Research Statement

Muthian Sivathanu

My research interests broadly span operating systems and distributed systems. Most of my recent work is centered on file and storage systems, and is focused on improving them along various dimensions such as availability, security, and performance. Efficient and robust persistent data storage is a fundamental prerequisite of existing computer systems, and is bound to become more important in future.

A key problem in storage systems today is that the range of functionality they can provide is fundamentally limited, despite the presence of sufficient processing power and significant low level information and control within them. One of the main reasons for this limitation is that storage systems lack higher level semantic understanding about the data. My research addresses this fundamental problem.

In this document, I first describe my thesis research, then discuss other research I have done on general distributed systems and operating systems, and finally present my future research agenda.

Dissertation Research

My dissertation work is comprised of two parts: the first part explores techniques to enable the placement of "smart" functionality and optimizations within the storage system. The second part involves utilizing those techniques to propose and implement new features for improving availability, security, and performance, that are impossible to do in existing storage systems.

Background: Storage systems have evolved significantly over the years. High-end storage arrays today have hundreds of processors and multiple gigabytes of memory in them, and utilize such power to implement a variety of optimizations. However, the interface they export to the outside world, *i.e.*, SCSI, has remained unchanged. Thus, higher layers of the system, such as the file system, are oblivious of complex functionality implemented within storage systems and still view the storage system as the simple disk that it was a few decades ago. As a result, file systems have very little information and control on the low-level details of data storage; for example, when running over an array volume, the file system has no control over which physical disk a block is laid out in, or the kind of redundancy employed for a particular block. Therefore, storage systems are becoming the inevitable locales to place functionality and optimizations that require such low level information.

Techniques: A key requirement for enabling complex functionality within storage systems is that the storage system needs to understand the higher level *semantics* of the data it stores. Current storage systems, by virtue of the simple block-based SCSI-like interface, only observe a raw stream of block reads and block writes. Higher level logical information such as the assignment of blocks to files, whether a block is a metadata or data block, whether a block is live, etc. are unavailable within the storage system. If the storage system had this information, it could combine this higher-level information with the low-level information and control that it already has, thus enabling optimizations that require both these classes of information. The first part of my thesis explores techniques to convey this higher-level semantic information to the storage system. While one obvious approach to convey semantic information would be to change the SCSI interface to a richer interface, such an approach is not pragmatic; modifying an interface as basic as SCSI requires broad industry consensus and raises legacy issues. In contrast, our approach is to acquire semantic information without changing the existing block-based SCSI interface. My dissertation proposes a new class of storage systems called "semantically-smart disk systems", that are capable of implicitly tracking semantic information about the file system, underneath an unmodified SCSI interface [FAST03]. A semantic disk infers higher level information by careful observation of block level read and write traffic, coupled with some static knowledge of the key on-disk structures of the file system.

Automatically tracking file system information underneath unmodified SCSI is challenging, because of the asynchrony exhibited by modern file systems. Buffering of metadata writes in the file system's buffer cache obscures the information available to the semantic disk and imposes fundamental limits on the degree of accuracy with which the information can be tracked. To explore the generality of the semantic inference techniques, I experimented with a variety of file systems: Linux ext2, ext3, and VFAT, besides some preliminary exploration of Windows NTFS. Since we observed that reasoning about what information can and cannot be tracked accurately is quite complex due to various file system behaviors, we wanted to formalize this. To this end, I identified a set of key dynamic file system properties that influence the accuracy of tracking information within a semantic disk. I also formulated a logical framework for

representing and reasoning about the update behavior of a file system. This logic was then used to reason about the extent of accuracy with which various pieces of semantic information can be tracked [LOGIC04].

Although the initial work on semantic disks was focused on file systems, I also subsequently explored the problem of extracting semantic information underneath a DBMS. With a prototype semantic disk running underneath a Predator/Shore DBMS, I demonstrated that such extraction was indeed feasible to a high level of accuracy [DB04].

Case Studies: The second part of my dissertation explores how to exploit semantic information within a storage system. This part proposes new optimizations that are impossible to implement in traditional storage systems, and demonstrates that they can be readily implemented within a semantic disk. One example is D-GRAID, a storage layout technique that greatly improves the availability of the storage system under multiple disk failures [FAST04]. Existing RAID schemes result in complete unavailability of data once the number of disk failures exceed the tolerance threshold of the array (typically 1). In contrast, D-GRAID ensures *graceful degradation* of availability. D-GRAID continues operation, providing access to a large fraction of semantically meaningful data, thus enabling most processes to complete oblivious of the failure. By making layout semantically-aware, D-GRAID prevents the availability cliff behavior seen in traditional RAID.

Another example of functionality that is enabled by a semantic disk is *secure deletion*, the ability to make data irrecoverable on deletes by repeated overwrites of the block with specific patterns [OSDI04]. Existing tools for this purpose that operate at the file system level are fundamentally incorrect when applied on modern storage systems that perform NVRAM buffering and block migration. The storage system is the only locale where secure deletion can be implemented reliably, but to implement this, the storage system also needs information on when deletes occur. I designed and implemented a prototype semantic disk that infers deletes and performs secure deletion. In contrast to the D-GRAID example, secure deletion was an extreme application for a semantic disk because it relied on the inferred information for correctness; inaccuracy could potentially result in trashing valid data. This example provided insights into the limits of semantic information that can be reliably tracked, and demonstrated that storage level functionality can even depend on inferred semantic information for correctness. Other examples of semantically-smart disk functionality include smart NVRAM buffering and exclusive caching within the storage system [ISCA04].

