International Journal of Applied Sciences and Smart Technologies

Volume 2, Issue 2, pages 91–110

p-ISSN 2655-8564, e-ISSN 2685-9432

RAID Level 6 and Level 6+ Reliability

Thomas Schwarz

Department of Computer Science, Klingler College of Arts and Sciences, Marquette University, Milwaukee, Wisconsin, U.S.A.
Corresponding Author: thomas.schwarz@marquette.edu

(Received 19-08-2020; Revised 09-11-2020; Accepted 09-11-2020)

Abstract

Storage systems are built of fallible components but have to provide high degrees of reliability. Besides mirroring and triplicating data, redundant storage of information using erasure-correcting codes is the only possibility to have data survive device failure. We provide here exact formula for the data-loss probability of a disk array composed of several RAID Level 6 stripes. This two-failure tolerant is not only used in practice but can also provide a reference point for the assessment of other data organizations.

Keywords: RAID level 6, robustness, reliability

1 Introduction

Storage systems are built of a large number of fallible components. Failure of some of these components might be independent or might be correlated. Often, the causes of failures are obscure. For example, Jim Gray's team at Microsoft research once observed that disk reliability in an experimental database for astronomy remarkably increased after switching to slightly more expensive enclosures [1]. In this case, the vibrations of a failing disk drives caused its neighbors to fail with a much higher probability.

Failures can happen at all levels in the storage hierarchy. Central components such as air conditioning or power supply can fail, and data centers can be flooded. Individual

Volume 2, Issue 2, pages 91–110
p-ISSN 2655-8564, e-ISSN 2685-9432

racks can also suffer failure of networking and electricity supply. Individual disks can fail in toto or suffer latent sector failures. These incidents happen at surprisingly high rates even in large state-of-the art data centers [2, 3, 4].

To protect against failures, data has to be stored redundantly. With the recognition of the importance of latent sector failures, intra-disk redundancy and other schemes were introduced [5], but it was also recognized that protection against single failure is not sufficient.

However, protection against single disk failures date back much further. Originally termed Redundant Arrays of Inexpensive Disks, Redundant Arrays of Independent Disks (RAID) originally were proposed to allow parallel I/O but were quickly recognized to solve an even more important problem, namely the tendency of magnetic hard drives to fail, often without warning. The original RAID architecture added a parity disk to an ensemble of n data disks. This parity disk would store the bit-wise exclusive-or of parallel sectors in the n data disks in its sectors. When a data disk failed or became otherwise inaccessible, its contents could be recovered by reading parallel sectors in all n surviving disks as the bit-wise exclusive-or of the sector contents. This arrangement is called a RAID Level 4. The final architecture, the RAID Level 5, would rotate the roll of parity disk among $n + 1$ disks and thus do away with the differences in the load. A dedicated parity disk is written with each update to any of the data disks, but not read with any look-up, so that its load is the same as that of a dedicated data disk only if the read and write loads of the ensemble fulfill a certain linear equation. Nevertheless, some important players in the storage industry use dedicated parity disks.

Not only the presence of latent sector errors (discovered only upon trying to access a disk sector) but also the coupling of complete disk failures motivated the introduction of higher degrees of failure tolerance. The RAID Level 6 adds one more parity disk to the ensemble. The contents of the new parity disk have to be calculated using a more involved code, usually defined using Galois field operations, though many other possibilities exist [e.g. 6, 7].

Since its inception, the architecture of failure tolerating disk arrays has undergone constant innovation and more erasure correcting codes and disk array layouts are proposed [8, 9]. To evaluate their effectiveness, a disk array consisting of several RAID

Volume 2, Issue 2, pages 91–110
p-ISSN 2655-8564, e-ISSN 2685-9432

Level 6 is used as a point of comparison. It seems however that no single calculation of RAID Level 6 reliability is published. This short contribution therefore aims at rectifying this lacuna. It is in part motivated by reviewing papers that make mistakes in the assessment of disk array reliability. We are right now experiencing a revolution in the storage industry where flash replaces disk and maybe soon PCM or another non-volatile memory architecture replaces both. Our analysis will remain valid independent of the chosen underlying technology.

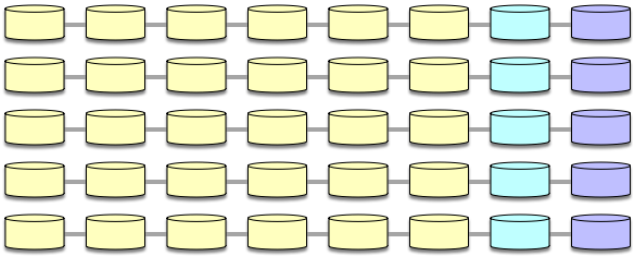


Figure 1. A disk array made up of five stripes organized as a Level 6 RAID. Each stripe consists of six data disks, a P-parity disk, and a Q-parity disk.

2 Failure Tolerance of an Ensemble of Raid Level 6 With or Without Distributed Parity

We first look at the classic Level 6 RAID, depicted in Figure 1. The RAID consists of m stripes ($m = 5$ in the figure), each with n data disks and *2* parity disks. In Figure 1, we have two types of parity disks, a P-parity, calculated as the bitwise exclusive-or of the data disks in the stripe, and a Q-parity, calculated using Galois field operations. If we distribute parity, we no longer have dedicated parity disks, but each disk will have P- and Q-parity blocks in equal proportion. We will also look at two variants. The first variant uses declustering [10], where all disks are divided into *disklets* of equal size. To mn of the $m(n + 2)$ disklets in parallel position, we assign the role of data disklet, to m we assign the role of P-parity disklet, and to the remaining m we assign the role of Q-parity disklet in a manner that the organization of Figure 1 is true at the disklet level. Before declustering, a single or double failure in a stripe made us read almost all or all of the disks in the reliability stripe. This recovery load is thus limited to

Volume 2, Issue 2, pages 91–110
p-ISSN 2655-8564, e-ISSN 2685-9432

a single stripe. In the declustered version, this recovery load is evenly distributed over all surviving disks. On the negative side, if three disks have failed, then it is likely (to be made more precise below) that three disklets that have failed are located in the same stripe in one of the many configurations. Thus, the possibility to withstand failures is smaller, but on the other side, recovery is faster.

