

Modeling Data Durability and Availability

Serkay Ölmez

serkay.olmez@seagate.com

January 07, 2021

Acknowledgements:

Ian Davies, Mike Barrell, John Bent, and Iman Anvari



SEAGATE



Outline

- Modeling Failures
 - Weibull Distribution
- Erasure Coding(EC)
 - General formalism, RAID 5& 6
- Hard Errors (UREs)
 - Modeling durability with UREs
- Distributed Parity
 - Improving data durability with ADAPT
- Multi-Layer EC
 - Improving durability with two layers of EC
- Availability
 - Modeling data availability
- Appendix
 - MACH2 and ReMan

Goals & Summary

Goals

- Provide a quick review of available models to compute data durability,
- Present an accurate and rigorous model,
- Establish a common language to compute these metrics.

Summary

A quick survey on literature

- The durability and availability of data can be predicted accurately with Markov Chains:
 - Based on rigorous math, and verified with Monte Carlo simulations.
 - Supports Distributed Parity, ReMan, UREs, Weibull failure modes, and multi-layer EC.
 - Developed in collaboration with the CORTX architects & sales team.
- Advanced features, such as Online ReMan, can be modeled too:
 - We continue to work on modeling latest and greatest CORTX features.

Modeling component failures

We will assume that individual component failures can be described by a Weibull distribution [10]. The failure probability density, cumulative failure distributions and the hazard rate (failure rate) are defined as follows:

$$f_{\alpha,\beta}(t) = \frac{\beta}{\alpha} \left(\frac{t}{\alpha} \right)^{\beta-1} e^{-\left(\frac{t}{\alpha}\right)^\beta}, \quad (1)$$

$$F_{\alpha,\beta}(t) = \int_0^t d\tau f_{\alpha,\beta}(\tau) = 1 - e^{-\left(\frac{t}{\alpha}\right)^\beta}, \quad (2)$$

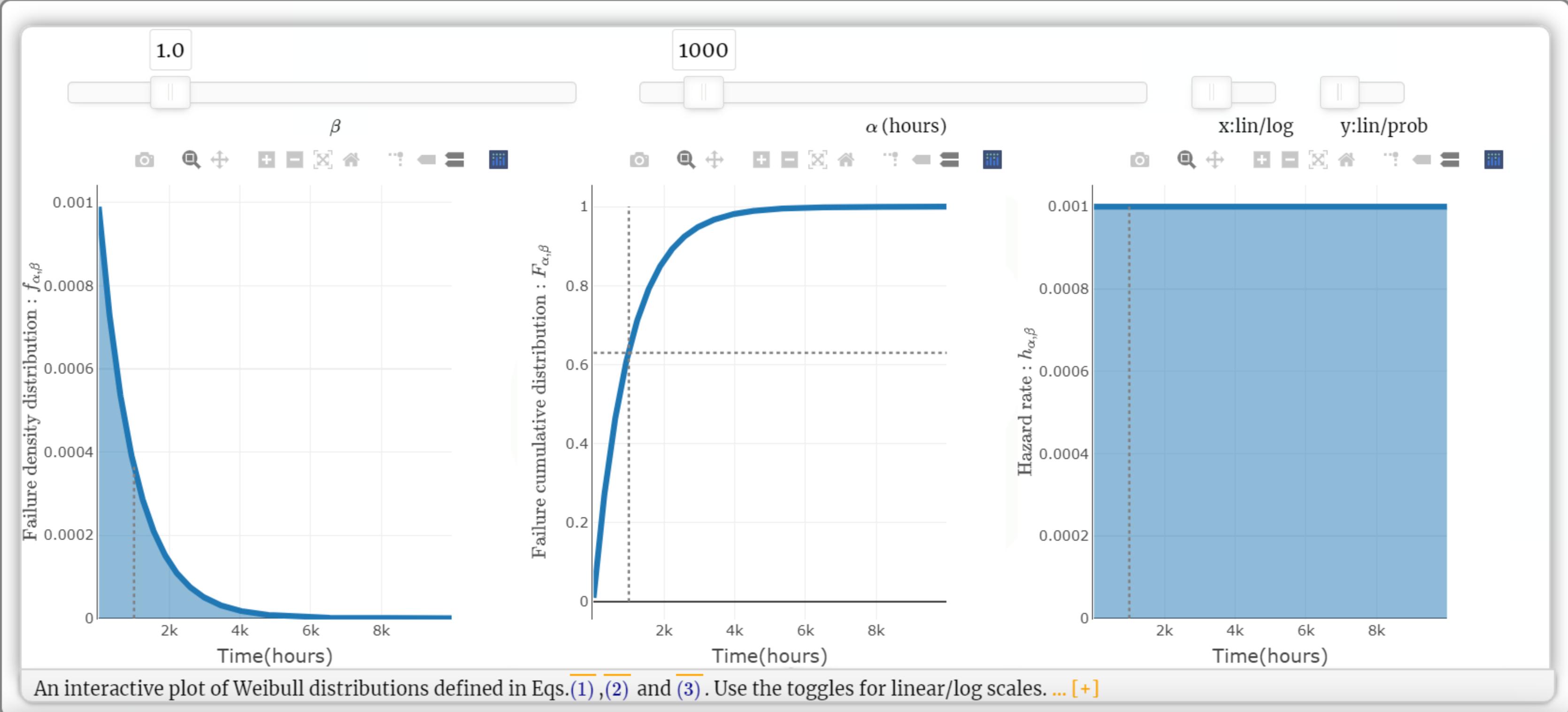
$$h_{\alpha,\beta}(t) = \frac{f_{\alpha,\beta}(t)}{1 - F_{\alpha,\beta}(t)} = \frac{\beta}{\alpha} \left(\frac{t}{\alpha} \right)^{\beta-1}. \quad (3)$$

The actual values of α and β vary from product to product: β is expected to be between 1 and 2.

$\beta = 1$ gives the exponential distribution, which has completely random head failure times with a fixed failure(hazard) rate: $h_{\alpha,\beta}(t) = \frac{1}{\alpha} \equiv \lambda$. This simplifies Eqs. (1), (2) and (3) to:

