

Validating communication network configurations in cloud and HPC systems using Metamorphic Testing

PABLO C. CAÑIZARES, Universidad Autónoma de Madrid, Spain

MIGUEL PÉREZ, Universidad Complutense de Madrid, Spain

ALBERTO NÚÑEZ, Universidad Complutense de Madrid, Spain

ROSA FILGUEIRA, School of Computer Science, University of St. Andrews, United Kingdom

During the last years, the fast evolution of computers and networks has led to the creation of a wide variety of services that have changed the way we live, like video streaming, on-line gaming and online shopping. These services are supported by complex systems, which require not only high computational power but high-speed and low-latency networks to fulfil the expected quality requirements. However, a misleading configuration in one of the thousand components that compose these systems may cause performance bottlenecks and functioning disruptions. Unfortunately, conventional testing methods are not adequate for checking these systems since, on many occasions, there does not exist a mechanism to determine if the behaviour of a system is the expected one. Fortunately, Metamorphic Testing is a valuable and promising testing technique that alleviates the two fundamental problems of testing: the *oracle problem* and the *reliable test set problem*.

In this paper, we combine Metamorphic Testing and simulation techniques for validating communication network configurations in HPC systems. For this, we rely on a catalogue of Metamorphic Relations, based on network communications knowledge, for checking its correctness. In addition, we have conducted an experimental study for analysing the communication network of HPC systems. The results show that Metamorphic Testing is appropriate for checking the correctness of communication networks supported by complex topologies in HPC systems.

CCS Concepts: • **Networks** → **Network simulations**; • **Software and its engineering** → **Software testing and debugging**.

Additional Key Words and Phrases: Communication networks, Simulation, Metamorphic testing, High Performance Computing

ACM Reference Format:

Pablo C. Cañizares, Miguel Pérez, Alberto Núñez, and Rosa Filgueira. 2022. Validating communication network configurations in cloud and HPC systems using Metamorphic Testing. 1, 1 (March 2022), 11 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

The revolution of computers and communication networks in the last decades has led to the creation of a wide variety of services that benefits society, shape and change the world [13]. Due to globalisation, these services must be distributed in such a way that they can be accessed by different demographic locations around the world. At present, there are many use cases that should be mentioned, such as, remote surgery [18], disease predictions [33], banking [30], and manufacturing [12]. All the aforementioned cases, not only require platforms

Authors' addresses: Pablo C. Cañizares, Universidad Autónoma de Madrid, Madrid, Spain, pablo.cerro@uam.es; Miguel Pérez, Universidad Complutense de Madrid, Madrid, Spain, miguel06@ucm.es; Alberto Núñez, Universidad Complutense de Madrid, Madrid, Spain, alberto.nunez@pdi.ucm.es; Rosa Filgueira, School of Computer Science, University of St. Andrews, St Andrews, United Kingdom, rf208@st-andrews.ac.uk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

XXXX-XXXX/2022/3-ART \$15.00

<https://doi.org/XXXXXXX.XXXXXXX>

providing a high level of computational power but also high-speed and low-latency communication networks to fulfil the rigorous requirements for their proper functioning. In addition, in some cases, these systems must handle the massive data generated by thousands of users concurrently accessing to these services.

High Performance Computing solutions (HPC) and data centres that support cloud systems, provide a suitable trade-off between performance-cost ratio and, in the major part of the cases, these are supported by high performance communication networks, which leads to achieve high throughput while maintaining low end-to-end latency. However, these systems usually consist of complex architectures that may not be allow a dedicated access to the hardware resources. In general, having a dedicated access to the system resources (i.e. communication network, computing nodes and storage nodes) to carry out an experimental study is not possible due to the multi-user nature of the target system. Thus, the deployment of different applications executed by other users may interfere in the measurements taken from the system under study. In addition, the concurrent access of multiple users may hamper the repeatability of the experiments to analyse the system.

Simulation techniques have been applied – during the last decades – by the research community to tackle these challenges [20, 31], which provides multiples advantages: (i) flexibility to design a wide spectrum of system architectures; (ii) it is not necessary to purchase specific hardware to execute the simulations since, in most cases, simulators can be executed in regular computers; (iii) simulated environments can represent scenarios with exclusive access to the system under study, which allows reproducing experiments in a controlled way.

Nowadays, there does exist a high number of simulation tools that allows to represent the behaviour of a great variety of scenarios, from individual components such as CPUs [29], and memories [16], to complete distributed system architectures [2]. These platforms enable modelling and representing the behaviour of both architecture and running applications in highly distributed environments.

Once the system under study is modelled and configured, it is necessary to check whether its behaviour corresponds to the expected one. Although it is possible to simulate the target system to analyse its overall functioning – for a few number of particular cases – by executing a reduced number of tests created manually by the tester, a large number of potential errors may remain unexplored. To face this issue, testing techniques can be applied. Testing is considered one of the most extended techniques to validate the correctness of systems [1, 5]. However, HPC and cloud systems often consist of thousands of computational elements, interconnected through complex network topologies, which hampers the applicability of traditional testing techniques. Indeed, an *oracle* to validate the behaviour of a system is often missing or it is computationally unaffordable. In testing, an *oracle* is a mechanism that determines if a test is correct, or not. Moreover, selecting a suitable test suite to validate the system in a reasonable time is also challenging. Metamorphic Testing (MeT) is a testing technique that uses expected knowledge of the system under study for results verification and for generating directed test suites [10], which can be applied to test complex systems. In the current literature we found few proposals that use MeT techniques for analysing general aspects of communication networks, and all of them address it in a superficial way. To the best of our knowledge, there does not exist any proposal that analyses – in detail – advanced topologies of HPC systems.

