POLITECNICO DI MILANO

Facoltà di Ingegneria dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica, Informazione e Bioingegneria

Strategies for Cloud Systems Run-time Adaptation

Relatore:

Prof. Danilo Ardagna — Politecnico di Milano

Co-relatore:

Dott. Michele Ciavotta — Politecnico di Milano

Tesi Magistrale di:

Michele Guerriero

Matricola 799702

Anno Accademico 2013-2014

To my family,
for all their love and support.

Acknowledgements

Desidero ringraziare innanzitutto la mia famiglia, per il sostegno, la comprensione e l'affetto dimostratomi durante l'intero percorso di studi universitari ed in particolare durante il periodo di produzione di questa tesi, ma soprattutto per gli insegnamenti da sempre impartitimi, senza i quali mai sarei stato in grado di raggiungere questo fondamentale obiettivo.
Ringrazio inoltre tutti i miei amici, per l'affetto sempre dimostratomi e per i mille momenti di gioia che tante volte mi sono stati d'aiuto nel superare momenti difficili.
Un ringraziamento particolare va al Professor Danilo Ardagna, per la comprensione, la professionalità, la competenza e la simpatia che ho avuto modo di conoscere ed apprezzare durante il periodo di svolgimento di questa tesi, ma anche per gli inestimabili insegnamenti, sia professionali che di vita, che ho avuto modo di assorbire.
Ringrazio il Dottor Michele Ciavotta, per l'immenso aiuto datomi nella realizzazione di questo lavoro con particolare riferimento agli aspetti più matematici e alla stesura di quest'opera.
Infine un ringraziamento particolare va alla mia fidanzata Kathly, per avermi sempre sostenuto ed in quest'ultimo difficile ed impegnativo periodo sopportato con grandissimo affetto, comprensione ed amore.

Contents

List of Figures

List of Tables

Abstract

Nowadays Cloud Computing is emerging as a major trend in the ICT industry. The wide spectrum of available Clouds, such as those offered by Microsoft and Google, is contributing to a predicted compound annual growth rate of 19.5% and provides a vibrant technical environment, where small and medium enterprises can create innovative solutions and evolve their existing service offer. Due to the large scale nature of the Cloud and the service centers, resource provisioning is one of the most important challenges. Indeed, modern cloud infrastructures and service centers are characterized by continuous changes in the environment and in the requirements they have to meet. Therefore, in order to provide Cloud services, advanced solutions have to be developed to be able to dynamically adapt the Cloud infrastructure, while providing continuous service and performance guarantees.

This thesis aims to develop capacity allocation techniques able to minimize the cost of the provided Cloud resources, while guaranteeing Quality of Service (QoS) constraints. Moreover the developed technique is employed within the MODAClouds Runtime Platform in order to realize an effective autoscaling mechanism. MODAClouds is a research project whose main goal is to avoid the Cloud vendor lock-in, while providing a model driven development environment and an appropriate runtime infrastructure in order to allow the design with quality assurance and the runtime management of multi cloud applications.
The overall goal of this thesis is to analyze performance of the proposed capacity allocation algorithm and to employ it within a real system, that is the MODAClouds Runtime Platform, developing a tool called Autoscaling Reasoner. We first evaluate, using a specific simulator, the more appropriate time scale, that results to be 5 minutes for spiky workload and 10 minutes for smooth traffic conditions, also with acceptable percentage of QoS violations. Then the proposed capacity allocation technique is compared with multiple heuristics provided in the literature. Finally we validated our solution within the MODAClouds Runtime Platform on

a real application. In this case results shown that the Autoscaling Reasoner is able to correctly scale in and out and to reduce management cost. Moreover also in this case the percentage of QoS violations results to be acceptable, varying from 10% to 25%.

