

A DISTRIBUTED INFORMATION RETRIEVAL MODEL FOR VIRTUALIZED RESOURCES IN CLOUD

M. Shoaib, Shazia, F. Ashraf, S.Majid*, and K. Kalsoom

Department of CS & Engineering, University of Engineering & Technology, Lahore, Pakistan

*Department of Computer Science, LCWU, Pakistan

Corresponding Author email: shoaib_uet@hotmail.com

ABSTRACT: Virtualized computing resources are made available for providing different services over the internet through cloud computing. Due to the hypervisor interface problem between computer hardware and operating system, this is an ultimate reason of delayed communication and inefficiency in search engines. A new model for virtualized resources in cloud has been proposed. This model has been evaluated using different performance parameters and results are satisfactory. Further work can be done for implementing this model for a real application.

Key words: Cloud Computing, Virtualization, Information Retrieval, Mobile agents.

INTRODUCTION

Various services and dynamically scalable virtualized resources are added to the cloud (Xu et al., 2009). Cloud makes the resources available globally with greater flexibility (James Farnhill, 2010).

The virtualization architecture is presented in figure 1. Hypervisor works as VMM to provide access to underlying hardware and to hide and multiplex it for VMs.

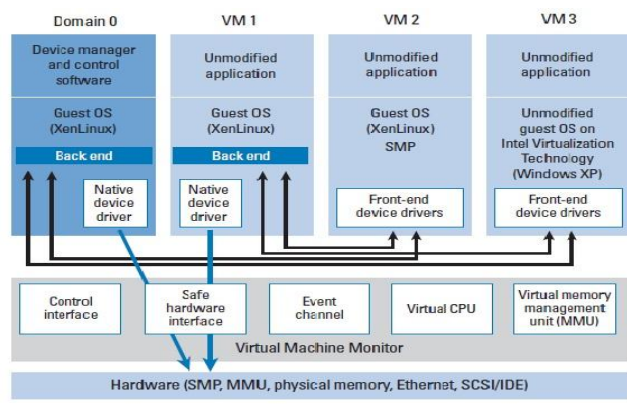


Figure 1- Virtualization architecture diagram (Ashraf, F., M. Shoaib, 2010)

Domain 0 is dominant and playing an important part. This is the only domain which has the access to the control interface of the hypervisor (Abels et al., 2005). All the other VMs are managed and controlled through this interface. Management of these domains includes create, delete, start and stop etc. Any VM can be created through this management software running inside domain 0. Special rights can be given to this VM. Like VM1 can directly access the hardware through an interface

provided by VMM like Xen (Barhan et al. 2009). Once virtualization software like Xen is configured on a server it will replace the operating system clock with its own clock (such as Linux clock with Xen clock). This will result into different types of overheads caused by virtualization including network overhead (Jamal, H., 2009).

The need for improvements in information services including information retrieval is now mandatory due to the rapid growth of virtualized resources in cloud (James Farnhill, 2010). All the cloud resources are distributed where as the existing search engines such as Yahoo, Google, MSN etc are centralized systems (Jingfang et al., 2005). Centralized systems are suffering from the different drawbacks including less scalability, frequent server failures and information retrieval issues as mentioned by (Htoon et al., 2008).

Document virtualization is also becoming popular over the last few years (Watters, C., 1999). Existing distributed IR models are also unable to search inside a virtualized physical node with multiple virtual systems running in parallel in the form of a grid.

(Jingfang et al., 2005) proposed a distributed IR model to resolve the issue of accurate and fast allocation of required information but still many issues are unsolved.

A modified IR model is the need of the time which can work efficiently with virtualized resources (James Farnhill, 2010).

In this research paper a distributed information retrieval model based on analyzing and researching the technologies of Semantic search, Mobile Agent and Peer to Peer (P2P), has been presented.

Results have approved that this model is effective, scalable for both simple and virtual grids.

The paper is outlined as follows: the section of materials and methods provides background information

on supporting technologies, describes the problem and presents the design of the proposed model. In the next section results and discussions are given. Conclusions of our work are given in the last section with future recommendations.

