Craig A. N. Soules

2135 Wightman St., Apt. 6
Pittsburgh PA 15217
(412) 726-6096
soules@cmu.edu
http://cv.craigsoules.com/

Research interests

File, storage, and operating systems.

Education

Carnegie Mellon University; Pittsburgh, PA
Ph.D. in Computer Science; expected graduation Summer 2006.
Dissertation title: Using context to assist in personal file retrieval.
B.S. in Computer Science with a Minor in Mathematics; graduated December 1999.

Professional experience

2000 - present **Research Assistant**, *Carnegie Mellon University*; Pittsburgh, PA
· Conducted research on various aspects of file and storage systems, focusing on problems related to file search and maximizing file system robustness to failures and attacks.
· Designed and implemented Connections, a context-enhanced desktop search tool for Linux and Windows used to evaluate my thesis work
· Designed and implemented CVFS, a comprehensive versioning file system used in Self-Securing Storage, PASIS, and Self-* Storage
· Assisted in both the Continuous Reorganization and Self-Securing Devices projects.

Summer 2003 **Summer Intern**, *HP Labs, Internet Systems and Storage Laboratory*; Palo Alto, CA
· Worked on FAB, a brick-based decentralized storage system
· Designed, implemented, and evaluated cache coordination and eviction algorithms for FAB

Summer 2001 **Summer Intern**, *IBM, K42 Group*; Yorktown, NY
· Worked on K42, a micro-kernel designed for multi-processor scalability
· Implemented a scalable, modular file system for K42
· Designed a general mechanism for interposing kernel components

1999 - 2000 **Research Assistant**, *Carnegie Mellon University*; Pittsburgh, PA
· Explored relative merits of journaling and soft-updates for protecting metadata integrity
· Designed and implemented a journaling file system based on FreeBSD FFS
· Added several performance optimizations to soft-updates under FreeBSD

Summer 1998 **Intern**, *Lycos, Search Group*; Pittsburgh, PA
· Tuned their spiders for increased performance and wrote new spidering tools
· Re-wrote client-side DNS code from Bind to be thread-safe

Teaching experience

Fall 2004 **Teaching Assistant**; 15-213, Intro to Systems
· Held weekly recitations and office hours
· Developed new homeworks and exams questions

Spring 2001 **Teaching Assistant**; 15-412, Operating Systems
· Developed new homeworks and exam questions
· Assisted in grading of labs, homeworks, and exams

Refereed publications

Craig A. N. Soules, Gregory R. Ganger. *Connections: using context to enhance file search.* Symposium on Operating System Principles, October 2005.

Adam G. Pennington, John D. Strunk, John Linwood Griffin, Craig A. N. Soules, Garth R. Goodson, Gregory R. Ganger. *Storage-based Intrusion Detection: Watching Storage Activity For Suspicious Behavior.* USENIX Security Symposium, August 2003.

Craig A. N. Soules, Jonathan Appavoo, Kevin Hui, Dilma Da Silva, Gregory R. Ganger, Orran Krieger, Michael Stumm, Robert W. Wisniewski, Marc Auslander, Michal Ostrowski, Bryan Rosenburg, Jimi Xenidis. *System support for online reconfiguration.* USENIX Annual Technical Conference, June 2003.

Brandon Salmon, Eno Thereska, Craig A. N. Soules, Gregory R. Ganger. *A Two-Tiered Software Architecture for Automated Tuning of Disk Layouts.* Workshop on Algorithms and Architectures for Self-Managing Systems, June 2003.

Craig A. N. Soules, Gregory R. Ganger. *Why Can't I Find My Files? New Methods for Automating Attribute Assignment.* Workshop on Hot Topics in Operating Systems, May 2003.

Craig A. N. Soules, Garth R. Goodson, John D. Strunk, Gregory R. Ganger. *Metadata efficiency in versioning file systems.* 2nd USENIX Conference on File Systems and Storage Technologies, March 2003.

