Article Type

Resource provisioning in Science Clouds: requirements and challenges

Álvaro López García,[1] Enol Fernández-del-Castillo,[1,2] Pablo Orviz Fernández,[1] Isabel Campos Plasencia,[1] and Jesús Marco de Lucas[1]

[1] *Advanced Computing and e-Science Group, Instituto de Física de Cantabria (CSIC - UC), Santander, Spain*
[2] *EGI Foundation, Amsterdam, The Netherlands*

Correspondence: A. López García, IFCA, Adva. los Castros s/n. 39005 Santander, Spain Email: aloga@ifca.unican.es

Summary

Cloud computing has permeated into the IT industry in the last few years, and it is nowadays emerging in scientific environments. Science user communities are demanding a broad range of computing power to satisfy high-performance applications needs, such as local clusters, High Performance Computing (HPC) systems and computing grids. Different workloads need from different computational models, and the cloud is already considered as a promising paradigm.

The scheduling and allocation of resources is always a challenging matter in any form of computation and clouds are not an exception. Science applications have unique features that differentiate their workloads, hence their requirements have to be taken into consideration to be fulfilled when building a Science Cloud. This paper will discuss what are the main scheduling and resource allocation challenges for any Infrastructure as a Service IaaS provider supporting scientific applications.

Keywords: Scientific Computing, Cloud Computing, Science Clouds, Cloud Challenges

1 Introduction

Cloud computing can be defined as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can

---

be rapidly provisioned and released with minimal management effort or service provider interaction." (44). This model allows many enterprise applications to scale and adapt to the usage peaks without big investments in hardware, following a pay-as-you-go model without needing an upfront commitment (4) for acquiring new resources.

This computing paradigm has achieved great success in the IT industry but it is still not common in the scientific computing field. Cloud computing leverages virtualization (40) to deliver resources to the users, and the associated performance degradation was traditionally considered as not compatible with the computational science requirements (54). However, nowadays it is widely accepted that virtualization introduces a CPU overhead that can be neglected (5, 12, 53). This has been confirmed by several studies that have evaluated the performance of the current cloud offerings both in public clouds such as Amazon EC2 (2, 19, 20, 49, 51, 55, 70, 73) or on private and community clouds (12, 18, 25, 26, 29, 57, 65). Moreover, other authors consider that the benefits that virtualization and cloud computing introduces are often more important than a small performance penalty (7, 12).

Therefore, and considering the expectations created around its promising features, scientific communities are starting to look with interest in the cloud. Some of the main characteristics are not novel ideas as they are already present in current computing environments (23): academic researchers have used shared clusters and supercomputers since long, and they are being accounted for their usage in the same pay-per-use basis —i.e. without a fixed fee or upfront commitment— based on their CPU-time and storage consumption. Nevertheless, facts such as the customized environments, resource abstraction and elasticity can fill some of the existing gaps in the current scientific computing infrastructures (23, 74).

Besides, current cloud middleware is designed to satisfy the industry needs. In a commercial cloud provider users are charged in a pay-as-you-go basis, so the customers pay according to their resource consumption. A commercial resource provider might not worry about the actual usage of the resources, as long as they are getting paid by the consumed capacity, even if they are idle resources. This situation is not acceptable in scientific facilities where the maximum utilization of the resources is an objective. Idle resources are an undesirable scenario if it prevents other users from accessing and using the infrastructure. Access to scientific datacenters is not based on a pay per use basis, as user communities are granted with an average capacity over long periods of time. This capacity, even if accounted, is not paid by the users, but it is rather supported by means of long-term grants or agreements.

In traditional scientific datacenters users execute their tasks by means of the well known batch systems, where the jobs are normally time-bounded (i.e. they have a specific duration). Different policies are then applied to adjust the job priorities so that the resources are properly shared between the different users and groups. Even if the user does not specify a duration, a batch system is able to stop its execution after a given amount of time, configured by the resource provider.

However, there is no such <u>duration</u> concept in the cloud model, where a virtual machine is supposed to live as long as the user wants. Users may not stop their instances when they have finished their job (they are not getting charged for them), ignoring the fact that they are consuming resources that may be used by other groups. Therefore, resource providers have to statically partition their resources so as to ensure that all users are getting their share in the worst situation. This leads to an underutilization of the infrastructure, since a usage spike from a group cannot be satisfied by idle resources assigned to another group.
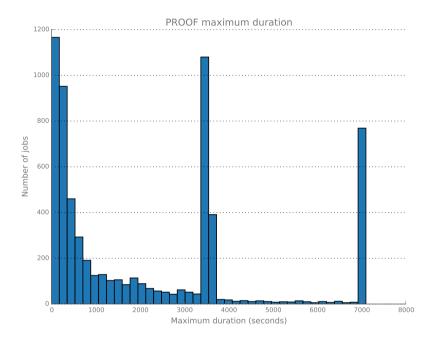

Figure 1: PROOF task duration.

To illustrate this problem we have collected the usage patterns for several months from a batch system specially configured to support this kind of tasks, regarding one application widely used by the High Energy Physics (HEP) community: the Parallel ROOT Facility (PROOF) (3). This tool is used to perform interactive analysis of large datasets produced by the current HEP experiments. Figure 1 shows the number of requests regarding the task duration. As it can be seen, all the requests can be considered short-lived, since its maximum duration is below 2 hours, with the highest concentration being below 1 hour. Figure 2 depicts the request pattern for a 3.5 year period. As it can be seen, this kind of jobs are executed in bursts or waves, meaning that a set of users will have a high demand of resources for short periods of time —i.e. when an analysis is at a final stage.

