An Open Source Solution for Virtual Infrastructure Management in Private and Hybrid Clouds

Borja Sotomayor, Rubén S. Montero, Ignacio M. Llorente, and Ian Foster

Abstract

More often than not, a "cloud" refers to an "Infrastructure-as-a-Service" (IaaS) cloud, such as Amazon EC2, where IT infrastructure is deployed in a cloud provider's datacenter in the form of virtual machines. With the growing popularity of IaaS clouds, an ecosystem of tools and technologies is emerging that can transform an organization's existing infrastructure into a *private cloud* or a *hybrid cloud*. In this article, we present OpenNebula, an open source virtual infrastructure manager that can be used to deploy virtualized services on both a local pool of resources and on external IaaS clouds, and Haizea, a resource lease manager that can act as a scheduling backend for OpenNebula providing features not found in other cloud software or virtualization-based datacenter management software, such as advance reservations and resource preemption, which we argue to be specially relevant for private and hybrid clouds.

Index Terms

C.0.a Emerging technologies, D.4.7.b Distributed systems

Cloud Computing is, to use yet another cloud-inspired pun, a nebulously defined term. However, it was arguably first popularized in 2006 by Amazon's Elastic Compute Cloud (or EC2, see http://www.amazon.com/ec2/), which started offering virtual machines (VMs) for $0.10/hour using both a simple web interface and a programmer-friendly API. Although not the first to propose a utility computing model, Amazon EC2 contributed to popularizing the "Infrastructure as a Service" (IaaS) paradigm, which became closely tied to the notion of Cloud Computing. An *IaaS cloud* enables on-demand provisioning of computational resources, in the form of VMs deployed in a cloud provider's datacenter (such as Amazon's), minimizing or even eliminating associated capital costs for cloud consumers, allowing capacity to be added or removed from their IT infrastructure in order to meet peak or fluctuating service demands, while only paying for the actual capacity used.

Over time, an ecosystem of providers, users, and technologies has coalesced around this IaaS cloud model. More IaaS cloud providers, such as GoGrid, FlexiScale, and ElasticHosts have emerged. A growing number of

Borja Sotomayor is with the University of Chicago (Chicago, IL, USA)
Rubén S. Montero and Ignacio M. Llorente are with the Facultad de Informática of the Universidad Complutense de Madrid (Madrid, Spain)
Ian Foster is with the University of Chicago (Chicago, IL, USA) and Argonne National Laboratory (Argonne, IL, USA)

companies base their IT strategy on cloud-based resources, spending little or no capital to manage their own IT infrastructure (see http://aws.amazon.com/solutions/case-studies/ for several examples). Some providers, such as Elastra and Rightscale, focus on deploying and managing services on top of IaaS clouds, including web and database servers that benefit from the elastic capacity of IaaS clouds, allowing their clients to provision services directly, instead of having to provision and setup the infrastructure themselves. Other providers offer products that facilitate working with IaaS clouds, such as rPath's rBuilder (http://www.rpath.org/), which allows dynamic creation of software environments to run on a cloud.

Although this ecosystem has evolved around *public clouds* —commercial cloud providers that offer a publicly-accessible remote interface to create and manage virtual machine instances within their proprietary infrastructure—there is also a growing interest in open-source Cloud Computing tools that allow organizations to build their own IaaS clouds using their internal infrastructure. The primary aim of these *private cloud* deployments is not to sell capacity over the Internet through publicly-accessible interfaces, but to provide local users with a flexible and agile private infrastructure to run service workloads within their administrative domain. Private clouds can also support a *hybrid cloud* model by supplementing local infrastructure with computing capacity from an external public cloud. Private and hybrid clouds are not exclusive with being a public cloud; a private/hybrid cloud can allow remote access to its resources over the Internet using remote interfaces, such as the web services interfaces used in Amazon EC2.

