Proactive Relocation for Survivable Cloud Services

Proactive Relocation for Survivable Cloud Services

Carlos Natalino and Paolo Monti

Electrical Engineering Dept., Chalmers University of Technology, 412 96 Gothenburg, Sweden
E-mail: {carlos.natalino, mpaolo}@chalmers.se

ABSTRACT

Improving the survivability of optical cloud services is costly due to the numerous resources involved. In this work, we introduce a proactive backup storage relocation (PBSR) strategy, which relocates backup storage in advance to better prepare services for potential failures and reduce the need for reactive relocation. Results show that our proposal enhances availability and restorability without increasing the service blocking ratio, while reactive relocations are reduced by a factor of up to half.

Keywords: Storage backup, availability, restorability, optical networks, cloud service, simulation, resiliency.

1. INTRODUCTION

Nowadays, telecommunication services require a combination of connectivity and computing resources. One approach to support them is the so-called *optical cloud network* paradigm, where distributed computing resources are connected via optical networks. It is well known that as services and networks evolve, the requirements for availability become more stringent [1]. In the case of optical networks, there are various sources of outages, ranging from hardware degradation and power outages to fiber cuts [2], [3]. However, ensuring survivability in optical cloud networks is challenging, mainly due to the numerous resources supporting these services and the large amount of data to be transmitted and processed. The latter is increasingly significant with, e.g., the growing popularity of large language models [4].

In traditional service management, survivability is ensured through *protection*, which involves setting up backup resources for when the primary resources fail. However, this method is costly and inefficient because many backup resources remain unused most of the time. On the other hand, services can rely on *restoration*, which does not involve provisioning backup resources. Instead, it relies on the on-the-fly re-provisioning of services around the failed resources. There are two main drawbacks to this approach. First, unlike protection techniques, it cannot guarantee 100% failure recoverability because bottlenecks may prevent the source node from accessing/using an alternative data center (DC). Second, services may experience non-negligible downtime due to the online computation and selection of recovery options.

To overcome these limitations, the paper proposes the proactive backup storage relocation (PBSR) strategy. PBSR utilizes a provisioning approach that assigns backups only for storage while relying on restoration for the connectivity and computing resources. PBSR then continuously monitors the fluctuations of resource usage across the infrastructure, detecting when bottleneck conditions appear. When a bottleneck condition is identified, PBSR evaluates whether services would benefit from a proactive relocation of their backup storage to a different DC in case of failure. Results demonstrate that the strategy enhances service availability and restorability performance while significantly reducing the necessity for reactive relocations that result in service downtime.

2. THE PROACTIVE BACKUP STORAGE RELOCATION (PBSR) APPROACH

This work considers a network of nodes interconnected by fiber links (Figure 1). The nodes are categorized as transit nodes, which forward traffic, and DCs nodes, which process traffic using processing and storage resources, measured in terms of processing units (PUs) and storage units (SUs), respectively. The link capacity is measured in terms of connectivity units (CUs). We assume that the number of DCs in a network is far lower than the number of transit nodes and that DC resources are abundant compared to link resources. Both nodes and links may be subject to failures, such as power grid failures or fiber cuts.

We assume a dynamic arrival of cloud service requests. Each cloud service request is associated with a source node requiring a certain amount of PUs, SUs, and CUs between the source node and the selected DC. The request does not specify any DC to be used. Instead, it leaves the decision to the provisioning algorithm to select a data center based on the network's current resource availability.

The entities requesting the service want to improve survivability while reducing idle resources. To achieve this, they use the storage protection with connectivity and processing restoration (SCORE) strategy [5]. SCORE provisions primary and backup resources to improve survivability while reducing overhead. It follows a *closest-available-DC* approach, provisioning primary processing, storage, and connectivity resources at the primary DC. Additionally, SCORE provisions one replica of the service storage at a DC different from the primary one.If a failure prevents the source node from reaching the resources at the primary DC, SCORE can tentatively restore the service by relying on the backup storage and then provisioning connectivity and processing resources. If there are insufficient resources (connectivity and processing) to restore the service to the backup DC, SCORE can finally attempt to perform a connectivity restoration with service relocation (CR+SR) [6].

