# Towards Migratable Elastic Virtual Clusters on Hybrid Clouds

Amanda Calatrava, Germán Moltó, Eloy Romero, Miguel Caballer, Carlos de Alfonso
*Instituto de Instrumentación para Imagen Molecular (I3M).*
*Centro mixto CSIC - Universitat Politècnica de València - CIEMAT*
*Camino de Vera s/n, 46022 Valencia, España*
*Email: {amcaar,elroal,micafer1,caralla}@i3m.upv.es*
*,gmolto@dsic.upv.es*

***Abstract*—This paper describes the research work in the context of the CLUVIEM project towards achieving migratable, self-managed virtual elastic clusters on hybrid Cloud infrastructures. These virtual clusters can span across on-premises and public Cloud infrastructures thus leveraging hybrid Cloud platforms. They are elastic since working nodes are automatically provisioned and relinquished to dynamically adapt the capacity of the virtual cluster (in terms of number of nodes) according to the current workload. They are self-managed since the elasticity rules are managed via the head node without requiring any external software entity for monitoring and deciding when to scale in and out. Finally, they are migratable since they consider both application migration, via application checkpointing, and infrastructure migration, by cloning infrastructures across multi-Clouds. These features introduce unprecedented flexibility for cost-effective cluster-based computing with minimal impact for cluster users. The paper summarises the current state of developments and future roads to achieve this vision.**

***Keywords*-Cloud computing; High Performance Computing; Virtualization; Elasticity;**

## I. Introduction

Clusters are one of the most widely used computing facilities across the world. They can be used for High Performance Computing (HPC), where tightly-coupled tasks require intensive communication, and for High Throughput Computing (HTC), where loosely-coupled tasks are typically executed as a Bag of Tasks (BoT) or a parameter sweep application. However, physical clusters suffer from several drawbacks which include, but are not limited to, an initial large capital investment, electricity costs for operation and refrigeration and the inability to cost-effectively enlarge and decrease the number of nodes according to the workload.

With the introduction of virtualization and the advent of Cloud Computing, the idea of deploying virtual clusters on computational resources provisioned from Cloud infrastructures took shape in the form of tools such as StarCluster [1] or Elasticluster [2]. StarCluster enables to provision a virtual cluster on top of Amazon Web Services (AWS). It also supports to automatically scale out the cluster (and scale in) considering the number of jobs queued up at the LRMS (Local Resource Management System). However, since this tool can only provision clusters from AWS, no virtual clusters can be deployed on on-premises Cloud platforms created with Cloud Management Platforms (CMPs) such as OpenNebula or OpenStack. In addition, the scaling capabilities of the virtual cluster require a client-side monitoring application that is always running and periodically polls the cluster. Therefore, the cluster is not self-managed and requires the StarCluster application running on the client side. In contrast, Elasticluster can be employed to create virtual clusters on several Cloud providers (Amazon EC2 and Google Compute Engine) as well as on-premises Cloud platforms (OpenStack supported). The clusters support elasticity but, unfortunately, the user decides when to scale the cluster by using the appropriate command. Therefore no automated elasticity is supported.

Other tools to deploy virtual clusters can be found in the literature, such as Wrangler [3] or the work by Niu et al. [4]. The former does not support elasticity while the latter, although it does include elasticity rules to scale the clusters, it does not consider support for *spot instances*, which is a cost-effective mechanism to provision computational resources for interruptible tasks, supported by Amazon EC2. In addition, none of the aforementioned tools support hybrid virtual clusters, where resources can span several Clouds (either on-premises or public).

In this paper we build on the state of the art and describe the goals, the road map and the milestones achieved so far in the CLUVIEM project. The project, funded by the Spanish government, aims at developing software (accessible via SaaS and CLI) to create migratable self-managed cost-effective virtual elastic clusters on hybrid Cloud infrastructures which, for the sake of brevity, will be named enhanced virtual clusters. After the introduction, the remainder of the paper is structured as follows. First, section II introduces the main architecture of the platform to be developed featuring capabilities such as automated elasticity, hybrid scenarios and migration. Next, section III addresses different scenarios in which these virtual clusters introduce significant benefits. Finally, section IV summarizes the paper and points to future work.