This kind of usage (i.e. short lived executions that are not constant over the time) is quite common for scientific applications (12, 22, 34, 48, 72) and presents a demanding challenge for resource providers. It is needed to deliver enough computing capacity for absorbing this kind or request, minimizing the reserved resources that will be idle for long periods of time.
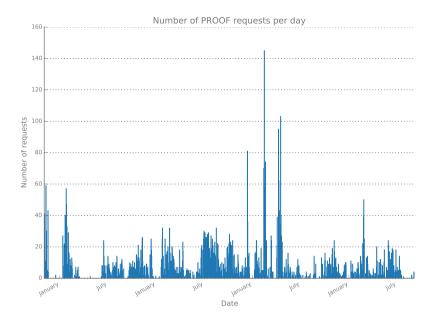
Figure 2: PROOF daily request pattern for a three and a half year period.

Implementing an effective scheduling and resource allocation policies to ensure that the elasticity is perceived as true is a challenging task. The issue in this context is the need of maximizing the utilization of the Infrastructure as a Service (IaaS) resources, so that a minimal amount of physical resources are provisioned and maintained. An allocation policy that is driven by a resource provider decision can result in a low value from the user standpoint, whereas an allocation under user control may result in a high cost for the provider (40).

In addition, satisfying elastic requests in an efficient way is not the sole challenge that a resource provider will face. Scientific applications have unique requirements, therefore Science Clouds shall provide unique features and face unique challenges. In this work we will focus on a gap analysis for a scientific IaaS provider, so that an effective resource allocation can be done. We will not focus in the mere virtual to physical resource mapping, but we will also cover other resource allocation problematic.

The rest of the paper is organized as follows. In Section 2 we will perform a review of the related work in the area. In Section 3 we will cover the open challenges that we have identified from the resource provisioning point of view. Finally, our conclusions are presented in Section 4.

2   Related work

To the best of our knowledge, there are not so many studies considering the resource allocation problematic from the resource provider point of view that take into account the specificity of the scientific application requirements.

There is a considerable amount of research works addressing cloud resource provisioning and scheduling from the user or consumer perspective (14, 31, 71, 72). Some authors have studied how to implement hybrid provisioning of

resources between several cloud providers (47, 63), or even between different computing infrastructures such as grids and clouds (11). The workflow model is widely used in many scientific computing areas, and there is a vast amount of studies regarding the feasibility and challenges of executing workflows in the cloud (27, 33, 38, 39, 56, 59, 67).

The systematic survey performed by Chauhan et al. (15) identified some challenges for High Performance Computing (HPC) and scientific computing in the cloud. More specifically, the survey points to the work by Somasundaram and Govindarajan (61) where the authors develop a framework focused on the execution of HPC applications in cloud environments by managing cloud resources where the user application is dispatched.

Regarding resource provisioning strategies from the provider standpoint, Sotomayor et al. studied how to account and manage the overheads introduced the virtual resources management (62). Hu et al. (30) studied how to deliver a service according to several agreed Service Level Agreements (SLAs) by using the smallest number of resources. Garg et al. (24) presented how to deal with SLAs that imply interactive and non-interactive applications. Cardonha et al. (13) proposed a patience-aware scheduling that take into account the user's level of tolerance (i.e. the patience) to define how to deliver the resources to the users.

There is large number of research works regarding energy aware resource provisioning in the clouds (6, 10, 50). Smith et al. (60) modelled how different workloads affected energy consumption, so that an accurate proper power prediction could be made to perform an efficient scheduling. Several authors have studied how the consolidation of virtual servers in a cloud provider could lead to a reduction of the energy consumption (17, 64). This fact can be used to increase the revenues by implementing energy-aware resource allocation policies (42).

Kune et al. (37) elaborated an exhaustive taxonomy of big data computing, including a discussion on the existing challenges and approaches for big data scheduling (among others). This work also includes an study of the underpinning technologies for big data cloud computing, as long as a gap analysis in the current architectures and systems.

Manvi et al. (40) performed an exhaustive review of the resource provisioning, allocation and mapping problems for a IaaS resource provider, stating some open challenges like **i)** how to design a provisioning algorithm for optimal resource utilization based on arrival data; **ii)** how and when to reallocate VMs; **iii)** how to minimize the cost of mapping the request into the underlying resources; **iv)** how to develop models that are able to predict applications performance ; among many others.

On the other hand, there are previous studies regarding the general challenges for Science Clouds. The work by Blanquer et al. (8), in the scope of the VENUS-C project, evaluated the requirements of scientific applications by performing a broad survey of scientific applications within the project. Their study showed that the cloud computing model is perceived as beneficial by the users (being one of the key expectations the elasticity), although some drawbacks need to be tackled so as to improve its adoption (such as interoperability, learning curve, etc.).

Juve et al. (35) outlines what is expected from a Science Cloud in contrast with a commercial provider (shared memory, parallel applications, shared filesytems) so as to effectively support scientific workflows. Besides, it concluded that cloud can be beneficial for scientific users, assuming that Science Clouds will be build ad-hoc for its users, clearly differing from commercial offers.

The United States Department of Energy (DOE) Magellan project elaborated an extensive report on the usage of cloud computing for science (75) by deploying several cloud infrastructures that were provided to some selected mid-range computing and data intensive scientific applications. Their key findings include **i)** the identification of advantages of the cloud computing model, like the availability of customized environments for the user or flexible resource management; **ii)** the requirement of additional programming and system administration skills in order to adopt the cloud computing model; **iii)** the economic benefit of the cloud computing model from the provider perspective due to the resource consolidation, economies of scale and operational efficiency; and **iv)** some significant gaps that exist in several areas, including resource management, data, cyber-security and others.

Regarding this last finding, the study concluded that there are several open challenges that science clouds need to address in order to ensure that scientists can harness all the capabilities and potential that the cloud is able to offer. These needs derive from the special requirements that scientific applications have, and were collected in a further publication by Ramakrishnan et al. (52). The authors conclude that science clouds **i)** need access to low-latency interconnects and filesystems; **ii)** need access to legacy data-sets; **iii)** need MapReduce implementations that account for characteristics of science data and applications; **iv)** need access to bare metal provisioning; **v)** need pre-installed, pre-tuned application software stacks; **vi)** need customizations for site-specific policies; and **vii)** need more sophisticated scheduling methods and policies. Some of those findings are coincident with the gaps that we have identified in this work specially those regarding with resource management (like access to specialized hardware) and scheduling policies, but further elaboration is needed on them.

