Towards Automated Personalized Data Storage

John Lange #1, Alexandros Labrinidis #2, Panos K. Chrysanthis #3

# *Department of Computer Science, University of Pittsburgh*
Pittsburgh, PA 15260, USA
1 jacklange@cs.pitt.edu
2 labrinid@cs.pitt.edu
3 panos@cs.pitt.edu

***Abstract*—User data is growing at an ever greater pace that threatens to overwhelm our ability to effectively manage it. As the types of data increase, and the storage environments become ever more heterogeneous, even reasoning about basic data management decisions becomes increasingly difficult. This expansion in complexity requires new methodologies for managing data that alleviate as much of the burden as possible from the individual user. Instead of requiring users to understand their full collection of data and the underlying storage architectures, future storage systems need to be able to decide on their own how to manage individual files both in terms of the appropriate storage medium as well as the necessary file operation semantics. In this paper we present a vision for future storage systems that address the dramatic increase in complexity and volume by providing autonomic storage management decisions based on dynamically collected metrics that measure the relationship between individual users and each of their personal files.**

I. Introduction

Managing large collections of data has become a serious challenge for modern computing platforms and users. While the challenges posed by managing this abundance of data are widely recognized and acknowledged by enterprise and scientific users [7], [14], [8], it is no less of an issue for ordinary consumers whose personal data collections have increased significantly as well. Numerous new terms have been coined to describe this issue in various environments such as the Data Deluge [7] and Big Data [19]. Unfortunately, while much work has been put into developing systems to cope with this massive volume of data, it has almost entirely focused on enterprise and scientific/high performance computing (HPC) environments [4], [33], [27], [15]. For ordinary users the task of managing their own personal data has not significantly changed even as its volume has become unmanageable. Furthermore, instead of addressing this problem directly, modern consumer platforms and interfaces (including smartphones and tablets) are taking the opposite approach and removing direct data management capabilities from consumer systems. Most consumer oriented solutions have tended to focus on the accessibility aspect of the data in question [32], [30], [6], [18]. While accessibilty is an important feature needed by users, it does not address the problem of how to actually manage the underlying data storage.

Along with data access, users are increasingly faced with a huge array of options in how to use and manage their personal data collections. Outside of social data services [5], more traditional services such as backup [31] and remote accessibility [13] are becoming available. Local file systems are also offering increased features such as change logs that track a file's history over its entire lifetime. While these new services greatly increase the number of options a user has to secure their data, they still either rely on the user to manage the data themselves (i.e. choosing which files to backup) or implement a global policy for every file in the system. Both of these approaches are sub-optimal in that they either place an increasingly complex onus on the user or waste resources to secure unimportant files and data.

We claim that as the amount of data increases along with the applications that use that data, users will become unable to effectively manage their own personal data collections.

Furthermore, it is becoming clear that most consumer oriented users are not interested in performing the management tasks explicitly, as the success of devices and interfaces such as tablets, phones, and Windows 8 demonstrates. In order to save users from a data deluge it will be necessary for emerging systems and platforms to take on a more active role in the management of the data they store, instead of simply hiding it from the users. Future systems will need to have the capability of automating the management of data and available storage devices and services in order to fully take advantage of the increasing variety of storage options while not overwhelming users with management requirements. In effect we are seeking to provide automated storage management across the entire range of storage options (such as local disks, cloud storage services, etc), similar to but much broader than existing solutions such as Apple's Fusion Drive [9]. Our vision of such systems is based around the idea of measuring the importance that each piece of data has to a specific user, and using that information to map files to a disparate set of storage devices and services which are presented as a unified file system to the user. This approach will allow a user to easily incorporate the large range of possible storage options into a single tractable and accessible interface.

Our high level vision is an autonomic storage system that "understands" both the implicit real world value each piece of data has to a user as well as which storage services are best suited for each data item.

While cloud services might appear to offer a solution to the problem we have so far described, there are still a significant number of issues that make fully migrating to the cloud unlikely to occur [23]. First, despite claims to the contrary, cloud services are not yet fully reliable in both technical and business terms. A number of high profile failures have occurred at all levels of cloud services, including technical failures resulting in service outages [1], political interference resulting in service cancellation [3], security failures both internal [23] and external [2], as well as direct subversion by foreign and domestic state actors. As such we claim that while cloud storage services will be a significant component of future storage systems, they will never be able to fully take over the role of sole storage provider.

