Virtual Machine Placement in Cloud Environments

Wubin Li

李务斌

LICENTIATE THESIS, MAY 2012
DEPARTMENT OF COMPUTING SCIENCE
UMEÅ UNIVERSITY
SWEDEN

Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden

wubin.li@cs.umu.se

ISBN 978-91-7459-453-9
ISSN 0348-0542
UMINF 12.13

Printed by Print & Media, Umeå University, 2012

Abstract

With the emergence of cloud computing, computing resources (i.e., networks, servers, storage, applications, and services) are provisioned as metered on-demand services over networks, and can be rapidly allocated and released with minimal management effort. In the cloud computing paradigm, the virtual machine is one of the most commonly used resource carriers in which business services are encapsulated. Virtual machine placement optimization, i.e., finding optimal placement schemes for virtual machines, and reconfigurations according to the changes of environments, become challenging issues.

The primary contribution of this licentiate thesis is the development and evaluation of our combinatorial optimization approaches to virtual machine placement in cloud environments. We present modeling for dynamic cloud scheduling via migration of virtual machines in multi-cloud environments, and virtual machine placement for predictable and time-constrained peak loads in single-cloud environments. The studied problems are encoded in a mathematical modeling language and solved using a linear programming solver. In addition to scientific publications, this work also contributes in the form of software tools (in EU-funded project OPTIMIS) that demonstrate the feasibility and characteristics of the approaches presented.

Preface

This thesis consists of an introduction to cloud computing, a brief discussion of virtual machine placement in cloud environments, and the below listed papers.

Paper I — Wubin Li, Johan Tordsson, and Erik Elmroth. Modeling for Dynamic Cloud Scheduling via Migration of Virtual Machines. In *Proceedings of the 3rd IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2011)*, pages 163–171, 2011.

Paper II — Wubin Li, Johan Tordsson, and Erik Elmroth. Virtual Machine Placement for Predictable and Time-Constrained Peak Loads. In *Proceedings of the 8th international conference on Economics of grids, clouds, systems, and services (GECON 2011)*, Lecture Notes in Computer Science, Vol. 7150, Springer-Verlag, pp. 120–134, 2011.

Paper III — Wubin Li, Petter Svärd, Johan Tordsson, and Erik Elmroth. A General Approach to Service Deployment in Cloud Environments. Technical Report UMINF-12.14, May, 2012. Department of Computing Science, Umeå University, 2012.

This research was conducted using the resources of the High Performance Computing Center North (HPC2N) and the UMIT research lab. Financial support has been provided by the European Community's Seventh Framework Programme ([FP7/2001-2013]) under grant agreements no. 257115 (OPTIMIS) and no. 257019 (VISION Cloud), and the Swedish Government's strategic effort eSSENCE.

Acknowledgments

The accomplishment of this licentiate thesis has been one of the most significant academic challenges I've ever had to face. Without the support, patience and guidance of numerous people, it would not have been completed. In particular, I would like to express my sincere and deepest gratitude to:

Erik Elmroth, my supervisor, for inviting me to Sweden and providing an excellent research environment, for his timely meetings, discussions and emails, for his enthusiasm and invaluable suggestions.

Johan Tordsson, my co-supervisor, for working over time with me for paper deadlines during the weekend(s), for sharing your knowledge and experience, for the inspiring discussions, for the outstanding job you did on proofreading my papers.

Peter Svärd, for your nice personality which always makes me feel easy and relaxed when working and travelling with you.

Daniel Espling and **Lars Larsson**, for sharing your knowledge and experience, for answering me tons of questions and helping me with various tools and systems.

P-O Östberg, for sharing your cabin and making my first ski-trip in Sweden a reality, and for the excellent lectures presented in the SOA course.

Other group members (in no particular order), **Ewnetu Bayuh Lakew**, **Lei Xu**, **Mina Sedaghat**, **Ahmed Ali-Eldin**, **Francisco Hernández**, **Peter Gardfjäll**, **Lennart Edblom**, and **Tomas Forsman**, for all their contributions to our collective effort.

All floorball team players, for creating the exciting and passionate moments every Tuesday in IKSU.