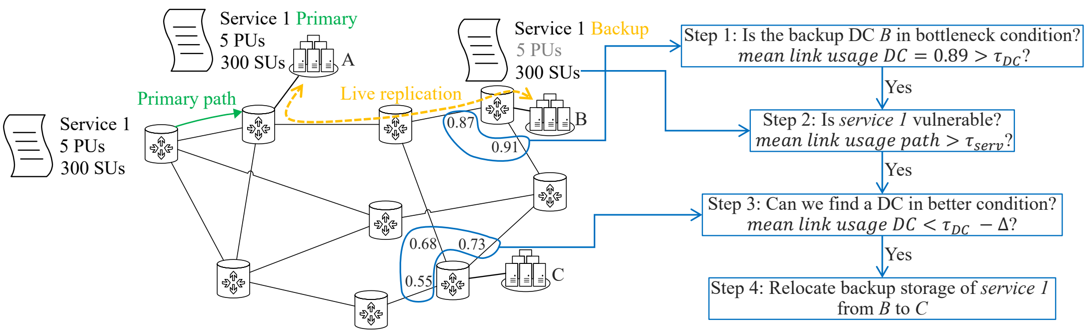

Service 1 Primary
5 PUs
300 SUs
A
Service 1 Backup
5 PUs
300 SUs
B
Service 1
5 PUs
300 SUs
Primary path
Live replication
0.87
0.91
0.68
0.73
0.55
C
Step 1: Is the backup DC B in bottleneck condition?
$mean\ link\ usage\ DC = 0.89 > \tau_{DC}$?
Yes
Step 2: Is *service 1* vulnerable?
$mean\ link\ usage\ path > \tau_{serv}$?
Yes
Step 3: Can we find a DC in better condition?
$mean\ link\ usage\ DC < \tau_{DC} - \Delta$?
Yes
Step 4: Relocate backup storage of *service 1* from B to C

Figure 1: Example of the proactive backup storage relocation (PBSR) strategy. Resource usage over the links connected to backup DCs is analyzed, and a proactive relocation decision is taken.

One drawback of SCORE is that although it considers available connectivity and processing resources when choosing the backup data center (DC), it does not reassess its decision on the best backup DC throughout the service's lifetime. Since they are not reserved beforehand, these resources may be unavailable when needed after a node or link failure. As a result, normal fluctuations in resource usage across the network could prevent SCORE from effectively utilizing the backup storage. This could lead to additional service relocations due to failures that could have otherwise been prevented. This may result in a significant number of reactive relocations, causing downtime and negatively impacting service availability.

The PBSR utilizes the backup storage provided by SCORE to improve the survivability of cloud services (Fig. 1). Fluctuations in resource usage are primarily caused by the provisioning or release of services, with only the provisioning of services leading to increased resource usage. As a result, PBSR is activated when new services are provisioned in the network. Subsequently, PBSR identifies potential bottlenecks in the connectivity of DCs, as depicted in *step 1* of Fig. 1. The specific criterion for identifying bottlenecks may vary in different scenarios. For example, a criterion could be the mean link usage across all the DCs reaching a predefined threshold τ_{DC}. Once the criterion is met, the proactive backup storage relocation process starts.

In the proactive backup storage relocation process, PBSR checks the condition of all (or a subset of) the services to verify their ability to be restored in case one of their primary resources fails. A service is considered *vulnerable* if the availability of the connectivity and processing resources that can potentially be used in the event of a failure falls below a predefined threshold (*step 2* in Fig. 1). When a vulnerable service is identified, PBSR attempts to find an alternative DC to host the storage backup (*step 3* in Fig. 1). This is accomplished by evaluating all available DCs (excluding the primary and backup ones) and selecting the one with the lowest resource usage. To ensure that an alternative DC is in significantly better condition than the current backup DC, PBSR looks for the alternative DC with significantly lower usage than the current backup one. If the resource usage difference exceeds a predefined threshold, the service storage backup is moved from the current backup DC to the alternative one (*step 4* in Fig. 1). Otherwise, the storage backup remains at its current DC