In this paper, we combine simulation and MeT techniques for validating network configurations in HPC and cloud systems. For this, we propose a catalogue of novel Metamorphic Relations (MRs) to analyse the communication networks of large-scale supercomputers. In particular, we focus the study in the *TOR* (Top-of-Rack) topology [26]. In order to check the feasibility of the proposal, we have conducted an experimental case study in which the MeT process has been carried out over this topology.

The remainder is structured as follows: First, in Section 2, we provide a preliminary background of MeT. Section 3 presents an overview of the existing approaches focusing on MeT techniques for testing communication networks. Next, we describe the contribution presented in this paper in Section 4. The experimental study to support the suitability and effectiveness of the proposal is presented in Section 5. Finally, conclusions and some lines of future work are presented in Section 6.

2 METAMORPHIC TESTING

Conventional testing methods require checking whether the output(s) returned by the system under test are the expected ones, or not. Schematically, let S be a system, I be the input domain and χ be a test selection strategy. Let $\mathcal{T} = \{t_1, t_2, \dots, t_n\} \subseteq I$ be the set of tests generated by using χ . When these tests are sequentially applied to the system S we obtain a sequence of outputs $S(t_1), S(t_2), \dots, S(t_n)$. Hence, if we have an oracle, called f , to check whether the output of S when applying any test of \mathcal{T} is the expected one, then we find an error if there exists $t_i \in \mathcal{T}$ such that $S(t_i) \neq f(t_i)$.

Testing has to deal with two fundamental problems: the *oracle problem* [32] and the *reliable test set problem* [3]. The former concerns the availability of an oracle, that is, a mechanism to check whether the behaviour of the system under study is correct or not. However, in some situations, the oracle is not available or its application is pragmatically unattainable. The latter refers to the need of generating an appropriate test suite for determining the correctness of the system under study. Since executing all the possible test cases over a system is considered computationally unaffordable, it is necessary to prioritise test cases, creating subsets of appropriated test suites.

MeT techniques can be applied to face these problems. In essence, MeT uses expected properties of the target systems, relating multiple test inputs/observed outputs for validating its correctness. These properties are formulated as Metamorphic Relations (MRs). An MR is a property of the system under study that entail multiple inputs and their outputs. Let us present an MR as a formula: $i(MR) \rightarrow o(MR)$, where $i(MR)$ refers to the relation between the input parameters of the test cases, and $o(MR)$ refers to the relation that must be fulfilled by the outputs obtained from the test cases. Thus, MRs can be used as an oracle, to determine if a test is correct, or not, and to create quality test cases focusing on the properties reflected in the MRs, making MeT suitable to face both fundamental problems of testing [10, 35].

For the sake of clarity, let us introduce an example for checking the implementation correctness of $\cos(x)$. It is widely known that calculating specific values of this function requires considerable efforts (e.g. $\cos(0, 31) = 0.95233\dots$). Therefore, MeT uses expected properties of the function to tackle this issue, such as $MR = \cos(x) = \cos(x + 2\pi)$, which always holds when randomly varying x . Hence, a failure in $\cos(x)$ is detected whether the results of the function $\cos(x)$ and its variant $\cos(x + 2\pi)$ differ.

3 RELATED WORK

This section summarises the existing proposals in the literature that are related to this work. Over the last few decades, different simulation tools focused on modelling and analyzing the network communication of distributed systems have been proposed by the research community. Among the most important ones, it is worth mentioning CloudSim [3], GreenCloud [17], CloudExp [15] and SIMCAN [24].

Considered as a *de facto* simulation platform for cloud computing environments, CloudSim [3] is particularly relevant. This approach is a simulation toolkit that supports modelling key aspects of cloud systems such as data centres, jobs queues, policies, and the communication between the different entities that compose the computational environment. Despite the extended functionality of CloudSim, it still has some drawbacks. As an example, it does not provide mechanisms for creating communications between VMs in several data centres, hence limiting the communication network capabilities.

GreenCloud is a simulation platform that focuses on modelling and simulating the communication network of distributed systems. Since it is built on the top of the NS2 network simulator [14], GreenCloud is able to define the intercommunication of the processes at a packet level. Moreover, GreenCloud supports plugins extensions to reproduce physical layer traces to increase the detail of the experiments. Regarding the experiments, this platform does not support heterogeneous and multi-cpu environments. In addition, it is necessary to implement them by using C++ and OTel for a single use case. The main drawback of this simulator is the time required to perform

the simulations. Since most of the proposals carry out the simulations in the range of seconds, GreenCloud needs from minutes to hours.