As the storage environment becomes more complex with cloud solutions as well as advanced and diversified local storage systems (both software and hardware), we posit that it is necessary to make the overall storage system responsive to the differentiated needs of users as well as the users' data. Past work has shown that there is a large degree of variability in the expectations held by users about how a system operates [12], [25], [26], and our own early results indicate that this variability holds among the files of each user.

In this paper we present results from a user study we conducted to measure the degree of familiarity that users have with their personal data. We outline this study and discuss its results in Section II. Overall, our study shows that users are unable to recognize the majority of files located in their own home directories. We claim that, based on these observations, a storage model that embraces this lack of familiarity will be necessary as the amount of personal user data continues to grow. We outline our vision of such a storage system in Section III, and discuss the challenges that must be addressed to achieve its implemenation.

II. Measuring Data Familiarity

To quantify the degree of difficulty faced by users when deciding how to manage their personal data we have conducted a survey-based user study to directly measure the degree of familiarity between users and their own personal data. To conduct this survey we implemented a small application that volunteers downloaded and ran on their personal computing environments that was designed to directly measure the degree to which each user understood the organization of their own personal files. To perform this measurement the survey presented each user with a series of files (including the absolute path) that were randomly selected from the users' home directory and/or other directories used to store personal data. For each file the participant was asked a series of questions to gauge both whether the user recognized the file in question as well as the level of importance the user assigned to the data contained in the file.

The goal of the study was to measure the variance in relevance and importance inside a user's personal data collection. In particular we sought to determine whether (1) a user's personal data collection could be managed uniformly as a whole or whether management decisions were necessary at per-file granularity, and (2) whether there was any variance among users in their understanding of their personal data files.

Recognizability of 1258 files among 15 users	
Not Recognized	50%
Recognized Parent Directory	37%
Recognized	13%

Fig. 1. Recognition of a random subset of files in users' home directories

The study participants consisted of 15 volunteers collected from inside our department consisting of graduate students and faculty members. The survey lasted at most 20 minutes and collected anywhere between 50 to 150 answers per user, with the survey ending after either 20 minutes had passed or 150 files had been classified. In total the survey collected results for 1258 different filenames for an average of 84 files per volunteer. The survey covered a combination of work and personal home computing systems including laptops and desktops and spanned across Linux (3 users), Windows (8 users), and MacOS (4 users) environments.

For the duration of the survey each user was repeatedly asked to identify a random file selected from their personal data directories. The user was first asked to identify the file and respond whether the user (1) recognized the file, (2) did not recognize it, or (3) did not recognize the file itself, but did recognize the file's parent directory. In addition the user was asked to rate the importance of the data included in the file on a scale of 1 to 10. Finally, the user was asked to provide additional classifications of services desired for each file (whether it should be backed up, encrypted, always available, etc). The full results of the survey were then uploaded to a MySQL database for offline analysis. To ensure anonymity, both file and directory names were randomized by hashing the name with a single salt value, generated randomly at the start of the survey. This ensured that neither filenames nor paths could be identified while retaining the hierarchical structure of the file organization. The full set of results were displayed to the participants at the end of the survey, at which time they were given the option to either upload them to the database or not.

A. Survey Results

The results of our survey are shown in Figures 1 and 2. As can be seen in Figure 1 of all the files scanned as part of the survey, very few (only 13%) were recognized directly by the participants. Furthermore, almost exactly half of the files were completely unrecognized, with the remainder (37%) only being recognized based on their parent directory. These results demonstrate the significant challenge faced by any system that relies on users to explicitly manage their own data, since most users are unable to even identify and recognize the majority of data that belongs to them. It is important to note again that this survey was conducted amongst graduate students and faculty inside the *Computer Science Department* at the University of Pittsburgh, so it is easily conceivable that a broader user base would report results that would in fact be significantly worse. One likely explanation for the degree of unfamiliarity with data files is the preponderance of state and configuration files

