8 Jan 2021

Introduction

I became overly concerned about data integrity on my computer system over Christmas in 2019. Perhaps some of it was due to a bit of available time on my hands, but nevertheless, I deep-dove into the subject. My rambling comments during those investigations is collected in the Appendix, but the final plan I ended up implementing is discussed in the pages which immediately follow.

Final Game Plan Regarding Data Integrity

After studying the issue of data integrity for myself over the past week or so, I have concluded the following points:

- The nominal write-error rate for most hard-disk drives has remained about 1 out of 10^{14} for each bit written for quite some time (e.g, > 10 years). This amounts to an average error rate of 1 unrecoverable bit-error out of every 12.5 TB written. While this error rate may have been acceptable when disk drives were several hundred GB, this is not acceptable (to me) with the TB-level drives of today and the future.
- Every day I use my regular hard-drive, I invite incurring this error rate and difficult-to-quantify *bit-rot*.
- As I retain photos, videos, and other source materials of my own creation for decades, the likelihood of incurring issues likely goes up faster than linear-time.
    - The forward error correction methods in play make use of the corrected error being representable as

$p_{unrecoverable} = sum\ p^n$

but p is actually an increasing function of time.

- Other than for complete hard-disk (or SSD) loss, it serves little purpose to use *perfect* back-up services to preserve my source materials <u>including</u> the errors & contaminations introduced when I touch the source information on my plain vanilla hard-drives.
    - In other words, it is silly to attempt introducing data integrity *after* the source information has already been potentially damaged.
- Therefore, unless I maintain the high level of data integrity I want from the very outset of the source material creation and from that point in time forward, data integrity is basically an illusion.
- Data integrity will never be better than at the point in time of its creation (i.e., entropy always decreases with time rather than decreases)

With respect to preserving data integrity from the very outset:

- Anyone storing or editing materials on a regular PC having a simple hard-drive for storage is likely the primary worst source of data integrity loss.
- If I want a data integrity level better than 1 bit out of 12.5 TB, every save to data storage must also be better.

It was a real revelation to me that although better data integrity performance is mathematically possible with consumer-grade RAID storage approaches, it is generally never done:

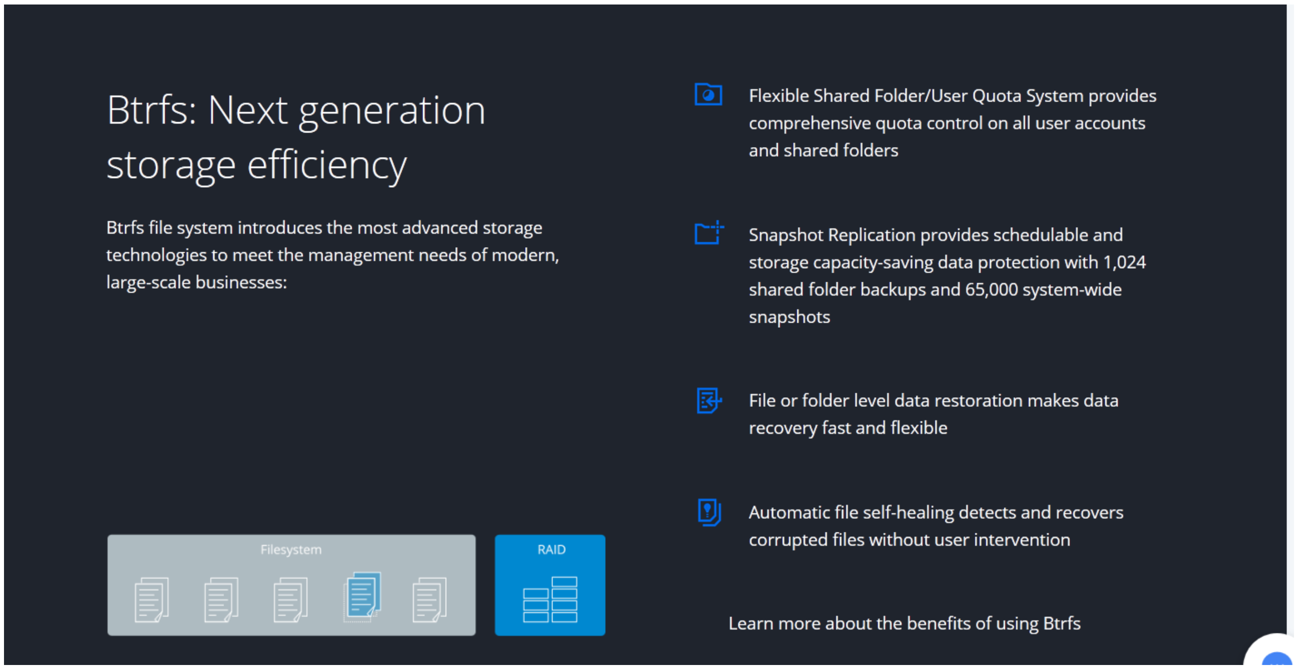
> *Nobody reads the checksums on reads. There's no point. Reading a checksum means you <u>have to read the entire slice plus checksum, then compute the checksum to verify you have the correct data</u>. <u>Plus the orthoganal checksum if you are running RAID-6</u> or whatever. It is a total performance killer because it breaks the ability to randomly seek to totally different sectors on different disks at the same time. Similarly, almost nobody reads both sides of a mirror in RAID-1 because if you only read one side you can alternate which side of the mirror you read from so that you get faster throughput, and if you suddenly have a mismatch, which disk do you take as correct and which do you take as broken? All modern RAID systems depend on the on-disk controllers to signal the RAID controller that they are in distress (through SMART or the like), at which point that disk is almost always kicked out of the array.* ***Checksums are used for rebuilding arrays, not for read-verification.***