CloudExp is a CloudSim extension to model and simulate cloud computing environments. This platform relies on its flexibility, being able to model and analyse a wide range of components, like data centres, storage systems, communication networks, virtualisation, Service Level Agreement, Service Oriented Architectures and business process management.

SIMCAN is a simulation platform – built on the basis of OMNeT++ and INET – for modelling and simulating large scale systems, such as HPC clusters and cloud data centres [24]. SIMCAN supports simulating MPI applications by providing an MPI-based API for executing them in the modelled architecture. Regarding the usability, it provides a GUI for graphically modelling distributed systems.

Although these simulators support the modelling and simulation of communication networks for a wide variety of systems, the validation of their models must be done manually. That is, these proposals do not provide testing mechanisms for automatically checking the correctness of the network communications of the systems under study.

During the last years, MeT [10, 35] has been applied in a great variety of areas such as hardware validation [4, 28], simulation [22, 27], security [21], and machine learning [8, 34]. The versatility provided by MeT allows us to analyse network communications in distributed systems, such as HPC clusters and cloud environments.

To the best of our knowledge, few proposals found in the current literature apply MeT for this purpose. Chen et al. propose a conformance testing of network protocols using MeT [11]. In this approach, the authors provide a set of MRs based on low-level properties of the network protocol to ensure that its behaviour corresponds with the expected one. Specifically, in the experimental phase, the ad-hoc on-demand distance vector protocol is selected for testing its conformance against the AODV protocol[9]. Although the proposal includes general MRs focused on low-level aspects (i.e. data packets, retransmissions, and routing tables) that are applicable to network protocols, it is not focused on the architectural components of distributed systems. This fact hampers its application in both HPC and cloud systems, since these usually consist of thousands of interconnected components, where a misconfiguration of any of them may cause performance issues.

Núñez and Hierons proposed a methodology – based on MeT and simulation – for validating and testing cloud systems [25]. To that end, the authors establish the theoretical foundations for formulating MRs focused on the architecture of the cloud, and analysing performance and energy aspects. Afterwards, as a result of improving that methodology, the authors presented TEA-Cloud [23], a complete framework for testing cloud systems. The proposal provides an extended collection of MRs, which focus on key aspects of the cloud, such as VMs, tenants, cloud providers, and analyse functional and non-functional aspects of the system. The experimental phase of these works demonstrates the applicability of MeT, in combination with simulation, to check the correctness of cloud systems by finding different synthetic errors. However, a reduced number of MRs, from the proposed catalogue of MRs, is focused on analysing the communication networks of the cloud. In contrast to the proposal presented in this paper, none of these works analyses in detail specific aspects of the topologies that support HPC systems.

Alternatively, MT-EA4Cloud [7] focuses on optimising the energy consumption of cloud computing systems. Evolutionary algorithms, simulation and MeT techniques are combined for evolving cloud system configurations towards energy awareness. Hence, all the proposed MRs, mutation and crossover operators are targeted at modifying different parts of the cloud to reduce power consumption. Even though MT-EA4Cloud combines simulation and MeT in cloud systems, it is oriented to reduce the energy consumption of the systems under study by modifying particular elements such as disks, cpus, and memories, among others, so it differs from the main objective of this work.

Although the previously presented works report promising results, their targets differ from the main objective pursued in this paper. Consequently, to the best of our knowledge, there does not exist, in the literature, any

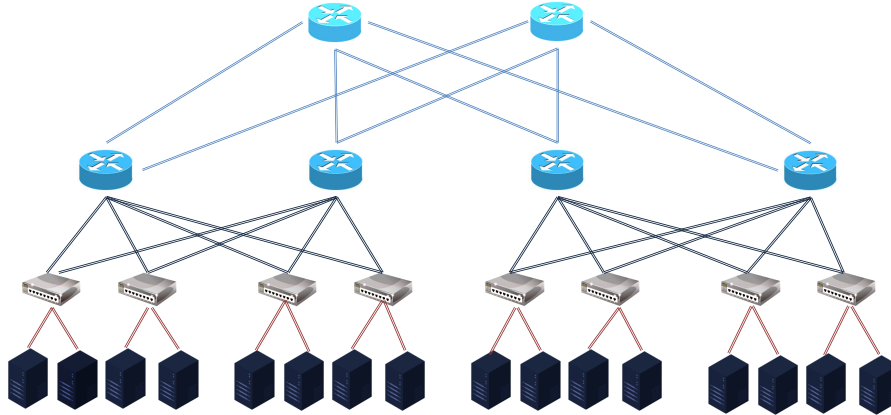


Fig. 1. Example of a TOR topology

proposal that addresses the checking of the proper functioning of network communications – supported by complex topologies – in HPC systems.

4 SIMULATING COMMUNICATION NETWORKS IN HPC ENVIRONMENTS

High Performance Computing clusters, and the data centres that support cloud systems, consist of a large number of computational nodes, which are interconnected through communication networks. Thus, a high number of tenants are able to launch multiple applications concurrently in the system, hence sharing some resources, like the network. In this context, checking whether the correct behaviour of these systems corresponds with the expected one is complex, since the *oracle* and *reliable test set* problems hinder the applicability of traditional testing techniques. To attack these problems, we combine simulation and MeT techniques for modelling, simulating and testing the communication network of the system under study.

