Exploiting Price and Performance Tradeoffs in Heterogeneous Clouds

Eduardo Roloff
Informatics Institute, Federal University of Rio Grande do Sul, Porto Alegre, Brazil
eroloff@inf.ufrgs.br

Matthias Diener
Informatics Institute, Federal University of Rio Grande do Sul, Porto Alegre, Brazil
mdiener@inf.ufrgs.br

Emmanuell D. Carreño
Informatics Institute, Federal University of Rio Grande do Sul, Porto Alegre, Brazil
edcarreno@inf.ufrgs.br

Francis B. Moreira
Informatics Institute, Federal University of Rio Grande do Sul, Porto Alegre, Brazil
fbmoreira@inf.ufrgs.br

Luciano P. Gaspary
Informatics Institute, Federal University of Rio Grande do Sul, Porto Alegre, Brazil
paschoal@inf.ufrgs.br

Philippe O. A. Navaux
Informatics Institute, Federal University of Rio Grande do Sul, Porto Alegre, Brazil
navaux@inf.ufrgs.br

ABSTRACT

Parallel applications are composed of several tasks, which have different computational demands among them. Moreover, most cloud providers offer multiple instance configurations, with large variations of computational power and cost. A combination between the application requirements and the variety of instance types of the cloud could be explored to improve the cost efficiency of the application execution. In this paper, we introduce the cost-delay product as a metric to measure the cost efficiency of cloud systems. With this metric, cloud tenants can evaluate different tradeoffs between cost and performance for their application, depending on their preferences. We explore the use of multiple instance types to create heterogeneous cluster systems in the cloud. Our results show that heterogeneous clouds can have a better cost efficiency than homogeneous systems, reducing the price of execution while maintaining a similar application performance. Furthermore, by comparing the cost-delay product, the user can select an instance mix that is most suitable for his needs.

KEYWORDS

Cloud computing, cost efficiency, heterogeneity, performance

1 INTRODUCTION

Cloud environments offer a large range of instance configurations, with several different aspects that affect the performance and price of the instance. When comparing the flexibility to build customized environments and the initial investments required to create an execution environment, the cloud is a good option in comparison to traditional environments, such as clusters [17]. The research on cloud computing for High-Performance Computing (HPC) is mainly focused on the evaluation and improvement of performance and communication and has received large amounts of attention in recent years [2–4, 14, 21].

However, building a cloud system out of more than one instance type is an area that has been researched less. A system composed of different instance types is considered a *heterogeneous cloud*, and we refer to it as such throughout this paper. Homogeneous multi-instance clouds are widely used, although in some cases different instance types are used only in the context of accelerators (such as GPUs) [5]. Large parallel applications typically have heterogeneous computational demands, with some tasks performing more work than others, since work can usually not be divided in a perfectly equal way among all tasks. By using heterogeneous clouds, tasks with higher loads can be executed on faster, more expensive instances, while tasks that perform less work can be executed on slower and more cost-efficient instances. The approach of balancing the load between tasks [13], such as work stealing [11], may not be possible or efficient for all types of applications and algorithms.

In this paper, we investigate the cost efficiency of application running on heterogeneous cloud instances. Our main contribution is the introduction of a metric for analyzing the tradeoffs between price and performance of an execution, which we call the *cost-delay product (CDP)*, inspired by the energy-delay product (EDP) [10] and based on our previous work with heterogeneous clouds [15]. By using the *CDP* metric, users can measure and compare different clusters in terms of their cost efficiency. Furthermore, our metric allows the user to emphasize lower cost or higher performance, by using the C^2DP and CD^2P variations of the metric.

We evaluate a cloud composed of different Microsoft Azure instances using MPI-based benchmarks. Results show that heterogeneous clouds are able to execute parallel applications with a reduced cost, while maintaining a similar performance as homogeneous clouds. This leads to substantial improvements in cost efficiency. We also show the impact of changing the focus between shorter execution time and lower price, which helps cloud tenants to choose their desired tradeoff for different execution scenarios.

