

# MY WEIRD PROMPTS

Podcast Transcript

EPISODE #409

## RAID Demystified: Speed, Safety, and Data Survival

Published February 01, 2026 • Runtime: 23:06

<https://myweirdprompts.com/episode/raid-storage-redundancy-explained/>

### EPISODE SYNOPSIS

In this technical deep dive, Herman and Corn explore the complex world of Redundant Arrays of Independent Disks (RAID). Inspired by a listener's "Frankenstein" workstation setup, the duo breaks down the trade-offs between performance, capacity, and redundancy across RAID 0, 1, 5, 6, and 10. They go beyond the basics to discuss the terrifying reality of rebuild times on modern 30TB drives, the mathematical magic of XOR parity, and why software RAID and ZFS have overtaken traditional hardware controllers. Whether you're a video editor looking for speed or a sysadmin guarding against bit rot, this episode provides the essential engineering insights to keep your data from vanishing into the void.

### DANIEL'S PROMPT

#### Daniel

Herman and Corinne, let's talk about RAID. It's an interesting subject often associated with high-end computing and servers, but it can also be used in workstations. I'm curious about the different types of RAID, the mathematics behind how they work, and if they really work as well as expected during a drive failure. What are the differences between physical and software RAID, and what are the trade-offs regarding performance and capacity?

# TRANSCRIPT

## Corn

Alright, we are back. I hope everyone is having a good week. We have got a real technical deep dive today, something that takes me back to my first custom PC builds, but with some very modern twists.

## Herman

Herman Poppleberry here, and I am ready. This is one of those topics where the more you know, the more you realize how much engineering goes into just keeping our data from vanishing into the void.

## Corn

Exactly. Our housemate Daniel actually sent this one in. He was talking about his current workstation setup here in Jerusalem. He is running this mix of N-V-M-e and Sata S-S-Ds, which he admits is a bit of a Frankenstein's monster, and it got him thinking about RAID.

## Herman

I love that Daniel is just throwing whatever drives he has into a pool. It is brave, but it is also the perfect starting point for this conversation. He wants to know about the different types of RAID, the math behind it, and whether it actually saves your skin when a drive dies.

## Corn

And that is the big question, right? Because RAID is one of those things that sounds like magic until you are sitting there watching a rebuild bar crawl across the screen for forty-eight hours. But before we get to the horror stories, let us lay the groundwork. Herman, for the uninitiated, what are we actually talking about when we say RAID?

### Herman

So, RAID stands for Redundant Array of Independent Disks. It used to stand for Redundant Array of Inexpensive Disks back in the late eighties when it was first conceptualized at Berkeley, but the industry shifted the naming. The core idea is simple: you take multiple physical hard drives or solid state drives and you combine them into one logical unit. To your operating system, it looks like one giant, fast, or reliable drive, but underneath, the RAID controller is doing a lot of heavy lifting to distribute data.

### Corn

Right, and it is all about those three pillars: performance, capacity, and redundancy. Usually, you have to trade one to get the others. Let us start with the most basic one, even though it is technically not redundant at all. RAID zero.

### Herman

RAID zero is the speed demon. We call it striping. If you have two drives, the controller splits every piece of data in half and writes one half to drive A and the other half to drive B simultaneously.

### Corn

So, in theory, you are doubling your write and read speeds because you are using two lanes of traffic instead of one.

### Herman

Exactly. The math is simple. Your total capacity is the sum of all drives, and your performance scales linearly with the number of drives, minus a tiny bit of overhead. But here is the catch, and it is a big one. If you have four drives in RAID zero and just one of them fails, you lose everything. Every single file is effectively shredded because half of its bits are on a dead drive.

### Corn

I always tell people RAID zero is called RAID zero because that is how much data you have left when a drive fails. It is great for temporary scratch space, like video editing cache, but never for anything you care about. Now, on the flip side, you have RAID one.

### Herman

RAID one is mirroring. It is the purest form of redundancy. You have two drives, and every single bit written to drive A is also written to drive B. If drive A dies, drive B just keeps humming along. The system does not even blink.

### Corn

But the trade-off there is capacity. If I buy two two-terabyte drives, I only have two terabytes of usable space. I am essentially paying double for my storage just for the peace of mind.

### Herman

That is the tax you pay. But for a workstation, RAID one is fantastic for your boot drive. It is simple, there is no complex math involved, and the read speeds can actually be faster because the controller can pull data from both drives at once, almost like RAID zero, though write speeds stay the same as a single drive.

