Analyzing P2P Cloud Architectures: A Research Perspective

[1]Ms. Amruta Deshmukh, [2]Dr. M. A. Pund
[1]*PhD Student, Computer Science*
Sant Gadge Baba Amravati University Amravati, Maharashtra (India)
[2]*Dept. of Computer Science*
Prof. Ram Meghe College of Engg Amravati, Maharashtra (India)

Abstract
Peer to Peer (P2P) computing has come up as an interesting technique for design of distributed systems. These systems vary from block chain miners to general cloud computing architectures. P2P computing is distributed, secure, robust, scalable and economical due to the collaborative working of multiple computing nodes in synchronization to achieve any task. In this paper, we compare various techniques and architectures used for design of a P2P cloud, and the issues faced by these implementations. Such a study will be useful for researchers to carefully design and develop their own P2P version of cloud by studying the nuances of typical P2P cloud architectures.

Index Terms—*Peer to Peer, cloud, robust, scalable, secure, and distributed.*

I. INTRODUCTION

Cloud computing has become additional and additional in style a day, and plenty of users and firms use these services to store their knowledge or to induce additional computing power. M. Armbrust [1], cloud computing refers to the applications delivery and services over the web, because of the hardware and program within the data centers that give these services. From this definition we can deduce that cloud computing could be a model that permits access to files, applications or services during a present and pervasive method through network to share a collection of configurable computing resources. These resources will be servers, storage, applications and services, which might be speedily provisioned and discharged with a lowest effort in commission management or interacting with the supplier [5]. Therefore, cloud computing provides the illusion of unlimited and on-demand quantifiability. Most ancient cloud computing systems area unit centralized and supported the client-server paradigm. A centralized structure introduces many drawbacks in cloud computing as storage dependence, privacy, quantifiability. During this situation, peer-to-peer (P2P) networks have emerged as a promising distributed information management platform. A P2P network could be a distributed network shaped by a gaggle of nodes, which build a network abstraction on prime of the physical network, referred to as an overlay network [7]. During a P2P system every peer will take the role of each, a server and of a shopper at identical time. A very important advantage of P2P networks is that each accessible resource like process, memory and information measure area unit provided by the peers [3]. Therefore, once a brand-new peer arrives to the P2P system the demand is augmented, however the capability too. This is often impracticable during a shopper server model with a set range of servers. P2P paradigm permits that a distributed platform distributes its load and duties on the taking part of peers. Main characteristic is its qualified storage methodology supported dependability indices, that area unit entirely clear to the user because it is within the centralized cloud computing. This research paper gives the visualization to design P2P cloud architecture with the help of its comparative analysis and study of typical cloud characteristics and its atmosphere. The next sections comprise a brief discussion about structured and un-structured P2P cloud implementations and its observation.