3 Resource provisioning in Science Clouds

Scientific workloads involve satisfying strong requirements. Resource allocation for scientific applications appears then as a demanding task that should take into consideration a number of hardware and software variables. As scientific applications started to move to cloud solutions, this number of requirements got increased: on top of the already existing needs, new requirements arose from the defining characteristics that the new paradigm of cloud computing offered to users: *on-demand self-service provisioning* needs richer computing capabilities definitions for applications with e.g. very specific demanding hardware requirements like guaranteeing a minimum network bandwidth for remote data access, commonly found in scientific environments. In the same line, *elastic provisioning*

is required to be highly customizable for the sake of minimizing customers' budgets and administrative costs for the service providers. Granular and customizable environments increase <u>predictability</u> so that the providers can offer performance guarantees to customers while estimating accurately the costs of resource utilization. Elasticity needs to be rapid as well: reducing e.g. <u>instance startup</u> will benefit a fast (auto-)scaling of resources.

In this section we will also cover other non-cloud inherent scientific requirements, most of which were traditionally tackled in previously proposed computing paradigms, such as grid computing and HPC clusters. Science Clouds will need to provide resource provisioning methods and policies to satisfy complex requirements such as <u>resource co-allocation</u> or <u>performance and data aware-based provisioning</u>. But popular open source cloud frameworks do not count with schedulers that have built-in mechanisms and policy-definition to satisfy these requirements. Cloud schedulers surely are not meant to offer the advanced set of scheduling possibilities that a standard batch system has, but they definitely need to address those requirements commonly found in scientific computations. A clear example is the execution of non-interactive applications. Batch executions are needed in multiple scientific use cases, so it appears to be reasonable to add <u>flexible allocation policies</u> to deal with this type of executions.

In the following lines we elaborate on the above identified requirements and, for some cases, depict what resource allocation challenges and solutions can be applied within Science Clouds.

3.1 Instance co-allocation

Compute and data intensive scientific workloads tend to use parallel techniques to improve their performance. Parallel executions are complex since they require intercommunication between processes, usually located in distributed systems, scenario in which resource provisioning task becomes even more challenging. Based on the assumption that a provider is capable of satisfying a request involving different instances, one have to consider the fact of managing them as a whole so to assure that these instances are actually being provisioned at the same time i.e. they are being <u>co-allocated</u> (in this context, instance co-allocation is not related with executing several instances in the same physical node, but rather that the instances are provisioned to the user at the same time). Proper co-allocation policies should take into account network requirements, such as satisfying low latencies and appropriate bandwidths, and fulfil any constraints imposed by the parallel framework being used, as e.g. OpenMPI's intra-subnet allocation check (19).

If a proper co-allocation mechanism is not in place, users and resource providers would need to coordinate in order to pre-provision the required instances (32), therefore hindering the on demand and self-service experience that is expected from a cloud system.

In homogeneous and static environments, guaranteeing ordered co-allocation of resources can be easily tackled, if compared to heterogeneous scenarios. In the specific case of cloud computing, the flexibility that it introduces, makes multi-resource allocation a challenging task that must take into consideration not only the synchronized startup (see

more at Section 3.7) of master and worker instances, but also how these resources are geographically distributed and what are the hardware constraints (network, cpu, memory) to be considered. Only by doing this, parallel tasks provisioned in clouds would have a similar application performance than what can be obtained with homogeneous ad-hoc resources, but getting rid of the rigidity that the introduce.

3.1.1 Instance co-allocation open challenges

The open challenges in this area are as follows:

- How to offer a proper SLA to ensure that instances need to be co-allocated.

- How to ensure that instances that need co-allocation are actually started at the same time.

- How to account (or not account) for instances that requiring co-allocation have been provisioned with an unacceptable delay. When a user is requiring this feature but the requirement cannot be fulfilled this should be taken into account.

- How to ensure that when the instances are already scheduled they are allocated within a time-frame. VM management introduces overheads and delays that should be taken into account to ensure a proper co-allocation.

3.2 Licensed software management

One of the major barriers scientists find when moving their applications to the cloud relies in licensing troubles. Software vendors that count with policies about how to deal with licensing in virtualized environments propose the usage of Floating Network Licenses (FNL). These special licenses usually increment costs, as they can be used by different virtual instances, and require the deployment of license managers in order to be able to use the software in the cloud infrastructures. Additionally, the license managers might need to be hosted within a organization's network.

Using FNLs are the most popular solution provided by vendors, but the imposed requirements mentioned above can be difficult to satisfy in some cases: hosting a license manager is not always possible by some scientific communities and it introduces maintenance costs, whose avoidance is one of the clear benefits of moving to a cloud solution.

The need for a more straightforward way of getting licensed or proprietary software to work in virtualized environments is a must that software vendors should consider. In commercial cloud infrastructures, like Amazon AWS, customers can make use of pre-configured images, license-granted, with the proprietary software locally available and ready to use. At the time of writing, Amazon AWS does not have agreements with all of the major software vendors, but it appears as a neat and smooth solution that requires no extra work from the end users side.

Besides the above administrative difficulties, the actual technical challenge in resource allocation for licensed software is that cloud schedulers are not license-aware. This gap needs to be filled by the cloud middleware stacks, as it was solved years ago in HPC clusters.

3.2.1 Licensed software management open challenges

The open challenges in this area are as follows:

- Persuade commercial vendors to release more flexible licensing methods, specific for the cloud.

- How to deal with license slots within the scheduler.

3.3 Performance aware placement

In order to improve resource utilization, cloud schedulers can be configured to follow a fill-up strategy that might end up in multiple virtual machines running concurrently on the same physical server. This scenario leads to resource competition which surely will affect application performance. In this regard, the scheduler that is in charge of provisioning the resources in Science Clouds needs to be performance-aware (or even degradation-aware), so that performance demanding instances do not share the physical resources with other instances that may impact its performance.

