RESERVOIR – When one cloud is not enough

Benny Rochwerger, Johan Tordsson, Carmelo Ragusa, David Breitgand, Stuart Clayman, Amir Epstein, David Hadas, Eliezer Levy, Irit Loy, Alessandro Maraschini, Philippe Massonet, Henar Muñoz, Kenneth Nagin, Giovanni Toffetti, and Massimo Villari

◆

Abstract—As cloud computing becomes more predominant, the problem of scalability has become critical for cloud computing providers. The cloud paradigm is attractive because it offers a dramatic reduction in capital and operation expenses for consumers. But as the demand for cloud services increases, the ensuing increases in cost and complexity for the cloud provider may become unbearable. We briefly discuss the technologies we developed under the RESERVOIR European research project to help cloud providers deal with complexity and scalability issues. We also introduce the notion of a federated cloud that would consist of several cloud providers joined by mutual collaboration agreements. A federated cloud can deal with scalability problems in a cost-effective manner. Providers in the federation who have excess capacity can share their infrastructure with members in need of additional resources.

1 INTRODUCTION

Cloud computing is essentially the latest incarnation of the utility computing model envisioned back in the '60s [1]. Just as we power a wide variety of devices from an electric utility hiding beyond the wall plug, individuals and organizations can now fulfill most of their computing needs from a computing utility hidden in the network [2].

Cloud computing's analogy to an electrical power grid does not end with the consumption model. Power generation plants are built to support a certain maximum capacity, which is determined by analyzing the average utilization and then over-provisioning for predicted spikes. Demand that exceeds this maximum capacity is delegated to neighboring providers. Similarly, cloud computing providers can handle requests that exceed their own capacity by delegating them to other cloud computing providers. This is a flexible and cost-efficient alternative to over-provisioning.

Grid computing [3], an earlier incarnation of the utility computing model, was driven by need for more compute power. Cloud computing, on the other hand, is driven by the need of companies and individuals to deal with the ever increasing cost and complexity of IT services. Whether you are a company looking to outsource all, or some, of your IT, or a start-up with a great idea but little funding, or you have a big one-time project that needs resources, cloud computing offers a seemingly infinite pool of resources, without any capital expenses or system administration overhead.

In grid computing, resource sharing is a goal by definition – scientific centers share their infrastructure with one another to achieve additional compute power. The need for resource sharing in the case of cloud computing, is not yet that clear. But we believe that as cloud computing becomes mainstream practice, providers will choose to support a federated model driven purely by business goals, i.e., be only as big as needed to be profitable, and rely on others when more resources are necessary.

RESERVOIR [4] is an European research initiative whose primary goal is to develop the technologies needed to deal with the scalability problem inherent in the single provider cloud computing model. RESERVOIR explores the notion of a *federated cloud* in which computing infrastructure providers with excess capacity lease it to provider in need of temporary additional resources.

The rest of this paper is organized as follows: In Section 2, we introduce a model for federated cloud computing and describe the challenges this model presents. In Section 3, we describe the experimental testbed built as part of the RESERVOIR project and in Section 4, we describe the technologies we developed to support advanced cloud management. Section **??** focuses on technologies we developed to meet the challenges. In Section 5, we present one of the applications we used for the validation of the model, and finally, we present our view of the future direction of cloud computing in Section 6.

2 THE FEDERATED CLOUD MODEL AND ITS CHALLENGES

In the RESERVOIR model for federated cloud computing, two or more independent cloud computing providers can join together to create a *federated cloud*. Participants of the federation who have excess capacity can share their resources, for an agreed-upon price, with those participants in need of additional resources. This sharing and paying model helps individual providers

---

The research leading to these results is partially supported by the European Community's Seventh Framework Programme (FP7/2001-2013) under grant agreement n° 215605.

