Progressive Recovery for Cloud-Based Infrastructure Services

Mahsa Pourvali[1], Kaile Liang[2], Feng Gu[3], Khaled Shaban[4], Samee Khan[5], Nasir Ghani[1]
[1]*University of South Florida,* [2]*University of New Mexico,* [3]*VM Ware,* [4]*Qatar University,* [5]*ND State University*

***Abstract*—Cloud-based services allow users to outsource their application and infrastructure needs over external datacenter/ networking facilities. However, as these offerings gain traction, there is a pressing need to address disaster recovery concerns, particularly for (virtual network) infrastructure services. However, only a handful of studies have looked at designing robust storage/computing/networking service interconnections for such scenarios. Moreover, these efforts have primarily focused on pre-fault protection design and not post-fault recovery. Indeed, the latter concern is of critical importance since physical repairs will likely occur in a time-staged manner. Hence this paper introduces the concept of progressive recovery for cloud-based infrastructure services and presents some of the first solutions in this space.**

***Index Terms*—Cloud networks, network virtualization, infrastructure as a service, disaster recovery, progressive recovery**

I. INTRODUCTION

Cloud services provide users with access to vast amounts of external storage, computing, and networking resources. Today many businesses are redeploying their applications to run in the cloud, achieving much-improved cost efficiency and scalability. Moreover, with advances in network virtualization, organizations are also starting to outsource their infrastructure needs, i.e., via *infrastructure as a service* (IaaS) offerings. Namely, advances in network virtualization [1] allow operators to provision custom *virtual network* (VN) services. Here, a typical configuration consists of a set of *distributed* storage/computing resources, i.e., VN nodes, interconnected by a set of bandwidth pipes, i.e., VN links. These VN nodes are embedded onto resource pools at datacenter nodes, and VN links are mapped onto underlying network connections.

Now many studies have looked at mapping VN demands over physical cloud infrastructures comprising of datacenter nodes and interconnecting network substrates [2], i.e., *VN embedding* (VNE) problem. However, as IaaS services gain traction, reliability and business continuity concerns are coming to the fore. In particular, large disaster events can cause extensive damage to physical facilities, e.g., natural disasters, cascading power outages, malicious *weapons of mass destruction* (WMD) attacks, etc [3]. In turn, these occurences can severely disrupt virtualized infrastructure services. Hence several studies have addressed VN survivability design. For example, researchers have proposed backup VN node and link provisioning schemes to recover from single link [4],[5] and single node failures [6],[7]. Others have also looked at disaster recovery with multiple node and link outages [8]-[10].

Nevertheless, the above schemes only focus on pre-provisioning backup resources prior to failures. Although such strategies may suffice for single isolated faults, they cannot guarantee IaaS service recovery against randomized multi-failure disaster events [3],[11]. It is therefore crucial to also consider *post-fault* restoration schemes to mitigate cloud IaaS (VN service) outages after a disaster. Now damaged infrastructures will likely be reparied in a staged *progressive* manner as backup resources become available. During these transitions, IaaS setups will still have to offer some level of partial/degraded services support to affected customers. Hence the detailed allocation (scheduling) of repair resources during multiple recovery stages will have a direct impact on service down times and operator penalties. Although some efforts have looked at progressive recovery for point-to-point flows [12],[13], there are no known studies within the context of more complex virtualized infrastructures (IaaS services).

In light of the above, this contribution presents one of the first studies on progressive post-fault disaster recovery for virtualized infrastructure services. The paper is organized as follows. First, Section II presents a brief survey of some related background work. Several new heuristics are then presented in Section III, along with peformance evaluation results in Section IV. Conculsions and future work directions are then presented in Section V.

II. BACKGROUND

Since the VNE design problem is known to be NP-hard [1], researchers have proposed a variety of solutions using optimization and heuristic methods, see survey in [2]. Most schemes here focus on minimizing resource utilization or maximizing carried loads/revenues. Furthermore, survivable VNE design has also been studied. For example, [4] and [5] have proposed link-disjoint path pair schemes for single link failures. Meanwhile, others have also studied VN recovery for single node failures, i.e., [6] presents two protection schemes using single or multiple backup nodes (along with backup VN link routing). Additionally, a more efficient (but also more complex) single node backup scheme is presented in [7] to re-map all nodes after a failure.

Recent studies have also looked at more robust VN disaster recovery schemes for multiple node/link failures. These strategies assume pre-defined disaster regions (a-priori risk models) with given occurrence probabilities and conditional node/link failure rates [11]. For example, [8] and [9] compute separate VN mappings for each region and then condense them using resource sharing. An incremental scheme is also proposed to add backup VN nodes/links to working mappings and protect against each failure region. However, these strategies still give high resource consumption. As a result, [10] proposes more efficient disaster recovery schemes using region partitioning. Namely, failure regions are separted into two groups and disjoint mappings computed for each (working, protection).