Lastly, I would like to thank my family for all their love and encouragement. Thank you.

Umeå, May 2012
Wubin Li

Contents

Introduction

By provisioning of shared resources as a metered on-demand service over networks, Cloud Computing is emerging as a promising paradigm for providing configurable computing resources (i.e., networks, servers, storage, applications, and services) that can be rapidly allocated and released with minimal management effort. Cloud end-users (e.g., service consumers and developers of cloud services) can access various services from cloud providers such as Amazon, Google and SalesForce. They are relieved from the burden of IT maintenance and administration and it is expected that their total IT costs will decrease. From a cloud provider's or an agent's perspective, however, due to the scale of resources to manage, and the dynamic nature of service behaviours (with rapid demands for capacity variations and resource mobility), as well as the heterogeneity of cloud systems, resource allocation and scheduling are becoming challenging issues, e.g., to find optimal placement schemes for resources, and resource reconfigurations in response to the changes of the environment [11].

There is a multitude of parameters and considerations (e.g., performance, cost, locality, reliability and availability, etc.) involved in the decision of where and when to place and reallocate data objects and computation resources in cloud environments. Some of the considerations are consistent with one another while others may be contradicting. This work investigates challenges involved in the problem of resource placement and scheduling in cloud environments, tackles the problem using combinatorial optimization techniques and mathematical modeling. Thesis contributions include scientific publications addressing, e.g., modeling for dynamic cloud scheduling via migration of Virtual Machines (VMs) in multi-cloud environments, as well as to optimal virtual machine placement within datacenters for predicable and time-constrained load peaks. In addition, this work also contributes in the form of software tools (in the EU-funded project Optimis [13]) that demonstrate the feasibility and characteristics of the proposed solutions.

Chapter 1

Cloud Computing

Cloud Computing provides a paradigm shift following the shift from mainframe to client-server architecture in the early 1980s [14] [32] and it is a new paradigm in which computing is delivered as a service rather than a product, whereby shared resources, software, and information are provided to consumers as a utility over networks.

1.1 Hardware Virtualization

Virtualization is a technology that separates computing functions and implementations from physical hardware. It is the foundation of cloud computing, since it enables isolations between hardware and software, between users, and between process and resources. These isolation problems are not well solved by traditional operating systems. Hardware virtualization approaches include *Full Virtualization*, *Partial virtualization* and *Paravirtualization* [31]. With virtualization, software capable of execution on the raw hardware can be run in a virtual machine. Cloud systems deployable services can be encapsulated in virtual appliances (VAs) [18], and deployed by instantiating virtual machines with their virtual appliances [17]. This new type of service deployment provides a direct route for traditional on-premises applications to be rapidly redeployed in a Software as a Service (SaaS) mode. By decoupling the hardware and operating system infrastructure provider from the application stack provider, virtual appliances allow economies of scale on the one side to be leveraged by the economy of simplicity on the other.

1.2 The XaaS Service Models

Commonly associated with cloud computing are the following service models:

- Software as a Service (SaaS)

In the SaaS model, software applications are delivered as services that execute on infrastructure managed by the SaaS vendor. Consumers are enabled to access services over various clients such as web browsers and programming interfaces, and are typically charged on a subscription basis [6]. The implementation and the underlying cloud infrastructure where it is hosted is transparent to consumers.

- Platform as a Service (PaaS)
  In the PaaS model, cloud providers deliver a computing platform and/or solution stack typically including operating system, programming language execution environment, database, and web server [5]. Application developers can develop and run their software on a cloud platform without having to manage or control the underlying hardware and software layers, including network, servers, operating systems, or storage, but maintains the control over the deployed applications and possibly configuration settings for the application-hosting environment [24].

- Infrastructure as a Service (IaaS)
  In IaaS model, computing resources such as storage, network, and computation resources are provisioned as services. Consumers are able to deploy and run arbitrary software, which can include operating systems and applications. Consumers do not manage or control the underlying cloud infrastructure but have to control its own virtual infrastructure typically constructed by virtual machines hosted by the IaaS vendor. This thesis work mainly focus on this model, although it may be generalized to also apply to the other models.