### Corn

Okay, so zero is for speed, one is for safety. But Daniel was asking about setups with four or five drives. That is where we get into the more "magical" math of RAID five and RAID six. This is where my brain usually starts to sweat a little. How do you get redundancy without losing half your space?

### Herman

This is where we talk about parity. RAID five is probably the most famous server configuration. You need at least three drives. Let us say you have three drives. The system stripes data across two of them, but on the third, it stores parity information.

### Corn

And parity is basically a mathematical summary of the data on the other drives, right?

### Herman

Precisely. It uses the X-O-R operation, which stands for Exclusive Or. For the listeners who remember their logic gates, X-O-R is a bitwise operation where the output is true if exactly one of the inputs is true. In the context of RAID, if you have data bit A and data bit B, you X-O-R them to get parity bit P.

### Corn

And the magic is that if you lose A, you can calculate it by X-O-R-ing B and P.

### Herman

Exactly! The math is reversible.  $A \text{ X-O-R } B \text{ equals } P$ .  $B \text{ X-O-R } P \text{ equals } A$ .  $A \text{ X-O-R } P \text{ equals } B$ . It is beautiful. In a RAID five array, the parity is not just on one drive, though. It is rotated across all the drives in the array. This is called distributed parity. If any single drive fails, the remaining drives use the parity bits to reconstruct the missing data in real time.

### Corn

So if I have four four-terabyte drives in RAID five, what is my actual usable capacity?

### Herman

The formula is  $N \text{ minus one times the capacity of the smallest drive}$ . So, four drives minus one is three. Three times four terabytes is twelve terabytes of usable space. You only "lose" the capacity of one drive to parity, but you can survive one failure.

### Corn

That sounds like a great deal. You get the speed of striping and the safety of mirroring but with a much lower capacity penalty. But I know you have some caveats here, Herman. Specifically about what happens when a drive actually fails in RAID five.

### Herman

This is the reality check Daniel was asking about. RAID five was amazing when drives were nine gigabytes. But today, in early twenty-twenty-six, we are seeing thirty-terabyte H-A-M-R mechanical drives. When a drive fails in a RAID five array of that size, the array enters what we call a degraded state. It is still working, but every time you read data, the controller has to do that X-O-R math on the fly to recreate the missing pieces. Performance tanks.

### Corn

And then you put in a new drive to replace the dead one, and the rebuild starts.

### Herman

And that is the danger zone. To rebuild that new drive, the controller has to read every single bit on every other drive in the array. On a thirty-terabyte drive, that could take days or even a week. This puts immense stress on old drives that are likely from the same manufacturing batch as the one that just died. If a second drive fails during that week-long rebuild, the whole array is toast.

### Corn

There is also the issue of Unrecoverable Read Errors, or U-R-Es. I remember reading that with modern high-capacity drives, the mathematical probability of hitting a read error during a multi-terabyte rebuild is actually quite high.

### Herman

It is terrifyingly high. If you hit a U-R-E on a healthy drive during a rebuild, the controller might not know how to finish the reconstruction, and you end up with a hole in your data. This is why many professionals have moved away from RAID five for large arrays and gone to RAID six.

### Corn

RAID six is just RAID five with an extra layer of protection, right?

### Herman

Yes, it uses double parity. It can survive two simultaneous drive failures. The math is more complex, using Reed-Solomon coding or Galois field theory instead of just simple X-O-R, but the result is that you can lose two drives and still be fine. You lose the capacity of two drives, but in a world of thirty-terabyte disks, that is a price many are willing to pay.

### Corn

I think it is important to mention RAID ten as well, because for workstations, that is often the gold standard if you have the budget.

### Herman

RAID ten is a nested level. It is a stripe of mirrors. You take two drives and mirror them, then take another two drives and mirror them, and then you stripe across those two pairs.

### Corn

So you get the massive performance boost of RAID zero and the security of RAID one.

### Herman

Exactly. It is very fast and very resilient. You can technically lose up to half your drives as long as you do not lose both drives in a specific mirror pair. Rebuilds are also much faster because you are just copying data from the surviving mirror, not doing complex parity calculations across the whole array.

### Corn

Let us talk about the physical versus software side of this. Daniel mentioned he is running an Ubuntu machine. In the old days, you had to buy a dedicated RAID controller card with its own processor and battery-backed cache. Is that still the case?

### Herman

Not really. For most users, software RAID is actually superior now. In the nineties, C-P-U-s were weak, so offloading the X-O-R math to a dedicated chip made sense. Today, your C-P-U is so fast that it can handle RAID calculations without breaking a sweat.

### Corn

Plus, if your physical RAID card dies, you often have to find the exact same model of card to get your data back. That is a single point of failure that people forget about.

