BRANDON SALMON, STEVEN W. SCHLOSSER, LORRIE FAITH CRANOR, AND GREGORY R. GANGER

perspective: semantic data management for the home

Brandon Salmon is finishing his PhD at Carnegie Mellon and will be joining Tintri in the fall. He is interested in bringing user-centered design techniques, which have been so effective at improving the usability of user interfaces, to bear on system architectures, making them easier to understand and more accommodating of users' social systems.

bsalmon@cs.stanford.edu

Steve Schlosser received his PhD from Carnegie Mellon University in 2004, studying the use of alternative storage technologies in computer systems. He was a senior researcher at Intel Research Pittsburgh from 2004 to 2009 and is now a member of the technical staff at Avere Systems.

schlos@averesys.com

Lorrie Faith Cranor is an associate professor of computer science and of engineering and public policy at Carnegie Mellon University, where she is director of the CyLab Usable Privacy and Security Laboratory (CUPS). See http://lorrie.cranor.org/.

lorrie@cmu.edu

Greg Ganger is a professor of electrical and computer engineering at Carnegie Mellon University and the director of the Parallel Data Lab (PDL). His broad research interests in computer systems includes storage, OS, security, and distributed systems.

ganger@ece.cmu.edu

DISTRIBUTED STORAGE IS COMING home. An increasing number of home and personal electronic devices create, use, and display digitized forms of music, images, and videos, as well as more conventional files (e.g., financial records and contact lists). In-home networks enable these devices to communicate, and a variety of device-specific and datatype-specific tools are emerging. The transition to digital homes gives exciting new capabilities to users, but it also makes them responsible for administration tasks which in other settings are usually handled by dedicated professionals.

It is unclear that traditional data management practices will work for "normal people" reluctant to put time into administration. For example, most home users are accustomed to semantic organization (via applications such as iTunes) when accessing their data for daily use, but are forced by filesystem design to use a hierarchy when managing this same data.

We present the Perspective distributed file system, part of an expedition into this new domain for distributed storage. You can think of Perspective as in "Seeing many views, one gains Perspective." One focus of Perspective is simplifying data management tasks for home users. For example, Perspective's design allows home users to manage their data using the same semantic primitives they utilize for daily access. As with previous expeditions into new computing paradigms, it is in order to gain experience that we are building and utilizing a system representing the vision. In this case, however, the researchers are not representative of the user population. Most users will be non-technical people who just want to use the system but must (grudgingly) deal with administration tasks or live with the consequences. Thus, organized user studies will be required as complements to systems experimentation.

Perspective's design is motivated by a contextual analysis and early deployment experiences [3]. Our interactions with users have made clear the need for decentralization, selective replication, and support for device mobility and dynamic membership. An intriguing lesson is that home users rarely organize and access their data via traditional hierarchical naming—usually they do so based on data attributes. Computing researchers have long talked about attribute-based data navigation (e.g., semantic file systems [1]), while continuing to use directory hierarchies. However, users of home and personal storage live it. Popular interfaces (e.g., iTunes, iPhoto, and even drop-down lists of recently opened Word documents) allow users to navigate file collections via attributes such as publisher-provided metadata, extracted keywords, and date/time. Usually, files are still stored in underlying hierarchical file systems, but users often are insulated from naming at that level and are oblivious to where in the namespace given files end up.

Users have readily adopted these higher-level navigation interfaces, leading to a proliferation of semantic data location tools. In contrast, the abstractions provided by file systems for managing files have remained tightly tied to hierarchical namespaces. For example, most tools require that specific subtrees be identified, by name or by "volumes" containing them, in order to perform *replica management tasks*, such as partitioning data across computers for capacity management or specifying that multiple copies of certain data be kept for reliability. Since home users double as their own system administrators, this disconnect between interface styles (semantic for data access activities and hierarchical for management tasks) naturally creates difficulties.

The Perspective distributed file system allows a collection of devices to share storage without requiring a central server. Each device holds a subset of the data and can access data stored on any other (currently connected) device. However, Perspective does not restrict the subset stored on each device to traditional volumes or subtrees. To correct the disconnect between semantic data access and hierarchical replica management, Perspective replaces the traditional volume abstraction with a new primitive we call a view. A *view* is a compact description of a set of files, expressed much like a search query, and a device on which that data should be stored. For example, one view might be *"all files with type=music and artist=Beatles stored on Liz's iPod"* and another *"all files with owner=Liz stored on Liz's laptop."* Each device participating in Perspective maintains and publishes one or more views to describe the files it stores. Perspective ensures that any file that matches a view will eventually be stored on the device named in the view.