PBSR has three critical parameters. Firstly, it involves a criterion for identifying which DC should be further analyzed (τ_{DC}). Relaxed criteria may fail to detect DCs with high resource usage, while strict criteria may result in additional processing overhead due to frequent execution of the proactive relocation process. Secondly, a threshold is required to identify vulnerable services (τ_{serv}). A relaxed threshold might miss too many vulnerable services, while a strict threshold may lead to unnecessary proactive relocations, incurring additional overhead. Finally, it involves setting a threshold for the minimum resource usage difference between the current and a potential alternate backup DC (Δ). A relaxed threshold will lead to more potentially unnecessary proactive relocations, increasing the overhead of PBSR. In contrast, a strict threshold may make PBSR less effective in preventing unreachable backup storage upon a link or node failure.

3. SYSTEM MODEL AND PROPOSED ALGORITHM

This section introduces an algorithm that implements the PBSR intuition described in the previous section. Let $G(N, L)$ be a graph with $|N|$ nodes and $|L|$ links. The set of nodes N consists of DC and transit nodes, denoted as N_{DC} and N_t, respectively, i.e., N=$N_{DC} \cup N_t$. Set S contains the active services, with S_n^{bkp} representing the services which have their backup storage provisioned at $n \in N_{DC}$. $P_s \in N_{DC}$ and $B_s \in N_{DC}$ represent the primary and backup DC of $s \in S$, respectively. S_n^{bkp} can be sorted by different criteria, such as the service time.

The first step of PBSR is to check whether or not a DC is in a bottleneck condition. Two alternatives for this stage are detailed next. The first one, called mean link usage (MLU), looks at all DCs together by computing the mean value of their link usage. If this value exceeds a predefined threshold τ_{DC}, all DCs are considered to be in a bottleneck state. The second alternative, called number of datacenters (NDC), looks at all DCs and checks if at least 50% of them have the value of their mean link usage exceeding a given threshold τ_{DC}. If this is true, all the DCs whose mean link usage value is above τ_{DC} are considered to be in a bottleneck state.

Algorithm 1: The PBSR proactive backup relocation process

```
  Input: $d$, $S_d^{bkp}$, $N_{DC}$, $\tau_{serv}$, $\Delta$
1 for each $s \in S_b^{bkp}$ do
2     $cur_{usage}$ = getBestPathToBackup($s$, $B_s$)
3     $alt_{dc}$, $alt_{usage}$ = getBestAlternativeBackup($s$, $N_{DC}$, $P_s$, $B_s$)
4     if $cur_{usage} > \tau_{serv}$ and $alt_{usage} < \tau_{serv}$ and $cur_{usage} - alt_{usage} \geq \Delta$ then
5         moveBackup($s$, $B_s$, $alt_{dc}$)
```

If a DC $d \in N_{DC}$ is in a bottleneck condition, the proactive relocation process described by Algorithm 1 is triggered. The algorithm iterates over S_d^{bkp} (line 1). For each service, the algorithm first checks all the path alternatives (up to a predefined number k) that can be used to connect the source node with the current backup DC. For each one of these path options, it computes the CU usage value and selects the one with the lowest cur_{usage} as the best (line 2). The algorithm then looks for an alternative backup DC that could be reached using a path with an even lower CU usage. For this reason (line 3), it iterates over N_{DC} and computes the CU usage value of the least used path to each one of the DC options. The DC (alt_{dc}) that can be reached with the best-performing path in terms of CU usage (alt_{usage}) is then considered as a potential candidate for storage relocation. The latter happens if the following conditions are all met: *(i)* s is a vulnerable service (i.e., $cur_{usage} > \tau_{serv}$), *(ii)* the path option to the potential new backup DC does not make s vulnerable (i.e., $alt_{usage} < \tau_{serv}$), and *(iii)* the difference between cur_{usage} and alt_{usage} exceeds a predefined threshold Δ.

4. PERFORMANCE ASSESSMENT

We assess the performance of PBSR via a custom discrete event simulator[1]. We consider the NSF topology with 14 nodes, out of which 4 are DCs, and 22 links. Each link contains 80 CU, represented as wavelengths in our simulation. We assume a single link failure scenario where failures are uniformly distributed among the links with a mean time to failure (MTTF) exponentially distributed with an average of $231 \cdot 10^{-5}$ [1/sec], and a mean time to repair (MTTR) exponentially distributed with an average $4,320$ [sec]. PUs and SUs at DCs are dimensioned such that they do not represent a bottleneck, following a similar process as in [5].