The second variant adds a second Q-parity to each stripe. Since there are too many contenders for the title of RAID Level 7, I just call the resulting organization a Level 6+ RAID. Now, the disk array architecture can withstand up to and including three failed disks in a single stripe without losing data.

A. Reliability of the Non-Declustered Level 6 RAID

We first calculate the robustness, that is, the probability with which an ensemble such as that shown in Figure 1 with m stripes of $n+2$ disks can withstand f individual disk failures. Clearly, in order to not have lost any data, each of the m stripes can have at most two failures. We call this selection of f disks a *benign failure pattern*. Let k_1 be the number of stripes with one failed disk and k_2 be the number of stripes with two failed disks in a benign failure pattern. Clearly, $k_1 + k_2 = f$ or, trivially equivalent, $k_1 = f - k_2$. We now count the number of benign failure patterns with f failed disks. We set $\kappa = k_2$. We first select $f - 2\kappa$ stripes with a single failure out of the m stripes. Then we select κ stripes out of the remaining $m - k_1 = m - f + 2\kappa$ stripes. Finally, we select one of the $n+2$ disks in each of the κ stripes and two of the $n+2$ disks in each of the $k_2 = f - \kappa$ stripes. This gives us for the number of good patterns

$$\gamma_{n,m,f} = \sum_{\kappa=0}^{\lfloor \frac{f}{2} \rfloor} \binom{m}{f-2\kappa} \binom{m-f+2\kappa}{\kappa} \binom{n+2}{2}^{\kappa} \binom{n+2}{1}^{f-2\kappa}$$

The probability of data loss is given by

$$p_{dl}(n,m,f) = 1 - \frac{\gamma_{n,m,f}}{\binom{m(n+2)}{f}}$$

as the opposite probability of the quotient of benign failure patterns of size f over all failure patterns of size f. Despite its seeming complexity, mathematical software like

Volume 2, Issue 2, pages 91–110
p-ISSN 2655-8564, e-ISSN 2685-9432

Mathematica or software developed using infinite integer precision as afforded by Python is able to calculate $\gamma_{n,m,f}$ exactly, though not instantaneously.

For small *f*, we can obtain formulae that are easier to evaluate. For $f = 0, 1, 2$, the probability of data-loss is zero since the disk array can tolerate always two failures. For $f = 3$, it is easier to calculate the malignant failure patterns. Three disks can only cause data-loss if they are all located in the same stripe. We model the failures as occurring serially in time. The first failure then determines a stripe. The next two failures need to fall into the same stripe. There are $\binom{n+2-1}{2}$ possibilities to form this malign pattern out of $\binom{m(n+2)-1}{2}$ possibilities to select two additional disks to fail. Thus, this happens with probability

$$p_{dl}(n, m, 3) = \frac{\binom{n+1}{2}}{\binom{m(n+2)-1}{2}}$$

$$= \frac{(n+1)n}{(m(n+2)-1)(m(n+2)-2)}$$

For $f = 4$, a malign failure pattern can have either four failed disks in the same stripe or three failed disks in the same stripe and another disk somewhere in another stripe. This gives

$$m\binom{n+2}{4} + m\binom{n+2}{3}(m-1)(n+2)$$

malign patterns and the corresponding data-loss probability evaluates to

$$p_{dl}(n, m, 4) = \frac{n\,(1+n)\,(4\,m\,(2+n) - 3\,(3+n))}{(-3+m\,(2+n))\,(-2+m\,(2+n))\,(-1+m\,(2+n))}.$$

B. Reliability of the Non-Decultered Level 6+ RAID

We extend our methodology to the three-failure tolerance of each single stripe. The number of benign failure patterns $\beta_{n,m,f}$ is obtained by first selecting positive integers κ_1, κ_2, and κ_3 such that $\kappa_1 + 2\,\kappa_2 + 3\,\kappa_3 = f$, representing stripes with one, two, and three different failures. Then we select κ_1 stripes out of *m*, then κ_2 stripes out of the remaining $m - \kappa_1$, then κ_3 out of the remaining $m - \kappa_1 - \kappa_2$ stripes, (or $\binom{n}{\kappa_1, \kappa_2, \kappa_3}$ so far), and then for each of the κ_1 stripes with one failure, one out of $n + 2$, for each of the κ_2 stripes with two failures, two out of $n + 2$, and finally, for each of the κ_3 stripes

Volume 2, Issue 2, pages 91–110

p-ISSN 2655-8564, e-ISSN 2685-9432

with three failures, three out of $n+2$ disks, giving us

$$\beta_{n,m,f} = \sum_{\kappa_1+2\kappa_2+3\kappa_3=f} \binom{n}{\kappa_1,\kappa_2,\kappa_3} \binom{n+2}{3}^{\kappa_3} \binom{n+2}{2}^{\kappa_2} (n+2)^{\kappa_1}.$$

