Analysing the performance instability correlation with various workflow and cloud parameters

Sasko Ristov, Roland Mathá, and Radu Prodan

Distributed and Parallel Systems Group, Institute for Computer Science, University of Innsbruck Technikerstr. 21a, A-6020 Innsbruck Email:{sashko, roland, radu}@dps.uibk.ac.at

Abstract—The cloud is an eco-system in which virtual machine instances are starting and terminating asynchronously on user demand or automatically when the load is rapidly increased or decreased. Although this dynamical environment allows to rent computing or storage resources cheaper rather than buying them. still it does not guarantee the stable execution during a period of time as the traditional physical environment. This is emphasised even more for workflows execution, since they consist of many data and control dependencies, which cause the makespan to be instable when a workflow is being executed in different periods of time in cloud. In this paper we analyse several parameters of workflow and the cloud environment that are expected to impact the workflow execution instability and investigate the correlation between them. The cloud parameters include the number of instances and their type, as well as the correlation with the efficient or inefficient execution of workflow parallel sections. We conduct a series of experiments, repeating each experiment by 30 test cases in order to evaluate instability for different cloud and workflow parameters. The results show a neglectfully correlation between each pair of parameters, as well as the tasks and file transfers within the workflow. Oppose to the expectations, the distribution of the makespan per experiment does not always comply with the normal distribution, which is also not correlated to a particular cloud or workflow parameter.

Index Terms—Cloud, makespan, performance modeling, workflow.

I. INTRODUCTION

Nowadays, companies' spending budgets are more distributed to management and automation software, rather than for the hardware. Almost all companies have already virtualised their data centers with some open source cloud framework, or have migrated their services in some of many public clouds. Besides that the Cloud became a platform for many applications and services, it is a promising platform even for scientific applications and workflows' execution [1].

Still, many challenges exist that could impact the decisions whether to migrate the computing to the Cloud. Although many researchers reported that the Cloud will reduce the costs for IT infrastructure, still its performance instability could change ones' minds [2]. Since the cost is directly connected with the time period that resources are leased, it can be neither estimated nor predicted.

The cloud performance instability is emphasised not only for the scientific applications that require massive data and computation over it, but even more for workflows executions, because they consist of many tasks that have control and data dependencies between them. A cloud virtual machine instance can provide a totally different and even unexpected performance in two different periods of time [3]. Many instances are straggling and thus providing some bottlenecks, while some others could fail, which additionally will make the performance instable. Predicting and modeling the cloud performance instability is important as the users can analyse the processes as close as possible to real environments.

Although many cloud simulators exist, which can simulate the execution of huge and complex workflows in seconds [4], still most of them provide a static simulation, that is, the same output is provided for the same input. The static simulation cannot predict the dynamism of the real cloud, where the makespan, and thus the cost, will be always different, due to its i) pay-as-you-go pricing model, ii) dynamics in starting and deployment time of instances [5] and iii) cloud performance fluctuation [6]. Neglecting this performance instability can later lead to wrong conclusions and decisions. Additionally, knowing the correlation of cloud environment parameters on one side, and workflow parameters on the other side, one can reduce the number of experiments in the Cloud for correlated parameters.For example, if the cloud performance instability does not depend on which type of virtual machine instance will be used, then one can do experiments with only one instance type, probably with the cheapest type in order to reduce the overall cost of the experiments.

This paper analyses the performance instability when a workflow is being executed in the cloud. We investigate if there exist a correlation between the performance instability and some cloud environment parameter, such as the number of instances and their type that are used for execution of a workflow, or with some workflow parameters, such as the size of parallel section. Also, we examine various workflow parameters that could lead to efficient or inefficient workflow execution. Our analysis shows that only when a workflow is executed sequentially, that is, with only one small instance with one virtual CPU core, all other experiments do not have any correlation with the environmental and workflow parameters. Additionally, greater correlation exist for task runtimes, rather than the file transfers.

The paper is organised in several sections that are described below. Section II presents the related works in the workflow execution and cloud performance instability. In Section III, we analyse which parameters cause the makespan instability when a workflow is executed in cloud. The testing methodology is

©2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. DOI: https://doi.org/10.1109/PDP.2017.80

described in Section IV, while the results of the experiments and evaluation of the parameters' correlation are presented in Section V. Finally, we conclude our paper and present the plans for future work in Section VI.

