

A Novel “Credit Union” Model of Cloud Computing

Dunren Che and Wen-Chi Hou

Department of Computer Science
Southern Illinois University Carbondale,
Illinois 62901, USA
{dche,hou}@cs.siu.edu

Abstract. Cloud Computing is drawing people’s attention from all walks of the IT world. It promises significant reduction of cost among many other advantages as it proclaims, including increased availability, fast provision, on-demand, and pay-per-use, etc. This paper presents a novel model of Cloud Computing, named “Credit Union” model (referred to as the CU model or CUM for short). This model is motivated by the cooperative business model of the many credit unions that have been widely practiced as a type of financial institutions world-wide. The CU model aims at utilizing the vast, underutilized computing resources in homes and offices, and transforming them into a self-provisioned community cloud that mimics the business model of a credit union, i.e., membership and credits are obtained by contributing spare computing resources. Clouds built based on the CU model, referred to as CU clouds, bear the following advantageous characteristics comparing to the general clouds: complete vendor independence, improved availability (due to reduced internet-dependence), better security, and superb sustainability (green computing). This paper expounds the principles and motivations of the CU model, addresses its implementation architecture and related issues, and outlooks prospect applications.

Keywords: Cloud Computing, Cloud Computing Model, Cloud Architecture, Green Computing, Sustainability, Community Cloud, Community Cloud Computing.

1 Introduction

Cloud Computing was the most discussed topic in the IT industry and academia in 2010, and it will likely remain to be the hottest IT topic this year and for the years to come. Cloud Computing proclaims many virtues or advantages over prior computing paradigms and models. Among them, cost reduction is probably the most attractive, at least to the CFOs (Executive Financial Officers). What are really tempting to the CFOs are the saved capital expenses that can then be turned into operational expenses. Additional cost-reduction may be obtained via improved hardware utilization, guaranteed availability (accompanied by the saved cost of failures and recovery), and the utility payment feature (i.e., the so-called pay-as-you-go model) of Cloud Computing. And for cloud service providers/vendors, cost reduction is often

realized through economy of scale. Improved hardware utilization straightforwardly implies less hardware needed, less power consumed, and less electric garbage to be processed. Therefore Cloud Computing is considered an enabling technology for green computing. While the current Cloud Computing technologies already promote environmental sustainability, we believe we can go a lot farther along the line of sustainability and practice green computing more thoroughly through a new model of Cloud Computing – the "Credit Union" Model – that forms the theme of this paper.

We see unused (and often wasted) computing resources everywhere and everyday - whether it is a power PC in office or a notebook at home, it constantly has idle CPU cycles and spare memory and disk spaces. After office hours, especially the period from mid-night to early morning, most computers are completely idle or turned off. However, on the other side of our planet, a new day is dawning, filled with complex activities that can only be perfectly and promptly accomplished with more CPU cycles and more memory spaces. Current Cloud Computing and the accordingly developed solutions cannot simply fit in here because they were not designed to utilize the vast amount, underutilized computing resources possessed by individuals and organizations for the good of the individuals and the communities.

Excessive computing resources are important assets to individuals and to the global village as a whole. To the resource owners, individuals or organizations, these assets are just like one's spare money. Spare money, if not invested, of course, will not yield any interests, but it does not flow away (let's intentionally turn a blind eye to the inflation that seems existent in every economy of this world). However, the matter is far worse when it comes to unused spare computing resources – these resources either completely vanish (like CPU cycles) or fast evaporate (depreciating their values in exponential speed) – which at the end causes a sheer waste of what might have started as a precious portion of one's capital spending.

Being practiced widely and successfully, credit unions are a type of cooperative financial institutions that are owned and controlled by their members and operated for the purpose of promoting thrift, providing credit at reasonable rates and other financial services to their members. Many credit unions exist to further community development or sustainable international development on a local level.

Our comparison between spare computing resources and spare money inspires creation of a special credit union so that both individuals and organizations can contribute their spare computing resources and transform into community benefits or individuals' credits or even monetary interests. In other words, we can construct a computing infrastructure that is community-based, relying on members' contribution of their excessive computing resources, such as CPU cycles, memory and disk spaces. Such a community computing infrastructure cannot be readily provisioned by current cloud vendors using existing technologies. CU clouds can only be made a reality via integration of multiple existing computing paradigms and technologies, including the fast fledging Cloud Computing, Grid Computing, and Peer-to-Peer (P2P) computing.