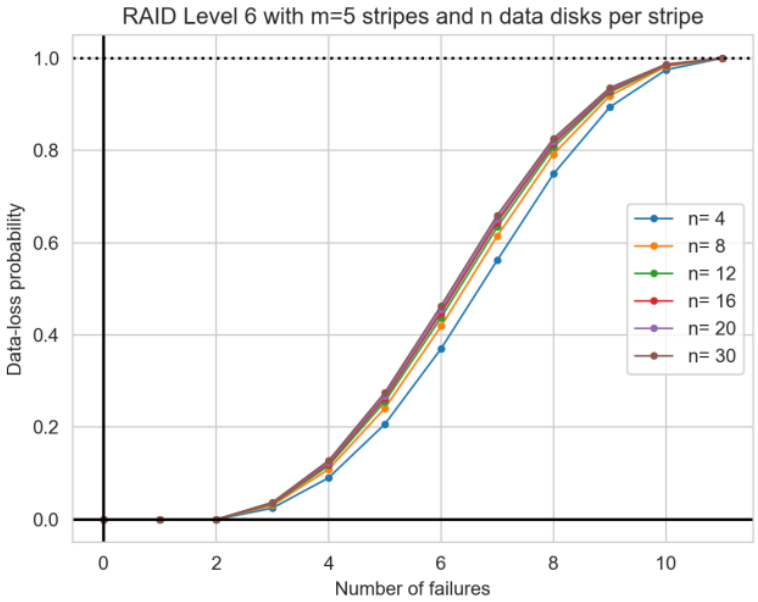


Figure 2. Data loss probability of a RAID Level 6 given f failures. The array has $m = 5$ stripes and n data disks and two parity disks per stripe

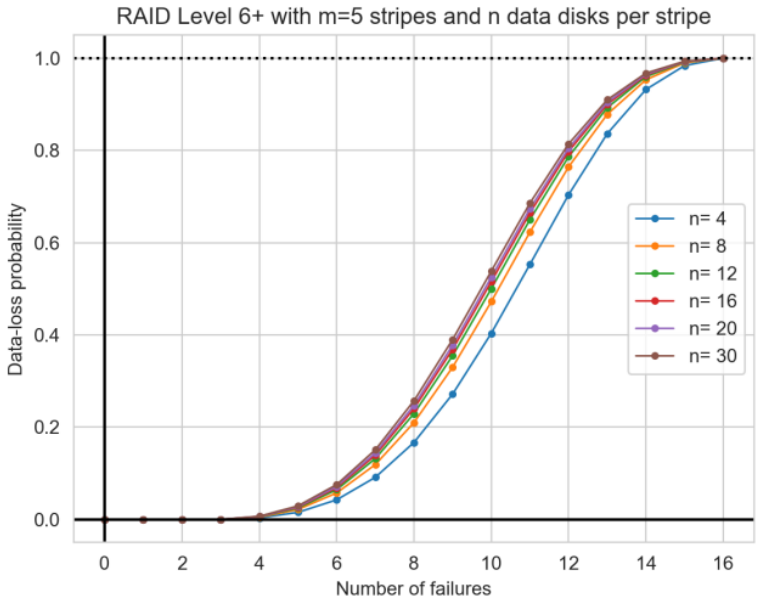


Figure 3. Data loss probability of a RAID Level 6 given f failures. The array has $m = 5$ stripes and n data disks and two parity disks per stripe.

International Journal of Applied Sciences and Smart Technologies
Volume 2, Issue 2, pages 91–110
p-ISSN 2655-8564, e-ISSN 2685-9432

We present an example for RAID Level 6 in Figure 2 and for RAID Level 6+ in Figure 3.

C. Reliability of Declustered RAID

There are many ways to decluster [10] a disk array. Sometimes, people have defined layouts, assuming that all disks are equal. In modern datacenters, it is more likely that Binary Large OBjects (BLOB) or streams are being stored by grabbing disklets on n different data disks and adding two or three parity disklets and then repeating the process until the object or the stream has been stored. The first method has the advantage that we can calculate data-loss probability exactly, but we can then use the result in the former case in order to approximate the second. Therefore, we assume here a pre-defined layout.

In more detail, we assume that all disks in the disk array have equal size. Since reliability striping at the individual sector level (4KB) would require much meta-data and gain little in dividing recovery workload among all disks in the array, we assume that each disk is split into disklets. In order to distribute workload, the number D of disklets should be at least in the hundreds, but it does not need to be much more.

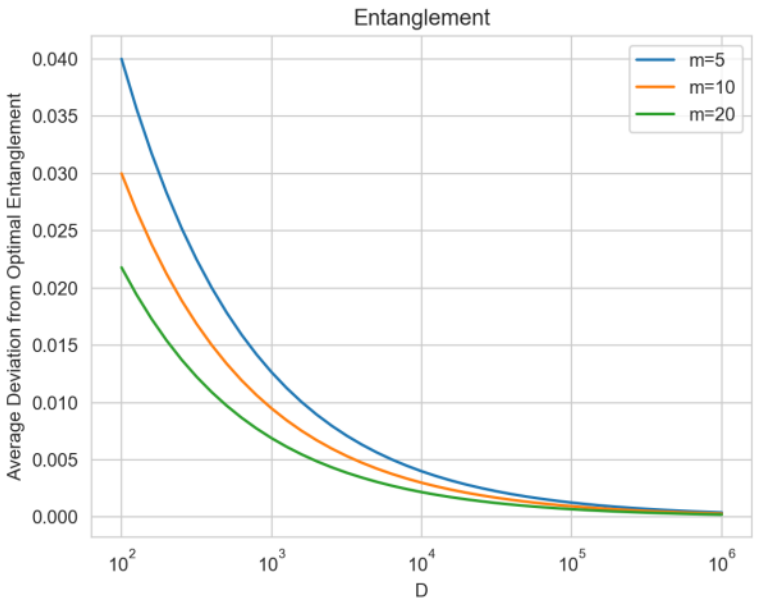


Figure 4. Average deviation from fair entanglement in dependence on the number D of disklets.