Sommario

In questi anni il Cloud Computing sta emergendo come soluzione principale nell'industria ICT. La sempre più ampia offerta, come nel caso di Microsoft e Google, contribuisce alla crescita annuale del settore, e ad un ambiente tecnico vivace, dove piccole e medie imprese concorrono a creare soluzioni innovative ed a migliorare i propri servizi. A causa delle dimensioni su larga scala del Cloud, la fornitura di risorse è una delle maggiori sfide. Infatti le infrastrutture delle Cloud moderne operano in un mondo caratterizzato da cambiamenti continui nell'ambiente e nei requisiti da soddisfare. Pertanto, devono essere sviluppate soluzioni avanzate in grado di adattare dinamicamente le infrastrutture Cloud, fornendo un servizio continuo e prestazioni garantite. Dal momento che la violazione della qualità del servizio (QoS) può portare ad una perdita di profitti, i fornitori di servizi investono numerose risorse nella ricerca di soluzioni che minimizzino i costi rispettando nel contempo la QoS.
Scopo di questa tesi è di studiare e proporre nuove tecniche di allocazione della capacità di calcolo, in grado di minimizzare i costi per le risorse Cloud allocate e di garantire allo stesso tempo i vincoli sulla qualità del servizio (QoS). La tecnica proposta è adottata all'interno della piattaforma di runtime del progetto MODAClouds, con lo scopo di realizzare un meccanismo di autoscaling. MODAClouds è un progetto di ricerca, finanziato dalla Comunità Europea nell'ambito del programma FP7, il cui scopo primario è quello di risolvere il problema del Cloud provider lock-in, realizzando un ambiente per lo sviluppo model-driven e un'infrastruttura di runtime atti ad abilitare il design e la gestione a runtime di application multi-cloud. L'obiettivo è da un lato, quello di definire la migliore granularità dell'intervallo di tempo su cui eseguire allocazione delle risorse al fine di minimizzare i costi e garantire la QoS, dall'altro quello di realizzare un Autoscaling Reasoner all'interno della piattaforma a runtime di MODAClouds e di valutarne le prestazioni. Innanzitutto abbiamo valutato la tecnica di allocazione della capacità proposta, confrontandola con alcune euristiche presenti in letteratura.

I risultati hanno mostrato che la scala temporale più appropriata è di 5 minuti per un profilo di traffico altamente variabile, con una percentuale accettabile di violazione degli SLA.
Abbiamo poi installato la piattaforma di runtime di MODAClouds in modo da valutare le performance dell'Autoscaling Reasoner implementato. In questo caso i risultati hanno mostrato che l'Autoscaling Reasoner è in grado di eseguire correttamente le azioni di scaling con conseguente riduzione dei costi. Inoltre anche in questo caso la percentuale di violazioni degli SLA è risultata essere accettabile e compresa tra il 10% e il 25%.

CHAPTER 1

Introduction

Nowadays Cloud Computing is emerging as a major trend in the ICT industry. The wide spectrum of available Clouds, such as those offered by Microsoft, Google, Amazon, and IBM, is contributing to a predicted compound annual growth rate of 19.5% [68] and provides a vibrant technical environment, where small and medium enterprises can create innovative solutions and evolve their existing service offer. The spread of cloud systems has unearthed the other side of the medal: if we use these systems, we have to take into account problems in terms of quality of service, service level agreements, security, compatibility, interoperability, cost and performance estimation and so on. Resource provisioning is one of the most important challenges for Clouds. Indeed, modern Cloud infrastructures live in an open world, characterized by continuous changes in the environment and in the requirements they have to meet. Continuous changes occur autonomously and unpredictably, and they are out of control of the Cloud provider. Therefore, in order to provide infrastructure or software as a service, advanced solutions have to be developed to be able to dynamically adapt the cloud infrastructure, while providing continuous service and performance guarantees.
In our work we take the perspective of a Software as a Service (SaaS) provider that deploys her applications as Web Service hosted by a single Infrastructure as a Service (IaaS) provider. This thesis aims to develop capacity allocation techniques able to minimize the cost of the provided Cloud resources at a single provider, while guaranteeing quality of service constraints. We deal with this problem on two different level of granularity: at the first layer, the capacity allocation is achieved considering each WS application as a black box, so without any notion of the