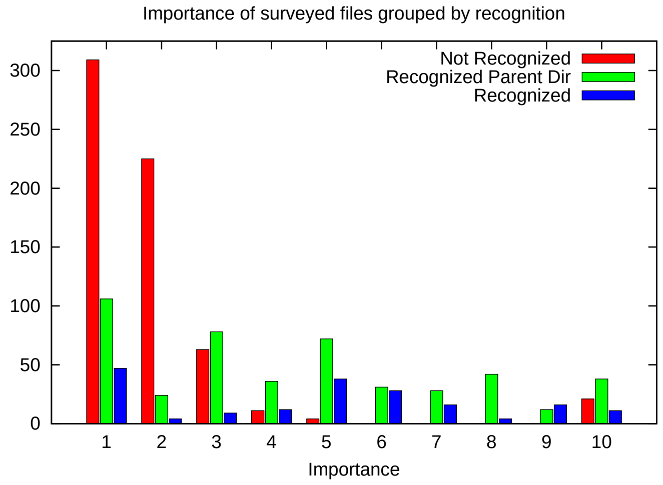

Fig. 2. The variability in importance of a random subset of files in users' home directories

that are stored on behalf of applications invisibly to users. An obvious example of this would be browser profiles that include bookmarks and caches that are stored as files in special application directories alongside other user data. While these files are by design hidden from the user, it is important to note that does not make them fully irrelevant nor does it mean that those files can simply be binned into a single classification. Rather it marks a disconnect between the data contents of a file and the filename of the file. This disconnect is further seen in the importance ratings each user assigned to particular files.

Figure 2 shows the results of the collected importance levels for each file included in the survey. The files are grouped by the recognizability classification assigned by the user. Unsurprisingly, the unimportant files are dominated by those that the users were completely unfamiliar with. Users would be unlikely to care if they lost a file which they never even knew existed. However, it should be noted that the importance of recognized files does not exhibit any clustering. In fact the number of recognized files with little to no importance (0) is greater than the number of recognized files rated between somewhat (7) to very (10) important. More surprisingly, of the files that users rated as very important (10), a greater number of them were in fact not recognized than were recognized. This would seem to indicate that even though users had no idea what data a file contained, they had some other implicit cognition that it was important in some way.

The results of our user survey indicate several important take away points. First, it is unworkable to require that users take on the responsibility of directly managing their personal data files, since users are completely unaware of the contents of over half of their personal data collection. Nor can data management services operate at only the user granularity and treat the full collection of a user's files as a uniformly important set of data or even as a single high level unit. Furthermore, there is an increasing trend for applications to implicitly manage their own internal data on behalf of the user, and even to hide the data itself from the user. This severely limits the users' control over the data storage system, and prevents the storage system from optimizing the storage resources based on the actual needs of the user instead of the application. In order to optimize the usage of storage systems and provide cost effective storage solutions for the user, the organization of the storage system must be based on per-file decisions, and reflect the relative importance that each file has to the user.

III. Enabling Automated Data Storage

Based on our survey results we claim that future data and file management systems must be capable of providing per file management that is responsive to actual user requirements in order to effectively utilize the proliferation of different storage platforms and services (such as solid-state drives, phase change memory, cloud storage services, and other emerging technologies). We envision that future computing platforms will need to provide a single unified storage management service that is capable of managing a user's personal data collection autonomously based on inputs it collects either explicitly or implicitly from the user. The goal of this system would be to develop a user profile that can be queried in order to dynamically optimize the organization of the underlying storage systems and user data in such a way that the user requirements are met in an optimal way, even if the user is not explicitly aware of what their requirements are. Our vision of such a system, which we call *MyFS*, is shown in Figure 3; we describe it next.

A. *MyFS*

Our hypothetical MyFS system would require two central components: a file profiler, which will determine importance characteristics on a per-file basis, and a file system dispatcher, which will manage the underlying storage systems and route I/O requests amongst them based on input from the profiler. The solid lines indicate data transfers between the different components of the storage system, the routing of which is controlled by the MyFS file system. The dashed lines indicate high level inputs to the file profiler, which can include explicit inputs set by the user, inferred values gathered from the user interface, or actual I/O traces collected by the file system. These inputs are then combined and used to generate an importance value for each file that the user interacts with. These importance values are fed back into the MyFS file system (the dotted line) where they are used to determine how each file is handled, which storage targets it is routed to, and the degree of replication.