International Journal of Applied Sciences and Smart Technologies
Volume 2, Issue 2, pages 91–110
p-ISSN 2655-8564, e-ISSN 2685-9432

Indeed, a disklet on a healthy disk in a disk array with one failed disk (the most common scenario) would be potentially involved only if the disklet is in the same stripe as the corresponding disklet in the failed disk. Therefore, the disklet is potentially involved with probability $1/m$, where, as we recall, m is the number of stripes. In a RAID Level 6 or 6+, we only need to read n of the remaining healthy $n+1$ and $n+2$ surviving disks, respectively, which gives the file system some choice, that can be used for further balancing. Even without taking advantage of this, the number of times t that a disklet on a healthy disk is entangled with the failed disk is governed by the binomial distribution:

$$\text{Prob}(t=\tau) = \binom{N}{\tau}\left(\frac{1}{m}\right)^{\tau}\left(\frac{m-1}{m}\right)^{N-\tau}.$$

The proportion of entangled disklets on a disk has expected value $1/m$, but the average disk has a proportion at distance

$$\frac{\sigma}{D} = \frac{\sqrt{D\frac{m-1}{m^2}}}{D} = \frac{\sqrt{\frac{m-1}{m^2}}}{\sqrt{D}},$$

which quickly approaches zero, as we can see in Figure 4. There, we depict the average deviation of disk from optimal entanglement. By multiplying this number with three, we get bounds on the difference to optimal entanglement in which approximately 99.7% of all disks fall. Because the numbers in Figure 4 are already quite small for *D* in the mid-hundreds, we can assume this magnitude for *D*.

Operationally, the larger *D*, the more metadata needs to be kept, but also the longer it takes for a stream to fill up a disklet.

In our context more importantly, the capability of a declustered RAID Level 6 or RAID Level 6+ to survive one additional disk failure beyond their guaranteed tolerance (of two and three, respectively) depends on *D*.

If a configuration without declustering tolerates f failures with a probability of p_f, then its declustered equivalent with *D* disklets per disk survives with a probability of $p_{f,D} = {p_f}^D$. Because data-loss probabilities with $f = 3$ in the case of RAID Level 6 and

Volume 2, Issue 2, pages 91–110

p-ISSN 2655-8564, e-ISSN 2685-9432

$f = 4$ in the case of RAID Level 6+ are so low, the data-loss probability of the declustered RAID is higher than what one (or at least me) would intuitively guess.

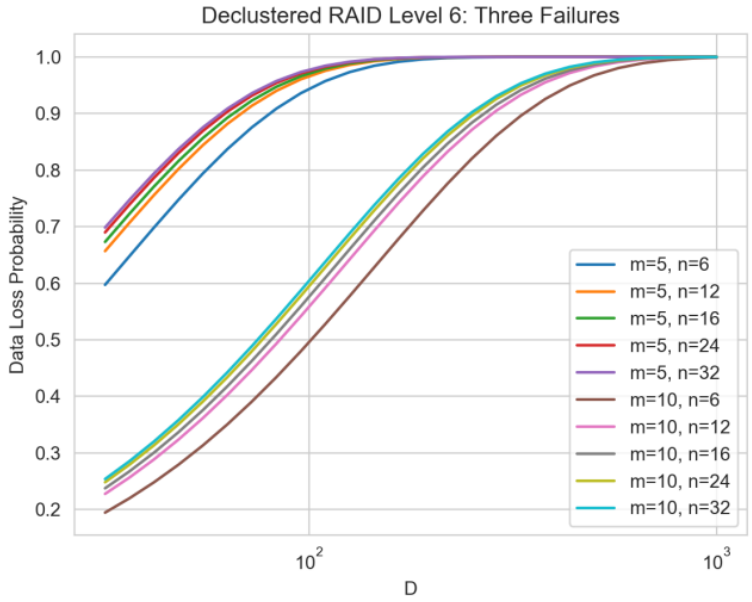


Figure 5. Dataloss probability in the presence of three failed disks in a declustered RAID Level 6 with *m* stripes and *n* data disks per stripe in dependence on *D*, the number of disklets per disk

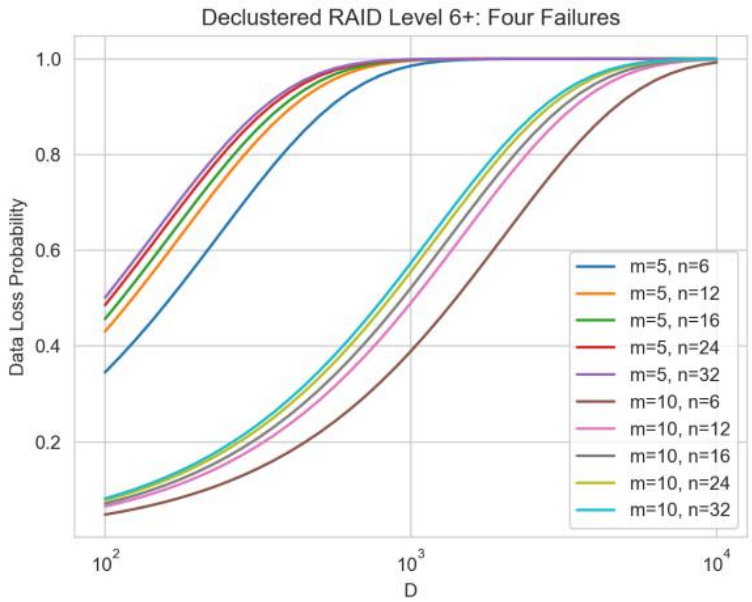


Figure 6. Dataloss probability in the presence of four failed disks in a declustered RAID Level 6+ with *m* stripes and *n* data disks per stripe in dependence on *D*, the number of disklets per disk.