However, to provide the same features found in commercial clouds, private/hybrid cloud software must meet a variety of requirements: provide a uniform and homogeneous view of virtualized resources, regardless of the underlying virtualization platform (e.g., Xen, KVM, VMWare, etc.); manage the full lifecycle of a virtual machine, including setting up networks dynamically for groups of VMs and managing the storage requirements of VMs, such as deployment of VM disk images or on-the-fly creation of software environments; support for configurable resource allocation policies to meet the specific goals of the organization (e.g., high availability, server consolidation to minimize power usage, etc.); and adaptability to an organization's changing resource needs, including peaks where local resources are insufficient, and changing resources, including addition or failure of physical resources.

Thus, a key component in private/hybrid clouds will be *virtual infrastructure (VI) management*, the dynamic orchestration of virtual machines on a pool of physical resources, meeting the requirements outlined above. In this article, we discuss the relevance of VI management not just for the creation of private/hybrid clouds, but also within the emerging cloud ecosystem. We present OpenNebula [1] (http://www.opennebula.org/) and Haizea [2] (http://haizea.cs.uchicago.edu/), two complementary open source projects that can be used to manage virtual infrastructures in private/hybrid clouds. OpenNebula is a virtual infrastructure manager that can be used to deploy and manage virtual machines, either individually or in groups that must be co-scheduled, on local resources or on external public clouds, automating the setup of the virtual machines (preparing disk images, setting up networking, etc.) regardless of the underlying virtualization layer (Xen, KVM, or VMWare are currently supported) or external cloud (EC2 or ElasticHosts are currently supported). Haizea is a resource lease manager that can act as a scheduling backend for OpenNebula, providing leasing capabilities not found in other cloud systems, such as advance reservations and

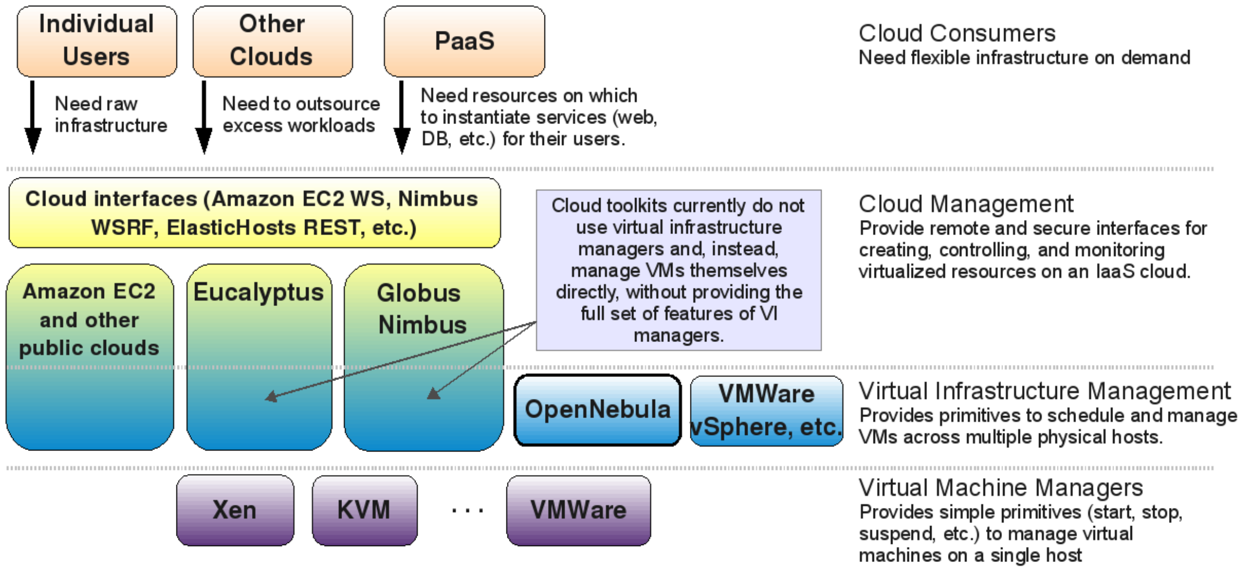

Fig. 1. The Cloud ecosystem for building private clouds

resource preemption, and which we argue to be specially relevant for private clouds.

