Selected Results of the Workshop on Data Storage Research 2025

GEORGE AMVROSIADIS, ALI R. BUTT, VASILY TARASOV, EREZ ZADOK, AND MING ZHAO

George Amvrosiadis is an Assistant Research Professor of Electrical and Computer Engineering at Carnegie Mellon University, and a member of the Parallel Data Lab. His current research focuses on distributed storage and data analytics, with an emphasis on high performance computing and machine learning. He co-teaches courses on cloud computing and storage systems. gamvrosi@cmu.edu

Ali R. Butt is a Professor and Associate Department Head for Faculty Development in the Department of Computer Science at Virginia Tech. He was also an organizer for the NAE US Frontiers of Engineering in 2010. Ali's research interests are in distributed computing systems, cloud/edge computing, file and storage systems, I/O systems, and system support for deep learning. At Virginia Tech he leads the Distributed Systems and Storage Laboratory (DSSL). butta@cs.vt.edu

Vasily Tarasov is a Researcher at IBM Research—Almaden. His most recent research focuses on new approaches for providing storage as a service in containerized environments. His broad interests include system and storage design, implementation, and performance analysis. vtarasov@us.ibm.com

With the emergence of new computing paradigms (e.g., cloud and edge computing, big data, Internet of Things, deep learning) and new storage hardware (e.g., non-volatile memory, shingled-magnetic recording disks) a number of open challenges and research issues need to be addressed to ensure sustained storage systems efficacy and performance. The wide variety of applications demands that the fundamental design of storage systems should be revisited to support application-specific semantics. Existing standards and abstractions need to be reevaluated; new sustainable data representations need to be designed to efficiently support emerging applications. To take advantage of hardware advancements, new storage software designs are also necessary to maximize overall system efficiency and performance.

Therefore, there is an urgent need for a consolidated effort to identify and establish a vision for storage systems research and comprehensive techniques that provide practical solutions to the storage issues facing the information technology community. In May 2018, a National Science Foundation (NSF) Workshop on Data Storage Research 2025 took place at the IBM Research—Almaden campus in San Jose, CA [1]. This two-day community-visioning workshop identified research challenges in designing novel and innovative systems to store, manage, retrieve, and efficiently utilize unprecedented volumes of data at increasingly faster speeds. Thirty-three researchers participated in the discussions. Participants came from academia, industry, and government to represent multiple storage, I/O, and distributed systems research communities.

In-depth discussions were carried out at the workshop along four major themes: (1) storage for cloud, edge, and IoT systems; (2) AI and storage; (3) rethinking fundamentals of storage systems design; and (4) evolution of storage systems with emerging hardware. The participants underscored the need for focused educational and training activities to instill storage system expertise and interest in the next generation of researchers and IT practitioners. Finally, the development of shared, scalable, and flexible community infrastructure to enable and sustain innovative storage research and verifiable evaluation was also discussed. This article summarizes the discussions on the interaction of cloud and AI with storage. For more details, see the full workshop report [10].

Storage for Cloud, Edge, and IoT Systems

The advent of cloud computing has transformed the basic substrate for systems building in the last decade, and the long-anticipated "Internet of Things" (IoT) has led to the emergence of edge computing that extends system boundaries pervasively. In such a dynamic context, the depth of the storage stack and the scope of storage systems are increasing rapidly. Storage systems will need to manage data collected, stored, transformed, and transferred from heterogeneous edge devices to back-end cloud services, which can involve more than 18 layers [8]. Moreover, there are potential gaps or miscommunications between layers and components, which increase the difficulty of providing end-to-end guarantees and achieving the ideal tradeoffs among performance, reliability, fairness, etc. To move data storage research forward for cloud, edge, and IoT systems, we summarize the research challenges and opportunities into nine key properties that are essential for future storage systems.