International Journal of Applied Sciences and Smart Technologies
Volume 2, Issue 2, pages 91–110
p-ISSN 2655-8564, e-ISSN 2685-9432

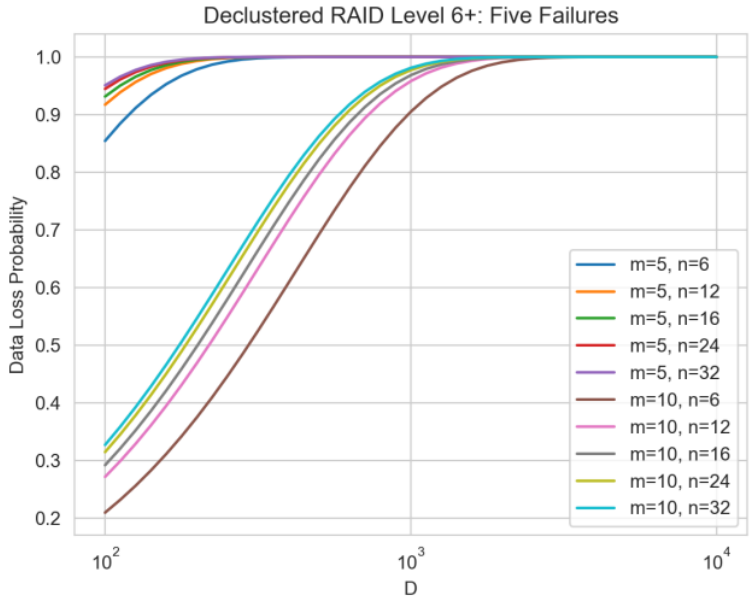


Figure 7. Dataloss probability in the presence of five failed disks in a declustered RAID Level 6+ with *m* stripes and *n* data disks per stripe in dependence on *D*, the number of disklets per disk.

As we can see from Figures 5 – 7, the data-loss probability in the declustered layout are substantial, but not zero for moderate values of *D*. This shows that a thorough reliability analysis of declustered disk arrays underestimates their reliability if it assumes that the array cannot tolerate more failures than its guaranteed tolerance, i.e. two in the case of Level 6 and three in case of Level 6+.

3 Markov Modeling

Reliability should not exclusively be assessed by using data-loss probabilities in dependence on failed disks. The true measures of reliability are Mean Time to Data-Loss (MTTDL), the survival rate of data for the economic life-span of a disk array, or the annual loss rate. We obtain these numbers using a Markov or a Semi-Markov model or even a Petri net. In this section, we give the formula for calculating transition probabilities and then apply it to calculate the MTTDL for declustered arrays using the usual simplifying assumptions.

International Journal of Applied Sciences and Smart Technologies
Volume 2, Issue 2, pages 91–110
p-ISSN 2655-8564, e-ISSN 2685-9432

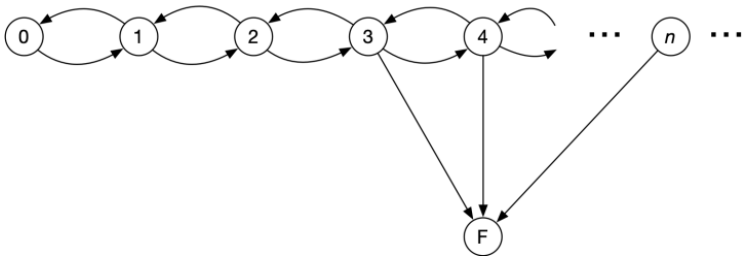


Figure 8. Generic Markov model

A. A Generic Markov Model

We should capture the current condition of a disk array in as few possible states as possible in order to simplify calculations. Often, it is possible to characterize the state of a disk array just by the current number of unavailable disks. The result is a state diagram as that presented in Figure 8. The normal, start state is State 0. There is also an absorbing state State F that characterizes a disk array that has lost data. The other states in Figure 8 represent situations where the array has suffered loss of access to a disk, but where all data stored in it can be recovered. State i is a state where i disks have failed.

We have a failure transition from State i to State $i + 1$ corresponding to the failure of an additional disk, if that failure does not lead to loss of data, and another transition from State i to State F in case that the failure has induced data-loss. If data on a disk has been recovered and saved on a spare or replacement disk, then we have a repair transition from State $i + 1$ to State i.

This model cannot capture all scenarios. For example, a disk array might have a small number of spared disks, but in case of failures, it might take a technician maybe weeks to replace the failed disks with new spare disks.

The rate at which transitions are taken in model reflect more or less reality. In the case of failures, a common first-degree approximation is the assumption of exponential failure times. In this case, the probability that a functioning disk fails at a given small time interval is constant. (Thus, the hazard rate in the lingo of reliability theory is constant.) It is however known that disk failures times are modeled with more accuracy using a Weibull distribution, or even better, using real disk lifespans [11].

Volume 2, Issue 2, pages 91–110
p-ISSN 2655-8564, e-ISSN 2685-9432

Much more problematic is the modeling of repair times as exponential repair times are hardly realistic. An exponential repair time assigns positive probabilities to physically impossible short repairs and very lengthy repairs. If technicians with normal working hours are involved, times to repair will also depend on the time that the failure occurred.