different types of requests it provides. Moreover we extend this formulation considering also requests implemented by each WS application in the optimization model. A receding horizon control which can be based either on the first or the second formulation is presented.
The overall goal is, starting from some analysis about the best time scale length for the capacity allocation algorithm in order to minimize cost and guaranteeing quality of service constraints, to implement and evaluate the receding horizon control in a real scenario, that is the MODAClouds project runtime platform. MODAClouds project aims to develop a software layer for deploying and migrating application transparently at runtime on multiple clouds. Further details about the MODAClouds project can be found in [27].
The thesis is organized as follows:

- In Chapter 2, we will introduce the Cloud Computing concepts and its three main paradigms: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS); for each of them some of the offerings available on the market will be presented. Then we will propose a classification of the literature approaches for runtime resource allocation, in terms of type of problem, solution found and discipline adopted.

- In Chapter 3 the optimization problem will be faced. We will start by stating the problem assumption and then we present a mathematical formulation of the problem. Then we also provide an extension of the first formulation. Finally the *receding horizon* approach is discussed.

- In Chapter 4 the reference system architecture, i.e. the MODAClouds runtime architecture, is introduced. The main components on which our receding horizon control leverages are discussed, focusing on the runtime requirements in order to implement the control.

- In Chapter 5 we will describe the tools relized in this work: we first introduce AMPL as the language adopted to implement the capacity allocation problem. The the *DesignTimeAdaptationPreparer* and the *AutoscalingReasoner*, which are the tools realized within the MODAClouds architecture, are described in details.

- Chapter 6 is devoted from one side to assess the quality of the described solution through experiments and numerical analysis, from

the other to evaluate performance of *AutoscalingReasoner* within the real scenario of the MODAClouds Runtime Platform. We will start presenting the results of the scalability analysis (in terms of time of execution of the problem) of the optimization model presented in Chapter 3 and a comparison with some heuristic decision methods. Then we consider the optimization model inside the reference runtime architecture. The Meeting in the Cloud (MiC) test application and the MiCMeter load injector are introduced. Then requirements to apply the receding horizon control are evaluated. Finally we describe the experimental results obtained on the MODAClouds Runtime Platform.

- In Chapter 7 some conclusion are drawn, underlying the achieved results and presenting future research directions.

CHAPTER 7

Conclusion and Future Work

In this thesis we proposed capacity allocation techniques able to minimize the cost of the allocated Cloud resources. Since the Cloud paradigm is getting every day more popular and data centers implementing the IaaS paradigm are spawning all around the world, the optimization of costs and resources is a central topic for SaaS providers. Indeed, in any time instant resources have to be allocated to handle effectively workload fluctuations, while providing QoS guarantees to end users.
The overall goal we addressed in the proposed technique is the minimization of the costs associated with the allocated virtual machine instances, while guaranteeing QoS constraints expressed as a threshold on the average response time. The capacity allocation techniques are formulated as optimization problems that, based on the workload prediction and the response time thresholds specified as SLAs, return as output the number of VM instances to allocate. In particular we first proposed a basic formulation of a capacity allocation technique and then we extend it considering also functionalities within the optimization model.
We performed an extensive analysis of our proposed solutions considering multiple workloads and system configurations. We analyzed the performance of our solution, exploiting the AMPL language and CPLEX MILP solver, comparing the achieved results with the ones obtained by the major techniques available in the literature or currently used by service providers. From these comparisons emerged that the proposed method scales linearly in time with respect to the number of class requests. The system was able to handle up to 160 different classes with time of execution of around 200 seconds, that it is an appreciable results since nowadays the mean number of application managed by SaaS are lower and the