In this paper we present a novel, "Credit Union" model of Cloud Computing that is a specialized Cloud Computing model to serve the particular needs of a community and its members who possess spare computing resources and allow them to invest that their spare resources (just like their spare moneys) for the common good of the community and/or extra individual benefits (earned in the form of credits). Such developed community clouds (or CU clouds) may be made open to the general public

to gain profits from outside; and the community members who hold sufficient credits may also choose to exchange for monetary benefits. Construction of CU clouds requires integration and utilization of several other related computing technologies that are reviewed in the next section.

The remainder of this paper is organized as follows: Section 2 reviews related technologies and related work. Section 3 defines our “credit union” model of Cloud Computing and discusses its relationships with other relevant computing models. Section 4 analyzes the scenarios of CU cloud applications and derives important characteristics that have influence on the architectural design of CU clouds. Section 5 presents illustrative architecture of CU clouds. Section 6 summarizes our discussion and points out future directions.

2 Related Technologies and Work

Our “credit union” model of Cloud Computing is not built from scratch, but on a series of underling, enabling technologies. This section, as a preliminary for the subsequent discussion, reviews the key supporting technologies and related work, with the hope of setting up a meaningful discussion context and clearing up possible confusions.

2.1 Cloud Computing

Although being widely discussed as a buzzword, *Cloud Computing* to different people still leads to rather different interpretations. We adopt the definition given by the National Institute of Standards and Technology (NIST) in 2009 [8], which we believe represents the most accepted definition for Cloud Computing [1]:

*Cloud Computing is a **model** for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*

Obviously, the term Cloud Computing refers to a general computing model or paradigm, being accompanied by a rich set of enabling/supporting technologies and products and services.

It is interesting to point out the key characteristics, delivery models, and deployment modes of Cloud Computing [2]:

The **key characteristics** of Cloud Computing include *on-demand self-service*, *ubiquitous network access*, *location independent resource pooling*, *rapid elasticity*, and *pay-per-use* (or pay-as-you-go). There are three primary **delivery models**, *cloud software as a service (SaaS)*, *cloud platform as a service (PaaS)*, and *cloud infrastructure as a service (IaaS)*, and four distinct **deployment models**, *private cloud*, *community cloud*, *public cloud*, and *hybrid cloud*.

There are several other computing models that are often regarded as the supporting technologies or closely related technologies to the Cloud Computing paradigm, from time to time causing confusions. We next briefly review these supporting or related technologies and highlight their characteristics.

2.2 Other Related Computing Models

A *Distributed System* consists of multiple autonomous computers that communicate through a computer network, and interact with each other in order to achieve a common goal. *Distributed Computing* generally refers to the use of a distributed system to solve a computational problem that is divided into many tasks, each of which is solved by one computer of the distributed system.

Generally speaking, *Grid Computing* is a form of distributed computing and parallel computing, whereby a 'super and virtual computer' is (formed) composed of a cluster of networked, loosely coupled computers acting in concert to perform very large tasks. The goal of Grid Computing is to provide a consolidated high-performance computing system based on loosely coupled storage, networking and parallel processing functions linked by high bandwidth interconnects.

Obviously, distributed computing denotes a rather general concept (or model) of computing. Both Cloud Computing and Grid Computing can be regarded as a particular kind of distributed computing; and both aim at delivering abstracted computing resources. But the two shall not be confused though their distinction is fairly subtle.

The comparison between the two models made by Frischbier and Petrov [1] are interesting. We quote and adapt their comparisons below:

The two paradigms (Cloud Computing and Grid Computing) differ in their particular approaches and subjects:

- (i) Cloud Computing aims at serving multiple users at the same time and elastically via resource *pooling* while Grid Computing is intended to deliver functionality at a scale and quality equivalent to a *supercomputer* via a *queuing* system;
- (ii) Grids consist of resources owned and operated by different organizations while clouds are usually under a single organization's control;
- (iii) Cloud services can be obtained by using a standardized interface over a network, while grids typically require running the grid fabric software locally (the fabric software was designed for unifying the interconnected grid nodes).

Here we want to point out that our "credit union" Cloud Computing model implicates a notion that is quite on the opposite of item (ii). This is just one of the several aspects that make our "credit union" model different from the general Cloud Computing model that most people have on their minds.

Two other computing notions are often referred to when Cloud Computing is discussed -- *utility computing* and *service-oriented computing* (typically manifested as *Software-as-a-Service* or *SaaS* for short). These two notions are rather generic terms, primarily referring to two aspects (or characteristics) of Cloud Computing. To help clearing up possible confusions, we provide following the definitions:

Utility computing — the packaging of computing resources, such as computation and storage, as a *metered service* similar to a traditional public utility, such as electricity.

