A Novel Cloud Bursting Technique

Sandor Acs*
*Institute for Computer Science and Control
MTA SZTAKI
1111 Budapest
Kende u. 13-17, Hungary
acs.sandor@sztaki.mta.hu

Miklos Kozlovszky*†
†Obuda University
OE NIK
1034 Budapest
Becsi u. 96/B, Hungary
kozlovszky.miklos@nik.uni-obuda.hu

Peter Kacsuk*‡
‡University of Westminster
UoW
115 New Cavendish Street
London W1W 6UW, United Kingdom
kacsukp@westminster.ac.uk

***Abstract*—Companies (even SMEs) are roused by the success and potentials of the public clouds and they build their own private cloud infrastructures. Thus, they open the door to an easier and more flexible way for outsourcing their IT services than before. However, the currently available software solutions still do not provide seamless extensibility by cloud bursting. Therefore, the IaaS users have to prepare their images in every infrastructure. This paper presents the criteria for idealistic cloud bursting and introduces a method that overcomes the current cloud bursting issues (e.g. different administration domains and networking policies). The proposed technique uses nested virtualization, that reduces the complexity of the cloud bursting procedure. Furthermore, we have evaluated the applicability of our design by performance tests. The evaluation showed that the seamless extensibility has a cost of 5-10% overhead on the deployment time.**

***Keywords*-cloud computing; cloud bursting; IT outsourcing;**

I. INTRODUCTION

Cloud computing is the dominant IT paradigm in distributed computing because it provides services on a pay-per-use model. Cloud solutions are built on virtualization technologies that support the extreme flexibility and scalability of services [1]. There are three major service models (Software/Platform/Infrastructure-as-a-Service) and three major deployment models (Public/Private/Hybrid) available for clouds [2]. This paper focuses on IaaS clouds that offer the fundamental IT resources, like networking, computation, and storage capacity.

Amazon launched the Elastic Cloud Computing (EC2) service in 2006, thus they were the first public IaaS provider. Since then, many providers entered the market and private cloud solutions emerged as well. Private IaaS clouds offer companies to benefit the cloud technology on their own IT resources. These software products usually provide support for combining the private and public clouds as a hybrid cloud environment. The technique called cloud bursting allows the companies to dynamically extend their infrastructure by renting third-party resources. However, seamless cloud bursting is not available in the current software products yet because of the following reasons. (i) The private and the public cloud are located in different administration domains from the application's point of view. Complex applications may not be split (distribute VM instances over multiple clouds) because of networking issues. (ii) IaaS clouds have different technical backgrounds (e.g. supported hypervisors and image formats) and they may require different API-s for cloud bursting. Therefore, the VMs—that contain the applications—need to be adopted for the target cloud infrastructures.

This paper is organized as follows: In Section 2 the related work is introduced, in Section 3 the design principles of the novel cloud bursting technique is described, Section 4 presents the evaluation of the proposed solution, and Section 5 concludes the paper.

II. RELATED WORK

There have been several recent efforts focusing on cloud bursting techniques and procedures.

Bicer [3] et al. presented a framework for supporting the development and execution of data-intensive computations. Their comprehensive evaluation shows that the hybrid clouds can be attractive environments for HPC and data-intensive computing as well.

Sriram [4] et al. concentrated on batch job scheduling techniques. They proposed different flavors of schedulers that are effective in cloud bursting large jobs and also honor various constraints and service level agreements.

Srijith [5] et al. presented the concepts of cloud bursting, cloud aggregation and cloud brokerage and they established the capability requirement of the entities.

Seagull [6] et al. provide a framework for cloud bursting that efficiently precopies and migrates applications to a public cloud when local infrastructure becomes overloaded. This solution enables performing agile provisioning of resources across private and public clouds.