overall execution time is less than the interval of 5 minutes, the finest grain time scale considered in our work. For what concerns the cost of the solution, our approach is cheaper than the heuristics of the state-of-the-art and it is competitive with the solution found by an oracle with perfect knowledge of the future. Moreover, we analyzed the solution performance in terms of SLA violations. From the SLA violation point of view, the algorithm has better performance in case of 5 minutes time scale. In this thesis we also employed the proposed capacity allocation technique within the MODAClouds Runtime Platform, in which we realized the Autoscaling Reasoner, that enacts an autoscaling mechanism based on the solution of the capacity allocation problem.
During this work we set up the MODAClouds Runtime Platform and evaluate the SDAs component, which provide the workload prediction required as input for the capacity allocation technique. The prediction error results on average to be between 30% and 40%, which is acceptable since the capacity allocation technique is proved to be effective also with an high level of noise, that corresponds to a significant prediction error.
Once SDAs performance and accuracy have been assessed and the Autoscaling Reasoner implemented, we evaluated the performance of the autoscaling mechanism within the MODAClouds Runtime Platform. We performed test with a timescale of 5 minutes and a window obesrvation of 5 steps. Furthermore, we consider single class problems, due to some limitations imposed by the current implementation of the MODACLouds Runtime Platform. Results show that the Autoscaling Reasoner is able to follow workload fluctuation employing scale in and out actions. We also observed that the percentage of SLA violations s between the 10% and 25%, which we can assume to be a good result, also considering that since MODAClouds is an ongoing research project, there are some components that are still under development or only at a prototypical stage and so further improvements could be achieved.
In this sense, future works will be first devoted to continue the integration process among all the components of the MODAClouds runtime platform. From one side, some recent changes in the MODACLouds Monitoring Platform allow us to evaluate also the Autoscaling Reasoner performance in the case of multiple classes. Furthermore, since CloudML can process one scale command at a time, we will also perform some tests using a timescale of 10 minutes, since we expect to achieve better performance in this case. Also the effects obtained from varying the number of timesteps in the observation windows could be interesting to study. Finally, since we proposed an extended formulation of the capacity allocation technique, which considers also functionalities in the optimization

model and from which we expect better performance with the respect to the basic formulation evaluated in this thesis, future efforts could be devoted to provide an implementation also for this extended version of the capacity allocation technique in order to compare its performance with the results obtained in this thesis.

Bibliography

[1] URL: http://www-01.ibm.com/support/knowledgecenter/SSSA5P_12.6.0/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.html.

[2] Bernardetta Addis, Danilo Ardagna, Barbara Panicucci, Mark S. Squillante, and Li Zhang. "A Hierarchical Approach for the Resource Management of Very Large Cloud Platforms". In: *IEEE Transactions on Dependable and Secure Computing* 10.5 (2013), pp. 253–272. ISSN: 1545-5971. DOI: http://doi.ieeecomputersociety.org/10.1109/TDSC.2013.4.

[3] L. Agostinho, G. Feliciano, L. Olivi, E. Cardozo, and E. Guimaraes. "A Bio-inspired Approach to Provisioning of Virtual Resources in Federated Clouds". In: *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*. 2011, pp. 598 –604.

[4] A. Aleti, B. Buhnova, L. Grunske, A. Koziolek, and I. Meedeniya. "Software Architecture Optimization Methods: A Systematic Literature Review". In: *Software Engineering, IEEE Transactions on* PP.99 (2012), p. 1. ISSN: 0098-5589.

[5] Alistair Croll. *Cloud performance from the end user perspective*.

[6] J. Almeida, V. Almeida, D. Ardagna, I. Cunha, C. Francalanci, and M. Trubian. "Joint admission control and resource allocation in virtualized servers". In: *Journal of Parallel and Distributed Computing* 70.4 (2010), pp. 344 –362.

[7] Amazon Inc. *Amazon Elastic Cloud*. http://aws.amazon.com/ec2/.

[8] Amazon Inc. *Amazon Web Services*. http://aws.amazon.com/.