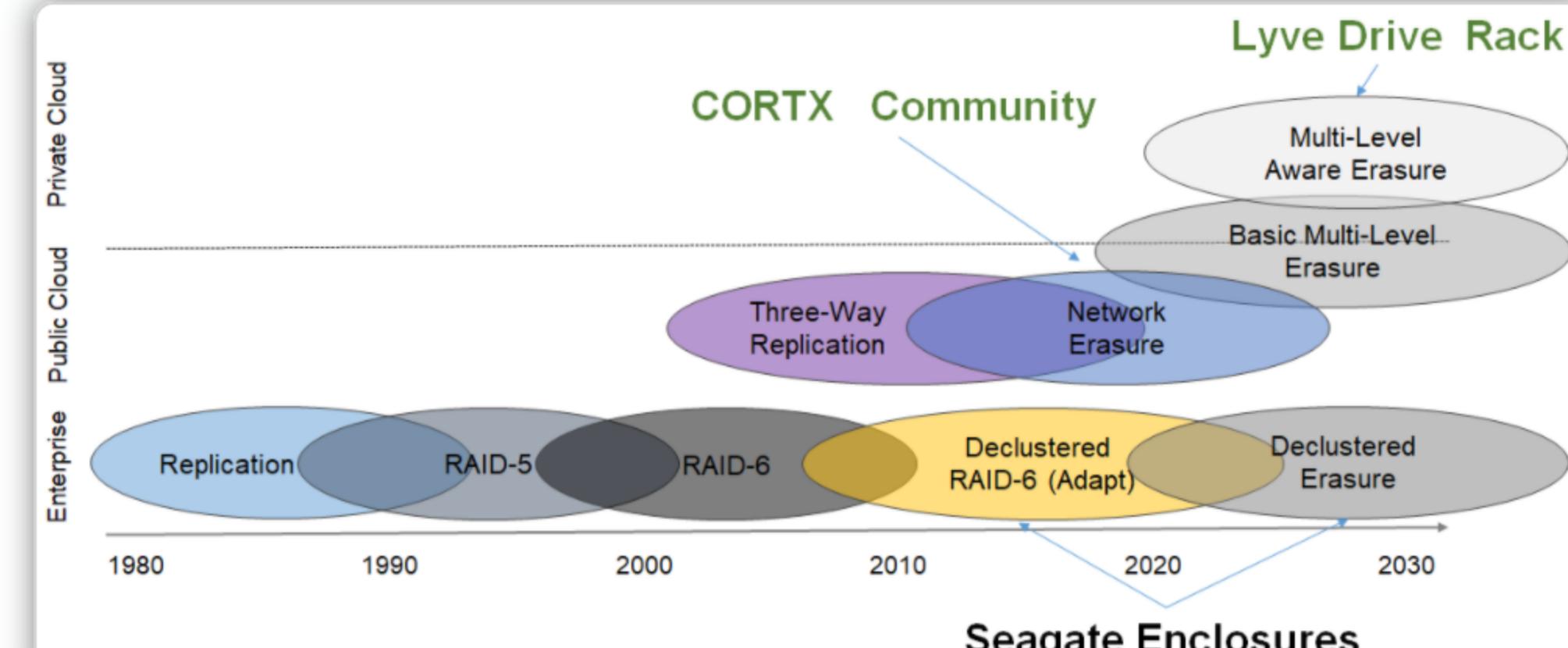
$$f_\lambda(t) = \lambda e^{-\lambda t}, \quad F_\lambda(t) = 1 - e^{-\lambda t}, \text{ and } h_\lambda(t) = \lambda. \quad (4)$$

Visualizing failure distributions



- It is important to understand how the storage devices fail. Weibull is typically a good fit.
- No matter how reliable individual devices are, failures are inevitable.

Erasure Coding



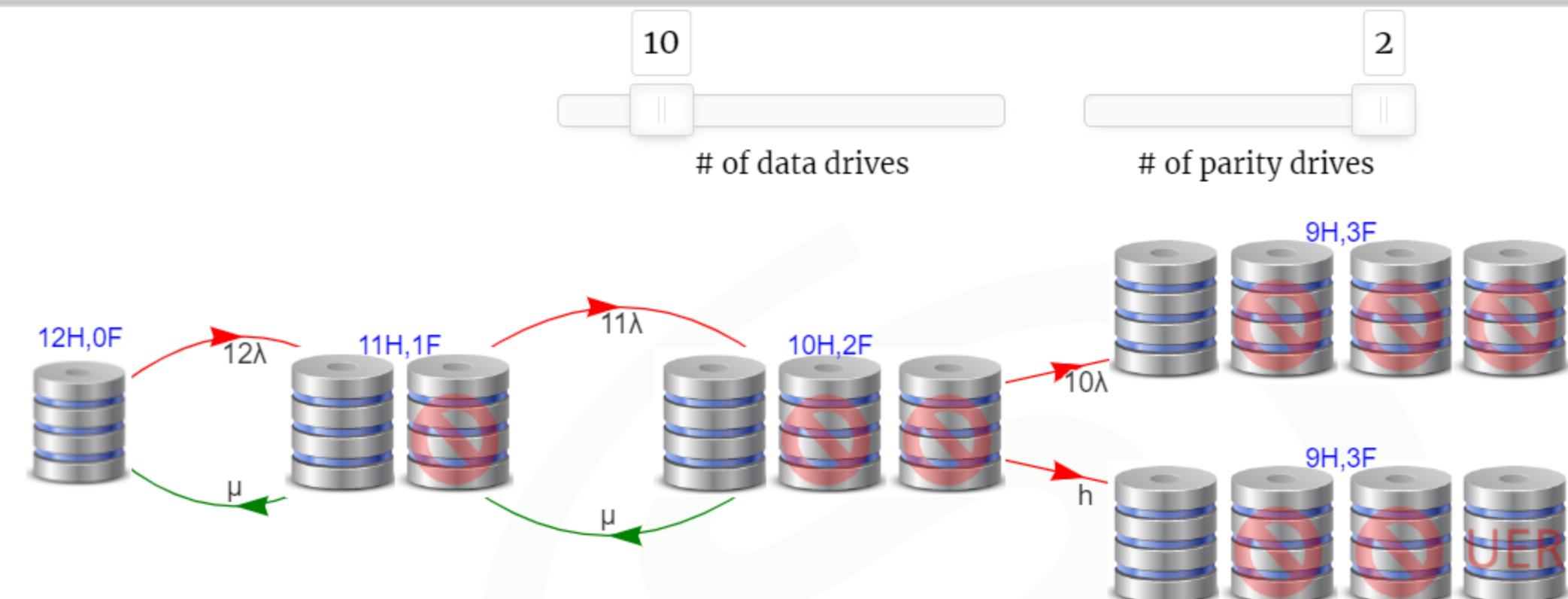
A brief timeline of the erasure coding. ... [+] Credit:John Bent

- The simplest way of creating redundancy is replication, but this has a very poor capacity efficiency.
- RAID 5 and Raid 6 introduce parities to protect data against device failures.
- Seagate enclosures supports declustered RAID6, which can be coupled with a top layer EC in CORTX to get the highest durability with best capacity efficiency.

Creating Redundancy

- EC adds fault-tolerance by creating redundant data pieces.
- Data is distributed across different storage media.
- Simplest example: compute and store the parity data.
 - Given two bytes of data: $B_1 = 01001001$ and $B_2 = 11011010$
 - The parity data: $P = B_1 \oplus B_2 = 10010011$
- Assume the drive that stored B_1 fails:
 - Compute $P \oplus B_2$, which is equal to B_1 .
- The data on the failed drive can be reconstructed.
 - Fault tolerant to one drive failure.
- c pieces of redundancy data:
 - Parities computed using *Reed-Solomon algorithm*.
 - Fault tolerant to c simultaneous drive failures.
- **Main Question:** What is the probability of having $c + 1$ simultaneous failures?

RAID 5 & 6 (with URE)



A Symbolic representation of the RAID6 including URE failures. Use the input sliders to change the number of drives or the redundancy. ... [+]

- The **red** arrows represent drive failures. Rate is scaled with the total number of drives:
 - λ is the failure rate per drive, and $n\lambda$ is the total failure rate for n drives.
 - The arrow denoted with h represents the data loss due to UREs- to be discussed in more detail later.
- The **green** arrows represent repairs. Failed drives are replaced, and data is rebuilt.
 - The repair time, $1/\mu$, depends on capacity and DR. It may be as long as several days.
- Data is lost when the system moves to the right-most state.
 - Key metric: Mean Time to Data Loss (MTTDL): it is a function of n , c , λ , μ , drive capacity and UER.

The simplest model

Consider the following set up:

(n) drives 20, capacity(C): 16 TB, redundancy(c): 3, recovery speed (S): 50 MB/s, and AFR: 1 %.

- At this recovery rate, the recovery time from a drive failure is: $(16 \text{ TB})/(50 \text{ MB/s}) = 3.7 \text{ days}$.
- The probability of losing a single drive in 3.7 days is: $1 \% * (3.7 \text{ days}/365) = 0.01\%$
- The system will lose data when there are 4 failures in 3.7 days, which has the probability: $(0.01\%)^4$.
- Data durability = 1 - Probability of 4 failure(s) in 3.7 days = $1 - (0.01\%)^4 = 15 \text{ nines}$.

