Software aging and rejuvenation in the cloud: a literature review

Roberto Pietrantuono, Stefano Russo
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione
Università degli Studi di Napoli Federico II
Via Claudio 21, 80125, Naples, Italy
{roberto.pietrantuono, stefano.russo}@unina.it

***Abstract*—With cloud computing becoming pervasive in IT enterprises, the long-running performance of cloud-based applications and services, of cloud computing platforms and of the underlying virtualization systems is a paramount concern. Researchers have been studying the phenomenon of software aging in cloud-related contexts, and the corresponding rejuvenation countermeasures, since about 10 years. This paper reviews the effort conducted so far in the software aging and rejuvenation (SAR) area in the cloud domain. Figures about current trends and future directions can be derived from the reported results.**

***Index Terms*—Software Aging; Rejuvenation; Cloud Performance; Virtualized Environment, Virtual Machine; VMM**

I. Introduction

Cloud computing is by now established as a predominant computing paradigm. Software and services based on the cloud are used in many different domains including healthcare, public transportation, mobile, and many more. According to Forrester's predictions for 2018, the total global public cloud market will be $178 billion in 2018, up from $146 billion in 2017, and will continue to grow at a 22% compound annual growth rate. Public cloud platforms, the fastest growing segment, will generate $44 billion in 2018 and more than 50% of global enterprises will rely on at least one public cloud platform to drive digital transformation.

Resources and services on the cloud need to be provided with no interruption and keeping acceptable performance over time. Performance degradation, and more generally the phenomenon of *software aging*, in this domain can seriously impact many IT enterprises, as an even short downtime or user dissatisfaction has huge economical impact. Much research has been conducted on aging in the cloud and, more broadly, on virtualized systems meant as the key technological enabler of the cloud. Researchers have studied *software aging and rejuvenation* (SAR) in this context, by proposing modelling approaches to analyze aging in common configurations (physical hosts, Virtual Machine Monitor (VMM) and Virtual Machines (VMs) running on top of them), as well as by monitoring perforamance of cloud platfroms to infer aging behaviours and decide *when* to rejuvenate. Also, the problem of *how* to rejuvenate these systems gained considerable attention by researchers. Indeed, rejuvenation is a key high-availability technique in this context, since the advantages given by VMs as application "containers" has led to novel and lightweight rejuvenation strategies (based on restarts, migration and failover) able of guaranteeing high levels of continuity of service in presence of aging problems.

This paper surveys the work that has been done so far to target SAR in cloud-based systems. A literature review has been conducted following the method described in Section II. High-level results are reported first (Section III), followed by detailed results on aging analysis approaches (Section IV) and on rejuvenation proposals for this kind of systems (Section V). A brief discussion concludes the paper (Section VI).

II. Method of analysis

A keyword-based search was performed on the on the *SciVerse Scopus* (http://www.scopus.com) search engine, which indexes the major scientific journals and proceedings from the main computer science and engineering publishers such as IEEE, ACM, Elsevier, Wiley, and Springer. The search required the words "software" and ("aging" or "rejuvenation") to appear together with at least "cloud" or "virtual" in the metadata (title, abstract, and keywords) of publications. This provided an initial raw set of 218 papers. The result has been manually inspected to remove the many irrelevant results, e.g., papers wherein the words "aging" or "rejuvenation" refer to other domains. The post-filtering set was of 88 papers. As further exclusion criterion, we eliminated the works wherein the word "virtual" referred to the Java virtual machine, because not related to the use of virtualization as cloud-enabling technology. No filtering based on the publication year and/or on the publication venue was applied, namely, we considered all the venues/years indexed by Scopus. The final set of analyzed papers is of 72 papers[1].

Figure 1 and 2 provide aggregate figures of the work in the surveyed area, reporting, respectively, the number of papers by year and the number of papers by venue with more than 1 publication. Publications in this area start from 2007, when the theme of cloud computing and virtualization started gaining popularity, with a peak from 2011 to 2015. The average number of cloud-related SAR publications was 7.5 papers per year. Papers appeared in 50 different venues[1], denoting a large diversity of research communities potentially interested in the topic. The highest number of papers appeared in WoSAR, because of its specific focus on SAR, and the journal special

[1]The full list is available at *https://github.com/rpietrantuono/Wosar2018*

issues promoted by WoSAR – Performance Evaluation (PE) and Journal on Emerging Technologies in Computing Systems (JETC) in Figure 2 are among these. The top venues include the best dependability/reliability conferences, such as DSN and ISSRE (WoSAR proceedings are part of ISSRE). It is worth noting the interest of cloud-related conferences, with 5 papers published in total in International Conference on Cloud Computing Technology and Science (CloudCom) and International Conference on Cloud Computing (CLOUD).

