C-Cloud: A Cost-Efficient Reliable Cloud of Surplus Computing Resources

Partha Dutta*, Tridib Mukherjee*, Vinay G. Hegde** and Sujit Gujar†
*Xerox Research Center India (XRCI), Bangalore, India
†Artificial Intelligence Laboratory, EPFL, Lausanne, Switzerland

***Abstract*—This paper presents C-CLOUD, a democratic cloud infrastructure for renting computing resources including non-cloud resources (i.e. computing equipment not part of any cloud infrastructure, such as, PCs, laptops, enterprise servers and clusters). C-CLOUD enables enormous amount of surplus computing resources, in the range of hundreds of millions, to be rented out to cloud users. Such a sharing of resources allows resource owners to earn from idle resources, and cloud users to have a cost-efficient alternative to large cloud providers. Compared to existing approaches to sharing surplus resources, C-CLOUD has two key challenges: ensuring Service Level Agreement (SLA) and reliability of reservations made over heterogeneous resources, and providing appropriate mechanism to encourage sharing of resources. In this context, C-CLOUD introduces novel incentive mechanism that determines resource rents parametrically based on their reliability and capability.**

***Keywords*-Cloud Computing, Reliability, Cost-efficiency**

I. Introduction

With the advent of cloud computing, owners of large-scale computing infrastructure (specifically, the cloud providers) can rent out their computation power to individual users. However, this does not tap into the enormous compute power available in non-cloud resources, i.e., resources that are not owned by the cloud providers and hence not part of any cloud infrastructure. Indeed, as technology has grown, the computing power with individuals as well as enterprises has grown exponentially, and many of these resources frequently remain idle [1]. Building a distributed system over surplus computing resources have been studied in Volunteer Computing (VC) systems. In VC, tasks are run on computing resources that are donated by their owners [3], [2], [4]. BOINC[1] is a widely used VC framework, and Entropia[2] and United Devices[3] are examples of commercial VC systems. End users sharing network bandwidth and network data plans have been considered in FlexiMart[4] and CrowdMAC [5]. The above systems, due to their voluntary nature, rarely provide SLA for the tasks and monetary incentives for the resource owners. In this context, this paper focuses on a novel *democratic cloud computing* eco-system that allows anybody to commoditize their surplus resources as part of a cloud infrastructure.

*The work was done during intern-ship at XRCI
[1]http://boinc.berkeley.edu/
[2]http://en.wikipedia.org/wiki/Entropia,_Inc._(company)
[3]http://en.wikipedia.org/wiki/United_Devices
[4]http://www.aflexi.net/home

A major obstacle to realize the objective is the guarantee in delivering any service over dynamically shared resources while minimizing the cost. Further, as opposed to the conventional cloud infrastructure, the resources in a democratic cloud would be typically out of control of the cloud service providers in terms of their availability and reliability. This is because the resources are often dynamically or temporarily shared by the owners. Further, it is also possible that the resources shared for a particular duration become unavailable, either retracted by the respective owners for personal requirements or unforeseen failures (e.g., connection outage or system crash). Ensuring a certain level of quality in service delivery (e.g. timely and successful completion of map tasks) to meet SLAs is a hard problem under the aforementioned settings. In this context, this paper proposes: (i) **C-CLOUD**, a cost-efficient democratic cloud infrastructure made of dynamically and temporarily shared surplus computing resources; and (ii) a novel **incentive mechanism** that allows C-CLOUD to provide incentives to resource owners while being aware of whether a resource meets task SLAs and expected success probability levels.

II. C-CLOUD: System Overview

Fig. 1 shows the vision of C-CLOUD. In particular, C-CLOUD generates a cloud from resources which are not part of any cloud infrastructure. Owners (individuals and enterprises) can rent out any of their surplus resources through C-CLOUD and get revenue in return from the C-CLOUD. Fig. 1 further shows a high-level functional architecture of C-CLOUD. Primarily, C-CLOUD takes reservation requests (analogous to requesting a certain VM instance in modern day public clouds, e.g. Amazon, Azure, etc) as inputs and allocates the requests to the dynamically shared resources. However, offering guarantees towards meeting such reservation requests can be particularly challenging since the resources can be highly unreliable, unlike in conventional cloud infrastructure. Thus, it is imperative to provide proper incentives to the resource owners such that the cost of serving the requests is low.

III. Incentive mechanism in C-CLOUD

Suppose the tasks requested to be executed over C-CLOUD are divided into subtask to allow for flexibility in resource allocation. For the j^{th} task, consider the allocation of k^{th} subtask $tk_{j,k}$ on some resource r_i.. We define *efficiency* $\rho(j,k,i)$ of this resource allocation as the product

