
**Email ID** : drajay.cse@gmail.com

### ABSTRACT

*Cloud Computing is basically a set of editable large-scale data, server clusters, providing computing and storage services to customers. The typical cloud-computing model is one, which consists of a central structure, one server connected to many nodes. This paper is about a cloud architecture which is based on P2P (Peer-to-Peer) architecture for implementation of cloud. There is no central structure; instead, all nodes are interconnected, forming a MESH architecture. A downtime or failure of one system thus does not affect the working of the entire cloud.*

**Keywords:** Cloud Computing; P2P Network; Storage

### INTRODUCTION

Cloud computing, currently, is the foremost impactful technological development to have emerged in data technology in lately. In less than a decade, cloud technology has reformed the process from plan to product launch, particularly in the case of net start-ups. Before the arrival of cloud computing era, any startup that wanted to deploy its “next big idea” on the webs had to create large (generally monetary) investments for geological real-estate, basic hardware components, network connectivity components and trained technical staff to sustain, maintain and deploy all of its features. Assuming the start-up might survive the monetary burden of construction and maintenance of a fully in situ computing system or resource, the unavoidable delays in releasing its product will reduce any competitive advantage it would have had on its opposing firms in the market.

### TERMINOLOGY

Cloud computing could be a model for enabling omnipresent, convenient, on-demand network access to a shared pool of editable computing resources (e.g., storage, applications, networks, services, and servers) which will be rapidly provided and freed with least managerial effort or interaction of service supplier and client.

There are four key attributes in this definition:

**Service on demand:** computing resources like storage, servers, applications and communication network components can also be provided dynamically, when needed, without the intervention of service provider;

**Remote access:** resources can also be accessed through the internet or other networks, using normal protocols and properly outlined interfaces; **Flexibility:** usually, the client realizes the cloud as an almost limitless and shareable collection of computing resources which can be acquired at any time and in any amount;

**Service billing:** cloud resource usage and maintenance service usage are billed using a pay-per-use (or PPU) model.

The first level of the cloud service model, IaaS, provisions every client their personal piece of the physical hardware through cloud computing and virtualization technologies. Additionally, to virtualize the bare metal or the base machines, an IaaS cloud service supplier might embrace a choice of installed base OS (operating systems) to run on the hardware. Familiar samples of IaaS clouds are “Amazon's Elastic Compute Cloud (EC2)” and “Google's Compute Engine.”

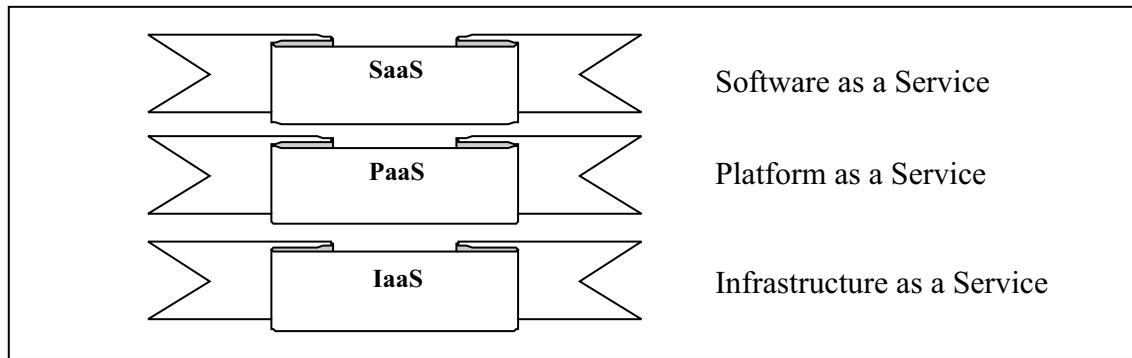


Fig. 1: Cloud Computing Service Models

A PaaS oriented cloud includes an ADE (“application development environment”) along with the underlying base hardware. The customers can evolve applications by taking help of libraries, application program interface (APIs) and software system tools provided may be by other companies. Samples of PaaS solutions are “Force.com from Sales Force” and “App Engine by Google”.

Thirdly, in a SaaS cloud, clients are provided with complete application packages for a selected domain and do not interact with the hardware resources itself. SaaS is often in all probability the model we tend to be most acquainted with in our lives.

#### THE CASE FOR P2P CLOUDS

Building a cloud infrastructure as one, large data-center has its blessings. Construction, equipment acquisition, installation and maintenance are usually simplified since everything is found in one place. Economies of scale permit price reductions so one large installation is typically cheaper than multiple smaller ones. On the opposite hand, running one massive data-center creates varied challenges. The acute density with which the hardware is packed inside a confined physical area generates overall power necessities as much as a small city. And also the heat that's generated has got to be dissipated somehow. It is no surprise that locations in cold climates and people near sources of low cost electric power generation are favored as potential data-center sites. May be the foremost serious disadvantage of a centralized cloud is that it represents a single-point of-failure, regardless however smartly it is designed. Redundant power provision, backup power generators or replicated network connections cannot give absolute protection against ruinous events like fires, hurricanes, earthquakes or floods. Unlikely as these

events could seem, can happen and various incidents were reported in places where entire data-centers with all of the cloud services hosted on them were knocked out by gentle weather storms. Another issue with centralized clouds arises from their geographical location, as what is best for the cloud service supplier might not be best for the client. This can be the case, as an example, once governments place restrictions on sensitive information crossing national borders. A data-center situated during a foreign country might prevent varied domestic customers.