Since views describe sets of files using the same attribute-based style as users' other tools, view-based management is easier than hierarchical file management. A user can see what is stored where, in a human-readable fashion, by examining the set of views in the system. She can control replication and data placement by changing the views of one or more devices. Views allow sets of files to overlap and to be described independently of namespace structure, removing the need for users to worry about application-internal file naming decisions or difficult volume boundaries. Semantic management can also be useful for local management tasks, such as setting file attributes and security, as well as for replica management. In addition to anecdotal experiences, an extensive lab study confirms that view-based management is easier for users than volume-based management [4].

Our Perspective prototype is a user-level file system which runs on Linux and OS X. In our deployments, Perspective provides normal file storage as well as being the backing store for iTunes and MythTV in one household and in our research environment lounge.

Storage for the Home

The home is different from an enterprise. Most notably, there are no sysadmins—household members generally deal with administration (or don't) themselves. The users also interact with their home storage differently, since most of it is for convenience and enjoyment rather than employment. However, much of the data stored in home systems, such as family photos, is both important and irreplaceable, so home storage systems must provide high levels of reliability in spite of lax management practices. Not surprisingly, we believe that home storage's unique requirements would be best served by a design different from enterprise storage. This section outlines insights gained from studying use of storage in real homes and design features suggested by them.

WHAT USERS WANT

A contextual analysis is an HCI research technique that provides a wealth of *in situ* data, perspectives, and real-world anecdotes on the use of technology. It consists of interviews conducted in the context of the environment under study. To better understand home storage, we extensively interviewed all members of eight households (24 people total) in their homes and with all of their storage devices present. We have also gathered experiences from early deployments in real homes. This section lists some guiding insights (with more detailed information available in technical reports [3]).

Decentralized and dynamic: The users in our study employed a wide variety of computers and devices. While it was not uncommon for them to have a set of primary devices at any given point in time, the set changed rapidly, the boundaries between the devices were porous, and different data was "homed" on different devices with no central server. One household had set up a home server, at one point, but did not re-establish it when they upgraded the machine due to setup complexity.

Money matters: While the cost of storage continues to decrease, our interviews showed that cost remains a critical concern for home users (note that our studies were conducted well before the fall 2008 economic crisis). While the same is true of enterprises, home storage rarely has a clear "return on investment," and the cost is instead balanced against other needs (e.g., new shoes for the kids) or other forms of enjoyment. Thus, users replicate selectively, and many adopted cumbersome data management strategies to save money.

Semantic naming: Most users navigated their data via attribute-based naming schemes provided by applications such as iPhoto, iTunes, and the like. Of course, these applications stored the content in files in the underlying hierarchical file system, but users rarely knew where. This disconnect created problems when they needed to make manual copies or configure backup/synchronization tools.

Need to feel in control: Many approaches to manageability in the home tout automation as the answer. While automation is needed, the users expressed a need to understand and sometimes control the decisions being made. For example, only 2 of the 14 users who backed up data used backup tools. The most commonly cited reason was that they did not understand what the tool was doing and, thus, found it more difficult to use the tool than to do the task by hand.

Infrequent, explicit data placement: Only 2 of 24 users had devices on which they regularly placed data in anticipation of needs in the near future. Instead, most users decided on a type of data that belonged on a device (e.g., "all my music" or "files for this semester") and rarely revisited these decisions—usually only when prompted by environmental changes. Many did regularly copy new files matching each device's data criteria onto it.

From the insights above, we extract guidance that has informed our design of Perspective.

Peer-to-peer architecture: While centralization can be appealing from a system simplicity standpoint and has been a key feature in many distributed file systems, it seems to be a non-starter with home users. Not only do many users struggle with the concept of managing a central server, many will be unwilling to invest the money necessary to build a server with sufficient capacity and reliability. We believe that a decentralized, peer-to-peer architecture more cleanly matches the realities we encountered in our contextual analysis.

Single class of replicas: Many previous systems have differentiated between two classes: permanent replicas stored on server devices and temporary replicas stored on client devices (e.g., to provide mobility) [5, 2]. While this distinction can simplify system design, it introduces extra complexity for users and prevents users from utilizing the capacity on client devices for reliability, which can be important for cost-conscious home consumers. Having only a single replica class removes the client-server distinction from the user's perception and allows all peers to contribute capacity to reliability.