### Herman

That is a huge point. If you use software RAID, like M-D-A-D-M on Linux or Z-F-S, you can take those drives, plug them into a completely different computer, and the software will recognize the array immediately. It is much more portable.

### Corn

You mentioned Z-F-S. We should probably explain why that is different from traditional RAID. Because it is not just RAID, it is a file system and a volume manager all in one.

### Herman

Z-F-S is the gold standard for data integrity. Traditional RAID is "dumb" in a way. It does not know what is a file and what is empty space; it just sees blocks of data. Z-F-S is "aware." It uses checksumming on every single block of data. If a bit flips on your hard drive, which happens more often than you would think due to cosmic rays or hardware degradation, a traditional RAID controller might just pass that corrupted data to the O-S.

### Corn

That is what they call "silent data corruption" or "bit rot."

### Herman

Right. But Z-F-S checks the data against the checksum. If they do not match, Z-F-S says, "Wait, this is wrong," and it automatically pulls the correct data from the parity or the mirror and heals the corrupted block. It is self-healing storage. It is incredible.

### Corn

So for Daniel, who is running Ubuntu, Z-F-S is definitely something he should look into, especially since he is mixing drives. Although, Z-F-S generally prefers drives of the same size.

### Herman

Yes, that is a universal RAID rule. Your array is only as big as your smallest drive multiplied by the number of drives. If Daniel has three one-terabyte S-S-Ds and one five-hundred-gigabyte S-S-D, a RAID array will treat all of them as five-hundred-gigabyte drives. He would be throwing away a lot of capacity.

### Corn

It is like a convoy. You can only go as fast as the slowest ship. And in RAID, you can only be as big as the smallest disk.

### Herman

Exactly. Now, Daniel also asked about the performance trade-offs. This is where it gets interesting with S-S-Ds. Back when we used spinning platters, RAID was essential to get decent speeds. But a single modern P-C-I-e Gen five N-V-M-e drive can do fourteen thousand megabytes per second. Do we even need RAID for performance anymore?

### Corn

That is a great question. For most people, no. A single Gen five N-V-M-e drive is faster than almost any Sata RAID array you could build. But if you are doing high-end video editing, like working with uncompressed eight-K footage, or if you are running massive databases, you might still want to stripe those N-V-M-e drives.

### Herman

There is a catch with N-V-M-e RAID, though. Often, the bottleneck is no longer the drives; it is the P-C-I-e bus or the C-P-U overhead. You might stripe four N-V-M-e drives and find that you are not getting four times the speed because you have hit the limit of how much data the processor can move through the memory bus. Plus, those Gen five drives get incredibly hot; putting four of them together requires some serious thermal management.

### Corn

And there is the latency issue. RAID controllers, especially older ones, can actually add a tiny bit of latency to every operation. With mechanical drives, you did not notice because the seek time of the drive was so slow. But with S-S-Ds that have microsecond latency, the overhead of the RAID logic can actually make the system feel slightly less responsive in some specific tasks.

### Herman

That is why for most workstations, I recommend RAID one for the O-S and maybe a large, single N-V-M-e for your current projects, with a separate RAID array for long-term storage or backups.

### Corn

Let us talk about the "does it work as expected" part of Daniel's question. Because I think there is a huge misconception that RAID is a backup. Herman, how many times have we seen people lose data because they thought RAID was enough?

### Herman

It is the number one mistake in computing. RAID is about uptime, not backups. If you accidentally delete a file, RAID will faithfully delete it from all your drives simultaneously. If a power surge fries your power supply and sends a spike into your drives, it can kill all of them. If ransomware encrypts your files, RAID will happily store the encrypted versions.

### Corn

There is a saying: RAID protects you against a drive failure, but a backup protects you against everything else.

### Herman

Exactly. And even the drive failure protection is not a guarantee. We talked about the rebuild stress. I have seen arrays where one drive fails, and during the rebuild, the heat and vibration of the intensive reading cause a second drive to fail. Suddenly, your "redundant" system is a brick.

### Corn

I think one thing that often surprises people is the "Write Hole" in RAID five and six. Can you explain that? It sounds like something out of a sci-fi movie.

### Herman

It is a bit of a nightmare scenario. Imagine the system is writing data and the parity bit to the disks. The data is written to drive A, but before the parity bit can be written to drive B, the power goes out. Now your parity is out of sync with your data. When the power comes back on, the controller has no way of knowing that the parity is wrong. If a drive fails later, the controller will use that "bad" parity to reconstruct data, and you will end up with corrupted files.