II. RELATED WORK

This section presents several examples that can be found in the literature, where researchers reported how instable performance the Cloud provides, when an application or a workflow is executed in different time periods.

Dejun et al. [7] reported that there is a high deviation of performance in Amazon EC2 for a VM of the same instance type during two different time periods. Additionally to this, Jackson et al. [8] have determined that the performance instability is caused by the different underlying hardware within the cloud, although the instances are declared of the same type.

Another performance instability is a long-term behavior of the Cloud, which was detected by Schad et al. [9] for Amazon EC2. That is, the behavior of Amazon EC2 is changed from one time period to another. They also reported an instability that is correlated to the CPU type of an instance and it depends also on the time period during a day and in which day of the week the experiments are conducted. Still, there are some cases where the Cloud provides a period of constant performance, as reported by Iosup et al. [10]. Still, this stability depends on the application type.

The performance instability is more ephasised for the workflows, as they contain dependencies between tasks. Dynamic schedules are more prone to the instability than static ones and can reduce the workflow makespan [11].

The cloud dynamic and instable environment can be simulated in order to predict or determine some limits of its behavior. Several tools are found where the cloud performance instability is simulated, even for workflow execution, such as WorkflowSim [12], which is an extension of Cloudsim [13] by introducing several parameters that are specific for workflows. The DynamicCloudSim simulator [14], which is an extension of CloudSim [13], introduces several additional parameters to simulate the cloud performance instability. Google's GloudSim [15] also simulates some dynamism in cloud, but only in horizontal or vertical resizing the instances, while the performance of the same instance is constant during a time period.

III. WHAT CAUSES WORKFLOW MAKESPAN INSTABILITY?

This section will elaborate the parameters that provide the makespan instability when a workflow is being executed in a dynamic cloud environment. Two main groups of issues can be classified: i) the cloud environment impact on the workflow execution, and ii) the workflow itself with its data and control dependencies and various task types.

A. Cloud (environmental) parameters

The virtualisation layer of the Cloud is the most important factor that impacts the performance instability. This additional layer cannot guarantee the same performance provided by the computing resources, memory and I/O virtual to physical translations.

The virtualisation causes the performance instability indirectly through the virtual machine instances, as well. Some of them are straggling or failing, or instability can be achieved due to the usage of carious resource overcommitment (oversubscription) techniques. Additionally, the multi-tenancy is another important factor. The number of active instances on the physical server and the total number of instances that are started in the whole cloud, as well as their utilisation over the time can impact on the instance straggling and failing. All these instances share the same underlying hardware resources, which sometimes can cause bottlenecks.

Cloud underlying hardware resources are heterogeneous, which additionally can cause performance instability. Virtual machine instances of the same type that are started in different time periods will probably be deployed on heterogeneous physical servers, and thus will provide different performance.

Instance types can cause additionally performance instability because horizontal or vertical scaling provides various performance for different types of applications or tasks. That is, the same amount of resources that will cost the same, sometimes provide slowdown, but in other cases can provide even superlinear speedup [16].

B. Workflow parameters

Workflows are consisted of tasks and connections (file transfers) between them. Some tasks are dependent and cannot be executed in parallel. A small delay in starting time of a task, can change the workflow scheduling, which will increase or decrease the file transfers by changing the virtual machine instance where some task will be executed. Therefore, the workflow type, and data and control dependencies are also a very important issue that impacts the performance instability additionally to the instability caused by the cloud's parameters.

C. A combination of cloud and workflow parameters

Also, a combination of cloud and workflow parameters can provide efficient or inefficient execution of workflow tasks, which can cause the additional performance instability. For example, if there is a parallel section with 20 tasks, using 10 instances should be more efficient than 13, since the execution time will be the same, or at least similar, while the cost is much greater for the latter case, which is more inefficient.

Tasks' interference appears when at least two tasks are executed concurrently in the same instance, which also impacts the performance. Additionally, execution of 10 tasks in 10 small instances, each with one CPU, and in five medium instances, each with two CPUs, will have different behavior.

IV. TESTING METHODOLOGY

This section explains the testing environment, test data, test cases and the experiments in order to make them reproducible. Our goal is to investigate the cloud and workflow parameters, as well as their correlation impact the workflow execution instability.