MATERIALS AND METHODS

We have used Mobile Agents to make information searching efficient for virtualized nodes in cloud due to their capabilities (Clark et al., 1997).

Semantic technologies are used to help in understanding the meanings of the information provided on web. The Topic map works as semantic web to find the nodes keeping the information most relevant to the user request. The generation of topic map module works with different useful modules (Maly et al., 2001).

Automatic Generation Module generates XTM file does the whole process with the help of standardized metadata that are considered to be the topics and the rules written inside the rule document are followed (Van et al., 2007).

Proposed IR Model for Virtualized Resources: The architecture of the proposed IR model is shown in figure 2 (Ashraf F., M. Shoaib, 2010).

The model consists of three Nodes; Common, Principal and Backup Principal Node.

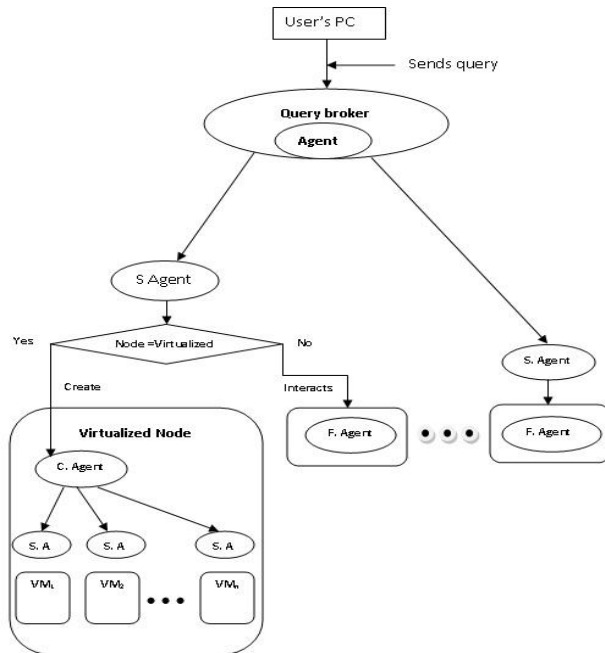


Figure 2- Architecture diagram of proposed Model (Ashraf F., M. Shoaib, 2010)

The functionality of each module is given below.

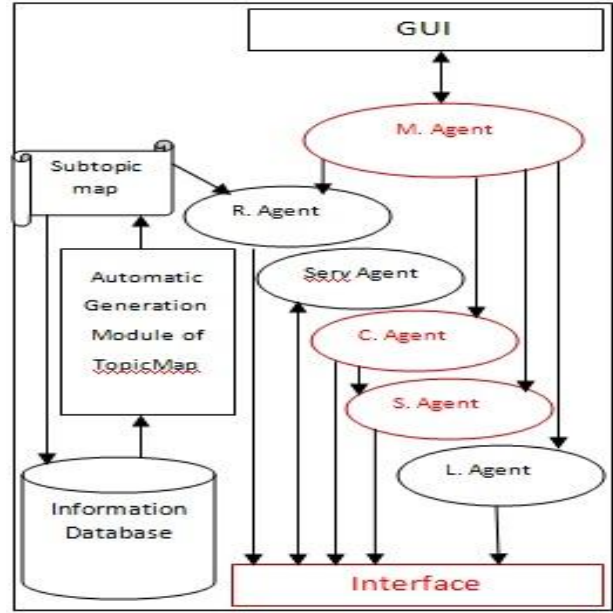


Figure 3- Common Node (Ashraf F., M. Shoaib, 2010)

Common Node. (depicted in figure 3) contains information. Common node consists of sub modules: Automatic Generation Module of Topic Map and Mobile and Fixed Agents.

Automatic Generation Module of Topic Map: Information is sorted out according the user query in conceptually relevant order. It has the ability to trace information recourses that are conceptually relevant. It is capable of not only sorting out information in a better way but also searching it from large number of scattered resources. It gathers the particulars and hints to understated knowledge models. It makes the information search more precise, accurate and relevant.