II. CLOUD COMPUTING ATMOSPHERE WITH P2P CLOUD CHARACTERISTICS

Service models outline the amount of abstraction at that a client interfaces a Cloud Computing atmosphere. These area unit the “Software as a Service” (SaaS) model, during this the Cloud client has no management over the infrastructure itself. Google Apps is an example of a SaaS Cloud. In distinction, the capability provided by a PaaS Cloud the “Platform as a Service” model accommodates programming languages, tools and a hosting atmosphere for applications developed by the Cloud client. Samples of PaaS solutions area units are AppEngine by Google, Force.com from Sales force, Microsoft’s Azure and Amazon’s Elastic Beanstalk. Finally, IaaS Cloud the “Infrastructure as a Service” (IaaS) model, provides its customers with basic computing capabilities like process, storage and networking wherever impulsive software system, as well as operational systems and applications, can be run. One among the earliest samples of IaaS Cloud is Amazon EC. Building and operational a Cloud knowledge center is dear [9], therefore solely giant corporations will af- ford such a large investment. However, the present centralized approach to Cloud computing isn't the sole risk, and in some cases may not even be the optimum alternative. In [17] the authors describe a spectrum of potential Cloud architectures: centralized, united and Peer-to-Peer (P2P). Centralized Clouds represent the present business offerings. The centralized model is suitable for applications like scientific computations, data processing, Internet scale net Services and delay-sensitive applications that can't tolerate high communication delays. United Clouds area unit a logical evolution of the centralized approach: they involve multiple Clouds that area unit tied along to make a bigger one. Federation will be wont to enhance dependableness through physical partitioning of the resource pool, and conjointly to handle communication latency problems by binding purchasers to the "nearest" knowledge center. Moreover, united Clouds area unit a stimulating different for those corporations UN agency area unit reluctant to maneuver their knowledge out of house to a service supplier because of security and confidentiality considerations. By operational on geographically distributed knowledge centers, corporations might still take pleasure in the benefits of Cloud computing by running tiny Clouds in-house and federating them into a bigger Cloud [18]. Finally, by stretching the concept of united Clouds to the acute, we can unit able to build a Cloud out of freelance resources that are opportunistically assembled. Such P2P Clouds may be designed by grouping individual peers with none central watching or coordination part. P2P Clouds will modify provisioning of resources at low or no cost; loosely-coupled distributed applications wherever the physical location of nodes is very important to stay data/computation close to the top user, will take pleasure in the P2P model. The Cloud API provides an interface for resource negotiation, allocation and watching, no matter the Cloud design. While P2P Clouds area unit unlikely to produce the options and Quality of Service (QoS) guarantees of a centralized or united Cloud, there are a unit however some usage situations that a completely distributed Cloud designs will be helpful. A P2P Cloud [19] [14] will be assembled at nearly no price victimisation existing resources; so, several tiny or medium-sized organizations might flip idle resources into a computing infrastructure which might be divided among variety of internal "customers". Some applications which might be dead on a P2P Cloud embrace [20] embarrassingly parallel computations, multimedia system streaming, on-line recreation requiring low latency and a high level of interactivity, collaboration tools with shared knowledge. Volunteer Computing (VC) could be a acknowledge computing paradigm, wherever users execute third-party applications. VC systems typically need users to put in a particular application on their PC; the applications fetch and processes computer file from a central location and uploads the results; VC systems area unit principally targeted at embarrassingly parallel scientific applications. The wide used BOINC system [12] [21] separates the shopper program from the application-specific part: users install the BOINC shopper and choose the project(s) they support. Clouds and VC systems have some necessary variations since they serve completely different purposes. The resources of a Cloud area unit usually owned by one entity (the Cloud provider), whereas VC depends on resources provided by third parties. A P2P Cloud differs from a VC system because of there's no central coordination or central repository of tasks.