Service-oriented computing – Cloud computing provides services related to computing and, in a reciprocal manner, *service-oriented computing* consists of the computing techniques that operate on software-as-a-service (SaaS).

It shall be clear that utility computing emphasizes the “metered” feature while service-oriented computing highlights on the “service-oriented” feature, which are both manifested by Cloud Computing.

2.3 Related Work

There is “tons” of reported work related to Cloud Computing. The leading Cloud Computing providers include Amazon [11, 12], Google [9], Microsoft [10], Salesforce [13], and more. The Linux website <http://linux.sys-con.com/node/1386896> even listed The Top 250 Players in the Cloud Computing in Year 2010, yet at the same time almost everyone agrees that Cloud Computing is still in its infancy. There are also numerous well-written surveys introducing and discussing Cloud Computing and related technologies [1, 2, 3, 4, 7, 8, 14]. So in this section, we are not going to review Cloud Computing in a general way. Instead, in the following we particularly look at SETI@home [5] and Seattle [6], two important projects that might be considered as overlapping somehow with our CU model and our CU cloud project that is currently being initiated. The overlapping is actually minimal as explained below.

SETI@home: SETI@home (“Search for Extra-Terrestrial Intelligence at home”) is an internet-based public volunteer computing project employing the BOINC software platform, hosted by the Space Sciences Laboratory, at the University of California, Berkeley, in the United States. Its purpose is to analyze radio signals, searching for signs of extra-terrestrial intelligence, and is one of many activities undertaken as part of SETI. Technically, SETI@home is a large Internet-based distributed system (project) for scientific computing; being an aspect of the P2P paradigm, involving shifting resource-intensive functions from central servers to workstations and home PCs. It uses millions of voluntary computers in homes and offices around the world to accomplish its computing tasks. Although it has not found signs of extraterrestrial life, the project has contributed to the IT industry and academia with the so-called “public-resource computing” model. In SETI@home, the client program repeatedly gets a work unit from the data/result server, analyzes it, then returns the results (candidate signals) to the server. The client can be configured to compute only when its host is idle or to run constantly at a low priority and as a background process. SETI@home does not contain any of the major ingredients of Cloud Computing but it is related to our work in that both use voluntary computers in office and homes with excessive computing capacities. SETI exercises *public-resource computing* for scientific discovery while our CU model aims at utilizing excessive computing resources owned by individuals and/or organizations for the common good of a community and/or the benefits of the participating individuals in the community.

Seattle: Seattle is a free, education research platform, implemented as a common denominator of Cloud Computing, Grid Computing, P2P network, distributed systems and networking on diverse platform types. As a project, Seattle reflects a *community-driven* effort that depends on resources donated by users of the software (and as such is free to use). A user (typically an educator) can install Seattle onto their personal

computer to enable Seattle programs to run using a portion of the computer’s resources. As an educational platform, Seattle provides many pedagogical contexts ranging from courses in Cloud Computing, Networking, and Distributed Systems, to Parallel Programming, Grid Computing, and P2P Computing. Seattle itself is not generally regarded as pursuing Cloud Computing. Seattle exposes locality and primarily provides a distributed system environment as an educational platform. Seattle uses configurable *sandboxes* to securely execute user code and monitors the overall use of key resources. Seattle is interesting to us because it relies also on public resources (resources are donated to Seattle and counted as credit for using Seattle after an instructor installs and runs the installer software on a local machine); in addition, Seattle has a preexisting base of installed computers to start with, that we might need as well for effective implementation of our CU clouds.

3 Defining the “Credit Union” Model of Cloud Computing

Much of the popularity of Cloud Computing attributes to its promise for cost reduction (from both the providers’ side and the consumers’ side). The reduction of cost directly contributes to environment-friendliness and thus Cloud Computing is said to be “green computing”. Even though, we observe that huge amount of CPU cycles and memory and disk spaces are underutilized and/or wasted in offices and at homes, day and night (especially after midnight). It would make greater contribution to the environmental sustainability that Cloud Computing has already promised if we can recycle and reuse the large amount of unused extra computing resources. The current Cloud Computing model and techniques were not particularly designed for increasing the resource utilization of the vast amount of client computers. We are thus motivated to develop a specialized Cloud Computing mode and the accompanying techniques to carry the sustainability feature of Cloud Computing a big step forward.