In this section, we provide a set of novel MRs, which are especially created to analyse the proper functioning of communication networks in a wide variety of architectures. Since the main objective of this work is to check the correctness of communication networks in HPC systems, we have considered one of the most popular topologies for building interconnection networks in large-scale supercomputers, that is, TOR (Top-of-Rack).

In order to support a flexible and accurate configuration of the topologies, we use the following notation. Let be s a test case that represents the target HPC system to be analysed, which provides details related to the underlying architecture of the system architecture. Similarly, a follow-up test case is represented by s' . In essence, a follow-up is generated by applying a slight modification over the source test case. Since the behaviour of the system is analysed using simulation, let be $\mathcal{S}(s)$ the result of the simulation for simulating the system represented in s using the simulator \mathcal{S} . We denote an MR as a tuple (MR_i, MR_o) , where MR_i represents a relation between two test cases, and MR_o denotes a relation between the results obtained from the execution of these test cases. It is worth noting that these test cases must fulfil the input relation MR_i . Hence, a metamorphic relation $MR_{\mathcal{S}}$ for testing s using the simulator \mathcal{S} , can be formally represented as a set of 4-tuples:

$$MR_{\mathcal{S}} = \{((s), (s'), \mathcal{S}(s), \mathcal{S}(s')) | MR_i((s), (s')) \rightarrow MR_o(\mathcal{S}(s), \mathcal{S}(s'))\}$$

The following notation has been used in the definitions of MRs:

- s_{netBan} denotes the maximum theoretical bandwidth of the communication network of s .
- s_{netLat} denotes the theoretical latency of the communication network of s .

- s_{swi} denotes the number of switches in s .
- s_{proc} denotes the number of processes executed in s .
- s_{link} denotes the number of communication links in s .
- $|s|$ denotes the number of physical machines in s .
- $\Theta(\mathcal{S}(s))$ represents the energy consumption required to simulate s .
- $\Lambda(\mathcal{S}(s))$ represents the data-centre load average of s .
- $\Gamma(\mathcal{S}(s))$ denotes the network load average of s .

For modelling the underlying infrastructure of HPC systems we define a catalogue of MRs. Next, we discuss its intuitive meaning.

- **MR₁**: $s_{swi} > s'_{swi} \wedge |s| = |s'| \rightarrow \Theta(\mathcal{S}(s)) \geq \Theta(\mathcal{S}(s')) \wedge \Gamma(\mathcal{S}(s)) \leq \Gamma(\mathcal{S}(s'))$
- **MR₂**: $s_{netLat} < s'_{netLat} \rightarrow \Theta(\mathcal{S}(s)) \leq \Theta(\mathcal{S}(s'))$
- **MR₃**: $s_{netBan} > s'_{netBan} \rightarrow \Theta(\mathcal{S}(s)) \leq \Theta(\mathcal{S}(s')) \wedge \Gamma(\mathcal{S}(s)) \leq \Gamma(\mathcal{S}(s'))$
- **MR₄**: $s_{proc} < s'_{proc} \rightarrow \Theta(\mathcal{S}(s)) \leq \Theta(\mathcal{S}(s')) \wedge \Lambda(\mathcal{S}(s)) \leq \Lambda(\mathcal{S}(s'))$
- **MR₅**: $s_{link} > s'_{link} \rightarrow \Gamma(\mathcal{S}(s)) \leq \Gamma(\mathcal{S}(s'))$

MR₁: If the number of switches that compose the network of s is greater than the number of switches of the network of s' , and the number of nodes of s and s' are equals, then the energy required to simulate s should be greater or equal than the energy required to simulate s' and the network load average of s should be lesser or equal than the network load average of s' .

MR₂: If the theoretical latency of the communication network of s is lesser than the latency of the network of s' , then the energy required to simulate s should be lesser or equal to the energy required to simulate s' .

MR₃: If the theoretical performance of the network of s is greater than the performance of the network of s' , then the energy required to simulate s should be lesser or equal to the energy required to simulate s' and the network load average of s should be lesser or equal than the network load average of s' .

MR₄: If the number of processes executed in s is lesser than the number of processes executed in s' then the energy required to simulate s should be lesser or equal than the energy required to simulate s' , and the load average of s should be lesser or equal than the load average of s' .

MR₅: If the number of links that interconnects the nodes with the switches in s is greater than the number of links that interconnects the nodes with the switches of s' , then the network load average of s should be lesser or equal than the network load average of s' .

5 EXPERIMENTS

This section presents an experimental study for evaluating the applicability and effectiveness of our proposal. The main goal of this study is to check the correctness of the communication network of HPC systems. For this, we have applied simulation techniques, for simulating HPC environments, in combination with MeT, for verifying the correctness of their results.