Several approaches have been raised in order to diminish degradation. Some do not act directly on pro-active scheduling but instead in reactive reallocation of the affected virtual instances by using underneath hypervisor capabilities like live migration. But, instead of relying in monitoring the application performance and take reallocation decisions based upon its degradation, a more pro-active scheduling is needed so to improve the suitability of the resource selection. Feeding the scheduler with more fine-grained hardware requirements, provided by the user request, such as low-latency interconnects (e.g. Infiniband, 10GbE) or GPGPU (16) selection, provides a better resource categorization and, consequently, will directly contribute to a more efficient execution of the application. To accomplish this, the specialized hardware must be exposed into the virtual instances, by means of PCI passthrough with IOMMU or Single Root I/O Virtualization (SR-IOV) techniques, and eventually managed by the cloud middleware using the underlying virtualization stack.

Therefore, consolidating virtual machines into the same physical host should not be applied when the instances are executing performance demanding applications. Virtualization in these cases is used only a as a way to provide customized environments for scientists. However, it should be noted that science clouds can apply consolidation techniques for non demanding applications, such as web portals or science gateways.

The hypervisor providing the virtualization appears as an important factor when measuring performance. It is widely accepted that virtualization introduces a penalty when compared with bare metal executions. However this
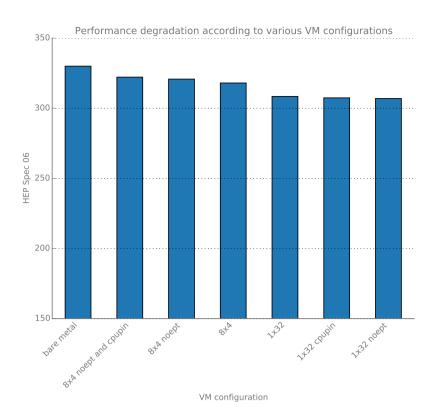
Figure 3: Aggregated performance regarding the HEP Spec 06 (45) benchmark, taking into account different virtual machine sizes and configurations for one host. The physical node consists on a node with two 8-core Intel®Xeon®E5-2670 2.60GHz processors, 128GB RAM and the virtual machines were dimensioned so as to consume —in aggregate— all the resources available on the host. The label "noept" means that the Extended Page Tables (EPT) support has been disabled. The label "cpupin" means that the virtual CPUs have been pinned to the physical CPUs

penalty depend on how the hypervisor is being used. Figure 3 shows the degradation of the aggregated performance delivered by a physical machine, using different vCPUs sizes.

Science Clouds need to deliver the maximum performance possible. Therefore, the cloud middleware should take this fact into account, by implementing scheduling policies that would help to prevent the above identified performance drops.

3.3.1 Performance aware placement open challenges

The open challenges in this area are as follows:

- How to minimize the performance loss when scheduling various virtual machines inside one host.

- How to proactively scheduling could be used to minimize resource competition.

- How to detect performance interferences between VMs and take appropriate actions (like live migration) to minimize them.

- How to redistribute the running instances between the resources without impacting the running applications.

- How to apply consolidation techniques that do not interfere with a scheduling strategy ensuring that performance demanding applications are not executed in a time sharing manner.

3.4 Data-aware scheduling

Several scientific disciplines —such as High Energy Physics (HEP), Astronomy or Genomics just to cite some of them— generate considerably large amounts of data (in the order of Petabytes) that need to be analyzed. Location and access modes have clear impacts to data-intensive applications (36, 37, 58, 68) and any platform that supports these kind applications should provide data-locality and data-aware scheduling to reduce any possible bottlenecks that may even prevent the actual execution of the application.

Storage in clouds is normally decoupled from the virtual machines and attached during runtime upon user's demand. This poses a bigger challenge to the scheduler since the location of data to be accessed is not known a priori by the system. Science Clouds should be able to provide high-bandwidth access to the data, which is usually accessed over the network (e.g. block storage may use ATA over Ethernet or iSCSI; object storage usually employs HTTP). This may require enabling the access to specialized hardware from the virtual machines (e.g. Infiniband network) or relocating the virtual machines to hosts with better connectivity to the data sources. Data-locality can also be improved by using caches at the physical nodes that host the VMs, by replicating locally popular data hosted externally to the cloud provider, or by leveraging tools like CernVMFS (9) that deliver fast access to data using HTTP proxies.

3.4.1 Data-aware scheduling open challenges

The open challenges in this area are as follows:

- How to take into account cloud data management specificities when scheduling machines.

- How to ensure that the access delivers high performance for the application being executed.

3.5 Flexible resource allocation policies

Long-running tasks are common in computational science. Those kind of workloads do not require from interactivity and normally are not time-bounded. Such tasks can be used as opportunistic jobs that fill the computing infrastructure usage gaps, leading to a better utilization of resources.

In traditional scientific datacenters and time-sharing facilities this is normally done in by means of several techniques, such as backfilling, priority adjustments, task preemption and checkpointing. Some of these techniques require that the tasks are time-bounded, but in the cloud a virtual machine will be executed as long as the user wants.

Commercial cloud providers have tackled this issue implementing the so called spot instances or preemptible instances. This kind of instances can be terminated without further advise by the provider if some policy is violated

(for example, if the resource provider cannot satisfy a normal request —in the preemptible case— or because the user is paying a prize that is considered too low over a published price baseline —in the spot mode, where the price is governed by a stock-options like market.

The usage of this kind of instances in Science Clouds could make possible that the infrastructure is filled with opportunistic (28) jobs that can be stopped by higher priority tasks, such as interactive demands. The Vacuum computing model (43), where resources appear in the vacuum to process some tasks and then disappear is an ideal candidate to leverage this kind of spot instances. Tools such as Vcycle (1) or SpotOn (66) are already being used to profit from opportunistic usage in existing commercial or scientific infrastructures.

