

Protecting Big Data – Erasure Coding

November 2015

Dr. Terry Critchley

IT Consultant

Dr. Bill Highleyman

Managing Editor, Availability Digest

Big Data has changed the landscape of data storage. A company's data is always precious, and the loss of any of it can be devastating to the IT functions upon which the company depends. With Big Data being stored on hundreds or even thousands of disks, how does one protect that data from loss? Erasure codes are the answer. 

Until recently, RAID storage satisfied most needs for data protection. RAID storage stripes data across several disks with additional parity information so that should a disk fail, the data the failed disk contained can be reconstructed on the fly from the data and parity information on the surviving disks. RAID 6 even allows two disks to fail without losing any data. A typical RAID system stripes data across five or six disks.

RAID systems still can provide sufficient protection for hundreds of terabytes of storage. However, with the advent of Big Data, the amount of storage required now far exceeds that possible with even the largest RAID systems. In many cases, the amount of Big Data storage required is measured in exabytes – a million times greater than the capabilities of the largest RAID systems. Big Data can require hundreds or even thousands of disks for storage. In a storage system so large, even with the most reliable disks industry has to offer, there almost always will be several disks in failure.

kilobyte	10^3 bytes
Megabyte	10^6 bytes
Gigabyte	10^9 bytes
Terabyte	10^{12} bytes
Petabyte	10^{15} bytes
Exabytes	10^{18} bytes
Zettabyte	10^{21} bytes
Yottabyte	10^{24} bytes
Lottabytes	$>10^{27}$ bytes

Erasure Coding

When a disk fails, it is said to be *erased*. Similar to RAID, erasure coding provides forward error-correcting codes on a set of additional disks. However, the error-recovery capabilities of erasure coding are far more powerful than RAID. An erasure-coded system can be structured so that data recovery can be achieved for any number of disk failures, a capability needed by the large number of disks in a Big Data storage system.

The core technology for erasure codes extends back over five decades. It has been in use in communication systems for that long, but is just now being applied to storage systems.

Referring to Figure 1, the number of disks used to store data is denoted by k . An additional m disks are provided for error-recovery coding. Thus, the total number of disks, n , is $n = m + k$. A measure of the

redundancy of the system is called the encoding rate, r , and is $r = m/n$. That is, r is the proportion of all disks in the system dedicated to redundancy. If $r = 0$, there is no redundancy. In summary,

- k is the number of data disks.
- m is the number of error-recovery disks.
- n is the total number of disks.
- r is the encoding rate m/n .

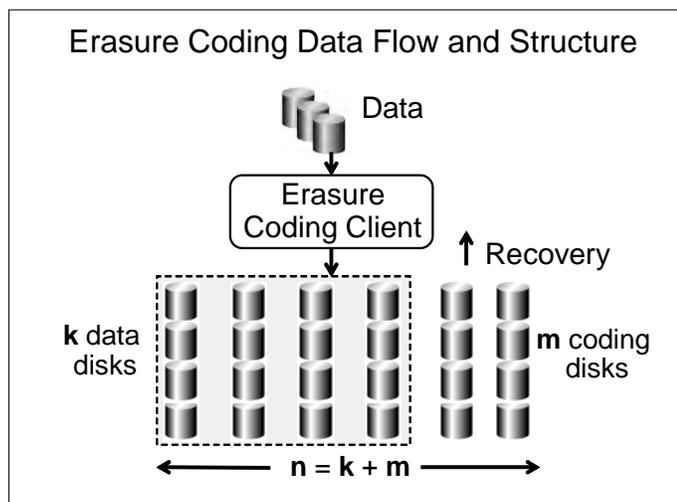


Figure 1: Erasure Coding Data Flow and Disk Configuration

The power of erasure coding is that the data in the storage system is available even if m of the disks in the storage system should fail.

How Does Erasure Coding Work?

The mathematics behind erasure coding is complex. However, as a simple statement, it depends upon a polynomial relationship between the data of all of the disks. If a disk should fail, its data can be reconstructed by solving the polynomial for the missing data using the data of the surviving disks.

A simple example is a parity check. In this case, there is one error disk ($k = 1$). The disk contains the parity check for the set of data disks.

As an example, consider four data disks and a table stored on these disks in which each row has four fields, with one field contained on each data disk. The values for one particular row in this table are 3, 7, 2, 4. We define a parity value as being the negative of the sum of the data values. Thus, the parity value is -16; and the sum of the data and parity values is zero.

Now let us assume that we lose disk 2 (the value of 7). We can reconstruct the value contained on that disk by combining the values on the surviving disks with the parity value:

$$\text{Missing value} = 16 - 3 - 2 - 4 = 7$$

In this case, our polynomial is the one-dimensional relationship:

$$d_0 = -\sum_{i=1}^4 d_i$$

where the data values are $d_1, d_2, d_3,$ and $d_4,$ and the parity value is d_0 .

A more interesting example is found in Wikipedia under “Erasure Code.” Though it is a communications-oriented example, it suffices to show the next step – the use of a two-dimensional polynomial. The example is called “err-mail.”

Err-mail works like e-mail except that about half of all mail gets lost, and messages longer than five characters are illegal.

Alice wants to send her telephone number, 555629, to Bob. Since she can only send up to five characters in each message, she breaks her telephone number into two parts and sends it as 555 followed by 629.

However, she knows that there is a good chance that one or both messages will be lost. So, in concert with Bob, she constructs a two-dimensional linear polynomial of the form:

$$f(i) = a + (b - a)(i - 1)$$

Setting $a = 555$ and $b = 629$, she has for her telephone number

$$f(i) = 555 + 74(i - 1)$$

Thus, $f(1) = 555$ and $f(2) = 629$. We will call these messages A and B.