Regardless, the above schemes only focus on pre-provisioning of backup resources. However it is very difficult (if not impossible) to predict all possible failure events and their node and link outages. Hence even the most robust survivability schemes will be vulnerable, and one must consider post-fault recovery strategies. Namely, infrastructure repairs will likely occur in a staged, progressive manner as backup resources become available. Hence there is a pressing need to develop *progressive* recovery schemes to carefully allocate/schedule resources and improve post-fault VN performance. This issue is now addressed further.

III. PROGRESSIVE RECOVERY SCHEMES

Progressive recovery focuses on multi-stage resource placement. Before detailing the related solutions, however, the required notation is introduced. Namely, the physical cloud substrate is given by a graph $\mathbf{G}_s$=($\mathbf{V}_s, \mathbf{E}_s$), where $\mathbf{V}_s$ is the set of datacenter nodes, and $\mathbf{E}_s$ is the set of network links. Here each node $v_i \in \mathbf{V}_s$ has maximum resource capacity of R^{max} units, and each physical substrate link $e_i \in \mathbf{E}_s$ has maximum bandwidth of B^{max} units. Meanwhile, an IaaS service request (i.e., VN demand) is represented by the graph, $\mathbf{G}_v$=($\mathbf{V}_v, \mathbf{E}_v$), where $\mathbf{V}_v$ is the set of virtual nodes and $\mathbf{E}_v$ is the set of virtual links. Without loss of generality, each VN node $v_i \in \mathbf{V}_v$ requires r_i in node resources, and each VN link $e_i \in \mathbf{E}_v$ requires b_i in bandwidth capacity.

Meanwhile, a disaster event occurs at T_0 and is followed by a series of recovery stages at times T_i, $i = 1, 2, ...T_i$, where $T_i > T_{i+1}$, see Figure 1. The resource levels of all failed nodes and links drop to zero after the initial disaster event and then start to increase as repair resources are installed during the recovery stages (until full recovery). Therefore all failed or partially-recovered physical nodes and links in stage i are eligible to receive repair resources, and these are denoted by the sets $\mathbf{F}_v^i$ and $\mathbf{F}_e^i$, respectively (where $\mathbf{F}_v^0$ and $\mathbf{F}_e^0$ are the initial set of failed physical nodes and links). Hence the initial node resource loss is given by $X_0 = \sum_{v_i \in \mathbf{F}_v^0} R_i$ and the initial network link capacity loss is given by $Y_0 = \sum_{e_i \in \mathbf{F}_e^0} B_i$ (Figure 1). Finally, the amount of datacenter repair resources received in the i-th stage is given by X_i (e.g., computing racks, storage disks) and the amount of link bandwidth repair resources is given by Y_i (e.g., optical link transponders, switching line card units). It is also assumed that all resources are assigned at the integral granularity level

Several progressive recovery heuristics are now presented. These schemes first distribute incoming repair resources to affected nodes/links and then use any standard VNE algorithms to re-map failed VN nodes/links (IaaS services). Now one of the main objectives here is to try to allocate resources to minimize the duration of service disruption, i.e., or maximize the number of recovered VN demands in each stage. To better achieve this aim, the schemes only consider eligible nodes that are one hop away from one/more working or partially-recovered nodes, i.e., *candidate node set*, $\mathbf{F}_v^{'i} \in \mathbf{F}_v^i$. Similarly, only eligible links with non-failed endpoint nodes are

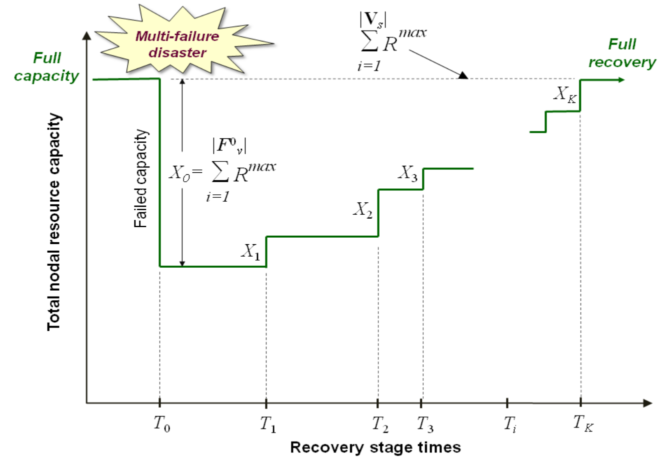

Fig. 1. Progressive recovery stages (case of node resources shown)

considered, i.e., *candidate link set*, $\mathbf{F}_e^{'i} \in \mathbf{F}_e^i$. This selectivity improves the chances that repair resources are accessible from working nodes/links more quickly. Consider the details.

