Simplified combinatorical analysis of burst failures

Serkay Ölmez

Abstract

Calculating data loss probability under bursts of failures.

I. The set up

Consider N drives such that $N = n_\mathrm{i} \times n_\mathrm{o}$, where the subscript o is the short for *outer* and i is the short for *inner*. Take n_i drives and apply an erasure coding with $n_\mathrm{i} = d_\mathrm{i} + p_\mathrm{i}$, where d_i stands for data drives and p_i stands for parity drives. This inner layer of erasure coding in this group of drives is capable of recovering from p_i simultaneous failures. Note that the number of such groups is n_o. We can now that this already erasure coded groups and create another erasure coding on top with $n_\mathrm{o} = d_\mathrm{o} + p_\mathrm{o}$. With this set up, we would like to study the resiliency of the data against failure bursts, i.e., will the data survive if N_f drives fail all at the same time?

It will take at least $p_\mathrm{i} + 1$ simultaneous failures for the inner layer to lose data. And we have to lose $p_\mathrm{o} + 1$ inner layers for the outer layer to lose data. Therefore the minimum number of failures that can cause data loss is:

$$n_\mathrm{min} = (p_\mathrm{i} + 1) \times (p_\mathrm{o} + 1) \tag{1}$$

II. Counting combinations

Consider the first of the erasure coded inner layer group that consists of n_i drives, and assume there are f_0 failed drives in this group where $0 \le f_0 \le n_\mathrm{i}$. The total number of creating a configuration with f_0 failed and $n_\mathrm{i} - f_0$ healthy drives is:

$$\mathcal{C}(f_0) = \frac{n_\mathrm{i}!}{(n_\mathrm{i} - f_0)!f_0!}. \tag{2}$$

We need to repeat this for all inner groups and multiply them together to get the total number of combinations

$$\mathcal{C}(f_0, f_1, \cdots, f_{n_\mathrm{o}-1}) \quad = \quad \prod_{k=0}^{n_\mathrm{o}-1} \mathcal{C}(f_k) = \prod_{k=0}^{n_\mathrm{o}-1} \frac{n_\mathrm{i}!}{(n_\mathrm{i} - f_k)!f_k!} = (n_\mathrm{i}!)^{n_\mathrm{o}} \prod_{k=0}^{n_\mathrm{o}-1} \frac{1}{(n_\mathrm{i} - f_k)!f_k!}. \tag{3}$$

Equation (3) is the total number of combinations for an arbitrary collection of failures per inner layer, i.e., $\{f_0, f_1, \cdots, f_{n_\mathrm{o}-1}\}$ with the only constraint being $0 \le f_k \le n_\mathrm{i}$. We want to consider cases where the total number of failed drives is a fixed number by imposing the following condition on f_k:

$$\sum_{k=0}^{n_\mathrm{o}-1} f_k = N_\mathrm{f}. \tag{4}$$

We can now sum over every possible value of f_k satisfying Eq. (4) to get the total number of combinations with total N_f failed drives.

$$\begin{aligned} \mathcal{C} \quad &= \quad \sum_{f_0} \sum_{f_1} \cdots \sum_{f_{n_\mathrm{o}-1}} \mathcal{C}(f_0, f_1, \cdots, f_{n_\mathrm{o}-1}) \delta \left(\sum_{l=0}^{n_\mathrm{o}-1} f_l - N_\mathrm{f} \right) \\ &= \quad (n_\mathrm{i}!)^{n_\mathrm{o}} \sum_{f_0} \sum_{f_1} \cdots \sum_{f_{n_\mathrm{o}-1}} \prod_{k=0}^{n_\mathrm{o}-1} \frac{1}{(n_\mathrm{i} - f_k)!f_k!} \delta \left(\sum_{l=0}^{n_\mathrm{o}-1} f_l - N_\mathrm{f} \right), \end{aligned} \tag{5}$$

where we imposed the condition in Eq. (4) using the Kronecker delta function, δ.

For computational purposes, we can eliminate the eliminate Kronecker delta function by carefully defining the range of the summation indices, i.e., f_k so that Eq. (4) is satisfied by definition:

$$\mathcal{C} \quad = \quad (n_\mathrm{i}!)^{n_\mathrm{o}} \sum_{f_0=0}^{N_\mathrm{f}} \sum_{f_1=0}^{N_\mathrm{f}-f_0} \cdots \sum_{f_{n_\mathrm{o}-2}=0}^{N_\mathrm{f}-\sum_{l=0}^{n_\mathrm{o}-3} f_l} \prod_{k=0}^{n_\mathrm{o}-1} \left. \frac{1}{(n_\mathrm{i} - f_k)!f_k!} \right|_{f_{n_\mathrm{o}-1}=N_\mathrm{f}-\sum_{l=0}^{n_\mathrm{o}-2} f_l}. \tag{6}$$

email: serkay.olmez@seagate.com

Note that we can use this equation as is although the summation indices f_k may exceed n_i. For those cases we will get 0 from the product term since negative factorials ,$f_k > n_\mathrm{i}$, become infinite. For numeric computations, the upper limit of the summation f_k can be truncated at $\min(N_\mathrm{f} - \sum_{l=0}^{k-1} f_l, n_\mathrm{i})$.

