POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione
Corso di Laurea Magistrale in Ingegneria Matematica

Game Theory Models for MapReduce: Joint Admission Control and Capacity Allocation

Relatore:
Prof. Danilo Ardagna — Politecnico di Milano
Correlatori:
Prof. Mauro Passacantando — Università di Pisa
Prof.ssa Michela Meo — Politecnico di Torino
Dott. Michele Ciavotta — Politecnico di Milano

Tesi Magistrale di:
Eugenio Gianniti
Matricola 799220

Anno Accademico 2014-2015

Contents

List of Figures

List of Tables

Abstract

Nowadays many companies have at their disposal large amounts of raw, unstructured data. With the term *Big Data* we refer to the analysis of huge datasets, allowing the extraction of information of utmost importance for business purposes. Among the enabling technologies, a central place is held by the MapReduce framework, in particular its open source implementation, *Apache Hadoop*. For cost effectiveness considerations, a common approach entails sharing server clusters among multiple user classes. Such a common infrastructure should provide every user with a fair share of computational resources, ensuring that Service Level Agreements (SLAs) are met and avoiding wastes.

In this work we consider mathematical programming problems that model the optimal allocation of computational resources in a cluster, in order to develop new capacity allocation techniques, allowing for better performance in shared datacenters. Our goal is the reduction of power consumption, while respecting the deadlines stated in the SLAs and avoiding penalties associated with job rejections. At the core of this approach there is the development of a distributed algorithm, based on Game Theory models and techniques, and enabling run-time capacity allocation, hence the need to split the original problem of resource allocation in MapReduce environments into two classes of problems, one for the central Resource Manager and the other for each Application Master. Further improvements could be gained taking into account also the issue of data locality, which can greatly impact performance in the execution of MapReduce jobs.

Sommario

Oggi molte aziende hanno a disposizione grandi quantità di dati molto spesso non strutturati. Con il termine *Big Data* si fa riferimento all'analisi di grandi moli di dati, permettendo l'estrazione di informazioni di massima importanza per gli obiettivi aziendali. Tra le tecnologie abilitanti, il framework MapReduce occupa un ruolo centrale, in particolare con la sua implementazione open source, *Apache Hadoop*. Per ragioni economiche, una prassi comune prevede la condivisione di cluster di server tra più classi di utenti. Tale infrastruttura comune dovrebbe fornire ad ogni utente una giusta quota di risorse di calcolo, garantendo che i contratti di Service Level Agreements (SLAs) siano soddisfatti ed evitando sprechi, come l'utilizzo inefficiente delle risorse.

In questo lavoro di tesi si propongono problemi di programmazione matematica per modellare l'allocazione ottimale delle risorse computazionali in un cluster, al fine di sviluppare nuove tecniche di allocazione delle risorse, consentendo una migliore performance in datacenter condivisi. L'obiettivo è la riduzione del consumo di energia, rispettando i termini indicati negli SLAs ed evitando penali associate ad obiettivi di prestazioni non raggiunti. Al centro di questo approccio si pone lo sviluppo di un algoritmo distribuito, basato su modelli e tecniche di Teoria dei Giochi, che consenta l'allocazione della capacità di calcolo a run-time. Da qui sorge la necessità di suddividere il problema originale di allocazione delle risorse in ambiente MapReduce in due classi di problemi, uno per il Resource Manager centrale e l'altra per ogni Application Master. Ulteriori lavori futuri faranno rifermiento ad estensioni possibili tenendo conto anche della data locality, che può influire notevolmente sulle prestazioni nell'esecuzione dei job MapReduce.

CHAPTER 1

Introduction

A large number of enterprises currently commits to the extraction of information from huge datasets as part of their core business activities. Applications range from fraud detection to one-to-one marketing, encompassing business analytics and support to decision making in both private and public sectors. In order to cope with the unprecedented amount of data that many companies need to process in a timely way, new technologies are increasingly adopted by the industry, following the *Big Data* paradigm. Among such technologies, Apache Hadoop [11] is already widespread and statistics suggest further increase in its future adoption. IDC estimates that, by 2020, nearly 40% of Big Data analyses will be supported by the public Cloud [29], while Hadoop is expected to touch half of the data worldwide by 2015 [43].