I. THE CLOUD ECOSYSTEM

Virtual infrastructure management tools for datacenters have been around since before Cloud Computing became the industry's new buzzword. Several of these, such as Platform VM Orchestrator (http://www.platform.com/Products/platform-vm-orchestrator), and VMware vSphere (www.vmware.com/products/vsphere/), and Ovirt (http://ovirt.org/), meet many of the requirements for VI management outlined earlier, providing features such as dynamic placement and management of virtual machines on a pool of physical resources, automatic load balancing, server consolidation, and dynamic resizing and partitioning of infrastructure. Thus, although creating what is now called a "private cloud" was already possible with existing tools, these tools lack other features that are relevant for building IaaS clouds, such as public cloud-like interfaces, mechanisms to add such interfaces easily, or the ability to deploy VMs on external clouds.

On the other hand, projects like Globus Nimbus [3] (http://workspace.globus.org/) and Eucalyptus [4] (http://www.eucalyptus.com/), which we term *cloud toolkits*, can be used to transform existing infrastructure into an IaaS cloud with cloud-like interfaces. Eucalyptus is compatible with Amazon's EC2 interface and is designed to support additional client-side interfaces. Globus Nimbus exposes EC2 and WSRF interfaces and offers self-configuring virtual cluster support. However, although these tools are fully functional with respect to providing cloud-like interfaces and higher-level functionality for security, contextualization and VM disk image management, their VI management capabilities are limited and lack the features of solutions that specialize in VI management.

Thus, an ecosystem of cloud tools is starting to form (see Figure 1) where cloud toolkits attempt to span both cloud management and VI management but, by focusing on the former, do not deliver the same functionality of software written specifically for VI management. Although integrating cloud management solutions with existing VI managers would seem like the obvious solution, this is complicated by the lack of open and standard interfaces between the two layers, and the lack of certain key features in existing VI managers (enumerated below). The focus of our work is, therefore, to produce a VI management solution with a flexible and open architecture that can be used to build private/hybrid clouds.

With this goal in mind, we started developing OpenNebula and continue to enhance it as part of the European Union's RESERVOIR Project (http://www.reservoir-fp7.eu/), which aims to develop open source technologies to enable deployment and management of complex IT services across different administrative domains. OpenNebula provides much of the functionality found in existing VI managers, but also aims to overcome the shortcomings in other VI solutions. Namely, (i) the inability to scale to external clouds, (ii) monolithic and closed architectures that are hard to extend or interface with other software, not allowing its seamless integration with existing storage and network management solutions deployed in datacenters, (iii) a limited choice of preconfigured placement policies (first fit, round robin, etc.), and (iv) lack of support for scheduling, deploying, and configuring groups of VMs (e.g., a group of VMs representing a cluster, which must all be deployed, or not at all, and where the configuration of some VMs depends on the configuration of others, such as the head-worker relationship in compute clusters). Table I provides a more detailed comparison between OpenNebula and several well-known VI managers, including cloud toolkits that perform VI management.

A key feature of OpenNebula's architecture, described in more detail in the next section, is its highly modular design, which facilitates integration with any virtualization platform and third-party component in the cloud ecosystem, such as cloud toolkits, virtual image managers, service managers, and VM schedulers. For example, all the actions pertaining to setting up a VM disk image (transferring the disk image, installing software on it, etc.) are specified in terms of well-defined hooks; although OpenNebula includes a default "transfer manager" that uses these hooks, it is possible to leverage existing transfer managers or VM image contextualizers just by writing code that interfaces between the hooks and the third-party software.

The Haizea project, described in detail in Section III and developed independently from OpenNebula, was the first project to leverage such an architecture in a way that was beneficial to both projects. Haizea, an open-source VM-based lease manager which originally could only simulate VM scheduling for research purposes, was modified to act as a drop-in replacement for OpenNebula's default scheduler, with few modifications required in the Haizea code and none in the OpenNebula code. By working together, OpenNebula was able to offer resource leases, such as advance reservation leases, as a fundamental provisioning abstraction, and Haizea was able to operate with real hardware through OpenNebula.