The presented works and software products did not take into consideration the following issues. (i) IaaS clouds provide different services and they may require different API-s for bursting. Therefore, the applications need to be adopted for each target cloud infrastructures. (ii) The target (public) cloud is a different administration domain. Complex applications (collections of VMs) may not be split because

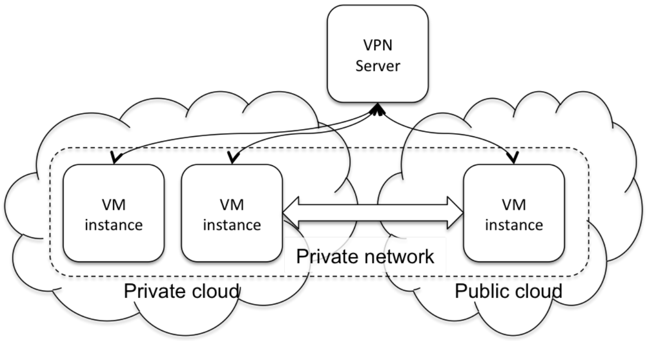

Figure 1: Virtual Private Networking

of networking or security restrictions. (iii) The live migration of the VM instances is essential to keep the flexibility and reliability of the cloud infrastructures.

There are some solutions that can overcome these issues. The HVX [7] is a cloud application hypervisor that allows easy migration of unmodified multi-VM applications between different private and public clouds effortlessly. However, the HVX is a proprietary product and we cannot adopt it for cloud bursting. The Xen-blanket [8] can also provide same feature as HVX, but it supports only the XEN hypervisor. (iii) The live migration of the VM instances is essential to keep the flexibility and reliability of the cloud infrastructures.

III. The novel cloud bursting technique

A. Principles

We believe that a cloud bursting technique should not incapacitate the running services. Therefore, we collected the criteria of seamless cloud bursting. (i) The applications have to operate exactly the same way on public clouds as on the original private one without any modification of VM images. (ii) The users of the services must not sense any difference between the private and the public clouds. The bursting procedure should be hidden by the underlying cloud layer, thus users are not forced to manually distribute their images and launch VM instances. (iii) The live migration must be available between the different clouds in order to keep the continuous operation of services.

B. Design

We use virtual private networking (VPN) and the nested virtualization technology to cover the differences between the local private and the public clouds. The VPN enables the VM instances—running on external public clouds— to join the local private network. Therefore, VM instances may have the same networking environment in both infrastructures.

Figure 1 presents a VPN server (top), a private (left), and a public cloud (right). The VPN server creates a private network between the clouds so the VM instances can communicate as if they were deployed into the same infrastructure.

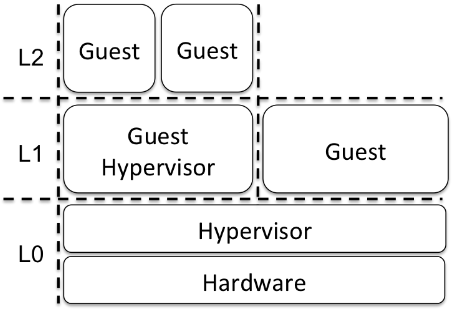

Figure 2: Nested virtualization

Figure 2 introduces the nested virtualization in a nutshell. The hypervisor (L0) on the top of a physical hardware enables launching guest VM instances (depicted as L1 Guest). The nested virtualization [9] allows to launch new VM instances (depicted as L2 Guest) on a VM instance. It is important to notice that the nested virtualization is not supported yet in public clouds, but it will be enabled in the future [10].

We use the nested virtualization to provide a generic virtualization layer on top of the clouds in order to cover the differences (e.g. hypervisor and image format) between the different IaaS solutions.

Figure 3 presents the current method of cloud bursting and it introduces that the users need to adopt their applications and upload their prepared images into both clouds.

The novel cloud bursting technique is presented in Figure 4. This solution launches virtual compute nodes as VM instances on the target clouds. The virtual compute nodes join the original infrastructure (depicted as Cloud#1) via VPN connections. Therefore, the bursted VM instances of the users become nested VM instances on the target clouds. Thus, the users can exploit the additional IaaS capacities without any adaptation of the applications and the bursted VM instances will be maintained in the same administrative domain as the original ones.

In this paper, we focus on extending computing capacity; however, networking and storage capacity can be bursted as well.

IV. Evaluation

The performance of the nested virtualization has been already studied [8] [11] so we focus on evaluating the overhead of the VM deployment.

Table I summarizes the technical details (in terms of CPU, memory, operating system, and hypervisor) of the test environment.