The rest of the paper is organized as follows. Section 2 discusses the characteristics of instances in major cloud providers, Microsoft and Amazon, which motivates our proposal of building heterogeneous clouds using different instance types. Section 3 introduces

UCC '17 Companion, , December 5–8, 2017, Austin, TX, USA.

ACM ISBN 978-1-4503-5195-9/17/12…$15.00
https://doi.org/10.1145/3147234.3148103

our metric to measure and adapt the cost efficiency, the cost-delay product (*CDP*). We evaluate the cost efficiency of a set of MPI applications running on heterogeneous clouds in Section 4. Section 5 contains an overview of related work. Finally, Section 6 summarizes our conclusions and discusses ideas for future work.

2 INSTANCE VARIATIONS IN PUBLIC CLOUDS

Two main public cloud providers of IaaS, Amazon's EC2 and Microsoft's Azure, have a large number of possible instance configurations. They provide instances from 1 core up to 128 cores, with different processor models. The available memory sizes range from less than 1 GB to up to 2 TB. Instances may come configured without a local disk, with increasing secondary memory sizes up to large local SSD drives.

Both providers group their instances such as to suit the purpose of the instances. The general purpose group presents machines with no specific optimization. Otherwise, there are groups of instances optimized for compute, memory, and storage roles. Amazon and Azure both provide groups of machines with accelerators as well. Azure has a group of HPC machines that provides Infiniband interconnections. Amazon does not provide Infiniband, but several machines of all groups support 10Gb Ethernet.

The price per hour of both providers presents a wide variation as well, with instances costing less than a cent to up to more than 14 dollars per hour. Table 1 summarizes the number of instance types per group and their price ranges.

In a previous work [16], we explored the tradeoff between price and performance of Amazon and Azure. We found that instances with higher prices do not always deliver higher performance than instances with cheaper price.

3 COST-DELAY PRODUCT: A METRIC TO MEASURE COST EFFICIENCY

An important aspect when evaluating cloud instance types is comparing their price/performance tradeoff. Several basic metrics have been proposed to evaluate this tradeoff. One metric that compares different configurations of the cloud is the cost efficiency [12, 14]. However, these simple metrics do not allow more comprehensive evaluations, for example, by placing more emphasis on price or on performance according to the user's preference.

Table 1: Instance characteristics of Microsoft Azure and Amazon EC2 in West USA datacenters (verified on July 25, 2017). Price ranges are given in US$/h.

Designation	Microsoft Azure		Amazon EC2	
	#Inst.	Price range	#Inst.	Price range
General Purpose	27	0.018–3.200	17	0.006–3.200
Compute Optimized	5	0.060–0.997	10	0.124–1.591
Memory Optimized	20	0.148–8.690	13	0.166–13.338
Storage Optimized	4	0.344–2.752	10	0.156–5.520
GPU/FPGA	7	0.900–4.370	8	0.900–14.400
HPC	6	0.971–2.136	—	—

We introduce a new metric, the *cost-delay product (CDP)*, which is modeled after the energy-delay product (EDP) [9, 10]. The basic *CDP* metric is defined as follows:

$$CDP = cost\ of\ execution \times execution\ time \quad (1)$$

The *cost of execution* represents the price of the environment used to execute the application. The majority of public cloud providers base their price model on hours of use, and the cost used in Equation 1 is the price per hour (in US$) of the instances allocated in a cloud provider. The *execution time* is the application's execution time in the allocated environment.

Lower values for the *CDP* indicate a better cost efficiency for a particular application in a particular environment. This metric can be used to directly compare two different cloud allocations, including in different providers. This basic version of the *CDP* resembles earlier proposals to quantify cost efficiency [14].