Cloud service suppliers have addressed the last mentioned issues by moving their infrastructure towards a federated model: multiple data-centers are created at distant geo locations interconnected through quick, personal networks. Doing this not solely will increase the independence of failure modes even during the face of catastrophes, it conjointly provides customers with a lot of choices for data location in keeping with expected access patterns and legal issues. If we continue the geographical distribution initiated by federated clouds to its logical conclusion, we might find ourselves with various individual nodes distributed across the world and interconnected through a public communication material just like the net. We name the ensuing design a Peer-to-Peer (P2P) Cloud since it shares several of the characteristic of different P2P systems developed for file sharing, content distribution, and a lot of recently created popular by virtual Cryptocurrency schemes like Bitcoin as their payment methods.

In principle, a P2P cloud may well be engineered out of ordinary computing, storage and communication resources found in several personal homes with basically zero initial investment. Broadband modems, routers, set-top boxes, game consoles and desktop pcs are all

candidates for participation during this design since nowadays most of those devices already give net property, affordable computation and a few storage capabilities. The challenge is to show this assortment of loosely connected, unreliable, heterogeneous, resource-constrained hardware into a coherent and usable infrastructure through existing P2P technologies therefore to provide an iaas cloud API to customers. We should additionally make sure that the salient features of clouds on-demand resource provisioning, elasticity and measured service are maintained. The fully redistributed design of a P2P cloud offers several fascinating properties.

First, there is no single entity that owns or controls it. Like most of the other P2P applications, a P2P cloud can be created and operated as a grassroots effort while not requiring the permission or consent of any authority. People decide unilaterally to participate in a P2P cloud by putting in the suitable consumer code on their native machines. The value of the ensuing P2P cloud infrastructure is equal with the quantity of people who have joined it. Even as no single entity is controls the creation of a P2P cloud, no single entity will unilaterally be able to shut it down.

A second advantage of a P2P cloud derives from the very fact that its elements are little, have low power consumption and are geographically distributed. In different words, power to a P2P cloud is provided through completely different power grids and network connections is provided through different isps, reducing drastically the one point-of-failure risk related to data-centers. Furthermore, the low-power nature of the elements and also the vital geographic separation between them obviates fully the heat dissipation considerations of data-centers. Whereas P2P clouds are unlikely to produce the standard of Service guarantees of centralized or federated clouds, there are however sure usage scenarios for which a totally distributed cloud design will be helpful.

Applications that may benefit from a P2P cloud include embarrassingly parallel computations, multimedia system streaming, online games with low-latency and high-interactivity necessities and cooperative tools with shared data. By cutting down the geographic distribution from international to one organization, we can additionally

imagine P2P clouds made out of idle in-house resources. For instance, associate engineering company might construct a P2P cloud out of its desktop pcs throughout off-load hours and build it accessible to the planning department for structural simulations, the IT cluster to research network access logs for intrusion detection, and allow the accounting department to figure income and alternative monetary indicators for analysis functions.

#### **CHALLENGES FOR P2P CLOUDS**

The extreme scale that a P2P cloud may reach, each within the range of components and in terms of geographic distribution, can mean that failures are going to be common place. To worsen matters more, people who operate the devices out of their homes may plan to flip on/off or plug/unplug them at will. The ensuing dynamic, whereby elements are perpetually joining and leaving the system, is known as churn within the P2P literature. The primary challenge for P2P clouds will be declared as: how will we produce and maintain a coherent computing infrastructure from a large number of unreliable resources and multiplex it among totally different users even within the presence of sustained rates of churn? This needs keeping track of all functioning and on-line devices, dynamically partitioning them among customers and reclaiming them once finished. And every one of this must be done in a totally decentralized manner with no master or controller and despite churn. Because it seems, these challenges are encountered in different P2P systems and economical gossip-based protocols are developed to unravel them. Gossip mongering may be a terribly straight forward interaction paradigm wherever peers in a massive, unstructured network exchange info with little range of their neighbors, presumably change their internal state based on the end result of such interactions. Gossip-based protocols are extensively studied and are accustomed model a various set of functions as well as the spreading of malware during an electronic network, diffusion of data during a social network and synchronization of light pulses during a swarm of fireflies. Gossip-based protocols are appealing for P2P clouds because they're very simple to explain and implement, nonetheless they will realize advanced international computations out of easy local

interactions with efficiency and quickly despite the chance of churn. During a prototype P2P cloud system engineered at the University of Bologna, we've used fully decentralized gossip-based protocols extensively for implementing totally different practicality necessary for P2P clouds as well as membership (figuring out who is up and connected), numeration (useful for cloud monitoring), slicing (partitioning a cloud into multiple sub-clouds), slice merging (useful for elastic resource allocation) and for supporting advanced queries over the set of peers (for example to spot the highest 100% most reliable nodes).

