A "No Data Center" Solution to Cloud Computing

Kartik Chhabra[1*], Dr. Amit Kumar Malik[2]

1 MBM Engineering College, Jodhpur-342011, India.
2 Dean and R&D, LIET College Alwar 301208
Email: Kartikchhabra1297@gmail.com, Tel: +917690883327
Email: malik250280@gmail.com, Tel: +919887490536

Abstract-
Current Cloud Computing is primarily based on proprietary data centres, 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 centres, 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 underutilized 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 centre-based cloud provision model. In this paper, we present our opportunistic Cloud Computing system, called cuCloud, that runs on scavenged resources of underutilized PCs within an organization/community. Our system demonstrates that the "no data centre" solution indeed works. Besides proving our concept, model, and philosophy, our experimental results are highly encouraging. Keywords-Cloud Computing; IaaS; Volunteer Computing; No Data Centre Solution; Credit Union Model; Credit Union Cloud

Keywords*-Cloud Computing; IaaS; Volunteer Computing; No Data Centre Solution; Credit Union Model; Credit Union Cloud.*

1. INTRODUCTION

Current Cloud Computing services operate on the "data Center" model, where numerous dedicated servers are deployed to provide these services. Establishing and maintaining such data Centers for cloud computing is a costly endeavor, requiring specialized expertise and significant resources, including substantial power for cooling, redundant power sources to ensure uninterrupted service, and other infrastructure components. To illustrate, approximately 45% of the total cost of a data Center is dedicated to procuring servers, while an additional 25% is allocated to specialized infrastructure for fault tolerance, redundant power systems, cooling solutions, and backup batteries. Furthermore, electrical costs associated with operating the machines account for 15% of the total amortized cost [1]. In addition to the extensive use of servers within data Centers, there exist billions of Personal Computers (PCs) owned by individuals and organizations worldwide. Regrettably, these PCs are largely underutilized, typically in operation for only a few hours each day. Research findings have demonstrated that desktop computers owned by organizations remain idle for as much as 97% of their operational time [2]. We have advocated in our previous work [3] that we should view the untapped processing power and available disk space within this vast pool of underutilized PCs as valuable assets, comparable to financial assets. These resources can be consolidated and repurposed for the benefit of both society and individuals, mirroring the cooperative nature of a credit union. This argument serves as the foundation for an alternative model of Cloud Computing provision, known as the "Credit Union Cloud Model" (CUCM). Cloud services, primarily Infrastructure as a Service (IaaS), established under the CUCM framework, are commonly referred to as "Credit Union Clouds" (CU clouds). A distinctive feature of CUCM is its "no data Center" approach to delivering Cloud Computing services for institutions, organizations, or communities. In the current landscape of public clouds, often more accurately termed "vendor clouds" since they are provided by vendors relying on dedicated data Centers, concerns related to security and loss of control present significant barriers for traditional IT adoption of cloud technology. It is well understood that businesses with highly confidential data are naturally apprehensive about entrusting that data to third parties. On-premises private clouds may offer a potential solution to alleviate these concerns. However, the substantial upfront investment required to establish a private cloud infrastructure can be financially burdensome. Among the many advantages of CU clouds, affordability stands out as particularly compelling. This affordability translates to minimal additional expenses for acquiring and operating an on-premise cloud infrastructure. By eliminating the need for upfront purchases of cloud servers, which would otherwise be mandatory, organizations can save up to 45% of the costs associated with building a data Center. Additionally, the Credit Union Cloud infrastructure does not require additional cooling systems, resulting in an additional 15% savings on data Center cooling expenses. In essence, our credit union cloud management system offers a viable on-premises solution for Cloud Computing, catering to institutions and organizations that prioritize cost-effectiveness and security. The structure of this paper is as follows: Section II provides an overview of CUCM, our "no data Center" cloud model; Section III presents an implementation overview of CUCM as demonstrated in cuCloud, along with empirical results and analysis; Section IV delves into related works; and finally, Section V concludes the paper and outlines directions for future research.

II. Credit Union Cloud Model

The Credit Union Cloud Model represents a unique cloud provisioning approach designed to leverage the surplus of idle or underutilized computers for cloud service delivery, in contrast to the conventional practice centred around dedicated data centres. CU clouds operate on existing infrastructures with excess capacities that were not originally configured to support Cloud Computing. These personal computers (PCs) are not exclusively reserved for the cloud infrastructure; instead, they continue to serve their intended users for tasks like word processing or web browsing, referred to as local/native applications. The CU cloud model enables PCs within an organization to participate in a "cloud credit union." In doing so, they contribute their underutilized resources, such as CPU cycles or disk space, to augment the union's cloud resource pool, reinforcing its cloud infrastructure [3]. Consequently, these contributing PCs are referred to as member or volunteer nodes.