In Section 2 we observed that public cloud providers offer a wide range of instance configurations with different performance and price ranges. Therefore, the user can choose an environment that prioritizes performance or price. To express such a preference, the *CDP* metric can be extended by applying a weighted approach, depending on whether priority should be given to higher performance or lower cost. The resulting metrics, C^2DP and CD^2P, are defined as follows:

$$C^2DP = (cost\ of\ execution)^2 \times execution\ time \quad (2)$$

$$CD^2P = cost\ of\ execution \times (execution\ time)^2 \quad (3)$$

Since they represent different units, values of *CDP*, C^2DP, and CD^2P can not be directly compared between each other. However, a user can easily calculate the three metrics for his target environments and compare them by the three different aspects, with *CDP* meaning the best cost efficiency without focus on execution time or performance the C^2DP representing the less expensive environment and the CD^2P showing the configuration with better performance. With these metrics, a cloud tenant can select whether performance or cost is more important to him, and select the comparison function accordingly. We exemplify a weighted approach using a power of 2 to introduce the metrics, although different weights can be used to fine-tune the desired tradeoff.

4 EVALUATION

This section presents the evaluation of heterogeneous clouds using our proposed metrics. We begin with an overview of our methodology, followed by an analysis of the cost efficiency and performance results.

4.1 Methodology

4.1.1 Cloud instances. For our experiments, we chose the Microsoft Azure cloud, as it offers more instance types compared to Amazon AWS. We chose instances with 8 cores since they present the largest number of different configurations. We made several performance tests with these instances, so we could choose the instances to be used in our evaluation. Instance types D4 and F8 were selected to evaluate our proposal, where D4 presents higher performance and price as well. The main reason for this choice is

the benefit per cost of these instances. In our tests, these instances presented more processing power per US$ than others, which is desirable for a better cost efficiency. Each instance of D4 costs 0.559 US$ per hour and F8 costs 0.513 US$ per hour per instance.

We used 8 instances to create a cluster of 64 processing cores, evaluating all combinations of the D4 and F8 instance types for the three metrics, *CDP*, C^2DP, and CD^2P.

The Microsoft Azure location used to allocate the machines was "West US". Machines of all instance types were only allocated once and not reallocated between executions. Further experiments after deallocating and allocating new instances of the same types (not shown in the paper) resulted in quantitatively and qualitatively very similar behaviors.

4.1.2 Software. The execution environment were machines with Ubuntu server 16.04 OS, with Linux kernel 4.4. The MPI environment used was Open MPI [7], version 1.10. The benchmarks used for evaluation were the the MPI implementations of the NAS Parallel Benchmarks (NPB) [1], version 3.3.1. The input size was class C, which is a medium-large input size. We used all NAS programs except *DT*, because *DT* needs at least 85 MPI processes for class C. Each application was executed 10 times and the results show the average value.

4.1.3 Assigning MPI ranks to instances. In order to exploit load imbalance when running on heterogeneous nodes, we use the following algorithm to assign the MPI processes of each application to the instance types. First, we measure the number of instructions executed by each MPI process with Linux' `perf` tool [6]. Ranks are then sorted according to the number of instructions they execute. Finally, we assign the MPI processes with more instructions to the faster instance types, maintaining the same number of ranks on each node.

4.2 Results

4.2.1 Cost efficiency. The cost efficiency results of the NAS benchmarks are shown in Figure 1 for the *CDP*, C^2DP, and CD^2P metrics. In each of the graphs, the bars represent the cost efficiency for one metric when varying the number of instances of each type. The y axes show the normalized values of the three metrics, and the x axes indicate the mix of D4 and F8 instance types. As discussed in Section 3, the absolute values of the three metrics are not comparable between each other, and we normalize them to fit into the figures.

The most important result of these experiments is that heterogeneous clouds were the most cost efficient environments for the majority of the benchmarks for all three metrics. For the *CDP* and CD^2P metrics, heterogeneous environments are beneficial for five out of the eight benchmarks (*BT*, *EP*, *FT*, *MG*, and *SP*), while homogeneous environments are more appropriate for *CG*, *IS*, and *LU*. For the C^2DP metric, the same five benchmarks are more efficient, as well as the *IS* application.