Our system model assumes a heterogeneous environment that supports multiple storage systems with differing behavioral semantics. We also assume that MyFS will operate on single user devices such as desktops, laptops, and netbooks, as well as emerging devices such as smartphones and tablets. The MyFS model assumes the ability to separate files, file systems and the underlying storage mediums from each other.

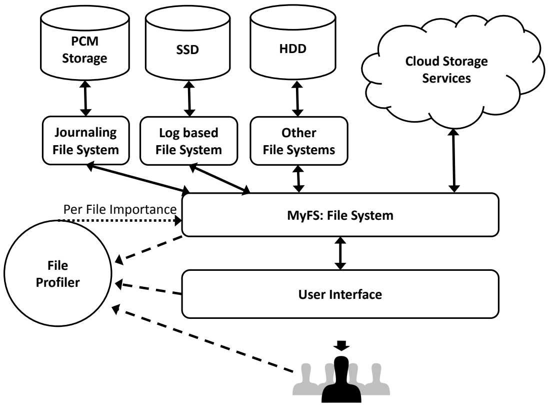

Fig. 3. MyFS System Diagram. Block IO is managed by the MyFS layer that determines storage location as well as replication based on per-file importance information collected implicitly and explicitly from the user.

That is, particular file system features will be capable of being dynamically selected based on the needs of a particular file and used on a specific storage device. Features such as journalling and logging can be enabled for particular files based on the consistency requirements of that file alone. This will allow different files to be accessed in a manner that is most suited to the file's contents and access patterns. For instance, when MyFS detects a file with a high importance value that is being heavily edited, it would be capable of enabling logging behavior on the file in order to optimize performance as well as maintain an update history to provide on the fly versioning. MyFS will also be capable of directing storage I/O to the appropriate hardware medium, based on the hardware's performance and reliability.

We also envision that MyFS will be capable of dynamically generating and placing replicas to provide the appropriate level of reliability for each particular file. MyFS would also need to integrate with cloud based storage systems that provide remote data services for storing and sharing a user's files. MyFS would treat these cloud services the same as local storage devices and differentiate among them only by their performance, availability, resiliency, cost models, and service pricing structure. We envision that the predominant use of these cloud services would be for replica storage, however there is no inherent restriction that would prevent them from being used as a primary storage medium.

Because the goal of our proposed system is to decouple a user's files from the devices/services those files are stored on, it would need to be able to use each device/service in multiple roles. MyFS should not mandate usage models for each component added to the system. For instance, such a system should not require transparent caching, wherein local storage devices are used simply as local caches for cloud based data. Instead, each device should be considered a primary component of the system that can be used for multiple purposes. While a local storage device could be used as a transparent cache, it could also simultaneously be used as the primary storage location for a newly created file that the user is currently editing, or as replica storage for a file stored on another local device. These details would be completely hidden from the user, who would only be presented with a global view of their personal file system that allows every file to be accessed in the same way.

Fundamentally the MyFS system would expose a small set of common operations that are in turn layered on top of underlying file system or cloud storage service interfaces. The focus of our proposed system would be the automated placement of files and data onto the appropriate resources that provide underlying services. Because MyFS would not be responsible for implementing the underlying storage architectures it would only be required to perform high level operations to control data movement: (a) copy to/from, (b) move to/from, and (c) delete. MyFS would automatically monitor the collection of storage resources and dynamically determine the files and targets for data placement. We envision the system continuously running in the background and adapting the data layout and organization based on dynamic measurements collected from both the storage resources themselves as well as from the set of profiles maintained by MyFS. In other words, the data management tasks of MyFS would be able to execute synchronously in reaction to actual file operations executed by the user or the user's applications, while also executing asynchronous operations, carried out in the background.

B. Research Challenges and First Steps

Past work has shown that it is entirely possible to design systems that respond directly to measured or inferred values of satisfaction, comfort, and irritation [16], [29], [10], [25], [24]. In the past we have applied this technique to remote display clients [21] and home broadband networks [25], [22]. In this same vein we posit that approaches similar to those used to infer a user's disposition can be used to measure the importance of the user's data. We envision incorporating these and other measurement techniques in order to develop both user and per-file profiles that can be used to generate classifications for data that in turn drive data organization decisions in the storage system. Fully realizing our envisioned system poses a number of significant research challenges that we now examine further.