A. Random Placement (RP)

This scheme performs random placement and is used for baseline comparison. Namely, node repair resources in stage i (X_i) are assigned to a random (failed, partially-recovered) candiate node in $\mathbf{F}_v^{'i}$. Similarly, link repair resources (Y_i) are assigned to a random candidate link in $\mathbf{F}_e^{'i}$ interconnecting two candidate nodes. If there are any leftover resources here, then the procedure is repeated for other randomly-selected nodes (links) until all resources are assigned or all candidate nodes (links) are restored to their pre-fault capacities.

B. Physical Node/Link-Degree (P-NLD)

This scheme assigns resources based upon (static) physical node degrees in the original working network. The aim here is to try to recover nodes with higher connectivity as they may provide more available routes to restore demands. Hence incoming node repair resources in stage i are assigned to the candidate node in $\mathbf{F}_v^{'i}$ with the highest node degree in $\mathbf{G}_s$=($\mathbf{V}_s, \mathbf{E}_s$). Next, the candidate links in $\mathbf{F}_e^{'i}$ are ordered by decreasing node degrees of their end-point nodes, and link repair resources are directed to the most-connected link. As per the RP scheme, this process is also repeated for any leftover/unused repair resources.

C. Virtual Node/Link-Degree (V-NLD)

This scheme assigns resources to candidate nodes based upon their dynamic IaaS (VN demand) load levels prior to the failure. The goal here is to place resources at damaged network regions carrying the most load and hence accelerate recovery of failed VN demands. Namely, the candidate nodes in stage i, $\mathbf{F}_v^{'i}$, are first ordered by decreasing VN load (measured in terms of embedded VN nodes). The first node is then selected for resource placement. Similarly, candidate links in $\mathbf{F}_e^{'i}$ are ordered by decreasing number of carried VN link connections (prior to failure), and link repair resources are assigned to the one carrying the highest load. These assignments are repeated for any leftover resources to speed up recovery.

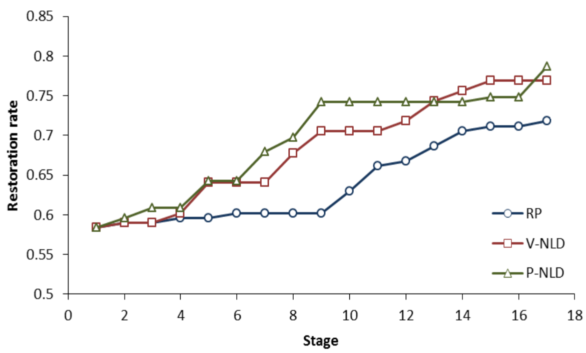

Fig. 2. Fraction of failed VN demands recovered

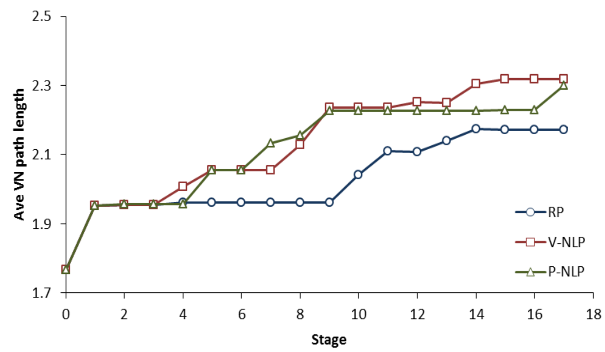

Fig. 3. Average VN link path lengths

IV. PERFORMANCE EVALUATION

Progressive recovery is tested using custom-developed models in the $OPNETModeler^{TM}$ simulation tool. All tests are done for a 48 node/118 link topology in which substrate nodes have 100 units of capacity and substrate links have 10,000 units of bandwidth. VN requests are also specified as random graphs averaging 3-5 nodes and 4-7 links each (generated via Inet topology tool). All VN nodes require between 10-20 units of capacity (uniform), and all VN links require between 50-1,000 units of bandwidth (uniform). Requests arrive randomly (exponential, mean 60 sec) and have infinite duration. These values model realistic long-standing IaaS setups. Furthermore, an average of X_i=100 (Y_i=20,000) units of node (link) repair resources are delivered in each stage, and all (partially-failed) VN demands are re-mapped using the single-stage VNE heuristic in [14] (albeit any other algorithm can be used).