When focusing on *CDP*, the five benchmarks have an increased cost efficiency ranging between 3.0% (*SP*) and 18% (*FT*) compared to the best cost efficiency of a homogeneous environment. For CD^2P, the cost efficiency is improved between 2.1% (*BT*) and 42.3% (*FT*). For C^2DP, cost efficiency increases range from 0.3% (*IS*) to 16.4% (*FT*). Over all benchmarks, cost efficiency was improved on average by 6.6%, 11.0%, and 10.8% for *CDP*, C^2DP, and CD^2P, respectively. These results show that cost efficiency can be improved substantially via heterogeneous clouds.

Another important result is that almost the whole spectrum of heterogeneous and homogeneous instances is the most cost efficient environment at least in one experiment. This shows that simple homogeneous clouds that do not take the specific application behavior into account can not result in optimal cost efficiencies.

Finally, it is necessary to point out that the most cost efficient mix of instance types not only changes between different benchmarks, but also between different metrics. A user that is interested in a different tradeoff might choose a different mix of instances to increase cost efficiency while reducing performance slightly.

An overview of the cost efficiency results for the NAS benchmarks is shown in Table 2. The table shows for each benchmark and metric the most cost efficient configuration and the cost efficiency gains of this configuration compared to the best homogeneous environment for each benchmark, as well as the average cost efficiency improvements over all benchmarks for each metric.

4.2.2 Performance. The performance analysis of the heterogeneous allocations is also an important aspect for the user, as increases in cost efficiency might cause a reduction of performance. Over all benchmarks, performance losses average 1.5% (when using

Table 2: Cost efficiency gains for the three metrics (*CDP*, C^2DP, CD^2P) compared to the best homogeneous configuration.

	CDP		C^2DP		CD^2P	
Application	Best configuration	Metric gains	Best configuration	Metric gains	Best configuration	Metric gains
BT	4-D4 4-F8	3.2%	1-D4 7-F8	8.9%	4-D4 4-F8	2.1%
CG	0-D4 8-F8	0.0%	0-D4 8-F8	0.0%	0-D4 8-F8	0.0%
EP	1-D4 7-F8	3.1%	1-D4 7-F8	2.0%	1-D4 7-F8	6.2%
FT	2-D4 6-F8	18.0%	1-D4 7-F8	16.4%	2-D4 6-F8	42.3%
IS	8-D4 0-F8	0.0%	1-D4 7-F8	0.3%	8-D4 0-F8	0.0%
LU	0-D4 8-F8	0.0%	0-D4 8-F8	0.0%	0-D4 8-F8	0.0%
MG	5-D4 3-F8	6.8%	1-D4 7-F8	4.7%	5-D4 3-F8	10.6%
SP	5-D4 3-F8	3.0%	1-D4 7-F8	2.8%	5-D4 3-F8	2.8%
Average	—	4.7%	—	6.2%	—	8.6%