We have selected the GreenCloud platform for modelling and simulating an HPC system supported by the TOR network topology, which is widely used for building these systems. Let us remark that the main objective of this approach is to *automatically* check the network behaviour of the system under study. To that end, we automatically generate test cases and compare the obtained result to determine the correctness of the system under test. This approach is *automatic* in the sense that the testing process is carried out without the intervention of a human user. That is, once the MRs are designed, and the source model – representing the system under study – is created, the testing process can be performed automatically.

In this experimental study, we use the catalogue of MRs presented in Section 4. These experiments have been performed in a HPC cluster with 280 nodes, which are interconnected through an Infiniband fabric network.

Each node contains two 18-core Intel(R) Xeon CPU@ 2.1 Ghz with hyper-threading, 256 GB of RAM and 406TB Lustre file system.

The modelled HPC system contains 1024 physical machines, which are interconnected through a communication network consisting of a 3-layers of switches, namely *Core level*, *Aggregation level* and *Access level*. Thus, as it is depicted in Figure 1, the *Core level* contains 2 switches, while *Aggregation level* and *Access level* contain, respectively, 4 and 8 switches. The physical machines are distributed among 64 racks. Each communication link has been configured with a latency of 0.0033 ms. However, each layer has been configured with different bandwidth. Thus, the *Core level* has a bandwidth of 100Gbps, while the bandwidth of the *Aggregation level* and *Access level* is 10 Gbps and 1 Gbps, respectively. For each MR, 100 different follow-up test cases have been automatically generated. From now on, we refer to *source test case* as the test case that represents the system under study.

Table 1 shows the results obtained from the ten most representative test cases. Let us remark that, for clarity purposes, this table shows only the most relevant output parameters obtained from the simulations. The first column shows the test case ID, the next three columns show the total energy consumption (in Wh) for the different switch layers, that is, *Core level*, *Aggregation level* and *Access level*. Next, the column labelled as *Energy Consumption* depicts the overall energy consumed by the system (in Wh), while the last two columns present the overall load (in %) of the computing resources of the data centre and the communication network. For the sake of clarity, the first row (with ID 0), represents the source test case, while the rest represent the generated follow-up test cases.

These results show how the modifications applied in the source test case – to create the follow-up test cases – have a direct impact in the obtained outputs. Specifically, the test cases 1–19 have been created by increasing the number of switches of the system (s'_{swi}). Hence, the energy consumption of the different switch levels increases ($\Theta(\mathcal{S}(s'))$), while decreasing the network load average ($\Gamma(\mathcal{S}(s'))$). The test case 29 has been created by increasing – with respect to the source test case – the theoretical latency of the system (s'_{netLat}). However, the simulation platform – supporting this feature to represent the target system – shows almost identical results for this test case and for the source test case and, therefore, the expected behaviour of the system is not reflected in these results. The test cases 35–58 have been generated by decreasing the bandwidth (s'_{netBan}), which is reflected in an increment over the network load average ($\Gamma(\mathcal{S}(s'))$). Finally, test cases 74 and 77 contain a higher number of processes (s'_{proc}) than the ones defined in the source test case, which is reflected in an increment of the data-centre and network load average.

We use these results to check if the MRs are satisfied, or not. The general results of the testing process are presented in Table 2, where we observe that the properties reflected in MR_1 , MR_3 , MR_4 and MR_5 properly represent the expected behaviour of the system. In these cases, 100% of the test cases satisfy these relations. On the contrary, MR_2 generates the opposite scenario, that is, none of the test cases satisfies this relation. This can be caused by a wrong representation of how the communication links of the systems affect the overall load of the network.

Figure 2 shows a comparison between the overall energy consumed by the system under study (see Figure 2.a) and the energy consumed by the system represented by the follow-up test case with ID 1 (see Figure 2.b). In this case, it is clearly shown that the energy required to execute the simulation of the system represented by the follow-up test case is distributed among the servers and the switches, while the energy required to simulate the source test case is practically focused on the servers.

Alternatively, figures 3 and 4 depict different measures – taken from the source model and from the follow-up test case with ID 1 – focusing on the communication network. In this case, since the follow-up includes a greater quantity of switches (s'_{swi}), in contrast to the source test case, the network load average is significantly lower than the one defined in the source test case. This is especially reflected on the links that interconnects the racks

Table 1. Summary of the results obtained during the testing process.

Id	Core switches	Aggregation switches	Access switches	$\Theta(S(s))$	$\Lambda(S(s))$	$\Gamma(S(s))$
0	8.8	17.6	4.2	62.4	52.7	5.5
1	89.0	177.7	230.6	529.0	52.7	0.7
15	8.8	17.6	58.1	116.3	52.7	1.7
19	4.3	8.6	58.1	102.8	52.7	4.1
29	8.8	17.6	4.2	62.4	52.7	5.5
35	8.8	17.6	4.2	68.8	52.7	48.7
53	8.8	17.6	4.2	68.8	52.7	90.1
58	8.8	17.6	4.2	62.4	52.7	9.5
74	8.8	17.6	4.2	88.2	94.6	9.1
77	8.8	17.6	4.2	87.2	93.7	8.3

Table 2. Adequacy (in %) of each MR using GreenCloud simulator.

