Future Generation Computer Systems 61 (2016) 13–25

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Self-managed cost-efficient virtual elastic clusters on hybrid Cloud infrastructures ☆

Amanda Calatrava *, Eloy Romero, Germán Moltó, Miguel Caballer, Jose Miguel Alonso

Instituto de Instrumentación para Imagen Molecular (I3M), Centro mixto CSIC - Universitat Politècnica de València - CIEMAT, Camino de Vera s/n, 46022 Valencia, Spain

HIGHLIGHTS

- Cost-efficient hybrid elastic virtual clusters are deployed across clouds.
- Spot instances and checkpointing reduce the costs of execution.
- Hybrid clusters reduce the total execution time by employing cloud bursting.
- Computationally intensive applications are executed easily with EC3.

ARTICLE INFO

Article history:
Received 24 April 2015
Received in revised form
27 November 2015
Accepted 27 January 2016
Available online 15 February 2016

Keywords:
Checkpointing
Cloud bursting
Cloud computing
Cluster computing
Hybrid cluster
Spot instance

ABSTRACT

In this study, we describe the further development of Elastic Cloud Computing Cluster (EC3), a tool for creating self-managed cost-efficient virtual hybrid elastic clusters on top of Infrastructure as a Service (IaaS) clouds. By using spot instances and checkpointing techniques, EC3 can significantly reduce the total execution cost as well as facilitating automatic fault tolerance. Moreover, EC3 can deploy and manage hybrid clusters across on-premises and public cloud resources, thereby introducing cloud bursting capabilities. We present the results of a case study that we conducted to assess the effectiveness of the tool based on the structural dynamic analysis of buildings. In addition, we evaluated the checkpointing algorithms in a real cloud environment with existing workloads to study their effectiveness. The results demonstrate the feasibility and benefits of this type of cluster for computationally intensive applications.

1. Introduction

The successful use of clusters of PCs as a computing facility is widespread in the scientific community for both high performance computing (HPC) and high throughput computing (HTC). However, these computing platforms have several drawbacks, such as the requirement for a large upfront investment and maintenance costs, which have major economic effects in small and medium-sized organizations. Moreover, the size of a physical cluster cannot be adapted easily to the application workload and they cannot provide customized environments for executing each separate application. In recent years, the development of hypervisors and virtualization technologies have paved the way for cloud computing. This paradigm can address those problems with customizable virtual machines (VMs), which decouple the execution of the application from the underlying hardware, where they are dynamically provisioned and released [1]. Thus, depending on the resource usage and cost model, it might be convenient to deploy a virtual cluster instead of a physical one, as suggested in a previous study [2]. Virtual clusters in the cloud are highly beneficial for many computational workloads, but particularly for highly parallel tasks. These benefits include the on-demand provision of per-application customized clusters as well as the ability to dynamically increase and decrease the number of working nodes in the virtual cluster according to the current workload, as demonstrated in our previous study [3]. Our previous study led to the development of the Elastic Cloud Computing Cluster (EC3)[1] [3] as an open-source tool for the deployment of customized virtual elastic clusters on different on-premises

☆ All the software described in this study is available at https://github.com/grycap.

* Corresponding author. Tel.: +34 963877356; fax: +34 963877359.
E-mail address: amcaar@i3m.upv.es (A. Calatrava).

1 EC3: http://www.grycap.upv.es/ec3.

http://dx.doi.org/10.1016/j.future.2016.01.018
0167-739X/

platforms, such as OpenNebula [4] and OpenStack [5], and public cloud providers, such as Amazon Web Services (AWS) [6]. In the present study, we build on our previous research by introducing two significant features: (i) automatic checkpointing coupled with cost-effective mechanisms for providing transient computing capacity (referred to as spot instances in AWS); and (ii) the ability to deploy hybrid virtual clusters across on-premises and public cloud platforms, according to an elastic scheme that spans different cloud providers.

For the first feature, we exploit the cost-effective advantages provided by cloud providers in order to reduce the total execution cost. This is the case for spot instances, which is a cloud pricing scheme available in Amazon EC2, where the users decide the maximum price that they are willing to pay for an instance, with savings of up to 86% compared with on-demand instances [7]. The user bids on spare Amazon EC2 instances and runs them whenever the bid exceeds the current spot price, which varies in real-time based on supply and demand. This variation causes the instance to terminate if the spot price is higher than the bid by the user, thereby interrupting the execution of the job. This situation is referred to as an "out-of-bid" situation. Recently, Amazon has included spot instance termination notices [8], which provide a two-minute warning before the provider terminates the spot instance. This improvement is useful for some applications, but two minutes is not sufficient time to checkpoint big applications, such as scientific applications, which might require additional time to save their content. Therefore, this type of instance is available at a lower cost but at the expense of reduced reliability. Thus, checkpointing allows the job progress to be saved periodically before the spot instance is terminated by the provider, thereby facilitating job resumption from the last checkpoint. In the present study, we review, propose, and implement checkpointing algorithms for this purpose.