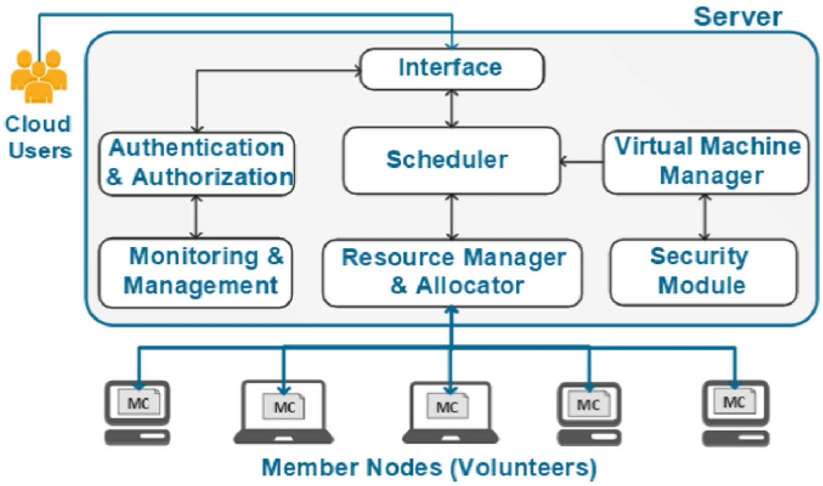

Fig. 1: Architecture of CUCM/cuCloud

UCM assumes a client/server architecture with member nodes as clients and dedicated management machine(s) as server(s). The server has different components as depicted in Fig.1. These components are briefly explained as follows. The Interface is the first port of communication between CU Cloud and its users/clients, whose access will be authenticated and authorized by the Authentication and Authorization module. The Resource Manager (RM) has the global picture of the resources that the cloud infrastructure has as a whole. The Resource Allocator (RA) component selects a list of suitable member nodes for the deployment of Virtual Machines (VMs) according to the resource requirements of the cloud customer, the Service Level Agreement (SLA), and the availability as well as reliability profiles of the member nodes. The Scheduler module accepts user requests and allocates or denies the requested resources in consultation with the Virtual Machine Manager (VMM) and the Resource Allocator. The VMM component handles the deployment of VMs on member nodes. The Security Module handles the security of the Virtual Machines. The Monitoring and Management module gives fine-grained resource information about the resources of the CU Cloud system.

On each member node that contributes resources to the CU Cloud system, there exists a crucial software component known as the Membership Controller (MC). The primary function of MC is to continuously monitor the utilization of resources on a member node and determine its membership status. An "active" status signifies the presence of sufficient resources to fulfill the minimum requirements for a virtual machine (VM), whereas an "inactive" status indicates an insufficiency of resources to support this need.

The Membership Controller performs periodic data collection and transmission to the server, conveying information about the available resources in terms of CPU, RAM, and Hard Disk. These resources are intended for contribution to the CU Cloud's resource pool.

The Membership Controller is composed of several key components:

1. Sensor Component: This component is responsible for actively monitoring the resource utilization of processes running on a member node. It then relays this resource usage information to the Reporter component.
2. Reporter Component: The Reporter component plays a pivotal role in determining the membership status of the node within the cloud infrastructure. It makes this determination based on the data received from the Sensor component. If the availability of resources, including RAM, CPU, and Hard Disk, exceeds or falls below a predefined threshold value, the Reporter component sends a message to the Resource Manager on the server. This message indicates whether the node should be categorized as an "active" or "inactive" member.
3. Virtual Environment Monitor Component: This component is responsible for the management of virtual machines (VMs) deployed on the member node. It oversees the handling and control of these VMs, ensuring their efficient operation within the CU Cloud environment.

In summary, the Membership Controller (MC) serves as a vital component on member nodes contributing resources to the CU Cloud system. It monitors resource usage, determines membership status, collects resource information for contribution, and consists of key components including the Sensor, Reporter, and Virtual Environment Monitor components.