Semantic naming for management: Using the same type of naming for both data access and management should be much easier for users who serve as their own administrators. Since home storage users have chosen semantic interfaces for data navigation, replica management tools should be adapted accordingly—users should be able to specify replica management policies applied to sets of files identified by semantic naming.

In theory, applications could limit the mismatch by aligning the underlying hierarchy to the application representation, but this alternative seems untenable in practice. It would limit the number of attributes that could be handled, lock the data into a representation for a particular application, and force the user to sort data in the way the application desires. Worse, for data shared across applications, vendors would have to agree on a common underlying namespace organization.

Rule-based data placement: Users want to be able to specify file types (e.g., "Jerry's music files") that should be stored on particular devices. The system should allow such rules to be expressed by users and enforced by the system as new files are created. In addition to helping users to get the right data onto the right devices, such support will help users to express specific replication rules at the right granularity to balance their reliability and cost goals.

Transparent automation: Automation can simplify storage management, but many home users (like enterprise sysadmins) insist on understanding and being able to affect the decisions made. By having automation tools use the same flexible semantic naming schemes as users do normally, it should be possible to create interfaces that express human-readable policy descriptions and allow users to understand automated decisions.

Perspective Architecture

Perspective is a distributed file system designed for home users. It is decentralized, enables any device to store and access any data, and allows decisions about what is stored where to be expressed or viewed semantically.

Perspective provides flexible and comprehensible file organization through the use of *views*. A view is a concise description of the data stored on a given device. Each view describes a particular set of data, defined by a semantic query, and a device on which the data is stored. A view-based replica management system guarantees that any object that matches the view query will eventually be stored on the device named in the view.

We envision views serving as the connection between management tools and the storage infrastructure. Users can set policies through management tools, such as the one described in Figure 1, from any device in the system at any time. Tools implement these changes by manipulating views, and the underlying infrastructure (Perspective) in turn enforces those policies by keeping files in sync among the devices according to the views. Views provide a clear division point between tools that allow users to manage data replicas and the underlying file system that implements the policies.

A primary contribution of Perspective is the use of semantic queries to *manage the replication of data*. Specifically, it allows the system to provide accessibility and reliability guarantees over semantic, partially replicated data. This builds on previous semantic systems that used queries to *locate* data and hierarchies to manage data.

View-based management enables the design points outlined above. Views provide a primitive allowing users to specify meaningful rule-based placement policies. Because views are semantic, they unify the naming used for data access and data management. Views are also defined in a human-understandable fashion, providing a basis for transparent automation. Perspective provides data reliability using views without restricting their flexibility, allowing it to use a single replica class.

The Perspective prototype is implemented in C++ and runs at user-level using FUSE to connect with the system. It currently runs on both Linux and Macintosh OS X. Perspective stores file data in files in a repository on the machine's local file system and metadata in a SQLite database with an XML wrapper.

A user study evaluation using this prototype shows that, by supporting semantic management, Perspective can simplify important management tasks for end users. View-based management allowed up to six times as many users to complete management tasks correctly than traditional hierarchical systems did [4].

PLACING FILE REPLICAS

In Perspective, the views control the distribution of data among the devices in the system. When a file is created or updated, Perspective checks the attributes of the file against the current list of views in the system and sends an update message to each device with a view that contains that file. Each device can then independently pull a copy of the update.

When a device, A, receives an update message from another device, B, it checks that the updated file does, indeed, match one or more views that A has registered. If the file does match, then A applies the update from B. If there is no match, which can occur if the attributes of a file are updated such that it is no longer covered by a view, then A ensures that there is no replica of the file stored locally.

This simple protocol automatically places new files, and also keeps current files up to date according to the current views in the system. Perspective's protocols ensure that this property holds in the face of disconnection, device addition, and device failure, without requiring any centralized control. Perspective's protocols also ensure that files and updates are never lost due to view changes [4].

Each device is represented by a file in the file system that describes the device and its characteristics. Views themselves are also represented by files. Each device registers a view for all device and view files to ensure they are replicated on all participating devices. This allows applications to manage views through the standard file system interfaces, even if not all devices are currently present.

VIEW-BASED DATA MANAGEMENT

In this subsection, we present three scenarios to illustrate view-based management. Each scenario assumes an interface that allows users to manipulate views. While we envision systems containing a number of tools and interfaces, Figure 1 shows the interface we currently provide Perspective users.

