Combinatorical analysis of burst failures for large-scale cluster

Meng Wang

November 15, 2022

1 Setup

We consider a storage system of N drives such that $N = X \cdot Y \cdot Z$, where there are X racks in the system, each rack contains Y enclosures, and each enclosure contains Z drives.

Let $M = Y \cdot Z$, so M denotes the number of drives per rack.

In particular, we are considering ORNL Alpine system, which is composed of 39 racks. 38 racks have 8 enclosures each, and 1 rack has 4 enclosures. Each enclosure has 106 drives.

For simplicity, we assume the system contains 40 racks. Each rack contains 8 enclosures. Each enclosure contains 100 drives.

2 Total instances with fixed number of affected racks

Consider f failures happen in r racks.

We first choose r racks from all the X racks, which has C_X^r combinations.

For each rack combination, without loss of generality, let's assume we picked racks $0, 1, 2, ..., r-1$. We consider all the possible failure_per_rack cases:

$$S(f, r, M) = \{(f_0, f_1, ..., f_{r-1}) | \sum_{\substack{0 \le i \le r-1 \\ 1 \le f_i \le M}} f_i = f\}. \tag{1}$$

For each $(f_0, f_2, ..., f_{r-1})$, there are $\prod_{i=0}^{r-1} C_M^{f_i}$ instances.

Therefore, the total number of instances is:

$$C_X^r \cdot \sum_{(f_0, f_1, ..., f_{r-1}) \in S} \prod_{i=0}^{r-1} C_M^{f_i} \tag{2}$$

One key here is how to get $S(f, r, M)$. Note that we have the following recurrence relation:

$$\begin{aligned} S(f, r, M) &= \{(f_0, f_1, ..., f_{r-1}) | \sum_{\substack{0 \le i \le r-1 \\ 0 \le f_i \le M}} f_i = f\} \\ &= \bigcup_{\substack{0 \le a \le M \\ a \le f-r+1}} \{(f_0, f_1, ..., f_{r-2}, a) | \sum_{\substack{0 \le i \le r-2 \\ 0 \le f_i \le M}} f_i = f - a\} \\ &= \bigcup_{\substack{0 \le a \le M \\ a \le f-r+1}} \{s \cup a | s \in S(f-a, r-1, M)\} \end{aligned} \tag{3}$$

By using formula 3, we can get $S(f, r, M)$ using backtracking algorithm or dynamic programming. Here is an implementation using backtracking algorithm: https://github.com/ucare-uchicago/mlec-sim/blob/main/src/theory/burst_theory.py#L14

3 Survival instances under RAID

Consider $k_l + p_l$ local-only SLEC. For easier deployment we assume $n_l = k_l + p_l$ is divisible by Z.

n_l drives in the same enclosure compose a RAID disk group. Therefore a rack contains $g_l = M/n_l$ RAID groups.

For each $(f_0, f_2, ..., f_{r-1}) \in S(f, r, M)$, rack i contains $fi \geq 1$ failures. We need to compute for each f_i, how many instances can survive the f_i failures in the rack.

Denote $\eta(f_i, g_l)$ as the number of survival instances in a rack when there are f_i failures in a rack containing g_l $k_l + p_l$ RAID groups.

We have the following recurrence relation (which is derived by considering what will happen if disk group 0 contains a failures):

$$\eta(f_i, g_l) = \sum_{\substack{0 \leq a \leq p_l \\ a \leq f_i}} \eta(f_i - a, g_l - 1) \cdot C(n_l, a) \tag{4}$$

We can then compute $\eta(f_i, g_l)$ based on recurrence relation 4 and backtracking algorithm. Here is an implementation: https://github.com/ucare-uchicago/mlec-sim/blob/main/src/theory/burst_theory.py#L69

Therefore, the total survival instances of $(f_0, f_2, ..., f_{r-1})$ is $\prod_{i=0}^{r-1} \eta(f_i, g_l)$.

Therefore, the total number of survival instances in the whole system is:

$$C_X^r \cdot \sum_{(f_0, f_1, ..., f_{r-1}) \in S} \prod_{i=0}^{r-1} \eta(f_i, g_l) \tag{5}$$

Therefore, the probability of data loss under f failures on r racks for RAID is:

$$\begin{aligned} \text{RAID data loss} &= \frac{C_X^r \cdot \sum_{(f_0, f_1, ..., f_{r-1}) \in S} \prod_{i=0}^{r-1} \eta(f_i, g_l)}{C_X^r \cdot \sum_{(f_0, f_1, ..., f_{r-1}) \in S} \prod_{i=0}^{r-1} C_M^{f_i}} \\ &= \frac{\sum_{(f_0, f_1, ..., f_{r-1}) \in S} \prod_{i=0}^{r-1} \eta(f_i, g_l)}{\sum_{(f_0, f_1, ..., f_{r-1}) \in S} \prod_{i=0}^{r-1} C_M^{f_i}} \end{aligned} \tag{6}$$