### Corn

This is why hardware RAID cards have those little battery packs, right? To finish the writes if the power cuts.

### Herman

Exactly. Or, if you use software like Z-F-S, it uses a copy-on-write mechanism that effectively eliminates the write hole by never overwriting data in place. But for traditional RAID five on a cheap motherboard controller, the write hole is a very real risk unless you have an Uninterruptible Power Supply, or U-P-S.

### Corn

Which, honestly, if you are running a RAID array, you should have a U-P-S anyway. It is the best fifty or one hundred dollars you can spend to protect your hardware.

### Herman

Absolutely. Especially here in Jerusalem, where we get those occasional winter power flickers.

### Corn

Oh, tell me about it. Every time the wind blows too hard, I am checking my server status. So, Daniel's current setup is a mix of N-V-M-e and four Sata S-S-Ds. If he wanted to make the most of that, what would you suggest?

### Herman

Since he is on Ubuntu, I would tell him to look at Z-F-S or maybe B-tr-f-s. He could put the four Sata S-S-Ds into a RAID ten equivalent. That would give him great speed and very high reliability. He would lose half the capacity, but for S-S-Ds, the rebuild would be incredibly fast, which minimizes the danger window.

### Corn

And the N-V-M-e should probably stay as a separate boot and scratch drive. Mixing N-V-M-e and Sata in the same RAID array is usually a bad idea because the whole array will be limited by the slower Sata speeds and higher latency.

### Herman

Right. It would be like putting a Ferrari and three tractors into a relay race. The Ferrari is going to spend most of its time waiting for the tractors to finish their laps.

### Corn

That is a great analogy. Now, let us look at the future a bit. We are seeing things like N-V-M-e over Fabrics and specialized storage controllers that handle data at a hardware level in ways that make traditional RAID look like a dinosaur. Do you think RAID has a place in five or ten years?

### Herman

I think the "concept" of redundancy will always be there, but the way we do it is changing. In large data centers, they are moving away from RAID and toward "Erasure Coding." It is like RAID five or six but much more flexible. You can distribute data across hundreds of servers, not just drives. You could lose an entire rack of servers and not lose any data.

### Corn

It is basically RAID at the network level.

### Herman

Exactly. For the individual user, I think we will see more "intelligent" storage where the operating system just manages a pool of drives and you tell it, "I want this folder to be mirrored and this folder to be fast," and it handles the block distribution behind the scenes.

### Corn

That sounds a lot like what Apple does with their Fusion drives, or what Windows does with Storage Spaces, though both of those have had their share of growing pains.

### Herman

Storage Spaces is actually quite powerful now, but it still lacks some of the robust "self-healing" features that Z-F-S has. For a pro workstation, I still think a dedicated Linux-based storage server or a high-end N-A-S is the way to go.

### Corn

Okay, so to summarize for Daniel and everyone else: RAID zero for speed you do not mind losing. RAID one for simple safety. RAID five for a balance of space and safety, but be careful with large drives. RAID six if you want to sleep better at night. And RAID ten if you want it all and can afford the disk tax.

### Herman

And never, ever forget that RAID is not a backup. If you do not have your data in two different physical locations, you do not really own that data. You are just borrowing it from fate.

### Corn

That is a bit dark, Herman Poppleberry, but it is the truth. Fate has a way of calling in those loans at the worst possible time.

### Herman

Usually at three in the morning when you have a deadline at eight.

### Corn

Exactly. Well, I think we have covered the basics and then some. This was a fun one. It is rare that we get to talk about X-O-R logic and file system architecture in the same breath.

### Herman

It is the heart of the machine, Corn. It is where the math meets the metal.

### Corn

Before we wrap up, I want to remind everyone that if you are finding these deep dives useful, or if you just like hearing two brothers talk about tech in Jerusalem, please leave us a review on your podcast app or on Spotify. It really helps the show find new listeners who share our particular brand of nerdiness.

### Herman

It really does. And check out our website at [myweirdprompts.com](http://myweirdprompts.com). We have the full archive there, and there is a contact form if you want to send us a prompt like Daniel did. We love getting into the weeds on these topics.

**Corn**

Definitely. We might have touched on some of these themes in our older episodes about data privacy or hardware history, so if you are interested, the website has a searchable archive for you to explore.

**Herman**

Thanks to Daniel for the prompt. I hope your Frankenstein workstation stays healthy, buddy.

**Corn**

Yes, keep those drives spinning, or, well, keep those electrons flowing in the case of the S-S-Ds.

**Herman**

This has been My Weird Prompts. Thanks for listening.

**Corn**

We will catch you in the next one. Peace.