Cost- and performance-based evaluation of cloud-based disaster recovery

Enrico Barbierato
Dip. di Scienze Matematiche, Fisiche e Naturali
Università Cattolica del Sacro Cuore
Via della Garzetta 48
25133 Brescia, Italy

Mauro Iacono
Dip. di Matematica e Fisica
Università degli Studi della Campania "L. Vanvitelli"
viale Lincoln 5
81100, Caserta, Italy

Marco Gribaudo
Dip. di Elettronica, Informatica e Bioingegneria
Politecnico di Milano
via Ponzio 51
20133, Milano, Italy

Michele Mastroianni
Dip. di Informatica
Università degli Studi di Salerno
Via Giovanni Paolo II, 132
84084 Fisciano, Italy

KEYWORDS

Performance evaluation; multiformalism modeling; cloud-based disaster recovery; simulation; cost evaluation

ABSTRACT

Cloud platforms offer not only the capacity to facilitate effective and scalable services for third-party applications and business solutions, but also present an opportunity to implement intricate disaster recovery strategies. For instance, a Chief Technical Officer may opt to maintain operations on private systems in order to effectively manage costs, privacy, and security, while leveraging at the same time the cloud as an autonomous and immediate disaster recovery support. This objective can be achieved by building a second leg of the IT system that functions as an online cold or hot spare, manages workload peaks, or handles a portion of the workload under normal conditions. To assess the cost-effectiveness of such solutions, appropriate models are essential to examine the trade-offs and explore the parameter space of possible alternatives.

The contribution of this paper is twofold. Firstly, it defines a multiformalism model for the design and evaluation of cloud-based recovery setups; secondly, it studies the time and the effects of transient management on costs, including losses due to a decreased capacity to serve requests.

INTRODUCTION

The utilization of information systems composes the cornerstone of operations for a significant proportion of enterprises, ranging from those of a modest scale to those of a much larger scale. Ensuring the continuity of business operations represents a primary concern for system administrators and Chief Technical Officers, given that any service outage may result in losses and additional costs arising from associated damages. Consequently, disaster recovery strategies need to be devised and implemented, encompassing the requisite hardware, appropriate software, and skilled personnel. Notably, the cloud has emerged as a promising alternative, offering disaster recovery services that are provided by third-party entities in an "as-a-service" format.

Various commercial offers are available that encompass different tiers of service levels, ranging from all-inclusive solutions in which the responsibility for planning, managing, implementing, maintaining, and administering disaster recovery is assumed entirely by the cloud vendor, to solutions in which complete control and decision-making power are retained by the enterprise, thereby providing maximum flexibility. Between these two extremes, a diverse range of degrees of the delegation is accessible. The selection of an appropriate solution is contingent on various factors, including the specific requirements of the business, internal expertise, the desired level of investment in IT technology, performance requirements, management approach, and scale.

Cloud-based disaster recovery involves the replication or partial execution of certain components of the in-house IT architecture in the cloud. At a minimum, this entails the replication of data, while at the other end of the spectrum, a comprehensive replica that can be invoked in the event of a main system failure represents a maximal choice. Intermediate options include the use of active Virtual Machines (VMs) in the cloud to manage workload peaks that surpass the capacity of the system, effectively serving as a permanent extension of the original system.

Irrespective of the selected strategy, cloud-hosted resources must be synchronized with the original system, with VMware solutions, for example, ensuring that the state of the cloud-based system is harmonized with the original system using a "Recovery Point Objective" (RPO) that, at worst, corresponds to the state of the original system up to 30 minutes before. As a consequence, this necessitates a continuous connection and the presence of active or inactive cloud resources, as well as hot or cold storage that are triggered in different ways depending on the chosen strategy, with varying associated costs.

Cloud-based disaster recovery can be implemented for either internal systems or those that provide services to customers and generate value due to external access.

Communications of the ECMS, Volume 37, Issue 1,
Proceedings, ©ECMS Enrico Vicario, Romeo Bandinelli,
Virginia Fani, Michele Mastroianni (Editors) 2023
ISBN: 978-3-937436-80-7/978-3-937436-79-1 (CD) ISSN 2522-2414

For the sake of clarity, this paper focuses on the latter case.