III. Implementation and Experimentation

A. Implementation

As a cloud management system and a platform for delivering Infrastructure as a Service (IaaS) Services, CU Cloud is expected to encompass all the key characteristics of Cloud Computing, including elasticity, metering services, multitenancy, and more. In the initial phase of our project, we have created a preliminary version of the CU Cloud Management (CUCM) system, referred to as "cuCloud," by utilizing the capabilities of Apache Cloud Stack. Apache Cloud Stack is an open-source IaaS platform that effectively manages and orchestrates a variety of resources, including storage, network, and computing resources. It facilitates the construction of public or private IaaS compute clouds [4]. Cloud Stack is equipped with management server(s) that can efficiently oversee tens of thousands of physical servers spread across geographically distributed data centres. The Management Server within the Cloud Stack framework establishes communication with the compute nodes, which are the physical servers. This interaction occurs through the hypervisors such as Xen, KVM, Hyper-V, and others installed on these machines. However, since Cloud Stack primarily caters to dedicated data centres with exclusively dedicated hosts, we needed to enhance and modify the Cloud Stack management server to make it compatible with non-dedicated member nodes that form the backbone of the cuCloud system. To address this requirement, we developed a specialized component known as the "Ad Hoc component" and seamlessly integrated it into the Cloud Stack management server. This integration resulted in the formation of the cuCloud management server, which is tailored to handle the unique characteristics of the CU Cloud infrastructure. On the other hand, for the Sensor component of the Membership Controllers (MC) residing on member nodes, we harnessed the capabilities of SIGAR (System Information Gatherer and Reporter) [5]. SIGAR is utilized to gather vital system information, including the number of CPU cores, RAM capacity, idle CPU percentage, free memory percentage, and the available free hard disk space. This collected data is then relayed to the Reporter module of the Membership Controller (MC). Each instance of the Membership Controller running on a member node continuously monitors the resource usage of processes running on the host. This comprehensive resource usage information is crucial for making informed decisions regarding the membership status of the node within the CU Cloud infrastructure.

When the resource utilization at a member node falls below a specific threshold, the Membership Controller (MC) instance sends an "active" message to the Ad Hoc component on the cuCloud management server. Conversely, if resource utilization exceeds this threshold, the MC sends an "inactive" message. The Ad Hoc component on the cuCloud management server then updates the resource base of Cloud Stack based on the messages received from the MCs.In this preliminary version of cuCloud, the other components of Cloud Stack remain unchanged. However, due to the fully autonomous nature of member nodes, traditional type I hypervisors like Xen or Hyper-V cannot be utilized. Instead, we opted for KVM (Kernel-based Virtual Machine), a virtualization solution that operates alongside other applications on the member nodes enhances the traditional kernel and user modes of Linux by introducing a new process mode known as "guest." This guest mode includes its own kernel and user modes, responsible for executing code from guest operating systems [6]. This unique characteristic of KVM allows us to run Virtual Machines (VMs) concurrently with local applications on member nodes, facilitating the efficient utilization of resources within the cuCloud system.

B. Experimentation

To evaluate the feasibility of the Credit Union Cloud Model (CUCM), we conducted two series of experiments utilizing one server and four client machines. For the experiments, we utilized the modified CloudStack version 4.9.0, as discussed in the implementation section. Here are the details of the experimental setup: - The Management Server of CloudStack ran on a machine equipped with 8 GB of RAM, an Intel 8 Core i7 processor clocked at 2.4 GHz, and a 250GB hard disk.

- Each computing node was configured with 8 GB of RAM, an Intel 4-core i3 processor operating at 3.1 GHz, and a 250GB hard disk. All machines ran Ubuntu 14.04, were connected to a 16Gbps switch, and supported Intel hardware virtualization (VT-x).

For the first set of experiments, we established a dedicated CloudStack infrastructure consisting of one Management Server and four compute nodes. We measured the performance and resource usage, specifically CPU and RAM, using well-known benchmarks, namely LINPACK [7] for CPU performance and STREAM [8] for RAM performance.

In the second set of experiments, we used a modified CloudStack version, renamed as cuCloud, to implement our CUCM model. cuCloud also comprised one Management Server node and four member nodes. We conducted the same benchmarks in this environment to gather a comparable set of measurements for analysis. Membership in cuCloud was contingent on a member node having a CPU idle percentage greater than or equal to 70%. Importantly, these experiments with cuCloud were carried out while local users continued to utilize the machines. We designed five scenarios to compare the performance of CloudStack when running on dedicated machines (or data centres) against our cuCloud model, which relies on contributing member compute nodes. Here are the five scenarios:

1. Scenario 1: In a dedicated CloudStack infrastructure, we deployed one small VM instance (1 vCPU, 512MB RAM, 20GB HD) on one of the computing nodes.

2. Scenario 2: In a dedicated CloudStack infrastructure, we deployed one medium-sized VM instance (2 vCPU, 1GB RAM, 20GB HD) and two small VM instances on one of the compute nodes. For one of the instances, we conducted performance measurement tasks, while the other two instances were set to busy states with 40% and 60% CPU usage, respectively.

