Replication and Erasure Coding Explained
Caringo Elastic Content Protection Technical Overview

Paul Carpentier, CTO, Caringo

Introduction: data loss

Unlike what most storage vendors will try to make you believe, data loss in storage systems is unavoidable. It’s right up there with death and taxes in that respect. Luckily enough, by selecting the right solutions and applying the appropriate parameters, we can control to a large extent both the probability of data loss happening, as well as the size of the data loss when it actually does happen. Providing a qualitative, straightforward view into the specifics of this process as related to the Caringo object storage software, specifically CAStor, is the purpose of this paper.

Elastic Content Protection (ECP): the scope

As always, if you are trying to protect something, you need to know the threats or exploits that you are expected to protect against: technical failure, human error or malicious intent. Even though the CAStor feature set offers specific ways to deal with the latter two categories, ECP primarily deals with the first. In other words, how do we make sure that a stream of bytes stored in a CAStor Object Storage Cluster can be retrieved intact, with a probability that meets or exceeds the Service Level Agreement (SLA) required.

In the course of IT history, many schemes have been devised and deployed to protect data against storage system failure, especially disk drive hardware. These protection mechanisms have nearly always been variants on two themes: duplication of files or objects (backup, archiving, synchronization, remote replication come to mind); or parity-based schemes at disk level (RAID) or at object level (erasure coding, often also referred to as Reed-Solomon coding). Regardless of implementation details, the latter always consists of the computation and storage of “parity” information over a number of data entities (whether disks, blocks or objects); when an entity fails later on, the lost information can be reconstructed by means of the remaining entities plus the parity information. Many different parity schemes exist, offering a wide range of protection trade-offs between capacity overhead and protection level - hence their interest.
As of late, erasure coding has received a lot of attention in the object storage field as a one-size-fits-all approach to content protection. As we will see below, this is quite a stretch. Erasure coding is a really interesting approach to storage footprint reduction for an interesting but bounded field of use cases, involving BOTH large streams AND large clusters. The Caringo development team wasn’t ready to sacrifice the numerous use cases that involve small streams, small clusters, or a combination of the two, so we implemented BOTH replication AND erasure coding in the same architecture, with flexible choices and automatic, adaptive transitions between either protection form. We call it Elastic Content Protection or ECP, and we are the only storage software developer in the market offering this level of flexibility or anything even remotely equivalent in function.

**Replication**

The simplest form of protective data redundancy is replication, with one or more additional copies of an “original” object being created and maintained to be available if that original somehow gets lost. In spite of the recent hype around erasure coding, we will see that there still are substantial use case areas where replication clearly is the superior option.

**How does the replication function of the CAStor object store work?**

For the sake of the example, imagine a cluster of a 100 CPUs with one disk drive each, and 50 Million objects with 2 replicas each, 100 Million objects grand total. When we speak of replicas in this context, we mean an instance - any instance - of an object; there is no notion of “original” or “copy”. 2 replicas equal a grand total of 2 instances of a given object, somewhere in the cluster, on 2 randomly different nodes. That latter statement entails that, when a disk drive fails, the replicas of the lost instances are randomly scattered over the cluster, but pretty much evenly spread across all the remaining nodes.

**When a drive volume fails, what happens?**

First of all there is a mechanism that nearly instantly detects drive or node failure based on observations by the other nodes. If a 1 Terabyte volume was lost, and there are 100 nodes in the cluster, then the replicas were spread over 99 nodes, each holding a bit more than 10 Gigabyte (1TB/99) worth of objects. After a settable “grace period”, recovery starts, i.e. the 99 nodes will start to work on re-replicating the siblings they hold of objects that were present on the now-lost volume. Since everything happens in parallel and each CPU has only 10GB to replicate, the recovery cycle will be over in a matter of minutes.
When does data loss occur in a replicated object store with two replicas per object?

The short answer is: when two or more drives fail simultaneously. The correct answer is: when a drive volume fails before the volume recovery cycle of a previous failure is complete, there will be at least some data loss. So it is clear that you want to avoid overlapping recovery cycles by optimizing program code to make recovery happen in the shortest possible amount of time. Performance of the code equals robustness of the storage solution. That is why we at Caringo have spent an extraordinary amount of creative energy over 6 versions of our software in refining the process we call Fast Volume Recovery. Especially when a cluster is filled up with small objects, this is a non-trivial optimization task, yet it is vital for the robustness of the cluster, or in other words the SLA it can offer against data loss.

When does data loss occur with three replicas per object?

The above can be extrapolated to provide guidance about more than two replicas: if there are three replicas per object, three overlapping recovery cycles - a very low probability event - will be required to cause any data loss. In CAStor, the desired replication level for objects can be controlled in a very fine-grained fashion: down to the object level, if required, and even automatically change over time. The parameter to drive this functionality is coded in the metadata of the object and is called a Lifepoint. It can express the specification that an object be stored in three replicas for the first year of its life, then automatically scales back to two replicas after that period. This means that the storage footprint - thus expense - of a given object can faithfully track its business value.