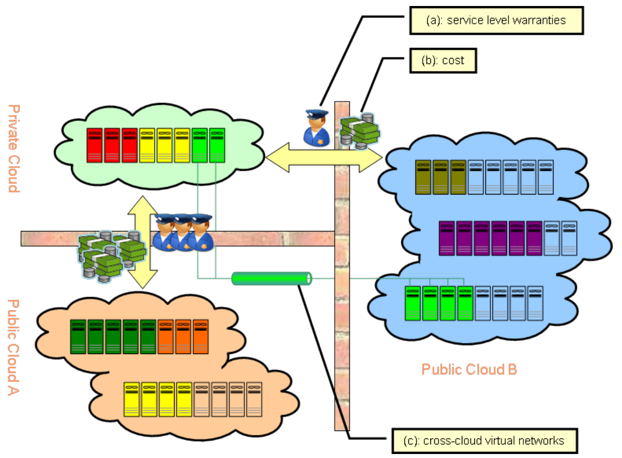

Fig. 1. Challenges in the federated cloud computing model: Finding the "best" cloud for any particular workload requires careful balancing among many parameters, such as (a) quality of service warranties and (b) cost. (c) Consistent behavior of applications regardless of the location of the different components (VMs).

reduce the need for over-provisioning of resources to deal with spikes in capacity demand.

In a multi-cloud environment, a system that makes automated decisions needs to address federated placement. Federated placement refers to the process of determining which cloud to use for a particular workload, given the fact that not all clouds are equal in terms of warranties and cost. A particular cloud could be inexpensive, for example, but might not provide availability warranties, making it inappropriate for mission-critical workloads. Another cloud, however, might guarantee five nines availability but be more expensive (see Figure 1(a,b)).

To support maximum optimization (from the infrastructure point of view), applications need to be completely location-free. In other words, the different components of the application (encapsulated in virtual machines) should be able to be deployed anywhere in the federated cloud. Moreover, the application components must be able to be migrated at any time, even across clouds. The development of location- and topology-independent technology is necessary to support this model. Such technology must (i) support inter-component communication and (ii) offer the virtual machines' (VMs) consistent access to data (see Figure 1(c)). On the other hand, this technology should enable limitations on its flexibility to support application correctness and compliance with government regulations and company policies.

For consumers, one of the main advantages of cloud computing is the capability to provide, or release, resources on-demand. These "elasticity" capabilities should be enacted automatically by cloud computing providers to meet demand variations. Clearly the behavior and limits of automatic growth and shrinking should be driven by contracts and rules agreed on between cloud computing providers and consumers. The ability of users to grow their applications when facing an increase of real-life demand need to be complemented by the provider's ability to scale and to overcommit resources.

Cloud infrastructures are subject to the same threats as other distributed systems. The security requirements for the cloud user are related to the ability to select or associate different security policies with cloud service deployments and the ability to monitor these policies. The security requirements for the cloud providers necessitate isolating the customer deployments at both the virtual and physical infrastructure levels. Isolation of the virtual infrastructure not only includes virtual machine isolation, but also isolation of virtual networks and virtualized storage. The ability to guarantee that services are provisioned only in clouds with the appropriate level of security policies is an important security requirement for federated clouds. Other important requirements are related to topics such as data location, regulatory compliance, recovery, investigative support, and long-term viability of cloud deployments.

3 THE RESERVOIR TESTBED

In order to develop, experiment, and gain insight into the possible roadblocks of the federated cloud model, we built a multi-cloud environment that aggregates resources from the University of Messina in Italy, a Thales Group site in France, and the Umeå University in Sweden, as follows:

- In Messina, we have 15 bi-processor dual-core machines, each with 8GB of RAM
- In Thales, we have 12 bi-processor quad-core machines, each with 4GB of RAM
- In Umeå, we have 3 quad-core machines, each with 8GB of RAM

All machines use KVM [5] as the hypervisor and the RESERVOIR cloud management middleware, which for maximum flexibility and ease of administration, is itself packaged in a self-contained virtual machine.

Although relatively small when compared with production clouds, this setup provides an excellent environment for experimentation with the issues of cloud federation, since these clouds are geographically distant, and are owned and managed by entirely different and independent organizations.

4 FEDERATED CLOUD MANAGEMENT SOLUTIONS

The main of the RESERVOIR project is to research and develop advanced technologies so that cloud infrastructure providers can efficiently run their businesses and provide value to their customers. In this section we briefly describe several of these technologies.

