POLITECNICO DI MILANO

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

Multiple time-scale Auto-Scaling Algorithms for Multi-Cloud IaaS Systems

Advisor: Prof. Danilo ARDAGNA
Co-Advisor: Dr. Michele CIAVOTTA

Master Thesis by:
Davide MOLINARI - 771111

Academic Year 2011/2012

Contents

1 Introduction **5**

2 State of the art **9**
2.1 Cloud Computing Overview 9
2.1.1 Infrastructure-as-a-Service (IaaS) 13
2.1.2 Platform-as-a-Service (Paas) 18
2.1.3 Software-as-a-Service (SaaS) 21
2.2 Cloud Computing and run-time research challenges 21
2.2.1 Problem 22
2.2.2 Solution 23
2.2.3 Discipline 24
2.3 Overview of related research approaches 25
2.3.1 Evaluation Criteria 35

3 Capacity Allocation Algorithm **39**
3.1 Problem statement 39
3.2 Design assumptions 42
3.3 Optimization Problem Formulation 46
3.3.1 Long-term problem 46
3.3.2 Short-term problem 51

4 Tools **57**
4.1 AMPL 57
4.2 CPLEX 58
4.2.1 CPLEX algorithms for continuous optimization 58
4.2.2 CPLEX for integer programming 59
4.3 SPECweb 2005 60
4.4 JMeter 62
4.5 SPECweb deployment 64
4.5.1 SPECweb tests 64
4.5.2 JMeter Extension 67

4.5.3 SPECmeter 68
4.6 Optimization Tools 70
4.6.1 Optimization Tools class diagram 70
4.6.2 Optimization Tools sequence diagram 72

5 Experimental Results **75**
5.1 Design of experiments 75
5.1.1 Performance parameters 75
5.1.2 Traffic generation 76
5.2 Scalability analysis 77
5.2.1 Long-term Allocation 80
5.2.2 Short-term Allocation 80
5.3 Heuristics 82
5.3.1 Heuristic 1 82
5.3.2 Heuristic 2 87
5.4 Multiple time-scale Analysis 88
5.4.1 Solutions under analysis 89
5.4.2 Solutions Cost 90
5.4.3 SLA violations 100

6 Conclusions **115**

List of Figures

2.1 Cloud service models 12
2.2 Taxonomy for optimization approaches. 26

3.1 Time-scale example. 42
3.2 System performance model 43
3.3 Cloud Infrastructure model. 47
3.4 Increasing workload. 53
3.5 Decreasing workload. 54
3.6 VMs counting 56

4.1 CPLEX mixed integer algorithm: The search tree. 60
4.2 SPECweb 2005 test diagram. 64
4.3 A simple two-state Markov chain. 66
4.4 SPECweb E-commerce test Markov chain. 66
4.5 Markov chain example in JMeter. 67
4.6 System test architecture. 68
4.7 Test automation sequence diagram. 69
4.8 Analysis class diagram. 71
4.9 Analysis sequence diagram. 73

5.1 Queueing delay time. 76
5.2 Service time. 76
5.3 Daily time distribution of the original trace of requests. 78
5.4 Randomly generated daily time distribution of request. 78
5.5 Long-term Problem scalability. 81
5.6 Short-term Problem scalability. 82
5.7 Hysteresis utilization control. 84
5.8 Solution cost, 5 minutes time scale, normal workload and low noise level. 92
5.9 Solution cost, 5 minutes time scale, normal workload and high noise level. 93

5.10 Solution cost, 5 minutes time scale, spike workload and low noise level. 94
5.11 Solution cost, 5 minutes time scale, spike workload and high noise level. 95
5.12 Solution cost, 10 minutes time scale, normal workload and low noise level. 96
5.13 Solution cost, 10 minutes time scale, normal workload and high noise level. 97
5.14 Solution cost, 10 minutes time scale, spike workload and low noise level. 98
5.15 Solution cost, 10 minutes time scale, spike workload and high noise level. 99