## CONCLUSIONS

Peer-to-Peer algorithms are historically related to file-sharing applications; but as we have shown in the above article, P2P techniques are powerful “building blocks” which will be used to produce massive scale, strong infrastructures out of unreliable parts. We have seen however, a P2P Cloud may well be assembled at just about zero price, and would offer computation and storage resources almost like a standard infrastructure hosted in a massive datacenter. A totally distributed system has the advantage of lacking centralized management, and thus does not have any single point-of-failure or any central dominant authority. Of course, there are disadvantages: security and trust are extremely tough, if not impossible, to attain in a totally distributed scenario. Moreover, one ought to expect a far lower Quality of Service from a P2P cloud compared to it provided by a centralized or federated Cloud. These issues show that the P2P paradigm is not however acceptable for public clouds serving a large user community. However, a P2P cloud would be ok for realizing a personal infrastructure out of already existing resources already owned by the organization (think of a personal cloud owned and operated by one company, or by a bunch of friends who trust each other). From a strictly technical purpose of view, we have already got all the ingredients to create a P2P cloud, however they have to be prepackaged together fittingly so as to get the general design right.

## REFERENCES

1. S. Androutsellis-Theotokis and D. Spinellis,

- "A survey of peerto-peer content distribution technologies," *ACM Comput. Surv.*, vol. 36, no. 4, pp. 335-371, 2004.
2. M. P. Papazoglou and W.-J. Heuvel, "Service oriented architectures: approaches, technologies and research issues," *The VLDB Journal*, vol. 16, no. 3, pp. 389-415, 2007.
  3. X. Jin and J. Liu, "From individual based modeling to autonomy oriented computation," in *Agents and Computational Autonomy*, ser. Lecture Notes in Computer Science, M. N. Et al., Ed., vol. 2969. Springer, 2003, pp. 151-169.
  4. J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in *OSDF04*, 2004.
  5. C. Olston, B. Reed, U. Srivastava, R Kumar, and A. Tomkins, "Pig latin: a not-so-foreign language for data processing," in *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2008, pp. 1099-1110. [Online]. Available: <http://dx.doi.org/10.1145/1376616.1376726>
  6. Y Koh, R. C. Knauerhase, P. Brett, M. Bowman, Z. Wen, and C. Pu, "An analysis of performance interference effects in virtual environments," in *ISPASS*. IEEE Computer Society, 2007, pp. 200-209.
  7. L. Kleinrock. A vision for the Internet. *ST Journal of Research*, 2(1):4-5, Nov. 2005.
  8. S. London. *INSIDE TRACK: The high-tech rebels*. Financial Times, 06 Sept. 2002. <Http://search.ft.com/nonftarticle?Id=020906000371> [18 July 2008]
  9. Foster and C. Kesselman (eds). *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann, San Francisco, USA, 1999.
  10. M. Chetty and R. Buyya. Weaving Computational Grids: How Analogous Are They with Electrical Grids? *Computing in Science and Engineering*, 4(4):61-71, July-Aug. 2002.
  11. A. Weiss. *Computing in the Clouds*. *Net Worker*, 11(4): 16-25, Dec. 2007.
  12. Twenty Experts Define Cloud Computing, [http://cloudcomputing.sys-con.com/read/612375\\_p.htm](http://cloudcomputing.sys-con.com/read/612375_p.htm)

12. Twenty Experts Define Cloud Computing, [http://cloudcomputing.sys-con.com/read/612375\\_p.htm](http://cloudcomputing.sys-con.com/read/612375_p.htm)
13. R. Buyya, D. Abramson, and S. Venugopal. The Grid Economy. Proceedings of the IEEE, 93(3): 698-714, IEEE Press, USA, March 2005.
14. S. Venugopal, X. Chu, and R. Buyya. A Negotiation Mechanism for Advance Resource Reservation using the Alternate Offers Protocol. In Proceedings of the 16th International Workshop on Quality of Service (iwqos 2008), Twente, The Netherlands, June 2008.
15. D. Hamilton. 'Cloud computing' seen as next wave for technology investors. Financial Post, 04 June 2008. [Http://www.financialpost.com/money/story.html?Id=562877](http://www.financialpost.com/money/story.html?Id=562877)
16. Morgan Stanley. Technology Trends. 12 June 2008. [H t t p : / / w w w . Morganstanley.com/institutional/techresearch/pdfs/techtrends062008.pdf](http://www.Morganstanley.com/institutional/techresearch/pdfs/techtrends062008.pdf) [18 July 2008].