Mean time to meaningless:
MTTDL, Markov models, and storage system reliability

Kevin M. Greenan
ParaScale, Inc.

James S. Plank
University of Tennessee

Jay J. Wylie
HP Labs

Abstract

Mean Time To Data Loss (MTTDL) has been the standard reliability metric in storage systems for more than 20 years. MTTDL represents a simple formula that can be used to compare the reliability of small disk arrays and to perform comparative trending analyses. The MTTDL metric is often misused, with egregious examples relying on the MTTDL to generate reliability estimates that span centuries or millennia. Moving forward, the storage community needs to replace MTTDL with a metric that can be used to accurately compare the reliability of systems in a way that reflects the impact of data loss in the real world.

1 Introduction

"Essentially, all models are wrong, but some are useful"
– George E.P. Box

Since Gibson's original work on RAID [3], the standard metric of storage system reliability has been the *Mean Time To Data Loss* (MTTDL). MTTDL is an estimate of the expected time that it would take a given storage system to exhibit enough failures such that at least one block of data cannot be retrieved or reconstructed.

One of the reasons that MTTDL is so appealing as a metric is that it is easy to construct a Markov model that yields an analytic closed-form equation for MTTDL. Such formulae have been ubiquitous in research and practice due to the ease of estimating reliability by plugging a few numbers into an expression. Given simplistic assumptions about the physical system, such as independent exponential probability distributions for failure and repair, a Markov model can be easily constructed resulting in a nice, closed-form expression.

There are three major problems with using the MTTDL as a measure of storage system reliability. First, the models on which the calculation depends rely on an extremely simplistic view of the storage system. Second, the metric does not reflect the real world, but is often interpreted as a real world estimate. For example, the Pergamum archival storage system estimates a MTTDL of 1400 years [13]. These estimates are based on the assumptions of the underlying Markov models and are typically well beyond the life of any storage system. Finally, MTTDL values tend to be incomparable because each is a function of system scale and omits the (expected) magnitude of data loss.

In this position paper, we argue that MTTDL is a bad reliability metric and that Markov models, the traditional means of determining MTTDL, do a poor job of modeling modern storage system reliability. We then outline properties we believe a good storage system reliability metric should have, and propose a new metric with these properties: NOrmalized Magnitude of Data Loss (NOMDL). Finally, we provide example reliability results using various proposed metrics, including NOMDL, for a simple storage system.

2 The Canonical MTTDL Calculation

In the storage reliability community, the MTTDL is calculated using Continuous-time Markov Chains (a.k.a. Markov model). The canonical Markov model for storage systems is based on RAID4, which tolerates exactly one device failure. Figure 1 shows this model. There are a total of three states. State 0 is the state with all n devices operational. State 1 is the state with one failed device. State 2 is the state with two failed devices, i.e., the data loss state. The model in Figure 1 has two rate parameters: λ, a failure rate and μ, a repair rate. It is assumed that all devices fail at the same rate and repair at the same rate.

At $t = 0$, the system starts pristinely in state 0, and remains in state 0 for an average of $(n \cdot \lambda)^{-1}$ hours (n device failures are exponentially distributed with failure rate λ), when it transitions to state 1. The system is then in state 1 for on average $(((n-1) \cdot \lambda)) + \mu)^{-1}$ hours. The system transitions out of state 1 to state 2, which is the data loss state, with probability $\frac{(n-1) \cdot \lambda}{((n-1) \cdot \lambda)) + \mu}$. Otherwise, the system transitions back to state 0, where the system

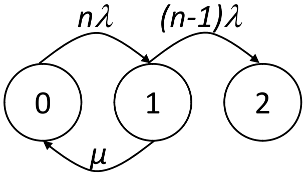

Figure 1: Canonical RAID4/RAID5 Markov model.

is fully operational and devoid of failures.

The canonical Markov model can be solved analytically for MTTDL, and simplified:

$$MTTDL = \frac{\mu + (2n-1)\lambda}{n(n-1)\lambda^2} \approx \frac{\mu}{n(n-1)\lambda^2}.$$

Such an analytic result is appealing because reassuringly large MTTDL values follow from disk lifetimes measured in hundreds of thousands of hours. Also, the relationship between expected disk lifetime ($1/\lambda$), expected repair time ($1/\mu$), and MTTDL for RAID4 systems is apparent.