List of Tables

2.1 Amazon EC2 Instances Types 15
2.2 Problem Category: Perspective and Quality Attribute sub-categories. 32
2.3 Problem Category: Dimensionality and Constraint sub-categories. 33
2.4 Solution Category: Type sub-category. 33
2.5 Solution Category: Degrees of Freedom sub-category. 34
2.6 Solution Category: Architecture Representation and Optimization Strategy. 34
2.7 Solution Category: Constraint Handling and Time scale. . . . 34
2.8 Discipline Category. 35

3.1 Parameters for Capacity Allocation. 45
3.2 Decision variables for Capacity Allocation. 45
3.3 Additional global parameters. 55

5.1 Performance parameters. 79
5.2 Cost parameters. 79
5.3 Long-term CA Problem Execution Time (sec). 80
5.4 Long-term CA Problem Execution Time (sec). 80
5.5 Short-term CA Problem Execution Time (sec). 81
5.6 Short-term CA Problem Execution Time (sec). 81
5.7 Noise level adopted. 89
5.8 Solutions cost percentage differences for 5 minutes time scale, normal traffic and low noise. 92
5.9 Solutions cost percentage differences for 5 minutes time scale, normal traffic and high noise. 93
5.10 Solutions cost percentage differences for 5 minutes time scale, spike traffic and low noise. 94
5.11 Solutions cost percentage differences for 5 minutes time scale, spike traffic and high noise. 95

5.12 Solutions cost percentage differences for 10 minutes time scale, normal traffic and low noise. 96
5.13 Solutions cost percentage differences for 10 minutes time scale, normal traffic and high noise. 97
5.14 Solutions cost percentage differences for 10 minutes time scale, spike traffic and low noise. 98
5.15 Solutions cost percentage differences for 10 minutes time scale, spike traffic and high noise. 99
5.16 M/G/1 equilibrium percentage violations for 5 minutes time scale, normal traffic and low noise. 102
5.17 M/G/1 equilibrium percentage violations for 5 minutes time scale, normal traffic and high noise. 102
5.18 M/G/1 equilibrium percentage violations for 5 minutes time scale, spike traffic and low noise. 103
5.19 M/G/1 equilibrium percentage violations for 5 minutes time scale, spike traffic and high noise. 103
5.20 M/G/1 equilibrium percentage violations for 10 minutes time scale, normal traffic and low noise. 104
5.21 M/G/1 equilibrium percentage violations for 10 minutes time scale, normal traffic and high noise. 104
5.22 M/G/1 equilibrium percentage violations for 10 minutes time scale, spike traffic and low noise. 105
5.23 M/G/1 equilibrium percentage violations for 10 minutes time scale, spike traffic and high noise. 105
5.24 Response Time percentage violations for 5 minutes time scale, normal traffic and low noise. 106
5.25 Response Time percentage violations for 5 minutes time scale, normal traffic and high noise. 106
5.26 Response Time percentage violations for 5 minutes time scale, spike traffic and low noise. 107
5.27 Response Time percentage violations for 5 minutes time scale, spike traffic and high noise. 107
5.28 Response Time percentage violations for 10 minutes time scale, normal traffic and low noise. 108
5.29 Response Time percentage violations for 10 minutes time scale, normal traffic and high noise. 108
5.30 Response Time percentage violations for 10 minutes time scale, spike traffic and low noise. 109
5.31 Response Time percentage violations for 10 minutes time scale, spike traffic and high noise. 109

5.32 Saturations percentage for 5 minutes time scale, normal traffic and low noise. 110
5.33 Saturations percentage for 5 minutes time scale, normal traffic and high noise. 110
5.34 Saturations percentage for 5 minutes time scale, spike traffic and low noise. 111
5.35 Saturations percentage for 5 minutes time scale, spike traffic and high noise. 111
5.36 Saturations percentage for 10 minutes time scale, normal traffic and low noise. 112
5.37 Saturations percentage for 10 minutes time scale, normal traffic and high noise. 112
5.38 Saturations percentage for 10 minutes time scale, spike traffic and low noise. 113
5.39 Saturations percentage for 10 minutes time scale, spike traffic and high noise. 113

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 at multiple providers, while guaranteeing Quality of Service (QoS) constraints. We deal with this problem on two different level of optimization: at the first layer, we provide the distribution of workload over multiple infrastructure provider and then at the second layer we implement capacity allocation of multiple class of requests at each provider on a short-term time scale. The second layer implements a short term receding horizon control. The overall goal is to analyze the best time scale length for the capacity allocation algorithm in order to minimize cost and providing QoS guarantees. We have performed also an extensive evaluation of our solution with multiple heuristics provided in the literature.