1.3 Cloud Computing Scenarios

Based on the classification of cloud services into SaaS, PaaS, and IaaS, two main stakeholders in a cloud provisioning scenario can be identified, i.e., the *Infrastructure Provider* (IP) who offers infrastructure resources such as Virtual Machines, networks, storage, etc. which can be used by *Service Providers* (SPs) to deliver end-user services such as SaaS to their consumers, these services potentially being developed using PaaS tools. As identified in [7], four main types of cloud scenarios can be listed as follows.

- Private Cloud

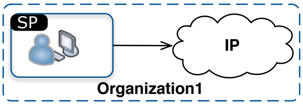

Figure 1: Private cloud scenario.

An organization provisions services using internal infrastructure, and thus plays the roles of both and SP and an IP. Private clouds can circumvent many of the security and privacy concerns related to hosted sensitive information in public clouds, the latter a case where the SP leases IaaS resources publicly available IPs. Private clouds may also offer stronger guarantees on control and performance as the whole infrastructure can be administered within the same domain.

- Cloud Bursting

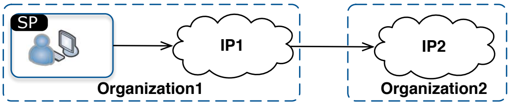

Figure 2: Cloud bursting scenario.

Private clouds may offload capacity to other IPs under periods of high workload, or for other reasons, e.g., planned maintenance of the internal servers. In this scenario, the providers form a hybrid architecture commonly referred to as a *cloud bursting* as seen in Figure 2. Typically, less sensitive tasks are executed in the public cloud instead while tasks that requiring higher levels of security are provisioned the private infrastructure.

- Federated Cloud

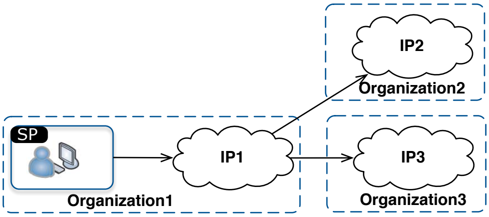

Figure 3: Cloud federation scenario.

Federated clouds are IPs collaborating on a basis of joint load-sharing agreements enabling them to offload capacity to each others [28] in a manner similar to how electricity providers exchange capacity. The federation takes place at the IP level in a transparent manner. In other words, an SP that deploys services to one of the IPs in a federation is not notified if its service is off-loaded to another IP within the federation. However, the SP is able to steer in which IPs the service may be provisioned, e.g., by specifying location constraints in the service manifest, Figure 3 illustrates a federation between three IPs.

- Multi-Cloud

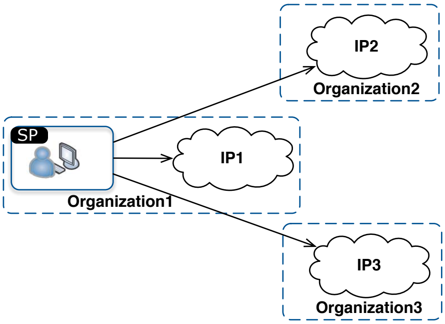

Figure 4: Multi-cloud scenario.

In multi-cloud scenarios, the SP is responsible for handling the additional complexity of coordinating the service across multiple external IPs, i.e., planning, initiating and monitoring the execution of services.

It should be remarked that the multi-cloud and federated cloud scenarios are commonly considered only for the special case where organization 1 does not possess an internal IP, corresponding to removing IP1 from figures 3 and 4.

Chapter 2

Virtual Machine Placement

Given a set of admitted services and the availability of local and possibly remote resources, there are a number of placement problems to be solved to determine where to store data and where to execute VMs. The following sections describe the challenges and state of the art of VM placement and scheduling in cloud environments.

2.1 Parameters and Considerations

There are a multitude of parameters and considerations involved in the decision of where and when to place/reallocate data objects and computations in cloud environments. An automated placement and scheduling mechanism should take into account the considerations and tradeoffs, and allocate resources in a manner that benefits the stakeholder for which it operates (SP or IP). For both of these, this often leads to the problem of optimizing price or performance given a set of constraints, often including the one of price and performance that is subject to optimization. Among the main considerations are:

- **Performance:** In order to improve the utilization of physical resources, data centers are increasingly employing virtualization and consolidation as a means to support a large number of disparate applications running simultaneously on server platforms. With different placement schemes of virtual machines, the performance achieved may differ a lot [29].