3.5.1   Flexible resource allocation policies open challenges

The open challenges in this area are as follows:

- How to maximize the resource utilization without preventing interactive users from accessing the infrastructure.

- How to specify dependencies between virtual machines so that workflows can be scheduled in a more easy way.

- How to account for resources that are suitable for being stopped or preempted.

- How to select the best instances that can be stopped to leave room for higer priority requests, with the compromise of reducing the revenue loss and with the smallest impact to the users.

3.6   Performance predictability

Popular open-source Infrastructure as a Service frameworks do not currently expose mechanisms for customers to define a specific set of hardware requirements that would guarantee a minimum performance when running their applications in the cloud. Real time demanding or latency sensitive applications are indeed seriously hit by this limitation, which appears as a big obstacle for integrating this type of applications into clouds.

Computing capabilities provide only a magnitude of multi-threading efficiency based on the number of virtual CPUs (vCPUs) selected. Customers are then tied to a generic vCPU selection that may be mapped to different processors by the underlying framework, in which case different performance results could be obtained based on the same set of requirements. This unpredictability will be increased whenever resource overcommit is in place, that could lead to CPU cycle sharing among different applications.

Lack of network performance guarantees contribute also to unexpected application behaviour. Enforcing network Quality of Service (QoS) to achieve customer-required network bandwidth can greatly improve application predictability, but network requirement selection are seldom offered by cloud providers (46).

Improved performance predictability is a key requirement for users (21) but also to providers. The lack of predictability leads to uncertainty (69), a fact that should be mitigated for both users and providers. An accurate provision of customer needs in terms of computing and network capabilities will not only boost customer experience but also will provide a clear estimation of cost based on the different service levels that the resource provider can offer.

3.6.1   Performance predictability open challenges

The open challenges in this area are as follows:

- How to expose enough granularity in the request specification without exposing the underlying abstracted resources.

- How to guarantee the performance predictability between different requests with the same hardware requirements.

3.7   Short startup overhead

When a request is made, the corresponding images have to be distributed from the catalogue to the compute nodes that will host the virtual machines. If the catalogue repository is not shared or the image is not already cached by the compute nodes, this distribution will introduce a penalty on the start time of the requested nodes. This overhead can be quite significant (52) and has a large influence in the startup time for a request. This is specially true when large (41) requests are made by a user. Figure 4 shows this effect in an OpenStack test infrastructure. The 2GB images were distributed using HTTP transfers to 35 hosts over a 1GbE network interconnect. As it can be seen, the time needed to get all the machines within a single request increased with the size of the request.

Parallel applications are common in scientific workloads, so a mechanism should be provided to ensure that these large request are not penalized by this transfer and deployment time. Users requiring interactivity cannot afford to wait for several minutes for an instance to be spawned, since interactivity implies immediateness. This is specially important for the co-allocation of instances, as described in Section 3.1, since the VM provision time may impact in the delivery time to the users, hindering the co-allocation of resources.

3.7.1   Short startup overhead open challenges

- How to deploy the images into the nodes in an efficient way.

- How to deal with spikes on the requests, so that the systems are not saturated transmitting the images into a large number of nodes.
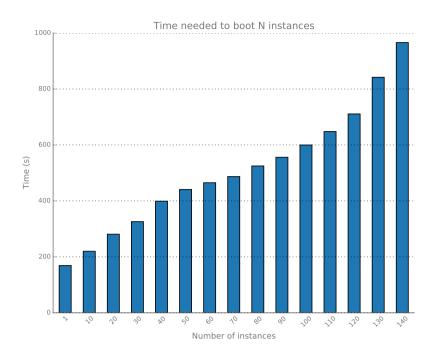
Figure 4: Time needed to boot the number of requested instances. Tests were performed in a dedicated infrastructure based on OpenStack with 35 hosts over a 1GbE network with an image of 2GB.

- How to implement cache mechanisms in the nodes, implementing sanity checks so that similar workloads are not constrained into a few nodes.

- How to forecast workloads, so that images can be pre-deployed, anticipating the user's requests.

4   Conclusions

In this paper we have depicted and elaborated on the resource allocation open challenges for cloud frameworks, based on the analysis of scientific applications requirements. In this context, we have identify cloud providers as Science Clouds, since they might not have the same expectations, requirements and challenges as any other private or commercial cloud infrastructure.

Cloud Management Frameworks (CMFs) are normally being developed taking into account the point of view of a commercial provider, focusing on satisfying the industry needs, but not really fulfilling academia demands. Scientific workloads are considered as high-performance computing tasks that need strong requirements. Some of them were tackled in previous computing paradigms and now there is the need to address them in Science Clouds. Other resource allocation requirements identified in this paper are inherent to cloud computing and would provide the predictability that cloud frameworks currently lack. These requirements naturally evolve into challenges that, as the time of writing, appear as obstacles for moving certain scientific workflows to Science Clouds.

The cloud is not a silver bullet for scientific users, but rather a new paradigm that will enter the ecosystem. In the upcoming years scientific computing datacenters have to move towards a mixed and combined model, where a given user will have access to the more <u>traditional</u> computational power, but also they should provide their users with additional cloud power that will complement the former computing infrastructures. These Science Clouds should need to be tuned to accommodate the demands of the user communities supported. This way, either the users the users will benefit from a richer environment, and resource providers can get a better utilization of their resources, since they will allow for new execution models that are currently not available.

References

1. *Vcycle: VM lifecycle management*; 2014

2. Afgan Enis, Baker Dannon, Coraor Nate, Chapman Brad, Nekrutenko Anton, Taylor James. Galaxy CloudMan: delivering cloud compute clusters. *BMC bioinformatics*. 2010;11(Suppl 12).

3. Antcheva I, Ballintijn M, Bellenot B, Biskup M, Brun R, Buncic N, Canal Ph., Casadei D, Couet O, Fine V, Franco L, Ganis G, Gheata A, Maline D Gonzalez, Goto M, Iwaszkiewicz J, Kreshuk A, Segura D Marcos, Maunder R, Moneta L, Naumann A, Offermann E, Onuchin V, Panacek S, Rademakers F, Russo P, Tadel M. ROOT – A C++ framework for petabyte data storage, statistical analysis and visualization. *Computer Physics Communications*. 2009;180(12):2499–2512.

