Technology as Magic, A Story of Errors

Paul Heath

[1]Seagate Technologies LLC

Abstract

Technological advances tend to appear as something magical when they are first introduced. An aspect of all technology that may not be considered in any depth, is how systems respond to error conditions. Often, the very essence of the technological solution is in how errors are handled. In this paper, several areas of technology are covered. Each one has different uses and applications. In each example, error handling is either a fundamental limit to the performance of the system, or it is a controlled parameter that is central to the behavior of the system. With the recent advent of artificial intelligence systems, we have to consider a few new aspects of how systems handle errors. AI systems have the ability to at least appear to have a point of view. They have been used to purposefully create products and responses that are objectively false or misleading. They also, due to the nature of a training/classification model, can be trained using data that is incorrect, or has a bias of some kind. We discuss the data security concepts of Confidentiality, Integrity, and Accessibility, and add a potential concept of Data Validity. This idea can lead to a discussion of how metrics might arise which enable the classification of data as being valid or not.

A History of Technologies

Digital Computing in 1975

The mid 1970s was early in the 'big iron' computer era. With IBM having moved from the 701 machine in the late 1950s to the 370 series in the mid 1970s. Mini-computing was in its early adoption. Digital Equipment was leading the way with the PDP series. ATT had a large operating system project called MULTICS underway. An alternative operating system was developed by Ken Thompson and Dennis Ritchie, this famously became Unix. Unix went on to become a dominant factor in both university and departmental computing. Unix was initially written in PDP-7 assembler language. It went on to be refined and rewritten in a new, portable language known as C. C remains an important language today in low level applications and embedded systems, including its use as the primary language used in the Linux operating system, a very popular Unix style system.

The author's first exposure to computing happened during the mid 1970s. The high school had a class in computer science. The system in use was a DEC PDP-11 running the RSTS/E operating system. Since the CPU was located at a small company, remote from the school, connections were made from a Teletype ASR-33 terminal, through a modem. The modem used a standard phone handset, and ran at 110 baud, or about 10 characters per second. The interaction with a remote computer looked like magic. A command went slowly into a teletype, and in a few seconds, a response came back. The limitations on communication speed, computing speed, and mechanical response of the teletype unit all combined to determine the response time. As we know today, all of those factors are now orders of magnitude faster. This is largely due to errors. Communication line speed is set by the rate at which data can flow with a low error rate. Computer clock rate is limited by silicon feature size, and heat transfer rates. These are also driven by errors. This is the first example of how errors shape our experience of technology.

The IBM Storage Subsystem

IBM pioneered the spinning disk storage device in 1956, with the 305 RAMAC system. This device used a large spindle stack of disks, and a single magnetic head / arm assembly to move from disk to disk within the collection of disks. The entire box stored 5 megabytes and weighed around 2000 pounds. The introduction of the disk storage concept began an industry that continues today, with disk storage being the most cost effective and widely deployed technology in use. By the early 1980s, IBMs disk subsystem had matured to what was called the 3380. This device was the first to offer more than 1 gigabyte of storage per spindle. A box, slightly larger than a refrigerator with a weigh of 1000 pounds, would store 2.52 gigabytes. In comparison, a modern high capacity disk drive can hold roughly 30 terabytes of data. Using the baseline of the 3380 device, if cars would have advanced at the same rate as disk drives, a single tank of gas would be able to take you the distance of 190 times the circumference of the Earth, or to the Moon and back 10 times.

How do errors shape the behavior of disk drives? The first job of a storage device is to save and recall data perfectly. The device presents a large expanse of 'logical blocks', each being 512 bytes, or in some cases 4096 bytes. Each block is numbered. The assumption is that when data is written to a given block, it can be read at some later time with exact recall. When a command to read a block comes down from the host computer, the drive first checks to see if it has that data in its cache memory. If so, it sets up a data transfer phase on the interface, and transmits the data to the host. If the data is not in the cache, the firmware of the drive tells the 'actuator' containing the magnetic heads, to move to the radial location containing the target block. While this is happening, the firmware sets up the read channel to trigger on the target block and transfer the data into cache. If that works, the data is transmitted to the host. If there is a read fault of some kind, the firmware goes through a complex sequence of retries, starting with simple re-reads and moving into channel equalization tweaks, off track read operations, and short write operations to reset magnetic domains in the head material. This process takes many dozens of disk revolutions to complete. A great deal of engineering effort has taken place in order to be able to deliver error-free data to the host device. In the past decade or so, a new use case has emerged. During the read of video frame data, a different perspective on errors is used. With video streaming, the requirement is that data be available at a rate of up to 25 frames per second at the video endpoint. If the disk system is spending dozens of revolutions trying to recover from a read error, it is unlikely to be able to keep up with the video frame rate. In this case, it is better to deliver data on time, with a few bytes incorrect than it is to deliver perfect data at too slow a rate. A single video frame with a few pixels of the wrong color will not be detectable by the viewer. The point here is that the definition of error can change with use case, and the means of dealing with errors has to also change accordingly.

