IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ASSISTED DISTRIBUTED KERNEL Self-Organizing Cloud system built using a peer-to-peer network

[1] Kumar K, [2] Anand T R, [3] Dhanvi P Desai, [4] Keerthana V
[1] Assistant Professor, [2,3,4] Students
[1,2,3,4] Department of Computer Science and Engineering
[1,2,3,4] K. S. Institute of Technology, Bengaluru-560109, Karnataka, India

Abstract: Current Cloud Computing is primarily based on proprietary data centers, where hundreds of thousands of dedicated servers are setup to host the cloud services. In addition to the huge number of dedicated servers deployed in data centers, there are billions of underutilized Personal Computers (PCs), usually used only for a few hours per day, owned by individuals and organizations worldwide. The vast untapped compute and storage capacities of the under- utilized PCs can be consolidated as alternative cloud fabrics to provision broad cloud services, primarily infrastructure as a service. This approach, thus referred to as "no data centre" approach, complements the data center based cloud provision model. In this paper, we discuss our opportunistic Cloud Computing system, called ADK that runs on scavenged resources of underutilized PCs within an organization/community- namely our college. Our system tries to identify that the "no data center" solution indeed works.

Index Terms **- Cloud Computing, Peer-to-peer network, Virtual Data Centre, ADK**

I. INTRODUCTION

The existing system utilized in cloud computing is centred on proprietary servers that are set up to host cloud services. Where hundreds of thousands of dedicated servers are setup to give the services. Setting up the data centre for cloud is expensive and running the infrastructure needs expertise as well as a lot of resources such as high power for cooling, redundant power for assured availability, etc. For example, 45% of the data centre cost goes to the acquisition of servers, 25% goes to specialized infrastructure for fault tolerance, redundant power, cooling systems, and backup batteries, while electrical cost consumed by the machines accounts for 15% of the amortized total cost. In addition to the vast number of servers used in data centres, there are billions of Personal Computers (PCs) owned by individuals and organizations worldwide. These PCs are mostly underutilized, usually used only a few hours per day. Researches show that desktop computers owned by organizations are idle up to 97% of the time. We had argue that we shall treat the untapped CPU cycles and disk spaces of the great many underutilized PCs as precious assets, like monetary assets, to consolidate and reuse them for the good of the society and of the individuals, similar to the way a peer-to-peer network utilizes required resources from the other nodes on the network.

17 years ago research on P2P systems has received a significant amount of attention in the academic and industrial sectors. The main purpose of the P2P design is for peers to correspond on the internet, without the need of new protocols on switches and routers in the internet core. In P2P computing, nodes organize themselves as an overlay network, in which packet transmission on each of the overlay links uses standard Internet protocols, that is Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). P2P file distribution systems allow nodes to collaborate among each another so that very bulky files can be distributed from one node to several receivers - gossiping. A file is divided into chunks, and nodes connect with one another in a random mesh topology, exchanging these chunks using random gossiping. Research on P2P systems began about 17 years ago and since its inception; it has gone through four major stages of evolution. The earliest stage paid attention on well-organized search protocols, either by using a distributed hash table, or by using a gossip protocol. The second stage started around the year 2007, when the research attention on P2P systems shifted towards optimizing protocols to transfer large files. The third stage began around 2010 when live and on demand streaming protocols have become the topic that attracted passionate research attention. The final stage is where we are now which is P2P communication networks. These stages however have substantial overlaps in time. For example mechanisms that distribute resources in P2P have started to receive attention from cloud computing today.

Our model is designed to work on the basis of the above idea where a P2P network is used to compile a Virtual Data Centre, where the accessible resources are distributed amongst requesting users on the bases of individual needs. Each user that opts to use the above mentioned service will be given a threshold as to how much of the shared resource he/she will be permitted to use for personal benefit until a voluntary contribution is made to the virtual machine. This interconnected architecture also makes shared resource identification

and utilization hassle free over the proposed platform. He/she can choose to leave or withdraw the shared resources from the platform at any time, but returning conditions will be met as per proposed threshold standards. Every user that requests to join the model is authorized and associated with a threshold based on the system specifications. This thresholds are tried and tested to ensure efficient parallel execution during voluntary contribution.