Number of nines is defined as the instances of leading 9's in reliability: 0.998 has 2 nines, 0.9991 has 3 nines.

$$\text{number of nines} = \text{Floor}(-\log_{10}(1 - \text{Durability})) \quad (5)$$

- This model is very simple, but... **it is wrong!**
- The metric calculated is **not** the data durability for an 3-redundant EC, it is for 1+3 [original+3 mirror(s)].
 - Note that the total number of drives, 20, did not even enter the equations!
 - There are binomial(20,4)=4845 combinations to choose 4 failure(s) out of 20 drives.
- This is not even the durability for over a year, it is just over 3.7 days. There are 99 such frames in a year.
- With the binomial coefficients and the # frames/year included, this model over-reports durability by at least 5 nines.
- This model also ignores UREs.

The BackBlaze Model

This is a model introduced by [11], [12].

(n) drives 20, capacity(C): 16 TB, redundancy(c): 3, recovery speed (S): 28.5 MB/s, and AFR: 0.4 %.

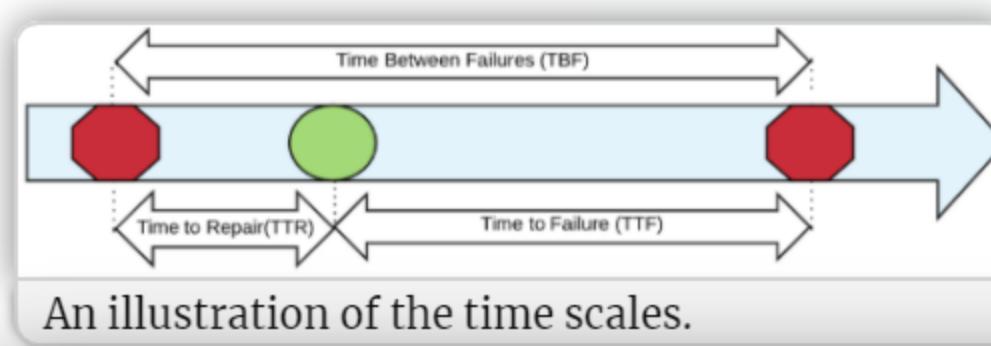
- The recovery time from a drive failure is: $T \equiv \frac{C}{S} = 6.5$ days.
- The AFR value can be converted to the failure rate as: $\lambda = -\frac{\ln(1-\text{AFR})}{\text{year}} \simeq \frac{\text{AFR}}{\text{year}} = 0.004/\text{year}$.
- The probability of a given drive to fail in 6.5 days is: $p_f \equiv \lambda T = 0.01\%$.
- The system will **not** lose data when there are at most 3 failures in 6.5 days.
 - The probability of not losing data in 6.5 days is: $p_{\text{no-loss}} = \sum_{i=0}^c \binom{n}{i} (p_f)^i (1-p_f)^{n-i} = 13 \text{ nines}$.
- There are $N_F = \frac{365}{T} = 56$ such frames in a year.
 - The probability of not losing data in a year is: $(p_{\text{no-loss}})^{N_F} = 11 \text{ nines}$.



An illustration of the frames in the BackBlaze model.

- This model is still simple, but... **it has a few issues:**
 - The segmentation of a year into 56 chunks of 6.5 days implies that data is lost only when 4 failures happen in a given frame. The cases of 4 failures within in 6.5 days but spanning two subsequent frames are missed.
 - It is implicitly assumed that every frame starts with a 3-redundant system. In reality, there may be up to 3 ongoing repairs exiting the previous frame, and a single drive failure early in the next frame will cause data loss.
- Ignoring UREs, BackBlaze model works reasonably well in the low AFR limit (up to a factor of ~ 2).
- Including UREs will reduce the data durability by ~ 2 nines.

The most intuitive model



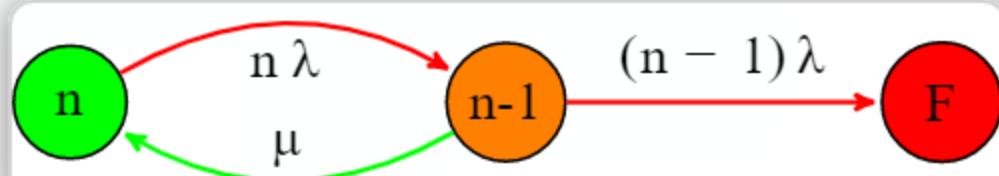
- The plot on the left show the time scales in a repairable system.
- A system of n drives has the Mean Time To Failure: $MTTF = 1/(n\lambda)$.
- A failed drive is replaced, and rebuilt. Mean Time To Repair: $MTTR = 1/\mu$.
- Fraction of the time spent for repair: $\frac{MTTR}{MTTF+MTTR}$

- Fraction of the time spent for repair: $\frac{MTTR}{MTTF+MTTR} \simeq \frac{MTTR}{MTTF} = \frac{n\lambda}{\mu}$ ($MTTF \gg MTTR$).
- There are $N - 1$ drives left running. The rate of failure: $(n - 1)\lambda$
- Multiply this rate with the fraction of time in recovery: $\frac{n\lambda}{\mu}(n - 1)\lambda = \frac{n(n-1)\lambda^2}{\mu}$
 - This is the rate of data loss. Inverting the expression: $MTTDL_1 = \frac{\mu}{n(n-1)\lambda^2}$
- When there are two parity drives, we can iterate to get: $MTTDL_2 = \frac{\mu^2}{n(n-1)(n-2)\lambda^3}$

- For c parity drives, we can recursively iterate to get: $MTTDL_c = \left[\frac{\mu}{\lambda}\right]^c \frac{(n-c-1)!}{\lambda n!}$
- $c = 1$ is a RAID5, and $c = 2$ is a RAID6 setup. $c \geq 3$ cases are refereed to as Erasure Coding in general.
- This model is very intuitive, and works great, but... **it is not clear how to include UREs.**

Durability with rigorous math

- The Markov chain shows a system of n drives with one redundancy.
- Markov chain represents state transitions and state probabilities $p_j(t)$.
- The change in $p_j(t)$ is dictated by incoming and outgoing arrows.



Markov Chain with one redundancy.

Show the details

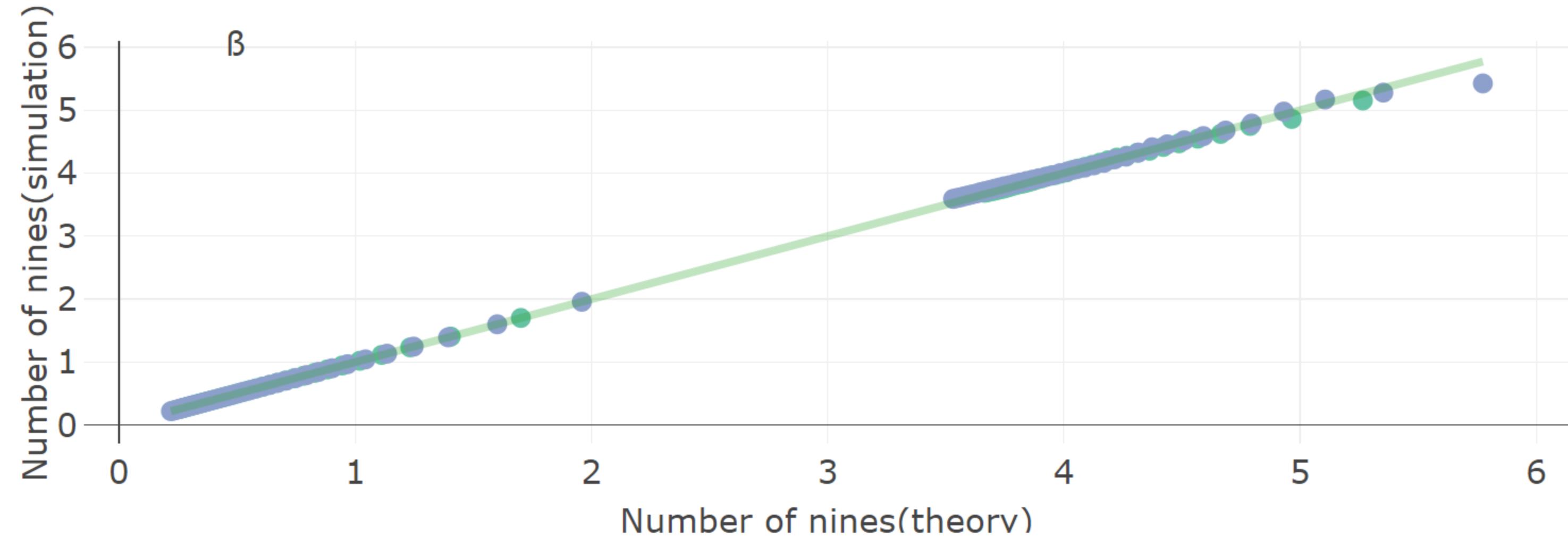
- The state transitions are described by a set of coupled DEs.
- They can be solved by Laplace transforms.
- The equations are solved for the data loss state.
- The reliability: $R(t) = 1 - p_F(t) \sim e^{-t/\text{MTTDL}_1}$
- $\text{MTTDL}_1 = \frac{1}{n(n-1)\lambda} \frac{\mu}{\lambda}$.