Id	MR_1	MR_2	MR_3	MR_4	MR_5
<i>GreenCloud</i>	100	0	100	100	100

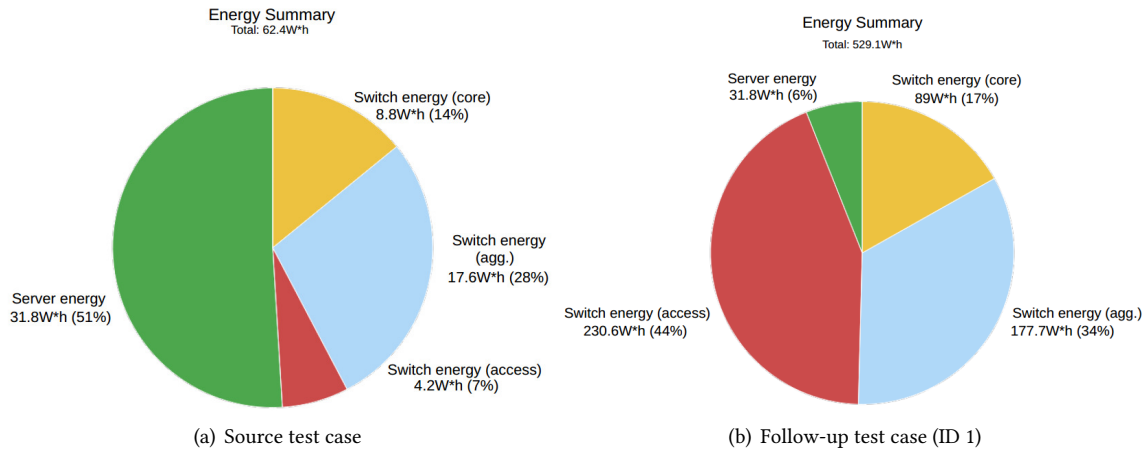


Fig. 2. Energy consumption comparison.

in the *Aggregation level* (See figure 3.a and 4.a) and in the links that interconnects the *Aggregation level* with the *Core level* (See figure 3.b and 4.b).

After a careful analysis of the results obtained from the testing process, we can conclude that Metamorphic Testing is suitable for analysing the behaviour of communication networks in complex systems, like clouds and HPC environments. We notice that increasing the total number of switches, the overall energy consumption is also increased. We expected that increasing the latencies of the communication links produce a drop in the overall performance, it requiring more energy consumption. However, the results show otherwise, that is, using different parameters to represent the latency in the network links produces similar results. Further experimental analysis is required to determine if this behaviour may be caused by a bug in the simulator. Regarding the load processed

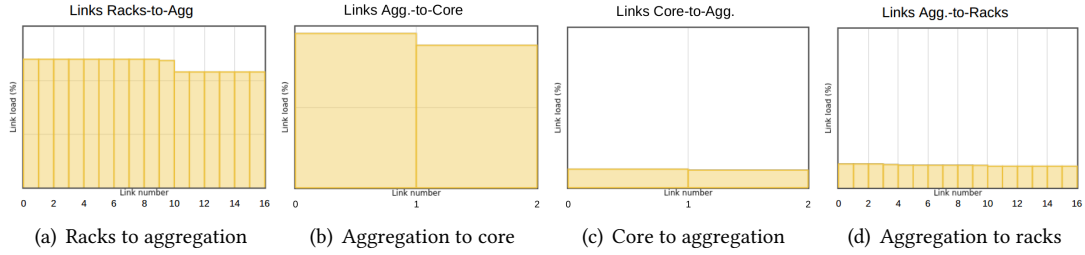


Fig. 3. Network load of the source test case.

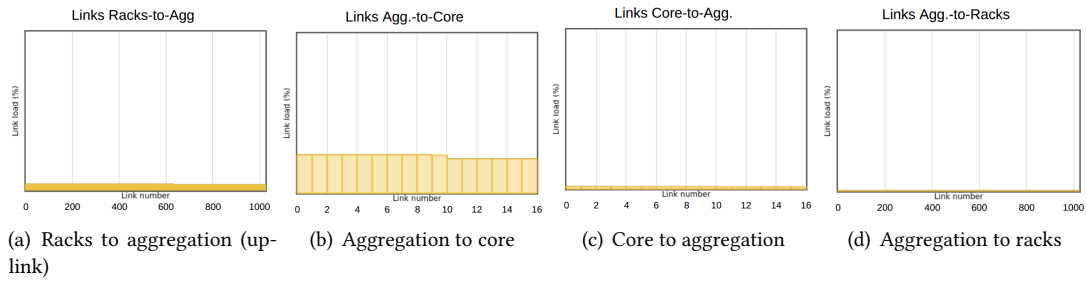


Fig. 4. Network load of the follow-up test case with ID 1.

by the cloud, the results clearly show the expected tendency, that is, when the system processes a greater load, it is reflected in greater energy consumption. Similarly, increasing the network bandwidth is reflected – as expected – in a greater system performance, which reduces the overall energy required by the system.