II. SURVEY

There are multiple existing systems that can be surveyed in order to identify and analyse the working of a P2P network, these surveys are used to cumulate an idea on how this network can be used to build a Self-organizing cloud.

BIT-TORRENT

Bit-Torrent appeared in 2005 as an application layer network protocol used to distribute files. Nodes here can download and upload a file at the same time. This does not necessary mean that a node finishes its download before uploading. Complete downloaded blocks can be uploaded to other nodes on the network. The serving capacity increases as the number of downloads increases making the system self-scaling. A tracker is a computer which acts as a server, and maintains a list of nodes which are participating in the system (i.e. computers that are currently downloading or uploading file data). Once a computer has established a connection to one other node in the system, it can begin downloading data. Faster download occurs based on the increased number of node connections hence the likelihood of obtaining a complete copy of the file. A typical Bit-Torrent system needs to have at least one seeder which has a complete copy of the data being shared. When a copy has been uploaded by the seeder to other nodes, the seeder may leave the network. However, for successful ongoing sharing of the data, at least one complete copy must remain available across the nodes in the network.

GNUTELLA

Gnutella is a large P2P network. Other networks adapted this model since it was first of its kind. Peers have same capability and responsibility. Nodes communicate in a symmetric manner in this type of network. Lack of a central directory server Index allows metadata of shared files to be stored locally among all peers. Each user acts independently. Nodes communicate with each other with the use of queries.

In order to build a self-organizing cloud, we consider each of the computers as nodes on the peer to peer network. These nodes will contribute their idle resources to the Virtual Data Centre and draw resources whenever there is a requirement from the native system. The communication is done by message passing and resource monitoring is done dynamically, these monitored resources are broadcasted onto the network for the VDC.

III. METHODOLOGY

MODULES

1. Free Resource Advertisement

The first module of our project checks for free memory and free processing power and broadcasts it on the network. On the Linux platform, we run basic Linux commands to obtain the information from the native OS. This is done to help nodes choose peers efficiently based on their requirements.

2. Peer to peer network

We aim to create a peer to peer network of nodes. After creation of the network, we will interconnect these nodes based on the user's requirements. This is done in order to create a Virtual Data Centre.

3. Virtualization

A pool of available resources on the network is created and this pool is used to create a Virtual Data Centre. We dynamically allocate a part of these resources to the nodes that require it. The user is essentially utilizing a part of this virtual pool of resources.

4. Parallelism

We split the process into small chunks and divide the execution of these process chunks by allocating them to nodes on the network that communicate by means of Message Passing. This is done in order to efficiently utilize resources and reduce time consumption.

WORKING

The resource information that is collected from nodes is advertised over the peer to peer ADK network and this information is referred by other nodes to select the peers that form their resource pool. These resource nodes are then allotted jobs based on the execution of the process in the master node. Here the master node is the node that selects the resource peers and forms a resource pool. It also monitors the status and progress of the allotted job to each resource peer. The master node waits for an acknowledgement to be sent by the resource peer once the job chunk is received by the node. If not, the master node selects another node for that particular job execution. If the node which is assigned a job fails, it forwards the task to another node on the network before it fails. This way the decision making mechanism is decentralised and the global working of the system is ensured by efficient communication between the nodes.