4.1 Dynamic service elasticity

The ability to dynamically scale a service up and down is key in cloud computing, since it enables the cloud computing user to avoid over-provisioning, while still being able to automatically adjust to changing loads. In RESERVOIR, the scaling process is automated through *elasticity rules* [6], a mechanism used to specify the dynamic capacity requirements of an application at deployment time. Elasticity rules follow the event-condition-action approach, in which automated actions to change the service capacity are triggered when certain conditions arise. Such capacity changes include (i) scaling up, i.e., resizing a running component of the service, such as increasing or decreasing the allocated memory of a component, or (ii) scaling out, i.e., adding or removing instances of service components.

Dynamic elasticity can be specified explicitly by adding elasticity rules to the *Service Manifest* – a descriptor of the service, based on the Open Virtualization Format (OVF) standard [7]. Such an approach benefits from the inherent knowledge providers have of their applications. For example, for the SAP application presented in Section 5, we defined a Key Performance Indicator (KPI) as the total number of active sessions currently served by the SAP system. We also defined an elasticity rule specifying that when this KPI exceeds a threshold, a new VM is automatically started.

An alternative to using explicit rules is implicit SLA protection. In this case, instead of elasticity rules the Service Manifest includes a section on performance objectives (e.g., response time must be below 50 milliseconds for 90% of the time for a 10-minute window). These objectives are coupled with a control strategy (e.g., minimizing the number of VMs within SLO boundaries).

We developed an engine that constructs an approximate model of the system response for each performance objective as a function of the service configuration in terms of VM instances, input workload, and other relevant KPIs. This model is updated in a continuous fashion to obtain autonomic control [8]. An approximate model is constructed in two steps: First, at service staging time, we provide artificial workloads to different system configurations and measure the system response. Second, at runtime, the engine controls the system configuration through the approximate models, enriching them with additional information as new combinations of workload, KPIs, and configurations are investigated. This continuous learning process is a key feature of the autonomic controller.

As part of the RESERVOIR validation, we ran several experiments to assess the system responses of different services under different working conditions. Figure 2 shows an example of a Kriging [8] model of a composite service throughput as a function of its incoming workload, the number of VM instances for a service composition engine (labelled SCE), and its application servers (labelled AS) of one of the atomic services.

4.2 Admission control

As the number of deployed services goes down, the probability that all elastic services will simultaneously request resources up to the maximal contracted capacity range diminishes. Moreover, as a system grows in size, the variance of total resource demand in the system becomes smaller. Drawing our inspiration from results in network bandwidth multiplexing, we defined a notion of equivalent physical capacity required to host the given mix of elastic services while keeping the probability of resource allocation congestion below the *acceptable risk level (ARL)*. This risk level is set by the infrastructure provider in accordance with its business goals. A conservative approach would set the ARL at the level of the strictest SLA availability percentile. As long as there is enough physical capacity to place equivalent capacity, a system will honor its SLA for all services, while efficiently multiplexing physical resources.

In RESERVOIR, we enhanced cloud management functionality with admission control. Admission control continuously calculates the anonymized equivalent capacity based on the statistics gathered for the service portfolio. When a new service is accepted into the cloud, RESERVOIR's admission control policy calculates its impact on the equivalent capacity. The policy assumes a pessimistic estimation of resource usage for the new service—namely that it would use its maximal resource allocation as specified in the service manifest. The new service is accepted if and only if the equivalent capacity resulting from service acceptance can be feasibly placed, using placement functions described in the previous section, on available physical resources. The placement optimizer functionality is addressed in the next section.

Table 1 presents a simplified simulation study of the theoretical multiplexing gain attainable for different ARL values ranging from 0.15 to 0.01. The simulation comprises three groups of experiments, in which the number of simulated services was 100, 200, and 300, respectively. Each experiment used 5760 data points corresponding to 2 months' worth of monitoring, in which each data point was collected at 15-minute intervals. Each service specified 20 compute units as its maximal demand. To simulate elasticity, the actual number of resources for each service at any given time was drawn from the uniform distribution in the range $[1, 20]$. The stability periods between these resource allotment changes were exponentially distributed with a mean of 50, which corresponds to about 12.5 hours of stability between conceptual changes in resources allocation due to elasticity.

