Constructing a resilient storage system poses a formidable challenge, even for seasoned professionals. A recent case in point is the incident at the Algolia data center, dubbed "When Solid State Drives Are Not That Solid." In this instance, Samsung SSDs were erroneously implicated in failures caused by a Linux kernel bug. As system complexity continues to escalate, the task of comprehending and diagnosing failures is poised to become even more arduous.

To gain a deeper understanding of real-world failures and improve the foundations of robust storage systems, we embarked on a comprehensive research journey. Initially, we conducted three empirical studies encompassing 207 bug reports sourced from Bugzilla, 59 bug patches for file system-aware applications, accompanied by reproducibility experiments, and a thorough analysis of 1553 bug patches related to persistent memory within the Linux kernel source tree.

Within these studies, we meticulously delineated issues across various dimensions such as resolution time, implicated kernel components, reproducibility, and root causes. This meticulous analysis provided us with a quantitative grasp of the challenges at hand. Furthermore, we delved deeply into a subset of these issues, and created a data set called "BugBench." This repository encompasses a wealth of information, including general bug patterns, consequences, fix strategies, triggering conditions, reproducibility assessments, and their implications for developing potent tools to tackle these intricate challenges.

Subsequently, we identified potential shortcomings in contemporary tools and, utilizing virtual machines and static analysis, embarked on an endeavor to enhance the efficacy of testing and debugging methodologies. Through rigorous experimentation, we successfully identified 29 potential bugs within persistent memory kernel drivers and uncovered several unexpected system behaviors. Our innovative correlation-based debugging technique proved to be instrumental in significantly curtailing the search space for root causes, whittling it down to a mere fraction (0.06% - 6.2%) of the original kernel function trees generated by FTrace.

Our enhancements transcend mere refinement; they open doors to heightened low-level observability, effectively extending their applicability to the latest state-of-the-art storage systems. We hold the aspiration that our diligent efforts and the open-source "BugBench" repository will serve as catalysts for further research, contributing to the measurement and enhancement of system reliability on a broader scale.