 TABLE I

 Instance Types of our private OpenStack cloud





Fig. 1. An example of the WIEN2k workflow

A. Testing environment

We have created a realistic cloud testing environment to evaluate the cloud instability of workflow executions. A private cloud with a total of 40 CPU cores of Intel(R) Xeon(R) CPU E5-2680 v2 with 2GHz based on Openstack cloud framework is used. Two instance types are used, small and medium, as specified in Table I, each with CentOS 6.3 operating system.

Askalon [17] is used to execute the workflows in cloud and to monitor and measure the execution of a workflow tasks.

In order to check as much parameters as possible that could impact on the workflow makespan's instability, we have chosen the WIEN2k workflow, which consists of two parallel sections (T2 and T4) with a single task before and after them for synchronisation (T1, T3, T5), as presented in Fig. 1. The number of tasks n within the parallel sections could be changed and is defined by an input parameter. The total number l of all tasks is given with $l = 2 \cdot n + 3$. We choose two different parameters, $n_1 = 13$ and $n_2 = 44$ in order to investigate if the number of instances and the size of the parallel sections are correlated with the makespan, that is, if there is a correlation when the number of instances is a divisor of the number of the tasks in parallel sections.

Although WIEN2k is a simple workflow, still, it offers various characteristics to be investigated, which could impact the performance instability.

B. Experiments and test cases

We created a total of 13 experiments. Twelve of them are defined by changing the workflow parameter (parallel sections size, 13 and 44), instance type (small or medium) and the number of instances (up to four). The 13th experiment is defined when WIEN2k/13 workflow is executed with only one small instance type. Table II describes each experiment along with its abbreviation.

When both WIEN2k/13 and WIEN2k/44 workflows are executed with three instances, then the executions are less

TABLE II EXPERIMENTS' DESCRIPTION

#	Abbreviation	Workflow	Instance type	Instances
1	13S1	WIEN2k/13	Small	1
2	13S2	WIEN2k/13	Small	2
3	13\$3	WIEN2k/13	Small	3
4	13S4	54 WIEN2k/13 Small		4
5	13M2	WIEN2k/13	Medium	2
6	13M3	WIEN2k/13	Medium	3
7	13M4	WIEN2k/13	Medium	4
8	44S2	WIEN2k/44	Small	2
9	44S3	WIEN2k/44	Small	3
10	44S4	WIEN2k/44	Small	4
11	44M2	WIEN2k/44	Medium	2
12	44M3	WIEN2k/44	Medium	3
13	44M4	WIEN2k/44	Medium	4





Fig. 2. Balanced and unbalanced execution of a parallel section

efficient because some instances are not utilised at the end of the parallel sections, while the experiments with both even number of tasks and instances utilise the instances better, as presented in Fig. 2.

Instances of medium type can execute up to two workflow tasks concurrently if they are independent. Oppose to them, instances of small type execute the tasks sequentially, which could cause more or longer file transfers between the instances because small instances could be deployed on various phisical cloud server nodes. In the experiments with only one small instance, the file transfers between instances are minimised because all file transfers are always within the same instance.

Each experiment is repeated 30 times (test cases) in order to investigate the workflow makespan instability in the cloud.

C. Test data

Several parameters that can impact the workflow makespan and thus make it unstable are measured. For each task we measure the time spent for files transfer and task's runtime.

We are using the Pearson's correlation coefficient in order to quantify the correlation between various parameters. the tasks execution correlation between two consecutive test cases within an experiment and the correlation of experiments makespans between each two experiments. We expect that the correlation between similar experiments (with only one different parameter) are more correlated compared to others.

V. EVALUATION

This section presents the results of the experiments and analyses the makespan correlation with the environment and



Fig. 3. Mean makespan deviation's distribution for each test case of all experiments experiments (environments)

workflow parameters. We present how much the consecutive tasks runtime and file transfers are correlated with the makespan for each experiment and how much the experiments are correlated among each other.

A. Makespan instability analysis

The results of mean makespan deviation's for all test cases of each experiment are presented in Fig. 3. We observe that although most of the test cases' makespans are within the interval of $Mean \pm 5\%$, some test cases provide up to 15% greater or up to 8.9% lower makespans than the corresponding experiment's mean value.

For better analysis, Table III presents the makespan's mean values and their relative standard deviations. We observe that the relative standard deviation for most experiments with WIEN2k/13 is 0.03 which is slightly greater than 0.02 that is the relative standard deviation for most of the experiments with WIEN2k/13. Apart of this, we cannot determine any correlation with some cloud or workflow parameters.