$$\begin{aligned}\dot{p}_n + n\lambda p_n - \mu p_{n-1} &= 0 \\ \dot{p}_{n-1} + (n-1)\lambda p_{n-1} + \mu p_{n-1} - n\lambda p_n &= 0 \\ \dot{p}_F - (n-1)\lambda p_{n-1} &= 0\end{aligned}\quad (15)$$

- Markov chain analysis is needed to address complicated cases:
 - Declustered Parity (ADAPT),
 - Re-Manufacturing in the field (ReMan),
 - Generic Weibull distribution for drive failures ($\beta \neq 1$),
 - Latent sector errors, i.e., hard errors (UREs).
- Most of the items above will be addressed in this presentation.

Monte Carlo Simulation

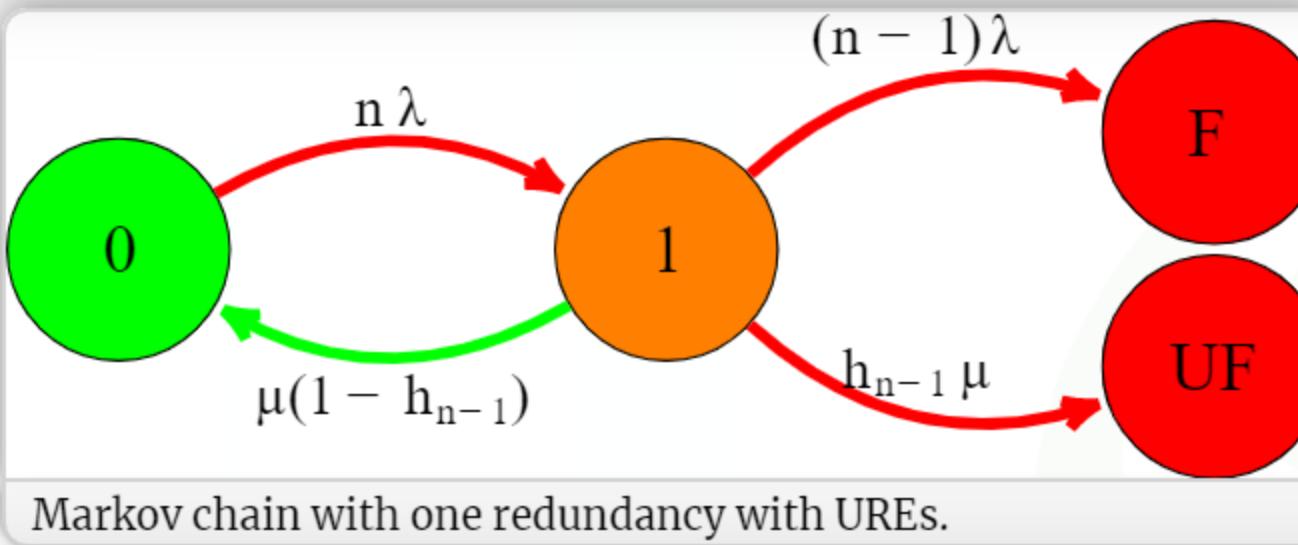
Show the simulation script (JSL)



The comparison of theory vs Monte Carlo results with various input parameters. Number of nines is defined in Eq. (5) where flooring is omitted. ... [+]

Silent data corruption (URE)

- UREs may arise due to thermal decays of bits. They are discovered only when data [is attempted to be] read.
- It is defined as the probability of a corrupted sector **per bits** read. A typical value is 10^{-15} .
- With tens of TBs data, observing at least one corrupted sector is very likely [13].



- Consider n drives with one redundancy.
- The S -matrix reads

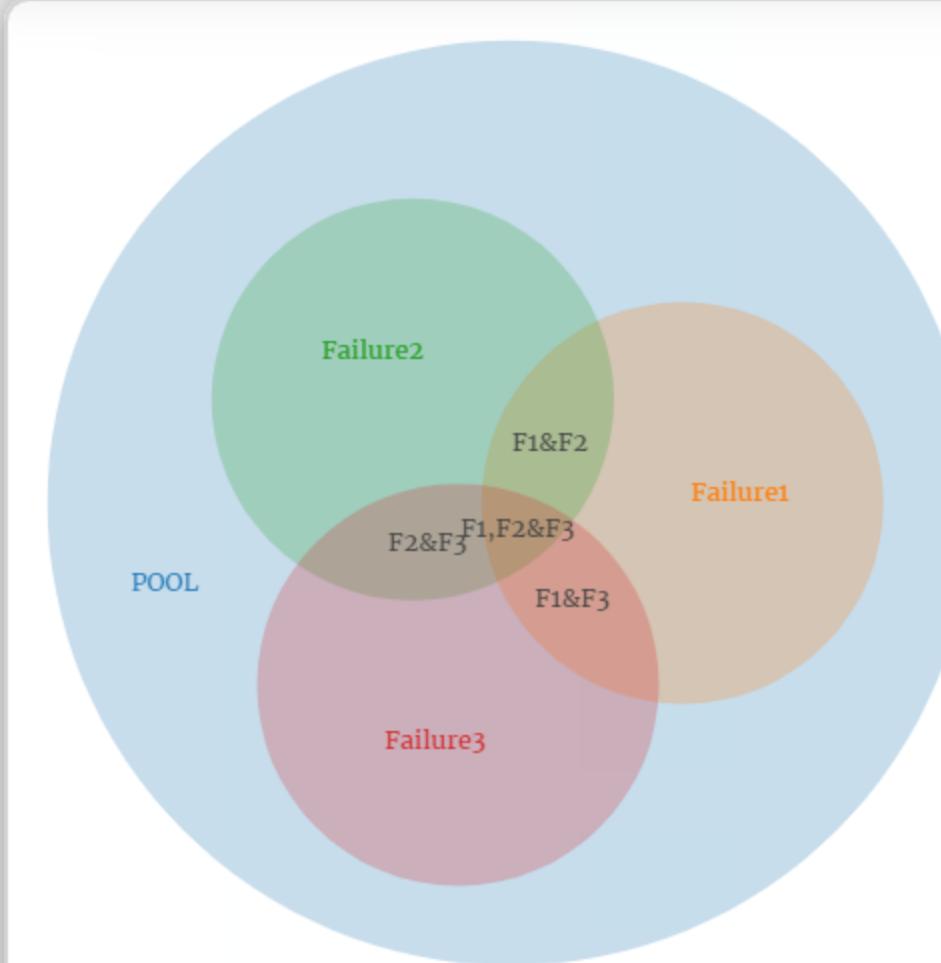
$$\begin{bmatrix} s + n\lambda & -\mu(1 - h_{n-1}) & 0 \\ -n\lambda & s + (n-1)\lambda + \mu & 0 \\ 0 & -(n-1)\lambda - h_{n-1}\mu & s \end{bmatrix}$$

where h_{n-1} : probability of observing UER(s) while reading $n-1$ drives.