The equivalent capacity grows as ARL diminishes. Moreover, the theoretically-attainable multiplexing gain computed as the ratio between the maximal demand and equivalent capacity ranges from 1.89 to 1.29. The actual multiplexing gain depends on the specific placement policy and available physical capacity.

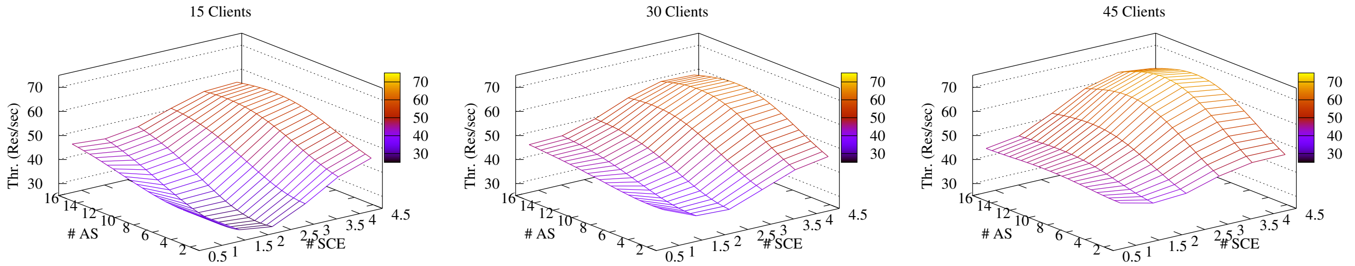

Fig. 2. Reduced surrogate model of service throughput as a function of the number of concurrent clients, and number of VM instances; These models are combined to find the optimal system configuration for a given workload and strategy (e.g., find the cheapest configuration for a throughput threshold)

TABLE 1
Equivalent capacity in compute units as a function of the number of services and acceptable risk level probability

Equivalent capacity vs maximal demand			
	100 services	200 services	300 services
Maximal Demand	2000	4000	6000
ARL = 0.15	1058	2331	3169
ARL = 0.1	1075	2370	3220
ARL = 0.05	1124	2489	3374
ARL = 0.01	1523	3443	4618

4.3 Policy-driven placement optimization

The ability to place virtual machines in an effective manner is vital for cost-efficient provisioning of services. Finding the best mapping or placement of virtual machines to physical machines is one of the most challenging problems for cloud management systems. For maximum flexibility in addressing the different needs and policies of each infrastructure provider, RESERVOIR supports dynamically-pluggable policies to calculate the placement. Each policy defines a different utility function to be optimized. For example, a *load-balancing policy* attempts to keep VMs equally distributed over physical machines, while a *power preservation policy* consolidates all VMs in the minimal number of physical machines, thus enabling unused machines to be turned off.

Another interesting use of the pluggable policy framework is the composition of multiple policies into policy chains. These are used to combine local placement on the physical machines in the own cloud with federated VM placement in partnering clouds. The resulting VM placement strategy is a two step process: the system first tries to place all VMs locally according to the active local placement optimization policy; then federated resources are considered for VMs that cannot be placed locally.

Remote placement constitutes a significant challenge as the clouds are separate management entities. As such, each cloud maintains individual management policies and probably wants to keep exact details of their infrastructure private. To overcome this problem we use a scheme where collaboration between two clouds is pre-defined in a *framework agreement* contract. This contract specifies the capacity, in terms of the number and sizes of VMs, that is available to others, together with various non-functional constraints, such as cost, levels of QoS, security levels, contract validity time, etc. Example remote placement policies include *revenue maximization* that weights income from service providers against provisioning costs and eventual SLA violation penalties and *consolidation* of VMs of each service across a minimal number of remote clouds.

We observed that cloud placement optimization problems can be modeled as extended classical combinatorial optimization problems, such as the Generalized Assignment Problem (GAP). To handle large-scale placement problems in the order of hundreds of physical hosts and remote sites and a few thousands VMs, we use a number of approaches, ranging from exact solutions based on techniques such as column generation to inexact solutions, such as approximation algorithms [9].