Although several researchers and simulators use normal distribution for cloud behavior, even for workflows, we applied the Kolmogorov-Smirnov (KS) test to determine if the makespan distribution within an experiment complies with the normal distribution. Fig. 4 presents the *p*-values of the KS test. We observe very discrepant *p*-values, that is, some experiments, such as all WIEN2k/44 with small instances, 13S3 and 44M2, have *p*-values between [0.55, 0.93] and are complying with the normal distribution, while others have much smaller *p*-value of up to 0.2. Just to note that the experiment 44M3 has p = 0. We can conclude that we cannot observe some correlation between the experiments.

B. Makespan's cross correlation between experiments

Further on, we analyse the cross correlation between each pair of experiments in order to determine if there is some parameter that impacts constantly on the makespan instability. We calculate the Pearson correlation coefficients, which are presented in Table IV. However, we cannot observe some correlation of makespan and some cloud or workflow parameter since the cross correlations cannot be classified for instance type or number, or the workflow parameter. Additionally, all



Fig. 4. p-values of the KS test for each experiments' makespan arrays

cross correlations are in the range of [-0.45, 0.41], which shows that there is no correlation between any two experiments and we cannot conclude which parameter impact is correlated with a workflow's makespan.

C. Tasks' runtime vs file transfer correlation

The workflow makespan consists of time spent for file transfers between tasks and task runtimes. Further on, we analyse if there is a correlation between these parameters and the makespan for all experiments.

Fig. 5 presents the average value of each pair's consecutive test cases correlations. We observe almost a total correlation between two consecutive workflow executions for tasks' runtimes, which are in the range of [0.97, 1], while there is smaller correlation for file transfers counterparts, which are in the range of [0.71, 0.91]. Still, the correlation for file transfer is enough to conclude that the file transfers for all tasks in a workflow are correlated in each two consecutive workflow executions.

Correlation for sequential execution in the experiment 13S1 shows much greater correlation of 0.95, which is expected because there is only one small instance and all file transfers are minimised within it. Slightly smaller average correlation is achieved for the experiment 13S4, which we believe is because of various placing of instances on physical servers. This should not be caused by the hyperthreading because the task runtimes have the similar correlation with other experiments.

D. Tasks' cross correlation

The results of the previous subsection showed that the executions of each two consecutive test cases (the whole workflows) are correlated. Since the workflow has different task types (tasks for synchronisation and tasks in parallel sections), now we analyse which task type is correlated with the whole workflow makespan.

Table V presents the mean values and relative standard deviation of task runtimes vs makespan cross correlation coefficient. Sequential experiment 13S1 also shows an average correlation of 0.52, which is much greater than other experiments, which is expected, but still not enough to say that the tasks' runtimes are correlated with the makespan.

		TABLE III		
MAKESPANS' MEA	N VALUES AND	THEIR RELATIVE STANDARD	DEVIATIONS FOR EACH E	XPERIMENT

	13S2	13S3	13S4	13M2	13M3	13M4	44S2	44S3	44S4	44M2	44M3	44M4
Mean makespan	213.29	182.72	172.32	160.37	159.74	161.73	503.48	408.66	371.26	358.39	337.54	326.68
Relative deviation	0.06	0.03	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.05	0.02

TABLE IV CROSS CORRELATION OF EACH EXPERIMENT'S MAKESPANS

	13S2	13S3	13S4	13M2	13M3	13M4	44S2	44S3	44S4	44M2	44M3	44M4
13S2	1	-0.246	-0.122	0.011	-0.157	-0.274	0.232	-0.005	0.135	0.001	0.136	-0.127
13S3	-0.246	1	-0.001	-0.173	-0.076	0.396	-0.091	-0.148	0.011	-0.017	0.113	-0.320
13S4	-0.122	-0.001	1	-0.186	0.132	0.199	-0.142	0.250	0.114	-0.177	-0.225	0.011
13M2	0.011	-0.173	-0.186	1	0.221	-0.168	0.090	0.020	-0.403	0.205	0.411	-0.104
13M3	-0.157	-0.076	0.132	0.221	1	-0.123	-0.142	-0.208	-0.431	0.021	0.318	0.131
13M4	-0.274	0.396	0.199	-0.168	-0.123	1	-0.102	0.075	-0.046	-0.097	-0.035	0.034
44S2	0.232	-0.091	-0.142	0.090	-0.142	-0.102	1	-0.171	0.108	-0.192	0.279	-0.237
44S3	-0.005	-0.148	0.250	0.020	-0.208	0.075	-0.171	1	0.103	0.041	-0.209	0.083
44S4	0.135	0.011	0.114	-0.403	-0.431	-0.046	0.108	0.103	1	-0.218	-0.453	-0.032
44M2	0.001	-0.017	-0.177	0.205	0.021	-0.097	-0.192	0.041	-0.218	1	0.147	0.211
44M3	0.136	0.113	-0.225	0.411	0.318	-0.035	0.279	-0.209	-0.453	0.147	1	0.026
44M4	-0.127	-0.320	0.011	-0.104	0.131	0.034	-0.237	0.083	-0.032	0.211	0.026	1