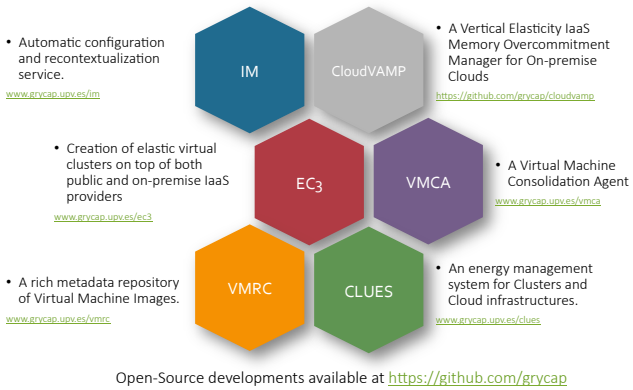


Figure 1. Open source software supporting the CLUVIEM project. See www.grycap.upv.es/{im,ec3,vmca,vmrc,clues}.

## II. Enhanced Virtual Clusters

Virtual clusters on the Cloud are composed of virtual machines which are provisioned from different Cloud providers. In the case of public Cloud providers such as AWS, a pay-as-you-go cost model is employed with no upfront investments. In the case of on-premises Clouds the cost of the provisioned resources is typically measured in terms of energy consumption. The greatest advantage of these virtual clusters is that they can naturally leverage the underlying elasticity of the Cloud platforms. You can start with a single head node (a.k.a. front-end node) that provides the users with the illusion of a fully active cluster and when they start submitting jobs to the LRMS, these are transparently intercepted to provision the required working nodes, using different customizable provisioning approaches, and are configured and automatically integrated in the LRMS before releasing the jobs to be executed. Therefore, users just notice a small delay until the jobs actually start their execution. The worker nodes are automatically relinquished whenever they are no longer used (or expected to be used according to a set of policies). This introduces a cost-effective approach for cluster-based computing where computational resources are provisioned and released as required.

Figure 1 shows the underlying software components[1] employed to create the platform to deploy these enhanced clusters: VMRC (Virtual Machine image Repository & Catalog), a catalog of Virtual Machine Images (VMIs) available in different Cloud platforms (e.g. AMIs in AWS and images in an OpenNebula repository), supporting matchmaking capabilities according to metadata; CLUES (CLUster Energy Saving), an elasticity management system for clusters and Cloud infrastructures; VMCA (Virtual Machine Consolidation Agent), a tool to consolidate VMs featuring migration across physical nodes; EC3 (Elastic Cloud Computing Cluster), a platform to create elastic virtual clusters on multi-Clouds; IM (Infrastructure Manager), a platform to provision and configure virtual infrastructure from different Cloud providers; and, finally, CloudVAMP, a memory overcommitment manager for on-premises Clouds, used to control vertical elasticity.

[1]These components have been released as open source software, available at https://github.com/grycap

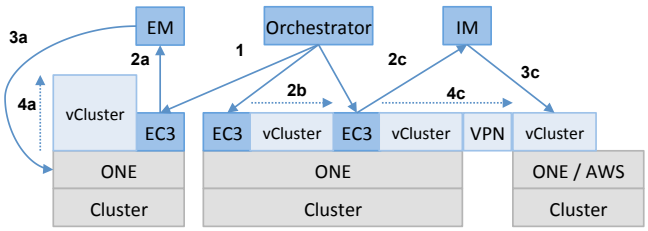


Figure 2. Elasticity schemes for virtual clusters: a) scaling up an already deployed working node (wn); b) deploying a new working node on the on-premises Cloud; c) deploying a new working node on another Cloud.