The utilization of multiformalism modeling methodologies relies on the amalgamation of constituents that are defined via multiple modeling languages or formalisms. This approach has two distinctive features. Firstly, it enables the modeler to utilize different formalisms to model various subsystems, which facilitates the selection of a more suitable or recognizable language, thereby reducing the learning curve or conforming to the user's preferred abstraction. Secondly, from the perspective of the solution, selecting an appropriate combination of formalisms entails the accurate alignment of model concepts with solver primitives, leading to an improved fit for the problem.

This study proposes a multiformalism modeling approach that facilitates the assessment of trade-offs between organizational choices and associated costs, including potential losses, to facilitate the design of cloud-based disaster recovery solutions. The paramount objective is to provide system administrators and Chief Technical Officers with a quantitative tool that empowers them to make informed decisions and formulate strategic plans as business requirements and workload dynamics evolve over time.

The paper is organized as follows: following the introduction, the scientific literature is reviewed and a reference ICT architecture is presented. Then, the authors discuss a model of the considered scenario and a simple case study. Finally, the conclusions close the paper.

RELATED WORK

Evaluating cloud disaster recovery can be rather complex, as it depends on different factors. For instance, the cloud model can be public or private, reflecting considerable differences in costs ([Chang et al., 2019], [Dreher et al., 2017]), or even hybrid (see [Malawski et al., 2013] for a full evaluation). With regard to disaster modeling, [Miles et al., 2019] present an overview of the scientific literature by examining the following perspectives: i) resource-constrained modeling, ii) machine learning, iii) dynamic economic impact modeling, iv) system dynamics simulation, v) agent-based simulation, vi) discrete-event simulation, vii) network modeling, and viii) stochastic simulation. The latter is described in terms of models performing sampling-based techniques and the exploitation of random variables, varying in space and in time, and changing their values according to probability.

A study of stochastic modeling of cloud disaster recovery is provided by [Andrade et al., 2017]. The authors define the concept of cloud disaster recovery and the need for stochastic modeling to estimate the likelihood and impact of potential disasters. Different approaches to stochastic modeling, including Markov processes, queuing theory, and simulation are taken into account and compared.

[Lenk and Tai, 2014] review a Markov Chain model for cloud disaster recovery taking into account the cost and time associated with different recovery options. The authors provide a model that can be used to make informed decisions about the appropriate disaster recovery strategy for a given cloud-based system. In particular, a discrete-time Markov Chain to model the state transitions of the cloud system is adopted. Each state denotes a particular system configuration, such as the system running normally, experiencing a partial or complete failure, or undergoing a recovery process. Furthermore, a method for optimizing the disaster recovery strategy using the Markov Chain model is proposed. Finally, the authors deploy a dynamic programming technique to determine the optimal recovery strategy based on the current system state and the expected cost and time of each recovery option.

[Silva et al., 2014] propose a Stochastic Petri Networks (SPNs) as a simulation model to evaluate the survivability of cloud computing systems in the presence of disasters. The proposed framework considers different factors, such as system availability, data loss, and recovery time, to evaluate the survivability of cloud computing systems. The use of SPNs allows for the modeling of complex systems, including cloud computing systems, by representing system states and transitions in a graphical manner. SPNs and fault-injection experiments [Mendonça et al., 2018] can be used to evaluate availability related metrics such as steady-state availability and downtime. Furthermore, [Andrade and Nogueira, 2019] use a similar approach for evaluating cloud-based data recovery solutions for IT environments. In [Nguyen et al., 2018], the authors avoid the space state explosion by using hierarchical modeling techniques based on stochastic reward net-based models.

Scientific literature offers numerous contributions concerning dedicated stochastic modeling tools based on the notion of multiformalism. For example, SHARPE (Stochastic Hierarchical Analysis for Reliability Performance Evaluation) [Trivedi, 2002] is a tool for reliability analysis and performance evaluation of computer systems. The SHARPE tool is designed to model and analyze complex systems that have a hierarchical structure and stochastic behavior. SMART (Stochastic Model-Checking Analysis and Random Testing) [Ciardo, 2006] is a tool for the analysis of stochastic models. The SMART tool is designed to provide both model-checking and random testing capabilities for the analysis of complex stochastic systems. Möbius [Deavours et al., 2002] is a modeling language to allow users to specify the behavior of a system at a high level of abstraction, while still providing the necessary detail for accurate performance analysis. A graphical interface allows users to visualize and edit models, as well as perform the simulation and analysis tools, which can be used to evaluate the performance of a system under a variety of different scenarios. OsMoSys [Vittorini et al., 2004] is an integrated tool environment for stochastic modeling and simulation. The authors describe the various components of OsMoSys, including its modeling language, simulation engine, and visualization tools. They also discuss the features and capabilities of the tool, including its ability to support both discrete-event and continuous-time simulations, as well as its support for hierarchical modeling.