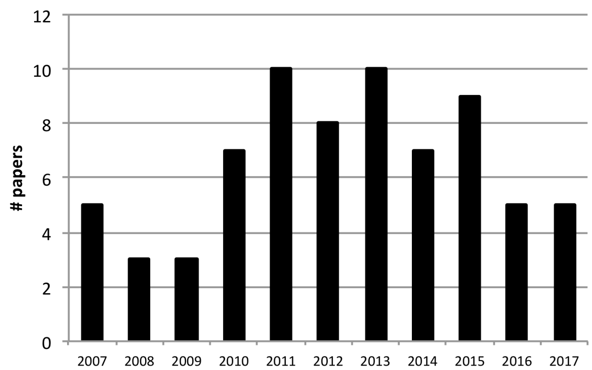

Fig. 1. Number of publications per year.

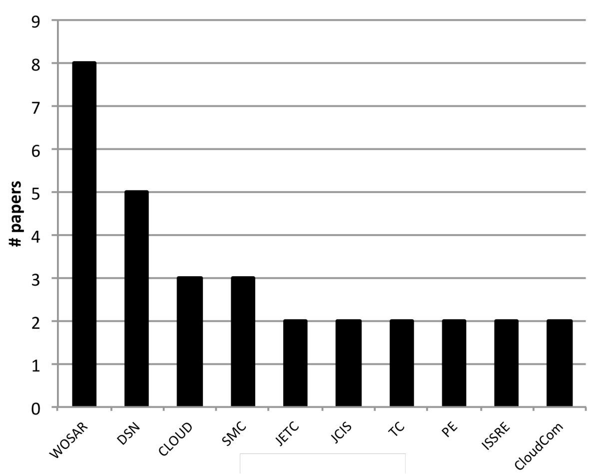

Fig. 2. Number of publications in major journals and conference proceedings.

III. High-level results

SAR research deals with *aging analysis* and with *rejuvenation*. The main goal of aging analysis – encompassing aging *detection and estimation* – is to determine the most likely time of aging failure occurrence, so as to figure out *when* to schedule a rejuvenation action. This is done either by formulating the problem via stochastic models and finding the best rejuvenation time or by monitoring system's health indicators and apply statistical inference techniques on time series to forecast future trends. These approaches, also known as **model-based** and **measurement-based**, are sometimes combined into a **hybrid** strategy, with models parametrized by field data. The goal of rejuvenation is to figure out *how* a rejuvenation action should be performed to minimize the negative impact on the user experience (e.g., downtime).

Figure 3 sketches the breakdown of the papers.[2] It highlights a balance between model-based and measurements-based studies and a big slice of works coping only with the rejuvenation problem. In fact, several researchers did not focus on aging analysis, but dealt exclusively with the rejuvenation technique. It should be noted that rejuvenation is also faced in most of the model-based, measurements-based and hybrid papers – specifically, in 23 out of 27 model-based papers, 15 out of 29 measurements-based papers and 5 out of 5 hybrid-strategy papers[3]. In the following, details about the aging assessment and then about rejuvenation are provided.

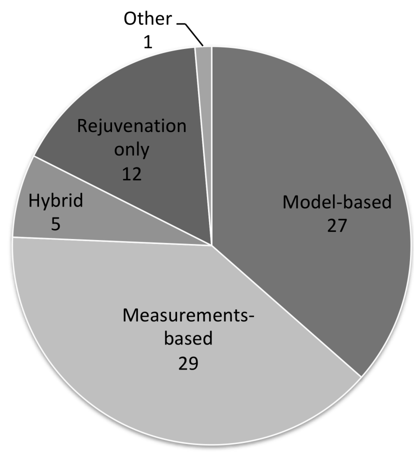

Fig. 3. Overview of analyzed papers.

IV. Aging analysis

A. Model-based aging analysis

Model-based studies are distinguished by *i)* the adopted modelling formalism and *ii)* the metric chosen to model the effect of software aging. Both aspects are explored hereafter. **Modelling formalism**: the aging process in more or less complex virtualized systems (with one or more VMs, physical hosts, and various rejuvenation strategies) is represented by several types of stochastic models, including state-space models such as stochastic Petri nets (SPN) and stochastic reward nets (SRN), continuous-time Markov chains (CTMC), semi-Markov processes (SMP), as well as combinatorial models such as reliability block diagrams (RBD) and dynamic fault trees (DFT). Figure 4 reports the number of number of times each formalism has been used for SAR in the cloud domain in model-based approaches.[4]

[2]The sum of papers is bigger than 72 because 2 papers adopt both a model-based and then a measurement-based approach

[3]The set includes 1 paper (indicated as "other"), in which there is neither a conventional aging analysis nor rejuvenation: it deals with a "static" analysis of aging-related bugs in cloud computing software

[4]The counting excludes the 5 hybrid studies, discussed later. Also, the sum is 29 instead of 27 because 2 studies adopt 2 formalisms each.