In fact, the integration of OpenNebula and Haizea provides the only VI management solution offering advance reservation of capacity. As shown in Table I, other VI managers use immediate provisioning, where VMs must be allocated right away or not at all, or best-effort provisioning, where VMs are queued until resources can be allocated. However, private clouds, specially those with limited resources where not all requests may be satisfiable immediately for lack of resources, stand to benefit from more sophisticated VM placement strategies supporting queues, priorities, and advance reservations. Additionally, Service provisioning clouds, such as the one being developed in the RESERVOIR project, have requirements that cannot be supported only with an immediate provisioning model, including the need for capacity reservations at specific times to meet service-level agreements or peak capacity requirements.

Other integration efforts with OpenNebula are currently underway (see http://opennebula.org/doku.php?id=ecosystem), including an implementation of the libvirt interface (http://libvirt.org/) and a VM consolidation scheduler designed to minimize energy consumption. The RESERVOIR project is also developing other tools around OpenNebula for service elasticity management, VM placement to meet SLA commitments, support for public cloud interfaces, and a VM scheduler (termed "policy engine" within the project) that adds support for policy-driven probabilistic admission control and dynamic placement optimization to satisfy site-level management policies. There has also been an experimental integration of OpenNebula with Globus Nimbus.

II. The OpenNebula Architecture

The OpenNebula architecture, depicted in figure 2, encompasses several components specialized in different aspects of virtual infrastructure management.

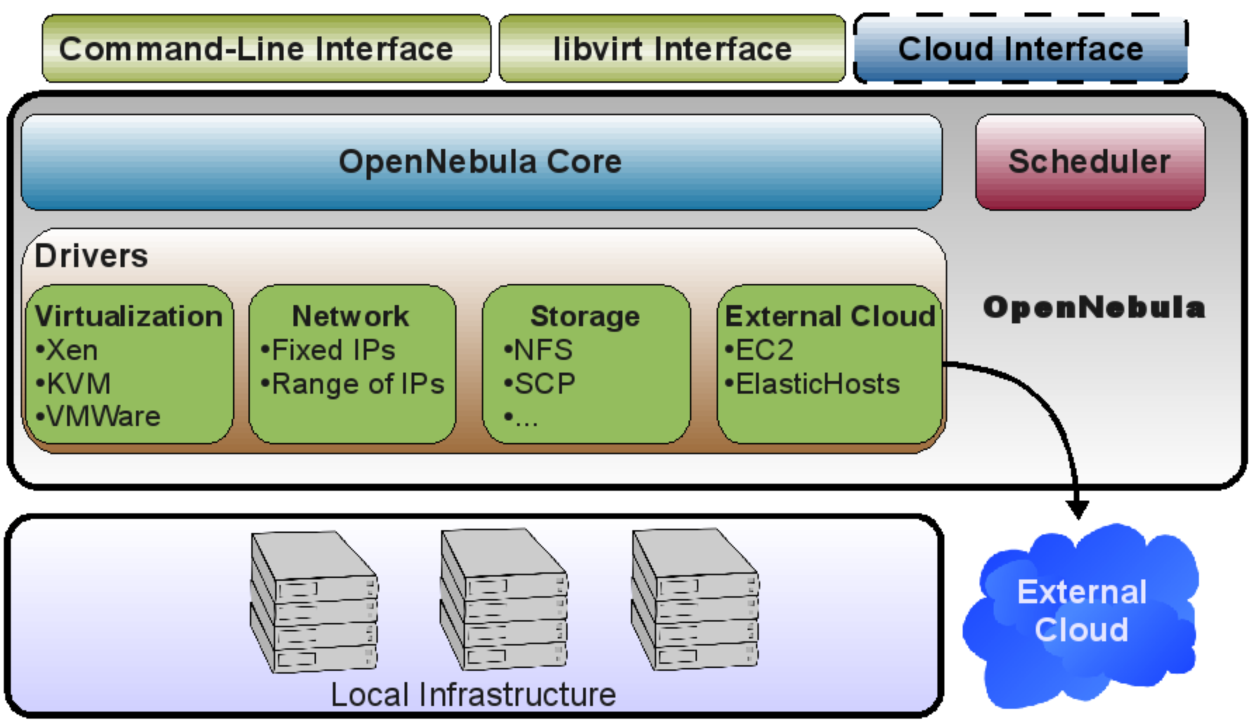

Fig. 2. OpenNebula virtual infrastructure engine components and its integration with an external cloud provider.