The following subsections outline the features of these enhanced virtual clusters and provide some additional details regarding the technologies and tools employed to achieve them.

### A. Elasticity

Virtual clusters must feature elasticity in order to cope with the dynamic computation requirements of the applications being executed. Figure 2 includes the different elasticity schemes addressed by CLUVIEM. First of all, vertical elasticity (Figure 2.a) enables to modify the capacities of the virtual machine at runtime without any downtime. Depending on the hypervisor support, these features include dynamic memory resizing, through memory ballooning, an dynamically adding virtual CPUs. Vertical elasticity allows, for example, to adapt the memory of the virtual machines that are executing a scientific application with dynamic memory requirements during its execution. A demonstration of that can be found in [5], in which it is monitored the memory consumption of an application within a VM and dynamically adapted the memory size of the VM to fit that memory consumption. In this way, the application does not incur in *thrashing* thus affecting its performance.

On the one hand, downsizing the memory of the VM when no longer required provides additional available free memory for other VMs that are currently being executed on the same physical host, a common situation on multi-tenant on-premises Cloud platforms. On the other hand, increasing the amount of memory of a VM might exceed the capacities of the underlying physical host. For that reason, live migration (without any downtime) of VMs to restore the Quality of Service is imperative. This enables to rebalance the workload of VMs across the datacenter (or across a physical cluster of nodes) so that the VMs have access to the required computational resources. This situation requires an appropriate migration plan that considers the whole state

of the underlying physical infrastructure and decides which VMs should be migrated to which nodes. For that, we plan to use VMCA, an add-on to CLUES that defragments the available resources by migrating VMs among physical hosts. This results in an increased density of VMs per real host.

Second, horizontal elasticity enables to shrink and grow the cluster size, in number of nodes, according to the values of some metrics such as the number of jobs queued up at the LRMS. Figure 2.b represents the scale out of a virtual cluster deployed on an on-premises Cloud managed by OpenNebula (ONE). The ability to resize a virtual cluster enables to cope with increased workloads at the expense of an increased cost (either in terms of energy, when running on an on-premises Cloud, or in terms of money, when running on a pay-as-you-go public Cloud). For that, we leverage the already-existing policies of CLUES, but adapted to a Cloud scenario (instead of powering on and off physical nodes through Wake-on-LAN or IPMI, VMs are deployed or terminated in a Cloud).

The Elasticity Manager (EM) is actually the aforementioned CLUES software, which runs in the head node (FE) of the cluster. Therefore, the cluster is self-managed and can scale in and out according to the elasticity rules without any user intervention. When the user deploys the cluster, the maximum number of nodes (VMs) is specified so that a reasonable cost per hour is never exceeded (when using a public Cloud). This means that these enhanced clusters automatically enter a low-cost mode (either energy or money) when no job workload is pending or expected to be executed. Notice that for certain workloads (e.g. burst of jobs) the cost of re-deploying a new cluster does not pay off when compared to provisioning additional worker nodes, which typically involves less configuration and time.

### B. Hybrid Scenarios

When trying to leverage both on-premises and public Cloud resources, hybrid scenarios arise (as depicted in Figure 2.c), in which VMs are deployed from different Cloud providers. For example, a virtual cluster is initially deployed on an on-premises Cloud and additional worker nodes are provisioned from another on-premises Cloud or a public Cloud to supplement the single virtual cluster with additional resources. Notice that the master node can either be deployed on the on-premises Cloud or on the public Cloud.