6 CONCLUSIONS AND FUTURE WORK

In this paper we have presented an approach for validating communication network configurations in cloud and HPC systems, using simulation and MeT techniques. The main goal of this proposal is two-fold. Firstly, providing a mechanism that acts as oracle to find inconsistencies in communication networks; and secondly, designing a method for *semi*-automatically construct valid test cases for conducting the verification of the results. In this case, we say this method is *semi*-automatic in the sense that the MRs must be designed by a human. However, the rest of the testing process, which involves generating follow-up test cases, executing the simulations and check if the test cases satisfy, or not, the relations, are performed without the intervention of a human user. To that end, the underlying architecture of one of the most popular topologies for building super-computers has been used to configure the system under study. Thus, the specific behaviour of this system has been captured in form of MRs.

In order to measure the effectiveness of our proposal, we have conducted an extended experimental study where a HPC data-center – built on the basis of the *TOR* topology – has been modelled and tested by applying up to 100 test cases and 5 MRs. The results show that MeT, in combination with simulation, is suitable for finding unexpected behaviours in the communication networks of HPC systems.

Regarding to future work, we plan to extend the catalogue of MRs, analysing more topologies and considering different metrics (such as cost, node coverage, and average distance) that aids to determine the expected results of communication networks from different perspectives. In addition, for the sake of knowledge transfer and

repeatability of experiments, we plan to integrate both the proposed MRs and the simulation environment in the generic framework of metamorphic testing Gotten [6].

7 ACKNOWLEDGMENTS

This work was supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with the Complutense University as part of the Program to Stimulate Research for Young Doctors in the context of the V PRICIT (Regional Programme of Research and Technological Innovation) under grant PR65/19-22452, the Spanish MINECO/FEDER project MASSIVE under Grant RTI2018-095255-B-I00, the *Comunidad de Madrid* project FORTE-CM under grant S2018/TCS-4314, and project S2018/TCS-4339 (BLOQUES-CM) co-funded by EIE Funds of the European Union and Comunidad de Madrid.

REFERENCES

- [1] K. Bäckström et al. 2022. Industrial Surveys on Software Testing Practices: A Literature Review. (2022).
- [2] A. Bernal, M.E. Cambronero, A. Núñez, P.C. Cañizares, and V. Valero. 2021. Evaluating Cloud Interactions with Costs and SLAs. *The Journal of Supercomputing* (2021), 1–27.
- [3] R. Buyya, R. Ranjan, and R.N. Calheiros. 2009. Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities. In *International Conference on High Performance Computing & Simulation (ICHPCS'09)*. IEEE, 1–11.
- [4] P.C. Cañizares, A. Núñez, and J. de Lara. 2019. An Expert System for Checking the Correctness of Memory Systems Using Simulation and Metamorphic Testing. *Expert Systems with Applications* 132 (2019), 44–62.
- [5] P.C. Cañizares, A. Núñez, and M.G. Merayo. 2018. Mutomvno: Mutation Testing Framework for Simulated Cloud and HPC Environments. *Journal of Systems and Software* 143 (2018), 187–207.
- [6] P.C. Cañizares, P. Gómez-Abajo, A. Núñez, E. Guerra, and J. de Lara. 2021. New Ideas: Automated Engineering of Metamorphic Testing Environments for Domain-Specific Languages. In *Proceedings of the 14th ACM SIGPLAN International Conference on Software Language Engineering*. 49–54.
- [7] P. C. Cañizares, A. Núñez, J. de Lara, and L. Llana. 2020. MT-EA4Cloud: A methodology for testing and optimising energy-aware cloud systems. *J. Syst. Softw.* 163 (2020), 110522.
- [8] D. Cao, H. Guo, and C. Tao. 2021. MT4ImgRec: A Metamorphic Testing Tool for Image Recognition Software. (2021).
- [9] I.D. Chakeres and E.M. Belding-Royer. 2004. AODV Routing Protocol Implementation Design. In *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings*. IEEE, 698–703.
- [10] T.Y. Chen, S.C. Cheung, and S.M. Yiu. 1998. *Metamorphic Testing: A New Approach for Generating next Test Cases*. Technical Report. Technical Report HKUST-CS98-01, Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong.
- [11] T.Y. Chen, F.-C. Kuo, H. Liu, and S. Wang. 2009. Conformance Testing of Network Simulators Based on Metamorphic Testing Technique. In *Formal Techniques for Distributed Systems*. Springer, 243–248.
- [12] A. Corallo, A.M. Crespino, M. Lazoi, and M. Lezzi. 2022. Model-Based Big Data Analytics-as-a-Service Framework in Smart Manufacturing: A Case Study. *Robotics and Computer-Integrated Manufacturing* 76 (2022), 102331.
- [13] Computers Science Degree Hub. 2022 [Online]. How Does Computer Science Help the World? <https://www.computersciencedegreehub.com/faq/how-does-computer-science-help-the-world/>
- [14] T. Issariyakul and E. Hossain. 2009. Introduction to Network Simulator 2 (NS2). In *Introduction to Network Simulator NS2*. Springer, 1–18.
- [15] Y. Jararweh, M. Jarrah, Z. Alshara, M.N. Alsaleh, and M. Al-Ayyoub. 2014. CloudExp: A Comprehensive Cloud Computing Experimental Framework. *Simulation Modelling Practice and Theory* 49 (2014), 180–192.
- [16] Y. Kim, W. Yang, and O. Mutlu. 2016. Ramulator: A Fast and Extensible DRAM Simulator. *IEEE Computer Architecture Letters* 15, 1 (2016), 45–49.
- [17] Kliazovich, Dmityry, Bouvry, P, and Khan, Samee Ullah. 2012. GreenCloud: A Packet-Level Simulator of Energy-Aware Cloud Computing Data Centers. *The Journal of Supercomputing* 62, 3 (2012), 1263–1283.
- [18] H. Laaki, Y. Miche, and K. Tammi. 2019. Prototyping a Digital Twin for Real Time Remote Control over Mobile Networks: Application of Remote Surgery. *IEEE access : practical innovations, open solutions* 7 (2019), 20325–20336.
- [19] C.E. Leiserson. 1985. Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing. *IEEE transactions on Computers* 100, 10 (1985), 892–901.
- [20] M. Mahbub. 2022. An Overview of IoT Infrastructure Architecture, Enabling Technologies, Issues, Integration of Cloud, and Simulation Tools. *Emerging Trends in IoT and Integration with Data Science, Cloud Computing, and Big Data Analytics* (2022), 20–38.