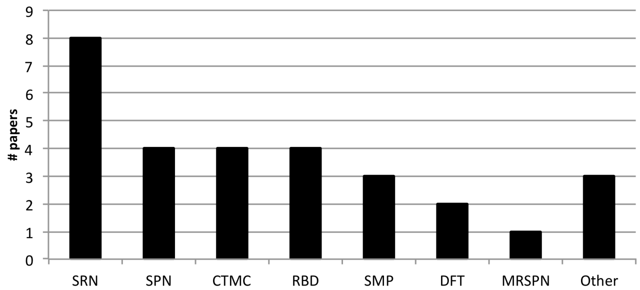

Fig. 4. Modelling formalisms in the model-based approaches

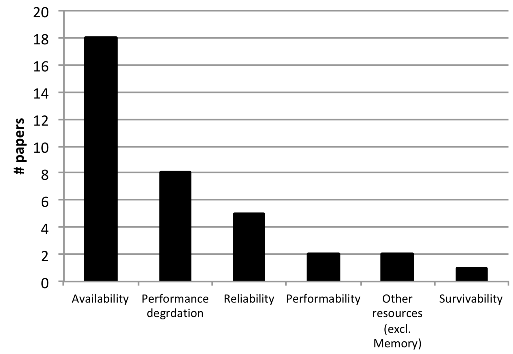

Fig. 5. Metrics used in model-based studies

Examples of works adopting the various models follow:

- *SPN-based models*. The most common formalisms are SPN-based ones. Examples are the work by Xu *et al.* [1] and Rezaei *et al.* [2], who use SRNs to model a *single-server* virtualized system with time-based rejuvenation applied at VMM level and a measurement-based rejuvenation at VM level. SRNs are also used by Nguyen [3] who considers various failure and recovery modes of multiple VMs and VMMs, and by Han and Xu [4] who consider three different rejuvenation policies (no rejuvenation, time-based rejuvenation, and time and load-based delay rejuvenation) for single-server virtualization systems with multiple VMs on a single VMM.
- CTMC are used often too. For instance, it is used in the work by Myint [5] presenting a *multiple-host multiple VMs availability analysis*, in which a primary-standby servers model is defined, encompassing a load balancer VM in each node responsible for monitoring resources and a rejuvenation agent installed on each VM.
- *Semi-Markov process* are used 4 out of 27 times. Examples include the works by Machida et *al.* [6], [7], which analyze the job completion time under aging and rejuvenation of the VMM.
- *Combinatorial models*: RBD is the most used combinatorial model, also in conjunction with state-space models, followed by DFTs. Examples are the works by Melo *et al.* [8], who formulate an availability model considering live migration for VMM rejuvenation based on extended RBD coupled with Deterministic Stochastic Petri Nets (DSPNs) and the one by Rahme *et al.* [9], who exploit DFT to model cloud-based software rejuvenation.
- *Other models* include *Markov Regenerative Stochastic Petri Nets (MRSPNs)*, adopted by Okamura et *al.* [10] to perform the transient analysis of the two main rejuvenation policies, *Cold-* and *Warm-VM* rejuvenation.

Metrics: model-based approaches do not usually refer to system-level resource depletion metrics, since the model is assumed to work regardless of which resource is being depleted. They mostly base their analysis on dependability attributes – the most common one being *availability* – and/or refer to user-perceived performance degradation metrics. Figure 5 reports the occurrences in model-based papers.

- *Availability* is by far the most common measure in model-based studies, 18 out of 27 occurrences. Availability is of particular relevance for cloud environments. In 13 cases, availability is analyzed with no reference to any specific user-perceived metric, while in the remaining 5 cases a performance degradation metric (e.g., response time, throughput) is also considered. Examples include the mentioned works by Xu *et al.* [1], Rezaei *et al.* [2], Nguyen [3]. Melo *et al.* [11] also model *cloud availability* by a SPN under two migration-based rejuvenation strategies (with and without a test before migration). Thein *et al.* analyzes *availability* with the time-based rejuvenation policy under different cluster configurations (2 VMs on a single server and 2 VMs per server in dual physical servers) and later propose a rejuvenation framework named VMSR for application servers [12].
- Other dependability attributes are: *reliability* in 5 out of 27 cases (e.g., when a RBD/DFT are used), *performability* and *survivability*. The mentioned work by Rahme *et al.* model reliability of the cloud-based software rejuvenation [9]. Escheikh analyzes power-management *performability*, evaluating the impact on performance and energy consumption in virtualized systems [13]. The work by Chang et *al.* focuses on *survivability* of virtualized systems with VM/VMM rejuvenation [14].
- Performance metrics, such as response time, job completion time or throughput, are used in 8 out 27 studies. In 1 case, the model is conceived specifically to analyze the job completion time with no explicit reference to a dependability attribute [6]; the others run the analyses looking at the impact on one among availability, reliability, survivability, performability. Examples are the work by Nguyen, where transaction loss, beside availability, is evaluated [3], and by Machida, where job time analysis is extended with steady-state availability analysis [7].
- In no case a memory-consumption metric is considered; 2 cases consider other resources in the model such as storage, CPU, network, power consumption – e.g., the works by Escheikh where performance and energy consumption are modelled together [13].