THE ICT INFRASTRUCTURE

The reference architecture which is taken into account for the evaluation is depicted in Fig. 1[1], and may be defined as one of the best practice in the field of Disaster Recovery as a Service (DRaaS). In particular, this work considers a service implemented using VMware Cloud Disaster Recovery, which is an on-demand disaster recovery and ransomware recovery service providing an easy-to-use and a cost-saving Software-as-a-Service Solution. The service is organized as follows: the VMs of the system to be protected are replicated in remote *instant power-on* VMs and by implementing them to a target VMware Cloud on AWS (Amazon Web Services) Software Defined Data Center (SDDC) on VMware Cloud on AWS. The system can be used as a fast-recovery replica minimizing the service downtime or, as another option, in case of a successful ransomware attack, as an isolated recovery environment (IRE) in which one can inspect, analyze, and repair snapshots of infected VMs in which it is possible to restore the service to a production environment by selecting the most recent non-corrupted replica.

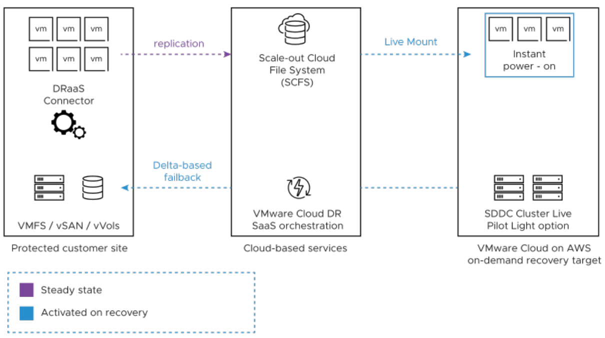

Fig. 1: The reference architecture

The network connection between the system and the AWS cloud service may be achieved using three different technologies:

- usual **Internet connectivity**, with very low costs but suffering from latency and security issues;
- **IPSec VPN connection**, with low bandwidth but a sufficient level of security, although not recommended for production environment;
- **AWS Direct Connect**, which is a dedicated AWS connection service that may guarantee high bandwidth (up to 100 Gbit/s) and very low latency at high price.

The operating cost for the system has been computed using AWS Pricing Calculator[2].

[1]Adapted from https://docs.aws.amazon.com/prescriptive-guidance/latest/disaster-recovery-vmware-cloud-on-aws/dr-options.html
[2]`https://calculator.aws`

MODELING APPROACH

The model of the ICT infrastructure has been implemented by using SIMTHESYS (Structured Infrastructure for Multiformalism modeling and Testing of Heterogeneous formalisms and Extensions for SYStems, see [Barbierato et al., 2012] for an introduction and a case study). It is a framework that offers a method for the formulation and resolution of multiformalism models through the production of intricate solvers, which are automatically generated by integrating general solution engines based on the rules that arise from formalism definitions. The formalisms themselves are defined by explicitly specifying both the syntax and semantics of all their atomic components. This approach offers several significant benefits, such as facilitating the rapid prototyping of new formalisms and solution techniques, enabling the deployment of new solvers without requiring the modification of existing ones, and providing an open architecture that allows for the creation of new interfaces that can be utilized to characterize different classes of formalisms.

The reference configuration of interest is based on N VMs that serve the operations needs. In normal conditions, VMs may be totally or partially run locally, while, in case of disasters, all VMs should run in the cloud after a transient. This work studies the effects of this transient, with particular reference to transient time and the effects of transient management on costs, including losses due to lower capacity to serve requests during the transient.

The configuration is parameterized on the number of VMs out of the total N that are run locally with respect to the normal state of the system. In the normal state, some VMs may be allocated in the cloud and ready to serve requests to implement a hybrid cloud solution that manages workload peaks.

One of the N VMs, namely a *Front End*, is used to accept requests and balance the workload; another is used to run a *Database Server* that implements data management for the application. With no loss of generality, requests are considered as generated by external traffic, and the routing of requests is managed by a DNS-as-a-service facility supplied by the cloud provider, redirecting traffic to the Front end replacement VM executed in the cloud in case of disaster.