III. Counting data loss instances

Not all of these combinations result in data loss. If $f_k > p_\mathrm{i}$ then the inner layer k has lost data. And if the number of inner layers that lost data is larger than p_o, the overall system has lost data. We can put this in using nested Θ functions:

$$\text{Data Loss} \quad = \quad \Theta\left[\sum_{l=0}^{n_\mathrm{o}-1} \Theta\left[f_l - p_\mathrm{i}\right] - p_\mathrm{o}\right], \tag{7}$$

where $\Theta[m]$ returns 1 for $m \geq 1$ and 0 otherwise.

IV. Probability of data loss

We have been counting the combinations that give data loss cases. We need to normalize that against the total number of combinations with N_f failed and $N - N_\mathrm{f}$ healthy drives which is simply:

$$\mathcal{C}_\mathrm{T} = \frac{N!}{(N - N_\mathrm{f})!N_\mathrm{f}!}. \tag{8}$$

And finally, the probability of losing data becomes:

$$\begin{aligned} \mathcal{P} \quad &= \quad \frac{\mathcal{C}_\mathrm{DL}}{\mathcal{C}_\mathrm{T}} \\ &= \quad \frac{(N - N_\mathrm{f})!N_\mathrm{f}!\,(n_\mathrm{i}!)^{n_\mathrm{o}}}{N!} \sum_{f_0=0}^{N_\mathrm{f}} \sum_{f_1=0}^{N_\mathrm{f}-f_0} \cdots \sum_{f_{n_\mathrm{o}-2}=0}^{N_\mathrm{f}-\sum_{l=0}^{n_\mathrm{o}-3} f_l} \prod_{k=0}^{n_\mathrm{o}-1} \frac{\Theta\left[\sum_{l=0}^{n_\mathrm{o}-1} \Theta\left[f_l - p_\mathrm{i}\right] - p_\mathrm{o}\right]}{(n_\mathrm{i} - f_k)!f_k!}\Bigg|_{f_{n_\mathrm{o}-1} = N_\mathrm{f} - \sum_{l=0}^{n_\mathrm{o}-2} f_l} . \end{aligned} \tag{9}$$

Numerically evaluating Eq. (9) is somewhat convoluted due to the changing number of summations as n_o changes. Below are the result for probability of losing data vs number of failures for selected erasure coding as shown in the title. Two traces (which are on top of each other) show the results from brute force counting and the formula. The code is somewhat hard coded to handle $n_\mathrm{o} = 3$ case.

V. Probability of data loss with fixed number of failed racks

Assume that in addition to fixing the number of failure, we want to fix the number of racks that failed. We can do this easily by modifying Eq. (9) where we multiplied combinations from 1 to n_o. We should truncate this at n_r, i.e., the number of failing racks. All we need to do is to replace n_o with n_r.

Since we are looking at n_r racks, the number of drives we are considering is now $N^r = n_\mathrm{i} \times n_\mathrm{r}$. The normalization factor becomes:

$$\mathcal{C}_\mathrm{T}^r = \frac{N^r!}{(N^r - N_\mathrm{f})!N_\mathrm{f}!}. \tag{10}$$

Equation (10) is almost correct. However it includes the cases where failures don't span n_r racks. For example, for $n_\mathrm{r} = 2$, if all failures fall into a single rack, we need to subtract them out. The number of such configurations is:

$$\mathcal{C}_\mathrm{T}^\mathrm{omit} = 2 \times \frac{n_\mathrm{i}!}{(n_\mathrm{i} - N_\mathrm{f})!N_\mathrm{f}!}. \tag{11}$$

Therefore the correct normalization for $n_\mathrm{r} = 2$ is:

$$\mathcal{C}_\mathrm{T}^r = \frac{N^r!}{(N^r - N_\mathrm{f})!N_\mathrm{f}!} - 2 \times \frac{n_\mathrm{i}!}{(n_\mathrm{i} - N_\mathrm{f})!N_\mathrm{f}!}. \tag{12}$$