In order to control the lifecycle of a VM, the OpenNebula *Core* orchestrates three different management areas, namely: (i) image and storage technologies (e.g. virtual appliance tools or distributed file systems) to prepare disk images for VMs; (ii) the network fabric (e.g. DHCP servers, firewalls or switches) to provide the VMs with a

TABLE I
COMPARISON OF TOOLS PROVIDING VIRTUAL INFRASTRUCTURE MANAGEMENT CAPABILITIES

Tool	**Provisioning Model**	**Default Placement Policies**	**Configurable Placement Policies**	**Support for Hybrid Cloud**	**Remote interfaces**
Amazon EC2	Best-effort	Proprietary	Proprietary	No	EC2 WS API
VMWare vSphere	Immediate	Initial placement on CPU load, and dynamic placement to balance average CPU or memory load and to consolidate servers	No	Only when both the local and external cloud use vSphere	vCloud API
Platform Orchestrator	Immediate	Initial placement on CPU load and migration policies based on policy thresholds on CPU utilization level	No	No	No
Nimbus	Immediate	Static greedy and round robin resource selection	No	Includes a “EC2 backend” that can forward requests to EC2, but local and remote resources have to be managed separately.	EC2 WS API and Nimbus WSRF
Eucalyptus	Immediate	Static greedy and round robin resource selection	No	No	EC2 WS API
oVirt	Immediate	Manual mode	No	No	No
OpenNebula 1.2	Best-effort	Initial placement based on a requirement/rank policies to prioritize those resources more suitable for the VM using dynamic information, and dynamic placement to consolidate servers	Support for any static/dynamic placement policy	Driver-based architecture allows interfacing with multiple external clouds. EC2-compatible clouds and ElasticHosts are currently supported.	No
OpenNebula 1.2/ Haizea	Immediate, Best-effort and Advance reservation	Dynamic placement to implement advance reservation leases	VM placement strategies supporting queues and priorities		
OpenNebula 1.2/ RESERVOIR	Immediate and Best-effort	Load balancing and power saving policies	Support for policy-driven probabilistic admission control and dynamic placement optimization to satisfy site-level management policies		RESERVOIR VMI

virtual network environment; and (iii) the underlying hypervisors to create and control VMs. The Core performs specific storage, network or virtualization operation through pluggable *Drivers*. Thus, OpenNebula is not tied to any specific environment, providing a uniform management layer regardless of the underlying infrastructure.

Besides managing the lifecycle of individual VMs, the Core is also designed to support deployment of *services*, which typically include of a set of interrelated components (e.g. web server, DB backend, etc.) requiring several VMs. Thus, a group of related VMs can be treated as a first-class entity in OpenNebula. Besides managing the VMs as a unit, the Core also handles delivery of context information (e.g. the IP address of the web server, digital certificates, software licenses, etc.) to the VMs.

Virtual Machine placement decisions are made by a separate *Scheduler* component. More specifically, the scheduler has access to information on all requests received by OpenNebula and, based on them, is responsible for keeping track of current and future allocations, creating and updating a resource schedule, and sending the appropriate deployment commands to the OpenNebula *Core*. The OpenNebula default scheduler provides a rank scheduling policy that places VMs on physical resources according to a ranking algorithm that is highly configurable by the administrator, and relies on real-time data from both the running VMs and available physical resources. The Haizea lease manager, described in the next section, can be used in place of the default scheduler.

OpenNebula offers *Management Interfaces* to integrate the *Core* functionality within other datacenter management tools, such as accounting or monitoring frameworks. To this end, OpenNebula implements the libvirt API, an open interface for VM management, and a command line interface (CLI). Also a subset of this functionality can be exposed to external users through a Cloud Interface.

