Approximate Code: A Cost-Effective Erasure Code for MultimediavApplications in Cloud Storage Systems

Huayi Jin[1]

***Abstract*—Erasure codes are commonly used to ensure data availability in cloud storage systems, where video data produced by autopilot, multimedia industry and security monitoring occupies large amounts of space. Typical erasure codes, such as Reed-Solomon (RS) codes or RAID 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 data, which results in multiple complete parity disks being excessive.**

Therefore, we propose Approximate Codes for video data, which significantly reduce storage overhead and increase availability for more important data segments. 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 schemes, Approximate Codes reduce the storage overhead by 7.64% at the expense of reasonable probability of video quality loss.

***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 parity 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.

Video data consumes massive space in cloud storage systems, and this trend is exacerbated as applications demand increased resolution and frame rates. Using multiple copies to ensure video data security will generate storage cost that are several times larger than the original data, which is obviously too expensive, while erasure codes can significantly reduce this cost.

Existing erasure codes are designed to completely recover corrupted data and use at least 3 additional parity 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 video 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 video 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 video data is usually stored after being encoded to save space, while the encoded video data stream is non-uniformly sensitive to data loss, which makes it inappropriate to provide uniform fault tolerance using conventional erasure codes. With the motion compensation mechanism, common video coding algorithms such as H.264 only needs to store the complete content of key frames and a little part of other frames, which makes other frames rely on the key frames for computation while decoding.

Therefore, we propose Approximate Codes for video 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 $m+s$ disks corruption, so we specifically store important segments of video data there. 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 corrupted, 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 disks corruptions and retains important data.

II. Related Work and Our Motivation

A. Existing Erasure Codes

B. Video Encoding

Video data is compressed using various formats to reduce storage costs. Lossy compression is a common method because it provides a much lower compression ratio than lossless compression while ensuring a tolerable loss of video quality, so we focus on this type of algorithm. Currently, H.264 is one of the most popular advanced algorithms in this type of work. H.264 is widely used on video sites such as YouTube because of its higher compression ratio and lower complexity than its predecessor.

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