[9] Amazon Inc. *AWS Elastic Beanstalk*. http://aws.amazon.com/elasticbeanstalk/.

[10] Amazon Inc. *Elastic Load Balancing*. http://aws.amazon.com/elasticloadbalancing/.

[11] *Amazon Web Services*. http://aws.amazon.com/.

[12] *AMPL. Ampl modeling language for mathematical programming*. http://www.ampl.com/.

[13] D. Ardagna, B. Panicucci, M. Trubian, and L. Zhang. "Energy-Aware Autonomic Resource Allocation in Multitier Virtualized Environments". In: *IEEE Trans. Services Computing* 5.1 (2012), pp. 2–19.

[14] D. Ardagna, S. Casolari, and B. Panicucci. "Flexible Distributed Capacity Allocation and Load Redirect Algorithms for Cloud Systems". In: *Proc. of the 4th International Conference on Cloud Computing (IEEE Cloud 2011)*. 2011.

[15] Danilo Ardagna, Sara Casolari, Michele Colajanni, and Barbara Panicucci. "Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems". In: *Journal of Parallel and Distributed Computing* 72.6 (2012), pp. 796 –808. ISSN: 0743-7315.

[16] Jeff Barr. *Host Your Web Site In The Cloud: Amazon Web Services Made Easy Amazon EC2 Made Easy*. 1st. Sitepoint, 2010.

[17] R. Van den Bossche, K. Vanmechelen, and J. Broeckhove. "Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads". In: *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*. 2010, pp. 228 –235.

[18] Nicolò Maria Calcavecchia, Bogdan Alexandru Caprarescu, Elisabetta Di Nitto, Daniel J. Dubois, and Dana Petcu. "DEPAS: A Decentralized Probabilistic Algorithm for Auto-Scaling". In: *CoRR* abs/1202.2509 (2012).

[19] Marco Caldirola. "Tecniche di Resource Allocation per sistemi virtualizzati di larga scala." MA thesis. Politecnico di Milano, 2010.

[20] Bogdan Alexandru Caprarescu, Nicolò Maria Calcavecchia, Elisabetta Di Nitto, and Daniel J. Dubois. "SOS Cloud: Self-organizing Services in the Cloud". In: *BIONETICS*. 2010, pp. 48–55.

[21] G. Casale, Ningfang Mi, L. Cherkasova, and E. Smirni. "Dealing with Burstiness in Multi-Tier Applications: Models and Their Parameterization". In: *Software Engineering, IEEE Transactions on* 38.5 (2012), pp. 1040 –1053.

[22] Giuliano Casale and Mirco Tribastone. "Fluid Analysis of Queueing in Two-Stage Random Environments". In: *QEST*. 2011, pp. 21–30.

[23] Giuliano Casale and Mirco Tribastone. "Modelling Exogenous Variability in Cloud Deployments". In: *SIGMETRICS Perform. Eval. Rev.* 40.4 (Apr. 2013), pp. 73–82. ISSN: 0163-5999. DOI: 10.1145/2479942.2479951. URL: http://doi.acm.org/10.1145/2479942.2479951.

[24] S. Casolari and M. Colajanni. "Short-term prediction models for server management in Internet-based contexts". In: *Decision Support Systems* 48.1 (2009).

[25] Sivadon Chaisiri, Bu-Sung Lee, and Dusit Niyato. "Optimization of resource provisioning cost in cloud computing". English. In: *IEEE Transactions on Services Computing* 5.2 (2012), pp. 164 –177. ISSN: 19391374.

[26] Wei Chen, Xiaoqiang Qiao, Jun Wei, and Tao Huang. "A Profit-Aware Virtual Machine Deployment Optimization Framework for Cloud Platform Providers". In: *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. 2012, pp. 17 –24.