Apache Hadoop is an open source project, backed by the Apache Foundation, developing a software suite that enables the elaboration of vast amounts of data on clusters of commodity hardware. To accomplish this goal, Hadoop features a distributed file system, a resource negotiator, and interfaces to the MapReduce framework. The latter allows for highly scalable parallel computation, automatically ensuring parallelization and distribution, fault-tolerance, reliability, and monitoring. The framework can offer all these interesting capabilities as it requires developers to write applications following the MapReduce programming model, which consists of a fixed workflow, with a map function reading unstructured input data and providing intermediate results to the reduce function, which aggregates them and outputs processed data.

Despite the convenience of this approach and the widespread Hadoop adoption within the Information Technology (IT) industry, still there are

not tools that support developers and operators in common activities related to the capacity planning of MapReduce applications. This thesis investigates the theoretical fundamentals needed to enable design space exploration in the early phases of application development, as well as the optimal management of cluster resources in private Clouds at run-time. To do so, we adopt Game Theory techniques, which found successful application in facing issues related to the Cloud computing paradigm, and use them to provide a scalable solution to the joint admission control and capacity allocation of multi-class Hadoop clusters.

This thesis is organized as follows. Chapter 2 discusses the state of the art, giving an overview of the technologies under study, in particular Cloud computing and Hadoop MapReduce, and exploring the usage of game-theoretic techniques and methods when dealing with related issues, as described in technical literature. Next, Chapter 3 shows how we developed models to solve the joint capacity allocation and admission control problem, interleaving the rigor of mathematical proofs with the intuition of mechanisms underlying the application under study. Then we analyze and validate our results in Chapter 4, where we carry out a thorough exploration of the characteristics of the proposed models. The solution methods are also validated comparing the predicted performance with simulations run exploiting the official simulator offered within the Hadoop suite. In the end, Chapter 5 wraps up this work and draws conclusions on the outcomes. Furthermore, it points out relevant issues that remain open and will be the focus of future work.

CHAPTER 5

Conclusions and Future Work

Throughout this thesis we investigated MapReduce applications running on Hadoop clusters. After exploring the technical aspects related to Cloud computing and Apache Hadoop, as well as the application of Game Theory techniques to similar problems, we focused on the development of models that could be exploited for solving the joint admission control and capacity allocation problem at run-time and in a scalable way. Then, these solution methods were validated through empirical analyses and their results compared with simulations run on the official simulator provided by Hadoop.

We have shown evidence to support the use of Algorithm 3.3.1 and the formulæ of Proposition 3.3.3 as a scalable and accurate distributed solution method for the joint admission control and capacity allocation problem. Furthermore, we compared the results of two different formulations of problem (3.14), using two possible virtual gain terms: $\tilde{r}_i = r_i$ and $\tilde{r}_i = r_i - r_i^{low}$. The former considers the contribute of the whole set of VMs to the virtual gain, whilst the latter does not take into account the virtual payments associated to the portion of resources guaranteeing minimal operation. In lay terms, the former is closer to the intuitive "revenues minus expenses" expression, whilst the latter comes from the observation that r_i^{low} VMs must be assigned to AM i independently of the strategy adopted by the RM, hence their contribution should not influence the outcomes of the optimization process. Multiple times the formulation adopting $\tilde{r}_i = r_i - r_i^{low}$ proved better, with respect to both robustness and accuracy. Indeed, in Section 4.3 the alternative missed the optimal solution in several experiments. Moreover, in Sections 4.4 and 4.5 we showed a loss of efficiency when executing the formulation with $\tilde{r}_i = r_i$.

All in all, we can conclude that Algorithm 3.3.1 with $\tilde{r}_i = r_i - r_i^{low}$ is the best approach to the solution of the problem of interest, both with respect to times of execution and efficiency, and considering the accuracy with which the optimal solution is caught. Furthermore, we showed that this solution method requires execution times below the second, with limited network traffic, then Algorithm 3.3.1 can support, in practice, both design phases and the run-time management of clusters.