III. LITERATURE REVIEW

In recent years, many authors have recognized the potential edges of P2P Cloud architectures. In [22] the authors sketched a general framework to support totally distributed applications running over a large-scale and dynamic pool of resources. The authors list many gossip-based protocols which will be applied to create the sub clouds and to implement bootstrapping, observance and management services. Also authors, discovered that the Cloud@Home design depends on centralized elements, whereas permitting finish users to contribute extra resources. In [23] a unique direction is take by the authors, they propose a distributed computing platform known as Nano knowledge Centers (NaDa). Zilch uses home gateways, controlled by ISPs, to produce computing and storage services. Employing a managed peer-to-peer model, Zilch type a distributed knowledge center infrastructure. They have also mention Wuala as Associate in nursing example of Cloud based mostly storage service. Wuala permits users to trade house on their onerous disks to receive encrypted chunks of files uploaded by different users. A PaaS Cloud provides virtual machines, in operation systems, application services, development frameworks, transactions and management structures for applications developed by the Cloud client [24] [25]. During this model, the users or customers will develop their applications at intervals the cloud infrastructure or use their applications regular. The service supplier manages the cloud infrastructure, software package, computer code or sanctionative. However, the purchaser's area unit answerable for the installation and maintenance of the applications they're developing. Authors in [26] state that a P2P Cloud permits organization or maybe individual to create a computing infrastructure out of existing resources, which may be simply allotted among completely different tasks. Potential edges of P2P Cloud computing have recognized throughout the last year and a number of other connected work are planned. Cloud@Home is conferred in [28] that could be a hybrid system that mixes characteristics from volunteer laptop and cloud computing paradigms. Xu et al [42] propose another distributed P2P Cloud system that is intended to produce storage service solely. Associate in Nursing design Associate in nursing its paradigm to produce an infrastructure and repair through a P2P cloud area unit conferred in [29]. Authors in [30] mix P2P and cloud computing to get a hybrid and distributed design for multimedia system streaming service. The following techniques implement the P2P cloud architectures: 1. Centralized methodology 2. Hierarchal methodology 3. Flooding methodology 4. DHT based methodology 5. Gossip based methodology The centralized methodology could be an answer used principally in early P2P streaming systems. During this methodology, the knowledge concerning all nodes, e.g., their address or offered information measure, is unbroken in a very centralized directory and therefore the centralized directory is accountable to construct and maintain the topology. CoopNet [31] and DirectStream [32] area unit 2 sample systems that use the central methodology. Since the central server features a world read of the overlay network, it will handle nodes change of integrity and going terribly quickly. one among the arguments against this model is that the server becomes one purpose of failure, and if it crashes, no different node will be part of the system. The quantifiability of this model, also, is another drawback. However, these issues will be resolved if the central server is replaced by a group of distributed servers. Ensuing answer for locating activity nodes is employing a hierarchal methodology. This approach is employed in many systems, like Nice [33], ZigZag [34], and BulkTree [35]. In Nice and ZigZag, as an example, variety of layers area unit created over the nodes, specified all-time low layer contains all the nodes. The nodes during this layer area unit sorted into some clusters, in line with a property outlined within the algorithmic program, e.g., the latency between nodes. One node in every cluster is chosen as a head, and therefore the chosen head for every cluster becomes a member of 1 higher layer. By cluster the nodes during this layer and choosing a head in every cluster, they type ensuing layer, and so on, till it finally ends up in a very layer consisting of one node. This single node that could be a member of all layers is named the rendezvous purpose. Whenever a replacement node comes into the system, it sends it's being part of request to the rendezvous purpose. The rendezvous node returns an inventory of all connected nodes on ensuing down layer within the hierarchy. The new node probes the list of nodes and finds the foremost correct one and sends it's be part of request thereto node. The method repeats till the new node finds a footing within the structure wherever it receives its desired content. Though this answer solves the quantifiability and therefore the single purpose of failure issues within the central methodology, it's a slow convergence time. The third methodology to find nodes is that the controlled flooding, that is originally planned by Gnutella [36]. Gnu Stream [37] could be a system that uses this idea to search out activity nodes. During this system, every node features a neighbor set that could be a partial list of nodes within the system. Whenever a node seeks a supplier, it sends its question to its neighbors. Every node forwards the request to any or all of its own neighbors except the one World Health Organization have sent the request. The question features a time-to-live (TTL) price, that decreases when every rebroadcasting. The broadcasting continues till the TTL becomes zero. If a node that receives the request satisfies the node choice constraints, it'll reply to the initial sender node. This methodology has 2 main drawbacks. First, it generates a major quantity of network traffic and second, there's no guarantee for locating applicable suppliers. Another answer for locating the activity nodes is to use Distributed Hash Tables (DHT), e.g., Chord [38] and Pastry [39], Split Stream [39] and [40] area unit 2 samples that job over a DHT. In these systems, every node