For the second feature, the coexistence of on-premises and public clouds has leveraged cloud bursting, where virtual clusters can be enlarged with resources outside the organization, and thus hybrid clusters can harness on-premises and public cloud resources simultaneously. This approach is highly advantageous because it allows users to seamlessly access cluster-based computing resources in addition to those available via their on-premises clouds. Other topologies can be considered for hybrid clusters when using virtual resources, such as heterogeneous clusters, where various nodes in the cluster have different hardware characteristics.

Therefore, in this study, we extend the capacities of EC3 to allow users to deploy self-managed cost-efficient virtual hybrid elastic clusters on top of Infrastructure as a Service (IaaS) clouds. The remainder of this paper is organized as follows. First, Section 2 reviews related research as well as the main contributions of the present study to the state of the art. Next, Section 3 focuses on the architecture and the new features included in EC3. In Section 4, we present two case studies that we conducted to assess the functionality and benefits of the new features incorporated in EC3. Finally, we give our conclusions in Section 5 and we discuss future research.

2. Related work

Previous studies have aimed to deploy virtual clusters on cloud infrastructures, e.g., StarCluster [9] is an open-source tool that provides clusters in Amazon EC2 based on a predefined configuration for applications (Open Grid Scheduler, OpenMPI, Network File System (NFS), etc.). In addition, CycleCloud [10] is a commercial service provided by CycleComputing that deploys virtual clusters. However, both tools can only provide resources from Amazon EC2 so virtual clusters cannot be deployed through on-premises cloud platforms created with cloud management platforms (CMPs) such as OpenNebula or OpenStack.

Elasticluster [11] can be employed to create virtual clusters on two cloud providers (Amazon EC2 and Google Compute Engine) as well as on-premises cloud platforms (OpenStack supported). The clusters can be scaled by the user but automated elasticity is not supported. Other tools for deploying virtual clusters have also been reported such as ViteraaS [12], which allows the creation of virtual clusters to manage the execution of user-defined jobs, but users are not provided with direct access to the cluster. There are also commercial solutions such as IBM Platform Dynamic Cluster [13], which aims to partition on-premises resources to deliver each user with a custom cluster that has specific features. The features of this system include live job migration and automated checkpoint restart. However, this product was designed for the management of on-premises infrastructures and it cannot be connected to commercial cloud providers.

In terms of the creation of clusters over hybrid cloud infrastructures, previous studies [14–16] have analyzed architectures, algorithms, and frameworks for deploying clusters over these infrastructures, where they analyzed the performance of virtual clusters deployed on top of hybrid clouds and obtained good results to demonstrate the feasibility of this type of deployment. These studies used a fixed number of on-premises nodes and they scaled up the cluster using public nodes. However, the migration of workloads among infrastructures was not considered. In [17–19], the Nimbus toolkit was employed to implement and evaluate an elastic site manager, which dynamically extends existing physical clusters based on Torque using computational resources provided by Amazon EC2 according to different policies. A similar approach was employed by [20], who investigated the benefits of using cloud computing to augment the computing capacity of a local infrastructure, although no details of the underlying technologies were given.

Regarding spot instances, many studies have attempted to develop predictive models of spot price variations, where some of the proposed solutions are based on Gaussian distributions [21] or Markov chains, such as [22,23]. However, other studies found that the spot price variation over time in Amazon EC2 did not seem to follow any particular distribution [24]. It was also observed [25] that Amazon might intervene with the prices artificially by setting a reserve price and generating prices at random, thereby further complicating the prediction of spot price variations.

Another field of research is the deployment of virtual clusters using spot instances. In [26], the economics of purchasing resources on the spot market were considered when handling unexpected load peaks in a cluster, but they did not consider checkpointing techniques. If the instance is terminated, the application is restarted from the beginning. This was the case in [27] where high bids were made rather than employing checkpointing strategies, but this solution can incur higher costs, thereby opposing the main advantage of spot instances. Moreover, Amazon has recently limited [28] the bids made by users to 10 times the on-demand price of the instance. Other solutions have involved deploying a cluster that fully comprises spot instances [29], but again it was assumed that the bid value was sufficiently large to avoid the spot instance being killed by Amazon. Finally, SpotMPI was presented in [30] as a toolkit to facilitate the execution of MPI applications on volatile auction-based cloud platforms. This toolkit can monitor spot instances and bidding prices to automate checkpointing at the bidding price and automatically restart the application after out-of-bid failures. However, this tool has the following limitations. First, it is based on StarCluster so it is restricted to AWS. Second, elasticity management for the clusters is not self-managed inside the cluster because StarCluster implements the elasticity using the Elastic