Global Positioning

The Global Positioning System or GPS is a satellite based technology which can locate a receiver device anywhere on Earth to a very accurate degree. It works by having a receiver detecting at least 4 signals from a constellation of satellites in medium earth orbit. Each satellite transmits a signal containing the time that the signal was sent. Each satellite is moving at roughly 8700 miles per hour. The satellites have an extremely accurate clock on board. The clocks have relativistic correction factors for both gravitational and velocity effects. The position of each satellite is monitored by ground based radar stations to ensure that their positions are always well known. Initially, the system had a random error programmed into it. This feature was called 'selective availability', and was meant to be a deterrent to potential tracking threats. The result of selective availability was that the entire location grid was shifted by some random amount at a random time. the random dither function). The solution to the problem was to purchase a number of commercial GPS receivers for the troops, and remove the dither function from the system. The selective availability function was permanently removed in 2000. This is an example of managing errors both to reduce threats and to solve operational issues.

Offensive Cyber Security

Stuxnet was the most complex example of malware ever developed. It was originally discovered in June of 2010. Its purpose was to disrupt the enrichment of uranium at the Natanz facility in Iran. It is understood that the software was developed by a state sponsored group, though the actual authors are a matter of speculation. The software gained entry into a air-gapped network of high speed centrifuges controlled by a large array of Siemens S7 programmable logic controllers. The software was designed to change the speed of individual centrifuges in ways that would cause early failure. The effect of this was that scientists spent a great deal of time trying to debug and repair equipment rather than to spend time perfecting their enhrichment process. The aggregate enrichment rate fell to below the point of producing weapons grade uranium 235. One aspect of the operation of Stuxnet is that the presentation of system status data to the monitor system was prerecorded and looped to the monitor to cover up the fact that the system was manipulating the centrifuges behind the scenes. This is similar to the bank robbery scene the movie Ocean's 11. This is an example of misrepresenting system status to make it look like the system is running error free.

Modern Problems

Artificial Intelligence

In the world of data security, there is a concept known as the CIA triad. This acronym is formed from Confidentiality, Integrity, and Availability. Confidentiality is the notion that data should only be accessed by authorized users and protected from unauthorized access. Integrity is the idea that data has not been tampered with or altered by unauthorized sources. Availability is a guarantee that data can be accessed when needed by authorized users. With recent advances in AI technology, we have to start thinking about a few new things. The xAi agent known as Grok, recently (early July, 2025) began exhibiting very different behavior in its responses to prompts. It started praising Hitler. This brings up troubling questions about how AI agents are responding. AI agents typically use large language models which are trained with extreme amounts of mostly internet available data. We've seen several examples of how technologies use and react to error conditions. In the case of AI, there are two central issues.

- Does the training data contain errors, and what in fact, constitutes and error.
- Can the agent adopt a point of view.

In the case of general AIs, it is certain that some of the training data has inaccuracies. In the specific case of Grok, it appears that its behavior changed its point of view over a very short period of time. This suggests that the point of view of an AI agent can be quickly manipulated. The existing notion of the CIA triad might benefit from a new aspect, one of data validity. Though there are many complexities at hand, the idea of data validity in some sort of metric, would enable a degree of accountability for AI agents. Count we adopt a 'peer review' model ? Could we learn from how accounts of history are documented? Could there be an established corpus if data of known provenance? There are many examples where both media outlets and general internet information sources are distributing data that is objectively false. This comes in the form of data meant to promote certain political points of view, or economic results, or even hate driven false narratives. In any case, it is important that we try to address this issue. There will come a time when the reasoning capacity of AI systems can rival or exceed that of our own human capacity. Without a reasonable method to validate inputs to these systems, we can expect at best, flawed results, and at worst, highly manipulated outcomes. Neither case is good.