Replication is part of a balanced IT strategy.

The storage industry has always been under economic pressure from its customer base to reduce storage infrastructure costs, which is a major bullet on any IT budget. A logical way to do so is by trying to reduce disk footprint by all means: compression, deduplication, RAID come to mind; these are all mechanisms to reduce redundancy. However, protective redundancy (backup, remote replication for business continuity) or performance related redundancy (caching) should not be adversely affected; the interplay - or interference - between all these potentially conflicting mechanisms is not always well understood, especially when multiple tiers of storage are involved - sometimes from different vendors. That is why a single tier storage model that can balance all requirements in a single management model is by far superior, both in reliability and in economic performance.
The core of this vision has been present in CAStor since its production release in 2006 and refinements have been added over the years. Recently, with CAStor 6, an important new technology chapter has been added to this single tier content protection story, warranting a fresh name - Elastic Content Protection - to label CAStor's overall capability in the matter, which we sincerely believe is currently industry-leading by a solid margin.

Erasure Coding

Most readers will be familiar with the concept of RAID content protection on hard disk drives. For example, the contents of a set of 5 drives is used to compute the contents of what is called a parity drive adding one more drive to the RAID set for a total of 6 drives. Of the total set of 6, if any single drive fails, the content that is lost can be rebuilt from the 5 remaining drives. Aside such a 5+1 scheme, many others are possible, where even multiple drives can fail simultaneously and yet the full content can be rebuilt: there is a continuum in the trade-off between footprint and robustness.

Erasure coding is similar to RAID on a per object level.

More recently, the same class of algorithms that is used for RAID has been applied to the world of object storage: they are commonly called Erasure Codes (http://en.wikipedia.org/wiki/Erasure_code). The concept is similar: imagine an object to be stored in a CAStor cluster. Now, rather than storing and replicating it wholly we will cut the incoming stream into (say) 5 segments and generate 1 parity segment, then store all 6 resulting segments. Similarly to the RAID mechanism above, any missing segment out of the 6 can be rebuilt from the 5 remaining ones. This provides a mechanism to survive a failed disk drive without making a full replica: the footprint overhead is just 20% here rather than 100%!

Beyond this “5+1” scheme, many more Erasure Coding (EC) schemes are thinkable. They can survive as many disk failures as their number of parity segments: an 8+3 scheme can survive 3 simultaneous disk failures without data loss, for instance.

Erasure coding comes with trade-offs.

The underlying objective is clear: provide protection against disk failure at lower footprint cost. However, as usual, there is no such thing as a free lunch. There are trade-offs to be considered when compared to replication, which may or may not be acceptable depending on the use case:
Trade-off #1: Larger data losses than replication when the # of failed disk drives grows.

Refer to figure 1 below. It represents the amount of data that will be lost when 1, 2, 3, 4,... drives fail. Both replication with two replicas (1:1) and erasure coding with 5 data and 1 parity segments (5:1) can sustain a single disk drive failure while keeping losses at 0. But beyond that, data loss grows much more quickly (blue curve) than for replication (red curve). One way to address that issue is to provide more parity segments. Let’s take a 5:3 scheme as shown in figure 2. The disk footprint overhead is now 60% rather than 20%, but we can now sustain 3 simultaneous drive failures with data losses remaining better than replication for anything less than 7 drive failures.
Another parameter that influences data loss amount is the cluster size. If we double the cluster size from 10 to 20, we get figure 3, which looks more interesting at first sight. However, we shouldn’t forget that when doubling the number of disks like we do, we also double the statistical probability of disk failure...

![Figure 3](image)

**Trade-off #2: Erasure coding is only really applicable in larger clusters**

It is clear that when we use an erasure code scheme like 10+6, we need at least 16 nodes to distribute all the segments making up the erasure set. But the real protection characteristics surface only at multiples of that minimum. In Figure 4 we compare the relative data losses incurred by a 5+2 scheme in clusters of 10 nodes (blue curve) and 30 nodes (red curve).

![Figure 4](image)
**Trade-off #3: Erasure coding is only really applicable for larger objects**

When chopping up objects to store the resulting segments across a bunch of nodes, the “physical” object count of the underlying storage system is multiplied (e.g., for a 10:6 scheme, it’s multiplied by 16). Not all competing object storage systems handle high object count with the same grace as CAStor, which only needs enough RAM to accommodate enough index slots for each of the physical erasure coded segments to be stored. It is also clear that the granularity (i.e., minimum file size) of the underlying file or object storage system will play a role in suggesting how small an object can be to be economically stored using erasure coding. Even as CAStor is more suited for the storage of very small streams than other architectures, it doesn’t really make sense from an efficiency perspective to store, say, a 50 K object using a 10:6 erasure coding scheme.