Building upon the outcomes of this work, it is possible to investigate further open issues and relevant research questions. An aspect that might strongly affect performance of MapReduce jobs in Hadoop clusters is data locality. When map tasks are started, the framework takes care of automatically providing slots with the slice of data they have to process. Now, if a slot can be scheduled on the same node where HDFS stored one of the blocks it should process, then computation will start right away. If this is not the case, instead, the network time required to transfer data between nodes will add to the job completion time. Similarly, reduce slots should be scheduled in order to minimize the data transfer needed to move intermediate values associated to the same key to the correct node. This behavior can be modeled adding specific parameters and constraints, thus allowing for results more adherent to actual cluster performance, since I/O is one of the factors that greatly affect it.

A project that is attracting increasing attention in the industry is Apache Tez [13]. It can be seen as the natural evolution of Hadoop, where workflows are not fixed anymore. When several data sources need to be accessed and combined, the MapReduce paradigm forces to use clumsy sequences of jobs, with the need to perform I/O on HDFS and explicitly enforce synchronization barriers after each step. On the other hand, Apache Tez abstracts the dependency relationships among input, output, and intermediate data with Directed Acyclic Graphs (DAGs). In this way, the framework itself can manage computation, launching tasks as needed and automatically ensuring that all the required intermediate data for each step is available when it starts. Moreover, this approach allows to avoid unneeded I/O between a step and the following. Another interesting development of this work is the extension of the model to consider the mechanisms governing Apache Tez, hence adapt our joint admission control and capacity allocation problem for the execution of complex DAGs.

In the end, we should be aware that the approximate formulæ we use to estimate performance might incur in large errors due to the inherent difficulty and unpredictability of application performance. A system for reliable performance prediction can greatly benefit from the coupling

with a local search method based on Petri Nets simulations. This technique allows to obtain very accurate predictions by simulating the whole Hadoop system, clearly at the price of longer execution times. In this vision, our models would provide a relevant initial guess for an iterative procedure relying on this more precise technique, in order to find out the optimal configuration or, conversely, certify that an application design will respect the constraints imposed on its execution due to business considerations.

Acronyms

AM Application Master. vii, 1, 3, 22, 23, 36, 39, 45, 47–51, 55, 56, 58, 59, 66, 70–81, 83, 85–89, 97

API application programming interface. 13, 17

ARIA Automatic Resource Inference and Allocation. 25

CPU central processing unit. 7, 11, 22, 24, 26, 64, 68

DAG Directed Acyclic Graph. 98

FIFO First In, First Out. 22–24, 34

GFS Google File System. 17

GNEP Generalized Nash Equilibrium Problem. 33, 52, 56

GPU graphics processing unit. 24

HDFS Hadoop Distributed File System. vii, 17–21, 98

I/O Input/Output. 26, 98

IaaS Infrastructure as a Service. 13–15, 31–33

ICT Information and Communication Technology. 7, 28, 36

IT Information Technology. 5, 8, 9, 30, 67

JN Journal Node. 21

KKT Karush-Kuhn-Tucker. 42, 52, 57, 65

NDFS Nutch Distributed File System. 19

NIST National Institute of Standards and Technologies. 10

OS operating system. 19, 20

PaaS Platform as a Service. 13–15

PoA Price of Anarchy. 29, 30

POSIX Portable Operating System Interface. 20

QN queueing network. 27, 28

QoS Quality of Service. 9, 28, 30–33, 59

RAM random access memory. 64

RM Resource Manager. 1, 3, 22, 23, 36, 46–51, 56, 58, 59, 79, 80, 89, 97

SaaS Software as a Service. 13–15, 32, 33

SLA Service Level Agreement. 1, 3, 9, 13, 33, 35, 36, 38–40, 59, 60

SLS Scheduler Load Simulator. 66, 93, 94

VM virtual machine. 11, 13, 31, 32, 34, 36, 39, 40, 45–51, 59, 64, 67, 68, 71, 75, 97

YARN Yet Another Resource Negotiator. vii, 21, 22, 24, 66, 93, 94

Bibliography

[1] V. Abhishek, I. A. Kash, and P. Key. "Fixed and Market Pricing for Cloud Services". In: *CoRR* (2012). arXiv: 1201.5621 [cs.GT].

[2] M. Abundo, V. Di Valerio, V. Cardellini, and F. Presti. "Bidding Strategies in QoS-Aware Cloud Systems Based on N-Armed Bandit Problems". In: *NCCA*. 2014.