We prepared three VM images for measuring the deployment time on layers L1 and L2. These images were built to reflect to real life use-cases. Therefore, the first

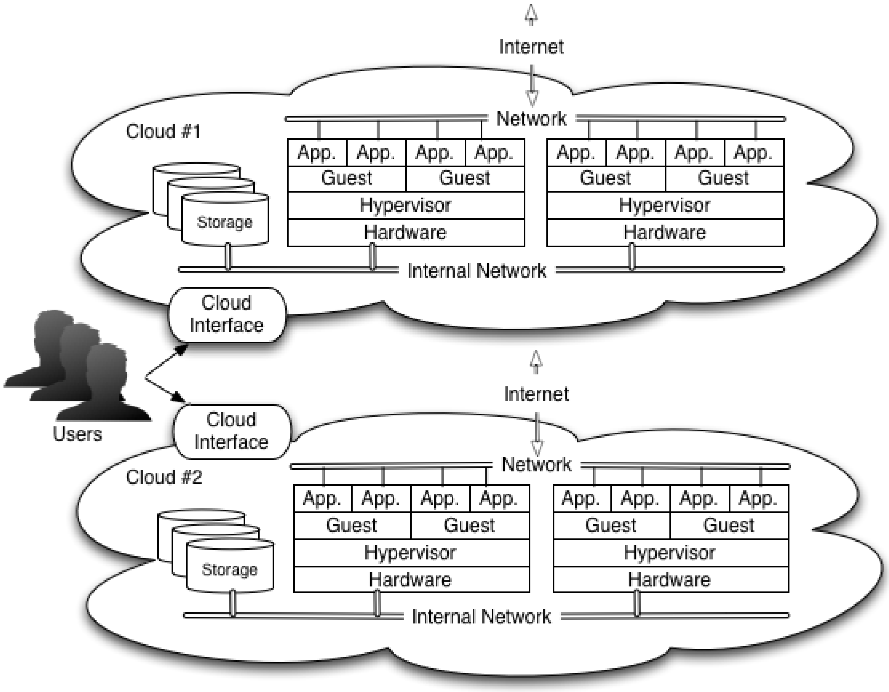

Figure 3: Classic cloud bursting technique

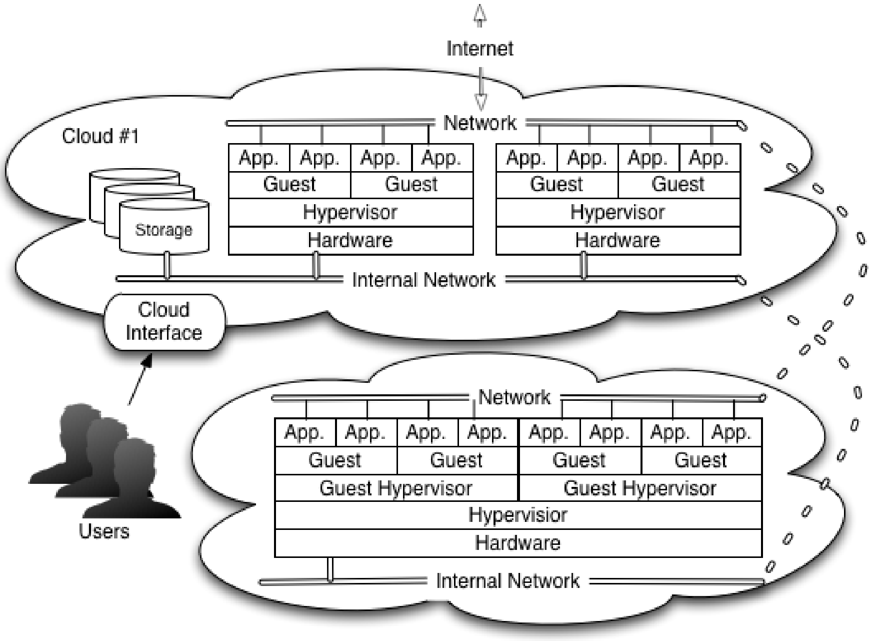

Figure 4: Novel cloud bursting technique

	Physical machines (L0)	Virtual Machine (L1)	Nested VM (L2)
Processor	Intel Core i7	N/A	N/A
CPU [cores]	2	2	2
Memory [GB]	8	4	4
OS	OS X v. 10.9.2	Ubuntu Linux v. 13.10	Ubuntu Linux v. 13.10
Hypervisor	Vmware Fusion 6.0.2	KVM v. 1.5.0	N/A