- **Cost:** The price model was dominated by fixed prices in the early phase of cloud adoption. However, cloud market trend shows that dynamic pricing schemes utilization is being increased [23]. Investment decreases by dynamically placing services among clouds or by dynamically reconfiguring services (e.g., resizing VM sizes without harming service performance) become possible. In addition, internal cost for VM placement, e.g., interference and overhead that one VM causes on other concurrently running VMs on the same physical host, should also be taken in to account.

- **Locality:** In general, for considerations of usability and accessibility, VMs should be located close to users (which could be other services/VMs). However, due to e.g., legal issues and security reasons, locality may become a constraints for optimal placement.
- **Reliability and continuous availability:** Part of the central goals for VM placement is service reliability and availability. To achieve this, VMs may be placed/replicated/migrated across multiple (at least two) geographical zones. During this procedure, factors such as the importance of the data/service encapsulated in VMs, its expected usage frequency, and the reliability of the different data centers, must be taken in to account.

2.2 Challenges

Given the variety of deployment scenarios, the range of relevant parameters, and the set of constraints and objective functions of potential interest, there are a number of challenges to the development of broadly applicable placement methods, some of which are presented below.

- Firstly, there exists no generic model to represent various scenarios of resource scheduling, especially when users' requirements are vague and hard to encode through modeling languages.
- Secondly, model parameterization, i.e., finding suitable values for parameters in a proposed model is a tedious task when the problem size is large. For example, in for a multi-cloud scenario that includes n cloud providers and m VMs, $m * n^2$ assignments are needed to express the VM migration overheads ignoring possible changes of VM sizes. Therefore, mechanisms that can help to automatically capture those values are required.
- Thirdly, the VM placement problem is typically formulated as a variant of the class constrained multiple-knapsack problem that is known to be NP hard [9]. Thus, tradeoffs between quality of solution and execution time must be taken into account. This is a very important issue given the size of real life data centers, e.g., Amazon EC2 [4], the leading cloud provider, has approximately 40,000 servers and schedules 80,000 VMs every day [12].

2.3 State of the art

Virtual machine placement in distributed environments has been extensively tudied in the context of cloud computing. Such approaches address distinct problems, such as initial placement, consolidation, or tradeoffs between honoring service level agreements and constraining provider operating costs, etc. [25]. Studied scenarios are usually encoded in mathematical models and are finally solved either by algorithms such as approximation, greedy packing and heuristic method, or by existing programming solvers such as Gurobi [1], CPLEX [2] and GLPK [3]. Those related work can be separated into two sets: (1) VM placement in single-cloud environments and (2) VM placement in multi-cloud environments.

In single-cloud environments, given a set of physical machines and a set of services (encapsulated within VMs) with dynamically changing demands, on-line placement controllers that decide how many instances to run for each service and where to put and execute them, while observing resource constraints, are NP hard problems. Tradeoff between quality of solution and computation cost is a challenge. To address this issue, various approximation approaches are applied, e.g., by Tang et al. [9] propose an algorithm that can produce within 30 seconds high-quality solutions for hard placement problems with thousands of machines and thousands of VMs. This approximation algorithm strives to maximize the total satisfied application demand, to minimize the number of application starts and stops, and to balance the load across machines. Hermenier et al. [15] present the Entropy resource manager for homogeneous clusters, which performs dynamic consolidation based on constraint programming and takes migration overhead into account. Entropy chooses migrations that can be implemented efficiently, incurring a low performance overhead. The CHOCO constraint programming solver [16], with optimizations e.g., identifying lower and upper bounds that are close to the optimal value, is employed to solve the problem. To reduce electricity cost in high performance computing clouds that operate multiple geographically distributed data centers, Le et al. [19] study the impact of VM placement policies on cooling and maximum data center temperatures, develop a model of data center cooling for a realistic data center and cooling system, and design VM distribution policies that intelligently place and migrate VMs across the data centers to take advantage of time-based differences in electricity prices and temperatures.