4. Armbrust Michael, Stoica Ion, Zaharia Matei, Fox Armando, Griffith Rean, Joseph Anthony D., Katz Randy, Konwinski Andy, Lee Gunho, Patterson David, Rabkin Ariel. A view of cloud computing. *Communications of the ACM*. 2010;53(4):50.

5. Barham Paul, Dragovic Boris, Fraser Keir, Hand Steven, Harris Tim, Ho Alex, Neugebauer Rolf, Pratt Ian, Warfield Andrew. (). *Xen and the art of virtualization*, Proceedings of the nineteenth acm symposium on operating systems principles se - sosp '03, pp. 164–177.

6. Beloglazov Anton, Abawajy Jemal, Buyya Rajkumar. Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Future Generation Computer Systems*. 2012;28(5):755–768.

7. Birkenheuer Georg, Brinkmann André, Kaiser Jürgen, Keller Axel, Keller Matthias, Kleineweber Christoph, Konersmann Christoph, Niehörster Oliver, Schäfer Thorsten, Simon Jens, Wilhelm Maximilian. Virtualized HPC: a contradiction in terms?. *Software - Practice and Experience*. 2012apr;42(4):485–500.

8. Blanquer Ignacio, Brasche Goetz, Lezzi Daniele. (). *Requirements of Scientific Applications in Cloud Offerings*, Proceedings of the 2012 sixth iberian grid infrastructure conference, pp. 173–182.

9. Blomer J, Buncic P, Charalampidis I, Harutyunyan A, Larsen D, Meusel R. Status and future perspectives of cernvm-fs. *Journal of Physics: Conference Series*. 2012;396(5):052013.

10. Buyya Rajkumar, Beloglazov Anton, Abawajy Jemal. Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges. 2010.

11. Calheiros Rodrigo N., Vecchiola Christian, Karunamoorthy Dileban, Buyya Rajkumar. The Aneka platform and QoS-driven resource provisioning for elastic applications on hybrid Clouds. *Future Generation Computer Systems*. 2012;28(6):861–870.

12. Campos Plasencia Isabel, Fernández-del Castillo Enol, Heinemeyer S., López García Álvaro, Pahlen F., Borges G., Lopez Garcia Alvaro. Phenomenology tools on cloud infrastructures using OpenStack. *The European Physical Journal C.* 2013;73(4):2375, available at arXiv:1212.4784v1.

13. Cardonha Carlos, Assunção Marcos D, Netto Marco A S, Cunha Renato L F, Queiroz Carlos. *Patience-aware scheduling for cloud services: Freeing users from the chains of boredom* (Basu Samik, Pautasso Cesare, Zhang Liang, Fu Xiang, eds.): Springer Berlin Heidelberg; 2013.

14. Chaisiri Sivadon, Kaewpuang Rakpong, Lee Bu Sung, Niyato Dusit. Cost minimization for provisioning virtual servers in amazon elastic compute cloud. *IEEE International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems - Proceedings.* 2011:85–95.

15. Chauhan Muhammad Aufeef, Babar Muhammad Ali, Benatallah Boualem. Architecting cloud-enabled systems: a systematic survey of challenges and solutions. *Software - Practice and Experience.* 2017;47(4):599–644, available at 1008.1900.

16. Chen Dan, Hu Yangyang, Cai Chang, Zeng Ke, Li Xiaoli. Brain big data processing with massively parallel computing technology: challenges and opportunities. *Software - Practice and Experience.* 2017;47(3):405–420.

17. Corradi Antonio, Fanelli Mario, Foschini Luca. VM consolidation: A real case based on OpenStack Cloud. *Future Generation Computer Systems.* 2014mar;32:118–127.

18. de Oliveira Daniel, Ocaña Kary a. C. S., Baião Fernanda, Mattoso Marta. A Provenance-based Adaptive Scheduling Heuristic for Parallel Scientific Workflows in Clouds. *Journal of Grid Computing.* 2012;10(3):521–552.

19. Evangelinos Constantinos, Hill Chris. (). *Cloud Computing for parallel Scientific HPC Applications: Feasibility of Running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2*, The 1st workshop on cloud computing and its applications (cca), pp. 2–34.

20. Expósito Roberto R., Taboada Guillermo L., Ramos Sabela, Touriño Juan, Doallo Ramón. Performance analysis of HPC applications in the cloud. *Future Generation Computer Systems.* 2013;29(1):218–229.

21. Fakhfakh F, Kacem H H, Kacem A H. Workflow Scheduling in Cloud Computing: A Survey. *Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), 2014 IEEE 18th International.* 2014;71(9):372–378.

22. Fernández Albor Víctor, Seco Marcos, Méndez Muñoz Víctor, Fernández Tomás, Silva Pena Juán Saborido, Graciani Diaz Ricardo. Multivariate Analysis of Variance for High Energy Physics Software in Virtualized Environments. *International Symposium on Grids and Clouds, Academia Sinica, Taipei, Taiwan.* 2015;160:1–15.

23. Foster Ian, Zhao Yong, Raicu Ioan, Lu Shiyong. (). *Cloud computing and grid computing 360-degree compared*, Grid computing environments workshop, 2008. gce'08, pp. 1–10.

24. Garg Saurabh Kumar, Gopalaiyengar Srinivasa K., Buyya Rajkumar. SLA-Based Resource Provisioning for Heterogeneous Workloads in a Virtualized Cloud Datacenter. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* 2011;7016 LNCS(PART 1):371–384.

25. Gunarathne Thilina, Wu Tak Lon, Choi Jong Youl, Bae Seung Hee, Qiu Judy. Cloud computing paradigms for pleasingly parallel biomedical applications. *Concurrency Computation Practice and Experience.* 2011;23(17):2338–2354.