B. Measurement-based aging assessment

Measurement-based approaches use observations gathered from monitoring relevant aging indicators to infer or predict the possible aging state of a system. The cloud-related approaches are hereafter distinguished by *i)* the technique to analyze the data and *ii)* the monitored aging metric.

Technique: Techniques are grouped as: *time series analysis*, *machine learning*, *threshold-based approaches* and *other*s. Figure 6 summarizes their occurrence.

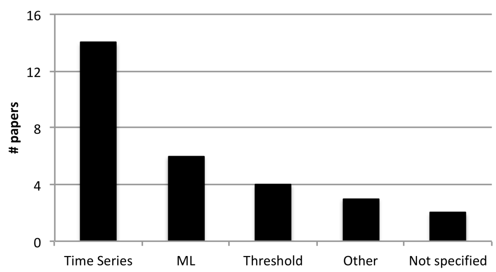

Fig. 6. Techniques used in the measurements-based approaches

- *Time Series Analysis* is based on trend detection and estimation of a set of aging indicators. Tests for trend detection aim at accepting/rejecting the hypothesis of no trend in data (e.g., Mann-Kendall test). Trend estimation can exploit many models, e.g, (multiple) linear regression, regression smoothing, Sen's slope estimation, autoregressive models, non-linear models. In the common case of presence of correlation among multiple aging indicators, data transformation, feature selection, or dimensionality reduction techniques are used. An example is Principal Component Analysis followed by regression [15]. Time series analysis was appleid in 14 out of 29 measurements-based studies. Relevant examples include the works by Araujo *et al.* on the Eucalyptus cloud computing framework [16]. They adopt several regression models including *linear*, *quadratic*, *exponential growth*, and *Pearl-Reed logistic* models, to predict memory consumption trends and schedule software rejuvenation properly.
- *Machine Learning*. Machine learning algorithms in SAR are used to detect/predict the possible aging system state. We found them applied 6 out of 29 times in the cloud domain. Cloud-related examples include the works by Sudhakar *et al.* [17], who use Artificial Neural Networks to capture non-linear relationships between resource usage and time to failure in cloud systems; Avresky *et al.* present a framework using machine learning for predicting failures caused by accumulation of anomalies and a proactive scale-up/scale-down technique in the cloud [18]. Simeonov proposes a framework with three VMs, one master and two identical slaves; the slaves send health data to the master, which predicts aging based on machine learning algorithms [19].
- *Threshold-Based Approaches* define thresholds for some aging indicators, so as to trigger rejuvenation when they are exceeded. an example is the work by Silva *et al.* adopts thresholds on mean response time and on quality of service indicators in a virtualization-based rejuvenation approach [20]. They propose a rejuvenation framework called *VM-Rejuv*, exploiting virtualization to optimize recovery, with a threshold-based aging detector module.

Other techniques include *grey correlation*, used to compute a metric of aging based on fuzzy evaluation [21], and two heuristic approaches not falling in the previous categories.

Metrics: measurement-based approaches gather aging data through probes at system level related to resources exhaustion (e.g., memory, storage), as well as at user level (e.g., response time, throughput). Figure 7 summarizes the used metrics.

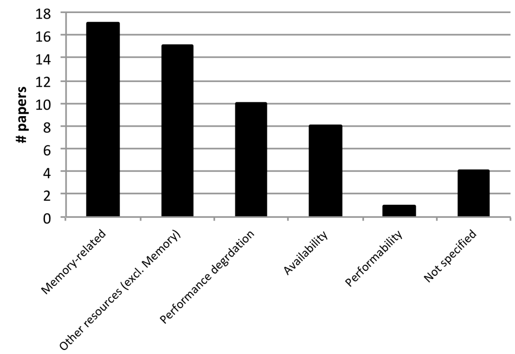

Fig. 7. Metrics used in measurements-based studies

- Memory indicators are very common in cloud-related studies, appearing in 17 out of 29 cases. Examples include the free RAM, caching, buffers size, the resident set size (RSS) of processes of interest, the swap space.
- Other resources include CPU, power consumption, number of threads/processes, storage space, network metrics. In cloud systems, indicators can be measured at any layer of the virtualization stack, e.g., at application/OS layer within the VM, at VM layer to probe the state and resource consumption of the VMs (e.g., for load balancing, scaling, VM migration and rejuvenation decisions) and at VMM layer. Besides system resources, indicators of interest include the number of VM allocations/releases, the VM start/stop time, the time to migration.
- Dependability attributes also appear in these studies, but never alone (as they are not the direct subject of "measurement"). Availability is the most common case.