4.4 Cross cloud virtual networks

To support applications built out of inter-communicating components that may be deployed and migrated across clouds, we need to take a novel approach to networking. Virtual Application Networks (VANs) [10] are a virtual and distributed switching service connecting VMs. VANs revolutionize the way networks are organized by using an overlay network between hypervisors. The hypervisors' overlay network decouples the virtual network services offered to VMs from the underlying physical network. As a result, the network created between VMs becomes independent from the topology of the physical network. Moreover, the resulting virtual networks can be migrated alongside the VMs connected to them.

In addition, VANs offer high levels of security by isolating virtual networks from one another and from the physical network. Such isolation is crucial for constructing large-scale clouds servicing many independent customers. Unlike virtual LANS (VLANs), which are physical resources, VANs do not introduce per virtual network service costs. At the same time, the network performance effect of VANs can be minimized. Providing network isolation of services is vitalnot only when they are running in the same cloud, but also in a federated cloud. Furthermore, cloud providers cannot be expected to coordinate their network maintenance, network topologies, and more with one another. VANs meet these requirements by separating clouds using VAN proxies, which act as gateways among clouds. A VAN proxy hides the internal structure of the cloud from other clouds in a federation. The VAN proxies of different clouds communicate to ensure that VANs can extend across a cloud boundary while adhering to the limitations discussed above.

4.5 Cross cloud monitoring

Monitoring in a federated cloud presents us with some interesting issues. Although the underlying cloud infrastructure is a distributed system, it is structured in a very particular way, with one large set of machines acting as one cloud. Most of the monitoring data stays within the cloud, as all of the service providers are within the cloud. The exception is for federated VMs. With many monitoring systems, the sources and consumers are often distributed arbitrarily across a network, and so the paths of data flow and the patterns of interaction are also arbitrary. Within RESERVOIR, the pattern is more predictable, and we are therefore able to design and build for this.

For cross-cloud federation of monitoring to operate, we need to undertake the following tasks:

- Address the setup of federated monitoring when the first VM for a service arrives at a cloud
- Create the cloud-to-cloud connections for sending measurements back to the home cloud
- Address the tear-down of remoting when the last VM for a service is migrated away from a cloud
- Ensure that remote and connected VMs are kept separate from other services

Within RESERVOIR, we have built a distributed monitoring system which has all of the necessary monitoring probes and consumers, and supports the closed control loops for the virtual infrastructures, including service clouds and virtual networks. For the monitoring data plane we use a combination IP multicast together with the Java Message Service (JMS) publish-subscribe system. We also use JMS for the federated monitoring, although cloud-to-cloud monitoring may use different protocols than the intra-cloud ones to ensure interoperability.

4.6 Cross cloud live migration

Live migration techniques require a direct communication link between the source and destination hypervisors. However, security and privacy considerations prevent a cloud provider from allowing another clouds direct access to its hypervisors. In order to overcome this apparent contradiction, we introduced a novel *federated migration channel* to transfer VMs from a source host in one cloud to a destination host in another cloud without directly addressing the destination host. Instead, the VM passes through a secure tunnel connecting proxies in the source and destination clouds. At the destination site, the VM is forwarded to the chosen destination host.

5 ON-DEMAND ENTERPRISE SYSTEMS

SAP systems are used for a variety of business applications that differ in version and functionality (such as Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP)). We deliberately chose SAP as the main application to demonstrate the challenges for cloud computing providers because it helps raise the enterprise-grade requirements not captured in the typical web-based applications prevalent as cloud-based offerings.

The main challenges facing a cloud computing provider wishing to host SAP applications are: (i) efficiently managing the life-cycle of the different SAP applications for hundreds or thousands of tenants while keeping a very low Total Cost of Ownership (TCO); (ii) consolidating many applications on the same infrastructure, thereby increasing hardware utilization and optimizing power consumption, while keeping the operational cost of a site to a minimum; and (iii) guaranteeing the individual Service Level Agreements (SLAs) of the infrastructure customers (i.e., the service providers).