26. Gupta Abhishek, Milojicic Dejan. (). *Evaluation of hpc applications on cloud*, Open cirrus summit (ocs), 2011 sixth, pp. 22–26.

27. Hardt Marcus, Jejkal Thomas, Campos Plasencia Isabel, Fernández-del Castillo Enol, Jackson Adrian, Weiland Michele, Palak Bartek, Plociennik Marcin, Nielsson Daniel. Transparent Access to Scientific and Commercial Clouds from the Kepler Workflow Engine. *Computing and Informatics*. 2012;31(1):119.

28. Hategan M., Wozniak J., Maheshwari K. Coasters: Uniform Resource Provisioning and Access for Clouds and Grids. *Fourth IEEE International Conference on Utility and Cloud Computing*. 2011:114–121.

29. Hoffa Christina, Mehta Gaurang, Freeman Tim, Deelman Ewa, Keahey Kate, Berriman Bruce, Good John. (). *On the Use of Cloud Computing for Scientific Workflows*, 2008 ieee fourth international conference on escience, pp. 640–645.

30. Hu Ye, Wong Johnny, Iszlai Gabriel, Litoiu Marin. Resource provisioning for cloud computing. *Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research - CASCON '09*. 2009:101.

31. Huang He, Wang Long, Tak BC, Tang Chunqiang. (). *CAP 3: A Cloud Auto-Provisioning Framework for Parallel Processing Using On-demand and Spot Instances*, Ieee sixth international conference on cloud computing (cloud), 2013, pp. 228–235.

32. Ismail Leila, Barua Rajeev. Implementation and performance evaluation of a distributed conjugate gradient method in a cloud computing environment. *Software - Practice and Experience*. 2012:1–27.

33. Jung Daeyong, Lim JongBeom, Yu Heonchang, Gil JoonMin, Lee EunYoung. A workflow scheduling technique for task distribution in spot instance-based cloud 2014 (pp. 409–416)., Ubiquitous information technologies and applications: Springer.

34. Juve Gideon, Deelman Ewa. Resource Provisioning Options for Large-Scale Scientific Workflows. *2008 IEEE Fourth International Conference on eScience*. 2008:608–613.

35. Juve Gideon, Deelman Ewa. Scientific workflows and clouds. *Crossroads*. 2010;16(3):14–18.

36. Kosar T. (). *A new paradigm in data intensive computing: Stork and the data-aware schedulers*, Challenges of large applications in distributed environments, 2006 ieee, pp. 5–12.

37. Kune Raghavendra, Konugurthi Pramod Kumar, Agarwal Arun, Chillarige Raghavendra Rao, Buyya Rajkumar. The anatomy of big data computing. *Software - Practice and Experience*. 2016;46(1):79–105, available at 1008.1900.

38. Lee Young Choon, Han Hyuck, Zomaya Albert Y., Yousif Mazin. Resource-efficient workflow scheduling in clouds. *Knowledge-Based Systems*. 2015;80:153–162.

39. Lin Xiangyu, Wu Chase Qishi. On scientific workflow scheduling in clouds under budget constraint. *Proceedings of the International Conference on Parallel Processing*. 2013:90–99.

40. Manvi SS Sunilkumar S. SS, Shyam GK, Krishna Shyam Gopal. Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. *Journal of Network and Computer Applications*. 2014;41:424–440.

41. Mao Ming, Humphrey Marty. (). *A Performance Study on the VM Startup Time in the Cloud*, 2012 ieee fifth international conference on cloud computing, pp. 423–430.

42. Mazzucco Michele, Dyachuk Dmytro, Deters Ralph. (). *Maximizing Cloud Providers Revenues via Energy Aware Allocation Policies*, Ieee 3rd international conference on cloud computing (cloud), 2010.

43. McNab a, Stagni F, Garcia M Ubeda. Running Jobs in the Vacuum. *Journal of Physics: Conference Series*. 2014;513(3):32065.

44. Mell Peter, Grance Tim. *The NIST definition of cloud computing*. Special Publication 800-145: National Institute of Standards and Technology ({NIST}); 2011.

45. Michelotto Michele, Alef Manfred, Iribarren Alejandro, Meinhard Helge, Wegner Peter, Bly Martin, Benelli Gabriele, Brasolin Franco, Degaudenzi Hubert, Salvo Alessandro De, Gable Ian, Hirstius Andreas, Hristov Peter. A comparison of HEP code with SPEC benchmarks on multi-core worker nodes. *Journal of Physics: Conference Series*. 2010;219(5):52009.

46. Mogul Jeffrey C, Popa Lucian. What we talk about when we talk about cloud network performance. *ACM SIGCOMM Computer Communication Review*. 2012;42(5):44–48.

47. Montero Ruben S., Moreno-Vozmediano Rafael, Llorente Ignacio M. An elasticity model for High Throughput Computing clusters. *Journal of Parallel and Distributed Computing*. 2011jun;71(6):750–757.

48. Nilsson Paul, De Kaushik, Filipcic Andrej, Klimentov Alexei, Maeno Tadashi, Oleynik Danila, Panitkin Sergey, Wenaus Torre, Wu Wenjing. (). *Extending ATLAS Computing to Commercial Clouds and Supercomputers*, The international symposium on grids and clouds (isgc), pp. 1–11.

49. Oesterle F., Ostermann S., Prodan R., Mayr G. J. Experiences with distributed computing for meteorological applications: grid computing and cloud computing. *Geoscientific Model Development*. 2015;8(7):2067–2078.

50. Orgerie Anne-Cécile, Assunção Marcos, Lefèvre Laurent. Energy Aware Clouds,In Cafaro Massimo, Aloisio Giovanni (eds.) 2011 (pp. 143–166)., Computer communications and networks London: Springer London.

51. Ostermann Simon, Iosup Alexandru, Yigitbasi Nezih, Prodan Radu, Fahringer Thomas, Epema Dick. A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing 2010 (pp. 115–131)., Cloud computing: Springer.

