Energy-Aware Massively Distributed Cloud Facilities: the DISCOVERY Initiative

Frédéric Desprez*, Shadi Ibrahim*, Adrien Lebre*, Anne-Cécile Orgerie†, Jonathan Pastor*, Anthony Simonet*
*Inria, France, E-mails:firstname.name@inria.fr
†CNRS, France, firstname.name@irisa.fr

***Abstract*—Instead of the current trend consisting of building larger and larger data centers (DCs) in few strategic locations, the DISCOVERY initiative proposes to leverage any network point of presences (PoP, i.e., a small or medium-sized network center) available through the Internet. The key idea is to demonstrate a widely distributed Cloud platform that can better match the geographical dispersal of users and of renewable energy sources. This involves radical changes in the way resources are managed, but leveraging computing resources around the end-users will enable to deliver a new generation of highly efficient and sustainable Utility Computing (UC) platforms, thus providing a strong alternative to the actual Cloud model based on mega DCs (i.e., DCs composed of tens of thousands resources). This poster will present the DISCOVERY initiative efforts towards achieving energy-aware massively distributed cloud facilities.**

***Keywords*—Cloud computing, green computing, decentralized systems.**

I. Motivation

To satisfy the escalating demand for Cloud Computing (CC) resources while realizing economy of scale, the production of computing resources is concentrated in mega data centers (DCs) of ever-increasing size, where the number of physical resources that one DC can host is limited by the capacity of its energy supply and its cooling system. To meet these critical needs in terms of energy supply and cooling, the current trend is toward building DCs in regions with abundant and affordable electricity supplies or in regions close to the polar circle to leverage free cooling techniques [1].

However, concentrating Mega-DCs in only few attractive places implies different issues. First, a disaster in these areas would be dramatic for IT services the DCs host as the connectivity to CC resources would not be guaranteed. Second, in addition to jurisdiction concerns, hosting computing resources in a few locations leads to useless network overheads to reach each DC. Such overheads can prevent the adoption of the UC paradigm by several kinds of applications such as mobile computing or big data ones.

The concept of micro/nano DCs at the edge of the backbone [2] is a promising solution to address the aforementioned concerns. However, operating multiple small DCs breaks somehow the idea of mutualization in terms of physical resources and administration simplicity, making this approach questionable in terms of energy efficiency. One way to enhance mutualization is to leverage existing network centers, starting from the core nodes of the backbone to the different network access points (a.k.a. PoPs Points of Presence) in charge of interconnecting public and private institutions. By hosting micro/nano DCs in PoPs, it becomes possible to mutualize resources, and so to make use of locally produced renewable energy, coming from solar panels or windmills for instance, which are often not dimension to power big facilities. This kind of architecture is also more suitable to deliver widely distributed CC platforms better suited to cope with disasters and to match the geographical dispersal of users and their needs.

A preliminary study has established the fundamentals of such an *in-network distributed cloud* referred by the consortium as the *Locality-Based Utility Computing* (LUC) concept [3]. However, the question of how operating such an infrastructure is still under investigations. Indeed, at this level of distribution, latency and fault tolerance become primary concerns, and collaboration between servers of different locations must be organized wisely, in order, for example, to balance the load according to renewable energy availability.

II. The LUC OS Design

The massively distributed cloud we target is an infrastructure that is composed of up to hundreds of micro DCs, which are themselves composed of up to tens of servers. Thus the system in charge of operating such an infrastructure should be able to manage up to thousands of servers spread geographically. Delivering such a system is a tedious task where wrong design choices could prevent to achieve our goal. In this section we first discuss few conceptual considerations that led us to the LUC OS proposal and second remain the major services that the LUC OS should deliver.

A. From Centralized to Distributed Management

The first way that comes generally to the mind to pilot and use distinct clouds is to rely on classical models like federated approaches: each micro DC hosts and operates its own Cloud infrastructure and a brokering service is in charge of resources provisioning by picking on each cloud. While such approaches can be acceptable for elementary usages, advanced brokering services are mandatory to meet production environment requirements (monitoring, energy-aware scheduling, automated provisioning, SLAs enforcement …). In addition to dealing with scalability and single point of failure (SPOF) issues, brokering services should integrate mechanisms similar to those that are already implemented at the level of IaaS managers [4]. Consequently, the development of a brokering solution is as difficult as developing an IaaS manager but with the complexity of relying only on the least common denominator APIs. While few standards such as OCCI [5] start to be adopted, they do not allow developers to manipulate low-level capabilities of each system, which is generally mandatory to finely administrate resources.

The other way to operate such an infrastructure is to design and build a dedicated system, i.e. the *LUC Operating System*, in charge of operating all the geographically spread micro DCs in a distributed manner. A LUC OS will define and leverage its own software interface, thus extending capacities of traditional Clouds with its API and a set of dedicated tools. This offers a unique opportunity to go beyond classical federations of Clouds by addressing all crosscutting concerns of a software stack as complex as a IaaS manager and by revising in a fully distributed manner, mechanisms that have been traditionally implemented in a centralized one (service nodes).