3 The MTTDL: Meaningless Values

Siewiorek and Swarz [12] define reliability as follows: "The reliability of a system as a function of time, $R(t)$, is the conditional probability that the system has survived the interval $[0, t]$, given that the system was operational at time $t = 0$." Implicit in the statement that the "system has survived" is that the system is performing its intended function under some operating conditions. For a storage system, this means that the system does not lose data during the expected system lifetime.

An important aspect of the definition of reliability, that is lost in many discussions about storage system reliability, is the notion of *mission lifetime*. MTTDL does not measure reliability directly; it is an expectation based on reliability: MTTDL $= \int_0^\infty R(t)dt$. MTTDL literally measures the expected time to failure over an infinite interval. This may make the MTTDL useful for quick, relative comparisons, but the absolute measurements are essentially meaningless. For example, an MTTDL measurement of 1400 years tells us very little about the probability of failure during a realistic system mission time. A system designer is most likely interested in the probability and extent of data loss, every year for the first 10 years of a system. The aggregate nature of the MTTDL provides little useful information, and no insight into these calculations.

4 The MTTDL: Unrealistic Models

While Markov Models are appealingly simple, the assumptions that make them convenient also render them inadequate for multi-disk systems In this section, we highlight three ways in which the assumptions significantly do not match modern systems. A more detailed discussion that includes additional concerns may be found in Chapter 4 of Greenan's PhD thesis [4].

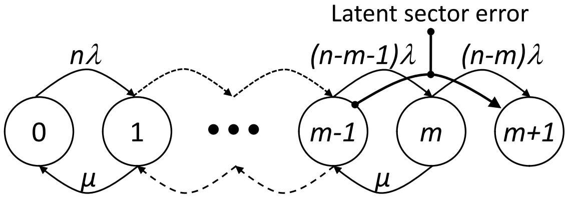

Figure 2: Illustrative multi-disk fault tolerant Markov model with sector errors.

For illustration, in Figure 2 we present a Markov Model based on one from [5] that models a n-disk system composed of k disks of data and m disks of parity, in the presence of disk failures and latent sector errors.

4.1 Exponential Distributions

Implicit in the use of Markov models for storage system reliability analysis is the assumption that failure and repair rates follow an exponential distribution and are constant. The exponential distribution is a poor match to observed disk failure rates, latent sector error rates, and disk repairs. Empirical observations have shown that disks do not fail according to independent exponential distributions [2, 6, 10], and that Weibull distributions are more successful in modeling observed disk failure behavior.

Latent sector failures exhibit significant correlation both temporally and spatially within a device [1, 11]. Beyond this, sector failures are highly usage dependent and difficult to quantify [2, 9]. The most recent synthesis of empirical data by Schroeder et al. suggests that Pareto distributions can best capture the burstiness of latent sector errors, as well as spatial and temporal correlations [11].

Disk repair activities such as rebuild and scrubbing tend to require some fixed minimal amount of time to complete. Moreover, in well-engineered systems, there tends to be some fixed upper bound on the time the repair activity may take. Events with lower and upper time bounds are poorly modeled by exponential distributions. Again, Weibull distributions capture reality better [2].

4.2 Memorylessness, Failure & Repair

Exponential distributions and Markov Models are "memoryless." When the system modeled in Figure 2 transitions into a new state, it is as if all the components in the system are reset. Available components' ages are reset to 0 (i.e., brand new), and any repair of failed components is forgotten. Both cases are problematic.

When the system transitions to a new state, it is as if all available disks are refreshed to a "good-as-new" state. In particular, the transition from state 1 to state 0 models a system where the repair of one disk converts all disks into their pristine states. In reality, only the recently repaired component is brand-new, while all the others have a non-zero age.

Now, consider the system under repair in state i such that $1 \leq i < m$. If a failure occurs, moving the system to state $i+1$, any previous rebuilding is assumed to be discarded, and the variable μ governs the transition back to state i. It is as if the *most recent* failure dictates the repair transition. In reality, it is the *earliest* failure, whose rebuild is closest to completion, that governs repair transitions.