- [21] P.X. Mai, F. Pastore, A. Goknil, and L. Briand. 2020. Metamorphic Security Testing for Web Systems. In *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*. IEEE, 186–197.
- [22] A. Núñez, P.C. Cañizares, and J. de Lara. 2022. CloudExpert: An Intelligent System for Selecting Cloud System Simulators. *Expert Systems With Applications* 187 (2022), 115955.
- [23] A. Núñez, P.C. Cañizares, M. Núñez, and R.M. Hierons. 2021. TEA-Cloud: A Formal Framework for Testing Cloud Computing Systems. *IEEE Transactions on Reliability* 70, 1 (2021), 261–284.
- [24] A. Núñez, J. Fernández, R. Filgueira, F. García, and J. Carretero. 2012. SIMCAN: A Flexible, Scalable and Expandable Simulation Platform for Modelling and Simulating Distributed Architectures and Applications. *Simulation Modelling Practice and Theory* 20, 1 (2012), 12–32.
- [25] A. Núñez and R.M. Hierons. 2015. A Methodology for Validating Cloud Models Using Metamorphic Testing. *Annales of Telecommunications* 70, 3-4 (2015), 127–135.
- [26] I. Patronas, A. Kyriakos, and D. Reisis. 2016. Switching Functions of a Data Center Top-of-Rack (ToR). In *2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, 364–367.
- [27] M.S. Raunak and M.M. Olsen. 2021. Metamorphic Testing on the Continuum of Verification and Validation of Simulation Models. In *2021 IEEE/ACM 6th International Workshop on Metamorphic Testing (MET'21)*. IEEE, 47–52.
- [28] F. Riese, V. Herdt, D. Große, and R. Drechsler. 2021. Metamorphic Testing for Processor Verification: A RISC-V Case Study at the Instruction Level. In *2021 IFIP/IEEE 29th International Conference on Very Large Scale Integration (VLSI-SoC)*. IEEE, 1–6.
- [29] R. Ubal, B. Jang, P. Mistry, D. Schaa, and D. Kaeli. 2012. Multi2Sim: A Simulation Framework for CPU-GPU Computing. In *2012 21st International Conference on Parallel Architectures and Compilation Techniques (PACT)*. IEEE, 335–344.
- [30] S. Vinoth, H.L. Vemula, B. Haralayya, P. Mamgain, M.F. Hasan, and M. Naved. 2022. Application of Cloud Computing in Banking and E-Commerce and Related Security Threats. *Materials Today: Proceedings* 51 (2022), 2172–2175.
- [31] T. Wegner, M. Lassnig, P. Ueberholz, and C. Zeitnitz. 2022. Simulation and Evaluation of Cloud Storage Caching for Data Intensive Science. *Computing and Software for Big Science* 6, 1 (2022), 1–17.
- [32] E.J. Weyuker. 1982. On Testing Non-Testable Programs. *Comput. J.* 25, 4 (1982), 465–470.
- [33] M. Xiao, G. Liu, J. Xie, Z. Dai, Z. Wei, Z. Ren, J. Yu, and L. Zhang. 2021. 2019nCoVAS: Developing the Web Service for Epidemic Transmission Prediction, Genome Analysis, and Psychological Stress Assessment for 2019-nCoV. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 18, 4 (2021), 1250–1261.
- [34] Z. Zhang, P. Wang, H. Guo, Z. Wang, Y. Zhou, and Z. Huang. 2021. DeepBackground: Metamorphic Testing for Deep-Learning-driven Image Recognition Systems Accompanied by Background-Relevance. *Information and Software Technology* 140 (2021), 106701.
- [35] Z.Q. Zhou, D.H. Huang, T.H. Tse, Z. Yang, H. Huang, and T.Y. Chen. 2004. Metamorphic Testing and Its Applications. In *8th International Symposium on Future Software Technology (ISFST'04)*. 346–351.