Results have shown that the most appropriate time scale is 5 minute for spiky workload and 10 minutes for smooth traffic conditions from the economical point of view, also with acceptable percentage of QoS violations. Furthermore, our solutions are very close to the ones found by an oracle with perfect knowledge of the future.

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.

Questa tesi si propone di sviluppare tecniche di allocazione delle risorse in grado di minimizzare i costi per le risorse allocate su più provider e al contempo garantire la QoS. Il problema viene affrontato attraverso due livelli di ottimizzazione: il primo livello consiste nella distribuzione del traffico in ingresso al sistema su più provider, mentre nel secondo si è implementato una tecnica di allocazione di più classi di richieste ad ogni provider, considerando intervalli di tempo brevi. Viene utilizzata la tecnica di controllo Receding Horizon. L'obiettivo è 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. Abbiamo eseguito anche una esaustiva valutazione della nostra soluzione attraverso il confronto con le euristiche proposte dalla letteratura.

I risultati dimostrano che l'intervallo di tempo più appropriato è di 5 minuti per traffico altamente variabile e di 10 per traffico uniforme, con una percentuale di violazione della QoS accettabile. Inoltre, la nostra soluzione è risultata confrontabile con quella ottenuta da un oracolo, che ha perfetta conoscenza del futuro.

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 his applications as Web Service hosted by multiple Infrastructure as a Service (IaaS) providers, thanks to a software layer developed within the MODAClouds[1] project [14], which aims to develop a software layer for deploying and migrating application transparently at run-time on multiple clouds.

This thesis aims to develop capacity allocation techniques able to minimize the cost of the provided Cloud resources at multiple providers, while

[1] `http://www.modaclouds.eu/`

guaranteeing quality of service constraints. We deal with this problem on two different level of optimization: at the first layer, we provide the distribution of workload over multiple infrastructure provider and then at the second layer we implement capacity allocation of multiple class of requests at each provider on a short-term time scale. Furthermore, the second layer implements a short-term receding horizon control. The overall goal is to analyze the best time scale length for the capacity allocation algorithm in order to minimize cost and guaranteeing quality of service constraints. We will perform also an extensive evaluation of our solution with multiple heuristics provided in the literature.

The thesis is organized as follows:

- In Chapter 2, we will introduce the Cloud Computing concepts and its three 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 proposed a classification of the literature approaches in terms of type of problem, solution found and discipline adopted. Finally, we will review the main works on run-time optimization and we will analyze them, according to some criteria of evaluation.

- In Chapter 3 the Capacity Allocation problem will be faced. We will start by stating the problem, then we will introduce the various assumptions and present the problem formulation, divided in long-term and short-term problem.

- In Chapter 4 we will describe all the tools used in our thesis work: At first we will present AMPL, the modeling language used to define the capacity allocation optimization problems (in order to be able to solve them using the CPLEX solver) and to implement the heuristics used in our solution. Then we will introduce SPECweb2005, JMeter and a workload injector we developed in order to automate testing in a Cloud platform.

- Chapter 5 is dedicated to assess the quality of our solution through experiments and numerical analysis. We will start presenting the results of the scalability analysis (in terms of time of execution of each problem) of the capacity allocation models presented in Chapter 3. Then we compare the results achieved by our solution (both in terms of costs and SLA violations) with the ones of the current state of the art techniques.

- In Chapter 6 are presented the work's conclusion, underling the achieved results and presenting future research directions.