The SAP use case made use of the federation capabilities of RESERVOIR for the sake of operational flexibility. We successfully experimented with scenarios of initial deployment of a multi-VM application across multiple data centers. We also successfully exercised automated elasticity to respond to changes in application load.

Our experiments show that it is feasible to deploy enterprise-grade complex applications in a federated cloud. However this does not come without problems and limitations. For example:

- When dealing with a complex multi-VM application, we encountered certain technical problems that stretched the naive mechanisms of rapid provisioning and elasticity to the limit. For example, the SAP system requires a special start-up sequence and SAP licensing is coupled to the identity of a real machine; but in a cloud infrastructure the application runs in a VM
- Some of the biggest obstacles in our experiments were the size of the images, the time it takes to create them, and the time it takes to start an application from an image; for example, an image for the DBMS component can use more than 100 GB, and take a few minutes to start.
- SAP applications use stateful and sticky sessions; that is, once a session is opened with a specific user, a state is maintained for that user in a specific server. As a consequence, even when the number of

active sessions decreases, the SAP sessions may be spread across servers in such a manner that makes it impossible to release resources.

6 CLOUDY FUTURE

Cloud computing is not a passing phenomenon. While companies may still be reluctant to fully embrace the hosted model that cloud computing presents, they are adopting cloud computing methodologies to organize their own data centers into private clouds. But the potential flexibility and cost savings are limited in private clouds, hence we are now seeing the rise of the hybrid cloud computing model. In this model, companies have their own private clouds but spill over some of their computing needs to a hosted public cloud, as needed [11]. This is essentially a partial realization of the RESERVOIR federated cloud computing model. All indications from the market show that this trend will continue.

While we believe that it is only natural that cloud computing providers will eventually reach their optimal capacity and adopt the federated cloud model, we are still a long way from instituting this model, particularly regarding standardization. Contemporary cloud technologies were not designed with interoperability in mind. But, just as with other utilities, in which we get service without knowing the internals of the utility provider, and with standard equipment not specific to any provider (such as telephones)—for cloud computing services to really fulfill the computing as a utility vision, providers will need to offer standardized services. This, in turn, will accelerate the adoption of the federated model. In the RESERVOIR project, we have shown, within the limitations of today's technologies (e.g., no interoperability between hypervisors), that a full federated cloud has huge potential.

Finally, we have also shown that deploying and running existing enterprise-grade applications (that were not originally designed for the cloud) is definitely possible, yet not a straight-forward process. A new generation of cloud-native business applications are likely to emerge. Such cloud-native applications may further utilize the unique capabilities of clouds, e.g., live migration across clouds. In the meantime, enterprises should adopt a model that is a hybrid of on-premise and on-demand models to fully leverage the benefits of the cloud computing paradigm while maintaining their current investments.

ACKNOWLEDGMENTS

We thank all members of the RESERVOIR consortium, without whose contributions this paper would not have been possible.

REFERENCES

[1] R. M. Fano, "The MAC system: The computer utility approach," *IEEE Spectrum*, vol. 2, p. 5644, January 1965.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the clouds: A Berkeley view of cloud computing," tech. rep., University of California, Berkeley, 2009.

[3] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *International Journal of High Performance Computing Applications*, vol. 15, no. 3, pp. 200–222, 2001.

[4] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Cáceres, M. Ben-Yehuda, W. Emmerich, and F. Galán, "The RESERVOIR model and architecture for open federated cloud computing," *IBM Journal of Research and Development*, vol. 53, no. 4, 2009.

[5] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "kvm: The linux virtual machine monitor," in *Proceedings of the Linux Symposium*, pp. 225–230, June 2007.

[6] C. Chapman, W. Emmerich, F. G. Márquez, S. Clayman, and A. Galis, "Software architecture definition for on-demand cloud provisioning," in *HPDC '10: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, pp. 61–72, 2010.

[7] DMTF, "Open virtualization format specification. specification dsp0243 v1.0.0d," September 2008.

[8] G. Toffetti, A. Gambi, M. Pezzè, and C. Pautasso, "Engineering autonomic controllers for virtualized web applications," in *Web Engineering* (B. Benatallah, F. Casati, G. Kappel, and G. Rossi, eds.), vol. 6189 of *Lecture Notes in Computer Science*, pp. 66–80, Springer Berlin / Heidelberg, 2010.