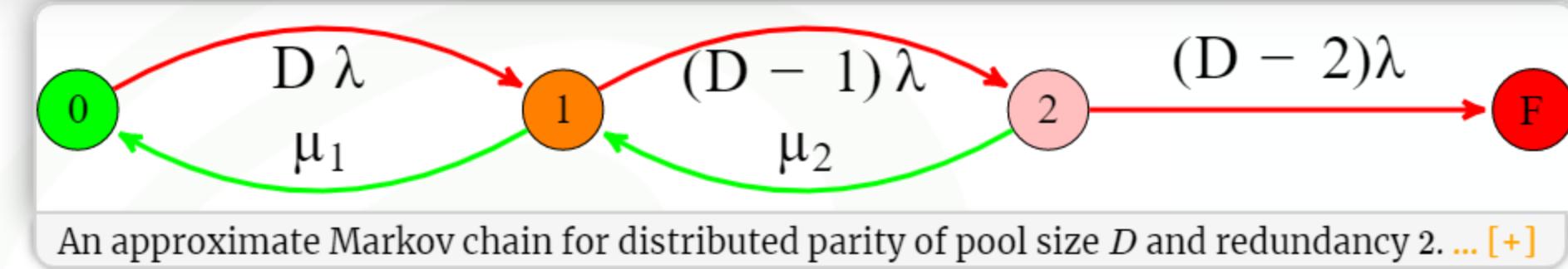
- Finding the poles of the determinant we get: $\frac{1}{\text{MTTDL}} = \frac{\mu}{n\lambda[(n-1)\lambda]} + \frac{1}{1/(n\lambda)} h_{n-1}$
 - This is an harmonic sum of MTTDLs for Drive failure mode and UER failure mode.
- The analysis can be extended to a generic c redundancy: $\frac{1}{\text{MTTDL}_c} = \frac{1}{\text{MTTDL}_{c,\text{DF}}} + \frac{h_{n-1}}{\text{MTTDL}_{c-1,\text{DF}}}$
 - If $h_{n-1} \equiv 1 - (1 - \text{UER})^{\text{bits}} \simeq 1 - e^{-\text{UER} \times \text{bits}}$ is not small enough, the second term is dominant.
 - Ex: $\text{UER} = 10^{-15}$, $n = 20$ and $C = 10\text{TB} \rightarrow h_{n-1} = 1 - (1 - 10^{-15})^{19*8*10^{13}} \simeq 1 - e^{-1.52} = 0.78$.
- Given the large amount of data to be read, the probability of UER is significant.
- Sector level data durability of an EC with c parities is at the order of drive level durability with $c-1$ parities.

Distributed Parity (ADAPT)

- Distributed raid (dRAID) uses all the drives in the pool to store data and parities.
- Rebuild is done by reading from all drives in the pool in parallel.
- ADAPT prioritizes the repair of critically damaged stripes[9]. This is the main reason for reli gain.

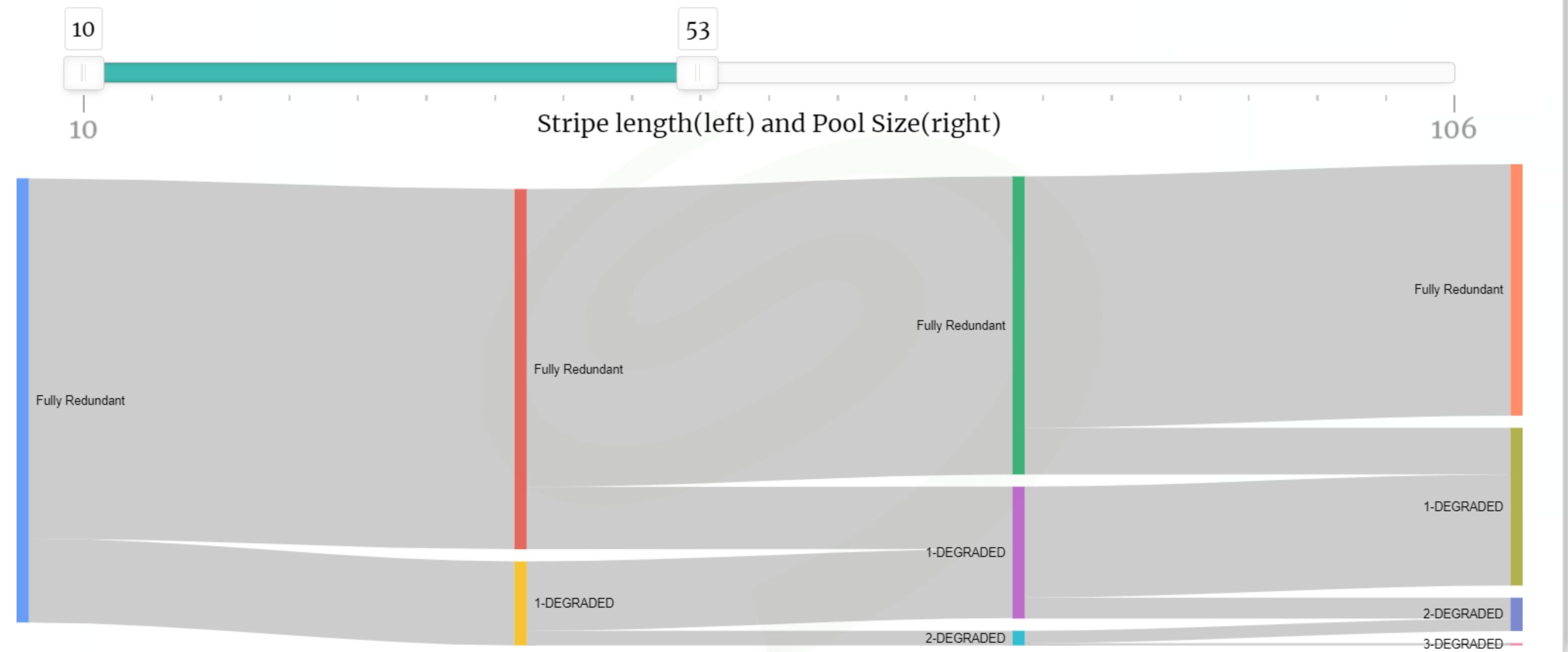


Venn Diagram of overlaps of 3 failures with 53-drive pool and EC size of 10.