Intelligent user profiling At the heart of our vision is the capability to monitor user and system activity in order to infer the relationship between users and their data. Building a completely accurate profile would certainly be an impossible task, so instead our system would be forced to rely on heuristic approximations and inferences generated from incomplete measurements. In order to converge as closely as possible to an accurate profile representation our system would require the implementation of an extensive monitoring framework throughout the storage system as well as the user interface.

Although there has been a lot of work on user profiling with regards to data access patterns, this has primarily been limited to the mobile data management domain, with the ultimate goal of supporting prefetching and other techniques to make data available anywhere, anytime [11], [17]. MyFS aims to manage data across different storage options and considers performance, reliability, availability, and cost as part of the optimization challenge.

In general, measuring user satisfaction and irritation has seen a significant amount of interest in both the systems and HCI communities. We believe that such frameworks could be extended and utilized in order to provide measurements applicable to our proposed storage framework. In particular many user monitoring systems gather data points by intentionally configuring the system to begin irritating the user, to detect an explicit user response in relation to a given system configuration. Once irritation is detected the configuration is quickly revoked, but not before a snapshot is taken and catalogued as a data point in the configuration space. By continuously perturbing the system a user profile is dynamically generated that allows the classification of states based on the level of irritation they cause the user. From this the system can infer both good and bad state configurations based on past experience. We believe that such an approach could be utilized to help classify data requirements in a similar manner. For example file accesses could be artificially slowed down while monitoring user behavior with the intent of detecting which files cause the most discomfort in the user. These measurements could be used to help classify files whose availability is directly correlated to user satisfaction.

Explicit user feedback Although we primarily want MyFS user profiling activities to be as transparent to the end user as possible, there are cases where it is beneficial for the user to provide explicit requirements for his/her files. An example is that of photo files, which typically are not accessed frequently (preventing implicit measurements of access behaviors), but are usually very important to the user and need to be backed up. The challenge here is two-fold: (a) how to identify the cases where it is best for the system to ask the user explicitly, and (b) how should the user specify these requirements for his/her per-file storage needs, especially when multiple dimensions are considered (performance vs reliability vs availability vs cost). One suggestion is to consider a variant of *Quality Contracts* [20], with step functions over the different dimensions used to express user satisfaction at different levels of service (over different dimensions of quality).

Efficient, holistic system profiling In addition to monitoring users, new mechanisms for maintaining persistent file usage measurements are needed in order to measure and classify file usage behavior. These measurements would feed into the central profile engine in order to provide per-file information and match actual usage patterns to the higher level user importance. This monitoring could further be used to detect unimportant files or files that would otherwise not require extensive storage features. Examples of such files include temporary files that are created and deleted quickly, application checkpoint files that maintain data persistence across application crashes, and other auxiliary files such as application caches or configuration settings that each place different requirements on the underlying storage system. We envision this file profile mechanism as collecting both file access behaviors as well as additional metadata associated with the files' contents. This metadata would be associated with each file and would be maintained at dynamic granularities and persistence depending on file types and access behaviors. Such metadata represents not only the access history of a given file, but also such things as content hashes that can be used to detect built in redundancy inside the files themselves. As an example, such measurements would be useful to detect backup files generated either explicitly by the user or automatically by applications. The metadata described above would need to be stored persistently in MyFS along with the actual user data.

SLA compliance and failure models One added benefit of performing holistic system profiling is having detailed information about the performance of the different storage alternatives. Such information can be further utilized in two ways: (a) if the storage provider is expected to adhere to a specific Service Level Agreement (SLA), the collected information can identify whether this is true or not, and warn the user if not, and (b) historical and real-time performance information for a particular storage option can be fitted against appropriate failure models and used to detect whether a failure is imminent (e.g., of a hard disk); MyFS in that case can take proactive measures.

Signals identification (implicit user-profiles) The identification of relevant signals from the breadth of measurements that could be collected is another challenge that must be addressed by our proposed storage system. A significant research problem exists in determining which measurement signals are necessary and useful for delivery to the policy engine. That is what behaviors and other observations are actually correlated to the importance and relevance for a particular file as well as the actual requirements needed from the storage system. Initial steps in such an effort would have to focus on not only developing new instrumentation methods and inputs, but also assessing inputs and signals that already exist and are present in modern system architectures.