[3] B. Addis, D. Ardagna, A. Capone, and G. Carello. "Energy-Aware Joint Management of Networks and Cloud Infrastructures". In: *Computer Networks* 70 (Sept. 9, 2014).

[4] E. Altman, U. Ayesta, and B. Prabhu. "Load Balancing in Processor Sharing Systems". In: *ValueTools*. 2008.

[5] E. Altman, U. Ayesta, and B. J. Prabhu. "Optimal Load Balancing in Processor Sharing Systems". In: *GameComm*. 2008.

[6] E. Altman, T. Boulogne, R. El-Azouzi, T. Jiménez, and L. Wynter. "A Survey on Networking Games in Telecommunications". In: *Comput. Oper. Res.* 33.2 (2006), pp. 286–311.

[7] *Amazon EC2 Pricing*. URL: http://aws.amazon.com/ec2/pricing/ (visited on 03/17/2015).

[8] *Amazon Web Services*. URL: http://aws.amazon.com/ (visited on 03/17/2015).

[9] *AMPL*. URL: http://www.ampl.com/ (visited on 03/17/2015).

[10] J. Anselmi and B. Gaujal. "Optimal Routing in Parallel, Non-Observable Queues and the Price of Anarchy Revisited". In: *ITC*. 2010.

[11] *Apache Hadoop*. URL: https://hadoop.apache.org (visited on 03/31/2015).

[12] *Apache Rumen*. URL: http://hadoop.apache.org/docs/r1.2.1/rumen.html (visited on 03/28/2015).

[13] *Apache Tez*. URL: http://tez.apache.org (visited on 04/01/2015).

[14] D. Ardagna, S. Casolari, and B. Panicucci. "Flexible Distributed Capacity Allocation and Load Redirect Algorithms for Cloud Systems". In: *CLOUD*. 2011.

[15] D. Ardagna, B. Panicucci, and M. Passacantando. "A Game Theoretic Formulation of the Service Provisioning Problem in Cloud Systems". In: *WWW*. 2011.

[16] D. Ardagna, B. Panicucci, and M. Passacantando. "Generalized Nash Equilibria for the Service Provisioning Problem in Cloud Systems". In: *IEEE Trans. on Services Computing* PP.99 (2012).

[17] S. Bardhan and D. A. Menascé. "Queuing Network Models to Predict the Completion Time of the Map Phase of MapReduce Jobs". In: *Proc. International Computer Measurement Group Conf.* (Las Vegas, NV). Dec. 2012.

[18] J. Barr. *Host Your Web Site in the Cloud: Amazon Web Services Made Easy*. 1st ed. Sitepoint, 2010. ISBN: 978-0980576832.

[19] J. Bredin, D. Kotz, D. Rus, R. Maheswaran, C. Imer, and T. Basar. "Computational Markets to Regulate Mobile-Agent Systems". In: *Autonomous Agents and Multi-Agent Systems* (2003), pp. 235–263.

[20] I. Carrera, F. Scariot, C. Geyer, and P. Turin. *An Example for Performance Prediction for Map Reduce Applications in Cloud Environments*.

[21] H.-L. Chen, J. R. Marden, and A. Wierman. "The Effect of Local Scheduling in Load Balancing Designs". In: *SIGMETRICS Perform. Eval. Rev.* 36 (2 2008), pp. 110–112.

[22] G. Cook. *How Clean is Your Cloud?* Tech. rep. Greenpeace International, Apr. 2012. URL: http://www.greenpeace.org/international/Global/international/publications/climate/2012/iCoal/HowCleanisYourCloud.pdf (visited on 04/02/2015).

[23] J. Dean and S. Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters". In: *6th Symposium on Operating Systems Design and Implementation*. 2004, pp. 137–149.

[24] V. Di Valerio, V. Cardellini, and F. Lo Presti. "Optimal Pricing and Service Provisioning Strategies in Cloud Systems: A Stackelberg Game Approach". In: *CLOUD*. 2013.

[25] P. Dube, Z. Liu, L. Wynter, and C. H. Xia. "Competitive Equilibrium in E-Commerce: Pricing and Outsourcing". In: *Computers & OR* 34.12 (2007), pp. 3541–3559.

[26] *EEX*. URL: http://www.eex.com/en/ (visited on 11/26/2014).