Credit unions as a type of financial institutions have been practiced very successfully for the good of a community and the benefits of its members by attracting and reinvesting the spare money owned by its members. The principles and basic ideas practiced by credit unions can be extended to and incorporated by Cloud Computing. This consideration leads to a new, specialized Cloud Computing model, which we refer to as the “credit union” model (CUM). The goal of this model is to consolidate and utilize the unused excessive computing resources possessed by individuals and/or organizations in a community. This novel Cloud Computing model is defined with more details as follows.

Definition of the “Credit Union” Model of Cloud Computing (CUM): The “credit union” model of Cloud Computing is a specialized model of the general Cloud Computing paradigm. It relies on the excessive computing resources owned by the individuals and/or organizations in a community. In the setting of the CU model, computing resources are contributed by individuals and/or organization either free or for credits. Clouds built according to the CU model are referred to as CU clouds. CU clouds stick to the general principles of Cloud Computing but require specialized architecture and implementation considerations for realizing its goal.

CU clouds aim at utilizing the elapsing CPU cycles and other excessive computing resources such as memory and disk spaces owned by the individuals

and/or organizations in a community; the community is not necessarily limited by the geographical boundaries. The developed CU clouds are typically deployed as a consolidated computing facility to be consumed by the community for the common good of the community, or the members (accordingly to the respective amounts of credits earned); alternatively, CU clouds as services can also open to the general public for gaining profits from the outside world – in other words, the community can independently function as a cloud service provider or simply sell its consolidated cloud infrastructures and facilities to a larger, enterprise cloud service provider.

The community oriented feature of a CU cloud makes it easy to get confused with the notion of community clouds in the general sense, but they are different. For a community cloud, in the general sense, the community and its members act only as the consumers of the cloud services that are typically provided by an enterprise cloud service provider; however, for a CU cloud, the community and its members not only are the consumers but also the service provider and owner -- they collectively own everything of their cloud, from its infrastructures, facilities, hardware and software, to full control and full management of everything of the cloud. The members of a CU cloud are voluntary participants – they are mainly attracted by the opportunity of earning credits that may be traded for monetary benefits or free use of the cloud facilities. Credits are earned through “depositing” (contributing) their excessive computing resources to the community, and, of course, the earned credits can also be donated to the community for the common good of the community.

We generally assume the opinion of Foster, *et al.* that Cloud Computing not only overlaps with Grid Computing, it shall be evolved out of Grid Computing and rely on Grid Computing as its backbone and infrastructure support [4]. Nevertheless, the two paradigms are far from being identical. Imagine that we can pull Grid Computing to one side and pull Cloud Computing to the other side, then our CU cloud model shall sit in the middle of the two paradigms, slightly shifted back from the cloud side to the grid side as shown in Figure 1, which is adopted from [4] but modified for illustrating the relationships between the new notion of CU cloud and other related computing models. In Figure 1, Web 2.0 covers almost the whole spectrum of service-oriented applications, where Cloud Computing lies at the large-scale side. Supercomputing and Cluster Computing have been more focused on traditional non-service applications. Grid Computing overlaps with all these fields and is generally considered of lesser scale than supercomputers and clouds. CU clouds sit at the low scale end of clouds but overlaps more with grid computing; in other words, CU clouds are clouds, but of less scale, and appears to have more in common with grids compared to the general cloud computing.

More specifically, what is the little extra common ground that CU clouds now find with grids? This question is answered by review the difference between clouds and grids [4]:

Clouds mostly comprise dedicated data centers belonging to the same organization, and within each data center, hardware and software configurations and supporting platforms are in general more *homogeneous* as compared with those in grid environments. In contrast, grids however build on the assumption that the resources are *heterogeneous* and *dynamic*, and each grid site may have its own administration domain and operation autonomy. However, in the context of CU clouds, we face even a more significantly heterogeneous environment of computing resources that are

possessed by individuals/organizations with exclusive privilege for use. From this point of view, CU clouds shall be relocated more toward the territory of grids. This characteristic of CU clouds raises a great implementation challenge – distributing coordination and load balancing within a highly dynamic and heterogeneous system.