3. Scenario 3: This scenario mirrored Scenario 1 but was implemented on the cuCloud non-dedicated infrastructure.

4. Scenario 4: Similar to Scenario 2, this setup was on cuCloud. However, we deliberately made the member node hosting the benchmark tasks busy to induce the live migration of the VM with the performance measurement tasks.

5. Scenario 5: Identical to Scenario 4, but in this case, we purposefully induced two live migrations of the VM hosting the performance tasks.

6. The LINPACK benchmark was employed in these experiments, tailored for CPU-intensive computations involving large linear equations. We conducted the LINPACK benchmark with matrix dimensions of 5000x5000 and collected the average of 10 executions.

7. The experimental results are depicted in Figures 2 and 3. Figure 2 illustrates that there is almost no discernible difference between running a task on a CloudStack dedicated compute node and on a cuCloud shared member node, provided there are sufficient resources to execute the task. However, Figure 3 demonstrates that when one or more migrations are involved, tasks running on cuCloud may take longer. This delay is a direct result of the induced migration of VMs. Notably, the performance gap between scenarios with one migration and two migrations appears relatively small (12.64 seconds vs. 12.71 seconds), although this may not hold as a general rule.

8. [Figures 2 and 3 would be included here to visually represent the experimental results.]

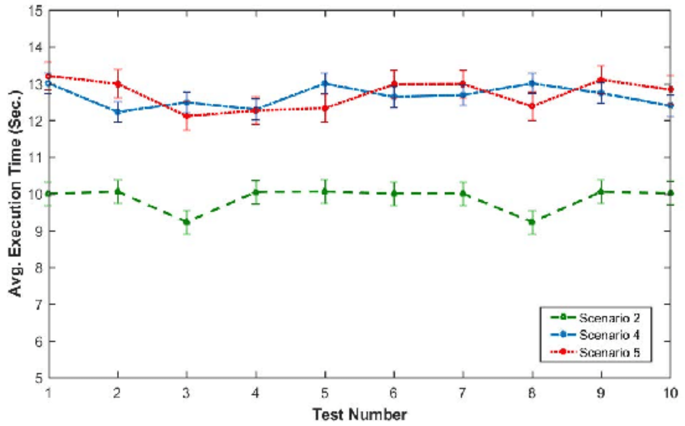

Fig. 2: LINPACK: Dedicated vs. cuCloud (no migration)

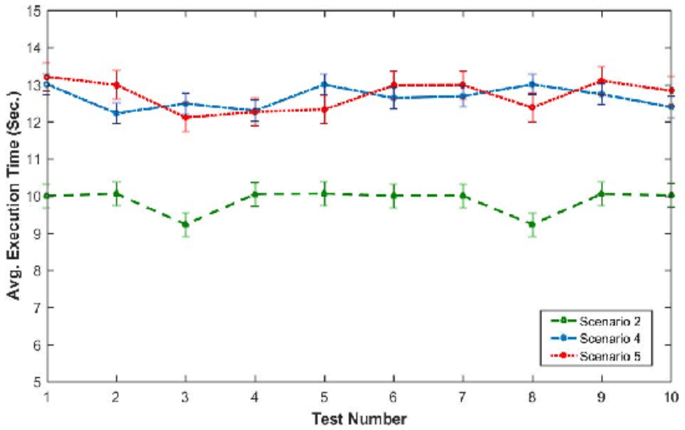

Fig. 3: LINPACK: Dedicated vs. cuCloud (with migration)

The STREAM benchmark is primarily designed to assess memory bandwidth by employing four fundamental operations: Add, Copy, Scale, and Triad. In our experiments, the STREAM benchmark was configured with an array size of 2,000,000.

Figure 4 provides a visual representation of the average bandwidth usage across 10 trials of running the STREAM benchmark under the five different scenarios. As depicted in Figure 4, the bandwidth usage for the four operations in the first three scenarios remains nearly identical. However, it is evident that VM migration induces a noticeable increase in bandwidth usage. This observation underscores the impact of VM migrations on memory bandwidth within the cuCloud system.