C. Hybrid analysis techniques

Hybrid solutions adopt a stochastic model to describe the phenomenon, and determine the model parameters through measurement, that is, via observed data. Solutions of this type, although not much common, are able to put together the advantages of both approaches. 5 out of 72 studies adopted this strategy. As modelling, they adopt 5 different formalisms (SPN, SRN, SMP, RBD and a *Markov Renewal Processes*

(MRP)). As for measurements, 3 of them use time series analysis for prediction-based rejuvenation while 2 use a threshold-based approach. As aging indicators, 3 use availability coupled with either memory-related or performance degradation indicators; 1 adopts reliability instead of availability, and 1 adopts only memory indicators. An example includes the hybrid solution by Liu *et al.* [22]: they measure the trends of various resources in a cloud-based streaming system with ATM endpoints, including CPU, storage, network, at several layers and use them to parametrize a model to schedule rejuvenation.

In the above classes of studies, a further analysis factor is workload dependency. Since aging has been shown to be correlated with workload [23], [24], some studies accounted for its impact. An example is by Bruneo *et al.*, who present a workload-based analysis of VMM aging and rejuvenation under different policies for availability maximization [25].

V. REJUVENATION TECHNIQUES

Software rejuvenation can act on the virtualization infrastructure (VI) and/or on the running VMs. About the VI rejuvenation, the key component subject to rejuvenation most of the times is the VMM. Alternative strategies depend on whether rejuvenation affects only the VMM, or also the VMs running on it [26]. Figure 8 schematizes the rejuvenation techniques at VMM and at VM level, while Figure 9 and 10 report the breakdown of papers by rejuvenation technique.

VMM can be rejuvenated by the following techniques [26]:

- **Cold-VM rejuvenation**, the simplest and most used approach, which simply shuts down the hosted VMs before triggering the VMM rejuvenation (e.g., via restart) and then restarts the VMs after the completion of VMM rejuvenation. This indirectly rejuvenates also the VMs.
- **Warm-VM rejuvenation**, in which the execution state of VMs is stored to persistent memory and resumed after the restart of the VMM, so as to reduce the downtime of VMs compared to Cold-VM. Though, in this case, VMs are not rejuvenated. This operation can be quickly performed by an on-memory suspend/resume mechanism (the memory image of VMs is preserved in RAM during the VMM restart rather than on persistent storage [27]).
- **Migrate-VM rejuvenation**, in which the downtime is further reduced compared to previous strategies, by migrating a VM to another host during the VMM rejuvenation, making VMs available. This does not rejuvenate VMs, and is limited by the capacity of other hosts to accept migrated VMs. The technique can be divided based on the type of live VM migration (*stop-and-copy* or *pre-copy*), and the policies to return back to the original host (*return-back* or *stay-on*). The *stop-and-copy* migration stops the VM operations and copy the memory contents to the destination server. In the *pre-copy* variant, the VM memory is copied to the destination server without stopping its operation, hence causing dirty pages in the copied memory that are then updated by a stop-and-copy approach. At VM restore, a *return-back* policy migrates the VMs back to the original host soon after the VMM rejuvenation; a *stay-on* policy allows the migrated VM to keep on running on the new server. Machida *et al.* found that a pre-copy migration and return-back policy are generally better than the a stop-and-copy migration and a stay-on policy [26]. Kourai *et al.* [28] present *VMBeam*, a technique using *zero-copy* migration. It starts a new virtualized system at the same host by using nested virtualization, and then migrates all the VMs from the aged virtualized system to the clean one by relocating the VMs memory, without any copy.

Rejuvenation at VI level is done also beyond the VMM, e.g.:

- **VI Micro-reboot**. This is a complex technique to perform a fine-grained reboot of software modules (micro reboot), tailored for the system-level VI software. For instance, Le *et al.* [29] applied micro-reboot to all modules of the Xen virtualization software, consisting of three main components: the privileged virtual machine, the device driver virtual machine, and the VMM.
- **VI Resource Management**. Dabrowsk *et al.* aim to detect and eliminate the number of orphan VMs, to quantify and reduce the leakage of VMs caused by attacks [30]. Although the cause of aging is security in this case, the approach can be generalized to improve the availability of the VI by periodically cleaning the resources it manages.

Cold-VM is the most used approach with 16 occurrences; Warm-VM and Migrate-VM count 10 occurrences. Micro-reboot appears 4 times, which is a non-negligible percentage.

In order to rejuvenate the VMs, besides Cold-VM, conventional replication approaches can be applied:

- **VM Failover** is a replication-based approach. The idea is to redirect, upon detection of aging in a VM, all the incoming requests to a another VM, and then rejuvenate the aged VM, e.g., by restart. *Passive* strategies are distinguished based on the replication policy as: *cold*, *warm* and *hot* standby, depending on whether the standby node is completely powered off (*cold*), it is up and running with data mirrored regularly but the replicated software is off (*warm*) or it is up and running with data mirrored in near real time and the replicated software is also up and running but without processing data or requests (*hot*). *Active* failover foresees that both replicas are up and running and process data or requests in parallel.