Mobile and Fixed Agent System Common Node is using two types of agents; Mobile and Fixed agent. Management agent and Service agent act as fixed agent. Mobile agent comprises Register, Search, Counter and Logout Agent. It generates a Register agent whenever any Common Node connects to P2P networks. The Register agent takes the information of URI and topic map to the Principal agent using Agent Transfer Protocol. It helps registering on the Principal Node. End user put forwards his search request by the use of GUI and Management agent creates a Search Agent according to the particulars given in search request. It also produces a Counter agent to check whether an encountering node is a virtualized system or not. If it is so then count how many virtual systems are running on this physical machine. All the virtual systems act as complete Principal node. In case, an encountering node is a virtualized machine, the counter agent then returns counted value of the number of running virtual systems to the search agent. The Search

agent coordinates with the driver domain Principal Node's Search Service agent. The Search Service agent generates a search agent per virtual system using the value returned by Counter agent.

Search Service agent generates the search agents equal in number to the counted value given by the counter agent. Search agents conduct search from every virtual system in parallel. The Search agents also bring search query information to Principal Node and present it to Search Service agents to enable to conduct search according to the user requirements. Search agents from the driver domain and other VMs then come back to the Common Node with a list of website addresses. This list helps end user to search out their wanted information. Finally the Common Node cooperates with other Nodes of the network which are included in the list of addresses returned by search agent. Common Node leaves the network and Management agent generates Logout agent which migrates to the Principal Node to logout itself from Principal Node.

On the other hand in case the encountered node is a simple network node, the Search agent gets the user's request information and takes it to the Principal Node to interact with Search Service agent. So that Search Service agent can conduct search according to the requested information. There is no need to get multiple agent instances to conduct search on a single system. Search agent returns to the Common Node with a list of website addresses in the same way. And all the other agents work in similar order as in case of virtualized node.

Principal Node (depicted in figure 4) serves as local servers; they are connected with each other. Such peers are liable for offering services to all the users. In our system we have a lot of such nodes; anyone of them can be selected by the Common node. Principal node inherits the functionality of common node along with some additional services like, registration, retrieval and index merger services. Topic Map Merger module, mobile agents and fixed agents are the components of a Principal node. Mobile agent consists of Search agent etc. is generated by other major agents. Fixed agents include Register Service Agent and Search Service Agent, that are not generated by other agent but they itself are major working agents and liable of generating more agents if required.

Topic Maps Merger Module The Register agent brings subtopic maps from the Common node to the Topic Map Merger Module. This module then merges them into an integrated topic map. To combine these topic maps following five merger operations are to be pursued. These are: Topic Merger operation, Subject-based Merger operation, Naming Constraint-based Merger operation, Explicit Topic Map Merger operation and Implicit Topic Map Merger operation.

The similar topic maps are merged resulting into an integrated one during the execution of topic map merger process. This module helps us improving the information search and retrieval using semantic retrieval technologies.

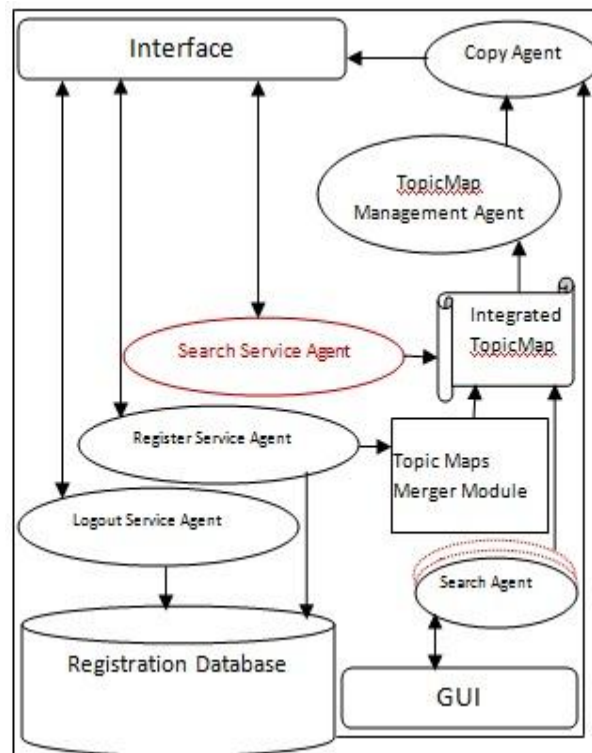


Figure 4- Principal Node (Ashraf F., M. Shoaib, 2010) Following is the detailed functionality performed by each.

