Efficient Restoration of Virtual Machine Snapshots

Giannis Tzouros, Vana Kalogeraki
Athens University of Economics and Business, Athens, Greece
tzouros@aueb.gr,vana@aueb.gr

1 INTRODUCTION

In recent years, virtual machines (VMs) have become essential for cloud environments, as they virtualize software and hardware for distributed applications. VMs can recover from errors or create versions of an application by suspending their memory and store it as a snapshot. However, VMs and their snapshots consist of multiple file components which are essential for the VM to be fully functional. This file architecture makes the VMs more susceptible to data losses from cloud storage failures (faulty data writes from malfunctions in the cloud environment and unavailable storage nodes due to connectivity issues within the cloud or hardware and software issues occurring in the machine that runs the node) as well as outer factors (packet losses from disruptions or unstable connectivity during network transfer and external threats from attackers such as hacking and malware injections on to-be-uploaded data).

Cloud systems today deploy two fault tolerance methods to prevent failures: Replication and Erasure Coding. Replication creates multiple copies of a single file and stores them in multiple storage nodes at the expense of a large storage overhead, making it unfeasible for cloud systems. Erasure coding, on the other hand, generates parity data for a group of data blocks, achieving a significant reduction in storage overhead compared to replication. However, the parity data generated by most erasure codes cover a limited amount of lost data, making it impossible to fix bigger data erasures.Furthermore, the inconsistency between the data size of the VM's components may cause problems for the load balance of a cloud storage, regardless of applied load balancing policies.

An additional challenge for recovering VMs is the "cold start" problem occurring for a short time period right after a storage node is disabled or unavailable. Essentially, VMs suffer from prolonged booting times during a cold start and the cloud will need to access at least a few thousand pages in order to read a VM stored through multiple nodes. Thus, the cloud will face significant delays trying to read any memory pages stored in the unavailable node.

2 PROPOSED APPROACH

We propose a fault tolerance middleware which operates between data storing applications and the cloud environment, and deploys Diagonally Interleaved Coding [1], an extensive erasure coding method based on Reed-Solomon. Initially we reduce the load imbalance problem by providing a lossless compression scheme for the virtual machine and its snapshots, where we enclose the VM files into a unified archive and split it into multiple parts of equal data size. Then we order the parts into a grid and encode them with Diagonally Interleaved Coding. According to figure 1, the split parts are arranged into diagonal groups, which are encoded using Reed-Solomon, generating parity data for each diagonal. Thus, our method can achieve higher fault tolerance than simple erasure codes. After encoding the diagonal groups, we upload and store them into the cloud. In the attached poster, we present a brief description of the compression and Diagonally Interleaved Coding uploading and decoding processes.

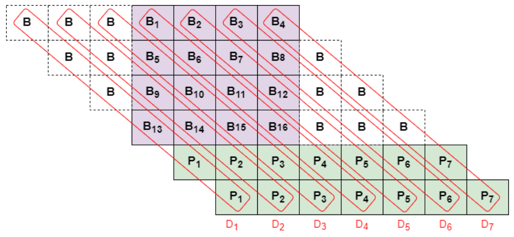
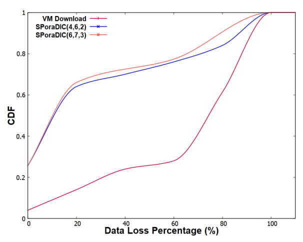
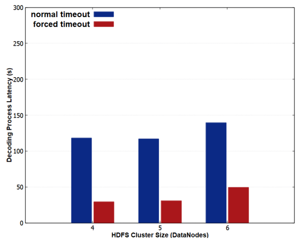


Figure 1: Representation of Diagonally Interleaved Coding



(a) CDF Data loss vs. simple VM



(b) Cold start latency

Figure 2: Early experiment results of our work

We present early experimental results to evaluate our middleware, as seen in Figure 2. Firstly, we compare our work in terms of data loss percentage during the recovery of a VM encoded with our work compared to recovering the same VM with no fault tolerance method deployed in case a storage node is unavailable. We show that our approach manages to suffer smaller data loss percentages compared to recovering a non-encoded virtual machine over multiple recovery attempts. Due to the random file placement policy of the cloud, the data loss results may vary each time we upload the VM. Secondly, we address the cold start problem after a data node failure.We deploy a forced timeout setting in our framework so that we can achieve significantly smaller delay times for degraded reads on missing data compared to a non-forced timeout method.

Acknowledgement. This research has partly been financed by the European Union through the EU ICT-48 2020 project TAILOR (No. 952215), the H2020 AutoFair project (No. 101070568) and the CoDiet project (No. 101084642).

REFERENCES

[1] D. Leong, A. Qureshi, and T. Ho. "On coding for real-time streaming under packet erasures". In: *2013 IEEE ISIT*. Istanbul, Turkey, pp. 1012–1016.