[27] Y. Feng, B. Li, and B. Li. "Price Competition in an Oligopoly Market with Multiple IaaS Cloud Providers". In: *IEEE Trans. on Computers* 63.1 (2014), pp. 59–73.

[28] *Flexyscale*. URL: http://www.flexiscale.com/ (visited on 03/17/2015).

[29] J. Ganz and D. Reinsel. *The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East — United States*. Tech. rep. IDC, Feb. 2013. URL: https://www.emc.com/collateral/analyst-reports/idc-digital-universe-united-states.pdf (visited on 04/07/2015).

[30] *Gestore Mercati Energetici*. URL: http://www.mercatoelettrico.org/En/Default.aspx (visited on 11/26/2014).

[31] *GoGrid*. URL: http://www.gogrid.com/ (visited on 03/17/2015).

[32] *Google App Engine*. URL: https://developers.google.com/appengine/ (visited on 03/17/2015).

[33] *Google Apps for Work*. URL: http://www.google.com/enterprise/apps/business/ (visited on 03/17/2015).

[34] *Google Compute Engine*. URL: https://cloud.google.com/products/compute-engine (visited on 03/17/2015).

[35] *Google Inc.* URL: http://www.google.com/about/company/ (visited on 03/17/2015).

[36] D. Grosu and A. Chronopoulos. "Noncooperative Load Balancing in Distributed Systems". In: *Journ. Parallel Distrib. Comput.* 65.9 (2005), pp. 1022–1034.

[37] *Hadoop MapReduce Next Generation - Capacity Scheduler*. URL: http://hadoop.apache.org/docs/r2.3.0/hadoop-yarn/hadoop-yarn-site/CapacityScheduler.html (visited on 03/12/2015).

[38] M. M. Hassan, M. S. Hossain, A. M. J. Sarkar, and E.-N. Huh. "Cooperative Game-based Distributed Resource Allocation in Horizontal Dynamic Cloud Federation Platform". In: *Information Systems Frontiers* (2012), pp. 1–20.

[39] M. M. Hassan, B. Song, and E.-N. Huh. "Distributed Resource Allocation Games in Horizontal Dynamic Cloud Federation Platform". In: *HPCC*. 2011.

[40] M. Haviv and T. Roughgarden. "The Price of Anarchy in an Exponential Multi-Server". In: *Oper. Res. Lett.* 35.4 (2007), pp. 421–426.

[41] H. Herodotou. *Hadoop Performance Models*. Tech. rep. Duke University, June 6, 2011. arXiv: 1106.0940 [cs.DC].

[42] M. Jebalia, A. Ben Letaïfa, M. Hamdi, and S. Tabbane. "A Comparative Study on Game Theoretic Approaches for Resource Allocation in Cloud Computing Architectures". In: *WETICE*. 2013.

[43] K. Kambatla, G. Kollias, V. Kumar, and A. Grama. "Trends in Big Data Analytics". In: *Journal of Parallel and Distributed Computing* 74 (July 2014), pp. 2561–2573. DOI: 10.1016/j.jpdc.2014.01.003.

[44] *KNITRO*. URL: http://www.ziena.com/knitro.htm (visited on 03/18/2015).

[45] J. Künsemöller and H. Karl. "A Game-Theoretic Approach to the Financial Benefits of Infrastructure-as-a-Service". In: *Future Generation Computer Systems* 41 (2014), pp. 44–52.

[46] *KVM*. URL: http://www.linux-kvm.org/ (visited on 03/17/2015).

[47] E. D. Lazowska, J. Zahorjan, G. S. Graham, and K. C. Sevcik. *Quantitative System Performance. Computer System Analysis Using Queueing Network Models*. Prentice-Hall, 1984. URL: http://homes.cs.washington.edu/~lazowska/qsp/ (visited on 04/07/2015).

[48] D.-R. Liang and S. K. Tripathi. "On Performance Prediction of Parallel Computations with Precedent Constraints". In: *IEEE Transactions on Parallel and Distributed Systems* 11.5 (May 2000), pp. 491–508.

[49] X. Lin, Z. Meng, C. Xu, and M. Wang. "A Practical Performance Model for Hadoop MapReduce". In: *International Conference on Cluster Computing Workshops*. IEEE. 2012. DOI: 10.1109/ClusterW.2012.24.