Network effects One final possible research direction is the exploitation of the effects of networking in order to better utilize user feedback (implicit or explicit) and make user profiling more efficient. In particular, we plan to consider two types. First of all, the networking of different MyFS installations – this would essentially allow for "collaboration" among different users and utilize patterns that are common. There is plethora of techniques to correlate profiles of one user to those of the community (e.g., collaborative filtering [28]), although we need to strike a balance between what information from one user's MyFS system is shareable with other MyFS systems. Secondly, the networking of MyFS with other user programs or interfaces. This would be facilitated with the establishment of an API to describe preferences (that the users have expressed or an application was able to detect) and sharing of user profiles.

IV. Conclusion

Personalized data management is quickly becoming an impossible task for many modern users. As the amount of personal data stored on local and remote computing platforms continues to proliferate, system architectures and user interfaces are failing to provide the capabilities necessary to effectively manage these personal data collections. As our study has shown, modern data computer systems often contain a large amount of data which users are completely oblivious too, even when that data is stored among their own personal files. Our results indicate that users are unable to recognize over half of the files stored in their home directories. While data management solutions are being actively and aggressively explored for enterprise, high performance, and other large scale entities, per-user management capabilities are either languishing or being actively stripped down even further in newer consumer environments. We have proposed a novel system architecture to address this problem through the introduction of an autonomic storage layer, that directly manages a user's files based on the inferred importance of each file to a particular user. The proposed architecture opens up a host of interesting research challenges which we plan to address in the future.

Acknowledgements:

We would like to thank the reviewers for their helpful comments. This work was partially supported by startup funds provided by the University of Pittsburgh, as well as NSF awards: IIS-0746696 and OIA-1028162.

References

[1] Amazon ec2 outage hobbles websites. http://www.informationweek.com/news/cloud-computing/infrastructure/229402054.

[2] Update on playstation network/qriocity services. http://blog.us.playstation.com/2011/04/22/update-on-playstation-network-qriocity-services/.

[3] Why did amazon web services shut down wikileaks? http://www.informationweek.com/news/cloud-computing/software/228500230.

[4] Mona Ahuja, Cheng Che Chen, Ravi Gottapu, Jörg Hallmann, Waqar Hasan, Richard Johnson, Maciek Kozyrczak, Ramesh Pabbati, Neeta Pandit, Sreenivasulu Pokuri, and Krishna Uppala. Peta-scale data warehousing at yahoo! In *Proc. of ACM SIGMOD Conference*, 2009.

[5] Sihem Amer-Yahia, Jian Huang, and Cong Yu. Building community-centric information exploration applications on social content sites. In *Proc. of ACM SIGMOD Conference*, 2009.

[6] Anne Aula, Rehan M. Khan, and Zhiwei Guan. How does search behavior change as search becomes more difficult? In *Proc. of the 28th international conference on Human factors in computing systems (CHI)*, 2010.

[7] G. Bell, T. Hey, and A. Szalay. Computer science: Beyond the data deluge. *Science*, 323:1297–1298, 2009.

[8] Scott Carlson. Lost in a sea of science data. *Chronicle of Higher Education 52(42)*, page A35.

[9] Feng Chen, David A. Koufaty, and Xiaodong Zhang. Hystor: Making the Best Use of Solid State Drives In High Performance Storage Systems. In *Proc. of the 25th International Conference on Supercomputing (ICS)*, pages 22–32, 2011.

[10] Kuan-Ta Chen, Cheng-Chu Tu, and Wei-Cheng Xiao. Oneclick: A framework for measuring network quality of experience. In *Proc. of the 28th IEEE Conf. on Computer Communications, (INFOCOM)*, 2009.

[11] Mitch Cherniack, Michael J. Franklin, and Stanley B. Zdonik. Expressing user profiles for data recharging. *IEEE Personal Commun.*, 8(4):32–38, 2001.

[12] Peter Dinda, Gokhan Memik, Robert Dick, Bin Lin, Arindam Mallik, Ashish Gupta, and Samuel Rossoff. The user in experimental computer systems research. In *Proc. of the Workshop on Experimental Computer Science (ExpCS)*, 2007.