What seemed quite ironic to me was that if the probability of a single unrecoverable bit error on one drive is p, I could not find anyone explicitly saving identical copies to 3 hard-drives in parallel and then performing a majority-vote of the 3 drives for every subsequent read operation. This would reduce the probability of an unrecoverable bit error to 3 p^2 which would be an astronomically good improvement (in the present content, 1 bit out of every million-billion-terabytes written!!!).

I found many references suggesting that the ZFS file system method is doing things correctly, but these always run on Unix-based systems and are very expensive. And I also knew I didn't want to implement my own approach because others had unquestionably solved this problem before me!

Buried in the specification sheets of multiple products is the acronym Btrfs (for B-tree file system). None of the product datasheets I encountered spent an iota of time on explaining the powerful concepts behind this acronym, features of which I had been hunting for! In short, this file system continually makes use of the stored parity and multiple-copy information available to scrub the stored data and maintain its integrity plus a whole lot more. Btrfs is the default file system used in modern distributions of Linux since about 2012. The details behind Btrfs go way beyond my interest level, but appear to fit my needs.

I have subsequently decided to purchase the Synology DS918+ NAS Server and populate it with 4 highly reliable SSD's (Samsung SSD 860 EVO 2TB). It is difficult to get bit error rate estimates for the individual SSD's but my reading suggests they are 100x to even 1000x more reliable than a standard hard-disk drive thereby driving the probability p used earlier to levels on the order of 10^{-16} or better. Put together when the data scrubbing and data redundancy I plan using with the DS918+, any concern over bit-rot I've had in the past will be obliterated.



Btrfs: Next generation storage efficiency

Btrfs file system introduces the most advanced storage technologies to meet the management needs of modern, large-scale businesses:

Flexible Shared Folder/User Quota System provides comprehensive quota control on all user accounts and shared folders

Snapshot Replication provides schedulable and storage capacity-saving data protection with 1,024 shared folder backups and 65,000 system-wide snapshots

File or folder level data restoration makes data recovery fast and flexible

Automatic file self-healing detects and recovers corrupted files without user intervention

Learn more about the benefits of using Btrfs

Figure 1 From Synology website

Figure 2 Synology DiskStation DS918+ NAS Server

Appendix: Early Ramblings of Late 2019

Computer Backups & Data Reliability

22 Dec 2019

The last couple of days I've been rather consumed thinking about my storage backup methodology and data integrity. This has been a painful adventure.

With the advent of the terabyte drives, the need for true data reliability is acute. Turns out most disk manufacturers target an *unrecoverable bit error rate* of 10^{14} which is about 12 TB. So on average, you can expect to lose a bit for every 12 TB read. That's not super great in my book in the context of 2 TB drives.