AND STORAGE

Erez Zadok is a Professor of Computer Science at Stony Brook University, where he directs the File Systems and Storage Lab (FSL). He received his PhD in computer science from Columbia University in 2001. His current research interests include file systems and storage, operating/distributed systems, energy efficiency, performance and benchmarking, big data, and applied ML/AI. Zadok has published over 100 refereed papers, which have been cited over 7,000 times. ezk@cs.stnybrook.edu

Ming Zhao is an Associate Professor of Computer Science at Arizona State University (ASU) , where he directs the Research Laboratory for Virtualized Infrastructures, Systems, and Applications (VISA, http://visa.lab.asu.edu). His research is in the areas of experimental computer systems, including cloud/edge, big data, and high-performance systems as well as operating systems and storage in general. He is also interested in the interdisciplinary studies that bridge computer systems research with other domains. mingzhao@asu.edu

1. Efficient systems. Similar to traditional systems, cloud-based systems also put great focus on efficiency. This is important for both users who pay for usage and for cloud service providers who need to maximize profit. However, compared to traditional systems, there are many more layers involved in cloud systems: different layers usually require different data formats and read/write strategies to achieve the best local efficiency; these may conflict with other layers. Moreover, the diverse hardware, dynamic workloads, multitude of customer-facing cloud services, and inherent multitenancy make achieving high efficiency even more difficult. Finally, more effort must be expended on capabilities that support local and end-to-end quality of service (QoS). We can no longer focus on a single layer or component. Instead, cross-layer and end-to-end solutions are needed for removing all excess resource allocations in different layers, saving various costs (e.g., CPUs, memory, energy) and achieving an overall high efficiency.

2. Unified systems. Modern systems are filled with diverse storage options (e.g., file systems, SQL databases, key-value and object stores). While each individual storage option usually provides some unique features, they often have similar functions or components (e.g., managing persistent data structures or data replication). This inherent overlap of functionality is one of the major sources of inefficiencies in today's systems. We should explore the possibility of extracting the unified core components as building blocks and providing generalized solutions for various higher-level services. Also, to make different services more unifiable, we should experiment with solutions that can automatically transform configurations based on the dynamic needs of workloads: addressing the underlying representation of data, the amount of resources allocated, and adapting configurations of durability and replication parameters.

3. Specified systems. Current approaches to system building are too prescriptive, rigid, and error prone. This has led to various problems, including downtime and data loss, reducing future storage systems' scalability. We envision that future systems and applications should be specified in terms of performance requirements, data persistence needs, etc. Correctness properties should be precisely specified throughout the systems, which could potentially lead to the holy grail of verified systems that never lose data. There are several open research questions: how to specify properties for the opaque cloud, how to identify the necessary properties and interfaces for each layer or component in the system, and how to specify the dynamic requirements of workloads.

4. Elastic systems. Unlike traditional storage clusters that are built on fixed hardware resources, cloud-based systems are naturally elastic. Such systems can be broken into constituent components that can be scaled up/down independently based on current workload demands. We envision that system elasticity can be utilized for handling storage infrastructure tasks in addition to the workloads, likely improving overall system utilization and efficiency. To utilize storage elasticity, more desegregated, composable software architectures are highly desirable. Instead of today's monolithic storage and file systems, we should experiment with different building blocks and microservices, which can be reused across domains and improve elasticity, long-term reusability, etc.

5. Explainable systems. Current cloud-based systems are opaque to users. Many services use relatively simple interfaces, which makes it difficult for users to reason about the provenance and layout of their data. Moreover, due to the complicated layering within the cloud, it is also difficult for system builders or administrators to explain abnormalities in system behaviors. We envision future systems as providing detailed provenance information at a configurable verbosity level regarding, for example, how a data object was created, the number of copies of it stored in the system, and who can access what and why. This will be helpful for improving security (e.g., how information is leaked), reliability (e.g., how data is corrupted), and performance (e.g., why this one run is slow).

6. Sharable systems. Unlike the first-generation cloud technologies that only run a single or a few applications for one entity, multitenancy is a new reality in modern cloud-based systems. We believe one fundamental requirement of multitenancy is effective sharing. However, achieving effective sharing is nontrivial as it involves many other systems aspects. For example, from efficiency's perspective, multitenancy may cause interference among different workloads at different system layers and thus violate QoS or service level objectives (SLOs). Similarly, security and privacy concerns need to be addressed in the multitenant environment to provide trustworthy sharing.