For VM placement across multiple cloud providers, information about the number of physical machines, the load of these physical machines, and the state of resource distribution inside the IP side are normally hidden from SP, and hence not parameters that can be used for placement decisions. Only provision-related information such as types of VM instance, price schemes, are exposed to SP. Hence, most works on VM placement across multi-cloud environments are focusing on cost aspects. Chaisiri et al. [8] propose an stochastic integer programming (SIP) based algorithm that can minimize the cost spending in each placement plan for hosting virtual machines in a multiple cloud provider environment under future demand and price uncertainty. Bossche et al. [10] examine the workload outsourcing problem in a multi-cloud setting with deadline-constrained, and present cost-optimal optimization to maximize the utilization of the internal data center and to minimize the cost of running the outsourced tasks in the cloud, while fulfilling the applications quality of service constraints. Tordsson et al. [30], propose a cloud brokering mechanisms for optimized placement of VMs to obtain optimal cost-performance tradeoffs across multiple cloud providers. Similarly, Vozmediano et al. [27] [26] explore the multi-cloud scenario to deploy a computing cluster on top of a multi-cloud infrastructure, for solving loosely-coupled Many-Task Computing (MTC) applications. In this way, the cluster nodes can be provisioned with resources from different clouds to improve the cost-effectiveness of the deployment, or to implement high-availability strategies.

Chapter 3

Summary of Contributions

3.1 Paper I

In Paper I [22], we investigate dynamic cloud scheduling use cases where parameters are continuously changed, and propose a linear programming model to dynamically reschedule VMs (including modeling of VM migration overhead) upon changed conditions such as price changes, service demand variation, etc. in dynamic cloud scheduling scenarios. Our model can be applied in various scenarios through selections of corresponding objectives and constraints, and offers the flexibility to express different levels of migration overhead when restructuring an existing virtual infrastructure, i.e., VM layout. In scenarios where new instance types are introduced, the proposed mechanisms can accurately determine the break-off point when the improved performance resulting from migration outweighs the migration overhead. It is also demonstrated that our cloud mechanism can cope with scenarios where prices change over time. Performance changes, as well as transformation of VM distribution across cloud providers as a consequence of price changes, can be precisely calculated. In addition, the ability of the cloud brokering mechanism to handle the tradeoff between vertical (resizing VMs) and horizontal elasticity (adding VMs), as well as to improve decision making in complex scale-up scenarios with multiple options for service reconfiguration, e.g., to decide how many new VMs to deploy, and how many and which VMs to migrate, is also evaluated in scenarios based on commercial cloud providers' offerings.

3.2 Paper II

In Paper II [21], the VM placement problem for load balancing of predictable and time-constrained peak workloads is studied for placement of a set of virtual machines within a single datacenter. We formulate the problem as a Min-Max optimization problem and present an algorithm based on binary integer programming, along with three approximations for tradeoffs in scalability and performance. Notably, two VM sets (i.e., VMs provisioned to fulfill services demands) may use the same physical resources if they do not overlap in runtime. We use an approximation based on discrete time slots to generate all possible overlap sets. Finally, a time-bound knapsack algorithm is derived to compute the maximum load of machines in each overlap set after placing all VMs that run in that set. Upper bound based optimizations are used to reduce the time required to compute a final solution, enabling larger problems to be solved. Evaluations based on synthetic workload traces suggest that our algorithms are feasible, and that these can be combined to achieve desired tradeoffs between quality of solution and execution time.

3.3 Paper III

The cloud computing landscape has developed into a spectrum of cloud architectures, leading to a broad range of management tools for similar operations but specialized for certain deployment scenarios. This both hinders the efficient reuse of algorithmic innovations for performing the management operations and increases the heterogeneity between different cloud management systems. A overarching goal is to overcome these problems by developing tools general enough to support the range of popular architectures. In Paper III [20], we analyze commonalities in recently proposed cloud models (private clouds, multi-clouds, bursted clouds, federated clouds, etc.) and demonstrate how a key management functionality - service deployment - can be uniformly performed in all of these by a carefully designed system. The design of our service deployment solution is validated through demonstration of how it can be used to deploy services, perform bursting and brokering, as well as mediate a cloud federation in the context of the OPTIMIS Cloud toolkit.