Alice also computes $f(3) = 703$ (message C), $f(4) = 777$ (message D), and $f(5) = 851$ (message E). She sends all five messages to Bob.

Bob receives messages A, B, and C garbled but receives messages D and E properly. Using the agreed-upon polynomial function, Bob can solve for messages A and B and recover Alice’s telephone number:

$$\begin{aligned} 777 &= a + 3(b - a) \\ 851 &= a + 4(b - a) \end{aligned}$$

Solving this pair of polynomials yields $a = 555$ and $b = 629$. Thus, Bob is able to recreate Alice’s telephone number even though three out of five of the err-mail messages were lost.

Erasure coding extends this use of polynomials to multi-dimensional polynomials that can protect entire disks and recover data even if m disks are lost.

How Powerful is Erasure Coding?

Let us compare erasure coding to a common way to protect data – mirroring. With mirroring (also known as RAID 1), the entire data set is duplicated. Thus, if any one disk is lost, the data is still available on its mirror. In fact, data can be recovered in the event of multiple disk failures so long as both mirrors of a single disk are not lost.

Mirroring

Let us take the case of eight data disks and eight coding disks. In the case of mirroring, the eight coding disks are copies of the eight data disks. The number of ways that one disk can fail in one set of eight disks and the same disk can fail in the other set of eight disks is 8.

Let us assume that the failure probability of a disk is $10^{-3} = .001$. Its availability (the proportion of time that it is up) is then .999, or three 9s (note that the number of nines of availability is equal to the exponent of the failure probability). The probability that two disks will fail is $(10^{-3})^2 = 10^{-6}$. Since there are eight ways that dual disk failures will result in data loss, the probability that data will be lost is $8 \cdot 10^{-6}$:

$$\text{probability of data loss for the mirrored system} = 8 \cdot 10^{-6}$$

Erasure Coding

Now instead of mirroring, let us use erasure coding on eight data disks and eight coding disks. That means that we can lose any eight disks in the storage subsystem and still not lose any data. In order to lose data, we will have to lose nine disks.

The number of ways that k disks out of n disks can fail is given by the relationship for combinations (i.e., how many ways can k items be chosen from n items when the order of the k items doesn't matter). The number of combinations of k items out of n items is

$$\frac{n!}{k!(n-k)!}$$

In our case, we want to know how many ways $m+1$ disks can fail out of n disks. This is what is required for data loss, and is

$$\frac{n!}{(m+1)!(n-m-1)!}$$

In our case example, $m = 8$ and $n = 16$. Thus, the number of ways that nine systems out of sixteen can fail is

$$\frac{16!}{9! * 7!} = 11,440$$

The probability that nine systems will fail is $(.001)^9 = 10^{-27}$. There are 11,440 ways in which nine systems out of sixteen can fail. Thus, the probability of data loss for the erasure coded system is $11,440 * 10^{-27} = 1.144 * 10^{-23}$:

$$\text{probability of data loss for the example erasure coded system} = 1.144 * 10^{-23}$$

The general form for this relationship is

$$\text{Probability of data loss for an erasure coded system} = \frac{n!}{(m+1)!(n-m-1)!} f^{m+1}$$

where

n is the total number of disks in the storage subsystem

m is the number of encoding disks

f is the probability of failure of a disk

Comparison of Mirroring and Erasure Coding

Based on the exponent of the failure probability, mirroring provides about six 9s of availability for this example. Erasure coding provides about twenty-three 9s of availability. Erasure coding is 10^{17} times more reliable than mirroring (100 quadrillion times more reliable)!

Substantial reliability using erasure coding can be achieved in this case with many fewer coding disks. For example, using only two coding disks instead of eight for a total of ten disks yields an availability of about nine 9s, a thousand times more reliable than mirroring with sixteen disks.

Further References

An in-depth discussion of erasure codes and the mathematics behind them can be found in Dr. Terry Critchley's excellent book on high availability, "High Availability IT Services." See the reference in the Acknowledgements section below.

A major researcher in erasure codes is James Plank, a professor in the University of Tennessee's Electrical Engineering and Computer Sciences Department. Papers presented by him can be found at <http://web.eecs.utk.edu/~plank/plank/papers/FAST-2013-Tutorial.html>.

Panasas (www.panasas.com) is a company specializing in very large storage systems. It promotes the use of erasure coding and notes that, with erasure coding, reliability can actually increase with scale. Material from the blog of Geoffrey Noer, Vice President of Product Management at Panasas, was used in part for this article. His blog reference is given in the Acknowledgements section below.

Summary

Big Data presents a significant challenge for storage vendors. Data at the exabyte level may be spread over hundreds or thousands of disks. At any point in time, many of these disks are liable to be out of service. How does a company maintain access to its Big Data in the presence of continual multiple disk failures?

The answer is erasure coding. By adding a set of coding disks that can be used to reconstruct data from downed disks, extremely high reliabilities can be obtained at a moderate cost. Erasure coding provides reliabilities that are orders of magnitude greater than more traditional methods such as mirroring or RAID.

Acknowledgements

Information for this article was taken from the following sources:

High Availability IT Services, Dr. Terry Critchley, *CRC Press*; 2015.

Erasure Codes for Storage Systems: A Brief Primer, James S. Plank, *Usenix*; December 2013.

https://www.usenix.org/system/files/login/articles/10_plank-online.pdf

The Increasing Need for High Reliability, High Performance Data Storage, Geoffrey Noer, *Panasas Blog*; October 27, 2015.

Erasure Code, Wikipedia.