Table I: Test environment

Application	Virtual Machine (L1)	Nested VM (L2)
Base [s]	14.29	15.72
LAMP [s]	16.55	17.69
LAMP+Tomcat7 [s]	20.42	21.93

Table II: Test results

image is a simple Ununtu Linux server edition without any additional services. This image was extended by the LAMP (Linux-Apache-MySQL-PHP) software stack for the second round. Then, the Tomcat7 was installed and tested. All of the measurements were repeated 10 times and the standard deviation was less than 3%.

The tests (Table II) showed that, the deployment times were increased by 5-10% on L2 according to layer L1.

V. Conclusion and future work

In this paper, we introduced the cloud bursting techniques and the related works. Then, we proposed a new bursting method that exploits nested virtualization and advanced networking technologies. The evaluation showed that seamless cloud bursting increases deployment time by 5-10%. In the future, we plan to extend our technique and focus on networking and storage capacity bursting as well.

Acknowledgment

The authors would like to thank Dr. Zsolt Németh, Márk Gergely, and Ádám Visegrádi for their valuable comments and suggestions.

References

[1] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Gener. Comput. Syst.*, 25(6):599–616, June 2009.

[2] P. Mell and T. Grance. The nist definition of cloud computing. *National Institute of Standards and Technology*, 53(6):50, 2009.

[3] T. Bicer, D. Chiu, and G. Agrawal. A framework for data-intensive computing with cloud bursting. In *Cluster Computing (CLUSTER), 2011 IEEE International Conference on*, pages 169–177, Sept 2011.

[4] S. Kailasam, N. Gnanasambandam, J. Dharanipragada, and N. Sharma. Optimizing service level agreements for autonomic cloud bursting schedulers. In *Parallel Processing Workshops (ICPPW), 2010 39th International Conference on*, pages 285–294, Sept 2010.

[5] S.K. Nair, S. Porwal, T. Dimitrakos, A.J. Ferrer, J. Tordsson, T. Sharif, C. Sheridan, M. Rajarajan, and A.U. Khan. Towards secure cloud bursting, brokerage and aggregation. In *Web Services (ECOWS), 2010 IEEE 8th European Conference on*, pages 189–196, Dec 2010.

[6] Tian Guo, Upendra Sharma, Timothy Wood, Sambit Sahu, and Prashant Shenoy. Seagull: Intelligent cloud bursting for enterprise applications. In *Proceedings of the 2012 USENIX Conference on Annual Technical Conference*, USENIX ATC'12, pages 33–33, Berkeley, CA, USA, 2012. USENIX Association.

[7] Alex Fishman, Mike Rapoport, Evgeny Budilovsky, and Izik Eidus. Hvx: Virtualizing the cloud. In *Presented as part of the 5th USENIX Workshop on Hot Topics in Cloud Computing*. USENIX, 2013.

[8] Dan Williams, Hani Jamjoom, and Hakim Weatherspoon. The xen-blanket: virtualize once, run everywhere. In *Proceedings of the 7th ACM european conference on Computer Systems*, pages 113–126. ACM, 2012.

[9] Muli Ben-Yehuda, Michael D Day, Zvi Dubitzky, Michael Factor, Nadav Har'El, Abel Gordon, Anthony Liguori, Orit Wasserman, and Ben-Ami Yassour. The turtles project: Design and implementation of nested virtualization. In *OSDI*, volume 10, pages 423–436, 2010.

[10] Michael Beham, Marius Vlad, and Hans P Reiser. Intrusion detection and honeypots in nested virtualization environments. In *Dependable Systems and Networks (DSN), 2013 43rd Annual IEEE/IFIP International Conference on*, pages 1–6. IEEE, 2013.

[11] HeeSeok Choi, TaeMuk Lyoo, JongBeom Lim, Daeyong Jung, Jihun Kang, Taeweon Suh, and Heonchang Yu. A study on performance comparison of cloud architectures using nested virtualization. In *Ubiquitous Information Technologies and Applications*, pages 77–84. Springer, 2014.