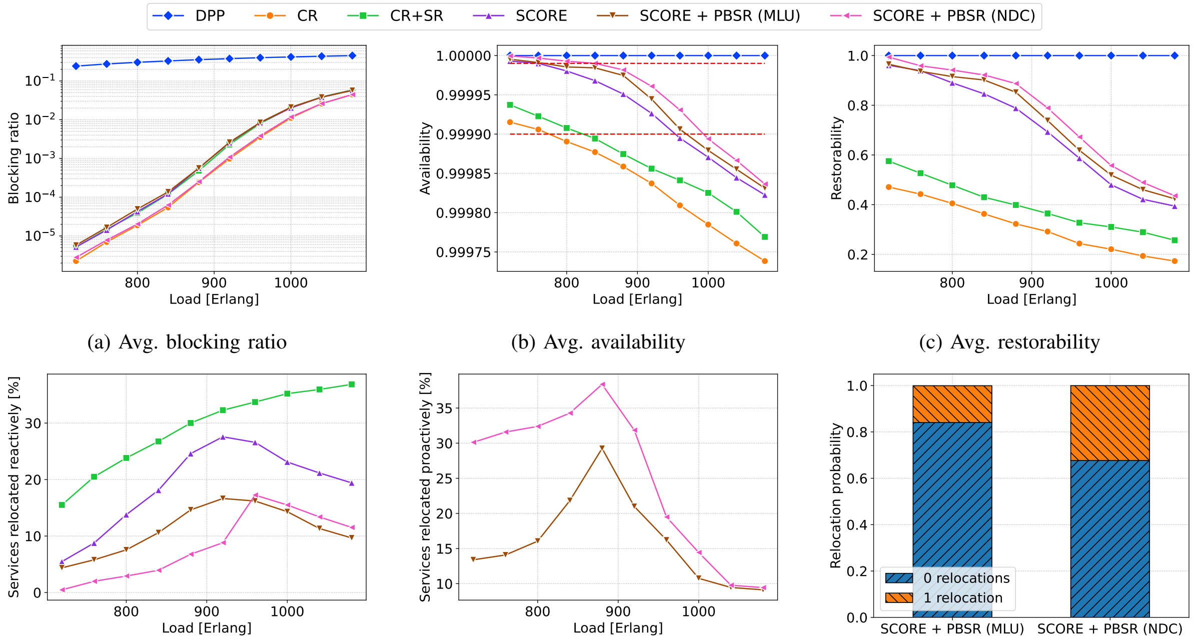
Cloud service requests arrive following a Poisson process. Their holding time is exponentially distributed with an average of 60 [h]. The mean inter arrival time is set based on the load [Erlang] chosen for the specific experiment. The source node of a request is selected uniformly among the transit nodes (N_t). Each service requires a single wavelength bi-directional lightpath between the source and a primary DC node. The number of required PUs follows a normal distribution with values 1, 2, 4, 8, 12, 16, 24, 32, 40, with average centered at 12. The number of required SUs also follows a normal distribution within the range of [10-500] [GB], with a 10 [GB] step and an average of 250 [GB]. Both distributions have a standard deviation equal to half the average value. The following results are the average of 400 experiments (seeds) per load, with 1 million arrivals per experiment. The values of the thresholds are set as follows: τ_{serv}=τ_{MLU}=τ_{NDC}=0.8, and Δ = 0.2.

PBSR is benchmarked against three strategies from the literature. Dedicated path protection (DPP) relies on the proactive assignment of connectivity, processing, and storage backup resources. Connectivity restoration (CR) relies on restoring connectivity to the primary DC by finding, after a failure, a new connectivity path, if available. Connectivity restoration with service relocation (CR+SR) also relies on connectivity restoration, but unlike CR, can, in the process, relocate the service to a different DC if the primary one cannot be reached [6]. Storage protection with connectivity and processing restoration (SCORE) is a hybrid approach that leverages backup storage resources while restoring connectivity and computing resources with service relocation if needed.

