

Research statement

Craig Soules

soules@cmu.edu

<http://cv.craigsoules.com/>

In a perfect world, file systems would be omniscient, helping users organize, locate, and access their data, acting almost like a personal assistant. Unfortunately, we are far from this ideal today. In fact, information management is one of the largest problems in computing today, as can be seen in the amount of time, money, and effort spent by both companies and individuals in addressing this problem.

The problem is that today's file systems are dumb. They provide only a mapping between a name and a piece of associated data (i.e., a file), leaving users unassisted with the task of organizing their data in such a way that they can find it again when it is needed. While this may have been sufficient in the past, recent increases in both the power and capacity of personal computers have drastically increased the amount of data generated by the average user, while simultaneously removing the need to delete old data. As our computers move from being bookshelves to being libraries, they must also move from being pieces of furniture to being librarians.

Before a file system can begin to assist the user in organizing or finding their data, it must first understand the data that it stores. What do a file's contents represent? How will the file be used? How and when will a user access the file? What is the file's relationship to both other files and user tasks? Without being able to answer these questions, the system will never be able to accurately determine or satisfy a user's information need. Most systems today use content analysis to try and answer these questions; however, a file's contents can only answer the first of these questions, and then only for the files containing structured, indexable content (e.g., text).

My thesis work proposes using *context* information to answer these remaining questions. The Merriam-Webster dictionary defines context as "the parts of a discourse that surround a word or passage and can throw light on its meaning." As such, context provides the system with knowledge about a file that is particular to both the current user and the current task. The problem is finding ways to automatically identify a file's context, and once identified, to store and use this context to improve file organization and search.

In my work, I developed *Connections*, a context-enhanced search tool that identifies a file's context using the temporal locality of file accesses to identify the contextual relationships between files. Once *Connections* captures these inter-file relationships, it uses them to extend and re-rank the results of traditional content-only search. By extending the results with contextually related results, *Connections* can find files that users believe are related to the given keyword, but do not contain that keyword in the file contents. Furthermore, by re-ranking the combined results using context, *Connections* is able to more accurately identify the files that the user is searching for. Through user studies, I was able to show that *Connections* reduces both false-negatives and false-positives when compared to traditional content-only search tools.

The results of my work provide two important lessons. First, context has a large role to play as we move toward the ideal of omniscient data storage. By giving the system a better understanding of how a user thinks about their data, its ability to accurately retrieve data can be significantly improved. Second, even with the advantages provided by context-enhanced search, there is still a large gap between our current retrieval accuracy and perfect accuracy. Finding ways to close this gap is a major goal of my future work.

Research Agenda

Looking forward, my research will continue to push toward the ideal of omniscient storage systems. Specifically, I plan to continue pursuing projects in automating file search and organization using context information.

Connections focuses on using context in the domain of file search, but I believe that context information can also be effectively applied to the domain of automated file organization. As a user works, identified inter-file relationships form an inherent collection, which the system can then use to organize all of the files associated with a particular user task. When a user continues that task, the set of files they require will then be quickly available.

Connections also focuses on only a single source of context, temporal locality; however, achieving truly omniscient storage will require additional sources of context, within both the system and user applications. In my future research, I plan to identify new forms of context and quantify their overheads and effectiveness. This information will help determine how context information should be gathered in practice and what context information is most valuable. Furthermore, methods of combining sources of context and, over time, having the system learn which sources are most successful for a given user are both open questions.

Connections gathers context information from system traces and stores it externally to the system. This separation prevents the use of context by other applications or even the system itself. I think such context should be tracked and exposed as a general system service. Research will be needed to understand how context information can be best integrated into the file system and used by other parts of the system. In addition to its utility in file search and organization, this integration could also inform performance-enhancing mechanisms such as file prefetching and mobile cache hoarding.

As file systems begin to change from simple mapping schemes to complex collections with detailed inter-relationships, it will be important to re-evaluate the interface by which users interact with the system. While other components of an operating system, such as scheduling, memory management, and networking, are all managed transparently to the user, the file system is the one aspect that the user interacts with directly. As such, automating file organization may not only result in changes to the file system's data allocation and management policies, but provides an opportunity to improve the higher-level naming schemes and potentially ease the way that users interact with their data.