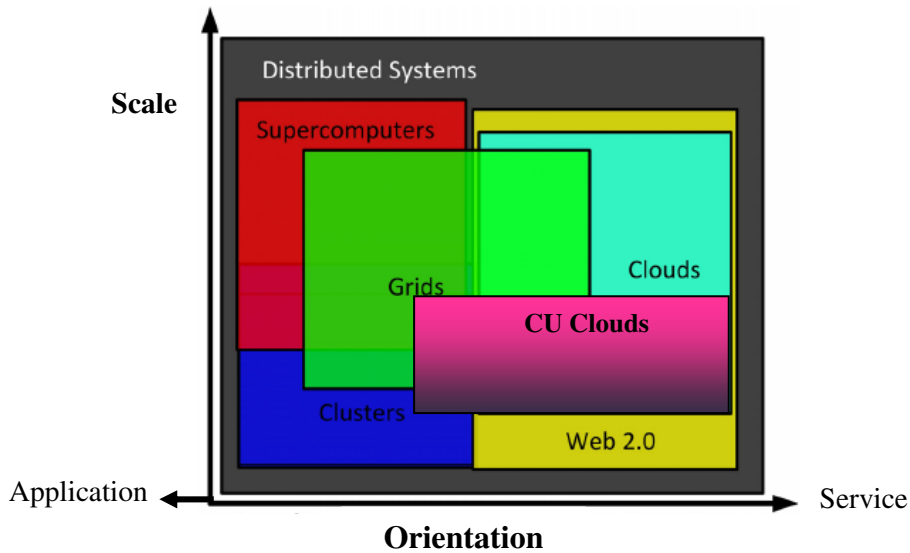


Fig. 1. Relationships between CU clouds and other related computing domains (adapted and modified from [4])

4 Prospect Applications and Implication on Implementation

The CU model has opened a brighter and broader horizon for future applications of Cloud Computing. It is more promising than the vendor-provision model of today's Cloud Computing because CU clouds have the potential to diminish the concerns and flaws associated with today's Cloud Computing.

As today's cloud services are exclusively provided by enterprise vendors such as Google, Amazon, and Microsoft, *vendor cloud* and *vendor Cloud Computing* are equivalent terms to today's Cloud Computing, which straightforwardly reflect the *vendor provision* nature of today's clouds. Cloud Computing is generally accredited of the feature of "green computing" owing to virtualization that has been successfully used to maximize resource utilization. However, with today's Cloud Computing, such effort (on maximizing resource utilization) has only been pursued on vendors' side where cloud resources are centralized and owned by the vendors; resource utilization on client machines has never been a concern of today's Cloud Computing.

Our CU model will help diminish the following aspects of concerns regarding today's Cloud Computing:

Security and Privacy: While the security concern of cloud consumers is not necessarily endemic only to Cloud Computing (noticing that vendors have been taking every means to protect consumer data and applications on their clouds), privacy concern seems an issue that can never be solved by the vendor-provision model of today's Cloud Computing. No matter what advance is made, users would never be without concerns when they run their mission-critical applications and/or store their sensitive data on clouds. We doubt US government departments such as DOD, CIA, and FBI will ever completely trust any vendor clouds, though they long for the convenience and benefits promised by Cloud Computing as any other consumers. These government branches would more willingly accept CU clouds that will situate on their own premises under their full control.

Cascading failure: Being centrally managed and maintained by best trained professionals, vendor clouds generally enjoy the good fame of improved availability. That does not mean that the clouds are absolutely isolated from failures, and when cloud failure indeed occurs, it causes *cascading effect* to all dependent applications and services. However, users want to get their things done even when the Internet and clouds are down or the network communication is slow. In such scenarios, CU clouds demonstrate great competency and advantage over vendor clouds. Moreover, in extreme situations (e.g., at war times) a community may want to completely shut off connection from the global domain of Internet, only CU clouds may give this security option without affect ongoing applications.

Underutilization on client resources: Maximization of resource utilization is only achieved by vendor clouds for the resources on the vendor's side. The CU cloud Computing can perfectly unify vendor resources and client resources, realize utilization maximization on all resources, and exercise the sustainability of green computing to the fullest.

By and large, CU clouds overcome several innate flaws (more accurately, most of the flaws are resulted from the *vendor-provision model* of today's Cloud Computing) and demonstrate undisputable advantages over the current Cloud Computing model, yet still retain all the advantage and benefits promised by Cloud Computing. CU clouds have a far better potential than vendor clouds for wide acceptance with regard to all walks of applications, from private sectors to the vast amount of communities and organizations at all levels, including government departments and those having extremely high demand for confidentiality and privacy.

As for application of CU clouds in public institutions in the United States, state laws typically disallow public assets (allocated to public universities, for example) from being used for other than the original purposes. Nevertheless, public institutions can use services delivered by their own CU clouds to enhance their original missions (education and/or research). Otherwise, the cloud services they need must be purchased from external, enterprise providers, which certainly means extra budgets must be allocated. Public institutions may choose to use their self-provisioned CU clouds to promote non-profit collaborations with other local institutions at all levels, including primary and secondary schools, and community colleges. Relatively large local communities such as community colleges may deploy their CU clouds, and multiple community CU clouds may further form a cloud federation at a larger scale

to better serve the varied needs of all potential consumers (individuals and organizations, local and distant) at all levels.