FIGURE 1: A SCREEN SHOT OF THE VIEW MANAGER GUI. ON THE LEFT ARE FILES, GROUPED USING FACETED METADATA. ACROSS THE TOP ARE DEVICES. EACH SQUARE SHOWS WHETHER THE FILES IN THE ROW ARE STORED ON THE DEVICE IN THE COLUMN.

Traveling: Harry is visiting Sally at her house and would like to play a new U2 album for her. Before leaving, he checks the views defined on his wireless music player and notices that the songs are not stored on the device, although he can play them from his laptop, where they are currently stored. He asks the music player to pull a copy of all U2 songs, which the player does by creating a new view for this data. When the synchronization is complete, the file system marks the view as complete, and the music player informs Harry.

He takes the music player over to Sally's house. Because the views on his music player are defined only for his household, and the views on Sally's devices for her household, no files are synchronized. But queries for "all music" initiated from Sally's digital stereo can see the music files on Harry's music player, so while he is visiting they can listen to the new U2 album from Harry's music player on Sally's nice stereo speakers.

Crash: Mike's young nephew Oliver accidentally pushes the family desktop off the desk onto the floor and breaks it. Mike and his wife Carol have each configured the system to store their files both on their respective laptops and on the desktop, so their data is safe. When they set up the replacement computer, a setup tool pulls the device objects and views from other household devices. The setup tool gives them the option to replace an old device with this computer, and they choose the old desktop from the list of devices. The tool then creates views on the device that match the views on the old desktop and deletes the device object for the old computer. The data from Mike and Carol's laptops is transferred to the new desktop in the background over the weekend.

Short on space: Marge is trying to finish a project for work on her home laptop. While she is working, a capacity automation tool on her laptop alerts her that the laptop is short on space. It recommends that files created over two years ago be moved to the family desktop, which has spare space. Marge, who is busy with her project, decides to allow the capacity tool to make the change. She later decides to keep her older files on the external hard drive instead, and makes the change using a view-editing interface on the desktop.

Conclusion

Home users struggle with replica management tasks that are normally handled by professional administrators in other environments. Perspective provides distributed storage for the home with a new approach to data location management: the view. Views simplify replica management tasks for home storage users, allowing them to use the same attribute-based naming style for such tasks as for their regular data navigation.

ACKNOWLEDGMENTS

We thank Rob Reeder, Jay Melican, and Jay Hasbrouck for helping with the users studies. We also thank the members and companies of the PDL Consortium (including APC, Cisco, DataDomain, EMC, Facebook, Google, HP, Hitachi, IBM, Intel, LSI, Microsoft, NetApp, Oracle, Seagate, Sun, Symantec, and VMware) for their interest, insights, feedback, and support. This material is based on research sponsored in part by the National Science Foundation, via grants #CNS-0326453 and #CNS-0831407, and by the Army Research Office, under agreement number DAAD19-02-1-0389. Brandon Salmon is supported in part by an Intel Fellowship.

REFERENCES

[1] David K. Gifford, Pierre Jouvelot, Mark A. Sheldon, and James W. O'Toole Jr., "Semantic File Systems," *Operating Systems Review* 25(5): 16–25, 1991: http://reference.kfupm.edu.sa/content/s/e/semantic_file_systems__50646.pdf.

[2] Daniel Peek and Jason Flinn, "EnsemBlue: Integrating Distributed Storage and Consumer Electronics," *Symposium on Operating Systems Design and*

Implementation (OSDI), USENIX Association, 2006: http://www.cs.ubc.ca/labs/dsg/Sem_Winter_2007/ensemblue.pdf.

[3] Brandon Salmon, Frank Hady, and Jay Melican, "Learning to Share: A Study of Sharing Among Home Storage Devices," Technical Report CMU-PDL-07-107, Carnegie Mellon University, October 2007: http://www.pdl.cmu.edu/PDL-FTP/Storage/CMU-PDL-07-107.pdf.

[4] Brandon Salmon, Steven W. Schlosser, Lorrie Faith Cranor, and Gregory R. Ganger, "Perspective: Semantic Data Management for the Home," *Proceedings of the 7th USENIX Conference on File and Storage Technologies (FAST '09)*, USENIX Association, 2009: http://www.usenix.org/events/fast09/tech/full_papers/salmon/salmon.pdf.

[5] M. Satyanarayanan, "The Evolution of Coda," *ACM Transactions on Computer Systems*, 20(2): 85–124, May 2002: http://www.cs.cmu.edu/~satya/docdir/p85-satyanarayanan.pdf.