Finally, OpenNebula can support a hybrid cloud model by using *Cloud Drivers* to interface with external clouds. In this way, the local infrastructure can be supplemented with computing capacity from a public cloud to meet peak demands, to better serve user access requests (e.g. by moving the service closer to the user), or to implement high availability strategies. OpenNebula currently includes an EC2 driver, which can submit requests to Amazon EC2 and Eucalyptus, and an ElasticHosts driver.

III. The Haizea Lease Manager

Haizea is an open-source resource lease manager, and can act as a VM scheduler for OpenNebula or used on its own as a simulator to evaluate the performance of different scheduling strategies over time. The fundamental resource provisioning abstraction in Haizea is the *lease*. Intuitively, a lease is some form of contract where one party agrees to provide a set of resources (an apartment, a car, etc.) to another party. When a user wants to request computational resources from Haizea, it does so in the form of a lease, which are then implemented as VMs managed by OpenNebula. The lease terms supported by Haizea include the hardware resources, software environments, and the availability period during which the hardware and software resources must be available. Currently, Haizea supports *advance reservation leases*, where the resources must be available at a specific time; *best-effort leases*, where resources are provisioned as soon as possible, and requests are placed on a queue if necessary; and *immediate leases*, where resources are provisioned when requested, or not at all.

Advance reservation of computational resources, one of the salient features in OpenNebula and Haizea, has been previously studied in the context of parallel computing [5], [6], [7] and, in the absence of suspension/resumption capabilities, is known to produce resource underutilization due to the need to vacate resources before an advance reservation (AR) starts. By using virtual machines to implement leases, ARs can be supported more efficiently [8], [2] through resource preemption, suspending the VMs of lower-priority leases before a reservation starts, resuming them after the reservation ends, and potentially migrating them to other available nodes or even other clouds.

Although resource preemption has also been studied in the context of parallel computing, VMs have the attractive quality of allowing computation to be suspended without having to make the applications inside the VM aware that they are going to be suspended, resumed, or even migrated. However, using VMs introduces overheads that pose additional scheduling challenges. More specifically, the *preparation overhead* of deploying the VM disk images needed by the lease can have a noticeable impact on performance if not adequately managed [9]. Haizea's approach is to separately schedule this overhead, instead of assuming it should just be deducted from a user's allocation. However, this is complicated by having to support multiple types of leases with conflicting requirements that must be reconciled; the transfers for a lease starting at 2pm could require delaying transfers for best-effort leases, resulting in longer waiting times. Haizea uses a number of optimizations, such as reusing disk images across leases, to minimize the impact of preparation overhead. Similarly, Haizea also schedules *runtime overhead*, such as VM suspensions, resumptions and migrations, which may also be required to finish before a specific time.

Haizea bases its scheduling on a resource slot table that represents all the physical nodes managed by Haizea over time. Best-effort leases are scheduled using a First-Come-First-Serve queue with backfilling (a common optimization in queue-based systems), while AR leases use a greedy algorithm to select physical resources that minimize the number of preemptions. Although the resource selection algorithm is currently hardcoded, future versions will include a policy decision module to allow developers to specify their own resource selection policies (e.g., policies to prioritize leases based on user, group, project, etc.). This policy decision module will also be used to specify the conditions under which a lease should be accepted or rejected.

IV. EXPERIENCES WITH OPENNEBULA AND HAIZEA

Although OpenNebula and Haizea both originated in research projects, one of our goals is to produce production-quality releases that meet the needs of other communities. In fact, we feel strongly about using a development model that, first and foremost, produces stable software, suitable for production environments, which we can also use for our own research, incorporating the results of our research into the next stable version. This allows us to support the requirements of virtual infrastructure users, while incorporating novel techniques and solutions into our releases. OpenNebula has already seen several stable releases, and has a growing user base through its inclusion in the popular Ubuntu GNU/Linux distribution (http://www.ubuntu.com/) starting with Ubuntu 9.04 ("Jaunty Jackalope"). Our first-hand experiences with OpenNebula have mostly taken place in the EU RESERVOIR project, where OpenNebula is being enhanced to meet the requirements of several business use cases [10]. In recent work, we have shown OpenNebula to be an effective tool for managing clustered services, using it to deploy and manage the backend nodes of a Sun Grid Engine compute cluster [1] and of a NGINX web server [11] on both local resources and an external cloud.