Jonathan Appavoo, Kevin Hui, Craig A. N. Soules, Robert W. Wisniewski, Dilma M. Da Silva, Orran Krieger, Marc A. Auslander, David J. Edelsohn, Ben Gamsa, Gregory R. Ganger, Paul McKenney, Michal Ostrowski, Bryan Rosenburg, Michael Stumm, Jimi Xenidis. *Enabling autonomic behavior in systems software with hot-swapping*. IBM Systems Journal, Volume 42, No. 1, 2003.

Jonathan Appavoo, Kevin Hui, Michael Stumm, Robert W. Wisniewski, Dilma Da Silva, Orran Krieger, Craig A. N. Soules. *An infrastructure for multiprocessor run-time adaptation.* Workshop on Self-Healing Systems, November 2002.

Gregory R. Ganger, Predeep K. Khosla, Mehmet Bakkaloglu, Michael W. Bigrigg, Garth R. Goodson, Semih Oguz, Vijay Pandurangan, Craig A. N. Soules, John D. Strunk, Jay J. Wylie. *Survivable Storage Systems.* DARPA Information Survivability Conference and Exposition, June 2001.

John D. Strunk, Garth R. Goodson, Michael L. Scheinholtz, Craig A. N. Soules, and Gregory R. Ganger. *Self-securing storage: Protecting data in compromised systems.* Operating Systems Design and Implementation, October 2000.

Margo I. Seltzer, Gregory R. Ganger, M. Kirk McKusick, Keith A. Smith, Craig A. N. Soules, and Christopher A. Stein. *Journaling versus Soft Updates: Asynchronous Meta-data Protection in File Systems.* USENIX Annual Technical Conference. June 2000.

Gregory R. Ganger, Marshall Kirk McKusick, Craig A. N. Soules, and Yale N. Patt. *Soft Updates: A solution to the metadata update problem in file systems.* ACM Transactions on Computer Systems, Volume 18 , No. 2, May 2000.

Recent talks

October 2005	*Connections: using context to enhance file search* 20th Symposium on Operating System Principles
October 2005	*Context-enhanced file search* 13th Annual Parallel Data Lab Workshop and Retreat
September 2004	*Improving File Search Using Context Information* 12th Annual Parallel Data Lab Workshop and Retreat
September 2003	*Distribution selection schemes in FAB* HP Labs
July 2003	*Why Can't I Find My Files?* Veritas
June 2003	*System support for online reconfiguration* USENIX Annual Technical Conference
May 2003	*Why Can't I Find My Files? New Methods for Automating Attribute Assignment* 9th Workshop on Hot Topics in Operating Systems
March 2003	*Metadata efficiency in versioning file systems* 2nd Conference on File and Storage Technologies

Patents

Method of Cooperative Caching for Distributed Storage System. Craig Soules, Arif Merchant, Alistair C. Veitch, Yasushi Saito, John Wilkes. Pending, filed March 2005.

Method of Hashing Address Space to Storage Servers. Craig Soules, Arif Merchant, Alistair C. Veitch, Yasushi Saito. Pending, filed March 2005.

Method of Caching Data. Craig Soules, Arif Merchant. Pending, filed March 2005.

Honors

Usenix Scholar 2000 & 2001

Research summary

2003 - present **Context-enhanced search**

My dissertation work seeks to improve on existing file search tools by utilizing contextual file inter-relationships identified from user access patterns. Traditional search tools are limited by two key factors. First, they are unable to effectively index any file that contains non-text or proprietary formats. Second, their literal interpretation of data makes it difficult to locate data when the search terms do not directly match the contents of the desired file or files. Context-enhanced search addresses these problems by identifying the connections between files that make up user tasks. Combining this information with traditional content analysis helps a search tool follow inter-file relationships to find desired files. Experiments with an implementation of context-enhanced search found significant reductions in both false-positives and false-negatives when compared to traditional content-only search.

In addition to introducing the concept of inter-file relationships for context-enhanced search, I have examined and evaluated a large variety of techniques for both identifying relationships and using the relationships to extend and rank results. Additionally, I have developed working implementations under both Linux and Windows for use in user studies aimed at validating the utility of context information.