Next we derive two aspects of expectations from the scenarios of CU cloud applications that have implication to the architecture and implementation of CU clouds.

First, a community with CU clouds is not an enterprise cloud service provider (but this does not eliminate the possibility that the community evolves into an enterprise cloud service provider in the future just as Amazon.com, which though might be considered as an exceptional example). The primary computing resources available to community CU clouds are extracted and consolidated from autonomous machines owned by various individuals and/or organizations and are geographically distributed. This important feature of CU clouds requires every participating computer to install and run specially designed software (we can vividly call as *membership software* or a *virtual box*) that collects and virtualizes excessive resources from each participating computer. A good metaphor for such membership software might be a boy scout who participates in a food drive program, comes to your house and collects the (spare) food items you are willing to contribute.

Second, a community typically does not have a dedicated cluster of commodity computers to support their community CU clouds. Once a community decides to build a CU cloud, for the sake of overall performance (regarding system monitoring, distributing coordination, and load balancing, etc.), procurement of a few dedicated machines might be necessary or at least recommended. They will be used to represent the community clouds in the cyberspace, serving as an access portal for internal consumers and also for potential outside customers. Overall, cloud resource consolidation and management are best carried out at such dedicated machines (which may alternatively be delegated to a few relatively powerful machines contributed to the community cloud especially in case of failure or experiencing severe performance degradation). The heterogeneous and highly dynamic nature of CU clouds (hosted by varied machines, which each runs a different pack of software and yet needs to support a range of fast changing and privileged local applications first) raises a greater technological challenge that the current cloud vendors have not been confronted with.

5 Architecture and Implementation Issues

The resource environment (including hardware, software, and applications) that a CU cloud is built on are highly heterogeneous and dynamic. The unused extra computing resources at each participating computer must be extracted and consolidated through virtualization and abstraction carefully carried out locally. Because each machine must be configured in a way not to affect its local users' daily work (we refer to these users as "native users" in the context of a CU cloud), at the hardware level, the hosted architecture [7] for virtualization becomes the only rational choice, which provides partitioning on top of a standard operation system, leaving the entire working environment of the native users completely intact. In contrast, the other popular virtualization architecture, bare-metal (Hypervisor) architecture [7], though often claims more efficiency, cannot fit into to the particular scenario of CU clouds as the hypervisors need to completely take over the original host operating system that, however, is still required by the native users and their applications.

Figure 2 illustrates the architecture for implementing a *core element* in a CU cloud (a core element is a node in a CU cloud denoting the abstracted and encapsulated resources from a participating machine via virtualization). The host operating system and all local applications as well as their running environments remain intact. The virtualization module (and added layer) above each participating machine provides the ability to emulate (multiple) guest operating systems that are open to cloud applications. As the host machine is dedicated to the CU clouds, the number of virtual machine instances (VMI), at one core element is usually limited to just a few; when the host machine is found having insufficient spare resources (e.g., CPU cycles), the number of virtual instances spawned from that machine shall be accordingly reduced (even to zero in case of severe resource completion). The virtual machine management (VMM) module provides a local management console, giving the owner or native users the means to interfere the resource completion between native applications and alien applications (i.e., cloud applications). Proper implementation of the virtualization layer shall not make the native users feel obvious performance degradation due to completion between native applications and alien applications. The virtualization layer (also called Hypervisor as in [7]) gracefully multi-plexes and encapsulates the computing resources (when their utilization rate is stably below a certain threshold). As an option, device emulation may be incorporated into the Hypervisor or VM.