Haizea is still in a “technology preview” stage, although a first stable release is planned later in 2009. In previous joint work [2] with K. Keahey (Argonne National Laboratory), we used Haizea to simulate 72 30-day workloads in six different configurations, or 36 years of lease scheduling, producing experimental results showing that, when using workloads that combine best-effort and advance reservation requests, a VM-based approach with suspend/resume can overcome the utilization problems typically associated with the use of advance reservations. More specifically, when measuring the total time required to process all the requests in the workload, we found that a VM-based approach performed consistently better (up to 32.97%), despite the overhead of using VMs. Our results also showed that the preparation overhead of VMs, in the form of transferring VM disk images from a repository, can be minimized through the use of image transfer scheduling and caching strategies. In more recent work [12], [13], we have used OpenNebula and Haizea together to perform experiments on a physical testbed and develop a resource model for predicting the runtime overhead of suspending/resuming VMs under a variety of configurations. We found that, similarly to scheduling preparation overhead, explicitly and separately scheduling suspensions and resumptions is necessary to avoid unnecessary delays in leases (e.g., if a lease must be suspended to make way for a higher-priority lease, like an advance reservation). Furthermore, we found that accurately estimating the time to suspend and resume leases depends on a variety of factors that must be taken into account when scheduling leases.

V. Conclusion

As interest in private and hybrid IaaS clouds grows, so will the need for a diverse ecosystem of tools and technologies that can be used as building blocks to create and manage these clouds. Although some solutions have emerged across three broad categories –cloud management, virtual infrastructure management, and virtual machine management–, the challenge ahead will be integrating multiple components to create complete IaaS cloud-building solutions.

Private and hybrid clouds will also face the challenge of efficiently managing a finite amount of resources. However, existing virtual infrastructure managers rely on an immediate resource provisioning that implicitly assumes that capacity is practically infinite. While this is a fair assumption for large cloud providers, such as Amazon EC2 which frequently refers to a cloud as having “infinite capacity”, it is not applicable to smaller providers where the likelihood of being overloaded is greater. In order to satisfy service-level agreements, requests for resources will inevitably have to be prioritized, queued, pre-reserved, deployed on external clouds, or even rejected, and VI management solutions with these capabilities will be required.

Here we have presented OpenNebula and Haizea, two open source projects that address these two challenges. By relying on a flexible, open, and loosely coupled architecture, OpenNebula is designed from the outset to be easy to integrate with other components, such as the Haizea lease manager. When used together, OpenNebula and Haizea are the only virtual infrastructure management solution that provide leasing capabilities beyond immediate provisioning, including best-effort leases and advance reservation of capacity.

ACKNOWLEDGMENTS

We gratefully acknowledge the hard work of the OpenNebula developers: Javier Fontán and Tino Vázquez. We also thanks our anonymous reviewers for their insightful and detailed comments. Development of OpenNebula is supported by Consejería de Educación de la Comunidad de Madrid, Fondo Europeo de Desarrollo Regional (FEDER) and Fondo Social Europeo (FSE), through BIOGRIDNET Research Program S-0505/TIC/000101, by Ministerio de Educación y Ciencia, and through the research grant TIN2006-02806, and by the European Union through the research grant RESERVOIR Grant Number 215605. Development of Haizea is supported by RESERVOIR, the University of Chicago, and the U.S. Department of Energy under Contract DE-AC02-06CH11357. Early work on Haizea was done in collaboration with K. Keahey (Argonne National Laboratory) and funded by NSF grant #509408 "Virtual Playgrounds".

REFERENCES

[1] I. Llorente, R. Moreno-Vozmediano, and R. Montero, "Cloud computing for on-demand grid resource provisioning," To appear in *Advances in Parallel Computing*, 2009.