52. Ramakrishnan Lavanya, Zbiegel PT T. (). *Magellan: experiences from a science cloud*, Proceedings of the 2nd international workshop on scientific cloud computing, pp. 49–58.

53. Ranadive Adit, Kesavan Mukil, Gavrilovska Ada, Schwan Karsten. (). *Performance implications of virtualizing multicore cluster machines*, Hpcvirt '08: Proceedings of the 2nd workshop on system-level virtualization for high performance computing, pp. 1–8.

54. Regola Nathan, Ducom Jean-Christophe. (). *Recommendations for Virtualization Technologies in High Performance Computing*, Second international conference on cloud computing technology and science, 2010 ieee, pp. 409–416.

55. Rehr John, Vila Fernando, Gardner Jeffrey, Svec Lukas, Prange Micah. Scientific Computing in the Cloud. *Computing in Science & Engineering*. 2011.

56. Rodriguez Maria Alejandra, Buyya Rajkumar. Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds. *IEEE Transactions on Cloud Computing*. 2014;2(2):222–235.

57. Rodríguez-Marrero Ana Y, González Caballero Isidro, Cuesta Noriega Alberto, Fernández-del Castillo Enol, López García Álvaro, Marco de Lucas Jesús, Matorras Weinig Francisco. Integrating PROOF Analysis in Cloud and Batch Clusters. *Journal of Physics: Conference Series*. 2012;396(3):032091.

58. Shamsi Jawwad, Khojaye Muhammad Ali, Qasmi Mohammad Ali. Data-Intensive Cloud Computing: Requirements, Expectations, Challenges, and Solutions. *Journal of Grid Computing*. 2013;11(2):281–310.

59. Smanchat Sucha, Viriyapant Kanchana. Taxonomies of workflow scheduling problem and techniques in the cloud. *Future Generation Computer Systems*. 2015;52:1–12.

60. Smith James W, Sommerville Ian. Workload Classification & Software Energy Measurement for Efficient Scheduling on Private Cloud Platforms. *Time.* 2011;abs/1105.2:10, available at 1105.2584.

61. Somasundaram Thamarai Selvi, Govindarajan Kannan. CLOUDRB: A framework for scheduling and managing High-Performance Computing (HPC) applications in science cloud. *Future Generation Computer Systems.* 2014;34:47–65.

62. Sotomayor Borja, Keahey Kate, Foster Ian. (). *Overhead Matters: A Model for Virtual Resource Management*, Proceedings of the 2nd international workshop on virtualization technology in distributed computing se - vtdc '06, pp. 5.

63. Sotomayor Borja, Montero Rubén S., Llorente Ignacio M., Foster Ian. Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Computing.* 2009;13:14–22.

64. Srikantaiah Shekhar, Kansal Aman, Zhao Feng. (). *Energy Aware Consolidation for Cloud Computing*, Proceedings of hotpower '08 workshop on power aware computing and systems.

65. Srirama Satish Narayana, Jakovits Pelle, Vainikko Eero. Adapting scientific computing problems to clouds using MapReduce. *Future Generation Computer Systems.* 2012;28(1):184–192.

66. Subramanya Supreeth, Guo Tian, Sharma Prateek, Irwin David, Shenoy Prashant. (). *Spoton: A batch computing service for the spot market*, Proceedings of the sixth acm symposium on cloud computing, pp. 329–341.

67. Szabo Claudia, Sheng Quan Z., Kroeger Trent, Zhang Yihong, Yu Jian. Science in the Cloud: Allocation and Execution of Data-Intensive Scientific Workflows. *Journal of Grid Computing.* 2014;12(2):245–264.

68. Tan Yu Shyang, Tan Jiaqi, Chng Eng Siong, Lee Bu Sung, Li Jiaming, Date Susumu, Chak Hui Ping, Xiao Xiong, Narishige Atsushi. Hadoop framework: Impact of data organization on performance. *Software - Practice and Experience.* 2013;43(11):1241–1260.

69. Tchernykh Andrei, Schwiegelsohn Uwe, Alexandrov Vassil, Talbi El-ghazali. Towards Understanding Uncertainty in Cloud Computing Resource Provisioning. *Procedia Computer Science.* 2015;51:1772–1781.

70. Vöckler Jens-Sönke, Juve Gideon, Deelman Ewa, Rynge Mats, Berriman Bruce. (). *Experiences Using Cloud Computing for a Scientific Workflow Application*, Proceedings of the 2nd international workshop on scientific cloud computing, pp. 15–24.

71. Voorsluys William, Buyya Rajkumar. Reliable provisioning of spot instances for compute-intensive applications. *Proceedings - International Conference on Advanced Information Networking and Applications, AINA.* 2012:542–549, available at 1110.5969.

72. Voorsluys William, Garg Saurabh Kumar, Buyya Rajkumar. Provisioning spot market cloud resources to create cost-effective virtual clusters. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* 2011;7016 LNCS(PART 1):395–408, available at 1110.5972.

73. Walker E. Benchmarking amazon EC2 for high-performance scientific computing. *Usenix Login.* 2008:18–23.

74. Wang Lizhe, Tao Jie, Kunze Marcel, Castellanos Alvaro Canales, Kramer David, Karl Wolfgang. (). *Scientific Cloud Computing: Early Definition and Experience*, 2008 10th ieee international conference on high performance computing and communications, pp. 825–830.

75. Yelick Katherine, Coghlan Susan, Draney Brent, Ramakrishnan Lavanya, Scovel Adam, Sakrejda Iwona, Liu Anping, Campbell Scott, Zbiegiel Piotr T, Declerck Tina, Rich Paul, Wright Nicholas J, Winkler Linda, Mitchell Nathan M, Guantonio Michael a, Lester Levi J, West Gabriel a, Skinner David, Lu Wei, Pershey Eric R. *The Magellan Report on Cloud Computing for Science*: U.S. Department of Energy Office of Science Office of Advanced Scientific Computing Research (ASCR); 2011

□