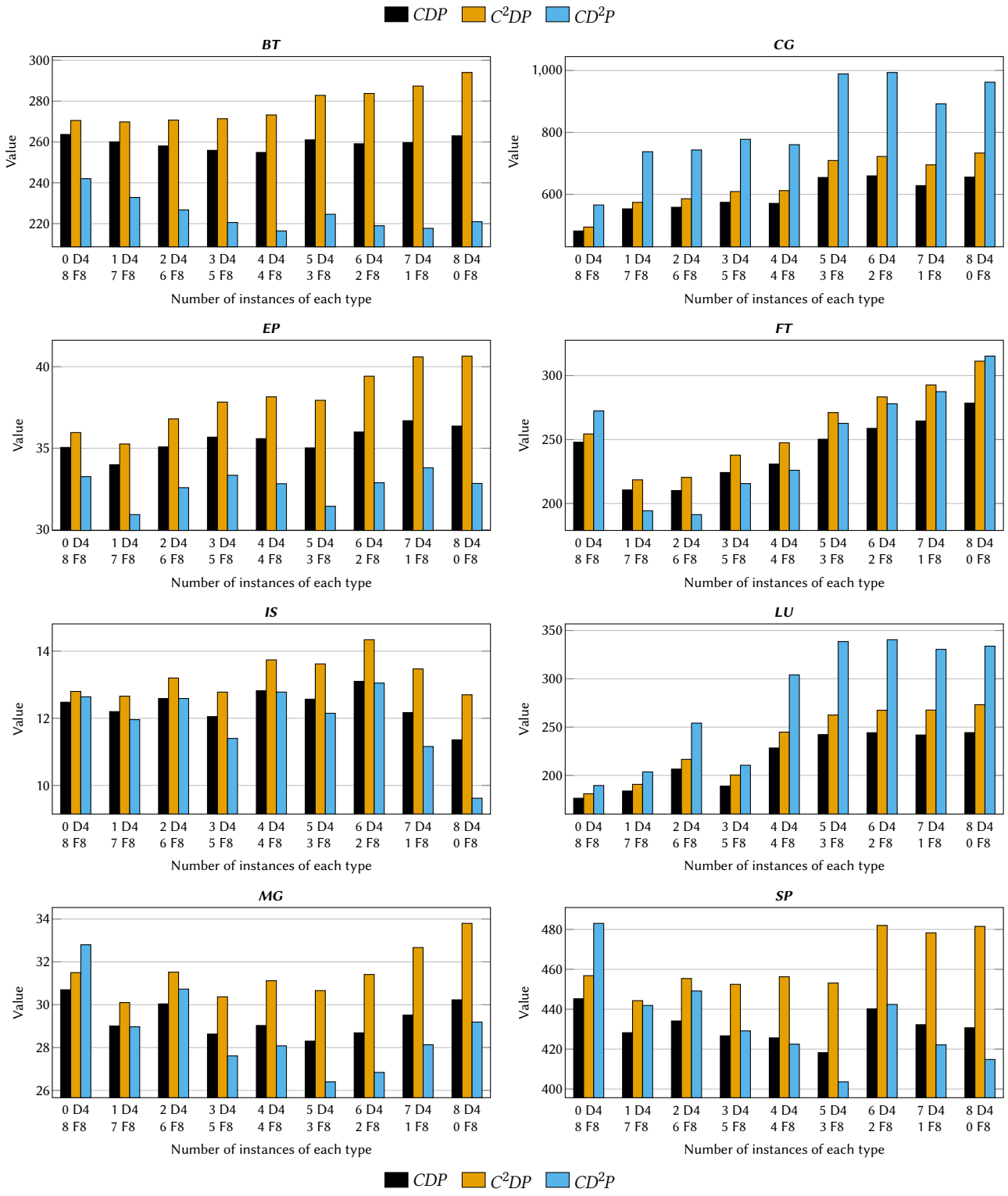

Figure 1: Cost efficiency results for the NAS benchmarks on the D4 and F8 instances for different combinations of instances. Lower values indicate a higher cost efficiency.

the cloud system with the best *CDP* or CD^2P) and 2.1% (when using the cloud system with the best C^2DP), compared to the fastest homogeneous cloud for each application. As expected, performance was impacted the most when focusing on C^2DP, as the price of execution received more importance. It is important to note that when comparing the performance loss with the cost efficiency gain, the cost efficiency gain is higher than the performance loss for all three metrics. This indicates that the cost efficiency improvements do not come with an exaggerated impact on performance, and therefore present a reasonable optimization possibility for the user.

5 RELATED WORK

Existing studies on improving the cost efficiency or performance in heterogeneous cloud environments focus on two types of situations, (1) benefits on the provider side, and (2) benefits for the end user.

5.1 Provider-side improvements

Yeo et al. [20] used mathematical models to analyze how the service provider could mitigate the performance impact of hardware heterogeneity to the final user. However, their models use information about the underlying infrastructure which is not available to the user to improve the cost/efficiency of their applications.