978-1-4799-5063-8/14 $31.00 © 2014 IEEE
DOI 10.1109/CLOUD.2014.152

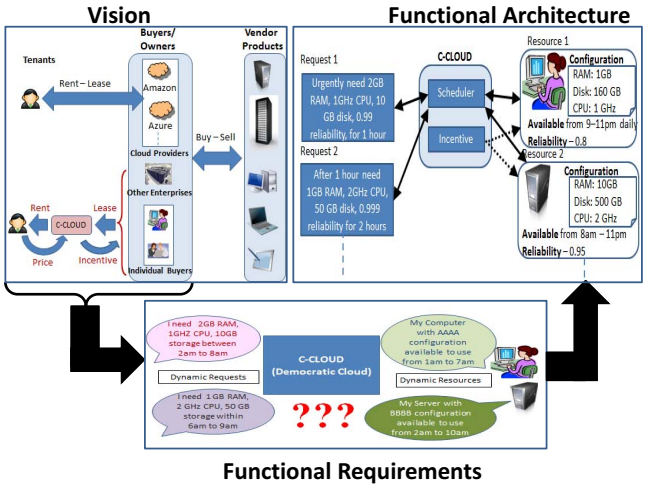

Figure 1. Vision, requirements, and high-level architecture of *C-CLOUD*

over all components of subtask requirement[5] of the ratio of allocated value of component to the desired value of the component. More specifically,

$$\rho(j,k,i) = \prod_{\text{each requirement of } tk_{j,k}} \frac{\text{value in the allocation over } r_i}{\text{desired value in } tk_{j,k}} \quad (1)$$

We next introduce a *penalty factor* pf which determines how severely a resource is penalized, in terms of monetary cost, when the allocation over the resource cannot satisfy the requirement of the subtask. In particular, the monetary cost of an allocation of $tk_{j,k}$ over resource r_i is defined in terms of the flat monetary cost $MC(i)$ of resource r_i as follows:

$$\begin{aligned} MC(j,k,i) &= \big(\rho(j,k,i)\big)^{pf}.MC(i) \qquad \text{if} \quad \rho \leq 1, \\ &= log_{10}\big(9+\rho(i,j)\big).MC(i) \quad \text{otherwise} \end{aligned} \quad (2)$$

We now discuss the effect of efficiency and the penalty factor on the monetary cost of a resource allocation.

- $\rho > 1$: This case implies that the resource can provide higher allocation than required, and thus can get some extra premium as compared to resources that do not. However, the premium is very small and increases logarithmically. This additional premium is small because additional allocation provided might not be always useful for the task.
- $\rho < 1$: Resources that provide allocation lower than the task requirement are penalized in terms of monetary cost. However, the degree to which the resource is penalized depends both on how far the allocation is from the required value (as captured using ρ), as well as an exponential penalty factor pf. Fig. 2 shows how the monetary cost depends on the efficiency ρ and the penalty factor pf.

A system designer can select a high value of pf to ensure sharp degradation of the monetary cost for resources that do not meet the task requirement. On the other extreme, the system designer can select a low value of pf to ensure slow degradation of the monetary cost as the efficiency, ρ, goes away from 1 towards 0.

[5]Namely, compute, memory, storage, success probability, and duration.

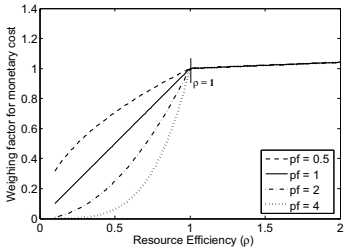

Figure 2. Dependency of weighing factor with resource efficiency.

IV. Discussion and Conclusion

This paper introduced C-CLOUD, a cost-efficient reliable democratic cloud of surplus resources dynamically and temporarily shared by the owners in return of incentives. To this effect, this paper proposed an incentive mechanism. The concepts of incentives for sharing resources and being aware of SLAs and task success probability make the proposed C-CLOUD unique to any other resource sharing paradigm, e.g. volunteer computing. C-CLOUD is further designed in a way that a user can get the flavour of modern day cloud infrastructure-as-a-service; albeit hosted over a set of dynamically shared and potentially unreliable resources with guarantees on task completion success. Major issues in the deployment revolve around interactions between the C-CLOUD and the resources shared for periodically checking the resource status (e.g. capability, reliability, availability, etc.). For handling these interactions, we use an existing open source distribution for volunteer computing.

References

[1] P. Goss, "30 million unused computers in the UK, says research," *Techradar*, Available: http://www.techradar.com/news/internet/pc/computing/30-million-unused-computers-in-the-uk-says-research-912899.

[2] D. Anderson and K. Reed, "Celebrating diversity in volunteer computing," in *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on*, 2009, pp. 1–8.

[3] J. Zhang and C. Phillips, "Job-scheduling via resource availability prediction for volunteer computational grids," *Int. J. Grid Util. Comput.*, vol. 2, no. 1, pp. 25–32, May 2011.

[4] E. Byun, S. Choi, H. Kim, C. Hwang, and S. Lee, "Advanced job scheduler based on markov availability model and resource selection in desktop grid computing environment," in *Meta-heuristics for Scheduling in Distributed Computing Environments*, ser. Studies in Computational Intelligence, F. Xhafa and A. Abraham, Eds. Springer, 2008, vol. 146.

[5] N. Do, C.-H. Hsu, and N. Venkatasubramanian, "Crowdmac: A crowdsourcing system for mobile access," in *Proceedings of the 13th International Middleware Conference*, ser. Middleware '12, 2012, pp. 1–20.