Mobile and Fixed Agent Principal node holds the following two types of agents: Mobile and Fixed agent. Fixed agents are not generated by any other agent but itself are responsible for generating other mobile agents. In our Principal node Register Service Agent, Search service Agent, Logout Service Agent and Topic Map Management Agent are some examples of fixed agent. On the other hand Mobile agents are generated from other fixed agents, for instance Search and Copy Agents. Principal node is responsible for four different tasks:

At first, Register Agent travels from Common node to the Principal node, it helps Register Service agent in completing registration of Common node in the Registration Database. Register Service agent also presents subtopic map to the topic maps merger module to be merged into a single topic map. These subtopic maps are brought by Register agent from Common node to Principal node. In case of virtualized machine the whole process will be repeated on every virtual system acting as Principal node.

Secondly, the Search agent hands over search request information along with the counted value of the number of virtual systems to Search Service agent of Principal node. In case of a virtualized machine it will copy and generate the equal number of Search agents to the counted virtual systems. The Search agents then coordinate with the search service agent of every virtual system in parallel. There Search Service agents search for the required information in parallel from integrated topic maps of every virtual system. It copies and creates new Search agents in case; it doesn't find the user's required information. For a simple non-virtualized node the Search Service agent does the same as described above. These new Search agents roam to the other Principal nodes (virtualized, non-virtualized) following the Directed Breadth-first algorithm. For more understanding Search agents take search request to other Principal nodes carrying the most relevant information by calculating the degree of similarity between the search request and the topics in those Principal nodes.

Thirdly, Logout Service agent removes the Common node information from Registration database. This way, it will log out the Common node from Principal node on leaving P2P network. Principal node also controls the updates of topic map by Topic Map Management agent. Topic Map Management Agent creates Copy Agent which migrates to Supporting Principal node to backup the information of registration and topic map.

Backup Principal Node (figure 5) is also nominated by Common node. The one with better performance is selected to support Principal node. It can not only share the workload of Principal node but also has the capability to work just like Principal node whenever it would be

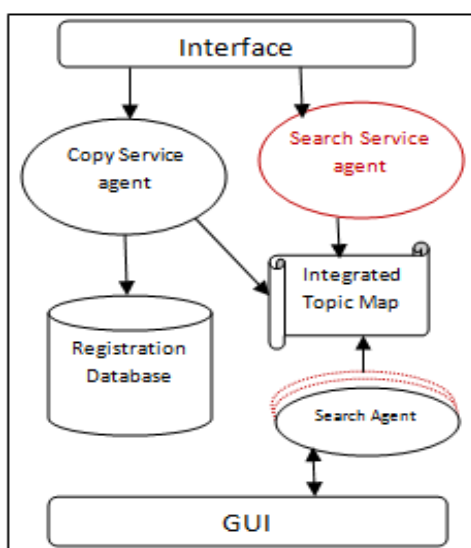


Figure 5- Backup Principal Node (Ashraf F., M. Shoaib, 2010)

down. Backup Principal Node produces Service agent to communicate with Copy agent of Principal node. This Copy agent moves from Principal node to Backup Principal Node with the backup of registration information and Integrated Topic Map on Principal node.

Mostly this part of model works in combination with Principal node to share its workload using Search Service Agent. In case of the Principal node breakdown, Backup Principal Node is responsible for the initiating selection of another Principal node, since it is the best in performance among all the other Backup Principal Nodes and it is the one who set up the crashed Principal node. There could be multiple numbers of Backup Principal Nodes depending on the situation demand. In case they are less in number than required, it is compulsory to elect more from Common node.