The RAID landscape is even more murky. RAID 5 uses one parity bit distributed across all of the disks used. I've repeatedly read that the biggest problem with RAID 5 is that when one disk is deemed bad, the remaining disks are worked extremely hard to rebuild a new replacement disk, and in doing so, a second disk frequently fails. So then comes along RAID 6 which uses two parity bits!

I've had some difficulty seeing how disk reliability versus bit-level reliability works. I don't believe they are the same.

I ran into repeated cautions about difficulties if a hardware RAID controller itself fails. Unless you can replace the controller with an exact twin, you will not be able to get your information back from the RAID storage! This argues in favor of something more like simple mirroring.

One thread for DIY systems I've run on to is FreeBSD, but I don't have it in me to start digging into UNIX:

Definition - What does FreeBSD mean?

FreeBSD is a free, open-source, Unix-like operating system based on Berkeley Software Distribution (BSD) Unix. It is the most popular among the BSD-based operating systems, with an installed base of more than 75%. Due to legal constraints, FreeBSD cannot be labeled as a Unix system, although it is compliant with Unix internals and application programming interfaces (APIs). FreeBSD's licensing terms give developers a high degree of freedom to reuse it, so other operating systems (such as MAC OSX) have reused many FreeBSD codes. Although FreeBSD is not categorized as Unix, MAC OSX does have a formal Unix branding.

FreeBSD is used by millions of users all over the world, as well as some of the busiest sites on the Internet such as Yahoo, Sony Japan, etc.

Learning about FreeBSD-derived projects

FreeBSD is widely used as a building block for other commercial and open-source operating systems. Some of the most widely used and publicly available systems are listed below.

- FreeNAS is a storage solution that can be installed on virtually any hardware platform to share data over a network. It uses ZFS to protect, store, backup, all of your data.
- TrueOS is a FreeBSD derivative with a graphical installer and impressive desktop tools aimed at ease of use for the casual computer user.
- pfSense is a FreeBSD based network security solution. pfSense software, with the help of the package system, is able to provide the same functionality or more of common commercial firewalls, without any of the artificial limitations. It has successfully replaced every big name commercial firewall you can imagine in numerous installations around the world.

https://www.trueos.org/

Wikipedia Entry for RAID:

Unrecoverable read errors during rebuild

Unrecoverable read errors (URE) present as sector read failures, also known as *latent sector errors* (LSE). The associated media assessment measure, *unrecoverable bit error* (UBE) rate, is typically guaranteed to be less than one bit in 10^{15} for enterprise-class drives (SCSI, FC, SAS or SATA), and less than one bit in 10^{14} for desktop-class drives (IDE/ATA/PATA or SATA). Increasing drive capacities and large RAID 5 instances have led to the maximum error rates being insufficient to guarantee a successful recovery, due to the high likelihood of such an error occurring on one or more remaining drives during a RAID set rebuild.[12][80] When rebuilding, parity-based schemes such as RAID 5 are particularly prone to the effects of UREs as they affect not only the sector where they occur, but also reconstructed blocks using that sector for parity computation.[81]

Double-protection parity-based schemes, such as RAID 6, attempt to address this issue by providing redundancy that allows double-drive failures; as a downside, such schemes suffer from elevated write penalty—the number of times the storage medium must be accessed during a single write operation.[82] Schemes that duplicate (mirror) data in a drive-to-drive manner, such as RAID 1 and RAID 10, have a lower risk from UREs than those using parity computation or mirroring between striped sets.[25][83] Data scrubbing, as a background process, can be used to detect and recover from UREs, effectively reducing the risk of them happening during RAID rebuilds and causing double-drive failures. The recovery of UREs involves remapping of affected underlying disk sectors, utilizing the drive's sector remapping pool; in case of UREs detected during background scrubbing, data redundancy provided by a fully operational RAID set allows the missing data to be reconstructed and rewritten to a remapped sector.[84][85]

From: https://www.partitionwizard.com/clone-disk/windows-10-storage-spaces-vs-raid.html

About Windows 10 Storage Spaces

Storage Spaces is a Windows built-in technology. Users can utilize this technology to group multiple drives together into a storage pool and then use the capacity of the pool to create virtual drives named storage spaces. This technology can help users to protect their data from drive failure through mirroring data and extend storage space through adding drivers to PC.

How to use Windows 10 storage spaces? Here is a tutorial.

Step 1: Create a storage pool.