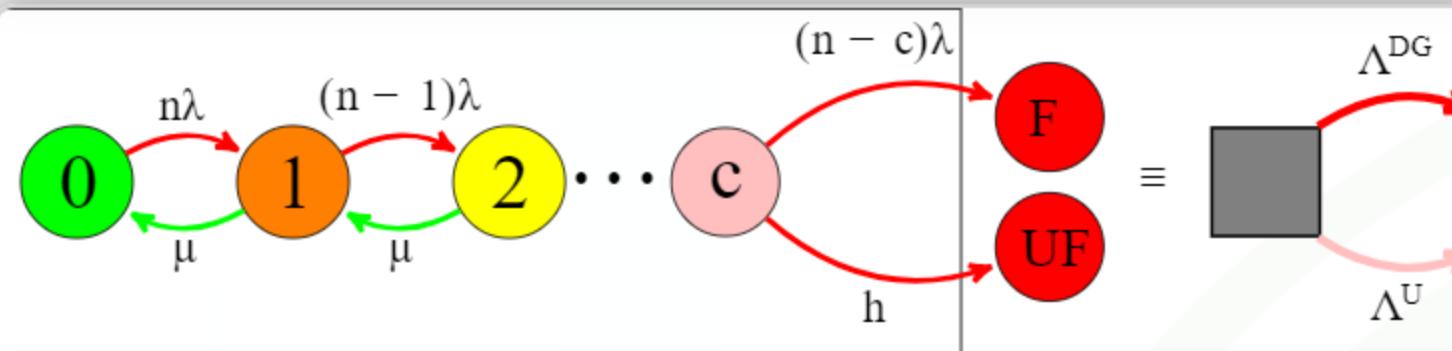
- Consider a system of pool-size D and EC size N , and redundancy c .
 - The overlaps are calculated for $D = 53$ and $N = 8 + 2$.
 - Geometric overlap areas scale with powers of N/D .
- Recovery rates: $\mu_1 \propto D$ and $\mu_2 \propto D^2$, failure rate: $\propto D$.
 - Increase in failure rate cancels with recovery speed up for $c = 1$.
 - Reli will benefit from ADAPT only if $c \geq 2$.
- ADAPT Reli can be expressed in terms of its RAID counterpart:
 - $\text{MTTDL}_{\text{dRAID}} = \left[\frac{D}{N}\right]^{\frac{c(c-1)}{2}} \text{MTTDL}_{\text{RAID}}$.
- For $D = 50$, $N = 10$, and $c = 2$: Adapt reli is 5x better than RAID6 reli.

Visualizing the damage



Modeling Multi-Layer Erasure Coding

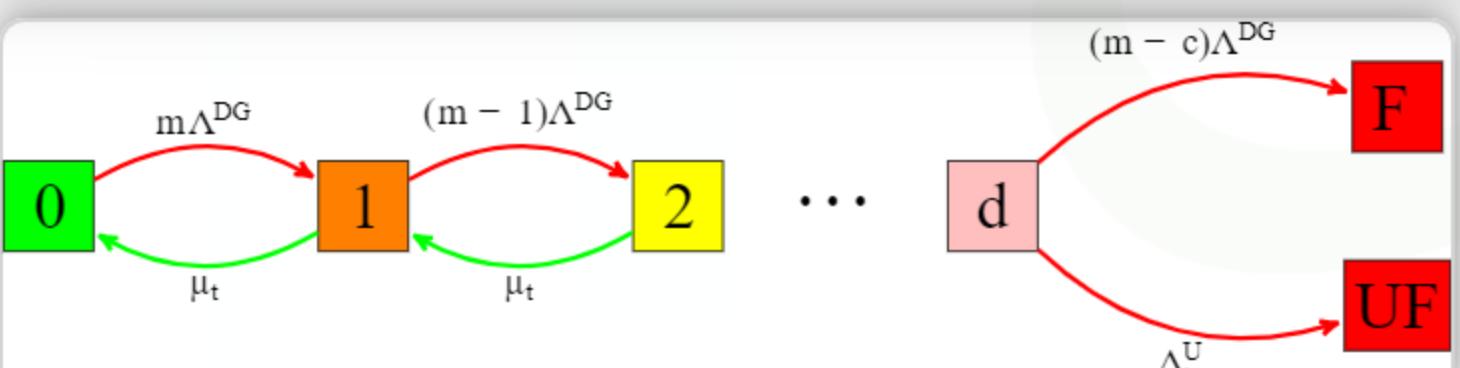
The overall data durability can be improved by implementing another layer of erasure coding. Top layer is composed of already erasure coded sub-elements.



Left: the first layer of the overall erasure coding of size n and c redundancies.
Right: A block diagram representation of the erasure coded system as a single element. ... [+]



Calm on the surface, but always paddling like hell underneath. A lot of data paddling inside the enclosure, but from outside, it is just a very reliable petabyte drive.



Top layer EC of size m and d redundancies built with self-erasure coded elements.

$$\frac{1}{\text{MTTDL}_d^t} = \frac{1}{\text{MTTDL}_{d,\text{DG}}^t} + \frac{h_{m-d}^t}{\text{MTTDL}_{d-1,\text{DG}}^t}$$

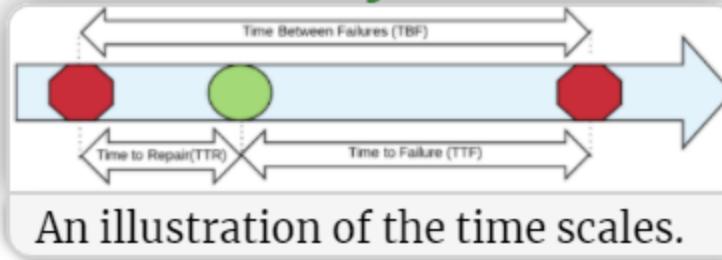
$$\text{MTTDL}_{d,\text{DG}}^t = \left(\frac{\mu_t}{\Lambda^{\text{DG}}} \right)^d \frac{(m-d-1)!}{\Lambda^{\text{DG}} m!},$$

$$h_{m-d}^t = 1 - (1 - \text{UER})^{(n-c)(m-d)C} \text{ probability of URE(s).}$$

- 16+2+Adapt (53-drive-pool) & 7+1 CORTX gives ~13 nines at the overall 73% capacity efficiency (27% overhead).
- ~12 nines can be reached with an 8+5 single layer EC at 62% capacity efficiency (38% overhead).

System Availability

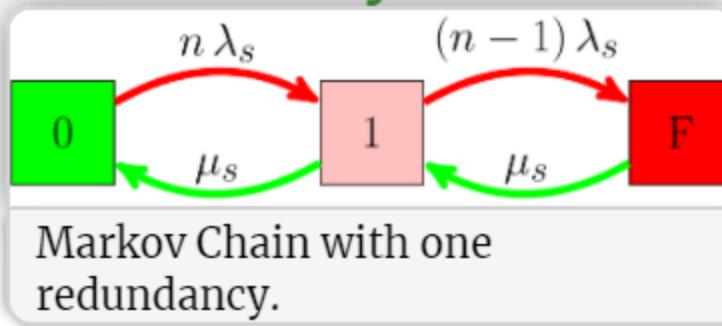
Availability with no redundancy



- A set of n devices: $\overline{\text{MTTF}} = 1/(n\lambda_s)$, and $\overline{\text{MTTR}} = 1/\mu_s$.
- Fraction of time spent for repair: $\frac{\overline{\text{MTTR}}}{\overline{\text{MTTF}} + \overline{\text{MTTR}}} \simeq \frac{\overline{\text{MTTR}}}{\overline{\text{MTTF}}} = \frac{n\lambda_s}{\mu_s}$.
- Availability=1-Fraction of time spent for repair: $A_0 \simeq 1 - \frac{n\lambda_s}{\mu_s}$. (16)

Show a rigorous proof

Availability with one redundancy

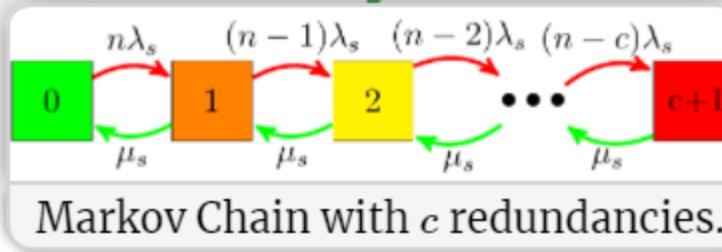


- With one redundancy, the data will be unavailable when 2 or more systems are down.
- Such a system can be modeled with the Markov chain shown on the left
- The availability can be computed as

$$A_1 = 1 - \left[n \frac{\lambda_s}{\mu_s} \right] \left[(n-1) \frac{\lambda_s}{\mu_s} \right] \quad (21)$$