  The most common strategy we encountered is by far the *passive failover*. An example is the mentioned framework, *VM-Rejuv*: a *load balancer* VM redirects requests to an *active* VM while it is correctly working and monitors it for aging; when rejuvenation is triggered, new requests are redirected to a *standby* VM [20]. Chang et *al.* model rejuvenation at both VMM- and VM-level, using, for the latter, a passive VM failover with replicas on a same host kept synchronized by a heartbeat mechanism [14]. In this and other cases (e.g., [1]), researchers refer to *cold* and *hot* standby as, respectively, *passive* and *active* strategy, to distinguish the cases in which the standby is off or is running with the state transferred

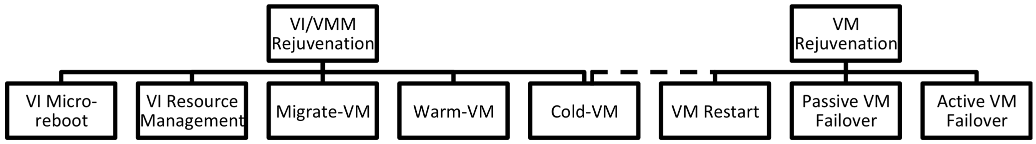

Fig. 8. Rejuvenation Techniques

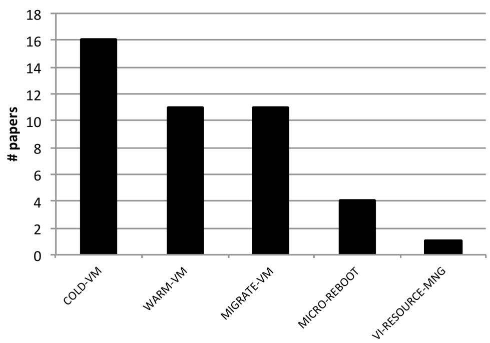

Fig. 9. VI/VMM Rejuvenation Techniques

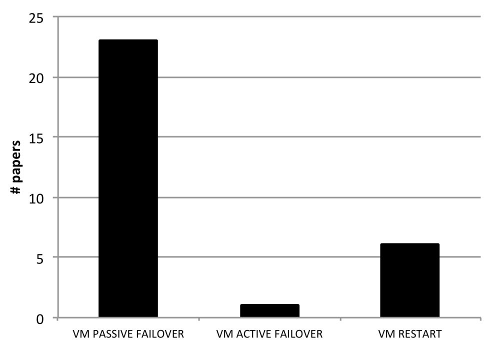

Fig. 10. VM Rejuvenation Techniques

at the end of each operation or periodically. A "pure" *VM Active Failover* is rarely encountered. We found it only in the work by Tan *et al.*, where active replicas (i.e., contemporarily processing requests) are used for intrusion tolerance, thus controlling the resource consumption by restarts, preventing the VMs from getting aged [31].

- **VM Restart**. Earlier studies applied raw restart of VMs, without any failover and no other means to reduce the downtime. This differs from *Cold-VM* just in the objective: *Cold-VM* aims at rejuvenating the VMM; in *VM Restart*, the goal is just to rejuvenate the VM. However, the action at VM level is exactly the same, i.e., a restart.

Besides rejuvenation, a further remediation action applied to VMs is *life extension*, in which a VM is allocated additional memory upon aging detection in order to temporarily prolong its life [32]. This however is not classified as a conventional rejuvenation action. Finally, note that other typical strategies, such as OS/node reboot, application/component restart, micro-reboot, can be appleid to hosts and/or to applications running in the VMs, but they are not specific to VIs or VMs.

VI. Discussion and conclusion

Some more isights beside what discussed so far are reported:

- SAR techniques in the cloud are increasingly emerging in the last years. 61% of research works in this area appeared in the last 5 years (2012-2017) and all of them appeared after 2007 – hence it is a relatively young research area. Looking at the increasing trend in cloud computing, it is easy to envision a similar increase of cloud-related SAR studies.
- *Model-based studies have a great importance for the cloud domain for their suitability to evaluate alternatives about the many rejuvenation strategies that can be applied*. The high number of studies about rejuvenation and the variety of techniques pushes researchers to develop new and more comprehensive models to account for cost/benefits of rejuvenation actions at several layers. The percentage of model-based studies discussing rejuvenation (23 out 27: 85%) witnesses such a trend.
- We detected an increasing trend in measurements-based approaches in the cloud domain, with even more papers than model-based ones (while results of a survey on all the SAR area in 2014 reported more model-based works [33]). The analyzed systems range from cloud applications (often web/application servers running on the cloud) to entire platforms, the most studied one being Eucalyptus. Likely, the wide availability of cloud computing software favours the experimentation of such strategies without excessive costs – a factor that could foster an increase of measurements-based analyses in the cloud domain in the future. Compared to the SAR literature, adopted techniques are basically the same, while metrics differ slightly because of the need of capturing behaviors related to the VMs management and/or of differentiating resources of virtual and physical host.
- *Rejuvenation in the cloud domain is addressed by many studies*. Overall, 76% of the studies (55 out of 72) studies include rejuvenation in their proposal and 16% (12 out of 72) deal exclusively with a rejuvenation technique. Indeed, the additional complexity and, at the same time, the possibilities offered by virtualization, push researchers for devising several new options for rejuvenation at several layers able to drastically reduced the cost. The increasing economical impact of performance of cloud-based systems will also likely push toward this direction. Research opportunities are, for instance, in investigating carefully the cost/benefit of replication-based strategies (including the scarcely addressed active replication), not only in