In each of these case studies, the complex behavior of modern file systems posed fundamental limits to the accuracy of semantic information that can be tracked, and required novel techniques to circumvent such uncertainty. Since I used prototype implementations to evaluate most of the case studies such as D-GRAID and secure delete, it provided a clear insight into various practical implementation issues that arise underneath modern file systems.

In addition, I also implemented the D-GRAID and secure delete case studies underneath a DBMS, and identified some minor modifications to database systems that would facilitate such semantically-smart functionality underneath them. This work is under submission [DB04].

Distributed Systems

I have also been greatly interested in distributed systems. For my masters project at UW, I designed and implemented WFS, a distributed file system targeted at network attached disks [MS01]. This file system was then used as the platform to explore and evaluate a new communication paradigm for distributed systems, called Scriptable RPC (SRPC) [ASPLOS02]. SRPC added a scripting capability to traditional RPC. By not requiring any additional effort by the developers, SRPC provided an easy migration path for existing distributed systems. I demonstrated the benefits of SRPC with a case study of an active storage system, showing that SRPC improves the performance, functionality, and design simplicity of distributed file systems.

In the context of WFS, I also explored distributed file system security. I proposed a mechanism for access anonymity in a distributed file system, that automatically adapts its message traffic in order to reduce network traffic overhead, while at the same time preserving complete anonymity [ANON02].

My summer internships have also given me plenty of opportunity to experiment with distributed systems, in particular distributed storage. In my internship with HP Labs in summer 2001, I proposed a new mechanism for scalable, fault-tolerant wide area replication that permits concurrent conflicting updates from all replicas in a geographically distributed storage system. The protocol involved constructing a logical chain topology out of the replicas and propagating updates along the chain. I showed that the technique had very good consistency and fault tolerance properties. In summer 2002, I interned with IBM Research Almaden, where I explored the problem of resource management in a wide-area storage system. To this end, I designed a scalable, decentralized architecture based on an economic

paradigm. The architecture, targeted at wide-area storage service providers, facilitated efficient replication of competing client "virtual disks", in terms of the number of replicas and their placement across the globe.

Operating Systems

General operating systems research is another area I am passionate about. In my summer internship at Google in 2003, I designed and implemented various techniques for performance isolation on Google's production servers running Linux. The techniques extended across all major resources such as CPU, memory, disk I/O, network, and the OS file cache. This project provided me a good learning experience in terms of real world constraints on operating systems innovation; for instance, their disinclination to modify the kernel was interesting. At Microsoft in summer 2004, I was part of a small core team that is working on a new operating system kernel from scratch. I designed and prototyped a key component of its IPC subsystem. This experience was extremely valuable in that it also involved interaction with various groups within the Windows division to learn about their experiences and common problems with existing operating systems.

Future Research Agenda

In the short term, I would like to extend my thesis work on smart storage systems. Specifically, one area I would like to investigate is the application of the semantic inference approach in the context of virtual machines. Since virtual machine monitors (VMMs) observe the same block-level traffic underneath guest operating systems, there could be missed opportunities in global resource management across multiple guest OSes. Inference of logical information about the file system can help, for example, in compacting unused free space due to deleted files in the guest system, or performing better disk scheduling across guest OSes by differentiating between "synchronous" writes and background writes. The logic framework that I developed for reasoning about file systems presents another avenue of future research. It would be interesting to explore how such a framework can augment existing techniques for verifying file system implementations. A more ambitious goal would be to explore if one can start with a logic specification of a file system's interaction with the disk, and automatically arrive at an implementation that preserves the ordering guarantees mentioned in the specification. Recent research has demonstrated serious correctness bugs in widely used file systems; automating the implementation of file system consistency management can help alleviate this problem.

In the longer term, I hope to work on areas that leverage my past experience, to build more robust and functional systems. One area I am interested in is distributed file and storage systems for emerging environments such as pervasive computing and sensor networks. Given that systems research is strongly influenced by technology trends, I am also interested in revisiting file system and general operating system design, as appropriate, in the face of technology changes such as ever increasing local and wide area network bandwidth, new storage media with varied characteristics, new interfaces to storage, and emerging classes of applications. I think an end-to-end vertical exploration, from the application layer all the way down to hardware, is crucial for effective research along these lines. My exploration of semantic storage in the context of DBMS systems is a step in this direction, where the storage hardware considers application-specific information to implement optimizations [DB04].

In summary, although my past research has mainly been in file and storage systems, I also have a deep interest in general operating systems and distributed systems. I like *building* real systems; most of the work I have done in the past have involved prototype implementations. I look forward to collaborating with others in systems and related areas such as networking, architecture, and databases, and hope to find a stimulating environment that facilitates such collaboration. I would also very much like to collaborate with industry in order to easily validate the timeliness and relevance of my future research; my various summer internships have given me good industry exposure that I hope will assist me in this direction.