Another interesting challenge facing file systems is the large number of devices and storage systems that the average user maintains. With the advent of mobile computing, and the constant increase in the amount of available storage on such devices, users are now faced with the problem of not only organizing large amounts of data, but also managing this data across devices. This leads to many interesting questions of how the work discussed in the previous three paragraphs applies to distributed devices. Collating context information across devices, managing naming and organization transparently, and distributing file system search are all areas that I plan to explore.

From another perspective, distributed file systems also have the potential to increase the amount of data shared among users. Shared file servers, peer-to-peer file sharing, and the internet are all examples in which users share information. While context is useful to an individual user, it may prove even more powerful when shared among users. After all, this is what makes web search (e.g., Google) work so well. Finding the right techniques to enable context-sharing transparently within distributed file systems, while maintaining privacy, will be an important area of future research.

Research Philosophy

Although I have done work in other areas, file and storage systems have been an emphasis throughout my research career. Generally speaking, file systems include some of the biggest usability, performance, and robustness challenges in modern computing. In fact, I believe that the data storage, organization, and retrieval provided by file systems is the most important aspect of computer systems from a user's perspective. To most users, the data stored on any given system is more valuable than the physical components that comprise the system. As a result, developing technologies that not only make data access efficient, but also make the data reliable, secure, easy to locate, etc., have been, and will continue to be, important goals of my research. For example, one of the motivations for my work on comprehensive versioning (CVFS) was the need for users to be able to identify and recover from malicious changes to their data. As discussed above, my thesis work begins to tackle the problem of helping users locate their data.

Furthermore, file systems generally represent the largest performance bottleneck for most workloads, caused by the growing discrepancy between memory and hard disk performance. As a result, the most profound effects on performance can be found at this layer. For example, in my work on comparing metadata journaling and soft-updates, I found that these techniques could provide performance benefits ranging up to 10x for various workloads when compared to a naive consistency technique.

My approach to research is the development and evaluation of full, working implementations. I believe that this is a requirement for fully understanding computer systems. Without usable implementations, researchers cannot see how proposed enhancements or changes interact with the rest of the components in the system, making thorough evaluation impossible. Working implementations can also provide useful tools and starting points for both developers and future research. I have found this to be true in all of my research endeavors. My work on soft-updates exposed several system-wide performance bottlenecks, which eventually led me to make changes in the virtual memory system of FreeBSD that improved system-wide performance. By developing a fully functional versioning file system, CVFS, not only was I able to evaluate my system using real benchmarks and traces, but other researchers were able to leverage my work to jump-start projects such as PASIS, Ursa Minor, and self-securing devices. My work on system reconfiguration resulted in tracing and timing tools that developers could utilize and laid the groundwork for additional work now being conducted by other researchers. Without a working implementation of context-enhanced search, there would have been no way to determine the effectiveness of the techniques, since the traces and user queries used to evaluate the system had to come from live systems. Furthermore, with this implementation now in place, I can use this tool as a foundation for my future research.

As a systems researcher, I am a strong believer in looking outside the realm of pure systems for both problems and solutions. Because systems manage all resources, and are transparent to users, they are often well situated to solve a wide variety of problems, and the solutions applied to them become available more broadly. As an example, my thesis work approaches a

problem of information retrieval, applies techniques from systems to provide a unified, context-enhanced search system, and evaluates the finished product using user study methods from human-computer interaction. Without drawing on each of these areas, proper treatment of this problem would have been impossible.

Because complex problems often require solutions from several areas of computing, collaboration is often a necessity, gathering the strengths of various individuals to contribute to the whole. Furthermore, when building complex systems, collaborated efforts can often accomplish far more, far more quickly, than working alone. As such, most of my past research has involved strong collaborations, and I plan for my future research to continue in this vein. Looking forward, I hope to find fellow researchers, both within systems and without, with interest in helping me bring storage one step closer to omniscience.