relation to the downtime reduction but also considering the cost of setting up and managing replicas, their energy consumption, the side-effects on security and on scalability, and so. Also, the overhead brought by rejuvenation is worth to be investigated and possibly reduced too, as the advantages of using VMs have been shown to be counterbalanced by a higher memory fragmentation induced by the virtualized environment [34].

- Emerging attributes related with aging causes and/or effects are being considered by some studies. Security is one of this: overall 6 out of 72 papers (8.3%) are related to security, wherein the cause of aging are attacks and intrusions. Energy consumption is a further example, very relevant for cloud contexts. 4 out of 72 papers (5.5%) dealt with the effect of aging in terms of energy consumption. These are worth to be investigated further.
- A final note is about the possibility of exploring SAR in emerging contexts related to the cloud/virtualization, such as: edge and fog computing, network function virtualziation and software-defined networks are some fields which could benefit from research on aging dynamics and rejuvenation counteractions.

REFERENCES

[1] J. Xu, X. Li, Y. Zhong, and H. Zhang, "Availability Modeling and Analysis of a Single-Server Virtualized System with Rejuvenation," *Journal of Software*, vol. 9, no. 1, pp. 129–139, 2014.

[2] A. Rezaei and M. Sharifi, "Rejuvenating high available virtualized systems," in *5th International Conference on Availability, Reliability, and Security (ARES)*. IEEE, 2010, pp. 289–294.

[3] T. A. Nguyen, D. S. Kim, and J. S. Park, "A comprehensive availability modeling and analysis of a virtualized servers system using stochastic reward nets," *The Scientific World Journal*, 2014.

[4] L. Han and J. guo Xu, "Availability models for virtualized systems with rejuvenation," *Journal of Computational Information Systems*, vol. 9, no. 20, pp. 8389–8396, 2013.

[5] M. Myint and T. Thein, "Availability Improvement in Virtualized Multiple Servers with Software Rejuvenation and Virtualization," in *Fourth International Conference on Secure Software Integration and Reliability Improvement (SSIRI)*. IEEE, 2010, pp. 156–162.

[6] F. Machida, V. F. Nicola, and K. S. Trivedi, "Job completion time on a virtualized server subject to software aging and rejuvenation," in *Third International Workshop on Software Aging and Rejuvenation (WoSAR)*. IEEE, 2011, pp. 44–49.

[7] ——, "Job Completion Time on a Virtualized Server with Software Rejuvenation," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 10, no. 1, pp. 10:1–10:26, 2014.

[8] M. Melo, P. Maciel, J. Araujo, R. Matos, and C. Araujo, "Availability Study on Cloud Computing Environments: Live Migration As a Rejuvenation Mechanism," in *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2013.

[9] J. Rahme and H. Xu, "A Software Reliability Model for Cloud-Based Software Rejuvenation Using Dynamic Fault Trees," *International Journal of Software Engineering and Knowledge Engineering*, vol. 25, no. 09n10, pp. 1491–1513, 2015.

[10] H. Okamura, K. Yamamoto, and T. Dohi, "Transient analysis of software rejuvenation policies in virtualized system: Phase-type expansion approach," *Quality Technology & Quantitative Management*, vol. 11, no. 3, pp. 335–351, 2014.

[11] M. Melo, J. Araujo, R. Matos, J. Menezes, and P. Maciel, "Comparative Analysis of Migration-Based Rejuvenation Schedules on Cloud Availability," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2013, pp. 4110–4115.

[12] T. Thein and J. S. Park, "Availability Analysis of Application Servers Using Software Rejuvenation and Virtualization," *Journal of Computer Science and Technology*, vol. 24, no. 2, pp. 339–346, 2009.

[13] M. Escheikh, K. Barkaoui, and H. Jouini, "Versatile workload-aware power management performability analysis of server virtualized systems," *Journal of Systems and Software*, vol. 125, pp. 365–379, 2017.