2003 **Caching in FAB**

The Federated Array of Bricks project at HP Labs explores cluster-based storage with the aim of combining small, low-cost "brick-based" storage nodes into a single, scalable storage system. In such a system, software must provide the required scalability, load balancing, and reliability that is normally provided by hardware solutions in monolithic storage systems. My contribution to FAB was an evaluation of several distributed caching schemes. Because of the requisite replication to provide reliability in the face of brick failures, it is often the case that several bricks could cache a given block of data. Some goals of distributed caching schemes are to maximize the amount of available cache in the system by reducing double-caching, to route requests to the brick with a cached copy when possible, and to spread requests evenly across the system according to brick capabilities. I implemented four schemes, both in simulation and in the FAB prototype, and evaluated their respective merits.

2001 - 2003 **Online system reconfiguration**

The K42 project at IBM is an object-oriented operating system that incorporates aspects of microkernels and exokernels to provide high performance and scalability on large multiprocessor machines. My contribution to K42 was the design and implementation of an online interposition and hot-swapping service. Interposition places a wrapper object around a given component that can track and/or modify calls into and returns from a component. Hot-swapping allows a component to be replaced by a second, interface-compatible component. By using interposition and hot-swapping in a running system, we had a general mechanism for dynamic tracing and timing, dynamic fault injection, dynamic performance optimization, and several other tools useful to both developers and researchers.

2000 - 2003 **Self-securing storage and comprehensive versioning**

Self-securing storage leverages the separation of control between the operating system and an underlying storage device to provide an additional layer of security. By having the device version all data changes and audit all requests, a system administrator gains a wealth of previously unavailable information about intruder actions. Even if the intruder gains full control of a machine's host operating system, they would have to further compromise the storage device to change or remove version and audit logs, a task made far more difficult by the narrow interface and specific functionality provided by such devices.

I was a founding member of the project. As it went on, my focus became the design and implemention of the Comprehensive Versioning File System (CVFS). CVFS combines versioned b-trees, metadata journaling, and a log-structured layout to provide efficient, fine-grained versioning, while maintaining performance that is on par with non-versioning file systems. Through trace analysis, I found that CVFS reduced the space overhead of versioning by up to 54% over a naive versioning approach.

1999 - 2000 **A comparison of journaling and soft-updates**

One of the inherent tensions in file write caching is that of performance versus integrity. By caching dirty metadata, the system can hide hard disk latencies, but without controlling the eviction order, the state of file system structures on the disk can become inconsistent. For example, the system might write a pointer to a block before the block itself. If the system crashes or loses power while the disk structures are in this state, the result can sometimes be catastrophic. The simplest solution to this problem is to synchronously write all metadata changes to disk as they occur, however two alternate solutions have been proposed: journaling and soft-updates. Journaling keeps a separate log of metadata changes that is written ahead of the change itself. This ensures that any changes that should occur before that metadata is written are synced into the log and can be replayed in the face of failure. Soft-updates maintains in-memory ordering structures that enforce a safe write order from the cache, ensuring that the on-disk structures never become inconsistent.

I worked with others to quantify the relative merits consistency schemes. To complete the evaluation, I designed and implemented a journaling file system based on the FreeBSD fast file system (FFS). My system, a second journaling file system developed at Harvard, soft-updates, and two implementations of standard FFS (asynchronous and synchronous) were evaluated using a series of micro- and macro-benchmarks. This evaluation also exposed a number of deficiencies that allowed me to enhance both the FreeBSD buffer cache and the FreeBSD soft-updates implementation.

References

Prof. Greg Ganger
Dept. of ECE
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
+1-412-268-1297
ganger@ece.cmu.edu

Dr. Alistair Veitch
Hewlett-Packard Laboratories
1501 Page Mill Road, MS 1134
Palo Alto, CA 94304
+1-650-236-5561
alistair.veitch@hp.com

Prof. M. Satyanarayanan
Computer Science Dept.
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh PA 15213
+1-412-268-3743
satya@cs.cmu.edu

Prof. Garth Gibson
Computer Science Dept.
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh PA 15213
+1-412-268-5890
garth.gibson@cs.cmu.edu