We need to extend this to generic n_r case, which is a work in progress. Plugging this in, we get the following:

$$\begin{aligned}\mathcal{P}^r &= \frac{\mathcal{C}_\mathrm{DL}}{\mathcal{C}^r_\mathrm{T}}\\ &= \frac{(n_\mathrm{i}!)^{n_\mathrm{r}}}{\mathcal{C}^r_\mathrm{T}} \sum_{f_0=0}^{N_\mathrm{f}} \sum_{f_1=0}^{N_\mathrm{f}-f_0} \cdots \sum_{f_{n_\mathrm{r}-2}=0}^{N_\mathrm{f}-\sum_{l=0}^{n_\mathrm{r}-3} f_l} \prod_{k=0}^{n_\mathrm{r}-1} \frac{\Theta\left[\sum_{l=0}^{n_\mathrm{r}-1}\Theta\left[f_l-p_\mathrm{i}\right]-p_\mathrm{o}\right]}{(n_\mathrm{i}-f_k)!f_k!}\Bigg|_{f_{n_\mathrm{r}-1}=N_\mathrm{f}-\sum_{l=0}^{n_\mathrm{r}-2} f_l}.\end{aligned} \tag{13}$$

VI. Plots

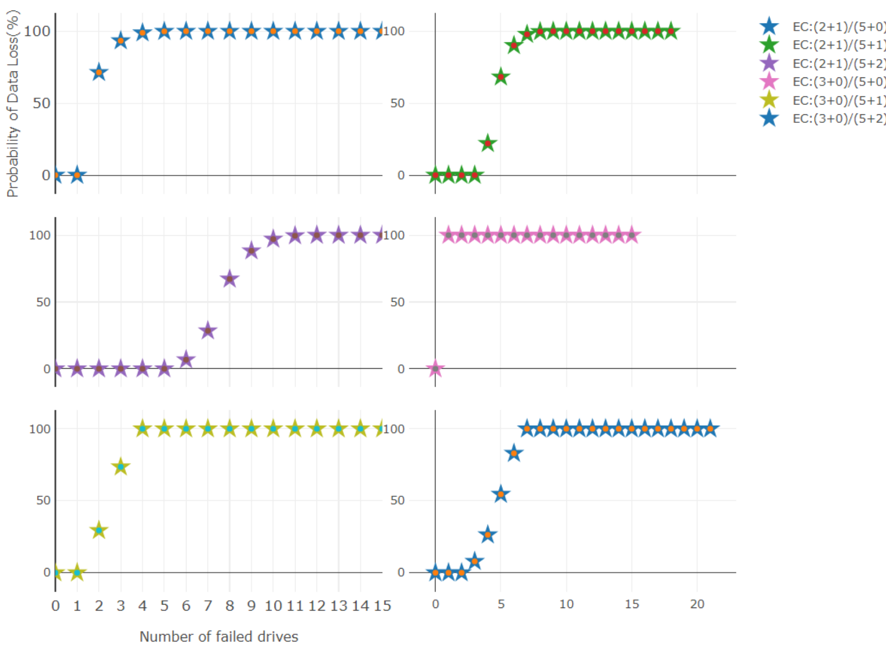

Figure 1: Data loss probability for various ECs. Stars show the results from formula and the dots are from brute force counting.)

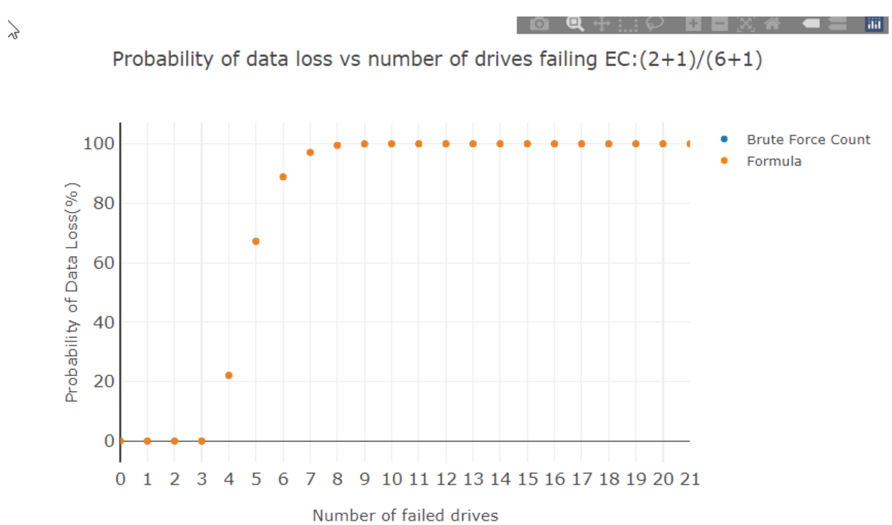

Figure 2: EC:(2+1)/(6+1)