[14] X. Chang, Z. Zhang, X. Li, and K. S. Trivedi, "Model-Based Survivability Analysis of a Virtualized System," in *IEEE 41st Conference on Local Computer Networks (LCN)*, 2016, pp. 611–614.

[15] D. Cotroneo, S. Orlando, R. Pietrantuono, and S. Russo, "A measurement-based ageing analysis of the JVM," *Software Testing, Verification and Reliability*, vol. 23, no. 3, pp. 199–239, 2013.

[16] J. Araujo, R. Matos, V. Alves, P. Maciel, F. V. d. Souza, R. M. Jr., and K. S. Trivedi, "Software Aging in the Eucalyptus Cloud Computing Infrastructure: Characterization and Rejuvenation," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 10, no. 1, pp. 11:1–11:22, 2014.

[17] C. Sudhakar, I. Shah, and T. Ramesh, "Software Rejuvenation in Cloud Systems Using Neural Networks," in *International Conference on Parallel, Distributed and Grid Computing*, 2014, pp. 230–233.

[18] D. R. Avresky, P. D. Sanzo, A. Pellegrini, and B. C. adn Luca Forte, "Proactive Scalability and Management of Resources in Hybrid Clouds via Machine Learning," in *14th International Symposium on Network Computing and Applications (NCA)*. IEEE, 2015, pp. 114–119.

[19] D. Simeonov and D. R. Avresky, *Proactive Software Rejuvenation Based on Machine Learning Techniques*, ser. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering. Berlin, Heidelberg: Springer, 2010, vol. 34, pp. 186–200.

[20] L. Silva, J. Alonso, and J. Torres, "Using Virtualization to Improve Software Rejuvenation," *IEEE Transactions on Computers*, vol. 58, no. 11, pp. 1525–1538, 2009.

[21] F. Liu, J. Cao, J. Guo, and B. Zhang, "Research the Measurement Method of Software Aging in Cloud," ser. Applied Mechanics and Materials, vol. 392, 2013, pp. 779–782.

[22] Y. Liu, W. Liu, J. Song, and H. He, "An empirical study on implementing highly reliable stream computing systems with private cloud," *Ad Hoc Networks*, vol. 35, no. C, pp. 37–50, 2015.

[23] A. Bovenzi, D. Cotroneo, R. Pietrantuono, and S. Russo, "Workload Characterization for Software Aging Analysis," in *22nd International Symposium on Software Reliability Engineering (ISSRE)*, 2011, pp. 240–249.

[24] M. Ficco, R. Pietrantuono, and S. Russo, "Aging-related performance anomalies in the apache storm stream processing system," *Future Generation Computer Systems*, vol. 86, pp. 975 – 994, 2018.

[25] D. Bruneo, S. Distefano, F. Longo, A. Puliafito, and M. Scarpa, "Workload-Based Software Rejuvenation in Cloud Systems," *IEEE Transactions on Computers*, vol. 62, no. 6, pp. 1072–1085, 2013.

[26] F. Machida, D. S. Kim, and K. S. Trivedi, "Modeling and analysis of software rejuvenation in a server virtualized system with live VM migration," *Performance Evaluation*, vol. 70, no. 3, pp. 212–230, 2013.

[27] K. Kourai and S. Chiba, "Fast Software Rejuvenation of Virtual Machine Monitors," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 6, pp. 839–851, 2011.

[28] K. Kourai and H. Ooba, "Zero-copy Migration for Lightweight Software Rejuvenation of Virtualized Systems," in *Proceedings of the 6th Asia-Pacific Workshop on Systems (APSys)*, 2015, pp. 7:1–7:8.

[29] M. Le and Y. Tamir, "Applying Microreboot to System Software," in *IEEE Sixth International Conference on Software Security and Reliability (SERE)*, 2012, pp. 11–20.

[30] C. Dabrowsk and K. Mills, "Vm leakage and orphan control in open-source clouds," in *2011 IEEE Third International Conference on Cloud Computing Technology and Science*, Nov 2011, pp. 554–559.

[31] Y. Tan, D. Luo, and J. Wang, "Cc-vit: Virtualization intrusion tolerance based on cloud computing," in *2010 2nd International Conference on Information Engineering and Computer Science*, Dec 2010, pp. 1–6.

[32] F. Machida, J. Xiang, K. Tadano, and Y. Maeno, "Software Life-Extension: A New Countermeasure to Software Aging," in *23rd International Symposium on Software Reliability Engineering*, 2012, pp. 131–140.

[33] D. Cotroneo, R. Natella, R. Pietrantuono, and S. Russo, "A survey of software aging and rejuvenation studies," *J. Emerg. Technol. Comput. Syst.*, vol. 10, no. 1, pp. 8:1–8:34, Jan. 2014.

[34] J. Alonso, R. Matias, E. Vicente, A. Maria, and K. Trivedi, "A comparative experimental study of software rejuvenation overhead," *Performance Evaluation*, vol. 70, no. 3, pp. 231–250, 2013.