keeps a routing table together with the address of other nodes within the overlay network. The nodes, then, will use these routing tables to search out activity nodes. This methodology is scalable, and it finds correct suppliers rather quickly. It guarantees that if correct suppliers area unit within the system, the algorithmic program finds them. However, it needs further effort to manage and maintain the DHT. The last approach to search out activity nodes are that the gossip-based methodology. Several algorithms area unit planned supported this model, e.g., New Cool streaming [41], DONet/- Cool streaming [42], PULSE [43], gradienTv [44] use a gossip-generated random overlay network to look for the activity nodes. We tend to use the gossip generated Gradient overlay [48] for node discovery in Sepidar and Glive. Within the gossip-based methodology, every node sporadically sends its knowledge convenience data to its neighbours, a partial read of nodes within the system, to alter them notice applicable suppliers, World Health Organization possess knowledge they're craving for. This methodology is scalable and failure-tolerant, however due to the randomness property of neighbour choice, generally the suitable suppliers aren't found in reasonable time. Many different overlay topologies are used for knowledge dissemination in P2P cloud systems. The most topologies used for this purpose are, a. Tree-based topology b. Mesh-based topology c. Hybrid topology The tree-based topology is split to single-tree and multipletree structures. Early knowledge delivery overlays use a singletree topology, wherever knowledge blocks square measure pushed over a arboriform overlay with a media supply because the root of the tree. Nice [33], ZigZag [34], and BulkTree [35] square measure samples of such systems. The low latency of information delivery is that the main advantage of this approach. Disadvantages, however, embrace the fragility of the tree structure upon the failure of interior nodes and therefore the indisputable fact that all the traffic is barely forwarded by them. The multiple-tree structure is associate improvement on single-tree overlays, that was planned for the primary time in SplitStream. During this model, the stream is split into substreams and every tree delivers one substream. Sepidar, CoopNet, gradienTv, woodlet, and Chunky Spread square measure some solutions happiness to the present category. Though multiple-tree overlays improve a number of the shortcomings of singletree structures, they're still liable to the failure of interior nodes T. Zahn, and J. Schiller have shown in [33] that mesh overlays have systematically higher performance than tree-based approaches for situations wherever there's churn and packet loss. The mesh structure is very resilient to node failures; however it's subject to unpredictable latency as a result of the frequent exchange of notifications and requests [12]. Glive, DONet/Cool streaming [25], PULSE [26], Gossip++ [34], power saw [35], and [28] square measure the systems that use a mesh-based overlay for knowledge dissemination. Another resolution for knowledge dissemination could be a hybrid model that mixes the advantages of the treebased structure with the benefits of the mesh-based approach. Example systems embrace NewCoolStreaming [24], Clique Stream [36], mTreebone [37], Prime [38], and [23]. A common downside in P2P streaming systems is free-riding. In P2P content distribution networks nodes ought to be incentivized to share their resources and contribute to knowledge dissemination; otherwise, opportunist nodes, known as free-riders, will use the system while not contributory any resources. This might have a significant impact on the standard of service of the P2P streaming system, resulting in measurability problems and repair degradation [46] [47]. The present solutions to handle the free-riding downside are often categorised as follows: a. Monetary-based b. Reciprocity-based c. Reputation-based In the monetary-based theme, users pay virtual currency to induce content from different nodes. Every node plays a twin role of a content client and supplier. A node, as a rational player, desires to maximise its profit, i.e., the standard of its received stream, however at the same time reduces its prices, i.e., the number of resources it contributes to the system. A preferred modelling tools to review strategic interactions among such rational players are that the scientific theory [46]. Some systems that use the sport theory to beat free-riders square measure [48]- [50]. Reciprocitybased mechanisms square measure like the tit-for-tat strategy in Bit Torrent [51]. Here, nodes live the number of received stream from their neighbours and keep the history of them. A node sporadically decides to transfer content to its neighbours, supported the native info concerning that neighbours have uploaded a lot of to that within the past. PULSE [26] and Bitos [44] square measure 2 systems that use the reciprocity based mechanism. Another mechanism to resolve the freeriding downside is that the reputation-based model. Nodes, during this model, receive scores supported their contribution to knowledge dissemination. The upper score a node has, the upper name it achieves, and consequently the upper priority it's for receiving knowledge. Nodes reputations square measure created supported feedbacks from different nodes within the system that have interacted with them. Sepidar and Glive, BarterCast [51], Eigen Trust, Give-toGet, and BAR gossip square measure variety of P2P streaming systems that use the reputation-based model. IV. COMPARATIVE ANALYSIS The contribution of comparative analysis provides the systematic analysis of structured and unstructured p2p cloud implementations. The structured p2p is a controlled topology with high maintenance and DHT-based resource discovery. This topology resembles structures like Ring, D-dimensional space, Butterfly network. It is a distributed and decentralized topology. Whereas unstructured p2p is a flexible node topology with low maintenance and flooding-like resource discovery. This topology is a type of flat and hierarchical and is a robust topology. We compared structured and un-structured P2P cloud implementations and found the following results:

Table I : Comparison Between Structured And Unstructured P2P Cloud

	Unstructured P2P	Structured P2P
Overlay construction	High flexibility	Low flexibility
Resources	Indexed locally (typically)	Indexed remotely in a distributed hash table
Query messages	Broadcast or random walk	Unicast
Content location	Best effort	Guaranteed
Performance	Unpredictable	Predictable bounds
Object types	Mutable, with many complex attributes	Immutable, with few simple attributes

Overhead of overlay maintenance	Relatively low	Moderate
Peer churn & failure	Supports high failure rates	Supports moderate failure rates
Applicable environments	Small-scale or highly dynamic environments with (im)mutable objects, e.g., mobile P2P	Large-scale & relatively stable environments with immutable objects, e.g., desktop file sharing
Examples	Gnutella, LimeWire, Kazaa	Chord, CAN, Pastry, Kademlia

The structured P2P approach outperforms unstructured P2P in terms of performance, but unstructured P2P has various flexibility advantages over the structured P2P clouds. In the next table we compare various structured P2P solutions, to evaluate the best one(s).

Table II : Comparison Between Structured P2P Solutions

Scheme	Arch.	Routing mechanism	Resilience
CAN	d-torus	Greedy routing through neighbors	M
Chord	Circular key space	Successor & long distance links	H
Cycloid	Cube connected cycles	Links to cyclic & cubical neighbors	M
Kademlia	Binary tree & XOR distance metric	Iteratively find nodes close to key	H
Mercury	Circular key space	Successor; predecessor; & long distance links	M
Pastry	Hypercube	Correct one digit in key at time	M
Tapestry	Hypercube	Correct one digit in key at time	M
Viceroy	Butterfly network	Predecessor & successor links	L

From the above table we can observe that Viceroy Protocol based P2P implementations would be better in terms of network resilience, moreover Viceroy has comparatively less overheads and computational complexity than other methods, which can be observed from the following table:

Table III : Viceroy Observation

Scheme	Lookup overhead	Routing table size	Join/leave cost
CAN	O(dn1/d)	2d	2d
Chord	O(log n)	O(log n)	O(log2 n)
Cycloid	O(d)	O(1)	O(d)
Kademlia	O(log n)	O(log n)	O(log n)
Mercury	O(1/k log2 n)	k + 2	k + 2
Pastry	O(logB n)	O(B logB n)	O(logB n)
Tapestry	O(logB n)	O(logB n)	O(logB n)
Viceroy	O(log n)	O(1)	O(log n)

Unstructured techniques also have their pros when compared to structured implementations. The following schemes compare most of the variations of structured and un-structured implementations:

Schemes-1: Centralized Applicable For: Small to medium scale, static networks, guaranteed resource discovery, centrally administered.

Applications

1. Single site grid cloud computing
2. Desktop grids

Strong Aspects: Complex mutable resources. All phases of resource collaboration O(1) advertises query cost. Extendable to support new attributes. Track enforce incentives trust. Act as a CA to enhance security. No routing state Weak Aspects: Single point of failure O(R) index O(Q) query load . Lower privacy trust as centralized node is the only source of trust

Schemes-2: Hierarchical Applicable For: Medium to large scale, geographically distributed, static networks. Guaranteed resource discovery. Multiple administrative domains.

Applications

1. Multi-site grid cloud computing
2. Desktop grids

Strong Aspects: Complex mutable resources. All phases of resource collaboration O(1) advertises query cost. Extendable to support new attributes. Track enforce incentives trust. Act as a chain of CAs to enhance security O(1) routing state. Weak Aspects: O(R) resource index at higher level nodes or resolution degradation if aggregated. Local failures partitioned hierarchy. Resource select, match bind require collaboration across multiple nodes in hierarchy. Lower privacy Chain of trust is broken if higher level nodes are compromised

Schemes-3: Unstructured + Flooding Applicable For: Small scale, static/dynamic networks Guaranteed resource discovery highly robust Applications

1. Mobile social networks

2. Mobile ad-hoc networks

Strong Aspects: Complex mutable resources. All phases of resource collaboration. Extendable to support new attributes Privacy through anonymity. Highly robust against node failures O(1) resource index. Simple to build maintain network. Preserve locality in mobile networks. Weak Aspects: Very high advertise query cost – message implosion. Match resources only if RSs are flooded. Nontrivial to enforce incentives, security, track trust O(n) routing state.

Schemes-4: Unstructured + Gossiping or Random walk Applicable For: Small to medium scale, static/dynamic networks. Best effort resource discovery.

Applications

1. Mobile social networks

2. Ad-hoc networks

3. P2P clouds

4. Desktop grids

5. File sharing

Strong Aspects: Complex mutable resources. All phases of resource collaboration. Extendable to support new attributes Privacy through anonymity Highly robust against node failures O(1) resource index. Simple to build maintain network. Better load distribution – no explicit load balancing necessary. Preserve locality in mobile networks. Weak Aspects: Very high advertise query cost. Not guaranteed to find resources. Limited ability to match resources. Bind resources only if query agents are used. Nontrivial to enforce incentives, security, track trust. Stale resource specifications O(n) routing state.

Schemes-5: Super-peer + Flooding, Gossiping, or Random walk Applicable For: Medium scale, static or semi dynamic networks. Best-effort resource discovery.

Applications

1. P2P clouds

2. Desktop grids

3. File sharing

Strong Aspects: Complex mutable resources. All phases of resource collaboration. Extendable to support new attributes. Relatively simple to build maintain. Relatively low advertise query cost O(1) advertising cost. Privacy of peers through anonymity. Ability to enforce incentives, trust, security – super-peers can act as Robust against node failures. Weak Aspects: Not straightforward to pick a super-peer. Not guaranteed to find resources. Super-peers can monitor peers are the only sources of trust O(n) resource index O(n) routing state.