[50] Z. Lu, X. Wen, and Y. Sun. "A Game Theory Based Resource Sharing Scheme in Cloud Computing Environment". In: *WICT*. 2012.

[51] M. Malekimajd, A. M. Rizzi, D. Ardagna, M. Ciavotta, M. Passacantando, and A. Movaghar. "Optimal Capacity Allocation for Executing MapReduce Jobs in Cloud Systems". In: *MICAS-SYNASC 2014 Workshops Proceedings*. (Timisoara, Romania). Forthcoming.

[52] P. Mell and T. Grance. *The NIST Definition of Cloud Computing*. Tech. rep. July 10, 2009. URL: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf (visited on 03/17/2015).

[53] I. Menache, A. Ozdaglar, and N. Shimkin. "Socially Optimal Pricing of Cloud Computing Resources". In: *VALUETOOLS*. 2011.

[54] *Microsoft Corporation*. (Visited on 03/17/2015).

[55] *Microsoft Office 365*. URL: http://office365.microsoft.com/ (visited on 03/17/2015).

[56] *Microsoft Windows Azure*. URL: http://www.windowsazure.com/ (visited on 03/17/2015).

[57] *Microsoft Windows Azure Virtual Machines*. URL: http://www.windowsazure.com/en-us/home/features/virtual-machines/ (visited on 03/17/2015).

[58] *Onlive*. URL: http://www.onlive.com/ (visited on 03/17/2015).

[59] R. Pal and P. Hui. "Economic Models for Cloud Service Markets: Pricing and Capacity Planning". In: *Theoretical Computer Science* 496 (2013), pp. 113–124.

[60] P. S. Pillai and S. Rao. "Resource Allocation in Cloud Computing Using the Uncertainty Principle of Game Theory". In: *Systems Journal, IEEE* PP.99 (2014), pp. 1–12.

[61] A. S. Prasad and S. Rao. "A Mechanism Design Approach to Resource Procurement in Cloud Computing". In: *Computers, IEEE Transactions on* 63.1 (2014), pp. 17–30.

[62] *Rackspace*. URL: http://www.rackspace.com/ (visited on 03/17/2015).

[63] N. S. V. Rao, S. W. Poole, F. He, J. Zhuang, C. Y. T. Ma, and D. K. Y. Yau. "Cloud Computing Infrastructure Robustness: A Game Theory Approach". In: *ICNC*. 2012.

[64] H. Roh, C. Jung, W. Lee, and D.-Z. Du. "Resource Pricing Game in Geo-distributed Clouds". In: *INFOCOM*. 2013.

[65] T. Roughgarden. "The Price of Anarchy is Independent of the Network Topology". In: *STOC*. 2002.

[66] *Salesforce*. URL: http://www.force.com/ (visited on 03/17/2015).

[67] N. Samaan. "A Novel Economic Sharing Model in a Federation of Selfish Cloud Providers". In: *IEEE Trans. on Paral. and Distr. Syst.* 25.1 (2014), pp. 12–21.

[68] T. Sandholm and K. Lai. "MapReduce Optimization Using Regulated Dynamic Prioritization". In: *SIGMETRICS/Performance '09*. (Seattle, WA, USA). June 15–19, 2009, pp. 299–310.

[69] *SAP*. URL: http://www.sap.com/ (visited on 03/17/2015).

[70] G. Song, L. Yu, Z. Meng, and X. Lin. "A Game Theory Based MapReduce Scheduling Algorithm". In: *Emerging Technologies for Information Systems, Computing, and Management*. Ed. by W. E. Wong and T. Ma. Lecture Notes in Electrical Engineering 236. Springer New York, 2013, pp. 287–296. DOI: 10.1007/978-1-4614-7010-6_33.

[71] Y. Song, M. Zafer, and K.-W. Lee. "Optimal Bidding in Spot Instance Market". In: *INFOCOM*. 2012.

[72] SPEC. *SPECpower_ssj2008*. URL: http://www.spec.org/power_ssj2008/ (visited on 11/26/2014).

[73] SPEC. *SPECvirt_sc2013*. URL: https://www.spec.org/virt_sc2013/ (visited on 11/26/2014).