Zhang et al. [23] developed a dynamic capacity provisioning manager for datacenters with heterogeneous hardware. They used a clustering approach to divide tasks with similar workloads to improve hardware utilization. The technique dynamically adjust the number of active bare machines to minimize provider costs. Their work takes the heterogeneity of VMs into account to characterize their workload, but their focus is on improvements from the provider perspective.

Zhang et al. [22] implemented task scheduling algorithms to reduce energy waste in cloud computing servers. Their algorithms adjust processor speeds on the server and allocate tasks in such servers to improve energy consumption. Based on simulations, they conclude that assigning more tasks to cloud servers using minimum energy is more beneficial for overall energy consumption than using a randomly selected cloud server. Their work take into account hardware heterogeneity but uses information unavailable to the end user.

5.2 End-user improvements

Gupta et al. [8] propose a technique to improve the performance of parallel applications in the cloud with task placement. The authors place the tasks according to the interference between different applications by analyzing their cache memory usage, and from a description provided by the user. They do not take different types of instances into account.

Zhang et al. [24, 25] exploited cloud heterogeneity in several MapReduce clusters to select the best cost/performance deployment. They simulate their configurations of 3 instance sizes looking to obtain the same application performance but with different provisioning costs. The validation was done on Amazon using MapReduce jobs with no data dependencies between them. Their results showed a difference in cost when using homogeneous or heterogeneous deployments. For some of the applications evaluated they obtained significant cost savings. Our work include MPI applications, and benchmarks with communication between instances.

Carreño et al. [4] created a communication-aware task mapping for cloud environments with multiple instances. Their work analyzes heterogeneity in communication between the tasks and in the network interconnections between cloud instances. They use this information to map tasks that communicate a lot to faster instances, improving inter-instance communication performance. However, their work uses the same type of VMs for each execution and they do not take computational performance into account. In our work, we compare the performance when mixing different types of VMs.

Wang et al. [19] developed a task-level scheduling algorithm to comply with budget and deadline constrains. They analyze heterogeneity as the variety of options of virtual machines from a provider and the underlying variations in hardware that exists for each instance. They developed a parallel greedy algorithm that improves deployment to comply with the constrains. Their work is different because it does not try to optimize the cost/efficiency of the solution but tries to respect the user constrains. Also their work was not validated using an actual public cloud infrastructure.

Su et al. [18] developed a cost-efficient task scheduling algorithm for executing large programs in the cloud. Their strategy was to map tasks to cost efficient VMs while preserving the expected performance of the application. Their algorithm decides which instance produce the best ratio, but is limited because does not select heterogeneous VMs instances, only same type from the provider offering. They were able to improve the scheduling time, but validate their approach using simulation only.

6 CONCLUSIONS

The use of the cloud as an environment for parallel applications execution is interesting due the well-know benefits of the model: No upfront costs and elasticity. The execution environment of parallel applications is normally a homogeneous cluster, composed of a number of machines with the same configuration, price, and performance. However, cloud providers offer large numbers of instances types, each one with different purposes, for example memory-bound applications.

We introduced a metric, the cost-delay product (CDP), that helps to analyze performance/price tradeoffs and validated it to be used to exploit the nonuniform application behavior and cloud heterogeneity for an improved cost efficiency. This metric can also be used to give emphasis on price or performance, depending on the user's preference. Our evaluation with MPI-based applications on an Azure cloud shows that the cost efficiency can be improved significantly, by up to 42.3%, depending on the application and the employed metric. Six out of the eight applications we tested improve their cost efficiency with a heterogeneous cloud. We conclude that the majority of parallel applications could improve their cost efficiency by using heterogeneous cloud instances. Despite the better cost efficiency, application performance was affected only slightly.

For the future, we will extend our analysis to consider more than two different types of instances with different numbers of cores on each type.

Acknowledgments

This research received funding from the EU H2020 Programme and from MCTI/RNP-Brazil under the HPC4E project, grant agreement no. 689772. This research received partial funding from CYTED for the RICAP Project. Additional funding was provided by FAPERGS in the context of the GreenCloud Project.