 TABLE V

 Mean and relative standard deviation of tasks' runtimes and workflow makespan correlations for each experiment

	13S1	13S2	1383	1384	13M2	13M3	13M4	44S2	44S3	44S4	44M2	44M3	44M4
Mean of runtime-makespan corr.	0.52	0.16	0.14	0.11	0.12	0.07	0.13	0.45	0.14	0.10	0.15	0.11	0.14
Relative stdandard deviation	0.34	0.57	0.57	0.88	0.78	0.95	0.74	0.27	0.78	0.85	0.72	0.82	0.73



Fig. 5. Comparison of average correlation of each test case' file transfer times and task runtimes with the workflow makespan for each experiment (environments)

All other experiments show much smaller average correlation, while much greater compatriot relative standard deviation.

In order to check the correlation of task types and the makespan, we calculate the average task runtime of the parallel sections in each test case of all experiments. Table VI shows the Pearson's correlation coefficient of mean values of tasks' runtimes per task group for each experiment. Again we observe that there is a correlation for the sequential experiment 13S1 only. For all others, the Pearson's correlation coefficient of all task groups' mean values are much smaller, which shows that there is no correlation between each task and the workflow makespan.

If we analyse by the task type, there is a slightly greater

TABLE VI PEARSON CORRELATION COEFFICIENT OF MEAN VALUES OF TASKS' RUNTIMES PER TASK GROUP FOR EACH EXPERIMENT

	T1	AVG T2	T3	AVG T4	T5
13S1	0.56	0.78	0.15	0.71	0.47
13S2	-0.01	0.07	0.23	0.23	0.28
1383	0.26	0.33	-0.08	-0.07	0.28
13S4	0.38	0.23	-0.14	0.19	0.10
13M2	-0.05	0.19	0.05	0.15	-0.30
13M3	0.22	0.10	0.26	0.04	-0.13
13M4	0.26	0.31	0.41	0.01	0.25
44S2	-0.14	0.39	-0.14	0.12	-0.11
44S3	0.16	0.17	0.03	0.21	0.11
44S4	-0.15	0.17	-0.14	-0.05	-0.23
44M2	0.35	0.43	0.17	-0.09	0.41
44M3	0.18	0.39	0.06	0.32	0.12
44M4	-0.01	0.37	-0.25	0.07	-0.09

correlation coefficient for the first parallel section (T2) for most experiments, but still this is inconsiderable compared to other tasks' correlation coefficient.

VI. CONCLUSION

Many parameters, both from the cloud environment and the workflow type itself, impact the makespan instability. However, the results of our analysis did not show any direct correlation between the instance type, number of instances and parallel section size of the workflow with the workflows' makespans, except for the sequential execution of a workflow with only one small instance.

There is a huge correlation (almost one) between each two consecutive workflow executions for the tasks runtimes.

Although the file transfers are also correlated (around 0.8), their correlation is smaller than the tasks runtimes. However, there is no correlation between neither the runtime of consecutive executions of a task, nor the file transfers with the whole makespan within the same experiment (cloud environment). Also, we could not observe any correlation of different task types within a workflow that will impact the makespan instability, regardless if it is either a synchronisation or a task of parallel section.

Several researchers [14], [18] used a normal distribution for the workflow execution to simulate the cloud instability, but we observed that it is not always the case. Additional analysis should be conducted for dynamical simulation of workflow execution in cloud that will simulate the totally randomly cloud behavior, regardless of the environment's and the workflow's parameters.

Although we used several cloud parameters and different workflow parallel section parameters, we used only one workflow type - WIEN2k. For future work we plan to investigate if there is any correlation for other realistic and synthetic workflows. Also, all experiments were conducted in the private OpenStack cloud, and we will analyse the correlation in some public commercial cloud.