[27] G. Casale D. Petcu P. Mohagheghi S. Mosser P.Matthews A. Gericke C. Ballagny F. D'Andria C.S. Nechifor C. Sheridan D. Ardagna E. Di Nitto. "MODACLOUDS: A Model-Driven Approach for the Design and Execution of Applications on Multiple Clouds". In: *Modeling in Software Engineering (MISE), 2012 ICSE Workshop* (2012), pp. 50 –56. ISSN: 2156-788.

[28] Brian Dougherty, Jules White, and Douglas C. Schmidt. "Model-driven auto-scaling of green cloud computing infrastructure". English. In: *Future Generation Computer Systems* 28.2 (2012), pp. 371 –378. ISSN: 0167739X.

[29] S. Dutta, S. Gera, A. Verma, and B. Viswanathan. "SmartScale: Automatic Application Scaling in Enterprise Clouds". In: *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. 2012, pp. 221 –228.

[30] W. Ellens, M. Zivkovic, J. Akkerboom, R. Litjens, and H. van den Berg. "Performance of Cloud Computing Centers with Multiple Priority Classes". In: *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. 2012, pp. 245 –252.

[31] M. Ferber, T. Rauber, M.H.C. Torres, and T. Holvoet. "Resource Allocation for Cloud-Assisted Mobile Applications". In: *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. 2012, pp. 400 –407.

[32] *Flexyscale*. http://www.flexiscale.com/.

[33] Anshul Gandhi, Varun Gupta, Mor Harchol-Balter, and Michael A. Kozuch. "Optimality analysis of energy-performance trade-off for server farm management". In: *Perform. Eval.* (2010), pp. 1155–1171.

[34] Anshul Gandhi, Varun Gupta, Mor Harchol-Balter, and Michael A. Kozuch. "Optimality analysis of energy-performance trade-off for server farm management". English. In: vol. 67. 11. P.O. Box 211, Amsterdam, 1000 AE, Netherlands, 2010, pp. 1155 –1171.

[35] Filippo Giove, Davide Longoni, Majid Shokrolahi Yancheshmeh, Danilo Ardagna, and Elisabetta Di Nitto. "An Approach for the Development of Portable Applications on PaaS Clouds." In: *CLOSER*. Ed. by Frédéric Desprez, Donald Ferguson, Ethan Hadar, Frank Leymann, Matthias Jarke, and Markus Helfert. SciTePress, 2013, pp. 591–601. ISBN: 978-989-8565-52-5. URL: http://dblp.uni-trier.de/db/conf/closer/closer2013.html#GioveLYAN13.

[36] *GoGrid*. http://www.gogrid.com/.

[37] *Google App Engine*. https://developers.google.com/appengine/.

[38] *Google Apps for Business*. http://www.google.com/enterprise/apps/business/.

[39] *Google Compute Engine*. https://cloud.google.com/products/compute-engine.

[40] *Google Inc.* http://www.google.com/about/company/. URL: http://www.google.com/about/company/ (visited on 2012).

[41] H. Goudarzi and M. Pedram. "Multi-dimensional SLA-Based Resource Allocation for Multi-tier Cloud Computing Systems". In: *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. 2011, pp. 324 –331.

[42] M. Hadji and D. Zeghlache. "Minimum Cost Maximum Flow Algorithm for Dynamic Resource Allocation in Clouds". In: *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. 2012, pp. 876 –882.

[43] Amazon Inc. *Amazon EC2 Pricing*. http://aws.amazon.com/ec2/pricing/.

[44] Waheed Iqbal, Matthew N. Dailey, David Carrera, and Paul Janecek. "Adaptive resource provisioning for read intensive multi-tier applications in the cloud". In: *Future Gener. Comput. Syst.* 27.6 (June 2011), pp. 871–879. ISSN: 0167-739X.

[45] Yang Jie, Qiu Jie, and Li Ying. "A Profile-Based Approach to Just-in-Time Scalability for Cloud Applications". In: *Cloud Computing, 2009. CLOUD '09. IEEE International Conference on*. 2009, pp. 9 – 16.

[46] *Kernel Based Virtual Machine*. http://www.linux-kvm.org/.