REFERENCES

[1] David H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simon, V. Venkatakrishnan, and S. K. Weeratunga. 1991. The NAS Parallel Benchmarks. *International Journal of Supercomputer Applications* 5, 3 (1991), 66–73. https://doi.org/10.1177/109434209100500306

[2] Christine Bassem and Azer Bestavros. 2015. Network-Constrained Packing of Brokered Workloads in Virtualized Environments. In *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*.

[3] Abhinav Bhatele, Andrew R. Titus, Jayaraman J. Thiagarajan, Nikhil Jain, Todd Gamblin, Peer-Timo Bremer, Martin Schulz, and Laxmikant V. Kale. 2015. Identifying the Culprits Behind Network Congestion. In *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. 113–122. https://doi.org/10.1109/IPDPS.2015.92

[4] Emmanuell D. Carreño, Matthias Diener, Eduardo H. M. Cruz, and Philippe O. A. Navaux. 2016. Communication Optimization of Parallel Applications in the Cloud. In *IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid)*. 1–10. https://doi.org/10.1109/CCGrid.2016.59

[5] S. P. Crago and J. P. Walters. 2015. Heterogeneous Cloud Computing: The Way Forward. *Computer* 48, 1 (Jan 2015), 59–61. https://doi.org/10.1109/MC.2015.14

[6] Arnaldo Carvalho de Melo. 2010. The New Linux 'perf' Tools. In *Linux Kongress*.

[7] Edgar Gabriel, Graham E. Fagg, George Bosilca, Thara Angskun, Jack J. Dongarra, Jeffrey M. Squyres, Vishal Sahay, Prabhanjan Kambadur, Brian Barrett, Andrew Lumsdaine, Ralph H. Castain, David J. Daniel, Richard L. Graham, and Timothy S. Woodall. 2004. Open MPI: Goals, concept, and design of a next generation MPI implementation. In *Recent Advances in Parallel Virtual Machine and Message Passing Interface (PVMMPI)*. 97–104. https://doi.org/10.1007/978-3-540-30218-6_19

[8] Abhishek Gupta, Laxmikant V. Kalé, Dejan Milojicic, Paolo Faraboschi, and Susanne M. Balle. 2013. HPC-aware VM placement in infrastructure clouds. In *IEEE International Conference on Cloud Engineering (IC2E)*. 11–20. https://doi.org/10.1109/IC2E.2013.38

[9] Mark Horowitz, Thomas Indermaur, and Ricardo Gonzalez. 1994. Low-power digital design. In *Low Power Electronics, 1994. Digest of Technical Papers., IEEE Symposium*. 8–11. https://doi.org/10.1109/LPE.1994.573184

[10] James H. Laros III, Kevin Pedretti, Suzanne M. Kelly, Wei Shu, Kurt Ferreira, John Vandyke, and Courtenay Vaughan. 2013. Energy Delay Product. In *Energy-Efficient High Performance Computing: Measurement and Tuning*. Vol. 3. 51–55. https://doi.org/10.1007/978-1-4471-4492-2

[11] Seung-Jai Min, Costin Iancu, and Katherine Yelick. 2011. Hierarchical work stealing on manycore clusters. In *Conference on Partitioned Global Address Space Programming Models*. 1–10.

[12] Ioannis A. Moschakis and Helen D. Karatza. 2012. Evaluation of gang scheduling performance and cost in a cloud computing system. *The Journal of Supercomputing* 59, 2 (01 Feb 2012), 975–992. https://doi.org/10.1007/s11227-010-0481-4

[13] Olga Pearce, Todd Gamblin, Bronis R. de Supinski, Martin Schulz, and Nancy M. Amato. 2012. Quantifying the effectiveness of load balance algorithms. In *ACM International Conference on Supercomputing (ICS)*. 185–194. https://doi.org/10.1145/2304576.2304601

[14] Eduardo Roloff, Matthias Diener, Alexandre Carissimi, and Philippe O. A. Navaux. 2012. High Performance Computing in the Cloud: Deployment, Performance and Cost Efficiency. In *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. 371–378.

