

Approximate Code: A cost-effective erasure code for multimedia applications in cloud storage systems

Huayi Jin¹

Abstract—Erasure codes are commonly used to ensure data reliability in cloud storage systems, where multimedia data produced by live video, autopilot, social media and security monitoring occupies large amounts of space. Typical erasure codes, such as Reed-Solomon (RS) codes, use several parity disks to fully recover failure disks. However, this is expensive, not only because the scenes in which multiple disks are simultaneously damaged are relatively rare, but also because they do not consider redundant information inside the multimedia data, which results in multiple complete parity disks being excessive.

Therefore, we propose Approximate Codes for multimedia data, which significantly reduce storage overhead by reducing the verification of redundant data. Approximate Codes provide complete recovery when fewer disks fail, and approximate recovery (recover most data) in the event of multiple disk failures. To demonstrate the effectiveness of Approximate Codes, we conduct several experiments in Hadoop and Alibaba Cloud systems. The results show that compared with the typical high-reliability erasure code scheme, Approximate Codes reduce the storage cost by 7.64% at the expense of reasonable video quality loss probability.

Index Terms—Erasure Codes, Approximate Storage, Multimedia, Cloud Storage

I. INTRODUCTION

Currently, many cloud storage systems use erasure codes to tolerate disk failures and ensure data availability, such as Windows [], Amazon AWS [] or Alibaba Cloud. It is known that erasure codes provide much lower storage overhead and write bandwidth than replication with the same fault tolerance.

Typical erasure codes schemes generate k check disks for a group of disks by calculation, which can tolerate any k disk failures in the group, such as RS-based code (RS, LRC), or XOR-based code (...). Other erasure codes (SD, STAIR) use the parity blocks to tolerate sector failures in addition to disk-level fault tolerance.

In cloud storage systems, erasure code schemes are essential for multimedia applications. Applications such as the video industry, autonomous driving, social media, and security monitoring generate numerous data each day while consuming large amounts of storage resources. This makes it extremely expensive to guarantee multimedia data availability through replication scheme.

Existing erasure codes are designed to completely recover corrupted data and use at least 3 additional check disks [] to ensure data availability. These methods are often excessive because scenes with 3 disks being corrupted at the same time are very rare as well as they do not consider that plenty of multimedia applications can tolerate a certain amount of data loss. For example, video data typically records at least 20 frames per second, which makes losing a few frames

difficult for a typical user to perceive. In addition, even if the multimedia data suffers a certain loss, the existing AI-based interpolation algorithm and super pixel algorithm can recover most of the damaged data [].

We also find that multimedia data is typically encoded and stored to save space, while the encoded multimedia data stream is non-uniformly sensitive to data loss, which makes it unsuitable to provide uniform fault tolerance using conventional erasure codes. Image coding algorithms such as JPEG convert pixels to the frequency domain, making some data segments more important than others, while video coding algorithms such as H264 only store key frames (I frame) completely, so that other frames need to rely on them for calculation.

Therefore, we propose Approximate Codes for multimedia data that significantly reduce storage overhead by reducing the parity of data that is not sensitive to errors. In the scenario shown in (Figure 1), the Approximate Codes are designed for systems composed of n disks where m disks are dedicated to coding and another s sectors encoded for the first strip. This allows the data of the first stripe to tolerate any $m+s$ disk corruption, so we specifically store important segments of multimedia data here. With an appropriate data distribution scheme, non-critical data segments will still retain $(n-m-s)/(n-m)$ data when any $m+s$ disks are lost, which makes recovery schemes such as interpolated or superpixel still effective. The approximate code provides two recovery modes, full recovery and approximate recovery. The former applies to no more than m disk corruptions and recovers all data, the latter applies to no more than $m+s$ disk corruptions and retains important data.

ACKNOWLEDGMENT

REFERENCES

- [1] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*. Elsevier, 1977, vol. 16.