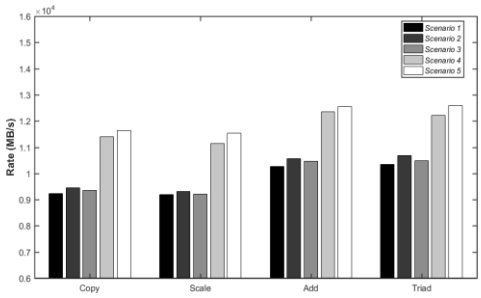

Fig. 4: STREAM Bandwidth Rate

Our preliminary implementation of the Credit Union Cloud Model (CUCM) and the experiments conducted with both cuCloud and CloudStack demonstrate not only the viability of the CUCM concept (i.e., the "no data Center" cloud solution) but also its potential as a promising alternative approach to Cloud Computing, offering numerous advantages.

IV. Related Works

In the realm of research related to the development of Cloud Computing models based on spare resources of personal computers (PCs), there have been relatively few works exploring this direction. One notable work in this domain is the concept of the "ad hoc cloud," which was reported in [9]. This work delved into various research aspects related to cloud provisioning using general-purpose computers. The proposed architecture for this ad hoc cloud infrastructure included several components: one for creating and destroying cloud elements, another for monitoring the effects of created cloud elements, one for handling Quality of Service (QoS) issues, and one for executing allocated tasks. However, the authors did not provide an actual implementation of their ad hoc cloud system.Another work in the realm of non-dedicated data Center-based Cloud Computing is discussed in [10]. This work aimed to investigate the feasibility, reliability, and performance of ad hoc Cloud Computing infrastructures. The ad hoc cloud system in this case was designed as a client/server system and was based on the well-known volunteer computing system BOINC, enhanced with virtualization support called V-BOINC. The server component of this system comprised three subcomponents: Cloud Interface, VM Service, and Job Service. The client was responsible for accepting and reliably executing jobs. The research concluded that ad hoc cloud is not only feasible but also a viable alternative to existing data centre-based Cloud Computing systems. However, the authors did not delve into aspects related to elasticity, multitenancy, and other critical characteristics of such a system.

In summary, there have been limited works exploring the concept of utilizing spare resources of PCs for Cloud Computing, and these studies have laid the foundation for novel approaches like the Credit Union Cloud Model (CUCM) that we have presented.

V. Conclusion and Future Work

The Credit Union Cloud Model (CUCM), designed to harness the underutilized computing resources within an organization or community instead of relying solely on dedicated servers, presents a promising alternative solution for Cloud Computing in various settings. Our work has not only validated the feasibility of the "no data Center" approach but has also demonstrated its highly competitive performance compared to traditional setups dependent on dedicated cloud servers. One of the most significant aspects of our work is the platform created by cuCloud, which opens the door to numerous exciting research opportunities for the future. Several critical research issues and areas of exploration lie ahead:

1. **Resource Sharing and Isolation**: cuCloud operates with a resource pool that is shared between virtual machines (VMs) and native users/tasks. It is essential to devise mechanisms that ensure cloud services run reliably and efficiently while preventing interference with native users/tasks on member nodes.
2. **Dynamic Resource Management**: CUCM demands a robust, dynamic, and efficient resource management and provisioning mechanism. This module must account for the dynamic and potentially unreliable nature of the member hosts contributing resources to cuCloud's resource pool.
3. **Scheduling Algorithms**: Novel and efficient scheduling algorithms are required, taking into consideration the availability, location, and reliability of member nodes used to deploy VMs. These algorithms play a crucial role in optimizing the allocation of resources.
4. **Security Measures**: cuCloud must implement strong security measures to safeguard member nodes against malicious cloud client processes and ensure the security of client VMs from potentially malicious native users on member nodes.

In conclusion, our work with the Credit Union Cloud Model has laid the foundation for a promising approach to Cloud Computing. The challenges and opportunities that lie ahead in this research direction are numerous and hold the potential to reshape the landscape of cloud service provisioning.

References [1]

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] D. Che and W. C. Hou, “A novel "credit union" model of cloud computing,” in International Conference on Digital Information and Communication Technology and Its Applications. Springer, 2011, pp. 714–727.

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

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

[6] 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 International Conference on, vol. 2. IEEE, 2008, pp. 381–386.

[7] J. J. Dongarra, P. Luszczek, and A. Petitet, “The linpack benchmark: past, present and future,” Concurrency and Computation: practice and experience, vol. 15, no. 9, pp. 803–820, 2003.
[8] J. D. McCalpin, “Stream: Sustainable memory bandwidth in high performance computers,” University of Virginia, Charlottesville, Virginia, Tech. Rep., 1991-2007.
[9] G. Kirby, A. Dearle, A. Macdonald, and A. Fernandes, “An approach to ad hoc cloud computing,” arXiv preprint arXiv:1002.4738, 2010.
[10] G. McGilvary, “Ad hoc cloud comp