Schemes-6: MAD Pastry Hypercube Backbone Applicable For: Medium to large scale, static or slowly moving networks. Guaranteed resource discovery.

Applications

1. Mobile social networks

2. Ad-hoc networks

Strong Aspects: Mutable resources. Resource selection match Physical locality somewhat preserved O(log n) point query cost. Match resources based on locality. Explicit static dynamic load balancing – Hypercube only Privacy through anonymity Ability to enforce incentives trust. Weak Aspects: Single attribute resources only O(n) range query cost. High cost of advertising mutable resources. No resource bind Match limited to locality. Lacks security unless all nodes have access to a CA.

Schemes-7: Resource aware overlay Applicable For: Large scale, relatively static networks. Immutable attributes guaranteed resource discovery.

Applications

1. Desktop grids

Strong Aspects: Complex immutable resources. Guaranteed performance. No RS advertise cost Select bind resources O(A) routing state. Static load balancing Privacy through anonymity Track enforce incentives trust. Weak Aspects: No mutable resources O(n) range query cost. No resource match. No dynamic load balancing. Moderate resilience to churn. Lacks security unless all nodes have access to a CA.

Schemes-8: Ring-like overlays: Mercury, LORM, MAAN, MURK, SWORD Applicable For: Large scale, relatively static networks. Guaranteed resource discovery SADQ or sub-queries.

Applications

1. CASA

2. GENI

3. P2P clouds

4. Desktop grid

5. File sharing

Strong Aspects: Complex mutable resources O(log n) point query cost O(log n) routing state. Lower query cost under SADQ. Lower advertise cost under sub-queries. Latency bandwidth match in SWORD. Easily extendable to support new attributes – except for LORM SWORD. Privacy through anonymity Track enforce incentives trust. Weak Aspects: O(n) range query cost routing state in Mercury. High advertise cost if attributes change frequently – particularly under SADQ. High query cost under sub-queries. No resource matches bind Index query load imbalanced. Moderate resilience to churn Incentives, trust, security has to be trade off with anonymity. Lacks security unless all nodes have access to a CA. In this comparison, A – No of attributes, CA – Certifying authority, Q – No of queries, R – No of resources, SADQ– Single-Attribute-Dominated queries.

IV. CONCLUSION

From this study, we can conclude that Viceroy based architecture is good for P2P structured implementation, but it can be improved by adopting some of the techniques mentioned by Mercury and SWORD techniques which are based on Ring like overlays. Moreover unstructured flooding can help in improving system robustness, and thereby increase the QoS of the P2P cloud. We can also integrate resource aware relay techniques for resource intensive applications, which might help in reducing the computation delay of the system.

REFERENCES

[1] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G.,Patterson, D. A., Rabkin, A., Stoica, I. and Zanaria, I.: Above the Clouds: A Berkeley View of Cloud. In: Technical Report No. UCB/EECS-2009-28, University of California at Berkeley, Berkeley, CA, USA, (2009).

[2] Mell, P. and Grance, T.: The NIST Definition of Cloud Computing. In: Special publication 800-145 (draft), Gaithersburg, MD, USA, (2011).

[3] "Afuronto: A six hop peer-to-peer network of zillions of Nodes" Hrishikesh Dewan, R.C Hansdah 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications 1550- 445X/18/IEEE DOI 10.1109/AINA.2018.00151.

[4] Babaoglu, O., Marzolla, M. and Tamburini, M.. Design and implementation of a P2P cloud system. In: 27th Annual ACM Symposium on Applied Computing (SAC'12), pp 412–417, Trento, Italy (2012).

[5] Sosinsky, B.: Cloud Computing Bible. In: Wiley Publishing Inc., Indianapolis, IN, USA (2011).

[6] Cunsolo, V., Distefano, S., Puliafito, A. and Scarpa, M.: Cloud@home: Bridging the gap between volunteer and cloud computing. In: 5th International Conference on Emerging Intelligent Computing Technology and Applications (ICIC'09), Ulsan, South Korea, (2009).

[7] "Escape From the Data Center: The Promise of Peer-to-Peer Cloud Computing" By Ozalp Babaoglu and Moreno Marzolla IEEE spectrum 22 Sep 2014: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/

[8] Xu, K., Song, M., Zhang, X. and Song, J.: A Cloud Computing Platform based on P2P. In: IEEE International Symposium on IT in Medicine Education (ITME '09), pp. 427-432, Shandong, China, (2009).