The following question is now to analyze whether the collaborations between instances of the system, that is the service nodes, should be structured either in a hierarchical way or in a P2P (i.e. flat) one. Few hierarchical solutions have been proposed during the last years in industry [6] and academia [7]. Although they may look easier than P2P structures, hierarchical approaches require additional maintenance costs and complex operations in case of failure.

On the other side, Peer-to-Peer (P2P) file sharing systems are a good example of software that works well at large scale and in a context where computing resources are geographically spread. While largely unexplored for building operating systems, peer-to-peer/decentralized mechanisms have the potential to natively handle the intrinsic distribution of LUC infrastructures as well as the scalability required to manage them. Hence, we propose to leverage advanced P2P mechanisms like overlay networks and distributed hash tables to design the LUC OS building blocks.

B. Cloud Capabilities

From the administrators and end-users point of views, the LUC OS should deliver a set of high level mechanisms whose assembly results in an operational IaaS system. Recent studies have shown that state of the art IaaS manager [8] were constructed over the same concepts and that a reference architecture for IaaS manager can be defined [9]. This architecture covers primary services that are needed for building the LUC OS :

- The **virtual machines manager** is in charge of managing VMs' cycle of life (configuration, scheduling, deployment, suspend/resume and shut down).
- The **Image manager** is in charge of VM' template files (a.k.a. VM images).
- The **Network manager** provides connectivity to the infrastructure: virtual networks for VMs and external access for users.
- The **Storage manager** provides persistent storage facilities to VMs.
- The **Administrative tools** provide user interfaces to operate and use the infrastructure.
- Finally, the **Information manager** monitors data of the infrastructure for the auditing/accounting.

The challenge is thus to propose a distributed version of the aforementioned services by relying on advanced P2P mechanisms. Instead of reinventing the wheel, we propose to minimize both design and implementation efforts by reusing as much as possible existing pieces of codes. With this in mind, we propose to investigate whether the OpenStack solution [10] can be revised to fulfill the LUC infrastructure requirements. Concretely, we propose to determine which mechanisms can be directly used and which ones must be revisited with P2P algorithms.

III. On-going work

We are currently performing an energy/cost-benefit analysis of the massively distributed Cloud architecture proposed in the Discovery project. Conducting such an in-depth analysis is mandatory (i) to design models that will allow us to extrapolate from the obtained values to the whole infrastructure and (ii) to determine the latitude we have to extend each PoP to a micro/nano DC. Such models will enable us to compare distributed vs. Mega-DC approaches in terms of energy, performance and financial aspects. Our current on-going work includes analyzing the energy footprint of OpenStack operations in order to improve their energy consumption.

Acknowledgments

This work was partially supported by the Inria Project Lab Discovery, an Open-Science Initiative aiming at implementing a fully decentralized IaaS manager: http://beyondtheclouds.github.io.

References

[1] J. V. H. Gary Cook, "How dirty is your data ?" Greenpeace International Report, 2013.

[2] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 68–73, 2008.

[3] M. Bertier, F. Desprez, G. Fedak, A. Lebre, A.-C. Orgerie, J. Pastor, F. Quesnel, J. Rouzaud-Cornabas, and C. Tedeschi, "Beyond the clouds: How should next generation utility computing infrastructures be designed?" in *Cloud Computing*, ser. Computer Communications and Networks, Z. Mahmood, Ed. Springer International Publishing, 2014, pp. 325–345. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10530-7_14

[4] I. Houidi, M. Mechtri, W. Louati, and D. Zeghlache, "Cloud service delivery across multiple cloud platforms," in *Services Computing (SCC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 741–742.

[5] N. Loutas, V. Peristeras, T. Bouras, E. Kamateri, D. Zeginis, and K. Tarabanis, "Towards a reference architecture for semantically interoperable clouds," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. IEEE, 2010, pp. 143–150.

[6] "Cascading OpenStack," https://wiki.openstack.org/wiki/OpenStack_cascading_solution.

[7] F. Farahnakian, P. Liljeberg, T. Pahikkala, J. Plosila, and H. Tenhunen, "Hierarchical vm management architecture for cloud data centers," in *6th International Conf. on Cloud Computing Technology and Science (CloudCom)*, Dec 2014, pp. 306–311.

[8] J. Peng, X. Zhang, Z. Lei, B. Zhang, W. Zhang, and Q. Li, "Comparison of several cloud computing platforms," in *2nd Int. Symp. on Information Science and Engineering (ISISE)*, 2009, pp. 23–27.

[9] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Iaas cloud architecture: From virtualized datacenters to federated cloud infrastructures," *Computer*, vol. 45, no. 12, pp. 65–72, 2012.

[10] "The Open Source, Open Standards Cloud," http://www.openstack.org.