REFERENCES

- C. Hoffa, G. Mehta, T. Freeman, E. Deelman, K. Keahey, B. Berriman, and J. Good, "On the use of cloud computing for scientific workflows," in *eScience*, 2008. *eScience* '08. *IEEE Fourth International Conference* on, Dec 2008, pp. 640–645.
- [2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California at Berkeley, Tech. Rep., February 2009. [Online]. Available: http://berkeleyclouds.blogspot.com/2009/02/above-clouds-released.html
- [3] A. O. Ayodele, J. Rao, and T. E. Boult, "Performance measurement and interference profiling in multi-tenant clouds," in *Cloud Computing* (*CLOUD*), 2015 IEEE 8th Int. Conf. on, June 2015, pp. 941–949.
- [4] W. Tian, M. Xu, A. Chen, G. Li, X. Wang, and Y. Chen, "Open-source simulators for cloud computing: Comparative study and challenging issues," *Simulation Modelling Practice and Theory*, vol. 58, Part 2, pp. 239 – 254, 2015, special issue on Cloud Simulation. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1569190X15000970

- [5] A. Tchernykh, U. Schwiegelsohn, V. Alexandrov, and E. ghazali Talbi, "Towards understanding uncertainty in cloud computing resource provisioning," *Procedia Computer Science*, vol. 51, pp. 1772 – 1781, 2015, int. Conf. on Computational Science, {ICCS} 2015.
- [6] F. Wu, Q. Wu, and Y. Tan, "Workflow scheduling in cloud: a survey," *The Journal of Supercomputing*, vol. 71, no. 9, pp. 3373–3418, 2015.
- [7] J. Dejun, G. Pierre, and C.-H. Chi, "Service-oriented computing. icsoc/servicewave 2009 workshops: International workshops, icsoc/servicewave 2009, stockholm, sweden, november 23-27, 2009, revised selected papers," A. Dan, F. Gittler, and F. Toumani, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ch. EC2 Performance Analysis for Resource Provisioning of Service-Oriented Applications, pp. 197–207.
- [8] K. R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. J. Wasserman, and N. J. Wright, "Performance analysis of high performance computing applications on the amazon web services cloud," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, Nov 2010, pp. 159–168.
- [9] J. Schad, J. Dittrich, and J.-A. Quiané-Ruiz, "Runtime measurements in the cloud: Observing, analyzing, and reducing variance," *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 460–471, Sep. 2010.
- [10] A. Iosup, N. Yigitbasi, and D. Epema, "On the performance variability of production cloud services," in *Cluster, Cloud and Grid Computing* (CCGrid), 2011 11th IEEE/ACM International Symposium on, May 2011, pp. 104–113.
- [11] M. Maławski, G. Juve, E. Deelman, and J. Nabrzyski, "Cost- and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds," in *Proc. of the Int. Conf. on HPC, Networking, Storage* and Analysis, ser. SC '12, 2012, pp. 1–11.
- [12] W. Chen and E. Deelman, "Workflowsim: A toolkit for simulating scientific workflows in distributed environments," in *E-Science (e-Science)*, 2012 IEEE 8th International Conference on, Oct 2012, pp. 1–8.
- [13] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.
- [14] M. Bux and U. Leser, "Dynamiccloudsim: Simulating heterogeneity in computational clouds," *Future Generation Computer Systems*, vol. 46, pp. 85 – 99, 2015.
- [15] S. Di and F. Cappello, "Gloudsim: Google trace based cloud simulator with virtual machines," *Softw. Pract. Exper.*, vol. 45, no. 11, pp. 1571– 1590, Nov. 2015.
- [16] M. Gusev and S. Ristov, "A superlinear speedup region for matrix multiplication," *Concurrency and Computation: Practice and Experience*, vol. 26, no. 11, pp. 1847–1868, 2013.
- [17] S. Ostermann, R. Prodan, and T. Fahringer, "Extending grids with cloud resource management for scientific computing," in *Grid Computing*, 2009 10th IEEE/ACM International Conference on. IEEE, 2009, pp. 42–49.
- [18] D. Poola, S. Garg, R. Buyya, Y. Yang, and K. Ramamohanarao, "Robust scheduling of scientific workflows with deadline and budget constraints in clouds," in Advanced Information Networking and Applications, 2014 IEEE Int. Conf. on, May 2014, pp. 858–865.