[9] C. Elliott, "GENI: exploring networks of the future," Mar. 2009, Available: http://www.geni.net

[10] I. A. Kash, J. K. Lai, H. Zhang, and A. Zohar, "Economics of BitTorrent communities," In Proc. 6th Workshop on Economics of Networks, Systems, and Computation (NetEcon '11), June 2011.

[11] M. Meulpolder et al., "Public and private BitTorrent communities: a measurement study," In Proc. 9th Int. Conf. on Peer-to-Peer Systems, Apr. 2010.

[12] "Effective High Performance Computing using Peer To Peer Networks" Nunziato Cassavia, Sergio Flesca, Michele Ianni, Elio Masciari, Giuseppe Papuzzo and Chiara Pulice 2017 International Conference on High Performance Computing Simulation.

[13] H. Shen, Z. Li, T. Li, and Y. Zhu, "PIRD: P2P-based intelligent resource discovery in Internet-based distributed systems," In Proc. 28th Int. Conf. on Distributed Computing Systems, June 2008, pp. 858–865.

[14] "Improving the Performance of Data Sharing in Dynamic Peer-to-Peer Mobile Cloud" Wenhua Xiao, Weidong Bao, Xiaomin Zhu, Wen Zhou and Peizhong Lu 2016 IEEE 22nd International Conference on Parallel and Distributed Systems

[15] H. M. N. D. Bandara and A. P. Jayasumana, "Characteristics of multiattribute resources/queries and implications on P2P resource discovery," In Proc 9th ACS/IEEE Int. Conf. on Computer Systems and Applications (AICCSA '11), Dec. 2011.

[16] J. Albrecht, D. Oppenheimer, D. Patterson, and A. Vahdat, "Design and implementation tradeoffs for widearea resource discovery," ACM Trans. Internet Technol., vol. 8, no. 4, Sep. 2008.

[17] Y-S . Kee, K. Yocum, A. Chien, and H. Casanova, "Improving grid resource allocation via integrated selection and binding," In Proc. Supercomputing, Nov. 2006.

[18] D. Irwin, N. Sharma, P. Shenoy, and M. Zink, "Towards a virtualized sensing environment," In Proc. 6th Int.Conf. on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom '10), May 2010.

[19] Y. Tan, J. Han, and Y. Lu, "Agent-based intelligent resource discovery scheme in P2P networks," In Proc. Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Dec. 2008, pp. 752–756.

[20] X. Sun, Y. Tian, Y. Liu, and Y. He, "An unstructured P2P network model for efficient resource discovery," In Proc. Int. Conf. on Applications of Digital Information and Web Technologies, Aug. 2008, pp. 156–161.

[21] A. R. Bharambe, M. Agrawal, and S. Seshan, "Mercury: Supporting scalable multi-attribute range queries," In Proc. ACM SIGCOMM '04, Aug./Sep. 2004.

[22] H. Shen, A. Apon, and C. Xu, "LORM: Supporting low-overhead P2Pbased range-query and multi-attribute resource management in grids," In Proc. 13th Int. Conf. on Parallel and Distributed Systems, Dec. 2007.

[23] M. Cai, M. Frank, J. Chen, and P. Szekely, "MAAN: A multi-attribute addressable network for grid information services," J. of Grid Computing, Jan. 2004.

[24] P. Ganesan, B. Yang, and H. Garcia-Molina, "One torus to rule them all: Multi-dimensional queries in P2P systems," In Proc. 7th Int. Workshop on Web and Databases (WebDB '04), June 2004.

[25] P. Costa, J. Napper, G. Pierre, and M. Steen, "Autonomous resource selection for decentralized utility computing," In Proc. 29th Int. Conf. on Distributed Computing Systems, June 2009.

[26] R. Ranjan, A. Harwood, and R. Buyya, "Peer-to-peer based resource discovery in global grids: a tutorial," IEEE Commun. Surv., vol. 10, no. 2.

[27] P. Trunfio, "Peer-to-Peer resource discovery in Grids: Models and systems," Future Generation Computer Systems archive, vol. 23, no. 7, Aug. 2007.

[28] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," IEEE Commun. Surv. Tutor., Mar 2004.

[29] K. Park and V. S. Pai, "CoMon: a mostly-scalable monitoring system for PlanetLab," ACM SIGOPS Operating Systems Review, vol. 40, no. 1, Jan 2006.