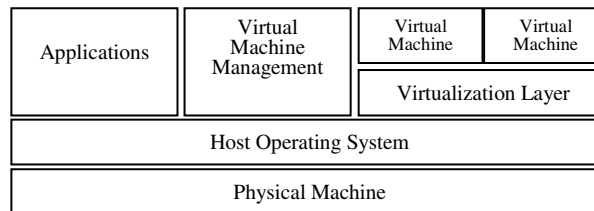


Fig. 2. Illustrative structure of a core element in a CU cloud

We take a similar way to consolidate the core elements as in [15]: taking the core element node of Figure 2 and multiply it on a physical network (typically the Internet), orchestrating the management over the entire infrastructure, and providing front-end coordination and load balancing for incoming connections with caching and filtering; this results in a whole range of consolidated virtual machine instances, which altogether are referred to as the *virtual infrastructure* hosting our CU clouds. The overall architecture that our CU clouds are situated on is depicted in Figure 3. Due to the fact that a CU cloud is physically hosted by a group of networked autonomous machines possessed by individuals or organizations, in the architecture of CU cloud (Figure 3), native users are granted (by the locally installed membership software) privileged accesses to respective host machines, as denoted by the module named “Desktop” on the upper right side of the structure (see Figure 3). As pointed out earlier, in the setting of CU clouds, we best have a few dedicated commodity machines installed to serve community-wide, virtual infrastructure management. The left column in the architecture (see Figure 3) explicitly indicates the infrastructure management module.

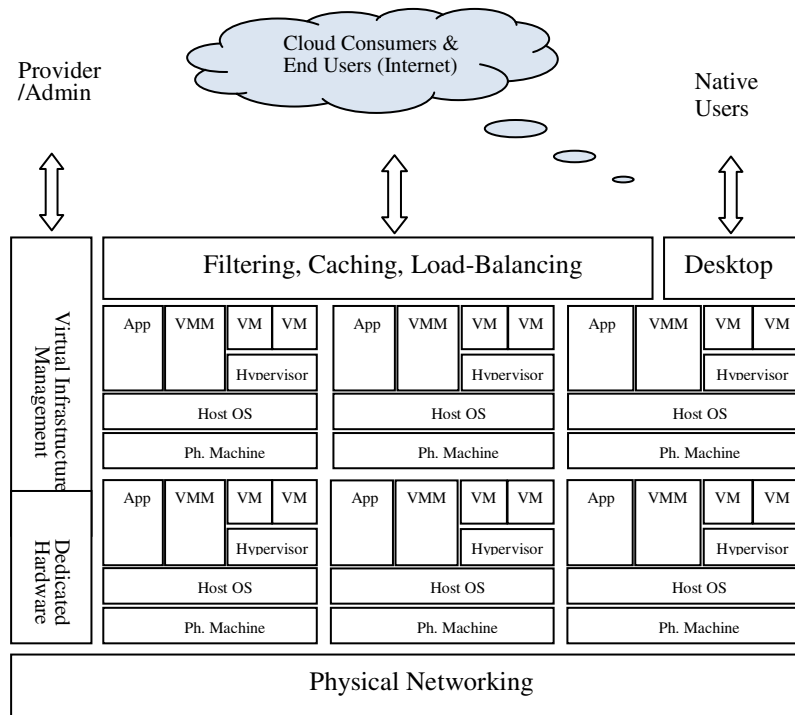


Fig. 3. Illustrative architecture of CU clouds

One prominent feature in the construction of CU clouds is resource-sharing between native users and alien applications, which happens at each core element node. While the idea of provisioning such a community-backed cloud infrastructure is exciting, the highly dynamic and heteronomous nature in a CU cloud environment implies new challenges. Virtualization must consider how to harmoniously reconcile the conflicts (resources competition) between native applications and alien applications. The inherent characteristics of a CU cloud determine that the resource management policy must first guarantee smooth running of all local applications. Furthermore, the virtualization module must be designed with the capability to automatically adjust the resources extracted at each core element node as cloud consumable resources. For example, each Hypervisor shall be able to switch to the full potential of each core element when the resource request from local applications drops to the minimum, which typically happens at the end of each work day for office computers and at the beginning of each work day for home computers.

The dynamic feature of the CU cloud infrastructure more resembles that in a grid environment. The relatively mature technologies developed by the grid community can be adapted for the construction of future CU clouds.

Before we end this section, let's summarize the key features of CU clouds as compared to vender clouds:

- **Different assumption:** CU clouds build on the assumption that computing resources are separately owned by the members of a community, and are highly dynamic and heterogeneous; typically each resource node is an autonomous node and has its own administration domain, comparable to grids, but more challenging for management because we have to coordinate local applications and large number of cloud applications (while in a grid environment, it is a small number of large applications). The governance in a CU cloud context is carried out at two levels: local administration (managing each individual machine) and federal administration (if viewing the whole cloud infrastructure as a federated system). The infrastructure of CU clouds is similar to that in a grid environment (though more complex). Construction of CU clouds may well leverage the results obtained from Grid Computing, which reflect more than a decade of community efforts in standardization, security, resource management, and virtualization support [4].
- **Community-centered:** A CU cloud is community-driven, community-provisioned, community-owned, and community-consumed – altogether, community-centered. Yet, a CU cloud reserves the option to open to the outside world for gaining profits for the community and its members.
- **CU clouds are “greener”:** as explained earlier, CU clouds have potential to carry the sustainability of “Green Computing” forward by a big step.
- **Smaller scale:** as the computing resources supporting a CU cloud come from the contributing members in a community, comparing with enterprise clouds, CU clouds are of relatively smaller scale. But multiple CU clouds may form a consortium or federation and result in larger-scale CU clouds. So, CU clouds are not necessarily smaller than enterprise vendor provided clouds when the envisioned technologies mature.
- **Natural digital ecosystems:** A CU cloud possesses most features of a digital ecosystem such as self-provision, self-organization, self-control, scalability, sustainability, and thus can naturally serve as an ideal platform for digital ecosystem development.
- **Ideal platform for education:** A CU cloud deployed for an educational institution can serve as a readily available and ideal platform for further the development of community education clouds.
- **Ideal platform for government:** due to the less security and privacy concern that inherently comes with the CU model.
- **Ideal platform for every community and organization.**

6 Summary

In this paper, we presented a novel Cloud Computing Model (CUM) which is based on and motivated by the widely practiced credit unions as a type of cooperative financial institutions world-wide. We discussed the architecture for CU cloud implementation and other related issues. CU clouds have important advantages over the current Cloud Computing model (which is basically a vendor-provision model). CU clouds do not come without new challenges, but that are not insurmountable. The new challenges are outlined below and form our future work to be investigated with the project that we are currently initiating:

- CU cloud specific virtualization technologies (of which a key point is how to gracefully balance cloud requirements with native applications)
- New host operating systems with built-in virtualization capability utilizing special hardware level support.
- Decentralized cloud facility management (including distributing coordination and load balancing, etc.)

PS: Just before we were to submit this paper, we notice Marinos and Briscoe's paper [16] that addresses on a highly relevant issue – Community Cloud Computing. Recognizing the potential overlaps, we highlight on a few points that differentiate our work from their: (1) CUM is based on the credit union business model; (2) CU clouds are open; (3) CUM draws upon voluntary computing [5, 6].

References

1. Frischbier, S., Petrov, I.: Aspects of Data-Intensive Cloud Computing. In: From Active Data Management to Event-Based Systems and More, pp. 57–77 (2010)
2. Tek-Tips: Defining Cloud Computing (2009), <http://tek-tips.nethawk.net/blog/defining-cloud-computings-key-characteristics-deployment-and-delivery-types>
3. Kossmann, D., Kraska, T.: Data Management in the Cloud: Promises, State-of-the-art, and Open Questions. *Datenbank-Spektrum* 10(3), 121–129 (2010)
4. Foster, I., Zhao, Y., et al.: Cloud Computing and Grid Computing 360-Degree Compared. In: Grid Computing Environments Workshop, pp. 1–10 (2009)
5. Anderson, D., Cobb, J., et al.: SETI@home: an experiment in public-resource computing. *Commun. ACM* 45(11), 56–61 (2002)
6. Cappos, J., Beschastnikh, I., et al.: Seattle: A Platform for Educational Cloud Computing. In: SIGCSE, pp. 111–115 (2009)
7. VMWare (White Paper): Virtualization Overview (2006), <http://www.vmware.com/pdf/virtualization.pdf>
8. Mell, P., Grance, T.: The NIST Definition of Cloud Computing. National Institute of Standards and Technology. Information Technology Laboratory (July 2009)
9. Google: What is Google App Engine (2010), <http://code.google.com/intl/en/appengine/docs/whatisgoogleappengine.html>
10. Microsoft: Windows Azure (2010), <http://www.microsoft.com/windowsazure/windowsazure/>
11. Amazon Elastic Compute Cloud (Amazon EC2) (2011), <http://aws.amazon.com/ec2/>
12. Varia, J.: Cloud Architectures (Amazon White Paper, June 2008), <http://jineshvaria.s3.amazonaws.com/public/cloudarchitectures-varia.pdf>
13. Salesforce (2011), <http://www.salesforce.com/>
14. Armbrust, M., Fox, A., et al.: A Berkeley View of Cloud Computing. Technical Report No. UCB/EECS-2009-28, <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>
15. Jones, M. T.: Anatomy of an Open Source Cloud (2010), <http://www.ibm.com/developerworks/opensource/library/os-cloud-anatomy/>
16. Marinos, A., Briscoe, G.: Community Cloud Computing. In: *CloudCom*, pp. 472–484 (2009)