**Durability: the common risk denominator**

The above calculations all deal with the potential size of a data loss as a function of its chosen protection scheme. It is clear however that if we want a generic expression of your exposure to data loss, the probability of such a data loss event will also have to enter the picture. It is also clear - all other things being equal - that both size and probability of data loss will be greater in larger clusters while ideally, we’d like to have a metric that mostly factors out that cluster size.

**How to calculate data durability as a number of 9s.**

That common denominator is derived as follows: the vertical axis in all the above charts shows the Fraction Lost, the (very small) percentage of the data that is lost when a data loss occurs. Without getting into computational detail, let us bring in data loss probability (based on things like disk drive MTTF and cluster recovery speed) and calculate the data fraction we would statistically lose on a yearly basis - if our calculations tell us a loss will happen only once every 17 years, then we divide expected fractional loss size by 17 to arrive at the YearlyExpectedFractionLost. This will be a very small number, typically something like 0.000000000001 or so. If we look at it from a positive angle, that means that 0.9999999999999 - or 99.99999999999% - of our data will be unharmed. A simpler way to express that would be “13 nines” of durability. For the mathematically inclined, that is -log(YearlyExpectedFractionLost).

Durability is quickly becoming the industry standard way to express the quality of an object cloud in terms of data protection over time. Different hardware/software options and parameters will influence the number of nines of durability. CAStor with ECP has a choice unmatched in the industry to tailor the durability of data stored in its clusters,
down to the object level, to a desired storage SLA. In this way, many different applications talking to a single CAStor cluster can all have their own SLA, or even a multiple of those: a consumer web service may offer different SLAs for its demo or premium accounts, for instance.

**ECP Durability Calculator as a practical tool**

At Caringo we have developed a very straightforward and simple to use ECP Durability Calculator to help with cluster capacity planning. It is based on a statistical function library running behind the scenes to compute durability as a function of a number of hardware parameters and chosen erasure coding and replication settings.

In fig.5 we see a first (edge) case of erasure coding: 1 data segment and 1 parity segment, which is actually replication with 2 replicas. Yellow parameters in rows 2 to 5 will be obvious. MTTF in row 6 stands for Mean Time To Failure of the disk drives used; the value used here is conservative at about ⅓ of what disk drive manufacturers specify, but more in line with scientific findings in published “Failure Trends in a Large Disk Drive Population” (by Google et al). The rate of repair is the speed with which a cluster will actively “heal” after losing a drive: this happens by re-replicating the lost objects (in case of replication) or by re-computing the lost segments (in case of erasure coding); the value used here is again on the low side, as one might see in a cluster with small objects; large objects will easily triple that recovery rate, hence raise the durability - all other things remaining equal. The results can be found in the green zone: storage efficiency is at 50% - pretty obvious for 2 replicas, and the durability is 8 nines.

![](image.png)  
**Figure 5**
In figure 6 we try a 5+2 Erasure Coding scheme to compare:

![Figure 6](image)

If the use case supports it, we are doing clearly better, both in terms of storage efficiency and in durability, even if our rate of repair has likely dropped (guesstimate of 2 MB/sec/node) as rebuilding the erasure coded set is more involved than re-replicating an object. We continue along the same lines for figure 7, while figure 8 shows a highly redundant 9+6 scheme that is suitable for spreading across three remote data centers (as reflected by the very low rate of repair) - the obtained durability is remarkable.

![Figure 7](image)

![Figure 8](image)
Conclusion

As so often in engineering, there is no single perfect solution to a wide array of use cases. In object storage applications, cluster sizes run the gamut between just a few nodes built into a medical imaging modality to thousands of nodes spanning multiple data centers, with object sizes ranging between just a few K for an email message and hundreds of GB for seismic activity data sets. If we want to fulfill the economic and manageability promises of the single unified object storage cloud, we need technology that is fully capable of seamlessly adapting between those use cases.

While erasure coding technology offers interesting new ways of saving raw disk footprint, its field of applicability is too narrow to serve as the single method of protection for general purpose object storage infrastructure for cloud storage or long-term retention of variable size unstructured content. General-purpose object storage needs to be capable of serving a wide range of use cases within a wide range of durability SLAs and associated price tags. To build that infrastructure, Caringo natively implements both replication and erasure coding into CAStor through Elastic Content Protection. Elastic Content Protection is enabled by parameter metadata (choice of mechanism, # of replica’s, # of parity segments) that is variable between individual objects over time and geographies (replicating clusters as part of the cluster parameters) as part of a programmed and automatically enforced life cycle. All this was not built as an exotic engineering stunt, but because without those capabilities one simply cannot speak of universally applicable object storage that delivers on our promise of truly changing the economics of storage.

Learn more
If you would like a Caringo representative to walk you through the Data Durability Calculator based on your storage efficiency and data durability needs please contact us at http://www.caringo.com/contact.html or email us at info@caringo.com.