Chapter 4

Future Work

Future directions for this work include to model the interconnection requirements that can precisely express the relationships between VMs to be deployed. Another area of future work is approximation algorithms based on problem relaxations and heuristic approaches such as greedy formulation for considerations of tradeoff between quality of solution and execution time. Additionally, for VM placement problems, interference and overhead that one VM causes on other concurrently running VMs on the same physical host should be taken in to account. In addition, we are working on a specific scenario where cloud users can specify hard constraints and soft constraints when demanding resource provisions. A hard constraint is a condition that to be satisfied when deploying services, i.e., it is mandatory. In contrast, a soft constraint (also called a preference) is optional. An optimal placement solution with soft constraints satisfied is preferable over other solutions. The hard and soft constraints can, e.g., be used to specify collocation or avoidance of co-location of certain VMs. We are also investigating how to apply multi-objective optimization techniques to this scenario.

Bibliography

[1] Gurobi Optimization, http://www.gurobi.com, visited October 2011.

[2] IBM ILOG CPLEX Optimizer, http://www.ibm.com/software/integration/optimization/cplex-optimizer/, visited October 2011.

[3] GNU Linear Programming Kit, http://www.gnu.org/s/glpk/, visited October 2011.

[4] Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2/, visited May, 2012.

[5] Platform as a Service. http://en.wikipedia.org/wiki/Platform_as_a_service, visited May, 2012.

[6] Software as a Service (SaaS). Cloud Taxonomy. http://cloudtaxonomy.opencrowd.com/taxonomy/software-as-a-service/, visited May, 2012.

[7] M. Ahronovitz et al. Cloud computing use cases white paper, v4.0. www.cloudusecases.org, visited May 2012.

[8] S. Chaisiri, B.-S. Lee, and D. Niyato. Optimal virtual machine placement across multiple cloudproviders. In *Proceedings of the 4th IEEE Asia-Pacific Services Computing Conference*, pages 103–110.

[9] T. Chunqiang, S. Malgorzata, S. Michael, and P. Giovanni. A scalable application placement controller for enterprise data centers. In *Proceedings of the 16th international conference on World Wide Web*, WWW'07, pages 331–340. ACM, 2007.

[10] R. V. den Bossche, K. Vanmechelen, and J. Broeckhove. Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads. In *Proceedings of the 2010 IEEE International Conference on Cloud Computing*, pages 228–235. IEEE Computer Society, 2010.

[11] E. Elmroth, J. Tordsson, F. Hernández, A. Ali-Eldin, P. Svärd, M. Sedaghat, and W. Li. Self-management challenges for multi-cloud architectures. In W. Abramowicz, I. Llorente, M. Surridge, A. Zisman, and J. Vayssière,

editors, *Towards a Service-Based Internet*, volume 6994 of *Lecture Notes in Computer Science*, pages 38–49. Springer Berlin/Heidelberg, 2011.

[12] D. Erickson, B. Heller, S. Yang, J. Chu, J. D. Ellithorpe, S. Whyte, S. Stuart, N. McKeown, G. M. Parulkar, and M. Rosenblum. Optimizing a virtualized data center. In *Proceedings of the 2011 ACM SIGCOMM Conference (SIGCOMM'11)*, pages 478–479, 2011.

[13] A. J. Ferrer, F. Hernádez, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R. M. Badia, K. Djemame, W. Ziegler, T. Dimitrakos, S. K. Nair, G. Kousiouris, K. Konstanteli, T. Varvarigou, B. Hudzia, A. Kipp, S. Wesner, M. Corrales, N. Forgó, T. Sharif, and C. Sheridan. OPTIMIS: A holistic approach to cloud service provisioning. *Future Generation Computer Systems*, 28(1):66–77, 2012.

[14] B. Hayes. Cloud computing. *Communications of the ACM*, 51(7):9–11, July 2008.

[15] F. Hermenier, X. Lorca, J.-M. Menaud, G. Muller, and J. Lawall. Entropy: a consolidation manager for clusters. In *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, VEE '09, pages 41–50, New York, NY, USA, 2009. ACM.