Show a rigorous proof

Availability with c redundancies



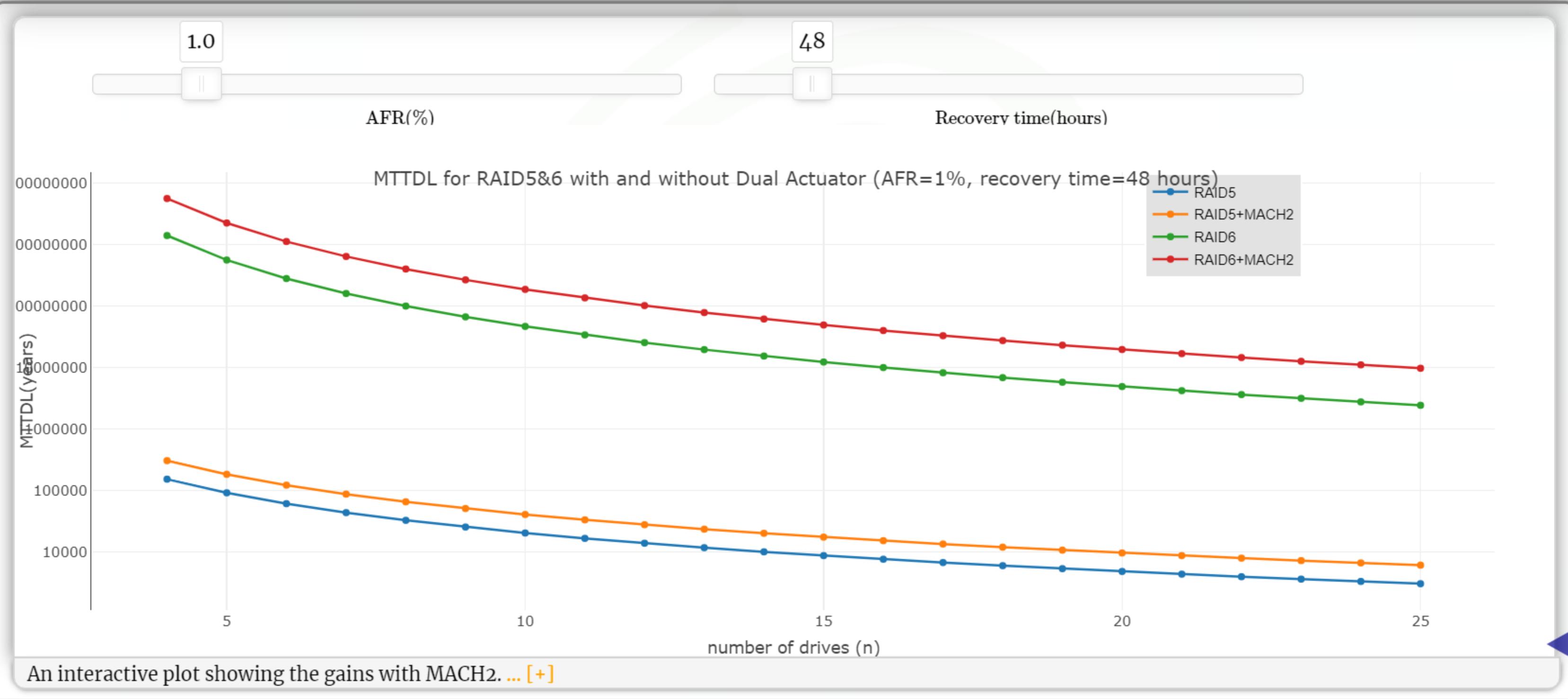
Recognizing the patterns in Eqs. (16) and (21) we can generalize the formula to a system with c redundancies:

$$A_c \simeq 1 - \frac{n!}{(n-c-1)!} \left[\frac{\lambda_s}{\mu_s} \right]^{c+1} \quad (27)$$

Durability with Mach2

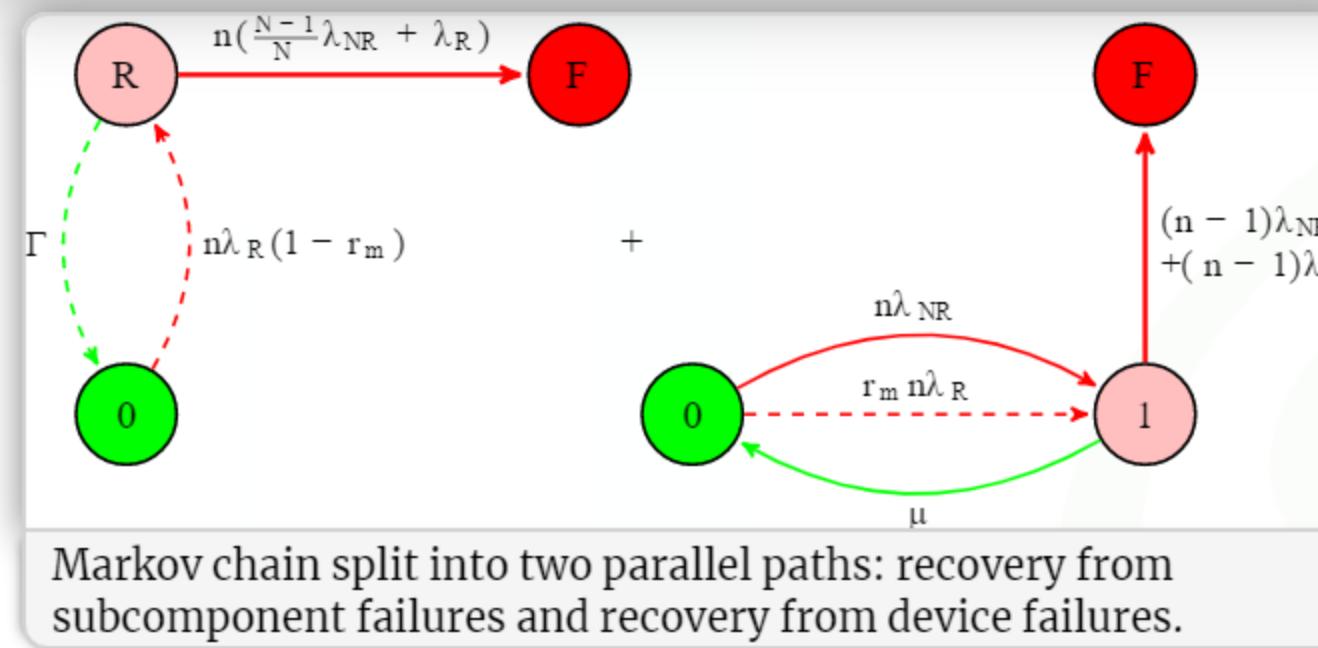
Show the GIF

- The reliability of RAID critically depends on the speed of the recovery.
- MACH2 doubles the data transfer \Rightarrow **2x/4x** reliability improvement for RAID5/6.



Durability with ReMAN

- Consider with the simplest case[14]: one redundancy. Given a head/drive failure, there are two paths to losing data:
 - A second failure while recovering from a head failure: less likely due to faster recovery.
 - A second failure while recovering from a drive failure: more likely due to longer recovery.