Tests are done for a large disaster region failing close to 30% of nodes/links near the network center, and all results are averaged over 10 runs. Foremost, Figure 2 plots the average percentage of *fully-restored* failed VN demands during the successive stages, i.e., restoration rate. These results indicate full *physical* network recovery after 17 stages, with the more selective node degree strategies yielding much faster *virtual* network recovery. For example, the P-NLD and V-NLD schemes give 10-15% higher recovery rates during the middle stages. However, the dynamic V-NLD scheme does not necessarily outperform the static P-NLD scheme. In addition, no scheme achieves full (i.e., 100%) VN recovery, since re-mapping prior demands over partially-recovered substrates yields different (i.e., less efficient) embeddings.

Meanwhile, Figure 3 plots the average path lengths of the recovered VN links, i.e., to measure bandwidth utilization. Here the baseline random (RP) scheme gives slightly lower hop count values than the other two strategies, particularly during the middle recovery stages. This is expected since this scheme gives lower VN restoration rates and thereby consumes fewer bandwidth resources. Meanwhile, the P-NLD and V-NLD schemes exhibit very close performances, with the latter showing slightly larger usages in the final recovery stages.

V. CONCLUSIONS & FUTURE WORK

This paper presents one of the first studies on progressive recovery for cloud-based infrastructure (network virtualization) services. The problem is introduced and several heuristic algorithms are proposed using random, physical node degree, and virtual load information to place repair resources. Overall, the findings indicate that the latter two strategies give faster recovery of failed demands. This contribution provides a good framework from which to expand in to more detailed studies using formal optimization and meta-heuristic strategies.

ACKNOWLEDGMENT

This work was made possible by the NPRP 5-137-2-045 grant from the QNRF (member of Qatar Foundation). The statements herein are solely the responsibility of the authors.

REFERENCES

[1] M. Chowdhury, R. Boutaba, "Network Virtualization: State of the Art and Research Challenges, *IEEE Communications Magazine*, Vol. 47, No.7, July 2009, pp. 20-26.

[2] A. Fischer, J. Botero, *et al*, "Virtual Network Embedding: A Survey", *IEEE Comm. Surveys & Tutorials*, Vol. 15, No. 4, 4th Quater 2013.

[3] M. Rahnamay-Naeini, J. Pezoa, G. Azar, N. Ghani, M. Hayat, "Modeling Stochastic Correlated Failures and Their Effects on Network Reliability," *IEEE ICCCN 2011*, Maui, Hawaii, August 2011.

[4] Y. Chen, *et al*, "Resilient Virtual Network Service Provision in Network Virtualization Environments," *IEEE International Conference on Parallel and Distributed Systems (ICPADS) 2010*, China, Dec. 2010.

[5] T. Guo, *et al*, "Shared Backup Network Provision for Virtual Network Embedding," *IEEE ICC 2011*, Japan, June 2011.

[6] H. Yu, *et al*, "Cost Efficient Design of Survivable Virtual Infrastructure to Recover from Facility Node Failures," *IEEE ICC 2011*, June 2011.

[7] C. Qiao, *et al*, "A Novel Two-Step Approach to Surviving Facility Failures," *OFC 2011*, Los Angeles, CA, March 2011.

[8] H. Yu, *et al*, "Survivable Virtual Infrastructure Mapping in a Federated Computing and Networking System under Single Regional Failures," *IEEE GLOBECOM 2010*, Miami, USA, December 2010.

[9] G. Sun, *et al*, "Efficient Algorithms for Survivable Virtual Network Embedding," *Asia Communications and Photonics Conference and Exhibition (ACP) 2010*, China, December 2010.

[10] F. Gu, H. Alazemi, A. Rayes, N. Ghani, "Survivable Cloud Networking Services," *IEEE ICNC 2013*, San Diego, January 2013.

[11] H. Lee, K. Lee, E. Modiano,"Diverse Routing in Networks with Probabilistic Failures, *IEEE/ACM Transactions on Networking*, Vol. 18, No. 6, December 2010, pp. 1895-1907.

[12] J. Wang, C. Qiao, H. Yu, "On Progressive Network Recovery After a Major Disruption," *IEEE INFOCOM*, Shanghai, China, April 2011.

[13] S. Nurre, *et al*, "Restoring Infrastructure Systems: An Integrated Network Design and Scheduling (INDS) Approach", *European Journal of Operational Research*, 2012, pp. 794-806.

[14] F. Gu, *et al*, "Regional Failure Survivability for Cloud Networking Services Using Post Fault Restoration," *IEEE Systems of System (SoSE) 2013*, Hawaii, June 2013.

[15] X. Cheng, *et al*, "Virtual Network Embedding Through Topology-Aware Node Ranking," *ACM SIGCOMM Computer Communication Review*, Vol. 41, No. 2, April 2011, pp. 38-47.