These two issues highlight the difficulty with the memorylessness assumption: each transition "forgets" about progress that has been made in a previous state – neither component wear-out nor rebuild progress are modeled. Correctly incorporating these time-dependent properties into such a model is quite difficult. The difficulty lies in the distinction between *absolute time* and *relative time*. Absolute time is the time since the system was generated, while relative time applies to the individual device lifetime and repair clocks. Analytic models operate in absolute time; therefore, there is no reasonable way to determine the values of each individual clock. Simulation methods can track relative time and thus can effectively model reliability of a storage system with time-dependent properties.

4.3 Memorylessness & Sector Errors

In a m-disk fault tolerant system, the storage system enters *critical mode* upon the m-th disk failure. The transition in the Markov model in Figure 2 from the $m-1$ to the $m+1$ state is intended to model data loss due to sector errors in critical mode. In this model, any sector errors or bit errors encountered during rebuild in critical mode lead to data loss. Unfortunately, such a model overestimates the system unreliability. A sector failure only leads to data loss if it occurs in the portion of the failed disk that is critically exposed. For example, in a two-disk fault tolerant system, if the first disk to fail is 90% rebuilt when a second disk fails, only 10% of the disk is critically exposed. Figure 3 illustrates this point in general. This difficulty with Markov models again follows from the memorylessness assumption.

5 Metrics

Many alternatives have been proposed to replace MTTDL as a reliability metric (e.g., [2, 7, 8]). While these metrics have some advantages, none have all the qualities that we believe such a metric must. In our opinion, a storage system reliability metric must have the following properties:

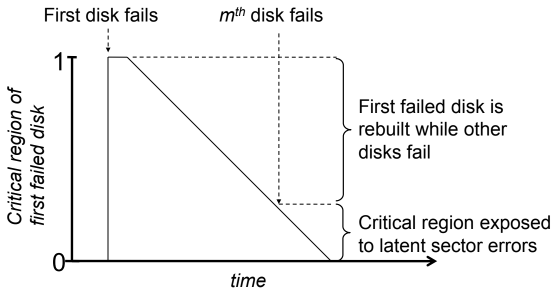
First disk fails
m^{th} disk fails
1
0
Critical region of first failed disk
time
First failed disk is rebuilt while other disks fail
Critical region exposed to latent sector errors

Figure 3: Critical region of first failed disk susceptible to data loss due to latent sector errors.

Calculable. There must exist a reasonable method to calculate the metric. For example, a software package based on well understood simulation or numerical calculation principles can be used.

Meaningful. The metric must relate directly to the reliability of a deployed storage system.

Understandable. The metric, and its units, must be understandable to a broad audience that includes developers, marketers, industry analysts, and researchers.

Comparable. The metric must allow us to compare systems with different scales, architectures, and underlying storage technologies (e.g. solid-state disk).

5.1 NOMDL: A Better Metric?

We start by creating a metric called Magnitude of Data Loss (MDL). Let MDL_t be the expected amount of data lost (in bytes) in a target system within mission time t. There are two advantages to MDL_t. First, like system reliability, the metric deals with arbitrary time periods. This means that a system architect can use MDL_t to estimate the expected number of bytes lost in the first year of deployment, or first ten years of deployment. Second, the units are understandable: bytes and years.

Unfortunately, MDL_t is not a metric that compares well across systems. Consider an 8 disk RAID4 array with intra-disk parity [11] and a 24 disk RAID6 array, both composed of the exact same 1 TB drives. The RAID6 array will result in a higher MDL_t, but has more than 3 times the usable capacity of the RAID4 array.

The MDL can be made comparable by normalizing to the system's usable capacity; doing so yields the NOrmalized Magnitude of Data Loss (NOMDL) metric. NOMDL_t measures expected amount of data lost per usable terabyte within mission time t. For example, the NOMDL_t of a system may be 0.001 bytes lost per usable terabyte in the first 5 years of deployment. Since

NOMDL$_t$ is a normalized version of MDL$_t$, both metrics can be output from the same base calculation.

5.2 Calculating NOMDL$_t$

NOMDL$_t$ is calculable. Markov models can measure the probability of being in a specific state within a mission time. With care, the probabilities can be calculated for all paths in a Markov model and used to derive the number of expected bytes lost. Unfortunately, as we have described above, we do not believe that Markov models accurately capture the behavior of contemporary storage systems. Maybe other modeling paradigms such as Petri Nets and such variants can be used to calculate NOMDL$_t$ while addressing the deficiencies of Markov models.