If both failures and repairs are exponentially distributed, then the state diagram of Figure 8 represents a true Markov model. For the researcher, this model is very attractive because it sometimes allows closed-form solutions for mean time to data-loss and similar measures, and failing that, simpler method for numerical solutions.

B. Calculating State Transition Probabilities

The rate with which failure transitions are taken depends on the number of disks that can fail and the probability that an additional failure will lead to data loss. In the simple case of a Markov model, these are the only contributors. We call $p_i$the probability that an additional disk failure in State i leads to data loss. (This is not the i^{th}disk failure since the system might have experienced a successful repair.)

$$p_i = \text{Prob}(\text{dataloss at } i+1 \mid \text{ no dataloss at } i).$$

If $p_{dl}(i)$ denotes the probability that the system has suffered data-loss in State i, then we have

$$p_{dl}(i+1) = p_{dl}(i) + (1-p_i)(1-p_{dl}).$$

Thus,

$$p_i = \frac{p_{dl}(i+1) - 1}{p_{dl}(i) - 1}.$$

If we assume exponential repair times that are independent of each other and exponential failure times, this gives us a Markov model with:

1) n non-failure states, where State i represents a disk array with i failed disks.
2) A failure state, State F, which is absorbing.
3) The start state is State 0.

International Journal of Applied Sciences and Smart Technologies

Volume 2, Issue 2, pages 91–110

p-ISSN 2655-8564, e-ISSN 2685-9432

4) There is a repair transition from State i to State $i-1$ taken with rate $i \cdot \rho$ and fixed "repair rate" ρ.

5) There is a failure transition not representing data-loss from State i to State $i+1$ taken with rate $(n-i) \cdot (1-p_i) \cdot \lambda$.

6) There is a failure transition representing data-loss from State i to State $i+1$ taken with rate $(n-i) \cdot p_i \cdot \lambda$.

C. MTTDL Calculation for the Markov Model

If we collect the current probabilities of being in State i at time t in an n-dimensional column vector $\mathfrak{p}(t)$and the transition rates into a transition matrix $\mathfrak{M}$, then we have the fundamental Chapman-Kolmogorov equation

$$\frac{d}{dt}\mathfrak{p} = \mathfrak{M} \cdot \mathfrak{p}.$$

The probability that a system has not suffered data-loss at time t is given by the sum of all coefficients of $\mathfrak{p}$, namely $\mathfrak{e}^t.\mathfrak{p}$ with a transposed column vector $\mathfrak{e}^t$ containing n one-coefficients. The MTTDL of the system is then given as

$$\begin{aligned} MTTDL &= \int_0^\infty t\,\frac{d\big(1-p(t)\big)}{dt}\,dt \\ &= -\int_0^\infty t\frac{dp(t)}{dt}\,dt \\ &= \lim_{t\to\infty}\big(-tp(t)\big) + \int_0^\infty p(t)\,dt \\ &= \int_0^\infty p(t)\,dt \end{aligned}$$

By applying the Laplacian transform to the fundamental differential equation, we obtain

$$s\mathfrak{L}(\mathfrak{p})(s) - \mathfrak{p}(0) = \mathfrak{M} \cdot \mathfrak{L}(\mathfrak{p})(s),$$

and after setting the new indeterminate s to zero, the definition of the Laplace transform gives us

International Journal of Applied Sciences and Smart Technologies

Volume 2, Issue 2, pages 91–110

p-ISSN 2655-8564, e-ISSN 2685-9432

$$-\mathfrak{p}(0) = \mathfrak{M} \cdot \mathfrak{L}(\mathfrak{p})(0) = \int_0^\infty \mathfrak{p}(t)dt.$$

We then multiply with e^t in order to obtain

$$MTTDL = \int_0^\infty p(t)\,dt$$

$$= \mathfrak{e}^t \cdot \int_0^\infty \mathfrak{p}(t)dt$$

$$= \mathfrak{e}^t \cdot \mathfrak{L}(\mathfrak{p})(0)$$

$$= -\mathfrak{e}^t \cdot \mathfrak{M}^{-1}\mathfrak{p}(0).$$

The value of $\mathfrak{p}(0)$ is of course $(1,0,\ldots,0)^t$ since the system starts out in State 1.

It should be noted that the calculation of the inverse is not without numerical challenges as the magnitude of repair transitions is much larger than the magnitude of failure transitions, so that matrix $\mathfrak{M}$ is ill-conditioned. However, this applies only to not using good software for numerical inversion and in many cases, just using double precision suffices for reasonable precision.

D. Example: Declustered RAID Levels

In the case of a highly declustered RAID Level 5 (one parity per stripe), a RAID Level 6 (two parities per stripe), and a RAID Level 6+ (three parities per stripe), the Markov models are fairly simple, because then any more than one, two, and three failures, respectively, have to lead to data-loss. In the case of RAID Level 5, the fundamental differential equation for the state vector made up of the probabilities of the system being in State 0 or State 1

$$\mathfrak{p}(t) = \begin{pmatrix} p_0(t) \\ p_1(t) \end{pmatrix}.$$

is simply

$$\frac{d}{dt}p_0(t) = -n\lambda p_0(t) + \rho p_1(t),$$

$$\frac{d}{dt}p_1(t) = n\lambda p_0(t) - ((n-1)\lambda p_1 - \rho p_1(t).$$

The first addend in the first equation corresponds to the failure transition, taken with rate $n\lambda$, from State 0 to State 1, the second summand to the repair transition, taken with