- Type "manage storage spaces" in Windows search box.
- Click Manage Storage Spaces button to open it.
- Click "Create a new pool and storage space".

Note: When users create a storage pool with formatted drives, data loss will occur. Please back up data in advance, or prepare for data recovery.

Step 2: Create storage space.

- Choose file system: NTFS or ReFS.
- Choose resiliency type: Simple (no resiliency), Two-way mirror, Three-way mirror, or Parity.

Attention:

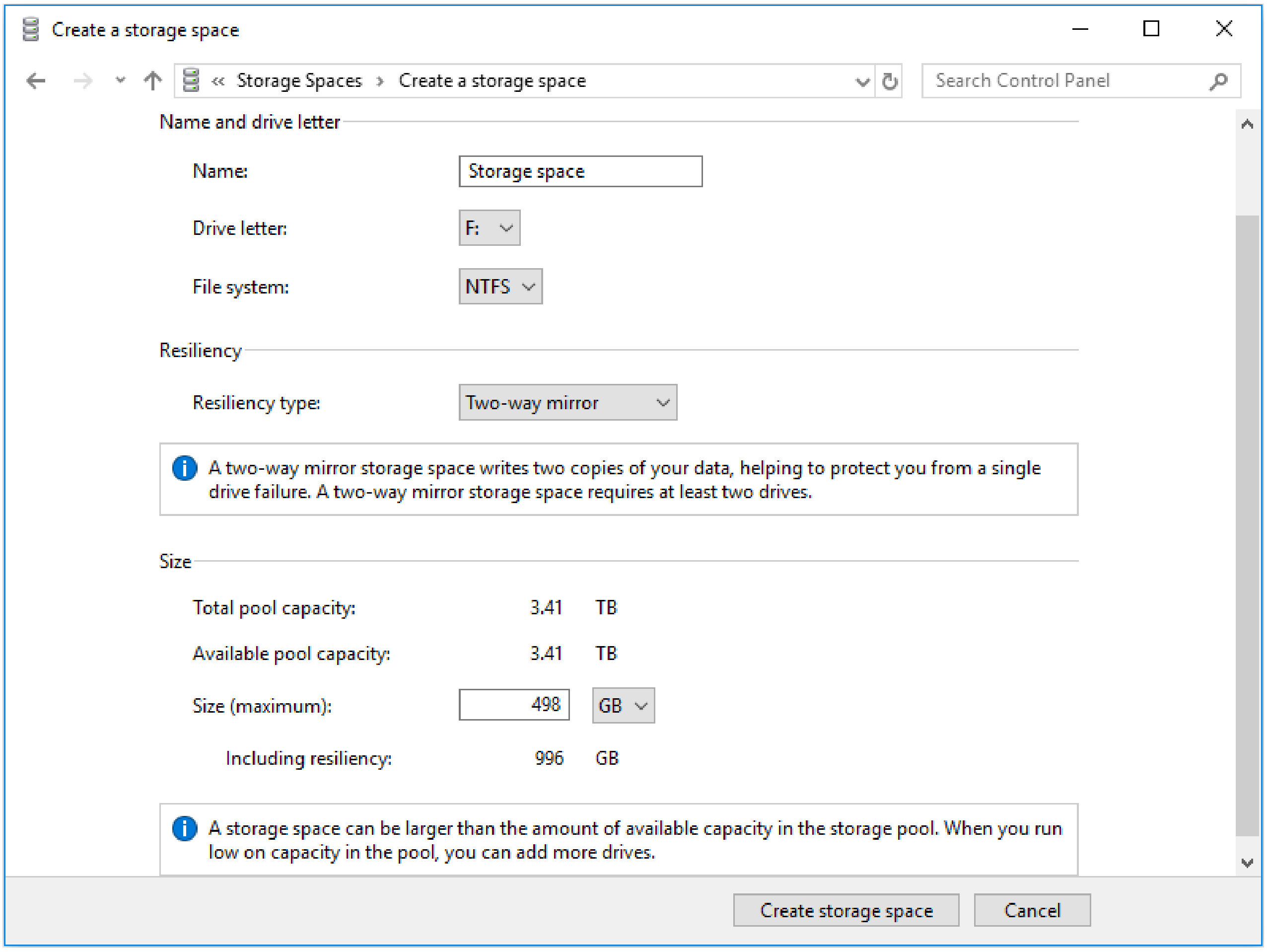
ReFS (Resilient File System): It is compatible with most of NTFS. It can verify data corruption automatically and try to recover it. Besides, the combination of ReFS and Windows 10 Storage Spaces can provide better data protection. At the same time, there are performance improvements in dealing with hundreds of millions of files.