Our recommendation is to use Monte Carlo simulation to calculate NOMDL$_t$. At a high level, such a simulation can be accomplished as follows. Initially, all devices are assigned failure and repair characteristics. Then, device failures and their repair times are drawn from an appropriate statistical distribution (e.g., Exponential, Weibull, Pareto) for each device. Devices are queued and processed by failure time and the simulation stops at a predefined mission time. Once a device is repaired, another failure and repair time is drawn for that device. Each time a failure occurs, the simulator analyzes the system state to determine if data loss has occurred. Detail of the techniques and overhead associated with simulation is discussed in Greenan's PhD thesis [4].

If the time of data loss is F and the mission time is t, then the simulator implements the function, $I(F < t) = \{0, 1\}$ (0 is no data loss, 1 is data loss). That is, the system either had a data loss event within the mission time or not. Many iterations of the simulator are required to get statistically meaningful results. The standard method of computing system reliability via simulation is to run N iterations (typically chosen experimentally) of the simulator and make the following calculation:

$$R(t) = 1 - \sum_{i=1}^{N} \frac{I(F_i < t)}{N}$$

Since $I(F_i < t)$ evaluates to 1 when there is data loss in iteration i and 0 otherwise, this directly calculates the probability of no data loss in $[0, t]$. Given the magnitude of data loss upon a data loss event, C_i, this standard calculation can produce the MDL$_t$:

$$\text{MDL}_t = \sum_{i=1}^{N} \frac{I(F_i < t) \cdot C_i}{N}.$$

The NOMDL$_t$ is the MDL$_t$ normalized to the usable capacity of the system, D: $\text{NOMDL}_t = \text{MDL}_t/D$. Using simulation thus produces $\{R(t), \text{MDL}_t, \text{NOMDL}_t\}$ which in our opinion makes NOMDL$_t$ a calculable, meaningful, understandable, and comparable metric.

5.3 Comparison of Metrics

Table 1 provides a high-level comparison of MTTDL and other recently proposed storage reliability metrics. We compare the metrics qualitatively in terms of the aforementioned properties of a good metric. We also perform a sample calculation for each metric of a simple storage system: an 8-disk RAID4 array of terabyte hard drives with periodic scrubbing for a 10 year mission time. The failure/repair/scrub characteristics are taken from Elerath and Pecht [2]. All calculations were performed using the HFRS reliability simulation suite (see §8).

Here we compare MTTDL, *Bit Half-Life* (BHL) [8], *Double-Disk Failures Per 1000 Reliability Groups* (DDF pKRG) [2], *Data Loss events per Petabyte Year* (DALOPY) [5] and NOMDL$_t$. MTTDL and DALOPY are calculated via Markov models. BHL is calculated by finding the time at which a bit has a 0.50 probability of failure, which is difficult to calculate via simulation and can be estimated using a Markov model. For BHL, this time is calculated for the entire system instead of a single bit. DDF pKRG and NOMDL$_t$ are computed using Monte Carlo simulation.

Both calculations for MTTDL and BHL result in reliability metrics that are essentially meaningless. Even in a RAID4 system with the threat of sector errors (2.6% chance when reading an entire disk) both metrics produce numbers that are well beyond the lifetime of most existing systems. In addition, both metrics produce results that are not comparable between systems that differ in terms of technology and scale.

DDF pKRG and DALOPY are interesting alternatives to the original MTTDL calculation. DDF pKRG is not sensitive to technological change, but is bound architecturally to a specific RAID level or erasure-coding scheme (double disk failure is specific to RAID4 or RAID5). DALOPY has most of properties of a good metric, but is not comparable. In particular, it is not comparable across systems based on different technologies or architectures. While DALOPY normalizes the expected number of data loss events to the system size, it does not provide the magnitude of data loss. Without magnitude it is hard to compare DALOPY between systems; data loss event gives no information on what or how much data was lost. In addition, the units of the metric are hard to reason about.