[16] N. Jussien, G. Rochart, and X. Lorca. The CHOCO constraint programming solver. In *Proceedings of the CPAIOR'08 workshop on OpenSource Software for Integer and Contraint Programming (OSSICP'08)*, 2008.

[17] G. Kecskemeti, P. Kacsuk, T. Delaitre, and G. Terstyanszky. Virtual Appliances: A Way to Provide Automatic Service Deployment. In F. Davoli, N. Meyer, R. Pugliese, and S. Zappatore, editors, *Remote Instrumentation and Virtual Laboratories*, pages 67–77. Springer US, 2010.

[18] G. Kecskemeti, G. Terstyanszky, P. Kacsuk, and Z. Neméth. An Approach for Virtual Appliance Distribution for Service Deployment. *Future Gener. Comput. Syst.*, 27(3):280–289, March 2011.

[19] K. Le, R. Bianchini, J. Zhang, Y. Jaluria, J. Meng, and T. D. Nguyen. Reducing Electricity Cost Through Virtual Machine Placement in High Performance Computing Clouds. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 22:1–22:12, New York, NY, USA, 2011. ACM.

[20] W. Li, P. Svärd, J. Tordsson, and E. Elmroth. A General Approach to Service Deployment in Cloud Environments. Technical Report UMINF-12.14, May, 2012. Department of Computing Science, Umeå University, 2012.

[21] W. Li, J. Tordsson, and E. Elmroth. Virtual Machine Placement for Predictable and Time-Constrained Peak Loads. In *Proceedings of the 8th international conference on Economics of grids, clouds, systems, and services (GECON'11).* Lecture Notes in Computer Science, Vol. 7150, Springer-Verlag, pp. 120-134, 2011.

[22] W. Li, J. Tordsson, and E. Elmroth. Modeling for Dynamic Cloud Scheduling via Migration of Virtual Machines. In *Proceedings of the 3rd IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2011)*, pages 163–171, 2011.

[23] J. Lucas Simarro, R. Moreno-Vozmediano, R. Montero, and I. Llorente. Dynamic Placement of Virtual Machines for Cost Optimization in Multi-Cloud Environments. In *Proceedings of the 2011 International Conference on High Performance Computing and Simulation (HPCS)*, pages 1 –7, july 2011.

[24] P. Mell and T. Grance. The NIST definition of cloud computing. *National Institute of Standards and Technology (NIST)*, 2011.

[25] K. Mills, J. Filliben, and C. Dabrowski. Comparing vm-placement algorithms for on-demand clouds. In *Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science*, CLOUDCOM '11, pages 91–98, Washington, DC, USA, 2011. IEEE Computer Society.

[26] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente. Elastic management of web server clusters on distributed virtual infrastructures. *Concurrency and Computation: Practice and Experience*, 23(13):1474–1490, 2011.

[27] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente. Multicloud deployment of computing clusters for loosely coupled mtc applications. *IEEE Transactions on Parallel and Distributed Systems*, 22:924–930, 2011.

[28] 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. Galan. The RESERVOIR model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53(4):1–11, 2009.

[29] O. Tickoo, R. Iyer, R. Illikkal, and D. Newell. Modeling Virtual Machine Performance: Challenges and Approaches. *SIGMETRICS Perform. Eval. Rev.*, 37(3):55–60, Jan. 2010.

[30] J. Tordsson, R. Montero, R. Moreno-Vozmediano, and I. Llorente. Cloud Brokering Mechanisms for Optimized Placement of Virtual Machines across Multiple Providers. *Future Generation Computer Systems*, 28(2):358 – 367, 2012.

[31] VMware. Understanding Full Virtualization, Paravirtualization, and Hardware Assist. `http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf`.

[32] P.-C. Yang, J.-H. Chiang, J.-C. Liu, Y.-L. Wen, and K.-Y. Chuang. An efficient cloud for wellness self-management devices and services. In *Proceedings of the Fourth International Conference on Genetic and Evolutionary Computing (ICGEC)*, pages 767 –770, Dec. 2010.