RESULTS AND DISCUSSION

In this section, we will present an evaluation of our work in terms of the qualitative and quantitative performance impact of the new features being added. In our proposed work the network load is tried to be balanced using parallel processing and efficiency is increased by avoiding repeated moves of agents. This New Scalable Distributed Information Retrieval Model for virtualized resources in cloud has the following advantages:

Resource Utilization: Proposed model helps making maximum use of the available resources. It allows the full utilization of unused computing power offered by multi-core processor systems, network bandwidth available in Gbits/secs and the information resources. The work in (Jamal H., et al., 2009) has provided a quantitative analysis of network bandwidth, memory and processor in virtualized environment on multi-core architecture. It claims that virtualization is a bottleneck when it comes to network and suggested different ways of improving it. This model deals in its best with virtualized nodes in terms of IR along with fully utilizing the available processing power.

Search Depth: Since P2P technology is used to build the basis, this model lets end user to search the required information deeper than any centralized search engine. This way it attains the benefits of Semantic search. It is independent of the host system in terms of hardware/software and even the format of the document in P2P network (Hong et al., 2003). This helps retrieving information in distributed way from cloud resources. It reaches to each node in the network regardless of its type (virtualized or non-virtualized) and search through all the information resources available on that node. On the other hand, previously available centralized search engines with server centric nature goes less deeply and ignore many of virtual/real resources.

Index Integration: The model consists of a component, named automatic generation and merger of topic map to get the indexes from scattered network nodes in distributed systems and unite them into one. The topic map is just like ontology, it discloses the Semantic relations between objects. Ultimately, we achieve a level's improvement in efficiency of information retrieval by the realization of Semantic Retrieval. The model covers up all the relevancy issues this way.

Surmount server malfunction: Whenever the Principal node crashes in the model, a new one is selected from the available Backup Principal nodes. This can successfully prevail over system breakdown.

Avoidance of Network overhead: Popularity of virtualization in cloud computing is due to its features like cost effectiveness, load balancing and utilization of resources. The throughput comparison readings are given in Table 1.

In figure 6, we observe a doubled difference of throughput in case of 5KB and 10KB files. And the throughput is almost equal in case of 100KB. The first two cases are most common so we consider them. This behavior shows that while retrieving data from a virtual system adds network overhead which will affect the overall throughput. Proposed IR model avoids excessive network involvement while processing any user query. This is done by involving a number of mobile agents with sufficient functionality.

Table1: Throughput comparison of virtual and non-virtual environment (Ashraf F., M Shoaib, 2010)

Sr-No	Doc Size	Virtualized Web Server		Non-virtualized Web Server	
		Time (s)	Transfer rate (Mb/s)	Time (s)	Transfer rate (Mb/s)
No. of requests= 1,000,000					
1	5KB	265.67	156.19	143.59	289.10
2	10KB	284.37	283.02	249.58	538.21
3	100KB	875.91	894.66	873.08	897.58
4	5KB	264.26	157.02	145.79	284.73
5	10KB	292.47	275.18	151.20	532.45
6	100KB	877.08	893.49	873.05	897.60
7	5KB	264.12	157.11	146.74	282.88
8	10KB	294.45	273.33	155.16	518.86
9	100KB	875.93	894.68	872.85	897.84
10	5KB	275.14	150.90	147.88	280.71
11	10KB	298.22	269.87	158.83	506.88
12	100KB	877.21	893.44	873.06	897.78

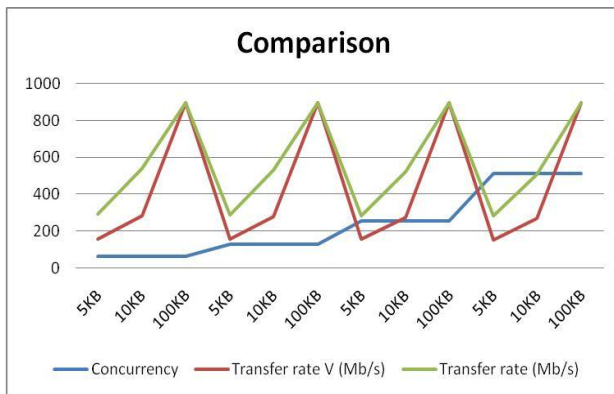


Figure 6- Throughput comparison of virtual and non-virtual web server (Ashraf F., M. Shoaib, 2010)

We use netperf and mpac benchmark for the performance analysis of our virtualized platform. The results obtained are as shown in figure 7 and parameters are defined in Table 2. Then we repeated the same experiments with our Mpac1.1 network benchmark

(Mpac1.1, Netperf2.4) and got the same results in both cases.