There are different scenarios in which provisioning from multi-cloud environments is beneficial. First, this approach can overcome a temporary shortage of computational resources within an on-premises Cloud. For example, when the requirements, in terms of number of nodes or their computational or memory capacities, of a virtual cluster exceed the capacities of an on-premises Cloud platform. A hybrid virtual cluster can span across an on-premises and a public Cloud to provide transparent Cloud bursting without affecting the users, which simply submit their jobs to be executed in the cluster through the LRMS. This is the case

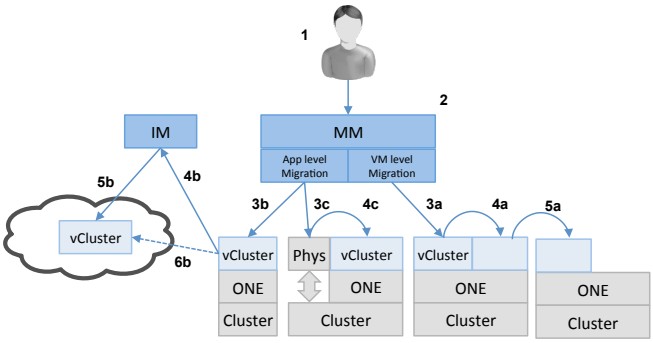


Figure 3. Migration schemes for virtual clusters.

of our previous work [6], in which hybrid virtual clusters are employed to execute a parallel computationally intensive gyrokinetic plasma turbulence code running on such hybrid clusters with resources provisioned from an on-premises OpenNebula Cloud and Amazon Web Services.

To create a common network among the VMs in disparate Cloud providers we provide support for Virtual Private Networks (VPNs) with OpenVPN, where the OpenVPN server is automatically deployed and configured on the head node of the cluster. Alternatively, we also support automatic deployment of SSH tunnels, customised with *iptables* rules to decide when to channel the traffic through the tunnels.

### C. Migration

Migrating virtual infrastructures is of interest both for datacenter administrators and for the owners of the virtual infrastructures themselves. On the one hand, datacenter administrators might require to decommission a physical node, perhaps because the SMART disk monitoring system alerts of an imminent failure. In this way, the ability to migrate virtual infrastructures enables to redistribute the virtual machines among the other physical nodes in the datacenter. For that, one can leverage the live migration capabilities available in hypervisors such as KVM or Xen so that VMs are migrated without any downtime or service disruption. Fortunately, CMPs such as OpenNebula leverage this ability to provide graphical tools to aid the sysadmin. However, migrating virtual infrastructures across Cloud providers is not a trivial task, where disparate hypervisors and platforms are employed. In the CLUVIEM project we address migration as shown in Figure 3.

First, the migration of virtual infrastructures can be achieved within the same Cloud on-premises (see arrows labeled $a$ in Figure 3) by using live migration. We have assessed the capabilities of KVM to perform live migration across physical nodes of the same OpenNebula deployment without any downtime. Migration can also occur across different on-premises Clouds (as shown in $5a$ and across public Clouds (shown in arrows labeled $b$). For that, we deploy a replica of the virtual cluster into a different Cloud provider,

coordinated by the Migration Manager (MM). Since clusters are created out of a high level language called RADL (Resource Application and Description Language; see [7] for details) it is possible to replicate the infrastructure into another Cloud provider by using the multi-Cloud capabilities of the IM. This involves deploying a new infrastructure with the same characteristics in another Cloud back-end. Transitioning from a physical cluster to a virtual one requires abstracting its hardware, software and data configuration to be expressed in RADL, what we intend to provide in a semi-automatic way but it is currently under research.

Second, the migration of running applications requires the introduction of application-independent checkpointing techniques in order to be able to resume a running application on the target virtual machine instance. For that purpose we have been using BLCR (Berkeley Lab Checkpoint/Restart for LINUX), a tool that introduces checkpoint capabilities both for sequential and parallel applications based on MPI. We use checkpointing both for migration of applications and as an application survival mechanism when using spot instances in Amazon EC2. A spot instance can be terminated if its price exceeds the bid of the user. For that, we developed a Checkpoint Manager that interacts with the SLURM LRMS supporting BLCR in order to checkpoint the jobs both at periodic interval and considering the evolution of the prices of the spot instances. This way, interrupted jobs can be resumed in newly deployed instances, which may be on a different Cloud (with the same virtual hardware).