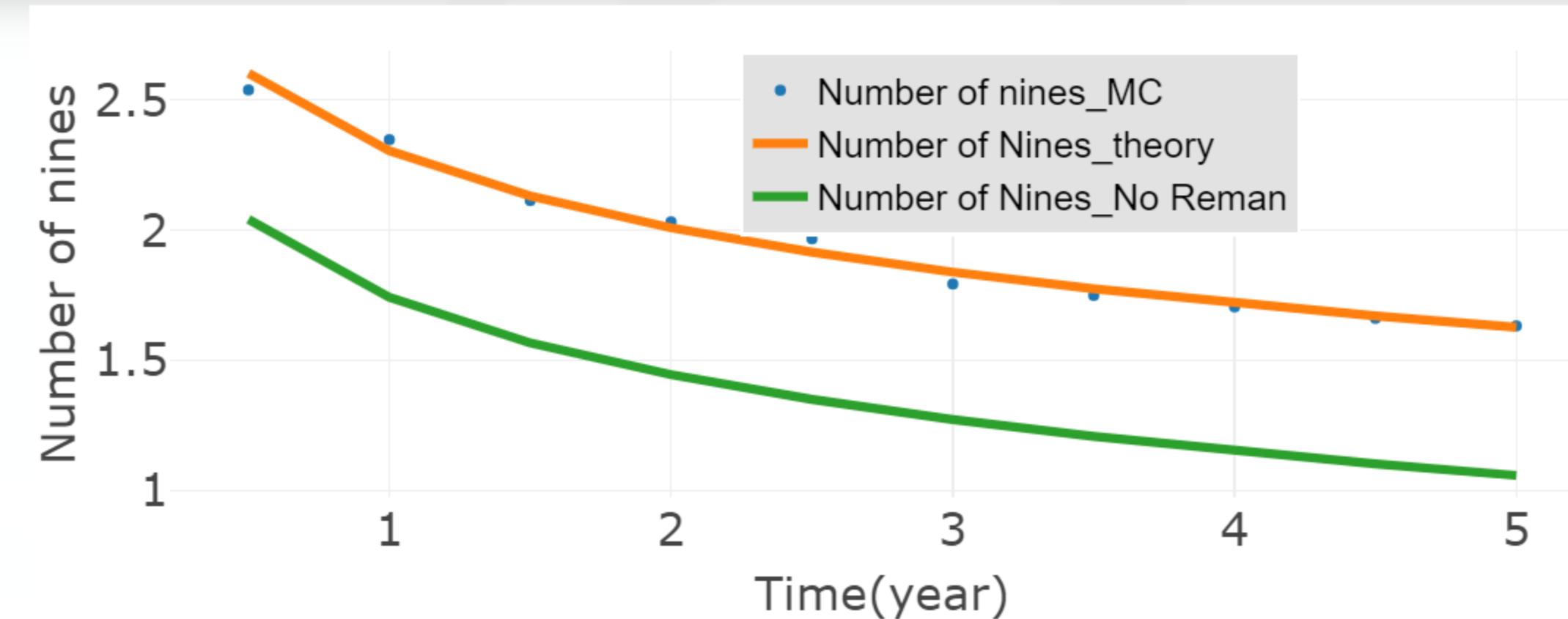
- The plot on the left shows two parallel Markov chains:
 - Left-most shows data recovery for a ReMan'able failure.
 - Right-most shows the standard recovery when a drive fails.
- r_m term represents the maximally ReMan'ed drive population.
 - ReMan'able failures on maximal drives trigger replacement.
 - This is represented by the dashed red curves.
- The transition rates are time dependent, and involve r_m functions.

- $\mathcal{R}_1^{(1)}(t) = \exp\left(-\frac{\mathcal{C}}{2\Gamma}\left[t + \frac{1-e^{-2\lambda_R t}}{2\lambda_R}\right] - \frac{\mathcal{C}}{2\mu}\left[(1+2\kappa)t - \frac{1-e^{-2\lambda_R t}}{2\lambda_R}\right]\right)$ where $\mathcal{C} \equiv n^2 \lambda_R^2 (1+\kappa)$, and $\kappa \equiv \frac{\lambda_{NR}}{\lambda_R}$
 - n : number of drives in the EC scheme, N : number of heads per drive,
 - Γ : ReMan Repair rate, μ : drive repair rate, λ_R : ReMan'able failure rate, λ_{NR} : non-ReMan'able failure rate.
- This is the expression when we allow for 1 ReMan/drive. Similar formulas are calculated for $R_{max} = 2, 3$.

- We have an analytical model of Erasure Coded systems that support ReMan.
- The closed mathematical form can be computed instantly enabling a real-time web application.

Simulation with online ReMan

- Below is a comparison of Monte Carlo Simulation and theoretical results:
 - ReMan'able AFR= 8% and non ReMan'able AFR=2%.
 - Note that AFRs are take unrealistically high to show the functional behavior.
 - Assuming 1-ReMan per drive is allowed.
- The plot shows the theoretical prediction is remarkably accurate.
 - The gain in durability coming from ReMan is about 4x.



Number of nines with $\lambda_R = 1/12$ and $\lambda_{NR} = 1/50$ (1/year), which correspond to 8% and 2% AFR respectively.

References

- [1] Y. Xie, *Dynamic documents with R and knitr*, 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC, 2015 [Online]. Available: <http://yihui.name/knitr/>
- [2] Hakim El Hattab, *Revealjs*. 2020 [Online]. Available: <https://revealjs.com/>
- [3] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (raid)," *SIGMOD Rec.*, vol. 17, no. 3, pp. 109–116, Jun. 1988, doi: [10.1145/971701.50214](https://doi.org/10.1145/971701.50214). [Online]. Available: <https://doi.org/10.1145/971701.50214>
- [4] L. Hellerstein, G. A. Gibson, R. M. Karp, R. H. Katz, and D. A. Patterson, "Coding techniques for handling failures in large disk arrays," *Algorithmica*, vol. 12, no. 2, pp. 182–208, Sep. 1994, doi: [10.1007/BF01185210](https://doi.org/10.1007/BF01185210). [Online]. Available: <https://doi.org/10.1007/BF01185210>
- [5] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theor.*, vol. 56, no. 9, pp. 4539–4551, Sep. 2010, doi: [10.1109/TIT.2010.2054295](https://doi.org/10.1109/TIT.2010.2054295). [Online]. Available: <https://doi.org/10.1109/TIT.2010.2054295>
- [6] G. Wang, L. Xiao-Guang, and L. Jing, "Parity declustering data layout for tolerating dependent disk failures in network raid systems," 2002, pp. 22–25, doi: [10.1109/ICAPP.2002.1173547](https://doi.org/10.1109/ICAPP.2002.1173547).
- [7] V. Venkatesan and I. Iliadis, "Effect of codeword placement on the reliability of erasure coded data storage systems," in *Quantitative evaluation of systems*, 2013, pp. 241–257.
- [8] A. Thomasian and M. Blaum, "Higher reliability redundant disk arrays: Organization, operation, and coding," *ACM Trans. Storage*, vol. 5, no. 3, Nov. 2009, doi: [10.1145/1629075.1629076](https://doi.org/10.1145/1629075.1629076). [Online]. Available: <https://doi.org/10.1145/1629075.1629076>
- [9] T. Kawaguchi, "Reliability analysis of distributed raid with priority rebuilding," 2013.
- [10] W. Padgett, "Weibull distribution," 2011, pp. 1651–1653.
- [11] B. Wilson, "Cloud storage durability," 2018 [Online]. Available: <https://www.backblaze.com/blog/cloud-storage-durability/>
- [12] B. Beach, "Python code to calculate the durability of data stored with erasure coding," 2018 [Online]. Available: <https://github.com/Backblaze/erasure-coding-durability/>
- [13] I. Iliadis, R. Haas, X.-Y. Hu, and E. Eleftheriou, "Disk scrubbing versus intra-disk redundancy for high-reliability raid storage systems," vol. 36, no. 1, pp. 241–252, Jun. 2008, doi: [10.1145/1384529.1375485](https://doi.org/10.1145/1384529.1375485). [Online]. Available: <https://doi.org/10.1145/1384529.1375485>
- [14] S. Olmez, "Reliability analysis of storage systems with partially repairable devices," 2021 [Online]. Available: Under peer review, available up on request