7. Application-driven systems. One major driving force of systems research is new application needs. There are many interesting new applications arising recently (e.g., augmented reality), which place new demands on storage systems (e.g., real-time processing). Given the diversity of applications, it is inefficient and impractical to build a highly specialized storage system for each application. Instead, we should explore the commonality among applications and automatically adapt storage systems for a range of applications. One unique challenge here is how to assemble a representative application suite and metrics for learning the common characteristics and demands.

8. Reliable systems. As the scale and complexity of systems keep increasing, failures become the norm rather than the exception. Therefore, we need to design systems to deliver high performance and other desired properties in the presence of failures. Future systems need to be formally specified, which could potentially lead to truly reliable storage that will never lose data. Existing efforts have shown that it is possible to formally specify and verify the crash consistency of one local file system built from scratch [6]. Nevertheless, it remains unclear how to scale formal methods to the vast majority of legacy software systems in cloud environments. More advanced mathematical methods and software engineering approaches are desirable.

9. Re-evaluable systems. A constant theme in storage research is the availability of suitable workloads. This is critical for fair comparisons between systems and for generating reproducible results. Unfortunately, compared to workloads for local storage systems (e.g., Filebench, SPEC-SFS), fewer cloud-based workloads are publicly available; a few such useful workload generators exist (e.g., YCSB [7], ATLAS [4]), but as systems keep evolving, more representative workloads are needed to advance research. Moreover, future storage systems should be built with easy evaluation in mind (e.g., exporting internal performance metrics) to facilitate the fair comparison of design tradeoffs under the same representative workloads.

Edge and Its Impact on Cloud

IoT is becoming a reality, causing an explosion of data collection, storage, and processing demands. The proliferation of IoT devices and the associated demands have led to the emergence of edge computing. Essentially, the edge model places a "mini datacenter" of compute and storage resources at the network edge, closer to end users. Compared with cloud computing, edge computing is less mature or standardized, and IoT devices can differ in capabilities, protocols, and data formats.

The service models for IoT applications are unclear. We envision that one possible direction is the serverless computing model, like AWS Lambda. However, additional research efforts are needed to integrate the spectrum of IoT devices into current models. Despite this heterogeneity, one common feature of all IoT devices is their limited hardware resources. To address this constraint, we should explore how to identify and discard unimportant data in a timely fashion—and how to balance among storage, preprocessing, and communication between IoT devices and clouds.

Cloud systems can be built for various workloads and adapt to demands on the fly. Conversely, edge computing has a large upfront cost to install edge nodes and a limited opportunity for ad hoc multiplexing at runtime, so we need to identify these workloads and match them to storage capabilities precisely.

One barrier to storage research in the era of cloud-edge computing is that no edge-to-cloud, holistic, persistent data storage capabilities exist today. Therefore, a realistic testbed involving both edge and cloud is highly desirable. Another barrier is the lack of agreed-upon workloads and traces for evaluation and comparison of new research designs. A realistic workload trace needs to track requests to read and write data across all devices, edge nodes, and cloud servers—including operations that transform or aggregate the data. Recent work on distributed system tracing [2] may provide the mechanism for collecting such traces; but the research community also needs to agree on a trace format, such as SNIA's DataSeries [5], and strategies for replaying such traces.

AI and Storage

Although AI and ML have existed as separate fields for decades, the last 5–10 years have witnessed an exponential growth in AI development and applications. Today, virtually all industries are either applying or planning to apply AI techniques. This shift is driven by three factors: data, compute, and algorithms. The confluence of these three factors has fueled AI's growth and, in turn, will drive the need for combined storage and AI research. Storage for AI focuses on how storage research that drives system designs can better serve AI workloads and data usage. Conversely, AI for Storage focuses on how storage systems can be improved via internal application of AI techniques.