Chapter 6

Conclusions

In this thesis we proposed capacity allocation techniques able to minimize the cost of the allocated Cloud resources at multiple IaaS providers. Since the Cloud paradigm is getting day by day more popular and the IaaS provider centers 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 the end users. The overall goal we addressed in our thesis 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.

In our work we proposed a formulation of two optimization problems, focused to provide a workload distribution on hourly basis over different IaaS providers and to achieve capacity allocation with cost minimization, while guaranteeing the respect of the SLA.

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 the CPLEX MILP solver, comparing the achieved results with the ones which can be 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 the class request, both for the long-term and short-term problems. The system was able to handle up to 160 different classes with time of execution of around 160 seconds for the long-term algorithm and of around 200 seconds for the short-term one, 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. In condition of normal traffic, the best time scale length is 10 minutes, while with spiky workloads the most appropriate time scale is 5 minutes. Moreover, we analyzed the solution performance in terms of SLA violations. According to these metrics we were able to verify the benefit of the adoption of the Receding Horizon Control, because we could register a substantial improvement of the performance thanks to the increases of the prediction forward step number. From the SLA violation point of view, the algorithm has better performance in case of 5 minutes time scale, that with receding horizon technique guarantees a maximum of 8 minutes of response time violation over 24 hours. Finally, from these results we can notice that a window of observation of three steps, that is 15 or 30 minutes according to the time scale chosen, represents the best configuration for the control method.

Future work will be devoted to the development of an adaptive approach that will be able to switch between different time scales according to the workload condition: A shorter interval of time in case of a less accurate prediction or a heavy traffic condition and a longer one in a smooth situation. Finally, experiments in a real prototype environment will be also performed.

Bibliography

[1] V. Abhishek, I.A. Kash, and P. Key. Fixed and market pricing for cloud services. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 157 –162, march 2012.

[2] B. Addis, D. Ardagna, B. Panicucci, M. Squillante, and L. Zhang. A hierarchical approach for the resource management of very large cloud platforms. *Dependable and Secure Computing, IEEE Transactions on*, PP(99):1, 2013.

[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*, pages 598 –604, dec. 2011.

[4] A. Aleti, B. Buhnova, L. Grunske, A. Koziolek, and I. Meedeniya. Software architecture optimization methods: A systematic literature review. *Software Engineering, IEEE Transactions on*, PP(99):1, 2012.

[5] J. Almeida, V. Almeida, D. Ardagna, I. Cunha, C. Francalanci, and M. Trubian. Joint admission control and resource allocation in virtualized servers. *Journal of Parallel and Distributed Computing*, 70(4):344 – 362, 2010.

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

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

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

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

[10] Amazon Inc. Amazon Price Scheme, 2013.

[11] AMPL. A Modeling Language for Mathematical Programming. http://www.ampl.com.

[12] Apache jMeter. http://jmeter.apache.org/.

[13] 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). To Appear*, 2011.

[14] D. Ardagna, E. Di Nitto, P. Mohagheghi, S. Mosser, C. Ballagny, F. D'Andria, G. Casale, P. Matthews, C.-S. Nechifor, D. Petcu, A. Gericke, and C. Sheridan. Modaclouds: A model-driven approach for the design and execution of applications on multiple clouds. In *Modeling in Software Engineering (MISE), 2012 ICSE Workshop on*, pages 50 –56, june 2012.

[15] D. Ardagna, B. Panicucci, M. Trubian, and L. Zhang. Energy-Aware Autonomic Resource Allocation in Multi-tier Virtualized Environments. *IEEE Trans. on Services Computing,* available on line.

[16] Danilo Ardagna, Sara Casolari, Michele Colajanni, and Barbara Panicucci. Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems. *Journal of Parallel and Distributed Computing*, 72(6):796 – 808, 2012.

[17] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Communications of the ACM*, 53(4):50 – 58, 2010.

[18] Bitcurrent. Cloud Computing Performance. Technical report, http://www.bitcurrent.com/new-report-on-cloud-performance/, 2010.

[19] Nicolò Maria Calcavecchia, Bogdan Alexandru Caprarescu, Elisabetta Di Nitto, Daniel J. Dubois, and Dana Petcu. Depas: A decentralized probabilistic algorithm for auto-scaling. *CoRR*, abs/1202.2509, 2012.