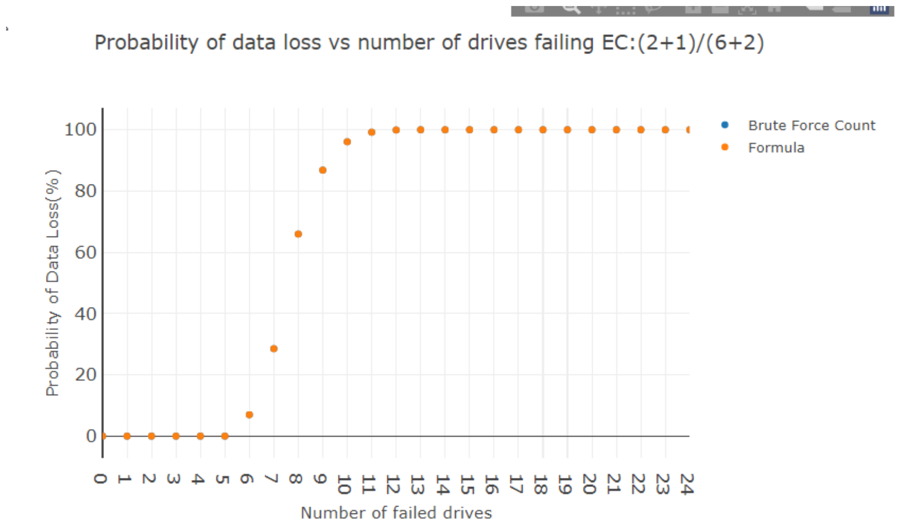

Figure 3: EC:(2+1)/(6+2)

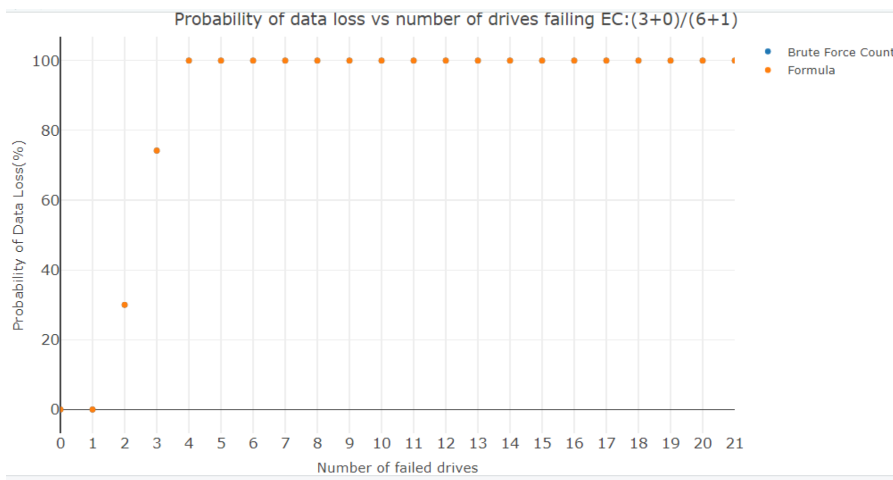

Figure 4: EC:(3+0)/(6+1)

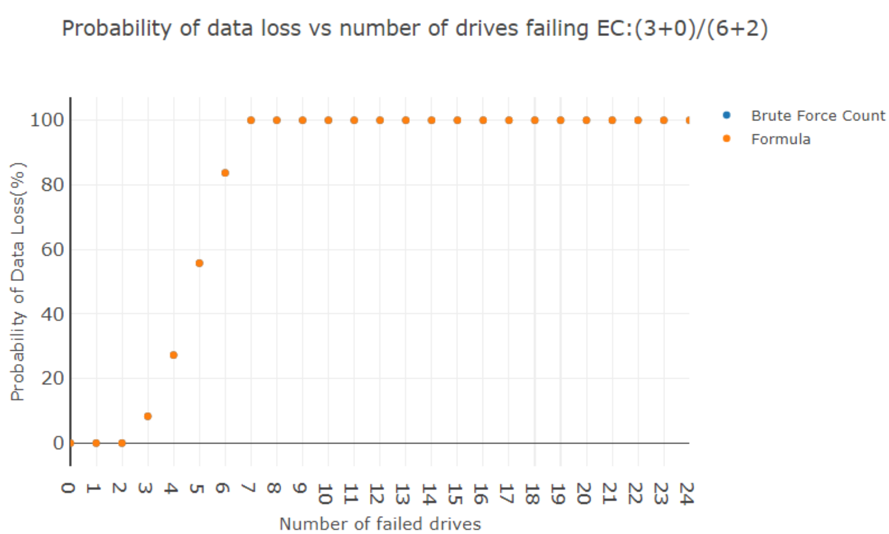

Figure 5: EC:(3+0)/(6+2)