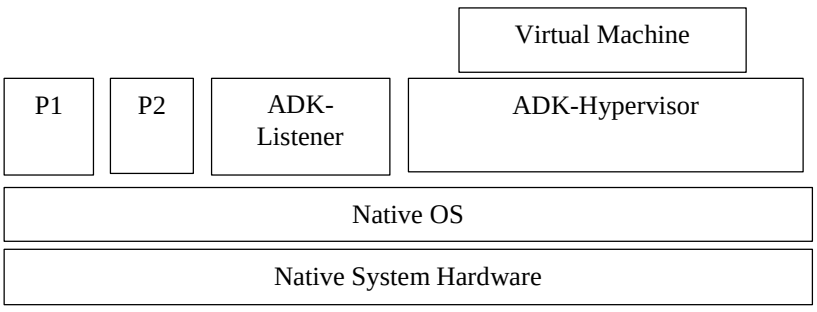

Fig1. ADK Architecture

ARCHITECTURE

The native OS that runs on the system manages the physical hardware of the node. The ADK hypervisor that runs on the OS is considered as a process. The hypervisor is built with the help of KVM hypervisor which is Type 2 hypervisor. The virtual machine that resides on the hypervisor is made from the resources donated by the resource peers of the network. The execution of processes in the virtual machine is handled by the hypervisor by distributing it to resource peers. The resource information is obtained by the native OS that runs on the system. This architecture is chosen to avoid deadlocks during the execution of processes. The ADK-Listener process that runs alongside the hypervisor listens to incoming job chunks and hands over the execution of the process to the operation system.

IV. CONCLUSION

Assisted Distributed Kernel, which aims at tapping into the underutilized computing resources available within an organization/community rather than dedicated servers, provides a promising alternative Cloud Computing solution for organizations and communities. Our work demon- states that the "no data center" solution indeed works. The "no data center" solution can gain highly competitive performance compared to its counter- part that depends on dedicated cloud servers. The most significant aspect of our survey so far is that using ADK we can setup a platform and have a door widely open for many exciting new research issues for the future. The Virtual Data Centre resource pool over which processes run in ADK is not dedicated but shared with native users/tasks. We need to devise a mechanism to provide cloud services reliably and efficiently, while keeping the services from interfering with the native users/tasks at member nodes. Another requirement from this idea is to have a robust, dynamic and efficient resource management and provisioning mechanism. The resource management and provisioning module should consider the dynamic and unreliable nature of the member hosts that contribute resources to the Virtual Data Centre We also need to investigate novel and efficient scheduling algorithms that consider the availability, location, and reliability of the member nodes used by the system to ensure efficient process execution. In addition, ADK requires strong security measures in order to insure the security of member nodes from malicious cloud client processes and client processes from malicious native users at member nodes.

V. REFERENCES

1] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, Dec. 2008.

[2] P. Domingues, P. Marques, and L. Silva, "Resource usage of windows computer laboratories," in *Parallel Processing, 2005. ICPP 2005 Workshops. International Conference Workshops on*. IEEE, 2005, pp. 469–476.

[3] Apache cloudstack. [Online]. Available: http://cloudstack. apache.org

[4] System information gatherer and reporter. [Online]. Available: https://support.hyperic.com/display/SIGAR/Home

[5] J. Che, Q. He, Q. Gao, and D. Huang, "Performance measuring and comparing of virtual machine monitors," in *Embedded and Ubiquitous Computing, 2008. EUC'08. IEEE/IFIP Interna- tional Conference on*, vol. 2. IEEE, 2008, pp. 381–386.

[6] J. J. Dongarra, P. Luszczek, and A. Petitet, "The linpack benchmark: past, present and future," *Concurrency and Com- putation: practice and experience*, vol. 15, no. 9, pp. 803–820, 2003.

[7] J. D. McCalpin, "Stream: Sustainable memory bandwidth in high performance computers," University of Virginia, Char- lottesville, Virginia, Tech. Rep., 1991-2007.

[8] G. Kirby, A. Dearle, A. Macdonald, and A. Fernandes, "An approach to ad hoc cloud computing," *arXiv preprint arXiv:1002.4738*, 2010.

[9] Tessema Mengistu, Abdulrahman Alahmadi, Abdullah Albuali, Yosef Alsenani and Dunren Che, "A No data centre solution to cloud computing" 2017 IEEE 10th International Conference on Cloud computing.