To manage the requests during the disaster recovery operations and to perform recovery, data must be replicated in the cloud, with periodic transfers occurring compatibly with application needs and according to cost constraints. The replication policy and frequency are also affected by the use of cloud VMs during normal operations, which may also induce a request for bidirectional synchronizations when necessary. Consequently, cloud storage may be solicited differently according to the workload dynamics during normal operations as well.

Requests that are received by the Front End are assigned to one of the available VMs, prioritizing local ones in case of hybrid cloud configurations. VMs serve the request, accessing the Database server and possibly modifying data. Updates are periodically sent to the cloud storage, asynchronously with respect to the effects of VM accesses to the Database Server.

When cloud resources are invoked, used resources are billed accordingly to their usage. Cloud counterparts of local resources may be configured in different readiness states, implying different activation times and costs. Part of cloud resources (at least, cloud storage) is always active.

Modeling is done considering different scenarios and different request rates and types.

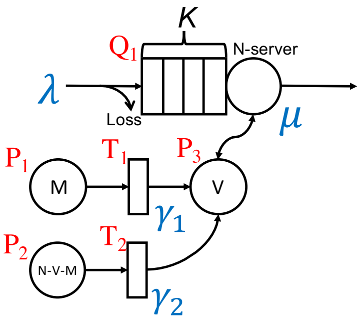

Fig. 2: The multiformalism model of the proposed scenario

The considered system is modeled with the Petri Net (PN) / Queuing Network (QN) multiformalism architecture presented in Figure 2. In particular, the service is modeled by a N server finite capacity queue, serving requests at an exponential rate μ. The system has a total capacity of K jobs, including the ones in service: the requests, arriving according to the Poisson process of rate λ, are lost if the system is full when they attempt to enter the server. The latter is controlled by a Generalized Stochastic Petri Net (GSPN), according to the test arc connecting queue Q_1 with place P_3. Following the semantics given in [Gribaudo and Iacono, 2023], the test arc controls the number of parallel servers of station Q_1. In particular, of the maximum of the N servers that compose the queue, only as many as the marking of P_3 are actually active at any time. Place P_1 models the M hot spare VMs, each one becoming available at rate γ_1, according to the firing of the infinite server timed transition T_1. Similarly, Place P_2 with initial marking $N - V - M$ and Transition T_2 of rate γ_2 model the activity of additional VMs not being supported as hot-spare. Place P_3 models the current number of active VMs: its initial marking V can be used to support hybrid cloud scenario, where part of the computation is initially deployed in the cloud .

The dynamics of the queue Q_1 is at least two order of magnitude faster than the one of the transitions: the transient time required to reach the steady state is negligible with respect to the time the system remains with the same number of servers. For this reason, it is possible to decouple the solution of the queuing network, from the one of the Petri Net, and use the steady state solution of the first as a reward for the second.

Specifically, following the classical theory of M/M/c/K queues, the loss rate $L_r(c)$ can be evaluated when only c out of N servers are active. Let be $\rho = \frac{\lambda}{c\mu}$, then:

$$L_r(c) = \frac{\frac{c^c(c\rho)^N}{c!}}{\frac{(c\rho)^c}{c!}\frac{1-\rho^{N-c+1}}{1-\rho} + \sum_{k=0}^{c-1}\frac{(c\rho)^k}{k!}} \tag{1}$$

The Petri Net component is mapped to a Continuous Time Markov Chain (CTMC), with usual state-space generation techniques. The corresponding CTMC has $(M + 1) \times (N - V - M + 1)$ states, which leads to an easily manageable model for a very large parameter space. By identifying each state $s_i = (n_1, n_2)$ with a tuple where n_1 and n_2 account respectively for the markings of places P_1 and P_2, the following holds:

$$q_{(n_1,n_2),(n_1-1,n_2)} = n_1 \cdot \gamma_1 \quad \text{with } n_1 > 0 \tag{2}$$
$$q_{(n_1,n_2),(n_1,n_2-1)} = n_2 \cdot \gamma_2 \quad \text{with } n_2 > 0 \tag{3}$$