Simple (no resiliency): This type needs at least one drive and will write one copy of data. So, it won't protect data from drive failure (click to know how to recover data from a failed hard drive). However, similar to RAID 0, it can improve read/write performance through adding drives.

Two-way mirror: It needs at least two drives and will write two copies of data and it will protect data from a single drive failure. It is similar to RAID 1.

Three-way mirror: It needs at least five drives and will write three copies of data. The usable space is a third of the total capacity. But it will protect data from two simultaneous drive failures.

Parity: It needs at least three drives and writes data with parity information. It will protect data from a single drive failure. It is similar to RAID 5.



Name and drive letter

Name:	Storage space
Drive letter:	F:
File system:	NTFS

Resiliency

Resiliency type:	Two-way mirror

A two-way mirror storage space writes two copies of your data, helping to protect you from a single drive failure. A two-way mirror storage space requires at least two drives.

Size

Total pool capacity:	3.41	TB
Available pool capacity:	3.41	TB
Size (maximum):	498	GB
Including resiliency:	996	GB

A storage space can be larger than the amount of available capacity in the storage pool. When you run low on capacity in the pool, you can add more drives.

Windows 10 Storage Spaces vs. RAID

Actually, Windows 10 storage spaces feature can also realize software RAID to some extent. However, the Windows Storage Spaces has some difference from RAID and we can't say that it is inferior to RAID. Let's see the difference of Windows 10 Storage Spaces vs. RAID.

Besides, Windows 10 Storage Spaces has ReFS, which can make users free from trouble of data verification.

Samsung 860 Pro 2 TB $468

This is the ultimate choice for users who want both high endurance and exceptional speeds. It is ideal for heavy workloads with high-end machines. Heavy IT users find Samsung 860 Pro’s performance perfect for their work. It comes with a 600 terabytes written (TBW) or a 10-year warranty. With AES 256 – bit engine encrypted on the hardware to secure your data and standards of TCG Opal v. 20 complied with. Also, the presence of Dynamic Thermal Hardware helps prevent overheating.

With 3D V-NAND technology, Samsung 860 Pro provides world – class read and write execution in both sequential as well as random tasks. Again, just like all Samsung components, 860 Pro is fine tuned in every stage of manufacturing to ensure perfect and consistent performance. This enhances its functioning and reduces power usage. The Samsung 860 Pro is ideal for upgrading, as the best ssd for macbook pro mid 2012 to perform large workloads. It was categorized as the best hdd for gaming 2017.

Samsung 850 EVO 1 TB

This is a remarkable choice, especially for large capacity users. The 1TB storage has even more endurance, and the capacity to write from 300 terabytes of data. It has Samsung Data Migration together with Magician software included to ease installation. It is compatible with many common operating systems, besides being particularly energy efficient. Samsung 850 Evo SSDs have a longer warranty and a reliable Samsung customer service. This brand featured as having the best 1tb ssd drive price in india, and the best budget ssd 120gb plus best 120gb ssd. It's also the cheapest ssd india and was the best ssd india 2016.