Storage for AI

Storage technologies are likely to be more complex in the future to support growing needs of big data and AI workloads. This complexity will demand support for different APIs at different levels. We expect to continue to see healthy use of block-level, file-level (e.g., POSIX), object, and key-value stores—and likely combinations thereof. There is a need for high-level, easy-to-use APIs that hide much of the internal complexity from users and developers; conversely, there is also a need to allow advanced users to access lower-level APIs to enable more effective custom optimizations. The key to the design of future storage systems and their APIs would be that they must be easy to use and logical for AI application developers and *at the same time* provide optimal storage at the lower levels. Specifically, the emerging AI field presents five trends that intersect with storage, where targeted storage research can benefit AI uses and applications:

1. Massive data sets. AI workloads require the ingestion, pre-processing, and, ultimately, analysis of massive amounts of data. Multiple stages exist in typical AI pipelines, from data ingestion such as ETL (Extract, Transform, Load), to pre-processing (e.g., feature engineering), to the ultimate execution of an AI algorithm in its training or inference phases. All of these can benefit from storage optimizations for performance and data management.

2. Awareness of AI stages. Storage that is aware of the distinct stages or phases of AI processing can optimize AI pipelines via techniques such as caching of intermediate results, tracking of lineage, provenance, and checkpointing.

3. Compute architecture and data optimization. AI platforms typically follow distinct distributed computation architecture patterns (e.g., data and model parallel). Memory hierarchy and data layout design for such computation patterns should be a focus for future storage research. APIs that express the data access intent of an AI algorithm can be a powerful tool to integrate optimizations with AI computation.

4. Unique characteristics. AI algorithms have unique characteristics that can be exploited to create efficient storage designs. Example characteristics include tolerance of small amounts of data loss, highly structured access patterns, and the ability to use and extrapolate from lossy compression. Emerging access methods and characteristics associated with AI workloads, such as stream processing or edge storage, also create unique challenges.

5. Security, traceability, and compliance. The use of AI brings a new dimension to data security. As industries and users demand that decisions made by AI algorithms be reproducible, transparent, and explainable, pressure builds on enterprises to use data-management mechanisms to govern what data is collected and how it should be used to generate AI models and consequent insights.

AI for Storage

AI techniques should be researched to improve storage systems with respect to performance, reliability, availability, and QoS. This can be accomplished using the large and growing amount of available storage systems' historical access data. Insights can be gained from training and thus be used to help design or optimize storage systems in five ways:

1. Data placement optimizations. ML algorithms can be applied to predict popular data and application patterns, which help improve various storage techniques, including tiered caching, prefetching, and resource provisioning. Adapting caching policies through online learning can have significant benefits: using ML techniques to select between LRU and LFU replacement policies, for example, improved cache hit rates significantly under tighter memory constraints [9]. We believe that ML can be successfully applied for other performance optimizations.

2. Failure prediction. Failure or error patterns in large storage systems, such as disk failures and silent data corruptions, can be predicted using ML techniques and early detection; then, cautious measures can be taken to prevent errors from propagating. For example, proactively replacing disks that are predicted to fail soon can reduce the cost of data loss or rebuilding.

3. Storage tuning. Storage systems typically evolve to have a large number of tunable parameters. Parameters include hardware composition, I/O schedulers, tiering thresholds, cache sizes, etc. Using learning and other black-box optimization techniques can help administrators build and maintain storage systems under dynamic workloads, informing them on the optimal parameter values to improve system performance and lower cost for given workloads.

4. Change and anomaly detection. Part of tuning for workloads is understanding when they change phases. Anomaly detection has been an application area for ML techniques for over 20 years, and many techniques from these fields can translate easily to storage domains.

5. Intelligent storage devices. Storage devices capable of carrying out computation can help reduce maintenance overhead for the overall storage system, potentially improving performance. Such devices, however, require that we determine what level of intelligence is appropriate to offload to the device and propose storage techniques to achieve the best synergy.

The key challenge in using AI for Storage is that training data will often be limited before decisions have to be made. For instance, systems to store and quickly process data in self-driving cars must exist and run fast even *before* enough data can be collected for automated system design. Similarly, as storage needs shift over time in an organization, there may not be enough training data to predict how best to deal with changing priorities when reconfiguring system parameters, tiers, data placement, and layout. Storage tuning may also be improved by considering more complex cost models, not just traditional throughput and latency: dollar cost, complexity, and power consumption are useful reward functions in a multi-objective optimization scheme.