[30] S. Kwan, and J. K. Muppala, "Bag-of-tasks applications scheduling on volunteer desktop grids with adaptive information dissemination," In Proc. 35th IEEE Conf. on Local Computer Networks (LCN '10), Oct. 2010, pp.560–567.

[31] M. Liu, T. Koskela, Z. Ou, J. Zhou, J. Riekki, and M. Ylianttila, "Superpeer-based coordinated service provision," J. Network and Computer Applications, vol. 34, pp. 1210–1224, doi:10.1016/j.jnca.2011.01.007

[32] H. M. N. D. Bandara and A. P. Jayasumana, "Evaluation of P2P resource discovery architectures using real-life multi-attribute resource and query characteristics," In Proc. IEEE Consumer Communications and Networking Conf. (CCNC '12), Jan. 2012.

[33] T. Zahn, and J. Schiller, "MADPastry: a DHT substrate for practicably sized MANETs," In Proc. 5th Workshop Applications and Services in Wireless Networks, June/July 2005.
[34] L. Dekar and H. Kheddouci, "A resource discovery scheme for large scale ad hoc networks using a hypercubebased backbone," In Proc Int. Conf. on Advanced Information Networking and Applications, May 2009, pp. 293–300.
[35] P. Golle, K. Leyton-Brown, and I. Mironov, "Incentives for sharing in peer-to-peer networks," In Proc. 3rd ACM Conf. on Electronic Commerce (EC '01), Oct. 2001, pp. 264–267.
[36] B. Q. Zhao, J. C. S. Lui, and D-M. Chiu, "A mathematical framework for analyzing adaptive incentive protocols in P2P networks," IEEE/ACM Trans. Network, vol. 20, no. 2, Apr. 2012, pp. 367-380
[37] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer, "KARMA: A secure economic framework for peer-to-peer resource sharing," In Proc. Workshop on the Economics of Peer-to-Peer Systems, June 2003.
[38] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The Eigentrust algorithm for reputation management in P2P networks," In Proc. ACM World Wide Web Conf. (WWW '03), May 2003.
[39] R. Zhou and K. Hwang, "PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing," IEEE Trans. on Parallel and Distributed Sys., vol. 18, no. 4, 2007, pp. 460–473.
[40] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," IEEE Trans. on Knowledge and Data Engineering, vol. 16, no. 7, 2004, pp. 843–857.
[41] R. Gupta and A. K. Somani, "Reputation management framework and its use as currency in large-scale peer-to-peer networks," In Proc. 4th IEEE Int. Conf. on Peer-to-Peer Computing (P2P '04), 2004, pp. 124–132.
[42] X. Zhao, L. Li, and G. Xue, "Authenticating strangers in fast mixing online social networks," In Proc. IEEE GLOBECOM '11, Dec. 2011.
[43] T. Choi, H. B. Acharya, and M. G. Gouda, "Is that you? Authentication in a network without identities," In Proc. IEEE GLOBECOM '11, Dec. 2011.
[44] D. S. Milojicic, "Peer-to-peer computing," HP Laboratories Palo Alto. Tech. Rep. HPL-2002-57, July 2002.
[45] H. Shen and C. Xu, "Performance analysis of DHT algorithms for rangequery and multi-attribute resource discovery in grids," In Proc. Int. Conf. on Parallel Processing, Sep. 2009. pp. 246–253.
[46] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," In Proc. ACM SIGCOMM '04, Aug. 2004.
[47] J. Ledlie, P. Gardner, and M. Seltzer, "Network coordinates in the wild," In Proc. USENIX NSDI '07, Apr. 2007.
[48] R. Beverly, K. Sollins, and A. Berger, "SVM learning of IP address structure for latency prediction," ACM SIGCOMM Workshop on Mining Network Data, Sep. 2006, pp. 299–304.
[49] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz, "P4P: Provider portal for applications,"ACM SIGCOMM Computer Communication Review, vol. 38, no. 4. Oct. 2008, pp. 351–362.
[50] J. Seedorf, S. Kiesel, and M. Stiemerling, "Traffic localization for P2Papplications: the ALTO approach," In Proc. 9th IEEE Int. Conf. on Peerto-Peer Computing (P2P '09). Sept. 2009, pp.171–177.
[51] M. S. Artigas and B. Herrera, "Social Helpers: Introducing social trust to ameliorate churn in P2P reputation systems," In Proc. IEEE Int. Conf. on Peer-to-Peer Computing, Aug.-Sep. 2011, pp. 328–337.