The model has a single absorbing state $s_{abs} = (0, 0)$, and its initial state is $s_0 = (M, N - V - M)$. Let $\mathbf{Q} = |q_{(n_1,n_2)}|$ be the infinitesimal generator of the CTMC, and $\mathbf{p}_0$ the initial state of the system, a zero-vector, with exception of the component corresponding to state s_0 that is set to one. The transient evolution $\mathbf{p}(t)$ of the system at time t can be computed as follows:

$$\mathbf{p}(t) = \mathbf{p}_0 \cdot e^{\mathbf{Q}\cdot t} \tag{4}$$

Let us call $\mathbf{r}$ a column vector, where component r_i corresponding to state $s_i = (n_1, n_2)$ accounts for the loss rate of that configuration:

$$r_i = \begin{cases} L_R(N - n_1 - n_2) & n_1 + n_2 > 0 \\ 0 & n_1 + n_2 = 0 \end{cases} \tag{5}$$

As there are not losses that might occur when the system is at full capacity (they may occur even when no recovery is in progress), the rate corresponding to the absorbing state is set to $r_{abs} = 0$. The instantaneous loss rate $\Phi(t)$ and the total accumulated losses $\Psi(t)$ at time t can then be defined as:

$$\Phi(t) = \mathbf{p}_0 \cdot e^{\mathbf{Q}\cdot t} \cdot \mathbf{r} \qquad \Psi(t) = \int_0^t \mathbf{p}_0 \cdot e^{\mathbf{Q}\cdot \tau} \cdot \mathbf{r} d\tau \tag{6}$$

Since the CTMC has a single absorbing state, the average total loss of the system $\bar{\Psi}$ can be easily computed until its full service capacity of N virtual machines is restored. Without loss of generality, let s_{abs} be the last state, and use $\hat{\mathbf{Q}}$ and $\hat{\mathbf{r}}$ to denote the sub-matrix and sub-vector that exclude the absorbing state:

$$\mathbf{Q} = \left\| \begin{array}{cc} \hat{\mathbf{Q}} & -\hat{\mathbf{Q}}\cdot\mathbf{1} \\ \mathbf{0} & 0 \end{array} \right| \qquad \mathbf{r} = \left\| \begin{array}{c} \hat{\mathbf{r}} \\ 0 \end{array} \right| \tag{7}$$

where $\mathbf{0}$ and $\mathbf{1}$ are respectively a zero-row and a one-column vector. Due to the matrix exponential definition of $\mathbf{p}(t)$, the following holds:

$$\bar{\Psi} = \lim_{t\to\infty} \Psi(t) = \mathbf{p}_0 \cdot \hat{\mathbf{Q}}^{-1} \cdot \hat{\mathbf{r}} \tag{8}$$

CASE STUDY EVALUATION

An e-commerce-oriented firm provides its services by means of an in-house solution. In case of problems, the cloud replaces the in-house solution simultaneously depending on the chosen recovery scenario. Some of the requests cannot be served until the backup cloud configuration is fully operational in equivalent conditions with respect to the in-house solution. This creates a loss depending on the value of each request and the number of loss requests: as a result, the focus of this scenario revolves around the study of the transient when the in-house system fails. Considering a constant rate of requests, performance measures are a proxy with respect to the overall loss, which can be compared against the costs of the chosen cloud-based recovery solution. For example, with regards to the cost of cloud services, considering AWS, a popular provider in the e-commerce world, costs related to the (virtual) servers and the other -aaS services requested for the operation of the online store are listed in Table I.

TABLE I: Cloud Service costs (AWS price list).

Item	**Description**	**Cost**
Front-end and hot-spare servers	Server always-on VM 4 CPU, 16GB RAM, 1TB disk	0.931 $/h
On-demand Server	Server on-demand, same characteristics as hot-spare, plus 300 MB/month provisioned network traffic	2.973 $/h
DB-as-a-Service	48 vCPU, 384GB Mem, 10 GB/month, 10M i/o operations	7.708 $/h
DRaaS	AWS Elastic Disaster Recovery service	0.357 $/h

The losses caused by the occurrence of a service interruption have been estimated with the following assumptions: the annual revenue is hypothesized to be 10M dollars, sufficiently high to consider the company able to easily manage complex ICT systems and services, and the mean value of orders is $40.

The E-commerce Conversion Rate (ECR) is used to assess the revenue loss occurred during the interruption. This rate, conceived to represent the (economical) performance of online shops, is defined as the number of visitors to an online shop who make an order versus the total number of visit of the store in a specified period [Pradana and Luxianto, 2020]. Therefore, the typical value of ECR for the chosen e-commerce business sector (home/office electronics), which is estimated at around 4% by literature [Saleh, 2022], was taken into account to estimate the revenue losses, as per Table II.