[47] H. Khazaei, J. Misic, V. Misic, and S. Rashwand. "Analysis of a Pool Management Scheme for Cloud Computing Centers". In: *Parallel and Distributed Systems, IEEE Transactions on* PP.99 (2012), p. 1. ISSN: 1045-9219. DOI: 10.1109/TPDS.2012.182.

[48] K. Konstanteli, T. Cucinotta, K. Psychas, and T. Varvarigou. "Admission Control for Elastic Cloud Services". In: *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. 2012, pp. 41 –48.

[49] Sanjay Kumar, Vanish Talwar, Vibhore Kumar, Parthasarathy Ranganathan, and Karsten Schwan. "Loosely coupled coordinated management in virtualized data centers". In: *Cluster Computing* 14.3 (2011), pp. 259–274. ISSN: 1386-7857. URL: http://dx.doi.org/10.1007/s10586-010-0124-9.

[50] D. Kusic and N. Kandasamy. "Risk-Aware Limited Lookahead Control for Dynamic Resource Provisioning in Enterprise Computing Systems". In: *Autonomic Computing, 2006. ICAC '06. IEEE International Conference on*. 2006, pp. 74 –83.

[51] D. Kusic, N. Kandasamy, and Guofei Jiang. "Approximation Modeling for the Online Performance Management of Distributed Computing Systems". In: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 38.5 (2008), pp. 1221 –1233. ISSN: 1083-4419.

[52] Dara Kusic, Jeffrey O. Kephart, James E. Hanson, Nagarajan Kandasamy, and Guofei Jiang. "Power and performance management of virtualized computing environments via lookahead control". In: *Cluster Computing* 12.1 (2009), pp. 1–15.

[53] Laurent Lefevre and Anne-Cecile Orgerie. "Designing and evaluating an energy efficient Cloud". In: *Journal of Supercomputing* 51.3 (2010), pp. 352–373.

[54] Ming Mao and Marty Humphrey. "Auto-scaling to minimize cost and meet application deadlines in cloud workflows". In: *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. SC '11. Seattle, Washington: ACM, 2011, 49:1–49:12. ISBN: 978-1-4503-0771-0.

[55] Yiduo Mei, Ling Liu, Xing Pu, S. Sivathanu, and Xiaoshe Dong. "Performance Analysis of Network I/O Workloads in Virtualized Data Centers". In: *Services Computing, IEEE Transactions on* 6.1 (2013), pp. 48–63. ISSN: 1939-1374. DOI: 10.1109/TSC.2011.36.

[56] Peter Mell and Tim Grance. *The NIST Definition of Cloud Computing*. Tech. rep. July 2009. URL: http://www.csrc.nist.gov/groups/SNS/cloud-computing/.

[57] *Microsoft Corporation*. http://www.microsoft.com/.

[58] *Microsoft Office 365*. http://office365.microsoft.com/.

[59] *Microsoft Windows Azure*. http://www.windowsazure.com/.

[60] *Microsoft Windows Azure Virtual Machines*. http://www.windowsazure.com/en-us/home/features/virtual-machines/.

[61] Amit Nathani, Sanjay Chaudhary, and Gaurav Somani. "Policy based resource allocation in IaaS cloud". English. In: vol. 28. 1. P.O. Box 211, Amsterdam, 1000 AE, Netherlands, 2012, pp. 94 –103.

[62] Elisabetta Di Nitto, Daniel J. Dubois, and Raffaela Mirandola. "On exploiting decentralized bio-inspired self-organization algorithms to develop real systems". In: *SEAMS*. 2009, pp. 68–75.

[63] K.Y. Oktay, V. Khadilkar, B. Hore, M. Kantarcioglu, S. Mehrotra, and B. Thuraisingham. "Risk-Aware Workload Distribution in Hybrid Clouds". In: *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. 2012, pp. 229 –236.