Each benchmarked strategy is evaluated according to the following metrics: *(i)* blocking ratio, i.e., number of rejected over the total number of cloud service requests; *(ii)* availability, i.e., ratio between service uptime and the total service time; *(iii)* restorability, i.e., number of successfully restored services over the total number of services disrupted by failures; *(iv)* percentage of relocations, i.e., the number of reactive service relocations performed over the total number of successfully restored services; *(v)* percentage of proactive relocations, i.e., the number of proactive relocations over the total number of successfully accommodated services in the network; and *(vi)* relocation probability, i.e., how probable it is (on average) to have services relocated a given number of times.

Figure 2 presents the results of the simulations. Fig. 2a shows the blocking ratio, highlighting that, except for DPP, all the other provisioning methods have similar performance. Fig. 2b highlights the four and five 9s availability thresholds. It can be seen how the introduction of PBSR, and in particular the version with the NDC criteria, extends the load value at which achieving five 9s availability is still possible, i.e., 760 Erlangs with SCORE vs. 840 Erlang with PBSR (NDC). Without PBSR, SCORE is able to guarantee five 9s availability only

[1] Available at https://github.com/carlosnatalino/java-anycast-cs-simulator



(a) Avg. blocking ratio

(b) Avg. availability

(c) Avg. restorability

(d) % of services relocated reactively

(e) % of services relocated proactively

(f) Relocation probability at 800 Erlang

Figure 2: Simulation results for the NSFnet topology.

up to 760 Erlang. Similar trends are observed in Fig. 2c, where PBSR (NDC) is able to improve the average restorability performance compared to SCORE.

However, most of the performance improvements of PBSR are in terms of the number of reactively relocated services, as shown in Fig. 2d. On average, over all values of the load, MLU reduces the ratio of services relocated reactively by 69% (from 15% to 9%), while NDC reduces the value by half. This improvement can be directly translated into lower service downtime, i.e., less time is needed to recover a service after a failure. Fig. 2e shows that the lower ratio of reactive relocations obtained by NDC comes at the cost of a higher number of proactive relocations than MLU. It is important to note, however, that the proactive relocations do not add downtime to the services. Finally, Fig. 2f provides a better understanding of how frequently services may be relocated for 800 Erlangs. Specifically, the services that are proactively relocated are moved only once.

5. FINAL REMARKS

In this work we proposed a new strategy to proactively relocate backup storage resources associated with cloud optical services. The proposed approach does not incur extra overhead in terms of blocking ratio while improving the average service availability and restorability performance. More importantly, the proposed strategy decreases the number of reactive relocations that cause service downtime upon failures by 69% and 100% depending on the criteria used to trigger the strategy. Potential future works encompass formulating the problem as linear programming and exploring more elaborate bottleneck detection mechanisms.

Acknowledgments: This work has been supported by Sweden's innovation agency VINNOVA within the framework of the EUREKA cluster CELTIC-NEXT project AI-NET-PROTECT (2020-03506).

REFERENCES

[1] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6g networks: Use cases and technologies," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 55–61, 2020.

[2] L. Wang, L. Wang, C. Wang, and C. Xie, "Unavailability analyses of hyperscale data center interconnect optical networks with optical layer protection," in *Proc. of ECOC*, 2024, p. Th3I.1.

[3] Q. Zhang, P. Layec, A. Pattavina, and M. Tornatore, "Resource re-allocation for pre-planned power outages in optical networks," in *Proc. of OFC*, 2024, p. W4I.7.

[4] J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn, and P. Villalobos, "Compute trends across three eras of machine learning," in *Proc. of IJCNN*, 2022, pp. 1–8.

[5] C. Natalino, A. Rostami, and P. Monti, "Storage protection with connectivity and processing restoration for survivable cloud services," in *Proc. of ICCCN*, 2021, pp. 1–9.

[6] C. Natalino, L. Wosinska, S. Spadaro, J. ao C. W. A. Costa, C. R. L. Francês, and P. Monti, "Restoration in optical cloud networks with relocation and services differentiation," *J. Opt. Commun. Netw.*, vol. 8, no. 2, pp. 100–111, Feb 2016.