Table 2: VM to VM throughput comparison using same and different host (Ashraf F., M. Shoaib, 2010).

Client	Server	Throughput (Gbps) on same host	Throughput (Gbps) on different host
Guest	Host	3 to 4	1 to 2
Host	Guest	2 to 3	0.9 to 1
Guest	Guest	5 to 6	0.9 to 2

We get these readings by running each experiment three times to make the collected data accurate. The presented values are the calculated average of all the readings. As shown every experiment is run for 180 sec. 10G network cards are used in the described experiments. The remarkably good throughput on the same host is due to the avoidance of the traversal of all virtual emulated devices. Memory to memory copying

process is much speedier than the traversal of the actual network and hypervisor layer.

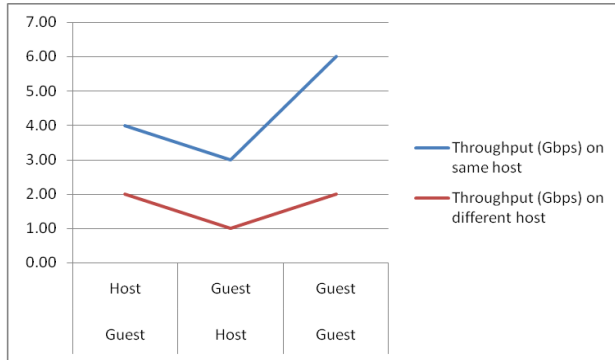


Figure 7- Performance comparison between intra and inter-host communication (Ashraf F., M. Shoaib, 2010).

Considering the case of virtualized node our model allows detection and search through all the VMs once an agent enters it. Rather than dealing with every VM as a standalone node to the network and traversing the same path for every VM over and over again an agent is used to enter into the virtualized system to conduct search. Instead of sending the search agent for every VM through the network, a single search agent creates its multiple instances on entering the driver domain of virtualized node. These instances work with one agent per VM. This will reduce the overall network traffic overhead. A mobile agent has the ability to learn from its working atmosphere which enables it to reply in suitable manner to any change in its surroundings. These agents are also capable of selecting the moving object/target on the basis of network load. Moreover, P2P distributed networks has the ability to use available network bandwidth to its full. The above arguments prove the realization of load balancing in our proposed model.

Fast Information Retrieval: Previously available centralized search engines work in server centric environment and dependent on the servers. This model also increases the search speed by adding parallel search to it.

Scalability: The proposed model is more scalable since it has the intrinsic worth of both centralized and distributed P2P structure. So it is capable of defeating all the causes of poor scalability resulted by central server in centralized systems. It can also prevail over all the network limitations of such systems.

Resource Sharing: All the nodes in a distributed network have the ability of resource sharing. They can simultaneously work and share resources among all nodes. This is done by generating multiple agents to conduct search inside a virtualized node. All of these

agents execute in parallel and make the idle parts of processor working. Similarly, it enables the network card to be used to its maximum capacity.

Fault-Tolerant systems: Mobile agents have the utmost competence against the time and unexpected situations occurred inside any system. This capability of agents helps us developing fault-tolerant distributed systems in a trouble-free manner. When the host computer turned-off all the mobile agents running on it are informed/notified, so that they must have enough time to migrate to any other node and carry on their execution there. These abilities of mobile agents can successfully prevail over the bad impacts of insecure and volatile P2P network.

Parallel Processing: Our model has very strong parallel processing facility. It can vigorously generate a number of mobile agents to work in parallel. It helps conducting tasks in a way to get better efficiency and decreases response time of any action. Mobile agents have many distinguishing features like vibrant and balanced/coherent dispense of themselves among the nodes of a network. While solving a problem, they have the ability to uphold the best possible configuration based on certain specific rules. In our proposed work we are using parallel processing ability of mobile agents to effectively perk up competence of indexing and information retrieval.