[64] *OMNeT++ Discrete Event Simulation System*. Tech. rep. 2014. URL: http://www.csrc.nist.gov/groups/SNS/cloud-computing/.

[65] *Onlive*. http://www.onlive.com/.

[66] Vern Paxson and Sally Floyd. "Wide Area Traffic: The Failure of Poisson Modeling". In: *IEEE/ACM Trans. Netw.* 3.3 (June 1995), pp. 226–244. ISSN: 1063-6692.

[67] *Rackspace*. http://www.rackspace.com/.

[68] N. Roy, A. Dubey, and A. Gokhale. "Efficient Autoscaling in the Cloud Using Predictive Models for Workload Forecasting". In: *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. 2011, pp. 500 –507.

[69] *Salesforce*. http://www.force.com/.

[70] *SAP*. http://www.sap.com/.

[71] SINTEF. http://cloudml.org/.

[72] Giuseppe Valetto, Paul L. Snyder, Daniel J. Dubois, Elisabetta Di Nitto, and Nicolò Maria Calcavecchia. "A Self-Organized Load-Balancing Algorithm for Overlay-Based Decentralized Service Networks". In: *SASO*. 2011, pp. 168–177.

[73] Emanuele Della Valle. http://streamreasoning.org/.

[74] Eveline Veloso, Virgílio Almeida, Wagner Meira Jr., Azer Bestavros, and Shudong Jin. "A Hierarchical Characterization of a Live Streaming Media Workload". In: *IEEE/ACM Trans. Netw.* 14.1 (Feb. 2006), pp. 133–146. ISSN: 1063-6692.

[75] *VMware Inc.* http://www.vmware.com/.

[76] W3C. http://www.w3.org/RDF/.

[77] Lijuan Wang and Jun Shen. "Towards Bio-inspired Cost Minimisation for Data-Intensive Service Provision". In: *Services Economics (SE), 2012 IEEE First International Conference on*. 2012, pp. 16 –23.

[78] W. Wang, B. Li, and B. Liang. "Towards Optimal Capacity Segmentation with Hybrid Cloud Pricing". In: *ICDCS*. 2012.

[79] Guiyi Wei, Athanasios V. Vasilakos, Yao Zheng, and Naixue Xiong. "A game-theoretic method of fair resource allocation for cloud computing services". English. In: *Journal of Supercomputing* 54.2 (2010), pp. 252–269.

[80] A. Wolke and G. Meixner. "TwoSpot: A Cloud Platform for Scaling Out Web Applications Dynamically". In: *ServiceWave*. 2010.

[81] Wook Hyun Kwon, Soo H. Han. *Receding Horizon Predictive Control: Model Predictive Control for State Models*. Ed. by Springer. 2005.

[82] *Xen Hypervisor*. http://www.xen.org/.

[83] Z. Xiao, Q. Chen, and H. Luo. "Automatic Scaling of Internet Applications for Cloud Computing Services". In: *Computers, IEEE Transactions on* PP.99 (2012), p. 1. ISSN: 0018-9340.

[84] S. Zaman and D. Grosu. "An Online Mechanism for Dynamic VM Provisioning and Allocation in Clouds". In: *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. 2012, pp. 253 –260.

[85] L. Zhang, X. Meng, S. Meng, and J. Tan. "K-Scope: Online Performance Tracking for Dynamic Cloud Applications". In: *ICAC*. 2013.

[86] Qi Zhang, Lu Cheng, and Raouf Boutaba. "Cloud computing: state-of-the-art and research challenges". In: *J. Internet Services and Applications* 1.1 (2010), pp. 7–18.

[87] X. Zhu, D. Young, B. Watson, Z. Wang, J. Rolia, S. Singhal, B. McKee, C. Hyser, D.Gmach, R. Gardner, T. Christian, and L. Cherkasova: "1000 Islands: An Integrated Approach to Resource Management for Virtualized Data Centers". In: *Journ. of Cluster Computing* 12.1 (2009), pp. 45–57.