TABLE II: Revenue loss caused by a disaster.

Annual revenue	10,000,000 $
Average amount for orders	40 $
Annual orders No.	250,000
E-commerce Conversion Rate	4.00 %
Annual No. of visits	6,250,000
Hourly No. of visits (λ)	713.5
Revenue loss (hour)	1,141.55 $/h

TABLE III: Service parameters.

Avg. Service time per req. (μ^{-1})	0.00889 h
Avg. Time to setup a hot spare (T_1)	2 h
Avg. Time to setup a new VM (T_2)	48 h
Maximum Queue Length (K)	16

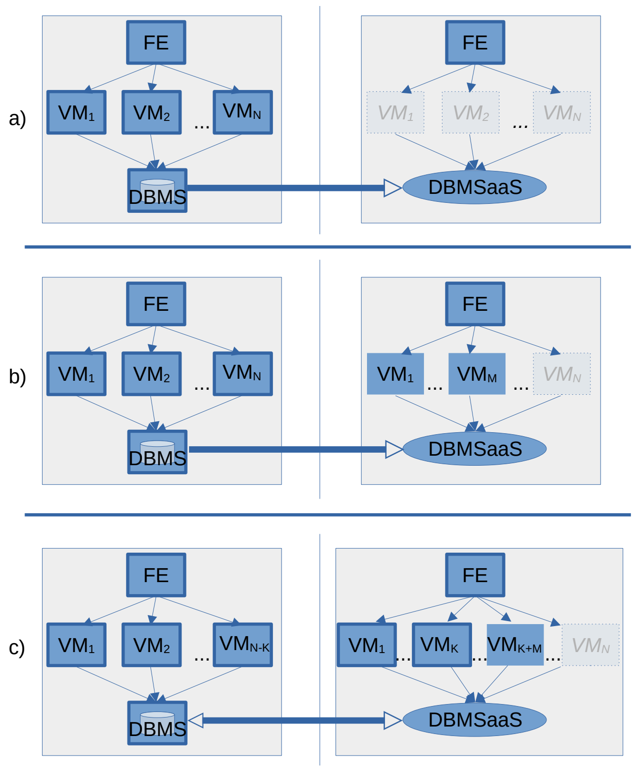

Fig. 3: Scenarios

Fig. 3 presents the three scenarios that have been evaluated. VMs are represented by rectangles and cloud services by ovals. VMs with a thick contour are active, VMs with a thin contour are in hot spare, so they can be activated in a short time, VMs with a dashed contour are in cold spare, so they need more time to be available for processing requests. The first scenario, represented in Fig. 3 a), is based on the case in which cloud costs are minimized, as for N VMs in the configuration of the in-house system N VMs are configured in the cloud, with none of them as hot spares. The second scenario, represented in Fig. 3 b), is based on the case in which for N VMs in the configuration of the in-house system, M of the N VMs are configured in the cloud as hot spares, with $M \leq N$. The third scenario, represented in Fig. 3 c), is based on the case in which the cloud is used as a hybrid resource, so that part of the normal workload of the system is managed by V cloud VMS, with $V < N$, which complete the overall N VMs configuration. These V VMs are consequently always active, while M VMs, with $M \leq N-V$, are configured as hot spares and $N-M-V$ VMs are configured as cold spares.

The three scenarios were evaluated using the parameters given in Table III, with case b) using respectively $M=2$ and $M=3$, and case c) with $V=2$ and both $M=0$ or $M=1$, and $V=2$, $M=0$. Fig. 4 shows the number of active VMs as function of time. The height of the curve is determined by the number of VMs that can become active in a limited time, which corresponds to $M+V$. Solutions that are hybrid, such as in case c), start with a higher number of VMs from time zero. The loss rate as function of time experienced in the various configurations is shown in Fig. 5: $V>0$ and $M>0$ can reduce, as expected, the loss rate in the moments immediately following the disaster. However, the cost might not be worth the gain, as shown in figure 6. In particular, the hybrid solutions seem not being worth the extra price, giving total losses very similar to hot-spare configuration with the same number of $V+M$ backup virtual machines.