[74] F. Teng and F. Magoules. "A New Game Theoretical Resource Allocation Algorithm for Cloud Computing". In: *GPC*. 2010.

[75] C.-W. Tsai and Z. Tsai. "Bid-Proportional Auction for Resource Allocation in Capacity-Constrained Clouds". In: *WAINA*. 2012.

[76] A. Verma, L. Cherkasova, and R. H. Campbell. "ARIA: Automatic Resource Inference and Allocation for MapReduce Environments". In: *Proceedings of the Eighth International Conference on Autonomic Computing*. June 2011.

[77] A. Verma, L. Cherkasova, and R. H. Campbell. "Resource Provisioning Framework for MapReduce Jobs with Performance Goals". In: *Middleware*. Vol. 7049. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 165–186. DOI: 10.1007/978-3-642-25821-3_9.

[78] E. Vianna, G. Comarela, T. Pontes, J. Almeida, V. Almeida, K. Wilkinson, H. Kumo, and U. Dayal. "Analytical Performance Models for MapReduce Workloads". In: *Int J Parallel Prog* 41 (2013), pp. 495–525. DOI: 10.1007/s10766-012-0227-4.

[79] E. Vianna, G. Comarela, T. Pontes, J. Almeida, V. Almeida, K. Wilkinson, H. Kumo, and U. Dayal. "Modeling the Performance of the Hadoop Online Prototype". In: *23rd International Symposium on Computer Architecture and High Performance Computing*. IEEE. 2011, pp. 152–159. DOI: 10.1109/SBAC-PAD.2011.24.

[80] V. Vinothina, R. Sridaran, and P. Ganapathi. "A Survey on Resource Allocation Strategies in Cloud Computing". In: *International Journal of Advanced Computer Science and Applications* 3.6 (2012).

[81] *VMware Inc.* URL: http://www.vmware.com/ (visited on 03/17/2015).

[82] J. Wan, D. Deng, and C. Jiang. "Non-Cooperative Gaming and Bidding Model Based Resource Allocation in Virtual Machine Environment". In: *IPDPS Workshops*. 2012.

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

[84] Y. Wang, X. Lin, and M. Pedram. "A Game Theoretic Framework of SLA-Based Resource Allocation for Competitive Cloud Service Providers". In: *GreenTech*. 2014.

[85] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong. "A Game-Theoretic Method of Fair Resource Allocation for Cloud Computing Services". In: *The Journal of Supercomputing* 54.2 (2010), pp. 252–269.

[86] *Xen Hypervisor*. URL: http://www.xen.org/ (visited on 03/17/2015).

[87] X. Xu, H. Yu, and X. Cong. "A QoS-Constrained Resource Allocation Game in Federated Cloud". In: *IMIS*. 2013.

[88] X. Yang and J. Sun. "An Analytical Performance Model of MapReduce". In: *CCIS*. IEEE. 2011.

[89] B. Yolken and N. Bambos. "Game Based Capacity Allocation for Utility Computing Environments". In: *ValueTools*. 2008.

[90] M. Zafer, Y. Song, and K.-W. Lee. "Optimal Bids for Spot VMs in a Cloud for Deadline Constrained Jobs". In: *CLOUD*. 2012.

[91] J. Zhang, F. Dong, D. Shen, and J. Luo. "Game Theory based Dynamic Resource Allocation for Hybrid Environment with Cloud and Big Data Applications". In: *International Conference on Systems, Man, and Cybernetics*. (San Diego, CA, USA). IEEE. Oct. 5–8, 2014, pp. 1128–1133.

[92] Q. Zhang, Q. Zhu, M. Zhani, and R. Boutaba. "Dynamic Service Placement in Geographically Distributed Clouds". In: *ICDCS*. 2012.

[93] Q. Zhang, L. Cheng, and R. Boutaba. "Cloud Computing: State-of-the-Art and Research Challenges". In: *J. Internet Services and Applications* 1.1 (2010), pp. 7–18. DOI: 10.1007/s13174-010-0007-6.

[94] Z. Zhang, L. Cherkasova, and B. T. Loo. "Parameterizable Benchmarking Framework for Designing a MapReduce Performance Model". In: *Concurrency Computat.: Pract. Exper.* (2014). DOI: 10.1002/cpe.3229.