NOMDL$_t$ is not sensitive to technological change, architecture or scale. The metric is normalized to system scale, is comparable between architectures and directly measures the expected magnitude of data loss. As shown in Table 1, the units of NOMDL$_t$—bytes lost per usable TB—are easy to understand. Beyond this, the subscript

	Meaningful	**Understandable**	**Calculable**	**Comparable**	**Result**
MTTDL			✓		37.60 years
BHL			✓		26.06 years
DDF pKRG	✓	✓	✓		183 DDFs
DaLoPY	✓	✓	✓		3.32 DL per (PB*Yr)
NOMDL$_{10y}$	✓	✓	✓	✓	14.41 bytes lost per usable TB

Table 1: Qualitative comparison of different storage reliability metrics.

$t = 10y$, clearly indicates the mission lifetime and so helps ensure that only numbers based on the same mission lifetime are actually compared.

6 Conclusions

We have argued that MTTDL is essentially a meaningless reliability metric for storage systems and that Markov models, the normal method of calculating MTTDL, is flawed. We are not the first to make this argument (see [2] and [8]) but hope to be the last. We believe NOMDL$_t$ has the desirable features of a good reliability metric, namely that it is calculable, meaningful, understandable, and comparable, and we exhort researchers to exploit it for their future reliability measurements. Currently, we believe that Monte Carlo simulation is the best way to calculate NOMDL$_t$.

7 Acknowledgments

This material is based upon work supported by the National Science Foundation under grants CNS-0615221.

8 HFRS Availability

The High-Fidelity Reliability (HFR) Simulator is a command line tool written in Python and is available at

`http://users.soe.ucsc.edu/~kmgreen/`.

References

[1] Bairavasundaram, L. N., Goodson, G. R., Schroeder, B., Arpaci-Dusseau, A. C., and Arpaci-Dusseau, R. H. An Analysis of Data Corruption in the Storage Stack. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST '08)* (San Jose, California, February 2008).

[2] Elerath, J., and Pecht, M. Enhanced reliability modeling of raid storage systems. In *Dependable Systems and Networks, 2007. DSN '07. 37th Annual IEEE/IFIP International Conference on* (june 2007), pp. 175–184.

[3] Gibson, G. A. *Redundant Disk Arrays: Reliable, Parallel Secondary Storage*. PhD thesis, Univeristy of California, Berkeley, December 1990. Technical Report CSD-91-613.

[4] Greenan, K. M. *Reliability and Power-Efficiency in Erasure-Coded Storage Systems*. PhD thesis, Univeristy of California, Santa Cruz, December 2009. Technical Report UCSC-SSRC-09-08.

[5] Hafner, J. L., and Rao, K. Notes on reliability models for non-MDS erasure codes. Tech. Rep. RJ–10391, IBM, October 2006.

[6] Pinheiro, E., Weber, W.-D., and Barroso, L. A. Failure trends in a large disk drive population. In *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST '07)* (2007), USENIX Association.

[7] Rao, K. K., Hafner, J. L., and Golding, R. A. Reliability for networked storage nodes. In *DSN-06: International Conference on Dependable Systems and Networks* (Philadelphia, 2006), IEEE, pp. 237–248.

[8] Rosenthal, D. S. H. Bit preservation: A solved problem? In *Proceedings of the Fifth International Conference on Preservation of Digital Objects (iPRES '08)* (London, UK, September 2008).

[9] Rozier, E., Belluomini, W., Deenadhayalan, V., Hafner, J., Rao, K., and Zhou, P. Evaluating the impact of undetected disk errors in raid systems. In *Dependable Systems Networks, 2009. DSN '09. IEEE/IFIP International Conference on* (29 2009-july 2 2009), pp. 83–92.

[10] Schroeder, B., and Gibson, G. A. Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you? In *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST '07)* (2007), USENIX Association, pp. 1–16.

[11] Schroeder, B., Gill, P., and Damouras, S. Understanding latent sector errors and how to protect against them. In *Proceedings of the 8th Conference on File and Storage Technologies (FAST '10)* (San Jose, California, February 2010).

[12] Siewiorek, D. P., and Swarz, R. S. *Reliable computer systems (3rd ed.): design and evaluation*. A. K. Peters, Ltd., Natick, MA, USA, 1998.

[13] Storer, M. W., Greenan, K., Miller, E. L., and Voruganti, K. Pergamum: Replacing tape with energy efficient, reliable, disk-based archival storage. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST '08)* (Feb. 2008), pp. 1–16.