Benchmarks and Workloads

Since AI techniques are heavily data dependent, any strategy for driving AI and storage research needs to factor in the need for publicly accessible data sets and benchmarks. Public data sets exist in ML but are in many cases too small to extract meaningful storage access patterns. Next, we describe three challenges that have to be overcome to drive expansive research into the storage and AI opportunities presented above:

1. Data-set generation and collection. We need some systematic and sustainable schemes to generate and collect data sets, including synthetic data generation of ML workloads, data sets from simulations and prior research, and long-term data collection and dissemination via shared community infrastructure.

2. Characterizing workloads across layers. How to benchmark and characterize workloads from different layers, including application, middleware, system, and storage-device layers, is challenging and needs investigation.

3. Workload classification. Classifying workloads has been studied for a long time [3]. As new storage platforms and applications are developed, there is a need to understand, in a way that is precise and communicable across different industries, what modern storage workloads look like. We could use ML techniques to improve workload characterization in four areas: (1) quantifying similarity among workloads; (2) tracking changes in how a workload functions on a given architecture; (3) learning mixes of customer workloads on shared storage systems; and (4) detecting phases of complex long-running workloads.

Conclusion

The NSF Workshop on Data Storage Research 2025 has unquestionably identified that the ongoing evolution of computing use cases, hardware technologies, and resource consumption patterns creates a multitude of new and complex challenges in data storage and management. We hope that this summary article and its associated, full-length public report [10] will serve as a useful guidance for data storage researchers in the coming years.

References

[1] National Science Foundation Visioning Workshop on Data Storage Research 2025: https://sites.google.com/vt.edu/data-storage-research/.

[2] A. Agelastos, B. Allan, J. Brandt, P. Cassella, J. Enos, J. Fullop, A. Gentile, S. Monk, N. Naksinehaboon, J. Ogden, M. Rajan, M. Showerman, J. Stevenson, N. Taerat, and T. Tucker, "The Lightweight Distributed Metric Service: A Scalable Infrastructure for Continuous Monitoring of Large Scale Computing Systems and Applications," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (*SC '14*), November 2014, pp. 154–165.

[3] A. K. Agrawala, J. M. Mohr, and R. M. Bryant, "An Approach to the Workload Characterization Problem," *Computer*, vol. 9, no. 6 (1976), pp. 18–32.

[4] G. Amvrosiadis, J. W. Park, G. R. Ganger, G. A. Gibson, E. Baseman, and N. DeBardeleben, "On the Diversity of Cluster Workloads and Its Impact on Research Results," in *Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC '18)* USENIX Association, 2018, pp. 533–546: https://www.usenix.org/system/files/conference/atc18/atc18-amvrosiadis.pdf.

[5] E. Anderson, M. Arlitt, C. Morrey, and A. Veitch, "DataSeries: An Efficient, Flexible, Data Format for Structured Serial Data," *ACM SIGOPS Operating Systems Review*, vol. 43, no. 1 (January 2009).

[6] H. Chen, D. Ziegler, T. Chajed, A. Chlipala, M. F. Kaashoek, and N. Zeldovich, "Using Crash Hoare Logic for Certifying the FSCQ File System," in *Proceedings of the 25th Symposium on Operating Systems Principles (SOSP '15)*, ACM, 2015, pp. 18–37.

[7] B. F. Cooper, A. Silberstein, E. Tam, R. Rakrishnan, and R. Sears, "Benchmarking Cloud Serving Systems with YCSB," in *Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC '10)*, ACM, 2010, pp. 143–154.

[8] E. Thereska, H. Ballani, G. O'Shea, T. Karagiannis, A. Rowstron, T. Talpey, R. Black, and T. Zhu, "IOFlow: A Software-Defined Storage Architecture," in *Proceedings of the 24th ACM Symposium on Operating Systems Principles (SOSP '13)*, ACM, 2013, pp. 182–196.

[9] G. Vietri, L. V. Rodriguez, W. A. Martinez, S. Lyons, J. Liu, R. Rangaswami, M. Zhao, and G. Narasimhan, "Driving Cache Replacement with ML-Based LeCaR," 10th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage '18), USENIX Association, 2018.

[10] G. Amvrosiadis, A. R. Butt, V. Tarasov, E. Zadok, M. Zhao, "Data Storage Research Vision 2025: Report on NSF Visioning Workshop," 2018: https://dl.acm.org/citation.cfm?id=3316807.