Volume 2, Issue 2, pages 91–110

p-ISSN 2655-8564, e-ISSN 2685-9432

rate ρ from State 1 to State 0, the first summand in the second equation to the failure transition from State 0 to State 1 seen above, the second summand to the failure transition taken with rate $(n-1)\lambda$, from State 1 to the Failure State, and the last addend to the repair transition from State 1 to State 0 again. The transition matrix is therefore

$$\mathfrak{M} = \begin{pmatrix} -n\lambda & \rho \\ n\lambda & -(n-1)\lambda - \rho \end{pmatrix},$$

so that the MTTDL is simply

$$\begin{aligned} &-(1,1) \cdot \begin{pmatrix} -n\lambda & \rho \\ n\lambda & -(n-1)\lambda - \rho \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \frac{(2n-1)\lambda + \rho}{(n-1)n\lambda} \ . \end{aligned}$$

In the case of a RAID Level 6, our transition matrix needs to capture transitions between three non-failure states and has therefore shape 3×3. In particular, we obtain the MTTDL as

$$\begin{aligned} &-(1,1,1) \cdot \begin{pmatrix} -n\lambda & \rho & 0 \\ n\lambda & -(n-1)\lambda - \rho & 2\rho \\ 0 & (n-1)\lambda & (-(n-2)\lambda - 2\rho \end{pmatrix}^{-1} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ &= \frac{(3n^2 - 6n + 1)\lambda^2 + (3n-2)\lambda\rho + 2\rho^2}{(n^3 - 3n^2 + 2n)\lambda^3} . \end{aligned}$$

In the case of the RAID Level6+ with a very high degree of declustering, we can assume that the storage system will suffer data-loss whenever four or more disks are currently not functioning. In this case, the Markov model becomes particularly easy and the MTTDL is given below, where λ is the individual disk failure rate, that is, the inverse of the expected Mean Time To Failure (MTTF) of a disk, where ρ is the repair rate, that is, the inverse of the expected Mean Time To Repair (MTTR), and n the number of disks. The MTTDL is then

$$-(1,1,1,1) \begin{pmatrix} -n\lambda & \rho & 0 & 0 \\ n\lambda & (1-n)\lambda - \rho & 2\rho & 0 \\ 0 & -(n-1)\lambda & (2-n)\lambda - 2\rho & 3\rho \\ 0 & 0 & (n-2)\lambda & (3-n)\lambda - 3\rho \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

which is given in closed form as

Volume 2, Issue 2, pages 91–110

p-ISSN 2655-8564, e-ISSN 2685-9432

$$\frac{(4n^3 - 18n^2 + 22n - 6)\lambda^3 + (6n^2 - 14n + 6)\lambda^2\rho + (8n - 6)\lambda\rho^2 + 6\rho^2}{(n^4 - 6n^3 + 11n^2 - 6n)\lambda^4}.$$

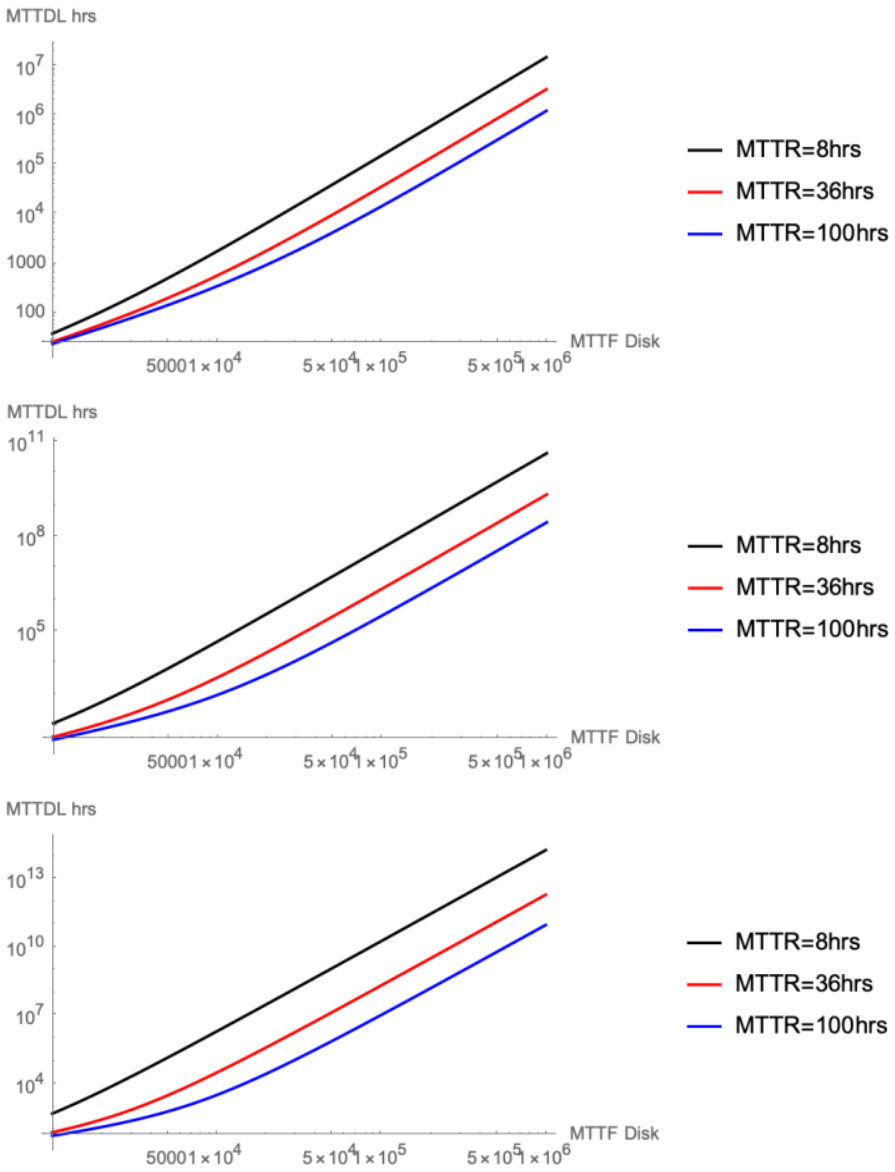


Figure 9. Mean Time to Data Loss in hours for a fully declustered Level 5 (top), Level 6 (middle) and Level 6+ (bottom) RAID in dependence on the MTTF of each of the $n = 90$ individual disks. The scales are both logarithmic.