[20] Marco Caldirola. Tecniche di resource allocation per sistemi virtualizzati di larga scala. Master's thesis, Politecnico di Milano, 2010.

[21] Bogdan Alexandru Caprarescu, Nicolò Maria Calcavecchia, Elisabetta Di Nitto, and Daniel J. Dubois. Sos cloud: Self-organizing services in the cloud. In *BIONETICS*, pages 48–55, 2010.

[22] G. Casale, Ningfang Mi, L. Cherkasova, and E. Smirni. Dealing with burstiness in multi-tier applications: Models and their parameterization. *Software Engineering, IEEE Transactions on*, 38(5):1040 –1053, sept.-oct. 2012.

[23] Sivadon Chaisiri, Bu-Sung Lee, and Dusit Niyato. Optimization of resource provisioning cost in cloud computing. *IEEE Transactions on Services Computing*, 5(2):164 – 177, 2012.

[24] 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*, pages 17 –24, june 2012.

[25] CPLEX. IBM ILOG CPLEX Optimization Studio. http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/.

[26] Brian Dougherty, Jules White, and Douglas C. Schmidt. Model-driven auto-scaling of green cloud computing infrastructure. *Future Generation Computer Systems*, 28(2):371 – 378, 2012.

[27] Dave Durkee. Why cloud computing will never be free. *Communications of the ACM*, 53(5):62 – 69, 2010. Cloud computing;.

[28] 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*, pages 221 –228, june 2012.

[29] 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*, pages 245 –252, june 2012.

[30] 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*, pages 400 –407, june 2012.

[31] FreshBooks. Freshbooks - online invoicing, time tracking & billing software - http://www.freshbooks.com/.

[32] Anshul Gandhi, Varun Gupta, Mor Harchol-Balter, and Michael A. Kozuch. Optimality analysis of energy-performance trade-off for server farm management. volume 67, pages 1155 – 1171, P.O. Box 211, Amsterdam, 1000 AE, Netherlands, 2010.

[33] GoGrid. Gogrid - www.gogrid.com.

[34] Google Inc. Google app engine - google code. http://code.google.com/intl/en-en/appengine/.

[35] Google Inc. Google apps for business — official website. http://www.google.com/apps/intl/en/business/index.html.

[36] Google Inc. Google inc. http://www.google.com/about/company/.

[37] 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*, pages 324 –331, july 2011.

[38] Bernd Grobauer, Tobias Walloschek, and Elmar Stocker. Understanding cloud computing vulnerabilities. *IEEE Security and Privacy*, 9(2):50 – 57, 2011.

[39] 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*, pages 876 –882, june 2012.

[40] Amazon Inc. Amazon simple storage service (amazon s3). http://aws.amazon.com/s3/.

[41] NetSuite Inc. Business software, erp software, business accounting software, crm and erp business software-netsuite. http://www.netsuite.com/portal/home.shtml.

[42] Terremark Worldwide Inc. Terremark cloud computing. http://www.terremark.com/services/cloudcomputing.aspx.

[43] Waheed Iqbal, Matthew N. Dailey, David Carrera, and Paul Janecek. Adaptive resource provisioning for read intensive multi-tier applications in the cloud. *Future Gener. Comput. Syst.*, 27(6):871–879, June 2011.

[44] I. Iyoob, E. Zarifoglu, and T.B. Dieker. Cloud computing operations research. *INFORMS Service Science*.

[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*, pages 9 –16, sept. 2009.

[46] JMeter Plugins. https://code.google.com/p/jmeter-plugins/.

[47] H. Khazaei, J. Misic, V. Misic, and S. Rashwand. Analysis of a pool management scheme for cloud computing centers. *Parallel and Distributed Systems, IEEE Transactions on*, PP(99):1, 2012.

[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*, pages 41 –48, june 2012.

[49] 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*, pages 74 – 83, june 2006.