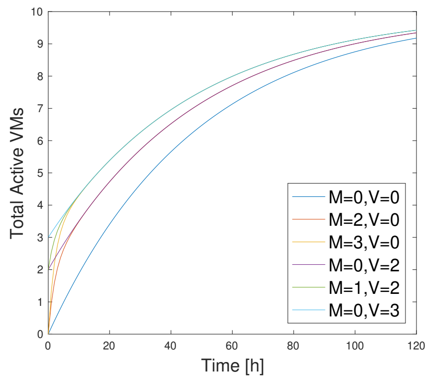

Fig. 4: VMs active as function of time for $N=10$

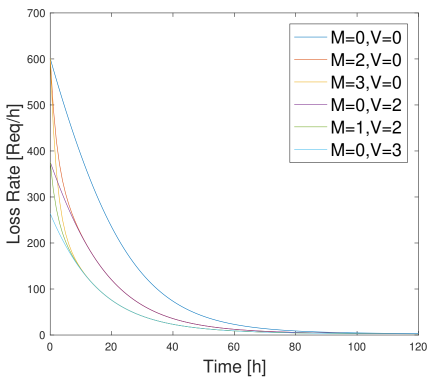

Fig. 5: Losses as function of time for $N=10$

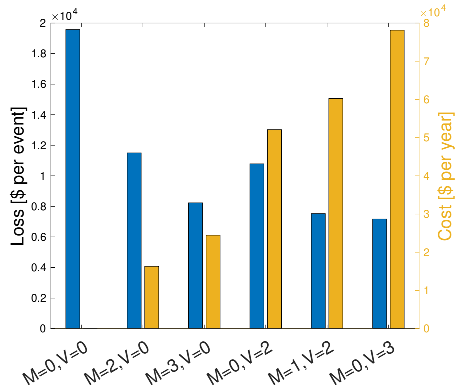

Fig. 6: Costs and losses for various configurations with $N=10$

CONCLUSIONS AND FUTURE WORK

This paper presented a parametric model to study the transient effects of the implementation of a cloud-based disaster recovery solution, based on commercial offers and on a typical application class. The discussed model will be the base for further work, more focused on cost evaluation and parameters selection, aiming at the design of a decision support methodology for Chief Information Officers and IT divisions.

ACKNOWLEDGEMENTS

This work is part of the research activity realized within the project PON "Ricerca e Innovazione" 2014-2020, action IV.6 "Contratti di ricerca su tematiche Green".

References

[Andrade and Nogueira, 2019] Andrade, E. and Nogueira, B. (2019). Performability evaluation of a cloud-based disaster recovery solution for it environments. *Journal of Grid computing*, 17:603–621.

[Andrade et al., 2017] Andrade, E., Nogueira, B., Matos, R., Callou, G., and Maciel, P. (2017). Availability modeling and analysis of a disaster-recovery-as-a-service solution. *Computing*, 99:929–954.

[Barbierato et al., 2012] Barbierato, E., Bobbio, A., Gribaudo, M., and Iacono, M. (2012). Multiformalism to support software rejuvenation modeling. In *2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops*, pages 271–276. IEEE.

[Chang et al., 2019] Chang, Y.-W., Hsu, P.-Y., Shiau, W.-L., and Hsu, M.-C. (2019). An empirical study on factors affecting switching intention to cloud enterprise resource planning: a comparison between public and private clouds. *Journal of Global Information Management (JGIM)*, 27(4):46–69.

[Ciardo, 2006] Ciardo, G. (2006). SMART: stochastic model-checking analysis and random testing toolkit. In *Proceedings of the 2006 international conference on Dependable Systems and Networks*, pages 231–240. IEEE.

[Deavours et al., 2002] Deavours, D. D., Clark, G., Courtney, T., Daly, D., Derisavi, S., Doyle, J. M., Sanders, W. H., and Webster, P. G. (2002). The Möbius Framework and Its Implementation.

[Dreher et al., 2017] Dreher, P., Nair, D., Sills, E., and Vouk, M. (2017). Cost analysis comparing HPC public versus private cloud computing. In *Cloud Computing and Services Science: 6th International Conference, CLOSER 2016, Rome, Italy, April 23-25, 2016, Revised Selected Papers 6*, pages 294–316. Springer.

[Gribaudo and Iacono, 2023] Gribaudo, M. and Iacono, M. (2023). Places, transitions and queues: New proposals for interconnection semantics. In Gilly, K. and Thomas, N., editors, *Computer Performance Engineering*, pages 216–230, Cham. Springer International Publishing.