[9] D. Breitgand and A. Epstein, "SLA-aware Placement of Multi-Virtual Machine Elastic Services in Compute Clouds." IBM Technical Report, H-0287, 2010.

[10] D. Hadas, S. Guenender, and B. Rochwerger, "Virtual network services for federated cloud computing," tech. rep., IBM, 2009. Research Report H-0269.

[11] D. Linthicum, "The case for the hybrid cloud," *Infoworld Cloud Computing*, March 2010.

BIOGRAPHIES

Benny Rochwerger is a senior technical staff member at the IBM Haifa Research Lab. Mr. Rochwerger has an M.S. in Computer Science from the University of Massachusetts at Amherst. His research interests include cloud computing, distributed systems and networking virtualization.

Johan Tordsson is a researcher at Umeå University. Dr. Tordsson holds a Ph.D in Computer Science from Umeå University. His research interests include automatic resource management for grid and cloud computing environments.

Carmelo Ragusa is a a Researcher at University of Messina. Dr. Ragusa has a Ph.D. in communication systems from the University of Surrey, U.K. His research interests include distributed computing, Cloud computing, Grid and SOA.

David Breitgand is a research staff member at the IBM Haifa Research Lab. Dr. Breitgand has a Ph.D. in Computer Science from the Hebrew University of Jerusalem. His research interests include end-to-end performance analysis and management of networked storage and systems.

Stuart Clayman is a Senior Research Fellow in the Dept. of Electronic Engineering at University College London. Dr. Clayman has a Ph.D. in Computer Science from the University College London. His research interests include distributed systems, management systems and time-ordered data processing.

Amir Epstein is a research staff member at the IBM Haifa Research Lab. Dr. Epstein has a Ph.D. in Computer Science from the Tel Aviv University, Israel. His research interests include approximation methods, algorithmic game theory and cloud computing

David Hadas is a research staff member at the IBM Haifa Research Lab. Mr. Hadas is at the final stages of his M.Sc. in Computer Science from the Tel Aviv University in Israel. His research interests include distributed networking, machine learning and biological learning processes.

Eliezer Levy is the Director of SAP Research in Israel. Dr. Levy has a Ph.D. in Computer Sciences from the University of Texas at Austin. His research interests include infrastructures for business applications, databases and distributed computing.

Irit Loy is a research staff member at the IBM Haifa Research Lab. Mrs. Loy has a M.Sc. in Electrical Engineering from the Israel Institute of Technology (Technion), Haifa. Her research interests include cloud computing, parallel and distributed systems and file systems.

Alessandro Maraschini is a Team Leader and Software Engineer in Elsag Datamat. Mr. Maraschini has a M. Sc. in Electronic Engineering from the University ”La Sapienza”, Rome. His research interests include grid and cloud computing technology-based projects.

Philippe Massonet is the scientific coordinator at CETIC, a Belgian ICT applied research center. Philippe holds the degrees of industrial engineer from the Université Catholique de Louvain in Belgium His research interests cover the areas of software/security engineering, and distributed systems such as Clouds and Grids.

Henar Muñoz is a Research and Development Engineer in Telefónica I+D. Ms. Muñoz has a M. Sc. in Telecommunication from the University of Valladolid, Valladolid (Spain). Her research interests include cloud computing and semantics technologies.

Kenneth Nagin is a research staff member at the IBM Haifa Research Lab. Mr. Nagin holds a B.S. in Computer Science from the University of Pittsburgh. His research interests include software based test generation consultation, cloud computing and virtualization.

Giovanni Toffetti is currently a postdoc researcher at University of Lugano. Dr. Toffetti received is Ph.D. in Computer Science from Politecnico di Milano. His main research interests are distributed systems, Web applications, and content-based routing.

Massimo Villari is an assistant professor at University of Messina. Dr. Villari has PhD in in Electronic Engineering from University of Messina. His research interests include cloud computing and security in distributed systems. He is member of IEEE.

Direct questions and comments about this article to Benny Rochwerger, rochwer@il.ibm.com