[50] D. Kusic, N. Kandasamy, and Guofei Jiang. Approximation modeling for the online performance management of distributed computing systems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(5):1221 –1233, oct. 2008.

[51] Dara Kusic, Jeffrey O. Kephart, James E. Hanson, Nagarajan Kandasamy, and Guofei Jiang. Power and performance management of virtualized computing environments via lookahead control. *Cluster Computing*, 12(1):1–15, 2009.

[52] Laurent Lefevre and Anne-Cecile Orgerie. Designing and evaluating an energy efficient cloud. *Journal of Supercomputing*, 51(3):352–373, 2010. Date revised - 2012-04-01; Pages - 352-373; Journal of Supercomputing [J Supercomput]. Vol. 51, no. 3, pp. 352-373. Mar 2010; 0920-8542.

[53] 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, pages 49:1–49:12, New York, NY, USA, 2011. ACM.

[54] Markov4jmeter. http://se.informatik.uni-kiel.de/markov4jmeter/.

[55] Peter Mell and Tim Grace. The nist definition of cloud computing. Techical report, July 2009.

[56] Microsoft. Hotmail - http://www.hotmail.com/.

[57] Microsoft. Microsoft corporation. http://www.microsoft.com/.

[58] Microsoft. Windows azure platform.

[59] Amit Nathani, Sanjay Chaudhary, and Gaurav Somani. Policy based resource allocation in iaas cloud. volume 28, pages 94 – 103, P.O. Box 211, Amsterdam, 1000 AE, Netherlands, 2012.

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

[61] 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*, pages 229 –236, june 2012.

[62] S. Pandey, Linlin Wu, S.M. Guru, and R. Buyya. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pages 400 –407, april 2010.

[63] Rackspace Cloud. Cloud Servers - Virtual Server Hosting & Dedicated Server Hosting. http://www.rackspacecloud.com/cloud_hosting_products/servers.

[64] N.S.V. Rao, S.W. Poole, Fei He, Jun Zhuang, C.Y.T. Ma, and D.K.Y. Yau. Cloud computing infrastructure robustness: A game theory approach. In *Computing, Networking and Communications (ICNC), 2012 International Conference on*, pages 34 –38, 30 2012-feb. 2 2012.

[65] 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*, pages 500 –507, july 2011.

[66] Mohsen Amini Salehi and Rajkumar Buyya. Adapting market-oriented scheduling policies for cloud computing. volume 6081 LNCS, pages 351 – 362, Busan, Korea, Republic of, 2010.

[67] Salesforce. saleforce.com - http://www.salesforce.com/.

[68] Pierre Audoin Consultants SAS. Economic and social impact of software & software-based services. 2011.

[69] SPECweb2005. http://www.spec.org/web2005/.

[70] 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*, pages 168–177, 2011.

[71] 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*, pages 228 –235, july 2010.

[72] 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*, pages 16 –23, june 2012.

[73] Wei Wang, Baochun Li, and Ben Liang. Towards optimal capacity segmentation with hybrid cloud pricing. In *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, pages 425 –434, june 2012.

[74] Guiyi Wei, Athanasios V. Vasilakos, Yao Zheng, and Naixue Xiong. A game-theoretic method of fair resource allocation for cloud computing services. *Journal of Supercomputing*, 54(2):252–269, 2010. Date revised - 2012-04-01; Pages 252-269; Journal of Supercomputing [J Supercomput]. Vol. 54, no. 2, pp. 252-269. Nov 2010; 0920-8542.

[75] B. Wickremasinghe, R.N. Calheiros, and R. Buyya. Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pages 446 –452, april 2010.

[76] A. Wolke and G. Meixner. Twospot: A cloud platform for scaling out web applications dynamically. In *ServiceWave*, 2010.

[77] Laurence A. Wolsey. *Integer Programming*. 1998.

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

[79] Z. Xiao, Q. Chen, and H. Luo. Automatic scaling of internet applications for cloud computing services. *Computers, IEEE Transactions on*, PP(99):1, 2012.

[80] 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*, pages 253 –260, june 2012.

[81] 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. *Journal of Cluster Computing*, 12(1):45–57, 2009.