[Lenk and Tai, 2014] Lenk, A. and Tai, S. (2014). Cloud standby: disaster recovery of distributed systems in the cloud. In *Service-Oriented and Cloud Computing: Third European Conference, ESOCC 2014, Manchester, UK, September 2-4, 2014. Proceedings 3*, pages 32–46. Springer.

[Malawski et al., 2013] Malawski, M., Figiela, K., and Nabrzyski, J. (2013). Cost minimization for computational applications on hybrid cloud infrastructures. *Future Generation Computer Systems*, 29(7):1786–1794.

[Mendonça et al., 2018] Mendonça, J., Lima, R., Matos, R., Ferreira, J., and Andrade, E. (2018). Availability analysis of a disaster recovery solution through stochastic models and fault injection experiments. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pages 135–142.

[Miles et al., 2019] Miles, S. B., Burton, H. V., and Kang, H. (2019). Community of practice for modeling disaster recovery. *Natural Hazards Review*, 20(1):04018023.

[Nguyen et al., 2018] Nguyen, T. A., Min, D., Choi, E., and Márquez, F. G. (2018). Stochastic reward net-based modeling approach for availability quantification of data center systems. In *Dependability Engineering*. InTech Rijeka, Croatia.

[Pradana and Luxianto, 2020] Pradana, R. and Luxianto, R. (2020). Analysis on user activity in e-commerce website for performance evaluation and decision making using big data analytics. In *Selected Papers from the 1st International Conference on Islam, Science and Technology, ICONISTECH-1 2019, 11-12 July 2019, Bandung, Indonesia*.

[Saleh, 2022] Saleh, K. (2022). The average website conversion rate by industry.

[Silva et al., 2014] Silva, B., Maciela, P. R. M., Zimmermannb, A., and Brilhantea, J. (2014). Survivability evaluation of disaster tolerant cloud computing systems. In *Proc. Probabilistic Safety Assessment & Management conference*, page 12.

[Trivedi, 2002] Trivedi, K. S. (2002). Sharpe 2002: Symbolic hierarchical automated reliability and performance evaluator. In *DSN '02: Proceedings of the 2002 International Conference on Dependable Systems and Networks*, page 544, Washington, DC, USA. IEEE Computer Society.

[Vittorini et al., 2004] Vittorini, V., Iacono, M., Mazzocca, N., and Franceschinis, G. (2004). The OsMoSys approach to multi-formalism modeling of systems. *Software and System Modeling*, 3(1):68–81.

AUTHOR BIOGRAPHIES

ENRICO BARBIERATO is an Assistant Professor in Data Science at Dipartimento di Matematica e Fisica, Università Cattolica del Sacro Cuore, Brescia, Italy. He worked for 25 years in IT consulting for the Banking, Telecommunications and Energy&Utilities industries. His research interests include performance evaluation through multiformalism and ethical AI. His email is enrico.barbierato@unicatt.it.

MARCO GRIBAUDO is an Associate Professor in Computer Science at Politecnico di Milano, Italy. He works in the performance evaluation group. His current research interests are multiformalism modeling, queueing networks, mean-field analysis and spatial models. The main applications to which the previous methodologies are applied comes from cloud computing, multi-core architectures and wireless sensor networks. His e-mail is marco.gribaudo@polimi.it

MAURO IACONO is an Associate Professor in Computing Systems at Dipartimento di Matematica e Fisica, Università degli Studi della Campania "L. Vanvitelli", Caserta, Italy, where he leads the Computer Science section of the Data and Computer Science research group. He received the Ph.D. in Electrical Engineering from Seconda Università degli Studi di Napoli. His research activity is mainly centred on the field of performance modeling of complex computer-based systems, with special attention for multiformalism modeling techniques. His email is mauro.iacono@unicampania.it. For more information: http://www.mauroiacono.com.

MICHELE MASTROIANNI is an Assistant Professor in Computer Science at Dipartimento di Informatica of Università degli Studi di Salerno, Italy, and is also a research associate at Dipartimento di Matematica e Fisica of Università degli Studi della Campania "L. Vanvitelli", Caserta, Italy (where he has been teaching and Data Protection Officer, Network Manager, project leader and expert for many local, regional and national technical projects), with the Data and Computer Science research group. He holds a M. Sc. degree in Electrical Engineering and a Ph.D. degree in Management Engineering. His email is mmastroianni@unisa.it.