Volume 2, Issue 2, pages 91–110
p-ISSN 2655-8564, e-ISSN 2685-9432

To illustrate this, Figure 9 gives the MTTDL in hours for an ensemble of 90 disks, for MTTF of disks ranging between 1000 hours and 1 million hours. The MTTR are 8 hours, 36 hours, and 100 hours.

E. Discussion

MTTDL figures are among the least useful figures of merit for the reliability of a disk array, but have the singular advantage of often being able to be derived exactly. Other figures of merit such as the probability to survive the economic lifespan of a disk array without data-loss are more important. The fantastic figures (a MTTDL of 100 million years! with normal disks) reflect in part the lack of modeling of catastrophic events and essential component failure and in part the implicit assumption that disks fail independently. Numerically, e.g. using the Euler method to solve the fundamental differential equation of the Markov model, we can derive numbers for the survival probability of the data in an array for 5 years.

Another, serious limitation of the current analysis is the concentration only on disk failures. However, as long as the merits of various disk array layouts are compared, this is more of a feature than a bug. With other words, even if one converts the MTTDL numbers into a failure rate per year, the numbers are not realistic, but they still serve to compare the different reliability merits of disk arrays. It is feasible to expand the Markov model to model latent sector errors and disk scrubbing, but this goes far beyond the goals of this paper. It is also possible to use Markov models in order to deal with infant disk mortality [12]. All these possibilities run soon into the difficulty of making good modeling assumptions.

Furthermore, if one is interested in actual failure rates and not in the relative merits of disk array layouts, one might have to use actual failure statistics or at least change the Markov model to a semi-Markov model in order to model the more generally applicable Weibull failure [11]. Finally, while the large-scale storage system of the foreseeable future might be disk based, the superior performance of Flash and soon PCM memories will see the expansion of this type of very fast storage. What we have learned about device failure behavior from disks might or might not transfer to calculating the reliability of these devices.

4 Conclusion

We defined two classical extensions of the RAID Level 5 and derived formulae for the capability of these disk arrays to survive a given number of disk failures. We then showed how to use these numbers in order to derive one particular figure of merit for the resilience of a disk array organized in this manner. No model reflects reality, but some are useful, as the saying goes. We have then shown by example how to use these calculations to determine the mean time to failure of various declustered RAID Levels.

RAID Level 6 and Level 6+ can serve as a reference point in order to compare the resilience resulting from using one of the many more sophisticated codes that trade off a slight increase in parity storage overhead for faster, average recoveries.

References

[1] J. Gray, personal communication.

[2] L. Bairavasundaram, G. Goodson, S. Pasupathy, and J. Schindler, "An analysis of latent sector errors in disk drives." *In SIGMETRICS Performance Evaluation Review*, **35** (1), ACM, 289–300, 2007.

[3] Bianca Schroeder, Sotirios Damouras, and Phillipa Gill, "Understanding latent sector errors and how to protect against them." *ACM Transactions on Storage (TOS)* **6** (3), 1–23, 2010.

[4] B. Schroeder and G.A. Gibson, "Understanding disk failure rates: What does an MTTF of 1,000,000 hours mean to you?" *ACM Transactions on Storage (TOS)* **3** (3), 1–16, 2007.

[5] A. Dholakia, E. Eleftheriou, X.Y. Hu, I. Iliadis, J. Menon, and K. K. Rao, "A new intra-disk redundancy scheme for high-reliability RAID storage systems in the presence of unrecoverable errors." *ACM Transactions on Storage* (*TOS*) **4** (1), 1–42, 2008.

[6] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures." *IEEE Transactions on computers* **44** (2), 192–202, 1992.

Volume 2, Issue 2, pages 91–110

p-ISSN 2655-8564, e-ISSN 2685-9432

[7] J.F. Pâris, A. Amer, and T. Schwarz, "Low-redundancy two-dimensional RAID arrays." *International Conference on Computing, Networking and Communications (ICNC)*, 507–511, IEEE, 2012.

[8] P. Chen, E. Lee, G. Gibson, R. Katz, and D. Patterson, "RAID: High-performance, reliable secondary storage." *ACM Computing Surveys* (*CSUR*) **26** (2), 145–185, 1992.

[9] T. Schwarz and W. Burkhard, "Reliability and performance of RAIDs." *IEEE Transactions on Magnetics* **31** (2), 1161–1166, 1995.

[10] M. Holland and G.A. Gibson, "Parity declustering for continuous operation in redundant disk arrays." *ACM SIGPLAN Notices* 27, Proceedings of ASPLOS **92** (9), 23–35, 1992.

[11] J. Elerath and M. Pecht. "A highly accurate method for assessing reliability of redundant arrays of inexpensive disks (RAID)." *IEEE Transactions on Computers* **58** (3), 289–299, 2008.

[12] Q. Xin, Q., T. Schwarz, and E. Miller. "Disk infant mortality in large storage systems." In *13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, 125–134, 2005.

International Journal of Applied Sciences and Smart Technologies

Volume 2, Issue 2, pages 91–110

p-ISSN 2655-8564, e-ISSN 2685-9432

This page intentionally left blank