Migrating workloads, such as independent jobs that arise from Hight Throughput Computing, can be efficiently achieved by deploying hybrid virtual clusters that dynamically remove and add nodes, from different Clouds, that are activated/deactivated from the LRMS so that jobs can be balanced across the working nodes without any user intervention, as performed in [6].

## III. Discussion and Application Scenarios

These enhanced virtual clusters can be employed for many applications in which cost-effective cluster-based computing is required. In particular, we are focusing on the following scenarios. First, the non-linear and dynamic structural analysis of buildings, where it is required to accurately simulate how a building is affected by external dynamic loads, such as an earthquake. This involves a parallel MPI-based applications. Second, the execution of Monte-Carlo simulations to describe the trajectories of particles used in radiotherapy dosimetry and PET devices. Finally, the deployment of virtual clusters as educational infrastructures for HPC-related subjects in Master's Degree.

## IV. Conclusion

This paper has summarised the developments towards self-managed cost-effective elastic virtual clusters on hybrid Cloud infrastructures. So far, the developments of this vision are based on the open source EC3[2] tool, which enables to provision virtual hybrid elastic clusters that span public Clouds (AWS and Google Compute Engine) and on-premises CMPs (OpenNebula, OpenStack and any other OCCI-compliant software), featuring checkpointing capabilities and spot instances support. Supporting OCCI enables the user to provision resources from EGI FedCloud, one of the largest scientific computing platforms. We have released an early version of this tool to the academic community together with the main underlying software components.

We expect to continue our early developments on migration of infrastructures and applications, which will introduce unprecedented flexibility for cluster-based computing.

## Acknowledgments

AC would like to thank the program "Ayudas para la contratación de personal investigador en formación de carcter predoctoral, programa VALi+d", grant number ACIF/2013/003, from the Conselleria d'Educació of the Generalitat Valenciana. Also, the authors would like to thank the Spanish "Ministerio de Economía y Competitividad" for the CLUVIEM project with reference TIN2013-44390-R.

## References

[1] MIT, "StarCluster." [Online]. Available: http://web.mit.edu/stardev/cluster/

[2] U. of Zurich, "Elasticluster." [Online]. Available: http://gc3-uzh-ch.github.io/elasticluster/

[3] G. Juve and E. Deelman, "Wrangler: Virtual Cluster Provisioning for the Cloud," in *Proceedings of the 20th international symposium on High performance distributed computing - HPDC '11*. New York, New York, USA: ACM Press, Jun. 2011, p. 277. [Online]. Available: http://dl.acm.org/citation.cfm?id=1996130.1996173

[4] S. Niu, J. Zhai, X. Ma, X. Tang, and W. Chen, "Cost-effective cloud HPC resource provisioning by building semi-elastic virtual clusters," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis on - SC '13*. New York, New York, USA: ACM Press, Nov. 2013, pp. 1–12. [Online]. Available: http://dl.acm.org/citation.cfm?id=2503210.2503236

[5] G. Moltó, M. Caballer, E. Romero, and C. de Alfonso, "Elastic Memory Management of Virtualized Infrastructures for Applications with Dynamic Memory Requirements," in *Proceedings of the International Conference on Computational Science (ICCS 2013)*. Elsevier, 2013, pp. 159–168.

[6] A. Calatrava, G. Moltó, M. Caballer, and C. D. Alfonso, "Virtual Hybrid Elastic Clusters in the Cloud," in *8th Iberian Grid Infrastructure Conference (IberGrid 2014)*, 2014, pp. 103–114.

[7] M. Caballer, I. Blanquer, G. Moltó, and C. de Alfonso, "Dynamic management of virtual infrastructures," *Journal of Grid Computing*, 2014. [Online]. Available: http://link.springer.com/article/10.1007/s10723-014-9296-5

[2] EC3 - http://www.grycap.upv.es/ec3