Conclusion: Cloud computing is a globally rising concept and virtualization is a way to implement the cloud. Virtualized resources are therefore unavoidable in the cloud. It is observed that virtualization is an overhead when it comes to information retrieval in cloud. This research work describes the design of a scalable IR model to effectively work with virtualized resources in cloud. Its performance has been evaluated using different performance analysis tools. It is concluded that the performance overhead in virtualized resources is due to the unstable network behavior, and the replacement of operating system clock. Better performance of search engines can be achieved using this improved IR model. Most promising feature of this model are parallel processing, load balancing, resource sharing, fault tolerance and many more. The qualitative goals have been met quite successfully, as the search can be conducted efficiently from the virtualized nodes.

Acknowledgements: We are thankful to the Higher Education Commission of Pakistan for the generous financial support. This research paper is extracted from dissertation of the third author. She has also awarded funding for this project implementation from HEC, Pakistan. She is very grateful to the UET Lahore Pakistan, to her supervisor and colleagues who helped her in extracting, writing and finalizing this research paper from her dissertation (Ashraf F., M. Shoaib, 2010).

REFERENCE

- Ashraf, F. and M. Shoaib, A Distributed Information Retrieval Model for Virtualized Resources in Cloud, MS Thesis, University of Engineering and Technology, Lahore Pakistan, (2010).
- Xu, K., M. Song, S. Zhang, and J. Song, A Cloud Computing Platform Based on P2P, Proceedings of IEEE International Symposium on IT in Medicine & Education (ITIME), 427 -432 (2009).
- Jingfang, X. and L. Xing, Design and implementation of a scalable distributed information retrieval, J. J Tsinghua Univ (Sci & Tech), 1842-1846 (2005).
- Htoon, H., and Thwin, Mobile Agent for Distributed Information Retrieval System, In the Proceedings of the 5th (ECTI-CON), 169-172 (2008).
- Voas, J., and J. Zhang, Cloud Computing: New Wine or Just a New Bottle?, IT Professional, IEEE Computer Society, 11(2): 15-17 (2009).
- James F., Article 'research on demand', Research Information, posted on 6 January (2010).
http://www.researchinformation.info/news/news_story.php?news_id=567.
- Abels, T., P. Dhawan, and B. Chandrasekaran, An Overview of Xen Virtualization. Dell Power Solutions, 109-111 (2005).
- Holmqvist, K., T. Halbach, and T. Kristoffersen, Virtualization as a Strategy for Maintaining Future Access to Multimedia Content, Proceeding of IEEE 1st intl' conference on Advances in multimedia, 29-32 (2009).
- Barhan, P. et al., Xen and the Art of Virtualization, Proceedings of the 19th ACM symposium on Operating systems principles, Bolton Landing, 164-177 (2009).
- Watters, C. Information Retrieval and the Virtual Document. Journal of the American Society for Information Science. 50(11), 1028-1029 (1999).
- Clark, K. L., S. Lazarou, A Multi-Agent System for Distributed Information Retrieval on the World Wide Web, In the Proceedings of 6th IEEE workshop on Enabling Technologies: Infrastructure for collaborative enterprises, 87-92 (1997).
- Maly, K., M. Zubair, and X. Liu, Kepler-An OAI Data/Service Provider for the Individual, J. D-Lib Magazine, 7(4): (2001).
- Xia, L. X., Z. Y. Wang, and C. Chen, Scalable Distributed Information Retrieval Model Based on Topic Map and Mobil Agent, In Proceedings of IEEE International Symposium on IT in Medicine and Education, 454-459 (2008).
- Sompel, V. D, and C. Lagoze, The Open Archives Initiative Protocol for Metadata Harvesting, 54-62 (2007).
- Jamal, H., A. Qadeer, W. Mahmood, and J. Ding, In proceedings of the IEEE International Conference on Virtual Machine Scalability on Multi-Core Processors Based Servers for Cloud Computing Workloads NAS, 90-97 (2009).
- Hong, C., L. Shuangyu, and Y. Yuhua, Development and Application of P2P Technology, Journal of Computer Engineering (2003).
- Wei, Z., and W. Beizhan, The Research and Application of Peer to Peer Networking Technology, J. Microprocessors, 4: 45-47 (2006).