[15] Eduardo Roloff, Matthias Diener, Emmanuell Diaz Carreño, Luciano Paschoal Gaspary, and Philippe O. A. Navaux. 2017. Leveraging Cloud Heterogeneity for Cost-Efficient Execution of Parallel Applications. In *Euro-Par 2017: Parallel Processing - 23rd International Conference on Parallel and Distributed Computing, Santiago de Compostela, Spain, August 28 - September 1, 2017, Proceedings*. 399–411. https://doi.org/10.1007/978-3-319-64203-1_29

[16] E. Roloff, M. Diener, L. P. Gaspary, and P. O. A. Navaux. 2017. HPC Application Performance and Cost Efficiency in the Cloud. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*. 473–477. https://doi.org/10.1109/PDP.2017.59

[17] Abdallah Saad and Ahmed El-Mahdy. 2013. Network Topology Identification for Cloud Instances. In *International Conference on Cloud and Green Computing*. 92–98. https://doi.org/10.1109/CGC.2013.22

[18] Sen Su, Jian Li, Qingjia Huang, Xiao Huang, Kai Shuang, and Jie Wang. 2013. Cost-efficient task scheduling for executing large programs in the cloud. *Parallel Comput.* 39, 4–5 (2013), 177 – 188. https://doi.org/10.1016/j.parco.2013.03.002

[19] Y. Wang and W. Shi. 2014. Budget-Driven Scheduling Algorithms for Batches of MapReduce Jobs in Heterogeneous Clouds. *IEEE Transactions on Cloud Computing* 2, 3 (July 2014), 306–319. https://doi.org/10.1109/TCC.2014.2316812

[20] S. Yeo and H. H. Lee. 2011. Using Mathematical Modeling in Provisioning a Heterogeneous Cloud Computing Environment. *Computer* 44, 8 (Aug 2011), 55–62. https://doi.org/10.1109/MC.2011.96

[21] Bassem El Zant and Maurice Gagnaire. 2015. Performance and Price Analysis for Cloud Service Providers. In *Science and Information Conference (SAI)*. 816–822.

[22] Luna Mingyi Zhang, Keqin Li, and Yan-Qing Zhang. 2010. Green Task Scheduling Algorithms with Speeds Optimization on Heterogeneous Cloud Servers. In *Proceedings of the 2010 IEEE/ACM Int'L Conference on Green Computing and Communications & Int'L Conference on Cyber, Physical and Social Computing (GREENCOM-CPSCOM '10)*. IEEE Computer Society, Washington, DC, USA, 76–80. https://doi.org/10.1109/GreenCom-CPSCom.2010.70

[23] Qi Zhang, Mohamed Faten Zhani, Raouf Boutaba, and Joseph L. Hellerstein. 2013. Harmony: Dynamic Heterogeneity-Aware Resource Provisioning in the Cloud. In *Proceedings of the 2013 IEEE 33rd International Conference on Distributed Computing Systems (ICDCS '13)*. IEEE Computer Society, Washington, DC, USA, 510–519. https://doi.org/10.1109/ICDCS.2013.28

[24] Zhuoyao Zhang, Ludmila Cherkasova, and Boon Thau Loo. 2014. Exploiting Cloud Heterogeneity for Optimized Cost/Performance MapReduce Processing. In *Proceedings of the Fourth International Workshop on Cloud Data and Platforms (CloudDP '14)*. ACM, New York, NY, USA, Article 1, 6 pages. https://doi.org/10.1145/2592784.2592785

[25] Zhuoyao Zhang, Ludmila Cherkasova, and Boon Thau Loo. 2015. Exploiting Cloud Heterogeneity to Optimize Performance and Cost of MapReduce Processing. *SIGMETRICS Perform. Eval. Rev.* 42, 4 (June 2015), 38–50. https://doi.org/10.1145/2788402.2788409