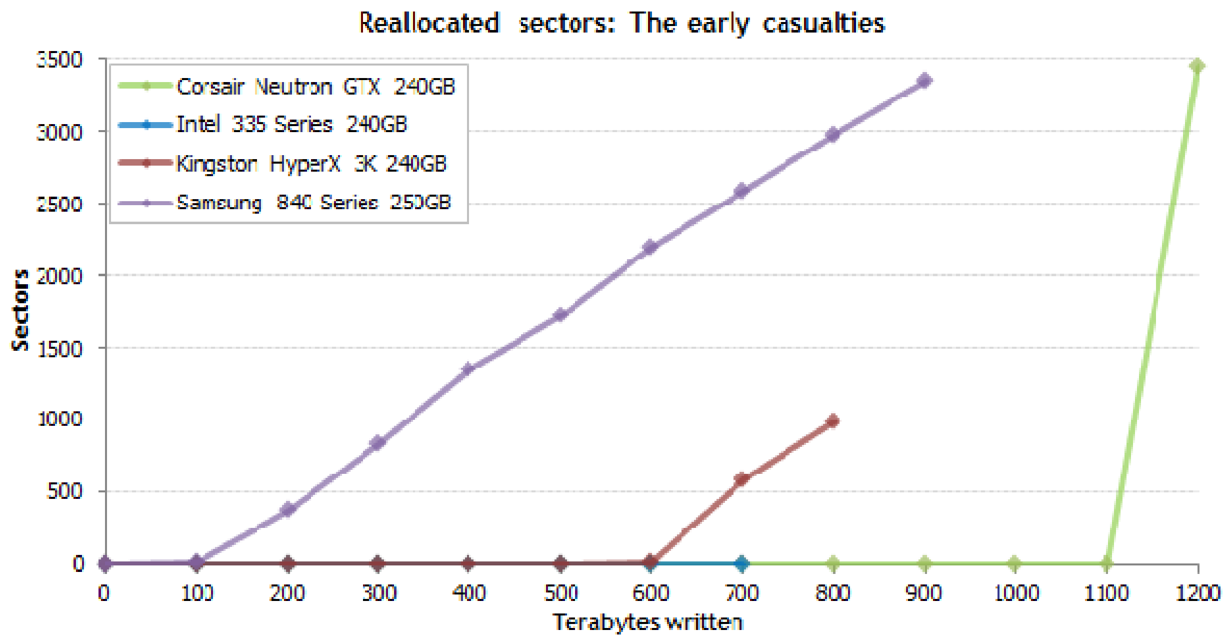


Figure 3 from https://www.extremetech.com/computing/201064-which-ssds-are-the-most-reliable-massive-study-sheds-some-light . Note, these are SSD RAMs, not 2.5 inch drives.

From: https://forums.servethehome.com/index.php?threads/win10-storage-spaces-is-it-that-bad.12227/

From 2016:

A lot of people just dislike Microsoft so no matter what they do, they get hated on. I played around with storage spaces and ended up returning to Stablebit Drivepool. With the exception of parity drives, it does similar things and I could just pull a drive out of machine "A" and it could be read on machine "B". Not so with storage spaces.

From 2013:

As long as you do mirroring, Storage Spaces is fine for all around utilization. I configure it in multiple ways depending on what I need, and I use powershell to do that.

Parity is slow on writes. so be careful there.

From 2016:

Hmm, the main reason I wanted to use this was for the benefit of protection against BitRot while still having one machine to do it all.
Storage Spaces has some great features, but it looks like performance is terrible and a lot of the features are still hidden away in Powershell or not even available on home OS.

I am ok with Powershell, I use it at work, but stripped features...

Now I am starting to look into DrivePool+Snap Raid+MD5Hash or perhaps UnRaid with some of its CRC plugins.

The real answer about unrecoverable error rate improvement (or not) with RAID systems:

> *Nobody reads the checksums on reads. There's no point. Reading a checksum means you <u>have to read the entire slice plus checksum, then compute the checksum to verify you have the correct data</u>. <u>Plus the orthoganal checksum if you are running RAID-6</u> or whatever. It is a total performance killer because it breaks the ability to randomly seek to totally different sectors on different disks at the same time. Similarly, almost nobody reads both sides of a mirror in RAID-1 because if you only read one side you can alternate which side of the mirror you read from so that you get faster throughput, and if you suddenly have a mismatch, which disk do you take as correct and which do you take as broken? All modern RAID systems depend on the on-disk controllers to signal the RAID controller that they are in distress (through SMART or the like), at which point that disk is almost always kicked out of the array. Checksums are used for rebuilding arrays, not for read-verification.*
>
> So the bottom line is that checksums could be computed to improve the error rate performance of RAID drives, but it is never done because of speed performance issues until the entire hard drive is declared bad.

At a minimum, you'd think you could at least have say 3 copies of everything and majority-vote the three to improve the error rate. This way, if the bit-error rate per disk was p, the probability of making the wrong decision on any given bit would be $3p^2$ would be astronomically small.

Maybe this is what I should do…use 3 SSD's with 3 identical copies, and if I ever encounter what appears to be an unrecoverable error, do my own voting of the 3 to get back to the best estimate for the true data values.