[13] Dropbox. http://www.dropbox.com.

[14] Ian Gorton, Paul Greenfield, Alex Szalay, and Roy Williams. Data-intensive computing in the 21st century. *Computer*, 41(4):30 –32, 2008.

[15] Naga Govindaraju, Jim Gray, Ritesh Kumar, and Dinesh Manocha. Gputerasort: high performance graphics co-processor sorting for large database management. In *Proc. of ACM SIGMOD Conference*, 2006.

[16] Ashish Gupta, Bin Lin, and Peter Dinda. Measuring and understanding user comfort with resource borrowing. In *Proc. of the 13th IEEE International Symposium on High Performance Distributed Computing (HPDC)*, 2004.

[17] Abdelsalam Helal and Joachim Hammer. Ubidata: Requirements and architecture for ubiquitous data access. *SIGMOD Rec.*, 33(4):71–76, December 2004.

[18] Stratos Idreos and Erietta Liarou. dbTouch: Analytics at your Fingertips. In *Proc. of the 6th Biennial Conference on Innovative Data Systems Research (CIDR)*, 2013.

[19] H. V. Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M. Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Communications of the ACM*, 2014.

[20] Alexandros Labrinidis, Huiming Qu, and Jie Xu. Quality contracts for real-time enterprises. In *Lecture Notes in Computer Science 4365: Post Proceedings of First International Workshop on Business Intelligence for the Real Time Enterprise*, pages pp. 143–156. 2007.

[21] John R. Lange, Peter Dinda, and Sam Rossoff. Experiences with client-based speculative remote display. In *Proc. of the USENIX Annual Technical Conference, (USENIX 2008)*, June 2008.

[22] John R. Lange, J. Scott Miller, and Peter Dinda. EmNet: Satisfying the individual user through empathic home networks: Summary (poster). In *Proc. of ACM SIGMETRICS 2009, (SIGMETRICS 2009)*, June 2009.

[23] Prince Mahajan, Srinath Setty, Sangmin Lee, Allen Clement, Lorenzo Alvisi, Mike Dahlin, and Michael Walfish. Depot: Cloud storage with minimal trust. In *Proc. of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2010.

[24] Arindam Mallik, Jack Cosgrove, Robrt Dick, Gokhan Memik, and Peter Dinda. Picsel: Measuring user-perceived performance to control dynamic frequency scaling. In *Proc. of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2008.

[25] J. Scott Miller, John R. Lange, and Peter A. Dinda. Emnet: Satisfying the individual user through empathic home networks. In *Proc. of the 29th IEEE Conf. on Computer Communications, (INFOCOM)*, 2010.

[26] J. Scott Miller, Amit Mondal, R. Potharaju, P. Dinda, and A. Kuzmanovic. Understanding end-user perception of network problems. In *Proc. of the Workshop on Measurements Up the STack (W-MUST)*, 2011.

[27] Andrew Pavlo, Erik Paulson, Alexander Rasin, Daniel J. Abadi, David J. DeWitt, Samuel Madden, and Michael Stonebraker. A comparison of approaches to large-scale data analysis. In *Proc. of ACM SIGMOD Conference*, 2009.

[28] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. of International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM.

[29] Alex Shye, Berkin Ozisikyilmaz, Arindam Mallik, Gokhan Memik, Peter Dinda, Robert Dick, and Alok Choudhary. Learning and leveraging the relationship between architectural-level measurements and individual user satisfaction. In *Proc. of the 35th International Symposium on Computer Architecture (ISCA)*, 2008.

[30] Vineet Sinha and David R. Karger. Magnet: supporting navigation in semistructured data environments. In *Proc. of ACM SIGMOD Conference*, 2005.

[31] SugarSync. https://www.sugarsync.com.

[32] Jinbao Wang, Sai Wu, Hong Gao, Jianzhong Li, and Beng Chin Ooi. Indexing multi-dimensional data in a cloud system. In *Proc. of ACM SIGMOD Conference*, 2010.

[33] Fei Xu, Kevin Beyer, Vuk Ercegovac, Peter J. Haas, and Eugene J. Shekita. E = mc3: managing uncertain enterprise data in a cluster-computing environment. In *Proc. of ACM SIGMOD Conference*, 2009.