[2] B. Sotomayor, K. Keahey, and I. Foster, "Combining batch execution and leasing using virtual machines," in *HPDC '08: Proceedings of the 17th International Symposium on High Performance Distributed Computing*. ACM, 2008, pp. 87–96.

[3] K. Keahey, I. Foster, T. Freeman, and X. Zhang, "Virtual workspaces: Achieving quality of service and quality of life on the grid," *Scientific Programming*, vol. 13, no. 4, pp. 265–276, 2005.

[4] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The eucalyptus open-source cloud-computing system," in *Cloud Computing and Applications 2008 (CCA08)*, 2008.

[5] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, and A. Roy, "A distributed resource management architecture that supports advance reservations and co-allocation," in *Proceedings of the International Workshop on Quality of Service*, 1999.

[6] W. Smith, I. Foster, and V. Taylor, "Scheduling with advanced reservations," in *IPDPS '00: Proceedings of the 14th International Symposium on Parallel and Distributed Processing*. IEEE Computer Society, 2000, p. 127.

[7] Q. Snell, M. J. Clement, D. B. Jackson, and C. Gregory, "The performance impact of advance reservation meta-scheduling," in *IPDPS '00/JSSPP '00: Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing*. London, UK: Springer-Verlag, 2000, pp. 137–153.

[8] B. Sotomayor, K. Keahey, I. Foster, and T. Freeman, "Enabling cost-effective resource leases with virtual machines," in *Hot Topics session in ACM/IEEE International Symposium on High Performance Distributed Computing 2007 (HPDC 2007)*, 2007.

[9] B. Sotomayor, K. Keahey, and I. Foster, "Overhead matters: A model for virtual resource management," in *VTDC '06: Proceedings of the 1st International Workshop on Virtualization Technology in Distributed Computing*. IEEE Computer Society, 2006, p. 5.

[10] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Caceres, M. Ben-Yehuda, W. Emmerich, and F. Galán, "The reservoir model and architecture for open federated cloud computing," *IBM Systems Journal*, October 2008.

[11] R.Moreno, R. Montero, and I. Llorente, "Elastic management of cluster-based services in the cloud," To appear in *Proceedings of the First Workshop on Automated Control for Datacenters and Clouds (ACDC 2009)*, June 2009.

[12] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Capacity leasing in cloud systems using the opennebula engine," in *Cloud Computing and Applications 2008 (CCA08)*, 2008.

[13] B. Sotomayor, R. Montero, I. Llorente, and I. Foster, "Resource leasing and the art of suspending virtual machines," To appear in *Proceedings of the The 11th IEEE International Conference onHigh Performance Computing and Communications (HPCC-09)*, June 2009.

BIOGRAPHIES

Borja Sotomayor is a PhD candidate in the Department of Computer Science at the University of Chicago. His research interests include resource provisioning and scheduling, distributed systems, and virtualization. Sotomayor has a M.Sc. degree from the University of Chicago, and a Computer Engineering degree from the University of Deusto (Bilbao, Spain). Contact him at borja@cs.uchicago.edu.

Rubén Santiago Montero is an Associate Professor in the Department of Computer Architecture at the Complutense University of Madrid. His research interests lie mainly in resource provisioning models for distributed systems, in particular: Grid resource management and scheduling, distributed management of virtual machines and cloud computing. Rubén has a PhD in physics (computer science program) from Complutense University. Contact him at rubensm@dacya.ucm.es

Ignacio Martín Llorente is a Full Professor and the Head of the Distributed Systems Architecture Research group at the Complutense University of Madrid. His research interests include advanced distributed computing and virtualization technologies, architecture of large-scale distributed infrastructures and resource provisioning platforms. Ignacio has a PhD in Computer Science and a Executive Master in Business Administration. Contact him at llorente@dacya.ucm.es.

Ian Foster is director of the Computation Institute at the University of Chicago and Argonne National Laboratory and the Arthur Holly Compton Distinguished Service Professor of computer science at the University of Chicago. His research interests include distributed computing, parallel computing, and computational science. Foster has a PhD in computer science from Imperial College, London. Contact him at foster@anl.gov.